Ref. Ares(2025)11587597 - 24/12/2025

HUMAN-CENTRIC AI-ENABLED EXTENDED REALITY APPLICATIONS FOR THE INDUSTRY 5.0 ERA

D2.3 – REFERENCE ARCHITECTURE MODEL AND TECHNICAL SPECIFICATIONS V2

Lead Beneficiary	UBI
Work Package Ref.	WP2 – Specifications, Architecture and EU XR Platform Integration
Task Ref.	T2.2, T2.3 – Technical Specifications of XR5.0 Components and Technologies, Reference Architecture Model for XR5.0 Applications
Deliverable Title	D2.3 – Reference Architecture Model and Technical Specifications v2
Due Date	2025-12-31
Delivered Date	2025-12-24
Revision Number	3.0
Dissemination Level	Public (PU)
Type	Report (R)
Document Status	Release
Review Status	Internally Reviewed
Document Acceptance	WP Leader Accepted and/or Coordinator Accepted
EC Project Officer	Ms. Marta PALAU FRANCO

HORIZON-CL4-2023 Research and Innovation Action

This project has received funding from the European Union's Horizon Europe Research and Innovation Programme under grant agreement no 101135209.

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

DISCLAIMER

COPYRIGHT MESSAGE

CONTRIBUTING PARTNERS

Partner Acronym	Role[1]	Name Surname[2]
UBI	Lead Beneficiary	Nikolaos Tousert, Dimitra Lipitaki, Thanos Giannakopoulos, Xanthi Papageorgiou, Dimitris Ntalaperas
IML	Contributor	Jorge Oliveira, Tomaz Saraiva
GFT	Contributor	
KUKA	Contributor	
SSF	Contributor	
OCU	Contributor	
HOLO	Contributor	
SYN	Contributor	
LXS	Contributor	
UNP	Contributor	
INNOV	Contributor	
ATB	Contributor	
UPRC	Quality Assurance and Internal Reviewer	Thanos Kiourtis, Argyro Mavrogiorgou
IPIA	Contributor	
CYENS	Internal Reviewer	
SUPSI	Internal Reviewer	Elias Montini, Samuele Dell'Oca
ALMER	Contributor	
SIE	Contributor	
TOG	Contributor	

[1] Lead Beneficiary, Contributor, Internal Reviewer, Quality Assurance
[2] Can be left void

Revision History

Version	Date	Partner(s)	Description
0.1	2025-09-12	UBI	ToC Version
1.0	2025-11-17	UBI, IML	1st Version
1.1	2025-12-1	UBI, IML, GFT, OCU, HOLO, SYN, LXS, UNP, INNOV, ATB, UPRC, IPIA, CYENS, SUPSI, ALMER, SIE, TOG	Inputs and Additions
1.2	2025-12-5	UBI	Version for Peer Reviews
2.0	2025-12-22	UBI	Version for Quality Assurance
2.1	2025-12-23	UPRC	Technical Review, Quality Assurance
3.0	2025-12-24	UBI, UPRC, GFT	Version for Submission

LIST OF ABBREVIATIONS

Acronym	Definition
AAS	Asset Administration Shell
ACID	Atomicity, Consistency, Isolation, Durability
ADM	Architecture Development Framework
AI	Artificial Intelligence
AMR	Autonomous Mobile Robot
AMT	Active Management Technology
AOSP	Android Open-Source Project
API	Application Programming Interface
AR	Augmented Reality
AWS	Amazon Web Services
BLE	Bluetooth Low Energy
CAD	Computer-Aided Design
CFS	Customer Facing Service
CI/CD	Continuous Integration/Continuous Deployment
CRM	Customer Relationship Management
CRUD	Create, Read, Update, Delete
CSV	Comma-Separated Values
C4	Context, Containers, Components, Code
DB	Database
DNS	Domain Name System
DT	Digital Twin
DTLS	Datagram Transport Layer Security
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load

ETSI MEC	European Telecommunications Standards Institute - Multi-access Edge Computing
FCAPS	Fault, Configuration, Accounting, Performance, Security
GATT	Generic Attribute Profile
GDPR	General Data Protection Framework
gRPC	Google Remote Procedure Call
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HDM	Historical Data Manager
HDT	Human Digital Twin
HITL	Human in the Loop
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity and Access Management
IDE	Integrated Development Environment
IIFA	Industrial Internet Architecture Framework
IIC	Industrial Internet Consortium
IIRA	Industrial Internet of Things (IIoT)
IoT	Internet of Things
IPC	Inter-Process Communication
ISO/IEC JTC 1	International Organization for Standardization / International Electrotechnical Commission - Joint Technical Committee 1
IT	Information Technology
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
JWT	JSON Web Token
KPI	Key Performance Indicator
LLM	Large Language Model

MQTT	Message Queuing Telemetry Transport
MR	Mixed Reality
MRTK	Mixed Reality Toolkit
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
OAuth2	Open Authorization 2.0
ODBC	Open Database Connectivity
OPC-UA	Open Platform Communications - Unified Architecture
OS	Operating System
PLC	Programmable Logic Controller
PPG	Photoplethysmography
PTSD	Post-Traumatic Stress Disorder
QoS	Quality of Service
RAMI 4.0	Reference Architectural Model for Industry 4.0
REST	Representational State Transfer
SAP	Systems, Applications and Products in Data Processing
SLAM	Simultaneous Localization and Mapping
SLB	Server Load Balancer
SQL	Structured Query Language
SRTP	Secure Real-Time Transport Protocol
SSO	Single Sign-On
STAR-RA	Reference Architecture for AI-Based Industry 5.0 Systems
STT	Speech-to-Text
STUN	Session Traversal Utilities for NAT
TCP	Transmission Control Protocol
TOGAF	The Open Group Architecture Framework

TTS	Text-to-Speech
TURN	Traversal Using Relays around NAT
UDP	User Datagram Protocol
URLLC	Ultra-Reliable Low-Latency Communication
VIM	Virtualization Infrastructure Manager
WebDAV	Web Distributed Authoring and Versioning
WebRTC	Web Real-Time Communication
WMP	Workers Movement Prediction
XAI	eXplainable Artificial Intelligence
XR	Extended Reality
XRSI	Extended Reality Safety Initiative

EXECUTIVE SUMMARY

Deliverable D2.3, titled “Reference Architecture Model and Technical Specifications v2”, is a cornerstone of the XR5.0 project, providing essential guidance for the development and integration of the platform. It serves as a central reference for developers and stakeholders, ensuring seamless connectivity and robust design principles across Industry 5.0 use cases such as product design, remote maintenance, production simulations and XR-based training.

The architecture emphasizes flexibility, scalability and interoperability. It integrates inputs from six pilots, aligning with their diverse needs and incorporating widely recognized standards like a high-level approach to design architecture framework with The Open Group Architecture Framework (TOGAF), an architecture modeling language ArchiMate and C4 model for visualising software architecture. The deliverable presents:

- A business-focused Enterprise Architecture offering strategic alignment across processes, applications and technologies.
- A Solution Architecture detailing system components, external integrations and information flows.

Additionally, the deliverable includes technical specifications of tools and new technologies used, ensuring modularity and compatibility. It outlines how XR5.0 aligns with existing industry standards while preparing for integration with the upcoming EU XR platform to support collaboration and innovation in Industry 5.0.

This second iteration further enriches the deliverable by incorporating detailed Component (C4) diagrams for XR5.0 components, three new architectural blueprints (Personalized Training, Training Workflow, and Security & Access Control), and updated technical specifications including APIs, internal architecture, software dependencies, and authentication mechanisms. These enhancements provide additional clarity, traceability, and alignment with relevant standards, while maintaining consistency with the architecture presented in the first iteration.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The content of this Deliverable serves three purposes: to act as an Architecture reference point for XR5.0 developers, to document the design and integration work in the context of WP2 and to serve as a blueprint for future endeavors in building systems that will try to address Industry 5.0 use cases by making use of XR and AI technologies.

The Deliverable first introduces the reader to the Architectural design process and the approach used for building the diagrams of the Architecture through requirements engineering. All user stories which have been created according to the needs of the 6 pilots have been reported since they played a major role in the design of the Business Layer of the Enterprise Architecture. Then, the standards and specifications used by the Architecture are listed and documented. More specifically, the TOGAF, ArchiMate and C4 specifications are analyzed in order for the user to be able to interpret the diagrams.

The technical specifications of the partners' tools have also been included in this document in high detail. Tools dedicated to various operations such as training, management of the XR environment, asset storage, AI processing, orchestration, etc. have been extensively documented in order to depict their integration in the ArchiMate Application Layer. In the context of this Deliverable, these tools are considered to be generic enough and modular in the sense that they could be replaced by similar tools with the same functionalities. To this end, the title given to the tools in the Architecture Application Layer reflects the functionality of the tool within the XR5.0 system.

After the presentation of the specifications employed by the Architecture (TOGAF, ArchiMate, C4), the diagrams of the Enterprise and Solution Architectures are presented and explained. These 2 architectural categories are supplementary, and they design and describe the same system through different viewpoints, each tailored to different aspects and levels of detail. The Enterprise Architecture offers guidance, alignment, and a holistic context, defining the organization's strategic goals and overall IT landscape, while the Solution Architecture provides operational design, delivering the system-level structure necessary to implement specific capabilities and business processes. Together they ensure that the solution is both strategically aligned and technically sound.

Furthermore, well-known standards in Industry 5.0, Industry 4.0, XR, and AI are included and briefly described in the document and their alignment with XR5.0 Architecture has been analyzed. Moreover, recommendations are being made in order to have in the future a standard capable of boosting compliance and interoperability with future Industry 5.0, XR and AI solutions

The XR5.0 Architecture has been structured to ensure future compatibility and alignment with the EU XR platform. This forward-looking approach aims to promote seamless interoperability and collaboration with the broader Industry 5.0 ecosystem.

This second and final iteration of the Deliverable extends the previous version by including detailed Component (C4) diagrams, additional architectural blueprints (Personalized Training, Training Workflow, and Security & Access Control), and updated technical specifications of XR5.0 components. These updates provide greater clarity, traceability, and alignment with standards while maintaining consistency with the architecture presented in the first iteration.

1.1 Objectives of the Deliverable

The objectives of this Deliverable are the following: to set the basis and the reference point for partners' development and integration efforts, to depict the transformation of various user stories derived from the needs of 6 different pilots into a comprehensive architectural business layer, to present how different tools can synergize and cooperate so as to deliver the so called XR5.0 platform that will boost the training, the effectiveness in troubleshooting and maintenance and the monitoring of the production line in Industry 5.0 environments. Moreover, this deliverable aims at reviewing current standards in Industry 5.0, Industry 4.0, XR and AI domains and examines their capabilities or deficiencies in supporting XR and AI-enabled Industry 5.0 person-centric applications.

Moreover, the Deliverable's objectives are to ensure all partners align on key architectural principles and are aware of how each tool fits within the overall XR5.0 architecture, as well as aligning the reference architecture with relevant European and international standards, ensuring compliance and fostering wider adoption. The Architecture and the technical specification of the tools have been documented in a manner that is accessible and understandable to all partners, including non-technical stakeholders. Furthermore, it acts as a reference document for onboarding new stakeholders in the project. Finally, the goal is through this document to provide a roadmap and guidelines for further development and enhancement of the tools and the architecture and ensure that the Deliverable can serve as a foundation for subsequent project phases or follow-up initiatives.

1.2 Insights from Other Tasks and Deliverables

This Deliverable entitled "Reference Architecture Model and Technical Specifications" has been published in the context of WP2, named "Specifications, Architecture and EU XR Platform Integration". To this end, along with the other tasks of WP2, it contributes to preparation of architectural guidelines that will drive the further development of the tools and their integration into a unified interoperable platform. The Deliverable was heavily based on the work of task T2.1 "Analysis of Reference XR Scenarios for Industry 5.0 Apps" which has been documented in the Deliverable D2.1 [2], entitled "Requirements and Reference Scenarios Analysis". Thereafter, the user stories of reference XR scenarios which have been gathered by the 6 pilots of the project were the cornerstones for building the architecture and especially for designing the business layer which represents the operational and strategic aspects of the project.

In addition to task T2.1, this deliverable has also been influenced by the findings of task T2.2, entitled "Technical Specifications of XR5.0 Components and Technologies" in the sense that the partners' tools form the building blocks of the system and shape the design and operational aspects of the architecture in several ways: the architecture must be designed to accommodate the tools capabilities within XR5.0 platform, the tools influence what modules or services need to be created for proceeding with the integration and determining the needed interfaces. Moreover, since each tool may depend on specific requirements (software, hardware, etc.), the architecture must ensure compatibility across all tools, and if tools require shared datasets, the architecture should include centralized storage.

Task T2.4 entitled "Cloud/Edge and BigData Infrastructures for XR5.0 Apps" also played a huge role in shaping the architecture and more specifically in designing the Enterprise Technology Layer. The task T2.4 which is responsible for the integration and more specifically for specifying the pipelines and workflows to seamlessly deploy XR5.0 platform across a wide range of industrial settings, was a roadmap for designing the architecture in the lowest level (i.e. Technology Layer) by depicting the resources, network and other tools needed for deployment and the continuous integration purposes.

The content of this Deliverable and more specifically the schema of the tools integration will feed the activities of task T2.5 "Design, Implementation and Documentation of Open APIs" which reports on the set of APIs that will facilitate the integration and communication between XR5.0 components and between XR5.0 and existing industrial systems. According to the Application Layer of this Deliverable, T2.5 will specify and document the APIs in order to facilitate the integration.

Moreover, the specification of the architecture is designed in a way that will boost compliance and interoperability with the EU XR platform, and thereafter, this document is highly relevant with the activities of task T2.6 entitled "Integration to the EU XR Platform".

The design of the Architecture and the content of this Deliverable are also in line with the Technical Management activities of task T1.2 and the goal is to constitute the blueprint for the activities of the technical packages WP3, WP4, WP5. Finally, the results of the Architecture will be evaluated and validated through the activities of WP6 and more specifically through the deployment in 6 diverse and representative pilots.

1.3 Updates from previous version

This second iteration of the deliverable builds upon the solid foundation established in the first version and introduces several significant enhancements aimed at improving architectural completeness, consistency, and technical precision. The updates strengthen the overall coherence of the XR5.0 architecture and provide partners and reviewers with a more detailed and structured perspective.

The first iteration already included an overview of the relevant standards as well as an initial analysis of their relevance to the XR5.0 Reference Architecture. In this second iteration, this work (chapter 7) has been expanded and refined. Key improvements include:

- ➢ Substantially enriched descriptions of how the XR5.0 architecture conceptually aligns with each standard, with clearer mappings to architectural layers, workflows, and design principles.
- ➢ A more consistent and traceable approach to cross-standard relationships, ensuring that alignment statements are evidence-based, neutral, and non-prescriptive.
- ➢ Addition of a Cross-Standard Analysis subsection and a Traceability Table, providing a clearer synthesis across standards.

These enhancements deepen the standards-related analysis without replacing the work of the previous version.

Moreover, to complement the high-level architectural views included in the first iteration, this version adds C4 Component Diagrams for all XR5.0 components. These diagrams enhance architectural transparency and:

- ➢ Present a uniform and widely accepted notation for describing internal component architecture.
- ➢ Clarify each component's responsibilities, interfaces, and dependencies.
- ➢ Allow clearer traceability between conceptual architecture, standards alignment, and component-level behavior.

Further to the aforementioned additions, three new architecture blueprints have been introduced to strengthen the operational and behavioral understanding of the XR5.0 ecosystem:

- ➢ Personalized Training Blueprint
- ➢ Training Workflow Blueprint
- ➢ Security and Access Control Blueprint

These blueprints complement the existing diagrams by providing higher-level, reusable architectural patterns that reflect key XR5.0 design principles.

Concerning the technical specifications of XR5.0 components, the first iteration already included technical descriptions of the XR5.0 components. In the present version, these descriptions have been updated in the following categories:

- ➢ API descriptions and interaction patterns
- ➢ Updated information on internal structure and software dependencies
- ➢ Updated security and authentication mechanisms
- ➢ Updated clarifications on runtime requirements and integration considerations

These updates strengthen the technical accuracy and completeness of the component documentation.

Consequently, the second iteration does not replace the content of the first version but extends and deepens it, ensuring higher architectural maturity, better traceability across viewpoints, and clearer alignment with relevant standards. The document now provides a more robust basis for internal integration, technical evaluation, and future work within XR5.0.

1.4 Structure

Chapter 2 presents the methodology used for designing XR5.0 Architecture. The review of standards in Industry 5.0 and Software Systems Architecture, the study of available XR5.0 documents and the collaboration with the partners are highlighted in this chapter. Moreover, an explanation is being provided for employing TOGAF and ArchiMate specifications for designing the Enterprise Architecture, and for employing C4 for designing the Solution Architecture. The process of Requirements Engineering and the list of XR5.0 user stories are also included in this chapter.

Chapter 3 presents the technical specifications of XR5.0 components and technologies. Building upon the first iteration, the descriptions have been updated and enriched to provide more complete information on each component's functionalities, interfaces, internal architecture, software dependencies, runtime requirements, and authentication mechanisms. These enhancements improve the technical clarity of the architecture and support traceability across components, services, and standards.

Chapter 4 explores the application of "The Open Group Architecture Framework" (TOGAF) [6] within the XR5.0 project. It is explained that TOGAF provides a structured methodology for developing and governing enterprise architectures. Moreover, five phases of TOGAF's iterative Architecture Development Method (ADM) are implemented in XR5.0, each contributing to the alignment of IT investments with Industry 5.0 goals.

Chapter 5 addresses the Enterprise Architecture of XR5.0, presenting the various layers (business, application, and technology), the vision of the architecture, and the organization viewpoint. In addition to translating user stories into business processes (business layer) and illustrating the integration of the various tools within the XR5.0 platform, this chapter now introduces three new architectural blueprints: the Personalized Training Blueprint, the Training Workflow Blueprint, and the Security and Access Control Blueprint. These blueprints provide reusable patterns and detailed visualizations that enhance understanding of XR5.0 operations, training processes, and security mechanisms.

Chapter 6 has been extended beyond the System Context and Container diagrams included in the first iteration, now incorporating detailed Component (C4 Level 3) diagrams for each architectural building block. These new diagrams provide a clearer and more granular view of the internal structure, responsibilities and interactions of the XR5.0 components, supporting a more complete and traceable architectural description.

Chapter 7 builds upon the initial standards analysis from the first iteration and significantly extends it by providing a more structured, detailed and traceable alignment between the XR5.0 Reference Architecture and relevant standards. The chapter now includes deeper mappings, clearer cross-standard relationships, and an expanded explanation of how the XR5.0 architectural concepts remain interoperable with established frameworks in Industry 4.0, Industry 5.0, XR and AI.

Finally, chapter 8 addresses the plan for integrating the XR5.0 Reference Architecture with the EU XR platform. It highlights the forward-looking strategies for ensuring interoperability, compliance and alignment with the EU XR platform's anticipated framework. Thereafter, it outlines initial steps, future plans and considerations to foster collaboration and compatibility.

2. Documentation of the process

The development of the system's architecture followed a structured methodology, ensuring alignment with the project's goals and stakeholder expectations. This chapter outlines the steps undertaken to gather, analyze and document the necessary inputs for the architecture design. The process combined the review of foundational project documents, active participation in collaborative workshops and discussions, and iterative refinement based on stakeholder feedback. Key inputs for this process included:

- ➢ Review and selection of the appropriate standards and specifications for the definition of XR5.0
- ➢ The Description of Action (DoA) of the project, which defined the overarching objectives and scope.
- ➢ Deliverable D2.1 entitled "Requirements and Reference Scenarios Analysis" [2] which provided user stories and scenarios critical for defining the system's functionality and behavior.
- ➢ Documentation from co-creation workshops conducted for the six pilots, which highlighted specific use cases and contextual needs.
- ➢ Insights and decisions derived from teleconferences and discussions with project partners, which played a pivotal role in refining the architecture design

More specifically, the first step began with the selection of the specifications and standards used for designing XR5.0 Architecture. The choice was to employ TOGAF [6] and ArchiMate [3] for defining the Enterprise Architecture. TOGAF is a widely adopted and well-established enterprise architecture framework with a comprehensive approach to structuring a project's architecture. On the other hand, ArchiMate is the official modeling language for TOGAF, providing a visual tool for representing architectural concepts, relationships and layers. The main reasons for using TOGAF (and consequently ArchiMate) for designing the XR5.0 Enterprise Architecture are the following:

- ➢ **Holistic Approach to Architecture**: TOGAF provides a holistic approach to aligning business and IT strategies with technical design and implementation. This is crucial for XR5.0, which aims at integrating human-centric solutions with advanced technologies (AI, IoT, XR, etc.)
- ➢ **Support for complex and evolving technologies**: TOGAF supports the development of flexible architectures that can evolve with these cutting-edge technologies. Flexible architectures are essential in XR5.0 considering the need for integrating with EU XR platform and diverse industrial settings
- ➢ **Cross-Disciplinary collaboration**: XR5.0 involves a diverse group of stakeholders across industries, research institutions, SMEs and countries. TOGAF (and thereafter ArchiMate) supports cross-functional collaboration through its structured methodology for governance, stakeholder engagement and decision-making processes.
- ➢ **Scalability and Flexibility**: XR5.0 accommodates a wide variety of actors, technologies and use cases. TOGAF offers adaptability in designing architectures that are both scalable and flexible, making it easier to incorporate new technologies, standards, or business processes as the project progresses. Moreover, ArchiMate enables clear communication of how changes at one level (e.g. XR integration) affect other areas (e.g. business processes, applications, etc.)
- ➢ **Alignment with European standards and regulations**: As the project is European, TOGAF provides a framework that is aligned with international standards and practices, ensuring compliance with EU regulations and standards, such as GDPR or the EU Digital Strategy. ArchiMate also supports this, since it provides a standardized notation for modeling that can be used across different countries and institutions within the EU, promoting consistency and interoperability.
- ➢ **Risk Management and Governance**: With a project of this scale and complexity, proper governance and risk management are vital. TOGAF has a strong focus on risk management, ensuring that architectural decisions are made with a full understanding of potential challenges, including technological, financial and operational risks.
- ➢ **Open and Community-driven**: Both TOGAF and ArchiMate are open frameworks with active communities. This ensures continuous development and support, along with access to a broad base

of knowledge and best practices. For a research project such as XR5.0, the ability to stay up to date with the latest trends and advancements in architecture is a major advantage.

Thereafter, these 2 architectural frameworks provide a robust, flexible and standardized approach to designing and managing the Enterprise Architecture of a complex European research project such as XR5.0. They ensure that the project can adapt to emerging technologies, foster collaboration across disciplines and align with industry and regulatory standards, while maintaining a human-centric focus and robust governance framework.

In addition to incorporating these specifications for defining the Enterprise Architecture, it has been decided to also include a separate section entitled “Solution Architecture” which makes use of the specification C4[4]. With C4 specification, the goal is to provide a more practical and actionable design for XR5.0 development teams, to provide a common understanding for both business and technical stakeholders, ensuring alignment across different levels of the project, and also to provide simple diagrams that help technical teams understand system components, interactions and dependencies. Moreover, C4 gives more specific, system-level design details (how the system operates and integrates). Thereafter, the reasons for choosing C4 (the Context, Containers, Components, and Code model) for the Solution Architecture alongside TOGAF and ArchiMate seems to be a beneficial approach for the following reasons:

- **Clear and Understandable Visual Representation**: The C4 model is designed to provide a clear, hierarchical set of diagrams that are easy to understand, even for stakeholders who may not be technically savvy. The focus is to provide diagrams that present a high-level overview of the system and its core components without overwhelming the XR5.0 audience with too many details, something extremely important in collaborative research environments with diverse stakeholders, including developers, industrial partners and researchers.
- **Focus on System Boundaries and External Interactions**: The diagrams presented in the context of the Solution Architecture through C4, as shown in chapter 6, establish system boundaries clearly. More specifically they focus on how XR5.0 interacts with external entities (users, pilots, etc.) and they highlight the **actors** and their relationships with the system, ensuring that all interactions, whether they are human-centric (such as XR interfaces) or machine-based (AI) are well understood.
- **Simplicity and Agility for Early-Stage Research Projects**: The C4 model that represents the Solution Architecture, with its 2 diagrams (System Context and Container) allows for a simplified approach that is agile and flexible, which is ideal for early-stage design work or projects where the exact specifications and technologies may evolve over time. XR5.0 falls into this category, and focusing only on these 2 diagrams keeps the design process light and adaptable, avoiding unnecessary complexity while still allowing for the necessary system design communication and iteration as the project progresses.
- **Complementing TOGAF and ArchiMate:** TOGAF and ArchiMate provide comprehensive enterprise architecture views, focusing on business, application and technology layers. These frameworks give a broad view of how the enterprise works, how processes are aligned and how technologies support the business. By adding the C4 model for the Solution Architecture, the architecture is enriched with a focused, system-specific view of how the core technology operates within the larger context. Thereafter, C4 bridges the gap between Enterprise Architecture and actual Solution Architecture, allowing to translate high-level enterprise objectives into the design of specific systems and components.

To this end, using C4 diagrams allow to complement the Enterprise Architecture with a focused, clear and flexible view. This enables stakeholders to understand the core system architecture, the external interfaces and the high-level components in a way that is accessible, scalable and adaptable to the evolving needs of XR5.0.

After the selection of the tools and standards for the definition of XR5.0 architecture that have been described above, the process continued with a detailed review of the project documents to identify key objectives and requirements. These documents include the Description of Action (DoA), the Deliverable

D2.1[2] “Requirements and Reference Scenarios Analysis” and documentation of the co-creation workshops. Deliverable D2.1 provided the Architecture high-level goals, user stories and scenarios that defined the functional scope of the system. Additionally, the co-creation workshops documentation captured pilot-specific use cases, offering an even deeper understanding of user needs and challenges.

It has to be mentioned that the collaborative nature of the process was central to its success. Regular telcos and discussions with project partners allowed for the validation and refinement of requirements, ensuring alignment with both technical feasibility and stakeholder expectations. This iterative approach ensured that the Architecture evolved in response to insights from all XR5.0 partners. Figure 1 illustrates the process undertaken to design the XR5.0 Architecture. **Document Analysis** serves as the foundation, incorporating static inputs such as the Description of Action (DoA), Deliverable D2.1 (which includes the user stories), and insights from co-creation workshops. These inputs feed into **Partners Collaboration,** which involves ongoing discussions with partners through telcos and other meetings to refine system requirements and validate architectural decisions. Then, the process includes the Architecture Design which consists of the Enterprise Architecture and the various layers (business, application, technology, strategy) and the Solution Architecture. It has to be noted that an iteration loop is indicated, showing how feedback from architecture design informs further partners' collaboration, ensuring the design aligns with XR5.0 goals and evolving requirements.

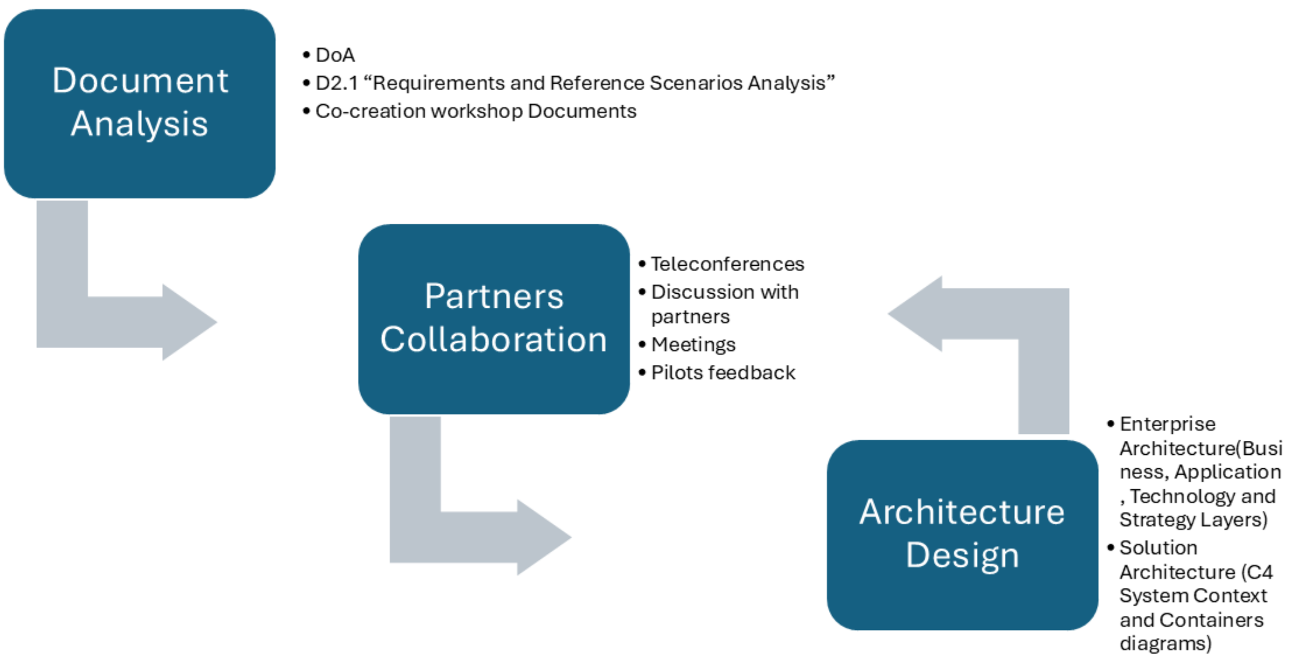


Figure 1 - Key Steps in XR5.0 Architecture Design

2.1 Requirements Engineering Process

This section documents the requirements engineering (RE) methodology and underlying basics which are applied to the XR5.0 project. To utilize the potential of existing structures and at the same time harmonize the RE process across the different technical stakeholders in the project, the RE methodology is primarily based on the international standards “ISO/IEC/IEEE 29148-2018 - Systems and software engineering — Life cycle processes – Requirements engineering”. Furthermore, the standard “ISO/IEC/IEEE 24765-2017 - System and software engineering – Vocabulary” is used as a clarification basis on Systems Engineering (SE) terminology. In accordance with ISO/IEC/IEEE 29148-2018, the XR5.0 project has adopted an iterative RE model which clusters the requirements engineering into three main steps (Figure 2) The steps are sequential; however, the iteration loops are designed so that they may include only certain steps.

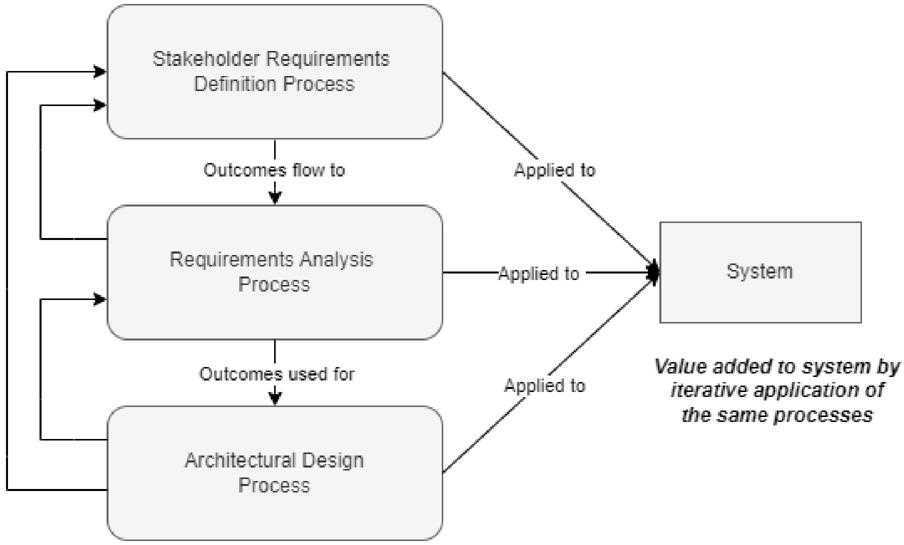


Figure 2 - Iterative systems engineering processes

This iterative approach to requirements engineering allows the project to adapt both the requirements engineering process and the systems development process to the changing dynamics of the three business cases. The process is not specific only to RE but is also applied to all development activities in XR5.0.

The objective of the requirements' engineering process is to develop a model of requirements that facilitates an agreed understanding between stakeholders (e.g., operators, customers, managers, virtual commissioners, etc.). This model must be validated against real-world needs, be implementable, and provide a basis for verifying designs and accepting solutions.

The identification and analysis of the requirements of reference XR scenarios within the context of Industry 5.0 has been presented in Deliverable D2.1 [2]. This analysis outlined how XR5.0 can effectively address these challenges, providing a comprehensive framework for integrating XR technologies in Industry 5.0 applications. The Deliverable D2.1 presents many XR5.0 aspects, such as pilot descriptions, background technologies and reference scenarios, but from the perspective of architecture, the user stories defined in D2.1 are considered of utmost importance since they serve as a foundation for designing the business layer of the enterprise architecture.

XR5.0 user stories encapsulate the key requirements, objectives and behaviors that the system must support to deliver value to stakeholders (i.e. operators, training managers, etc.). By focusing on user-centric goals and outcomes, these stories translate business needs into actionable features and processes. Each story typically defines "who" the user is, "what" they need, and "why" it matters, which helps architects prioritize functionalities and design modular, scalable components. In the enterprise business layer, user stories guide the alignment of services, workflows and data models to ensure they fulfill real-world use cases. This iterative and incremental approach fosters collaboration across teams, improves understanding of complex business requirements and ensures the resulting architecture is both adaptable and user focused. For a complete reference of the user stories, the user can refer to Deliverable D2.1 [2].

3. Technical Specifications of XR5.0 Components and technologies

This section elaborates on the technical specifications of the technologies used in XR5.0. This will describe each component in detail to inform the reference architecture for XR5.0. The structure of this description is provided according to the following sections:

Section 3.1. Description and Functionalities and Components, introduces the purpose and utility for each component. This description is provided to demonstrate how technology supports XR5.0. It also indicates whether the technologies were previously developed or developed specifically under the project. Most of the technologies are software components, but some hardware technologies are also considered.

Section 3.2. APIs and Interfaces describe how each component interacts within XR5.0, highlighting protocol support and methods for external systems to interact with XR5.0.

Section 3.3. Input and Output Data Formats outline the data formats for input and output across components to enable interoperability between platforms.

Section 3.4. Schematics and Internal Architecture contain the architecture and schematics of each component, including logical flow diagrams, interactions between components, and structural layouts.

Section 3.5. Software Requirements and Dependencies detail the requirements, including required or compatible versions, operating systems, and libraries for use by each component.

Section 3.6. Authentication Mechanisms provides a summary on the security protocols and authentication mechanisms related to every described component such as encryption algorithms or authentication protocols.

3.1 Description and Functionalities of the Components

For Section 3.1 depicting the Description and Functionalities of Components of XR5.0, an overview of each technology's intended use and its role within the project is provided, illustrating how each element contributes to achieving XR5.0's objectives. The following technologies represent different hardware and software components, each uniquely integrated to support XR5.0's goals in data processing, remote assistance, user interactions, XR and explainable AI (XAI) capabilities. Additional information is also provided to describe whether the technologies were previously developed (background technologies) or were developed specifically for the XR5.0 project (foreground technologies).

Hardware and Operating Systems

The Hardware and Operating Systems section describes the underlying technologies that will support XR5.0 XR immersive and functional scenarios. This hardware consists of sophisticated XR devices and operating systems that together improve usability, data capture, and user experience in different industry contexts. The RealWear Arc 2 is a second-generation head-mounted display from RealWear being a lightweight and ergonomic device, optimized for prolonged, hands-free use in environments ranging from logistics to remote support. This device is coupled with Almer OS, an Android-based OS specifically devised to optimize the interaction with AR capabilities, providing seamless and intuitive use for real-world applications. Another XR device available at the project is the RealWear Navigator from RealWear, a hands-free AR headset designed for industrial use. Oculavis SHARE provides device-agnostic connectivity for remote troubleshooting and operational efficiency between equipment manufacturers, technicians and experts from locations around the globe. For data capture, the project also counts on eye-tracking technologies, which enables collecting eye behavioral and physiological responses for serving as an interface between the eye-tracking device and the Clawdite platform that processes the collected data. The combination of these hardware and operating system contributes to an integrated and robust infrastructure of XR5.0.

Augmented and Virtual Reality Platforms

The Augmented and Virtual Reality Platforms section describes the technologies that drive XR5.0 featuring advanced and immersive tools for industry use cases. The solutions combine real-time guidance, high-end streaming, and high-quality visualizations, addressing a range of use cases from remote support to training modules in off-the-job and on-the-job training. TeamViewer Frontline running on the RealWear Arc 2 headsets, supports hands-free interaction via the overlay of digital instructions over physical environments to improve workflow efficiency for field workers and technicians.

Another technology used in XR5.0 that supports management and streaming of high-fidelity visualizations is the HOLO's XR solutions. The following figure illustrates three core components: Hololight Space, Hololight Stream and Hololight Hub (Figure 3).

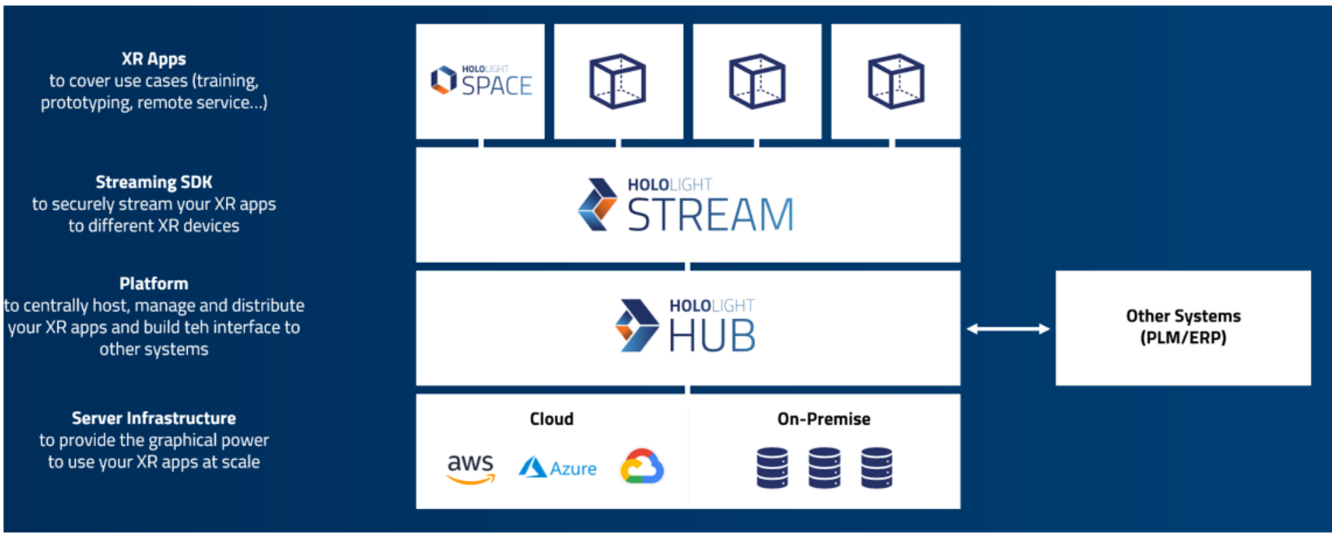


Figure 3 - HOLO's components supporting XR solutions in XR5.0

Hololight Hub and Hololight Stream provide flexible, scalable solutions for streaming and managing XR applications, giving users an effortless experience with low-latency visualizations, while Hololight Space provides high-fidelity, interactive 3D visualizations that enhance engineering and development workflows. These technologies were already available before the project but were further enhanced to support the XR5.0 Training Platform. Specifically, the Hololight Hub will be the backbone of the XR5.0 Training Platform with features to create and store training programs. It will also provide extension of user profile to include training data and will integrate an intermediate layer to handle training data, assets and materials and to store in a database and AWS S3.

The Training Platform with associated modules are also available for training at different industrial use cases. This platform has its predecessor as a Virtual Reality Training module for Aircraft Maintenance Technicians offering a step-by-step procedure for guidance in essential tasks, thereby promoting hands-on learning and skill refinement. In XR5.0 this platform evolved to XR combining two main modules, the Training Programs Authoring Tool (TPAT), a web application for managers to create and manage XR Training Programs and Materials, and the XR Training Plugin (XRTP), the Unity plugin that enables the visualization of XR Training Programs and Materials in XR devices. It also includes XR5.0 Training Asset Repository, a cloud base storage solution for XR training programs, and an AI tool, the AI Voice Assistant, one application using voice commands to assist VR navigation in the training platform. Overall, these technologies are aimed at improving XR5.0's training, usability, and real-time support capabilities.

Smart Service, IoT, and IIoT Integration

Smart Service, IoT and IIoT Integration section outlines the technologies that support connectivity, monitoring and collaboration through XR5.0. Clawdite is an extensible and flexible IIoT-based platform supporting the creation of customized data representations of production systems and their entities, including humans. It also enables dynamic and quasi-static data management. Some technologies relying in the Clawdite are the following: 1) Training Platform – Data & Authentication, 2) Workers movement prediction, it is designed to detect the real-time positions and orientations of human workers in a shared industrial environment; 3) Worker Shadowing and Monitoring for combining expert knowledge with process tracking, enabling the virtual shadowing of experienced operators, 4) UI Adaptation Engine for integrating user-related data and biometric inputs to compute engagement and stress metrics for dynamic adjustment of the training; 5) Troubleshooting Assistance Service consisting in real-time insights and support to workers to be exploited by XR devices. The Smartwatch Connector is based on Wear OS and enables real-time collection of physiological data from industrial workers and send these signals to Clawdite for data processing, which is critical for XR5.0 monitoring and data-driven applications. A different technology also part of Smart Service, IoT and IIoT technologies, is Kubernetes, being defined is an open-source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications. Kubernetes abstracts underlying infrastructure resources and provides a resilient, declarative control plane for running distributed systems at scale.

Explainable AI (XAI) and Machine Learning Technologies

The Explainable AI (XAI) and Machine Learning Technologies section describes the advanced AI models that enhance transparency, adaptability, and user understanding within XR5.0. The integration of Explainable AI (XAI) and Deep Learning Models, XR5.0 contributes to explainable predictions using advanced AI models. That improves transparency in the AI decision-making process and enhances trust in the system. Generative AI as a Service component adds a next level of interactivity within XR environments, providing context-aware responses that adapt based on the user's environment, inputs, and tasks, thus potentially contributing to user engagement. This technology is being developed in the project and leverages existing open-source technologies such as LlamaIndex, FastAPI, React 18, MongoDB, Redis and LangChain & HuggingFace Transformers. The project will also comprise advanced AI paradigms such as Active Learning and Neurosymbolic AI for supporting AI systems in the project. The Azure AI Services (Azure OpenAI) will be used to process and summarize maintenance reports, whereas OpenAI API Services provides advanced AI capabilities. Including natural language understanding, generation, and audio processing. A service for documents processing based on a RAG Application is implemented for document processing and conversion, and knowledge base querying through retrieval augmented generation. Another XAI with Domain-Specific Knowledge component uses ontologies, knowledge graphs, and domain-specific knowledge to create context-specific explanations for the user to support their decision-making and understanding that was defined as background technology of this project. Last, Integration and Visualization of AI/XAI provide these AI explanations and recommendations in a clear manner in the XR environment to follow the open science principles and to support the user with understanding of the AI-driven process.

Data Management and Analytics

The Data Management and Analytics section describes the infrastructure supporting XR5.0 to handle large datasets and provide insight in real-time, crucial for data-driven and immersive experiences. This is supported by the Ultra-Scalable Relational Database – LeanXcaleDB and the Training Repository. The LeanXcaleDB is a technology able to handle big data with high efficiency. It includes scalability capabilities for retrieving, processing, and storage, necessary for the real-time analytics that support XR5.0's functionalities. It also includes abstraction layers as AAS OPC-UA Connector and AAS Pupil Labs API Connector for enabling data exchange, interoperability, and secure integration in different industry use cases. Finally, MageAI ETL pipelines for data processing, formatting and provisioning are also included in the list of available technologies for the XR5.0 project. Finally, Grafana is an open-source platform that enables users to create dashboards from the data saved in Clawdite.

Table 1 – *Summary Table of XR5.0 technologies*

Technology		Role	Brief description
Hardware and Operating Systems	RealWear Arc 2	Background/Foreground	Lightweight AR headset
	RealWear Navigator 520	Background/Foreground	Hands-free AR headset
	Almer OS	Background	Android-based OS for ALMER headsets
	Smartwatch connector (Wear OS)	Foreground	PineTime SmartWatch, based on the InfiniTime OS
	Eye tracker	Background	Collects behavioural and cognitive data from workers
Augmented and Virtual Reality Platforms and associated Modules	Hololight Hub	Background	Orchestration platform for AR/VR contents
	Hololight Stream	Background	Remote rendering
	Hololight Space	Background	Industrial VR and AR visualization software
	TeamViewer - Frontline	Foreground	AR software solution tailored for industrial use.
	SLB for AMT training	Background	Immersive tool for AMT training
	Training Programs Authoring Tool (TPAT)	Foreground	Web application for managing training materials and programs, with the Unity plugin that enables XR visualizations.
	XR Training Plugin (XRTP)	Foreground	Unity plugin that enables the visualization of XR Training Programs and Materials in XR devices.
	XR5.0 Training Asset Repository	Foreground	Cloud Base Storage Solution for XR modules/materials
	Oculavis SHARE	Foreground	Remote Maintenance Platform for different OS
	Ari – Smart Assistance	Background	Smart assistant for AR devices
	AI Voice Assistant	Foreground	AI Voice Assistant application using voice commands to assist VR navigation
Smart Service, IoT, and IIoT Integration	Clawdite	Background	Extensible and flexible IIoT platform for customised data representations.
	Training Platform – Data & Authentication	Foreground	Collects the information from the workers' training preferences

	Workers Movement Prediction	Foreground	Component of the Clawdite platform. It is designed to detect the real-time positions and orientations of human workers in a shared industrial environment.
	Worker Shadowing and Monitoring	Foreground	Combines expert knowledge with process tracking, enabling the virtual shadowing of experienced operators
	UI Adaptation Engine	Foreground	Integrates user-related data and biometric inputs to compute engagement and stress metrics, supporting dynamic adjustment
	Troubleshooting Assistance Service	Foreground	Provision of real-time insights and support to workers, making available a single point of interaction to be exploited by XR devices
	Smartwatch Connector (BLE Watch)	Foreground	Enables real-time collection of physiological data from industrial workers
	Kubernetes	Background	Open-source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications.
Explainable AI (XAI) and Machine Learning Technologies	Active Learning	Foreground	Deep Learning models
	Generative AI as a Service	Foreground	Fast API-based web service with enhanced interactive capabilities integrated with OpenAI API
	CYENS Centre of Excellence	Background	Integration and Visualization of AI/XAI
	NeuroSymbolic AI	Foreground	AI system combining deep learning with context based features
	Azure AI Services (Azure OpenAI)	Background	AI system to process and summarize maintenance reports.

	OpenAI API Services	Background	Provides advanced AI capabilities. Including natural language understanding, generation, and audio processing.
	Documents processing/RAG Application	Foreground	A 2-step process application for document processing and conversion, and knowledge base querying through retrieval augmented generation (RAG).
Data Management and Analytics	LanXcal eDB	Background	Ultra-scalable relational database for real time analytics
	AAS OPC-UA Connector	Background	Abstraction layer for industrial assets, enabling data exchange, interoperability, and secure integration with I4.0
	AAS Pupil Labs API Connector	Background	Abstraction layer for industrial assets for enabling data exchange between different devices.
	MageAI ETL pipelines	Foreground	MageAI ETL pipelines for data processing, formatting and provisioning.
	Grafana	Background	Open-source platform that enables dashboards leveraging dynamic data saved within Clawdite (e.g., the worker/cell/robot monitoring dashboards)

The analysis of the technologies across the five categories shows that a significant portion of the ecosystem is built upon background technologies, which serve as the foundational layer of XR5.0. These pre-existing hardware platforms, XR frameworks, AI services, and data-management components provide mature, reliable capabilities that the project can leverage and extend. Their role is essential in ensuring interoperability, stability, and scalability, allowing the project to focus development efforts on higher-level functionalities rather than rebuilding core infrastructures.

In parallel, the project introduces a substantial number of foreground technologies, developed specifically to address XR5.0's requirements for industrial training, intelligent assistance, worker monitoring, adaptive interfaces, and integrated data analytics. These new components expand the functionality of the underlying systems, enabling richer XR experiences, personalized support services, and AI-driven insights. The interplay between background and foreground technologies demonstrates how XR5.0 combines the technological foundations with targeted innovation to deliver the XR5.0 ecosystem.

3.2 Specification of APIs and Interfaces provided and exposed by each Component

Section 3.2 related to Specification of APIs and Interfaces Provided and Exposed by Each Component, describes the communication protocols, APIs, and interfaces available for each technology integrated within XR5.0. The components span the area types identified above where the technologies of XR5.0 are organized.

Hardware and Operating Systems

The RealWear Arc 2 devices are supported by a customer-facing application available as an APK, facilitating installation and access to AR functionalities on Android-based devices. Almer OS is compatible with a wide range of Android applications, allowing users to install and operate standard Android apps directly on their AR devices without additional integration components, apart from compatible AR hardware. As for the other OS, Wearable OS, communication between Bluetooth Low Energy (BLE) devices is managed using the Bluetooth GATT (Generic Attribute Profile) protocol, enabling efficient and low-power data exchange, ideal for wearable applications. For the Smartwatch connector (Wear OS) for the API it uses BLE watch connector.

Augmented and Virtual Reality Platforms

The TeamViewer Frontline APK is compatible with any AR hardware running Almer OS or Android OS version 9 or higher, making this component compatible with a wide range of AR devices. This platform easily integrates with enterprise systems like ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) platforms to enable efficient data capture, compliance, and workflow optimization.

Streaming: The following image illustrates the XR streaming approach of Hololight Stream for real time streaming of XR applications cloud server to an XR device. A high-level overview of the components and data flow in the system is depicted in Figure 4.

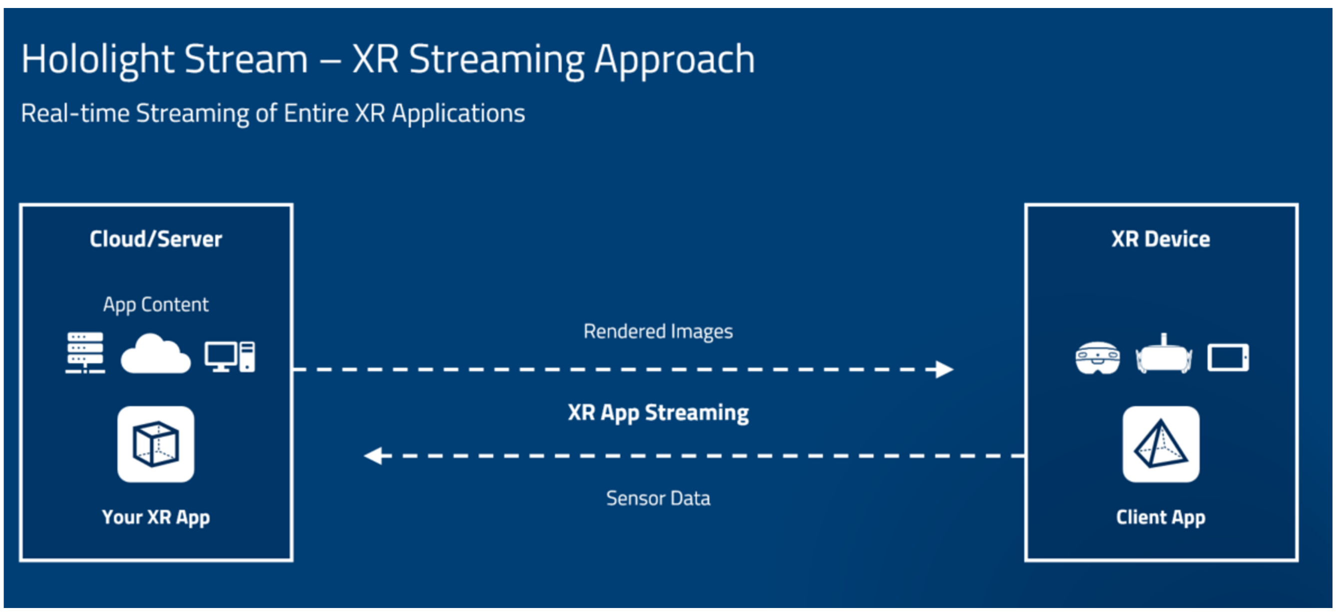


Figure 4 - Hololight Stream – XR Streaming Approach

Hololight Hub uses Web Real-Time Communication (WebRTC) protocols for secure and low-latency streaming, providing instantaneous access to data and visuals. In addition, Secure Real-Time Transport Protocol (SRTP) and Datagram Transport Layer Security (DTLS) are also implemented to ensure data integrity and confidentiality. Web Access is based on HTTPS for reliable web interactions. This component provides APIs to allow integration with other services, enabling customized extensions and ensuring compatibility with systems in XR5.0. See Figure 5 for more details about the Training platform that uses the above components.

Hololight Stream supports Unity Plugin and Software Development Kit (SDK) integrations, offering a developer-friendly environment for incorporating high-quality streaming into XR applications. It also supports Mixed Reality Toolkit (MRTK), facilitating development for mixed reality. Hololight Stream utilizes WebRTC for efficient real-time streaming, optimized to deliver high-quality visuals with low latency.

The Training Repository Storage Solution is based on Web Distributed Authoring and Versioning (WebDAV) which is an HTTP-based protocol allowing users to collaboratively edit and manage files stored on a remote server. The service availability will be done through RESTful APIs.

On the other hand, Oculavis SHARE uses Oculavis API and standard connectors for Salesforce, SAP and Microsoft. The AI Voice Assistant uses REST API communication endpoints, while the Ari – Smart Assistance requires Almer OS, or any other Android operating system that is equal or above Android 9 with the APK installed on the desired Hardware.

Smart Service, IoT, and IIoT Integration

Clawdite supports integration through two main APIs: the Orchestrator API and the HDM (Historical Data Manager) API. It also features MQTT broker for data sharing and modules communication. These APIs allow for historical data managing and the customization and creation of data representations for production systems, including human interactions. Through these interfaces, Clawdite enables enhanced monitoring and control over connected industrial environments, aligning human and machine data for optimal operational insights.

The Workers Movement Prediction component sends its output through MQTT, while the Worker Shadowing and Monitoring exposes the WSaM API (Worker Shadowing and Monitoring API), used to deliver monitoring data, activities feedback/suggestion and digital-shadow state information to other XR5.0 modules. The UI Adaptation Engine exposes a dedicated Adaptation Engine API, which provides access to adaptive UI elements configuration and parameters. The Troubleshooting Assistance Service exposes the TAS API, a web-service interface responsible for delivering real-time troubleshooting recommendations. Communication occurs via HTTP-based API calls, returning results in JSON.

The smartwatch connector for Wear OS uses the Google Health Connector for data retrieval. Communication is via standard Wear OS and Google Health APIs, relying on Bluetooth Low Energy (BLE).

The Kubernetes rely on RESTful, resource-oriented API exposed by the kube-apiserver. Exposes an OpenAPI (Swagger) specification.

Explainable AI (XAI) and Machine Learning Technologies

The Active Learning component operates as a web service where data is exchanged in JSON format. NeuroSymbolic AI also functions as a web service, with JSON as the primary data format for communication with other components.

The Generative AI as a Service system is built using FastAPI and supports components such as OpenAI API, ChromaDB/Pinecone, Redis Chat Store, LlamaIndex, LangChain, and Hugging Face Transformers. The service is accessible via a RESTful API, enabling scalable, flexible interaction with external applications for generating context-aware responses. This REST API setup provides developers with a standardized method to access and implement generative AI capabilities within XR5.0.

Explainable AI (XAI) with Domain-Specific Knowledge component offers multiple interfaces to ensure comprehensive integration and functionality. The Data Collection Interface integrates with various data sources, supporting the flow of relevant information into the XAI system. Ontology/Knowledge Graph Interface allows for creating, editing, managing, and querying knowledge structures, supporting the development and use of domain-specific knowledge within the AI system. Context Extraction Interface manages context models, using standardized data to load and interpret contextual information, enhancing the relevance and accuracy of explanations provided by the XAI system.

Integration and Visualization of AI/XAI component supports API calls using JSON for structured data exchange and integrates various multimedia services, such as Speech to Text and Text to Speech for audio interactions, and Virtual multimedia content support (3D models, images, metadata, sounds, videos) to enhance the user experience and provide rich, visual explanations within XR5.0's environment. The Documents Processing/RAG tool is supported by REST API communication endpoint. It also uses OpenAI API and Azure AI API for these services.

Data Management and Analytics

The LeanXcaleDB component is designed to be integrated into diverse environments, adhering to industry standards for communication and data processing, such as JDBC/ODBC communication protocols providing also a standard python-based driver. It is already integrated with popular data processing frameworks (i.e. Apache Spark, Flink, Kafka, etc.). This combination of industry-standard communication protocols, versatile API specifications, and integration with popular data frameworks ensures that LeanXcaleDB is highly adaptable and meets the demands of modern data-intensive environments. It also uses AAS OPC-UA Connector and the AAS Pupil Labs API Connector. MageAI ETL pipelines uses REST API communication endpoint.

3.3 Description of each Component input and output data format

In this section related to the Description of Each Component Input and Output Data Format, the data formats used by each XR5.0 component for both input and output is depicted in a structured table (Tables 2-6).

Table 2 – Hardware and Operating Systems

Technology	Input	Output	I/O data format and samples
RealWear Arc 2	Camera Microphone Inertial Sensors Voice control Programmable buttons	Display HD Audio Remote support	-- -- –
Almer OS	Via Android applications Voice commands Accelerometer	Interface optimized for AR interactions Executes small-factor algorithms directly on the device	– Bluetooth GATT protocol
Smartwatch connector (Wear OS)	PPG Heart Rate Sensor	Open to community contributions for customization	–
Eye tracker	--	--	--
RealWear Navigator 520	Programmable buttons	--	--

Table 3 – Augmented and Virtual Reality Platforms

Technology	Input	Output	I/O data format and samples
TeamViewer Frontline	Voice Commands and gesture controls Gathers real-time data from the environment and devices	Displays instructions, highlights objects, and provides real-time annotations Offers step-by-step instructions, checklists, and guided workflows for consistent and accurate task execution	– –
Hololight Hub	Web Interface: -User data -Application data -Training data (files, workflows, assets) XR Client: -User input	User session data Application instance on cloud	--
Hololight Stream	User Input	Pixel Stream	--
Hololight Space	3D models (CAD/Digital Twin files) User input through interaction Custom data - AI enhanced content, JSON, text/image/speech, PDFs etc. Sensor data - camera, microphone	XR representation/XR space; Sensor data - head pose, SLAM, user input; Visualization of 3D models	--
SLB for AMT training	User input through interaction with XR device	3D visualization	--
Training Programs Authoring Tool (TPAT)	--	--	--
XR Training Plugin (XRTP)	--	--	--

XR5.0 Training Asset Repository	--	--	JSON, XML
Oculavis SHARE	Input from technicians and experts for remote guidance Data from maintenance processes managed through the integrated ticketing and work order system User Data for Login	Provides virtual elements overlaying the real world to guide technicians Delivers precise instructions during visual assistance calls User Session Data Provides a platform for resolving support tickets and accessing knowledge resources (Service and machine documentation)	--
Ultron - Smart Assistance	Gathers information about its environment from the sensors of the AR platform Voice commands Detects state of the user	Provides information access to the device Allows communication with others	–
AI Voice Assistant	text or speech (audio files)	text or speech (audio files)	Input - POST requests requiring query params and form body params

Table 4 – Smart Service, IoT, and IIoT Integration

Technology	Input	Output	I/O data format and samples
Clawdite	Streams data from devices such as smartphones, Raspberry Pi, PLCs, etc., through dedicated agents Collects and manages historical data from gateways and functional modules	Over API to any given service Offers components for modeling digital twins that can be reused and extended. Supports modular integration, allowing for additional functionalities as needed Interface with Clawdite APIs and JSON data format	JSON data format

Service Connector Core	Collects data from different clouds, databases, APIs, and services.	Data conversion	MQTT
	Each connector can have specific options passed as arguments to providers, parsers, and consumers	Engine for data interoperability across various systems	JSON; Easily extendable to support other data formats
Training Platform – Data & Authentication	--	--	--
Workers Movement Prediction	RGB-D cameras feed	It returns the current and predicted workers position coordinates over time windows of 1, 3 and 5 seconds	JSON (via MQTT text message)
Worker Shadowing and Monitoring	Visor tracking about operator head and hands	Activity monitoring feedback and suggestions (related to body parts)	JSON
UI Adaptation Engine	Adaptation space and elements	UI elements to be visualized (type, format, position, etc...)	JSON
Troubleshooting Assistance Service	A troubleshooting request (either textual or recorded voice)	It returns troubleshooting instructions as suggested by the AI engine. Also, it creates a new "case" in SHARE.	JSON (documented with Open API)
Smartwatch Connector (BLE Watch)	Heart rate	Clawdite	--
Kubernetes	Components manifest descriptions in yaml	XR5.0 components deployed in the XR5.0 central/cloud infrastructure	Defined in the official documentation

Table 5 – Explainable AI (XAI) and Machine Learning Technologies

Technology	Input	Output	Input/Output data format and samples
Active Learning	Labels dataset Human annotations to the model It will be a web service; the data transferred to	Trained model with better predictions and accuracy	--

	the Web Service would be in a JSON format		
NeuroSymbolic AI	Web Service that will take as inputs images / texts / videos. Recognizes and processes attributes such as color, shape, and size to make predictions.	Identifies and recognizes images, attributes, and produces results based on the embedded representations	JSON
Generative AI as a Service	Chat messages via OpenAI API Data formats JSON, CSV, PDF, TXT Proprietary knowledge for LLMs Use-case specific metadata for context User input/query in text form	Generated text responses to be rendered in XR	Chat messages via OpenAI API Data formats JSON, CSV, PDF, TXT Proprietary knowledge for LLMs Use-case specific metadata for context
Explainable AI (XAI)	Ontology/Knowledge graph interface for creating, editing, managing and querying ontologies/knowledge graphs Context extraction interface for loading and managing context models based on standardized data	Generates explanations contextualized to specific domains and knowledge	--
Integration and Visualization of AI/XAI	Input of 3D virtual content for rendering Data regarding user interactions within the XR environment Inputs from speech-to-text and text-to-speech services for AI API calls (i.e. json)	AI/XAI visualizations rendered in XR environments Rendering of virtual multimedia content including 3D models AI-driven recommendations displayed within the XR environment	--

Azure AI Services (Azure OpenAI)	Existing maintenance case data	Provides a platform for resolving support tickets and accessing knowledge resources (Service and machine documentation)	--
OpenAI API Services	Existing maintenance case data	Generated summary of case data	--
Documents processing/RAG Application	Documents processing: PDF docs, query params RAG: text query"	json response	--

Table 6 – Data Management and Analytics

Technology	Input	Output	Input/Output data format and samples
LeanXcaleDB	Historical data that can be ingested for further analytical processing or real-time data generated by the XR devices	Provides real-time analytical results while ensuring data consistency	Standard JDBC/ODBC Python drivers Integration with Spark, Flink, Kafka via the usage of custom connectors
AAS OPC-UA Connector	Robot coordinate system	Clawdite platform	--
AAS Pupil Labs API Connector	Eye movements	--	
MageAI ETL pipelines	query params	json response	--

3.4 Schematics and Internal Architecture/Design of each Component

In this section that concerns the Schematics and Internal Architecture/Design of each Component, the information is described for the different categories of technologies in XR5.0.

Hardware and Operating Systems

This schematic provides an overview of the Almer OS technology stack for AR devices, showing the various components and stakeholders involved, along with their functions (Figure 5).

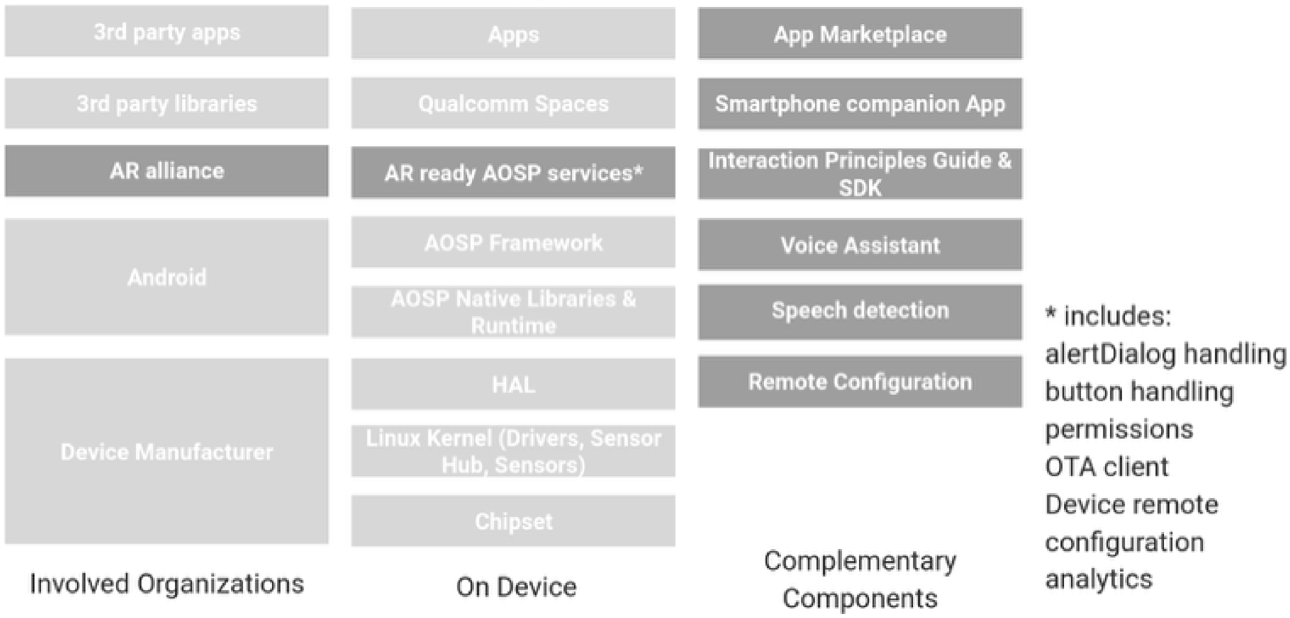


Figure 5 - Schematics concerning the Almer OS component

Almer OS Technology integrates hardware and software for a flexible, Android-based AR platform. This consists of partnerships with third parties and uses Qualcomm's AR technologies, creating a robust environment for developers. The architecture of the device also includes basic components for voice-control, remote configuration and a smartphone companion app that can make it feasible in a wide range of AR settings.

Augmented and Virtual Reality Platforms

The below figure (Figure 6) depicts a modular and integrated approach for the XR training environment using XR5.0 technologies. This training platform is scalable and adaptable to different training needs being a centralized interface for XR managing applications and accessing training resources. This platform will be accessible by a Custom Frontend Admin View providing an administrative view, allowing admins to monitor and configure various elements on the platform such as the available XR applications, Cloud Repository for accessing training assets, Training programs, and User management and Settings panels. API connections will be done by Backend APIs connect the Hololight Hub, Cloud Repository, and Training Assets & Programs modules to the main training platform, whereas 3rd Party Provider Authoring Tool also integrates with the platform, enabling content from external providers to be accessible within the training environment.

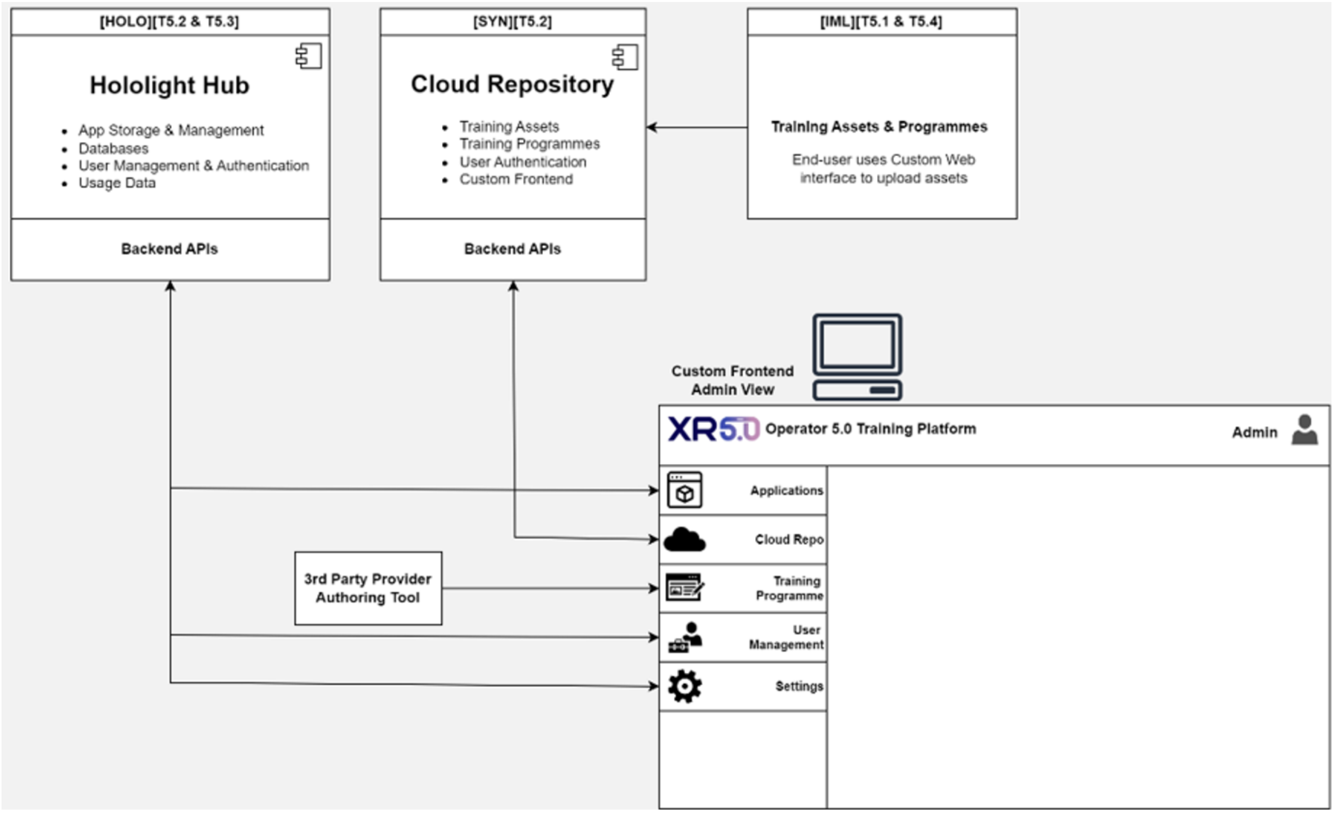


Figure 6 - Visual schema of the XR5.0 Operator 5.0 Training Platform

The architecture of the XR5.0 Operator 5.0 Training Platform integrates various components to deliver XR applications for training. This setup provides a cloud-based solution to deliver XR training applications, allowing resource-intensive XR experiences to run on lightweight XR devices. The centralized architecture enables scalability and AI driven models allowing to create a powerful training environment that leverages cloud computing and streaming technology. The components are the Human-Centric Digital Twin, AI Models and Cloud Repository (WP5): Stores training assets, applications, and related data, making them accessible for XR applications. An AWS Cloud Server allows multiple XR apps to operate in parallel and provides scalability to handle high-performance needs. The streaming will be done by the Hololight Stream, which streams XR applications directly from the cloud to XR devices, ensuring that computationally intensive tasks are processed remotely, while the XR device only handles the display and interaction. The Hololight Hub acts as the central hub for managing and storing applications. Figure 7 illustrates the architecture of the XR5.0 Operator 5.0 Training Platform.

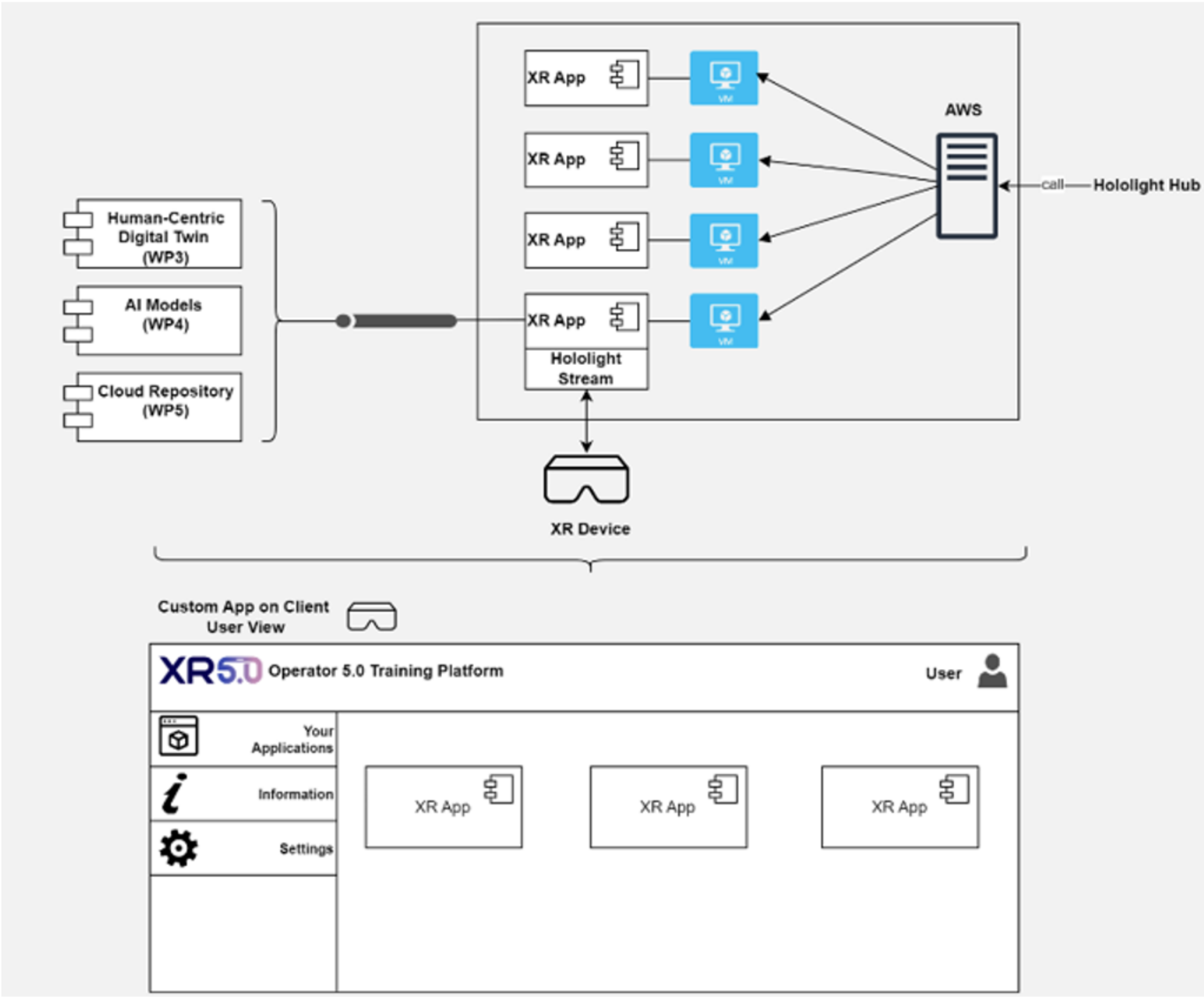


Figure 7 - Architecture of the XR5.0 Operator 5.0 Training Platform

The front end of the XR5.0 Training Platform comprises the Training Programs Authoring Tool (TPAT) that allows to manage and create the training programs from traditional training materials. These programs are then displayed in the XR environment using the XR Training Plugin (XRPT). Figure 8 depicts the diagram of the TPAT, whereas Figure 9 describes the XRPT.

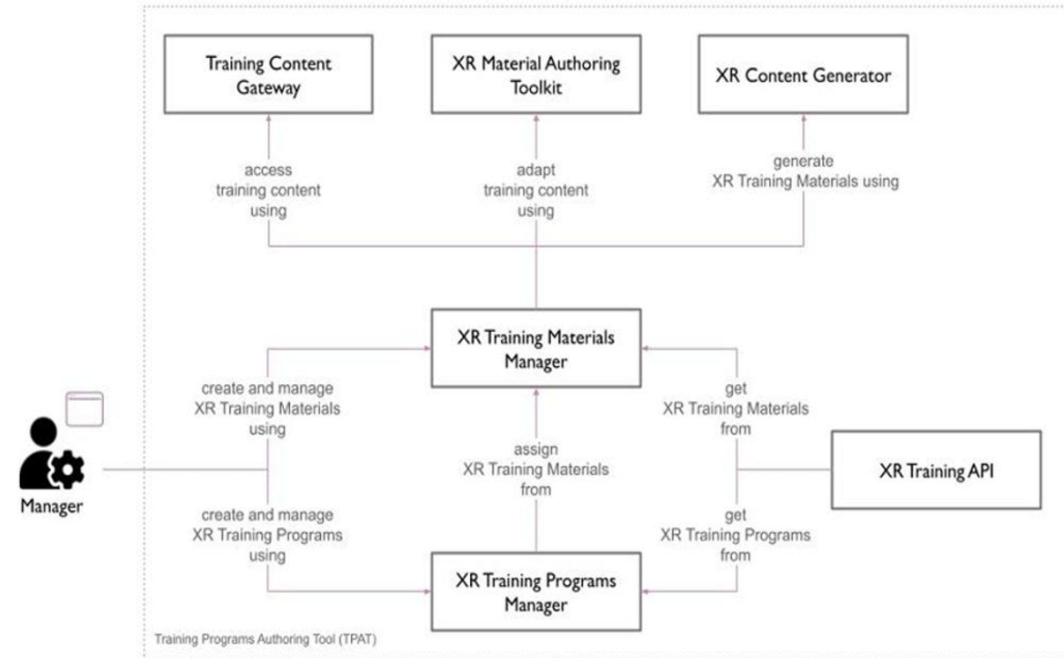


Figure 8 - Diagram of the Training Programs Authoring Tool

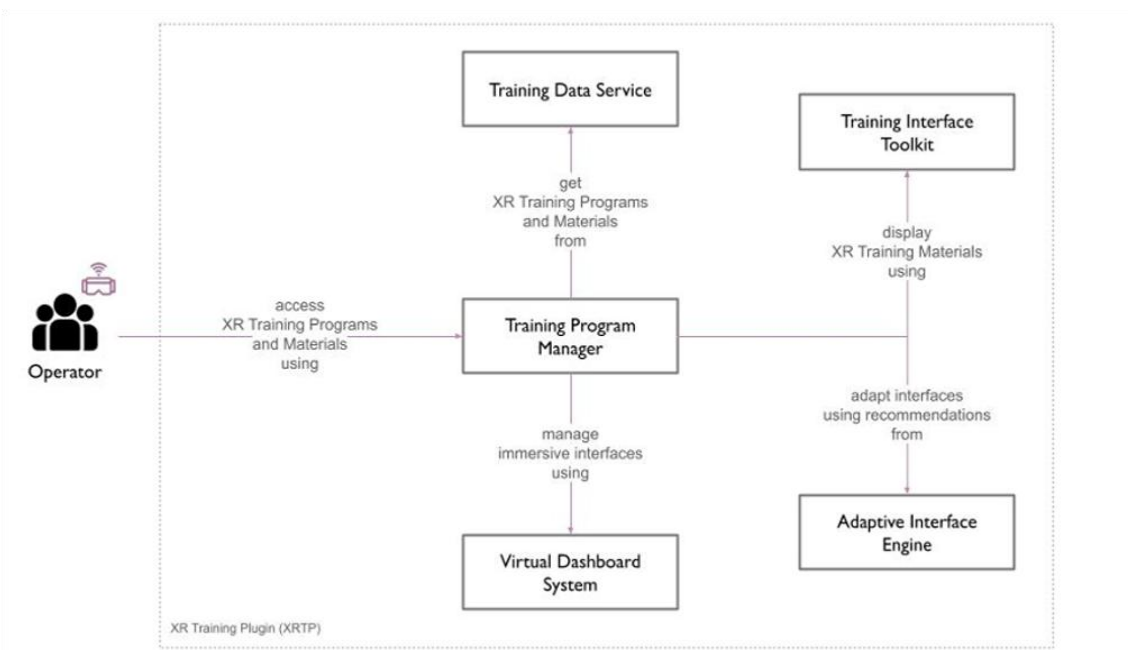


Figure 9 - Diagram of the XR Training Plugin

The general architecture of Oculavis SHARE is described in the following image. This provides a holistic view of the entire system, emphasizing multi-platform client integration and additional services such as 3D asset delivery (Figure 10).

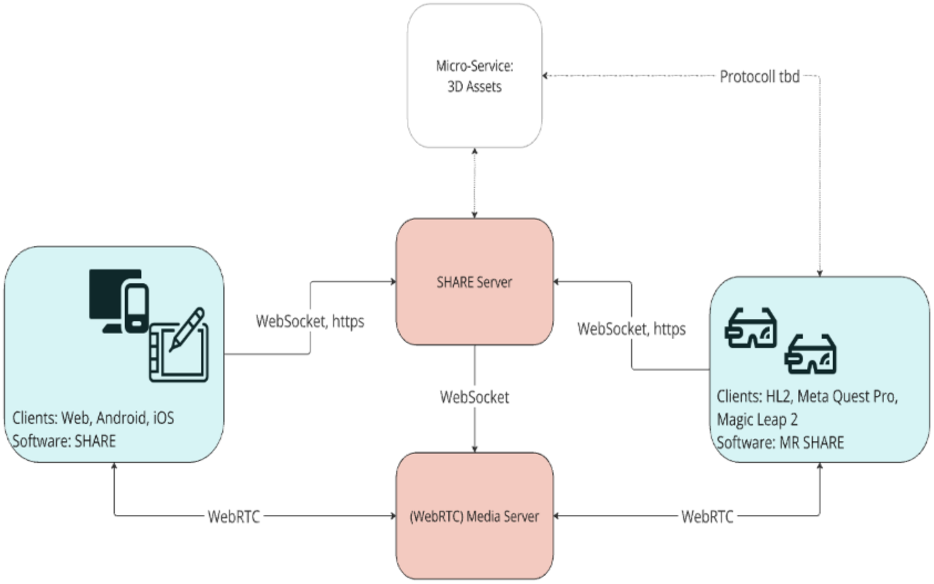


Figure 10 - General architecture of Oculavis SHARE

The Star Topology Call is an important component of Oculavis SHARE. The Star Topology Call Component is designed for a real-time communication system. It integrates multiple services to manage effectively peer-to-peer connections, media streaming, and signaling (Figure 11).

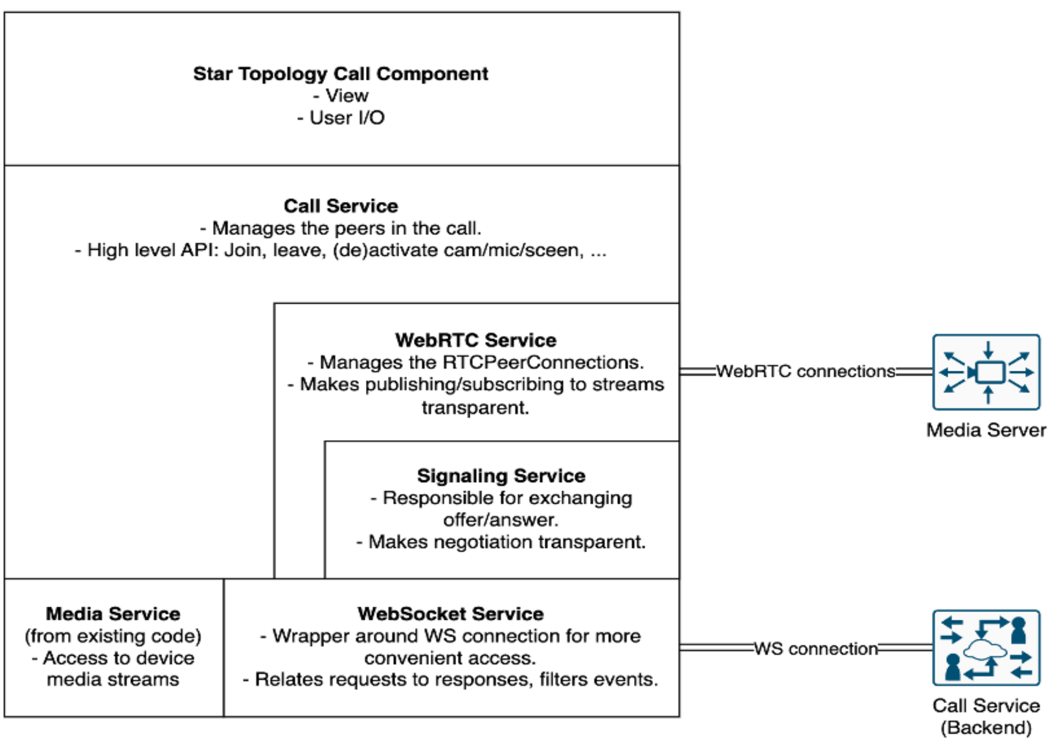


Figure 11 - Logic flow of the Star Topology Call Component

Smart Service, IoT, and IIoT Integration

The Clawdite platform architecture offers a modular and adaptable framework optimized for Industrial IoT use cases. It focuses on facilitating data exchange, across different industrial systems to ensure interoperability and scalability. Its core components are as follows: i) Gateways: Act as interfaces to stream data to the IIoT Middleware in predefined formats. They can be deployed on various devices like Raspberry Pi, smartphones, tablets, and PLCs; ii) IIoT Middleware: Manages and distributes data streams across architectural components, integrating different middleware solutions to suit specific application needs; iii) Historical Data Manager (HDM): Stores and retrieves historical data from Gateways and Functional Modules, enabling reporting and analytics since most middleware solutions lack this functionality; iv) Orchestrator: Oversees platform organization and management, including Digital Twin instances, installed modules, sensors, workers, and adopted message schemas; and v) Functional Modules: Pluggable external components that enhance the platform's capabilities (e.g., fatigue monitoring, worker movement prediction). They process data from the Middleware and Orchestrator and share the results with the platform. Figure 12 depicts these components.

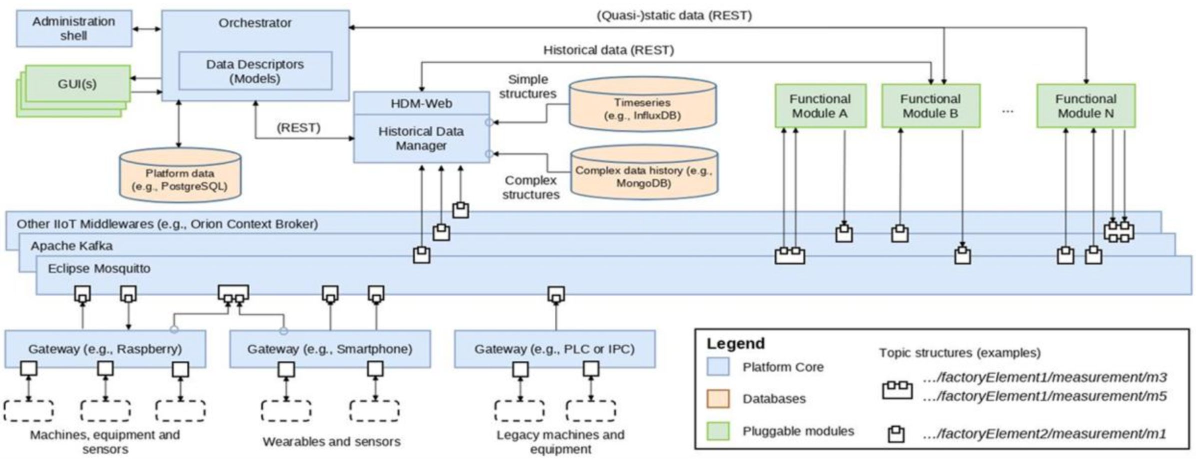


Figure 12 - Architecture of Clawdite

The Functional Modules realized within XR5.0 and plugged to the shared Clawdite instance, together with their related components, are depicted in the following figure (Figure 13).

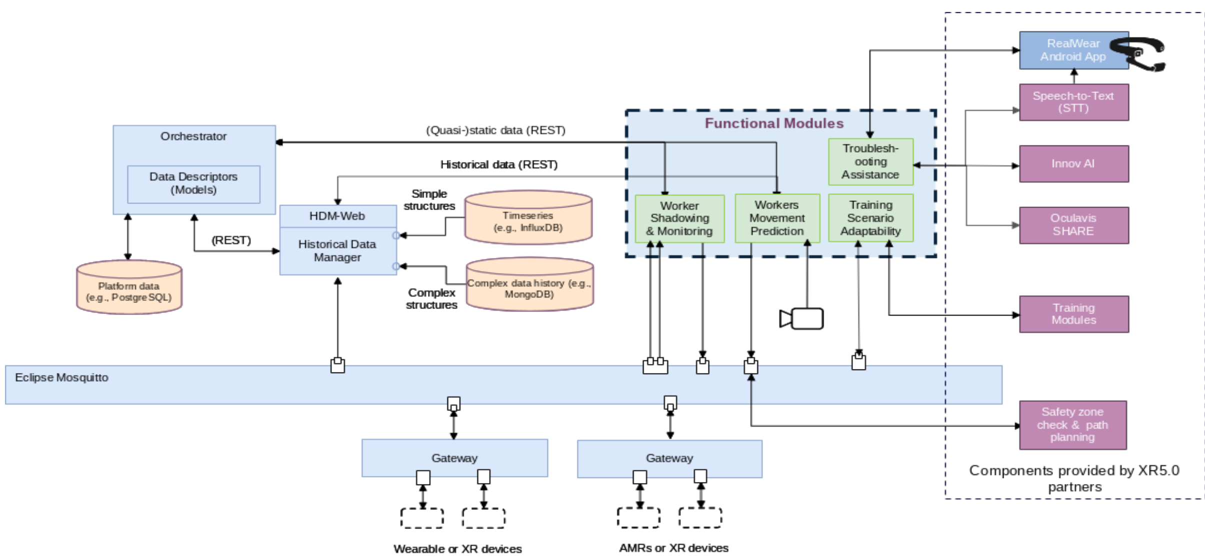


Figure 13 - XR5.0 Functional Modules within Clawdite architecture

The Troubleshooting Assistant Service is a Clawdite's Functional Module designed to enhance the troubleshooting process for workers. It integrates multiple modules (i.e., Innov AI Chat Engine, SIE Speech-to-Text, Oculavis SHARE) to provide real-time insights and support to workers, enabling them to resolve issues more efficiently. The diagram related to this module is depicted in Figure 14.

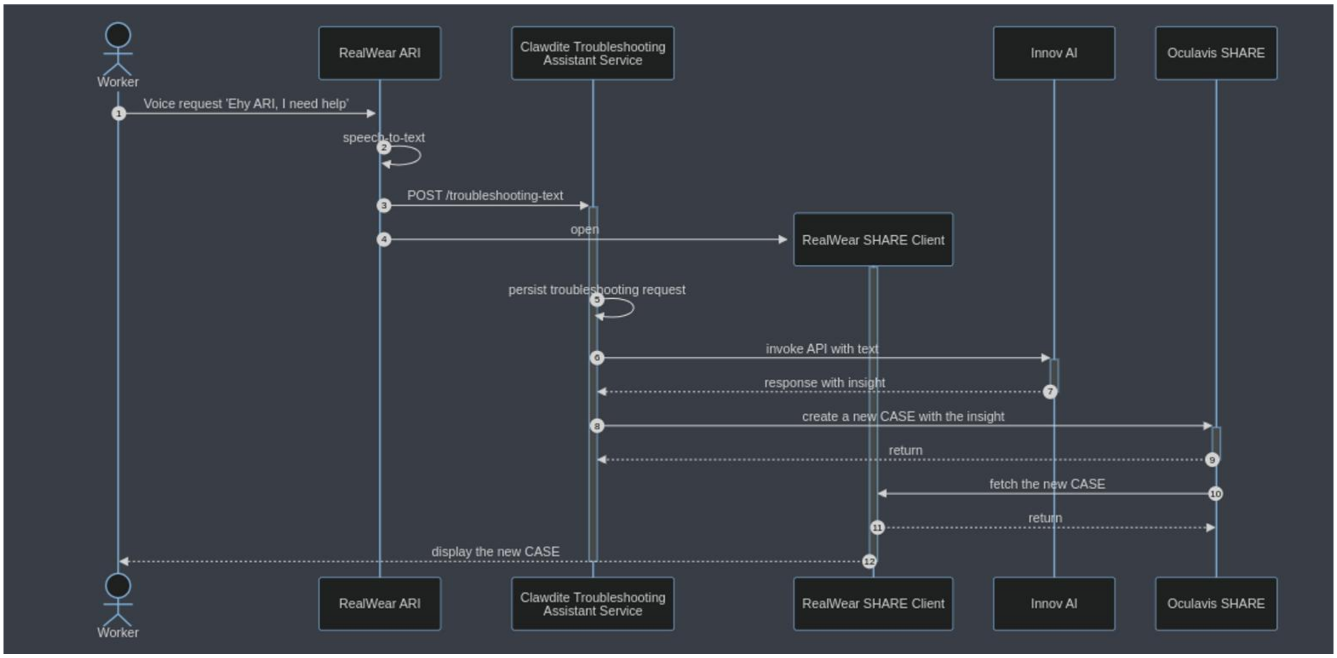


Figure 14 - Troubleshooting Assistant Service

Explainable AI (XAI) and Machine Learning Technologies

Generative AI as a Service joins expert domain-specific knowledge with advanced language processing to provide real-time and personalized context-driven responses. Figure 15 describes the architecture of this tool for XR5.0 project.

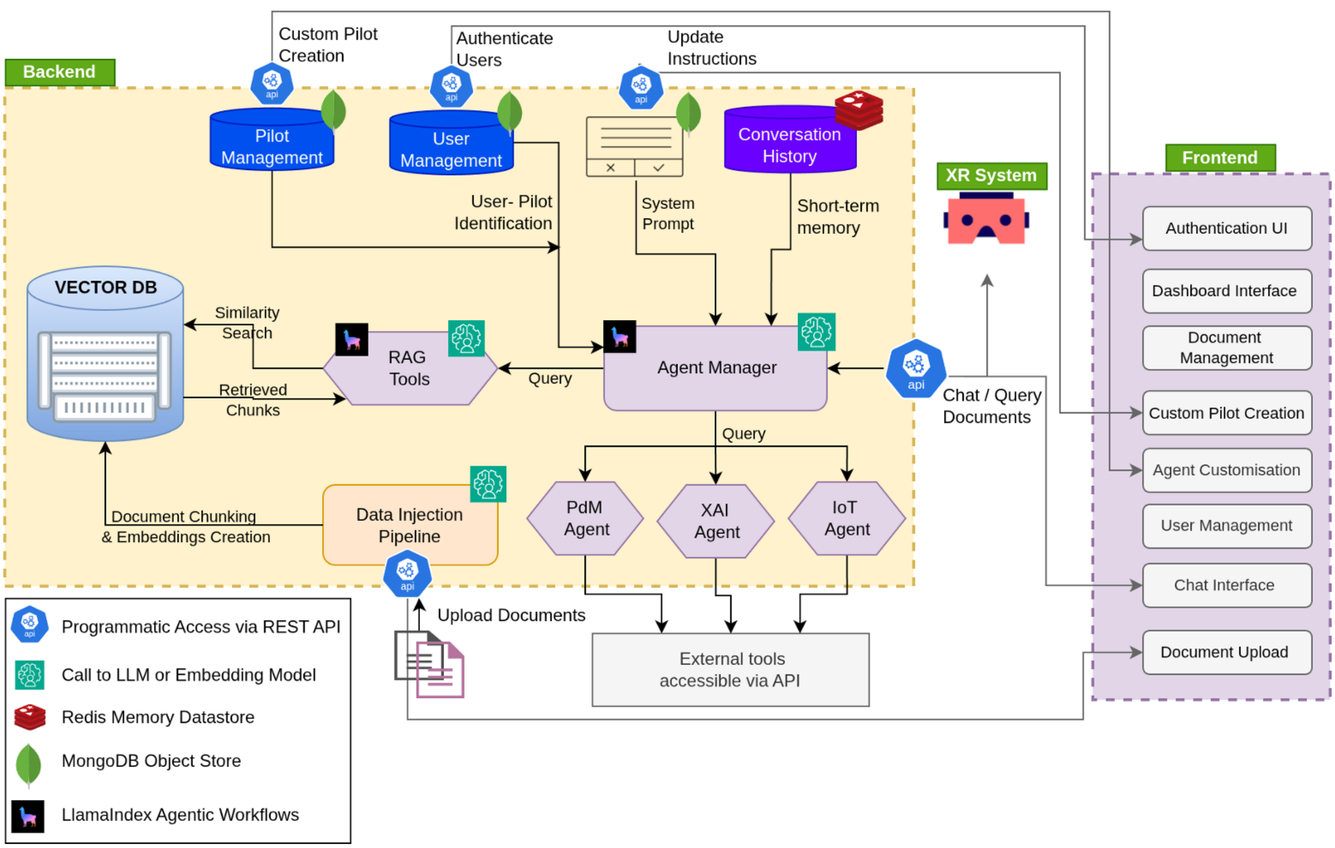


Figure 15 - Architecture of Generative AI as a Service

A potential application within XR5.0 project is for transforming user manual PDFs into an AI-accessible, quarriable format, allowing for interactive, context-aware responses to user questions. The integration of text chunking, embeddings, and a vector database ensures that the information is stored and queried efficiently. This diagram illustrates a workflow for processing a User Manual (PDF) and making it accessible through a query AI agent using OpenAI embeddings and a Chroma DB for document indexing and querying.

This diagram (Figure 16) illustrates a workflow for processing a User Manual (PDF) and making it accessible through a quarriable AI agent using OpenAI embeddings and a Chroma DB for document indexing and querying.

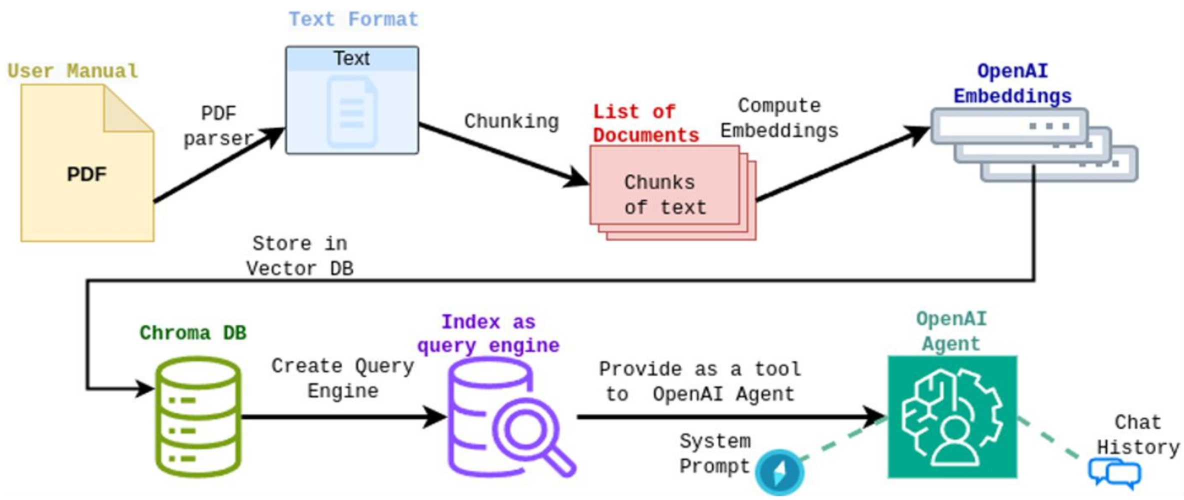


Figure 16 - Logic workflow for processing PDF text using Generative AI

Advanced AI paradigms are also used in the XR5.0 project, as illustrated in the figure, which shows how NeuroSymbolic AI and Active Learning are combined into a unified intelligent pipeline. Data from permanent storage is first prepared and then processed by NeuroSymbolic AI, where neural networks perform object and outlier detection while symbolic reasoning generates interpretable rules. Active Learning further enhances the system by incorporating uncertainty estimation, algorithmic feedback, labeled dataset preparation, and human expert interpretation, creating a continuous improvement loop that strengthens the accuracy of XR5.0 AI tools.

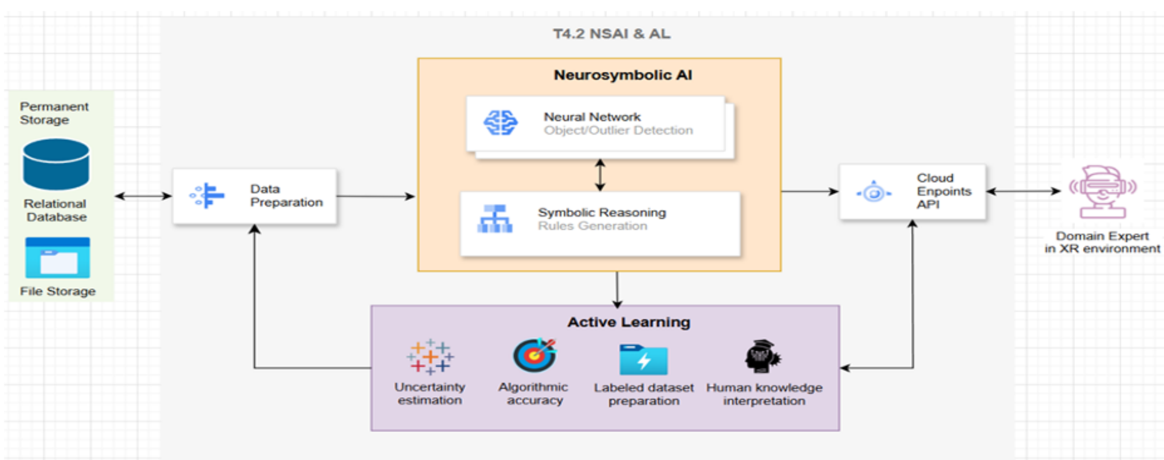


Figure 17 - NSAI and AL advanced paradigms

The documents processing/RAG is a fundamental tool to process the text information from PDF for the training platform, as it provides a complete pipeline for uploading, extracting, enriching, and indexing document contents for intelligent querying. As illustrated in the figure, documents are first received through dedicated API endpoints and converted into structured elements such as text, tables, and page images. These elements are cleaned, filtered, chunked, and enriched by an LLM before being formatted into JSON and versioned in S3 storage. The system then generates vector embeddings and stores them in a vector database, enabling semantic and metadata-driven retrieval.

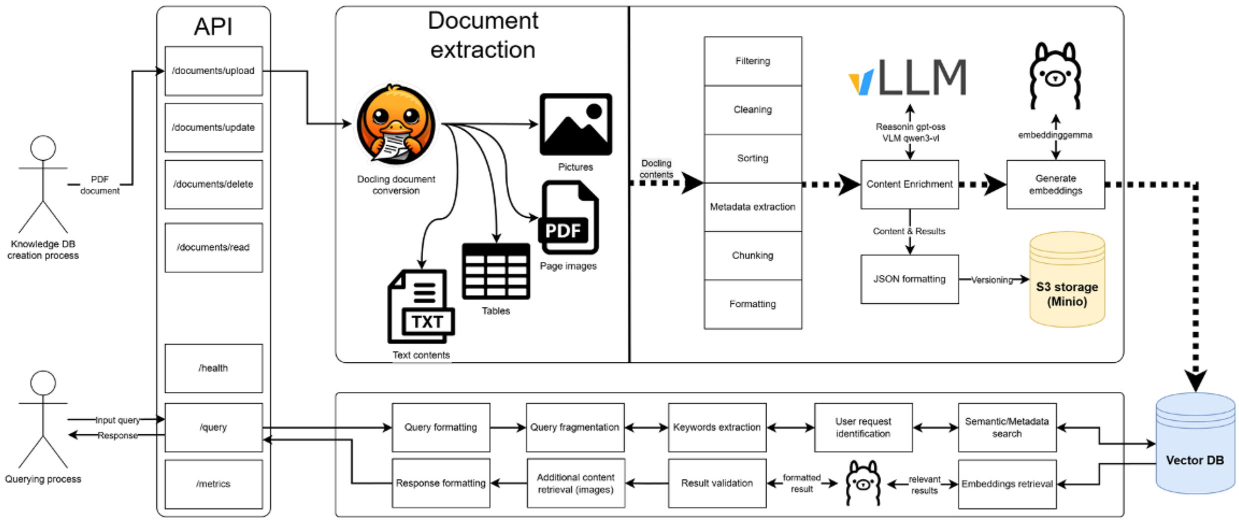


Figure 18 - Documents processing/RAG workflow

Data Management and Analytics

As for the LeanXcaleDB, the following diagram depicts its components, data flow, and integration with external tools and frameworks. The key components include the i) Data Lake that acts as the main repository for storing data; ii) Query Engines to handle data retrieval and processing, iii) Distributed Data Nodes to store portions of the data and enable distributed query execution; iv) Transactional Engine and Data Node Management to coordinate data node operations; v) Direct API providing unified interface for interacting with external tools and services; vi) Connectors for External Frameworks (i.e. Kafka, Spark, Flink, OData) as Data Sources, Spark Processing, Stream Processing and Mobile Apps.

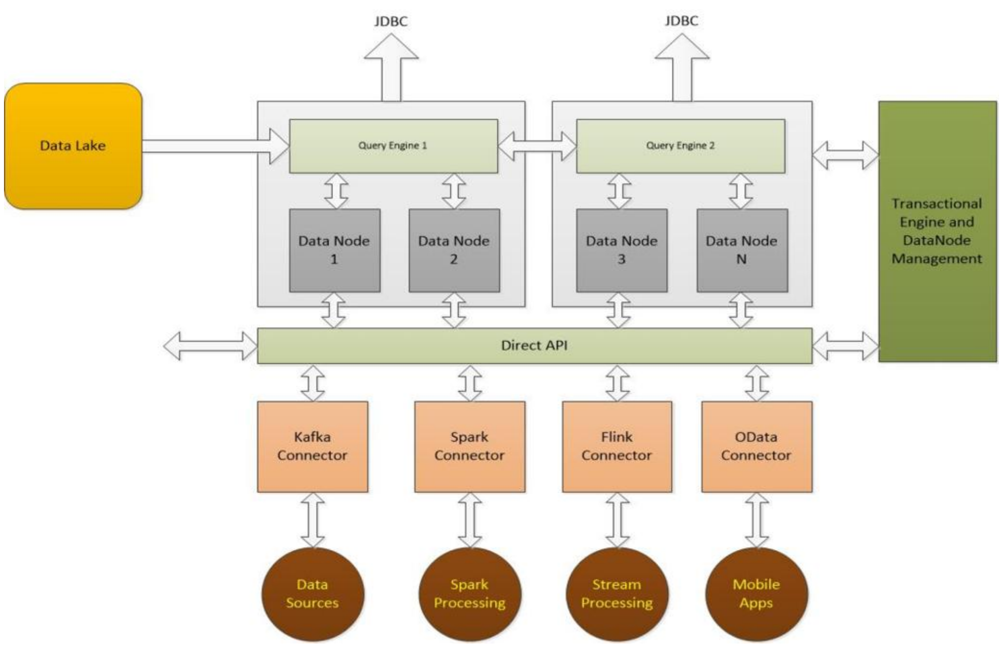


Figure 19 - Schema describing LeanXcaleDB

The ETL pipelines for data processing, formatting and provisioning include time-series anomaly detection and processing (e.g., aggregation and reduction) and geo-location calculation. The figure illustrates how sensor data flows through an anomaly-detection stage, where faults can be confirmed by an on-site technician, before triggering the ETL workflow. This workflow invokes the MageAI pipeline to transform and prepare the data, which is then exposed through an API endpoint for consumption by downstream XR applications.

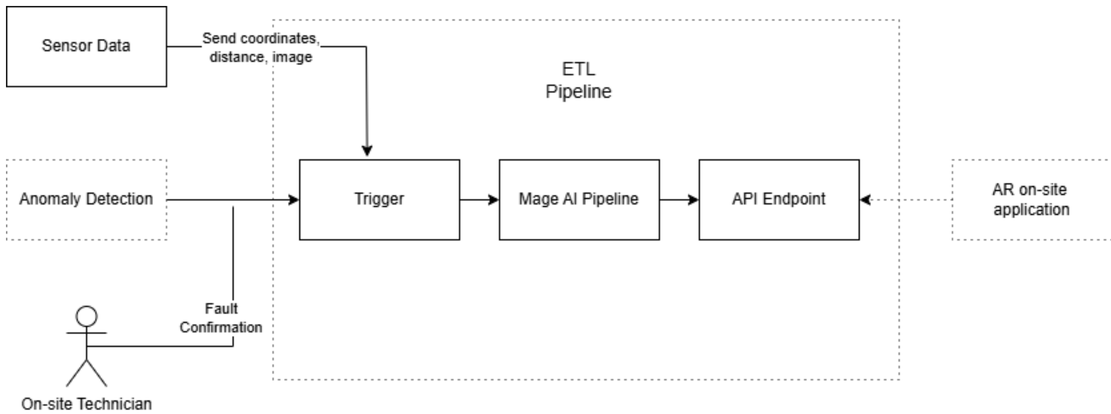


Figure 20 - ETL workflow

3.5 Software Requirements and Dependencies of each Component

Hardware and Operating Systems

The hardware and operating systems used in the project are complimentary, having unique needs and requirements to work appropriately. The Almer 2 Arc has no software or hardware dependency and only requires a stable Wi-Fi connection. Almer 2 can operate standalone without using any external applications or third-party software components. The Almer OS is designed to operate on AR hardware, such as the RealWear Arc or RealWear Navigator. For compatibility, the AR device must support Android 9 or higher. Since Almer OS relies on the Android operating system environment, the devices should be compatible with this OS specification and have AR capabilities. Beyond this, no additional dependencies are required. The Smartwatch connector has no specific software or hardware requirements or dependencies as well as the eye tracker.

Augmented and Virtual Reality Platforms

For the Augmented and Virtual Reality platforms, namely, TeamViewer Frontline, Hololight Hub, Hololight Stream, Hololight Space, and SLB for AMT training, these have specific requirements and dependencies that are described below. TeamViewer Frontline requires minimal setup, needing only web access and an AR device running at least Android 9. The Training Platform components like the TPAT and XRPT do not have specific requirements, just the hardware that is based on Meta Quest 3 devices. The Training Repository component Docker, OwnCloud, and MySQL/MariaDB. Docker provides a containerized environment. OwnCloud serves as the primary file storage and management platform, while MySQL/MariaDB acts as the database backend. The requirements for the Hololight technologies are described in Table 7.

Table 7 – Requirements for the Hololight technologies

Hololight Hub	**Hardware:** Device-specific client application for XR access. **Network:** A stable, low-latency internet connection. **Software:** Unity-based applications built with Hololight Stream SDK deployed on Microsoft Windows OS.
Hololight Stream	**Hardware** (Minimum): • OS: Windows 10, 11, or Server 2019 • Memory: 16 GB • CPU: Intel i5 8th Gen. (6 Cores) or AMD Ryzen 7 3700X • GPU: NVIDIA GTX 1070Ti or NVIDIA GRID for VMs • Storage: SSD or NVMe • Network (Minimum): • Network: 5 GHz Wi-Fi • Bandwidth: 20 Mbps • Latency: <100 ms **Development Environment**: • IDE: Visual Studio 2019 or 2022 with components for Universal Windows Platform, Unity, and C++ development. • Unity Version: 2021.3.x with Universal Platform Build Support and Windows Build Support IL2CPP. • MRTK Version: 2.7.x or 2.8.x. **XR End-Device**: Requires internet access and must support STUN (Session Traversal Utilities for NAT) and TURN (Traversal Using Relay around NAT) protocols.

Hololight Space	**Server Requirements:** • OS: Windows 10 20H2 or above • DirectX: Version 10 or above • Memory: 8 GB • Graphics Memory: 4 GB on a dedicated graphics card • Processor: x86, Quad-core **Client Requirements:** • OS: Windows 10 version 17763.0 or higher • Architecture: ARM **Connectivity:** • Hololight Stream Ports: Port 9999, UDP and TCP protocols. • WebRTC Ports: Ports 16384-32768 for SRTP/TURN, using UDP for incoming media streams. • Standalone Runtime Broker Ports: Port 17321, TCP for the Hololight Space double-click receiver.

Oculavis SHARE is a web-based platform designed for remote collaboration and maintenance tasks. It includes MR applications and dedicated apps for iOS and Android, being accessible across different devices. The supported devices are the following: i) Microsoft HoloLens 2 for immersive MR applications; ii) Magic Leap 2 for advanced AR functionalities; iii) Smartphones and Tablets where native apps are available for both iOS and Android platforms for mobile access; and iv) PC for accessing through web browsers for desktop use. The Ari - Smart Assistance and Oculavis SHARE have minimal or no dependencies, making these technologies easy to deploy, needing only internet access and an AR device running at least Android 9. The AI Voice Assistant is built in Python with a FastAPI backend, incorporating TTS Conqui for text-to-speech, Whisper for speech-to-text processing, and the Ollama client for large-language-model integration.

Smart Service, IoT, and IIoT Integration platforms

The Clawdite platform requires IIoT Middleware to enable connectivity with industrial IoT devices; a Historical Data Manager, Orchestrator; Functional Modules, where additional external components can be plugged in to provide specific functionalities. It also demands the following software requirements: Docker, for containerized deployment, enabling scalability and consistency, and NodeJS, for running JavaScript-based applications and services within the platform. The Workers Movement Prediction is developed using Python and relies on pyrealsense, OpenCV, MediaPipe, TensorFlow, Ultralytics, Supervision, and paho-mqtt for data processing, model inference, and communication. The Troubleshooting Assistance Service and Worker Shadowing and Monitoring are developed in Python with a FastAPI framework and packaged as a Docker container for scalable and portable deployment, relying on paho-mqtt for data exchange.

The Training Scenario Adaptability and the UI Adaptation Engine modules are currently being designed, and no information is yet available at the time of writing this deliverable. The Smartwatch Connector (BLE Watch) The Smartwatch Connector (BLE Watch) operates using Bluetooth Low Energy (BLE) and connects through an Android phone. In addition, all components intended for deployment on Kubernetes must be packaged as virtual Docker images to ensure proper orchestration and scalability.

Explainable AI (XAI) and Machine Learning Technologies

Most components within this suite are designed with minimal or no specific requirements. ATB's Explainable AI (XAI) requires Docker, as the tool is developed in Java and packaged within a Docker container, ensuring compatibility across different environments. The Generative AI as a Service is designed as a RESTful API and documented through OpenAPI 3.0 specification to ensure standardized integration and interoperability. The Documents Processing/RAG application is built using Python and FastAPI, leveraging Torch for machine learning tasks, Docling for document parsing and processing, the Ollama client for LLM inference, Chroma for vector-based retrieval, and MinIO as an S3-compatible storage backend.

Data Management and Analytics

The LeanXcaleDB has no specific requirements, being a flexible component not requiring additional software or hardware. The AAS OPC-UA Connector operates within a Docker runtime environment, ensuring a consistent and portable set-up across different systems. The AAS Pupil Labs API Connector also runs in a Docker runtime environment, but in addition requires the Pupil Labs software to be installed to enable device communication and data streaming. The MageAI ETL pipelines are implemented using Python and integrate with a PostgreSQL database to support data extraction, transformation, and loading operations.

3.6 Authentication Mechanisms supported by each Component

Hardware and Operating Systems

Almer 2 Arc relies on unique device-based identities managed through MongoDB Realm. Each RealWear Arc device has its own identity, which is independent of the end user, ensuring device-level authentication. Wearable OS employs Bluetooth GATT (Generic Attribute Profile) for authentication, supporting low-energy, device-specific connections.

Augmented and Virtual Reality Platforms:

TeamViewer - Frontline uses custom authentication tailored to the platform's specific requirements. Hololight Hub integrates OAuth JWT for secure token-based authentication and supports Microsoft Single Sign-On (SSO) for enterprise-level access. The development of the XR5.0 Training Platform modules is still in place as well as the definition of the authentication process for this platform. The Training Repository uses Auth0 to manage authentication and authorization; JWT (JSON Web Tokens) a token-based authentication mechanism; Username/Password; or None indicating that no authentication mechanism is implemented or required for specific components.

Smart Service, IoT, and IIoT Integration:

Ari - Smart Assistance does not require authentication, as it is a service that runs directly on the device and operates independently of user-specific credentials. Oculavis SHARE uses Microsoft SSO authentication enabling users to authenticate using their existing Microsoft accounts. Clawdite and Service Connector Core do not require authentication.

Explainable AI (XAI) and Machine Learning Technologies:

Generative AI as a Service employs Auth0 for identity management, JWT for secure token-based authentication, and standard Username/Password credentials for access control. Integration and Visualization of AI/XAI CYENS' technology incorporates authentication mechanisms to ensure secure access, including the following: API Authentication – OAuth (Open Authorization) to allow secure, token-based access to the XAI component's APIs; and Token-based authentication for session management and secure user access. These mechanisms work together to protect data integrity and confidentiality while supporting integration into other project frameworks. Active Learning, NeuroSymbolic AI and ATB's Explainable AI (XAI), and do not require authentication.

Data Management and Analytics

LeanXcaleDB component complies with the standard authentication mechanisms defined in JDBC/ODBC protocols for secure and reliable access to the system. For MageAI ETL pipelines the authentication process is still being defined as the XR5.0 evolves.

4. THE ARCHITECTURE DESIGN METHODOLOGY WITHIN TOGAF

The Open Group Architecture Framework (TOGAF) is an established, widely used methodology for developing, managing, and governing enterprise architecture. Initially developed by The Open Group in the mid-1990s, TOGAF has evolved into one of the most recognized and adaptable frameworks for aligning business and IT strategies [6]. Its structured approach provides a comprehensive methodology for designing, planning, implementing, and maintaining enterprise architectures that align IT investments with business goals, enabling organizations to be more agile and responsive in a rapidly changing environment.

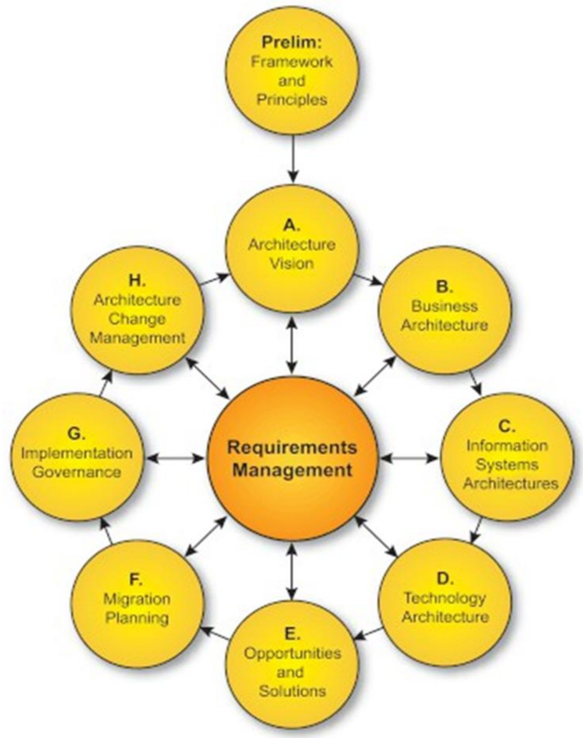


Figure 21 - TOGAF Architecture Development Cycle

TOGAF has nine ADM phases of iterative development. This ADM cycle according to TOGAF is depicted in Figure 21. Briefly, the phases can be categorized as follows:

- ➢ The first category consists of the preliminary and phase A, phases that set out the key issues of the problem and serves as a starting point to create the enterprise architecture team and for establishing the vision of the Architecture.
- ➢ The “development” category, which consists of phases B to D. These are phases in which the requirements engineering takes place and in which the various aspects of the architecture are implemented.
- ➢ The “transitional” category, which consists of phases E and F and concerns transitioning and changes of the architecture.
- ➢ The governance category, which establishes how changes in architecture are implemented and how it is maintained.

Not all, but 5 ADM phases were implemented in XR5.0. The following sub-chapters outline the application of TOGAF’s ADM phases in the XR5.0 project, detailing the structured development of Business, Application and Technology layers to achieve Industry 5.0 goals. It is highlighted how foundational principles, stakeholder engagement and iterative design ensured a scalable and integrated architecture leveraging XR and AI technologies.

4.1 Preliminary phase - Framework and principles

The Preliminary Phase establishes the foundational elements necessary for guiding the architectural development process. For this project, this phase involved defining the scope of the architecture, identifying key stakeholders, and aligning with organizational objectives and industry standards.

Key activities included:

- ➢ **Establishing the Architecture Vision**: Ensuring alignment with Industry 5.0 goals to enhance operator effectiveness through XR and AI technologies.
- ➢ **Defining Architecture Principles**: Emphasizing modularity, scalability, and interoperability to support evolving needs and integration across layers.
- ➢ **Identifying Key Stakeholders**: Addressing the needs of operators, training managers, remote experts, and developers as central to architectural design.
- ➢ **Laying the Framework**: Selecting a structured approach incorporating the business, application, and technology layers for iterative development and alignment with TOGAF principles.

This phase provided the necessary framework and principles to ensure the architecture is robust, adaptable, and aligned with project objectives, setting the stage for detailed development in subsequent phases.

4.2 Phase A - Define the architectural vision

Within TOGAF, Phase A serves to define the architecture scope, include stakeholder concerns and establish any hard constraints imposed by any internal (technical) or external source (organizational or legal). It is within Phase A that the adherence to the vision is continuously adhered to.

In XR5.0, we made sure that the vision was well defined in both design and implementation. Moreover, there was a strong effort to meet requirements being collected and discussed with the pilots before any major design decisions were made.

4.3 Phase B through D - Define the Business, the System and the Technical Architecture

This chapter addresses the detailed architectural development undertaken during the **Business Architecture (Phase B)**, **Application Architecture (Phase C)**, and **Technology Architecture (Phase D)** phases of the TOGAF ADM cycle for the XR5.0 project.

Business Architecture

In Phase B, the project identified the key business functions and services needed to support operators in an Industry 5.0 environment. These include training services, maintenance assistance, and real-time monitoring services. Business services such as **Personalized Training Service**, **Predictive Maintenance Service**, and **Production Line Monitoring Service** were defined to meet the operational goals of efficiency, safety, and skill enhancement. These services are underpinned by business functions like **Visualization of Augmented Information** and **Fusion of Data and Services into XR Environments**, which access critical business objects such as IoT data, ERP data, and training workflows.

Application Architecture

Phase C focused on mapping the application components and services needed to realize the defined business architecture. The project designed key application components such as:

- **XR Apps**, including training and assistance plugins managed by the Central XR Hub
- **AI Models**, enabling advanced capabilities like generative AI, defect recognition, and process optimization.
- **Cloud Repository**, offering storage and retrieval functionalities for training materials and

workflows.

These components collectively deliver services like **XR Visualization Service** and **Data, Image, and Video Processing Service**, ensuring seamless integration of XR and AI technologies into the operator's workflows.

Technology Architecture

In Phase D, the technical foundation supporting the application and business layers was established. A **Kubernetes Cluster** hosts the applications, with automated deployment pipelines leveraging CI/CD principles. The **Virtualization Platform** provides scalable resource provisioning for virtual machines, while communication networks ensure connectivity between external stakeholders and internal systems. Services such as **Application Deployment Service** enable efficient and automated operations across the architecture.

Together, the business, application, and technology layers are integrated to deliver a comprehensive and scalable architecture. This alignment ensures that the XR5.0 framework addresses the project's objectives while remaining adaptable to future enhancements.

5. ENTERPRISE ARCHITECTURE

Enterprise Architecture serves as a structured approach to align an organization's business goals, application systems and technology infrastructure, ensuring that all components work cohesively to achieve strategic objectives. In this project, Enterprise Architecture is leveraged to design and implement the XR5.0 framework, a cutting-edge solution that integrates AI and XR technologies within Industry 5.0 environments.

The architectural design follows the TOGAF ADM framework [6] introduced in chapter 4 and is represented using ArchiMate [3], a standardized modeling language that facilitates clear and consistent communication between stakeholders. ArchiMate's layered approach ensures that the business, application and technology dimensions are comprehensively addressed, while its viewpoints enable focused exploration of specific concerns, such as organizational roles or value streams. It enables a structured and visual representation of complex systems, and the language supports a layered approach, with the Business Layer capturing the services, processes and functions that deliver value to stakeholders; the Application Layer focusing on the software systems and data flows that support business operations; and the Technology Layer addressing the physical and virtual infrastructure needed to deploy and operate the applications. It also includes a variety of viewpoints, such as the organizational, value and motivation viewpoints, which enable stakeholders to examine the architecture from different perspectives. This flexibility ensures that the architecture remains comprehensive and adaptable to evolving requirements. In this chapter, ArchiMate diagrams are used to convey the XR5.0 architecture in a structured and transparent manner, demonstrating the alignment between Industry 5.0 business goals, innovative applications and the supportive technical infrastructure. This document will not cover ArchiMate in detail, however the reader is encouraged to explore its online documentation [3] or study Annex I which contains a brief explanation of the various ArchiMate elements that were leveraged in order to design the various layers of the XR5.0 in this chapter.

To this end, this chapter presents the Enterprise Architecture of the XR5.0 project, starting with a high-level vision (Chapter 5.1) and an organizational perspective (Section 5.2). Subsequently, the layered architecture is detailed in section 5.3, where the Business, Application and Technology layers are described, emphasizing their interactions and the realization of Industry 5.0 services.

Through this structured design, the architecture ensures that business requirements are effectively realized through robust applications and reliable technical infrastructure. The use of ArchiMate models supports the clarity and traceability of design decisions, enabling future iterations to build upon this initial framework.

5.1 Vision of the Architecture

The architecture is designed to align with XR5.0 strategic objectives, ensuring efficient delivery of value to all stakeholders. The Vision of Architecture provides a high-level overview of how key processes, capabilities and technologies work together to achieve the desired outcomes. This vision serves as the foundation for the subsequent design and implementation phases.

The value stream diagram included in this section illustrates how value is created and delivered, offering a clear understanding of the flow of activities and the enabling capabilities. The value stream diagram is a critical tool that visualizes:

- ➢ The stages where value is created or transformed within the project.
- ➢ The XR5.0 capabilities and resources required at each stage.
- ➢ The flow of value from initial inputs to final delivery to stakeholders

By providing a holistic view, the diagram ensures that the architecture remains focused on delivering value efficiently and effectively. For better comprehension of the diagram in figure 22, we provide the following definitions which are directly aligned with both ArchiMate's modelling language and TOGAF.

- **Activity (represented by an arrow, indicating the flow of task):** The specific tasks or actions performed at each stage of the value stream to transform inputs into outputs. The activity describes what is being done to create or deliver value.
- **Capability (depicted as a rectangle with rounded corners):** The skills, processes, systems or resources required to perform the relevant activity effectively. A set of capabilities ensure that XR5.0 can support the relevant activity and deliver the intended outcomes.
- **Value (represented by a rectangle with an ellipse inside):** The benefit or outcome delivered to XR5.0 stakeholders (training managers, operators, trainees, etc.) at each stage of the value stream. Value represents the reason the activities are performed, and the capabilities are leveraged.
- **Outcome (represented by a rectangle that includes an arrow pointing towards a target):** This element shows the focus on achieving the desired outcome, which is in XR5.0 the following: "Provision of a full-fledged XR and AI environment for improving Industry 5.0 operations"

According to Figure 22, the key stages of XR5.0 value stream are the following:

- **Stage 1:**
  - **Activities:** Provision of Augmented Information
  - **Capabilities Involved:** Creation of Digital Twins, Integration with IoT sensors and ERP systems, Defect Recognition, Product Variant Detection
  - **Value Delivered: Improve** efficiency and reduce onboarding time
- **Stage 2:**
  - **Activities:** Provision of Immersive Training
  - **Capabilities Involved:** Creation of Training Workflows through Generative AI, Rendering of Training Workflows in XR Environment, Conversion of traditional training assets into XR assets, Provision of environment for manually creating Training Workflows
  - **Value Delivered:** Boost training engagement and reduce failure
- **Stage 3:**
  - **Activities:** Step By Step Guidance
  - **Capabilities Involved:** Provision of a chatbot that guides the operator, recognition of technical components, provision of step-by-step instructions concerning maintenance procedures
  - **Value Delivered:** Increase efficiency and procedural accuracy, reduce complexity of assembly lines
- **Stage 4:**
  - **Activities:** Provision of Remote Assistance
  - **Capabilities involved:** Permission control, provision of industry internal data to remote expert, establishment of remote connection
  - **Value Delivered:** Improve troubleshooting and reduce on-site service
- **Stage 5:**
  - **Activities:** Provision of handsfree digital assistance 24/7

- **Capabilities involved:** Automatic production of Training Workflows, adaptation of maintenance tasks to the operator stress level, 24/7 support for maintenance and troubleshooting
- **Value Delivered:** Increase sustainability and improve troubleshooting

➢ **Stage 6:**

- **Activities:** Personalization
- **Capabilities involved:** Make use of personalized operator's feedback, optimization of human-machine interaction, processing and integration of operator's biometric data, adaptation of training workflows based on biometrics

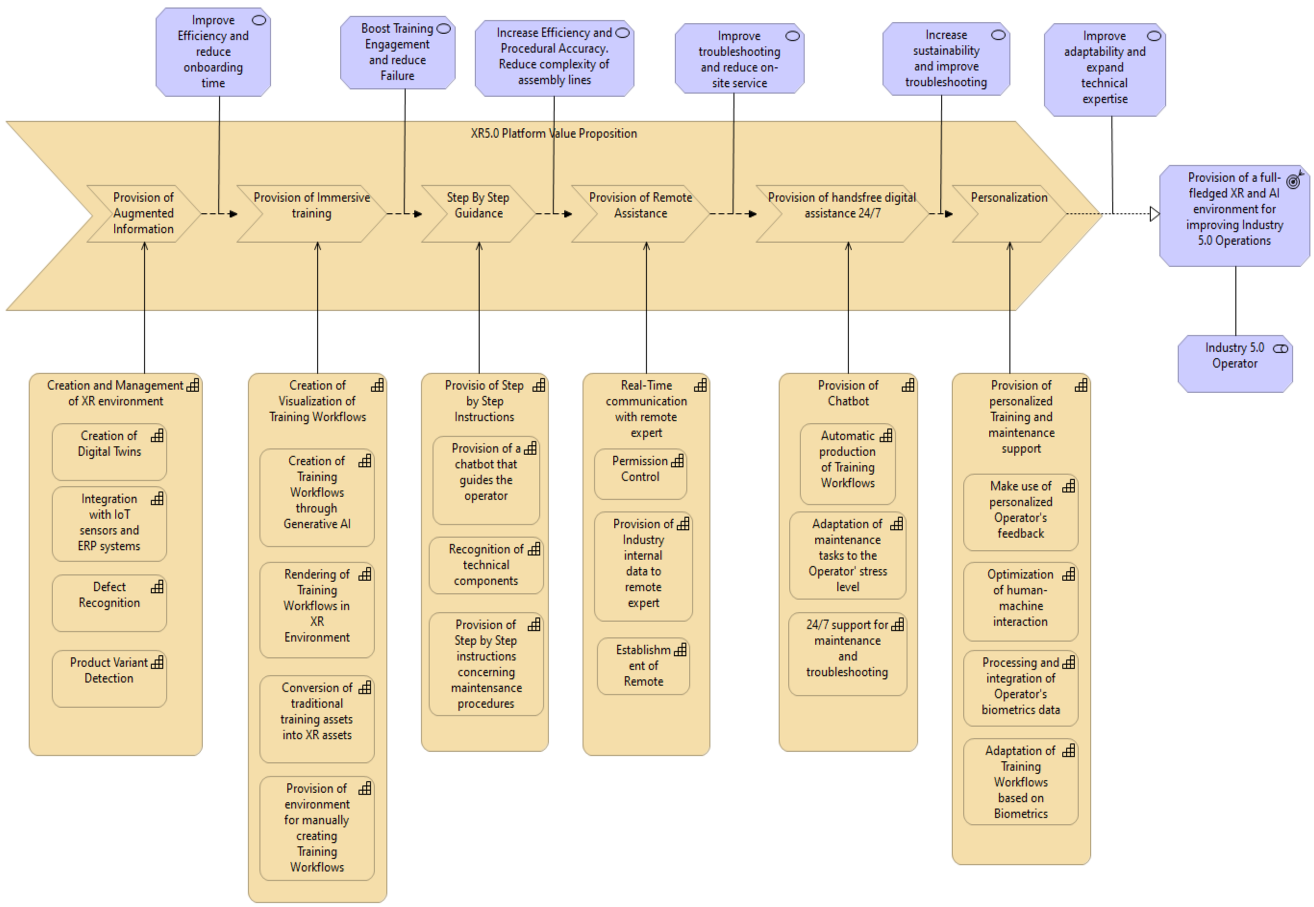


Figure 22 - Vision of the XR5.0 Architecture (Value Stream- Strategy layer)

The vision of the architecture sets a clear direction for how this project will deliver value, ensuring alignment between XR5.0 strategic goals, operational processes and supported technologies. The value stream diagram highlights the flow of value and provides a guide for deeper analysis in subsequent chapters. In the next chapters we will explore the specific layers of architecture, namely business, application and technology layers.

5.2 Organization Viewpoint

The Organization Viewpoint focuses on visualizing the organizational structure of an enterprise. It is used to define and analyze the roles, responsibilities, collaborations and hierarchical relationships within an organization. The Organization Viewpoint is particularly valuable for addressing concerns about how the enterprise is structured and how different organizational entities interact. In this document, the Organization Viewpoint models and communicates the organizational structure of typical Industry 5.0

organization based on the feedback provided by the pilots as stated in Deliverable D2.1[2]. The Organization Viewpoint in XR5.0 is shown in Figure 23, and the key characteristics are the following:

- **Purpose of the diagram:** To model and communicate the organizational structure in Industry 5.0 settings. Also designing, deciding and informing.
- **Concerns:** Organizational hierarchy and roles, identification of competencies
- **Elements included:** Business Actors
- **Scope:** Single Layer/Single Aspect
- **Stakeholders:** Managers, employees, shareholders

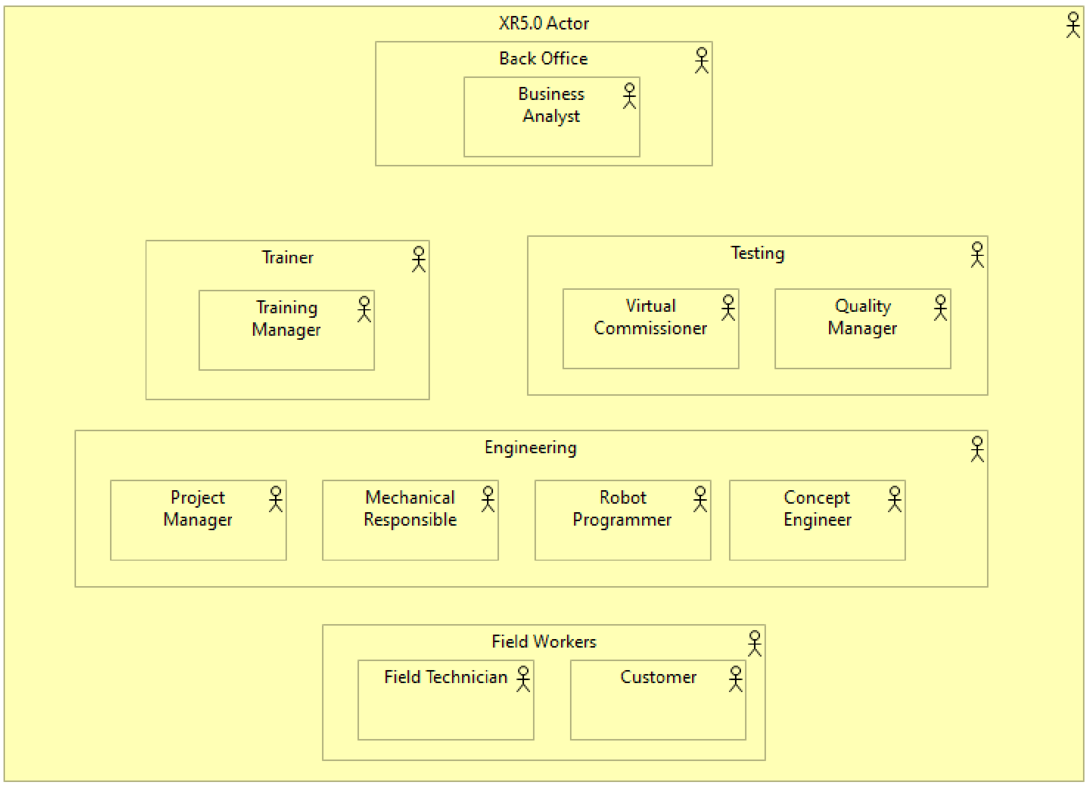


Figure 23 - Organization Viewpoint of the Architecture

The interactions of those various actors with the XR5.0 system are described in chapter 6 of the Solution Architecture and more specifically, they are depicted in the System Context diagram.

5.3 Layers

The architecture of XR5.0 is structured using a layered approach, aligning with the principles of the ArchiMate framework. This approach ensures clarity, modularity and consistency across the various aspects of the system, allowing stakeholders to easily understand and analyze the relationships between needs, application capabilities and technological infrastructure.

The layers in this architecture serve distinct but interconnected purposes:

- **Business Layer**: This layer captures the high-level services, processes and functions that deliver value to stakeholders. It focuses on the activities and objectives of the operator and other roles involved in the XR5.0 ecosystem, emphasizing how the system supports training, maintenance and other business outcomes.

- **Application Layer**: This layer focuses on the software systems and their interactions, ensuring that the business services are effectively supported by well-defined application services, components and data flows. It includes innovative AI models, XR applications and other key components that bridge business needs with underlying technology.
- **Technology Layer**: This layer defines the physical and virtual infrastructure required to support the application layer. It includes elements such as virtualization platforms, communication networks and deployment services, ensuring the seamless operation of the applications and alignment with business goals.

By organizing the architecture into these layers, the result is having a holistic view of the XR5.0 system. Each layer plays a crucial role in achieving the overall vision while maintaining traceability and alignment between XR5.0 strategic objectives, implemented solutions and supporting infrastructure.

5.3.1 Business Layer

The business layer of the architecture presents the foundational services and workflows that empower the operator in an Industry 5.0 environment, leveraging cutting-edge XR and AI technologies. This layer bridges the needs of industrial operators with advanced technological solutions, ensuring efficient training, maintenance, and production monitoring.

The Business Layer diagram includes key ArchiMate elements that represent essential components of business architecture (see also Annex I):

- **Business Function**: Represents a specific activity or operation performed within the business, depicted as a yellow rectangle with an arrow inside pointing on the top.
- **Business Process**: A sequence of tasks or activities that together achieve a specific outcome, typically shown as a yellow rectangle with an arrow inside pointing on the right.
- **Business Object**: Represents information or data used or produced by business processes, shown as a yellow rectangle which holds another rectangle inside.
- **Junction**: Used to indicate decision points or paths where multiple flows converge or diverge, represented by a circle. (circle with black inside is an “and” junction and circle with white inside is an “or” junction)
- **Business Service**: Represents a service provided by the business to external stakeholders, shown as a yellow rectangle with ellipsis inside
- **Actor**: Depicts an entity, typically an individual or role, that interacts with the business, represented by a yellow human-like icon

As shown in figure 24, the business layer is enabling the operator to perform tasks more effectively through enhanced visualization, real-time assistance, and streamlined data integration.

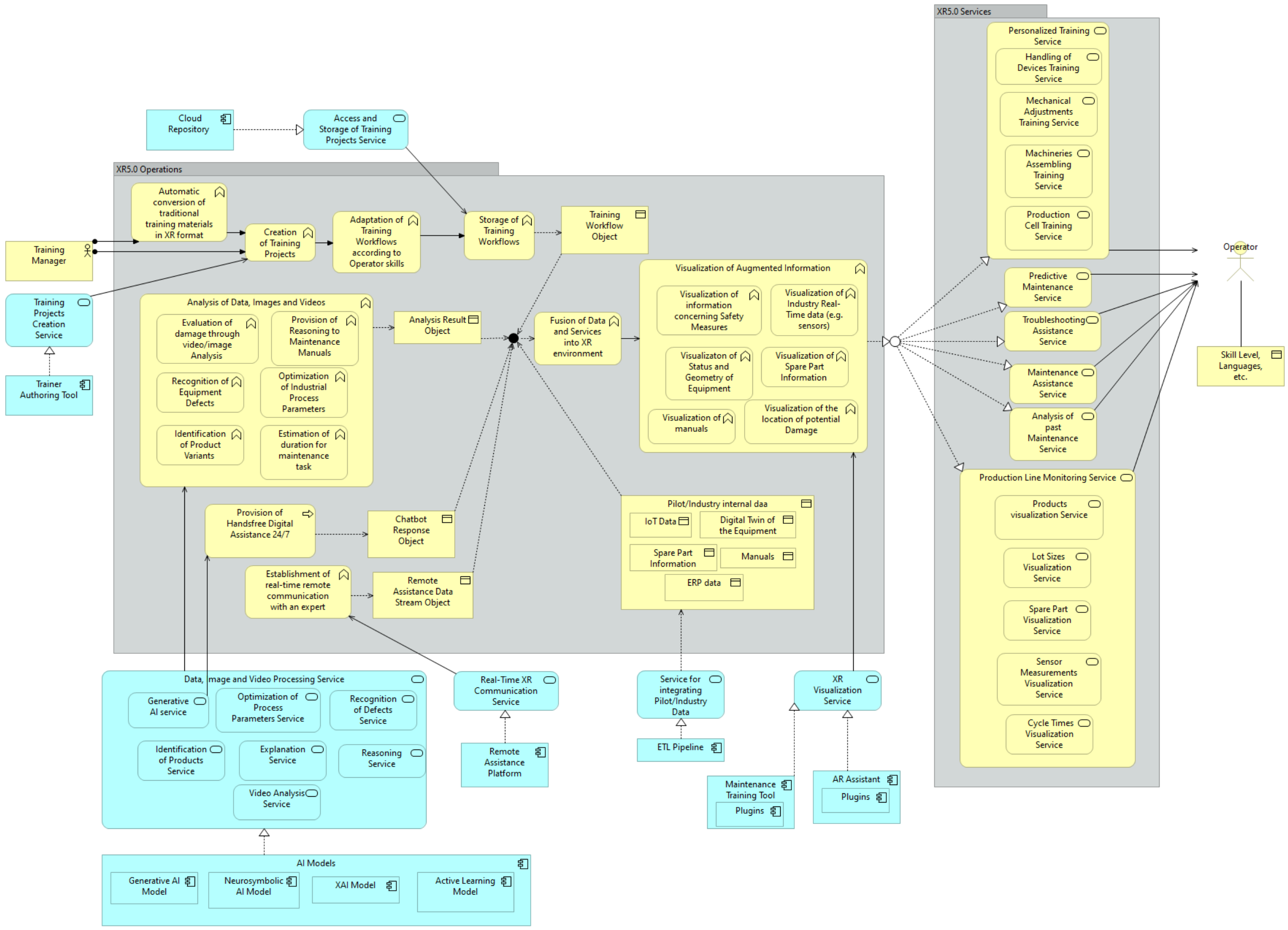


Figure 24 - Business Layer of the Architecture

At the heart of the business layer lies the **Visualization of Augmented Information** function, which synthesizes and displays critical industrial data to the operator through XR glasses. This function supports a variety of sub-functions, such as visualizing real-time sensor data, safety measures, equipment geometry, spare part details, manuals, and potential damage locations. These capabilities are powered by the integration of diverse industrial data sources, represented collectively as the **Pilot/Industry Internal Data**, which includes IoT data, digital twins, ERP information, and more.

The **Fusion of Data and Services into XR Environment** function plays a pivotal role, acting as a junction where multiple workflows converge. These workflows ensure that the operator is equipped with accurate and up-to-date information. Inputs to this fusion process include training workflows, analysis results derived from AI-driven image and video processing, chatbot responses for 24/7 digital assistance, and real-time data streams from remote communication with experts. Together, these inputs enable a seamless and immersive XR experience for the operator.

Supporting this ecosystem are a range of business services that cater to the operator's key tasks. For instance, the **Personalized Training Service** delivers tailored training programs, covering areas like device handling, mechanical adjustments, and production cell operations. The **Maintenance Assistance Service** and related services provide proactive and reactive support for industrial equipment, while the **Production Line Monitoring Service** offers insights into product visualization, cycle times, and sensor measurements. These services collectively ensure that the operator has the tools and knowledge needed for effective decision-making.

The business layer is not isolated but tightly integrated with the underlying application layer. Key application components like the **Trainer Authoring Tool**, **Cloud Repository**, **AI Models**, and **ETL Pipeline** provide the technological backbone that realizes the necessary business services. For example, the **Trainer Authoring Tool** facilitates the creation of training workflows, while the **Cloud Repository** ensures storage and access to training materials. Similarly, the **AI Models** application component powers the analysis of data, enabling predictive maintenance and optimized industrial processes.

To further enhance operator capabilities, specific XR applications like the **Maintenance Training Tool** and the **AR Assistant** deliver immersive visualization and real-time assistance. These applications ensure that operators can interact intuitively with complex industrial systems, receiving relevant insights at the right time.

Finally, the inclusion of services like **Real-Time XR Communication** ensures that operators can connect with remote experts for collaborative troubleshooting, leveraging XR for effective guidance.

This business layer is a dynamic and interconnected framework that aligns industrial operations with state-of-the-art technological advancements. By focusing on visualization, real-time assistance, and seamless integration of data and services, it lays the groundwork for achieving the goals of Industry 5.0—enhancing human-machine collaboration and ensuring sustainable, efficient, and human-centric industrial practices.

5.3.2 Application Layer

In ArchiMate, the Application Layer describes the software applications and components that provide functionality to support business and technology operations. It includes application services, data objects, interfaces and relationships among these elements, defining how systems collaborate to meet organizational goals. For the XR5.0 project, the Application Layer outlines the integration of AI models, XR applications, cloud resources and supporting platforms to deliver interactive and intelligent solutions.

As shown in figure 25, the Application Layer for XR5.0 project comprises a sophisticated ecosystem of software components and services that collectively support the creation, management and deployment of XR content and AI functionalities. The architecture is designed to facilitate immersive operator training, maintenance assistance and advanced AI-driven insights. The key elements and their roles are summarized below:

- **AI Models and Services:**
  - The **AI Models** application component consists of four specialized models: The **Generative AI**, **NeuroSymbolic AI**, **Active Learning** and **XAI Models**.
  - These models produce the **AI Produced Data**, which comprises data objects such as **Generated Content**, **Explanations**, **Learning Data**, **Symbolic Knowledge** and **Malfunction Detection Data**.
  - The AI Models also realize the **AI Services** grouping element, which includes the following services: (i) **Content Generation Service**, (ii) **Explanation Service,** (iii) **Model Update Service,** (iv) **Reasoning Service** and (v) **Video Analysis for Malfunction service**
  - These AI services are consumed by the **AR Apps** grouping element to enhance user experiences.
- **XR Apps and the Central XR Hub:**
  - **XR Apps** is a grouping element that integrates various plugins and application components for immersive content creation and streaming. Key components include:
    - **Streaming Solution Plugin: It** interfaces with **WebRTC Interface** to stream XR content, enabling rendering on the server rather than the XR glasses.
    - **Cloud Repository Plugin**: It facilitates content retrieval from the **Cloud Repository** application component.

- **XR Training Plugin**: It is a useful plugin for visualizing training workflows.
- **AR Assistant**: It visualizes the location of the damage, and it also utilizes AI for damage recognition. Furthermore, it renders sensor data and checklists.
- **Virtual Training Tool**: It processes and renders training programs using the XR Training Plugin. By using this tool, the operator can utilize a step-by-step guidance tool that enhances his training and improves his efficiency in maintenance actions. Also, it visualizes 3D models and AI recommendations.

- All XR Apps are orchestrated by the **Central XR Hub and Orchestration Platform,** which manages and integrates XR applications. As shown in figure 25, the hub provides the following functionalities:
  - It provides **User Management and Authentication** as well as **Orchestration and Management** of the **XR Apps**.
  - It delivers the service "**Update and Configuration of Embedded XR Apps**" which is consumed by the actor XR **App Developer** in order to be able to configure the various XR apps that have been integrated into the hub.

➢ **Cloud Repository and Training Management**:

- The **Cloud Repository** serves as a central storage and retrieval platform. It stores training materials and training programs, and it exposes the interface "**Storage API**". Through this interface, the **Cloud Repository plugin** is able to retrieve the content of the repository which comprises PDFs, videos, 3D models and training workflows. As shown in figure 25, the training workflows are authored by the Trainer Authoring Tool which is a web application that provides the service "**Training Programs Creation**" to be consumed by the actor named **Training Manager.** Furthermore, the repository also provides authentication functionalities.

➢ **Pilot Environment and Sensor Data Integration**:

- The **Pilot Environment** exposed the interfaces named "Pilot Interfaces" which is an ArchiMate group element. This element comprises the following interfaces:
  - The **Sensors API Interface** which exposes sensor data available in the industrial environment (e.g. air quality sensors, temperature and humidity sensors, etc.)
  - The **Biometrics Interface** exposes through wearable sensors measurements related to heart rate, temperature, motion, etc. in order to monitor the operator's stress levels, health and safety.
  - The **Gesture Data Interface** monitors the operator's movements, posture and actions in order to deliver those measurements back to XR Apps so as to update the XR experience and thereafter to enhance the operator's productivity and efficiency.
- Through the aforementioned interfaces, the **ETL Pipeline Tool** which acts as a data pipeline tool, can access the pilot data, transform them in a suitable format and deliver the measurements in the XR Apps for further processing

➢ **Remote Assistance Platform**:

- Finally, as shown in the figure below, XR5.0 also offers the **Remote Assistance Platform** application component which establishes communication between the operator and a remote expert. This enables real-time assistance through XR technologies, enhancing operational support and troubleshooting.

- ➢ **Application Deployment Service: This** is a service produced in the Technology Layer (see chapter 5.3.3). The service is depicted also in the Application Layer in order to denote that the Kubernetes cluster, and especially the worker nodes offer deployment service to the application components entitled "**AI Models**", "**Cloud Repository**", "**Central XR Hub**" and "**XR Apps**". In other words, this means that the infrastructure resources, as described in the Technology Layer in chapter 5.3.3, support the deployment of the aforementioned XR5.0 tools and this is achieved through well-defined CI/CD procedures and cloud provision. To this end, the **Application Deployment Service** is the ArchiMate element that connects the Application and Technology Layers.

A small description of each of the ArchiMate elements used in the figure below is the following (see also Annex I):

- ➢ **Application Service**: Represents a service provided by an application to support business processes, depicted as a rectangle with an ellipsis inside.
- ➢ **Application Component:** A modular part of an application, shown as a rectangle, indicating a specific software element responsible for functionality (e.g. Central XR Hub and Orchestrating Platform).
- ➢ **Group**: Represents a collection of elements within the application layer, typically shown as a dashed box enclosing related components.
- ➢ **Application Interface**: Defines interaction points between application components, depicted as a rectangle with a circle inside (e.g. WebRTC Interface)
- ➢ **Application Function**: Represents a unit of behavior or activity performed by the application shown as a rounded rectangle with an arrow inside which points to the top.
- ➢ **Data Object**: Represents an item of information used or produced by application functions, shown as a rectangle, which has another rectangle inside.
- ➢ **Junction**: Represents a decision point or flow diverging/converging, shown as a circle (a solid circle is an "and" junction while an empty circle is an "or" junction)

It has to be noted that all elements in the Application Layer in this document are depicted in light blue.

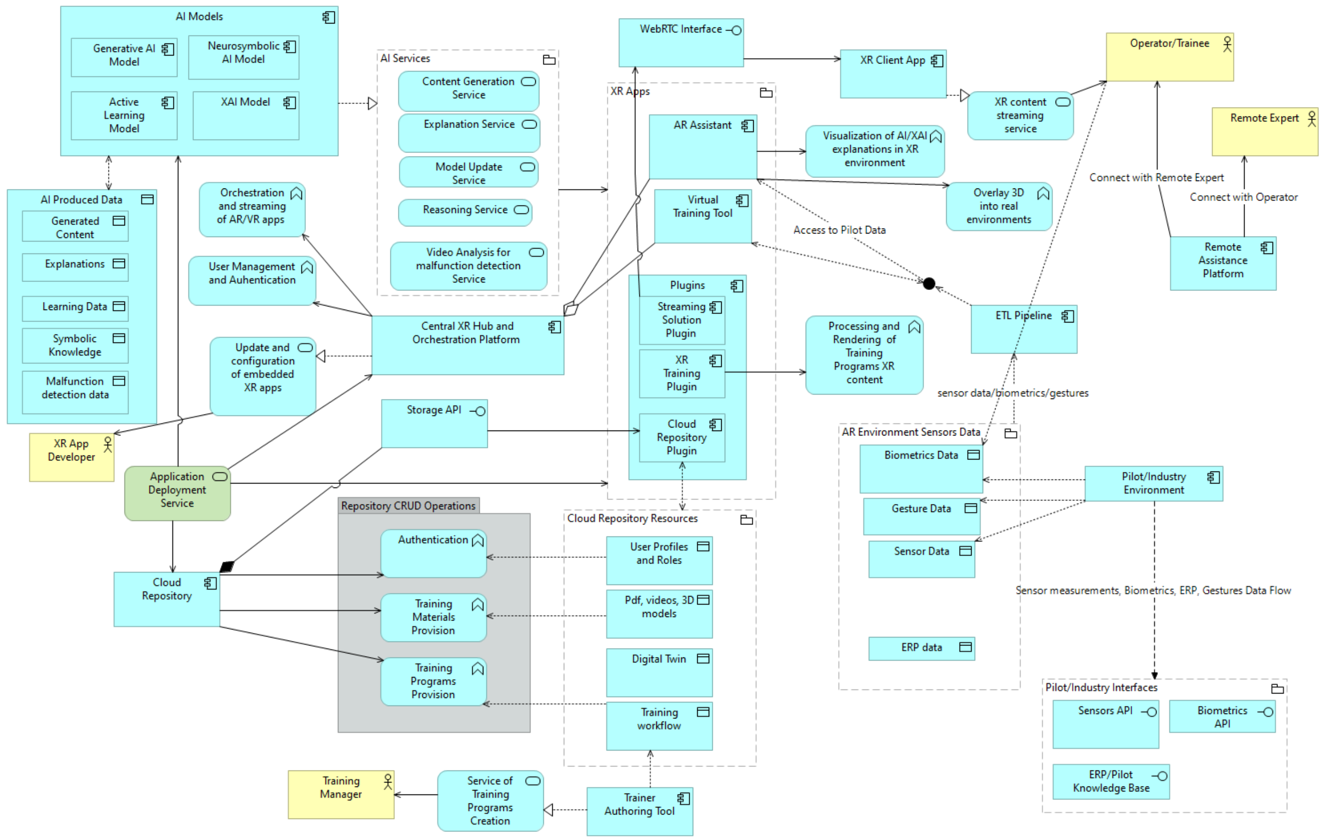


Figure 25 - Application Layer of the Architecture

As a result, the XR5.0 Application Layer establishes a robust and interconnected architecture to deliver advanced AI-driven XR experiences. By integrating AI models, XR apps, cloud-based repositories and remote assistance capabilities, the design supports interactive operator training, maintenance and decision-making processes. Each component is carefully orchestrated to ensure scalability, seamless integration and alignment with project goals.

5.3.3 Technology Layer

The Technology Layer in ArchiMate defines the infrastructure services and physical or logical components that support application and business processes. This layer describes elements such as devices, nodes, system software, networks, and their relationships, providing a foundation for higher layers in the enterprise architecture. In this context, the Technology Layer illustrates the project's virtualized infrastructure and network connectivity, which supports DevOps practices and automated deployment pipelines.

A description of the elements used in the following figure is the following (also refer to Annex I):

- **Node**: Represents a physical or virtual computational resource, depicted as a rectangle with a 3D box inside.
- **System Software**: Software that supports the operation of applications, shown as a rectangle with 2 circles inside.
- **Artifact**: Represents an actual piece of data or software deployed, depicted as a rectangle with a document symbol inside.
- **Technology Service**: A service offered by technology components, shown as a rectangle with rounded corners and ellipsis inside.

- ➢ **Device**: Represents hardware devices like servers or workstations, depicted as a rectangle with a computer symbol inside.
- ➢ **Communication Network**: Represents communication infrastructure, shown as a rectangle with a network symbol inside.
- ➢ **Group**: Represents a collection of technology elements, shown as a dashed box.
- ➢ **Technology Function**: Represents a technology-specific function or task, shown as a rounded rectangle with an arrow inside pointing to the top.

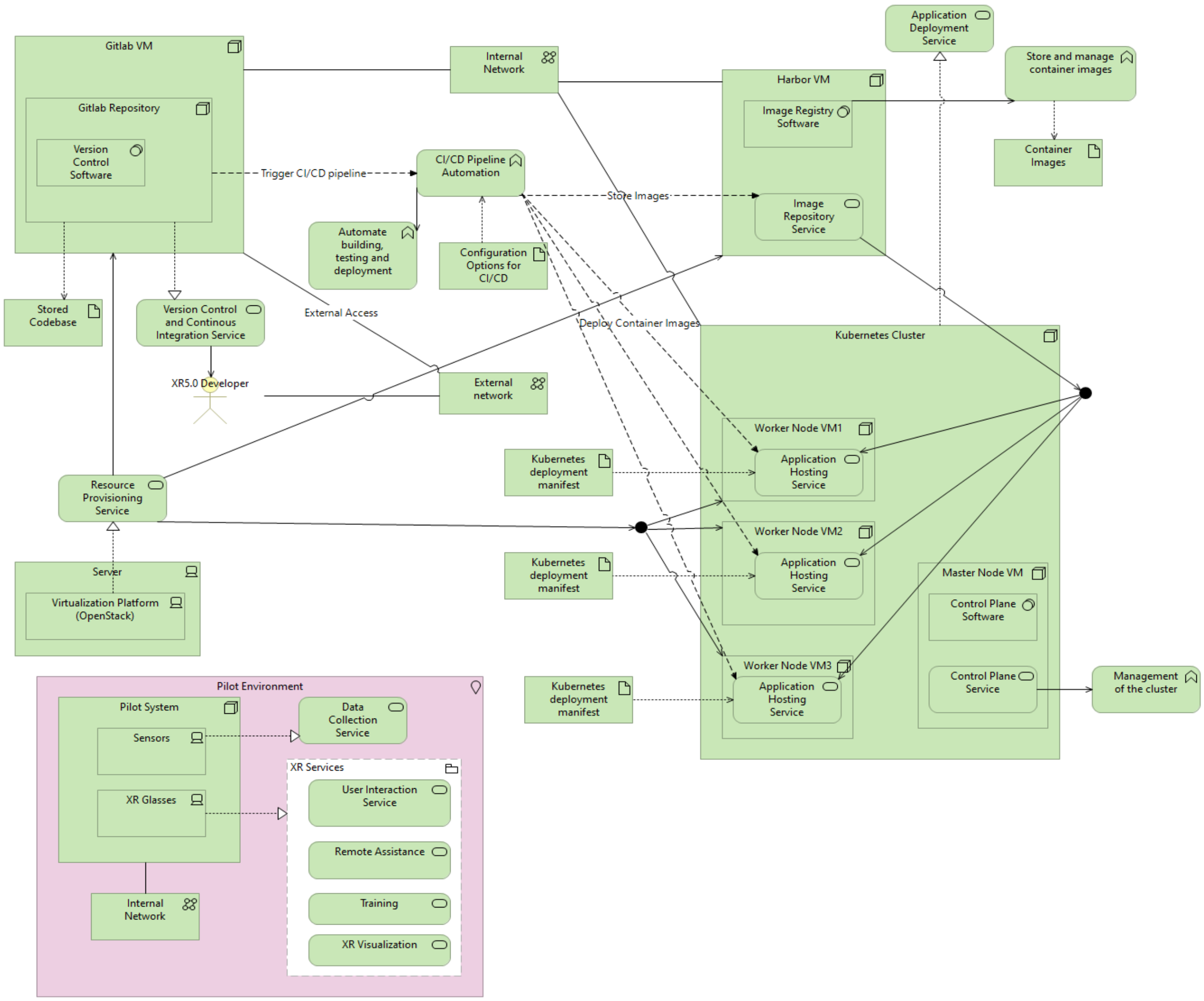


Figure 26 - Technology Layer of the Architecture

The diagram depicted in Figure 26 models XR5.0 technical infrastructure for software development, artifact management and automated deployments using ArchiMate's Technology Layer concepts. It highlights the relationships between the physical hardware, virtualization layer, network services and the virtual machines (VMs) deployed for various purposes. The key components and their interactions are detailed below:

- ➢ **Physical Server and Virtualization Layer**: A physical server device has been provided in the project in order to host the project's resources. This server represents the underlying hardware. Moreover, the Virtualization Platform (system software) is realized on the server, providing the necessary Resource Provisioning Service to host and manage multiple VMs. The server and virtualization layer ensure resource efficiency and scalability while enabling isolation for different workloads.

- **Virtual Machines (VMs):** Several VMs are provisioned on the virtualization platform to support specific functions as follows:
    - **GitLab VM:** It hosts the Gitlab system software, which provides the shared repository for version control and continuous integration/continuous delivery (CI/CD) pipelines. Moreover, it enables the project partners to push and manage their codes collaboratively.
    - **Harbor VM**: It acts as a container image registry to securely store and manage Docker images generated by the CI/CD pipelines.
    - **Kubernetes**: It consists of a Master Node and multiple Worker Nodes, each realized as a VM. Moreover, it provides the platform for orchestrating and deploying project artifacts (containerized XR5.0 applications)
- **Communication Networks**: As shown in Figure 26, two types of networks exist in the XR5.0 system:
    - **Internal Network**: There is an internal network that connects all VMs hosted on the server. It facilitates internal communication between the Gitlab VM and Harbor VM for artifact transfer. And it also allows workload orchestration between the Kubernetes master node and worker nodes. Moreover, there is the internal network in the pilot environment which ensures isolation of the industry systems and the operator devices (ERP, sensors, databases, machines, etc.). Access control is the most central aspect of the internal networks and also the privacy of internal processes and data flows plays a big role.
    - **External Network**: It represents the external internet or network layer enabling partners from external organizations to connect to the Gitlab VM securely. It provides access for remote code submission and repository management.
- **Key Data Flows and Processes**:
    - **External Access**: Project partners connect to the Gitlab VM through the external network to push their code to the shared repository.
    - **Internal CI/CD Workflow**: Code from the repository is processed through Gitlab CI/CD pipelines, producing container images stored in the Harbor image registry.
    - **Artifact Deployment**: The Kubernetes master node pulls images from Harbor and deploys them to the worker nodes, using the internal network for communication.
    - **Pilot Application Use**: In the pilot settings, the operator will use both the deployed applications hosted in Kubernetes worker nodes but also their internal application and services which are internally deployed in order to perform their operations. These applications will enable the operator to perform operations aligned with the project's objectives, ensuring that the XR5.0 infrastructure meets the requirements identified in the Deliverable D2.1[2].

Thereafter, the Technology Layer ensures the following:

- **Scalability**: The virtualization platform allows for flexible resource allocation across VMs.
- **Security**: The separation of internal and external networks protects internal resources while allowing controlled access for external partners.
- **Automation**: The integration of Gitlab CI/CD, Harbor and Kubernetes streamlines the build, store and deploy cycle, reducing manual intervention.

- ➢ **Collaboration**: External access to the Gitlab VM supports seamless collaboration with partners across different organizations.

The Technology Layer for XR5.0 system effectively balances resource provisioning, secure networking and automation to meet the project's requirements for DevOps and artifact deployment. This architecture provides a robust foundation for the project's operations, ensuring efficiency, security and scalability.

5.4 Solution Blueprints

In the context of the XR5.0 project, solution blueprints provide a concrete, structured representation of how the envisioned system will be realized across multiple dimensions. While previous chapters have outlined the system architecture from strategic, business, application and technology viewpoints, solution blueprints serve as a bridge between high-level architecture models and component-level design, offering a more detailed depiction of workflows, interactions and data flows across the system.

Solution blueprints are particularly valuable in complex, multi-layered environments such as XR5.0, where immersive experiences, personalized training and worker-centric solutions intersect with Industry 5.0 processes. They allow stakeholders to visualize how business objectives, technical requirements and operational processes translate into actionable system components. By providing a layered and detailed perspective, solution blueprints enhance understanding, facilitate communication among consortium partners, and support alignment with both the user stories and the technical standards.

The solution blueprints are represented using ArchiMate in order to depict the interrelations between business processes, applications and underlying technology layers. This approach ensures consistency with the architectural views previously presented, while adding the granularity necessary to understand end-to-end workflows, system interactions and functional dependencies. ArchiMate-based blueprints allow us to explicitly capture the flow of information and responsibilities across XR5.0 components, highlighting the integration of Human Digital Twins, XR training personalization and XR content delivery mechanisms.

The solution blueprints in this chapter focus on system-level workflows and cross-layer integration, providing a high-level yet sufficiently detailed visualization of key XR5.0 capabilities. They complement the architectural diagrams in earlier sections by:

- ➢ Depicting multi-layer interactions across business, application and technology viewpoints.
- ➢ Illustrating specific workflows for critical use cases such as personalized training, training lifecycle management, and system security.
- ➢ Highlighting data flows and decision points that enable real-time adaptation and intelligent support mechanisms.
- ➢ Serving as a reference for implementation teams to guide component development, integration and deployment.

It is important to note that the blueprints presented in this chapter focus on the overall system perspective using ArchiMate. In contrast, chapter 6.3 will present component-level solution blueprints using the C4 modeling approach, detailing the internal structure, interactions and technical interfaces of each XR5.0 module. This dual-level representation ensures that the XR5.0 architecture is fully traceable from strategic objectives down to implementation details, providing a robust framework for both development and evaluation.

In the following sub-sections, we provide three specific solution blueprints that illustrate the following key XR5.0 workflows:

- ➢ **Personalized Training Blueprint**: A multi-layered model illustrating the creation of personalized training projects for Industry 5.0. It integrates Human Digital Twins to enable both pre-designed and real-time adaptation of XR training, with personalization driven by machine learning models mapping emotional states to training recommendations. Data is collected before, during and after training, including heart rate and skin conductance.

- **Training Workflow Blueprint**: An end-to-end depiction of the training lifecycle, detailing the pipeline from traditional training material transformation to XR content delivery. It captures both technical and business processes supporting training creation, deployment and adaptation.
- **Security View Blueprint**: A security-focused architecture diagram illustrating OAuth 2.0 authentication and authorization flows across XR5.0 components. It ensures secure access and integrates seamlessly with the other solution workflows.

Collectively, these solution blueprints provide a comprehensive view of XR5.0, showing how the system delivers adaptive, immersive and secure worker-centric training experiences. Combined with the C4 component-level blueprints in chapter 6, they ensure that the system is fully specified, traceable and actionable across all layers of architecture.

5.4.1 Personalized Training Blueprint

The personalized XR training process within XR5.0 is designed to support adaptive learning pathways that are tailored to the skill level, performance behavior and operational needs of individual workers. The process combines immersive XR environments with performance monitoring and AI-driven personalization mechanisms, ensuring that training becomes both effective and contextually relevant to the user.

The diagram depicted in Figure 27 comprises a multi-layer approach spanning from low-level technology layer that is related to the pilot/Industry 5 premises, to high-level business processes that model the business workflow required for the delivery of the personalized and user-centric training. As shown at the bottom of Figure 27, the technology layer consists of the tools responsible for capturing values related to worker physiological data and pilot specific data (ERP). All this data is delivered to the application component named "OPC-UA Connector". More specifically, the elements of the technology layer are the following:

- **Pilot Camera**: Real-time camera images are being used to detect the worker in order to predict his short-term movement.
- **Pilot Enterprise Resource Planning (ERP)**: Pilot ERP data that represents information related to worker assigned tasks, are delivered to the Adaptor/Connector (i.e. OPC-UA Connector).
- **Smartwatch**: As shown in the figure below, the smartwatch element consists of the following sub-elements:
  - **Smartwatch OS**: It is a system-level software that manages the core operations of the smartwatch device. It provides the foundational services required for hardware interaction, application execution, connectivity, power management, and user interface rendering.
  - **Smartwatch app**: The Smartwatch app is an application running on the smartwatch that continuously collects physiological data from the device's embedded sensors, such as heart rate and skin conductance. It processes and displays these measurements locally to the user and securely transmits the biometric measurements to OPC-UA Connector.
  - **Heart Rate Sensor**: It is an embedded hardware component in the smartwatch that measures the user's pulse by detecting changes in blood flow, typically using optical photoplethysmography (PPG) technology. It continuously captures and transmits heart rate data to the system software and application, enabling real-time monitoring and feedback.
  - The skin conductance sensor is an embedded hardware component that measures the electrical conductivity of the user's skin, which varies with physiological responses such as stress, excitement or exertion. By detecting small changes in sweat gland activity, the sensor

provides data related to emotional arousal and autonomous nervous system activity during the training.

All of the aforementioned unprocessed measurements (worker location and speed, body and head orientation, biometrics and assigned tasks) are captured through the Technology Layer and delivered to the application component named “OPC-UA Connector”. As shown in the Application Layer, the adaptor adapts the data and delivers them to the Clawdite platform, which instantiates the Human Digital Twins (DTs). Clawdite then stores the data in twins’ representation and exposes them through the corresponding API. As shown in Figure 27, Clawdite consists of the following key sub-components:

- **Historical Data Manager (HDM)**: It retrieves and stores historical data from the gateways and the functional modules and enables advanced reporting and analytics.
- **Orchestrator**: The Orchestrator manages the platform’s overall structure, including Digital Twin instances, worker profiles, installed modules, connected sensors and message schemas.
- **Functional Modules**: External components that can be plugged in to provide additional functionalities.

According to the Business Layer at the top of the diagram (i.e. elements shown in yellow), the worker collected data are gathered through 3 phases: (i) pre-training data, (ii) on-the-training data and (iii) post-training data which are triggered by the following business events: (i) Training Task Preparation, (ii) Training Task Begins and (iii) Training Task Ends. Moreover, a live questionnaire is provided through a chatbot to the worker, before and after the training. Based on all these aforementioned data (questionnaires, biometrics, operator body status), the digital representation of the worker is updated by making use of the Human Digital Twin Creation and Storage service of Clawdite component.

The Clawdite platform exposes the following application services:

- **Worker Experience Service**: Through this service, experience level data of the worker is provided to the Training Authoring Tool application component, in order for the trainer to create a new personalized training project for the worker. In this case, the Authoring Tool takes into consideration the skills of the worker, in order for the trainer to create a suitable training project.
- **Worker Emotional Service**: The emotional status of the worker is provided to the Active Learning Model. This emotional status has been modeled based on the heart rate and skin conductance sensor measurements which have been derived through the smartwatch. The Active Learning Model then maps the emotional state to training recommendations by making use of trends analyzation, prediction of learning performance and emotional data interpretation.
- **Prediction of Operator Movement**: Real-time camera images feed the corresponding functional module of Clawdite in order to detect the worker and predict his short-term movement. This prediction is useful because it can be used by the path planning algorithm of an Autonomous Mobile Robot (AMR) in order to avoid collisions with the worker.

As shown in Figure 27, the application service named “Mapping of emotional state to training recommendations” is provided to the XR app through the corresponding APIs. Then the XR app, according to the aforementioned mapping, monitors and adjusts the training in real-time. This results in providing the application service named “Real-time adaptation of XR training” which serves the business service named “Provision of personalized training to the Operator”. The latter serves the Operator who makes use of XR5.0 in order to consume user-centric Industry 5.0 training services.

The workflow depicted in Figure 27 enables ongoing skill development, gradual competence building, and training experiences tailored to each worker’s performance evolution. The approach also supports training efficiency by reducing unnecessary repetition and ensuring that learners focus effort on areas where improvement is needed.

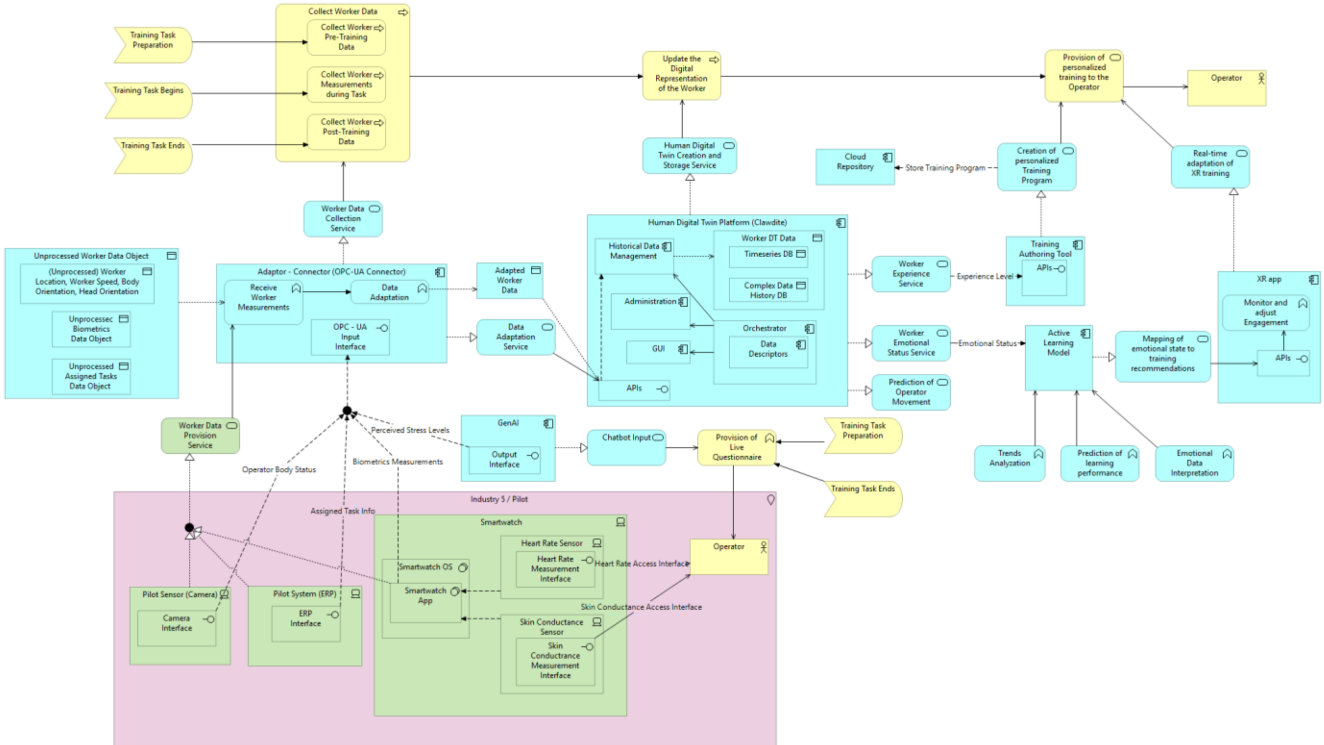


Figure 27 - Personalized Training Blueprint

5.4.2 Training Workflow Blueprint

The Training Workflow Blueprint illustrates the complete end-to-end process of the XR5.0 training ecosystem, supporting the delivery of immersive, user-centric training experiences in the context of Industry 5.0. This workflow enables a seamless transformation of traditional training content into interactive Extended Reality (XR) experiences, fostering enhanced engagement, personalized learning, and improved operational skills for industrial workers.

As shown in Figure 28, two primary roles participate in this workflow:

- ➢ The Training Manager, who is responsible for creating and managing the training projects, and
- ➢ The Operator, who engages in the immersive XR-based training sessions to develop competencies in maintenance, troubleshooting, and other industrial processes.

To enable immersive experiences, conventional training materials (e.g., PDF documents, slide decks, videos, or images) are transformed into XR Training Objects (e.g., annotated videos, checklists, 3D interactive elements). This transformation is performed through the application services of the Training Programs Authoring Tool, which leverages Generative AI models to automate the conversion and enrichment of training assets.

Once generated, the XR Training Objects are integrated into a Training Program, which is securely stored within the XR5.0 Cloud Repository. According to the business layer of the architecture, the workflow proceeds with the business function "Retrieval of Training Program", which is performed via the respective XR Applications (such as the *AR Assistant*, *Virtual Training Tool*, or other compatible XR solutions).

The retrieval process is facilitated by the Cloud Repository Plugin, ensuring secure and efficient access to stored training content. Subsequently, the visualization and delivery of the training program are enabled through the XR Training Plugin and the Stream Plugin, which allow real-time rendering and interaction with the XR content on the trainee's device.

At the highest business level, as depicted on the right-hand side of the business layer in Figure 28, the workflow culminates in the delivery of the "Personalized Training Service". This service provides adaptive, context-aware, and value-added learning experiences tailored to each operator's role, performance, and learning needs, fully aligning with the Industry 5.0 vision of human-centric, intelligent, and collaborative industrial environments.

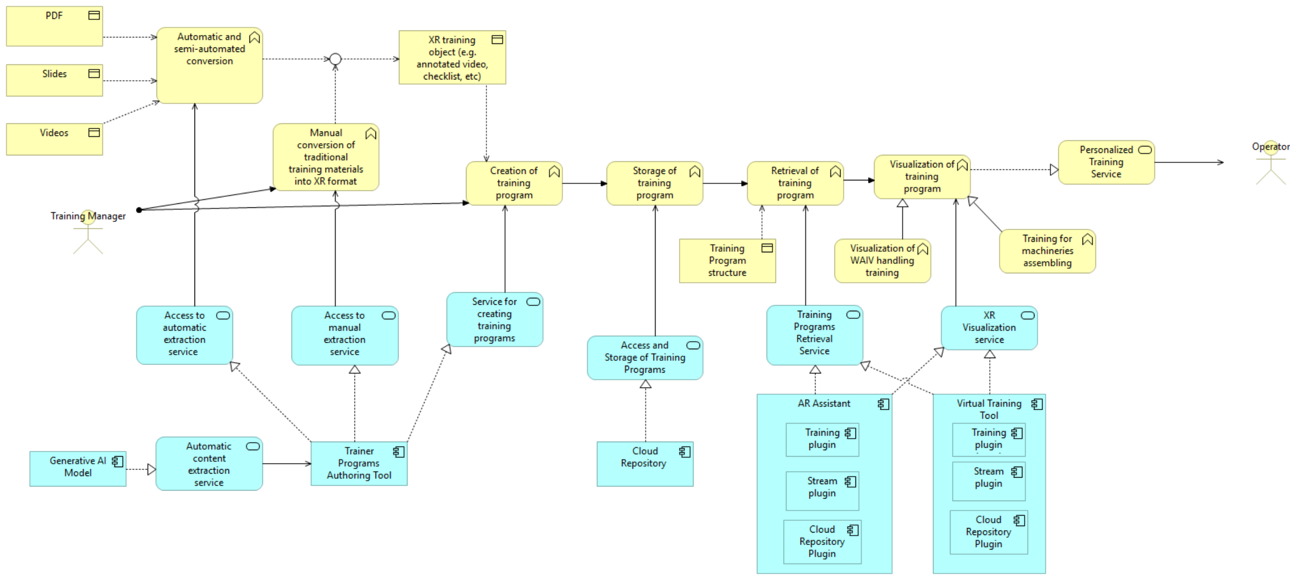


Figure 28 - Training Workflow Blueprint

5.4.3 Security and Access Control Blueprint

The XR5.0 ecosystem integrates multiple AI services, XR applications, training tools and cloud backend components. To ensure secure, controlled and standardized access to these distributed resources, an OAuth2-based security architecture has been adopted. This approach provides unified authentication, fine-grained authorization, token-based access control and interoperability with existing identity management systems.

The workflow ensures that all interactions between client applications (service consumers) and AI/cloud services (service providers) follow a consistent and secure pattern, regardless of the hosting environment (edge, cloud, or XR device). This enhances trustworthiness across XR5.0 architecture and supports its alignment with modern cybersecurity and data-protection requirements.

The following actors have been identified in the workflow:

➢ **Service Consumption Components (Clients):** These components which request access to XR5.0 services are the following:

  - **XR Training Authoring Tool**: Needs to access AI services (e.g. GenAI for content generation) and the Cloud Repository to store training assets.
  - **XR App (AR/VR Training Client)**: Requires access to AI models (e.g. Active Learning, GenAI) and training content stored in the Cloud Repository.
  - **Cloud Repository User Interface**: Enables authenticated users to upload, retrieve and manage training assets, models or metadata

  These components behave as OAuth2 clients, obtaining access tokens to call secured backend APIs

➢ **Service Provisioning Components (Resource Servers)**: These components expose protected APIs:

  - **Active Learning Model Backend**

- **Generative AI Services**
- **XAI Model Backend**
- **Neurosymbolic AI System**
- **Cloud Repository Backend**
- **Human Digital Twin Platform (Clawdite)**

As shown in Figure 29, the XR5.0 security architecture introduces an Identity and Access Management (IAM) system which is part of XR Hub and Orchestration Platform. The IAM component enforces strict tenant isolation, ensuring that each tenant (pilot or industry) maintains a distinct and separate set of users with no cross-visibility or shared identity space. Moreover, the IAM provides the following services:

➢ **Application Management Service**: It is responsible for securely registering, updating and maintaining XR applications used throughout the XR5.0 ecosystem. It ensures that only trusted and verified XR applications are onboarded into the XR5.0 Operator Training Platform. This service enforces application lifecycle policies and coordinates secure distribution of application packages to XR headsets and operator devices. By controlling the full application lifecycle, it prevents unauthorized or tampered applications from entering the training environment.

➢ **Security Management Service**: The Security Management Service provides the central security governance layer of the platform. It coordinates identity, access control and security policy deployment across all XR5.0 components. The service enforces compliance with defined standards and ensures that all interactions within the XR ecosystem meet Industry 5.0 security and safety requirements, safeguarding both operators and digital assets.

➢ **User Management Service**: The User Management Service handles the secure administration of user identities, profiles and role assignments across the XR5.0 ecosystem. It stores user attributes, manages onboarding workflows and defines access privileges for different user categories (trainees, trainers, administrators, etc.). The service supports multi-role environments and ensures that access to XR training content, authoring tools and cloud services is aligned with organizational policies.

➢ **Login Service**: The Login Service authenticates users across XR headsets, workstations, and cloud-based tools. It supports password-based login, initiates secure authentication flows, issues session tokens and ensures that users gain access only after their identity has been verified through approved authentication methods. It serves as the main entry point for secure access to XR applications.

➢ **Token Generation Service:** The Token Generation Service provides secure, cryptographically signed tokens that represent user or application identities during platform interactions. These tokens carry role, permission and session information used by downstream XR5.0 services to enforce access control. The service uses secure algorithms and key-management practices to ensure token integrity and prevent forgery. By generating short-lived, scoped tokens, it supports fine-grained authorization for XR training scenarios, cloud repository access and content authoring workflows.

➢ **Policy Enforcement Service:** The Policy Enforcement Service ensures that every request (whether coming from an XR headset, the cloud-based training platform or the authoring tool) adhered to the defined access control and secure policies. It evaluates access attempts using a combination of user roles, contextual information and operational policies defined for the XR5.0 environment. The service enables precise and dynamic control over who can view, modify or publish training content, supporting safety-critical training workflows.

➢ **Token Validation and Revocation Service:** The service validates all incoming tokens to confirm their authenticity, expiration time, signature and assigned permissions. It ensures that only legitimate tokens originating from the trusted Token Generation Service are accepted. Additionally, it provides real-time token revocation capabilities, allowing administrators to immediately disable compromised credentials or terminate active sessions. This supports continuous security assurance during immersive training operations.

The IAM supports standard OAuth2 grant types, enabling flexibility for browser-based tools, XR headsets and backend-to-backend communication. To ensure interoperability and future integration with the EU XR Platform, the adopted workflow is compliant with OAuth2 and OpenID Connect principles.

As depicted in Figure 29, the XR5.0 OAuth2-based workflow follows five main steps:

1. **Step 1: User Authentication:**
    a. The user initiates access from a client application (XR app, authoring tool, cloud UI).
    b. The application redirects the user to IAM's login screen.
    c. The user authenticates via username/password.
    d. The IAM issues an authorization code (for interactive clients) or a client credential token (for non-interactive backend services).
2. **Step 2: Token Issuance**: The client application exchanges the authorization code for access token (used to call protected APIs) or Refresh Token (used to obtain new access tokens without re-authentication). Access tokens include: (i) token expiration, user identity or client ID and authorization scopes.
3. **Step 3: API Request to Service Provider**: The client (XR app, authoring tool, cloud repository UI, etc.) invokes a backend API by including the access token in the request header. Depending on the client's purpose: (i) the XR app may request personalized training recommendations from the Active Learning or HDT service, (ii) the Authoring Tool may call the GenAI to convert traditional material to XR training objects and (iii) the Cloud Repository may request access to repository files or metadata.
4. **Step 4: Token Verification by Resource Server**: Each AI service or backend component performs the following checks:
    a. **Token validity** (not expired, not revoked)
    b. **Signature verification** using the Authorization Server's public keys
    c. **Scope validation** ensuring the token contains the required permissions.
    d. **Role validation**, if the resource requires specific user roles (e.g. trainer, operator, admin)
5. **Step 5: Service Execution and Response**: The resource server executes the requested operation and returns:
    a. AI inference results (e.g. active learning suggestions, generative outputs)
    b. XAI explanations or model decision justifications
    c. User-specific training content or resource updates
    d. Repository items, metadata or storage confirmations

This ensures end-to-end controlled access under a unified security layer.

According to Figure 29, the XR5.0 platform defines a clear governance model for security management. A central XR5.0 Security Manager oversees authentication, authorization and security policies across all tenants, supported by the Security Management Service. At the tenant level (i.e. pilot level), each Tenant Administrator manages the tenant's user base through the User Management Service, ensuring full separation and autonomy of user domains.

The Key Architectural Benefits of the XR5.0 security workflow are the following:

- ➢ **Unified security across heterogeneous components**: Regardless of the technology stack used by AI services or XR clients, OAuth2 provides a uniform security model.
- ➢ **Fine-grained access control**: Through scopes, roles, and policies, each service defines precisely what resources a user or application may access.
- ➢ **Interoperability and standards alignment**: The OAuth2 workflow aligns with ETSI MEC security principles, OpenID Connect, ISO/IEC 27001 controls, and EU XR Platform requirements.
- ➢ **Secure scalability**: Additional services (such as new AI models or external platform components) can easily be integrated into the authorization framework without major redesign.

To this end, the OAuth2-based workflow ensures that all XR5.0 components (XR applications, authoring tool, AI models, cloud repository, etc.) operate within a secure, consistent and standards-based access control environment. This workflow is central to ensuring trustworthy human-AI interaction, compliant data management, and scalable integration with future platforms such as the EU XR Platform.

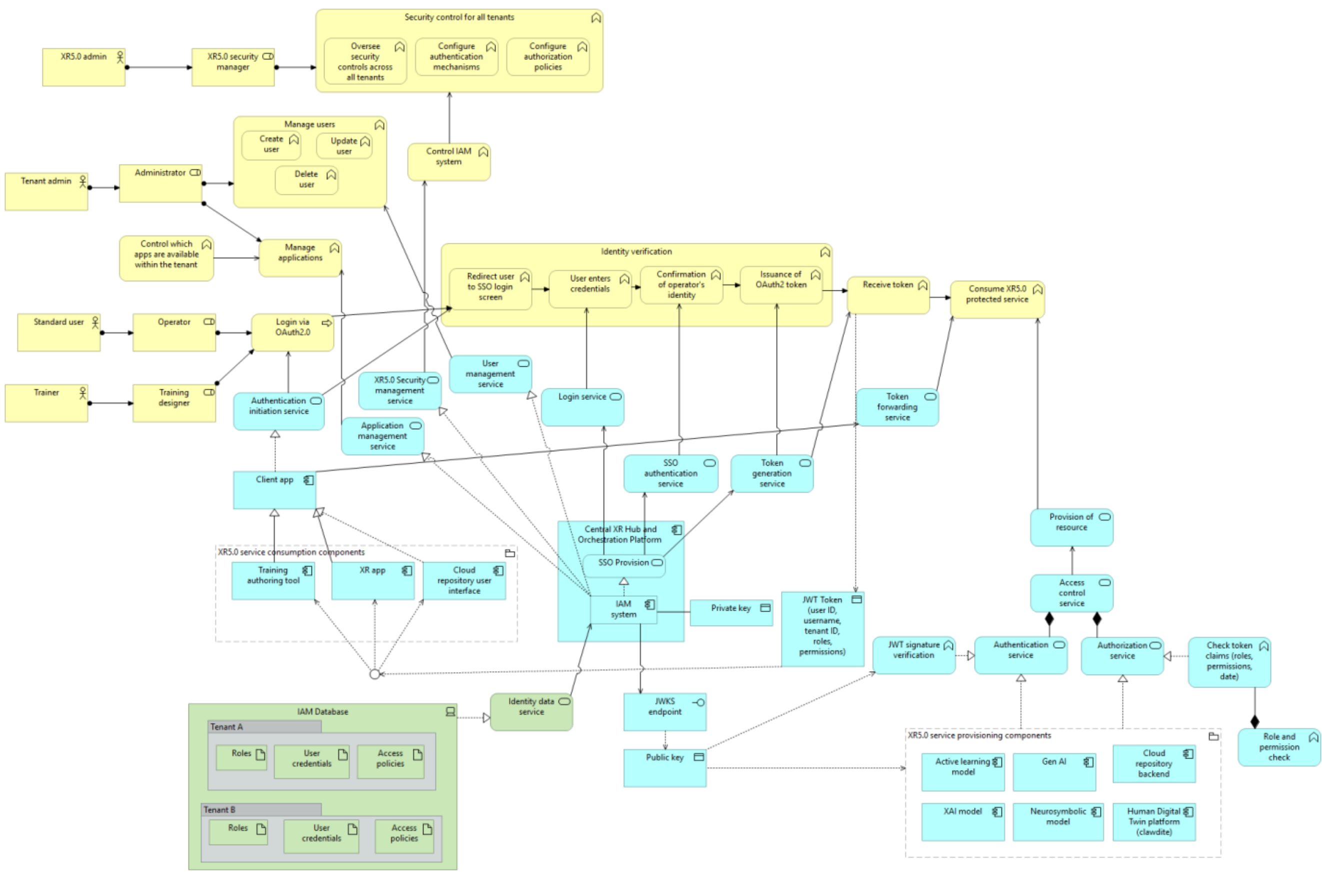


Figure 29 - Security and Access Control Blueprint

6. Solution Architecture

This section transitions from the broader enterprise architecture explored earlier, to a more detailed methodology of solution architecture specifically focusing on visualizing software architecture. While the enterprise architecture provides an overview of interactions between the business, application, and technology layers enabling the identification of the XR5.0 system keys objectives, the solution architecture offers a detailed examination of the software components within these layers, including their structural configuration and operational characteristics.

In order to achieve this, the C4 model [1] is adopted for visualization of software architecture. The C4 model enables the decomposition of advanced software systems into four levels such as Context, Containers, Components and Code. This methodology will enable the alignment of the components of the system with both business goals and technical requirements, bridging the gap between the high-level enterprise architecture and the specific software design. As we proceed, the focus will shift to how software components function within the larger architecture and to their contribution to operational efficiency and scalability.

6.1 System Context

The **System Context Diagram** provides an overview of the XR5.0 platform and its interactions with external entities. The ***XR5.0 platform*** is defined as a central software hub that leverages XR and AI technologies to enhance safety, productivity, and user experience in Industry 5.0 operations. It facilitates external connections in order to utilize sensor measurements, enterprise resource planning data, operational manuals, maintenance procedures, and training content.

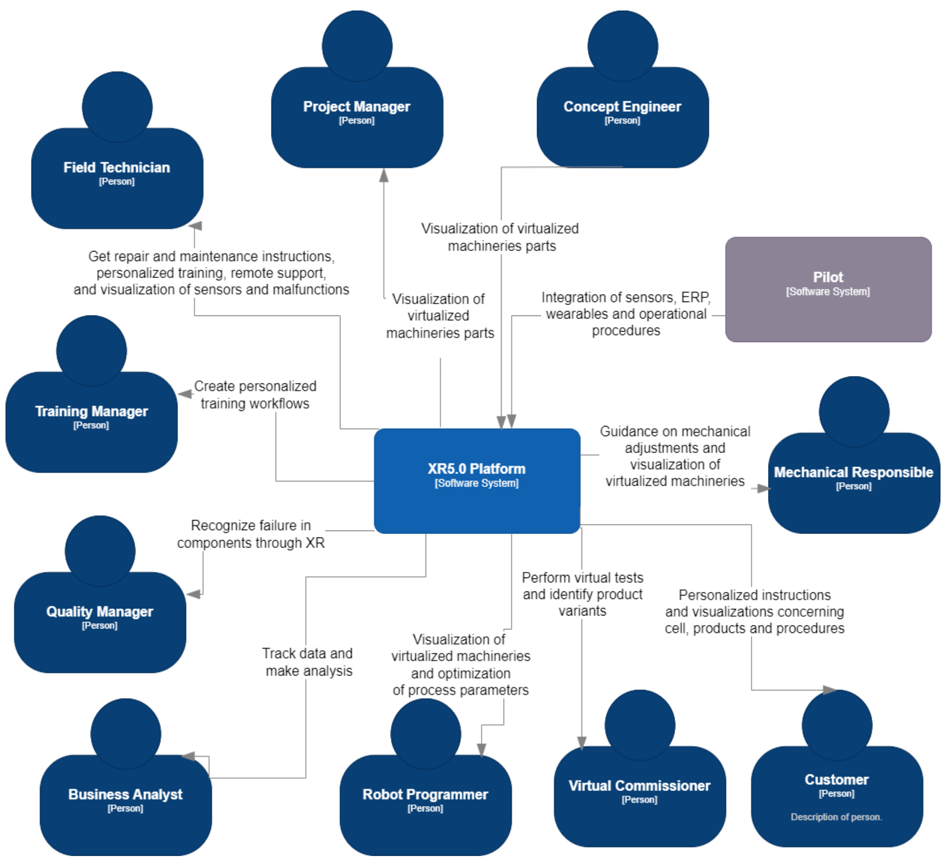


Figure 30 - System context diagram of Solution Architecture

Figure 30 provides an overview of the high-level interactions and roles, offering clarity on the position of the system within the enterprise landscape and emphasizing its integrative functionality for Industry 5.0 initiatives.

This includes human roles, such as ***Field Technicians***, ***Training Managers***, and ***Quality Managers***, as well as third-party systems like external ***Pilots*** and external ***Data Sources***. This diagram illustrates the key external actors involved in the system, along with their primary interactions, as stated in the XR5.0 Deliverable D2.1 [2]:

- ➢ **Field Technician**: A skilled professional responsible for installing, maintaining, troubleshooting, and repairing equipment or systems directly on-site in industrial settings. This role relates to the XR5.0 platform through an intuitive XR environment that provides: (i) digital assistance (e.g., chatbot), (ii) step-by-step instructions, (iii) personalized training, (iv) remote assistance support, (v) malfunction detection, (vi) monitoring of the production line and (vii) safety measures alerts. The integration allows the Field Technician to perform tasks efficiently, utilising advanced XR and AI functionalities to enhance productivity and ensure safety.
- ➢ **Training Manager**: A training manager is responsible for creating training programs that enhance employee skills and productivity across the organization. This role is connected with the XR5.0 platform through the ability to create training workflows personalized to individual technician skill levels. The integration enables Training Managers to devise targeted and efficient training programs that are aligned with both organisational goals and individual learning needs, utilising the advanced tools provided by the XR5.0 platform.
- ➢ **Quality Manager**: A Quality Manager ensures that products meet quality standards by conducting testing of raw materials, goods, and finished products. This role is connected with the XR5.0 platform through functionalities that allow them to (i) view information concerning testing, (ii) perform testing using a Digital Twin, (iii) recognize failure conditions (e.g., identifying leak conditions). The integration allows Quality Managers to maintain high standards of product reliability and safety while utilising sophisticated tools in the XR5.0 system for real-time testing and fault detection.
- ➢ **Business Analyst:** A Business Analyst plays a crucial role in analyzing system-generated data and aligning system performance with business objectives. This role relates to the XR5.0 platform through functionalities that allow them to (i) track user actions and data generated by the system, (ii) analyze past maintenance activities to identify patterns, and (iii) flag recurring problems for predictive maintenance and improved efficiency. The integration allows Business Analysts to derive actionable insights from the system's data, thereby facilitating continuous improvement and alignment with organisational goals.
- ➢ **Robot Programmer:** A Robot Programmer is responsible for configuring and optimizing robotic systems used in industrial automation. This role is connected with the XR5.0 platform through functionalities that allow them to get a virtual presentation of the current state of machinery and their missing parts and receive instructions concerning the setting and optimization of process parameters for Programmable Logic Controller (PLC) programming. The integration facilitates the automation process for Robot Programmers, ensuring the seamless operation of robotic systems through the utilisation of XR5.0's advanced visualisation and instructional tools.
- ➢ **Mechanical Responsible:** Responsible for the mechanical aspects and systems, ensuring proper maintenance and optimization of mechanical components. This role relates to the XR5.0 platform through functionalities that allow them to get instructions concerning the mechanical adjustments. The integration allows the mechanical responsible party to perform mechanical optimisations in an efficient manner and to guarantee seamless system functionality, making use of XR5.0's support for precision and operational effectiveness.

- **Virtual Commissioner:** The person who ensures that machinery, equipment, and systems are tested and validated before full-scale deployment. This role is connected with the XR5.0 platform through functionalities that allow them to (i) view information concerning the monitoring of the production line (e.g., cycle times, lot sizes, pictures of products as they are embedded in the production line), (ii) view highlighted identified product variants in AR glasses, and to (iii) view test results during campaigns. The integration allows the Virtual Commissioner to validate and optimize production processes, utilising the capabilities of XR5.0 to guarantee system readiness and performance prior to deployment.
- **Project Manager**: A Project Manager oversees the entire project lifecycle. He relies on reports from the technical team to monitor progress and ensure project success. In the context of XR5.0, the project manager needs to see the virtualized machineries parts, in order to have a better impression of the current state and the relation to the still upcoming installation work. Moreover, the visualization of information concerning the order in which parts need to be installed is a desired functionality that needs to be offered to this role.
- **Concept Engineer: This** role revolves around bridging technology, creativity and sustainability to design innovative solutions and systems that integrate advanced technologies with a human-centric and environmentally conscious approach. More specifically, key responsibilities of a Concept Engineer are the following: (i) to develop futuristic concepts for products, processes and systems by leveraging emerging technologies, (ii) to design systems and processes and evaluate the lifecycle impact of proposed concepts, (iii) to work with software and hardware teams to prototype concepts and test their feasibility and (iv) to develop concepts that enhance resilience of supply chains and manufacturing processes. In the context of XR5.0, a Concept Engineer would like to view the virtualized machineries parts in order to get a better impression of the current state and the relation to the still upcoming work. Moreover, the visualization of the order in which the parts need to be installed is also important for this role.
- **Customer: This** role could represent either an operator or a dedicated maintenance department ensuring the continuous operation of the machines. According to feedback from pilot 6, a customer is responsible for contacting the company in case of problems or preventive maintenance. He has access to relevant documentation and the technical support portal. The skill level of this role varies greatly, from deep technical knowledge to basic operation and maintenance abilities.
- **Pilot:** XR5.0 Pilots are used to drive the technical development, evaluate, and validate XR5.0 technology, and to experiment with diverse industry environments. In this diagram the pilot is recognized as a Software System and is connected with the XR5.0 platform through functionalities that allow them to (i) provide APIs for integrating ERP, sensors, and wearables, and (ii) provision information concerning operational procedures, historical data for training models, and manuals. This integration enables Pilots to serve as experimental and validation environments, ensuring XR5.0's adaptability to various industry scenarios and its alignment with enterprise systems. The pilot "box" in this diagram could represent any Industry 5.0 environment.

The System Context Diagram presented in this chapter aims at providing a high-level view of the XR5.0 system and shows how the system interacts with external entities, such as users, pilots, etc. Thereafter it focuses on:

- **The XR5.0 system as a whole: Represented** as a single box.
- **External interactions:** Highlighting relationships between the XR5.0 system and its external actors (e.g. users, pilots)
- **Purpose and boundaries:** Clarifying what the system does and where its responsibilities begin and end.

To this end, in this diagram all actors, as stated in Deliverable D2.1, have been presented in every small detail to focus on their specific needs. In the other diagrams throughout this document, for reasons of simplicity all these actors have been merged into 2 different personas: (i) the Training Manager who creates the training workflows and (ii) the operator who makes use of XR5.0 technology in order to facilitate his operations in the Industry 5.0 environment.

6.2 Container View

Following the System Context Diagram, the Container Diagram divides the system into its principal structural components, or "containers," which represent applications, services, databases, or other operational elements. The diagram illustrates how these containers communicate, the technologies employed, and the pathways through which data flows within the system. The Container Diagram provides a detailed view of how the XR5.0 architecture is structured and how it interacts, establishing a deeper understanding of the system's technical foundation. Together with the System Context Diagram, it helps stakeholders grasp the system's architecture, paving the way for further investigation into the underlying components and code.

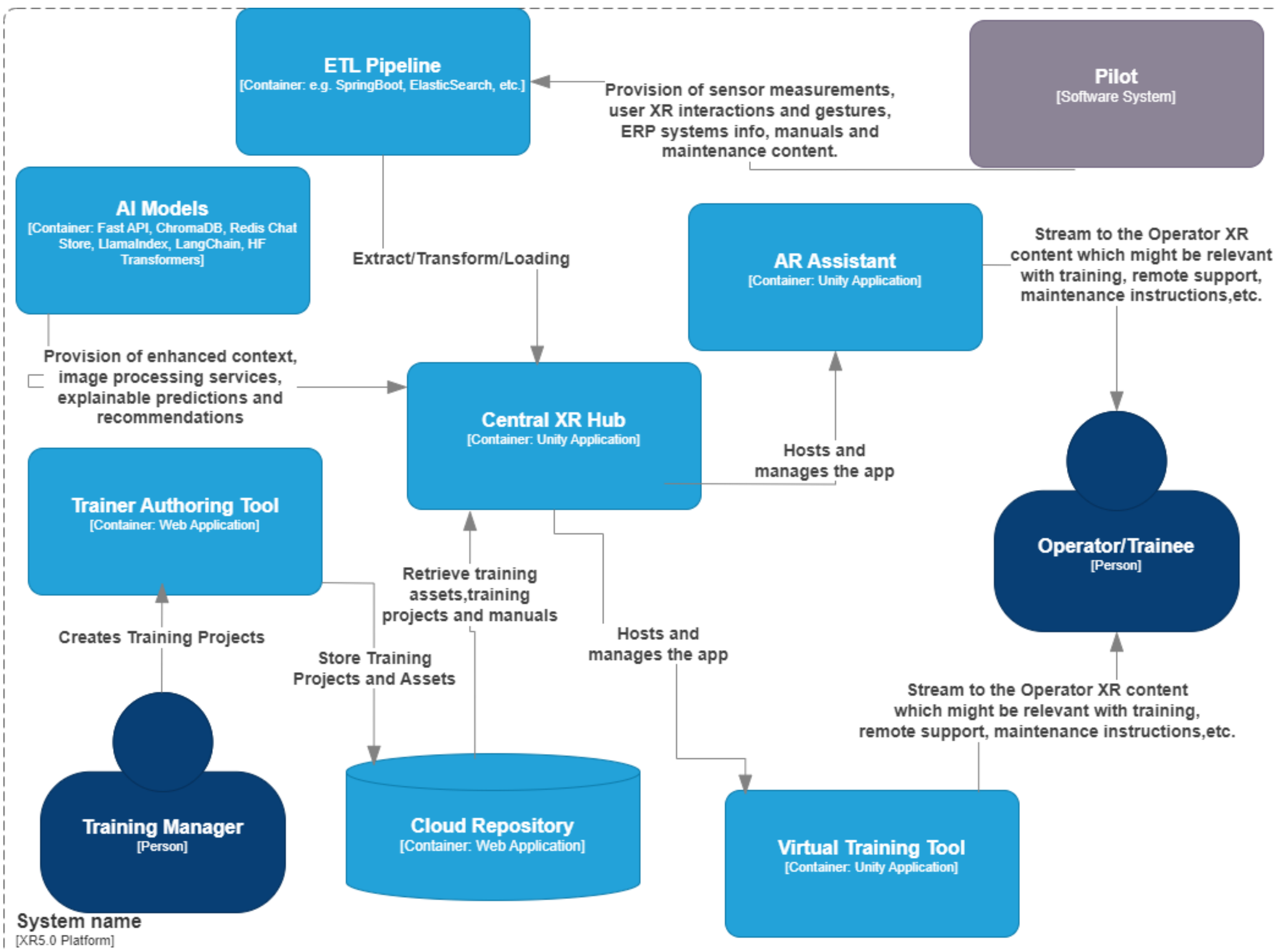


Figure 31 - Container diagram of Solution Architecture

According to C4 Model specification [4] a container is a separately runnable/deployable unit (e.g. a separate process space) that executes code or stores data. Thereafter, every XR5.0 component that has been separately deployed in the project, has been identified in this diagram as a container. The Container diagram shows the high-level shape of the software architecture and how responsibilities are distributed across it.

It also shows the major technological choices and how the containers communicate with one another. The containers. persons and software systems presented in the diagram in Figure 31 are the following:

- **The Pilot**: Each pilot is considered to be a Software System in the Container Diagram according to C4 specification. In a broader perspective, the grey box named "Pilot" could also represent any Industry 5.0 environment. The Pilot provides the ETL Pipeline sensor measurements, user XR interactions and gestures, ERP information and manuals and maintenance content. In other words, every piece of information that could be exploitable by the XR5.0 system is potentially provided to XR5.0 through the ETL Pipeline.
- **ETL Pipeline**: The Extract-Transform-Load (ETL) Pipeline is a Container. It is a core data integration component designed to facilitate the seamless transfer of data from pilot sites to the central hub. Its primary function is to ensure efficient and reliable ingestion, processing and storage of data across distributed environments while maintaining data quality and integrity. It interfaces with heterogeneous data sources across pilot sites, including sensors, databases, APIs, etc. It supports diverse data formats, and it implements secure and efficient data extraction mechanisms to minimize network latency and overhead. It standardizes data from diverse sources into a unified format suitable for analysis and storage and it transfers processed data to the central hub for XR visualization, analysis or storage.
- **AI Models:** For simplicity reasons, all AI Models (XAI Model, Active Model, Generative AI, NeuroSymbolic Model) have been placed as a single container in the diagram. These models provide to the Central Hub (and more specifically to the AR applications) AI processing capabilities and other services, such as image recognition, human-AI collaboration, XR content generation, scenarios generation, operational optimization, explanations of AI decisions, etc.
- **Central XR Hub:** This Container is the orchestration platform that hosts and manages the various XR applications. Thereafter it is the underlying backbone technology of XR5.0 system and hence all XR applications that have been used within each pilot have been hosted on the hub. It provides centralized application management and secure access to 3D content from anywhere, on any device.
- **Cloud Repository:** This container is a cloud-based storage solution that hosts XR and conventional training modules. The storage solution is based on WebDAV and it is made available as RESTful service. The XR applications that are managed by the Central XR Hub can retrieve the content of the repository.
- **Training Manager**: The Training Manager plays a pivotal role in bridging human expertise and advanced technologies, ensuring a harmonious and productive collaboration between people and machines. He identifies skill gaps and provides training programs to keep employees updated with emerging technologies and processes in Industry 5.0. Moreover, he designs human-centric training projects tailored to individual workers' needs, considering their learning styles, roles and responsibilities. By leveraging XR and AI-driven platforms he paves the way for operators' adaptive learning and personalized skill development. In this way, he ensures operators are engaged, informed and motivated as they adapt to Industry 5.0 processes. He makes use of the Trainer Authoring Tool in order to create training projects.
- **Operator**: For simplicity reasons, in C4 Container Diagram, the Operator represents all personas that have been recognized through the pilot user stories, apart from the Training Manager. Thereafter, the Operator is the persona that makes use of XR5.0 technologies so as to: (i) oversee automated systems and machines, (ii) identify anomalies, (iii) troubleshoot errors (iv) analyze real-time data generated by IoT devices and sensors, (v) resolve technical issues (vi) adjust parameters as required for the production workflow and (vii) validate final outputs. To this end, XR5.0 streams XR content to the Operator in order to facilitate his operations. This XR content might be relevant with training, remote support, maintenance instructions, etc.

- ➢ **Trainer Authoring Tool**: The Trainer Authoring Tool is a tool used for creating training projects and programs. It is primarily used by Training Managers, and it automatically stores all the training assets and programs into the Cloud Repository. Each training project is defined by many attributes such as name, description, learning objectives, learning time, target audience, level, requirements, industrial domain and activities. Furthermore, each Training Program is associated with many XR-based materials which are defined by the attributes: name, description, type and configuration.
- ➢ **AR Assistant**: AR Assistant is an XR app managed through the Central XR Hub. This AR app is capable of the following: (i)visualizing the damage location, (ii) consuming AI services in order to analyze the damage, (iii) visualizing sensor data and (iv) providing checklists in the AR environment It offers text to speech and speech to text service, and it can visualize virtual multimedia content (3D models, images, metadata, videos, etc.). It is a Unity-based application, and it has been integrated into the Central Hub. With the use of the Streaming plugin, it can stream relevant XR content to the operator.
- ➢ **Virtual Training Tool**: The Maintenance Training Tool provides a virtual reality environment to be used in off-the-job settings providing a step-by-step procedure for handling devices. This VR app offers training in procedures that require following a predetermined sequence for the maintenance of equipment in industrial settings. It immerses technicians in interactive XR environments for virtual training on maintenance procedures. The app displays step-by-step instructions of selected maintenance procedures, allowing technicians to interact with a realistic 3D replica of the target equipment using virtual projections of their hands. It has been integrated into the Central Hub and with the use of the streaming plugin, it can stream relevant XR content to the operator.

It has to be noted that in addition to the AR Assistant and the Virtual Training Tool, the Central Hub can accommodate even more XR apps according to the specific use case, so the aforementioned apps act as representatives in the C4 Container Diagram.

6.3 Component Diagrams

This section presents the component-level architecture of the XR5.0 system using the C4 modeling approach. Each diagram illustrates the internal structure, responsibilities, and interactions of individual XR5.0 components, providing a detailed view of the system beyond the high-level Solution Architecture.

The purpose of these diagrams is to support developers, integrators, and stakeholders in understanding the design, dependencies, and interfaces of each component. They also provide a foundation for identifying potential integration points and ensuring consistency across the architecture.

The following sub-sections (6.3.1 - 6.3.11) present each component's diagram along with a brief description of its functionality, relationships, and key interfaces.

6.3.1 Almer OS Component Diagram

The Almer OS Component Diagram, shown in Figure 32, represents the architecture of an Android-based AR platform, highlighting the software and hardware components required to deliver a flexible, context-aware augmented reality experience. This diagram provides a high-level view of the main components, their responsibilities and their interactions, illustrating how the system integrates hardware, middleware, OS services and applications to support AR scenarios.

The key components of the Almer OS Component Diagram are the following:

- ➢ **Apps & App Marketplace**:
  - ○ It represents end-user applications and the marketplace for discovering and installing AR apps.
  - ○ The applications communicate with Qualcomm Spaces and other platform services to leverage AR capabilities.

- **Qualcomm Spaces**:
  - Serves as the central AR framework that orchestrates AR content and spatial computing functionality.
  - Provides APIs and services to enable apps to interact with AR-ready OS components, sensors, and peripherals.
  - Acts as the bridge between the high-level application layer and the lower-level AR-ready AOSP services.
  - Facilitates context-aware AR experiences, device integration and the use of companion smartphone applications.
- **Smartphone Companion App**:
  - Supports device management, configuration, and remote interactions, allowing apps to coordinate AR experiences between the headset and mobile devices.
- **AR-ready AOSP Services & AOSP Framework**:
  - Provides core AR services integrated into the Android OS, including sensor data management, interaction principles, and voice assistant capabilities.
- **AOSP Native Libraries & Runtime, HAL and Linux Kernel**:
  - Represents the lower layers of the Android stack, including hardware abstraction, runtime libraries, and kernel-level drivers for sensors, chipset, and other peripherals.
- **Voice Assistant & Remote Configuration**:
  - Components that support voice-controlled interactions and remote device management, ensuring hands-free AR operation and dynamic system updates.

Concerning the interactions between components within Almer OS, the following have been identified:

- Applications interact with Qualcomm Spaces to access AR capabilities and retrieve spatial computing services.
- Qualcomm Spaces communicates with AR-ready AOSP Services to leverage system-level services like sensor access, context extraction and interaction guidelines.
- Low-level OS layers, including HAL and the Linux kernel, provide the foundational support for hardware interaction, enabling real-time AR experiences.
- Voice commands and remote configuration flows allow users and administrators to interact seamlessly with the system without direct hardware intervention.

The diagram in Figure 32 emphasizes the integration of hardware and software necessary for a robust, flexible AR platform. By structuring the system into components ranging from applications to kernel-level services, it demonstrates how Qualcomm Spaces acts as the central orchestrator, enabling developers to build AR applications that are responsive, context-aware and compatible with a wide range of devices. The design also highlights modularity, allowing the platform to support voice control, remote configuration and companion app integration for scalable and adaptable AR experiences.

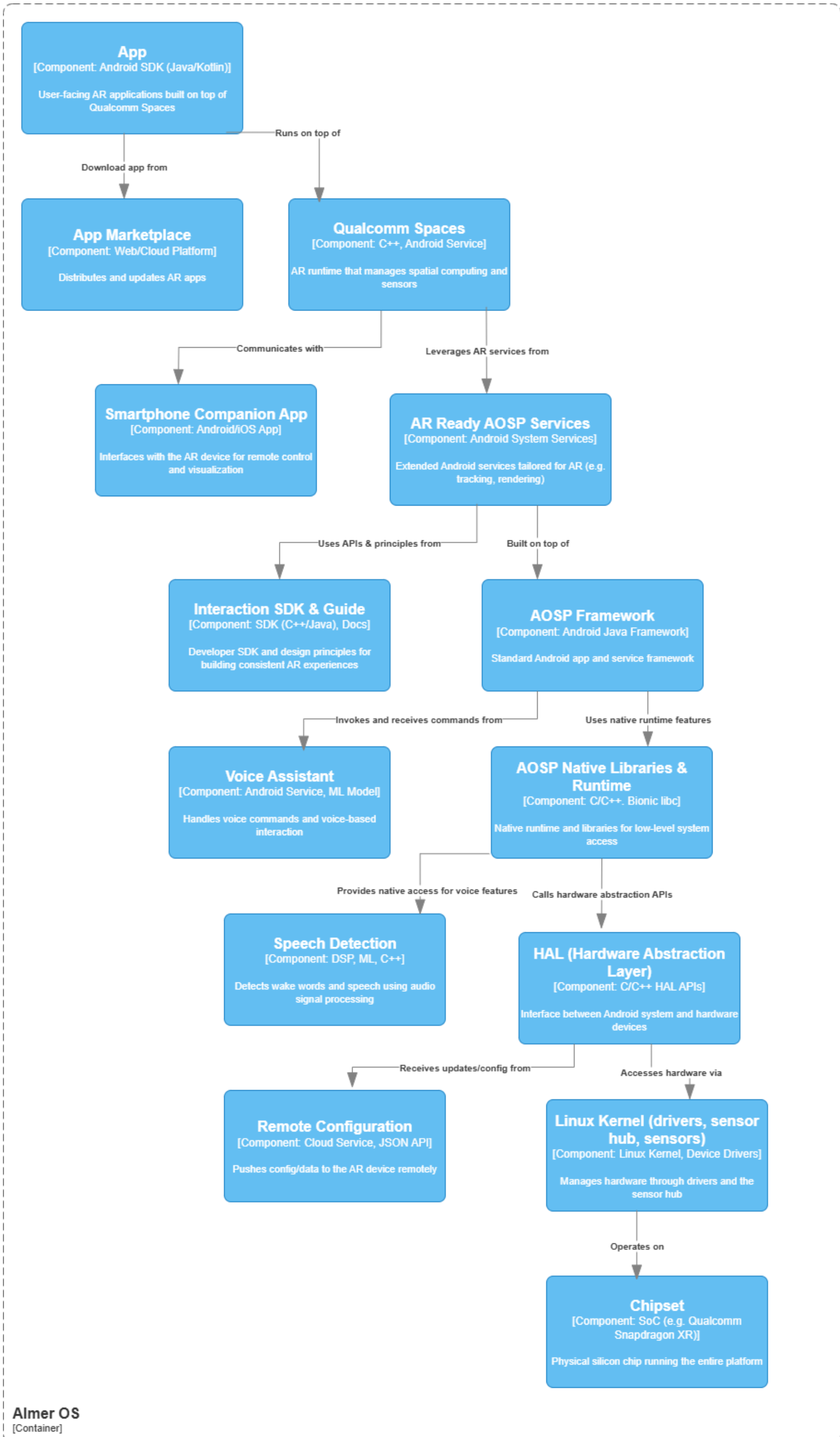


Figure 32 - Almer OS Component Diagram

6.3.2 Context-Based Explainable AI Component Diagram

The C4 Component Diagram in Figure 33 illustrates the architecture of a context-based Explainable AI (XAI) tool designed to generate domain-specific and context-aware explanations for AI outputs. The system integrates data, knowledge, and contextual information to provide transparent and interpretable insights for users, particularly within Industry 5.0 XR applications.

The key components of this tool are the following:

- **Data Collection Interface**:
  - Responsible for connecting to heterogeneous data sources (e.g., IoT devices, databases, logs, external APIs).
  - Provides standardized data ingestion mechanisms to ensure consistency and reliability.
- **Ontology/Knowledge Graph Interface:**
  - Supports the creation, editing, management, and querying of ontologies and knowledge graphs.
  - Enables the integration of domain-specific knowledge, ensuring that explanations are not only data-driven but also semantically grounded.
- **Context Extraction Interface**:
  - Loads and manages context models based on standardized data.
  - Extracts contextual information relevant to the current use case (e.g., environment conditions, operator profiles, task-specific parameters).
- **Context-Based XAI Engine**:
  - Core component that integrates inputs from the knowledge graph, context models, and collected data.
  - Generates explainable**,** contextualized outputs tailored to the user's domain and situation.
  - Combines symbolic reasoning and data-driven approaches to produce human-understandable explanations.
- **Vector Database (Vector DB)**:
  - Stores embeddings derived from domain or proprietary datasets.
  - Provides semantic search capabilities to retrieve relevant contextual information that enriches explanations.
- **XR Application (External System)**:
  - Represents the client interface where operators or domain experts receive AI explanations.
  - Provides immersive access to contextualized insights in industrial training and operations.

Concerning the interactions between the components, the following have been identified:

- **XR App → XAI API:** Users submit voice or text commands that are transformed into queries.
- **XAI Engine → Vector DB:** Searches for relevant domain-specific information to enrich the explanation.
- **Vector DB → XAI Engine:** Returns embeddings and contextual data derived from domain/proprietary datasets.

- ➢ **XAI Engine → LLM** (OpenAI or similar)**:** Sends enhanced prompts combining data, context, and knowledge for explanation generation.
- ➢ **LLM → XAI Engine:** Returns generated responses.
- ➢ **XAI Engine → XR App:** Provides the personalized**,** context-based explanation back to the user.

In summary, this architecture emphasizes the fusion of domain knowledge, contextual information and AI reasoning to generate explanations that are not only accurate but also meaningful to end users. The XAI tool goes beyond generic interpretability techniques by leveraging ontologies**,** context models**,** and embeddings, ensuring that explanations are adapted to the specific industrial scenario. In the context of Industry 5.0, this supports operator training, safety, and decision-making by providing trustworthy AI outputs in XR environments.

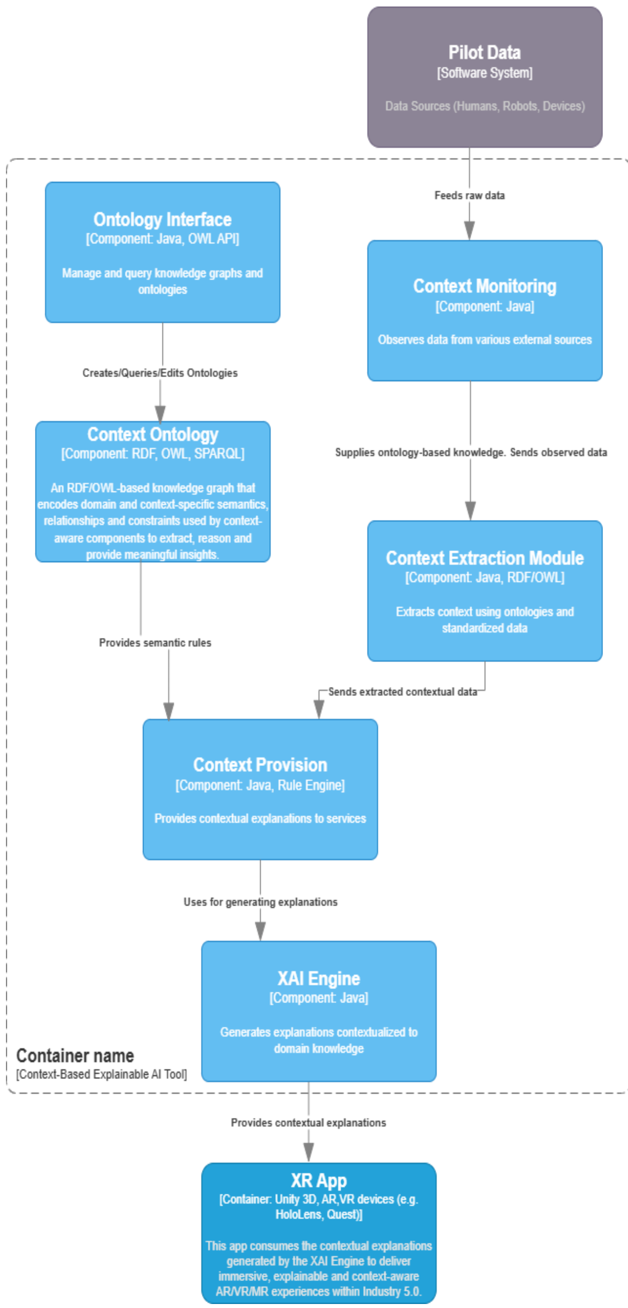


Figure 33 - Context-Based Explainable AI Tool Component Diagram

6.3.3 Operator 5.0 Training Platform Component Diagram

The XR5.0 Training Platform is a cloud-based XR training and streaming platform that enables Industry 5.0 operators to be able to run XR training applications on -demand for various industrial scenarios within the I5.0 ecosystem. The end-users/operators will be able to leverage the training platform to undergo immersive training in AR/VR mode with real-time interactions. The platform is based on HOLO's enterprise streaming platform Hololight Hub, which in-itself is an XR platform for hosting, managing and collating enterprise XR applications. The platform provides a cloud repository that enables industrial enterprises to upload and store their training materials and assets. The repository is integrated with the XR platform to enable streaming of XR training assets in real-time to end-users. In addition, XR5.0 Training Platform offers the industrial enterprise the functionality to design and develop immersive training programs that will address the needs of various operators based on their expertise and skills in relation to specific maintenance and troubleshooting tasks. As a result, it will enable industrial operators to be completely immersed during training sessions in a high-quality and robust virtual environment grounded in a real-world industrial context.

Figure 34 depicts the internal architecture of the XR5.0 Training Platform in the C4 Component Diagram specification. As shown in Figure 34, in addition to hosting and streaming XR applications to end devices (i.e. AR/VR headsets), the platform also supports administrative operations through web browsers and seamless management of applications and users. Moreover, it offers robust orchestration of cloud resources and efficient XR content delivery.

According to Figure 34, the core components of XR5.0 Training Platform are the following:

- **Cloud Repository Interface**: This module is needed for interfacing the Cloud Repository.
- **Authoring Tool Interface**: This module is needed for interfacing the Training Programs Authoring Tool.
- **Settings Module**: The Settings Module is the core configuration and control unit of the XR5.0 Training Platform. It allows users to manage system preferences and adjust parameters.
- **User Management Module**: The User Management Module handles identity, access control and role-based permissions across the XR5.0 Operator Platform. It supports multi-tenant architecture and independent administration for each pilot.
- **Front-end admin view component**: This module enables access of the Training Platform through a custom front-end admin view.
- **Applications Module**: This module is needed for interfacing the Hololight Hub in order to manage the XR applications.
- **Training Repository**: It consists of the back end and front-end of the cloud base storage solution that hosts XR and conventional training modules. This tool is supported by a WebDAV server and offers RESTful services for the provision of resources.
- **Training Programs Authoring Tool**: It's a web application for managers to create and manage XR training programs and materials. It is exposed to trainers through the Authoring Tool Interface.
- **Hololight Hub**: Hololight Hub is the underlying backbone technology of XR5.0 Training Platform and hence all applications developed can be hosted on Hololight Hub.
- **Trainer**: A human expert who uses XR5.0 Training Platform to deliver personalized, contextual and adaptive learning experiences.
- **Administrator**: The persona who administers the XR5.0 Training Platform.
- **Developer**: The persona who develops XR apps.

The following interactions have been identified for this tool:

- ➢ **Trainer → Cloud Repository Interface → Training Repository:** The trainer creates and stores training assets and projects through the Cloud Repository Interface. The assets are permanently stored in the Training Repository database.
- ➢ **Trainer → Authoring Tool Interface → Training Programs Authoring Tool**: The trainer creates training projects through the Authoring Tool Interface which is hosted by the XR5.0 Training Platform. Then, the Authoring Tool Interface contacts the Training Programs Authoring Tool.
- ➢ **Administrator → Settings Module → Hololight Hub**: The administrator configures the XR5.0 Training Platform through the Settings Module.
- ➢ **Administrator → User Management Module → Hololight Hub**: The administrator manages users and configures the authentication mechanisms through the User Management Module.
- ➢ **Administrator → Front-end admin view component → Hololight Hub**: The administrator accesses the front-end of XR5.0 Training Platform through the Front-end admin view component.
- ➢ **Developer → Applications Module → Hololight Hub**: The developer is able to manage his XR applications through the Applications Module.

In conclusion, the XR5.0 Training Platform is designed to support immersive training in Industry 5.0 environments. It integrates XR applications, authoring tools, training program management and adaptive interfaces into a unified ecosystem that enables both managers and operators to benefit from personalized and interactive training experiences.

The platform provides managers with authoring and program management tools to create, assign and monitor training materials, while operators access these programs through immersive XR plugins and adaptive user interfaces. The inclusion of AI-driven components such as context-based explainability, adaptive training and voice assistance ensures that training sessions are tailored to the specific needs of the worker and the industrial context.

By combining immersive XR technologies, data-driven personalization and cross-device interoperability, the XR5.0 Training Platform offers an innovative, human-centric approach to workforce development. It not only improves training efficiency and worker safety but also fosters adaptability and resilience aligning with the principles of Industry 5.0.

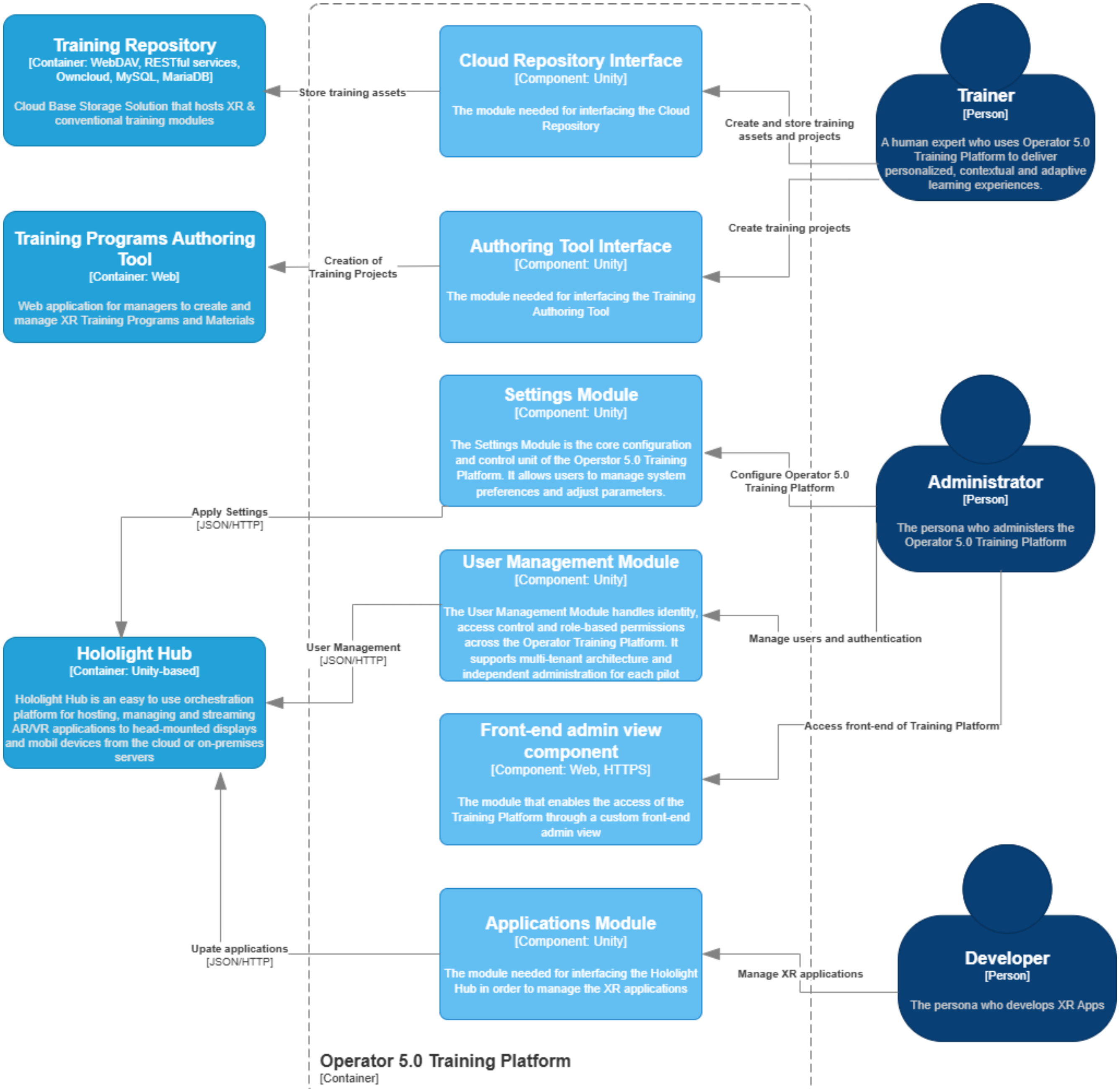


Figure 34 - XR5.0 Training Platform Component Diagram

6.3.4 LeanXcale Component Diagram

The LeanXcale Component Diagram in Figure 35 describes the architecture of an ultra-scalable relational database system designed for big data management and real-time analytics. The system integrates relational database principles with distributed architectures, enabling scalability, interoperability with big data ecosystems, and low-latency query execution.

The key components of the tool are the following:

➢ **Data Lake**:
  - Stores large volumes of raw historical and real-time data.
  - Acts as the primary ingestion point for structured, semi-structured and unstructured data.

- **Query Engine**:
  - Provides the interface for processing SQL-like queries across distributed data nodes.
  - Ensures efficient query planning and optimization for both analytical and transactional workloads.
- **Data Nodes**:
  - Distributed storage and processing units that hold partitions of the dataset.
  - Enable horizontal scalability and fault tolerance.
  - Connected bidirectionally with the Query Engine to handle query execution and data retrieval.
- **Transactional Engine & Data Node Management**:
  - Ensures ACID compliance, transaction handling and coordination between distributed nodes.
  - Provides metadata services and global consistency management across the system.
- **Direct API**:
  - Exposes database services via a direct programmatic interface.
  - Used for low-latency access, bypassing JDBC/ODBC layers.
  - Connects with external connectors such as Spark for real-time data integration.
- **Spark Connectors**:
  - Provide integration with big data and streaming ecosystems such as Apache Spark, Flink, Kafka.
  - Facilitate data ingestion, real-time stream processing and analytical workflows.
- **External Systems**:
  - **Data Sources**: Raw input from XR devices, IoT, enterprise systems, or legacy databases.
  - **Mobile Applications**: Consume processed analytical results in Industry 5.0 settings.

Concerning the interactions between the components, the following have been identified:

- **Data Lake → Query Engine:** Provides bulk or historical datasets for querying.
- **Query Engine ↔ Data Nodes**: Executes distributed queries and retrieves data partitions.
- **Query Engines (Cluster) ↔ JDBC/ODBC Interface**: Supports standardized SQL access for client applications
- **Data Nodes ↔ Direct API**: Allow direct programmatic access for low-latency applications.
- **Direct API ↔ Spark Connectors**: Facilitate real-time integration with big data frameworks.
- **Transaction Engine ↔ Query Engines/Data Nodes**: Manages transaction coordination, consistency and recovery.
- **Spark Connectors ↔ External Systems**: Enable ingestion of real-time or historical data and dissemination of analytical results to consumers.

In summary, the C4 component diagram of LeanXcale tool illustrates how the Ultra-Scalable Relational Database bridges traditional relational database models with modern big data and streaming requirements. By combining query engines, distributed data nodes and a transactional engine, it delivers scalability, consistency and performance. Its support for JDBC/ODBC compliance and custom connectors for Spark, Flink and Kafka ensures smooth integration into existing enterprise and Industry 5.0 ecosystems.

In industrial XR scenarios, the database enables both historical analysis (e.g. long-term equipment performance) and real-time analytics (e.g. device telemetry), making it a cornerstone for decision support, predictive maintenance and immersive data-driven applications.

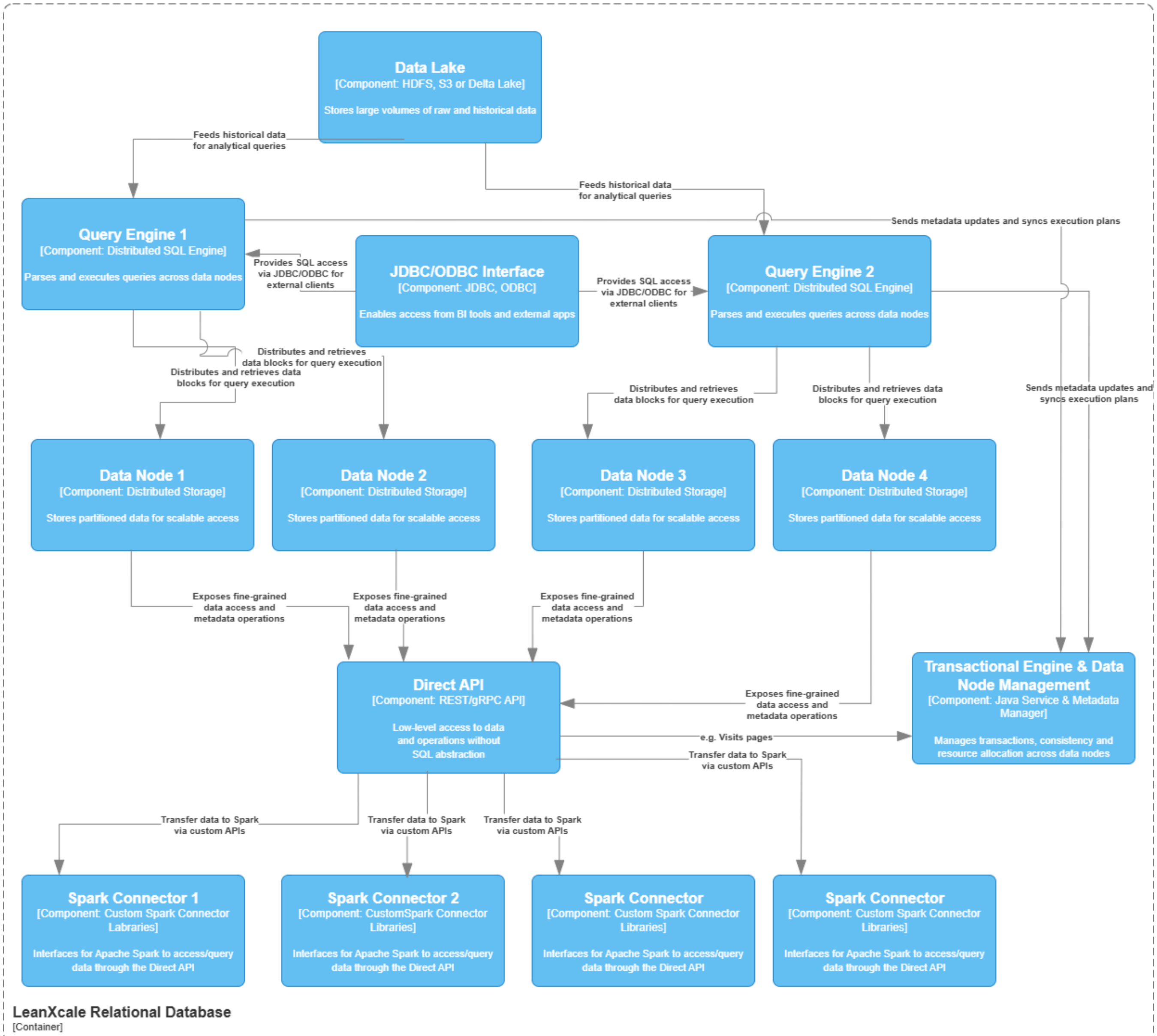


Figure 35 - LeanXcale Relational Database Component Diagram

6.3.5 Generative AI Component Diagram

The C4 Component Diagram of Generative AI tool illustrates the architecture of a system designed to provide personalized, context-aware responses for users in immersive XR environments. This tool that is depicted in Figure 36, integrates advanced natural language processing with contextual retrieval mechanisms, ensuring that responses are accurate, domain-specific and adapted to the user's interaction mode (voice or text).

The Generative AI tool consists of the following key components:

- ➢ **XR Application (External System)**:
  - The client-facing application where users interact through voice or text commands.
  - Provides an immersive interface for receiving contextualized responses from the AI system.
- ➢ **XR_AI API**:
  - Acts as the entry point for all requests from XR applications
  - Manages input parsing (speech/text) and output delivery
  - Provides REST endpoints for integration with XR platforms.
- ➢ **Prompt Manager**:
  - Structures and optimizes user input into well-formed prompts suitable for large language models.
  - Enriches prompts with contextual metadata and ensures consistency in interactions.
- ➢ **Functions**:
  - Encapsulate system-level capabilities, external APIs, or integration points with other services.
  - Allow the AI to go beyond text generation and interact with external logic or datasets.
- ➢ **Agents**:
  - Orchestrate reasoning processes, deciding when to query external knowledge, invoke functions, or reformat outputs.
  - Support adaptive responses by combining LLM outputs with contextual knowledge.
- ➢ **Vector Database (Vector DB)**:
  - Stores embeddings derived from domain or proprietary datasets.
  - Provides semantic search to retrieve relevant contextual information that enriches queries.
- ➢ **Large Language Model (OpenAI or similar)**:
  - Core AI engine is responsible for generating natural language responses.
  - Operates on enhanced prompts (user query + contextual data + domain knowledge) to ensure relevance.

The following interactions occur between the components of the Generative AI tool:

- ➢ **XR App → XR-AI API:** Users send voice or text commands.
- ➢ **XR-AI API → Prompt Manager:** Processes input into prompts and queries.
- ➢ **Prompt Manager → Vector DB:** Retrieve relevant contextual information from domain datasets.
- ➢ **Agents → LLM:** Forward enriched queries combining user input, context, and retrieved knowledge.
- ➢ **LLM → Agents:** Return generated responses for post-processing and personalization.
- ➢ **Agents → XR-AI API → XR App:** Deliver personalized responses as either text or speech.

This architecture highlights how the Generative AI tool blends large language models with contextual retrieval and domain knowledge to produce trustworthy and useful responses in XR settings. By combining the Prompt Manager, Functions and Agents with the Vector DB and LLM, the system ensures that outputs are not only technically accurate but also adapted to the user's task and environment.

In the context of Industry 5.0, this tool enables operators, trainers and workers to receive on-demand explanations, instructions and contextual guidance, enhancing both productivity and safety in XR-based industrial workflows.

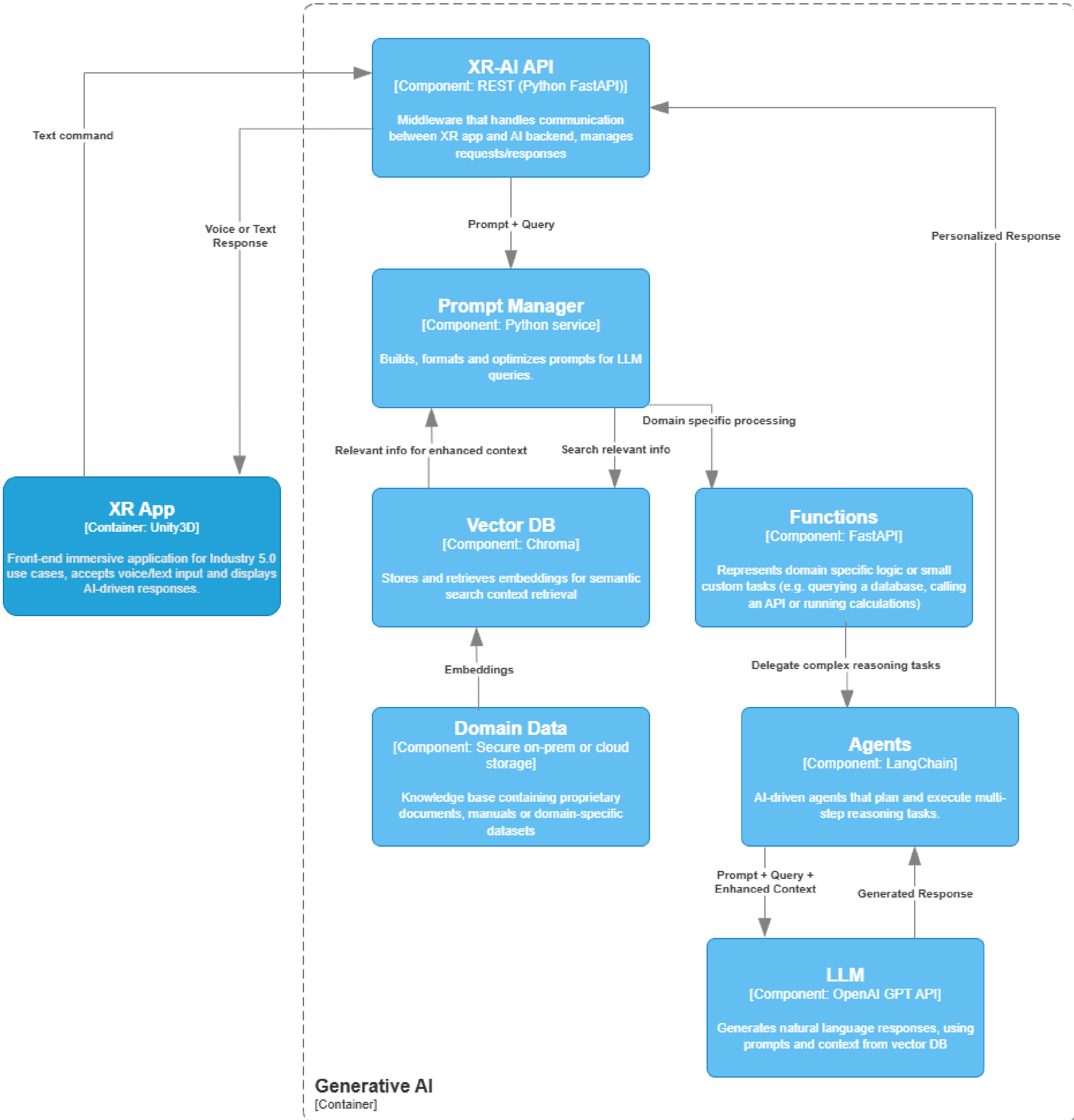


Figure 36 - Generative AI Component Diagram

6.3.6 Oculavis SHARE Component Diagram

The Oculavis SHARE Component Diagram, which is depicted in Figure 37, illustrates the architecture of the SHARE Platform, a smart service and connected worker solution designed for machine and equipment manufacturers. The platform is device-agnostic and enables technicians, experts and users to collaborate in real time across web, mobile and XR devices. Its primary goal is to support remote assistance, immersive collaboration and access to digital assets in industrial environments.

The Oculavis SHARE Component Diagram has the following key components:

- **Clients: Web, Android, iOS (Software: SHARE)**:
  - End-user applications providing access to the SHARE platform through traditional devices.
  - Enable communication with the SHARE Server for session management and with the Media Server for real-time streaming.
- **Clients: HL2, Meta Quest Pro, Magic Leap 2 (Software: MR SHARE)**
  - XR applications providing immersive collaboration experiences.
  - Support streaming via WebRTC and content synchronization with the SHARE Server.
- **SHARE Server**:
  - Central coordination component that manages user sessions, authentication, data synchronization, and workflow logic.
  - Interfaces with the Media Server for real-time communications and with the 3D Assets Micro-Service for content delivery.
- **(WebRTC) Media Server**:
  - Handles real-time audio, video, and data streaming between clients.
  - Works in conjunction with the SHARE Server for session control and signaling.
- **Micro-Service: 3D Assets**:
  - Manages 3D models and digital twins used in XR training, remote support, or equipment visualization.
  - Provides assets to both the SHARE Server and XR clients via defined protocols.

Concerning the internal interactions between the components, the following have been identified:

- **Clients (Web/Android/iOS) → SHARE Server:** User requests, session management, and data synchronization (via WebSocket, HTTPS).
- **Clients (Web/Android/iOS) ↔ Media Server:** Real-time audio, video, and data streaming (via WebRTC).
- **Clients (HL2/Meta Quest Pro/Magic Leap 2) → SHARE Server:** User authentication, collaboration events, and workflow synchronization (via WebSocket, HTTPS).
- **Clients (HL2/Meta Quest Pro/Magic Leap 2) ↔ Media Server:** Immersive audio/video/data streaming (via WebRTC).
- **SHARE Server ↔ Media Server:** Signaling and session control (via WebSocket).
- **SHARE Server ↔ Micro-Service: 3D Assets:** 3D asset management, queries, and retrieval (via REST API / WebSocket).
- **Micro-Service: 3D Assets ↔ XR Clients:** Delivery of 3D models and digital twins (protocol TBD, likely gRPC or custom streaming).

Overall, the SHARE platform architecture demonstrates how multi-device interoperability and real-time collaboration are achieved through the integration of servers, micro-services, and immersive clients. By combining Web, Mobile and XR applications with WebRTC-based media streaming and 3D asset management, the system enables global collaboration between experts, technicians and end-users.

In the context of Industry 5.0, SHARE supports connected worker scenarios, allowing field operators and remote experts to exchange knowledge, visualize digital twins and resolve issues efficiently, regardless of location or device.

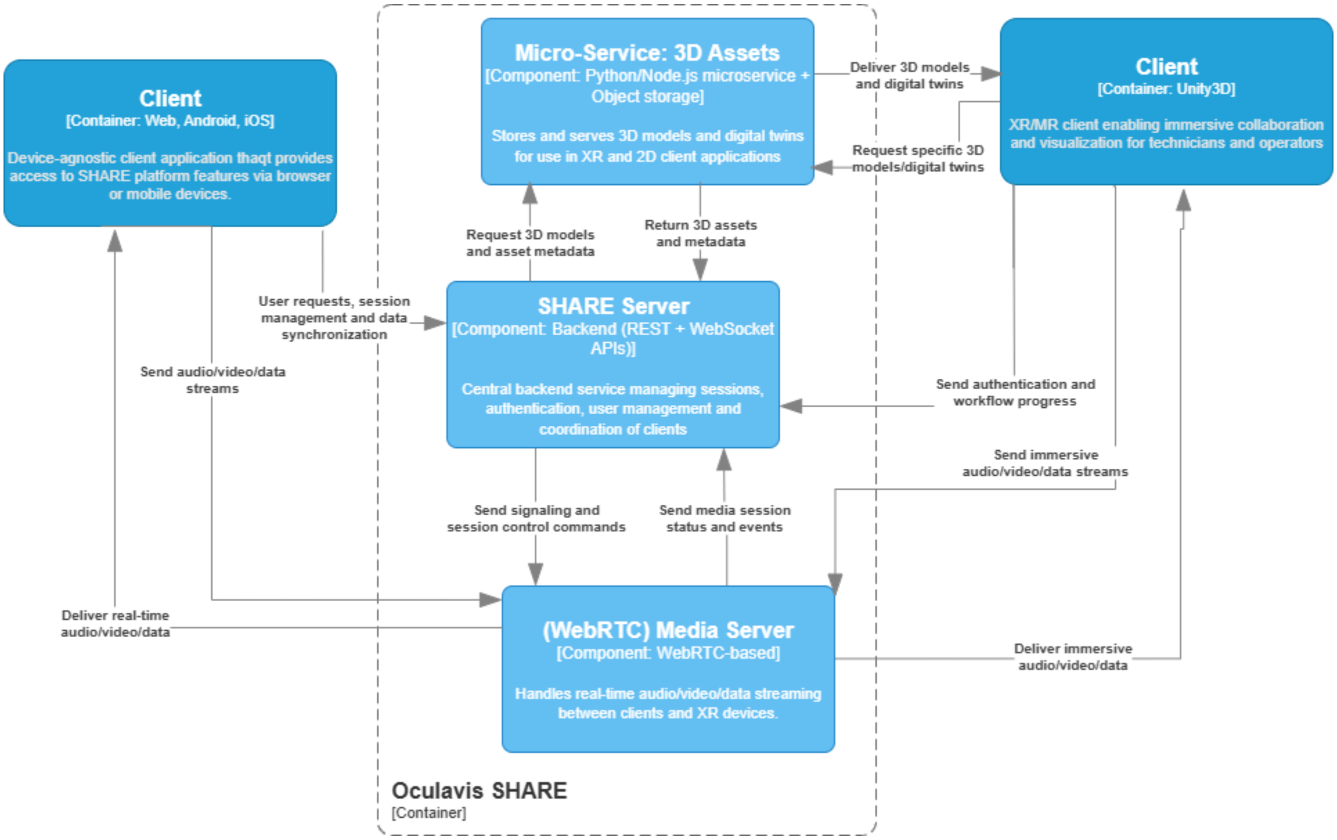


Figure 37 - Oculavis SHARE Component Diagram

6.3.7 AI Voice Assistant Component Diagram

The C4 Component Diagram depicted in Figure 38, illustrates the architecture of the AI Voice Assistant, an application designed to help users navigate and interact with virtual and augmented reality environments using natural voice commands. The system integrates speech-to-text, text-to-speech and large language models to provide responsive, context-aware and personalized assistance for XR users.

As shown in Figure 38, the following key components constitute the AI Voice Assistant:

- **XR Application (External System)**:
  - Represents the client-facing immersive application used by operators or users in VR/AR environments.
  - Accepts voice or text input and displays system responses in real time.
- **Text-to-Speech (TTS) Component**:
  - Converts generated text responses into natural speech output for XR users.
  - Interfaces with TTS models to ensure high-quality, intelligible audio.
- **Speech-to-Text (STT) Component**:
  - Converts user’s spoken commands into text that can be processed by the system.
  - Interacts with STT models to accurately capture user intent.

- ➢ LLM Client:
  - ○ Sends processed queries to large language models (LLMs) to generate intelligent responses.
  - ○ Receives textual outputs from the LLM and forwards them to the TTS component or the XR App.
- ➢ **TTS / STT / LLM Models**:
  - ○ Represent the AI models used for speech synthesis, speech recognition, and language understanding.
  - ○ Enable accurate, responsive, and context-aware interactions within XR environments.

The aforementioned components interact with each other as follows:

- ➢ **XR App → Text-to-Speech / Speech-to-Text:** Sends text/speech input from the user.
- ➢ **TTS Service → TTS Models:** Converts textual output into speech for user feedback.
- ➢ **STT Service → STT Models:** Converts user speech into textual commands.
- ➢ **LLM Client → LLM Models:** Queries the language model to generate contextualized responses.
- ➢ LLM Models → **LLM Client → TTS Models → TTS Service → AR Application**: Delivers text or speech query responses back to the user.

The AI Voice Assistant Demonstrates how AI models can be integrated into XR applications to provide intuitive, voice-driven interactions. By combining speech-to-text, text-to-speech and LLMs, the system enables hands-free operation, personalized responses and real-time guidance. This architecture is particularly useful in Industry 5.0 environments, where operators benefit from voice-controlled assistance while performing complex or immersive tasks, improving efficiency, safety and user experience.

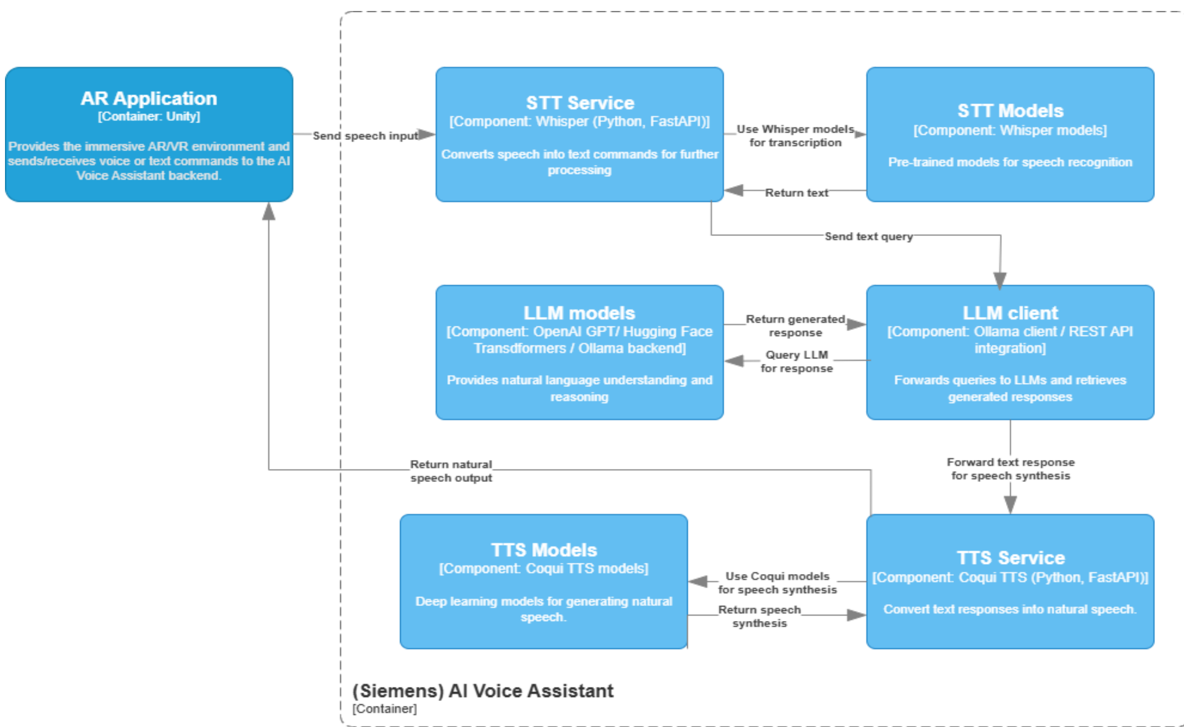


Figure 38 - Siemens AI Voice Assistant Component Diagram

6.3.8 MageAI ETL Pipeline

The MageAI ETL Pipeline is a system designed to ingest, transform and store data from multiple sources for analytics, reporting and downstream applications. As shown in the C4 component diagram of Figure 39, this pipeline enables scalable, automated data processing and provides a modular architecture that integrates data from heterogeneous sources into a unified, clean, and quarriable format.

Its key components are the following:

- **Pipe Sensors**:
  - The sensors that measure values relevant with the pipes. These measurements (e.g. vibrations) dictate whether there is damage in the pipes.
  - According to the C4 specification, the pipe sensors are an external software system, in the sense that they do not belong in the XR5.0 system.
- **AR app**:
  - It’s an AR on-site application that is used by the on-site technician in order to inspect and troubleshoot the pipes.
  - This app is integrated with the XR5.0 system and the AR app receives processed pipe sensor data through the MageAI ETL pipeline.
- **Trigger**:
  - This component triggers the data preprocessing, formatting and provisioning of data, upon anomaly detection in pipes.
- **MageAI pipeline**:
  - It processes, formats and provisions time-series anomaly detection data
- **API endpoint**:
  - API endpoint for providing processed pipe sensor data to the AR app.

Concerning the interactions between the components, the following workflow is achieved:

- **On-site Technician → Trigger:** When the on-site technician opens the app, the “Trigger” component initiates data preprocessing and provides data upon detecting anomalies in the pipes. The technician then confirmed the fault.
- **Pipe sensors → Trigger:** The pipe sensors send the measurements, the coordinates of the damage location as well as the corresponding images to the “Trigger” component.
- **Trigger → MageAI pipeline:** The “Trigger” component sends processed time-series pipe sensor data (vibration, anomalies, etc.) to the MageAI pipeline component.
- **MageAI pipeline → API endpoint:** The MageAI pipeline exposes the processed sensor data to the API endpoint.
- **API endpoint → AR app:** The API endpoint exposes the processed pipe sensor data to the XR app.

In summary, the MageAI ETL pipeline architecture highlights a modular, scalable approach to data processing, enabling organizations to unify data from diverse sources and make it quarriable. By separating ingestion, transformation, storage and orchestration, the pipeline ensures data quality, consistency and reliability supporting real-time and batch analytics.

In the context of XR5.0, the pipeline provides the foundation for inspecting and troubleshooting the pipes using machine learning and decision-making workflows, allowing the operators to gain actionable insights into the damage and the overall condition of the infrastructure, enabling faster response and reduced downtime.

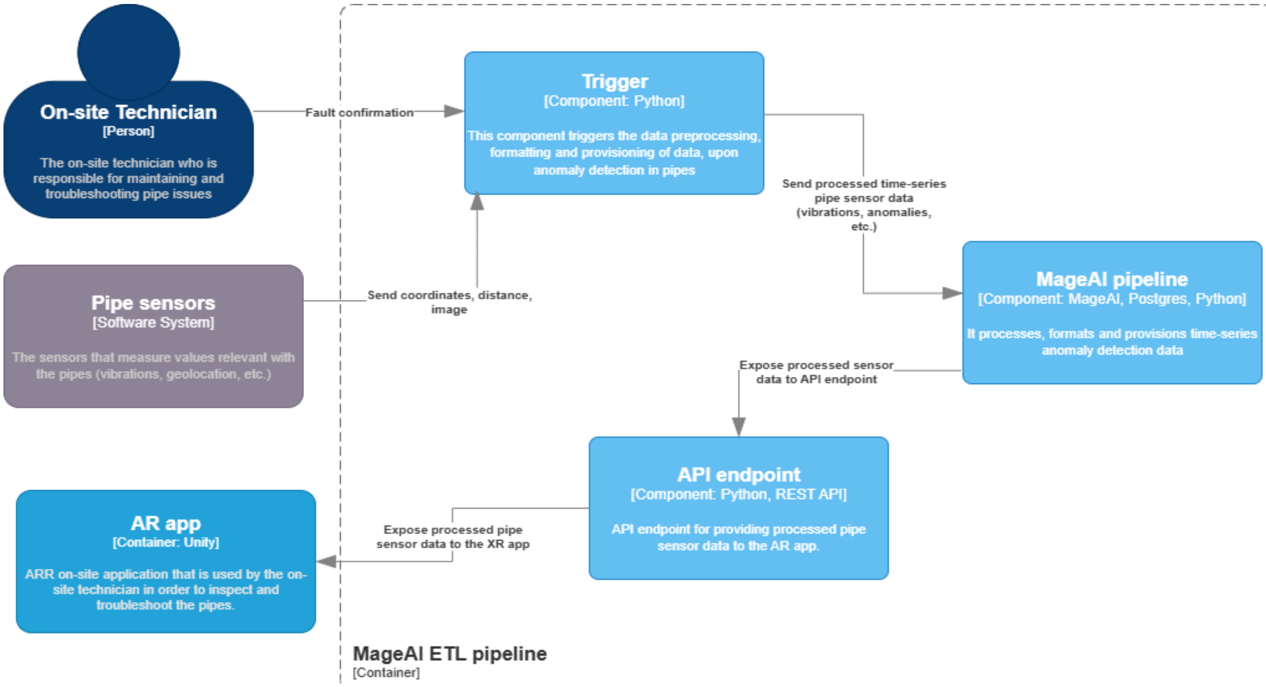


Figure 39 - MageAI ETL Pipeline Component Diagram

6.3.9 IIoT-based Platform (Clawdite)

The C4 Component Diagram depicted in Figure 40 illustrates the architecture of the Clawdite IIoT Platform, a scalable solution designed to collect, manage, and analyze industrial data from heterogeneous sources such as machines, wearables, XR devices, and autonomous systems. The platform integrates real-time data ingestion with historical data management to provide analytics-ready datasets for monitoring, prediction, and decision support in Industry 5.0 scenarios.

The key components of his platform are the following:

- **Gateways (Raspberry Pi, Smartphone, PLC/IPC)**:
  - It acts as entry points for data from machines, sensors, wearables, and XR devices.
  - Handle protocol translation and initial preprocessing before forwarding data to the IIoT middleware.
- **IIoT Middleware (e.g. Orion Context Broker, Apache Kafka)**:
  - Manages streaming data ingestion and distribution in real time.
  - Provides publish/subscribe capabilities to ensure scalable data exchange between devices and platform services.
- **Historical Data Management (HDM)**:
  - **TimeSeries (InfluxDB)**: Stores short-term, high-frequency sensor readings for monitoring and anomaly detection.
  - **Complex Data History (MongoDB):** Maintains long-term, structured and unstructured datasets, enabling advanced analysis of historical and contextual information.
- **Platform Data**:

- Centralized repository for platform-level information such as metadata, configuration, and aggregated results.

➢ **Functional Modules**:

- Extend the platform with domain-specific services, e.g., predictive maintenance, anomaly detection, and optimization engines.
- Consume data from middleware and HDM to deliver actionable insights.

➢ **Orchestrator**:

- Coordinates workflows, manages execution of functional modules, and handles integration with external systems via APIs.

As far as the component interactions are concerned, the following workflow takes place:

➢ **Machines/Wearables/XR Devices → GateWays:** Raw data acquisition and preprocessing.

➢ **Gateways → IIoT Middleware:** Streaming data forwarded in standardized formats.

➢ **IIoT Middleware → HDM (TimeSeries/Complex Data History):** Real-time and historical data storage.

➢ **HDM → Functional Modules / Orchestrator:** Retrieval of historical and live data for analysis and workflows.

➢ **Orchestrator ↔ Functional Modules:** Execution of analytics tasks and coordination of services.

In overall, the Clawdite IIoT Platform provides a modular and extensible architecture for managing the complete data lifecycle in industrial environments. By combining real-time streaming with long-term historical data management, it enables predictive analytics, decision support, and digital twin integration. It's an extensible and flexible IIoT-based platform supporting the creation of customized data representations of production systems and their entities, including humans. It also enables dynamic and quasi-static data management.

In the context of XR5.0, Clawdite empowers manufacturers and operators to leverage interoperable, human-centric, and resilient data-driven services, improving productivity, safety, and adaptability in complex industrial ecosystems. In XR5.0, the platform is being expanded to support an enhanced Digital Twins human modelling, and it will provide real-time data streaming and access for analysis and decision-making. Since it supports modular integration, three functional modules will be incorporated in order to support the following functionalities: (i) the anticipation of technician movements to be used as input for robot path planning algorithms, (ii) the provision of expert technician shadow and newbie technician monitoring information to the visor gateway, which will visualize the movements shadow while showing feedback about the actual maintenance process and (iii) the enhancement of troubleshooting process for workers.

More specifically, the Workers Movement Prediction (WMP) Functional Module is designed to detect the real-time positions and orientations of human workers in a shared industrial environment, and to forecast their future movements to support intelligent robot behavior and safety. On the other hand, the Worker Shadowing and Monitoring Module is a Functional Module that stores and provides the shadow of the expert AMT. Moreover, it tracks, monitors and elaborates feedback about the maintenance process executions by using the movement information (i.e. positions of head and hands) which are tracked directly through the visor. Finally, the Troubleshooting Assistant Service Functional Module integrates multiple modules (INNOV Generative AI, Siemens Speech-to-Text, Oculavis SHARE) to provide real-time insights and support to workers/operators, enabling them to resolve issues more efficiently. Figure 40 depicts this modular and extensible architecture which manages the Human Digital Twin (HDT) representations and the integration of various functional modules according to the respective use case to be supported in Industry 5.0.

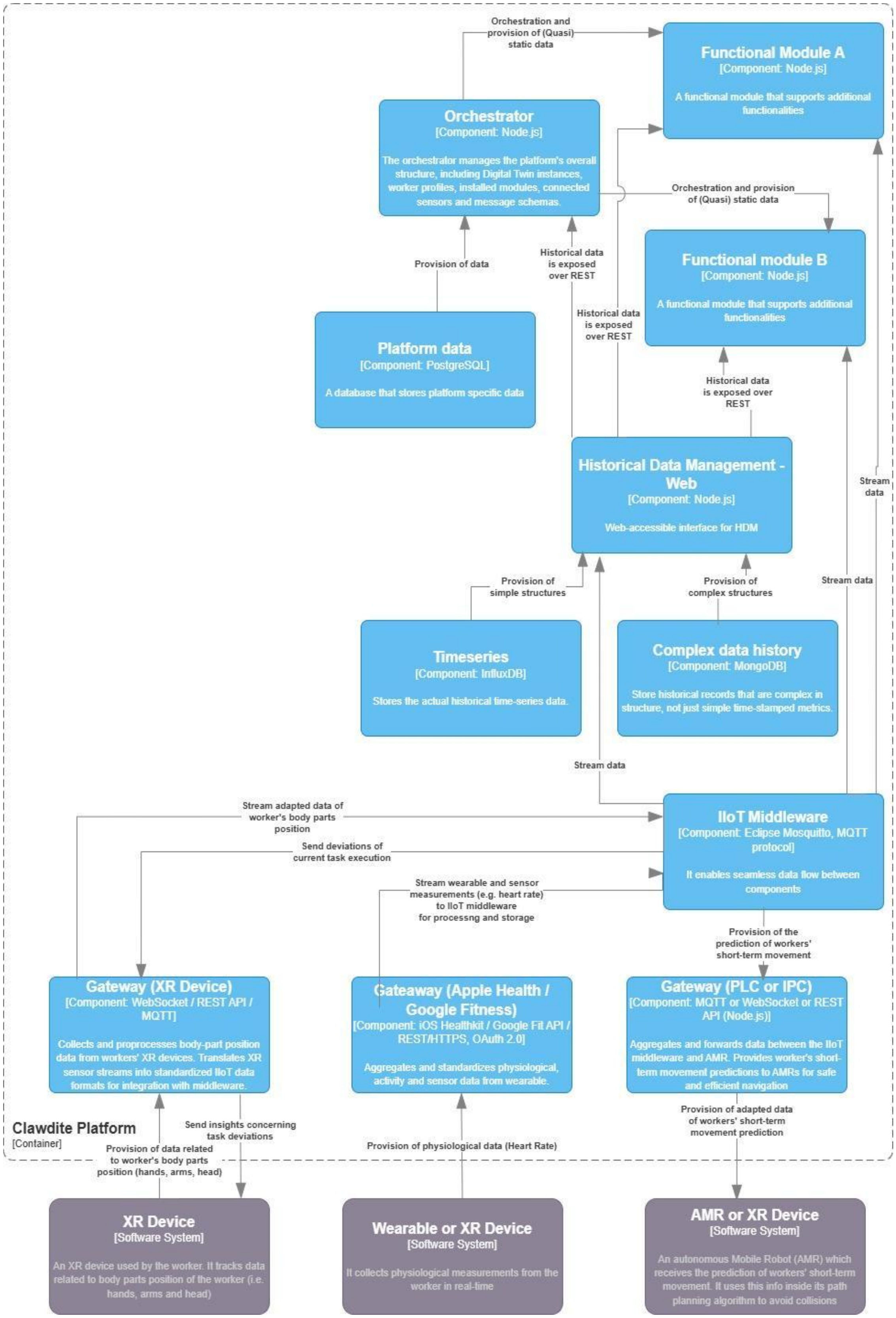


Figure 40 - IIoT-based Platform (Clawdite)

6.3.10 Training Programs Authoring Tool Component Diagram

The C4 Component Diagram depicted in Figure 41 illustrates the architecture of the Training Programs Authoring Tool, a web-based application that enables managers and instructors to create, manage and adapt XR training programs and materials. The system supports immersive learning experiences by integrating authoring, content management, and program assignment functionalities into a single, user-friendly platform.

The key components of the Training Programs Authoring Tool are the following:

- **XR Training Materials Manager**:
  - Central component for managing XR training content, including tutorials, 3D assets, simulations, and interactive materials.
  - Allows creation, editing, adaptation, and assignment of training materials to specific programs.
  - Interfaces with the XR Material Authoring Toolkit and XR Content Generator for content creation and enrichment.
- **XR Training Programs Manager**:
  - Enables managers to create, assign and manage XR training programs.
  - Integrates with the XR Training Materials Manager to link training content to specific programs.
  - Provides workflows for scheduling, grouping and tracking learner progress.
- **XR Material Authoring Toolkit**:
  - Toolset for instructors and content creators to adapt training content for different learning scenarios.
  - Supports editing, simulation preparation and customization of XR assets.
- **XR Content Generator**:
  - Automates generation of XR training materials, such as 3D simulations or interactive tutorials.
- **XR Training API**:
  - Provides programmatic access to training materials and programs, enabling integration with other systems or XR clients.
  - Supports retrieval and updates of content and program metadata.
- **Manager (Actor)**:
  - Represents the user responsible for creating, managing and assigning training programs and materials.
  - Interacts with both the XR Training Materials Manager and XR Training Programs Manager to control the full training lifecycle.

Concerning the interactions between the components of the Training Programs Authoring Tool, the following have been identified:

- **Manager → XR Training Materials Manager:** Create, manage and adapt XR training materials.
- **Manager → XR Training Programs Manager:** Create, assign and manage XR training programs.

- **XR Training Programs Manager → XR Training Materials Manager:** Assign training materials to programs.
- **XR Training Materials Manager / XR Training Programs Manager → XR Training API:** Access training content and program information programmatically.
- **XR Training Materials Manager → XR Material Authoring Toolkit: Adapt** training materials using authoring tools.
- **XR Training Materials Manager → XR Content Generator:** Generate XR training content automatically based on learning objectives.

In summary, the Training Programs Authoring Tool enables managers to efficiently design, customize and deploy XR training programs. By centralizing training materials management, program assignments and content generation, the system ensures that operators receive consistent, high-quality and immersive training experiences.

In the context of Industry 5.0, this tool supports skill development, operator training and knowledge transfer, allowing organizations to train personnel effectively in complex, technology-driven environments.

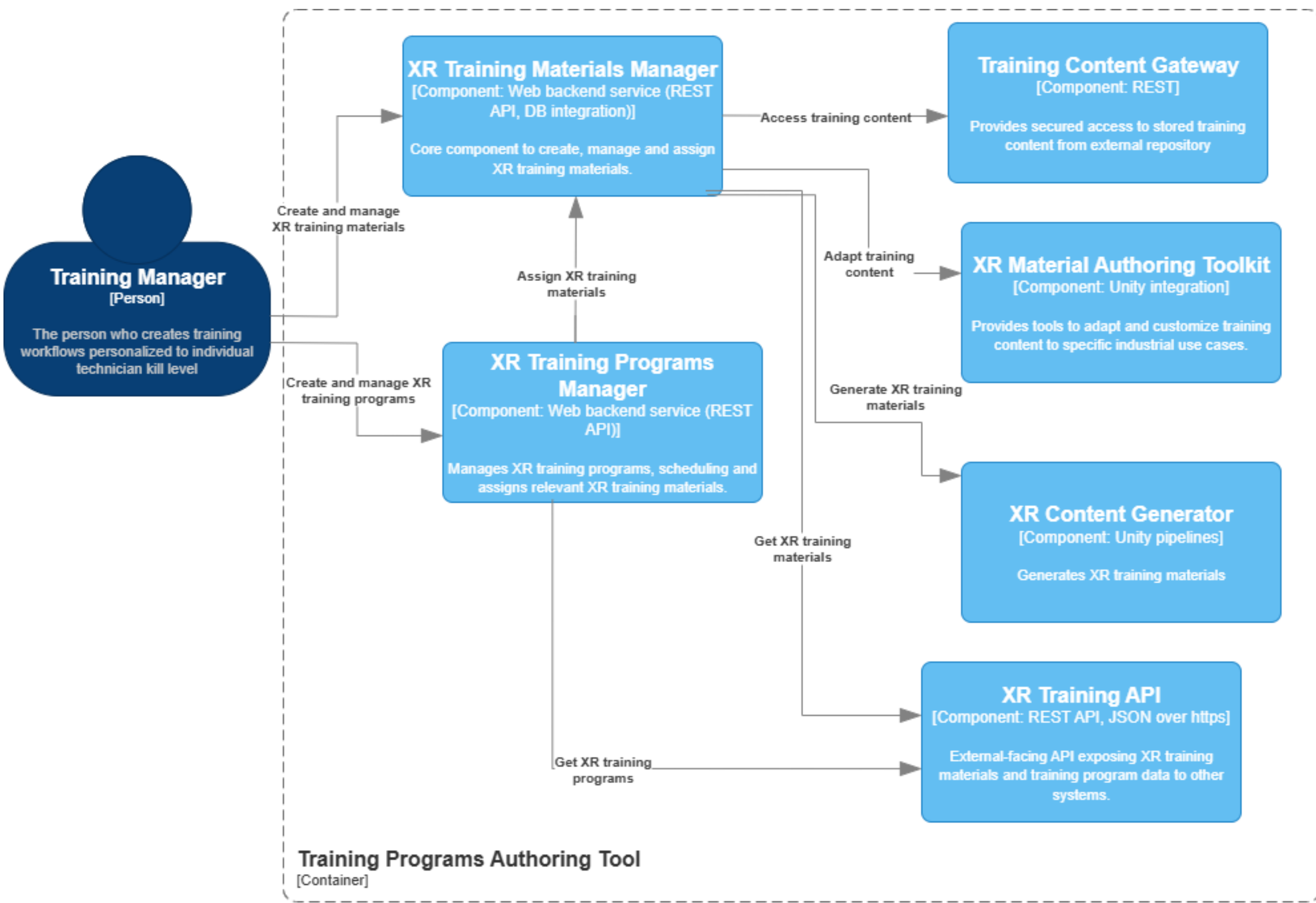


Figure 41 - Training Programs Authoring Tool Component Diagram

6.3.11 XR Training Plugin Component Diagram

As shown in Figure 42, the XR Training Plugin is a visualization tool designed to enable operators to access and interact with XR-based training programs and materials in immersive environments. It serves as a bridge between training content, adaptive interfaces and the end-user, ensuring that the delivery of XR training is both effective and user-friendly. The actor who uses the plugin is the operator who accesses XR training programs and materials, receiving adaptive and immersive interfaces to support learning activities in industrial contexts.

The core components of the Training Plugin are the following:

- ➢ **Training Program Manager**: Central controller of the plugin, responsible for orchestrating training program access, managing materials and coordinating with other services.
- ➢ **Training Data Service**: Supplies the Training Program Manager with XR training programs and materials.
- ➢ **Virtual Dashboard System**: Provides the operator with immersive management interfaces, enabling seamless interaction with the training content.
- ➢ **Training Interface Toolkit**: Handles the rendering and display of XR training materials, ensuring accessibility and usability across XR devices.
- ➢ **Adaptive Interface Engine**: Adapts interfaces based on recommendations, tailoring the XR training experience to user needs and preferences.

The following interactions between the components and the actor have been identified:

- **Operator → XR Training Program Manager**: Access XR training programs and materials using the plugin.
- **Training Program Manager → Training Data Service**: Get XR training programs and materials from the service.
- **Training Program Manager → Virtual Dashboard System**: Manage immersive interfaces using the dashboard.
- **Training Program Manager → Training Interface Toolkit**: Display XR training materials using the toolkit.
- **Training Program Manager → Adaptive Interface Engine**: Adapt interfaces using recommendations from the engine.

In summary, the XR Training Plugin ensures that XR training programs are delivered to operators in a dynamic and adaptive way. By combining content management, immersive visualization, and adaptive interface capabilities, it plays a crucial role in enhancing learning efficiency and user engagement in Industry 5.0 training scenarios.

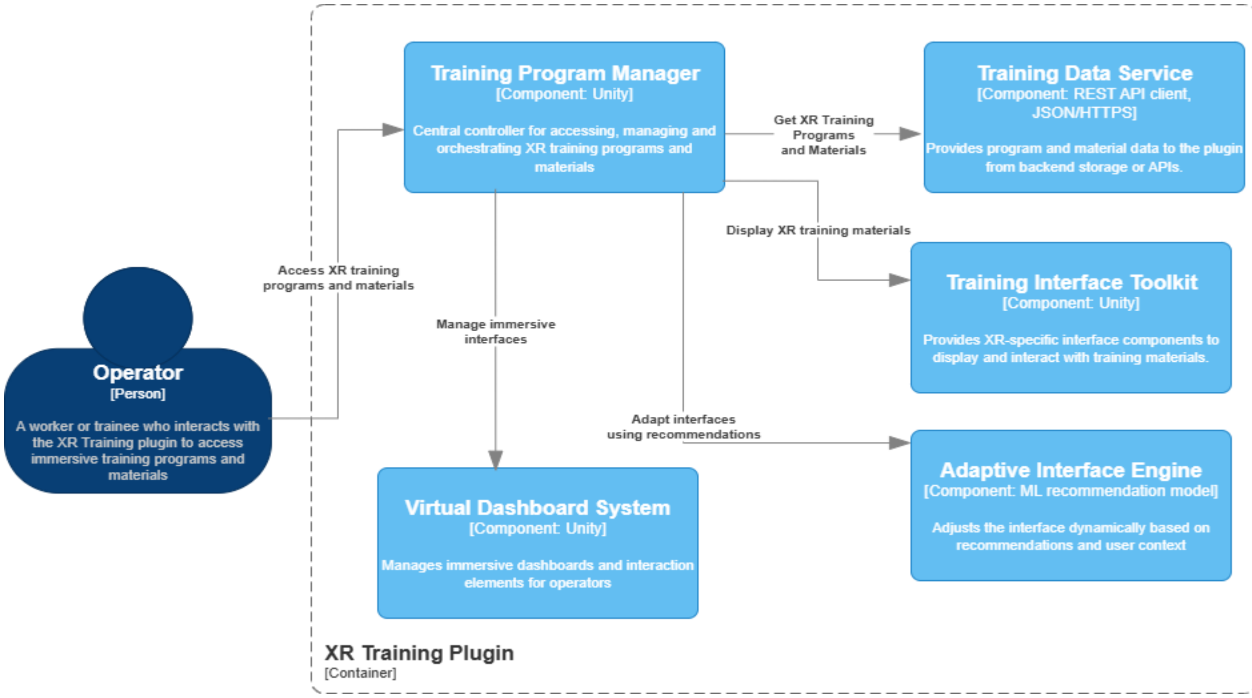


Figure 42 - XR Training Plugin Component Diagram

6.3.12 Neurosymbolic AI and Active Learning Component Diagram

The C4 Component Diagram depicted in Figure 36 illustrates the architecture of the Neurosymbolic AI and Active Learning component, a system designed to enable explainable AI (XAI) in XR and industrial contexts. It combines deep learning with symbolic reasoning and integrates human-in-the-loop feedback to ensure models are both accurate and interpretable.

The actor in the diagram is the operator. He provides annotations, feedback and domain-specific knowledge to improve model predictions, ensure explainability and guide active learning cycles.

The core components of the Neurosymbolic AI and Active Learning are the following:

- **Data Preparation**:
  - Responsible for cleaning, preprocessing and preparing datasets, both labeled and unlabeled for training and inference.
  - Supports iterative improvement by feeding curated data back into the learning process.
- **Neurosymbolic AI**:
  - Combines Neural Network Object/Outlier Detection with Symbolic Reasoning Rules Generation to produce predictions that are both accurate and interpretable.
  - Integrates symbolic reasoning with deep learning to enhance explainability and reliability.
- **Active Learning**:
  - Enables models to learn efficiently from limited labeled data.
  - Incorporates feedback from domain experts to prioritize which data points are most valuable for training.
- **Permanent/File Storage**:
  - Stores raw and processed datasets for historical analysis, reproducibility and training.
- **Cloud Endpoints API**:

- Provides access to Neurosymboli AI predictions and active learning workflows.
- Acts as the interface between domain experts and the system.

The components of Neurosymbolic AI and Active Learning interact with each other as follows:

- **Data Preparation → Neurosymbolic AI:** Provides cleaned and structured data for training and inference.
- **Neurosymbolic AI → Cloud Endpoints API:** Exposes predictions and insights to domain experts and downstream services.
- **Cloud Endpoints API → Active Learning:** Sends model outcomes and feedback for selection of next training samples.
- **Active Learning → Data Preparation:** Feeds curated and labeled data back into the preparation process for iterative improvement.
- **Domain Expert ↔ Cloud Endpoints API:** Provides annotations, guidance, and feedback to improve model performance and explainability.

Overall, the Neurosymbolic AI and Active Learning component demonstrates how deep learning, symbolic reasoning and human expertise can be combined to produce explainable and adaptive AI models. By integrating human-in-the-loop feedback and iterative active learning, the system ensures that models remain interpretable, accurate and tailored to domain-specific contexts. In Industry 5.0 scenarios, this architecture enables XR operators and domain experts to interact with AI systems confidently, leveraging both machine intelligence and human knowledge for decision support and predictive insights.

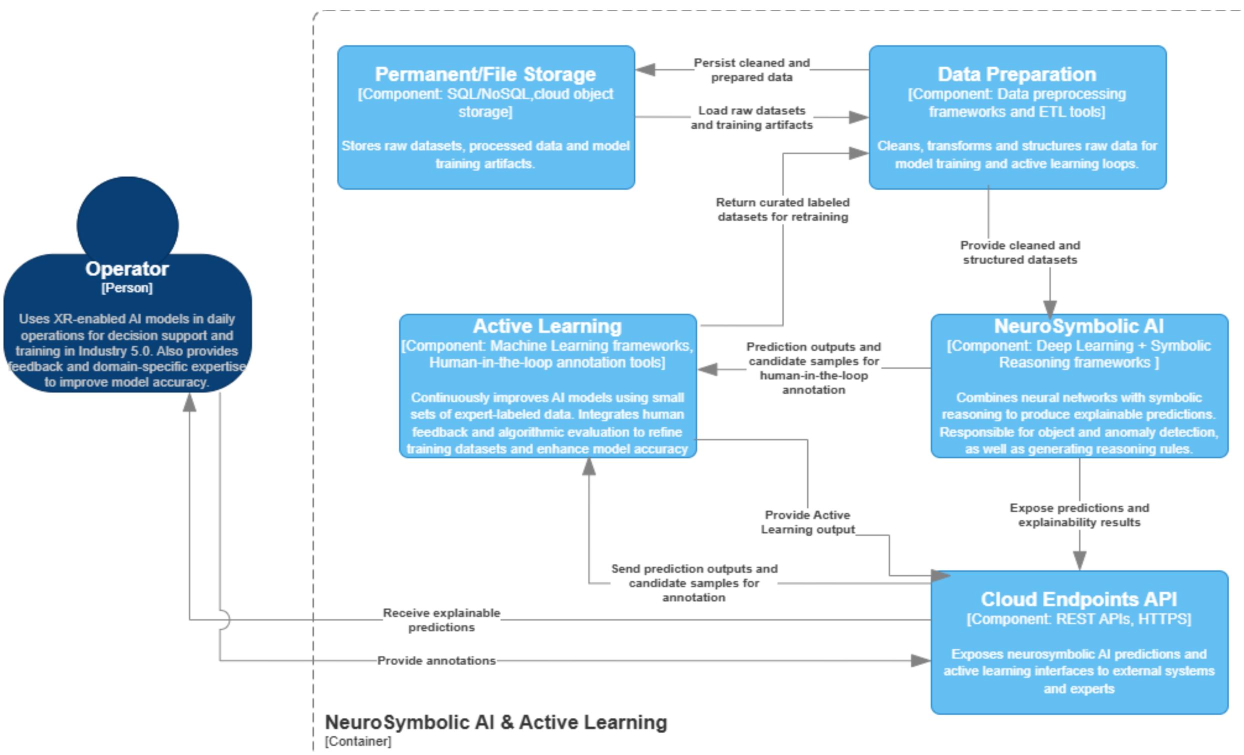


Figure 43 - Neurosymbolic AI and Active Learning Component Diagram

6.3.13 XR Training Asset Repository

The Web App of the **XR Training Asset Repository** functions as a mediator between the other components of the Training Platform (Hololight Hub, XR Training Management System) and the underlying storage provided by Owncloud. The XR Training Asset Repository is implemented as an ASP.NET Core 8.0 web application and serves as a central hub for managing and delivering XR training content. The repository exposes well-defined APIs for tenant and asset management, interacts with external storage, and provides a web-based admin interface for system administrators.

As shown in Figure 44, the XR Training Asset Repository consists of the following components:

- **API Controllers**:

  The API Controllers component is the entry point for all HTTP requests. It handles tenant provisioning, storage setup, management of training programs, learning paths, materials, and assets, and provides user management within tenants. The component also exposes **interactive OpenAPI/Swagger documentation** for developers and integrators. Controllers orchestrate operations by calling the Services component.

- **Services (Business Logic)**:

  The Services component implements the repository's core business rules and workflows. Responsibilities include dynamic tenant database management, tenant provisioning workflows, automated schema creation per tenant, material type handling, asset CRUD operations, type inference, training program and learning path management, and providing a **storage abstraction layer** for multiple backends such as AWS S3 and ownCloud/WebDAV. The Services component ensures that business logic is reusable, centralized, and decoupled from both API endpoints and database persistence.

- **Data Layer (Entity Framework Core)**:

  The Data Layer provides all database access and persistence. It manages dynamic tenant-specific DbContexts, resolves per-tenant connection strings, and supports the schema-per-tenant pattern. The Data Layer interacts with both the **Admin Database**, which stores global metadata (tenants, users, configuration, groups), and **Tenant Databases**, which store assets, materials, training programs, learning paths, and relationship tables.

- **Admin Dashboard (Web UI)**:

  The Admin Dashboard is a web-based interface that allows administrators to manage tenants, users, training materials, and system settings. It communicates with the API Controllers over HTTPS/JSON and leverages the repository's OpenAPI interface for operations.

- **External Systems**:
  - **AWS S3**: Provides scalable object storage for large training assets such as images, videos, PDFs, and 3D models. The Services component accesses S3 through a storage abstraction interface.
  - **XR5.0 Training Platform**: The central platform that orchestrates and delivers XR training applications by consuming the repository APIs.
  - **XR App**: XR training applications that retrieve assets and metadata from the repository.
  - **Repository Plugin**: A library for XR application developers to simplify interaction with the repository APIs.

- **Databases**:
  - **Admin Database**: Stores global system configuration, tenant metadata, global users, tenant administrators, and groups.
  - **Tenant Databases**: One per tenant, storing tenant-specific data including assets, materials, training programs, learning paths, and junction tables. The schema-per-tenant pattern ensures data isolation and simplifies multi-tenant management.

Moreover, the following interactions take place between the aforementioned components:

- ➢ API Controllers call Services to execute business workflows.
- ➢ Services interact with the Data Layer to persist or query data from the Admin and Tenant Databases.
- ➢ Services manage storage through either AWS S3 or ownCloud/WebDAV adapters.
- ➢ External clients (XR 5.0 Training Platform, XR App, Repository Plugin) consume the API Controllers over HTTPS/JSON.
- ➢ The Admin Dashboard uses the API Controllers to provide administrative functions to system administrators.

Consequently, this component-based architecture separates API endpoints, business logic, persistence, and external storage clearly, supporting scalability, maintainability, and extensibility. The multi-tenant design, combined with storage abstraction and external integrations, allows the repository to serve multiple XR training applications while keeping tenant data isolated and secure.

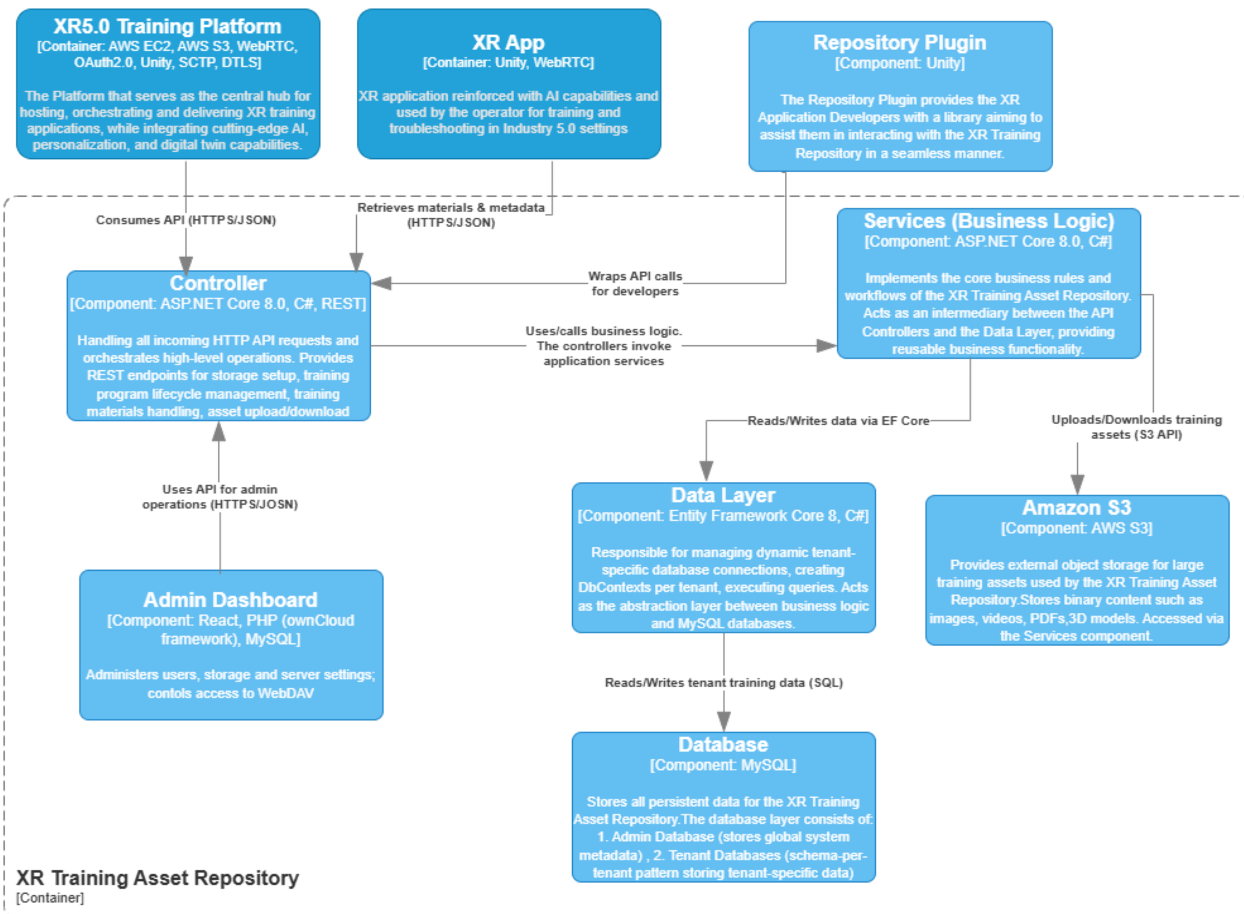


Figure 44 - XR Training Asset Repository Component Diagram

7. STANDARDS

7.1 Overview of Standards

Industry 4.0 standards

Industry 4.0 emphasizes the integration of cyber-physical systems, IoT, and data analytics to improve automation and manufacturing efficiency. Key frameworks and standards include:

7.1.1 RAMI 4.0 (Reference Architectural Model for Industry 4.0)

RAMI 4.0 [13] offers a structured framework for the implementation of Industry 4.0 principles. Developed by German organizations such as ZVEI, VDMA and BITKOM, RAMI 4.0 serves as a critical tool for navigating the complexities of industrial digital transformation. It provides a common language and methodology for implementing smart manufacturing practices and supports the adoption of emerging technologies such as the Industrial Internet of Things (IIoT) and Artificial Intelligence (AI) It organizes processes and technologies across a three-dimensional map comprising layers, life cycle/value stream, and hierarchy levels. This ensures a comprehensive approach to smart manufacturing and facilitates interoperability across systems. The three-dimensional relationship, which is guided by the RAMI 4.0 model, establishes connections between hierarchies, functions, and the product lifecycle.

As shown in Figure 45, RAMI 4.0 spans across 3 axes as follows [9]:

Vertical Axis: Functional Layers RAMI 4.0 consists of six functional layers that describe the abstraction levels of an Industry 4.0 system. These layers facilitate the digital integration of physical and business processes:

- **Asset Layer**: It represents the physical elements, such as equipment or products, that serve as the foundation for data sources in other layers and functions.
- **Integration Layer**: It is used to obtain and digitise information, transforming data from physical phenomena into formats that can be quantified and utilised digitally.
- **Communication Layer**: It is responsible for facilitating the exchange of information between different entities within the system. It defines the data standards and protocols that are required for the standardisation of Industry 4.0.
- **Information Layer**: It encompasses all forms of information, including real-time data (e.g. production data) and the regulations or directives that govern operational procedures.
- **Functional Layer**: It is responsible for defining the technical and logical functions necessary for the system's operations. It delineates the technical and logical functions that are indispensable for the system's operational efficacy.
- **Business Layer**: It represents the operational and functional aspects of the enterprise, encompassing the organisational structure, business models, financial management and legal considerations. It encompasses the company's organisational structure, including business models, financial management, and legal considerations.

Horizontal Axis: Hierarchy Levels Derived from the ISA-95 model, this axis defines levels of industrial operation, from physical devices to enterprise-level systems:

- **Field Device Level**: Physical devices like actuators, sensors or edge nodes.
- **Control Device Level**: PLCs and other controllers for real-time process management.
- **Station Level**: Workstations or machine groups involved in production tasks.
- **Work Center Level**: A collection of stations functioning as part of a production line.
- **Enterprise level**: Strategic and resource planning systems, typically ERP solutions.

Third Axis: Life Cycle and Value Streams Based on ISO 62890, this dimension represents the lifecycle stages of a product and the associated value streams. It ensures that all phases, from design to decommissioning, are accounted for:

- **Development**: It focuses on design, prototyping and engineering.
- **Production**: It covers manufacturing, assembly and quality assurance.
- **Service**: It includes operation, maintenance and eventual disposal.

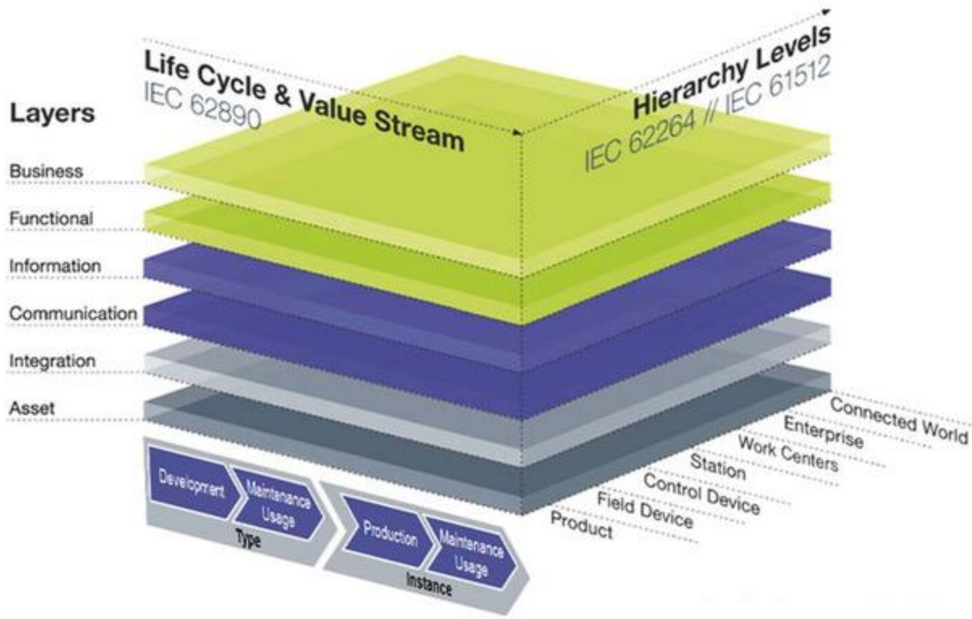


Figure 45 - RAMI4.0 as three-dimensional Layer model

A central feature of RAMI 4.0 is the Asset Administration Shell (AAS), which acts as a digital interface for managing both physical and digital assets [10]. Each asset, whether a machine, product or software module is accompanied by an AAS that describes its properties, capabilities and interfaces. Key functionalities of the AAS include:

- ➢ **Digital Twin Integration**: It enables real-time synchronization between physical assets and their digital counterparts.
- ➢ **Interoperability**: It uses standardized semantic models to ensure compatibility across systems.
- ➢ **Data Management**: It facilitates data collection, analysis and decision making by exposing relevant information.

The adoption of RAMI 4.0 offers numerous advantages for industries seeking to modernize their operations:

- ➢ **Standardization**: It aligns with global standards like IEC 62264, ISO 62890 and ISA-95, ensuring consistency and interoperability
- ➢ **Modularity**: It provides a scalable architecture that industries can implement incrementally, reducing risk and cost.
- ➢ **Interoperability**: It encourages seamless communication between heterogeneous systems and technologies, fostering collaboration

➢ **Productivity Gains**: It optimizes manufacturing processes through automation, predictive maintenance and real-time insights.

Despite its benefits, implementing RAMI 4.0 comes with challenges:

➢ **Complexity**: Mapping existing processes to the RAMI 4.0 framework requires significant expertise and resources.

➢ **Legacy Systems**: Integrating outdated equipment with Industry 4.0 solutions can be technically challenging.

➢ **Skills Gap**: Organizations must train personnel in both operational technology and information technology

➢ **The abstract nature** of RAMI 4.0 hinders its applicability for real world use cases and thus also the implementation of the design principles [11].

RAMI 4.0 framework is applicable in diverse industrial contexts such as Smart Factories for automating production lines with IoT-enabled devices, **Predictive Maintenance** for using AAS and digital twins to anticipate and address equipment failures and **Supply Chain Optimization** for enhancing transparency and collaboration through real-time data sharing.

7.1.2 IIRA (Industrial Internet Reference Architecture)

The Industrial Internet Reference Architecture (IIRA) serves as a blueprint for designing and implementing Industrial Internet of Things (IIoT) systems [7]. Developed by the Industrial Internet Consortium (IIC), it aims to bridge the gap between operational technology and information technology, fostering interoperability and scalability while addressing the unique challenges of industrial applications. The objective of the IIRA is to enhance interoperability across different IoT systems, thereby providing a framework for leveraging various technologies in the development of IIoT solutions. The IIRA identifies the essential components of any such system, including sensing, connectivity, analytics and control, which are integral to its implementation. The IIRA is described based on the ISO/IEC/IEEE 42010:2011 standard [8], which has been adopted by the IIC to define its Industrial Internet Architecture Framework (IIAF).

IIRA is structured into four architectural layers:

➢ **Business Layer**: This defines the overall objectives of the system, focusing on value creation and aligning IIoT deployments with business goals such as cost reduction, productivity, improvements and new revenue streams [7].

➢ **Usage Layer**: It captures how users interact with the system, defining operational scenarios and functional requirements. This layer ensures that system design considers all user roles [12].

➢ **Functional Layer:** At this level, the system is broken into domains such as sensing, analytics and control. It specifies the interactions between these domains, ensuring seamless functionality.

➢ **Implementation Layer**: This layer outlines the technologies, protocols and standards required to realize the functional components. Examples include communication protocols like MQTT and edge computing techniques such as fog computing.

The functional domains, crosscutting functions and system characteristics framework, derived from RAMI 4.0, offers a structured approach to understanding the various operational layers and their interactions within an industrial system. Functional domains represent key operational areas such as business processes, manufacturing execution and control systems. These domains are underpinned by crosscutting functions, including interoperability, security and scalability which enable seamless integration across systems.

Additionally, system characteristics such as modularity, reusability and real-time responsiveness are critical to achieving the dynamic and adaptable architectures envisioned by both RAMI and IIRA. The interplay between these layers and characteristics ensures that industrial systems can scale, integrate new technologies and maintain robust operations across diverse use cases.

By visualizing these relationships, Figure 46 underscores the holistic nature of modern industrial reference architectures, which aim to bridge the gap between operational technology and information technology.

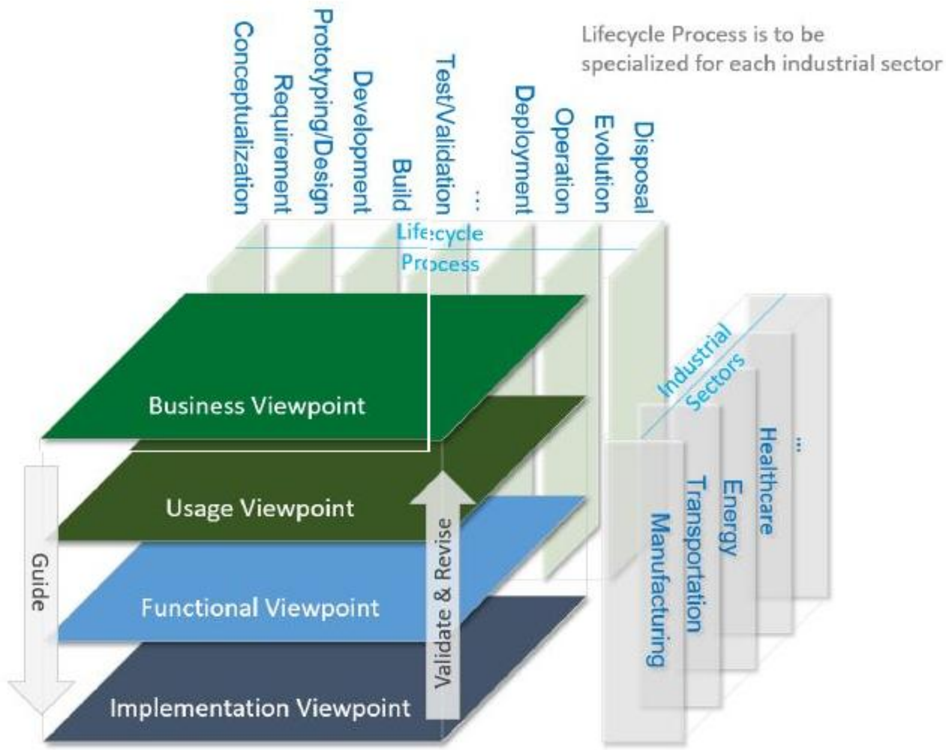


Figure 46 - Relationship among IIRA Viewpoints, Application Scope and System Lifecycle Process

The IIRA framework emphasizes several critical characteristics to ensure the effective design, deployment and operation of IIoT systems. These characteristics address both technical and operational needs, ensuring alignment with industrial requirements.

Interoperability is a cornerstone of IIRA, ensuring seamless communication and data exchange between heterogeneous devices, platforms and systems [13]. The architecture facilitates integration across legacy systems, new technologies and multi-vendor environments through standardized communication protocols and shared data models. This characteristic is essential in environments where devices from multiple manufacturers must coexist and collaborate efficiently.

Scalability is another fundamental aspect of IIRA, enabling systems to expand from pilot projects to full-scale enterprise implementations without significant reengineering [7][12]. The modular design of the architecture supports both vertical scaling (e.g. increasing computational resources) and horizontal scaling (e.g. adding more devices or nodes to the network). This flexibility is critical in dynamic industries like manufacturing, where the scale of operations can change rapidly.

Security is integral to IIRA, given the sensitivity and criticality of industrial data [12]. The architecture incorporates security measures across all layers, from physical devices to cloud services. This includes features such as encryption, authentication protocols, and intrusion detection systems. Security by design ensures the resilience of IIoT systems against cyber threats, protecting both data integrity and operational continuity.

Real-Time Processing and Low Latency are emphasized in the architecture to support applications requiring immediate responses, such as robotic control or autonomous vehicle systems. By leveraging edge computing technologies, IIRA reduces reliance on centralized cloud processing, enabling faster data analysis and decision-making closer to the source of data generation.

Manageability and Maintainability are also prioritized. The architecture supports remote monitoring and management of devices and systems, reducing downtime and maintenance costs. Features such as predictive maintenance, enabled by data analytics and machine learning, further enhance operational efficiency.

The flexibility and robustness of the IIRA framework make it applicable across a wide range of industrial sectors, addressing specific challenges and enabling innovative solutions.

In **smart manufacturing**, IIRA facilitates real-time monitoring of production lines, predictive maintenance of machinery, and optimization of supply chains [12]. By connecting equipment, sensors, and analytics platforms, the architecture supports adaptive manufacturing processes that can respond dynamically to changing demands and conditions.

The **energy and utilities** sector benefits significantly from IIRA, particularly in the management of smart grids and renewable energy systems. The architecture enables real-time monitoring and control of distributed energy resources, integration of renewable energy sources, and predictive analytics to forecast energy demand. These capabilities improve grid reliability, enhance energy efficiency, and reduce operational costs.

In **transportation and logistics**, IIRA underpins autonomous vehicle ecosystems, intelligent transportation systems, and smart logistics networks [13]. The architecture supports low-latency communication between vehicles and infrastructure, enabling features such as collision avoidance and traffic optimization. In logistics, it facilitates real-time tracking of goods, warehouse automation, and route optimization.

The **healthcare industry** leverages IIRA for applications such as remote patient monitoring, connected medical devices, and hospital automation. By integrating sensors, analytics, and secure communication protocols, the architecture supports timely and accurate medical interventions while protecting patient data.

In **agriculture**, the architecture enables precision farming practices. IoT-enabled sensors monitor soil conditions, weather, and crop health, while analytics platforms provide actionable insights for optimizing irrigation, fertilization, and pest control. These capabilities enhance productivity and sustainability.

7.1.3 ETSI Industry Specification

ETSI (European Telecommunications Standards Institute) plays a pivotal role in developing globally recognized standards for telecommunications and electronic systems, aligning with the growing demands of Industry 4.0 and IoT ecosystems. ETSI’s contributions address critical challenges, such as ensuring interoperability, securing communication systems, and enabling scalable industrial solutions through standardized frameworks. Below is a comprehensive exploration of ETSI's contributions to Industry 4.0, highlighting its key standards, applications, and the transformative role it plays in industrial settings.

ETSI’s Multi-access Edge Computing (MEC)

Multi-access Edge Computing (MEC) is one of ETSI's foundational contributions to Industry 4.0, enabling the decentralization of computing resources by bringing them closer to devices at the network edge [14]. MEC is instrumental in reducing latency and enhancing real-time analytics, crucial for applications like autonomous robotics, predictive maintenance, and augmented reality systems in manufacturing.

MEC operates through a structured framework that integrates edge devices with cloud platforms. This architecture supports data-intensive processes at the edge, minimizing the need for centralized processing and enabling quicker decision-making. The seamless connectivity between edge and cloud environments provided by MEC aligns with the dynamic requirements of modern industrial networks, where agility and rapid adaptation to operational changes are paramount.

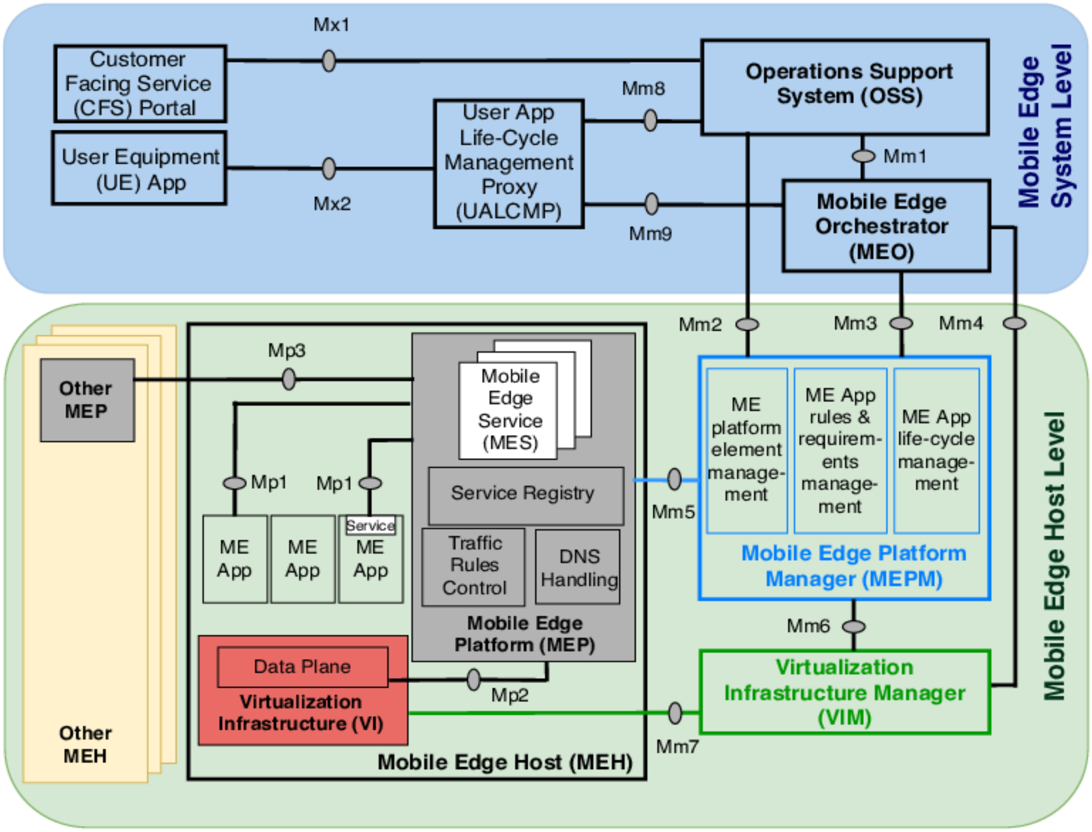


Figure 47 - MEC Reference Architecture

Figure 47 depicts MEC ETSI Architecture. There are three distinct components, MEC Host, MEC Platform Manager, MEC Orchestrator and the Virtualization Infrastructure Manager (VIM).

The **Virtualization Infrastructure Manager** has the purpose of managing the virtual machines (VMs) on top of physical infrastructure (compute, storage and networking). It is responsible for allocating, maintaining and releasing virtual resources of the virtualization infrastructure. MEC Apps are the actual applications that are run in MEC on top of VMs. **MEC Service** is an important block in MEC. The network-related APIs are exposed by MEC service to MEC Apps through reference point Mp1, as shown in Figure 47. Also, the MEC platform can consume these services. MEC Apps are network-aware (so that MEC Apps can act based on that) and MEC Service can help by exposing the network information through APIs. According to ETSI, at least three types of following services must be exposed by MEC Service: (i) the radio network conditions, (ii) the location information and (iii) the bandwidth manager.

Traffic Rules Control is an important piece of MEC Platform. As Mec Platform is serving multiple applications, simultaneously, it should be able to assign priorities through **Traffic Rules Control**.

The mobile edge platform shall provide functionality that supports routing all DNS traffic received from any UE to a local DNS server/proxy. This is where **DNS Handling** comes into play since the benefit of the MEC is to process a lot of information locally in MEC instead of sending it to the internet. Thus, it should have a way to handle DNS redirection to a local DNS server so that traffic can be diverted for local processing instead of sending to the internet. Moreover, it should be noted that there could be other MEC hosts connected to the existing MEC host through the Mp3 interface.

Concerning **MEC Platform Manager,** it performs the following:

- ➢ It manages MEC Apps life cycle; therefore, it instantiates, maintains and tears down MEC Apps on VMs.

- ➢ It is responsible for FCAPS (Fault, Configuration, Accounting, Performance, Security) management for the MEC platform.
- ➢ It manages the application rules, traffic rules and DNS configuration

The **Mobile Edge Platform Manager** assumes a central role in the architectural framework, and it is responsible, among other things, for the following:

- ➢ It oversees the deployment, configuration, scaling, updating and termination of MEC applications on the MEC host.
- ➢ It configures MEC platforms, including platform services like DNS, service discovery and application enablement.
- ➢ It manages resource allocations for applications hosted on MEC servers.
- ➢ It implements resource policies, including CPU, memory and network bandwidth usage, for applications.
- ➢ It monitors the performance of MEC platform components and hosted applications
- ➢ It acts as the intermediary between the MEC Orchestrator and the MEC platform.

Regarding the **CFS Portal** (Customer Facing Service Portal), the mobile operators' customers can order new MEC applications or monitor the service SLAs.

Finally, the **User App LCM Proxy** is an optional feature in the MEC system. This is applicable when the system supports a feature called "UserApps". When the mobile edge system supports the feature UserApps, the system shall allow the establishment of connectivity between a UE (that runs device application) and a specific instance of a mobile edge application. The user application lifecycle management proxy allows device applications to request on-boarding, instantiation and termination of user applications. In simple words, the user can now trigger specific applications in the MEC system from his device if this feature is supported by the MEC system.

Concerning **IoT Standards and Interoperability**, it must be noted that **ETSI** has significantly advanced the IoT domain through its **oneM2M** standards, which focus on creating a common service layer for diverse IoT applications [15]. The oneM2M architecture fosters interoperability by enabling seamless communication and integration between devices from different manufacturers and across multiple industrial sectors. By employing semantic data models, the framework ensures consistent interpretation of data across heterogeneous systems, a critical factor in large-scale IoT deployments.

This approach promotes horizontal scalability, allowing systems to expand across different industries without the need for extensive reengineering. Additionally, oneM2M emphasizes secure data exchange and device lifecycle management, ensuring robustness and reliability in industrial operations. Figure 48 depicts how the employment of a oneM2M Service Layer boosts interoperability and facilitates seamless interaction between heterogeneous applications and devices.

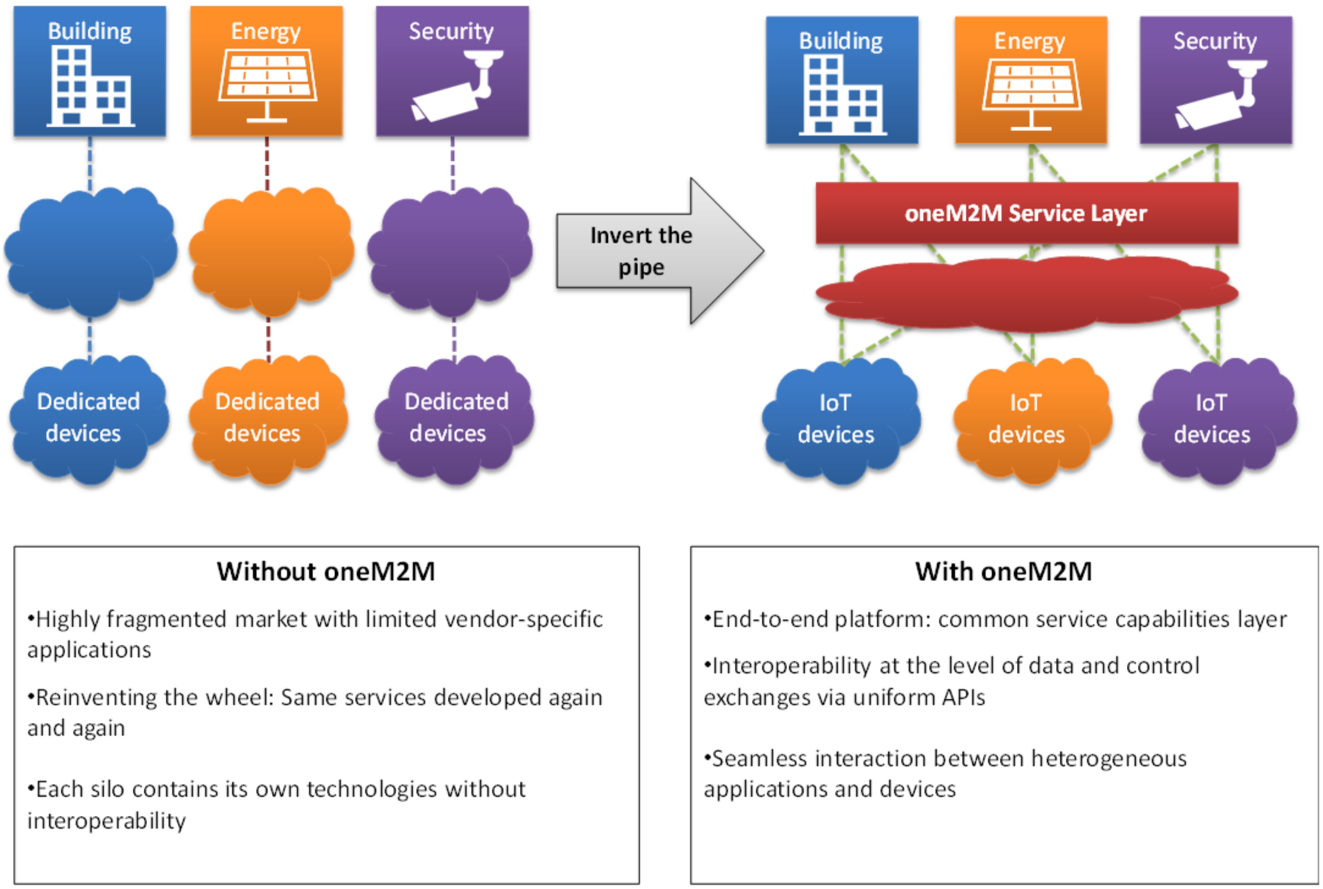


Figure 48 - Benefits of using oneM2M standard

To this end, the oneM2M initiative can remove fragmentation of the IoT world. Because it is independent of the access or protocol technology that is used for transport, it is designed to be a long-term solution for IoT deployment.

Concerning ETSI's **cybersecurity standards**, they address the growing need for secure industrial communication networks in the industry 4.0 landscape [16]. These standards encompass a range of critical areas, including risk assessment frameworks, encryption techniques, and device identity management. By securing communication channels and ensuring robust access control, ETSI standards mitigate vulnerabilities inherent in legacy systems and emerging IoT environments.

A notable aspect of ETSI's cybersecurity efforts is the focus on end-to-end security solutions. These solutions cover everything from device onboarding and authentication to secure data transmission and system-wide intrusion detection mechanisms. This comprehensive approach ensures that industrial systems remain resilient against evolving cyber threats while maintaining operational efficiency.

Moreover, ETSI's **5G standards** are foundational for enabling advanced capabilities in industrial networks, such as ultra-reliable low-latency communication (URLLC), massive IoT connectivity, and network slicing [17]. These features support diverse Industry 4.0 applications, ranging from remote robotic surgery to automated logistics systems.

The integration of 5G within industrial environments allows for unprecedented levels of automation and coordination among devices. By leveraging URLLC, manufacturing systems can achieve real-time synchronization between robots and machinery, enhancing precision and productivity. Similarly, network slicing enables the creation of dedicated virtual networks tailored to specific industrial use cases, ensuring optimal resource allocation and performance.

Including a visual representation of ETSI's 5G network slicing or URLLC architecture can help readers understand how these technologies are structured and deployed in industrial contexts.

7.1.4 ISO/IEC Standards

ISO 22400 (Expected to be replaced by ISO/DIS 22400-2)

ISO 22400 provides a standardized framework for defining and monitoring key performance indicators (KPIs) specifically for manufacturing operations, making it a cornerstone for Industry 4.0 implementations. The standard supports the integration of advanced technologies such as IoT, AI and Extended Reality (XR) by enabling consistent and effective tracking of performance metrics that are critical to operational efficiency and decision-making. In the context of Industry 4.0, ISO 22400 helps organisations align their processes with data-driven insights, facilitating real-time analysis and optimisation of production workflows. By ensuring interoperability and consistency in KPI definitions across systems and platforms, this standard promotes smarter, more connected manufacturing environments that drive innovation and competitiveness in the era of digital transformation.

ISO 26262

ISO 26262 is an internationally recognised standard that provides a comprehensive framework for ensuring the functional safety of automotive systems. It addresses the unique challenges posed by increasing complexity and autonomy of modern vehicles and provides guidelines for the design, development and validation of systems critical to safety. In Industry 4.0, ISO 26262 is highly relevant to autonomous functions such as robotics, automated manufacturing and AI-driven processes. The standard emphasises risk assessment, hazard analysis and the implementation of safety mechanisms to mitigate potential system failures. By following ISO 26262, organisations can ensure the reliability and safety of autonomous operations, fostering trust and compliance in environments where advanced technologies are seamlessly integrated with human-centric workflows and Industry 4.0 principles.

7.1.5 Industry 5.0 standards

Industry 5.0 builds on Industry 4.0 by emphasizing human-centricity, sustainability and resilience. While standards are still emerging, existing frameworks and tools are being adapted:

1. Human-centric AI standards:

   Human-centric AI standards, such as those developed by ***ISO/IEC JTC 1/SC 42*** [18], are increasingly aligned with the objectives of Industry 5.0. These standards emphasise the importance of explainable and trustworthy AI, ensuring that artificial intelligence technologies are used ethically to enhance and support human workers in a transparent and reliable manner.

   The ***IEEE 7000-2021 standard*** [19] provides further guidance by outlining ethical concerns in system design, with a particular focus on integrating ethical AI concepts throughout the development lifecycle. This approach supports the development of human-centric and ethical AI systems, thereby reinforcing the principles of industry 5.0.

2. **XRSI (Extended Reality Safety Initiative)**:

   The **Extended Reality Safety Initiative (XRSI)** is a globally recognized non-profit organization committed to the advancement of safety, privacy, and ethical considerations in the evolving domains of extended reality (XR), including virtual reality (VR), augmented reality (AR), and mixed reality (MR). Headquartered in the San Francisco Bay Area, XRSI serves as a multidisciplinary platform for developing frameworks, fostering collaboration, and advocating for responsible innovation in immersive technologies [20].

XRSI's mission revolves around addressing the privacy, security, and ethical challenges posed by immersive technologies. By uniting experts from diverse fields such as cybersecurity, law, healthcare, AI, and product design, XRSI aims to create global standards and best practices. Its initiatives focus on safeguarding vulnerable populations and ensuring inclusivity in XR applications, thereby reducing potential digital divides.

One of XRSI's flagship projects is the **XRSI Privacy and Safety Framework**, a regulation-agnostic guide designed to address privacy and safety concerns in XR environments [25]. This framework emphasizes the protection of sensitive data, including biometrically inferred information, and aligns with global standards such as the GDPR and NIST guidelines. The framework introduces novel considerations like informed consent, data minimization, and anonymization to mitigate risks associated with immersive experiences. It also provides actionable insights into content moderation and proactive safety measures, making it adaptable to evolving technological landscapes.

XRSI plays a pivotal role in exploring the potential of XR in transformative sectors such as healthcare and education. For instance, the **Medical XR Privacy and Safety Framework** targets the ethical use of XR in healthcare, ensuring that patient data is handled securely and equitably. XR's capability to create immersive training environments has also been highlighted in education, where it facilitates interactive learning and high-fidelity simulations for medical procedures, promoting accessibility and skill development across various demographics.

Recognizing the impact of data biases in AI-driven XR systems, XRSI advocates for the inclusion of underrepresented groups in the development and governance of XR technologies [26]. This focus aims to prevent the perpetuation of gender, racial, or socioeconomic disparities in virtual spaces. XR platforms also benefit from accessibility features, such as voice guidance for visually impaired users, which align with XRSI's broader goals of fostering equity and diversity [27].

XRSI has been instrumental in advising policymakers and industry stakeholders on the responsible adoption of XR technologies. Collaborations with academic institutions, such as the University of California San Diego, and participation in international forums highlight its dedication to shaping a safe and inclusive metaverse. By emphasizing cross-sector engagement, XRSI seeks to build a sustainable ecosystem that prioritizes user trust and safety [27].

With the rapid advancement of XR and related fields like AI, 5G, and brain-computer interfaces, XRSI's efforts are increasingly vital. As immersive technologies become mainstream, the organization continues to refine its frameworks, expand its partnerships, and advocate for global standards that balance innovation with ethical accountability.

3. **OpenXR**:

   OpenXR is an open-source and royalty-free standard for access to virtual reality (VR) and augmented reality (AR) platforms and devices. It simplifies the development and deployment of XR applications across various hardware platforms. By ensuring interoperability, OpenXR supports the seamless integration of XR technologies, which is vital for the creation of integrated and efficient ecosystems in the Industry 5.0 era. These standards provide a framework that facilitates the ethical, safe, and effective adoption of advanced technologies in a human-centric industrial landscape. It encompasses technologies like Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR). As immersive environments become integral to industries such as healthcare, education, manufacturing, and entertainment, XRSI provides critical frameworks to address unique challenges related to data security, user safety, and ethical deployment.

   The cornerstone of XRSI's contributions is its **Privacy and Safety Framework**, a comprehensive guideline that ensures the responsible development and use of XR technologies [20]. The framework aligns with global standards like the **General Data Protection Regulation (GDPR)**, the **NIST Privacy Framework**, and ISO standards, offering a robust reference for organizations across sectors. Key areas covered by the framework include:

- **Privacy and Data Governance:** The framework emphasizes privacy by design, urging developers to incorporate secure data-handling practices from the outset. This includes anonymization techniques, secure storage, and encryption to protect user identities and prevent misuse of behavioral data. Furthermore, XRSI promotes informed consent, ensuring users understand what data is collected and how it is used. XR systems collect vast amounts of personal and sensitive data, such as:
    - Biometric data (e.g., eye tracking, facial expressions, voice patterns)
    - Behavioral data (e.g., movement patterns, gaze duration)
    - Location-based data
- **Safety in Immersive Environments: The** physical and psychological risks of XR are unique and varied [21]. XRSI provides guidelines for minimizing these risks, including ergonomic hardware design, well-calibrated user interfaces, and clear policies against harassment in shared virtual environments. The risks fall into these categories:
    - **Physical Safety Risks**: Issues like motion sickness, spatial disorientation and repetitive strain injuries.
    - **Psychological Risks**: Cognitive overload, virtual harassment and addiction.
- **Ethical Considerations**: Ethical concerns are central to XR, as immersive technologies can influence user behavior and perceptions. XRSI advocates transparency in algorithms, the prevention of misinformation, and inclusivity in XR content design to avoid bias or exclusion.

  XRSI Guidelines apply to many sections such as the following:

    - **Education**: In virtual classrooms, XRSI ensures that educational XR applications safeguard students' privacy and provide harassment-free environments. For example, the framework recommends encrypted communication for online lectures and secure user authentication for accessing XR educational content.
    - **Healthcare**: In healthcare, XR technologies are transforming diagnosis, therapy, and training. XRSI frameworks guide organizations to secure sensitive medical data and ensure ethical use, such as in VR-based therapy for PTSD or AR-assisted surgeries. Figure 49 depicts an example of XR usage in biomedical applications and mainstream working principles [22].
    - **Entertainment**: With the growing popularity of VR gaming and social XR platforms, XRSI focuses on mitigating risks related to addictive content, virtual harassment, and identity theft. The framework includes recommendations for parental controls and content moderation.
    - **Industrial Applications:** XR is increasingly used in training simulations, remote collaboration, and product design in industrial contexts. XRSI frameworks guide the ethical deployment of these technologies, ensuring workplace safety and data protection.

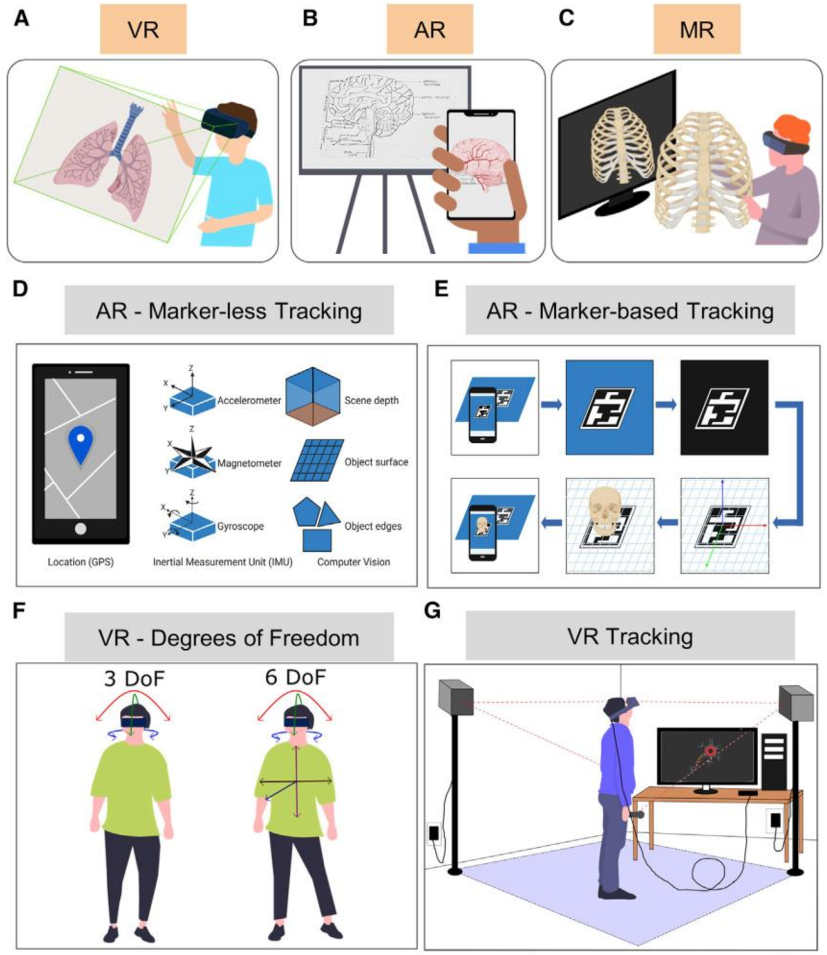


Figure 49 - (A) Visualizing 3D image of a lung, (B) Smartphone based AR, (C) Visualization of a 3D image of the rib cage using MR glasses, (D) Marker-less tracking in AR, (E) Marker-based tracking in AR, (F) Degrees of freedom in VR, (G) Tracking VR principles

Even if the rise of XR technologies introduces many opportunities, however it also comes with complex challenges that XRSI seeks to address, such as:

- **Interoperability**: XR systems often lack standardization, leading to fragmentation across platforms. XRSI promotes open standards to foster interoperability, enabling seamless communication between different XR applications and devices.
- **Cybersecurity Threats**: Immersive environments are vulnerable to hacking and data breaches. XRSI recommends multi-factor authentication, end-to-end encryption and robust cybersecurity protocols to mitigate these risks.
- **Behavioral Profiling and Privacy Invasion**: XR's ability to track detailed behavioral data raises concerns about profiling and surveillance. XRSI advocates clear data minimization policies and strict regulatory compliance.
- **Digital Harassment and Safety**: The immersive nature of XR environments increases the risks of harassment and abuse. XRSI proposes real-time content moderation, robust reporting mechanisms, and safe interaction zones to protect users.

7.1.6 STAR-RA

The ***Reference Architecture for AI-Based Industry 5.0 Systems*** (STAR-RA) is a high-level framework designed to provide guidance for the development of human-centric, resilient, and sustainable AI-driven industrial systems that are aligned with the principles of Industry 5.0. It builds upon the concepts of Industry 4.0 reference architectures, such as ***IIRA and RAMI4.0***, but provides functionalities and structuring principles that are adapted to the requirements of Industry 5.0. The key features include:

1. **Human-Centricity**: A central tenet of Industry 5.0 is the collaboration between humans and machines. The STAR-RA integrates the concept of **human-in-loop (HITL)**, allowing human operators to interact with AI systems (such as robots and digital twins) in real-time. This interaction aims to leverage the unique capabilities of both human intuition and machine intelligence. The architecture allows for **adaptive systems** that support human decision-making, increase productivity and improve safety in industrial environments [23]. The human-centric approach ensures that AI does not replace human workers but rather enhances their decision-making and operational capabilities [24].
2. **Sustainability:** In alignment with European initiatives such as the European Green Deal and the Circular Economy Action Plan, the STAR-RA framework prioritizes sustainability. By utilizing **AI for resource optimization,** the architecture supports efficient manufacturing processes that reduce waste, lower energy consumption, and promote environmentally friendly practices [23]. Moreover, AI can be used to develop **digital twins** (i.e. virtual representations of physical assets) which can simulate different scenarios to improve lifecycle management and predict the impact of various production strategies on sustainability [24].
3. **Trustworthiness:** The **trustworthiness** of AI systems is a core principle of STAR-RA. To build reliable, transparent, and accountable AI systems, the architecture incorporates **Explainable AI (XAI)**, ensuring that the decisions made by AI models are understandable and justifiable to human users. This feature is crucial in industrial environments, where regulatory compliance, data privacy, and operational safety are of utmost importance [23]. The architecture also includes **cybersecurity measures** to protect both data and the integrity of AI-driven systems. These systems are designed to comply with European standards such as the **General Data Protection Regulation (GDPR)** and the **EU Cybersecurity Act**, ensuring secure and ethical use of AI [23].
4. **Integration of Advanced Technologies:** STAR-RA enables the integration of various advanced AI technologies, such as **NeuroSymbolic AI**, **active learning**, and **multi-agent systems**. **NeuroSymbolic AI** combines the strengths of neural networks and symbolic reasoning to enhance the flexibility and interpretability of AI systems. This integration allows AI systems to process both structured and unstructured data, making them more adaptable to complex industrial environments [24]. Additionally, **active learning** enables AI models to continuously learn and improve over time by selecting the most relevant data for training. **Multi-agent systems** facilitate coordination between different AI agents, optimizing the entire production process across multiple systems, which is particularly valuable in large, complex manufacturing environments [23].
5. **Resilience:** The STAR-RA architecture also emphasizes the **resilience** of industrial systems by incorporating mechanisms for predictive maintenance and real-time **decision-making.** By analyzing data from IoT sensors and other sources, the system can predict equipment failures before they occur, ensuring higher operational uptime and minimizing costly disruptions. This feature is vital for industries that rely on continuous, high-efficiency operations [24]. Furthermore, STAR-RA's design allows it to be easily adapted to different industrial sectors, making a highly versatile and resilient framework for modern manufacturing [23].

To this end, the **STAR-RA** reference architecture provides a robust framework for the deployment of AI-based solutions in Industry 5.0 environments, focusing on human-centric, sustainable and resilient systems. By integrating advanced AI technologies with a focus on trust, transparency and human-machine collaboration, the architecture enables a shift from traditional industrial automation to a more collaborative, ethical and eco-friendly approach. This model not only supports operational efficiency but also ensures that AI applications are safe, compliant with regulations and aligned with sustainability goals.

7.2 Relevance of XR5.0 Reference Architecture with existing Standards

The XR5.0 architecture represents a cutting-edge approach to enabling Industry 5.0 principles, including human-centric operations, real-time decision-making and the integration of advanced XR and AI technologies. To ensure alignment with established practices and foster interoperability, XR5.0 draws upon a variety of existing standards and frameworks, including RAMI 4.0, IIRA, ISO/IEC specifications, ETSI standards, XRSI, OpenXR and the STAR Reference Architecture.

These standards provide foundational guidance for system structure, interoperability and functional layering, while XR5.0 extends their scope to address emerging requirements like immersive visualization and predictive analytics. This alignment ensures that XR5.0 is not only innovative but also adheres to established industry norms, facilitating its adoption and integration into diverse industrial ecosystems.

In the following subsections, we explore how XR5.0 aligns with each standard, emphasizing shared principles, functional similarities and key innovations that bridge Industry 4.0 and Industry 5.0 paradigms.

7.2.1 XR5.0 Reference Architecture and RAMI 4.0

The Reference Architecture Model for Industry 4.0 (RAMI 4.0) provides a structured, three-dimensional framework for designing, implementing and integrating industrial systems. It organizes the enterprise landscape across three axes: **Hierarchy Levels**, **Life Cycle & Value Stream**, and **Layers (Business, Functional, Information, Communication, Integration, and Asset layers)**. RAMI4.0 has become a widely adopted standard in industrial automation, providing clear guidance on **system modularity**, **interoperability**, and **lifecycle management**.

The XR5.0 Reference Architecture aligns closely with RAMI4.0 principles, while extending its scope to accommodate **human-centric AI and immersive Extended Reality (XR) technologies**, in line with Industry 5.0 objectives. Specifically, XR5.0 integrates AI-driven digital twins, personalized XR training, and secure operator assistance services, extending RAMI 4.0's focus on machine-to-machine and system-level interoperability to include **human-in-the-loop interactions and adaptive learning workflows**.

Layer and Component Mapping between XR5.0 and RAMI4.0

RAMI4.0 defines multiple layers, and XR5.0 components map naturally onto these:

Table 8 – Mapping of RAMI4.0 Layers to XR5.0 Layers/Components

RAMI4.0 Layer	XR5.0 Layer / Components	Notes on Mapping
Business Layer	XR5.0 Business Layer, Value Stream Workflows, Personalized Training Service	Aligns XR5.0 business services with RAMI4.0's value stream perspective. The XR5.0 value stream diagram maps six stages of value delivery, ensuring that XR-enabled training, augmented information, and operator assistance are integrated into the industrial value creation process.
Functional Layer	XR Applications (XR Training Plugin, XR apps), AI models (Gen AI, Active Learning, Neuro-symbolic AI, XAI)	Provides the functional services that realize business processes. XR5.0 extends RAMI4.0 functional concepts to include AI-driven analytics, immersive XR content, and real-time adaptive guidance.
Information Layer	Human Digital Twin Platform (Clawdite), Cloud Repository, ETL Pipeline	Ensures standardized representation of operational and biometric data. RAMI4.0 emphasizes structured information flow. XR5.0 incorporates **operator digital twins, sensor data integration, and adaptive training metrics**.

Communication Layer	OPC-UA Connector, MQTT Interfaces, XR Hub Network Components	Manages interoperability and communication between devices, sensors and applications. XR5.0 adheres to industrial protocols and RAMI4.0 communication principles while supporting low-latency XR streaming.
Integration Layer	Cloud Repository Plugin, XR Hub, Remote Assistance Platform	Bridges applications with physical and virtual resources. XR5.0 extends RAMI4.0 integration by handling XR content orchestration and secure real-time human-machine interaction.
Asset Layer	Pilot Environment (IoT Sensors, Cameras, Smartwatch), Operator Devices, AMRs	Physical infrastructure elements aligned with RAMI4.0 assets. XR5.0 emphasizes human-in-the-loop operations, incorporating devices that monitor operator performance, biometrics and interactions with machines.

Implementation Examples of RAMI 4.0 Alignment
The XR5.0 Reference Architecture operationalizes RAMI4.0 principles in several key areas:

- **Value Stream Mapping**:
  - XR5.0's six-stage value stream diagram directly implements RAMI4.0's life cycle and value creation perspective.
  - Each stage (from provision of augmented information to personalized training) corresponds to specific capabilities and business services, reflecting **value-driven** design.
- **AI and XR Functional Services**:
  - RAMI4.0's functional layer emphasizes modular service components. XR5.0 realizes this through AI models and XR applications that support operator decision-making, predictive maintenance, and immersive training.
  - For example, the Active Learning Model adapts training content in real-time based on operator performance, extending RAMI4.0's functional concept to **human-centered adaptability**.
- **Information and Digital Twin Integration**:
  - The Human Digital Twin platform (Clawdite) integrates multiple data sources (IoT sensors, ERP, operator biometrics) reflecting RAMI4.0's **information layer**.
  - RAMI4.0 does not explicitly address human digital twins, so XR5.0 extends the standard by capturing and integrating human state and performance metrics into operational workflows.
- **Communication and Orchestration**:
  - OPC-UA connectors and MQTT pipelines implement RAMI4.0's communication principles, ensuring secure and standardized data exchange between applications, devices and cloud components.
  - XR5.0 also orchestrates XR content delivery and remote expert communication, demonstrating integration of real-time immersive feedback within industrial standards.
- **Asset Layer and Human-in-the-Loop Operations**:

- Operator devices (XR glasses, smartwatches) and AMRs are treated as assets in RAMI4.0 terms, but XR5.0 expands the concept to include human physiological and behavioral data as first-class "operational assets", supporting adaptive training and safety workflows.

Added Value of XR5.0 Beyond RAMI4.0

While RAMI4.0 provides a **solid industrial architecture foundation,** XR.5.0 extends it in several key ways:

➢ **Human-Centric Industry 5.0 Alignment**:

RAMI 4.0 focused on system and process alignment, while XR5.0 explicitly includes human operators as central participants, integrating digital twins, biometric data and adaptive learning mechanisms.

➢ **Immersive XR Integration**:

XR5.0 implements RAMI4.0's functional and information layers using XR applications and AI-driven immersive environments, enabling real-time, context-aware visualization and guidance.

➢ **AI-Enhanced Functional Services**:

RAMI4.0 describes functional capabilities at a high-level. On the other hand, XR5.0 realizes these with object recognition, step-by-step guidance and personalized training, all powered by AI models integrated into the industrial workflow.

➢ **Enhanced Interoperability and Orchestration:**

XR5.0 extends RAMI4.0 by managing heterogeneous devices, cloud services, XR content and remote assistance under a unified orchestration and security framework.

➢ **Operator Wellbeing and Safety**:

By including biometric monitoring and adaptive XR experiences, XR5.0 ensures operator health, safety and skill development, going beyond RAMI4.0's machine-centric focus.

The XR5.0 Reference Architecture is compatible with RAMI4.0, while significantly extending it to address human-centric, AI-enabled, and XR-immersive aspects of Industry 5.0 This alignment ensures that XR5.0 maintains industrial standard compliance, while delivering novel capabilities in operator training, human-machine collaboration, and adaptive industrial operations.

7.2.2 XR5.0 Reference Architecture and Industrial Internet Reference Architecture (IIRA)

The Industrial Internet Reference Architecture (IIRA) provides a standards-based framework for designing, integrating and analyzing Industrial Internet of Things (IIoT) systems. IIRA defines multiple viewpoints (Business, Usage, Functional, Implementation) and tiers (Edge, Platform, Enterprise) to support interoperability, modularity, and scalability across industrial systems. It also defines cross-cutting functions such as security, connectivity and data management.

The XR5.0 Reference Architecture aligns closely with IIRA principles while extending them to include human-centric, AI-enabled, and XR-immersive operations, consistent with Industry 5.0 objectives. XR5.0 incorporates human digital twins, adaptive XR-based training, AI-driven decision support and real-time operator assistance, ensuring that IIRA's guidelines are realized in a human-centric industrial ecosystem.

Layer and Component Mapping between XR5.0 and IIRA

As shown in the following table, the XR5.0 architecture components can be mapped to IIRA viewpoints and tiers:

Table 9 – Mapping of IIRA Viewpoints/Tiers to XR5.0 Layers/Components

IIRA Viewpoint / Tier	XR5.0 Layer/Component	Notes on Alignment
Business Viewpoint	Value Stream, Personalized Training Service, Operator Assistance Services	Maps XR5.0 business objectives and workflows to IIRA business-level viewpoint
Usage Viewpoint	Training Workflow Blueprint, XR Training Plugin, AR Assistant	Defines operator usage scenarios and interactions, reflecting IIRA usage viewpoint principles.
Functional Viewpoint	XR applications, AI models (GenAI, Active Learning, XAI), Remote Assistance Platform	Implements IIRA functional requirements for analytics, decision-making, and operational support.
Implementation Viewpoint	XR Hub, Cloud Repository, OPC-UA Connector, IAM	Maps software and hardware integration to IIRA implementation viewpoint.
Edge Tier	XR devices, IoT sensors, smartwatches, AMRs	Handles real-time data acquisition and local processing.
Platform Tier	Human Digital Twin platform (Clawdite), AI engines, ETL pipelines	Centralized processing, analytics, and orchestration aligned with IIRA platform tier.
Enterprise Tier	ERP integration, dashboards, IAM governance	Business-level decision-making, reporting, and compliance.

Implementation Examples of IIRA Alignment

The XR5.0 Reference Architecture operationalizes IIRA principles in several areas:

- **Business Viewpoint**:
  - XR5.0's value stream and personalized training services implement IIRA's business viewpoint, aligning industrial objectives with operator training and performance improvement.
- **Usage Viewpoint**:
  - XR5.0 defines operator interactions with XR applications, chatbots, and AI services, reflecting usage scenarios required by IIRA.
  - **Example**: An XR app guides operators through maintenance tasks in real time, aligning with IIRA's usage focus.
  -
- **Functional Viewpoint**:
  - XR5.0's AI models and XR applications implement modular functional services, consistent with IIRA functional viewpoint guidance.
  - Examples include predictive maintenance, real-time guidance, and adaptive training workflows.
- **Implementation Viewpoint**:
  - XR5.0 orchestrates software and hardware components via the XR Hub, OPC-UA connectors, and IAM system.
  - These components implement IIRA's integration principles for heterogeneous systems, supporting secure, standardized, and scalable deployment.

- **Tiers (Edge, Platform, Enterprise)**:
  - Edge Tier: Captures sensor data, operator biometrics, and XR device input
  - Platform Tier: Processes data through Human Digital Twin, AI engines, and cloud services.
  - **Enterprise Tier**: Integrates ERP systems, dashboards, and governance, ensuring business-level decision support.

Added Value of XR5.0 Beyond IIRA
While IIRA provides a comprehensive framework for IIoT systems, XR5.0 extends it in several key ways:

- **Human-Centric Operations**:

  XR5.0 places the human operator at the center of the industrial ecosystem, integrating biometric monitoring, adaptive XR training, and real-time decision support, which goes beyond IIRA's machine and system-centric perspective.

- **Immersive XR Integration**:

  XR5.0 implements functional and usage viewpoints using AI-powered immersive XR applications, enabling context-aware guidance and training that IIRA does not explicitly address.

- **AI-Enhanced Functional Services**:

  Functional components like Active Learning models, GenAI, and XAI provide real-time adaptive support to operators, extending IIRA's functional viewpoint with predictive and personalized capabilities.

- **Enhanced Interoperability and Orchestration**:

  XR5.0 manages heterogeneous devices, cloud services, and XR applications under a unified orchestration and security framework, ensuring seamless human-machine collaboration.

- **Operator Wellbeing and Safety**:

  By integrating biometric feedback and personalized training, XR5.0 supports operator health, skill development, and safety, which are beyond the explicit scope of IIRA.

The XR5.0 Reference Architecture is compatible with IIRA principles while significantly extending them to support human-centric, AI-enabled, and immersive XR operations. This ensures alignment with Industry 5.0 objectives and establishes a traceable, standards-compliant architecture that enhances operator performance, safety and industrial efficiency.

7.2.3 XR5.0 Reference Architecture and ISO/IEC JTC 1 Standards

ISO/IEC JTC 1 covers a broad set of international standards across **AI (SC 42), IoT (SC 41), cloud computing (SC 38), cybersecurity (SC 27), and data management/interoperability (SC 32)**. These standards provide a foundation of high-level principles for secure, interoperable and scalable digital systems.

The XR5.0 Reference Architecture has been designed in a way that **acknowledges and conceptually aligns with the general direction and principles** of these standards. This alignment serves primarily as a **guiding framework** for architectural thinking rather than a claim of full technical conformity or certification.

Conceptual Layer and Component Alignment between XR5.0 and ISO/IEC JTC Standards
Several high-level correspondences can be identified with ISO/IEC JTC 1 standardization domains:

- **SC 42 - Artificial Intelligence**

The AI-enabled components of XR5.0 (e.g., Generative AI, Active Learning, the XAI layer, Neuro-symbolic reasoning) follow general principles related to transparency, explainability and responsible use of AI, as described in SC 42.

- **SC 41 - IoT and Digital Twins**

  Elements such as the Human Digital Twin (Clawdite) and the integration of sensor or contextual data reflect concepts outlined in SC 41, such as digital representations, data collection models and high-level IoT architectures.

- **SC 38 - Cloud Computing and Distributed Platforms**

  XR5.0 includes cloud-based repositories, AI services and the XR Hub, which follow the general architectural principles of cloud service provisioning, scalability and API-based access, consistent with SC 38 guidance.

- **SC 27 - Information Security and Privacy**

  The OAuth2-based Identity and Access Management layer incorporates foundational principles of secure identity management, access control and data protection, which correspond broadly to SC 27 recommendations.

- **SC 32 - Data Interoperability and Metadata Management**

  The management of XR Training Objects, metadata and content representation follows generic data structuring and interoperability practices, in line with SC 32.

Illustrative Examples of Conceptual Alignment
The following examples show how the architecture follows general principles from the standards, without requiring detailed implementation mapping:

- **AI Components (SC 42)**

  The architecture supports explainability and transparency through the inclusion of the XAI layer and structured AI workflows.

- **Digital Twin and IoT Concepts (SC 41)**

  The Human Digital Twin conceptually reflects digital representation principles and modular data inputs from different sources.

- **Cloud and Security (SC 38 + SC 27)**

  The IAM system, token management and secure access control, follow high-level security and cloud governance principles.

- **Data Management (SC 32)**

  The handling of XR metadata and training assets respects generic data-structuring and interoperability approaches.

Added Conceptual Value
The XR5.0 Reference Architecture extends the high-level principles of ISO/IEC JTC 1 standards to a human-centric XR + AI + Digital Twin environment. The architecture:

- Integrates multiple technology domains in a unified XR-focused ecosystem
- Follows security-by-design and modular design principles
- Supports future scalability and potential alignment with additional standards or certifications if required.

This ensures that XR5.0 remains forward-looking and adaptable.

To this end, the XR5.0 Reference Architecture demonstrates a high-level alignment with ISO/IEC JTC 1 standardization principles across AI, IoT, cloud, security and data interoperability. This alignment supports trustworthy, secure and human-centric system design.

7.2.4 XR5.0 Reference Architecture and ETSI Industry Specifications

ETSI provides a set of specifications that are widely used across industrial, telecommunication, edge computing, and immersive systems. While the XR5.0 architecture is not required to fully implement any specific ETSI standard, it aligns conceptually with several ETSI frameworks that support interoperability, edge/cloud integration, low-latency processing, and trustworthy data handling.

This subsection provides a high-level (non-binding) mapping between XR5.0 architectural elements and the relevant ETSI specifications.

Conceptual Alignment between XR5.0 and ETSI MEC Architecture

ETSI MEC (Multi-access Edge Computing) defines an architecture where computation can be offloaded to edge nodes close to the end user, enabling low-latency, context-aware services. In XR5.0, several architectural behaviors follow similar principles, even if no strict implementation is required. The following conceptual points of alignment have been identified:

- XR5.0 uses a distributed deployment model, where XR applications, AI services, and training tools may operate across cloud and on-premises infrastructures, a pattern consistent with MEC's separation of user equipment, edge nodes, and cloud platforms.
- The **real-time adaptation of XR training** benefits from latency-sensitive data flows (e.g., physiological measurements, movement prediction), aligned with MEC objectives for low-latency processing.
- The **use of connectors** (e.g., OPC-UA Connector) reflects MEC's focus on standardized interfaces enabling access to local data sources such as sensors, IoT devices, and industrial systems.
- XR5.0 incorporates **context-aware personalization** (e.g., emotional state estimation or movement prediction), which is conceptually aligned with MEC's application enablement APIs for providing location, context, and QoS information.

These parallels do not imply technical conformance but show that the XR5.0 architecture has been designed in a way that is compatible with MEC-style distributed processing and data locality requirements.

Conceptual Alignment with ETSI AR Framework and Immersive Technologies Work

ETSI ISG AR (Augmented Reality) and related groups define reference architectures for interoperable and scalable AR systems.

The XR5.0 architecture follows several conceptual principles from ETSI AR work items as follows:

- **Separation of content creation, content delivery, and rendering**, reflected in XR5.0's:
    - Training Authoring Tool
    - Cloud Repository
    - XR Training Clients (XR/VR apps)
- **Support for modular, reusable XR assets**, consistent with ETSI's emphasis on componentized AR content pipelines.
- **Use of standardized APIs** for accessing training content and AI services, which aligns with ETSI recommendations for API-driven interoperability.
- **Cross-device support**, since training content is intended to run on multiple XR interfaces (headsets, mobile devices), also consistent with ETSI's device-agnostic AR vision.

It has to be noted that the aforementioned relationships indicate architectural alignment and not technical certification.

Security and Trust Alignment with ETSI Cybersecurity Standards
Several ETSI specifications address authentication, authorization, token management, and trust frameworks across distributed systems. The **XR5.0 security blueprint** adopts industry best practices that are compatible with general ETSI cybersecurity principles.

The following areas of alignment have been identified:

- ➢ The use of OAuth2/OpenID Connect (IAM, Token Generation Service, Policy Enforcement Service) fits within ETSI's guidance for standardized, interoperable identity and access management.
- ➢ The architecture supports **tenant separation;** a principle found in ETSI standards related to virtualization and multi-tenant cloud infrastructures.
- ➢ XR5.0's **token validation and revocation processes** reflect ETSI's emphasis on secure lifecycle management of credentials.
- ➢ The adoption of role-based and scope-based access control is consistent with ETSI" s broader recommendations for fine-grained authorization in multi-service environments.

Added Value of XR5.0 beyond ETSI Specifications
While ETSI specifications provide general architectural patterns and guidelines, the XR5.0 Reference Architecture extends them in several ways that are specific to human-centric Industry 5.0 operations:

- ➢ XR5.0 introduces Human Digital Twin-driven personalization, a focus not explicitly covered in current ETSI documents related to edge computing or AR frameworks.
- ➢ The architecture integrates multiple AI components (Active Learning Model, Generative AI services, Neurosymbolic AI, Explainable AI) in a coordinated workflow tailored to training use cases.
- ➢ XR5.0 provides a complete value stream perspective for immersive industrial training, going beyond the technology-centric scope of typical ETSI frameworks
- ➢ XR5.0 delivers a multi-layer enterprise architecture (Business, Application, Technology layers), offering a broader strategic perspective compared to the component-level focus of ETSI specifications.
- ➢ The security blueprint applies OAuth2 in a domain-specific context (XR training, operator safety, cross-pilot tenanting), which extends beyond standard telecom use cases.

These contributions demonstrate that the XR5.0 Reference Architecture is not only aligned with ETSI best practices but also advances a more integrated, human-centric design approach tailored to Industry 5.0 needs.

7.2.5 XR5.0 Reference Architecture and XRSI

The **XR Safety Initiative (XRSI)** provides a widely recognized set of principles and high-level frameworks aimed at ensuring **privacy, safety, security, ethical behaviour, and responsible design** within immersive technologies. These frameworks do not act as strict technical standards but as guidance and best-practice baselines for organizations that develop or deploy XR solutions. The XR5.0 Reference Architecture, particularly in its emphasis on human-centric design, aligns well with the objectives and spirit of the XRSI guidelines.

Because XR5.0 targets the industrial domain of **Industry 5.0**, where operators interact with immersive training content, real-time guidance, and biometric-enabled personalization mechanisms, considerations related to **trust, user safety, secure data flows, and responsible handling of human-related information** become essential. Even though XRSI guidelines are not implemented as formal compliance artefacts within the project, they serve as **reference points** for framing the architectural logic around user protection and trustworthy XR experiences.

Conceptual Alignment
The XR5.0 Architecture demonstrates conceptual alignment with several foundational XRSI themes:

Human-Centricity and Wellbeing

XRSI places the user at the center of immersive interactions, emphasizing wellbeing, workload balance, and the minimization of cognitive or physical strain. This connects naturally with XR5.0's focus on:

➢ Adaptive training workflows

➢ Personalized guidance

➢ Operator stress-awareness

➢ Clear, understandable information presentation

These principles ensure that the XR5.0 ecosystem is designed to support safe and manageable operator experiences.

Privacy and Protection of User-Related Information

XR5.0 highlights the need for careful management of user-identifying or biometric information. In XR5.0, this is addressed conceptually through:

➢ Secure access control in training and assistance workflows

➢ Repositories and application services designed with privacy-aware logic

No claim is made about specific implementation mechanisms; however, the Architecture incorporates the idea that sensitive data must flow through controlled and well-defined services, which XRSI's recommendations.

Security and Trust in XR Environments

XRSI stresses system trustworthiness and the avoidance of harmful or unsafe interactions. Correspondingly, XR5.0's Vision of Architecture and layered model emphasize:

➢ Controlled retrieval of training programs

➢ Safe delivery of XR experiences through dedicated plugins

➢ Oversight mechanisms (Training Manager, Remote Assistant)

➢ Prevention of unverified modifications of training assets

These architectural constructs support a trusted operational context, aligning with XRSI's safety-oriented goals.

Architectural Touchpoints
While XRSI does not define explicit architectural layers or interfaces, several XR5.0 components naturally reflect XRSI's protective principles:

➢ **Training Workflow Blueprint** provides structured, guided interactions, reducing the risk of operator confusion or misuse.

➢ **Personalized Training Service** promotes socially responsible adaptation, resonating with XRSI's emphasis on the wellbeing of XR users.

➢ Access Control, Cloud Repository and XR Delivery Plugin offer points where privacy-aware and safety-aware logic can be integrated.

- Generative AI-enabled content transformation is associated with XRSI's calls for transparency and responsible behaviour in AI-supported XR environments.

Even if no technical compliance claims are made, however these elements provide opportunities to incorporate safety considerations that XRSI promotes.

Summary and Added Value
The XR5.0 Reference Architecture does not attempt to fully implement or certify XRSI guidelines, nor does the project require strict adherence to them. However, the conceptual compatibility between XR5.0 and XRSI is significant:

- Both emphasize human-centric and safe XR interactions
- Both promote trust and responsible processing of user-related information, and secure operational environments.
- Both highlight the importance of clear workflows, control guidance, and protective system behaviour.

Consequently, XR5.0 uses XRSI as a reference framework to ensure that the architecture remains aligned with emerging expectations around XR ethics, safety, and user protection. This high-level alignment supports the long-term sustainability and trustworthiness of XR solutions developed within the Industry 5.0 context.

7.2.6 XR5.0 Reference Architecture and OpenXR

OpenXR, developed by the Khronos Group, is an open and royalty-free API standard designed to promote interoperability across XR devices and runtimes. Its core objective is to reduce fragmentation by enabling applications to run on different XR hardware through a unified interface. Although XR5.0 currently relies on a remote-rendering–based XR delivery model (Hololight Stream), OpenXR remains an important reference point for ensuring long-term portability and alignment with emerging XR standards.

XR5.0's architectural principles (such as modular XR applications, device-agnostic design, and support for heterogeneous XR hardware) are consistent with the goals of OpenXR. However, XR5.0 does not require or mandate OpenXR for current implementations. Instead, OpenXR is considered relevant at a conceptual and future-interoperability level, ensuring that the architecture remains open to potential integration with OpenXR-based runtimes should project needs evolve.

Conceptual Alignment with OpenXR
XR5.0 employs Hololight Stream, a specialized remote-rendering solution that streams high-fidelity XR content from cloud-based servers (Amazon Cloud Computing) to AR/VR devices. This setup relies on:

- Unity plugin integration
- MRTK support
- WebRTC communications
- Session traversal utilities (NAT/STUN/TURN) on XR end-devices

While this approach differs from the native-runtime model typically associated with OpenXR, several conceptual alignments still exist:

- **Device abstraction and heterogeneous hardware support**

  XR5.0's use of remote rendering inherently enables device-agnostic operation, which is philosophically aligned with OpenXR's goal of decoupling content from hardware differences.

- **Modular XR application design**

The XR applications, the XR training plugin and visualization modules in XR5.0 are structured in a way that could, in the future, interface with OpenXR-compliant runtimes if required

- **Interaction and input models**

  XR5.0 supports natural user interactions through MRTK and Unity. These interaction paradigms overlap conceptually with the input abstraction model defined in OpenXR, although current implementation does not rely on OpenXR APIs.

- **Future-proof extension capabilities**

  OpenXR's extension mechanism aligns with the XR5.0 philosophy of supporting additional XR features, devices and rendering pipelines.

Relevance to XR5.0's Remote Rendering Approach
Because XR5.0 uses Hololight Stream for delivering XR content, several architectural benefits arise that parallel OpenXR's principles:

- **Cross-device compatibility** enabled through streaming rather than local execution.
- **Unified access to XR Training Projects**, regardless of device capabilities.
- **Reduced device fragmentation**, since rendering is centralized.

This means XR5.0 maintains future compatibility with OpenXR-aligned environments without committing to OpenXR today.

Potential Future Interoperability Pathways
While XR5.0 does not currently integrate OpenXR at the implementation level, the architecture allows for future interoperability in areas such as:

- Integration of OpenXR-based runtimes on new XR headsets
- Support for OpenXR interaction profiles (hand controllers, hand tracking, gaze, etc.)
- Compatibility with EU XR Platform initiatives promoting OpenXR adoption
- Extending the XR Training Plugin to interface with OpenXR APIs if required.

Summary
The relationship between XR5.0 and OpenXR is best characterized as strategic and conceptual:

- XR5.0 currently uses Hololight Stream, Unity, MRTK and WebRTC.
- The XR5.0 Architecture aligns with OpenXR principles at a high level, ensuring future compatibility.

This approach keeps XR5.0 open to mainstream XR standards while accurately reflecting the project's current technological choices.

7.2.7 XR5.0 Reference Architecture and STAR-RA

The STAR-RA reference architecture (Reference Architecture for AI-Based Industry 5.0 Systems) provides a conceptual framework for designing AI-enabled, human-centric industrial systems oriented to Industry 5.0 goals. STAR-RA emphasizes integration of trustworthy AI, human-centric interaction, digital twins, modular services, and governance mechanisms that ensure transparency, safety, and operator wellbeing. It is positioned as a high-level blueprint that supports the development of interoperable, auditable, and ethically aware AI-driven industrial solutions.

The XR5.0 Reference Architecture is designed with goals and components that are conceptually compatible with STAR-RA principles. XR5.0's focus on human digital twins, adaptive AI models, XR-based operator interfaces, and governance (security & IAM) reflects the same broad objectives that STAR-RA promotes. The following subsections describe this conceptual alignment, provide illustrative examples from XR5.0, and explain the added value XR5.0 brings relative to STAR-RA, while avoiding any claim of formal conformance or certification.

Layer and Component Mapping (Conceptual)

Below is a high-level mapping between STAR-RA's main concerns and XR5.0 architectural elements. This mapping is intentionally conceptual: it shows correspondences in intent and design rather than asserting technical compliance.

- **Human-AI Interaction & Governance (STAR-RA governance layer)**
  - **XR5.0 components:** IAM/Security Hub, Policy Enforcement Service, Token Generation/Validation, XAI services.
  - **Notes**: XR5.0 incorporates governance building blocks (authentication, scoped tokens, policy enforcement and explainability services) that conceptually address STAR-RA's governance and trust requirements.
- **AI Services & Lifecycle (STAR-RA AI Layer)**
  - **XR5.0 components:** GenAI, Active Learning, Neurosymbolic AI, XAI models, model management.
  - **Notes**: XR5.0 defines AI components and pipelines for content generation, adaptation and explanation, which align with STAR-RA emphasis on trustworthy AI lifecycles (training, validation, deployment, monitoring).
- **Human Digital Twins & Contextual Models (STAR-RA digital-twin domain)**
  - **XR5.0 components:** Human Digital Twin platform (Clawdite). Clawdite Historical Data Manager, Clawdite Orchestrator, Clawdite functional modules.
  - **Notes**: XR5.0's Human Digital Twin concept corresponds to STAR-RA's focus on contextual, human-centric models that feed AI personalization and safety logic.
- **Interaction & XR Interfaces (STAR-RA interaction / UX domain)**
  - **XR5.0 components:** XR Applications, XR Training Plugin, Stream Plugin, Trainer Authoring Tool.
  - **Notes:** XR5.0 provides immersive, adaptive interfaces that mirror STAR-RA's recommendations for ergonomics, explainability and operator-centric interaction.
- **Data & Integration (STAR-RA data/integration domain)**
  - **XR5.0 components:** OPC-UA Connector, ETL pipelines, Cloud Repository, Pilot ERP, sensor and biometric inputs.
  - **Notes:** XR5.0 collects and harmonizes multimodal data (sensors, ERP, biometric, XR telemetry) consistent with STAR-RA's emphasis on reliable, auditable data flows.
- **Runtime & Infrastructure (STAR-RA platform/infra domain)**
  - **XR5.0 components:** Kubernetes cluster, VMs, virtualization layer, streaming infrastructure (remote rendering), internal/external networks.
  - **Notes:** XR5.0's technology layer supports deployment, monitoring and scalable delivery, which conceptually map to STAR-RA's platform requirements for trustworthy, auditable execution.

These mappings illustrate intentional architectural alignment and design compatibility; they are not statements of certified or tested conformity.

Illustrative Implementation Examples (Conceptual)
The following illustrative examples show how XR5.0 implements design choices that are consistent with STAR-RA principles:

- **Governance & Explainability Example**

  XR5.0 exposes an XAI service that provides model explanations for active-learning decisions used in training adaptation. Conceptually, this addresses STAR-RA requirements for model transparency and operator-facing explainability. The IAM and Policy Enforcement Services mediate access to XAI outputs, aligning with governance expectations.

- **Human Digital Twin Usage Example**

  The Clawdite Human Digital Twin ingests sensor streams, operator questionnaire inputs, and historical performance data. STAR-RA emphasizes human-centered contextual models; XR5.0 uses the digital twin to generate personalized training adaptations and to inform safety-related decisions (e.g., adapting task difficulty based on stress indicators).

- **AI Lifecycle & Monitoring Example**

  XR5.0 model retraining workflows and active learning loops (fed by operator performance data) reflect STAR-RA's focus on lifecycle governance, monitoring and continuous validation at a conceptual level.

- **Interaction & UX Safety Example**

  The Personalized Training Blueprint defines pre/on/post training phases, live questionnaires and real-time biometric monitoring. STAR-RA recommends usability and safety evaluation: XR5.0 uses these artifacts to capture operator experience metrics that can be used in later validation and compliance activities.

- **Data Auditing & Traceability Example**

  XR5.0's Historical Data Manager collects time-stamped records of training sessions and operator responses. This supports traceability, auditability and post-hoc analysis, which are key STAR-RA concerns for accountability.

Added Value of XR5.0 Relative to STAR-RA
XR5.0 brings several domain-specific emphases to the general STAR-RA framing for AI-based Industry 5.0 systems:

- **Operationalized Human-Centric Training**

  XR5.0 combines immersive XR, AI personalization and human digital twins into a cohesive workflow specifically targeted at operator training and maintenance scenarios. This provides a concrete operational domain (training) where STAR-RA principles can be exercised.

- **Multimodal Human Context Integration**:

  XR5.0 explicitly integrates biometric data, behavioral telemetry and subjective feedback into the digital twin and AI loops, extending STAR-RA's conceptual guidance toward concrete multimodal application in XR training.

- **Security by Design in XR Contexts**:

  XR5.0's OAuth2-based IAM, tenant isolation and policy enforcement construct a governance pattern that targets the particular needs of XR environments (content integrity, streaming security, scoped access to biometric data) and complements STAR-RA's governance recommendations.

- **Plug-in and Orchestration Focus**:

The XR5.0 orchestration design (Hub, plugins, cloud repository) demonstrates a practical, modular approach to deploying STAR-RA-style components in a heterogeneous environment, easing future validation and incremental compliance where needed.

Conclusion

The XR5.0 Reference Architecture and STAR-RA share common high-level objectives: trustworthy AI, human-centric operation, auditable data flows and robust governance. XR5.0 is intentionally designed to be conceptually compatible with STAR-RA principles and to provide concrete domain applications (notably personalized XR training) where STAR-RA objectives can be realized.

7.3 Traceability Table

This section provides a consolidated view of how the standards discussed in this chapter conceptually relate to the XR5.0 Reference Architecture. The purpose of the traceability table is to ensure transparency and clarity regarding the architectural influences that guided the design of XR5.0. The table below highlights three elements for each standard:

- **XR5.0 Architectural Layers Influenced**: Business, Application, Technology.
- **Relevant XR5.0 Components or Concepts**: Taken exclusively from the validated diagrams (e.g. Training Workflow, Cloud Repository, XR Training Plugin, Streaming Plugin, Clawdite Platform, AI models)
- **Nature of Alignment**: A general description of how the standard conceptually informs XR5.0.

To this end, table 10 consolidates the conceptual alignment of all standards discussed in sections 7.2.1-7.2.7, mapping them to XR5.0 architectural layers and components. This traceability view provides a clear overview of how the XR5.0 Reference Architecture is informed by recognized frameworks.

Table 10 – Traceability: Standards vs. XR5.0 Architecture

Standard	XR5.0 Architectural Layers Influenced	Relevant XR5.0 Components/Concepts	Nature of Alignment (High-Level)
RAMI 4.0	Business, Application, Technology	Training Workflow Blueprint, Cloud Repository, Asset Representation, Interoperability Services	Provides general structuring principles and a multi-layer model that conceptually informs how XR5.0 separates business services, application logic, and technological enablers.
IIRA (Industrial Internet Reference Architecture)	Business, Application, Technology	XR Applications, Authoring Tool, System Management	Offers functional domains and cross-cutting concerns that support XR5.0's structuring of training services, data flows, and lifecycle management
ISO/IEC JTC (SC 41 focus)	Application, Technology	Device Interaction Layer, Data Exchange Services, Human Digital Twin concepts	Provides general guidance on interoperability, IoT-enabled environments, and digital twin conceptual models relevant to connected XR devices
ETSI Industry Specifications	Technology	Network Connectivity, Edge/Cloud Access, Remote Rendering Pathways	Offers high-level guidance on reliable networking, latency considerations, and distributed service environments that

			relate to XR5.0's streaming and deployment options
XRSI Frameworks	Business, Application	Identity Handling, Privacy Controls, Safety Considerations	Provides high-level principles for user safety, data protection, and responsible XR deployment, conceptually supporting XR5.0's human-centric orientation
OpenXR	Application, Technology	XR Runtime Interaction Layer, XR Training Plugin, Stream Plugin	Provides an interoperability model that informs future-proofing and potential runtime portability; conceptual influence only (no implementation)
STAR-RA (Reference Architecture for AI-Based Industry 5.0 Systems)	Business, Application	AI-Based Training and Authoring, Personalized Training Service, Human-Centric Learning	Informs the human-centric and AI-supported design principles of XR5.0, especially regarding personalization, human-AI teaming, and adaptability

The traceability table demonstrates that XR5.0 has been developed with awareness of relevant architectural frameworks and standards while maintaining flexibility for diverse industrial needs and partner technologies. The mapping reinforces that XR5.0 aligns conceptually with widely recognized reference models. This ensures that the architecture remains transparent in its design rationale and robust in terms of industry alignment.

7.4 Cross-Standard Analysis

Building on the conceptual mappings presented in section 7.3, this chapter analyzes the standards collectively to highlight cross-standard synergies, potential complementarities, and conceptual guidance for XR5.0 This cross-standard perspective ensures coherent alignment across architectural layers. The Cross-Standard Analysis provides a holistic view of how the various standards referenced in XR5.0 interact conceptually, highlighting areas of convergence, complementary guidance, and potential gaps. While each standard has been discussed individually in sections 7.2.1 to 7.2.7, examining them collectively allows for a higher-level understanding of how XR5.0 aligns with recognized architectural, interoperability, and human-centric principles in Industry 5.0 environments.

The primary objectives of this analysis are:

➢ **Identify Complementarities**:

Many standards address overlapping concerns from different perspectives. For example, RAMI 4.0 and IIRA both provide multi-layered reference models, though RAMI focuses on the manufacturing context and IIRA on the Industrial Internet. By considering both together, XR5.0 can leverage their shared structural principles (layered architecture, functional domains and cross-cutting concerns) while remaining flexible in implementation.

➢ **Highlight Conceptual Synergies**:

Standards such as STAR-RA, ISO/IEC JTC, and XRSI complement each other by emphasizing human-centric design, digital twin frameworks, and privacy/security considerations. XR5.0 benefits from these synergies by designing components such as the Clawdite Platform (Human Digital Twin), Personalized Training Services, and secure data flows in a way that conceptually aligns with multiple frameworks.

➢ **Ensure Layer-Level Consistency**:

A cross-standard perspective enables the alignment of Business, Application and Technology layers across the different standards.

- **Business Layer**: Concepts from RAMI 4.0, STAR-RA, and XRSI inform service-oriented structuring, role definitions, and human-centric workflows.
- **Application Layer**: IIRA and ISO/IEC JTC 1 provide guidance on modularity, interoperability, and AI-driven services, informing the orchestration of XR applications, AI services, and the Cloud Repository.
- **Technology Layer**: ETSI Industry Specifications and OpenXR inform networking, runtime compatibility, and remote rendering capabilities, as seen in Hololight Stream integration, without imposing strict implementation requirements.

➢ **Identify Potential Gaps and Future Opportunities**:

The cross-standard view highlights areas where standards could be more strongly leveraged in future extensions. For example, OpenXR is currently considered at a conceptual level to ensure future interoperability, while STAR-RA principles offer guidance for AI-based adaptability, which can be further extended as operational deployments evolve.

➢ **Facilitate Consistency and Traceability**:

By considering multiple standards together, XR5.0 can ensure that the rational for architectural decisions is consistently aligned with recognized frameworks. This cross-standard analysis reinforces the traceability introduced in section 7.3, showing how conceptual alignment is maintained across standards.

The Cross-Standard Analysis demonstrates that XR5.0 is not only aligned with individual standards but also designed to leverage their combined guidance in a coherent and consistent way. This approach enables conceptual interoperability, human-centric design, and layered architectural clarity. It provides the foundation for XR5.0 implementations, where additional compliance or adoption of further technical specifications could be explored as the platform matures.

7.5 Summary

This chapter has examined the relationship between the XR5.0 Reference Architecture and a set of widely recognized international standards and reference models, spanning industrial architectures (RAMI 4.0, IIRA), XR interoperability frameworks (OpenXR, XRSI), security and governance standards (including relevant ISO/IEC JTC1 families), and human-centric Industry 5.0 frameworks (such as STAR-RA). Across all analyses, the intention has not been to claim full compliance with individual technical specifications, but rather to demonstrate how XR5.0 conceptually aligns with established best practices while remaining flexible and implementation-agnostic.

The comparison shows that XR5.0 is consistent with the layered architectural principles promoted by RAMI 4.0 and IIRA, particularly in terms of distributed system organization, separation of concerns, and lifecycle-aware thinking. XR5.0 adopts XR interoperability principles (e.g. content abstraction, runtime independence, secure service access) that are complementary to initiatives such as OpenXR and XRSI. Likewise, the architecture reflects key ideas from ISO/IEC JTC 1 domains (such as modularity, data governance, cybersecurity and digital representation) at a conceptual level suitable for a cross-partner research project.

In the context of Industry 5.0, the XR5.0 architecture also conceptually resonates with the emerging Reference Architecture for AI-Based Industry 5.0 Systems (STAR-RA), particularly in areas such as human-centric design, explainable and trustworthy AI services, and interaction between digital models and physical operations. This alignment highlights that XR5.0 contributes to broader efforts in the European and international ecosystem toward safe, transparent, and human-focused cyber-physical systems.

Overall, the standards considered in this chapter serve as **reference points** for ensuring that the XR5.0 architecture remains interoperable, extensible, and compatible with evolving industrial requirements. The analyses provided here reinforce that XR5.0 is well-positioned to integrate with future platforms and standards developments, including the EU XR Platform, without constraining partners to strict technical dependencies. This conceptual alignment supports long-term sustainability of the project outcomes while allowing each pilot and technological partner to retain flexibility in their chosen implementation approach.

8. Plans for Boosting Compliance and Interoperability with the EU XR Platform

As the XR5.0 Reference Architecture is designed with the goal of aligning seamlessly with European initiatives, a major focus of the project is ensuring interoperability with the EU XR platform being developed as part of the XR2Industry project of the Innovation Type II call. While, at the time of writing, detailed documentation about the EU XR platform is not yet publicly available, XR5.0 has developed a proactive strategy that will allow for smooth integration once the relevant specifications and architecture are released.

Interoperability Goals and Strategic Design Considerations

XR5.0 aims to serve as a foundational architecture for human-centric, AI-enhanced XR applications within the context of Industry 5.0. The following interoperability goals have been identified to ensure XR5.0's alignment with the EU XR platform:

- **Flexible Integration through Standardized Interfaces**: The architecture of XR5.0 emphasizes modularity and flexibility, enabling straightforward integration with external platforms. XR5.0 employs standardized interfaces (e.g. RESTful APIs, WebSockets) for communication between the system's components. This allows for smooth connection with third-party platforms like the EU XR platform, ensuring compatibility with future data structures and communication protocols that may be specified by the EU project.
- **Data Interoperability and Semantic Alignment**: XR5.0 is committed to semantic interoperability, ensuring that data flows seamlessly between the two systems, regardless of the underlying technologies or platforms. By leveraging industry standards like RAMI 4.0 and OpenXR, XR5.0 guarantees that the data format and terminology used within its architecture can easily map to those in the EU XR platform. A key aspect of this is ensuring that business-level processes and application-level interactions in XR5.0 follow common ontologies that will likely be adopted by the EU XR platform.
- **Cross-Platform Accessibility:** Given the varied nature of devices and hardware in both XR and industry 5.0 environments, XR5.0 adopts cross-platform compatibility principles. The application layer of XR5.0 is built with frameworks that are not only flexible but also hardware-agnostic. For instance, XR5.0 has the capability to stream high quality XR content from cloud-based infrastructure to XR glasses and smartphones. Thereafter, the rendering process shifts from the low-performance XR device to the high-performance server and the XR app does not need to be installed on an end device. This feature characterizes XR5.0 as hardware-agnostic which ensures that the platform can operate across a range of devices and integrate easily with emerging XR hardware being developed for the EU XR platform.

Planned Actions for Integration

Although, at the time of writing, detailed information regarding the EU XR platform is not yet available, XR5.0 has developed a detailed roadmap for ensuring alignment as more information emerges as follows:

- **Engagement with the EU XR Platform Consortium:**
    - The XR5.0 team will actively seek to establish communication channels with the EU XR platform stakeholders. This will involve setting up initial meetings and exploring their vision for interoperability. The goal is to discuss and analyze the platform's architecture, expected use cases and alignment opportunities with XR5.0.
    - Collaborative workshops will be held to map out integration points, addressing challenges such as data flow, platform-specific requirements and user interface compatibility.

- ➢ **Revisiting and Extending the Architecture**: Once detailed specifications for the EU XR platform is available, XR5.0 will undergo an evaluation phase to determine which parts of its current architecture need to be adapted. Key areas of focus include:
  - **Business Layer: User** stories and personas in XR5.0 will be examined to ensure alignment with the EU XR platform's business processes. Potential changes in workflows, resource management, and human-centered application design will be incorporated to ensure that the two platforms complement each other seamlessly.
  - **Application Layer:** Integration points such as AI models XR interfaces and modular application containers will be assessed. XR5.0 is designed with flexibility in mind, allowing for the introduction of new application containers that integrate external systems from the EU XR platform with minimal disruption to existing workflows.
  - **Technology Layer:** The technical components of XR5.0, including its data storage, networking protocols and hardware abstractions will be analyzed for compatibility with the EU XR platform. New cross-platform tools and technologies may be incorporated to ensure that XR5.0 works across a broad range of hardware, from edge devices to cloud-based resources.
- ➢ **Proactive alignment with Industry 5.0 Standards**: XR5.0 aligns with leading standards in Industry 5.0, XR and AI. By utilizing frameworks such as RAMI 4.0, IIRA and XRSI, XR5.0 ensures that its architecture will comply with emerging standards set by the EU XR platform, particularly in terms of:
  - **Security and Privacy:** By adopting security and privacy standards, XR5.0 will align with the EU's data protection and privacy requirements, ensuring that it meets regulatory guidelines likely to be set by the EU XR platform.
  - **AI and Automation:** The incorporation of AI-based decision-making systems within the XR5.0 architecture (e.g. malfunction detection, process optimization, equipment recognition) positions XR5.0 as a natural complement to any AI-enhanced functionalities the EU XR platform may employ.
- ➢ **Validation:** Continuous validation with end-users from the XR5.0 will ensure that the integration does not impact the user experience while adding value through enhanced interoperability with the EU XR platform.

By following the aforementioned strategic plan, XR5.0 aims to achieve the following outcomes:

- ➢ **Enhanced Interoperability**: With a focus on standardized interfaces and data formats, XR5.0 ensures that both systems can exchange information effortlessly.
- ➢ **Scalability**: XR5.0's modular and scalable architecture is well-positioned to adapt to the evolving needs of the EU XR platform
- ➢ **Wider Adoption and Collaboration**: By aligning with EU standards and initiatives, XR5.0 ensures compatibility with broader Industry 5.0 frameworks. This positions the architecture as a practical and adaptable reference point, facilitating its use across diverse industrial applications and encouraging collaborative efforts in the XR domain.

9. CONCLUSIONS

This document presents the main outcomes of Deliverable D2.2, further enhanced and extended in this second iteration (D2.3). It reflects the updated XR5.0 Reference Architecture Model, the refined technical specifications of its components, and a more detailed alignment with relevant standards. This iteration provides a comprehensive and structured view of the architecture, incorporating new blueprints, detailed component diagrams, and enriched specifications, thereby offering increased clarity, consistency, and support for developers and stakeholders.

9.1 Concluding Remarks

The second iteration of the XR5.0 Reference Architecture (D2.3) has delivered a more complete and detailed description of the system, enhancing both clarity and usability for developers and stakeholders. Key updates include:

- ➢ **C4 Component Diagrams**: Detailed visualizations of each XR5.0 component, showing internal architecture, interactions, and dependencies. These diagrams facilitate understanding of system-level and component-level design decisions.
- ➢ **New Blueprints**: Three additional blueprints were introduced: (i) the Personalized Training Blueprint, (ii) the Training Workflow Blueprint, and (iii) the Security and Access Control Blueprint. They provide operational insight into XR5.0 processes, business functions, and secure access mechanisms.
- ➢ **Standards Alignment**: The analysis of relevant standards (RAMI 4.0, IIRA, ISO/IEC JTC 1, ETSI Industry Specifications, XRSI, OpenXR, STAR-RA) has been expanded and structured. This iteration incorporates more detailed mapping to the XR5.0 architecture, including layer-to-component alignments, conceptual integration with the technical specifications, and a comprehensive cross-standard comparison.
- ➢ **Technical Specifications Update**: Component descriptions were updated with details on APIs, internal architecture, software requirements, dependencies, and authentication mechanisms. This ensures modularity, interoperability, and easier integration across pilot scenarios.

Overall, the deliverable represents a robust, flexible, and interoperable framework, aligned with Industry 5.0 principles and forward compatible with the EU XR platform. It provides a strategic reference for development and integration activities, supporting human-centric XR and AI solutions across multiple industrial use cases.

9.2 Lessons Learned

Several practical insights have emerged during the creation of this second iteration:

- ➢ **Visualization Benefits**: Using C4 component diagrams and multiple architectural blueprints significantly improved the clarity of the architecture and helped communicate complex interactions between layers and components.
- ➢ **Blueprint Utility**: The addition of the three new blueprints clarified operational workflows, personalized training mechanisms, and security/access control procedures, providing a more actionable perspective on system behavior.
- ➢ **Standards Alignment Strategy**: The process of analyzing and aligning the XR5.0 architecture with existing standards highlighted several key insights. First, a structured approach to mapping architecture layers and components against standards such as RAMI 4.0, IIRA, ISO/IEC JTC 1, and STAR-RA ensures clarity in understanding relationships and dependencies, which is essential for interoperability and future integration. Second, conceptual alignment provides a framework that accommodates evolving technologies and allows the architecture to remain adaptable. Finally, documenting alignment at multiple levels (business, application, and technology) supports

traceability, informs future design decisions, and establishes a foundation for any subsequent efforts to achieve formal standard conformance in XR5.0 components or services.

- **Enhanced Technical Documentation**: Detailed technical specifications (APIs, dependencies, authentication mechanisms) proved crucial for ensuring modularity, maintainability, and integration readiness across the XR5.0 ecosystem.
- **Integration Across Layers**: Coordinating the business, application, and technology layers reinforced the need for consistency and traceability, especially when aligning diagrams, blueprints, and technical specifications.
- **Forward-Looking Perspective**: Structuring the architecture to remain open to future standards and EU XR platform integration underlined the importance of adaptability in long-term project planning.

These lessons provide guidance for future development, integration, and standardization efforts, ensuring that XR5.0 can evolve effectively while remaining aligned with human-centric Industry 5.0 objectives.

REFERENCES

[1] Vázquez-Ingelmo, A., García-Holgado, A., & García-Peñalvo, F. J. (2020, April). C4 model in a Software Engineering subject to ease the comprehension of UML and the software. In 2020 IEEE Global Engineering Education Conference (EDUCON) (pp. 919-924). IEEE.

[2] XR5.0, D2.1 - Requirements and Reference Scenarios Analysis, 2024.

[3] The Open Group, ArchiMate 3.2 Specification. The Open Group, 2023. [Online]. Available: https://pubs.opengroup.org/architecture/archimate3-doc/

[4] S. Brown, "C4 Model for Software Architecture," [Online]. Available: https://c4model.com/. [Accessed: 28-Nov-2024]

[5] Adolphs, P., Bedenbender, H., Dirzus, D., Ehlich, M., Epple, U., Hankel, M., Wollschlaeger, M. (2015) "Reference Architecture Model Industrie 4.0 (RAMI 4.0)". VDI/VDE and ZVEI. Retrieved from https://www.zvei.org/en/subjects/industry-4-0/the-reference-architectural-model-rami- 40-and-the-industrie-40-component/ [last accessed 21/05/2021]

[6] The Open Group, *TOGAF® Standard, 10th Edition: A Pocket Guide*. The Open Group, 2022.

[7] Industrial Internet Consortium.: The Industrial Internet Reference Architecture v 1.9. Available at: https://www.iiconsortium.org/IIRA.htm. Last accessed 26 June 2023

[8] ISO/IEC/IEEE.: ISO/IEC/IEEE 42010:2011 Systems and software engineering – Architecture description. http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508 (2011). Last accessed 26 June 2023

[9] A. Buchner, M. Höchtl, T. Kramler, and E. Veith, "Modeling of Customer-Oriented Applications in Product Lifecycle Using RAMI 4.0," *International Journal of Modeling and Optimization*, vol. 9, no. 1, pp. 1–7, Feb. 2019. DOI: 10.7763/IJMO.2019.V9.676.

[10] F. Figueroa, W. Pavon, H. Calvopiña, D. Mideros, and F. Castro De la Cruz, "Integration of the Asset Administration Shell (AAS) for Smart Manufacturing: State-of-Art and Future Opportunities," *Sciety*, Jun. 2024. [Online]. Available: https://sciety.org/articles/activity/10.20944/preprints202406.1271.v1.

[11] Binder, M., Müller, G., & Schmidt, S. (2024). Ensuring the application of Industry 4.0 design principles by using reference architectures. *Sensors & Transducers*, 265(2), 97–105. Available: https://sensorsportal.com/HTML/DIGEST/may_2024/Vol_265/p_3338.pdf

[12] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*, Springer, 2020, pp. 169–186. [Online]. Available: https://www.researchgate.net/publication/264589945_Research_Directions_for_the_Internet_of_Things .

[13] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2020. [Online]. Available: https://www.researchgate.net/publication/264589945_Research_Directions_for_the_Internet_of_Things

[14] ETSI, "ETSI Multi-access Edge Computing (MEC)," [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing. [Accessed: Dec. 7, 2024].

[15] ETSI, "IoT and oneM2M Standards," [Online]. Available: https://www.etsi.org/technologies/internet-of-things. [Accessed: Dec. 7, 2024].

[16] ETSI, "Cyber Security for IoT Standards," [Online]. Available: https://www.etsi.org/technologies/cyber-security. [Accessed: Dec. 7, 2024].

[17] ETSI, "5G and Industrial Networks," [Online]. Available: https://www.etsi.org/technologies/5g. [Accessed: Dec. 7, 2024].

[18] Chang, W. (2022). ISO/IEC JTC 1/SC 42 (AI)/WG 2 (data) data quality for analytics and machine learning (ML). *Information Technology Laboratory*.

[19] Spiekermann, S. (2021). What to expect from IEEE 7000: The first standard for building ethical systems. *IEEE Technology and Society Magazine*, *40*(3), 99-100.

[20] XRSI, "XRSI Privacy and Safety Framework," [Online]. Available: https://xrsi.org/. [Accessed: Dec. 7, 2024].

[21] M. Slater and M. V. Sanchez-Vives, "Enhancing Our Lives With Immersive Virtual Reality," *Frontiers in Robotics and AI*, vol. 3, no. 74, 2016, [Online]. Available: https://www.frontiersin.org/. [Accessed: Dec. 7, 2024].

[22] M. Venkatesan, H. Mohan, J. R. Ryan, C. M. Schürch, G. P. Nolan, D. H. Frakes, and A. F. Coskun, "Virtual and augmented reality for biomedical applications," *Cell Reports Medicine*, vol. 2, no. 7, Jul. 2021, Art no. 100348, doi: 10.1016/j.xcrm.2021.100348.

[23] STAR-AI EU Funded Project, "Human-Centric Artificial Intelligence Architecture for Industry 5.0 Applications," *STAR-AI*, 2024. [Online]. Available: https://star-ai.eu/new-publication-human-centric-artificial-intelligence-architecture-industry-50-applications

[24] I. Papanikolaou, et al., "Reference Architecture for AI-Based Industry 5.0 Applications," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/378099277_Reference_Architecture_for_AI-Based_Industry_50_Applications

[25] XRSI, "Medical XR Privacy & Safety Framework," Medical XR Council, 2021. [Online]. Available: https://xrsi.org/medical-xr-privacy-safety-framework. [Accessed: Dec. 10, 2024].

[26] XRSI, "Ethics in Immersive Technologies," XR Safety Initiative, 2021. Available: https://xrsi.org/ethics. [Accessed: Dec. 10, 2024].

[27] XRSI, "Collaborating for a Safer Metaverse," XR Safety Initiative, 2021. Available: https://xrsi.org/metaverse-collaboration. [Accessed: Dec. 10, 2024].

ANNEX I.

In this annex, a brief explanation of the various ArchiMate elements that were leveraged to design the various layers of the XR5.0 (Human-Centric AI-Enabled Extended Reality Applications for the Industry 5.0 Era) will be presented. For a more thorough review, the reader may consult the thorough online tutorial provided by the Open Group located at:

https://pubs.opengroup.org/architecture/archimate3-doc/

The descriptions of the elements used are taken from the online tutorial. Table 11 lists the main motivation and strategy elements used for the strategy layer. Motivations are used for modelling the reasons that guide the design of the architecture, while strategy elements model the directions and choices of the enterprise that have a direct impact on how the architecture is designed. Business elements that are used to model the operational and organizational aspects of the enterprise are depicted in Table 12. The application elements of an architecture model, the structure, behavior and interaction of the applications that compose the solution are depicted in Table 13. Finally, the structure and behavior of the technology infrastructure is described by the technology elements in the ArchiMate approach; these elements are depicted in Table 14.

Table 11 – Strategy and motivation elements used

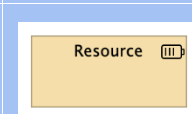
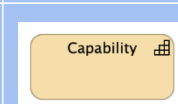
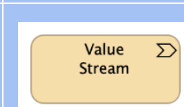
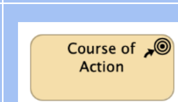
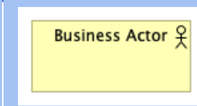
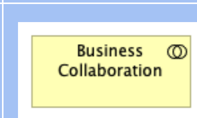
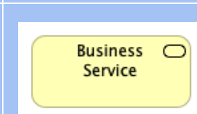
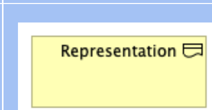
Element	Description	Notation
Goal	Represents a high-level statement of intent, direction, or desired end state for an organization and its stakeholders.	
Resource	Represents an asset owned or controlled by an individual or organization.	
Capability	Represents an ability that an active structure element, such as an organization, person, or system, possesses.	
Value Stream	Represents a sequence of activities that create an overall result for a customer, stakeholder, or end user	
Course of action	Represents an approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal.	

Table 12 – Business elements used

Element	Description	Notation
Business Actor	Represents a business entity that is capable of performing behavior.	
Business Role	Represents the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.	
Business Collaboration	Represents an aggregate of two or more business internal active structure elements that work together to perform collective behavior.	
Business interaction	Represents a unit of collective business behavior performed by (a collaboration of) two or more business actors, business roles, or business collaborations.	
Business service	Represents explicitly defined behavior that a business role, business actor, or business collaboration exposes to its environment.	
Contract	Represents a formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.	
Representation	Represents a perceptible form of the information carried by a business object.	

Product	Represents a coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.	

Table 13 – Application elements used

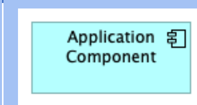
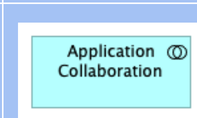
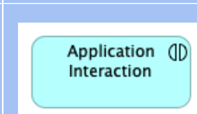
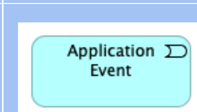
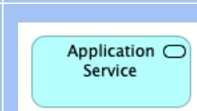
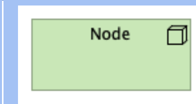
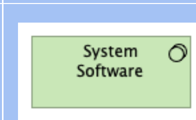
Element	Description	Notation
Application Component	Represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable.	
Application Collaboration	Represents an aggregate of two or more application internal active structure elements that work together to perform collective application behavior.	
Application function	Represents automated behavior that can be performed by an application component.	
Application Interaction	Represents a unit of collective application behavior performed by (a collaboration of) two or more application components.	
Application event	Represents an application state change.	
Application Service	Represents an explicitly defined exposed application behavior.	
Data Object	Represents data structured for automated processing.	

Table 14 – Technology elements used

Element	Description	Notation
Node	Represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.	
System Software	Represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it	
Technology Process	Represents a sequence of technology behaviors that achieves a specific result.	
Facility	Represents a unit of collective application behavior performed by (a collaboration of) two or more application components.