Ref. Ares(2026)9527728 - 09/10/2026

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

D6.4 – INTEGRATED PILOT SYSTEMS AND USE CASES DEPLOYMENT V2

Lead Beneficiary	ATB
Work Package Ref.	WP6 – XR5.0 Apps Integration, Validation and Evaluation
Task Ref.	T6.2 – Rapid Human-Centric AI-enabled Product Design T6.3 – Human Centred Remote Maintenance and Asset Management T6.4 – XR Streaming for Operator 5.0 Training T6.5 – Personalised Aircraft Maintenance Training T6.6 – XR Training for Effective Assembly Operations T6.7 – Human Centric Guidance and Troubleshooting
Deliverable Title	D6.4 – Integrated Pilot Systems and Use Cases Deployment V2
Due Date	2026-09-30
Delivered Date	2026-10-08
Revision Number	3.0
Dissemination Level	Public (PU)
Type	Demonstrator (DEM)
Document Status	Release
Review Status	Internally Reviewed and Quality Assurance Reviewed
Document Acceptance	WP Leader Accepted and 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

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union. Neither the EASME nor the European Commission is responsible for any use that may be made of the information contained therein.

COPYRIGHT MESSAGE

This report, if not confidential, is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0); a copy is available here: https://creativecommons.org/licenses/by/4.0/. You are free to share (copy and redistribute the material in any medium or format) and adapt (remix, transform, and build upon the material for any purpose, even commercially) under the following terms: (i) attribution (you must give appropriate credit, provide a link to the license, and indicate if changes were made; you may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use); (ii) no additional restrictions (you may not apply legal terms or technological measures that legally restrict others from doing anything the license permits).

CONTRIBUTING PARTNERS

Partner Acronym	Role[1]	Name Surname[2]
ATB	Lead Beneficiary	Fulya Horozal Sebastian Scholze
	Contributor	Fulya Horozal Umesh Sevra
COR	Contributor	Marcelo Corrales
CYENS	Contributor	Andreas Andreou Sophia Ppali
EKSO	Contributor	Karin Ladjeri
HOLO	Contributor	Harsh Manoj Shah
IML	Contributor	Jorge Oliveira Tomás Saraiva
INNOV	Contributor	George Fatouros Despina Tomkou
IPIA	Contributor	António Rosinha Toacy Oliveira Joaquim Reis
KUKA	Contributor	Georgios Karantinakis Christian Behrens
LNS	Contributor	Philipp Saurer
OCU	Contributor	Mathias Horner
RW	Contributor	Nataly Ramirez
SH	Contributor	Joshua Barbian
SIE	Contributor	Robert Fejers
SPACE	Contributor	Nikos Papadakis Nikolaos Zafeiropoulos
SSF	Contributor	Bastien Perrissoud
SUPSI	Contributor	Elias Montini Samuele Dell'Oca Vincenzo Cutrona
SYN	Contributor	Emmanouil Mavrogiorgis Stylianos Tsanakas
	Internal Reviewer	Emmanouil Mavrogiorgis
TAP	Contributor	Luís Pimentel de Oliveira
TOG	Contributor	Scott Hansen
UBI	Contributor	Nikolaos Tousert Pavlos Kranas
UNP	Contributor	Bruno Almeida Giovanni Di Orio

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

UPRC	Contributor	Dimitrios Dardanis Stergios Chairistanidis
	Internal Reviewer	Dimitrios Dardanis
	Quality Assurance	Thanos Kiourtis

REVISION HISTORY

Version	Date	Partner(s)	Description
0.1	2026-08-01	ATB	ToC Version
0.2	2026-08-18	ATB	Initial Version
0.3	2026-09-18	ATB, CYENS, HOLO, IML, INNOV, KUKA, SIE, SUPSI, SSF, SYN, UPRC	Partner Contributions in Sections 2 - 7
0.4	2026-09-22	OCU, UNP	Partner Contributions in Sections 1 and 3
0.5	2026-09-23	UNP	Partner Contributions in Section 6
0.6	2026-09-23	ATB	Harmonization of Partner Contributions
0.7	2026-09-24	ATB	Addition of Sections 1 and 8
1.0	2026-09-25	ATB	1st Version for Peer Reviewers
1.1	2026-09-29	SYN, UPRC	1st and 2nd Internal Review Feedback
1.2	2026-09-30	ATB	Integration of 1st and 2nd Internal Review Feedback
1.3	2026-10-01	INNOV	Integration of 1st and 2nd Internal Review Feedback
1.4	2026-10-05	SSF	Integration of 1st and 2nd Internal Review Feedback
2.0	2026-10-06	ATB	Version for Quality Assurance
3.0	2026-10-08	ATB	Version for Submission

LIST OF ABBREVIATIONS

Acronym	Definition
AAS	Asset Administration Shell
AAC	Advanced Audio Coding
AI	Artificial Intelligence
AL	Active Learning
AMT	Aircraft Maintenance Technician
API	Application Programming Interface
AR	Augmented Reality
ASR	Automatic Speech Recognition
BLE	Bluetooth Low Energy
CAD	Computer-Aided Design
CBM	Concept Bottleneck Models
CCTV	Closed-Circuit Television
CPU	Central Processing Unit
CRSM	Customer Relationship & Service Management
CRUD	Create, Read, Update and Delete
CSV	Comma-Separated Values
CUDA	Compute Unified Device Architecture
DT	Digital Twin
EP	Early Prototype
ETL	Extract, Transform and Load
FBX	Filmbox
FLAC	Free Lossless Audio Codec
FP	Full Prototype
GenAI	Generative AI
GPS	Global Positioning System
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
GradCAM	Gradient-weighted Class Activation Mapping
HDT	Human Digital Twin
HF	High Frequency
HMD	Head-mounted Display
HRV	Heart Rate Variability
HTTP	Hypertext Transfer Protocol
ID	Identifier
IIoT	Industrial Internet of Things
IoT	Internet of Things
I4.0	Industry 4.0
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LF	Low Frequency

LIME	Local Interpretable Model-agnostic Explanations
LLM	Large Language Models
MQTT	Message Queuing Telemetry Transport
MPEG	Moving Picture Experts Group
MP3	MPEG-1 Audio Layer III
MP4	MPEG-4 Part 14 (audio-only MP4)
M4A	MPEG-4 Audio
NSAI	Neurosymbolic Artificial Intelligence
OPC UA	Open Platform Communications Unified Architecture
OGG	Ogg Vorbis
PLC	Programmable Logic Controller
QR	Quick Response
RAG	Retrieval Augmented Generation
RAMI 4.0	Reference Architecture Model for Industry 4.0
REST	Representational State Transfer
RMSSD	Root Mean Square of Successive Differences
SDK	Software Development Kit
SSO	Single Sign-On
STT	Speech-to-Text
TPAT	Training Programmes Authoring Tool
TTS	Text-to-Speech
UC	Use Case
URDF	Unified Robot Description Format
USD	Universal Scene Description
UI	User Interface
UUID	Universally Unique Identifier
VR	Virtual Reality
XAI	Explainable Artificial Intelligence
XR	Extended Reality
XRTP	Extended Reality Training Plugin
VITS	Variational Inference Text to Speech
WAIV	Wing Anti-Ice Valve
WAV	Waveform Audio File Format
WebM	Web Media
WLAN	Wireless Local Area Network
WMA	Windows Media Audio
WP	Work Package

EXECUTIVE SUMMARY

Deliverable D6.4 – *Integrated Pilot Systems and Use Cases Deployment V2* documents the XR5.0 project's final integration activities and outcomes across six industrial pilots. Each pilot is implemented within its designated task (T6.2 – T6.7) and is designed to demonstrate how Extended Reality (XR) and Artificial Intelligence (AI) can transform industrial training, maintenance, troubleshooting and operations. D6.4 extends the integration activities reported in D6.3, which documented the Early Prototypes of the six XR5.0 industrial pilots. D6.4 covers the remaining integration activities carried out during the second Pilot Integration Phase, from Month 25 to Month 32, and presents the Full Prototype implementations of all six pilots.

The key outcome of this phase is the successful complete integration of XR5.0 components into working Pilot Final Prototypes, thereby validating the viability of the XR5.0 Reference Architecture from work package WP2. D6.4 examines how the full prototypes bring together XR5.0 software components from WP3–WP5 within each pilot's application environment. Their implementation provides further evidence of the applicability of the WP2 Reference Architecture across different industrial settings and of the interoperability of its components. Alongside the integration outcomes for each pilot, the deliverable brings together findings on usability, data availability, and system integration. The pilot integration activities demonstrate the technical feasibility, interoperability and value of human-centric, AI-enabled XR solutions across diverse industrial domains while revealing shared challenges and lessons relevant to their deployment in real-world settings.

TABLE OF CONTENTS

1. Introduction 14
1.1 Objectives of the Deliverable 14
1.2 Insights from Other Tasks and Deliverables 14
1.3 Updates from Previous Version 15
1.4 Structure 16
2. Pilot 1 Full Prototype 17
2.1 Pilot-specific Infrastructure 18
2.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies 19
2.2.1 XR Application 19
2.2.2 XR-enabled AI Assistant 43
2.2.3 XR-based Training Program 49
2.2.4 XR5.0 Clawdite Platform – Workers' Movement Prediction Module 50
2.2.5 Safety-Risk Zone Checker 51
2.2.6 NOVAAS for Digitalizing XR Physical Assets and a AASs Manager 52
2.2.7 KUKA OPC UA Server 52
2.2.8 XR Worker-Robot Position Prediction Prototype 52
2.3 Component Usage Matrix 58
2.4 Discussions 61
3. Pilot 2 Full Prototype 63
3.1 Pilot-specific Infrastructure 66
3.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies 66
3.2.1 Mobile Troubleshooting AI Assistant 66
3.2.2 Mainstatic SHARE Platform 67
3.2.3 XR Human-Centered XAI Models 68
3.2.4 QR-Code Workflow Handover 69
3.3 Component Usage Matrix 70
3.4 Discussions 72
4. Pilot 3 Full Prototype 73
4.1 Pilot specific Infrastructure 74
4.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies 74
4.2.1 AR Application 74
4.2.2 XR-enabled AI Assistant 79
4.2.3 Active Learning and Neuro-symbolic AI 84
4.2.4 Explainable AI 85

4.2.5 Smart Pipes API 86
4.3 Component Usage Matrix 88
4.4 Discussions 90
5. Pilot 4 Full Prototype 92
5.1 Pilot-specific Infrastructure 93
5.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies 94
5.2.1 XR-based Training Program and Materials 94
5.2.2 Hololight Hub 95
5.2.3 Training Material: 3D Model Explorer Addon 96
5.2.4 Training Material: 3D Workflow Tool 97
5.2.5 XR-enabled AI Assistant Addon 97
5.2.6 XR-enabled AL & NSAI Models for XR-AI Detector 99
5.2.7 Workers' Digital Twin: Worker Shadowing and Monitoring 107
5.2.8 Adaptation Engine: Adaptation of Personalised XR Content 108
5.3 Component Usage Matrix 108
5.4 Discussions 110
6. Pilot 5 Full Prototype 112
6.1 Pilot specific Infrastructure 113
6.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies 114
6.2.1 XR App 114
6.2.2 Hololight Hub/Stream 119
6.2.3 XR5.0 Training Management Platform 119
6.2.4 XR5.0 Clawdite Platform 121
6.2.5 Smartwatch Connector 121
6.2.6 Stress Index Calculator 122
6.2.7 Stress Detection Module 125
6.2.8 XR5.0 Chat Engine 127
6.3 Component Usage Matrix 127
6.4 Discussions 129
7. Pilot 6 Full Prototype 132
7.1 Pilot specific Infrastructure 132
7.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies 134
7.2.1 Troubleshooting Assistant (Gateway) 134
7.2.2 AR Visualisation and RealWear Arc 3 Integration 135
7.2.3 XR-enabled AI Assistant and Expertise-level Personalisation 136
7.2.5 End-to-End Integration and Technical Testing 138
7.3 Component Usage Matrix 139

7.4 Discussions 141
8. Conclusions 143
8.1 Summary 143
8.2 Lessons Learned 147

LIST OF FIGURES

Figure 1 – WP6 Roadmap 14
Figure 2 – Relation to Other Work Packages 15
Figure 3 – Pilot #1 Architecture 18
Figure 4 – Interaction with the AR Overlay 20
Figure 5 – Interaction with the AR Overlay – Cross-sectional View 20
Figure 6 – The Safety-risk Zone Checker enabled and disabled via Voice Commands 21
Figure 7 – Users can trigger the component navigation functionality via the hand-menu or via the voice command "Component Navigator" 22
Figure 8 – The Final Component Navigator User Interface Panel with the Button Functions to speak, reset and receive Information 23
Figure 9 – The button function "Info" further displays the available components to navigate within the 3D model with an option to click and isolate 24
Figure 10 – The component navigator feature enabled wherein the "Milling Tool" is isolated with the rest of the components rendered in X-ray view allowing for easier navigation and component identification 25
Figure 11 – Once a component is isolated, RoboSpace also allows for easy 3D manipulation of the isolated component - move, rotate, scale 26
Figure 12 – Example of isolating and identifying the robot arm component of the KUKA robot crawler model 27
Figure 13 – The KUKABot interface provides assistance to a user's technical query. RoboSpace displays the results of the query in 2D panels 29
Figure 14 – RoboSpace additionally also provides the option to view the reasoning of the model's response 30
Figure 15 – A realistic use case of KUKABot and Component Navigator in assessing and understanding the robot arm of the robot crawler model 31
Figure 16 – Robot Arm motion panel with two buttons, one to connect to the live OPC UA data source and the second to connect to a simulated data source 32
Figure 17 – When the user selects the OPC UA data source, the application loads another panel with options to connect, disconnect and reset. "Connect" makes a successful connection to the robot's PLC via OPC UA and via a VPN 32
Figure 18 – When the user selects "Connect", the robot starts reading and consuming the values from the PLC and the user visualises robot movement virtually. Appropriate error handling has been implemented in case a connection to the PLC is not successful 33
Figure 19 – When the user selects "Disconnect", the robot movement stops as the application stops reading the live values 34
Figure 20 – When the user selects the simulation data source, the application loads preloaded robot kinematic data with options to play, pause and stop 35
Figure 21 – Pilot 1 RoboSpace XR application available on the XR5.0 Training Platform within KUKA's tenancy 36
Figure 22 – The TPAT interface enables KUKA admins to create training programs for their technicians and operators 37
Figure 23 – The TPAT interface enables users to create different models of training materials such as Checklist, Image, Quiz, Video and Workflow. These materials can be added to several different training programs 37
Figure 24 – An Example of a Training Material for KUKA 38
Figure 25 – The TPAT interface allows admins to view the training program progress status of their users 39
Figure 26 – The training programs available within the RoboSpace app via XRTP. These programs were created by the admin and assigned to the user by their role 40
Figure 27 – Interactive panels of the XRTP within RoboSpace allowing users to perform training easily .. 41

Figure 28 – Admins can add images to their training programs and users can view them directly on the XR headset 42
Figure 29 – XRTP allows users to view the status and completion of their training programs 42
Figure 30 – An Example of a Checklist Training Material 43
Figure 31 – DataLens High-level Architecture 44
Figure 32 – High-level Overview of the Authentication Integration with the XR5.0 Hub 47
Figure 33 – Operator Interface Depiction showing Collections, Documents and Analytics 48
Figure 34 – Workflow Depiction for the Deployment Process of the DataLens Service 49
Figure 35 – Training Program in Pilot #1 50
Figure 36 – Training Program Interactive Instruction Panels 50
Figure 37 – Architecture of the XR Worker-Robot Prediction Prototype 53
Figure 38 – Logical Data Flow based on Classical Broker Architectural Pattern 54
Figure 39 – Worker UI Page 56
Figure 40 – XR Worker-Robot Position Prediction Prototype - Humans Detection from Camera System 57
Figure 41 – XR Worker-Robot Position Prediction Prototype - Humans Detection from Camera System 58
Figure 42 – Pilot #2 Architecture 65
Figure 43 – User Interaction with the Mobile Troubleshooting AI Assistant 67
Figure 44 – Visualization of textual XAI explanations 68
Figure 45 – QR Code with Workflow IDs 69
Figure 46 – Pilot #3 Architecture 74
Figure 47 – AR Pipe Model Visualization 75
Figure 48 – Sensor Data Visualization in Pilot #3 AR App 76
Figure 49 – Visualization of Anomaly Detection in Pilot #3 AR App 77
Figure 50 – Chatbot Conversation Window 78
Figure 51 – Pilot #3 AR App UI 79
Figure 52 – Pilot #3 XR Training Plugin 79
Figure 53 – XR5.0 AI Assistant Overview 80
Figure 54 – XR5.0 AI Assistant Knowledge Base 81
Figure 55 – XR5.0 AI Assistant Query Pipeline 82
Figure 56 – XR5.0 AI Chat View 82
Figure 57 – XR5.0 API Overview A. 83
Figure 58 – XR5.0 API Overview B. 84
Figure 59 – Active Learning Integration in Pilot #3 85
Figure 60 – NSAI Integration in Pilot #3 85
Figure 61 – XAI Integration in Pilot #3 86
Figure 62 – High Level Architecture of Smart Pipes API 87
Figure 63 – Pilot #4 Architecture Diagram 93
Figure 64 – XR Training Program Outline 94
Figure 65 – Training Program using the XR Versions of Training Materials 95
Figure 66 – Hololight Hub Architecture 95
Figure 67 – 3D Model Explorer Interface 96
Figure 68 – 3D Model Explorer View 96
Figure 69 – 3D Model Workflow XR View 97
Figure 70 – XR-enabled AI Assistant Addon 98
Figure 71 – Integration of the SIE Conversational AI Services within the XR5.0 99
Figure 72 – System Architecture 100
Figure 73 – Annotation of a Test Image using the XR-AI Annotator 101
Figure 74 – Image captured and annotated using the XR-AI Annotator 102
Figure 75 – WingValve Detection 103
Figure 76 – On-demand Explanation of the AI Detection using Heatmaps 104
Figure 77 – API Endpoints 106

Figure 78 – Shadow Operator 108
Figure 79 – Pilot #5 Architecture 113
Figure 80 – Pilot #5 XR Application Main Menu 115
Figure 81 – 3D Guardian with Scenario Wiring in the Pilot #5 XR Application 115
Figure 82 – Training Objectives for the Commissioning Scenario 116
Figure 83 – In-headset Knowledge Check Retrieved from TPAT 117
Figure 84 – Assessment Result with Stress Intervals 117
Figure 85 – User Panel with Worker Profile and Live Biometrics 118
Figure 86 – XR5.0 Chat Panel in the Pilot #5 XR Application 118
Figure 87 – Scenario Questions and Related Materials in TPAT 120
Figure 88 – Asset Extracted from the Guardian Manual in TPAT 120
Figure 89 – Training Program Retrieval through the XR5.0 Training Platform REST API 121
Figure 90 – Training Platform Materials Rendered in the Headset 121
Figure 91 – Smartwatch and Companion Application in Use during a Pilot #5 Session 122
Figure 92 – Stress Index Calculator Dashboard 123
Figure 93 – Stress Index Calculator Integration Architecture Diagram 124
Figure 94 – Stress Index Calculator Input Data Flows and Topics 124
Figure 95 – Stress Index Calculator Output Data Flows and Topic 125
Figure 96 – Stress Detection Module – Technical Architecture 126
Figure 97 – Guardian Manual Ingested in the XR5.0 Chat Engine Document Store 127
Figure 98 – Pilot #6 Architecture including Helen S, AI Troubleshooting Services and RealWear Arc 3 134
Figure 99 – Pilot #6 Troubleshooting Assistant Service Architecture 135
Figure 100 – Visualization of Troubleshooting Instructions 136
Figure 101 – Helen S and Progressive Escalation Workflow 137
Figure 102 – Pilot #6 LNS Troubleshooting Process 138

LIST OF TABLES

Table 1 – KUKABot API Interfaces used within Pilot #1 44
Table 2 – XR Worker-Robot Position Prediction Prototype Outcomes 52
Table 3 – Mapping of Pilot #1 Technical Components to Pilot Use Cases in Full Prototype 61
Table 4 – XR5.0 Technical Components in the Pilot #2 Full Prototype 70
Table 5 – Mapping of Pilot #2 Technical Components to Pilot Use Cases in Full Prototype 71
Table 6 – SmartPipes API Endpoints 88
Table 7 – XR5.0 Technical Components in Pilot #3 Full Prototype 88
Table 8 – Mapping of Pilot #3 Technical Components to Pilot Use Cases in Full Prototype 90
Table 9 – XR5.0 Technical Components in the Pilot #4 Full Prototype 108
Table 10 – Mapping of Pilot #4 Technical Components to Pilot Use Cases in Full Prototype 110
Table 11 – XR5.0 Technical Components in Pilot #5 Full Prototype 128
Table 12 – Mapping of Pilot #5 Technical Components to Pilot Use Cases in Full Prototype 129
Table 13 – Pilot #6 Technical Test Scenarios and Acceptance Criteria 139
Table 14 – XR5.0 Technical Components in Pilot #6 Full Prototype 140
Table 15 – Mapping of Pilot #6 Technical Components to Pilot Use Cases in Full Prototype 141
Table 16 – Mapping of XR5.0 Technical Components to the Pilots' Full Prototypes 144

LIST OF CODES

Code 1 – DataLens RAG Pipeline Response Schema 46

1. INTRODUCTION

The scope of Deliverable D6.4 – *Integrated Pilot Systems and Use Cases Deployment V2* is to document the second and the final phase of integration and deployment activities for the six XR5.0 industrial pilots within WP6 – *XR5.0 Apps Integration, Validation and Evaluation*. It provides a consolidated view of how pilot-specific requirements were aligned with the XR5.0 reference architecture from WP2 and how software and infrastructure components developed in earlier work packages were deployed to build Pilot Full Prototypes.

In WP6, each pilot is implemented as part of a designated task T6.2-T6.7:

- Pilot #1: T6.2 – Rapid Human-Centric AI-enabled Product Design
- Pilot #2: T6.3 – Human Centred Remote Maintenance and Asset Management
- Pilot #3: T6.4 – XR Streaming for Operator 5.0 Training
- Pilot #4: T6.5 – Personalised Aircraft Maintenance Training
- Pilot #5: T6.6 – XR Training for Effective Assembly Operations
- Pilot #6: T6.7 – Human Centric Guidance and Troubleshooting

1.1 OBJECTIVES OF THE DELIVERABLE

The primary objective of this deliverable is to present the outcomes of the final pilot integration phase (Pilot Integration Phase 2, see Figure 1) from Month 25–Month 32, focusing on the technical components integrated and the deployment environment in each pilot. The emphasis is on describing how the pilots were assembled from XR5.0 technical components and deployed in their respective industrial contexts. The document captures integration outcomes in order to inform subsequent development, evaluation and refinement activities within the project. The Pilot Full Prototypes will be evaluated in the second and final Pilot Evaluation Phase (Month 33-Month 36) as part of T6.8 (XR5.0 Apps Evaluation and Stakeholders' Feedback).

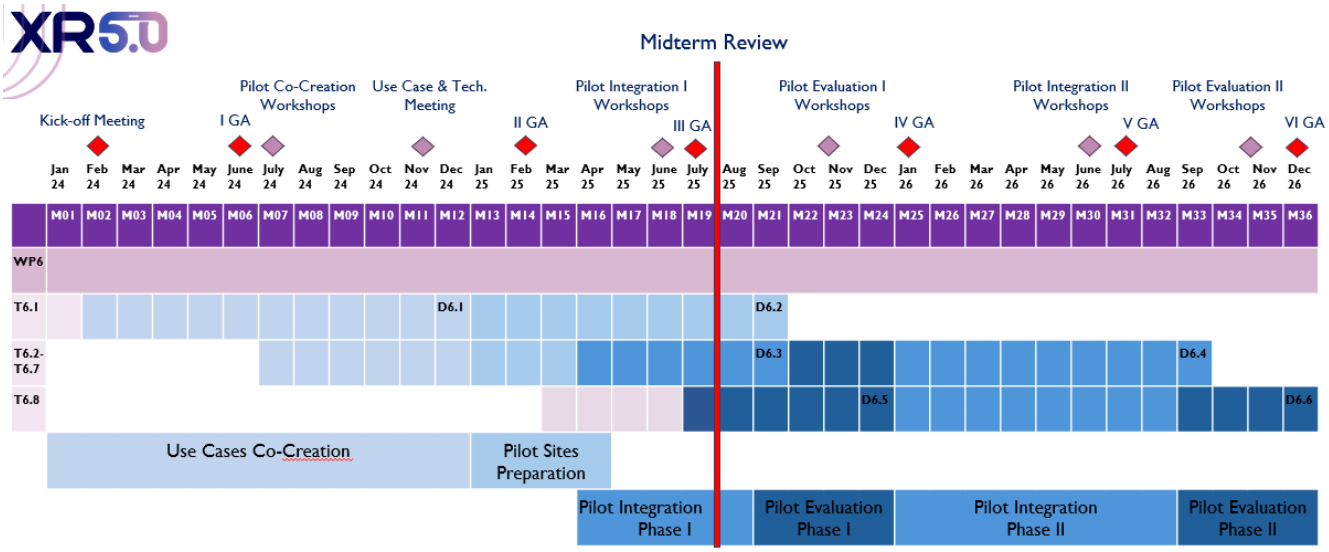


Figure 1 – WP6 Roadmap

1.2 INSIGHTS FROM OTHER TASKS AND DELIVERABLES

This deliverable is directly linked to the broader project workflow. It builds on the technical developments from WP3 (Human Centric Digital Twins and XR Personalisation), WP4 (AI/XR Symbiosis for Augmented Intelligence), and WP5 (XR-enabled Operator 5.0 Training), and demonstrates their application within the pilot environments coordinated under WP6. D6.4 acts as a bridge between technical development and pilot

evaluation to ensure continuity and alignment across XR5.0 work packages. The lessons captured in this deliverable will provide critical input for the upcoming deliverable on final pilot evaluation (D6.6).

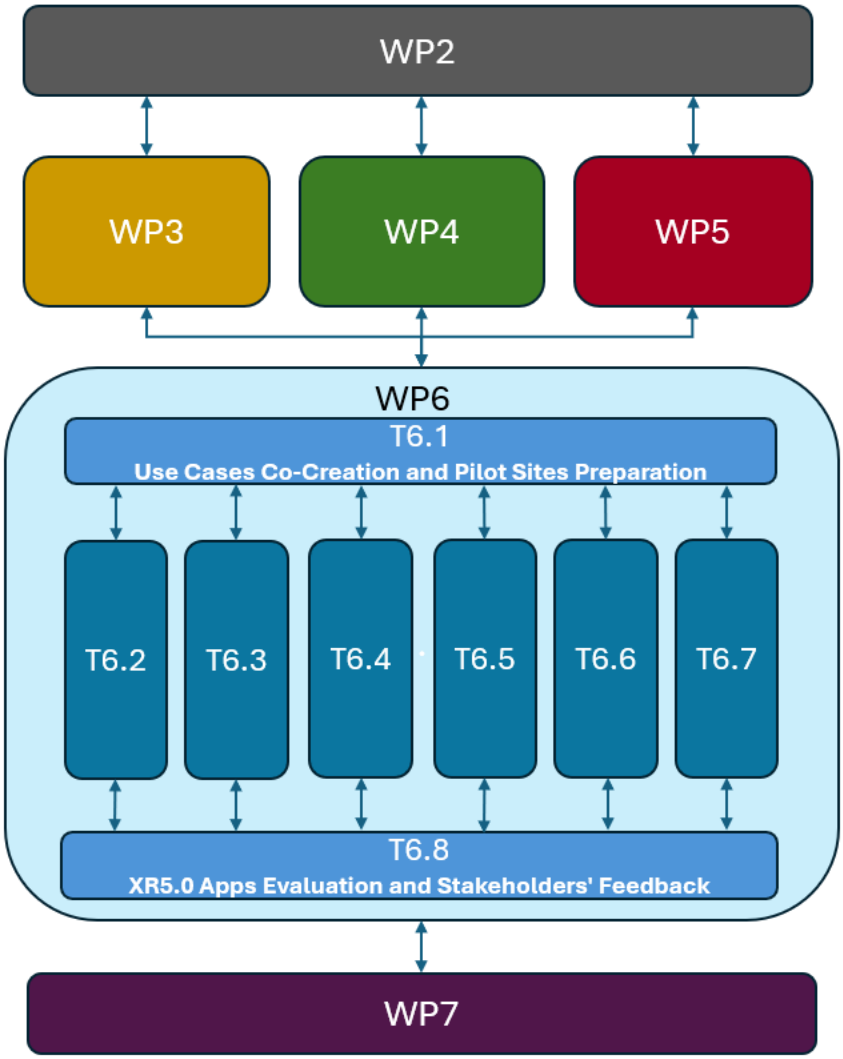


Figure 2 – Relation to Other Work Packages

Figure 2 illustrates the interactions and dependencies between the various WPs and tasks of XR5.0 that interact with WP6. WP2 focuses on the design, implementation, and analysis of XR5.0 technologies, including the XR5.0 reference architecture, and provides inputs to the technical work packages WP3-WP5. The technical work packages WP3-WP5 contribute to WP6 by providing technical components for each pilot in WP6. Finally, WP7 supports the overall project exploitation and dissemination activities to ensure the seamless interaction and flow of resources between these interconnected components.

1.3 Updates from Previous Version

This section summarises the main changes in each pilot since the early prototypes reported in D6.3. It highlights how the full prototypes extend the XR applications, AI support, training functions and integration of pilot systems.

Pilot 1 has expanded its XR capabilities to support detailed inspection of robot parts and visualisation of simulated and live robot movements. Cloud-based delivery now enables central management of applications and training, while the AI assistant provides improved conversational support and more

transparent responses. The full prototype also integrates visual and audible safety warnings and extends the exchange of live robot and worker-position data across the pilot systems.

Pilot 2 has connected mobile AI troubleshooting, asset and ticket information, remote expert support, and XR guidance into a more complete maintenance workflow. The assistant now uses the active service context to retrieve equipment-specific documentation and guide technicians towards relevant procedures. Reusable XR maintenance and training content has been integrated with the XR5.0 Training Platform. The combined workflow was validated in an industrial maintenance scenario at SCHMIDT + HAENSCH.

Pilot 3 has shifted from an AR headset concept to a mobile AR application in response to operators' practical and security requirements. The mobile interface has been simplified, with improved spatial positioning and access to sensor data, anomaly findings, explanations and recommendations. The AI assistant now supports both text and voice interaction, and structured training has been integrated and tested on mobile devices. The Smart Pipes backend has also gained more robust data handling, API access and authentication.

Pilot 4 has developed its aircraft maintenance training into distinct mixed reality and virtual reality applications. The training environment now includes a documentation-grounded voice assistant, while aircraft-part recognition has been extended with expert annotation and feedback, active learning, and explanations of detections. The worker digital twin has been refined to replay expert movements and assess trainees against key procedural milestones, alongside added heart-rate monitoring and personalised content adaptation.

Pilot 5 has expanded from an early XR assembly training prototype into a more complete training and repair support application. Training supervisors can now author levels, materials and assessments that are delivered in the headset, including a practical device commissioning scenario. Biometric data is used to show stress feedback and adjust the difficulty of the next training level. For repair support, technicians can consult device documentation through a voice-enabled AI assistant and monitor device messages in XR. The application now accesses these training and AI capabilities through integrated services.

Pilot 6 has refined its AI troubleshooting guidance with an explicit expertise level, allowing instructions to vary more precisely in detail for different technicians. The integrated workflow has been tested on the RealWear Arc 3, extending support to newer wearable hardware. Helen S has also been added as an entry point for straightforward service requests, with users able to move to hands-free XR guidance for more complex procedures.

1.4 STRUCTURE

The deliverable is structured systematically to present the integration outcomes of each XR5.0 pilot (T6.2-T6.7) in a designated section. Following this introductory section, Section 2 to Section 7 present each Pilot Full Prototype from T6.2 to T6.7, respectively. Each pilot section consists of an overview, an introduction of pilot-specific infrastructure, a description of pilot-specific installation, customization and configuration of XR5.0 technologies, component usage matrix, and a discussion of key achievements, identified challenges, findings, planned improvements and risks. Section 8 concludes the deliverable.

2. PILOT 1 FULL PROTOTYPE

Pilot #1 – T6.2 Rapid Human-Centric AI-enabled Product Design, led by ATB, has developed a full prototype XR application aimed at enabling personalized XR experiences coupled with AI services in assembly line production and virtual commissioning. It offers an interactive augmented reality experience for industrial maintenance and training tasks addressing scenarios from the pilot use cases. Pilot #1 use cases are as follows (details of the pilot use cases have been documented in deliverable D6.1 Use Cases Co-Creation and Pilot Sites Preparation):

- **UC 1.1 – Personalised XR in Assembly Line Production:** This use case focuses on on-site technician support through AI-enabled XR-based solutions.
- **UC 1.2 – Virtual Commissioning and Generative AI in Robotics**: This use case focuses on technician training and support for virtual commissioning operations using XR-based solutions.

The Pilot #1 Full Prototype XR application and the technical components in the pilot have been co-developed with partners ATB, KUKA, HOLO, SIE, IML, UNP and SUPSI. The key technical components developed and integrated include a Generative AI module (T4.3), textual XAI support (T4.1), XR5.0 Training Platform (T5.4) providing an XR-based training programme using KUKA training materials (T5.1), a safety risk warning module using the XR5.0 Clawdite Platform (T3.3) for workers movement prediction and an Asset Administration Shell (AAS) framework (T3.2).

The Pilot #1 Full Prototype builds on the RoboSpace XR application and the component integrations documented for the Early Prototype in D6.3. RoboSpace streams true-to-scale, interactive robot models to HoloLens 2, overlays live sensor and machine data on the corresponding virtual components, and supports hands-free interaction through voice commands. Its Component Navigator now lets users isolate and inspect individual robot parts, while the robot-motion view supports both simulated playback and visualisation of live joint-position data received through OPC UA. The application integrates KUKABot, a retrieval-augmented AI assistant developed by SIE, to provide voice and text access to KUKA technical documentation. The updated assistant supports contextual follow-up questions and returns source references and textual reasoning information for display in XR. IML's interactive SmartPad-2 training programme is delivered through the XR Training Plugin, and the integration with the XR5.0 Training Platform enables training content to be authored, assigned and tracked alongside the RoboSpace application. The Workers Movement Prediction module of the XR5.0 Clawdite Platform predicts worker positions over short time horizons. ATB's Safety-Risk Zone Checker evaluates these positions and RoboSpace supplies visual and audible warnings when a worker is identified as being at risk. The full prototype also connects worker-position information to Asset Administration Shells (AASs) developed with UNP's NOVAAS framework, supporting standardised exchange with the robot's data systems.

Figure 3 shows the architecture of the Pilot #1 Full Prototype and its integration with the XR5.0 Training Platform. When a user launches the AR application on HoloLens 2, Hololight Hub orchestrates the virtual machines that run the XR application and stream it to the headset in real time. The Full Prototype also incorporates cloud deployment.

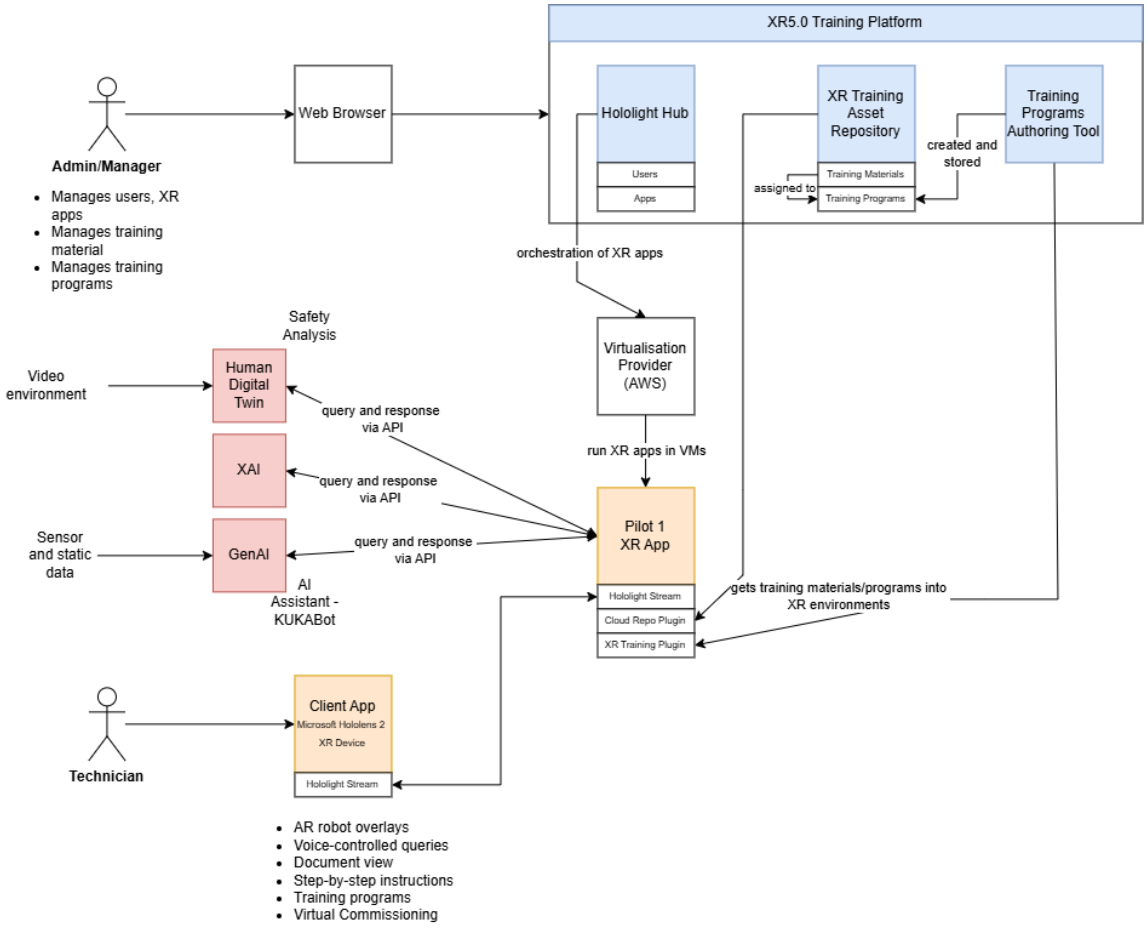


Figure 3 – Pilot #1 Architecture

2.1 PILOT-SPECIFIC INFRASTRUCTURE

Pilot #1 Full Prototype specific infrastructure consists of the following components:

- KUKA construction robot
- 3D models of the KUKA construction robot
- Hardware:
  - Microsoft Hololens 2 XR device
  - A high-performance PC
- Software:
  - RoboSpace – Pilot #1 XR application
  - Hololight Space XR engineering application for AR/VR devices
  - Hololight Stream SDK streaming technology
  - XR-based AI Assistant KUKABot
  - XR-based Training Platform
  - XR5.0 Clawdite Platform
  - Safety-risk Zone Checker API
  - KUKA OPC UA Server
  - NOVAAS for the KUKA construction robot
  - NOVAAS for the WPM functional module
  - XR Worker-Robot Position Prediction Prototype

2.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies

2.2.1 XR Application

The Pilot #1 Full Prototype XR application, RoboSpace, is developed by HOLO for the end-user KUKA. Built on Hololight Space, it is hosted through Hololight Hub within the XR5.0 Training Platform and securely streamed to Microsoft HoloLens 2 using Hololight Stream. The application presents interactive, true-to-scale robot models as AR overlays aligned with the physical environment. It combines inspection and analysis tools with component-level navigation, real-time machine data, and robot-motion visualisation using simulated sequences or live joint positions. Voice interaction and the integrated KUKABot AI assistant provide access to technical documentation and textual explanations, while XR training programmes guide technicians through SmartPad-2 operation. The full prototype also integrates visual and audible safety warnings based on current and predicted worker positions.

True-to-Scale Robot Visualization

RoboSpace displays a detailed KUKA robot model at its physical dimensions, allowing operators to examine it from different viewpoints and position it within their workspace. This supports training and operational inspection by helping users understand the robot's spatial arrangement and its relationship to surrounding equipment. Alignment with the physical environment is achieved through QR-code markers or the Two Point Alignment Tool, which uses two reference points to establish the model's position and orientation.

Interactive Tools for Inspection and Analysis

RoboSpace provides several tools for examining robot models in an industrial setting, as illustrated in Figure 4 and Figure 5:

- **Explosion view**: Separates the model's components to reveal their arrangement without dismantling the physical robot.
- **3D model manipulation**: Allows users to select, move, rotate and resize models using grab and bounding-box controls.
- **Cross-sectional views**: Uses planar, cubic or spherical cutting tools to expose internal structures.
- **Measurement and annotation tools**: Enables users to measure model features and attach notes to support inspection and communication.
- **Collision detection**: Helps identify potential intersections between robot components or surrounding equipment.

These functions support on-site inspection, analysis and technical decision-making within the XR environment.

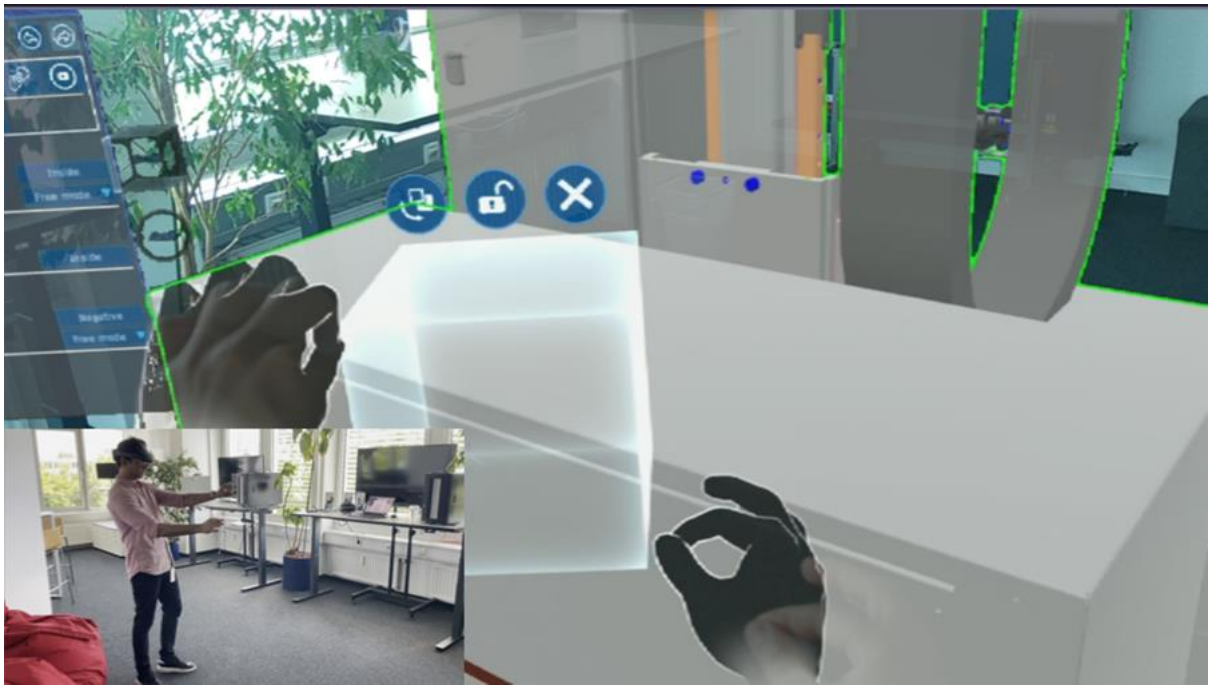


Figure 4 – Interaction with the AR Overlay

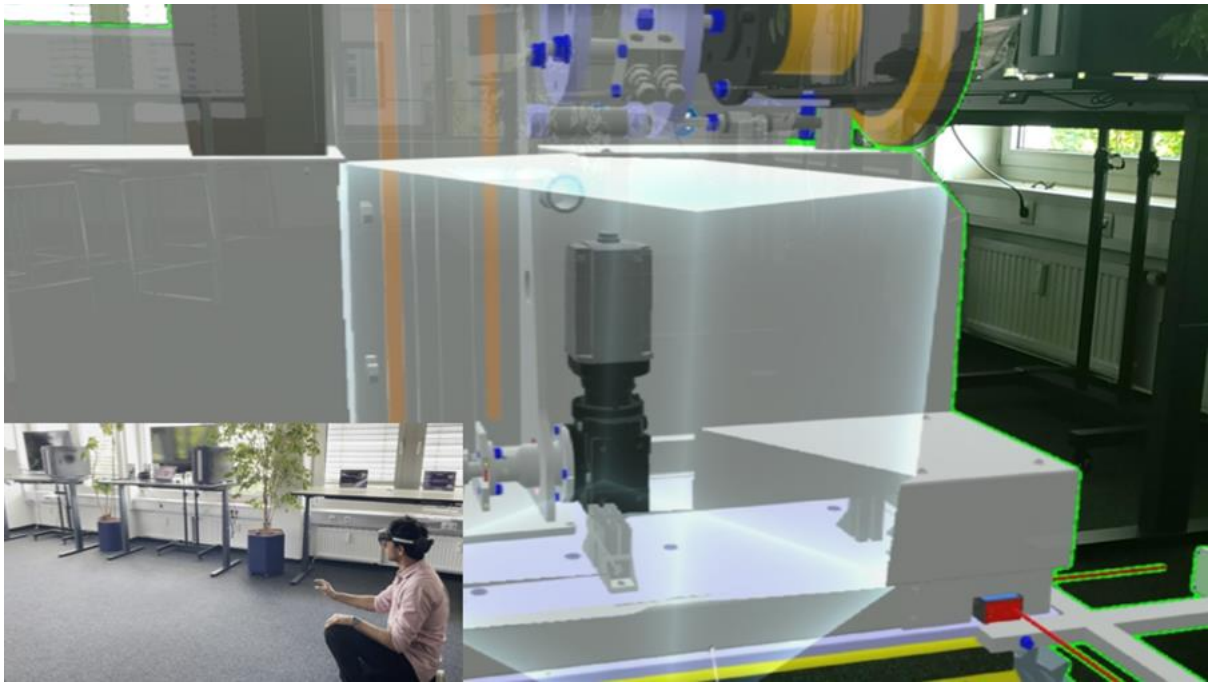


Figure 5 – Interaction with the AR Overlay – Cross-sectional View

Real-Time Data Visualization with OPC UA

A major strength of the application lies in its integration with KUKA's **OPC UA servers**. Integration with KUKA's OPC UA servers connects the virtual robot model to live information from machine controllers and sensors. Process values are displayed at the corresponding locations on the model; for example, distance-

sensor readings can appear beside the relevant part of the robot arm. This contextual presentation lets operators inspect machine data within the XR view alongside the equipment it describes.

Voice-Driven UI for Industrial Environments

In production settings, hands are often occupied, and using controllers or touch interfaces is impractical. Our application supports **voice-driven interaction**, enabling users to control the interface, switch tools, or request information through simple spoken commands. This feature makes the system far more usable in noisy, hands-on environments.

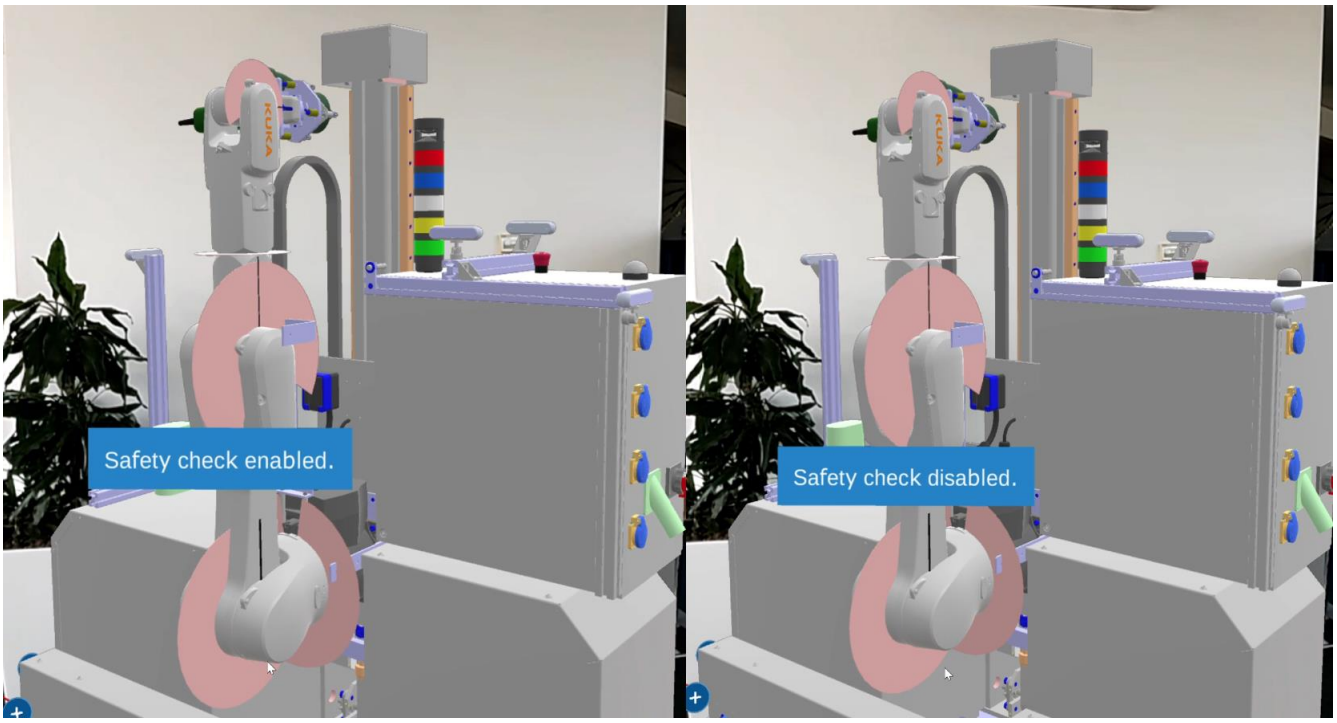


Figure 6 – The Safety-risk Zone Checker enabled and disabled via Voice Commands

Model Component Navigation

Component Navigator is a feature within RoboSpace that allows the user to isolate and inspect a single component of the loaded 3D model — for example, one part of the robot crawler or robot arm — rather than viewing the model as a whole. The feature can be triggered either by speaking the name of the desired component or by opening a dedicated panel and selecting it from a list of currently available parts.

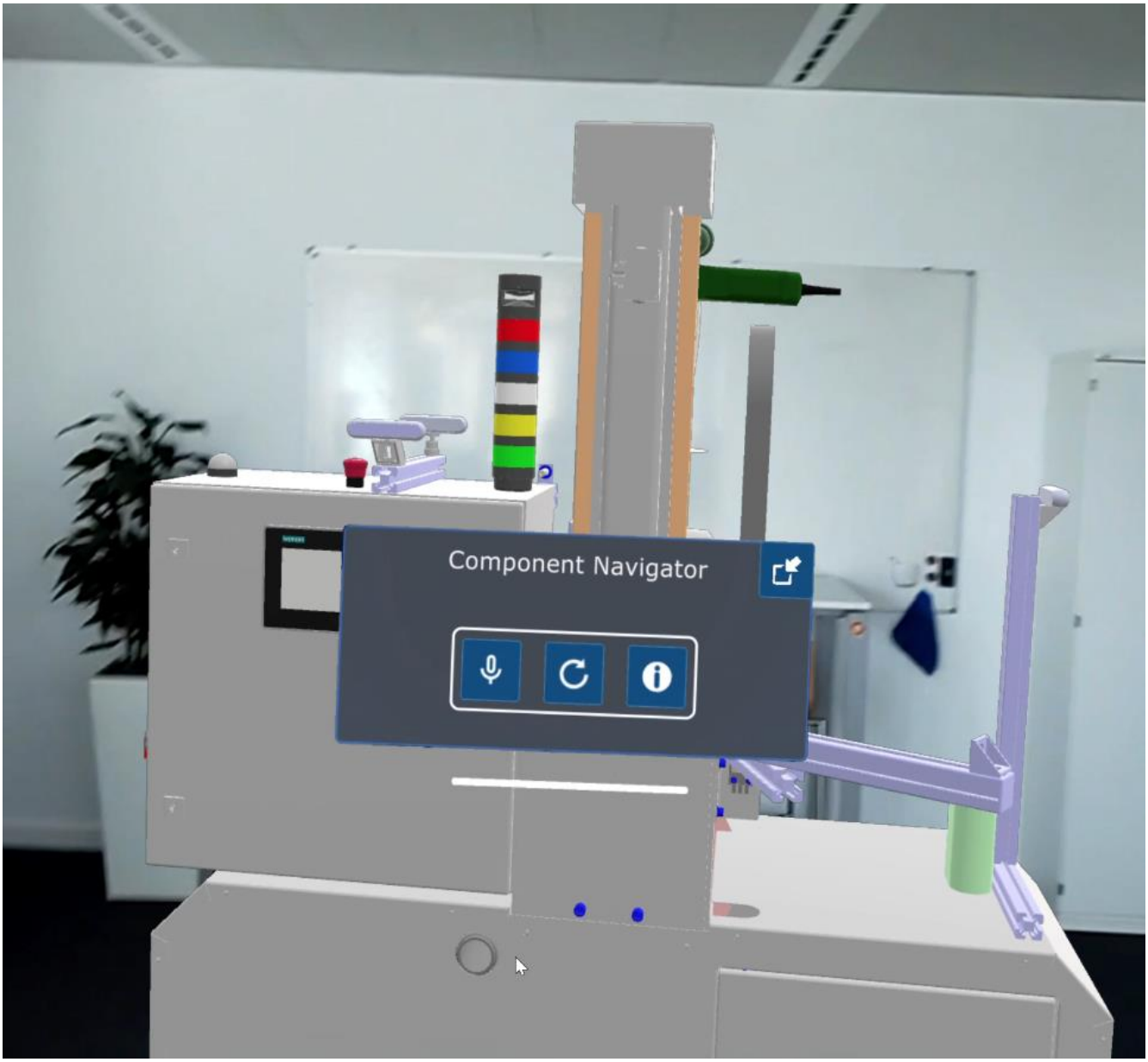


Figure 7 – Users can trigger the component navigation functionality via the hand-menu or via the voice command “Component Navigator”

Development began by identifying a way to isolate a single component, either by voice or keyboard input, with the rest of the model dimmed, hidden, or rendered in X-ray view so the selected part stands out clearly. This was extended so that a spoken, plain-language name for a part (a "function text") could be mapped onto the underlying CAD data: an algorithm was built that reads the mechanical component identifiers from the model's CAD-derived data and links each one to its corresponding function text. This mapping was first developed for the robot crawler, validated against a full station model, and later re-adapted when an updated CAD source file was supplied by the manufacturing partner.

The user-facing panel went through several rounds of refinement. An early version added a dedicated navigator button opening a small panel where the user could activate the microphone briefly to speak a component name, with the panel also listing available components for reference and providing clear feedback when the feature is unavailable for the currently loaded model. The design was then simplified to

a compact interface with a microphone control, a reset option, and an information button revealing the full list of components — each one clickable, so a user can select a component directly if voice input fails or isn't preferred. Hovering over a highlighted part also now reveals its internal identifier for reference.

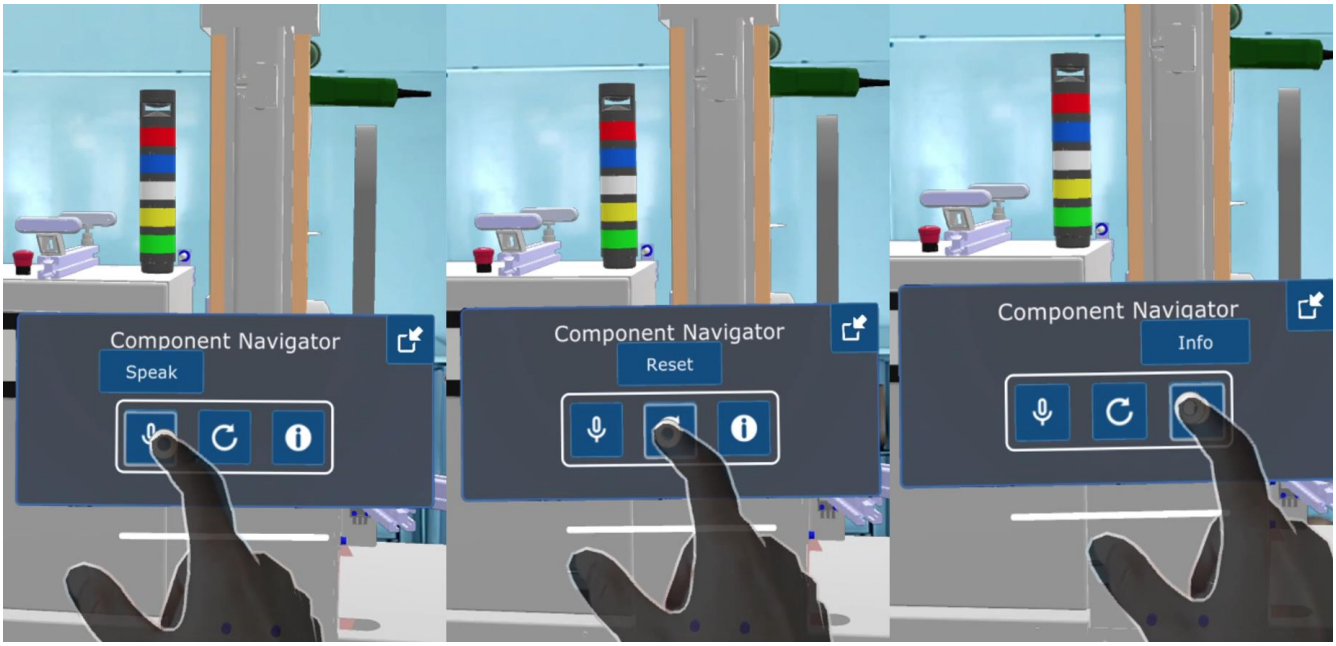


Figure 8 – The Final Component Navigator User Interface Panel with the Button Functions to speak, reset and receive Information

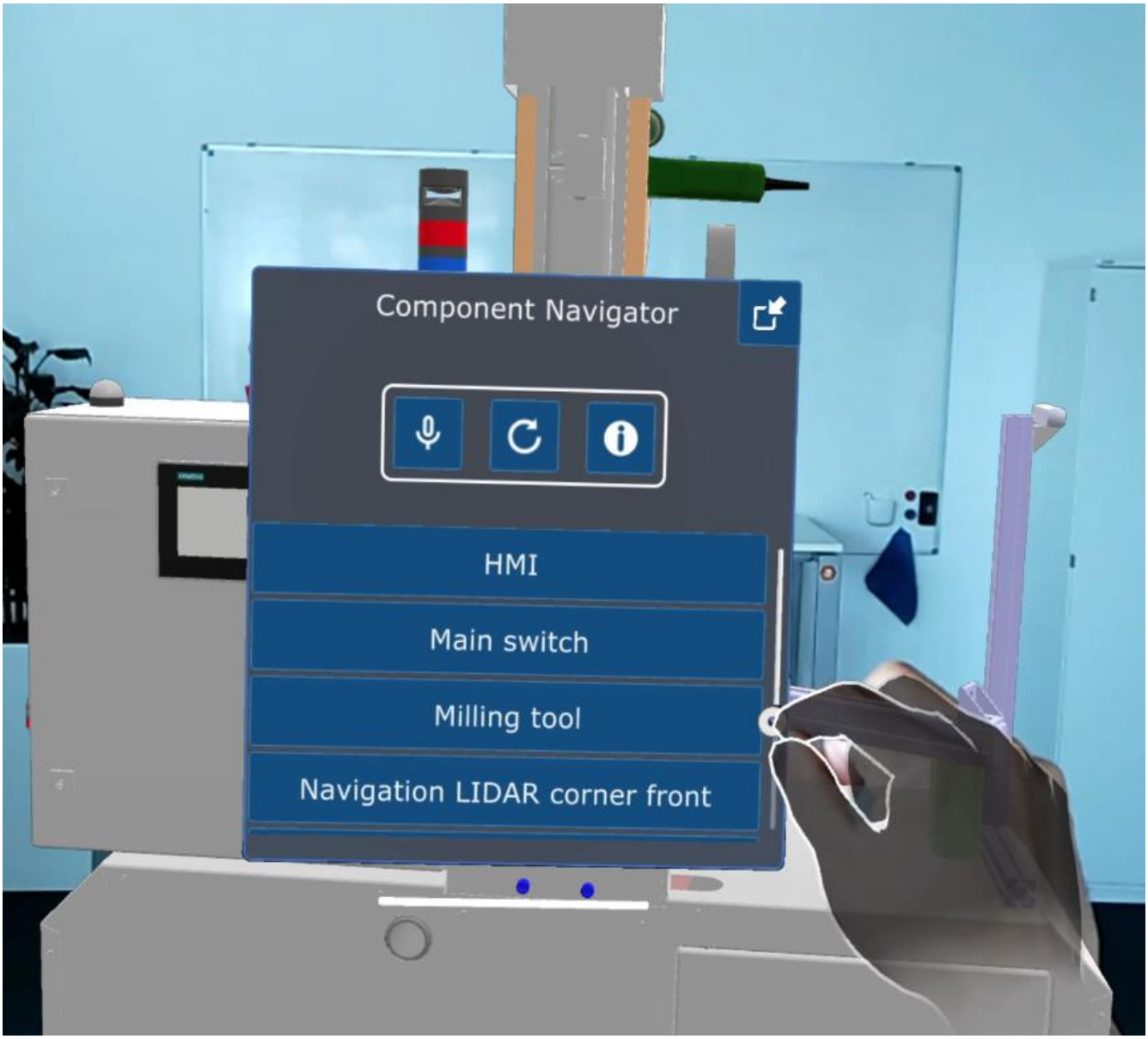


Figure 9 – The button function “Info” further displays the available components to navigate within the 3D model with an option to click and isolate

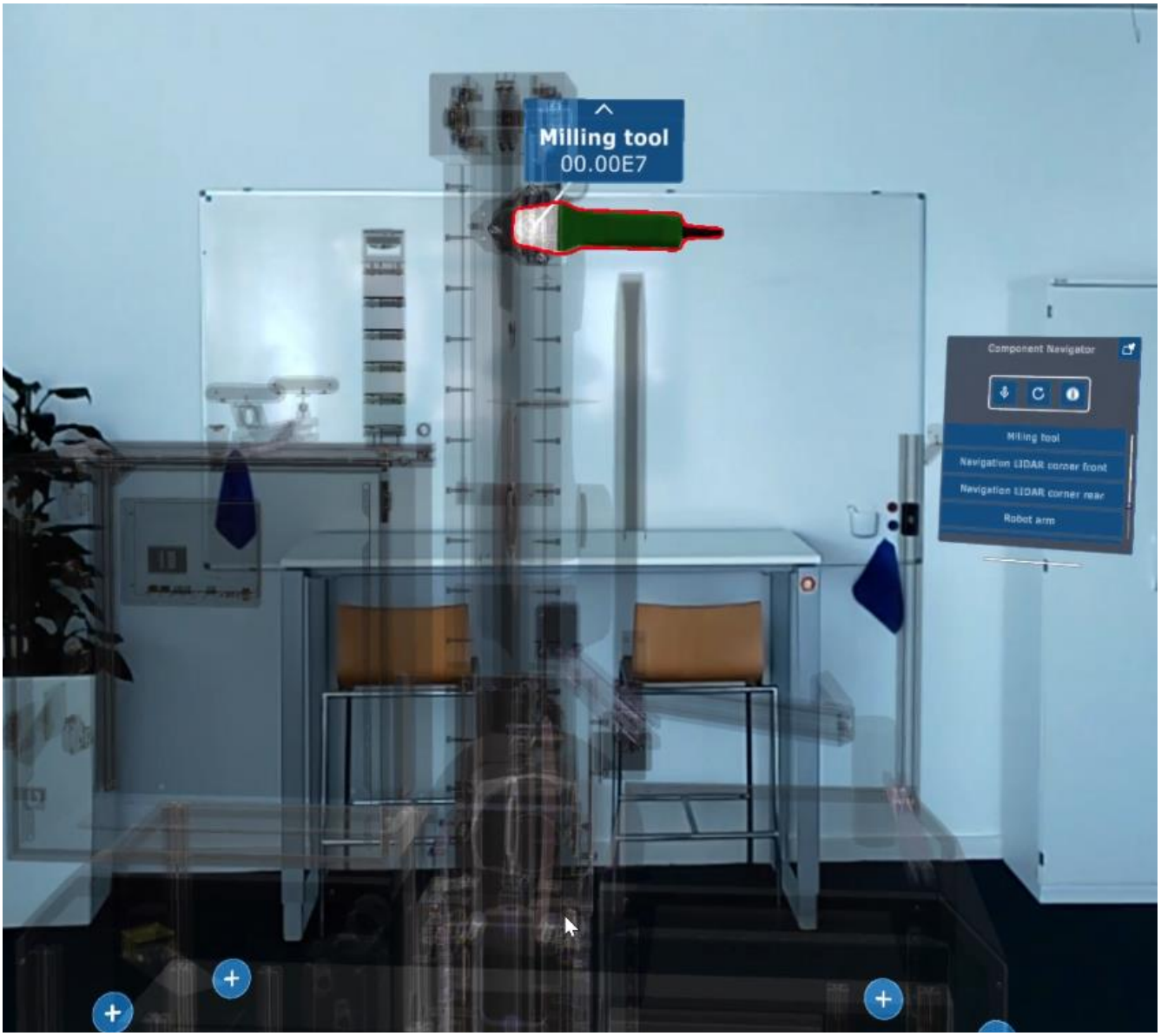


Figure 10 – The component navigator feature enabled wherein the "Milling Tool" is isolated with the rest of the components rendered in X-ray view allowing for easier navigation and component identification

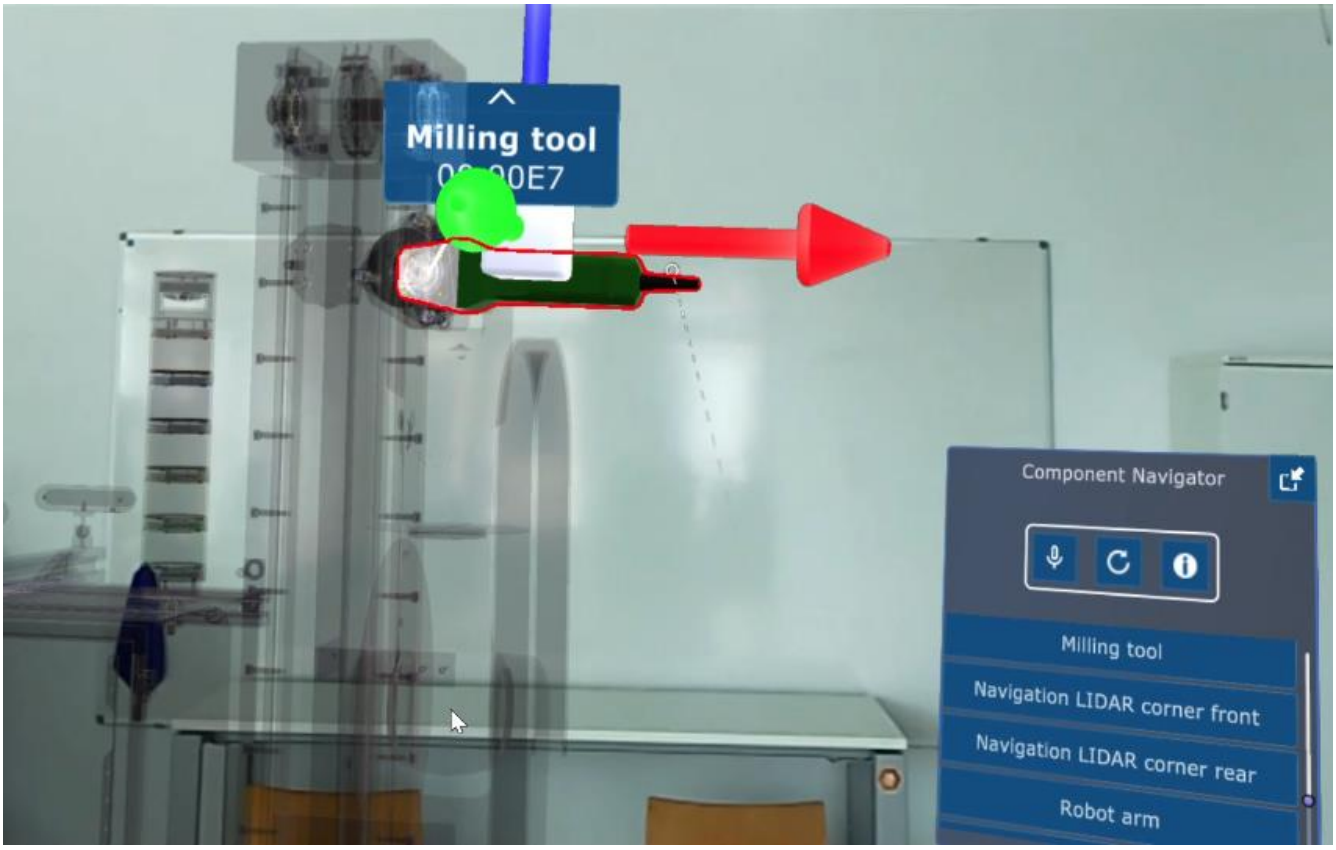


Figure 11 – Once a component is isolated, RoboSpace also allows for easy 3D manipulation of the isolated component - move, rotate, scale

Because voice-driven selection depends on reliable speech recognition, a dedicated investigation addressed the accuracy of the underlying recognition engine: adjusting its confidence threshold, testing alternative speech-to-text approaches, and redesigning problematic phrases (spelling out numbers, avoiding short or easily confused words) to reduce misrecognition of component names.

As a result, RoboSpace now offers a robust, dual-mode (voice and manual) means of isolating and inspecting individual robot components, adaptable across CAD sources and resilient to common speech-recognition failure modes.

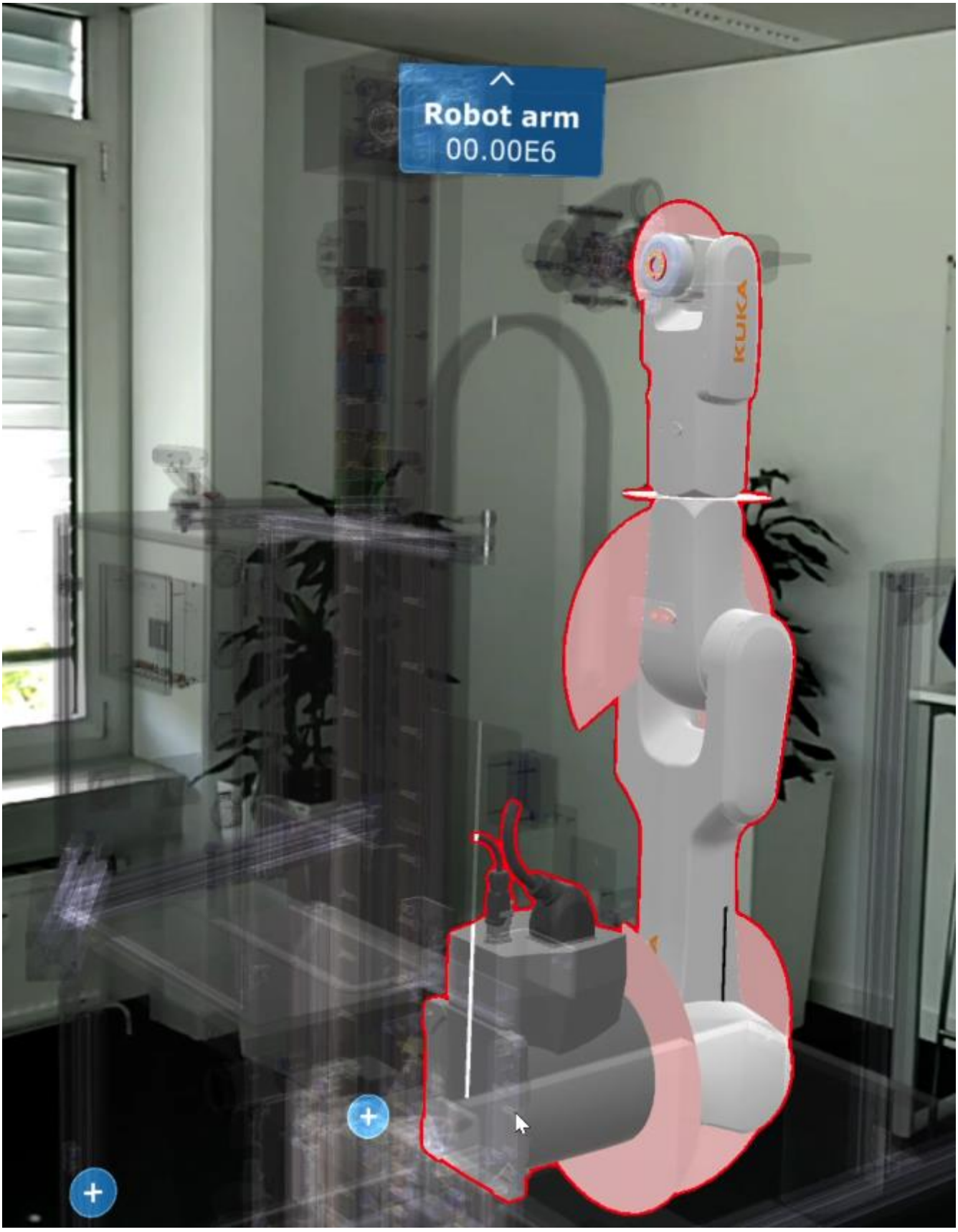


Figure 12 – Example of isolating and identifying the robot arm component of the KUKA robot crawler model

AI-powered Assistance for Design Review

KUKABot is a conversational AI assistant integrated into RoboSpace that allows users to ask natural-language questions and receive spoken and text responses, supported by a knowledge-retrieval service provided by SIE and is described in more detail in subsection 2.2.2.

Work on the user experience focused on making the assistant clear and comfortable to use: the response panel was made to resize dynamically to fit its content (such as returned images), avatars for the assistant and application were updated, and voice-output timing was tuned to stay synchronised with on-screen text. Detached response panels now persist in the scene until explicitly closed by the user rather than disappearing when a new question is asked. Since RoboSpace also includes its own local assistant, logic was implemented so the two assistants can operate side by side without overlapping prompts or spoken responses, with visual cues indicating which assistant is currently active.

On the integration side, the assistant's backend — hosted and developed by SIE— was updated twice to follow significant upgrades to that service: first to a revised API with token-based authentication and persistent conversation sessions, then to a new standardised interface accompanying an upgraded language model with improved reliability and an expanded range of supported question types. A "reasoning" element now returned alongside the assistant's answers was surfaced in a dedicated panel so users can inspect its underlying reasoning, with an associated display issue subsequently corrected.

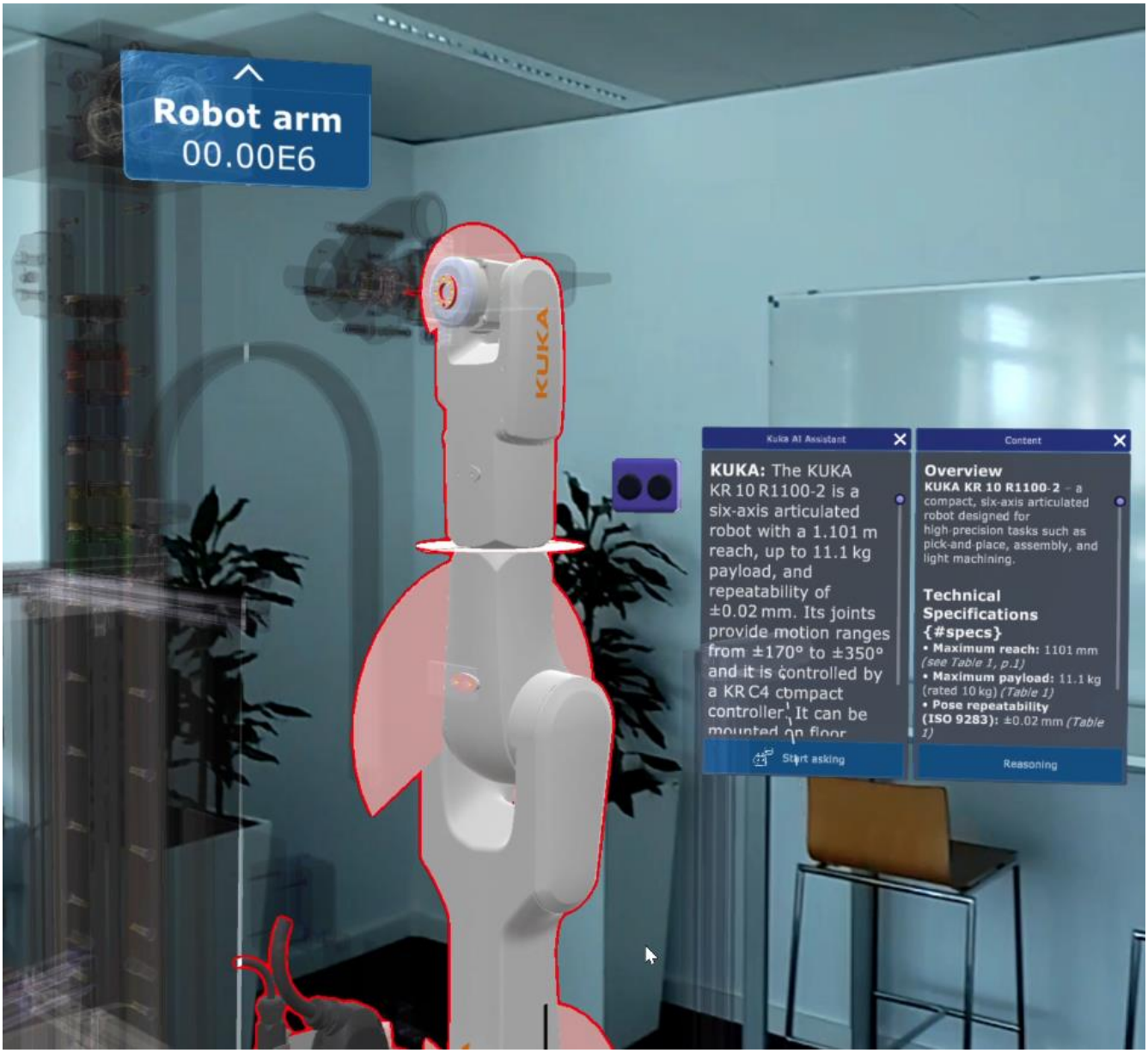


Figure 13 – The KUKABot interface provides assistance to a user's technical query. RoboSpace displays the results of the query in 2D panels

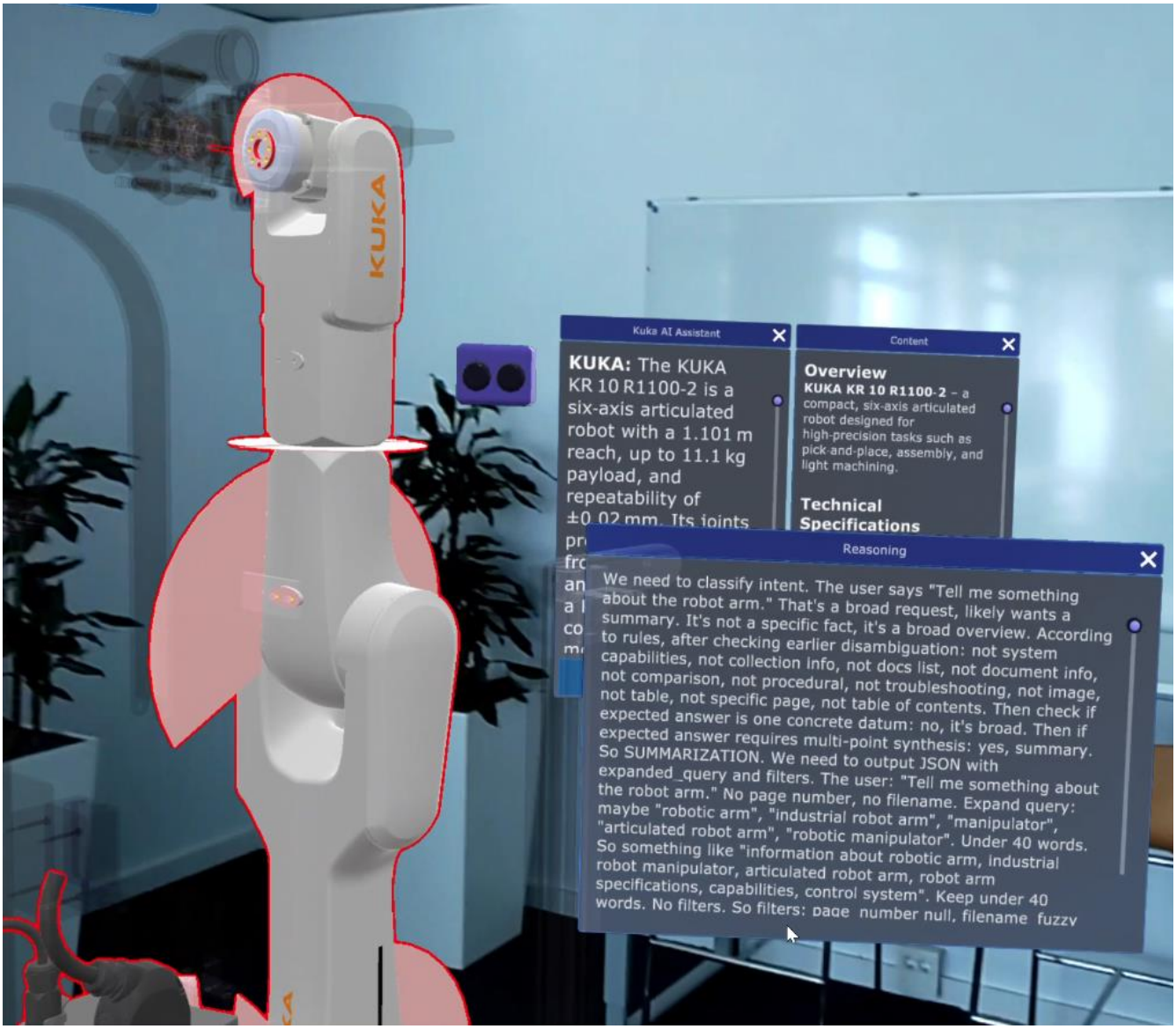


Figure 14 – RoboSpace additionally also provides the option to view the reasoning of the model's response

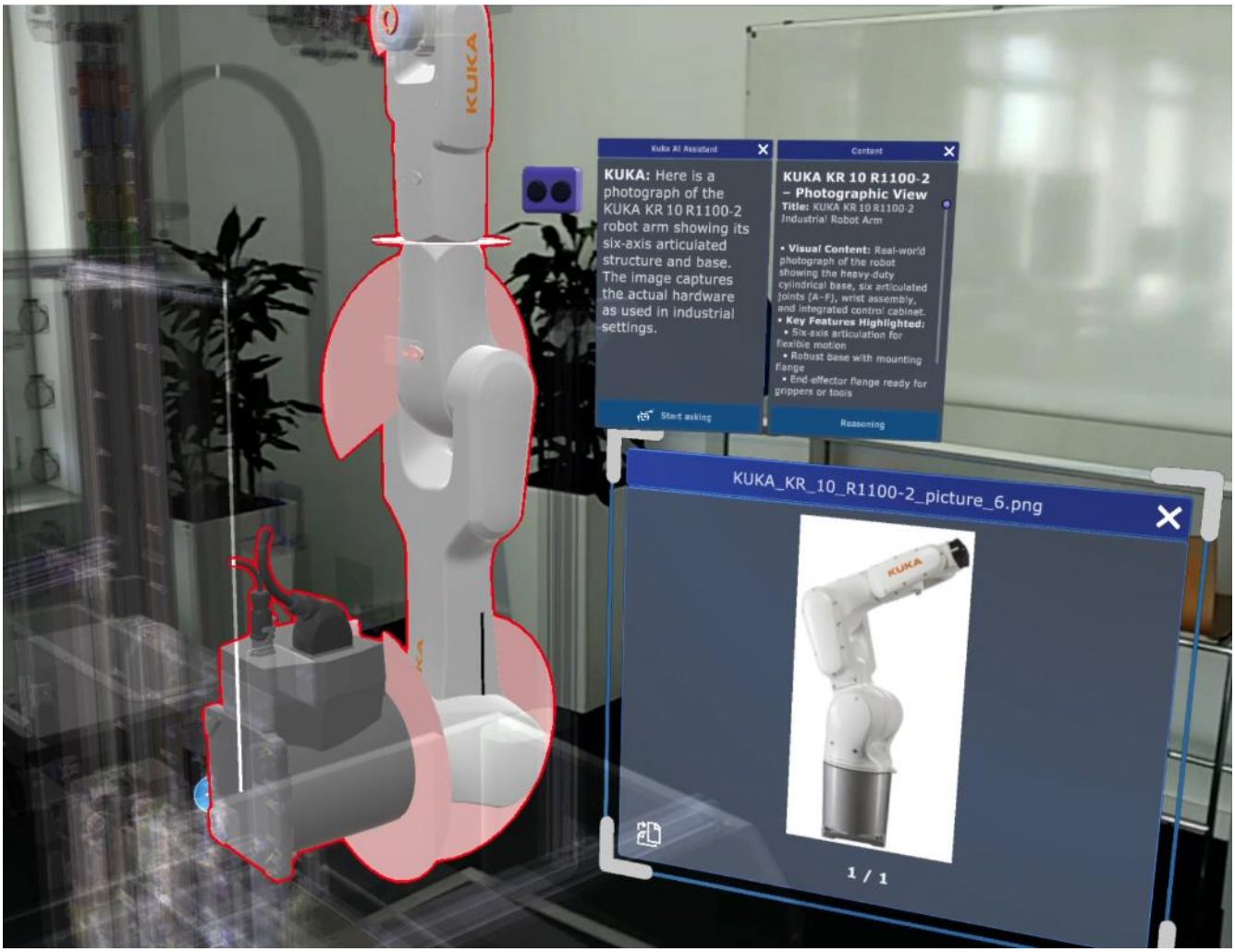


Figure 15 – A realistic use case of KUKABot and Component Navigator in assessing and understanding the robot arm of the robot crawler model

Robot Simulation and Remote Interaction

This feature lets the user visualize and control the motion of the KUKA robot arm within RoboSpace, either by replaying a pre-recorded simulated motion sequence or by connecting to live data from the physical robot via the KUKA CRSM portal and OPC UA connection.

The initial development began with detailed kinematic data provided by KUKA — describing the movement of each of the robot's six axes between default, minimum and maximum positions, with precise timing and angular-velocity parameters — reproduced first as a proof of concept and then integrated natively into RoboSpace with in-app play, pause and stop controls.

The motion data source evolved progressively from static, locally stored datasets toward a live connection to the PLC via OPC UA: a service was introduced to read joint-angle values from the robot's PLC in real time, and the interface was extended with a dedicated control panel letting the user choose between simulated playback and live data, each with appropriate controls and error handling. The panel was later refactored to match updated interface standards, and motion smoothness was further tuned. Most recently, connectivity was upgraded so RoboSpace application connects directly to the live data source over a secure channel, removing the need for a separate intermediary service.



Figure 16 – Robot Arm motion panel with two buttons, one to connect to the live OPC UA data source and the second to connect to a simulated data source



Figure 17 – When the user selects the OPC UA data source, the application loads another panel with options to connect, disconnect and reset. "Connect" makes a successful connection to the robot's PLC via OPC UA and via a VPN



Figure 18 – When the user selects “Connect”, the robot starts reading and consuming the values from the PLC and the user visualises robot movement virtually. Appropriate error handling has been implemented in case a connection to the PLC is not successful

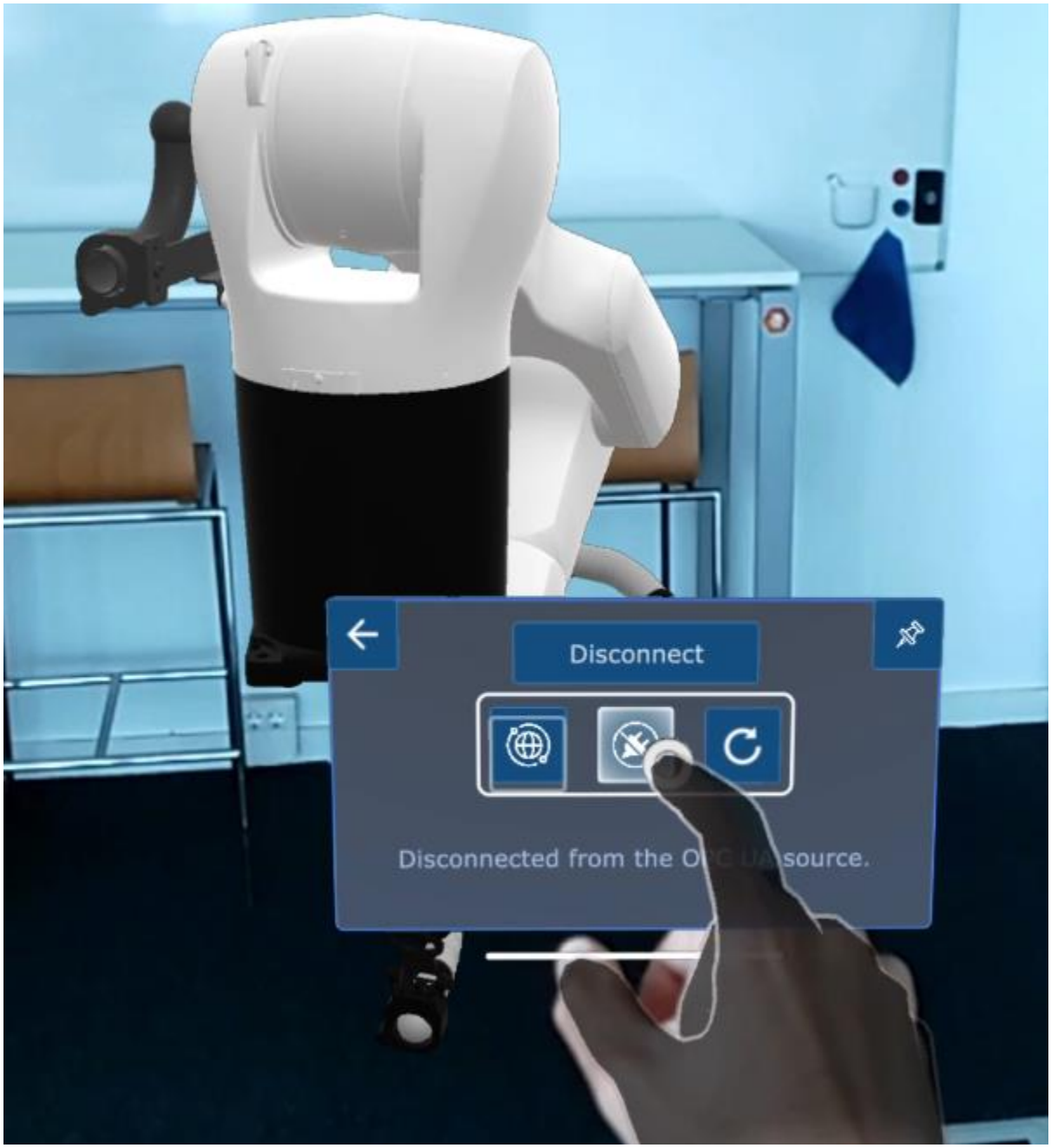


Figure 19 – When the user selects “Disconnect”, the robot movement stops as the application stops reading the live values

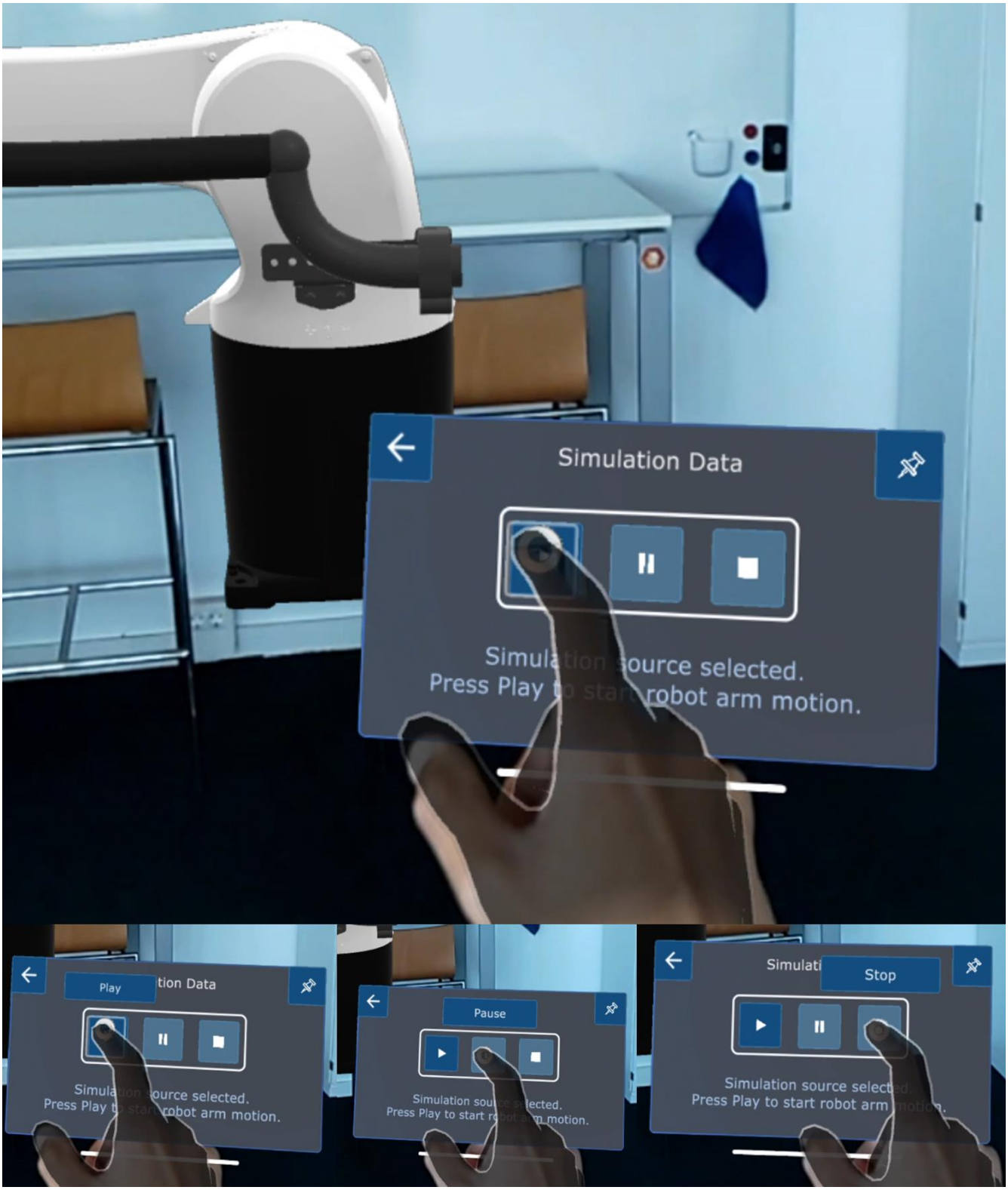


Figure 20 – When the user selects the simulation data source, the application loads preloaded robot kinematic data with options to play, pause and stop

Overall, the robot arm simulation has matured from a static proof of concept using pre-recorded data into a fully interactive, dual-source visualization connected to the robot's real-time sensor infrastructure provided by KUKA.

XR5.0 Training Platform Integration

For the KUKA pilot, the XR5.0 Training Platform is delivered as a single, integrated experience built on Hololight Hub (described in detail in deliverables D5.3 and D5.4), which serves as KUKA's dedicated cloud environment. The platform hosts and streams the pilot's RoboSpace application from the cloud, while giving administrators a centralised web interface to manage users — adding operators, assigning roles and access — and applications, including uploading, versioning and configuring RoboSpace for KUKA's tenant, independently from other partners on the platform.

When an operator starts a session, the platform provisions a cloud virtual machine on demand and streams the running application to the operator's headset (Microsoft HoloLens 2) in real time, so no processing or sensitive data resides on the device itself. To support live interaction with KUKA's physical robot, each application VM also establishes a secure VPN connection into KUKA's network at launch and communicates with the robot controller through a dedicated remote interface, allowing RoboSpace to read live sensor and joint-position data directly from the robot arm rather than relying on pre-recorded data. An application portal capability, added at the infrastructure level, further allows this environment to be reached directly where needed, complementing the standard streamed session with controlled direct access.

Training content itself is authored and delivered through the platform's Training Programs Authoring Tool (TPAT) and received via XR Training Plugin (XRTP) within the RoboSpace application. Training managers use the authoring tool to build structured programs, checklists and step-by-step workflows from existing documentation, images and videos; these are then rendered inside RoboSpace so operators can follow guided training material immersively, in the same session as the live robot connection.

Together, this gives KUKA a single, centrally managed environment: administrators control users and applications, technicians train immersively with real-time visibility into the robot's physical state, and secure connectivity between the cloud platform and KUKA's on-site systems is handled transparently underneath.

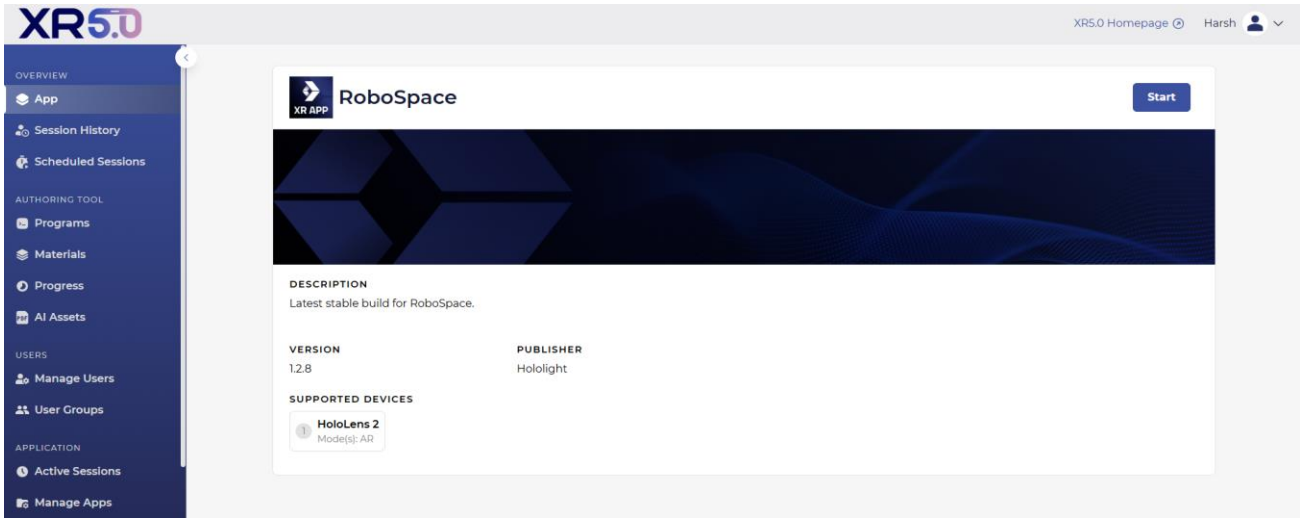


Figure 21 – Pilot 1 RoboSpace XR application available on the XR5.0 Training Platform within KUKA's tenancy



Figure 22 – The TPAT interface enables KUKA admins to create training programs for their technicians and operators

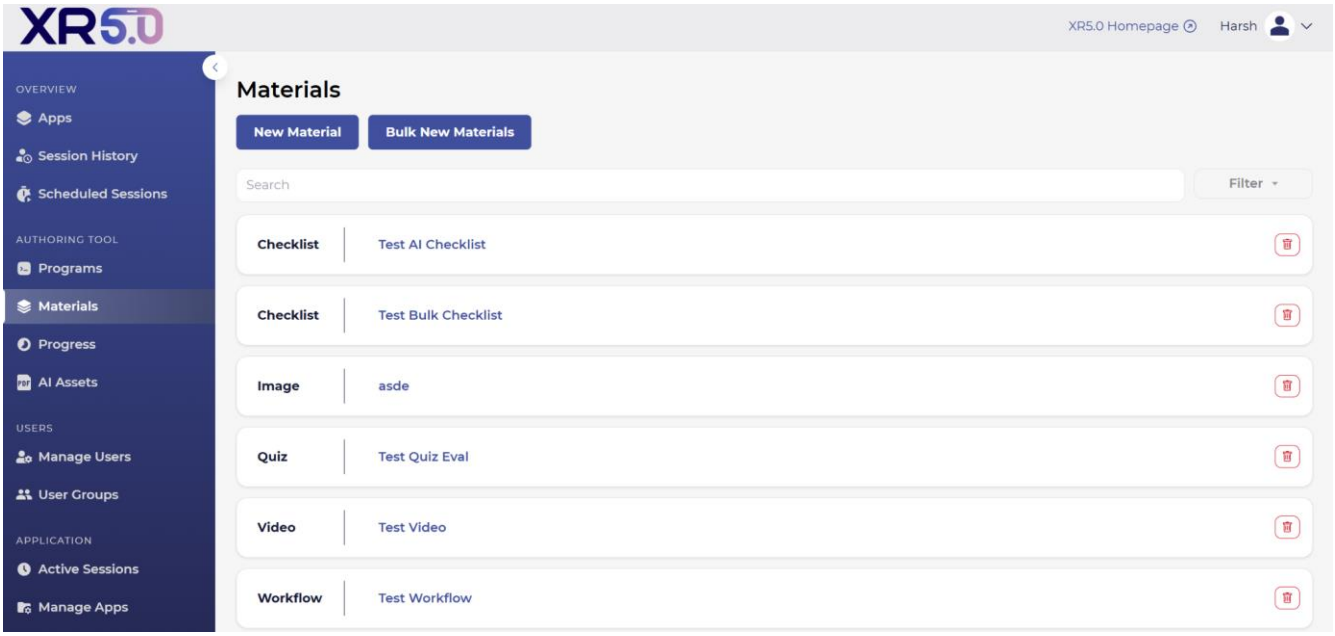


Figure 23 – The TPAT interface enables users to create different models of training materials such as Checklist, Image, Quiz, Video and Workflow. These materials can be added to several different training programs

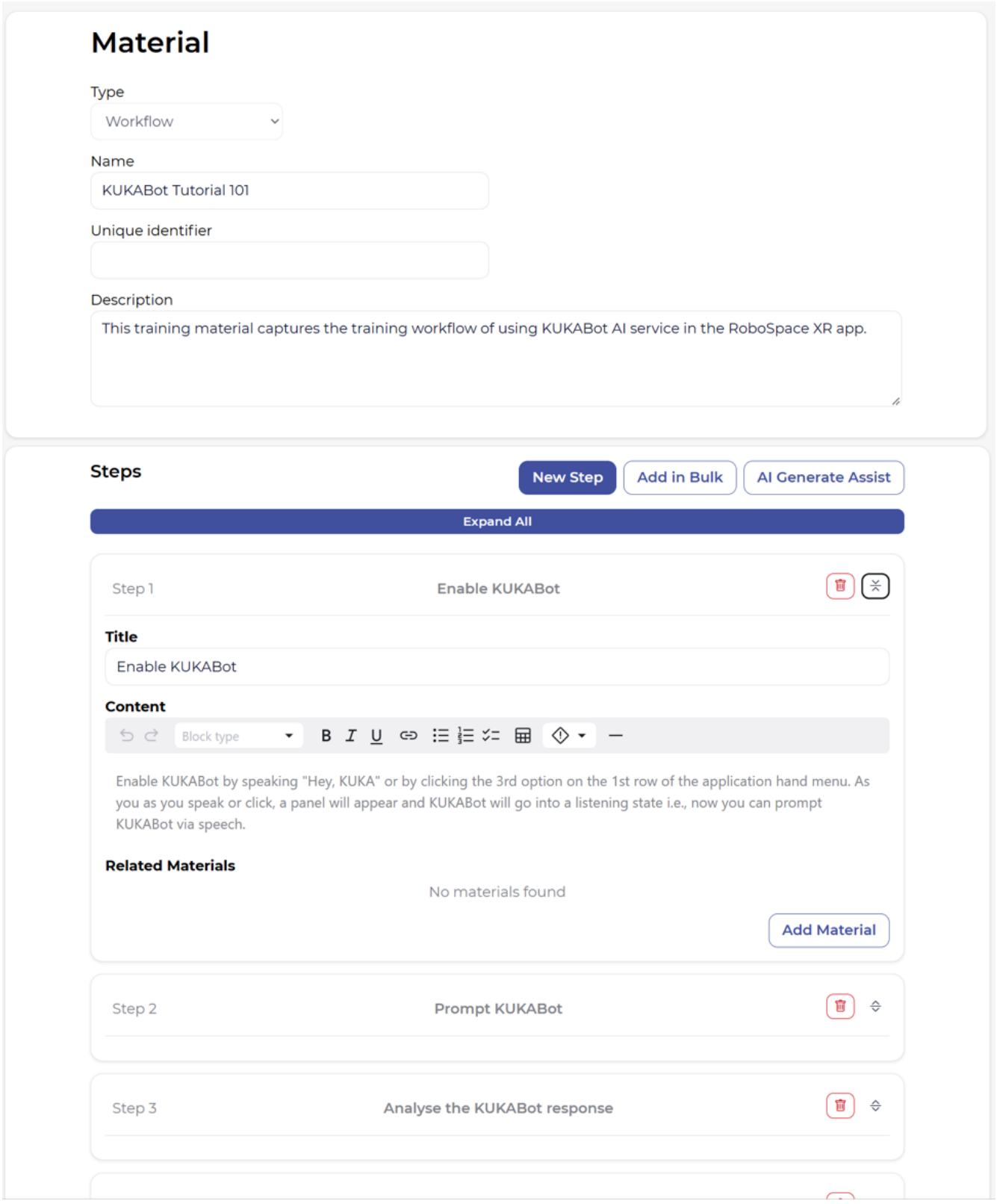


Figure 24 – An Example of a Training Material for KUKA

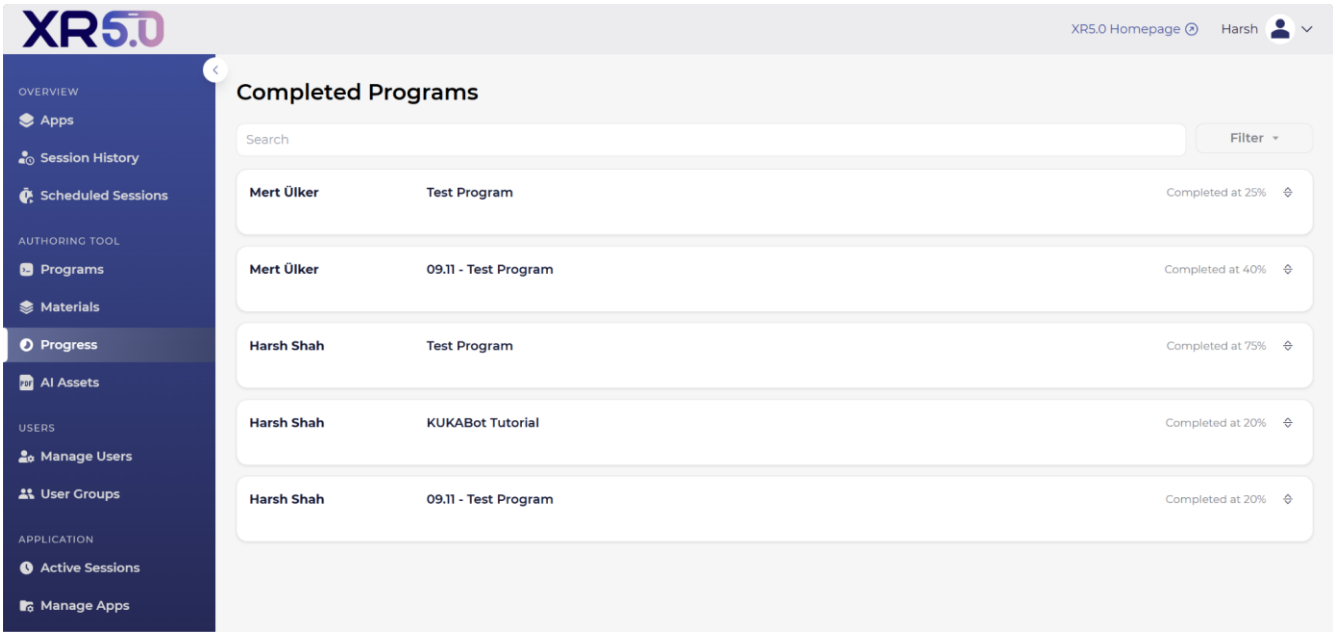


Figure 25 – The TPAT interface allows admins to view the training program progress status of their users

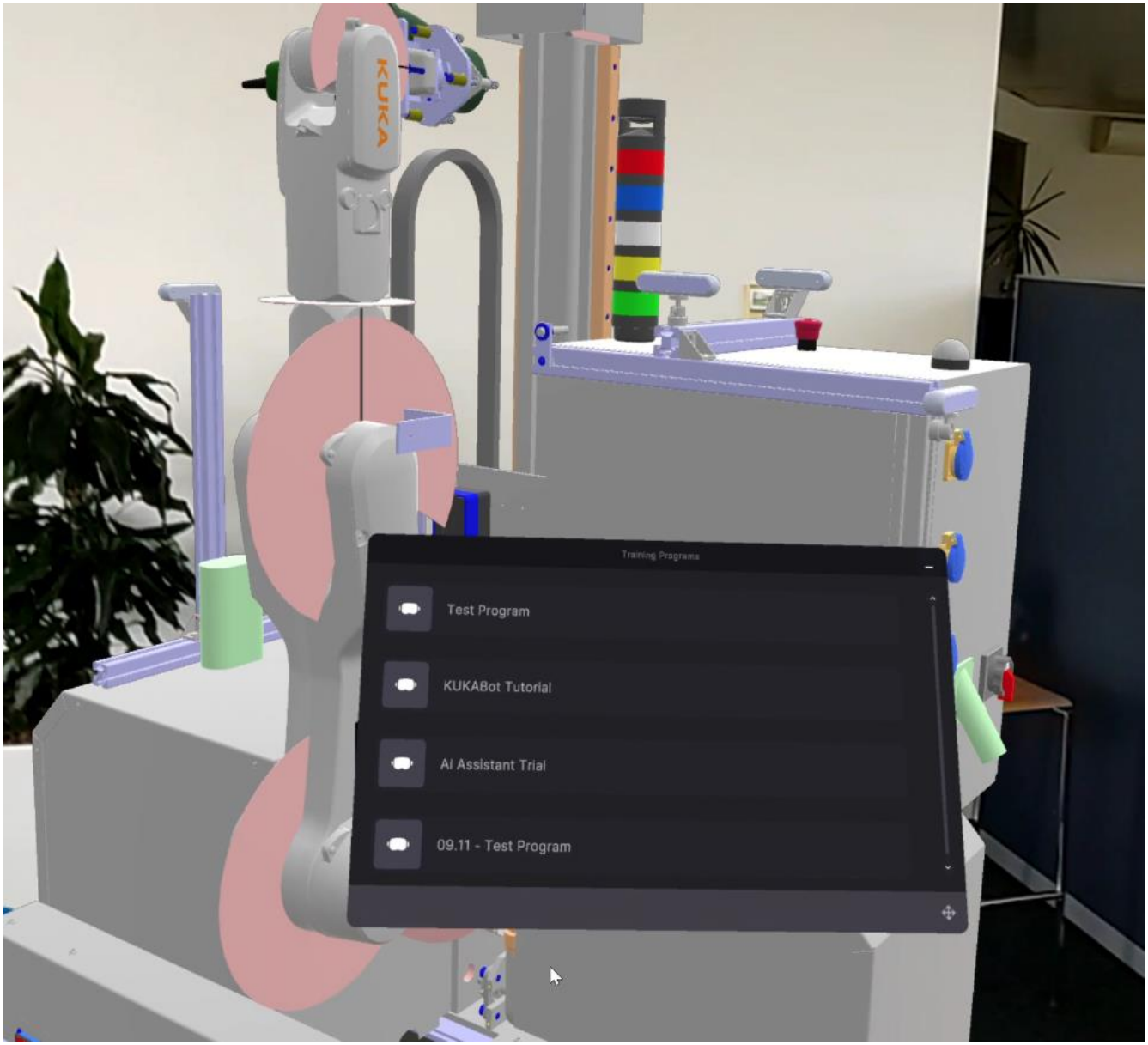


Figure 26 – The training programs available within the RoboSpace app via XRTP. These programs were created by the admin and assigned to the user by their role

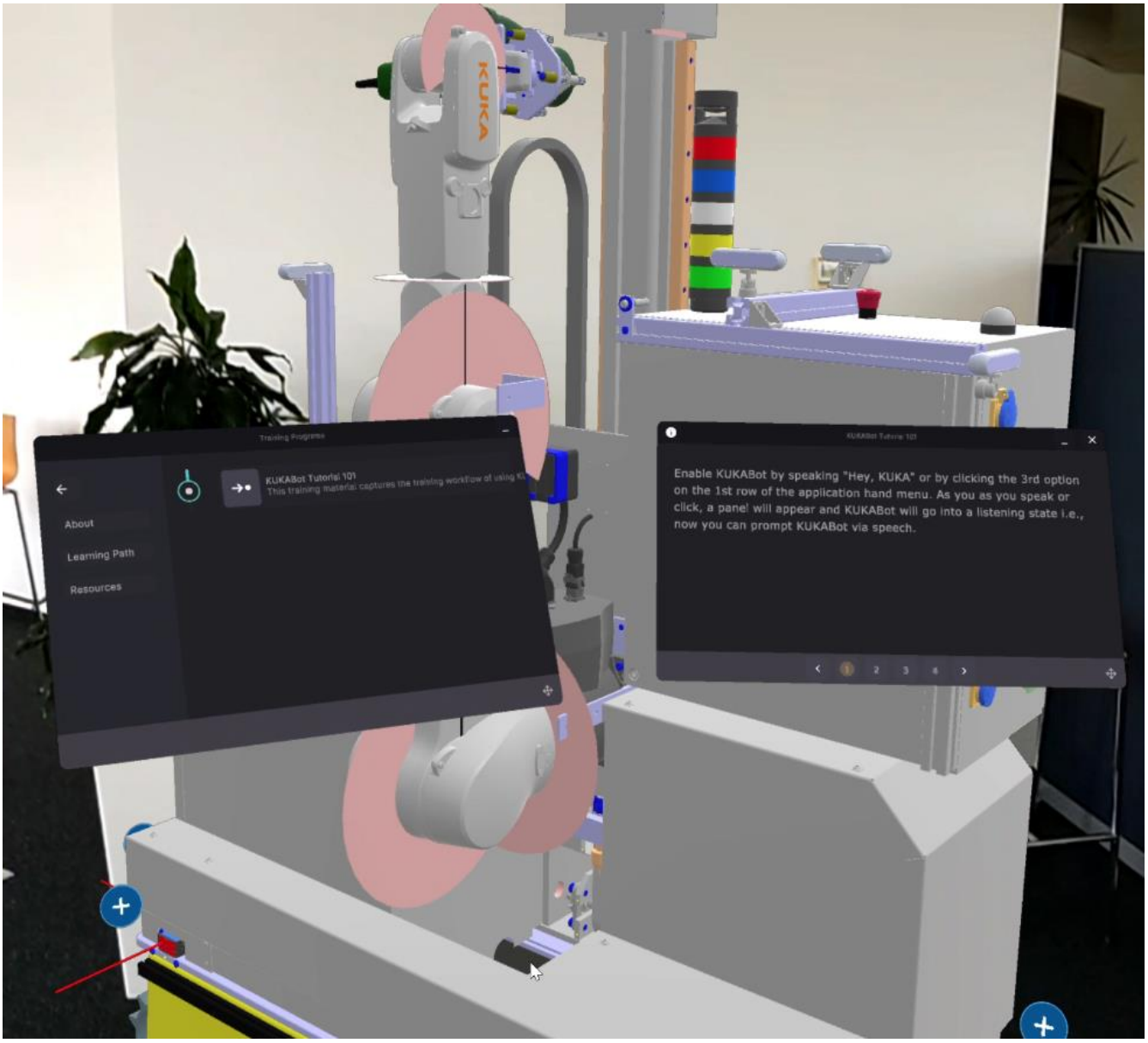


Figure 27 – Interactive panels of the XRTP within RoboSpace allowing users to perform training easily

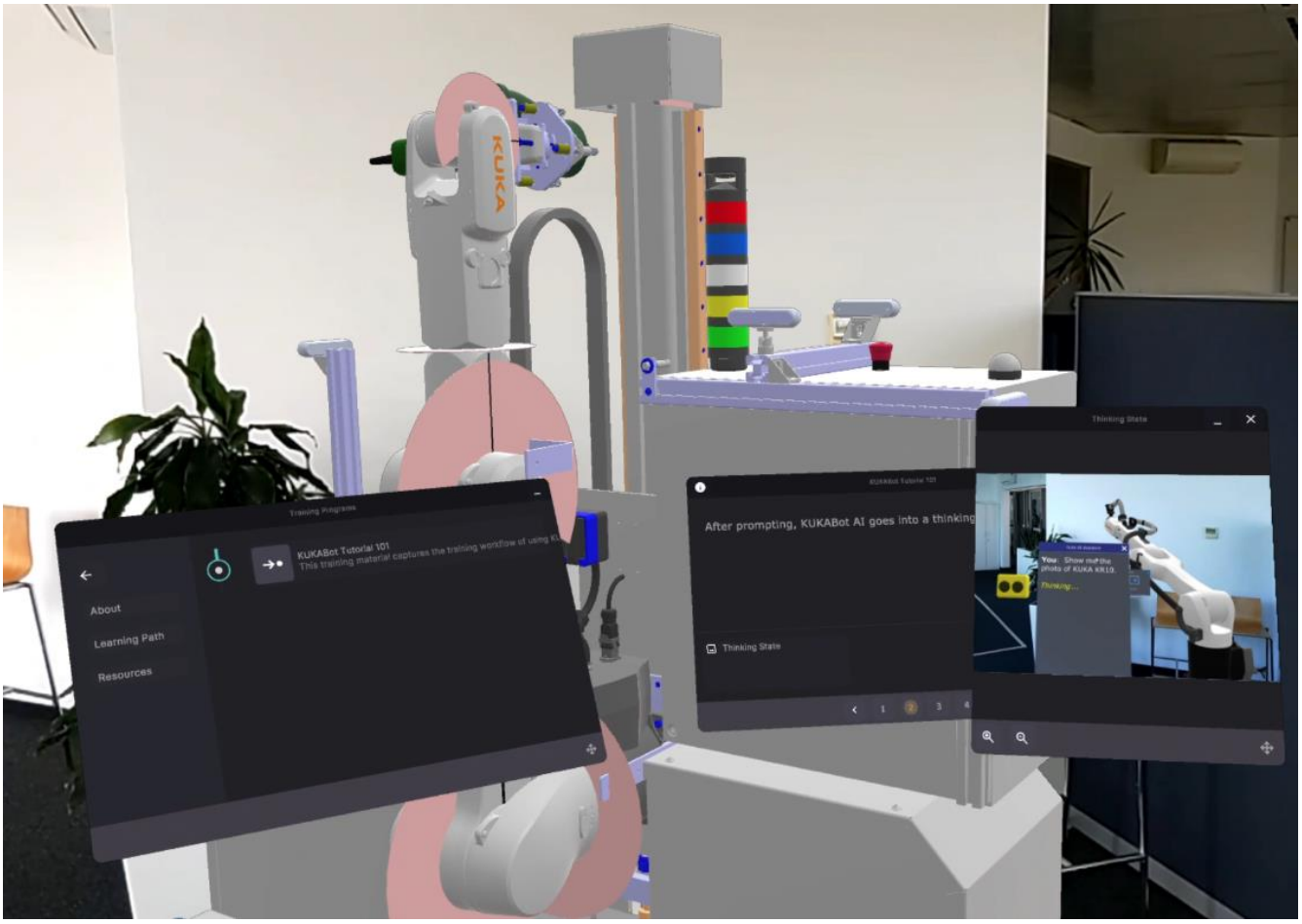


Figure 28 – Admins can add images to their training programs and users can view them directly on the XR headset

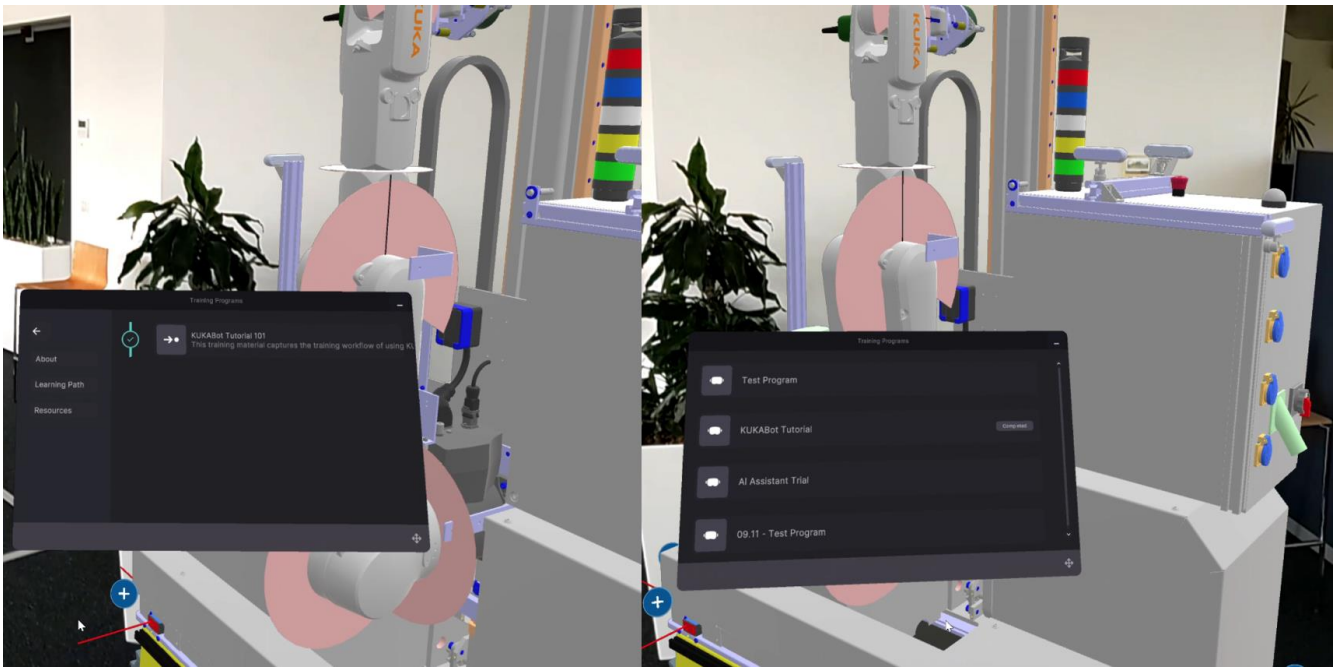


Figure 29 – XRTP allows users to view the status and completion of their training programs

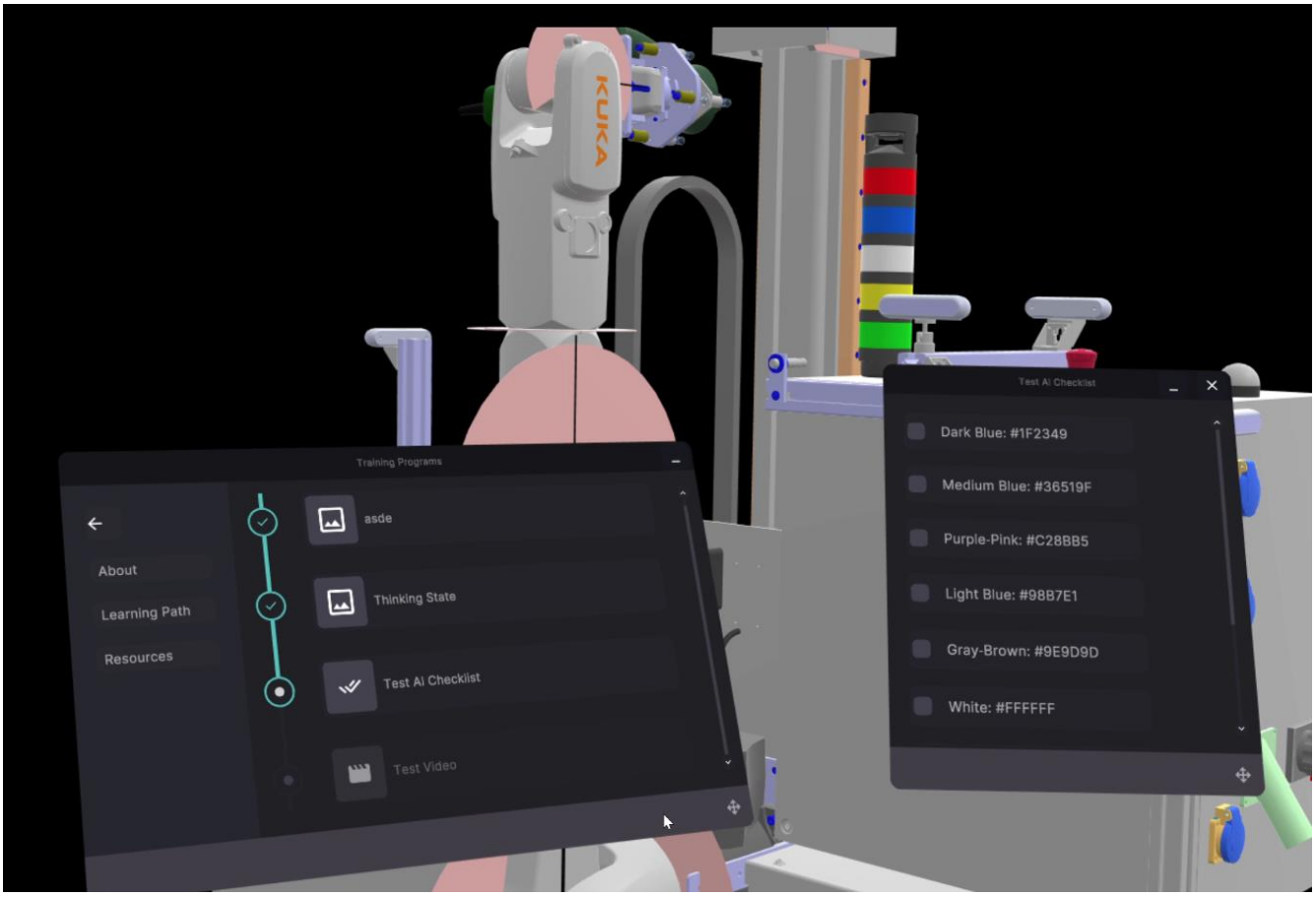


Figure 30 – An Example of a Checklist Training Material

2.2.2 XR-enabled AI Assistant

The KUKA Pilot's XR-enabled AI Assistant use case is enabled by KUKABot, a conversational AI assistant developed by Siemens and integrated by HOLO into the RoboSpace XR application. KUKABot gives technicians on-demand, spoken access to KUKA technical documentation without requiring them to leave the immersive environment.

KUKABot is implemented as a deployment of the DataLens Conversational AI Framework, a privacy-first, on-premise retrieval-augmented generation platform contributed by Siemens and described in detail in deliverable D4.2. For the KUKA Pilot, the framework operates with a dedicated knowledge base containing KUKA technical manuals, equipment datasheets, and electrical design schematics. This separation allows the pilot to benefit from the DataLens platform while ensuring that KUKA's documentation remains confined to its own storage and retrieval namespace.

To provide a secure and privacy-preserving solution, the framework was built exclusively on open-source components:

- Ollama for language and embedding models,
- Docling for document conversion,
- MinIO for object storage,
- ChromaDB for the vector knowledge base.

This architectural choice ensures that document ingestion, retrieval, and inference are performed entirely on KUKA-controlled hardware, without exposing document content to external services or third-party infrastructure. It therefore addresses a central concern of the pilot: ensuring that proprietary engineering documentation and the data associated with its users remain under organisational control, in line with the

principles of the GDPR and the EU AI Act. The high level depiction of the DataLens architecture is presented in Figure 31.

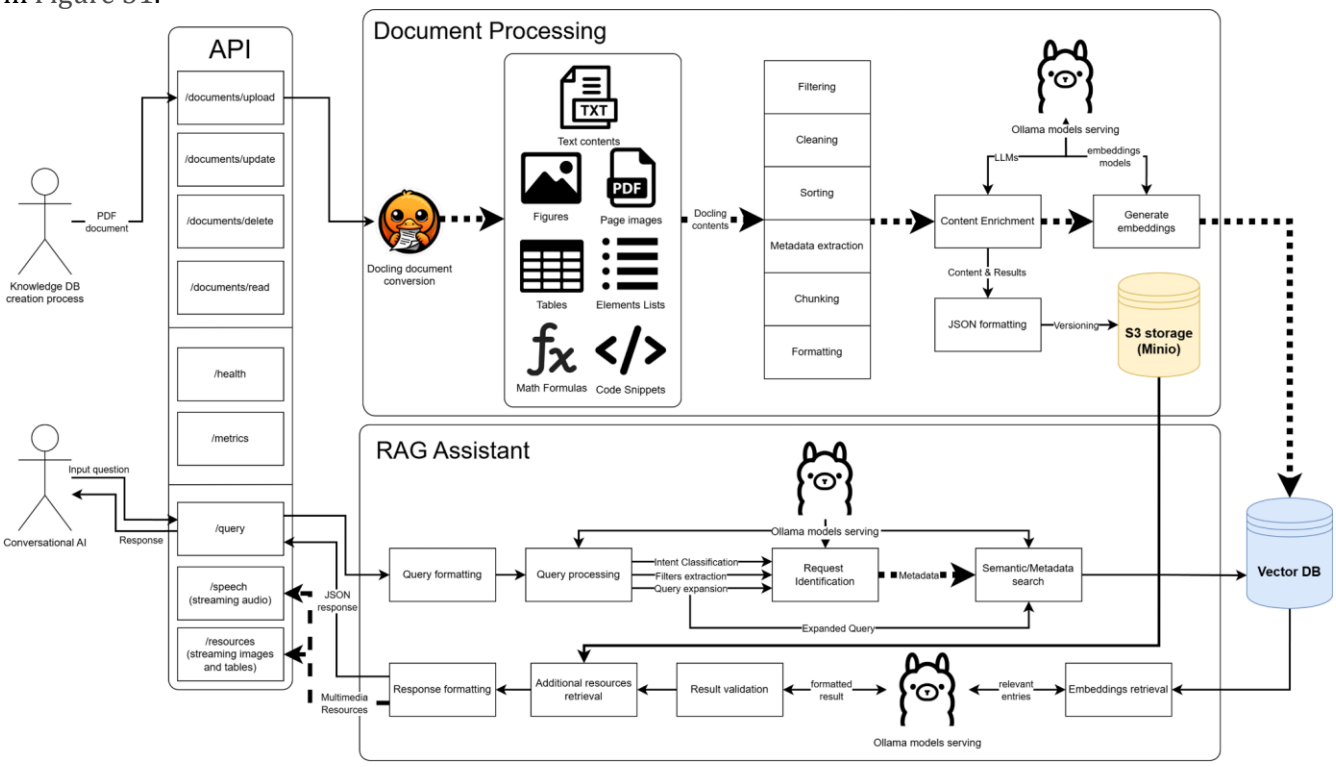


Figure 31 – DataLens High-level Architecture

Within the RoboSpace XR application, KUKABot is consumed as a service through a versioned REST interface. The application submits the technician's question as text or recorded speech and receives a structured response containing the elements required for presentation in the immersive environment. Depending on the request and the retrieved documentation, the response may include a spoken answer, formatted text, figures, tables, document pages, and source references. Media content is retrieved separately through dedicated endpoints, allowing the XR application to request only the assets required for the current interaction. The main endpoints used by the Pilot 1 integration are summarised in Table 1, while the complete API specification is provided in D4.2.

Table 1 – KUKABot API Interfaces used within Pilot #1

Method	*Endpoint*	Purpose
POST	*/api/v1/collections/{collection}/inferences*	Submit a question as text or recorded speech, returns the structured reply.
GET	*/api/v1/collections/{collection}/audio/{speech_id}*	Retrieve the synthesised spoken answer referenced by the reply.
GET	*/api/v1/collections/{collection}/content/{file_name}*	Retrieve a figure, table or document page referenced by the reply.
POST	*/api/v1/collections/{collection}/documents*	Add a document to the knowledge base,

		while processing continues in the background.
GET	*/api/v1/collections/{collection}/jobs/{job_id}*	Follow the progress of a document being processed.
GET	*/health*	Report a service and dependency status.

Response Generation

When a technician submits a question, KUKABot accepts it either as text or as recorded speech. Speech input is transcribed before entering the conversational pipeline. Rather than forwarding the question directly to a single language model, KUKABot processes it through several stages to determine the conversational context, the user's intent, and the most relevant information sources.

First, the assistant determines whether the technician is asking a new question, following up on a previous answer, accepting a suggestion, or engaging in general conversation. This allows follow-up questions such as "What is its maximum reach?" to be interpreted in the context of the preceding exchange rather than as isolated queries. The question is then reformulated to improve retrieval quality. During this step, abbreviations and related engineering terminology may be expanded, while explicit references to pages, figures, tables, or specific documents are identified.

KUKABot subsequently classifies the request according to the type of information required. Most questions are answered using document content retrieved through semantic similarity against the KUKA knowledge base. These questions may concern factual information, equipment or document overviews, comparisons between models or configurations, step-by-step operating procedures, fault diagnosis, or specific figures, tables, or document pages.

Other questions concern the knowledge base itself, for example which documents are available, what topics the collection covers, or which functions the assistant supports. These requests are answered directly using catalogue and collection metadata rather than through document retrieval. If the requested information is not available in the loaded documentation, KUKABot indicates this explicitly instead of generating an unsupported answer.

Once the question has been processed and the relevant information retrieved, KUKABot returns a JSON structured response designed for use in the XR environment. A response illustration of the Service is presented in Code 1 and can include a spoken answer optimised for speech synthesis, a link to the corresponding audio file, and a formatted text answer for display on an XR panel. Where relevant, it also contains links to figures, tables, or document pages associated with the answer, together with source references identifying the document, content type, and page from which the information was retrieved.

```
{
  "session_id": "uuid of the current session",
  "query": "User's input query text",
  "collection_name": "default",
  "response": {
       "speech": {
       "text": "Voice-optimised plain text response for TTS synthesis",
       "link": "https://datalens.xr50.work/api/v1/collections/default/audio/{speech_id}"
       },
       "markdown": "# Section\n**Key value**: 42 Nm\n\n- Bullet 1\n- Bullet 2",
       "images": [
      "https://kukabot.xr50.work/api/v1/collections/default/content/pump_diagram.png"
       ]
  },
  "reasoning": "## Conversation Routing: ...\n## Processing user query: ...\n## Generating
final response: ...",
  "sources": [
       "Operating Manual, ContentType: text, Page 24, Path: .../contents/...",
       "Datasheet, ContentType: table, Pages 8 - 9, Path: .../contents/..."
  ]
}
```

Code 1 – DataLens RAG Pipeline Response Schema

A conversation identifier links successive exchanges and enables follow-up questions to be interpreted in their previous context. Audio files and images are referenced through dedicated links rather than embedded directly in the response. This allows the XR application to retrieve only the assets required for the current interaction, keeping the response compact and reducing unnecessary data transfer.

The inclusion of source references makes the assistant's responses auditable, since individual claims can be traced back to the relevant technical documentation. This is particularly important in an engineering context, where the technician may need to verify the information before acting on it. It also supports the transparency and human-oversight principles relevant to the use of AI-based assistance in safety-relevant environments.

Authentication and Access

To secure communication between the RoboSpace XR application and KUKABot, requests carry an encrypted XR5.0 Hub session token issued when the technician signs in to the Hololight Hub. KUKABot validates the token with the Hub and obtains the identity associated with the request, including the organisation, application, and user. This allows each request to be attributed to an authenticated user and to inherit the session lifetime and central session control provided by the Hub. The approach replaces the

static, collection-specific credentials used previously, which did not identify the individual user or provide an expiry mechanism. This authentication workflow is illustrated within Figure 32.

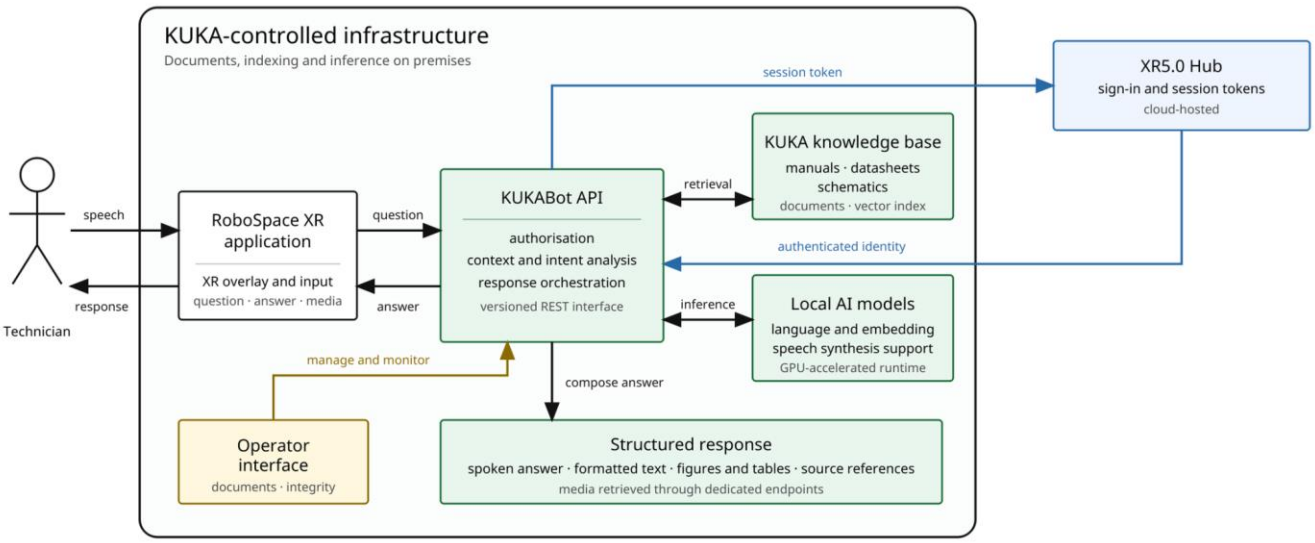


Figure 32 – High-level Overview of the Authentication Integration with the XR5.0 Hub

Authentication establishes the identity of the requester, while authorisation remains managed by the KUKABot deployment. Each deployment is configured for the organisation it serves and can optionally be restricted to specific applications or users. Requests that do not meet these requirements are rejected before the documentation is accessed, preventing unintended access between pilots or organisations. Administrative operations, such as creating knowledge bases, ingesting documents, repairing knowledge-base integrity, and accessing usage reports, continue to require separate operator credentials.

Because the Hub is a cloud-hosted service, network connectivity is required for token validation. However, all documentation, document processing, retrieval, and inference remain on KUKA-controlled infrastructure. No documents, questions, or generated answers are sent to the Hub; only the session token and the identity information returned during validation cross this boundary.

Operator Interface

The KUKABot knowledge-base administrator interacts with the system through a web application deployed alongside the assistant. Figure 33 provides an overview of the operator interface, which offers centralised access to the following functions:

- Knowledge-base configuration: configure the knowledge base and the assistant's conversational behaviour.
- Document management: add, update, and manage documents.
- Processing monitoring: monitor document-processing jobs and review their status.
- System health monitoring: check the health of KUKABot and its dependencies.
- Interactive testing: submit test questions to verify that the expected content is retrieved and that the assistant behaves as intended.
- Knowledge-base integrity: check and repair the consistency between stored documents and their corresponding index entries, particularly after bulk ingestion or migration.

Administrative actions are protected through the XR5.0 Hub authentication mechanism. When an operator signs in, the Hub provides an authenticated session token that is validated by KUKABot before access to the

operator interface is granted. The Hub establishes the operator's identity, while KUKABot applies the authorisation rules that determine which administrative functions are available. Where required, the interface can be operated in read-only mode, preventing modifying actions while still allowing monitoring and verification activities.

KUKABot also records usage information to support the evaluation of the pilot. The operator can review usage frequency, response times, the distribution of question types, the documents consulted, and the outcomes of document processing. In line with the privacy-preserving design of the component, the text of questions is not retained by default; usage records contain only the question length and an irreversible identifier, unless explicit configuration enables the retention of the complete question text.

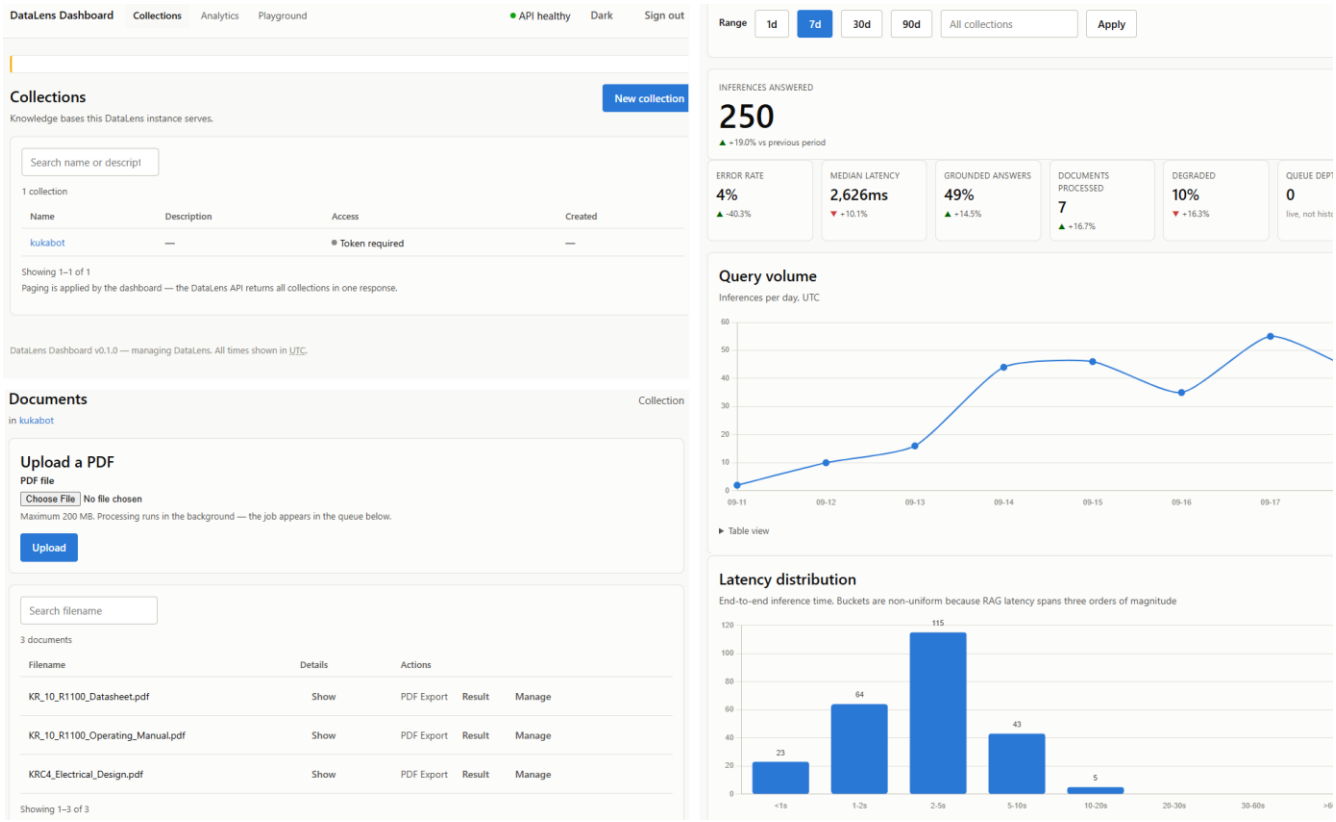


Figure 33 – Operator Interface Depiction showing Collections, Documents and Analytics

Deployment Architecture

KUKABot is deployed as a containerised stack defined as a single topology. It comprises the conversational AI service, the document object store, the language and vision model runtime, the initialisation procedures required to prepare storage and obtain the models, and the browser-based operator interface.

Deployment consists primarily of supplying the required configuration and starting the stack, as shown in Figure 34. Every dependency comes bundled within this stack, including the runtime environment, object store and model server. Once the stack is running, GPU acceleration is passed through to the model runtime, and long-running services restart automatically after a failure or host restart.

The assistant also waits for its dependencies to become available before accepting requests. During the initial start-up, the required models are downloaded automatically, which may take several minutes; subsequent restarts reuse the downloaded models. This mechanism supports a replicable and portable

on-premise installation at the pilot site without requiring separate manual configuration of the individual services.

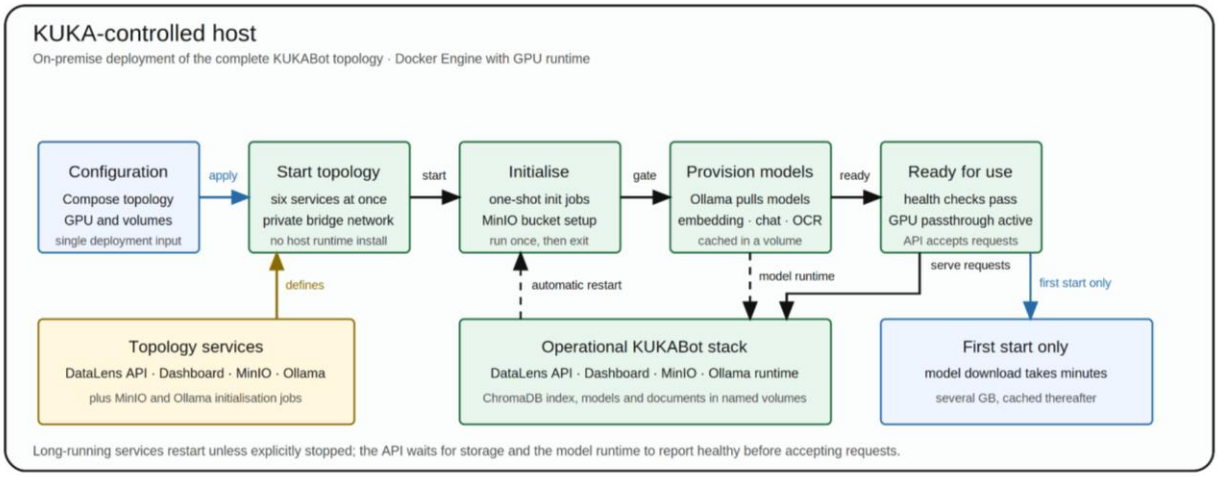


Figure 34 – Workflow Depiction for the Deployment Process of the DataLens Service

2.2.3 XR-based Training Program

The Pilot #1 Final Prototype incorporates an immersive training programme for UC 1.2, *Virtual Commissioning and Generative AI in Robotics*, which offers an interactive, step-by-step tutorial on operating the KUKA SmartPad-2 as shown in Figure 35 and Figure 36. Accessed through the XR Training Plugin (XRTP) developed by IML, the programme guides commissioning technicians through the operation of the KUKA SmartPad-2, the handheld device used to control and programme KUKA robots. Interactive XR panels introduce the device's main controls and provide step-by-step guidance on operating modes, system status indicators, user groups and commissioning workflows. Technicians can therefore practise these tasks in a virtual environment before using the physical equipment.

The XRTP supports the creation and adaptation of training programmes from existing materials, allowing their content and structure to reflect pilot or organisational needs. Its integration in the Final Prototype enables Pilot #1 to deliver contextual training alongside the virtual commissioning application. The training programme and its interactive panels are described in detail in D6.3, while D5.1, *XR Training Materials, Demonstrators, and Programs*, provides further information on the XRTP.

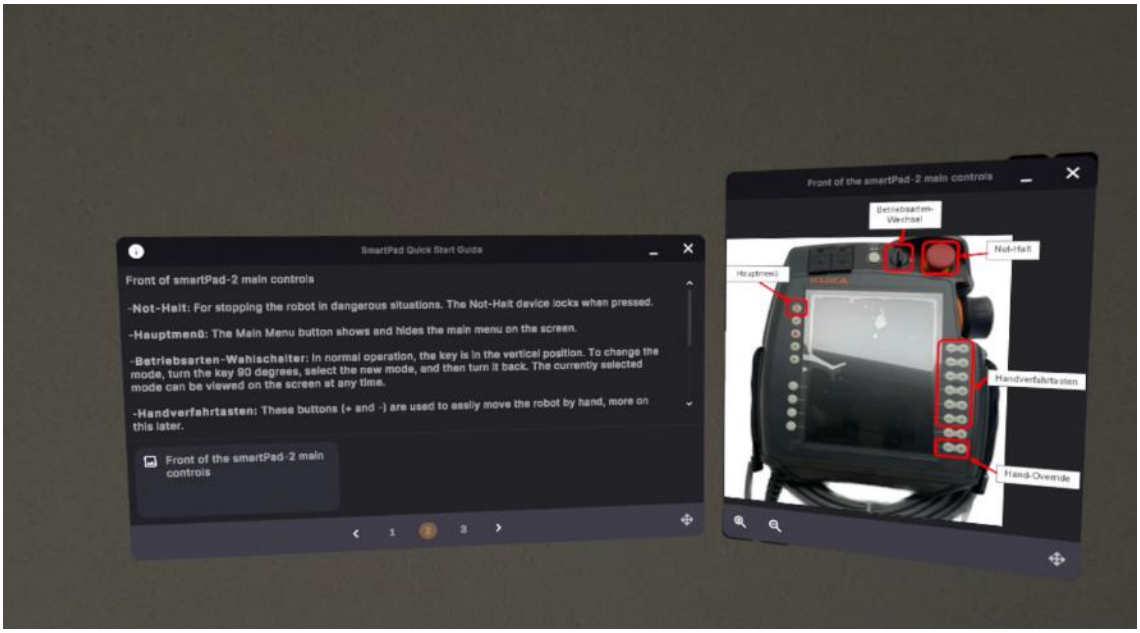


Figure 35 – Training Program in Pilot #1

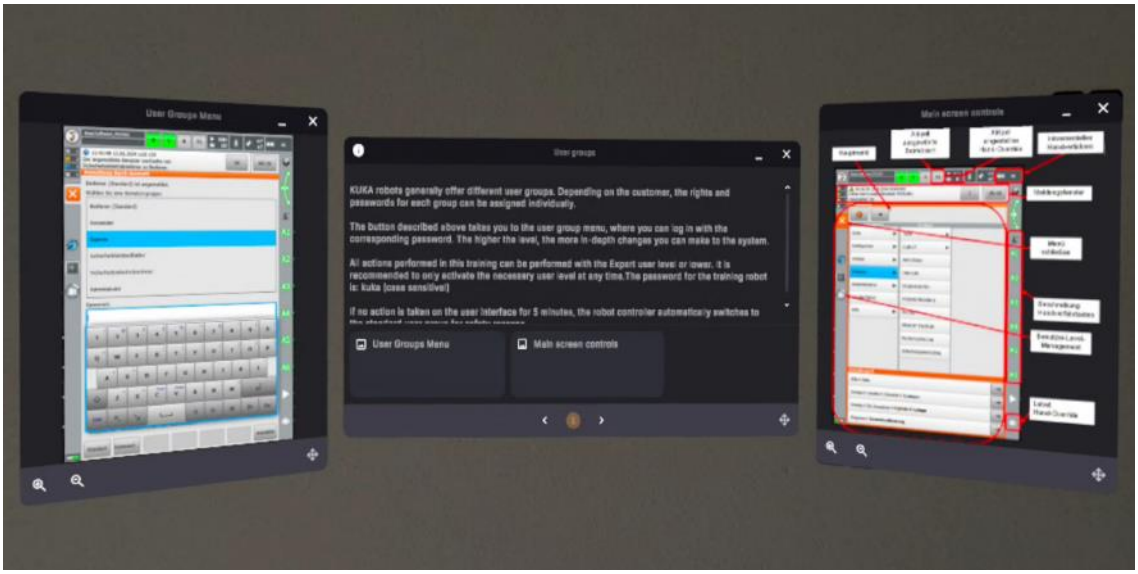


Figure 36 – Training Program Interactive Instruction Panels

2.2.4 XR5.0 Clawdite Platform – Workers' Movement Prediction Module

As introduced in D6.3, a Human Digital Twin (HDT) is instantiated through the XR5.0 Clawdite Platform by SUPSI, as a shared instance available to all partners involved in the XR5.0 project. The HDT hosts the Workers' Movement Prediction (WMP) Functional Module, which monitors, detects and predicts worker movements in real time over three short-term horizons (1s, 3s, 5s).

Detection relies on real-time multi-camera pose estimation (YOLO11-Pose) with worker ID tracking across fields of view, while prediction is performed through an enhanced Kalman filter tracking framework that also propagates orientation to estimate directional intent, enabling proactive rather than purely reactive robot behaviour. The module continuously publishes current and predicted worker positions to the shared Clawdite instance via MQTT. Its design and implementation are detailed in D3.1 (Section 5.3.1) and D3.2 (Section 4.3.1). The module enhances situational awareness and supports adaptive system behaviour in collaborative, safety-critical environments.

SUPSI facilitated the integration and deployment of the module at KUKA's premises, where it was preliminary evaluated in a real industrial, multi-worker scenario (D6.5, Section 3.2.4). For deployment, the module was optimised for NVIDIA Jetson embedded platforms and complemented by a runtime validation subsystem that continuously evaluates prediction accuracy. This lays the basis for future AMR path planning and risk-aware optimisation logics, to improve safety, coordination and efficiency in human-robot shared environments

2.2.5 Safety-Risk Zone Checker

The safety-risk zone checker is a service that complements the prediction module by analyzing anticipated worker positions to detect potential risks. The safety-risk zone checker service is implemented to prevent hazardous situations between humans and robots during active construction robot operations.

Worker positions are predicted by the external Human Digital Twin (HDT) through XR5.0 Clawdite platform by SUPSI. The safety-risk zone checker evaluates these predictions against the configured risk zone surrounding the robot. The safety-risk zone checker provides the current risk information through the HTTPS endpoint "/check-risk-zone". The XR application/device uses this endpoint and, based on the returned severity level, displays the corresponding visual warnings and audible warning on the XR device.

XR5.0 Clawdite HDT instance provides the current position of the worker and predicted worker movement around the robot at several short-term prediction horizons: 1 second, 3 seconds, and 5 seconds. The safety-risk zone checker retrieves these current positions together with the predicted positions of all workers and maps them to the following severity levels:

- Severity level 3 - Critical: A worker predicted to enter the risk zone in 1 second.
- Severity level 2 - Important: A worker predicted to enter the risk zone in 3 seconds.
- Severity level 1 - Moderate: A worker predicted to enter the risk zone in 5 seconds.
- Severity level 0 - Current: It represents the current position of the worker.

The latest worker IDs and their predicted movement at 1 second, 3 seconds and 5 seconds were used to check whether any worker is expected to enter a hazardous area. The risk zone is defined as the area 1.5 meters around the robot.

The safety-risk zone checker can be configured to select which worker position is used for the safety-risk check. Depending on this configuration, the service can use the worker's current position or one of the predicted positions horizons when checking whether the worker is inside the defined risk zone.

The following endpoints were introduced by the service,

- **/check-risk-zone** endpoint returns a JSON list of workers who are currently at risk, together with severity level. The XR application/device uses this information to present the visual warning alert and warning sound.
  - The endpoint available at: https://xr50.atb-bremen.de/check-risk-zone. Access to the endpoint requires a session-token header for secure communication.
  - An example response to this request: {"workers_in_risk_zone":[{"worker_id":"1","safety_risk":true,"safety_risk_message":"Severity level 3 - Critical"}]}
- **/latest-message** endpoint returns the latest workers movement prediction messages. This endpoint is useful for debugging and monitoring the raw messages being processed by the service.
  - The endpoint available at: https://xr50.atb-bremen.de/latest-message. Access to the endpoint requires a session-token header for secure communication.

- An example response of this request is:
  {"workers":[{“id”:11,”positions”:{“current”:{...},”1s”:{...},“2s”:{...},”5s”:{}},..}],”timestamp”:17
  6095053}

The complete workers positions are not shown here for the sake of brevity.

This safety-risk zone has already been deployed at KUKA’s facilities, and these endpoints have been integrated into the XR application. RoboSpace application communicates to “/check-risk-zone” endpoint to retrieve information of workers who are in hazardous situations close to the robot. RoboSpace app running in the Hololens shows the warning sign along with the warning sound to the worker of detected safety risk.

2.2.6 NOVAAS for Digitalizing XR Physical Assets and a AASs Manager

Pilot #1 uses Asset Administration Shells (AASs) to represent physical and XR-related assets in a standardised form, supporting data exchange across the pilot’s systems. The AASs were developed and deployed using UNP’s NOVAAS implementation. Their submodels describe each asset’s digital nameplate, technical data, handover documentation and operational data. NOVAAS connects these models to KUKA’s OPC UA servers, linking them with live robot data such as operational status and 3D position. This provides an AAS-based view of both asset information and current measurements for the Final Prototype. D6.3 details the asset modelling, AAS packaging, NOVAAS setup and OPC UA integration carried out during the Early Prototype phase.

In the Pilot #1 Final Prototype, the Clawdite Manager, developed using NOVAAS as its foundation, supports the integration of Asset Administration Shells (AASs) with the XR5.0 Clawdite platform. It detects and registers AASs on the network, makes their models and real-time data available to the platform, and monitors their availability. If an AAS becomes unavailable, the manager updates the platform accordingly. D6.3 describes the registration process, logging interface and methods for verifying AAS registration in detail.

2.2.7 KUKA OPC UA Server

The Pilot #1 Final Prototype uses OPC UA to transfer process data from the KUKA robot’s PLC to the RoboSpace XR application. RoboSpace displays values such as sensor readings, operating states and alarms as contextual AR overlays, allowing users to inspect machine data where it is relevant. The OPC UA client can connect locally through the robot’s Wi-Fi network or remotely through KUKA’s maintenance system. D6.3 describes the server and client setup, access procedures, PLC interface and tested OPC UA implementations in detail.

2.2.8 XR Worker-Robot Position Prediction Prototype

The prototype connects a pilot worker, the WMP functional module, a Position Estimation AAS, a Robot AAS, and the Clawdite HDT Platform. Its purpose is to demonstrate an end-to-end flow in which worker position observations are received, estimated, published, persisted as digital-twin data, and used alongside robot context.

The outcomes of the prototype are described in Table 2.

Table 2 – XR Worker-Robot Position Prediction Prototype Outcomes

Outcome	Definition of done
Ingestion	A worker position event is published to the available MQTT broker (part of the Clawdite platform) and consumed by the WPM functional module.

Estimation	The WPM validates the event, calculates a normalized position, and publishes an estimated-position state.
Digital twin	Worker and robot entities, descriptors, and recent state values are available through Clawdite.
AASs	The Position Estimation AAS and Robot AAS are thus responsible for consuming the estimated positions from Clawdite platform, translating into AAS Basic events, sharing these events over the local MQTT instance, and updating the robot controller registries (within the PLC data blocks) using the available OPC-UA server.
Demonstration	A repeatable scenario shows a worker entering a robot cell, moving through it, and leaving it.

Architecture Overview and Responsibilities

Figure 37 shows the architecture of the XR Worker-Robot Position Prediction Prototype.

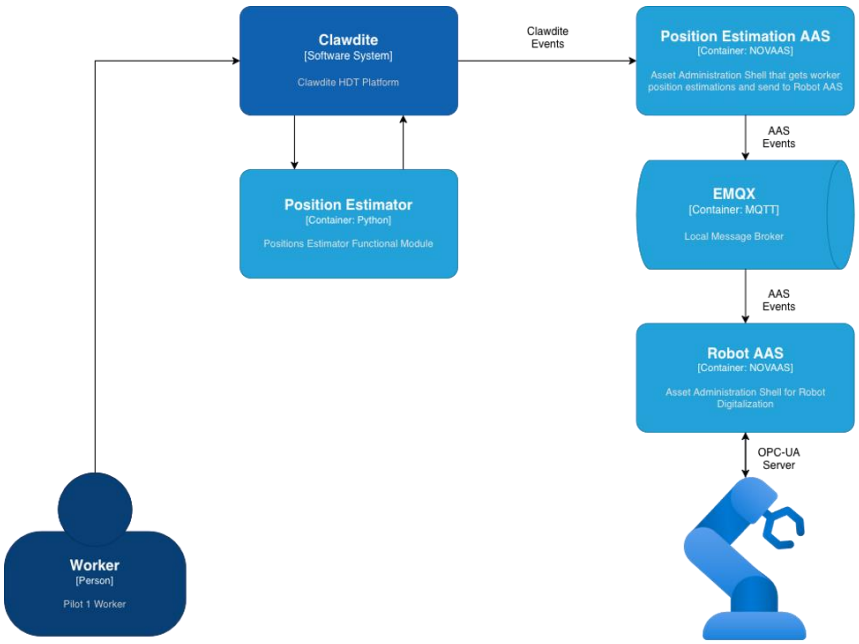


Figure 37 – Architecture of the XR Worker-Robot Prediction Prototype

Component	Role	Primary interface
Worker	Source of observations about a person in the pilot scenario.	MQTT publish via gateway or simulator
EMQX	Local broker that decouples producers from consumers.	MQTT 3.1.1 or 5.0; topics below
Position Estimator	Validates raw observations, performs filtering and coordinate normalization, and publishes estimated states to Clawdite using the internal platform broker.	MQTT subscribe/publish; optional health HTTP endpoint; Clawdite HDM REST client

Position Estimation AAS	AAS representation of the estimation capability and its latest output. It consumes the estimation from the Clawdite platform and translate them into aas-aligned basic events to be published over the local MQTT broker.	NOVAAS HTTP API; ingests latest state via MQTT subscription
Robot AAS	AAS representation of the robot and its operational context. It is connected to the robot/PLC controller using the provided OPC-UA server.	NOVAAS HTTP API; ingests position estimation from Position Estimation AAS via MQTT subscription and send it to the robot controller internal registries via OPC-UA.
Clawdite	HDT platform for workers, factory models, descriptors, states, sessions, and history.	HDT and HDM REST APIs

Logical Data Flow

The canonical data path for Prototype 1.0 is a classical broker architectural pattern (see figure xx):

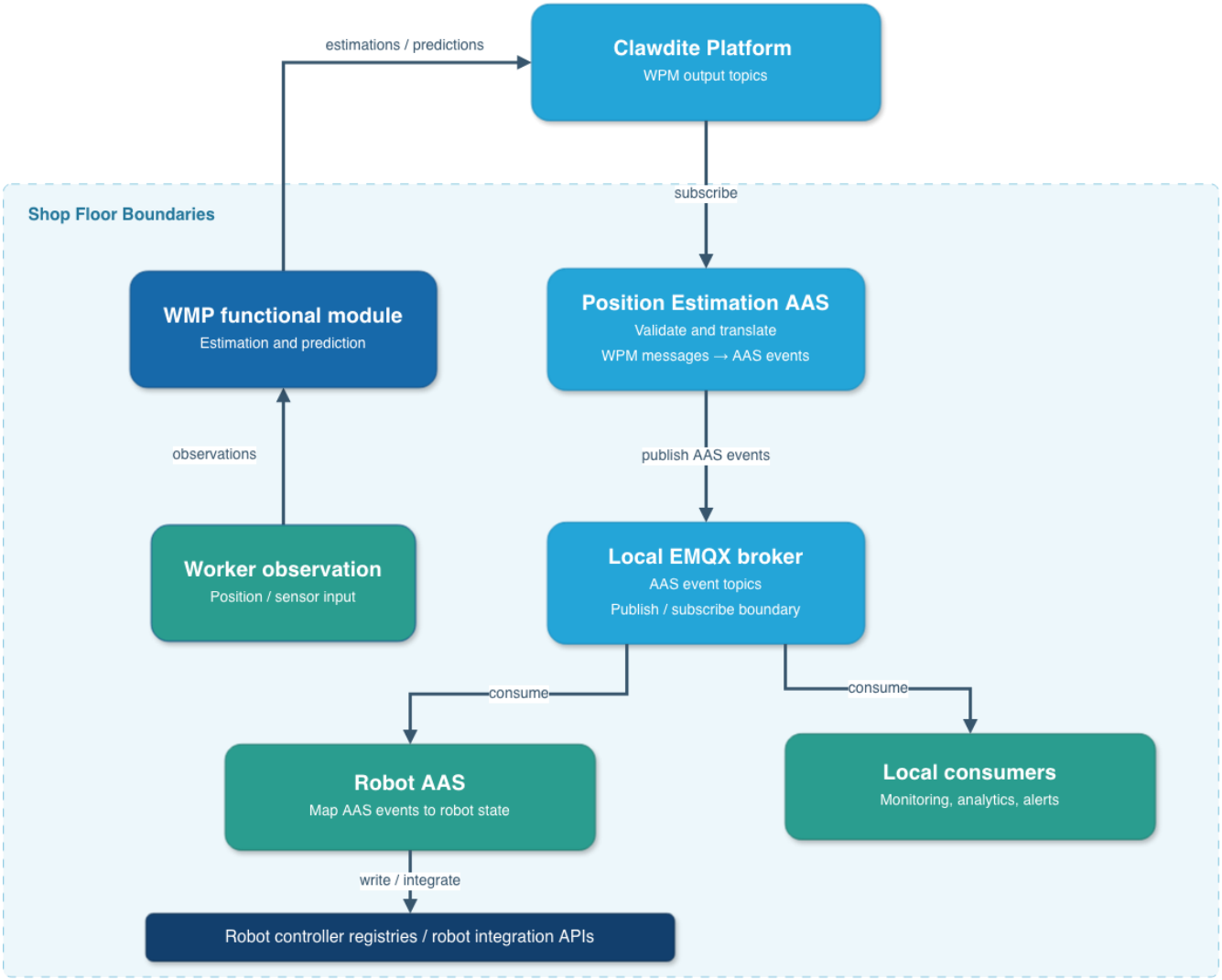


Figure 38 – Logical Data Flow based on Classical Broker Architectural Pattern

The WMP is the producer of estimation/prediction results. It publishes to MQTT (real time) and writes to Clawdite HDM. No other component writes estimation output to either channel. **Position Estimation AAS** obtains its latest values by subscribing to the estimated-state MQTT topics via NOVAAS's internal MQTT connector. Robot context flows separately: Robot AAS is connected to the KUKA construction robot and reads and writes data within the controller registries. The WPM does not write to any robot control system. Estimations/Predictions are ingested by the Position Estimation AAS, translated into aas suitable events and published within the local MQTT to be ingested by the Robot AAS.

Overview on the Data Streams

- **MQTT Contracts**
  - WPM functional module is publishing predictions on this topic:
    - HDT/{workerId}/state/{stateDescriptorId}
    - The payload contains worker positions with current, 1s, 3s, and 5s values.
  - The Position Estimation AAS translates predictions into AAS suitable events on this topic
    - aas/{base64(aasId)}/sm/{base64(submodelId)}/estimations
- **OPC-UA Write Contracts**
  - The robot AAS converts every worker estimation/prediction into OPC-UA write items, for each worker it writes:
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Human_ID"
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Position_actual"."Pos_x"
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Position_actual"."Pos_y"
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Position_1s"."Pos_x"
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Position_1s"."Pos_y"
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Position_3s"."Pos_x"
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Position_3s"."Pos_y"
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Position_5s"."Pos_x"
    - ns=3;s="DB_INTERFACE_OPC_UA"."Human_Safety"."IN"."Human"[i]."Position_5s"."Pos_y"

Position Estimation AAS UI

The Figure 39 shows the Worker page of the Position Estimation AAS.

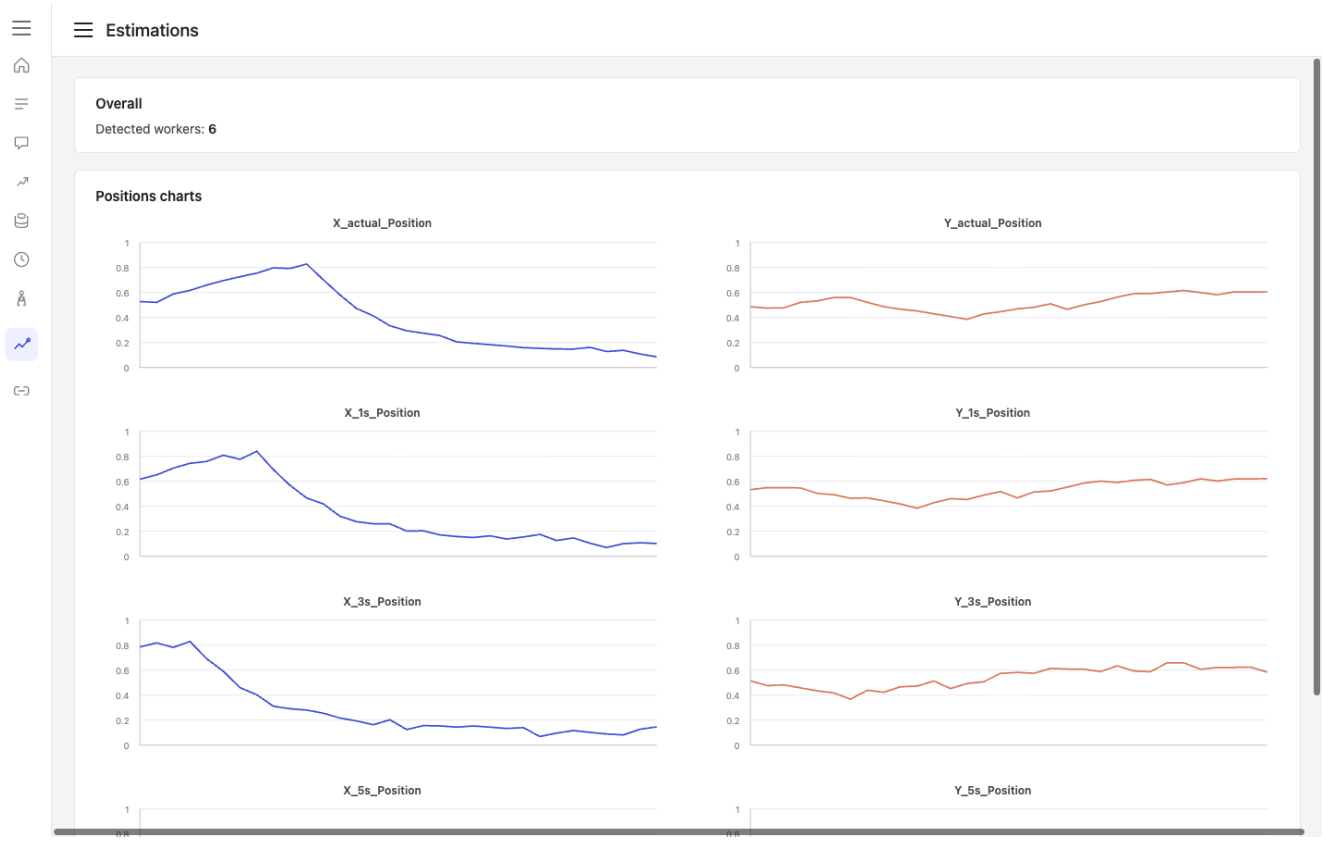


Figure 39 – Worker UI Page

WMP Functional Module UI

The Worker Movement Prediction (WMP) functional module provides the worker-tracking and short-term movement-prediction capabilities used by the XR worker–robot interaction scenario. As described in D6.3 and D3.1, the module processes synchronized camera streams to detect workers, estimate their positions and orientations, predict their short-term movement, and derive occupancy information.

The user interface provides an operational view of these outputs. It displays the camera feeds, tracked-worker identifiers and coordinates, current and predicted positions, movement directions, and occupancy-map information. The prediction horizons shown in the prototype include approximately one, three, and five seconds.

The WMP module outputs are published to the Clawdite internal MQTT broker, where they are consumed by the Position Estimation AAS. The Position Estimation AAS translates them into AAS-suitable events and publishes them to the local EMQX broker. The Robot AAS consumes the resulting events and maps the worker-position and prediction data to the robot integration and controller interfaces.

The screenshots below are included as a visual illustration of the WMP prototype interface. The detailed description of the detection, tracking, prediction, camera configuration, and occupancy algorithms is intentionally not repeated here; the reader is referred to D6.3 and D3.1 for the complete functional-module description.

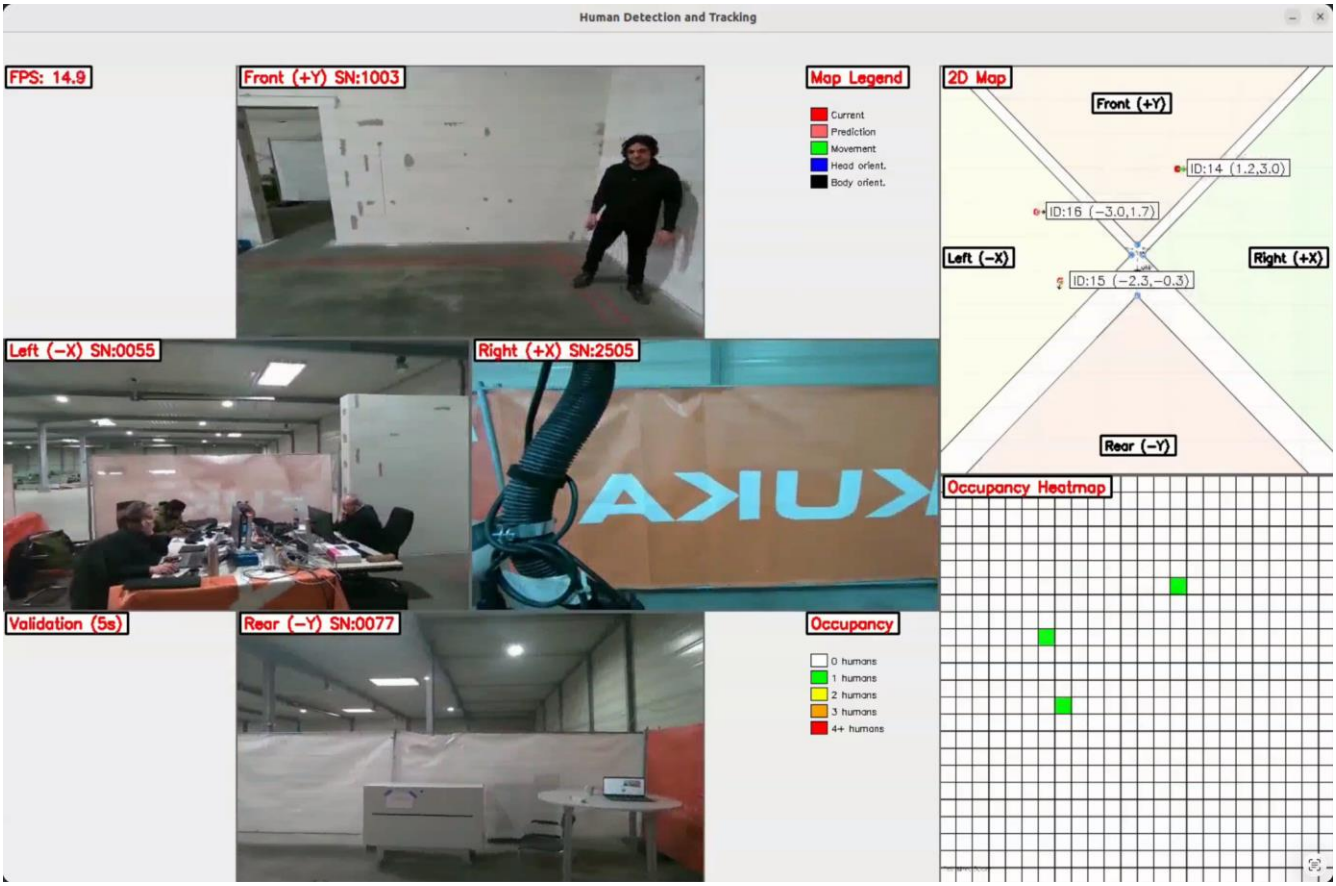


Figure 40 – XR Worker-Robot Position Prediction Prototype - Humans Detection from Camera System

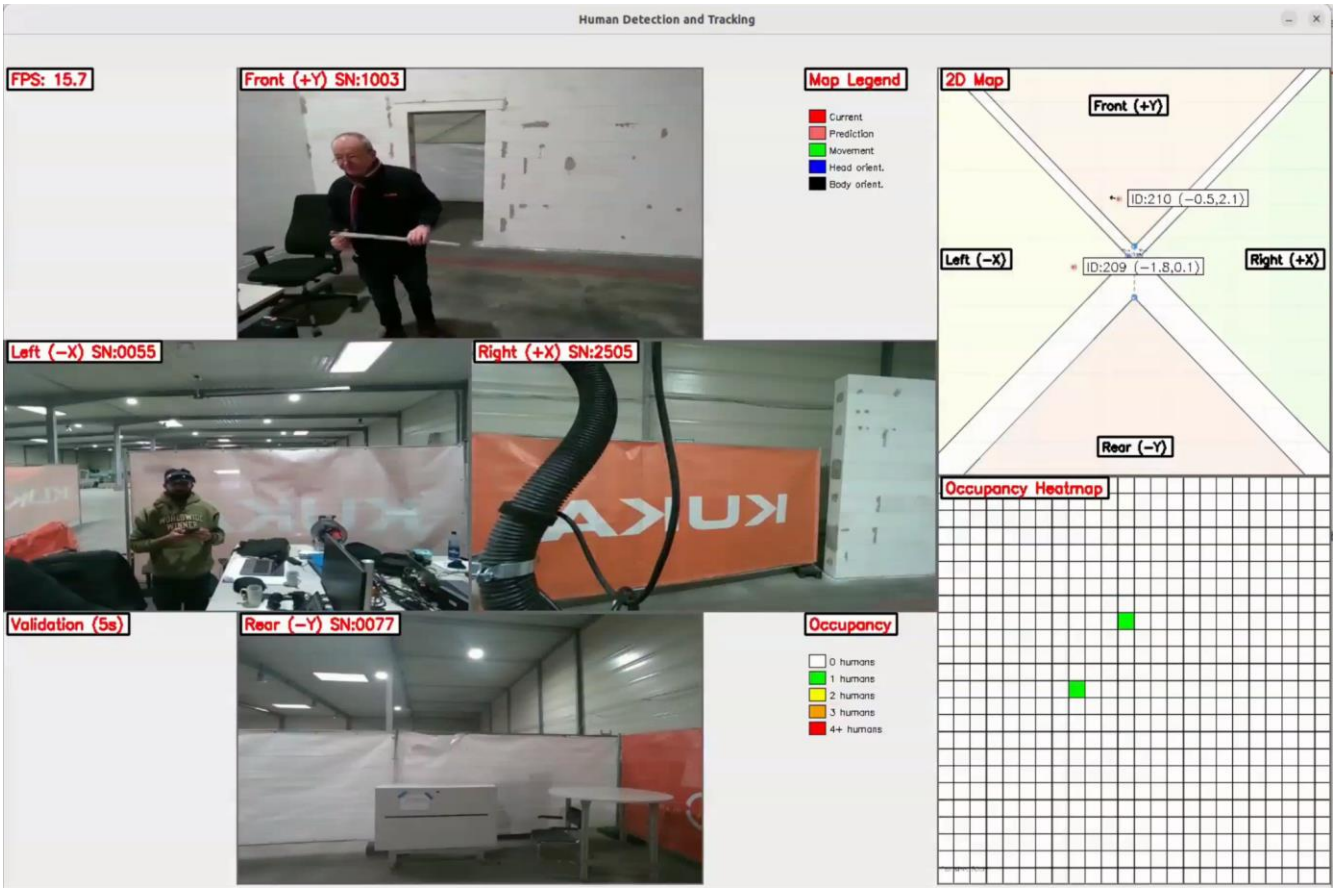


Figure 41 – XR Worker-Robot Position Prediction Prototype - Humans Detection from Camera System

The prediction information is particularly relevant to the position-estimation scenario because it provides short-term estimates of where a worker is expected to move. In the current prototype, these values correspond to future position horizons such as one, three, and five seconds. The occupancy view represents the monitored area as a grid. Each cell indicates the number of detected workers in that location. The two figures show different moments of the same monitoring process. In the first figure, several workers are detected simultaneously and appear as individual markers in the 2D map and as occupied cells in the heatmap. In the second figure, the detected worker identifiers and positions have changed, demonstrating that the WMP module continuously updates worker tracking as people move through the environment.

2.3 Component Usage Matrix

Table 2 gives an overview of the XR5.0 technical components in the Pilot #1 Full Prototype.

Table 2– XR5.0 Technical Components in the Pilot #1 Full Prototype

XR5.0 Technical Tasks	Technical Component	Technical Partner	How is the Component Used?
T3.1 Specifications for Worker Digital Images	N/A		

T3.2 Workers' Data Collection & Analytics	NOVAAS Framework and Asset Administration Shells (AASs)	UNP	Provides standardized data exchange across pilot components and interoperable services for industrial assets: Represent robot and XR-related assets, link asset information to live OPC UA data, and exchange worker-position estimates through the Position Estimation AAS and Robot AAS.
T3.3 Workers' Digital Twins	XR5.0 Clawdite Platform HDT and Workers' Movement Prediction module	SUPSI	Track workers and provide current and predicted positions over short time horizons of 1, 3 and 5 seconds for the worker–robot interaction scenario.
T3.4 Personalized XR Content Development	N/A		
T3.5 Adaptation of Personalized XR Content	N/A		
T4.1 Human-centered XAI Models	Textual transparency information provided with KUKABot responses	SIE	KUKABot returns reasoning information and source references that users can inspect alongside its answers.
T4.2 XR-enabled AL & NSAI Models	N/A		
T4.3 Generative AI Models	AI Chat/Voice Assistant KUKABot, deployed through the DataLens Conversational AI Framework	SIE	Provides the privacy-first, on-premise conversational AI backend: ingests KUKA technical documentation into an isolated knowledge base and answers spoken natural-language questions with structured, source-referenced replies comprising speech, formatted text, figures and tables, using locally hosted models. Requests are authenticated with XR5.0 Hub session tokens and authorised against the organisation the deployment serves. Operated through a browser-based administration interface with usage reporting.
T4.4 Visualization of XAI Explanations	KUKABot reasoning display in RoboSpace	SIE	Provides eXplainable AI output for KUKABot as a Chain-of-Thought reasoning trace returned with every reply, recording each stage of the decision path from question to answer alongside source references to the originating documents. Exposed

			as a first-class field for display in the XR application, supporting the transparency and human-oversight requirements of the EU AI Act.
T4.5 Visualization of AI Recommendations	KUKABot UI in RoboSpace XR application	HOLO	Provides an AR-based interactive for the XR-enabled AI Assistant KUKABot in the RoboSpace XR application
T5.1 Training Material	KUKA SmartPad-2 Training Material	IML	Converts traditional training materials into customized XR tutorials.
T5.2 Cloud-based XR Training Asset Repository	Cloud-based XR Training Asset Repository	SYN	Provides central storage and access to reusable XR training assets, supporting the creation and delivery of Pilot #1 training programmes through the XR5.0 Training Platform.
T5.3 Hololight Hub	Hololight Hub	HOLO	Hosts and streams RoboSpace from an on-demand cloud virtual machine to HoloLens 2, with central management of users, access and application versions.
T5.4 Training Programs	KUKA SmartPad-2 training programme, Training Programs Authoring Tool (TPAT) and XR Training Plugin (XRTP)	IML	Provides an XR-based immersive training program on operating the KUKA SmartPad-2, delivered via the XR Training Plugin. Enables training managers to author and assign structured training content and technicians to follow interactive SmartPad-2 guidance within RoboSpace.
Pilot-specific Components	Safety-risk Zone Checker	ATB	Evaluates current or predicted worker positions against the configured robot risk zone and supplies information for visual and audible warnings in RoboSpace.
	KUKA OPC UA Server	KUKA	Acts as a communication platform to enable process data exchange and AR visualization of real time data on XR devices. Exchanges robot process and joint-position data with pilot components, supporting live AR data overlays and robot-motion visualisation.
	RoboSpace Relay Service	KUKA	Provides the described middleware for conveying plant-control data to the cloud-hosted RoboSpace application, using plant-specific data-point configurations.

Table 3 maps the components of the Pilot #1 Full Prototype to UC1.1 (*Personalised XR in Assembly Line Production*) and UC1.2 (*Virtual Commissioning and Generative AI in Robotics*). In UC 1.1, the focus is on on-site technician assistance, worker monitoring and safety through XR. RoboSpace supports on-site technicians with contextual robot data, voice interaction and KUKABot assistance. The Workers' Movement Prediction module provides worker-position estimates to the Safety-Risk Zone Checker, whose integration with RoboSpace enables visual and audible risk warnings. D6.4 also describes a data flow that passes worker-position estimates through the Position Estimation AAS and Robot AAS to the robot controller.

UC1.2 combines the interactive SmartPad-2 training programme with robot visualisation and virtual commissioning support. The full prototype adds simulated robot-arm motion and visualisation driven by live joint-position data. The XR5.0 Training Platform enables training content to be authored, assigned and delivered within the cloud-hosted RoboSpace application. Across the pilot, NOVAAS provides structured asset representations and links them to live data, while OPC UA supports data exchange with the physical robot.

Table 3 – Mapping of Pilot #1 Technical Components to Pilot Use Cases in Full Prototype

	UC1.1	*UC1.2*
XR-enabled AI Assistant KUKABot	X	
XR-based Training Program		X
Worker Movement Prediction (XR5.0 Clawdite Platform)	X	Not planned
Safety-risk Zone Checker	X	Not planned
NOVAAS Asset Administration Shell Framework		X
KUKA OPC UA Server	X	X

2.4 DISCUSSIONS

Pilot #1 Full Prototype brings together XR visualisation, live robot data, AI assistance, technician training and worker-safety information in a single application. RoboSpace retains the true-to-scale robot overlays, inspection tools and voice-driven interaction demonstrated in the Early Prototype. It now adds a Component Navigator that lets technicians isolate individual parts by voice or manual selection. The robot-motion view can replay simulated movements or visualise live joint-position data received from the physical robot through OPC UA. These additions address the Early Prototype's principal limitation concerning the display of robot kinematics in XR.

The integration with the XR5.0 Training Platform extends the pilot from a locally streamed application to a centrally managed, cloud-hosted experience. Hololight Hub provisions and streams RoboSpace to HoloLens 2, while the platform supports user and application management, training authoring, assignment and progress tracking. Technicians can access the SmartPad-2 training programme within the same XR environment used to inspect the robot and view its live state.

KUKABot has also been extended. Its updated conversational service can interpret follow-up questions, retrieve information from a dedicated KUKA knowledge base and provide answers with source references. RoboSpace presents the returned text, speech, media and reasoning information in XR. These features improve access to technical documentation and give users more information with which to assess an answer; the displayed reasoning should not, however, be treated as independent verification of its correctness.

For the worker–robot interaction scenario, the Safety-Risk Zone Checker is now integrated into RoboSpace and provides visual and audible warnings based on worker-position information from the Human Digital Twin. The Full Prototype also describes an end-to-end data flow in which worker-position estimates pass through the Position Estimation AAS and Robot AAS to the robot controller interface. This demonstrates how prediction outputs can be exchanged across the pilot's XR, digital-twin and industrial data systems. The headset warnings provide information to the worker; the description does not establish them as a certified robot-safety function.

Some boundaries remain relevant to future real-world deployment. The new motion view supports simulated playback and visualisation of live robot movement, but the Full Prototype does not establish remote control of the physical robot or collaborative manipulation of a shared kinematic simulation. Reliable operation also depends on connectivity among the cloud-hosted application, KUKA's network, the robot controller and the supporting AI and prediction services. Further deployment work would need to assess performance, robustness, access control and the role of XR warnings within the site's established safety procedures.

3. PILOT 2 FULL PROTOTYPE

Pilot #2 – T6.3 Human-Centred Remote Maintenance & Asset Management, led by OCU, demonstrates how extended reality (XR), explainable AI (XAI) and AI-supported knowledge retrieval can be combined in industrial maintenance workflows. The Full Prototype focuses on reducing service downtime, improving access to equipment-specific knowledge, supporting less experienced technicians and enabling remote expert assistance through a device-agnostic workflow across smartphones, tablets, laptops and Microsoft HoloLens 2.

The Full Prototype was validated using a realistic maintenance scenario at SCHMIDT + HAENSCH: troubleshooting a software-related failure of a VariPol polarimeter and carrying out the corresponding SD-card replacement procedure. The scenario combines three complementary interaction modes: mobile remote support through Maintastic SHARE, AI-supported troubleshooting and training through a voice/text assistant, and XR-based step-by-step guidance on HoloLens 2.

The pilot addresses two complementary use cases:

- **UC2.1 – Remote Expert Support**: Enable off-site experts to provide real-time assistance to technicians through a shared XR workspace.
- **UC2.2 – AI-Supported Troubleshooting & Knowledge Management**: Integrate explainable AI models and a retrieval-augmented knowledge base to guide service processes and reduce time-to-resolution.

The Full Prototype architecture is aligned with the XR5.0 Remote Maintenance concept and is illustrated in Figure 21. Maintastic SHARE serves as the central workflow and collaboration backbone, connecting asset management, service tickets, maintenance knowledge, remote expert sessions and XR workflows. The AI and knowledge layer combines structured application data with vector-based retrieval over equipment manuals, workflows and training material. The XR5.0 Training Platform provides reusable XR training and maintenance assets that can be linked to the same service context.

The Pilot #2 Full Prototype architecture is aligned with the XR5.0 Remote Maintenance framework and is illustrated in Figure 21. The architecture combines industrial equipment, AI-supported knowledge services, maintenance management, remote expert collaboration and XR-based guidance within a modular system. The individual components are connected through Maintastic SHARE and the OCU KnowledgeBase, allowing maintenance information and workflows to be used consistently across mobile devices, XR devices and remote expert interfaces.
The architecture supports both Pilot #2 use cases. In UC2.1, the focus is on remote expert support, where technicians can initiate maintenance cases and collaborate with off-site experts through Maintastic SHARE. In UC2.2, the architecture supports AI-assisted troubleshooting and XR-based maintenance guidance by combining equipment-specific knowledge with contextual workflows and step-by-step XR instructions.
The main components of the Pilot #2 architecture are described below:

- **Target Equipment – SCHMIDT + HAENSCH Laboratory Instruments**: SCHMIDT + HAENSCH laboratory instruments represent the physical assets used within the pilot. Equipment-specific information, including operating status, measurements, error information and associated technical documentation, provides the context for maintenance and troubleshooting activities. The VariPol polarimeter is used as the main equipment for the Full Prototype validation scenario.
- **Maintastic SHARE Collaboration Platform**: Maintastic SHARE provides the main operational interface for technicians, maintenance experts and supervisors. The platform combines asset management, maintenance tickets, workflows, technical documentation and remote expert collaboration. Maintenance cases are linked to the corresponding equipment, allowing relevant

information and workflows to remain associated with the asset throughout the service process. SHARE additionally provides XR-enabled video communication for remote support scenarios in UC2.1.

- **OCU KnowledgeBase and Application Services**: The OCU KnowledgeBase provides the application and data layer connecting Maintastic SHARE with AI-supported knowledge retrieval and maintenance workflows. It combines structured application data stored in PostgreSQL with vector-based semantic retrieval and external AI services. Interfaces to services such as the Azure SDK and OpenAI SDK support the processing of maintenance requests and the generation of context-aware troubleshooting guidance.
- **Ticketing and Asset Management**: Maintenance tickets provide the common context for reported faults, troubleshooting activities and subsequent workflow execution. Each ticket can be associated with the corresponding asset, technical documentation and maintenance procedure. This enables equipment-specific information to be reused throughout the maintenance process and provides a structured basis for escalation to remote experts or XR-supported workflows.
- **Workflow Orchestration and XR Guidance**: Maintenance and training procedures are represented as structured step-by-step workflows. Depending on the task, these workflows can be presented on mobile devices or transferred to an XR device for spatial guidance. Integration with the XR5.0 Training Platform enables reusable training and maintenance content to be incorporated into the Pilot #2 workflow.
- **Technician Interaction – UC2.2**: On-site technicians access maintenance information, troubleshooting support and workflows through mobile devices and, where spatial guidance is beneficial, through HoloLens 2. AI-supported instructions use the available asset, ticket and knowledge context to provide relevant troubleshooting steps. XR workflows can subsequently guide the technician through maintenance procedures using step-by-step instructions and spatial information.
- **Remote Expert Interaction – UC2.1**: Remote experts and supervisors access the maintenance case through Maintastic SHARE. The platform provides the relevant asset, ticket and documentation context and enables real-time communication with the on-site technician. Where required, visual information from the technician's environment can be shared to support diagnosis and guidance. AI-supported information and retrieved technical documentation can additionally be reviewed by the expert before recommendations are communicated to the technician.

The resulting architecture provides a common information flow from the physical equipment and associated maintenance data to AI-supported troubleshooting, structured workflows and remote or XR-based technician assistance. This modular approach allows the interaction method to be selected according to the maintenance task: mobile interfaces provide low-friction access to information and troubleshooting, while XR and remote expert functions are used when spatial guidance, hands-free interaction or additional expert support is required.

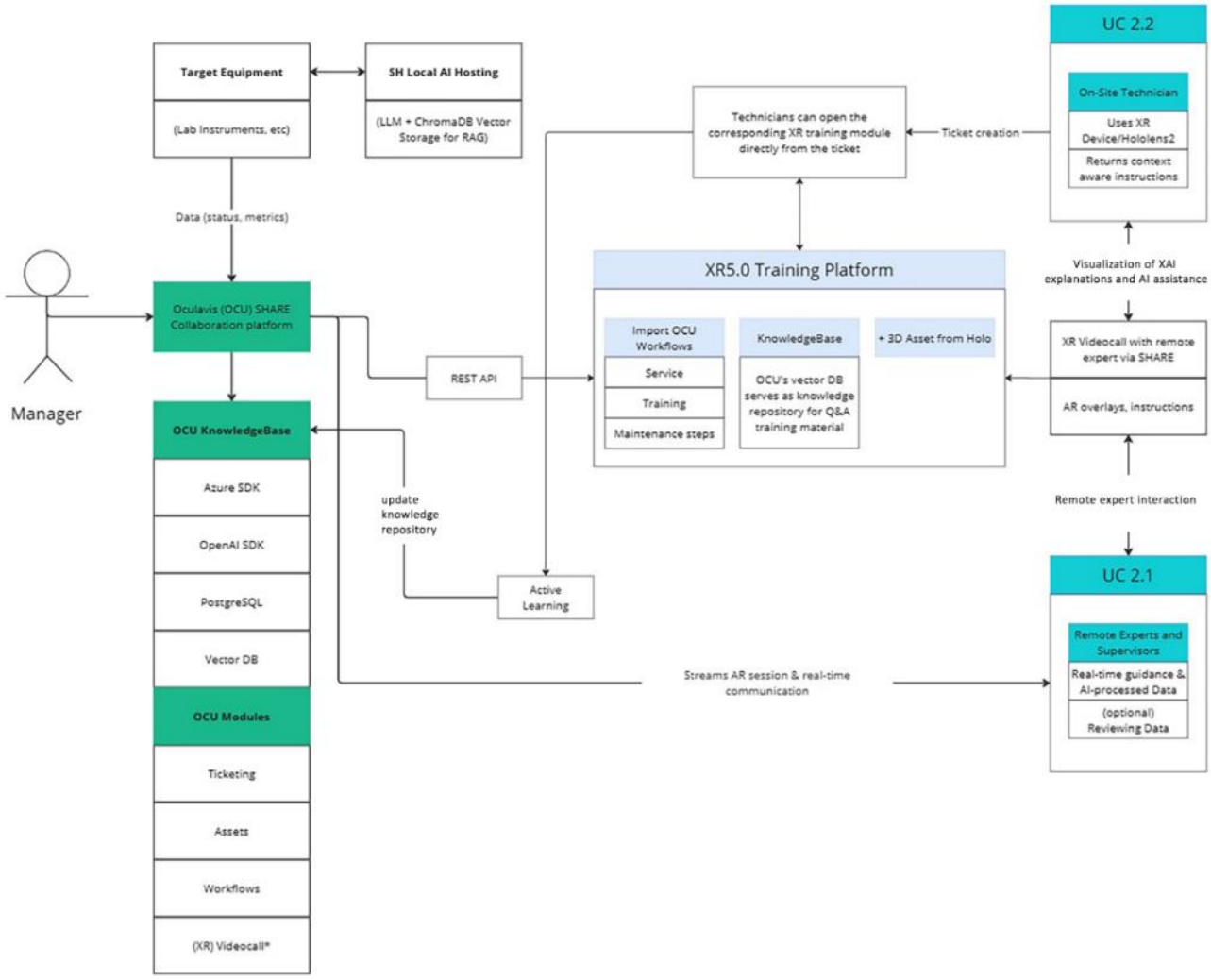


Figure 42 – Pilot #2 Architecture

3.1 PILOT-SPECIFIC INFRASTRUCTURE

The general infrastructure established during the Pilot #2 Early Prototype remains valid. Since D6.3, the infrastructure has been extended and integrated into a more complete end-to-end maintenance workflow.

A central development is the closer integration of the mobile troubleshooting functionality with Maintastic SHARE and the equipment-specific knowledge layer. Maintenance requests can be associated with the corresponding asset and service case, allowing documentation, troubleshooting information and maintenance workflows to be retrieved within the same operational context.

The Full Prototype also extends the XR workflow by integrating reusable maintenance and training content with the XR5.0 Training Platform. This enables maintenance procedures and associated XR assets to be used consistently for both technician training and operational maintenance.

Finally, the interaction concept has evolved from an XR-centred approach towards a device-agnostic workflow. Smartphones and tablets are used for low-friction troubleshooting and information access, Maintastic SHARE provides remote expert collaboration, and HoloLens 2 is used where spatial visualisation or hands-free guidance provides additional value.

Pilot #2 Full Prototype specific infrastructure consists of the following components:

- Hardware
  - Microsoft HoloLens 2 AR headset
  - iOS/Android smartphones and tablets (technician devices)
  - Remote expert workstation (PC with Maintastic client)
  - Local/edge compute server (for RAG/LLM hosting, ChromaDB-based)
- Software
  - Mobile troubleshooting AI assistant (React Native prototype)
  - Maintastic SHARE platform (remote assistance, asset management, workflows)
  - XR5.0 Human-centered Explainable AI (XAI) services (T4.1)
  - Local and cloud-based RAG pipeline for domain knowledge management

3.2 PILOT-SPECIFIC INSTALLATION, CUSTOMISATION AND CONFIGURATION OF XR5.0 TECHNOLOGIES

3.2.1 Mobile Troubleshooting AI Assistant

The Mobile Troubleshooting AI Assistant developed during the Early Prototype remains the primary AI-supported interface for technicians. The technical principles of the component are described in D6.3.

Since D6.3, the main development effort has focused on integrating the assistant more closely with equipment-specific maintenance information and the wider Pilot #2 workflow. Instead of treating the AI interaction as an isolated chatbot, the Full Prototype combines the technician's request with the relevant asset, maintenance case and available technical knowledge.

The assistant uses a retrieval-augmented generation approach in which relevant information is retrieved from equipment manuals, maintenance documentation and validated workflows before a response is generated. The resulting guidance can therefore refer directly to equipment-specific information rather than relying only on the general knowledge of the underlying language model.

Technicians can interact with the assistant through either text or natural speech. The assistant can provide troubleshooting recommendations, sequential maintenance instructions and references to relevant

technical information. Where a suitable structured workflow is available, the assistant can additionally direct the technician towards the corresponding maintenance or XR procedure.

The Full Prototype integration activities demonstrated that smartphones and tablets provide an effective first interface for troubleshooting. The familiar interaction model and low navigation overhead allow technicians to access information quickly without requiring an XR device for every maintenance request.

Figure 43 illustrates the user interaction with the Mobile Troubleshooting AI Assistant using a speech prompt.

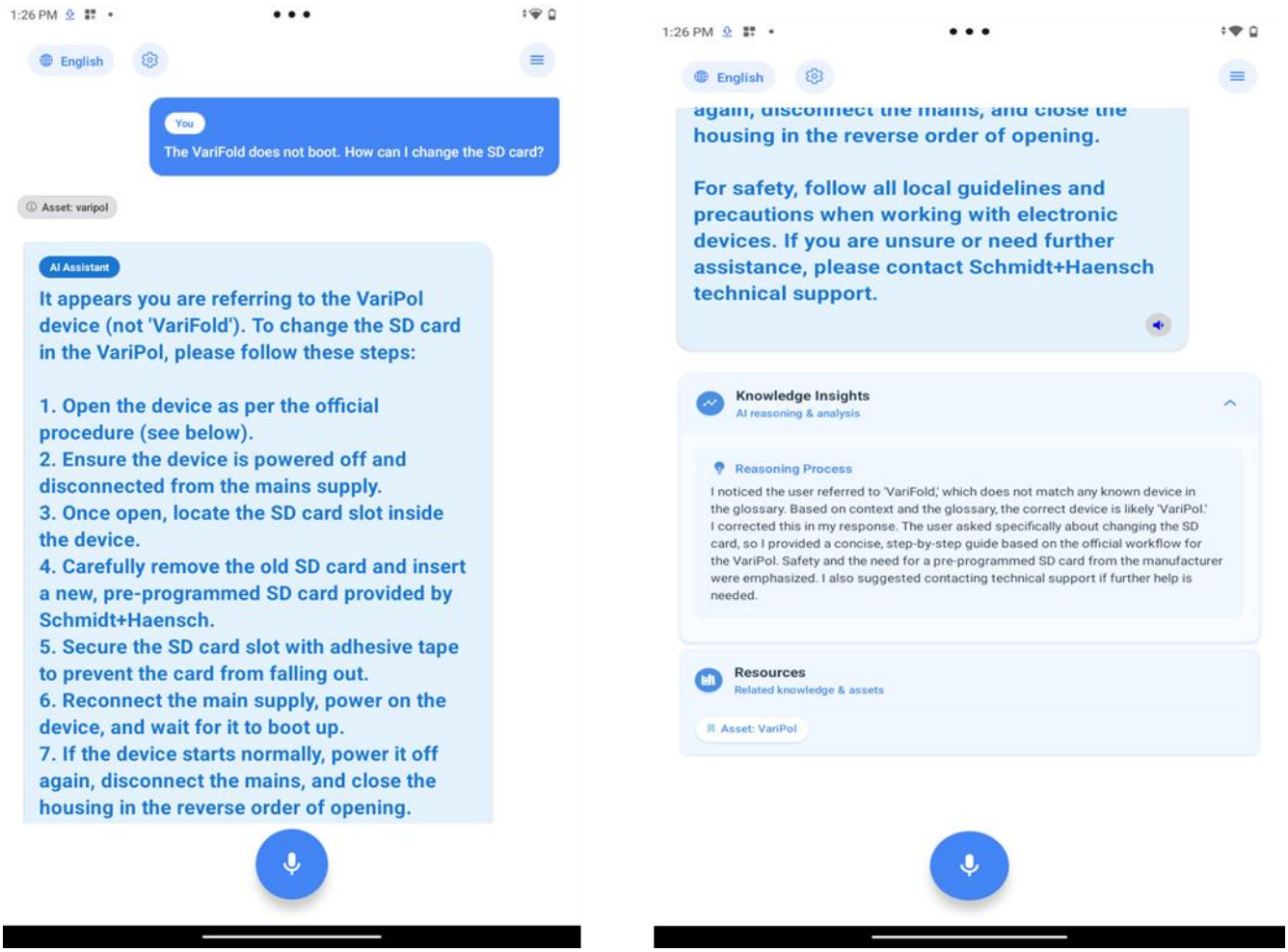


Figure 43 – User Interaction with the Mobile Troubleshooting AI Assistant

3.2.2 Mainstatic SHARE Platform

Maintastic SHARE remains the central maintenance and collaboration platform of Pilot #2. The platform combines asset management, maintenance tickets, workflows, technical documentation and real-time collaboration within one environment.
Since the Early Prototype, the main focus has been on using SHARE as the common operational backbone connecting the individual Pilot components. Maintenance activities are associated with the corresponding asset and ticket so that troubleshooting information, workflows and expert communication remain linked to the same service case.

For UC2.1, Maintastic SHARE provides the basis for remote expert support. An on-site technician can escalate a maintenance case to an experienced expert without requiring the expert to travel to the equipment location. The expert can access the relevant asset and service information and communicate with the technician through the remote collaboration functionality.
This connection between structured maintenance information and real-time expert communication is important for the Pilot because remote assistance is treated as part of the maintenance workflow rather than as a separate communication channel.

For UC2.2, SHARE provides the asset and workflow context used by the AI-assisted troubleshooting and XR-supported maintenance functions. The same service case can therefore progress from initial fault reporting to AI-supported troubleshooting and, where required, to an XR workflow or remote expert session.

3.2.3 XR Human-Centered XAI Models

The human-centred XAI functionality introduced in the Early Prototype was further integrated into the troubleshooting workflow during the Full Prototype development.
The objective of the explanation functionality is to provide technicians with additional information about the basis of an AI-supported recommendation without unnecessarily increasing the amount of information displayed during normal operation.

For this purpose, AI-supported recommendations can be complemented by an optional textual explanation. The explanation can indicate how the user request was interpreted, which equipment-specific information was considered and which technical sources or workflows support the recommended action.
Where available, recommendations are linked to the corresponding knowledge assets, such as equipment manuals or validated maintenance procedures. This provides traceability and allows technicians or remote experts to review the information on which the recommendation is based.
The explanation remains secondary to the actual maintenance instruction and can be accessed when required. This approach supports transparency while avoiding additional cognitive load during routine maintenance activities.

Figure 44 illustrates the visualisation of the textual XAI information.

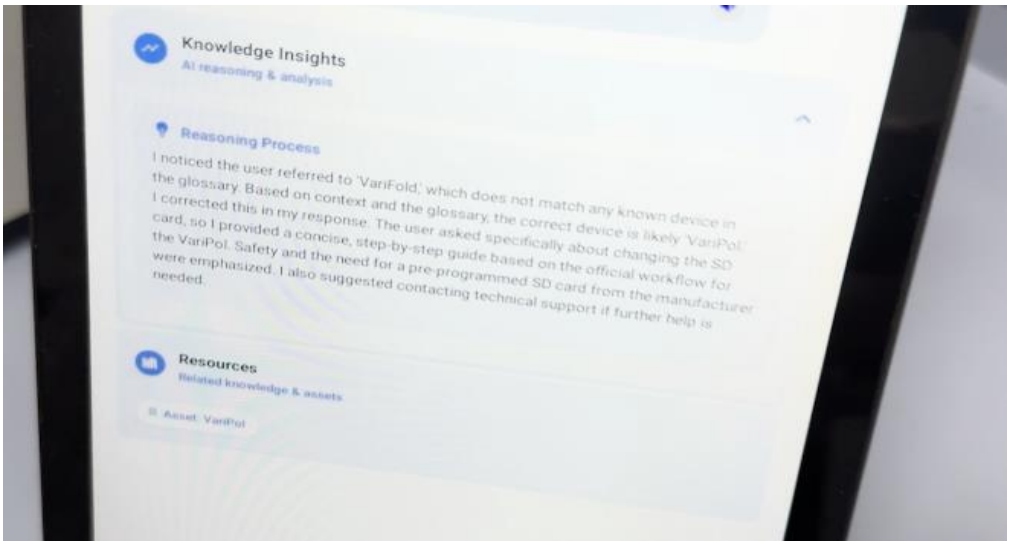


Figure 44 – Visualization of textual XAI explanations

If available, the explanation is linked to resources and knowledge assets (e.g., manuals, device-specific workflows) so technicians can trace recommendations back to authoritative sources. This approach provides transparency without overloading the user, as explanations remain hidden until requested.

3.2.4 QR-Code Workflow Handover

The QR-code workflow handover developed during the Early Prototype remains an important element of the Full Prototype interaction concept.
The purpose of the handover is to separate workflow selection and information retrieval from the spatial interaction required on the XR device. A technician can first identify the appropriate maintenance workflow using the mobile troubleshooting interface. Once the corresponding XR workflow has been selected, the system generates a QR code containing the relevant workflow reference.
The QR code can then be scanned with HoloLens 2, which opens the corresponding maintenance procedure directly. The technician does not need to repeat the troubleshooting process or manually search for the workflow within the XR interface.

The Full Prototype integration tests confirmed the value of this approach. Interaction with spatial interfaces can introduce additional navigation effort, particularly for users with limited experience using XR devices. Moving workflow selection to a familiar mobile device reduces this interaction overhead while retaining the hands-free and spatial advantages of XR during the maintenance procedure itself.
Figure 45 illustrates the QR-code handover between the mobile interface and HoloLens 2.

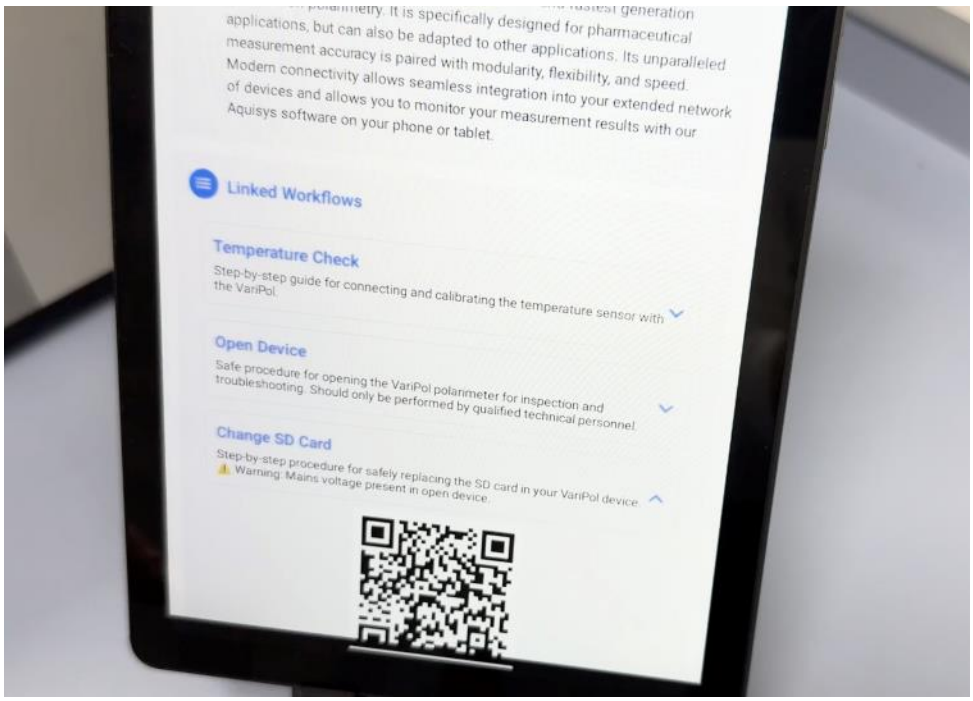


Figure 45 – QR Code with Workflow IDs

3.3 COMPONENT USAGE MATRIX

Table 4 gives an overview of the pilot-specific and XR5.0 technical components in the Pilot #2 Full Prototype.

Table 4 – XR5.0 Technical Components in the Pilot #2 Full Prototype

XR5.0 Technical Tasks	Technical Component	Technical Partner	How is the Component Used?
T3.1 Specifications for Worker Digital Images	N/A		
T3.2 Workers' Data Collection & Analytics	N/A		
T3.3 Workers' Digital Twins	N/A		
T3.4 Personalized XR Content Development	N/A		
T3.5 Adaptation of Personalized XR Content	N/A		
T4.1 Human-centered XAI Models	CoT-enabled Textual Explanations in in the Mobile Troubleshooting AI Assistant	OCU	Provides textual explanations that outline how the AI Assistant arrived at its recommendation including terminology, correction of user errors and step-by-step guidance.
T4.2 XR-enabled AL & NSAI Models	N/A		
T4.3 Generative AI Models	Mobile Troubleshooting AI Assistant	OCU	Provides voice-to-voice and text troubleshooting guidance referencing structured equipment manuals and past service records.
T4.4 Visualization of XAI Explanations	Textual XAI Visualizations in the Mobile Troubleshooting AI Assistant	OCU	Textual XAI explanations are provided as an extension to the AI outputs and linked to resources & knowledge assets if available.

T4.5 Visualization of AI Recommendations	Textual Visualizations in the Mobile Troubleshooting AI Assistant	OCU	AI explanations are displayed in a collapsible text panel within the mobile interface.
T5.1 Training Material	Equipment-specific Maintenance and Training Material	IML	Provides reusable step-by-step instructions and supporting technical content for technician training and maintenance, including the VariPol polarimeter SD-card replacement procedure. The material supports guided workflow execution on mobile and XR devices.
T5.2 Cloud-based XR Training Asset Repository	XR Training and Maintenance Asset Repository	SYN	Stores reusable XR training and maintenance assets for incorporation into Pilot #2 workflows. Selected procedures are linked to the maintenance context and launched on HoloLens 2 through the QR-code workflow handover.
T5.3 Hololight Hub	N/A		
T5.4 Training Programs	N/A		
Pilot-specific Components	QR Code Workflow Handover	OCU	QR code generated by the Mobile Troubleshooting AI Assistant is scanned by the Hololens 2 XR device to launch the corresponding step-by-step AR workflow.

Table 4 illustrates how the core prototype components are mapped to the Pilot #2 use cases. The mapping reflects the hybrid interaction model validated in the pilot: mobile AI and Maintastic SHARE provide the primary maintenance entry point, while remote expert collaboration and XR workflows are activated according to the complexity and spatial requirements of the task.

Table 5 – Mapping of Pilot #2 Technical Components to Pilot Use Cases in Full Prototype

	UC2.1	*UC2.2*
Mobile Troubleshooting AI Assistant	X	X
Maintastic SHARE	X	X
Human-centered XAI Models and Visualization of XAI Explanations	X	X
QR-Code Workflow Handovers	Not Planned	X

In UC2.1, Maintastic SHARE acts as the central entry point for the maintenance case and remote expert session. The technician can first use mobile troubleshooting and available knowledge to clarify the problem

and then escalate the case to an experienced expert while retaining the relevant asset, ticket and documentation context. XR can be used when a shared spatial view or hands-free guidance improves remote collaboration.
In UC2.2, the mobile assistant and equipment-specific knowledge layer support troubleshooting and training directly from the active maintenance context. Validated procedures can be continued as structured XR workflows. When spatial presentation provides an advantage, the selected workflow is transferred to HoloLens 2 through the QR-code handover and executed using reusable training and maintenance content from the XR5.0 workflow layer.

3.4 Discussions

The Pilot #2 Full Prototype was validated at the SCHMIDT + HAENSCH headquarters in Berlin using a VariPol polarimeter maintenance scenario. The practical test involved an experienced maintenance expert and an untrained technician and compared complementary approaches for resolving the same maintenance problem: mobile remote support through Maintastic SHARE, AI-supported troubleshooting and guidance, and a HoloLens 2 XR workflow for the SD-card replacement procedure.
A key success of the Full Prototype is the integration of maintenance management, equipment-specific knowledge, AI assistance, remote expert support and XR workflows into one connected process. Maintastic SHARE provides a common asset and ticket context, while the AI layer helps technicians retrieve relevant knowledge and identify appropriate next steps. The QR handover allows a procedure selected on a mobile device to be continued on HoloLens 2 without forcing the technician to repeat workflow navigation in the spatial interface.
The evaluation confirmed the importance of a mobile-first interaction strategy. For straightforward troubleshooting and information retrieval, the technician preferred smartphones or tablets because interaction was faster and more familiar. XR provided additional value when instructions needed to remain visible while both hands were occupied, when spatial information was relevant or when remote support benefited from a shared view of the working area.
The main XR usability challenge was interaction with the spatial user interface. Gesture precision, spatial navigation, overlay density and device responsiveness can increase cognitive load, particularly for maintenance procedures that are comparatively simple. The Full Prototype therefore treats XR as an optional escalation and execution layer rather than as the mandatory first interface for every maintenance task.
The pilot also demonstrated the importance of explainability and source traceability. Technicians and experts need to understand the basis of AI-supported guidance and must be able to connect recommendations to validated documentation or workflows. Providing explanations and retrieved source information on demand supports trust without overloading the primary maintenance interface.
Further work should focus on simplifying HoloLens interaction, expanding the equipment-specific knowledge base, increasing the number of validated XR maintenance workflows and strengthening the reuse of training content through the XR5.0 Training Platform. Additional field trials should evaluate task completion time, error rate, usability, trust, remote-resolution rate and the extent to which expert travel can be reduced across a broader range of service scenarios.
A further development direction is a closed operational knowledge loop in which validated expert corrections, technician feedback and completed maintenance cases can improve future knowledge retrieval and workflow selection. This would allow the Pilot #2 architecture to evolve from a static assistance solution toward a continuously improving maintenance knowledge system while keeping human validation at the centre of the workflow.

4. PILOT 3 FULL PROTOTYPE

Pilot #3 – T6.4 XR Streaming for Operator 5.0 Training, led by HOLO, integrates mobile augmented reality (AR), artificial intelligence (AI) and structured XR-based training to support water infrastructure inspection and maintenance. The full prototype combines geolocated visualisation of underground pipe networks and sensor data with explainable anomaly detection, text- and voice-enabled AI guidance, and operator training. Adapted to smartphones and tablets in response to operational and security requirements, the solution aims to improve situational awareness, support informed maintenance decisions and strengthen operator skills.

The pilot addresses three core use cases:

- UC3.1 - XR-Based Training on Visual Inspection and Anomaly Detection
  - UC3.1.1 - AR application to detect leaks and anomalies faster by visualizing geolocated pipe networks and sensor-based alerts directly in AR.
  - UC3.1.2 - AI-supported CCTV operations during pipe inspection.
- UC3.2 - XR-Based Training on Condition Based Maintenance
  - UC3.2.1 - AI-powered AR-based training support for maintenance activities.

Pilot #3 Full Prototype has been co-developed by EKSO, CYENS, UPRC, INNOV, SIE, HOLO and IML, building on the use case co-creation reported in D6.1 and the early prototype described in D6.3. The implementation integrates Human-centred XAI models (T4.1), XR-enabled Active Learning and Neurosymbolic AI models (T4.2), Generative AI models (T4.3), Visualisation of XAI explanations (T4.4) and Visualisation of AI recommendations (T4.5) within a mobile AR application for smartphones and tablets. WP5 contributes Training Material (T5.1), covering equipment and safety checklists and operational procedures, and Training Programs (T5.4), supporting water-pipe maintenance training through the integrated, mobile-adapted XR Training Plugin. The Smart Pipes API complements these capabilities by providing access to sensor data and geolocated anomalies.

Following feedback from EKSO, the Pilot #3 implementation was adapted from the initially envisaged AR-glasses-based solution to a mobile AR application due to operational and security constraints. The CYENS AR application, developed using the Unity game engine, enables workers to visualise geolocated smart pipe networks, sensor information and detected anomaly locations in real-world environments. The application also integrates AI-supported guidance and voice-based interaction to assist operators during inspection and maintenance activities.

On the AI side, neuro-symbolic models trained using EKSO's CCTV inspection data support the automatic detection of anomalies, while XAI visualisations provide explanations and contextual information behind AI-generated findings. AI recommendations and Generative AI capabilities further support operators in interpreting inspection results and making intervention informed decisions. In parallel, the XR-enabled Active Learning environment supports worker training and iterative improvement of AI models.

The training dimension is complemented by the IML XR Training Plugin, which has been integrated and tested within Pilot #3 to support structured training scenarios and operator familiarisation with inspection and maintenance procedures. The current implementation therefore combines mobile AR, AI/XAI, Active Learning, Generative AI, AI recommendations and XR-based training capabilities. The initially planned integration with the Hololight Hub and Hololight Stream for cloud-based XR streaming is not part of this phase of implementation following the transition to mobile devices for the above mentions reasons.

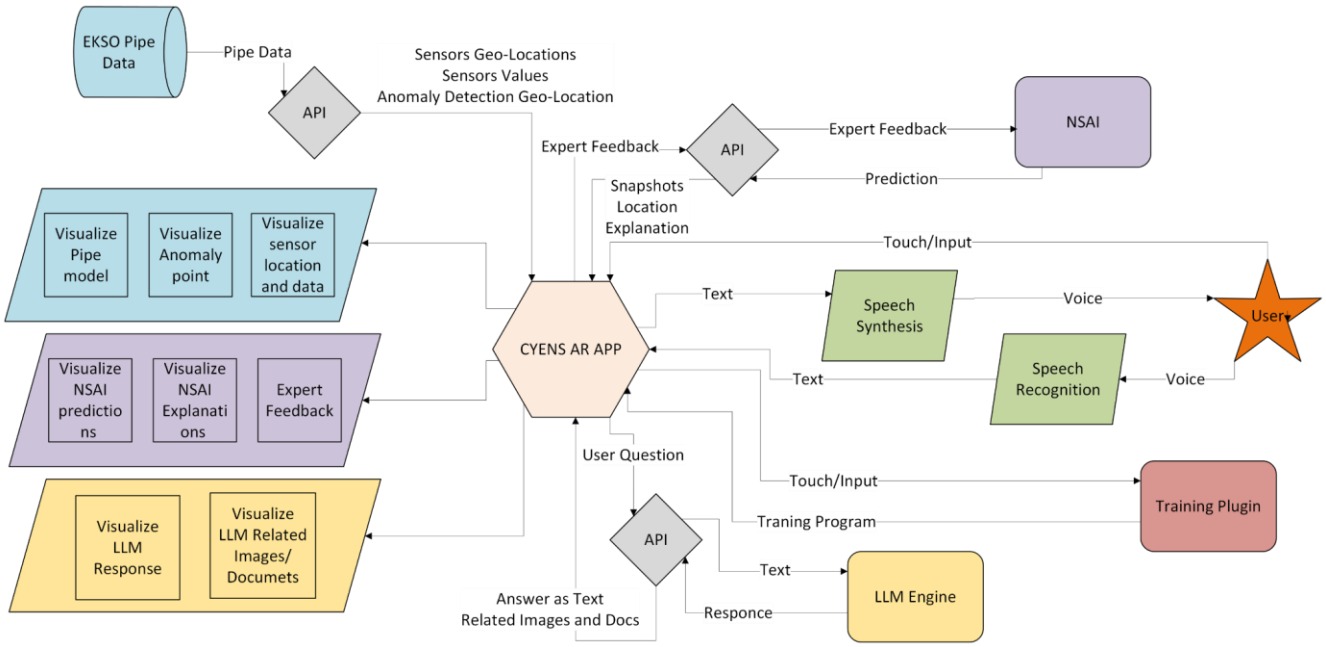


Figure 46 – Pilot #3 Architecture

4.1 Pilot specific Infrastructure

Pilot #3 Full Prototype infrastructure consists of the following:

- Hardware
  - Android Smartphone/Tablet
  - Rovion Robot Crawler
- Software
  - Unity-based AR application
  - Outlier/Anomaly Detection with Active Learning
  - Neurosymbolic AI Module
  - LLM Chat Engine
  - ETL workflows through the Smart Pipes API
  - IML's Training Program platform
  - Cloud-servers

4.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies

4.2.1 AR Application

The Pilot #3 AR application, developed within the XR5.0 framework, showcases how Augmented Reality (AR) and AI-supported technologies can support inspection, training, monitoring, and decision-making in ordinary/extraordinary maintenance operations contexts. The application enables users to visualise smart pipe infrastructure, anomaly locations, sensor information, AI-generated explanations and recommendations directly within their physical surroundings, creating a spatially aware interface that connects real-world assets with digital intelligence to fully assist operators.

The application has been implemented by CYENS in close collaboration with project partners providing the backend components and AI services for the EKSO end-user. These include AI/NSAI and Active Learning models, Generative AI capabilities, XAI explanations, AI recommendations and the Smart Pipes API. The training dimension is further supported by the IML XR Training Plugin, which has been integrated and

tested within Pilot #3 to provide structured training scenarios and support operator familiarisation with inspection and maintenance procedures.

The application was initially envisaged for AR head-mounted devices such as Magic Leap 2, with XR streaming through the Hololight platform. Following feedback from EKSO and the practical requirements identified during the pilot, the implementation was transitioned to a mobile-based AR solution for smartphones and tablets, that could fit the security and operational needs of the operators. Mobile devices therefore constitute the current primary deployment platform, providing a more accessible and practical solution for the pilot environment. The Hololight Hub and Hololight Stream infrastructure are not part of the first-phase implementation.

A key element of the mobile AR application is spatial alignment of digital pipe infrastructure with the physical environment in order to give a realistic picture of intervention to the operator. The application uses GPS and compass information together with marker-based positioning to support the placement and alignment of pipe models, sensors and anomaly information in the user's surroundings for a precise location and intervention on the infrastructure. This enables operators to interact with relevant infrastructure and AI-generated information in a contextualised AR environment while carrying out inspection, maintenance or training activities.

Pipe Model Visualization

One of the central systems of the application is the visualization of pipe infrastructure models in AR (as illustrated in Figure 47. The system loads 3D representations of underground civil or industrial pipelines and anchors them in the physical environment using geo-location input. Users can move around the augmented scene to inspect the pipelines from different perspectives, while models are scaled and oriented to match real-world positioning. This creates a clear and intuitive link between the physical space and its corresponding digital infrastructure giving guidance to the operators for any on the ground intervention.



Figure 47 – AR Pipe Model Visualization

Geo-Location and Orientation Integration

To ensure that digital models are correctly aligned with the real environment, the AR application integrates geo-location and orientation data. On mobile and tablet devices, this is achieved directly through their built-in GPS and compass sensors, which provide continuous positioning information.

Sensor Data Visualization

The sensor data visualization system overlays operational values directly onto the pipeline models within the AR environment. In the prototype, sensor readings are primarily retrieved from the backend through a

REST API, which allows for flexible data access and supports filtering options so that users can request specific time windows, sensor types, or anomaly-related measurements for a complete diagnosis of the anomaly. This makes the system adaptable to different inspection scenarios and ensures that only relevant information is displayed in the AR view (see Figure 48). In addition to the backend connection, an alternative implementation has been developed that retrieves sensor values from CSV files, providing a lightweight solution for cases where offline use or simplified data integration better fits the user's needs (granting full info access in all contexts). In both modes, the sensor readings are anchored to the corresponding pipeline segments, enabling users to correlate physical infrastructure with real-time or recorded operational conditions.



Figure 48 – Sensor Data Visualization in Pilot #3 AR App

Anomaly Detection Visualization

The anomaly detection component extends the pipe model visualization by anchoring predicted faults directly within the AR scene. Anomalies are identified through the Neurosymbolic and Active Learning AI services, which analyse recorded CCTV inspection data to detect irregularities along the pipeline infrastructure. These detections are then communicated to the AR application and represented as clear spatial markers, such as pins or spheres, placed at the corresponding locations on the digital pipe model. By interacting with a marker, users can open a popup window that provides additional information about the anomaly type and related metadata. This direct link between AI-generated predictions and their physical context allows inspectors to quickly understand the position of faults/anomalies, improving the efficiency and accuracy of field inspections.

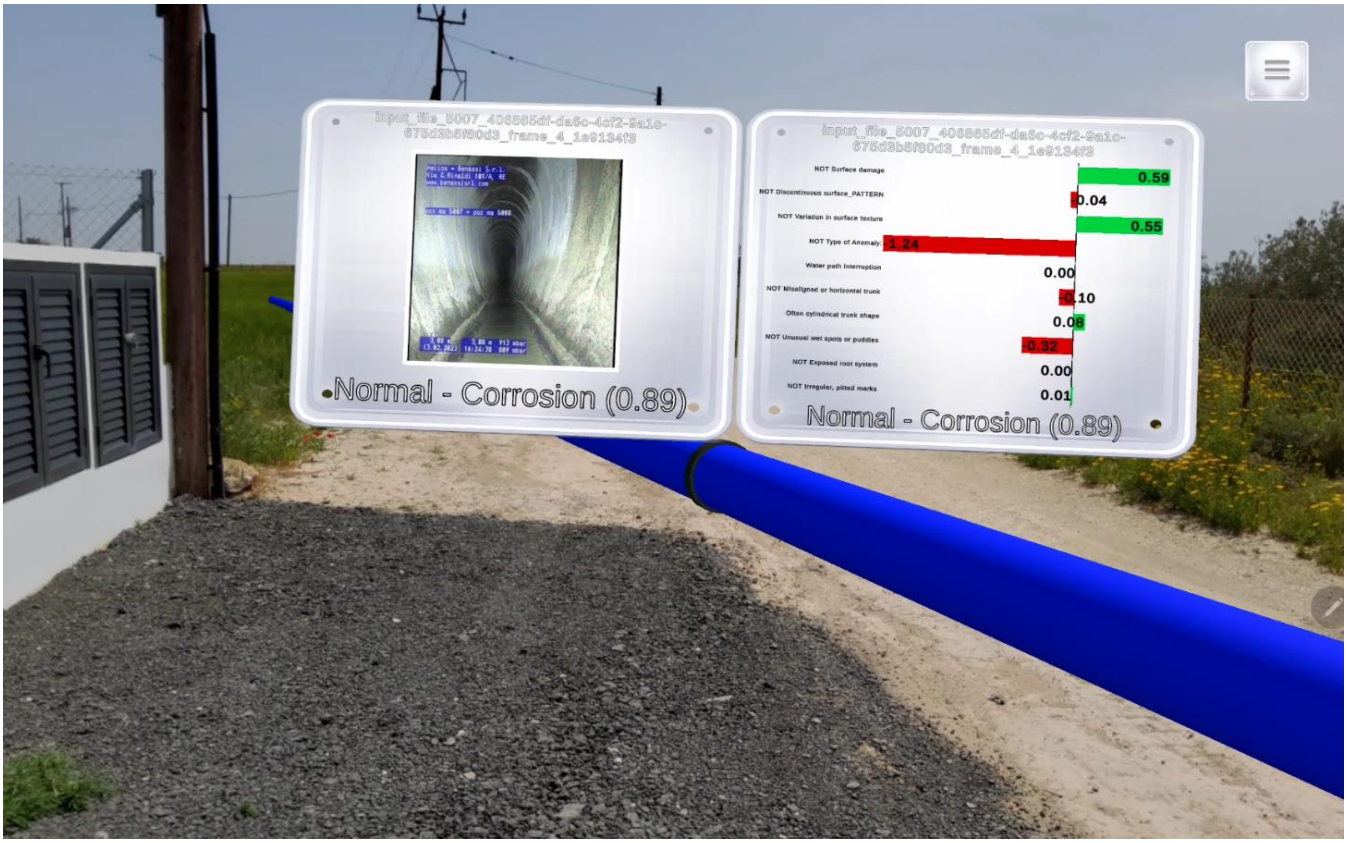


Figure 49 – Visualization of Anomaly Detection in Pilot #3 AR App

AI Chatbot Integration

Another core component of the prototype is the integration of the AI chatbot, which acts as a digital assistant to support workers during inspection, training and maintenance activities. The chatbot supports both voice-based (handsfree) and text-based interaction, allowing users to submit questions or requests either by speaking or through the chat interface. Voice interaction is enabled through Speech-to-Text (STT) and Text-to-Speech (TTS) technologies, while the chat interface provides direct textual interaction with the AI assistant.

The chatbot provides contextual guidance and explanations based on the user's needs and the relevant operational scenario. Responses can include step-by-step instructions/checklists, maintenance and inspection guidance, explanations of detected anomalies, and recommendations for appropriate actions. Where relevant, the chatbot can also provide supporting documentation and visual material, such as related technical documents and images, helping operators better understand the issue and the required procedure. By integrating the chatbot directly into the AR application, the system provides an interactive and support mechanism that complements the visualisation of infrastructure, sensor data, AI findings and recommendations, assisting workers in making informed decisions during training and field activities.



Figure 50 – Chatbot Conversation Window

UI and Interaction System

Finally, interaction within the AR application is supported through a lightweight and modular user interface and popup system (see Figure 51). Following feedback received from EKSO, the application menu and overall UI were simplified and reorganised to provide a more intuitive, accessible and user-friendly experience for field operators. The updated interface reduces unnecessary navigation and presents the most relevant inspection, training and AI-supported functionalities in a clearer and more streamlined manner.

Anomalies, sensor readings, AI explanations and recommendations, training information, and chatbot responses are presented as interactive elements that users can access through touch-based interaction on mobile devices. The chatbot supports both text and voice interaction, providing contextual guidance, explanations, maintenance procedures, recommendations and, where relevant, supporting documentation and images. Additional usability features were also introduced, including a double-tap gesture to switch popup content to a full-screen mode, improving the readability and accessibility of detailed information on mobile screens.

The XR Training Plugin was also adapted to operate on mobile devices, allowing the structured training functionality to be used within the final mobile-based Pilot 3 implementation (see Figure 51). In addition, a Single Sign-On (SSO) mechanism was introduced, allowing the application to use a single authentication token to access the different integrated services and systems. This reduces the need for repeated authentication and contributes to a simpler and more seamless user experience. The SSO approach also supports further simplification of the UI by reducing the need for separate authentication interfaces for individual services.

Overall, the UI has evolved from the initial AR-glasses-oriented interaction concept into a simplified, mobile-focused and user-centred interface for a more easy-to-use and secure deployment on field. The modular design allows additional data sources, AI services, training content and interaction functionalities to be integrated as the Pilot evolves, while maintaining a clear and accessible experience for inspection, maintenance and training activities.



Figure 51 – Pilot #3 AR App UI

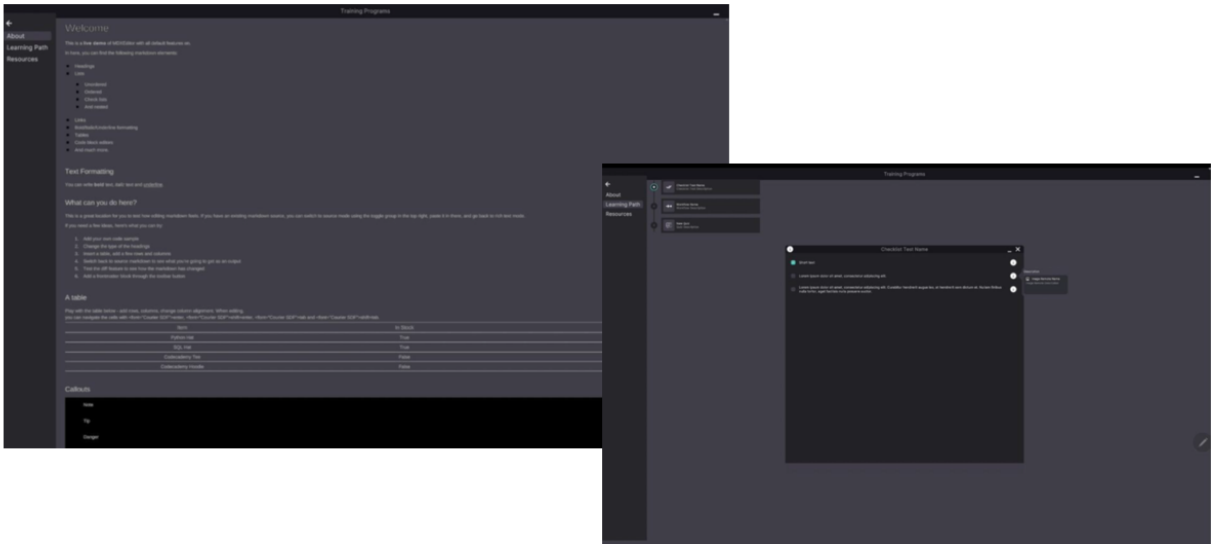


Figure 52 – Pilot #3 XR Training Plugin

4.2.2 XR-enabled AI Assistant

The XR-enabled AI Assistant developed by INNOV supports EKSO's smart pipes pilot by providing context-aware guidance for the inspection, installation and maintenance of water pipes. Operators and trainees ask questions in natural language and receive answers grounded in the technical documentation of the pilot, such as installation manuals, equipment specifications and troubleshooting guides. Compared to the version reported in D6.3, which relied on text-only ingestion and a single retrieval and generation step, the component has evolved into a multimodal, multi-stage assistant. It now presents the relevant figures from the source manuals next to its answers, selects which documents to search based on the user's question, and provides more accurate responses on specific equipment, procedures and error codes. The objective is therefore not only to answer questions from documentation, but to deliver the right information in the form that is most useful to the operator, including the diagrams and photographs on which technical procedures typically rely.

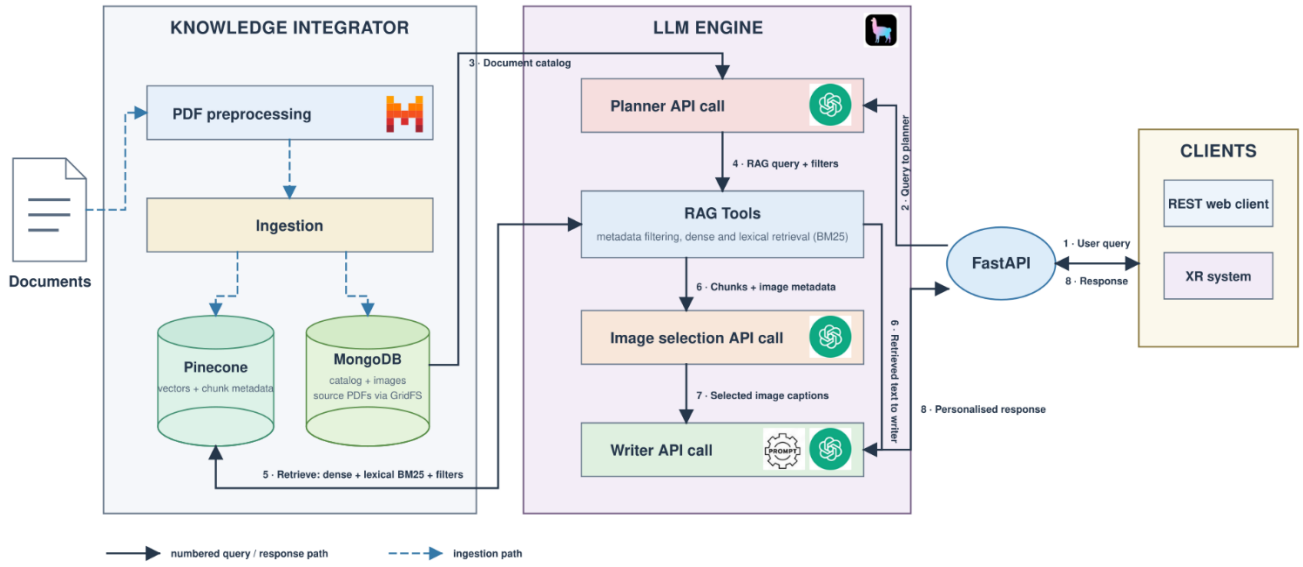


Figure 53 – XR5.0 AI Assistant Overview

The component follows an API-first, service-oriented architecture implemented with FastAPI, separating the client applications from the document processing, retrieval and generation logic. Figure 53 presents the high-level architecture, which consists of the Knowledge Integrator, the LLM Engine and the clients. The Knowledge Integrator is responsible for ingesting the technical documentation. Uploaded PDF manuals are processed by an OCR-based preprocessing step (Mistral OCR) that extracts text, tables and embedded images, while preserving the link between each image, its page and its surrounding content. Text chunks are embedded and stored in a Pinecone vector database together with their metadata. A MongoDB store holds the document catalogue, the extracted images and the original source PDFs. For each document, the system records structured metadata such as pilot, document type, manufacturer and page count, along with an automatically generated summary. The knowledge base is shown in the Knowledge Base view in Figure 54.

The LLM Engine answers each question through a three-stage pipeline, illustrated in Figure 55:

- Planner: receives the user query together with the document catalogue of the active pilot. It identifies the relevant documents, translates them into metadata filters and produces an optimised search query.
- Image selection: examines the retrieved chunks and their associated image metadata, and keeps only the figures that support the answer.
- Writer: generates the final response, grounded in the retrieved content and referring to the selected figures and pages.

Retrieval combines dense semantic search with lexical (BM25) search. Both the optimised query from the planner and the original user query are used, filtered by the planner's metadata, and their results are merged. This allows the system to capture the intent of the question while keeping exact terms such as product names, part numbers and error codes, which semantic search alone tends to miss.

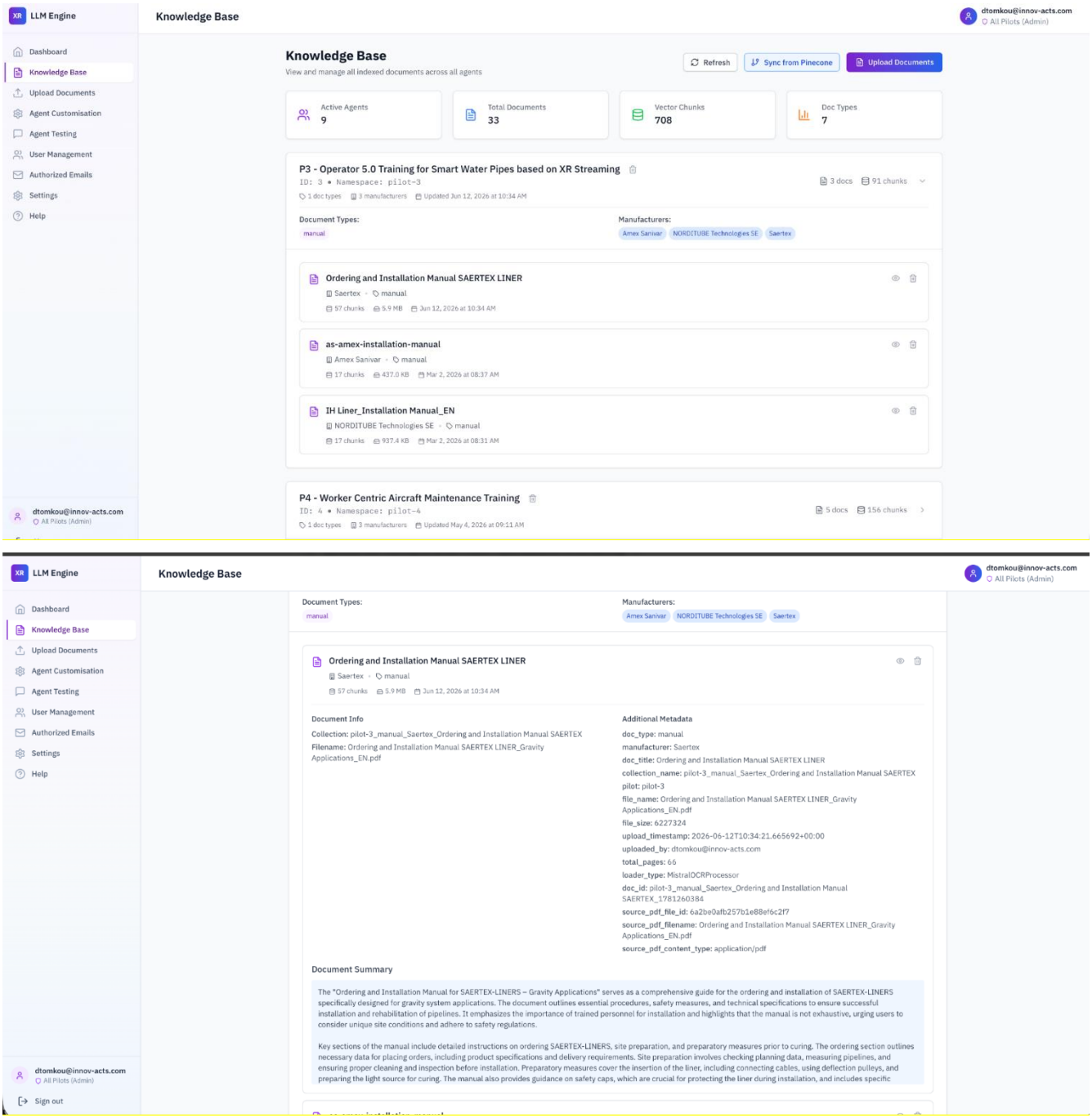


Figure 54 – XR5.0 AI Assistant Knowledge Base

The web application provides the user interface for administrators and operators to upload and manage documents, configure agents and test the assistant. As shown in Image W, answers are presented together with a references panel that lists the selected images with their source document and page, and allows the original PDF to be opened. Each pilot is configured as a separate agent with its own namespace, documents and system prompt, so the same engine serves multiple XR5.0 pilots without changes to the underlying implementation. For integration with XR devices and external systems, the REST API (Figure 57 and Figure 58) has been extended with endpoints for chat, document and namespace management, custom pilots, system prompts, image serving and source document retrieval. XR applications therefore receive the same

multimodal output as the web client through standard HTTP calls, while remaining independent of the retrieval and language-model implementation.

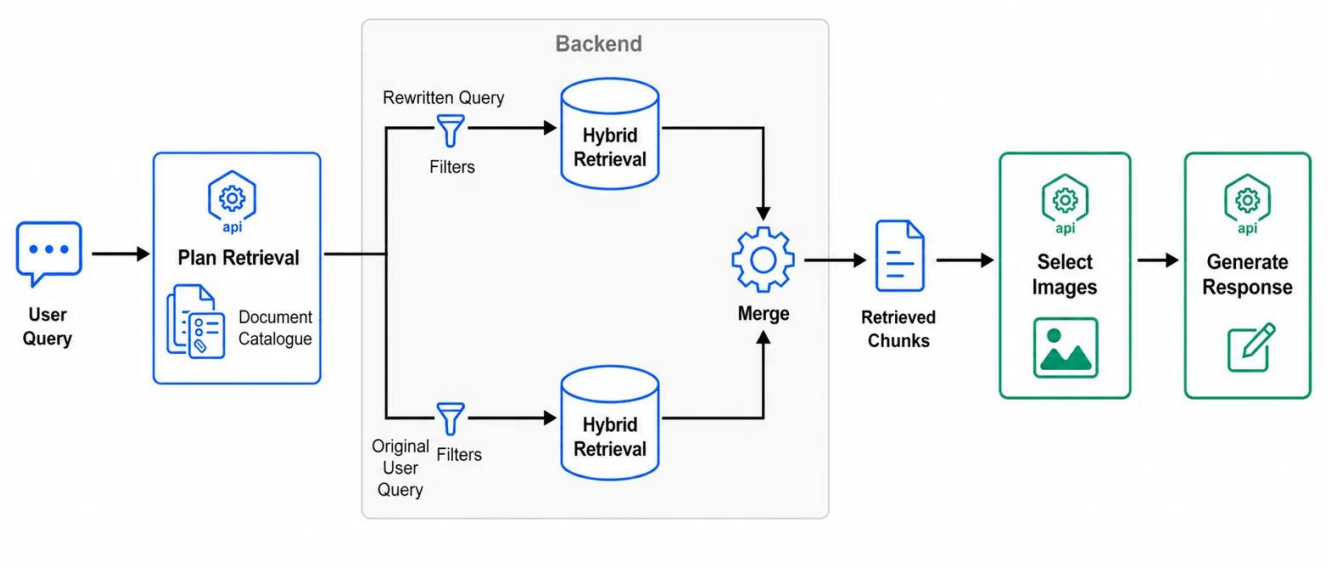


Figure 55 – XR5.0 AI Assistant Query Pipeline

The updated component demonstrates the technical feasibility of turning image-rich industrial documentation into a searchable, multimodal knowledge base with minimal manual effort. Its pilot-agnostic design and documented API demonstrate interoperability across both pilots and client applications. Its human-centric value lies in giving field operators concise, traceable answers accompanied by the visual material they need to carry out a procedure safely and correctly.

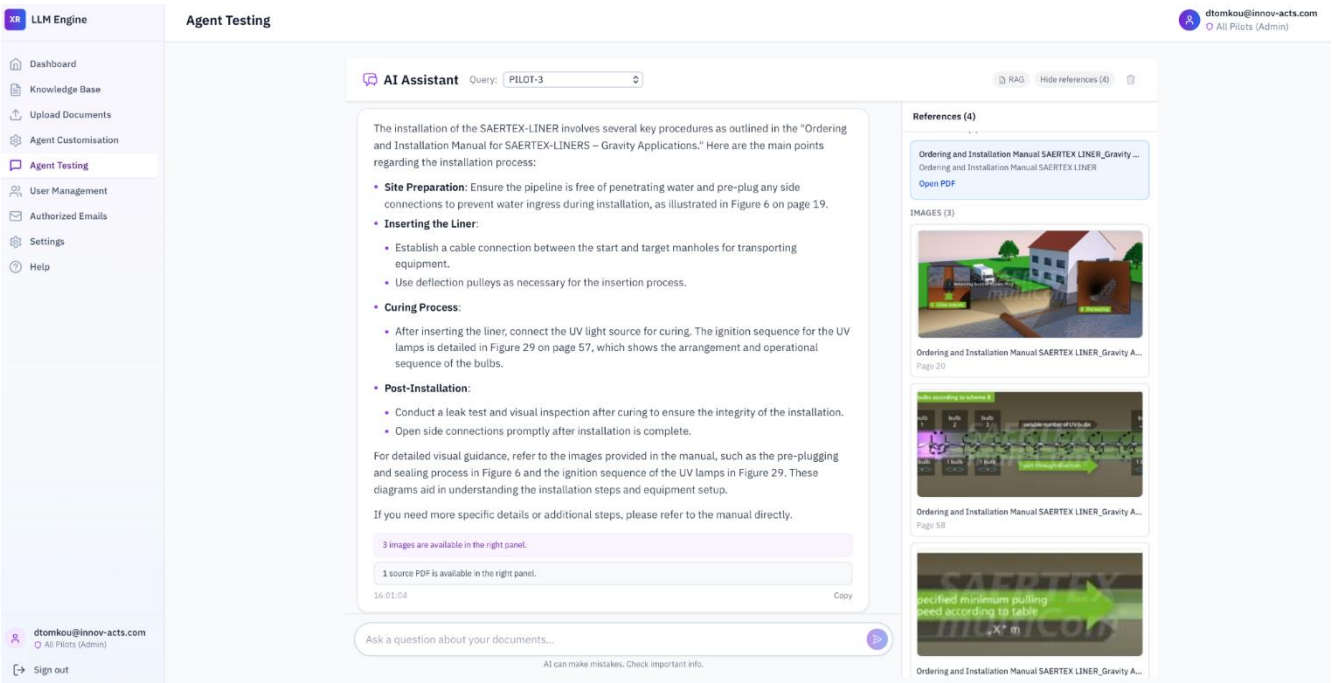


Figure 56 – XR5.0 AI Chat View

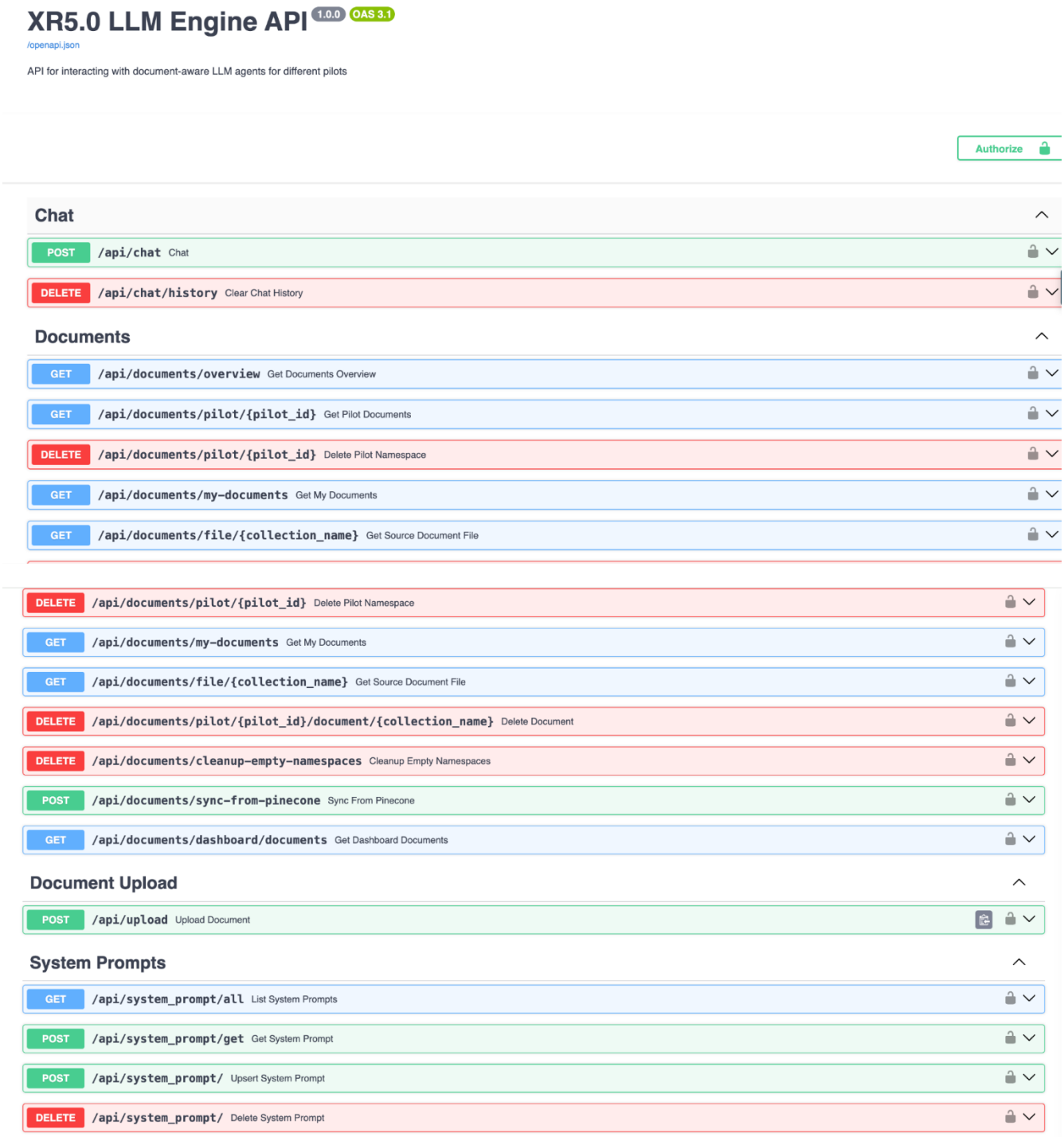


Figure 57 – XR5.0 API Overview A.

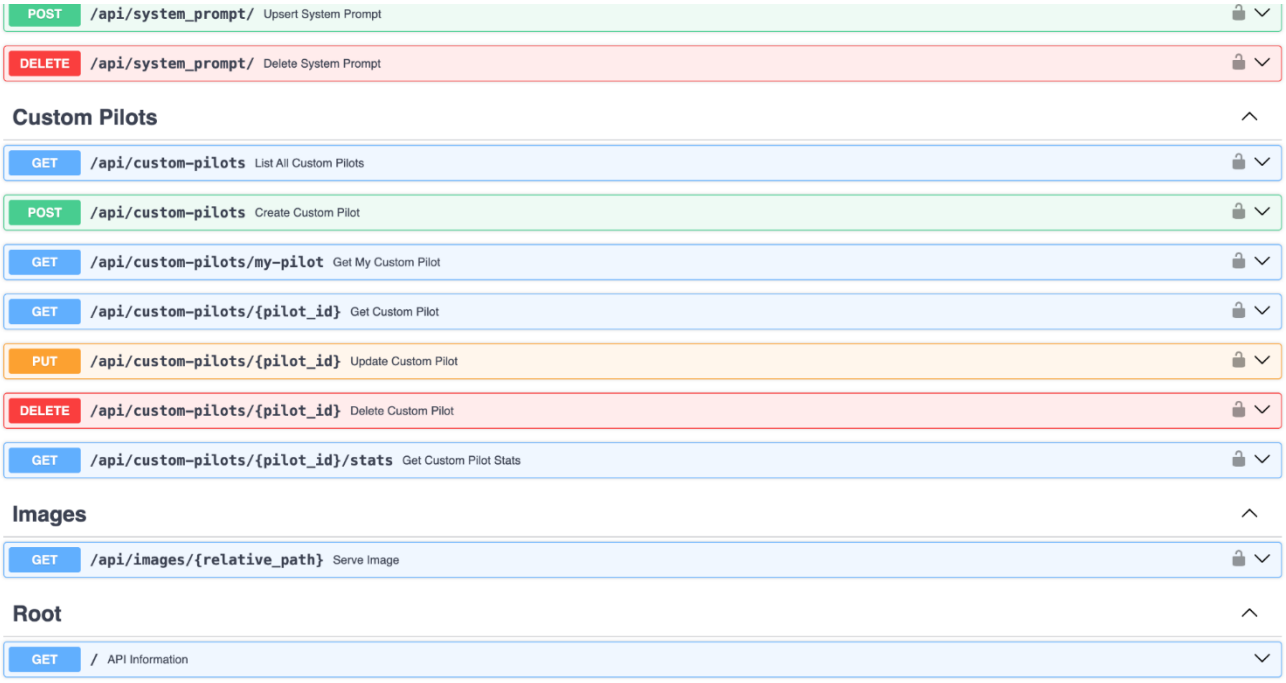


Figure 58 – XR5.0 API Overview B.

4.2.3 Active Learning and Neuro-symbolic AI

Active Learning Component

The Active Learning (AL) component, developed by UPRC supports the selection of informative training examples and the involvement of human experts in improving AI models. Sampling methods identify examples with high potential information gain, helping to prioritise expert review and annotation.

The component assesses the uncertainty of predictions produced by the Neurosymbolic AI (NSAI) model for CCTV-based pipe anomaly detection. Predictions exceeding a predefined uncertainty threshold are referred to human experts for validation or correction (Figure 59). Experts can also revise AI outputs on demand through the *ai_correction* service. These corrections trigger the *ai_retraining* service, which updates the underlying training datasets and initiates a new training cycle. This feedback loop connects expert judgement with NSAI model development, strengthening human oversight and supporting iterative improvements in anomaly detection.

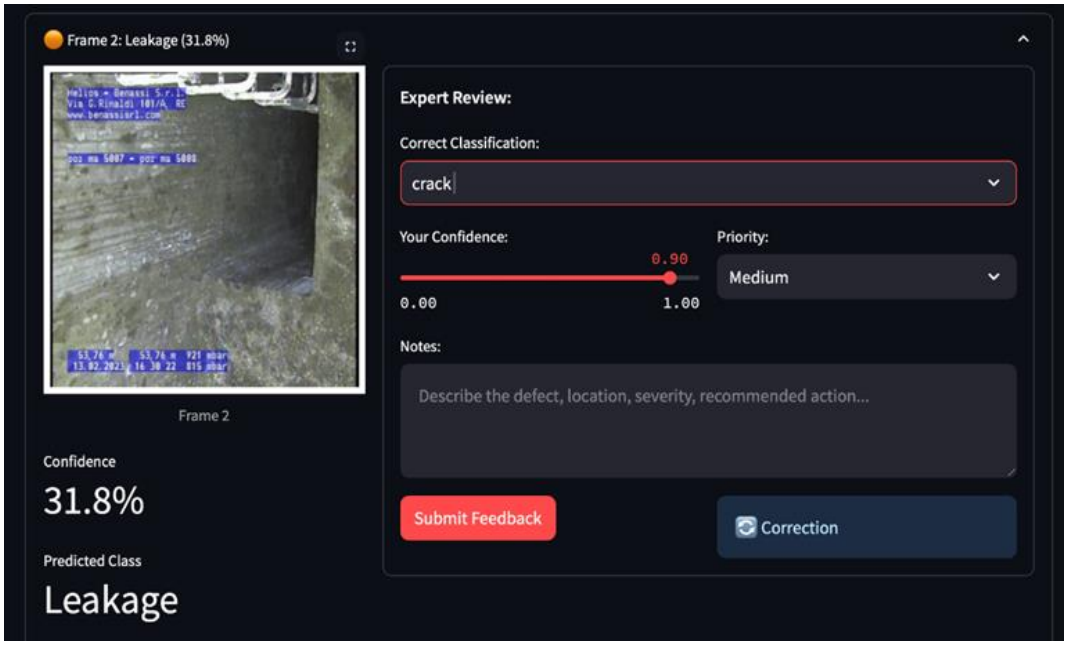


Figure 59 – Active Learning Integration in Pilot #3

Neurosymbolic AI Component

The Neurosymbolic AI (NSAI) component, developed by UPRC supports anomaly detection in CCTV footage of water-pipe inspections. It combines deep neural networks with symbolic reasoning to identify visible characteristics associated with potential defects and provide interpretable explanations for its predictions. Detection results and associated explanations are made available to operators through the mobile AR application.

As illustrated in Figure 59, the model identifies a potential defect based on concepts such as "variation in surface texture" and "surface damage". The contribution of these human-understandable concepts to the prediction can be represented through rules within the symbolic reasoning framework. This approach helps operators understand the basis of detected anomalies and supports expert validation and correction through the Active Learning component.

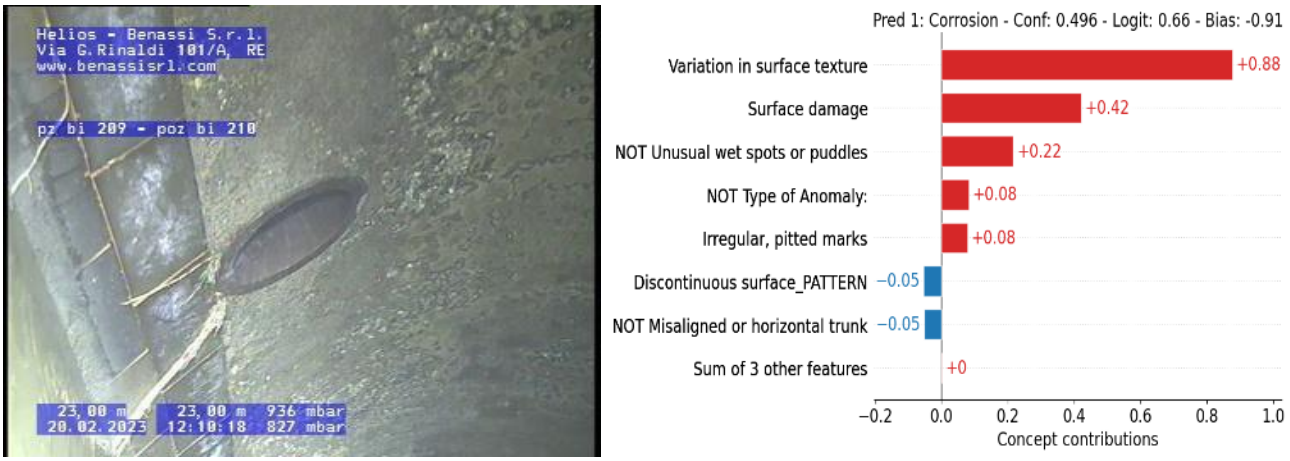


Figure 60 – NSAI Integration in Pilot #3

4.2.4 Explainable AI

The eXplainable AI (XAI) capabilities developed by UPRC help operators understand and assess AI-generated anomaly detections. The approach combines inherently interpretable, or glass-box, models with

post-hoc explanation methods, supporting the presentation of relevant explanations within the mobile AR application.

Concept Bottleneck Models (CBMs) are integrated into the Neurosymbolic AI (NSAI) component to connect visual features extracted from CCTV inspection footage with human-understandable concepts, such as surface damage or texture variations. These concepts support the identification of potential pipe anomalies and can be translated into formalised rules within the symbolic reasoning framework. By combining concept-based predictions with logical reasoning, the component provides insight into the characteristics underlying each detection and supports expert review. Figure 61 illustrates the CBM output and the associated explanation of an AI prediction.

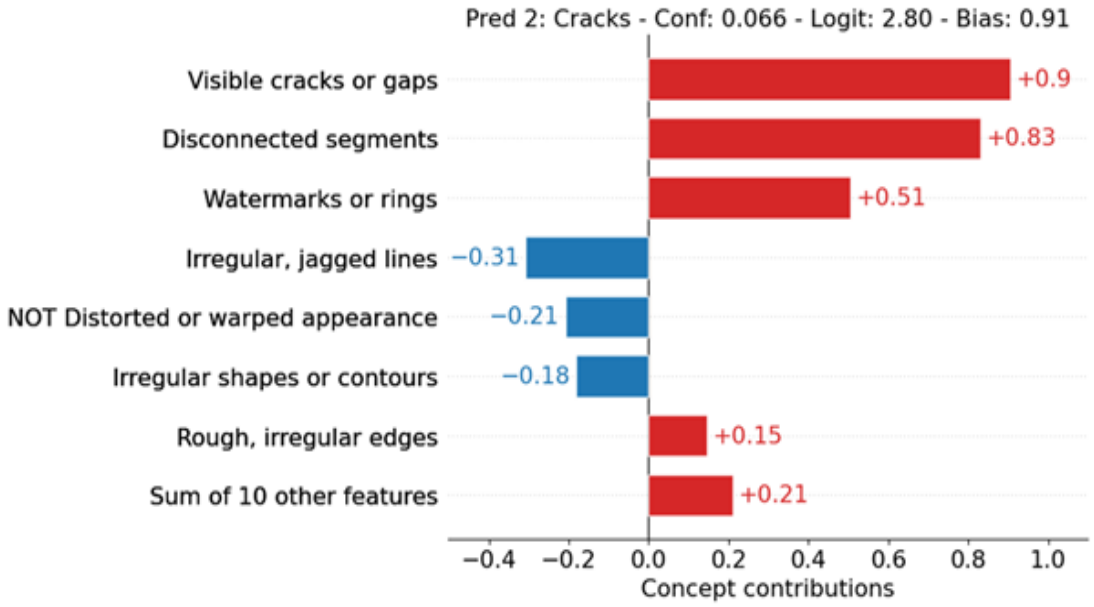


Figure 61 – XAI Integration in Pilot #3

Complementing this approach, LIME and Grad-CAM provide post-hoc explanations for black-box models without requiring their retraining. Further details on these methods are available in D4.1. Both methods are accessible through a separate API and can be requested on demand, allowing experts to decide when additional explanations are needed. This selective access helps keep the inspection interface focused and limits unnecessary visual content that could increase cognitive load or cause confusion.

4.2.5 Smart Pipes API

This subsection provides an introduction to the Smart Pipes API solution provided by Siemens for the EKSO pilot. The Smart Pipes system is a comprehensive solution for monitoring water infrastructure and detecting anomalies, leveraging advanced ETL (Extract, Transform and Load) workflows, real-time data processing, and geospatial analytics to detect and localize anomalies along water distribution networks.

Furthermore, the Smart Pipes system is a service solution designed to facilitate partners in developing their own solutions for the EKSO pilot's use cases. It achieves this by enabling a streamlined integration process through its API with other applications, such as the "AR Application" described in Subsection 4.2.1.

The architectural design of the Smart Pipes application depicted in Figure 62, is structured into two core components as follows:

- ETL workflows - a flexible and modular framework that handles the core processing logic.

- The Smart Pipes Interfaces - the storage solution and the API endpoints through which the system is accessed.

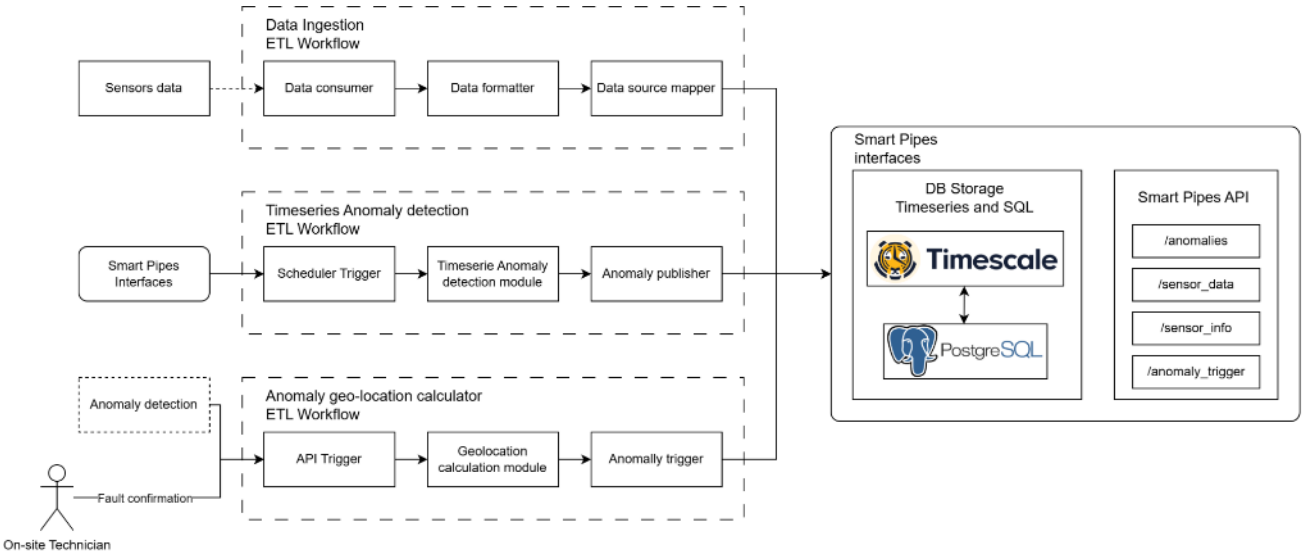


Figure 62 – High Level Architecture of Smart Pipes API

ETL workflows represent the main processing logic of the application. Each follows the pattern of consuming, processing and publishing data, and the modular design allows further workflows to be developed and integrated as operational requirements evolve. The workflows are as follows:

- **Data Ingestion** - consumes sensor measurements and applies pre-processing so that granularity and data-point integrity are consistent, then formats and publishes them to the storage layer. In the current version this operates on the static datasets provided by the EKSO pilot, loaded in batches with tracking of the files already processed so that repeated imports do not duplicate data; a generator produces synthetic readings for development and testing. In an operational deployment the same path would receive streaming measurements from field devices.
- **Time-series Anomaly detection** - queries the most recent measurements from the storage layer and determines whether sensor behaviour is consistent with normal operation or should be flagged as anomalous, storing any anomaly detected.
- **Anomaly geo-location calculation** - computes the position of a detected anomaly along the network from distance measurements and the pipe infrastructure identifier, and stores the resulting anomaly metadata. It is triggered through the API rather than on a schedule, so that a calculation can be requested at the point it is needed.

The Smart Pipes Interfaces provide the data access layer, comprising both storage and the API. Storage is a single PostgreSQL database with the TimescaleDB extension enabled, which holds the time-series sensor measurements efficiently alongside the sensor, pipe network and anomaly records and the relationships between them.

The API is organised into four groups of endpoints, summarised in Table 6: sensor and network metadata, time-series measurements, anomaly retrieval and triggering, and administrative operations. Time-series queries support filtering by sensor name, adjustment of granularity and data reduction, with selectable strategies governing how these are applied. Results may be returned either inline within the response or as a stream, which allows a consuming application such as the AR Application described in Subsection 4.2.1 to choose the delivery mode appropriate to its use. Repeated queries are served from a time-based cache, and a group of administrative endpoints exposes cache statistics, selective and full invalidation, and control of cache monitoring.

The system is container-deployed: the database, the API service and the data importer are defined as a single topology, so that installation consists of supplying the required configuration and starting the stack. The ETL orchestrator is deployed alongside it.

Access to the API is authenticated through the XR5.0 Hub. The session token accompanying each request identifies the user and the application making it, and the deployment accepts only the organisations and applications configured for it. A consuming application such as the AR Application therefore reaches the service with the credential the operator already holds from signing in to the platform, rather than through a separate credential issued and maintained for each integration.

Table 6 – SmartPipes API Endpoints

Method	**Endpoint**	**Purpose**
GET	/ekso/sensor/info	Sensor infrastructure metadata.
GET	/ekso/chain/info	Sensor chain metadata for the pipe network.
GET	/ekso/sensor/query	Time-series measurements, with name filtering, granularity adjustment and data reduction.
GET	/ekso/sensor/streaming	The same measurements are delivered as a stream rather than inline.
GET	/merge_data/anomalies	Retrieve detected anomalies, with filtering.
POST	/merge_data/anomaly_trigger	Trigger the anomaly geo-location calculation.
GET, POST	/ekso/admin/cache/…	Administrative group: cache statistics, selective and full invalidation, and monitoring control (six endpoints).
GET	/health	Service status. No credential required.

4.3 Component Usage Matrix

Table 7 gives an overview of the pilot-specific and XR5.0 technical components used in Pilot #3 Full Prototype.

Table 7 – XR5.0 Technical Components in Pilot #3 Full Prototype

XR5.0 Technical Tasks	**Technical Component**	**Technical Partner**	**How is the Component Used?**
T3.1 Specifications for Worker Digital Images	N/A		
T3.2 Workers' Data Collection & Analytics	N/A		

T3.3 Workers' Digital Twins	N/A		
T3.4 Personalized XR Content Development	N/A		
T3.5 Adaptation of Personalized XR Content	N/A		
T4.1 Human-centered XAI Models	XR5.0 XAI Suite for Glass-box Models and Post-hoc Methods	UPRC	Provide explanations for the NSAI anomaly detection in CCTV inspections and allows experts on human understandable concepts.
T4.2 XR-enabled AL & NSAI Models	AL & NSAI-based Anomaly Detection	UPRC	Detects anomalies in CCTV tube inspection videos, offering explainable results by linking visual features like surface texture variations or damage to rule-based predictions of defects. Active Learning supports expert validation and correction of uncertain predictions and initiates dataset updates and model retraining.
T4.3 Generative AI Models	AI Chat/Voice Assistant	INNOV	Provides contextual inspection, maintenance and training guidance through text and voice interaction, using EKSO's technical documentation to support instructions, explanations and recommendations.
T4.4 Visualization of XAI Explanations	XAI Visualizations in the AR App	CYENS	Presents explanations of AI-generated anomaly detections within the mobile AR application to support operator understanding and assessment.
T4.5 Visualization of AI Recommendations	AI Visualizations in the AR App	CYENS	Provides visualizations of the recommendations from the NSAI anomaly detection and the AI chat/voice assistant in the AR app.
T5.1 Training Material	Training Materials for Pilot 3	IML	Training materials provide XR-based content based on traditional materials for equipment and safety checklists, access and procedure workflows.
T5.2 Cloud-based XR Training Asset Repository	N/A	HOLO	
T5.3 Hololight Hub	N/A	HOLO	
T5.4 Training Programs	XR-based Training	IML	Training programs provide a basic program for maintenance of water pipes

	Programs for Pilot 3		
Pilot-specific Components	ETL Workflow and Smart Pipes API	SIE	Enables real-time monitoring and anomaly detection in water infrastructure by combining advanced ETL workflows with geospatial analytics.

Table 8 maps the technical components for the three use cases – UC3.1.1, UC3.1.2 and UC3.2.1. The AR application serves as the main interface for operators and will be used across all three use cases. Active Learning and Neurosymbolic AI are primarily applied to the analysis of CCTV videos for detecting pipe anomalies, with Neurosymbolic AI also being extended for use in training programs under UC3.2.1. A GenAI/LLM-based chat engine acts as a general-purpose assistant for pipe maintenance and inspection, supporting all three use cases. Meanwhile, the ETL Workflow and the Smart Pipe API are implemented in UC3.1.1, specifically to deliver geolocated anomaly points to the AR application based on vibration sensor data.

Table 8 – Mapping of Pilot #3 Technical Components to Pilot Use Cases in Full Prototype

	UC 3.1.1	*UC 3.1.2*	*UC 3.2.1*
AR Application	X	X	X
Anomaly Detection with Active Learning		X	
Neurosymbolic AI Module		X	X
LLM Chat Engine	X	X	X
ETL Workflow	X		
Smart Pipes API	X		
Training Platform Authoring Tool			X

4.4 DISCUSSIONS

Pilot #3 demonstrated significant progress in integrating AR, AI and XR-enabled training technologies for smart pipe inspection and maintenance. The CYENS mobile AR application communicates with the Smart Pipes backend and AI services, enabling the visualisation of pipe infrastructure, sensor data, anomaly detection outcomes, XAI explanations and AI recommendations within the physical environment. The application also integrates the AI chatbot, supporting both text- and voice-based interaction and providing contextual guidance, documentation and visual material to assist operators during inspection and maintenance activities.

Following continuous feedback from EKSO (direct operators experience on field), the solution was transitioned and fine tuned from the initially envisaged AR-glasses-based implementation to a mobile AR application. The user interface and menu structure were consequently simplified and reorganised to provide a more intuitive and accessible experience for field operators needs and necessities. Additional usability features, including full-screen popup interaction through double-tap, were introduced to improve information accessibility on mobile devices. The IML XR Training Plugin was also adapted, integrated and tested for mobile operation, extending the training capabilities of the pilot.

Several technical challenges were addressed during the implementation, through a continuous feedback/feedforward process with users, particularly regarding spatial alignment, GPS and compass accuracy, mobile device limitations and the integration of multiple AI and backend services. The use of GPS- and marker-based localisation provides a practical approach for positioning digital pipe infrastructure and anomaly information in the physical environment. Optimisation of the UI, data processing and visualisation was also required to ensure that AI-generated information could be presented without unnecessarily increasing user cognitive load.

The findings demonstrate the value of combining contextual AR visualisation with AI-supported decision-making and structured training. XAI visualisations help operators understand AI-generated anomaly detections, while AI recommendations and the Generative AI chatbot provide a powerful support and additional guidance easily accessing relevant operational information during operations. The NSAI and Active Learning components further support human-in-the-loop processes, allowing expert feedback and validation to contribute to the continuous improvement of AI models.

Further improvements will focus on continued optimisation of mobile performance, refinement of the user interface and interaction mechanisms, and validation with end users under representative operational scenarios. Additional testing will assess the stability, usability and effectiveness of the integrated AR, AI, chatbot and training functionalities. Partner collaboration will continue to refine the integration of the different components and ensure alignment with the technical objectives of WP4 and the broader XR5.0 project goals and for a real-life powerful device.

5. PILOT 4 FULL PROTOTYPE

Pilot #4 – T6.5 Personalized Aircraft Maintenance Training, led by IML, develops immersive, AI-enhanced training applications for the maintenance of the Wing Anti-Ice Valve (WAIV). Building on the early prototype reported in D6.3, the full prototype combines interactive XR training materials, conversational AI, aircraft-component recognition, worker digital twins, and personalised content adaptation. These capabilities aim to support junior Aircraft Maintenance Technicians (AMTs) in learning and performing maintenance procedures, improving procedural understanding, supporting error prevention, and facilitating the transfer of expert knowledge. The pilot addresses two use case:

- **UC4.1 – Virtual Training:** A mixed reality application that combines the XR5.0 Training Platform with interactive 3D models, procedural guidance, conversational AI, and aircraft-component recognition to support training in WAIV removal and installation.
- **UC4.2 – AMT Digital Twin:** A virtual reality application that uses recorded expert movements and live trainee monitoring to provide visual guidance, assess task execution, and deliver corrective feedback during WAIV maintenance training.

Pilot #4 Full Prototype has been developed collaboratively by TAP, IML, UNP, IPIA, SIE, UPRC, and SUPSI, following the use case co-creation activities documented in D6.1. The key technical components have been developed including AI-assisted training environments, Human Digital Twins (HDT) consisting in a shadow operator based on real-time feedback systems, immersive VR/AR content, and stress monitoring through wearable sensors. Since D6.3, the technical developments have extended the conversational assistant with retrieval-augmented generation, enabling answers grounded in training documentation and accompanied by source references. The aircraft-component detection framework now supports image capture and annotation, expert validation and correction, Active Learning, on-demand explanations, and controlled evaluation and deployment of updated models. The earlier Shadow Operator concept has evolved into Worker Shadowing and Monitoring, combining expert-motion replay with phase-based assessment of trainees' movements and procedural milestones. The full prototype also incorporates heart-rate collection to support worker-state assessment and an Adaptation Engine for personalising the presentation of training content.

The architecture is centred on the XR5.0 Training Platform, which supports the authoring, storage, management, and delivery of training materials and programmes. Training managers prepare content through the Training Programs Authoring Tool and store it in the cloud-based XR Training Asset Repository. Hololight Hub manages access to the XR applications and orchestrates their cloud execution and streaming to Meta Quest 3 headsets. Interactive 3D model exploration and step-by-step workflows provide spatial and procedural guidance for WAIV maintenance.

The training applications connect to complementary AI and worker-support services. The conversational AI services process training documentation into programme-specific knowledge bases and provide spoken and visual responses within the XR environment. The computer vision backend supports component detection, human feedback, iterative learning, and explanations of individual detections. The worker digital twin processes head and hand tracking data through the XR5.0 Clawdite Platform, while physiological data collection provides additional information on the operator's state. Using training content and worker context, the Adaptation Engine returns presentation decisions that the XR interfaces apply to tailor the learning experience. Figure 63 presents the logical architecture of the Pilot #4 Full Prototype.

.

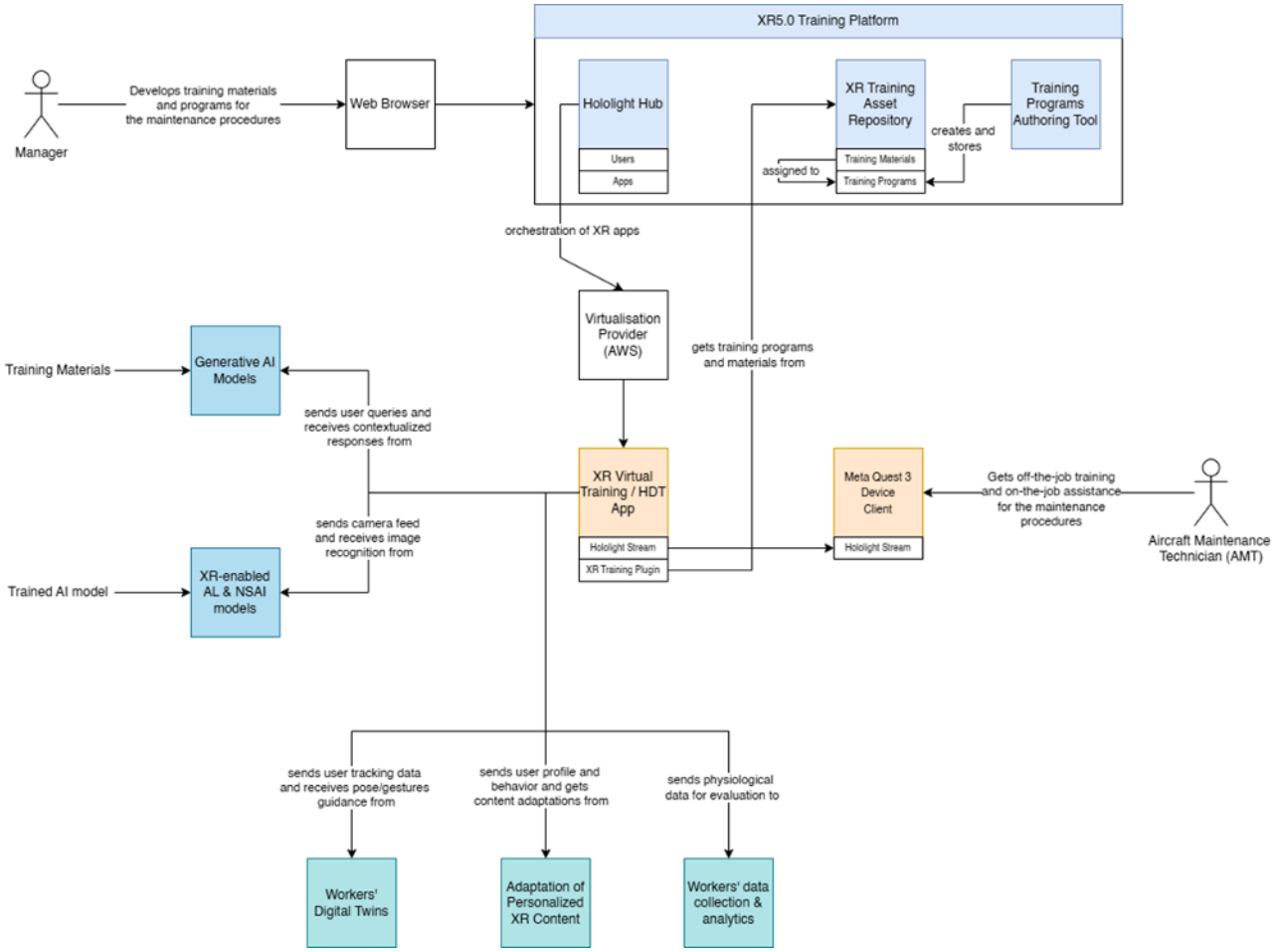


Figure 63 – Pilot #4 Architecture Diagram

5.1 PILOT-SPECIFIC INFRASTRUCTURE

Pilot #4 infrastructure is the following:

- Shopfloor at TAP including aircraft and WAIV device
- 3D models of the aircraft and WAIV device
- Hardware:
  - Meta Quest 3 XR device
  - Pixel Watch 2
- Software:
  - Unity-based XR Virtual Training / HDT Application
  - XR5.0 Training Platform
  - Hololight Hub Meta Quest Application
  - XR5.0 Clawdite Platform (workers' digital twins)
  - XR5.0 Activity Metrics Mobile Application

5.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies

5.2.1 XR-based Training Program and Materials

In aircraft maintenance training, technicians typically rely on digital manuals and videos, instruction, and hands-on practice to learn the procedures for maintaining and repairing critical aircraft components such as the Wing Anti-Ice Valve (WAIV). Training sessions may be conducted both in on-the-job and off-the-job contexts to learn how to disassemble and repair in controlled environments before applying these skills in real-world scenarios.

The XR-enabled Training Program developed by IML enhances traditional aircraft maintenance training by introducing several features that contribute to support the learning method and that will aim to improve training retention. With XR streaming, immersive XR experiences are not dependent on processing capacity of the devices, enabling flexible and scalable deployment across different locations.

The conversion of training materials (PDFs, videos, manuals) into XR-compatible formats transforms passive content into interactive and engaging training programs as shown in Figure 64.

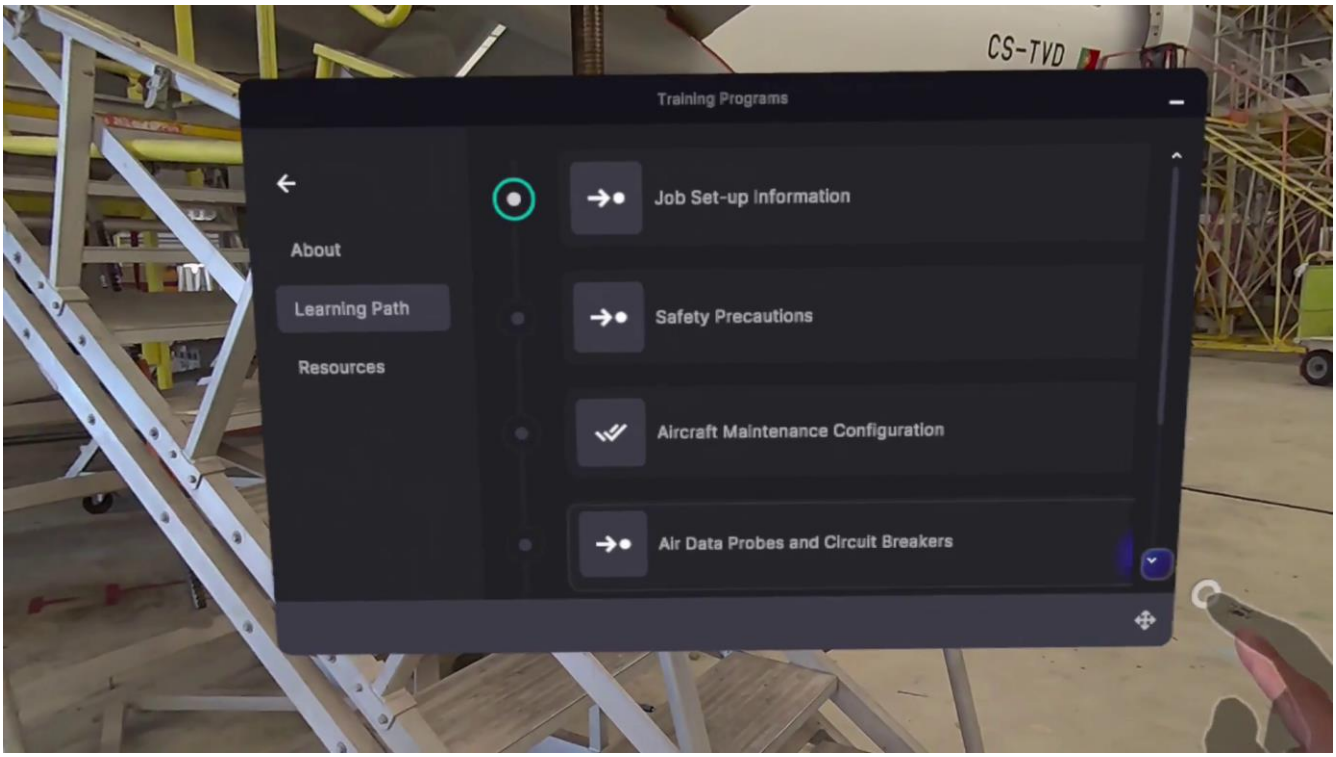


Figure 64 – XR Training Program Outline

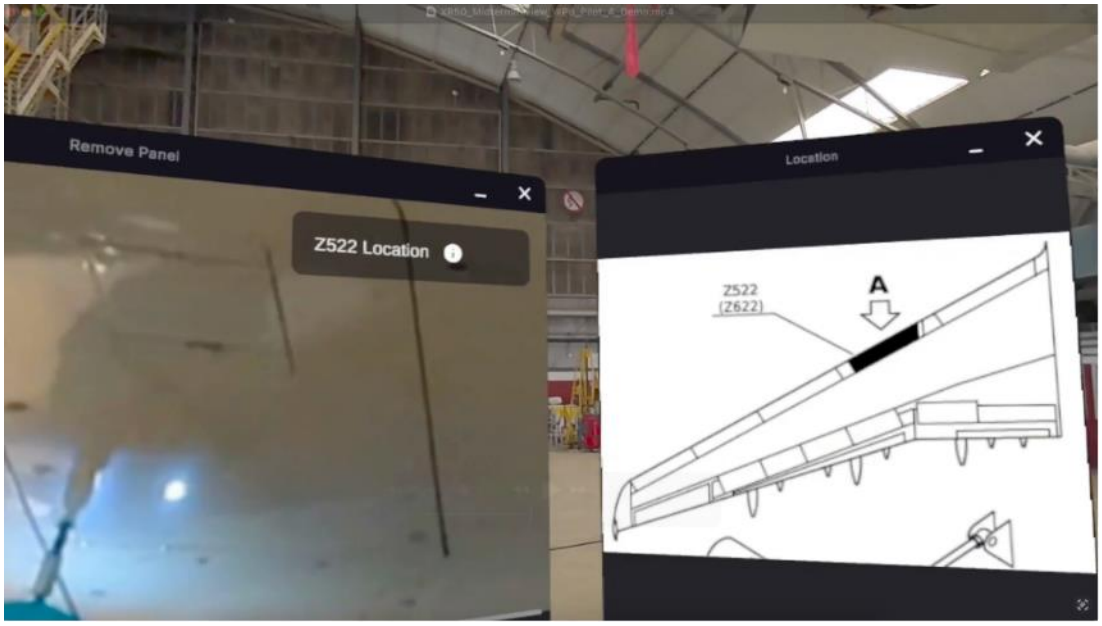


Figure 65 – Training Program using the XR Versions of Training Materials

5.2.2 Hololight Hub

In Pilot #4 Hololight Hub provides the cloud infrastructure for hosting, managing, and streaming the XR training applications to users' devices. It centralises the administration of users, applications, and configuration settings, while orchestrating application execution on AWS infrastructure. Together with Hololight Stream, it supports secure, low-latency delivery of immersive training experiences to Meta Quest headsets.

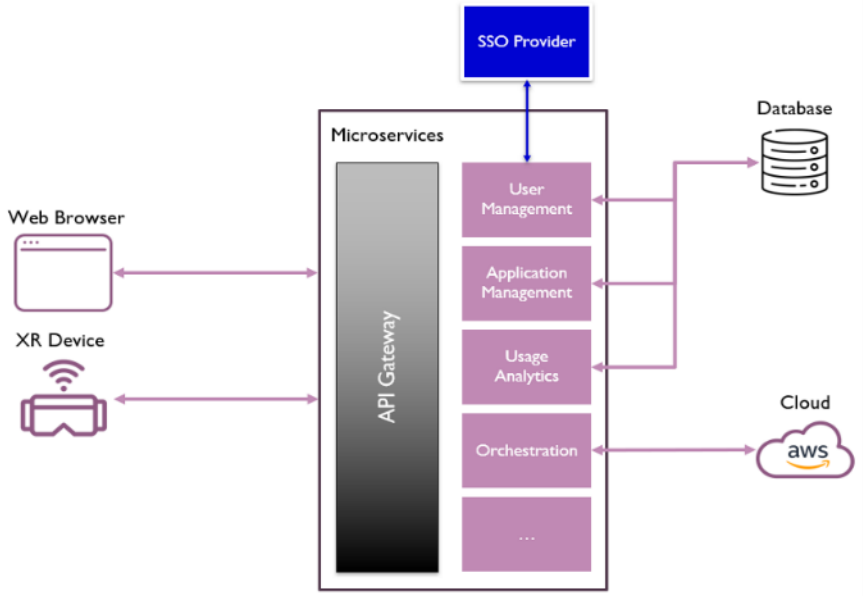


Figure 66 – Hololight Hub Architecture

The architecture follows a microservices approach, with an API Gateway routing requests to services responsible for user and application management, analytics, and resource orchestration. Single Sign-On

(SSO) supports authenticated access, a database maintains platform data, and AWS virtual machines provide the processing resources required for XR application execution and rendering.

5.2.3 Training Material: 3D Model Explorer Addon

The 3D Model Explorer by IML is a core XR training component that transforms static 3d CAD assets into interactive learning experiences. In the TPAT, managers upload 3D models to the XR Asset Repository, where the importer auto-generates exploded views of 3d model. Authors can customize parts visualization and identification. XR-based visualization allows AMT technicians to explore and conduct exploded views of 3D models such as those involving the WAIV device in on-the-job settings.

In use, operators open the Explorer from a training program and freely explore and identify components, toggle exploded views and highlight parts of the model.

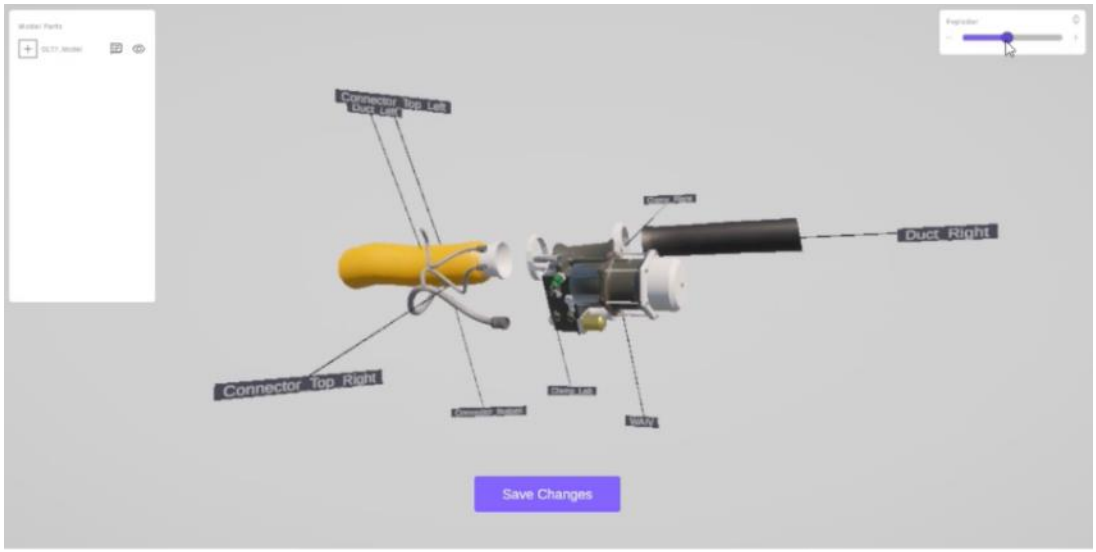


Figure 67 – 3D Model Explorer Interface

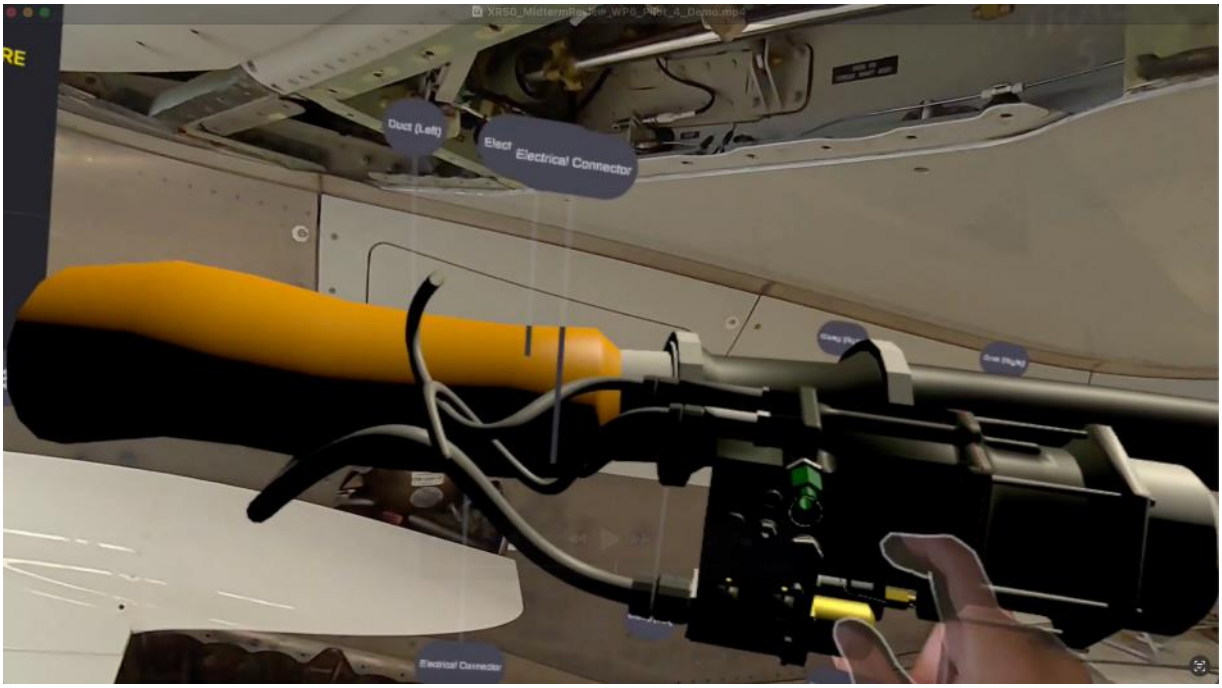


Figure 68 – 3D Model Explorer View

5.2.4 Training Material: 3D Workflow Tool

The 3D Workflow Tool by IML enables the creation of step-by-step, model-based procedures that combine spatial guidance with instructional logic. Within the XR application, developers import a 3D asset and build a workflow as an ordered sequence of steps, each linked to specific parts or subassemblies. For each step, authors can define animations (e.g., highlight), add procedural notes.



Figure 69 – 3D Model Workflow XR View

5.2.5 XR-enabled AI Assistant Addon

The XR-AI Assistant provides contextual support to operators during immersive training sessions. Integrated into the XR5.0 Training Platform, it enables operators to submit natural-language questions within the XR environment and receive responses grounded in the training documentation associated with the active programme. During the authoring phase, training managers configure the assistant through the Training Platform Authoring Tool (TPAT) by associating relevant procedural documentation with the corresponding AI-enabled training activity. Once deployed, operators can ask questions without leaving the immersive session, while the assistant retrieves the relevant information and presents the response within the XR environment. A captured interaction with the AI Assistant within the XR environment is illustrated in Figure 70.

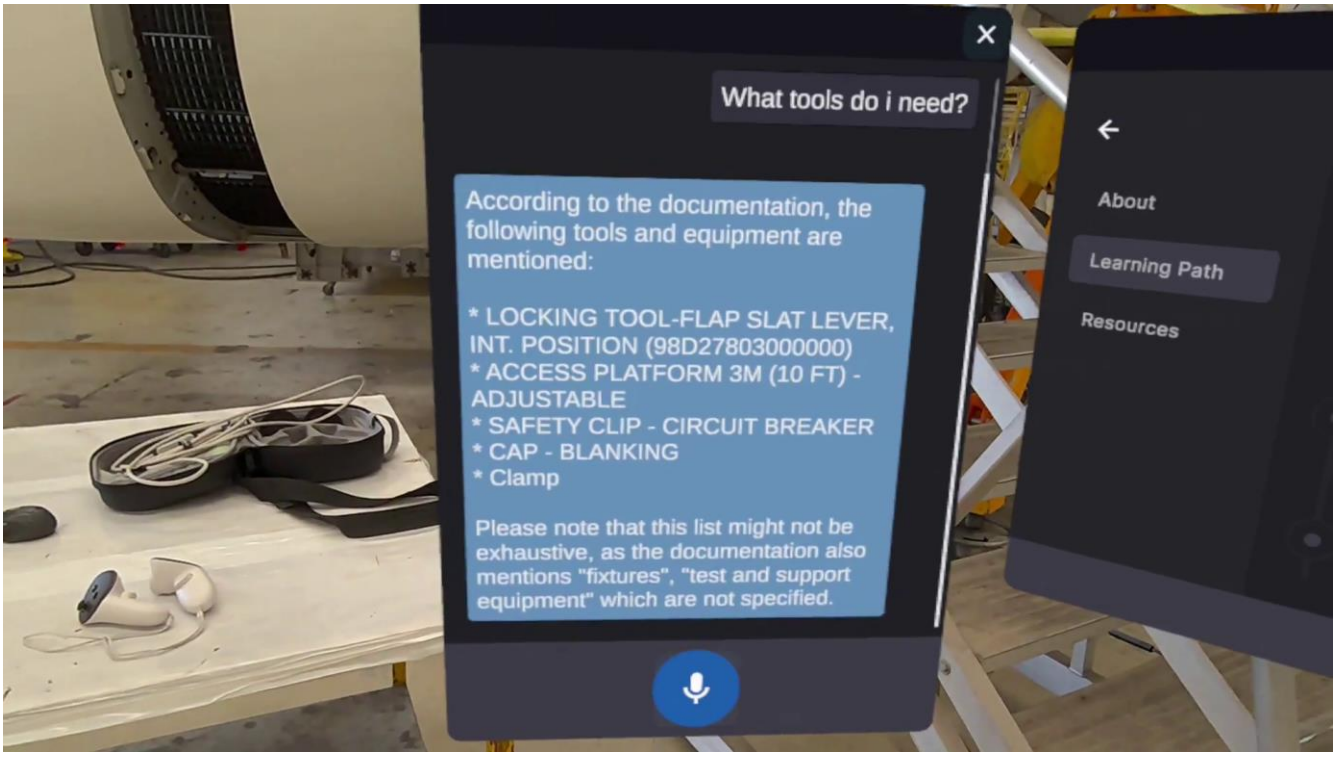


Figure 70 – XR-enabled AI Assistant Addon

The capability is implemented through two interoperating SIE services. The AI Voice Assistant provides the speech recognition, conversational interaction, and speech synthesis functions, while the DataLens Conversational AI Framework processes and retrieves information from the associated documentation. Together, these components form a voice-driven retrieval-augmented generation pipeline that provides contextually grounded answers during the training session. The integration of both services within the Training Platform is shown in Figure X. The AI Assistant has been implemented within the XRTP and is being progressively integrated with the pilot use cases during the current project phase.

When procedural documentation is associated with a training activity in this way, DataLens processes it once into a knowledge base for that programme. Each document is converted with Docling, which extracts its structure — text, tables and figures — and every element is then described and turned into a vector embedding, so that it can be retrieved by meaning rather than by keyword alone. The embeddings are held in a vector database and the source files and extracted assets in object storage, so that no document processing takes place while an operator is in session.

During a session, the AI Voice Assistant handles the spoken exchange. A question captured in the XR environment is transcribed by a speech-to-text (STT) engine based on the Whisper model, and the resulting text is passed to DataLens. The answer is rendered back into speech by the text-to-speech (TTS) engine based on the Kokoro model, which supports selectable speaker characteristics and a range of audio encoding formats so that the output can be matched to the XR client. Both functions are exposed through a REST interface.

DataLens resolves the question against the knowledge base. The question is first expanded into a form better suited to retrieval and classified according to the kind of answer required, then matched against the stored embeddings to obtain the most relevant passages, from which a locally hosted language model composes the answer. The response presented in the XR environment combines the spoken answer with formatted text and, where the question calls for it, a figure or table drawn from the documentation, together with references to the source document and page so that an operator can confirm where the guidance originated.

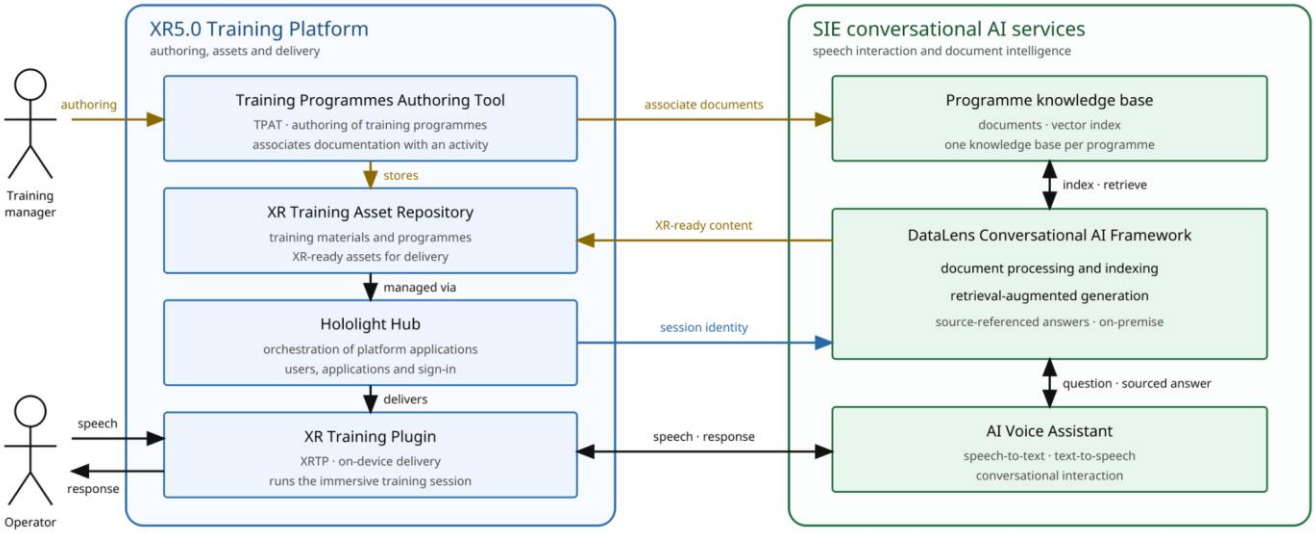


Figure 71 – Integration of the SIE Conversational AI Services within the XR5.0

Both services run on infrastructure controlled by the hosting organisation. They are container-deployed, so that installation at the pilot site consists of supplying the required configuration and starting the stack. Language and embedding models are served locally through Ollama, with GPU acceleration where CUDA-capable hardware is available and automatic fallback to CPU where it is not; models are loaded on demand and released after a period of inactivity, keeping memory use within the capacity of the host. The language model runtime and the object storage can be deployed on separate machines where resource demand requires it. Training documentation and the answers derived from it remain within this environment, allowing proprietary material to be used in immersive training without being exposed to external services.

Access is authenticated through the Hololight Hub. The session token accompanying each request identifies the user, and each deployment accepts only the organisations and applications configured for it, so a session issued for one organisation cannot be used to reach another's training documentation. The Voice Assistant passes the token on when it calls DataLens, so identity is preserved across both services.

5.2.6 XR-enabled AL & NSAI Models for XR-AI Detector

The developed solution is an AI-based computer vision system designed to support the detection and inspection of the Wing Anti-Ice Valve in aircraft imagery. The component incorporates a Human-AI collaboration approach in which an object-detection model provides automated detections, while human experts can validate and correct its outputs. Rather than treating the AI model as a static detector. It combines object detection, Human-in-the-Loop (HITL) feedback, Active Learning (AL), Explainable AI (XAI), and model governance within a common backend framework. The objective is therefore not only to detect the target component, but also to enable the system to collect expert knowledge, progressively improve from reviewed examples, provide explanations for individual detections, and control the deployment of newly trained models.

The component follows an **API-first, service-oriented backend architecture** implemented with FastAPI and deployed through Docker. The backend provides a stable REST interface that can be consumed by an external frontend, including the planned native XR application. The frontend is intentionally separated from the AI and data-management logic: it submits aircraft images, presents detection results, collects human feedback, requests explanations, and initiates authorized Active Learning workflows. The backend remains responsible for model inference, validation and persistence of feedback, dataset construction, candidate-model training and evaluation, XAI generation, and model governance. This separation is also reflected in the current frontend integration contract.

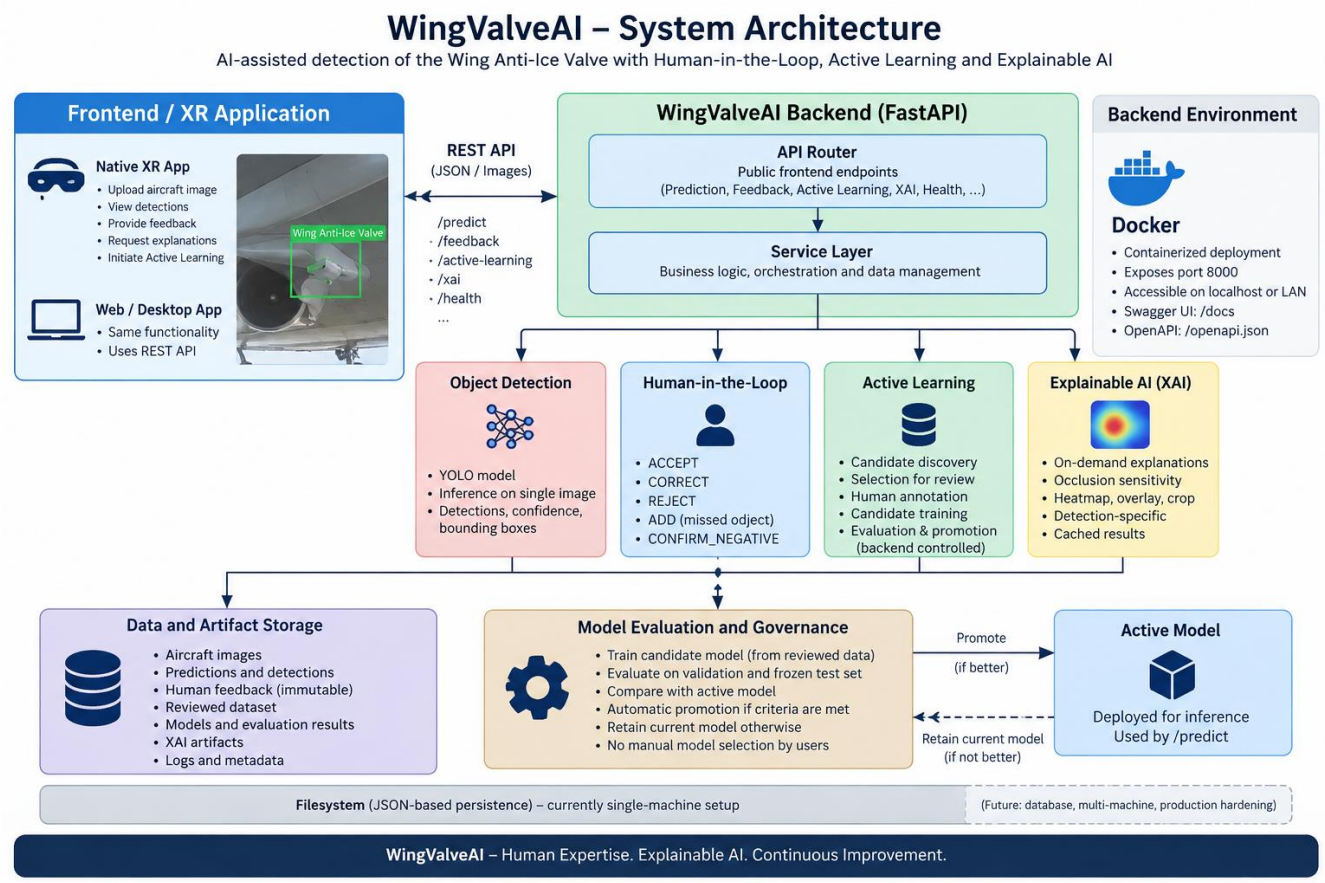


Figure 72 – System Architecture

Figure 72 presents the high-level architecture of an **AI-assisted computer vision framework for aircraft component inspection**. The system follows an API-first and modular design that separates the user-facing application from the underlying AI, data-management, learning, explainability, and model-governance processes. This separation enables different external applications, including native XR or desktop interfaces, to interact with the same backend services without being directly coupled to the implementation of the AI components.

The **Frontend/XR Application** constitutes the main interaction point for the operator. It integrates a custom component - XR-AI Annotator - developed for both annotation and detection of aircraft image. It communicates with the WingValveAI backend enabling users to submit aircraft images, inspect object detections and associated confidence values, provide feedback on predictions, request explanations for individual detections, and initiate authorized Active Learning processes. In addition, the component enables users to capture and annotate images directly in the XR environment for initial training of the detection model.

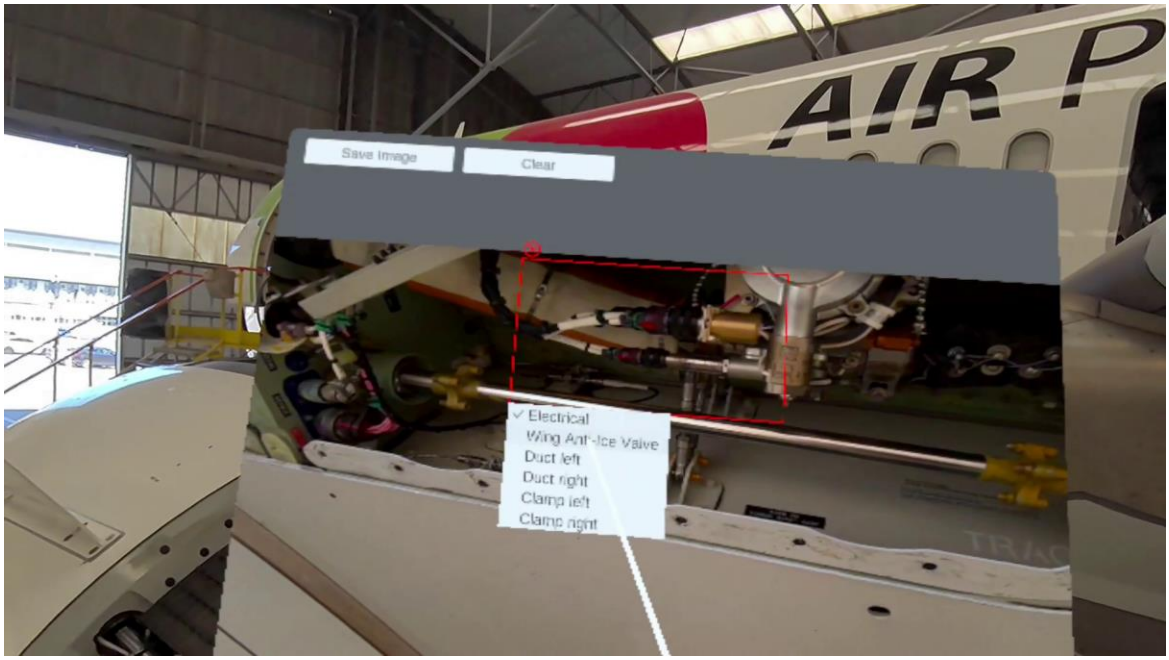


Figure 73 – Annotation of a Test Image using the XR-AI Annotator

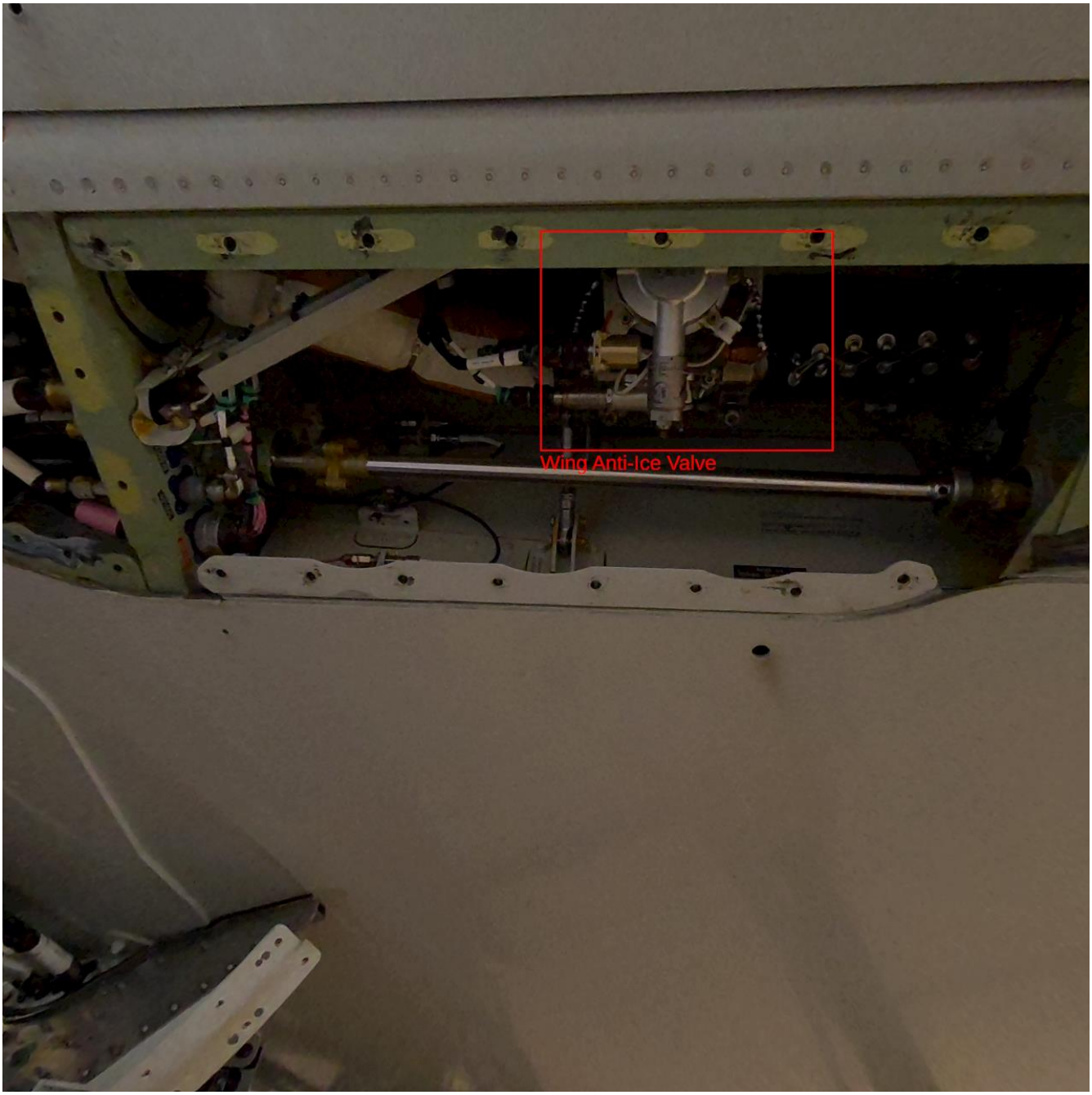


Figure 74 – Image captured and annotated using the XR-AI Annotator

Communication with the backend is performed through a REST API, allowing the frontend to remain independent of the underlying machine-learning implementation. The **FastAPI backend** provides the central communication and orchestration layer. Public endpoints expose the functionality required by external applications, while the internal service layer implements the corresponding business logic. Parameters related to model selection, inference, training, dataset construction, explainability, and model governance remain controlled by the backend. This ensures a clear separation between user interactions and internal AI operations. The **Object Detection component** analyses individual aircraft images using a trained object detection model. It identifies the target aircraft component and returns bounding boxes, confidence values, and persistent identifiers for predictions and individual detections.



Figure 75 – WingValve Detection

These identifiers maintain the relationship between an initial AI prediction and subsequent human feedback or explanation requests. The **Human-in-the-Loop component** enables domain experts to contribute directly to the interpretation and refinement of AI outputs. Users can accept valid detections, correct inaccurate bounding boxes, reject incorrect detections, add objects missed by the model, or confirm that an image contains no target object. The original machine prediction is preserved, while human feedback is stored independently to maintain traceability and provide reviewed information for subsequent learning processes. The **Active Learning component** exploits human-reviewed examples to progressively extend the available training information. Candidate images are identified for review based on the results of the detection process. Following human annotation or validation, the reviewed examples can be incorporated into a new dataset and used to train candidate detector models. In this way, expert feedback

becomes a structured source of information for iterative model development rather than remaining solely a correction of individual predictions. The **Explainable AI component** provides additional information about individual detections when requested by the user. The current implementation employs occlusion sensitivity to examine the contribution of different image regions to a selected detection. Explanation artifacts include heatmaps, overlays, and detection crops. Explanations are generated on demand, avoiding the computational cost of automatically producing them for every inference request.



Figure 76 – On-demand Explanation of the AI Detection using Heatmaps

The **Model Evaluation and Governance component** controls the transition between model development and operational inference. Newly trained models are treated as candidates and evaluated against the

currently active detector according to predefined performance and governance criteria. A candidate can replace the active model only when these criteria are satisfied. Otherwise, the existing model remains in operation. Human experts therefore contribute knowledge through feedback and annotation, while final model deployment remains governed by predefined backend rules rather than manual model selection. The **data and artifact management layer** supports the different components by maintaining aircraft images, predictions, human feedback, reviewed datasets, model versions, evaluation results, explanation artifacts, logs, and associated metadata. The current implementation primarily relies on filesystem- and JSON-based persistence. The backend is containerized using Docker, supporting reproducible deployment and network-based access by external frontend or XR applications.

Overall, the architecture integrates **object detection, human expertise, Active Learning, Explainable AI, and controlled model governance within a unified Human–AI collaboration framework**. Human feedback can improve the information available for future model development, while explainability supports the interpretation of individual predictions and the governance mechanism controls whether newly trained models are suitable for deployment.

API Interaction

The **REST API provides the main communication interface between the frontend/XR application and the backend services**, exposing a controlled set of endpoints for interacting with the system. Through these endpoints, external applications can submit images for object detection, provide Human-in-the-Loop feedback, participate in Active Learning workflows, and request Explainable AI information for individual detections. The API follows a clear separation of responsibilities: user-facing operations are publicly accessible, while model configuration, dataset construction, training, evaluation, and model governance remain internally managed by the backend. This design provides a stable integration interface while maintaining centralized control over the AI lifecycle.

List of Endpoints:

System Health Check — Checks whether the backend service is available and operational.

Image Prediction — Uploads a single aircraft image and returns detected objects, bounding boxes, confidence scores, and the corresponding prediction and detection identifiers.

Submit Human Feedback — Records Human-in-the-Loop feedback for an existing prediction, including ACCEPT, CORRECT, REJECT, and CONFIRM_NEGATIVE actions.

Add Missed Detection — Allows the user to manually annotate a target object that was missed by the detector.

Feedback History — Retrieves the complete feedback history associated with a specific prediction.

Final Reviewed Prediction — Retrieves the latest consolidated human-reviewed state of a prediction.

Active Learning Candidate Discovery — Identifies candidate images or predictions that may benefit from human review.

Active Learning Run Status — Retrieves information and status for a specific Active Learning run.

Active Learning Candidates — Retrieves the candidate samples associated with a particular Active Learning run.

Candidate Selection — Selects candidate samples to be considered for human review.

Candidate Review — Records the review state of a selected Active Learning candidate.

Start Active Learning Cycle — Initiates a complete backend-managed Active Learning cycle using the reviewed information.

Active Learning Cycle Status — Retrieves the progress and final governance outcome of an Active Learning cycle.

Generate XAI Explanation — Generates an on-demand **Occlusion Sensitivity** explanation for a selected detection.

Explanation Information — Retrieves metadata associated with a previously generated explanation.

Explanation Overlay — Retrieves the explanation visualization superimposed on the original image.

Explanation Heatmap — Retrieves the Occlusion Sensitivity importance heatmap.

Detection Crop — Retrieves the cropped image region corresponding to the selected detection.

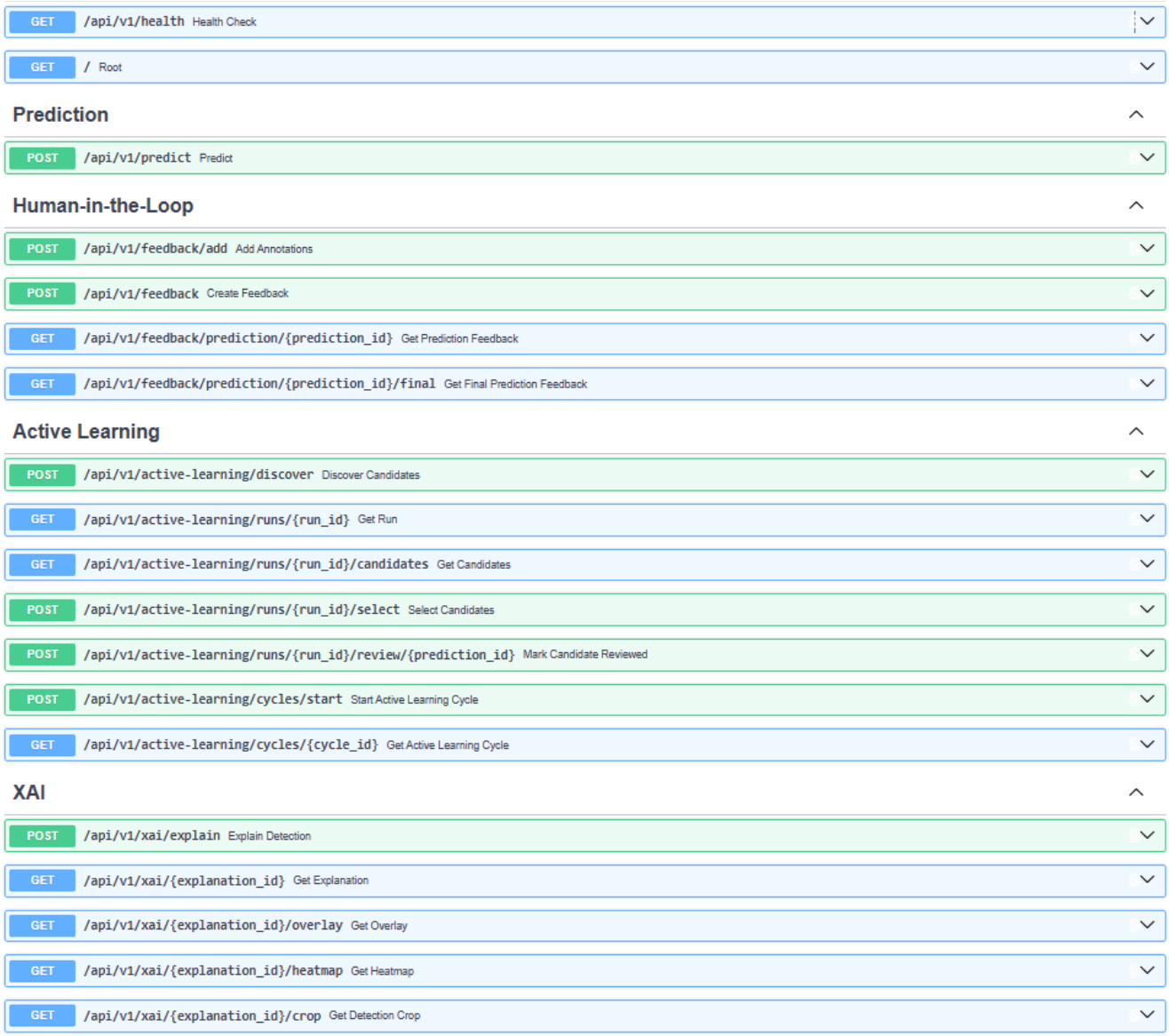


Figure 77 – API Endpoints

5.2.7 Workers' Digital Twin: Worker Shadowing and Monitoring

As anticipated in D6.3 (under the Shadow Operator concept, now revised), the Worker Shadowing and Monitoring (WSaM) Functional Module is a Human-Centric Digital Twin developed by SUPSI (with IML providing the XR tracking application and integration). The WSaM extends the HDT within XR5.0 to support Aircraft Maintenance Technicians (AMTs) during aircraft valve maintenance procedures. It provides two complementary capabilities: shadowing, where expert movements are replayed as an XR overlay to guide technicians, and monitoring, where live execution is analysed to provide real-time feedback and performance assessment. The module's design and implementation are detailed in D3.2 (Section 4.3.2).

During shadowing, the WSaM module uses the recorded motion signature of expert technicians as a reference. Rather than enforcing exact trajectory replication, it adopts a hybrid, milestone-driven monitoring strategy (a Human Activity Recognition, Monitoring and Analysis perspective): maintenance procedures are segmented into meaningful operational phases provided by the XR application, and monitoring evaluates whether the critical milestones of each phase are correctly reached, supported by motion analysis. This makes feedback robust to execution-speed variations, individual motion styles, and left- or right-handed execution.

Specifically, head and hand movements of expert technicians are captured through the XR headset (Meta Quest 3) and replayed as an XR overlay that provides trainees with a visual benchmark for correct task execution. Simultaneously, the module monitors the trainee's movements over rolling temporal windows and compares them against the expert reference, combining body-part positions, trajectories and timing. Normalised deviation scores are computed per tracked body part and evaluated against adaptive thresholds that depend on task phase and operator characteristics (e.g., skill level, handedness). When deviations exceed the threshold, corrective feedback is produced in real time.

Tracking data is captured by the XR headset, streaming approximately 53 head- and hand-related measurements per update at 10 Hz. Each update is serialised in JSON with timestamps, anatomical identifiers (e.g., wrist, palm, thumb) and contextual metadata such as activity phase, user profile, handedness and skill level. This data is streamed to the XR5.0 Clawdite Platform via MQTT, where it is processed by the WSaM Functional Module. Deviations are translated into human-understandable guidance by two interchangeable feedback engines: a deterministic rule-based engine that guarantees immediate, always-available feedback, and an LLM-based engine that adds context-aware, natural-language guidance when AI services are available, with automatic fallback to the rule-based engine. Monitoring results are published via MQTT and exposed through REST APIs compatible with the Clawdite ecosystem, and are visualised on a real-time operator dashboard (D3.2, Section 4.3.2).

This approach supports XR5.0's vision of human-centric, adaptive and data-driven industrial environments, where workers are empowered through intelligent systems that enhance learning, safety and productivity.

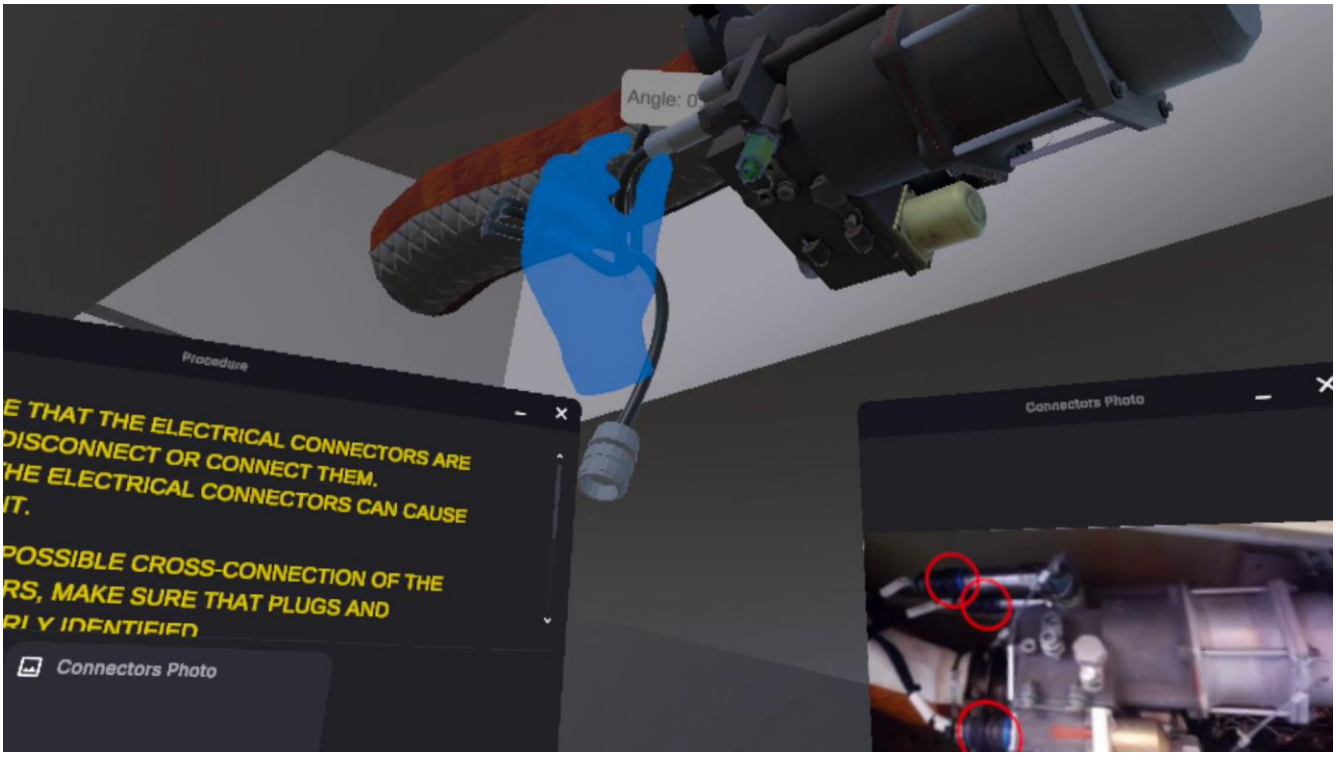


Figure 78 – Shadow Operator

5.2.8 Adaptation Engine: Adaptation of Personalised XR Content

Personalised presentation of training content in Pilot #4 is delivered through the Adaptation Engine, a service-oriented Functional Module jointly developed by SUPSI and IML and deployed on the shared XR5.0 Clawdite instance.

The engine is framework-independent and stateless, exchanging JSON over a REST API: it consumes training content from the XR5.0 Training Platform (XR Training Plugin) and worker context from the Clawdite HDT (worker profile and multimodal human-state indicators), and returns per-material and per-path presentation decisions that the XR clients apply to their interfaces, deciding what a given technician should see, expand, annotate, collapse or skip. Because the same service drives both web front-ends and the Unity-based headset UIs, it is reused across the XR5.0 pilots. The Adaptation Engine is described in D3.4 (Chapter 4).

5.3 Component Usage Matrix

Table 9 gives an overview of the XR5.0 technical components in the Pilot #4 Full Prototype.

Table 9 – XR5.0 Technical Components in the Pilot #4 Full Prototype

XR5.0 Technical Tasks	Technical Component	Technical Partner	How is the Component Used?
T3.1 Specifications for Worker Digital Images	N/A		

T3.2 Workers' Data Collection & Analytics	XR5.0 Activity Metrics Application	UNP	Collects operators' heart rates in real time to detect stress and evaluate their state.
T3.3 Workers' Digital Twins	Worker Shadowing and Monitoring (XR5.0 Clawdite Platform)	SUPSI	Captures expert technicians' movements and projects them as a digital avatar to guide trainees in real time with visual benchmarks and feedback during aircraft valve maintenance.
T3.4 Personalized XR Content Development	N/A		
T3.5 Adaptation of Personalized XR Content	Adaptation Egine	SUPSI	Provides real-time UI adaptation suggestions based on operators' profiles and interactions.
T4.1 Human-centered XAI Models	N/A		
T4.2 XR-enabled AL & NSAI Models	WingValve AI	UPRC	Provide real-time object recognition to help the operator identify valve components.
T4.3 Generative AI Models	AI Voice Assistant	SIE	Operates as a plugin within training programs, providing answers to user queries based on the training materials.
T4.4 Visualization of XAI Explanations	N/A		
T4.5 Visualization of AI Recommendations	N/A		
T5.1 Training Material	Training Materials for WAIV Removal and Installation generated using 3D Workflow Tool and 3D	IML	Training materials provide XR-based content based on traditional materials for equipment and safety checklists, access and procedure workflows.

	Model Explorer		
T5.2 Cloud-based XR Training Asset Repository	Cloud-based XR Training Asset Repository	SYN	Training materials are saved in the Cloud-based XR training asset repository.
T5.3 Hololight Hub	Hololight Hub	HOLO	Provides remote access to the XR app through streaming to VR headset Meta Quest 2/3.
T5.4 Training Programs	XR-based Training Programs for WAIV Removal and Installation	IML	Training programs provide removal and installation programs for the WAIV.

Table 10 maps the technical components/features for both the UC4.1 and UC4.2. The first use case is focused on Virtual training and aims to integrate the project's AI tools for the in an XR environment designed for training the maintenance process of the WAIV device, whereas the second use case focuses on AMT Digital Twin and aims to develop a human digital twin to guide junior technicians in performing the maintenance process of the WAIV device.

Table 10 – Mapping of Pilot #4 Technical Components to Pilot Use Cases in Full Prototype

	UC4.1	*UC4.2*
XR5.0 Activity Metrics Application	X	X
Worker Shadowing and Monitoring (XR5.0 Clawdite Platform)		X
Adaptation Engine	X	X
WingValve AI	X	
AI Voice Assistant	X	X
Training Materials for WAIV Removal and Installation	X	X
Cloud-based XR Training Asset Repository	X	X
Hololight Hub	X	X
XR-based Training Programs for WAIV Removal and Installation	X	X

5.4 DISCUSSIONS

In Pilot #4, the XR Training Plugin and core infrastructure have been established, and initial testing with an AMT technician provided encouraging feedback. The integration of the Hololight Hub has enabled remote

rendering of 3D content. Furthermore, the involvement of multiple partners in the pilot has facilitated the sharing of expertise and strengthened collaboration.

Despite these successes, several challenges remain. Training activities remain dependent on aircraft availability, which limits scheduling flexibility. Full XR training cycles have not yet been completed. The integration of biometric data collection is still in progress, and the limited availability of an on-site XR support team poses additional constraints.

Findings from the pilot indicate that the use of XR can significantly enhance training engagement and support error reduction. However, deploying XR solutions in a real hangar introduces logistical constraints. Moreover, workers' acceptance and adaptation to XR will be critical factors for ensuring both feasibility and effectiveness.

To address these challenges, several improvements are currently underway. These include the integration of the XR Training Plugin with the XR5.0 Cloud Repository and the extension of AI models (AL/NSAI) for object recognition of aircraft parts. In addition, integration with the XR5.0 Clawdite platform is underway to manage and leverage human data for the adaptation feature and Human Digital Twins (HDTs).

At the same time, certain risks have been identified. The unavailability of aircraft may cause delays in real-world testing. Connectivity issues in the hangar could negatively affect streaming quality. Furthermore, the confidentiality of training content requires the implementation of strict measures to ensure data protection.

The immediate next steps involve addressing the main risks associated with connectivity issues. Integration efforts will continue for AI models supporting object recognition, worker analytics, and the HDT. Preparations will also be made to conduct feasibility studies to further validate the approach.

6. PILOT 5 FULL PROTOTYPE

Pilot #5 – T6.6 XR Training for Effective Assembly Operations, led by SPACE, demonstrates an integrated XR environment supporting technicians in the assembly, commissioning and repair of edge devices. The Full Prototype combines immersive training, worker profiles, biometric stress monitoring and AI-assisted guidance. Training scenarios progress from equipment familiarisation to practical commissioning, with difficulty adjusted according to the technician's expertise and stress outcomes. The pilot addresses two core use cases:

- **UC5.1 – Personalized Training for Edge Device Assembly**: Provides structured VR training through levels of increasing difficulty, supported by technical documentation, guided objectives and knowledge assessments. Technicians practise assembly and commissioning procedures while interacting with a virtual device connected to physical hardware. Worker profiles inform initial level selection, while stress measurements from completed activities guide automatic adjustment of subsequent training difficulty.
- **UC5.2 – Remote Instructions for Edge Device Repairs**: Provides XR-based support for technicians working on deployed edge devices. Technicians can consult relevant documentation, monitor device messages and obtain repair and configuration guidance through a voice- and text-enabled AI assistant grounded in SPACE's technical manuals.

The Pilot #5 Early Prototype is documented in D6.3 Section 6. It established the Unity-based XR application by SYN, streamed to Meta Quest headsets through Hololight Hub/Stream (T5.3, HOLO), the first integration with the XR5.0 Training Platform (T5.2/T5.4, IML), the worker profile in the XR5.0 Clawdite platform (T3.3, SUPSI) and the Smartwatch Connector (T3.2, UNP) as the biometric data source. D6.3 also recorded that the Early Prototype relied on the pilot application as a glue layer between components that were not yet mutually integrated, that mock components stood in for unavailable modules, and that the AI tasks T4.1–T4.5 were not yet part of the pilot. This section follows the structure of D6.3 (Section 6) and focuses on developments since then.

Two main developments define the Full Prototype. First, an AI and Adaptation Services tier has been added, comprising the Stress Index Calculator by UNP/IPIA/SUPSI (T3.2), the Stress Detection Module by UPRC (T4.1, T4.2) and the XR5.0 Chat Engine by INNOV (T4.3) with speech input through Whisper; the XR application consumes these services through MQTT and REST interfaces rather than embedding their logic, which removes the glue-layer role noted in D6.3. Second, the XR5.0 Training Platform is now fully integrated: training scenarios, related materials and assessment questions are authored by the training supervisor in the Training Programs Authoring Tool (TPAT) and retrieved by the headset at run time. The resulting architecture is shown in Figure 79 and replaces the Early Prototype architecture of D6.3 Section 6.

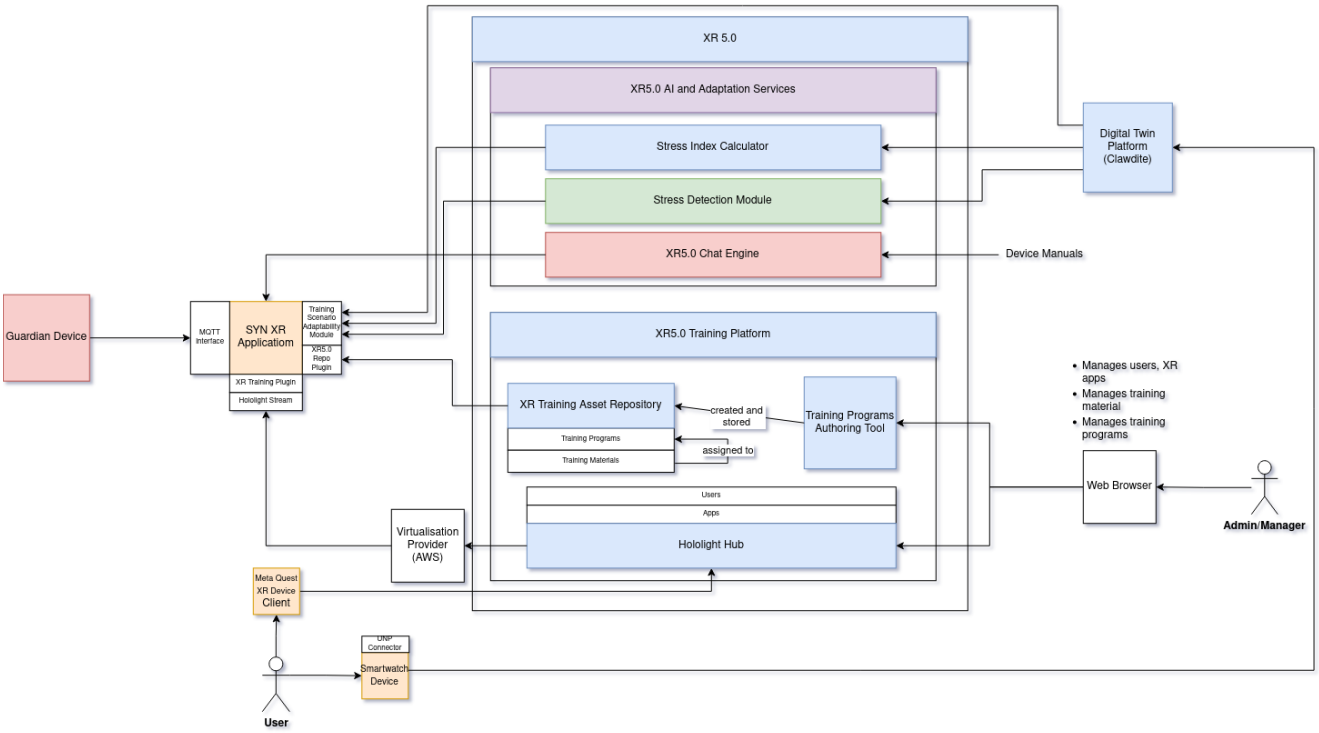


Figure 79 – Pilot #5 Architecture

In Figure 79, the XR operator wears the headset and a smartwatch paired with the companion smartphone application. The XR application, streamed through Hololight Hub, hosts the Training Scenario Adaptability Module and communicates with the Guardian over its MQTT interface. Speech is transcribed by Whisper and forwarded to the XR5.0 Chat Engine. The AI and Adaptation Services consume the biometric stream from the XR5.0 Clawdite platform and return stress state to the application. The XR5.0 Training Platform provides the Training Asset Repository and TPAT, through which the training supervisor authors scenarios and materials and maintains worker attributes in Clawdite.

6.1 PILOT SPECIFIC INFRASTRUCTURE

The Pilot #5 Full Prototype infrastructure consists of the following components:

- Hardware:
  - Guardian edge device (MDUINO 19R PLUS industrial controller), bench-wired for the commissioning scenario. The door actuated in the scenario is rendered in VR, and the magnetic door sensor is emulated by inserting pins into the corresponding digital input terminals of the Guardian, so that the relay outputs, inputs and MQTT traffic exercised in training are those of the real device.
  - Meta Quest 2/3 headsets for immersive training and support.
  - Wear OS smartwatches for biometric data collection.
  - Android smartphone running the XR5.0 Health Monitor companion application, mediating between the smartwatch and the XR5.0 Clawdite platform.
- Software:
  - Pilot #5 XR application (Unity), including the Training Scenario Adaptability Module, the MQTT Listener and the XR5.0 Chat panel (SYN).
  - Hololight Hub and Hololight Stream for hosting and streaming the XR application to the headset (HOLO).
  - XR5.0 Training Platform

- XR5.0 Clawdite Human Digital Twin Platform
- Smartwatch Connector
- Stress Index Calculator
- Stress Detection Module
- XR5.0 Chat Engine
- Whisper speech-to-text service
- Guardian Digital Twin: the interactive 3D representation of the Guardian in XR

6.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies

6.2.1 XR App

The Pilot #5 XR application, developed by SYN, is the single interface through which technicians access both use cases. It is built in Unity, streamed to the headset through Hololight Stream, and integrates the Guardian's MQTT interface, the XR5.0 Training Platform, the XR5.0 Clawdite platform and the AI and Adaptation Services. Its Early Prototype implementation, including the 3D Guardian model and the MQTT device integration, is described in D6.3 Section 6.2.1. The developments since D6.3 concern the session structure, the hardware-in-the-loop commissioning scenario, assessment, the worker panel with live biometrics, the chat panel and the MQTT listener.

Session Structure and Navigation

On launch the technician authenticates against the XR5.0 Training Platform and selects a use case from the main menu (Figure 80). The session interface is organised in five tabs: Task/Objectives, XR5.0 Training Platform, User, MQTT Listener and Settings. The tab structure is almost identical in both use cases, as UC5.2 replaces “Task” tab with “Chat”; however the main difference lies in the content loaded from the Training Platform and in whether the session is run in guided training mode with objectives and assessment (UC5.1) or in support mode against a deployed device with free-form chat and MQTT monitoring (UC5.2). Training content is organised as a series of training levels of increasing difficulty, from equipment-familiarisation quizzes to full commissioning procedures, and the level presented to a technician is selected according to their skill level in the worker profile and their stress state in the previous level (see Training Scenario Adaptability below).



Figure 80 – Pilot #5 XR Application Main Menu

Commissioning Scenario

The Early Prototype already operated against the physical Guardian over MQTT (D6.3 Section 6.2.1), but its training content consisted of an abstract objective list. The Full Prototype replaces it with a set of training levels, the most advanced of which is a commissioning scenario that exercises the device end to end. The technician wires the Guardian to power and network, connects the door-opening and door-closing circuits to relay outputs, connects the magnetic door sensor to a digital input, and then verifies the installation by closing the door and observing the resulting MQTT message on the Guardian's broker. In VR the Guardian is shown at bench scale with colour-coded cables (Figure 81), and a virtual monitor shows the door that the relays actuate. A relay energised from the VR interface is a real relay closing, and the sensor state reported over MQTT is the real input state of the device; only the door and the sensor's mechanical trigger are simulated, as described in the infrastructure section.

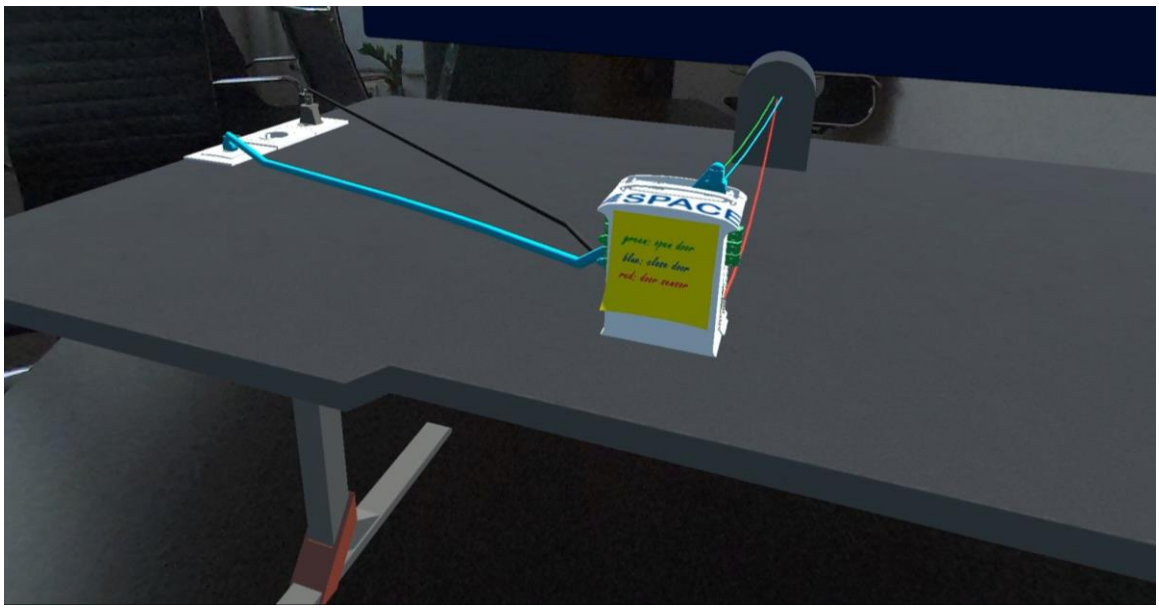


Figure 81 – 3D Guardian with Scenario Wiring in the Pilot #5 XR Application

Training Objectives and Assessment

In guided training mode the scenario is presented as an objective checklist retrieved from the Training Platform (Figure 82). Objectives are ticked as the technician completes them, and each step can be accompanied by the related material from the Training Asset Repository (connector diagrams, relay output and input descriptions, MQTT command reference). At the end of the scenario the application runs the knowledge check authored in TPAT for that scenario: questions are presented in the headset with the Guardian in view, so that the technician answers by inspecting the device (Figure 83). The result screen (Figure 84) combines the assessment score with the stress information produced during the session by the Stress Detection Module and Stress Index Calculator, reporting the total percentage of time during which the user was flagged as stressed. This is the point at which the biometric loop closes: the same level yields a competence outcome and a physiological outcome, and both feed the selection of the next level.

Training Scenario Adaptability

The Training Scenario Adaptability Module, internal to the XR application, uses the stress outcome of a completed level to adjust the difficulty of the next one. After a level is completed the proportion of the task duration during which the technician was flagged as stressed is computed from the per-interval stress state. If the technician was stressed for more than 50% of the task duration, the difficulty of the next level is lowered; if no stress was detected at all, it is raised; a small amount of stress is treated as desirable, since it sustains the trainee's focus and supports learning. The rule is deliberately simple and transparent, and it is the first automatic adaptation of content in the pilot; D6.3 described adaptation as a planned capability.



Figure 82 – Training Objectives for the Commissioning Scenario

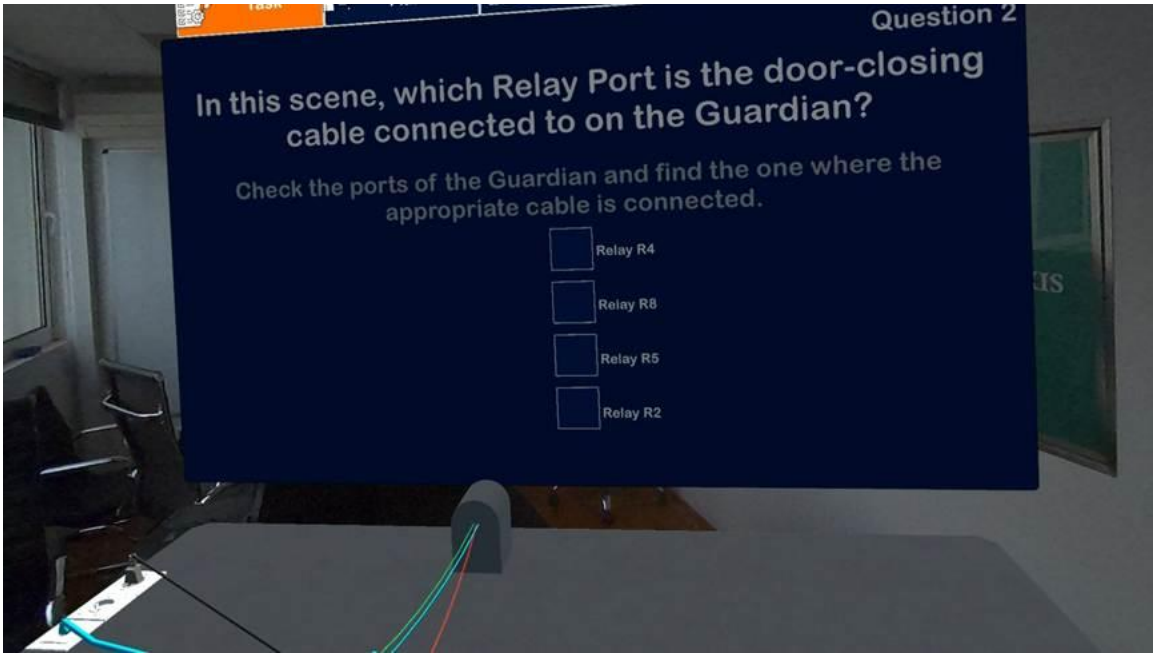


Figure 83 – In-headset Knowledge Check Retrieved from TPAT

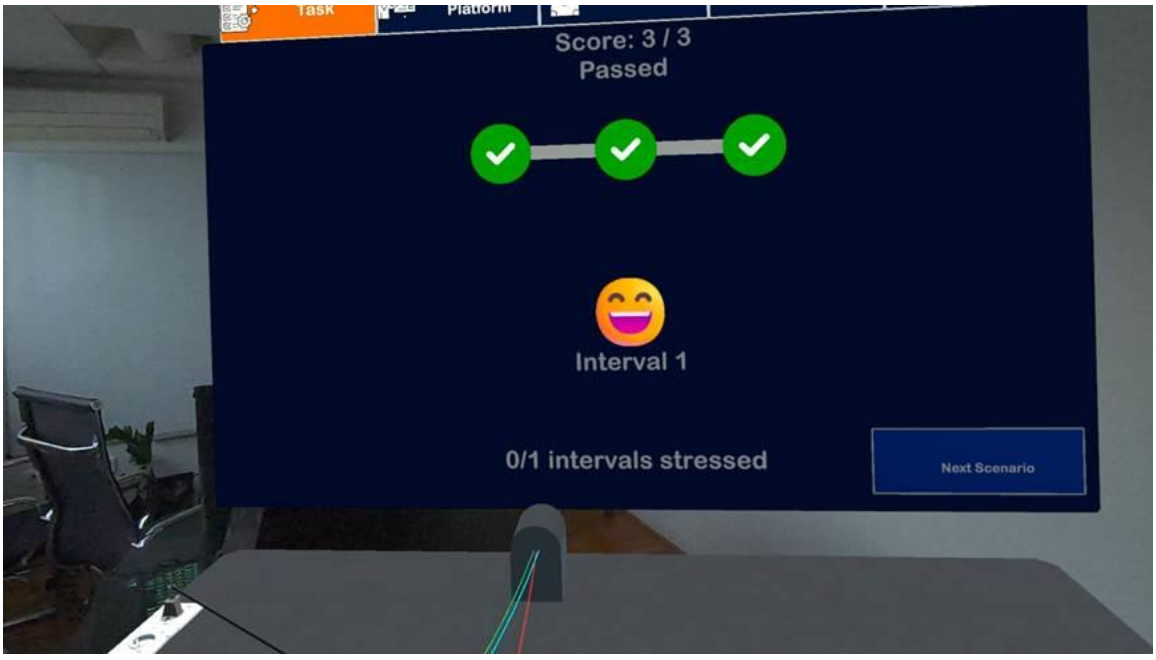


Figure 84 – Assessment Result with Stress Intervals

Worker Profile and Live Biometrics

The User tab (Figure 85) presents the worker profile retrieved from the XR5.0 Clawdite platform (name, gender, age, skill level) together with the live heart-rate trace and the stress level and HRV value returned by the Stress Index Calculator. In D6.3 this panel displayed raw biometrics only; it now displays the derived state that drives adaptation, which keeps the adaptation transparent to the technician.



Figure 85 – User Panel with Worker Profile and Live Biometrics

XR5.0 Chat Panel

The chat panel (Figure 86) gives the technician voice or text access to the XR5.0 Chat Engine. Speech captured by the headset is transcribed by Whisper and sent to the engine; the answer, grounded in the Guardian documentation ingested in the engine's document store, is displayed in the panel. Typical queries during the scenario are procedural ("how to open the door at relay R5?") or descriptive ("what is the communication shield?"). The panel is available in Use Case 5.2, being its primary instruction channel, where the technician works on a Guardian installation without a guided objective list and obtains repair and configuration instructions from the assistant.

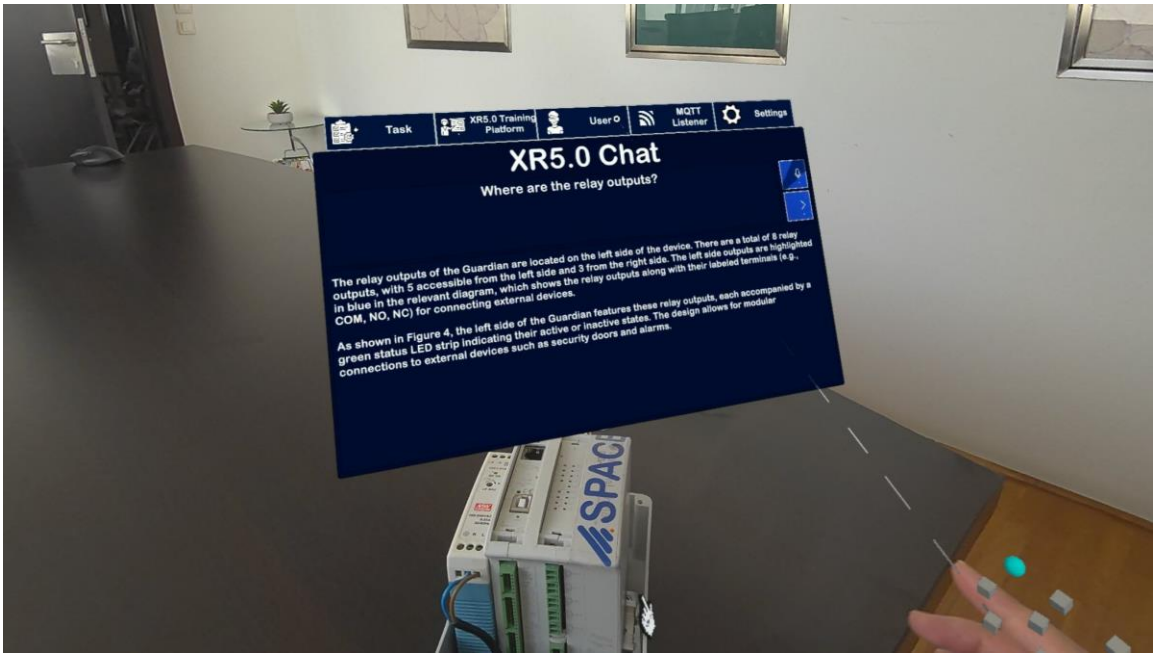


Figure 86 – XR5.0 Chat Panel in the Pilot #5 XR Application

MQTT Listener

The MQTT Listener tab subscribes to the Guardian's broker and shows incoming messages in the headset,

so that the technician can confirm that a wiring step has had the intended effect (for example, the input change published when the door sensor is triggered) without leaving VR. In support mode it serves as the diagnostic view of a deployed device.

6.2.2 Hololight Hub/Stream

Hololight Hub/Stream by HOLO offloads rendering and application hosting from the headset and streams the XR application to the Meta Quest. Its use in Pilot #5 is described in D6.3 Section 6.2.2 and the general Hololight Hub architecture in Section 5.2.2 of this deliverable (Pilot #4). No architectural change has been made since D6.3; the streaming configuration was retained for the Full Prototype and the additional AI services communicate with the streamed application instance rather than with the headset.

6.2.3 XR5.0 Training Management Platform

The XR5.0 Training Platform (IML, HOLO, SYN) holds the training materials and programs for the Guardian and delivers them to the XR application. Early Prototype integration is described in D6.3, Section 6.2.3, which also noted that integration with the Training Programs Authoring Tool (TPAT) was planned for a later phase. That integration has now been completed and training managers can author programs via a UI offered by the training platform. The Full Prototype uses the platform's image, quiz, workflow and checklist material types, and the training program now comprises several levels of varying difficulty, from familiarisation with the Guardian device to the commissioning procedure, serving both UC5.1 and, for reference materials, UC5.2.

Scenario Authoring in TPAT

The training supervisor authors each training level in TPAT: the objective sequence, the knowledge-check quiz questions (Figure 87) and the related materials attached to the scenario. Related materials for the commissioning scenario include the Communication Shield, Relay Outputs, Inputs, Connector Diagram and Sending MQTT Commands sections of the Guardian manual. Assets are created directly from the manual PDF using the tool's extraction function (Figure 88), so that the schematic images shown in the headset are the ones in SPACE's documentation.

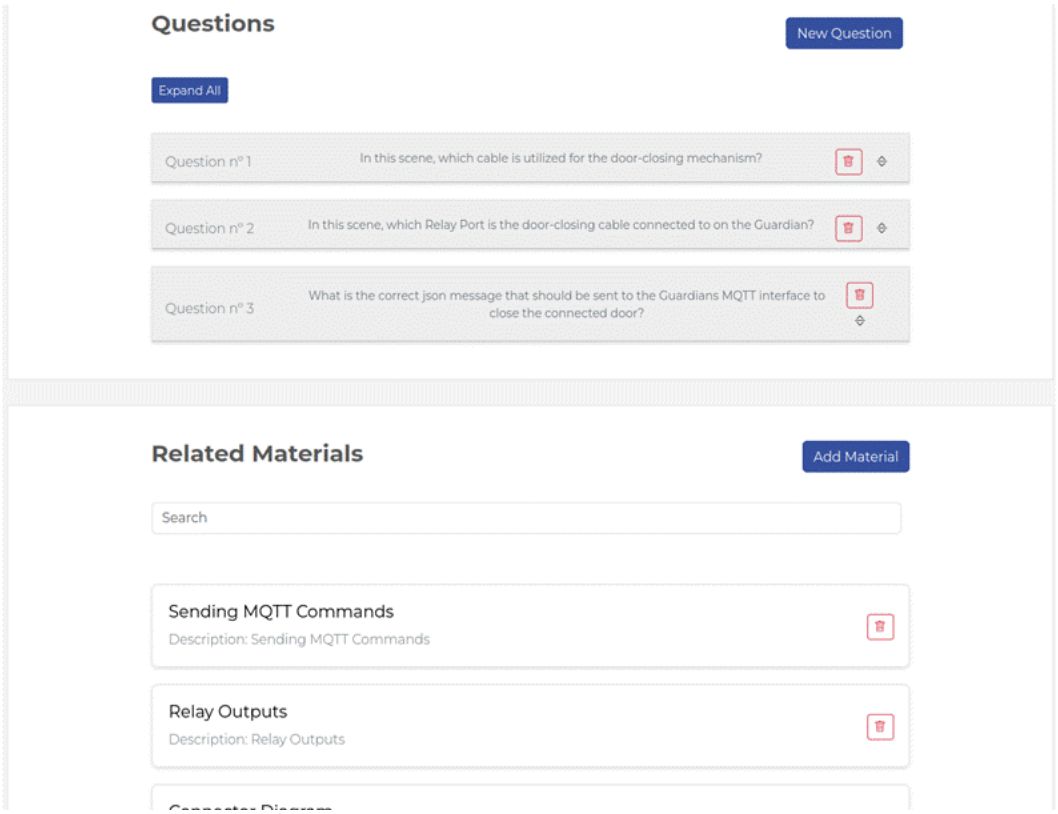


Figure 87 – Scenario Questions and Related Materials in TPAT

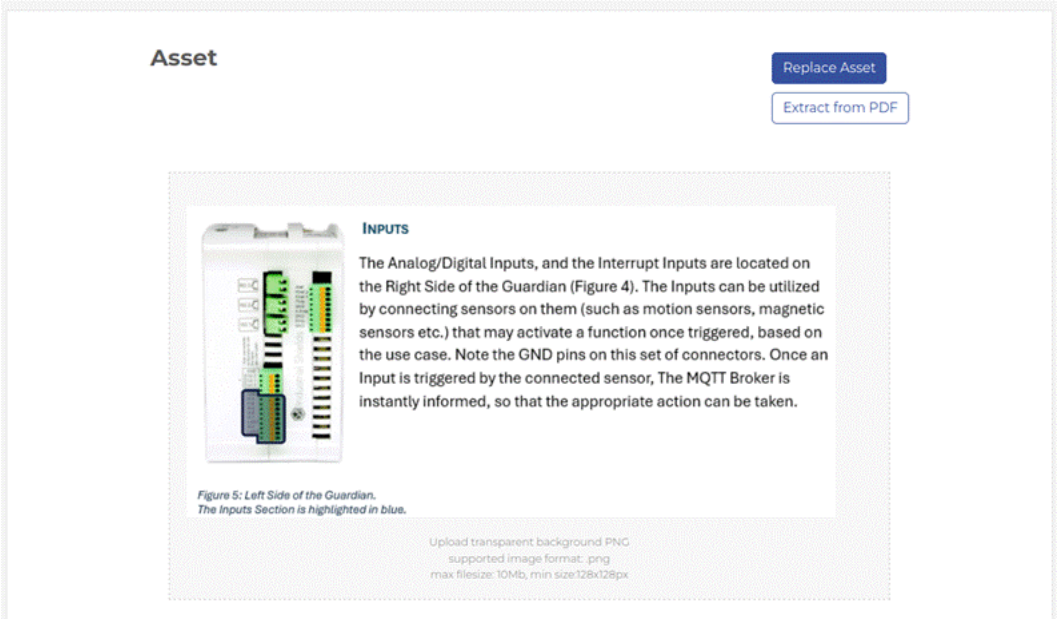


Figure 88 – Asset Extracted from the Guardian Manual in TPAT

Run-time Retrieval

At run time the XR application retrieves the training program for the pilot through the platform's REST API (Figure 89). The response carries the program's materials with their type, description and asset references; for the current program these include the MDUINO Schematic image and the Pre-Deployment Checklist. The materials are rendered in the headset's Training Platform tab as the technician works through the scenario

(Figure 90).

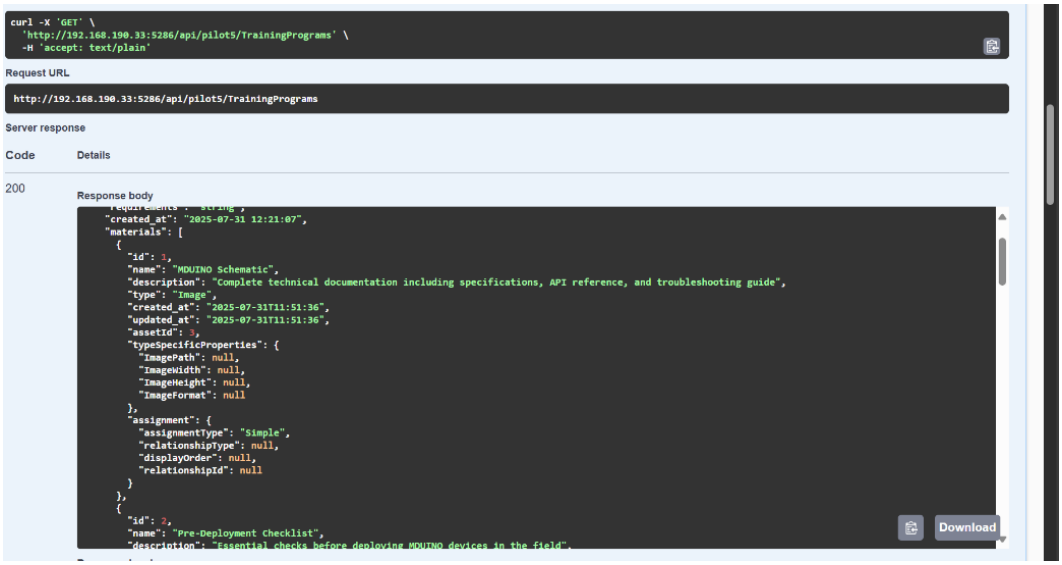


Figure 89 – Training Program Retrieval through the XR5.0 Training Platform REST API

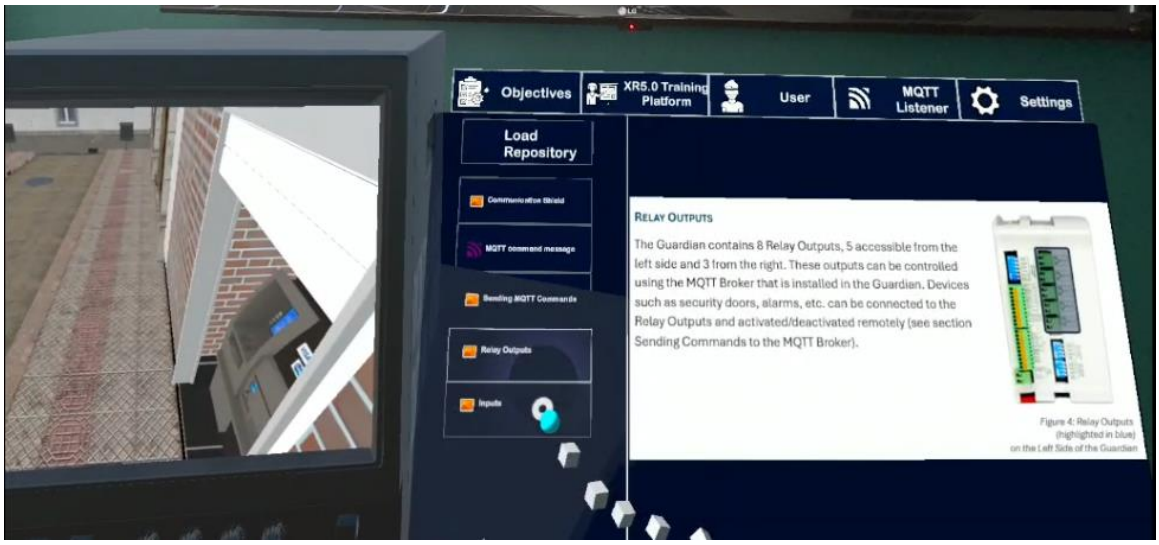


Figure 90 – Training Platform Materials Rendered in the Headset

6.2.4 XR5.0 Clawdite Platform

The XR5.0 Clawdite platform by SUPSI hosts the Human Digital Twin for Pilot #5: the worker profile (age, skill level, gender) and the biometric stream from the Smartwatch Connector. Its data model and the worker characteristics implemented for the pilot are described in D6.3 Section 6.2.4. Since D6.3 the platform's role has changed from a store that the XR application reads to the hub through which the AI and Adaptation Services operate: the Stress Index Calculator subscribes to the biometric stream on the platform's MQTT broker, and the training supervisor maintains worker attributes, in particular skill level, as the profile input to scenario selection.

6.2.5 Smartwatch Connector

The Smartwatch Connector by UNP collects heart rate and heart-rate variability from the technician's smartwatch and streams them to the XR5.0 Clawdite platform through the Android companion application. Its dual-app architecture, the BLE and Wear OS integration modes, the QR-code session start and the metrics collected are described in D6.3 Section 6.2.5. For the Full Prototype the Wear OS native integration was

deployed in the pilot with a Wear OS watch and the XR5.0 Health Monitor companion application (Figure 91). The connector is the sole physiological data source for the Stress Index Calculator and the Stress Detection Module described next.

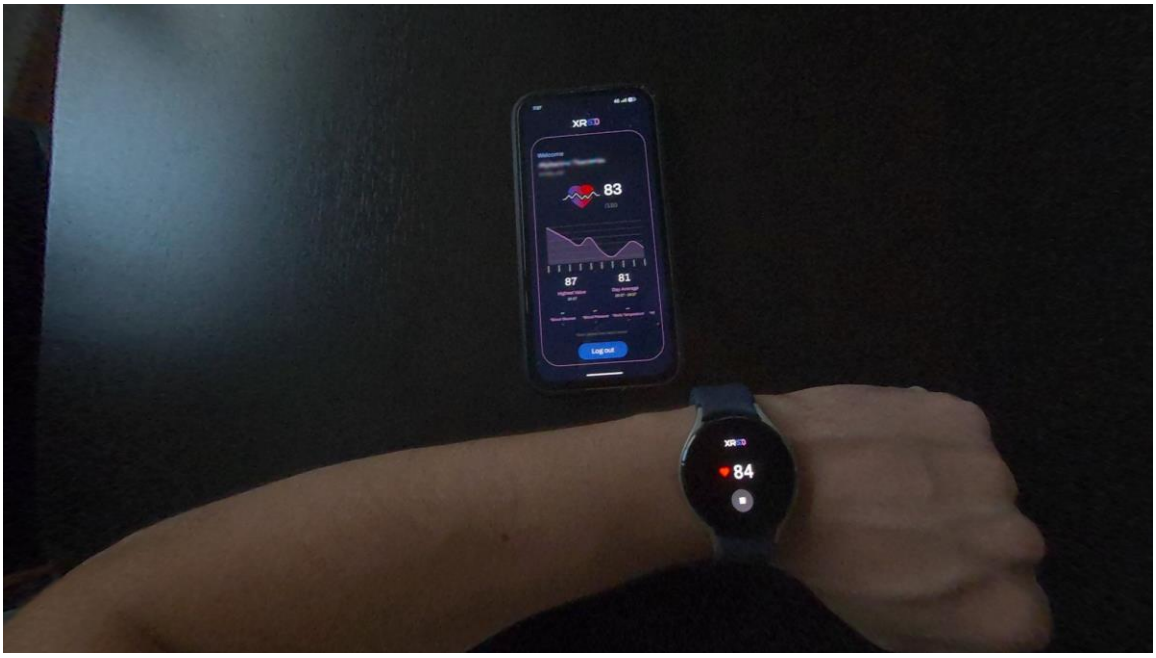


Figure 91 – Smartwatch and Companion Application in Use during a Pilot #5 Session

6.2.6 Stress Index Calculator

The Stress Index Calculator (SIC) functional module, developed by UNP, IPIA and SUPSI under T3.2 and new since D6.3, is a deterministic real-time service that turns the biometric stream into a per-worker stress state that the XR application can act on immediately. It was introduced so that the biometric loop could be closed in the pilot with a transparent, always-available rule, while the learned Stress Detection Module (next subsection) matures on real WearOS data.

The service runs as an MQTT listener on the AI and Adaptation Services host and exposes a web dashboard (Figure 92) with Dashboard, Baselines, Configuration, Logs and Payload views. For each worker it maintains an RMSSD baseline established at the start of a session and computes, over a rolling window, the current RMSSD, the relative drop against the baseline, an HRV score, and derived mental-workload and psychological-stress indicators. The XR Application calculates the percentage of time user stress was raised beyond the configured threshold during the training scenario, and uses this value —in conjunction with the Stress Detection Module— to drive the Training Scenario Adaptability Module.

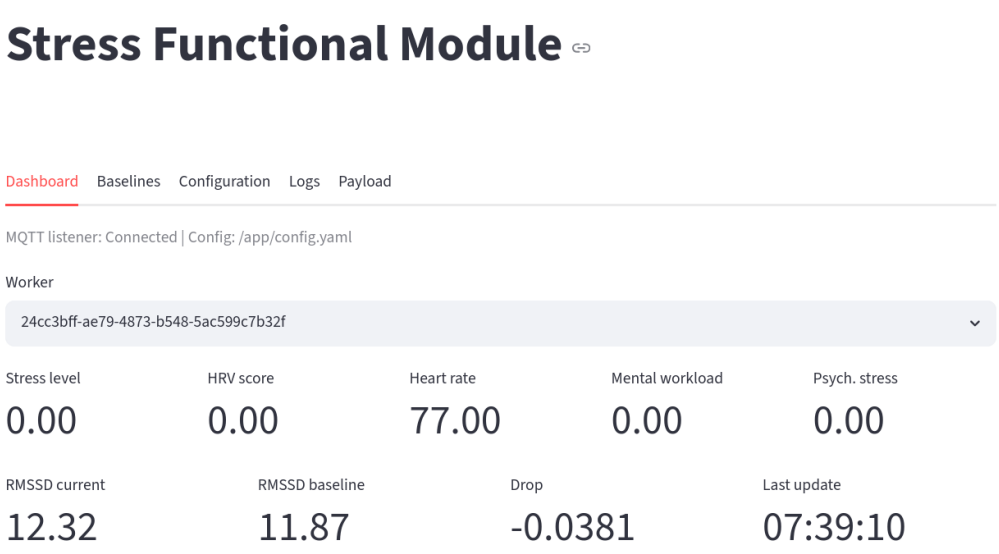


Figure 92 – Stress Index Calculator Dashboard

The SIC functional module evolved through three main stages. Version 1 established the core integration with the Clawdite platform, combining REST-based entity management with MQTT-based physiological data processing. It introduced the functional module structure with heart-rate, mental-workload, and psychological-stress inputs, together with HRV and stress-index calculation. However, entity lifecycle management and worker baselines were primarily handled at runtime, limiting recovery and reuse.

Version 2 focused on operational maturity. Functional-module entities were cached and validated before reuse, activation and deactivation controls were introduced, and a Streamlit interface enabled operators to start, stop, configure, monitor, and recover the module. Baseline collection became explicit and user-controlled, while missing or invalid platform entities could be replaced without terminating the service. These changes improved reliability and made the module more suitable for continuous operation.

Version 3 addressed the remaining limitation of runtime-only baselines by associating baseline data with eligible workers and persisting it in Clawdite as characteristic values. On startup, the system validates the worker identity and factory entity model, restores any available baseline, and maintains an in-memory copy for efficient processing. Newly collected baselines are persisted for future sessions. This version therefore improves continuity across restarts and deployments, while also making worker eligibility, baseline origin, persistence status, and loading errors visible to operators.

Overall, the development path moved the functional module from an initial proof of concept, through an operator-controlled and recoverable service, to a deployment-ready system with durable worker-specific context.

Integration Architecture Diagram

Figure 93 shows the SIC functional module integration architecture. The diagram shows how the stress-level functional module integrates with the Clawdite platform over REST and MQTT. The service creates or reuses functional module entities, activates them, and publishes computed state. It also maintains a local entity cache and optionally launches a Streamlit UI for monitoring.

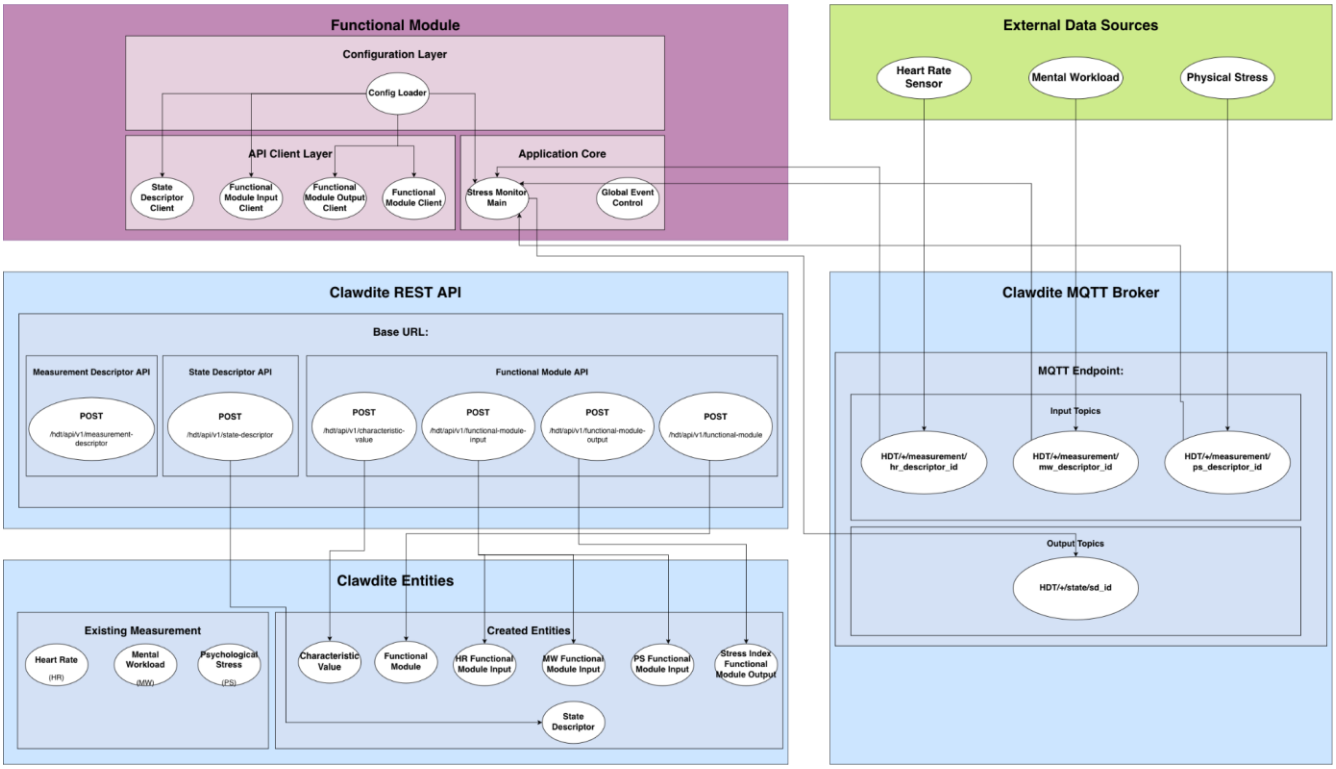


Figure 93 – Stress Index Calculator Integration Architecture Diagram

MQTT Contracts

- **Input data flow and topics**

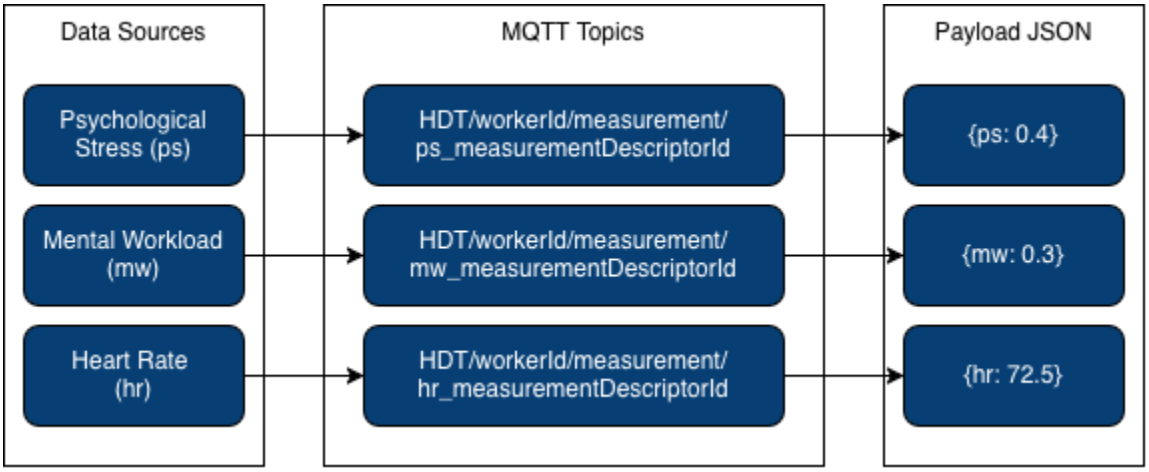


Figure 94 – Stress Index Calculator Input Data Flows and Topics

- **Output data flow and topics**

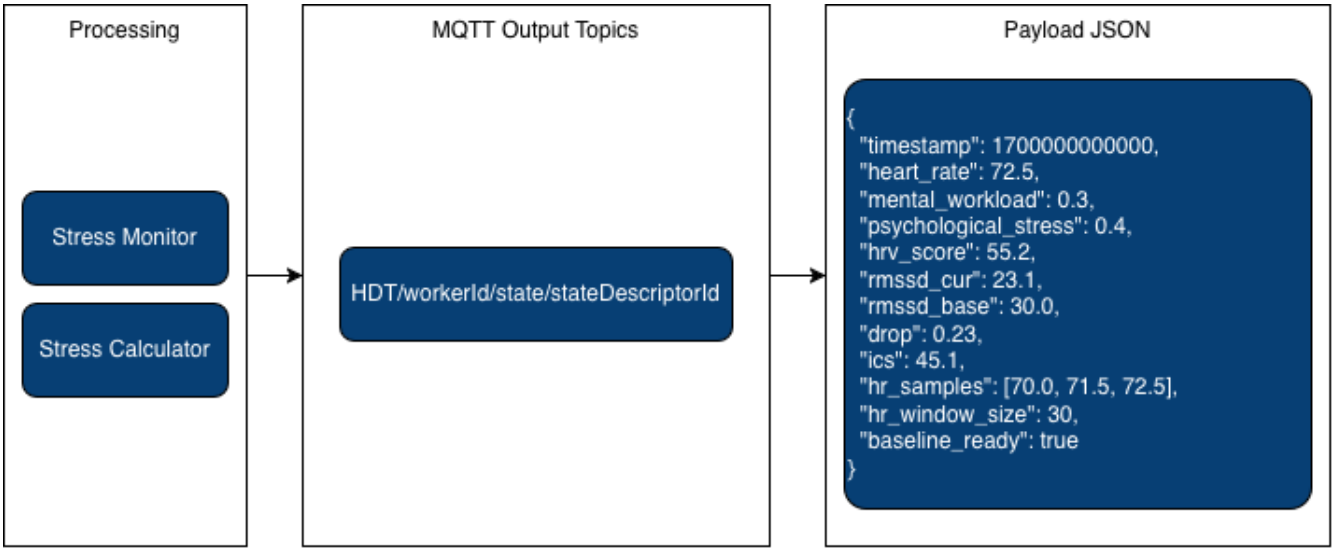


Figure 95 – Stress Index Calculator Output Data Flows and Topic

6.2.7 Stress Detection Module

The Stress Detection Module, developed by UPRC under T4.1 and T4.2 and new since D6.3, identifies and assesses the stress level of users during training activities by analysing physiological data acquired through wearable devices. The per-interval derived stress flags are published to the XR Application, which displays them on the assessment result screen (Figure 84) and uses them—in conjunction with the Stress Index Calculator—to adjust the difficulty of subsequent training scenarios.It thus forms part of a Human–AI collaboration approach in which AI supports training officers in understanding users' physiological responses and adapting training programmes to individual needs. In Pilot #5 it complements the Stress Index Calculator: where the calculator applies a fixed rule to the live stream, the module provides a learned estimate with explainability and a human-in-the-loop feedback path, and is intended to replace or augment the deterministic index as operational WearOS data accumulates. Its development follows a hybrid data approach that combines physiological data from the WESAD (Wearable Stress and Affect Detection) dataset with data originating from the project, transformed into a representation compatible with the target WearOS-based environment. The AI component estimates the stress state of the user, Explainable AI (XAI) provides interpretable information about the physiological factors contributing to individual predictions, and a Human-in-the-Loop Active Learning mechanism enables authorised training officers to provide contextual feedback on selected observations, supporting controlled model improvement and informing decisions on training difficulty.

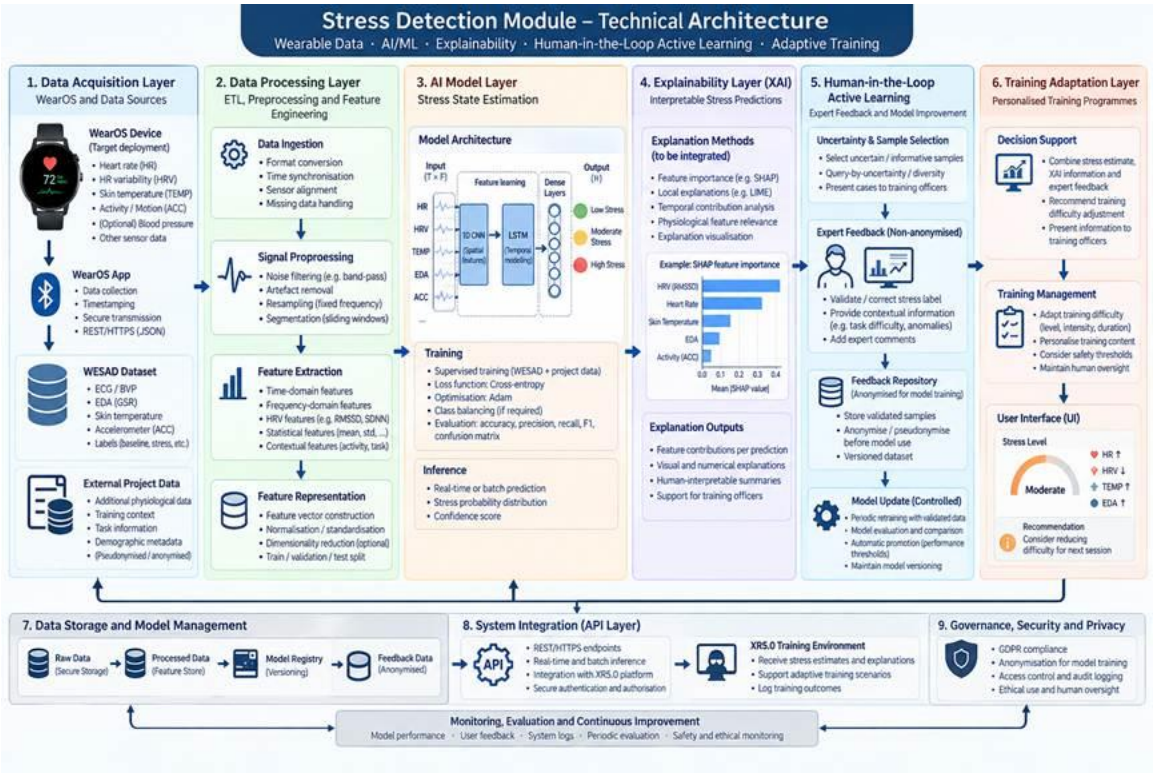


Figure 96 – Stress Detection Module – Technical Architecture

Figure 96 shows the module's architecture, which combines wearable sensing, physiological data processing, machine learning, XAI, Active Learning and human-supervised training adaptation in the following layers:

- **Data Acquisition Layer.** Manages the physiological and contextual data required for stress detection. The target deployment uses WearOS devices collecting heart rate, heart-rate variability, temperature and activity data; during development, WESAD-derived data are combined with project-specific data and transformed into a representation compatible with the target environment.
- **Data Processing Layer.** An ETL and preprocessing pipeline performs data harmonisation, temporal alignment, filtering, missing-value handling, segmentation, normalisation and feature extraction, producing statistical and HRV-related features for training and inference.
- **AI Stress Detection Layer.** A supervised model estimates the user's stress state, distinguishing between stress levels and providing confidence information alongside the prediction. Performance is evaluated with precision, recall, F1-score, accuracy and confusion matrices.
- **Explainable AI Layer.** Identifies the physiological features that contributed to the model's decision through methods such as SHAP, LIME or model-specific techniques, providing local and global explanations.
- **Human-in-the-Loop Active Learning Layer.** Identifies uncertain or informative cases for expert validation. Authorised training officers review these cases, confirm or correct model outputs and provide training-related context; validated feedback contributes to controlled retraining.
- **Training Adaptation Layer.** Stress estimates, XAI information, contextual data and training-officer expertise support the personalisation of training activities. The AI provides decision support; it does not autonomously modify the training programme.
- **Data and Model Management, API and System Integration, Governance and Monitoring.** Dedicated repositories maintain processed data, validated feedback, model versions and evaluation results; the functionality is exposed as a service through an API; security, privacy, access control

and pseudonymisation are applied throughout, and model performance is periodically re-evaluated as real WearOS data become available.

API Interaction

The module is exposed to the XR application as a REST service, which separates the AI implementation from the applications consuming its results. The provided endpoints allow the XR application to trigger stress calculation on demand for a specific user just for the duration of each training scenario.[C3]

6.2.8 XR5.0 Chat Engine

The XR5.0 Chat Engine by INNOV (T4.3) is a retrieval-augmented assistant that answers technicians' questions from SPACE's Guardian documentation. It is new to Pilot #5 since D6.3, where generative AI was listed as planned for the Full Prototype. The engine itself, its web administration interface and its REST API are described in Section 4.2.2 of this deliverable (Pilot #3); this subsection covers the Pilot #5 configuration and integration.

The Guardian documentation is ingested into the engine's document store through its OCR pipeline, chunked and tagged to the pilot (Figure 97), so that retrieval is restricted to Pilot #5 content. In the headset, queries are entered by voice, transcribed by Whisper and sent to the engine's chat endpoint; the answer is displayed in the chat panel described in the XR App subsection. The engine's answers are grounded in the same manual sections that TPAT exposes as related materials, so that the technician receives consistent instructions whether they consult the material or ask the assistant.

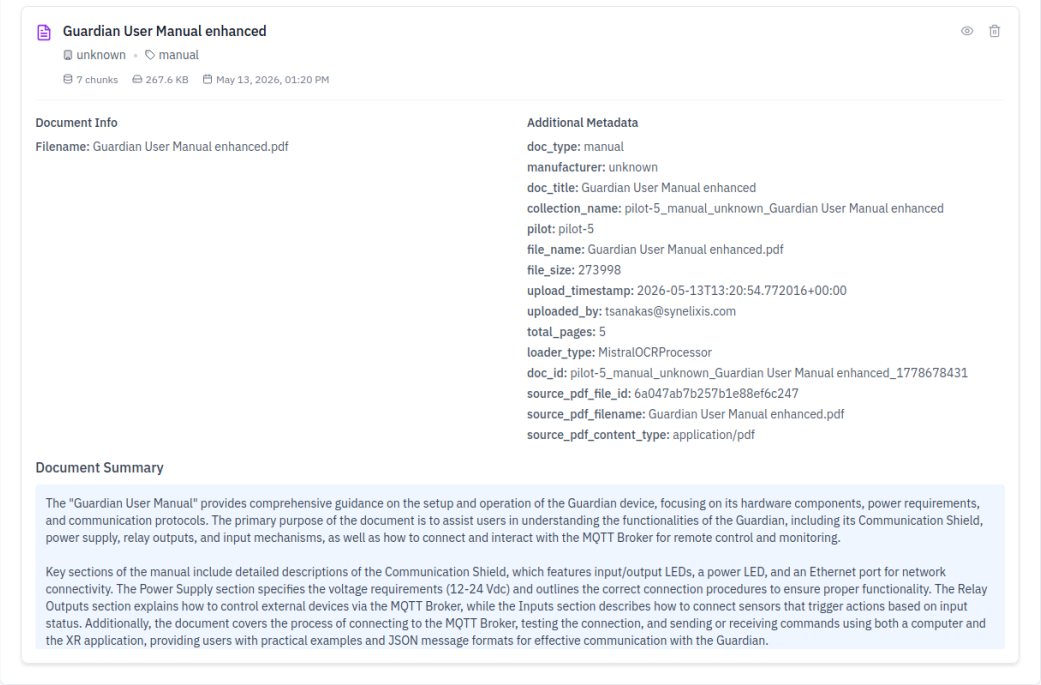


Figure 97 – Guardian Manual Ingested in the XR5.0 Chat Engine Document Store

6.3 Component Usage Matrix

Table 11 gives an overview of the XR5.0 technical components in the Pilot #5 Full Prototype. Components marked as not part of the Early Prototype in D6.3 Section 6.3 and now integrated are the Stress Detection Module (T4.1, T4.2), the XR5.0 Chat Engine (T4.3) and the stress and assessment visualisations in the XR application (T4.4, T4.5); the T3.2 entry is extended by the Stress Index Calculator.

Table 11 – XR5.0 Technical Components in Pilot #5 Full Prototype

XR5.0 Technical Tasks	Technical Component	Technical Partner	How is the Component Used?
T3.1 Specifications for Worker Digital Images	N/A		
T3.2 Workers' Data Collection & Analytics	Smartwatch Connector; Stress Index Calculator	UNP / IPIA / SUPSI	Streams heart rate and HRV from the technician's Wear OS smartwatch to the XR5.0 Clawdite platform, and computes a deterministic RMSSD-baseline stress index per worker in real time.
T3.3 Workers' Digital Twins	Human Digital Twin (XR5.0 Clawdite Platform)	SUPSI	Stores the worker profile (age, skill level, gender) and the biometric stream; hub through which the Stress Index Calculator and the training supervisor operate.
T3.4 Personalized XR Content Development	Training levels of varying difficulty in the XR application	SYN / SPACE	Level selection and material presentation according to the worker's skill level and stress state.
T3.5 Adaptation of Personalized XR Content	Training Scenario Adaptability Module	SYN	Automatically raises or lowers the difficulty of the next training level from the proportion of the completed level during which the technician was stressed.
T4.1 Human-centered XAI Models	Stress Detection Module – XAI layer	UPRC	Explains which physiological features drive a stress prediction.
T4.2 XR-enabled AL & NSAI Models	Stress Detection Module – AI and Active Learning layers	UPRC	Learned stress-state estimation with human-in-the-loop validation by training officers.
T4.3 Generative AI Models	XR5.0 Chat Engine	INNOV	RAG assistant over the Guardian documentation, queried by voice from the headset.
T4.4 Visualization of XAI Explanations	User panel in XR App	SYN	Displays the derived stress level and HRV value driving adaptation alongside the live heart-rate trace.
T4.5 Visualization of AI Recommendations	Assessment result screen and chat panel in XR App	SYN	Shows stress intervals with the assessment score and renders Chat Engine answers in the headset.
T5.1 Training Material	SPACE Guardian Device Training Materials	IML / SPACE	Manual sections and schematics extracted into the Training Asset Repository through TPAT.

T5.2 Cloud-based XR Training Asset Repository	XR5.0 Training Platform – Asset Repository	HOLO / SYN	Stores materials and programs; retrieved by the XR application through REST.
T5.3 Hololight Hub	Hololight Hub/Stream	HOLO	Hosts and streams the XR application to the Meta Quest headset.
T5.4 Training Programs	Guardian training levels (familiarisation quizzes to commissioning scenario)	IML / SYN	Authored in TPAT by the training supervisor; delivered as objectives, materials and quiz questions in the headset.
Pilot-specific Components	Guardian MQTT Integration	SYN / SPACE	Bidirectional MQTT link between the XR application and the physical Guardian for commands, telemetry and input-state monitoring.

Table 12 maps the components to the two use cases. Both use cases run through the same XR application against the same Guardian. UC5.1 runs the commissioning scenario in guided training mode, with objectives, related materials, the knowledge check and the stress-annotated result. UC5.2 runs against a deployed device in support mode, in which the technician works from the MQTT Listener, the Training Platform materials and the Chat Engine rather than from an objective list, and receives repair and configuration instructions from the assistant.

Table 12 – Mapping of Pilot #5 Technical Components to Pilot Use Cases in Full Prototype

	UC5.1	*UC5.2*
Hololight Hub/Stream	X	X
XR5.0 Training Platform (incl. TPAT)	X	X
XR5.0 Clawdite Platform	X	X
Smartwatch Connector	X	X
Guardian MQTT Integration	X	X
Stress Index Calculator	X	Not planned
Stress Detection Module	X	Not planned
XR5.0 Chat Engine	Not planned	X

6.4 DISCUSSIONS

Since D6.3, Pilot #5 has progressed from an Early Prototype in which the XR application acted as a glue layer over partly integrated components to a Full Prototype in which the application is a consumer of platform services. Four developments account for most of the change.

The first is that the biometric loop is closed. In D6.3 the smartwatch stream reached the headset as raw values displayed for self-awareness. In the Full Prototype, the stream is turned into a stress metric by the

Stress Index Calculator and into a per-interval stress state by the Stress Detection Module, shown to the technician during the session and combined with the assessment score at the end of it. A training level therefore produces a competence outcome and a physiological outcome together, and the stress outcome now selects the difficulty of the next level automatically, which is the adaptive training that the pilot set out to demonstrate.

The second is the way the AI layer was introduced. The three AI services are integrated through MQTT and REST interfaces, and none of their logic lives in the XR application. The decision to deploy a deterministic stress index alongside the learned Stress Index Calculator was taken deliberately: the rule-based index is transparent, always available, and sufficient to close the loop for the evaluation phase, while the learned module, which was trained on WESAD data because operational WearOS data were not yet available in quantity, can be validated against pilot data and brought in through the same API without changing the application. The same two-tier pattern (deterministic fallback, learned enhancement) appears in Pilot #4's Worker Shadowing feedback engines, and is likely to be a reusable lesson for human-state estimation in industrial XR.

The third is that scenario authoring has moved to the training supervisor. With TPAT in use, objectives, questions, and materials are authored in the platform and retrieved at runtime, and the manual sections that the Chat Engine answers from are the same sections that TPAT exposes as materials. Adding a scenario no longer requires a change to the application.

The fourth is expanding upon the commissioning scenario. The Early Prototype already ran against the physical Guardian; what is new is the variety of scenarios offering an in-depth training experience that suits not just experienced users, but also trainees who are seeing the Guardian device—or even experiencing XR training—for the first time. This way, new trainees can ease into the process and avoid being overwhelmed by having to handle a complex assembly process using hand controls in an XR setting. On the contrary, beginners are tasked with going through the documentation, observing existing virtual installations, and answering quizzes—a process that gives them valuable familiarization with both the device itself and the training controls. Once the Application Training Adaptability Module deems them ready, they are assigned the more complex tasks.

The challenges recorded in D6.3, disparate component maturity and hardware fragmentation, have largely been resolved by the service-based integration, and the mock components of the Early Prototype have been replaced. The remaining risks are the small volume of operational biometric data, which limits validation of the learned module before the evaluation phase, and the dependence of the stress index on a per-session baseline, which requires a short calm period at the start of each session and may be sensitive to how that period is conducted.

Next steps toward the final evaluation in T6.8 are the validation of the Stress Detection Module on pilot data collected during evaluation sessions and its promotion to the live path where it outperforms the index, and the collection of user feedback on the transparency of the stress display and on the assessment result.

7. PILOT 6 FULL PROTOTYPE

Pilot #6 - T6.7 Human Centric Guidance and Troubleshooting, led by RW, focuses on supporting maintenance and troubleshooting activities for LNS equipment through human-centric, AI-enabled guidance. The pilot combines an AI-based troubleshooting assistant with hands-free XR visualisation in order to provide technicians with contextual instructions directly during maintenance and repair activities.

The pilot addresses two principal activities:

- **Maintenance guidance:** accessing and following equipment-specific procedures while working on LNS equipment.

- **Troubleshooting assistance:** describing a technical problem, receiving relevant diagnostic information and following corrective instructions.

In both activities, the level of detail of the guidance is adapted to the user's technical knowledge. Less-experienced users can receive detailed explanations and intermediate checks, while experienced technicians can use more concise instructions through the additional expert level.

The Pilot #6 Full Prototype has been co-developed by LNS, SSF, RW, SUPSI, INNOV and OCU. LNS contributed domain knowledge and industrial expertise, while RealWear enabled hands-free interaction and supported integration and testing on the Arc 3. SUPSI provided the troubleshooting gateway connecting the services and directing requests within the XR workflow. INNOV refined the XR-enabled AI Assistant (T4.3) to deliver equipment-specific guidance tailored to users' expertise, including an additional expert level. Maintastic's SHARE platform delivered the generated procedures as step-by-step instructions through the wearable interface, and the Swiss Smart Factory (SSF) supported integration and technical testing in an industrial setting.

The broader support concept also incorporates Helen S, LNS's existing digital customer-support assistant, as a complementary entry point for requests that do not initially require wearable assistance. Users can manually transition to the XR application when hands-free guidance is beneficial while working directly on the equipment, re-entering the relevant equipment and problem information.

The technical baseline of the Pilot #6 Early Prototype is described in D6.3, Section 7, with component integration and initial deployment covered in Sections 7.3 and 7.4.

7.1 PILOT SPECIFIC INFRASTRUCTURE

The Pilot #6 Full Prototype infrastructure is as follows:

- Hardware:

  - Pilot 6 uses LNS equipment as the target of maintenance and troubleshooting activities. The pilot infrastructure includes an LNS bar feeder at the Swiss Smart Factory, conventional computing devices for accessing digital support, and RealWear wearable hardware for hands-free interaction.

  - The Early Prototype configuration included the RealWear Navigator 520. For the Full Prototype, the Pilot #6 application and associated troubleshooting workflow were additionally integrated and tested on the RealWear Arc 3.

  - The RealWear Arc 3 provides a see-through display and voice-controlled interaction, allowing technicians to consult instructions while keeping their hands available for work on the

equipment. In Pilot #6, the wearable device acts as the interaction and presentation interface for the troubleshooting workflow.

- The additional deployment supports the objective of reducing dependence on a single wearable configuration. Operation on the Arc 3 provides evidence of portability to an additional platform; compatibility with further XR devices remains subject to integration and testing.

- Software:

  - **T3.3 — XR5.0 Clawdite Platform**, provided by SUPSI, used for the Troubleshooting Assistant integration layer to connect pilots' software components for a smooth interaction.
  - **T4.3 — XR5.0 AI Assistant**, provided by INNOV, used for retrieval-augmented generation of equipment-specific maintenance and troubleshooting guidance.
  - **T4.5 — Maintastic SHARE AR Visualisation**, provided by OCU, used to deliver and present workflows and guidance through the wearable application.

  - RealWear ARI AI voice interface to enable hands-free interaction with the AR glasses.

  - Helen S, LNS's existing digital customer-support assistant, used to support concept as an additional user entry point. Helen S is distinguished from the XR5.0 AI Assistant developed under T4.3. This extension allows users to access AI-supported troubleshooting without requiring XR hardware for every support request.

  The separation between troubleshooting services and the user-facing application supports reuse of the underlying functionality across different interfaces. However, each additional interface requires verification of its connections, supported interactions and presentation requirements.

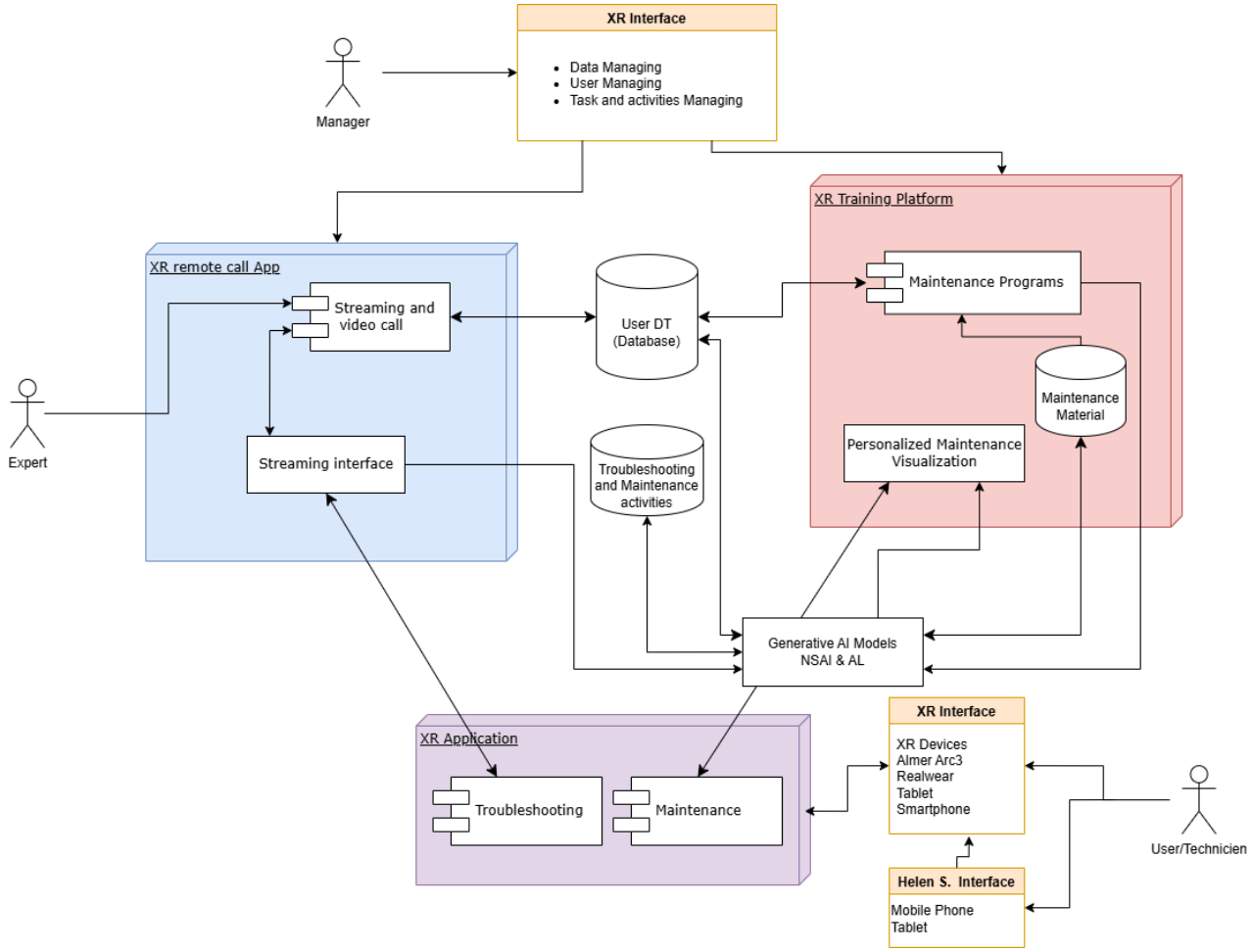


Figure 98 – Pilot #6 Architecture including Helen S, AI Troubleshooting Services and RealWear Arc 3

7.2 Pilot-specific Installation, Customisation and Configuration of XR5.0 Technologies

7.2.1 Troubleshooting Assistant (Gateway)

The Troubleshooting Assistant and its gateway remain the central connection between the troubleshooting logic and the user-facing interfaces. The technical principles of the component are described in D6.3 Section 7.2.1.

Since D6.3, significant effort has been dedicated to improving and testing the overall integration rather than fundamentally changing the component architecture. The interfaces between the AI troubleshooting functions and the XR application were repeatedly tested using representative maintenance and troubleshooting interactions. Particular attention was given to the consistency of the generated instructions, the transfer of troubleshooting steps to the user interface, and the behaviour of the complete workflow when executed on the target wearable device.

The integration work also supported the extension towards multiple user interfaces. The troubleshooting functionality is no longer considered to be accessible only through an XR headset. Instead, the gateway enables the troubleshooting logic to form part of a broader support workflow in which users can begin with

a conventional AI support interface and move to XR when additional visual or hands-free assistance is required.

This separation between the troubleshooting logic and its presentation is an important step towards the Pilot's objective of a more device-agnostic support architecture.

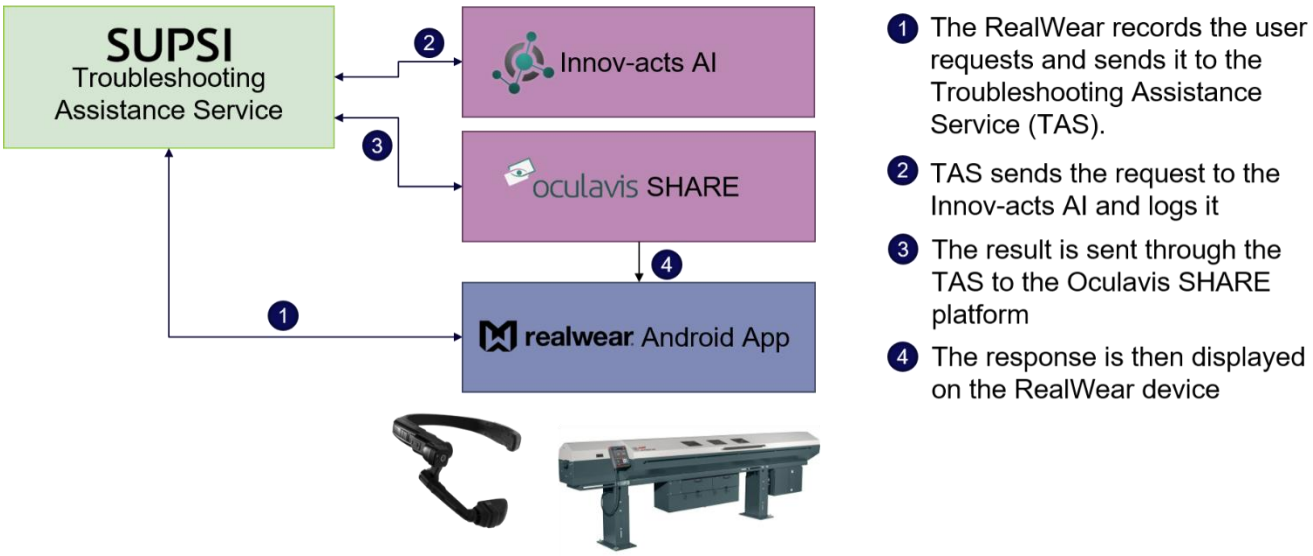


Figure 99 – Pilot #6 Troubleshooting Assistant Service Architecture

7.2.2 AR Visualisation and RealWear Arc 3 Integration

The AR visualisation principles and the initial RealWear integration are described in D6.3 Section 7.2.2. D6.3 already demonstrated the visualisation of troubleshooting instructions and interaction through the RealWear ARI AI voice interface.

Following D6.3, the solution was additionally integrated and tested on the new RealWear Arc 3 platform. The integration was successful and demonstrated that the Pilot #6 application and associated AI-supported workflow can be transferred to a new generation of wearable hardware without redesigning the complete troubleshooting solution.

This validation is particularly important for the Pilot because one of the deployment objectives is to reduce dependence on a single XR device. The application logic and AI services are therefore kept as independent as possible from the specific visualisation hardware, while the wearable device acts primarily as the interaction and information-presentation layer.

Testing on the Arc 3 included the execution of the Pilot #6 troubleshooting workflow, interaction with the AI-supported instructions and the presentation of maintenance guidance through the wearable interface. The successful deployment provides evidence that the architecture can accommodate evolving XR hardware and supports the longer-term objective of making the Pilot solution usable by a wider range of industrial users.

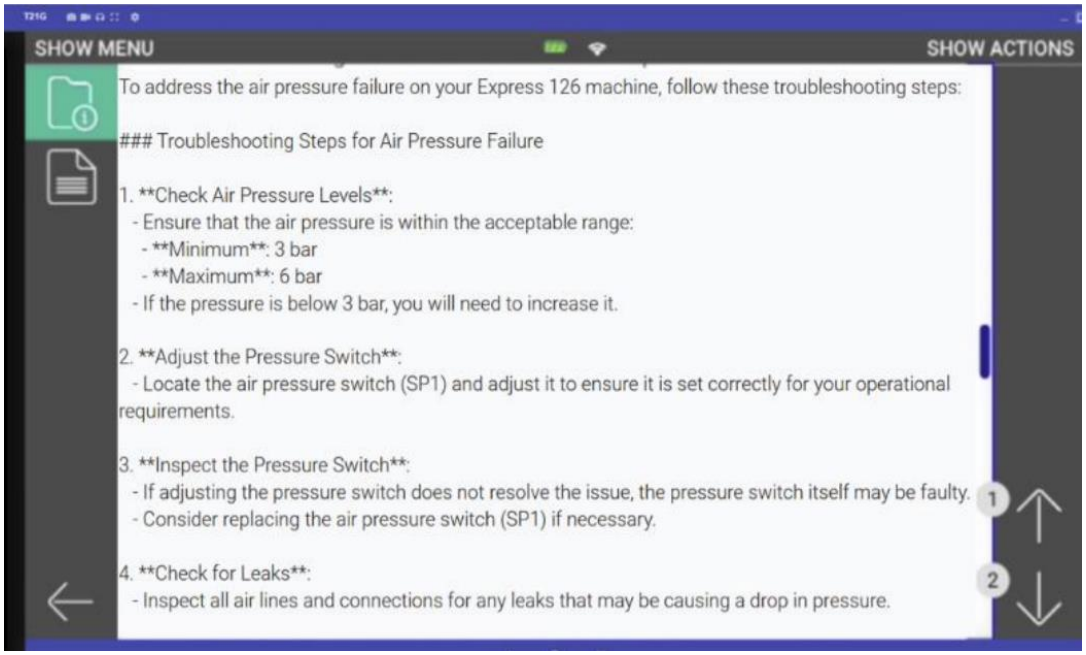


Figure 100 – Visualization of Troubleshooting Instructions

7.2.3 XR-enabled AI Assistant and Expertise-level Personalisation

The XR-enabled AI Assistant described in D6.3 Section 7.2.3 has undergone the largest functional improvements since the Early Prototype.

The main development focused on improving the quality and usability of the AI-generated troubleshooting guidance. During the integration and testing activities, the AI behaviour was iteratively refined so that responses are more suitable for practical service and maintenance situations. In particular, the instructions were improved to provide clearer actions and a more appropriate amount of information for the person carrying out the task.

An important new development is the introduction of an explicit **expertise-level parameter**. The level of technical knowledge of the user can now be taken into account when generating the troubleshooting or maintenance instructions. This makes it possible to vary both the amount of explanation and the granularity of the generated procedure.

For a user with limited experience, the AI can provide detailed, sequential instructions with additional explanations and intermediate checks. For a user with greater technical knowledge, unnecessary explanatory steps can be reduced. With the newly added **expert level**, the system can provide significantly more concise instructions that focus on the relevant technical action and assume knowledge of standard maintenance operations.

The expert-level implementation therefore goes beyond simply changing the wording of the response. It influences how the troubleshooting procedure is presented to the user: elementary actions that an expert can reasonably be expected to know do not need to be explained in detail, while important diagnostic information, warnings and equipment-specific actions remain available.

This development directly implements one of the central human-centric objectives of Pilot #6: avoiding both information overload for experienced technicians and insufficient guidance for less-experienced users. The same underlying technical knowledge can therefore be presented differently according to the user's know-how.

7.2.4 Integration of Helen S and Progressive Escalation to XR

A new component integrated within the Pilot #6 Full Prototype is Helen S, LNS's AI-supported customer-service assistant, for overall support workflow. Helen S is an existing LNS digital customer-support assistant and is distinguished from the XR5.0 AI Assistant developed under T4.3. Helen S provides users with continuous access to answers and step-by-step support for common service activities. LNS describes Helen S as providing personalised and intuitive AI assistance for tasks such as installing a filter, replacing a belt or troubleshooting a problem.

Within Pilot #6, Helen S is used as a complementary first level of assistance. Not every service request requires the user to put on XR glasses. Straightforward questions or relatively simple procedures can be solved directly through Helen S. The purpose of this extension is to make initial assistance accessible without requiring wearable hardware for every request. Users can begin through a conventional support interface and access XR guidance when the procedure benefits from hands-free consultation of instructions.

At the support-concept level, the workflow comprises:

1. The user raises a service question through Helen S.
2. Initial assistance is provided through the conventional support interface.
3. If additional wearable assistance is appropriate, the user accesses the Pilot #6 XR troubleshooting workflow.

For more complex cases, the workflow can transition from this conventional support channel to the XR-supported troubleshooting process. XR is selected when the procedure benefits from hands-free interaction, contextual visual information or a longer sequence of actions that must be performed directly on the machine.

This approach extends the available interaction options while retaining XR for activities where it provides practical value. Task complexity alone does not determine the interface: access to hardware, the working environment and the need for hands-free operation are also relevant.

The resulting support concept can therefore be described as a progressive assistance approach:

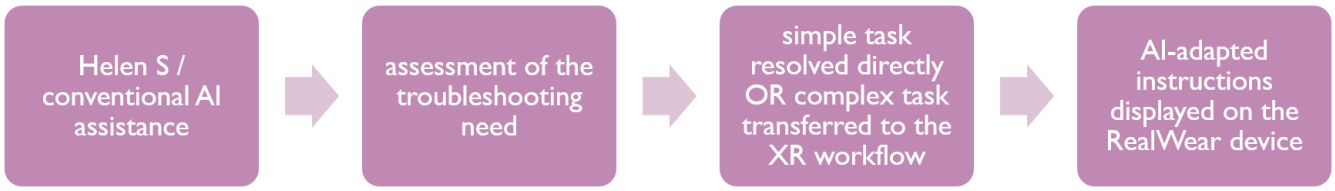


Figure 101 – Helen S and Progressive Escalation Workflow

This extension is particularly relevant to the device-agnostic objective of Pilot #6. The goal is not to require XR for every customer interaction, but to use XR where it provides a clear additional benefit. Consequently, the Pilot can reach a larger population of users while preserving advanced XR support for technically demanding maintenance and troubleshooting operations.



Figure 102 – Pilot #6 LNS Troubleshooting Process

7.2.5 End-to-End Integration and Technical Testing

Technical testing of the Pilot 6 Full Prototype covered three aspects: 1) Deployment on the RealWear Arc 3, 2) expertise-dependent guidance, and 3) a manual transition from Helen S to the XR application. These activities checked the operation of the wearable workflow, the presentation of instructions for different user profiles, and the use of conventional support alongside hands-free assistance.

The Arc 3 deployment test covered access to the Pilot #6 application, submission of an equipment-specific troubleshooting request and presentation of the resulting instructions through the wearable interface. The expertise-dependent guidance test compared responses to the same request at different expertise levels, focusing on the amount of explanation and the granularity of the procedure.

The Helen S scenario involved a manual transition between two separate interfaces. The user began with conventional assistance through Helen S and then opened the Pilot #6 XR application when hands-free guidance was appropriate. The relevant equipment and problem information was entered again in the XR application. No automated handover or transfer of conversation history was involved.

Table 13 – Pilot #6 Technical Test Scenarios and Acceptance Criteria

Test activity	Configuration and scenario	Observed result	Scope and limitations
Arc 3 deployment	The Pilot 6 application was deployed on the RealWear Arc 3 and tested using an LNS equipment troubleshooting workflow. The interaction covered submission of a request and consultation of the resulting guidance.	The application operated on the Arc 3, and AI-supported instructions were presented through the wearable interface. The test demonstrated execution of the troubleshooting workflow on the additional platform.	The result applies to the tested configuration and workflow. Compatibility with further devices and operation across different industrial conditions require additional testing.
Expertise-dependent guidance	The same troubleshooting request was examined using different expertise settings, including the newly introduced expert level. Responses were compared for explanatory detail and procedural granularity.	Less-experienced users received more detailed instructions, while the expert setting produced more concise guidance focused on the relevant technical actions. The comparison informed further refinement of the AI responses.	Expertise adaptation changes how guidance is presented. Technical correctness, completeness and retention of essential warnings remain important criteria for continued evaluation.
Manual transition from Helen S to XR	The user began a support interaction through Helen S, then manually opened the Pilot 6 XR application and re-entered the relevant equipment and problem information to continue with wearable guidance.	The scenario demonstrated a support journey combining conventional digital assistance with hands-free XR guidance through a manual transition.	The interfaces operated separately. No shared session, automated handover or transfer of conversation history was implemented; continuity depended on the information re-entered by the user.

These activities demonstrated operation of the tested Arc 3 workflow and supported refinement of expertise-dependent instructions. They also clarified Helen S's role as a complementary support entry point, with a manual transition to the XR application. The results provide evidence of technical functionality within the tested scope; improvements in maintenance time, troubleshooting effectiveness and user acceptance require separate evaluation with representative users.

7.3 Component Usage Matrix

Table 14 gives an overview of the XR5.0 technical components integrated within the Pilot #6 Full Prototype.

Table 14 – XR5.0 Technical Components in Pilot #6 Full Prototype

XR5.0 Technical Tasks	Technical Component	Technical Partner	How is the Component Used?
T3.1 Specifications for Worker Digital Images	N/A		
T3.2 Workers' Data Collection & Analytics	N/A		
T3.3 Workers' Digital Twins	XR5.0 Clawdite Platform (for Troubleshooting Assistant)	SUPSI	Acts as an integration layer between external services (voice interface, AI assistant and Maintastic SHARE) and the XR environment.
T3.4 Personalized XR Content Development	N/A		
T3.5 Adaptation of Personalized XR Content	N/A		
T4.1 Human-centered XAI Models	N/A		
T4.2 XR-enabled AL & NSAI Models	N/A		
T4.3 Generative AI Models	XR5.0 AI Assistant	INNOV	Deployed as a RAG-based chatbot adapted with LNS's industrial maintenance documentation to support bar feeder operations by tailoring guidance to worker experience and assisting with predictive maintenance and troubleshooting.
T4.4 Visualization of XAI Explanations	N/A		
T4.5 Visualization of AI Recommendations	Maintastic SHARE AR Visualization	OCU	Maintastic SHARE delivers digital workflows and step-by-step guidance from the AI Assistant directly to RealWear XR devices, which are visualized in AR by the Android RealWear app.
T5.1 Training Material	N/A		

T5.2 Cloud-based XR Training Asset Repository	N/A		
T5.3 Hololight Hub	N/A		
T5.4 Training Programs	N/A		
Pilot-specific Components	RealWear ARI AI Voice Interface	RW	Enables hands-free, voice-controlled interaction on Navigator XR glasses by converting speech to text and forwards requests to the Troubleshooting Assistant on the Arc 3.
Pilot-specific Components	Helen S - LNS	LNS	Users can enter the support process through Helen S for straightforward requests and use XR for procedures where contextual and hands-free guidance offers additional value.

Table 15 provides a comprehensive overview of the components employed in the two use cases of Pilot #6. The primary distinction lies in the integration of an AI chatbot in UC6.1 (Human Centred Preventative Maintenance Guidance), where responses and instructions are customized to the specific needs of the workers during a troubleshooting process. In contrast, UC6.2 (Troubleshooting for repair) involves steps that are predetermined and adaptable to varying levels of expertise among workers.

Table 15 – Mapping of Pilot #6 Technical Components to Pilot Use Cases in Full Prototype

	UC6.1	*UC6.2*
Troubleshooting Assistant (Gateway)	X	X
AR Visualisation (Maintastic SHARE)	X	X
AI chatbot (XR5.0 LLM Engine)	X	
RealWear ARI AI Voice Interface	X	X
Helen.S digital customer-support assistant		X

7.4 DISCUSSIONS

Since D6.3, Pilot #6 has progressed from an Early Prototype demonstrating the core interaction between AI-supported troubleshooting and XR visualisation towards a more mature and flexible support workflow.

The most important improvement has been the refinement of the AI assistance. Testing showed that the usefulness of troubleshooting instructions depends strongly on the amount of prior knowledge of the user. Providing identical instructions to every technician would either result in insufficient support for inexperienced users or unnecessary information for experienced personnel. The implementation of the additional expert level addresses this issue by enabling the same troubleshooting knowledge to be delivered at an appropriate level of detail.

Integration on the Arc 3 extends the wearable configurations supported by the pilot. The results reported in Section 7.2.5 identify the functions tested and the adaptations required. This provides a concrete basis for assessing portability while recognising that additional platforms require their own integration and verification.

A third development is the integration of Helen S into the Pilot workflow. This creates a broader and more practical support concept in which XR is not treated as the mandatory starting point for every maintenance request. Users can initially solve simple questions through an easily accessible AI assistant, while XR becomes an escalation path for complex operations where visual, contextual and hands-free instructions are advantageous.

These developments make Pilot #6 more accessible to a larger user population while at the same time strengthening its human-centric characteristics. The final Pilot 6 system combines the accessibility of conventional AI assistance, the personalisation of expertise-aware troubleshooting instructions, and the contextual support capabilities of industrial XR devices.

8. CONCLUSIONS

8.1 SUMMARY

The six XR5.0 full prototypes demonstrate how Extended Reality (XR), Artificial Intelligence (AI), and Human Digital Twins (HDTs) support industrial training, inspection, maintenance and troubleshooting. Building on the early prototypes reported in D6.3, D6.4 documents more complete workflows, closer integration between technical components, and changes made in response to pilot users and operational constraints. The pilots differ in how they use XR: some rely on immersive training or hands-free guidance, while others use mobile devices as the most practical starting point.

Pilot #1 – T6.2 Rapid Human-Centric AI-enabled Product Design combines digital twins, generative AI, XR training and Industry 4.0 standards to support technicians during assembly and virtual commissioning and improve safety in human–robot collaboration. Its augmented reality application allows technicians to inspect KUKA robots at true scale, isolate individual components, and visualise simulated movements or live joint positions. Real-time machine data appears alongside the virtual model, while voice interaction and an AI assistant provide access to technical documentation and explanations. Delivered through the XR5.0 Training Platform, the application supports central management of users, applications and immersive training programmes. Predictive safety functions use worker-position data to identify potential risks and issue visual and audible warnings through the headset. The Asset Administration Shell framework supports standardised exchange of asset and operational data across the pilot's systems.

Pilot #2 – T6.3 Human-Centred Remote Maintenance & Asset Management integrates generative AI, explainable AI and XR to support technicians across the maintenance process of specialized laboratory equipment. Technicians can begin troubleshooting by voice or text on a mobile device. The AI assistant uses asset and service-case information to retrieve equipment-specific documentation, provide guidance and explain the sources behind its recommendations. Maintenance cases can also be escalated to remote experts through Maintastic SHARE, with the relevant asset, ticket and documentation context available during the session. When a task benefits from spatial or hands-free guidance, technicians can select a structured procedure on the mobile device and open its step-by-step instructions on HoloLens 2 through a QR-code handover. Reusable XR content supports both maintenance and training.

Pilot #3 – T6.4 XR Streaming for Operator 5.0 Training integrates mobile AR, **AI-based anomaly detection, explainable AI, generative AI and structured training** to support water infrastructure inspection and maintenance. Adapted to smartphones and tablets in response to operational and security requirements, the application enables field operators to view geolocated underground pipelines, sensor information, detected anomalies, explanations and recommendations in their working environment. The Smart Pipes API supplies sensor and anomaly data, while Neurosymbolic AI analyses CCTV inspection footage and Active Learning enables experts to validate and correct predictions, supporting iterative model improvement. A text- and voice-enabled AI assistant provides contextual guidance and access to technical documentation. The integrated, mobile-adapted XR Training Plugin delivers structured training scenarios covering equipment and safety checklists and water-pipe maintenance procedures.

Pilot #4 – T6.5 Personalised Aircraft Maintenance Training provides mixed reality and virtual reality training for Aircraft Maintenance Technicians (AMTs), focusing on Wing Anti-Ice Valve (WAIV) removal and installation. The full prototype combines **XR5.0 training materials and programmes**, **generative AI**, **AI-based object recognition**, **human digital twins**, **workers' data collection**, and **personalised content adaptation**. Technicians explore interactive 3D models, follow step-by-step procedures, and access a voice-enabled AI assistant that answers questions using training documentation and provides source references. Aircraft-component recognition supports expert annotation and correction, Active Learning, and on-demand explanations. The human digital twin replays expert movements and monitors trainees against

procedural milestones to provide corrective feedback, while heart-rate monitoring supports worker-state assessment and an adaptation service tailors content presentation to each technician's profile and context.

Pilot #5 – T6.6 XR Training for Effective Assembly Operations supports SPACE's technicians in assembling, commissioning and repairing the Guardian edge controller through a single XR application integrated with XR5.0 services. Technicians can inspect a detailed virtual device, send configuration commands and monitor real-time device status. The Full Prototype combines the **XR5.0 Training Platform**, immersive **training programs**, **worker data collection**, **stress analysis and adaptation services**, and **generative AI**. Training supervisors author progressively challenging scenarios, from device familiarisation to practical commissioning, with objectives, documentation and knowledge checks delivered in the headset. Worker profiles and smartwatch biometrics support personalisation, while deterministic and learned stress estimation provide feedback used to adjust the difficulty of subsequent levels. Technicians can view their physiological state alongside assessment results. For repair support, a voice-enabled AI assistant provides guidance grounded in Guardian documentation, alongside device-message monitoring in the headset.

Pilot #6 – T6.7 Human Centric Guidance and Troubleshooting demonstrates AI-driven, human-centric support for industrial maintenance and troubleshooting through **generative AI** and **AR visualisation**. The full prototype combines LNS equipment (an upgraded LNS bar feeder at the Swiss Smart Factory), AI troubleshooting services and voice-controlled XR interfaces (RealWear ARI), with wearable support extended to the **RealWear Arc 3**. AI-generated instructions have been refined through technical testing, and an explicit expertise-level parameter, including an additional expert level, adapts their detail to technicians' knowledge. **Helen S**, LNS's existing digital customer-support assistant, provides a complementary entry point, with users manually transitioning to the XR application when hands-free guidance is beneficial.

Table 16 shows the mapping of XR5.0 technical components to the Pilots' Full Prototypes. The cell in dark gray colour indicate the technical components that were integration into the pilot full prototypes after D6.3.

Table 16 – Mapping of XR5.0 Technical Components to the Pilots' Full Prototypes

	Pilot #1	**Pilot #2**	**Pilot #3**	**Pilot #4**	**Pilot #5**	**Pilot #6**
T3.1 Specifications for Worker Digital Images	N/A	N/A	N/A	N/A	N/A	N/A
T3.2 Workers' Data Collection & Analytics	NOVAAS Frame-work (UNP)	N/A	N/A	XR5.0 Activity Metrics Application for physiological data (UNP)	Smartwatch Connector and Stress Index Calculator (UNP, IPIA, SUPSI)	N/A
T3.3 Workers' Digital Twins	XR5.0 Clawdite Platform for worker movement	N/A	N/A	Worker Shadowing and Monitoring through the XR5.0	XR5.0 Clawdite Platform HDT for worker profiles and	XR5.0 Clawdite Platform (Trouble-shooting

	and position data (SUPSI)			Clawdite Platform (SUPSI)	biometric data (SUPSI)	Assistant) (SUPSI)
T3.4 Personalized XR Content Development	N/A	N/A	N/A	N/A	Training levels tailored to worker skill and stress state (SYN, SPACE)	N/A
T3.5 Adaptation of Personalized XR Content	N/A	N/A	N/A	Adaptation Engine for training-content presentation (SUPSI, IML)	Training Scenario Adaptability Module for selecting the next training level (SYN)	N/A
T4.1 Human-centered XAI Models	Textual transparency for AI assistant responses (SIE)	CoT-enabled Textual Explanations in the Mobile Trouble-shooting AI Assistant (OCU)	XR5.0 XAI Suite for Glass-box Models and Post-hoc Methods (UPRC)	N/A	Stress Detection Module, XAI layer (UPRC)	N/A
T4.2 XR-enabled AL & NSAI Models	N/A	N/A	AL & NSAI-based Anomaly Detection (UPRC)	WingValve AI object detection with expert feedback and Active Learning (UPRC)	Stress Detection Module, AI and Active Learning layers (UPRC)	N/A
T4.3 Generative AI Models	AI Chat/Voice Assistant KUKABot (SIE)	Mobile Trouble-shooting AI Assistant (OCU)	AI Chat/Voice Assistant (INNOV)	AI Chat/Voice Assistant (SIE)	XR5.0 Chat Engine for device documentation (INNOV)	AI Chat/Voice Assistant (INNOV)
T4.4 Visualization of XAI Explanations	Assistant response	Textual XAI Visualization	XAI Visualiza-	N/A	Stress and biometric	N/A

	and reasoning display in the XR application (HOLO)	s in the Mobile Trouble-shooting AI Assistant (OCU)	tions in the mobile App (CYENS)		information displayed in the XR application (SYN)	
T4.5 Visualization of AI Recommendations	KUKABot UI in XR App (HOLO)	Textual Visualizations in the Mobile Trouble-shooting AI Assistant (OCU)	AI Visualizations in the AR App (CYENS)	AI Chat/Voice Assistant UI in VR App (IML)	Assessment results and AI assistant responses in the XR application (SYN)	Troubleshooting workflows and guidance displayed through Maintastic SHARE on the XR device (OCU)
T5.1 Training Material	KUKA smartPAD training materials and platform-authored content (IML)	Reusable XR maintenance and training content (IML)	Water-pipe inspection and maintenance training materials (IML)	Training Materials generated using 3D Workflow Tool and 3D Model Explorer for WAIV Removal and Installation (IML)	SPACE Guardian Device Training Materials (IML)	N/A
T5.2 Cloud-based XR Training Asset Repository	Training assets delivered through the XR5.0 Training Platform (HOLO)	Reusable XR maintenance and training assets linked to the XR5.0 Training Platform (OCU)	N/A	Cloud-based XR Training Asset Repository (SYN)	XR5.0 Training Platform Asset Repository (HOLO, SYN)	N/A
T5.3 Hololight Hub	Cloud hosting and streaming of the XR application (HOLO)	N/A	N/A	Cloud delivery and streaming of the XR training	Hosting and streaming of the XR application (HOLO)	N/A

				application (HOLO)		
T5.4 Training Programs	KUKA smartPAD-2 and platform-authored training programs (IML)	N/A	Water-pipe maintenance training program (IML)	Training Programs for WAIV Removal and Installation (IML)	SPACE Guardian Device Training (IML)	N/A
Pilot-specific Components	Safety-risk Zone Checker (ATB)	Maintastic SHARE maintenance and remote-support workflow (OCU)	ETL Workflow and Smart Pipes API (SIE)	XR-AI Annotator for aircraft-component images (UPRC)	Bidirectional MQTT integration with the physical Guardian device (SYN, SPACE)	RealWear ARI AI voice interface and Arc 3 integration (RW)
	KUKA OPC UA Server (KUKA)	QR-code handover to HoloLens 2 (OCU)				
	Robot and position-estimation Asset Administration Shells (UNP)					Helen S customer-support assistant (LNS)

8.2 LESSONS LEARNED

The six XR5.0 full prototypes demonstrate the technical feasibility of combining XR, AI, training services and industrial data across robotics, remote maintenance, water infrastructure, aviation, edge-device operations and equipment troubleshooting. Compared with the early prototypes reported in D6.3, they bring more components into connected workflows and provide a clearer picture of how workers can use these technologies in practice. The results also show that the value of XR depends on the task, the working environment and the quality of the guidance presented to the user.

- **Key successes:** The pilots integrated capabilities that had previously been demonstrated separately. Pilot 1 connects robot visualisation, live data, AI assistance, training and safety warnings in a cloud-delivered XR application. Pilot #2 links asset and service-case information with mobile troubleshooting, remote experts and XR maintenance procedures. Pilot 3 combines mobile AR with pipe and sensor data, anomaly analysis, explanations, AI guidance and training. Pilot #4 brings

immersive aircraft-maintenance training together with AI assistance, component detection, worker monitoring and content adaptation. Pilot #5 delivers authored training scenarios connected to a physical device, with biometric feedback and adaptive level selection. Pilot 6 combines expertise-sensitive troubleshooting with conventional AI support and hands-free XR guidance. Together, these implementations show that XR5.0 components can support connected industrial workflows across different domains.

- **Challenges:** Integration remains demanding when industrial systems, AI services, training platforms, worker-data services and XR devices are developed independently. Although more mature service interfaces have reduced the need for pilot applications to connect components through custom logic, reliable end-to-end operation still depends on compatible data formats, authentication, network access and clear responsibility for each part of the workflow. Device performance and interaction also affect usability: headset gestures, voice recognition, visual clutter and response times can make otherwise useful guidance difficult to follow. Mobile AR introduces its own constraints, including GPS and compass accuracy, spatial alignment and limited screen space. Cloud streaming and connected AI services depend on stable connectivity, which cannot be assumed throughout a hangar, shop floor or field site.
- **Cross-pilot findings:** The pilots support selecting the interaction method according to the task. In Pilot #2, familiar mobile devices provided an efficient starting point for troubleshooting, while XR was valuable when technicians needed spatial information or hands-free instructions. Pilot #3's move to mobile AR responded to operators' practical and security requirements. Pilot #6 similarly allows straightforward questions to be addressed through a conventional support assistant before moving to XR for more complex procedures. These examples suggest that XR is most useful when it adds a clear benefit to a specific stage of work, rather than serving as the required interface for every request.

  The quality and presentation of AI guidance are equally important. Equipment-specific documentation, source references and explanations help technicians and experts assess recommendations. Optional explanations can provide transparency without adding information to every routine instruction. Human review is particularly valuable for image-based AI: in Pilots #3 and #4, expert feedback can correct or validate findings and contribute to subsequent model development. Pilot #6 shows why instructions must also reflect the worker's expertise; the same procedure may require detailed intermediate steps for a novice and a concise presentation for an experienced technician.
- **Adaptation and worker data:** The full prototypes provide more concrete examples of personalised support. Pilot #4 uses worker context and movement information to tailor training presentation and provide procedural feedback. Pilot #5 combines worker profiles, smartwatch measurements and assessment results to select training levels. These approaches show how worker-related data can inform an XR experience, but also reveal the need to make adaptation understandable to its users. In particular, the learned stress-detection module in Pilot #5 requires validation with representative pilot data; its implemented stress-index path provides a basis for adaptation in the meantime.
- **Remaining risks and limitations:** The prototypes establish working capabilities, but their results should not be interpreted as measured improvements in task completion time, error rates, learning outcomes or long-term worker acceptance across all six pilots. Some functions have been tested in representative scenarios rather than across the full range of operating conditions. Access to aircraft and specialised equipment can constrain testing, while variable connectivity can affect streamed applications and remote services. The use of proprietary maintenance documentation, training content and worker-related data also requires appropriate access controls and data handling. For broader industrial use, organisations will need to assess the reliability of AI guidance, the suitability of each device and the support needed by workers with different levels of XR experience.
- **Implications for evaluation and future deployment:** Evaluation should examine complete tasks with end users, including whether guidance is accurate and understandable, whether workers can

move smoothly between mobile, XR and expert support, and whether adaptive features provide useful assistance. Technical assessment should cover connectivity, spatial accuracy, device responsiveness, service integration and recovery from failures. For future real-world deployment, the pilots point to modular services, reusable training content, traceable AI recommendations and a choice of interaction methods as important design principles. The appropriate combination should be determined by the operational task and validated with the workers who will use it.

The XR5.0 Pilot full prototypes provide a stronger basis for evaluation and potential real-world deployment than the early prototypes documented in D6.3. The full prototypes identify the conditions that will matter most when assessing their practical value. The next evidence needed is systematic assessment with end users under representative conditions, including usability, technical reliability, the quality of AI guidance and adaptation, and the practical value of XR within each workflow. These findings can inform future deployment of human-centric, AI-enabled XR solutions beyond the project.