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

D4.2 – ADVANCED AI PARADIGMS FOR HUMAN-AI COLLABORATION V2

Lead Beneficiary	ATB
Work Package Ref.	WP4 – AI/XR Symbiosis for Augmented Intelligence
Task Ref.	T4.1 – AI Models and Infrastructures for Trusted Human-AI Collaboration T4.2 – XR-Enabled Human-AI Collaboration (Neurosymbolic AI, Active Learning) T4.3 – XR-Enabled Generative AI
Deliverable Title	D4.2 – Advanced AI Paradigms for Human-AI Collaboration v2
Due Date	2026-03-31
Delivered Date	2026-03-31
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

COPYRIGHT MESSAGE

CONTRIBUTING PARTNERS

Partner Acronym	Role[1]	Name Surname[2]
ATB	Lead Beneficiary	Fulya Horozal, Sebastian Scholze
INNOV	Contributor	George Fatouros, Despoina Tomkou
OCU	Contributor	Mathias Horner
SIE	Contributor	Robert Stefan Fejer
UPRC	Contributor	Stergios Chairistanidis Dimitrios Dardanis
SUPSI	Internal Reviewer	Sara Masiero
UBI	Internal Reviewer	Nikolaos Tousert
UPRC	Quality Assurance	Thanos Kiourtis, Argyro Mavrogiorgou

REVISION HISTORY

Version	Date	Partner(s)	Description
0.1	2026-01-22	ATB, INNOV, OCU, SIE, UPRC	ToC Version
0.2	2026-02-12	ATB, INNOV, OCU, SIE, UPRC	Initial Partner Contributions
1.0	2026-03-18	ATB, INNOV, OCU, SIE, UPRC	1st Version
1.1	2026-03-19	ATB	Version for Peer Reviews
1.2	2026-03-28	ATB, INNOV, OCU, SIE, UPRC	Addressed review comments
2.0	2026-03-30	ATB	Version for Quality Assurance
2.1	2026-03-30	UPRC	Quality Assurance revision
2.2	2026-03-30	ATB	Addressed QA comments
3.0	2026-03-31	ATB. GFT	Version for Submission

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

LIST OF ABBREVIATIONS

Acronym	Definition
AI	Artificial Intelligence
AL	Active Learning
API	Application Programming Interface
AR	Augmented Reality
BERT	Bidirectional Encoder Representations from Transformers
CBM	Concept Bottleneck Models
CCTV	Closed-Circuit Television
CLIP	Contrastive Language–Image Pre-training
CRUD	Create, Read, Update, Delete
CNN	Convolutional Neural Network
CoT	Chain-of-Thought
DL	Deep Learning
DoA	Description of Action
ETL	Extract, Transform, Load
FIFO	First In, First Out
GenAI	Generative AI
GPT-4	Generative Pre-trained Transformer 4
Grad-CAM	Gradient-weighted Class Activation Mapping
HITL-ML	Human-in-the-loop Machine Learning
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
JWT	JSON Web Token
LIME	Local Interpretable Model-agnostic Explanations
LlaMA	Large Language Model Meta AI
LLM	Large Language Model
LNN	Logical Neural Network
LTN	Logical Tensor Network
ML	Machine Learning
MPNet	Masked and Permuted Pre-training for Language Understanding
MT	Machine Teaching
NN	Neural network
NSAI	Neurosymbolic Artificial Intelligence
OCR	Optical Character Recognition
RAG	Retrieval Augmented Generation
SaaS	Software as a Service
STT	Speech-to-Text
TRL	Technology Readiness Level
TTS	Text-to-Speech
URL	Uniform Resource Locator
VLM	Vision Language Model
VMLC	Visual Machine Learning Control
VR	Virtual Reality
VSA	Vector Symbolic Architecture
WP	Work Package
XAI	eXplainable AI
XR	Extended Reality

EXECUTIVE SUMMARY

Deliverable D4.2, *Advanced AI Paradigms for Human-AI Collaboration v2*, represents the follow-up of Deliverable D4.1, *Advanced AI Paradigms for Human-AI Collaboration v1,* and constitutes the final deliverable covering Tasks T4.1 – T4.3 of WP4 – *AI/XR Symbiosis for Augmented Intelligence* within the XR5.0 project. Building upon the foundations established in Deliverable D4.1, this deliverable consolidates the progress achieved during Months 13 – 27 of the project and presents the final advancements in AI methodologies and their application in real world Industry 5.0 environments to support human-AI collaboration.

The deliverable reports the development, implementation, and validation of advanced AI approaches across three key tasks: T4.1 – AI Models and Infrastructures for Trusted Human-AI Collaboration; T4.2 – *XR-Enabled Human-AI Collaboration (Neurosymbolic AI and Active Learning)*; and T4.3 – *XR-Enabled Generative AI*. These tasks collectively address the need for transparent, adaptive, and human-centric AI systems capable of supporting industrial operators through explainable decision-making, intelligent knowledge retrieval, and immersive XR-based assistance.

The results presented in this deliverable demonstrate significant progress in the integration of eXplainable AI (XAI), Neurosymbolic AI (NSAI), Active Learning (AL), and Generative AI (GenAI) within the XR5.0 platform. The developed solutions enable enhanced interpretability of AI decisions, continuous human-in-the-loop learning, and dynamic generation of knowledge-based assistance for industrial scenarios. In particular, the Neurosymbolic AI framework combines deep learning perception with symbolic reasoning to provide transparent inspection and classification processes, while Active Learning mechanisms incorporate expert feedback to improve model performance over time.

Complementary to these approaches, eXplainable AI techniques provide additional transparency by revealing the internal reasoning of AI models through visual explanations, feature attribution methods, and interpretable concept-based reasoning. The Post-hoc XAI methods integrated within XR5.0 represent a significant advancement in human-centric explainability for black-box AI models and provide improved transparency and interpretability in industrial AI applications, enabling operators to better understand model predictions, identify potential errors, and maintain trust in AI-assisted decision-making processes within industrial inspection and monitoring tasks. Moreover, glass-box models further strengthen transparency by enabling fully interpretable decision-making processes, where predictions are directly linked to human-understandable concepts and their contributions, allowing domain experts to inspect, validate, and intervene in the reasoning process of AI systems. In addition, human-centric GPT-based interaction mechanisms using Chain-of-Thought (CoT) reasoning enable large language models to externalize their reasoning processes through structured intermediate steps to allow users to trace how AI-generated responses are derived and strengthen transparency and human oversight in AI-supported decision-making.

Furthermore, the deliverable introduces advanced Generative AI-based systems that enable conversational access to industrial documentation and maintenance knowledge through Retrieval-Augmented Generation (RAG) pipelines, customizable AI agents, and multimodal interaction interfaces. These solutions allow operators to interact with complex technical documentation through natural language queries, voice interfaces, and XR-supported workflows, improving efficiency and knowledge accessibility in industrial environments. The integration of explainable reasoning mechanisms, such as Chain-of-Thought prompting, further enhances these systems by providing traceable reasoning steps behind generated responses to allow users to better understand and validate AI-generated recommendations in industrial contexts.

The developed technologies have been implemented and demonstrated across XR5.0 pilots, including industrial inspection, maintenance training, and customer service support scenarios. Evaluation results

indicate that the proposed AI solutions effectively support human-AI collaboration, improve transparency in AI decision-making, and facilitate knowledge-driven assistance in XR environments.

Deliverable D4.2 demonstrates the successful advancement of AI/XR symbiosis for augmented intelligence, contributing to the XR5.0 vision of human-centric, trustworthy, and adaptive AI systems aligned with Industry 5.0 principles. The outcomes presented in this deliverable represent the final stage of the WP4 technical developments and provide a solid foundation for further deployment, evaluation, and exploitation of AI-enabled XR solutions within industrial contexts.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Codes

1. INTRODUCTION

1.1 Objectives of the Deliverable

The primary objective of D4.2, *Advanced AI Paradigms for Human-AI Collaboration v2*, is to present the final progress and advancements achieved in tasks

- T4.1 – *AI Models and Infrastructures for Trusted Human-AI Collaboration*,
- T4.2 – *XR-Enabled Human-AI Collaboration (Neurosymbolic AI, Active Learning*),
- T4.3 – *XR-Enabled Generative AI*

within WP4, *AI/XR Symbiosis for Augmented Intelligence*, of the XR5.0 project. This deliverable consolidates the efforts and outcomes achieved during Months 13–27 of the project and highlights the advancements in AI methodologies and XR integration for augmented intelligence, including the initial implementation of these innovations in the project pilots. The objectives specific to each task are outlined as follows:

Task T4.1 – AI Models and Infrastructures for Trusted Human-AI Collaboration

T4.1 focuses on the integration of trustworthy AI models into XR solutions for Industry 5.0 applications, ensuring transparent, understandable and user-adaptive AI systems that foster effective human-AI collaboration. Main goals include:

- Developing human-centred eXplainable AI (XAI) models tailored for industrial end-users that provide personalized, trusted recommendations and ensure ethical compliance and trustworthiness through responsible AI design.
- Building tools and infrastructures for efficient training, inference, and integration of XAI models into XR platforms within XR5.0.
- Provide Human-Centric GPT-Based interactions to leverage Large Language Models (LLMs) for intuitive and interactive AI explanations.

Task T4.2 – XR-Enabled Human-AI Collaboration (Neuro-Symbolic AI, Active Learning)

T4.2 focuses on the development and integration of Neurosymbolic AI (NSAI) and Active Learning (AL) models into XR environments to enhance human-AI collaboration within Industry 5.0 applications, improving accuracy, efficiency and explainability of AI systems. Main goals include:

- Developing NSAI and Active Learning models for XR-enabled human-AI collaboration
- Integrating NSAI into XR-based industrial applications, combining deep learning and symbolic reasoning to create more robust and XAI models.
- Integrating XR-enabled active learning techniques that involve humans in the machine learning process for better and faster results with XR interfaces to enable a natural interaction between humans and the AI systems.

Task T4.3 – XR-Enabled Generative-AI

T4.3 focuses on the development and integration of Generative AI (GenAI) models into XR environments to facilitate improved collaboration and enhanced creativity between humans and AI systems. Main goals include:

- Leveraging GenAI to create novel, context-aware content and solutions for more efficient and effective collaboration in various industrial scenarios.
- Developing solutions based on state-of-the-art generative models, retrieval-augmented generation, agentic workflows and prompt engineering to enable real-time assistance based on proprietary data (e.g., technical manuals) to human users in I5.0 contexts via user-friendly interfaces for (e.g., chat, speech-to-text and text-to-speech).

Several XR5.0 pilots, including industrial inspection, remote maintenance, asset management, and customer service support scenarios have integrated the outputs of T4.1, T4.2 and T4.3. Table 1 presents an overview of all the AI solutions developed within WP4 and how they align with XR5.0 pilots. All the AI solutions have followed the guidelines for Responsible and Ethical AI as defined by the European Commission in the EU AI Act. To increase the trustworthiness, reliability and interpretability of the AI systems, XAI solutions were integrated to provide explanations in human understandable concepts. Initial demonstration results revealed that the proposed AI solutions effectively foster human-AI collaboration, improve transparency of AI decision-making processes and provide knowledge driven assistance in various I5.0 scenarios.

The list of all six XR5.0 pilots are as follows:

- Pilot 1 (KUKA): T6.2 – Rapid Human-Centric AI-enabled Product Design.
- Pilot 2 (SH): T6.3 – Human Centred Remote Maintenance and Asset Management.
- Pilot 3 (EKSO): T6.4 – XR Steaming for Operator 5.0 Training.
- Pilot 4 (TAP): T6.5 – Personalised Aircraft Maintenance Training.
- Pilot 5 (SPACE): T6.6 – XR Training for Effective Assembly Operations.
- Pilot 6 (LNS): T6.7 – Human Centric Guidance and Troubleshooting.

Table 1 – Mapping of WP4 Technologies to XR5.0 Pilots

WP4	Methodology / Solution Implemented	Description	Pilots Integrated	Technical Partner
T4.1 – AI Models and Infrastructures for Trusted Human-AI Collaboration	Post-hoc eXplainable AI (XAI) – LIME, Grad-CAM	Visual and feature attribution explanations to interpret AI model predictions in inspection scenarios	Pilot 3	UPRC
	Glass-box AI / Concept-based reasoning (CBM reasoning layer)	Transparent reasoning layer enabling concept-level explanations for model decisions	Pilot 3	UPRC
	Human-centric GPT-based Interaction using Chain-of-Thought (CoT)	Structured reasoning traces for conversational AI explanations and decision transparency	Pilot 1	SIE
			Pilot 2	OCU
T4.2 – XR-Enabled Human-AI Collaboration (Neurosymbolic AI, Active Learning)	Neurosymbolic AI (Concept Bottleneck Models)	Combines neural perception with symbolic reasoning for interpretable AI predictions	Pilot 3, Pilot 4	UPRC

	Active Learning (Human-in-the-loop)	Expert feedback integrated into AI training to improve model accuracy and adaptability	Pilot 3, Pilot 4, Pilot 5	UPRC
T4.3 – XR-Enabled Generative AI	LLM Chat Engine with Retrieval-Augmented Generation (RAG)	Conversational AI for querying industrial documentation and providing contextual assistance	Pilot 3, Pilot 5, Pilot 6	INNOV
	GenAI Maintenance Assistant (voice-based troubleshooting assistant)	Mobile conversational AI supporting maintenance diagnostics and XR collaboration	Pilot 2	OCU
	DataLens Conversational AI Framework	Privacy-preserving RAG-based conversational AI for querying proprietary technical documentation	Pilot 1, Pilot 4	SIE

1.2 Relation to Other Work Packages and Deliverables

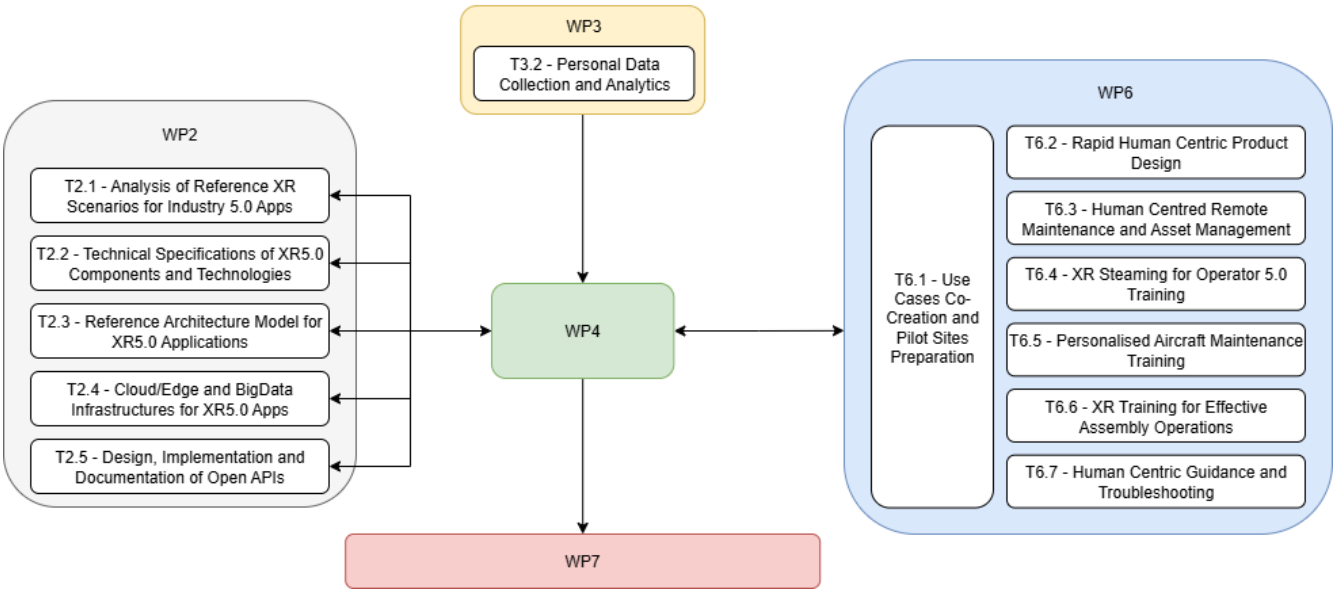

Figure 1 – Relation to Other Work Packages

Figure 1 presents the interactions and dependencies between WP4 and various tasks from other WPs of the XR5.0 project. WP4 interacts with WP2 through the analysis of reference XR scenarios for Industry 5.0 applications (T2.1), the technical specifications for XR5.0 components and technologies (T2.2), the reference Architecture Model (T2.3) as well as the design and implementation of Open APIs (T2.5).

Moreover, WP4 is also associated with T2.4 activities which ensure that XR5.0's models, tools and software can seamlessly operate across diverse cloud, edge, and hybrid computing infrastructures via DevOps pipelines.WP3 contributes to this integration through the task of personal data collection and analytics (T3.2), ensuring that data-driven insights feed into WP4. WP4, in turn, interacts with WP6, which include various deployment, operational and pilot evaluation activities, as well as system implementation and management across tasks T6.1 through T6.6. Finally, WP7 supports the overall project exploitation and dissemination activities, ensuring the seamless interaction and flow of resources between these interconnected components.

WP4 tools, including XAI, AI, NSAI, GenAI, and AR/VR visualization capabilities, have been designed, developed, integrated, and deployed in accordance with the updated XR5.0 reference architecture described in D2.3 – *Reference Architecture Model and Technical Specifications v2*, thus ensuring advanced Industry 5.0 functionalities. This alignment is depicted in the layered architectural framework, which incorporates the Business, Application, and Technology layers, as detailed in D2.2 – *Reference Architecture Model and Technical Specifications v1* and D2.3 – *Reference Architecture Model and Technical Specifications v2*. The Business Layer emphasizes workflows and services that rely on XR-enhanced training, maintenance, and production monitoring, where tools like XAI provide interpretable insights, and GenAI powers dynamic content generation for operator assistance. For instance, GenAI, facilitates the transformation of static user manuals into interactive, queryable formats accessible through the XR platform. This aligns with the XR5.0 objective of delivering value-added services such as immersive training and real-time operator guidance, which are essential for Industry 5.0 scenarios.

The XR5.0 platform facilitates the integration of AI models like XAI, NSAI, and GenAI as key components within the system, enabling capabilities such as operational optimization and decision support. The Container Diagram from D2.3 highlights the orchestration role of the Central XR Hub, which consolidates these AI-driven functionalities while ensuring seamless interaction with AR/VR systems for visualizing real-time analytics and recommendations. The Technology Layer underpins this integration by providing the necessary virtualization, deployment pipelines, and network connectivity. This setup ensures scalability and secure operation of XR applications across various pilots.

The first integration phase of the AI models and XR applications within the XR5.0 pilots was reported in Deliverable D6.3 – *Integrated Pilot Systems and Use Cases Deployment v1*, while a comprehensive evaluation of technical metrics and qualitative results is presented in Deliverable D6.5 – *Socio-technical Evaluation of XR5.0 Systems v1*. The evaluation in D6.5 covers the six XR5.0 pilot applications and is based on a structured, survey-driven assessment framework designed to systematically capture user feedback while allowing for in-depth analysis of pilot-specific innovations. Surveys were adopted as the primary evaluation instrument to ensure consistency, comparability, and scalability across heterogeneous use cases, technologies, and participant profiles. They provide a coherent evaluation framework that supports standardization, enables cross-pilot comparison, and facilitates the collection of both quantitative and qualitative insights, allowing for a detailed assessment of effectiveness, usability, and user acceptance within realistic industrial contexts.

The final integration and Full Prototype versions will be presented in the upcoming deliverable D6.4 – *Integrated Pilot Systems and Use Cases Deployment v2* and their qualitative and quantitative results will be reported in D6.6 – *Socio-technical Evaluation of XR5.0 Systems v2*.

1.3 Relation to Other WP4 Tasks

The integration and workflow of advanced AI methodologies within WP4, highlighting the roles of NSAI, AL and GenAI are illustrated in Figure 2 – Figure 4, emphasizing their contributions to creating trusted human-AI collaboration frameworks, dynamic content generation, and interpretability through XAI. Additionally, it discusses how these technologies connect to downstream tasks for deployment, refinement, and their application in Immersive XR5.0 training platform. Technical evaluation metrics for all the solutions developed for the purposes of WP4, will be defined, standardised and presented in the upcoming and final deliverable D4.4 – *XR Visualisations of AI and XAI Outcomes v2*, which represents the final deliverable of WP4.

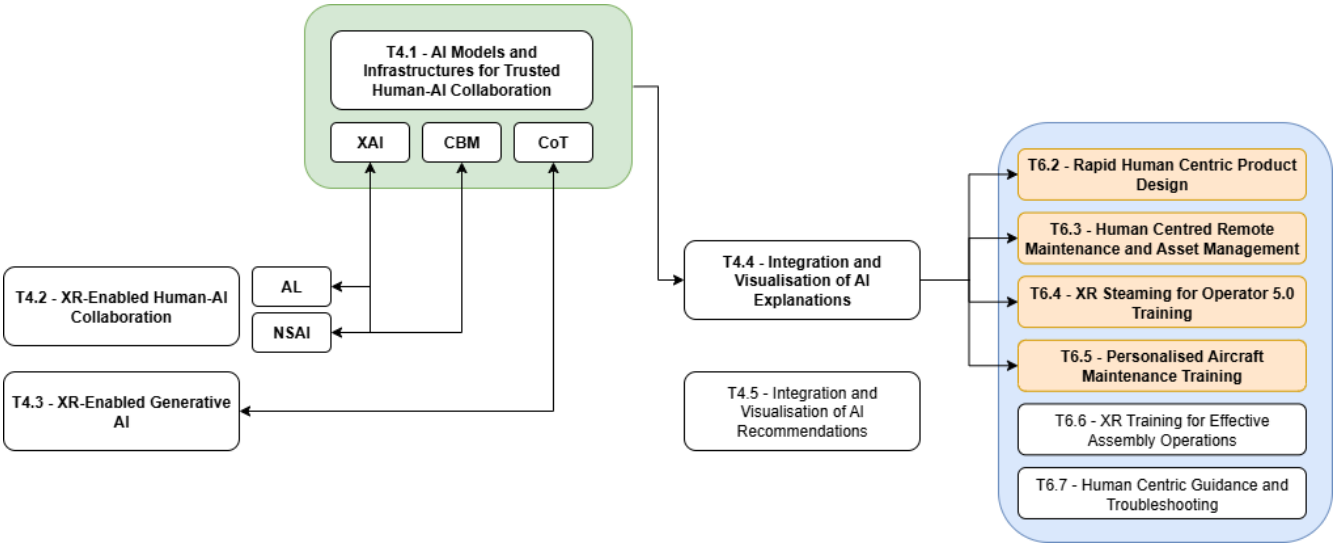

Figure 2 – T4.1, XAI in Relation to Other Tasks

Figure 2 presents the workflow of WP4, emphasizing the development and integration of eXplainable Artificial Intelligence solutions (T4.1) in relation with other WP4 and WP6 tasks. Task T4.1 serves as the foundational component, creating AI models and infrastructures for trusted human-AI collaboration. It incorporates Post-hoc XAI methods, glass-box models (CBM) for the purposes of T4.2 and Chain-of-Thought prompting (T4.3) to illustrate the reasoning of Large Language Models (LLMs) and enhance trust between the AI systems and the human experts. The explanations generated by the models of T4.1 are integrated into Immersive Environments (T4.4) and visualised to end users.

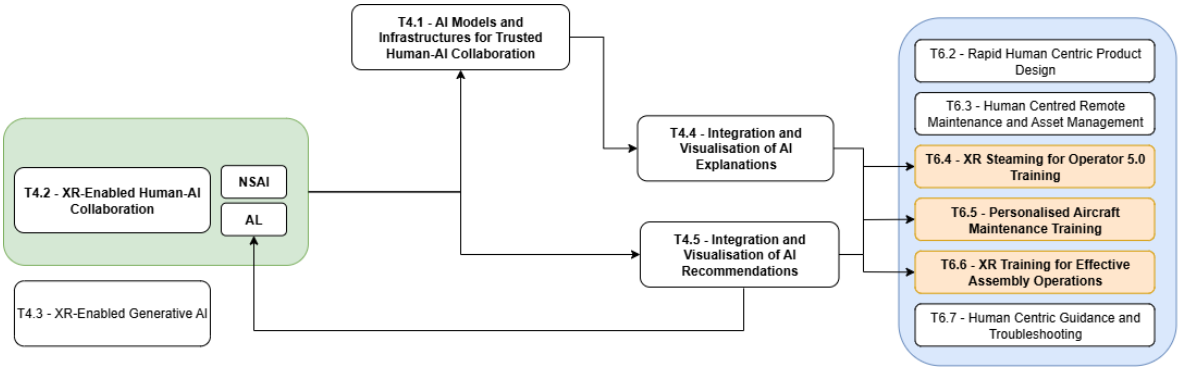

Figure 3 – T4.2, in Relation to Other Tasks

Figure 3 illustrates the workflow of WP4, emphasizing the development and integration of Neurosymbolic AI (NSAI) models (T4.2). It leverages inputs from key modules, including Context-Based Models (CBM), that, as described in section 4.2, are capable of extracting "concepts" leveraging LLM technologies, in order to make each dataset/AI model self-explainable and NSAI-ready. NSAI (T4.2) is trained on the contextually enhanced data offering a self-explainable model permitting also human-AI interactions. The outcomes from these tasks are directed to tasks T4.4 and T4.5, which facilitate the integration and visualization of AI explanations and recommendations, respectively. Furthermore, it illustrates the pipeline showcasing the utilization of Active Learning (AL) models (T4.2) for XR-enabled Human-AI collaboration. In this context, AL is applied to improve the adaptability and efficiency of XR systems by iteratively relabelling data points to refine the learning process. AL operates both independently and integrated with NSAI in T4.2. The insights generated by AL within T4.2 feed directly into T4.1, which provides XAI methods for trusted

human-AI collaboration. From T4.1 and T4.2, the outputs are progressing to T4.4 and T4.5, where the focus shifts to the integration and visualization of AI and XAI outputs. T4.4 ensures that AI explanations are interpretable and accessible to end-users, enabling better understanding and trust in the system, while T4.5 focuses on delivering recommendations tailored to specific contexts. These results are then routed to WP6 for deployment and are further refined and implemented across downstream tasks.

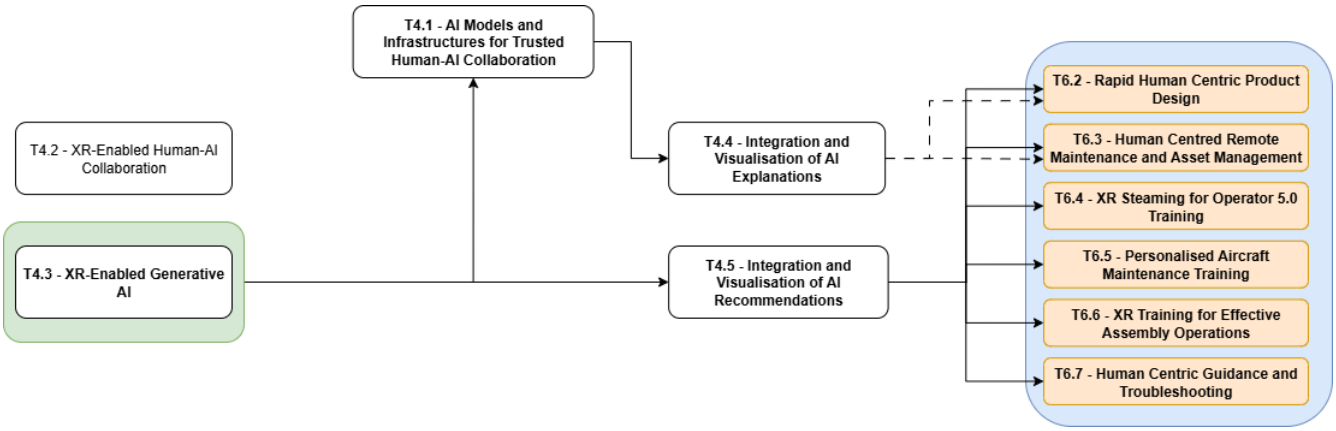

Figure 4 – T4.3, GenAI in Relation to Other Tasks

Figure 4 illustrates the pipeline showcasing the role of Generative AI (GenAI) models (T4.3) and its broader integration into the XR-enabled framework. T4.3 focuses on employing GenAI to dynamically create content, scenarios, and interactions specifically tailored to enhance XR systems. Beyond its standalone applications in T4.3, GenAI also plays a critical role in T4.1, where GenAI will be utilized as part of Human-Centric XAI using Chain-of-Thought (CoT) prompting, enabling the generation of human-friendly explanations and contextual responses. Additionally, GenAI serves as a key interface for the XR5.0 training platform developed in WP5, facilitating immersive, personalized, and adaptive training experiences tailored to users' needs. The outputs from T4.3 and their integration into T4.1 feed directly into T4.4 and T4.5.

1.4 Updates from previous Version

The second version of the Advanced AI Paradigms for Human-AI Collaboration deliverable D4.2 aims to build upon the solid foundations and findings as presented in its first version D4.1. It demonstrates the final stage of the developed Artificial Intelligence models in the fields of eXplainable AI (XAI), Neurosymbolic AI (NSAI) & Active Learning (AL) and Generative AI (GenAI), their implementation, system architecture and how they address specific needs and requirements of Industrial 5.0 applications.

1.4.1 AI Models and Infrastructures for Trusted Human-AI Collaboration

This deliverable presents XAI solutions from Task T4.1 – *AI Models and Infrastructures for Trusted Human-AI Collaboration* in three different categories:

- Post-hoc XAI Models.
- AI Glass-box Models for Computer Vision.
- Human-centric GPT-based Interaction using CoT.

Post-hoc XAI Models

Deliverable D4.1 presented the overall approach and theoretical framework that was designed for the research and development of Post-hoc XAI models. In the last reporting period an XAI suite was developed to provide explanations for black-box models, allowing their integration into existing workflows without the need for retraining. The Post-hoc XAI suite includes Local Interpretable Model-agnostic Explanations

(LIME) and Gradient-weighted Class Activation Mapping (Grad-CAM) models. LIME was incorporated in the XAI suite due to its model agnostic nature and its ability to support multiple data modalities (e.g. images, text). Furthermore, due to the specific needs and requirements (computer vision tasks) as defined by the XR5.0 Pilot owners, Grad-CAM was also preferred since it generates rigorous visual explanations and highlights image regions that contribute the most to an AI model's prediction. Both solutions are provided in an ad-hoc manner in the Immersive Environments through a separate API to avoid cognitive overload. A comprehensive analysis of the Post-hoc XAI suite will be presented in Section 4.1.

AI Glass-box Models for Computer Vision

As presented in Deliverable D4.1, Concept Bottleneck Models (CBM) were preferred as a meaningful way to enhance interpretability by aligning neural network components with human-understandable concepts. In the second reporting period, CBMs were directly integrated with NSAI models to provide further clarity and insight into AI decision making processes. CBMs significantly contributed to the symbolic layer of the NSAI component and are responsible for producing the final predictions based on the detected concepts on the corresponding problem. A full description of the Concept Bottleneck Models and their integration with the NSAI solutions is presented in Section 4.2.

Human-centric GPT-based Interaction using CoT

Human-centric GPT-based Interaction using Chain-of-Thought (CoT) is an additional category of XAI that is included in T4.1 to accommodate for the needs of Pilot 1 and Pilot 2, where for each pilot a custom solution is implemented as part of their GenAI integration from T4.3.A detailed description of Human-centric GPT-based Interaction using Cot in Pilot 1 and Pilot 2 are given in Section 4.3.

A key distinction between Deliverables D4.1 and D4.2 lies in the integration status of the Context Awareness Framework. While D4.1 presented the architecture and intended role of this framework, it has not been integrated into any XR.50 pilot activities reported in D4.2. This is due to the fact that only three pilots within the XR.50 initiative incorporated XAI components and in these cases, the identified XAI requirements did not necessitate integration with the Context Awareness Framework. Instead, the pilots adopted alternative solutions, as listed above, which were deemed sufficient to address their explainability needs within the specific operational contexts.

1.4.2 XR-Enabled Human-AI Collaboration

In this deliverable the Reasoning Context and Contextual Models from Task T4.2 – *XR-Enabled Human-AI Collaboration (Neurosymbolic AI, Active Learning)* are presented. Deliverable D4.1 provided a solid foundation to incorporate NSAI solutions and AL in I5.0 applications. During the last reporting period, NSAI models were developed for the purposes of Computer Vision tasks and stress detection systems. The developed NSAI models combine sub-symbolic neural perception with structured, human-interpretable symbolic reasoning to enhance human-AI collaboration. Furthermore, the integration of NSAI models with AL methods, such as human-in-the-loop, assisted in alleviating the problem of unlabeled datasets that appear highly in industrial applications. Combining human expertise in cases of algorithmic uncertainty proved to increase algorithmic accuracy as well as human trust towards the AI decision making processes. In order to provide the model's predictions in various pilots' applications, as well as integrating the AL component which would allow the pilots experts to provide their feedback to the model's prediction, an API service was developed. The API along with the necessary authorization component is responsible for providing predictions for the corresponding pilots needs as well as gathering the expert's feedback which is valuable and used for model retraining. A detailed description of the Reasoning Context and Contextual Models is presented in Section 2.1.

1.4.3 XR-Enabled Generative-AI

This deliverable presents three distinct solutions from Task T4.3 – XR-Enabled Generative-AI.

- XR-enabled GenAI – LLM Chat Engine with Customizable Agents,
- GenAI for XR-based Maintenance Instructions,
- DataLens Conversational AI Framework.

XR-enabled GenAI – LLM Chat Engine with Customizable Agents is the updated version of the XR-enabled GenAI solution from D4.1. Deliverable D4.1 reported the foundational implementation of the LLM Chat Engine, built around a RAG pipeline and a RESTful API serving as the integration point for XR devices and applications. The current reporting period has seen the system grow substantially in capability and maturity. A web-based dashboard now complements the API, giving operators and administrators a dedicated interface for document management, pilot configuration, and direct interaction with the chat engine. Document processing has moved beyond simple text extraction, with the pipeline now capable of preserving document structure, handling tables, and generating automated descriptions of images found within technical documents. The query processing pipeline has evolved from a single-agent design into a three-stage architecture where dedicated agents handle retrieval, image selection, and answer generation in sequence. A user and pilot management system has also been introduced, supporting role-based access control, custom pilot creation, and independent configuration of each pilot's knowledge base and behaviour. These advances are described in detail in Section 3.1.

In addition to this, two new solutions are developed within Task 4.3.

GenAI for XR-based Maintenance Instructions is based on advanced Generative AI-based systems that enable conversational access to industrial documentation and maintenance knowledge through natural language queries, voice interfaces, and XR-supported workflows. Integrated within Pilot 2, It supports maintenance troubleshooting and operational guidance, enabling technicians to retrieve relevant knowledge from maintenance documentation and historical service records. The solution facilitates XR-assisted workflows and interaction through intuitive interfaces, supporting operators during maintenance tasks. In addition, Chain-of-Thought reasoning from T4.1 enables large language models to externalize their reasoning processes, allowing users to trace how AI-generated responses are derived and supporting transparency in AI-assisted decision-making. A detailed description of GenAI for XR-based Maintenance Instruction is given in Section 3.2.

DataLens Conversational AI Framework provides a privacy-preserving, on-premises solution for conversational access to industrial knowledge using retrieval-augmented generation (RAG) pipelines. Integrated within Pilot 1, it enables operators to query technical documentation and operational data through natural language interactions, delivering grounded responses. The integration of Chain-of-Thought reasoning from T4.1 improves transparency by allowing users to follow the reasoning behind generated outputs. The framework also supports multimodal interaction, including voice and XR interfaces, enabling flexible deployment in industrial environments. A detailed description of DataLens Conversational AI Framework is given in Section 3.3.

1.5 Structure of the Deliverable

This deliverable presents the updates of the tasks T4.1, T4.2 and T4.3 since it is the second and final version of the Advanced AI Paradigms for Human-AI Collaboration. The deliverable content is structured as follows:

- Section 2, following this introductory section, presents the progress and updates from Task T4.2, XR-Enabled Human-AI Collaboration (Neurosymbolic AI, Active Learning).
- Section 3 presents the progress and updates on Task T4.3, XR-Enabled Generative AI.
- Section 4 presents the progress and updates on Task T4.1, AI Models and Infrastructures for Trusted Human-AI Collaboration.
- Section 5 concludes the deliverable.

The technologies in each task are presented across several key dimensions addressing:

- *Role and functionality*, defining the specific contributions of the respective technology,
- *System architecture*, detailing the key components associated with the respective technology,

- *Implementation,* detailing the implementation of the system,
- *Demonstration scenarios and mapping to pilots*, connecting the technology application to specific project pilots,
- *Challenges and limitations* encountered or anticipated during the technology application in the project pilots,
- *Evaluation*, summarizing the validation of the technology,
- *Discussion and lessons learned*.

2. XR-ENABLED HUMAN-AI COLLABORATION

2.1 Reasoning Context and Contextual Models

Visual inspection of infrastructure assets via CCTV presents a fundamental tension in applied AI: deep neural networks achieve strong perceptual performance but produce opaque, unauditable decisions, while symbolic reasoning systems are transparent but struggle with raw perceptual inputs such as video. This tension is especially acute in industrial scenarios such as infrastructure inspection and maintenance where image and object classifications carry operational and regulatory consequences that demand both accuracy and explainability. The approach adopted is a Neurosymbolic AI — a paradigm that integrates sub-symbolic neural perception with structured, human-interpretable symbolic reasoning. Specifically, the system is built around Concept Bottleneck Models (CBMs), a Neurosymbolic architecture that enforces an explicit symbolic bottleneck between perception and decision-making. Rather than mapping raw visual inputs directly to output labels, a CBM first grounds perception in a set of human-defined semantic concepts, then applies a transparent linear reasoning step over those concepts to reach a classification. This design satisfies the dual requirement of perceptual power and symbolic accountability that XR5.0 project's pilots' demand. To enhance the human - AI collaboration, an Active Learning system was also employed to empower human experts and provide domain knowledge to the AI models. Leveraging human-in-the-loop techniques the human actor is placed in the center of the AI decision making process, since he participates in the AI lifecycle by providing input and oversight to improve algorithmic accuracy and safety.

2.1.1 Role and Functionality

The CBM serves as the central reasoning component of the inspection pipeline. Its Neurosymbolic structure is operationally significant: the symbolic bottleneck layer ensures that every classification is fully traceable to a weighted combination of interpretable concepts (e.g., surface texture irregularity, discolouration, structural deformation), rather than latent neural activations with no semantic grounding. In the CCTV inspection context, the model receives individual video frames as input and produces:

- A classification label from the set {Normal, Crack, Corrosion, Leakage},
- A concept activation vector quantifying each concept's contribution to the prediction — the symbolic reasoning trace,
- A confidence score for the predicted class.

This dual output serves two distinct operational roles. The neural component (the CLIP backbone) handles the perceptual complexity of raw video frames; the symbolic component (the concept bottleneck and sparse linear classifier) produces decisions that domain experts can inspect, contest, and correct at the concept level. The model also supports test-time concept intervention: an expert can override a specific concept activation and observe the resulting change in classification, enabling counterfactual reasoning of the form *"if no discoloration were present, would this frame still be classified as corrosion?"* This is a defining capability of Neurosymbolic systems — the symbolic layer permits direct human manipulation of intermediate reasoning states in a way that pure neural models do not support. This is particularly valuable in regulatory and audit contexts where decision accountability is required. Expert corrections feed into a continuous active learning loop: labelled feedback is accumulated and used to periodically fine-tune the symbolic classification layer, allowing the model's reasoning to adapt to domain-specific knowledge without retraining the neural perception backbone.

2.1.2 System Architecture

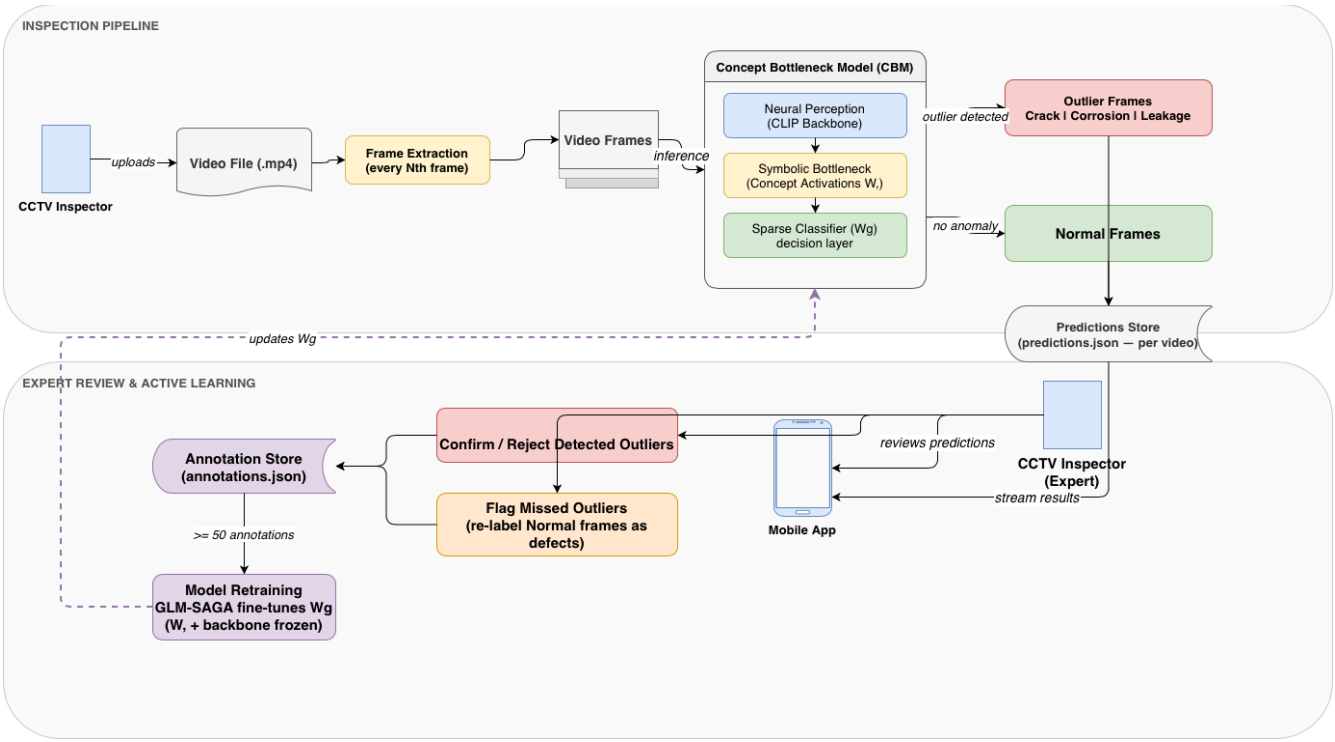

Figure 5 – High-level Architecture of Neurosymbolic AI Component

The architecture in Figure 5 reflects the Neurosymbolic decomposition directly, separating the neural perception stack from the symbolic reasoning and knowledge update components.

Data Ingestion Layer

Videos are submitted via a RESTful API. Upon receipt, frames are extracted at a configurable interval (default: every 50th frame) using OpenCV and persisted to a shared storage volume organised by video identifier.

Neural Perception Layer

A pre-trained **CLIP (Contrastive Language-Image Pretraining)** vision encoder serves as the neural backbone. CLIP was selected for its strong alignment between visual representations and semantic textual descriptions — a property that directly supports concept-grounded supervision. The backbone operates in frozen inference mode, producing high-dimensional visual embeddings that are passed to the symbolic bottleneck.

Symbolic Bottleneck Layer

This is the Neurosymbolic core. The projection layer maps the neural embedding into a concept activation space defined by human-interpretable concepts.

Active Learning & Knowledge Update Layer

Expert corrections are accumulated alongside frame images, organised by corrected labels. When a configurable threshold (default: 50 annotations) is reached, the symbolic layer is retrained while the neural backbone and concept projection remain frozen. This *partial retraining* strategy is appropriate for Neurosymbolic active learning: the neural perception and concept grounding remain stable, and only the

symbolic decision boundary is updated to incorporate expert knowledge. Weights are backed up before each update. All capabilities are exposed through a FastAPI backend secured with OAuth2/JWT authentication and role-based permission scopes.

2.1.3 Implementation

The system is containerised via Docker Compose, with Kubernetes manifests provided for production environments. Separate Persistent Volume Claims maintain shared data and trained model weights across pod restarts. HTTPS is supported via configurable SSL certificates.
An admin user interface has been developed in order for the CCTV inspector to be able to upload the videos he wants. These videos are first framed and analyzed in the web service, and after a while are available to be presented in the mobile app.

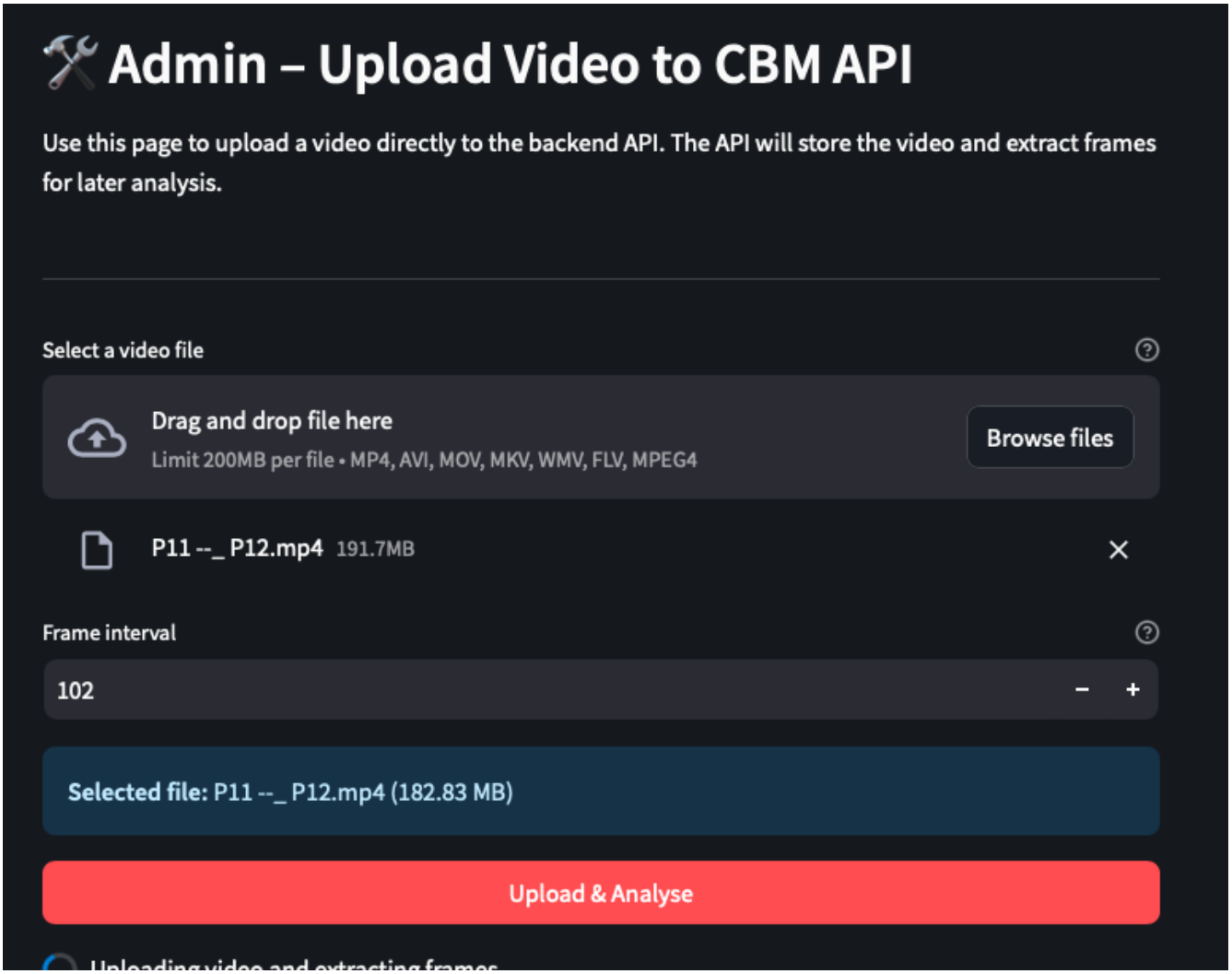

Figure 6 – Neurosymbolic AI Administrator's Page

Then, an API has been developed and deployed in the following address: https://cmb-api-xr50.euprojects.net/

The API includes an AUTH login system, which is responsible for the issuance of the JWT tokens which are required in the calls towards the API.

Table 2 shows the calls that can be made to the API, and some example responses as well.

Table 2 – API Specification and Public Endpoints

Method	Endpoint	Auth	Description	Input
POST	*/upload_video*	Admin Token	Upload a video file and extract frames at specified intervals	**file** (multipart/form-data): Video file (.mp4, .avi, .mov, .mkv) **frame_interval** (form field, optional): Extract every Nth frame (default: 50)
POST	*/predict_frames/{video_name}*	User Token	Run CBM model predictions on extracted frames for a specific video	**video_name** (path parameter): Name of the video from upload response
GET	*/predictions/{video_name}*	User Token	Retrieve existing predictions for a specific video with optional filtering	**video_name** (path parameter): Name of the video **include_normal** (query, optional): Include normal predictions (default: true) **class_filter** (query, optional): Filter by specific class ('normal', 'crack', 'corrosion', 'leakage') **confidence_threshold** (query, optional): Minimum confidence threshold (0.0-1.0)
POST	*/feedback/expert*	Admin Token	Submit expert feedback for active learning (model improvement)	**frame_image** (multipart/form-data): The frame image file **frame_id** (form field): Unique frame identifier from predictions **video_name** (form field): Name of the source video **model_prediction** (form field): Model's original prediction **expert_classification** (form field): Expert's classification ('normal', 'crack', 'corrosion', 'leakage')

				confidence_score (form field, optional): Expert's confidence (0.0-1.0) **expert_notes** (form field, optional): Additional notes

2.1.4 Demonstration Scenarios and Mapping to Pilots

The Reasoning Context and Contextual Models asset has been applied in Pilot 3 and Pilot 4 for the purposes of image classification and object detection in offline and streaming scenarios.

Pilot 3 (EKSO) – XR Steaming for Operator 5.0 Training

For the purposes of Pilot 3, we developed a scenario where the CCTV inspector would be benefited by the use of the service. The computer vision task is to identify whether there are outliers and they can be identified, as well as the type of outlier the model is able to identify in a certain video. For each different outlier type, the model is able to identify specific concepts that are present on videos that are relevant to each outlier type and responsible for the corresponding prediction. The inspector is able to view the outliers in an offline manner and also get the reasoning of the model for each prediction. In order to utilize the solution, he adheres to the following steps:

1. First, he uploads the video using the admin environment. Since all services follow the authentication protocols of the whole project, first of all the user has to login to the corresponding service using the authentication parameters he has been provided.
2. When the video finishes uploading and inference is complete, the user is able to view the video and the frames labeled as outliers via the mobile app.
3. (still under development) CCTV inspector (who is considered the "Expert") is able to provide the feedback to the model's predictions.

Pilot 4 (TAP) – Personalised Aircraft Maintenance Training

In the context of Pilot 4, the computer vision task is to identify specific engine aircraft parts. The component is able to serve classifications in an offline and streaming setting, depending on the user requirements. Due to the nature of the specific use case and the lack of available data set, human-in-the-loop techniques were employed from the training set generation stage. Figure 7 presents a high-level architecture of the Reasoning Context and Contextual AI model component for the purposes of Pilot 4.

1. Expert operators use the Immersive Environment (XR app) to capture and tag specific aircraft engine parts.
2. Human inputs are stored in cloud storage. Data is processed and cleaned to create an initial training set to be used for the AI training.
3. Once full training is complete, the human operator is able to use the AI object detection module directly from the XR app.
4. The AI produces the engine detection part (object detection) output by also incorporating its decision-making confidence level. The outputs are given in the forms of images and/or videos in the offline setting and as a constant stream in the online setting.
5. Human experts can also provide corrective input into the AI model which triggers a retraining of the model should the Information Gain is modified.

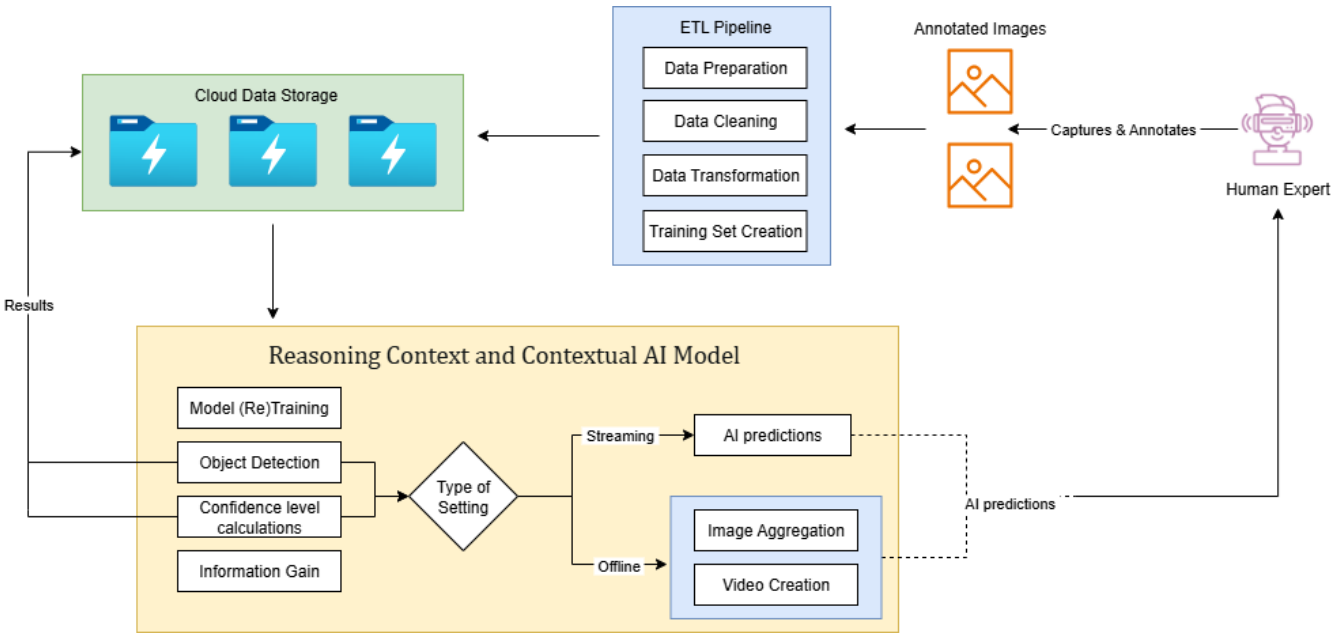

Figure 7 – High-level Component Architecture for Pilot 4

Pilot 5 (SPACE) – XR Training for Effective Assembly Operations

The Reasoning Context and Contextual Models component integrates with Pilot 5 for the purpose of identifying stress detection levels of operators/workers during training sessions or assembly tasks. The main motivation in determining proper stress levels while performing particular tasks, is to identify learning gaps and propose personalised training material that matches the workers/operators needs. To that end, wearables (watches) using Wear OS are provided to individual workers while performing a specific task or training session. The worker activates the wearable App as described in D3.3 – *Personalised XR Content Development and Adaptation v1* and the App transmits the data directly to the XR5.0 Clawdite Platform which is also described in D3.3.

As illustrated in Figure 8, the Reasoning Context and Contextual AI component retrieves all relevant anonymous data to proceed with the stress level detection of a particular task for a specific time range. The data are initially passed through an Extract Transform Load (ETL) pipeline to prepare the training or test sets that will feed the AI model. Considering that in real-world industrial situations, some tasks might slightly raise the stress levels of individuals by nature and not by lack of knowledge or expertise, we employ human-in-the-loop techniques to incorporate that knowledge by defining expected stress levels for a specific task by a human expert. The confidence level calculations as well as the Information Gain identified in the training/test sets are reported to the human expert in case there should be adjustments that would increase algorithmic accuracy and precision. Finally, the stress level detection module provides the relevant results in forms of reports that can be integrated directly in Immersive Environments.

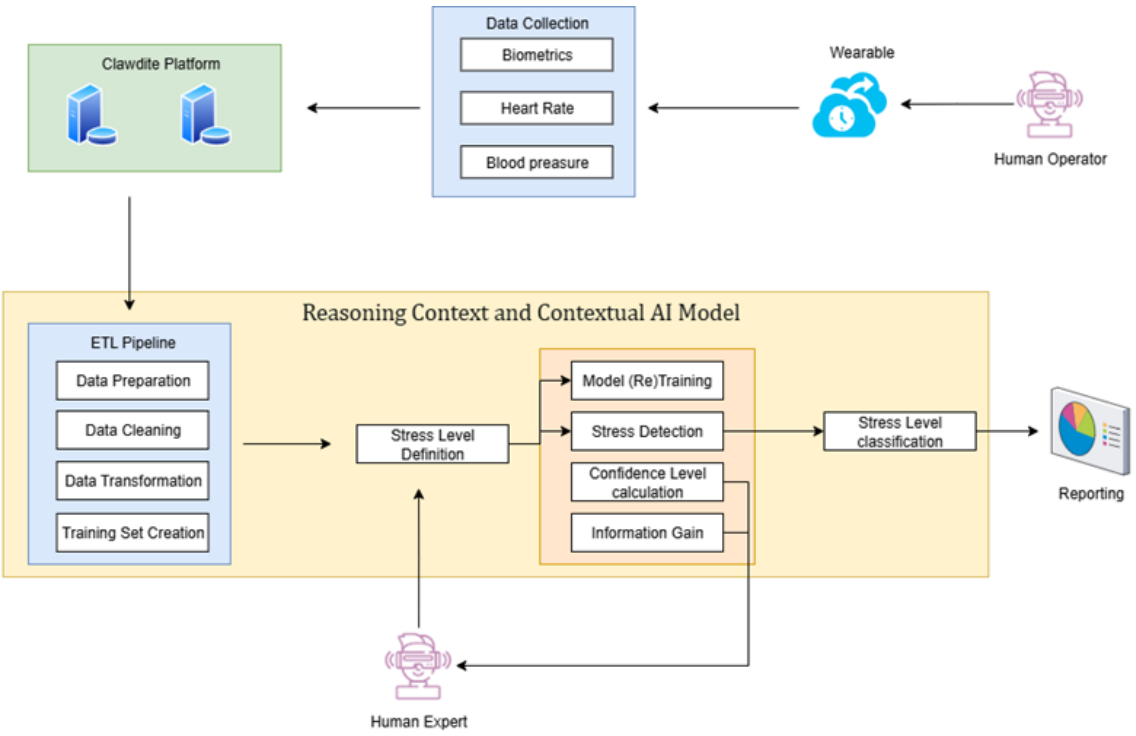

Figure 8 – High-level Component Architecture for Pilot 5

2.1.5 Challenges and Limitations

One major challenge has been the lack of big datasets. AI models are trained with small instances of outliers and leverage the Active Learning component and the expert's knowledge to be able to provide better predictions concerning the pipe outliers.

Another limitation of the current implementation is that currently is applied in an offline environment, since the latency and the bandwidth required for a video to be uploaded on the API and the predictions to be fetched real time is quite challenging to be achieved.

2.1.6 Evaluation

For the evaluation of the component, a specialized User Interface environment was developed. The user had the following capabilities via the user interface that connected directly to the API of the solution:

- Upload a CCTV inspection video.
- Identify outlier frames and display them along with the model's confidence score for each detected outlier.
- Display normal frames (frames not labeled as outliers) for user review.
- Show model explanations alongside each prediction to help users understand the reasoning behind the results.

To conduct an initial evaluation of the NSAI component and its performance during this preliminary user evaluation phase, we used one report provided by the pilot owner, EKSO. The report was linked to two different CCTV inspection videos and included the operator's inputs collected in a laboratory setting.

To compare and validate the NSAI outputs against real operator feedback, we analysed and processed the CCTV inspection videos and cross-matched the NSAI results with the outliers and anomalies tagged by the operator.

The NSAI component successfully identified all outliers that had also been tagged by the human experts, with an excess of +5 and +4 detections respectively due to the frame-by-frame analysis (Table 3). In

addition, the system flagged extra frames for expert evaluation, which could further support the assessment of KPI #5 – Reduced Human Error. In KPI #5, the goal of the Human - AI collaboration is to reduce human errors. The human operator as well as the model would be flagging anomalies in inspection videos and in the end one would be compared to the other. By flagging more outliers than what the expert identified, we may encounter False Positive predictions but in case something an outlier is indeed missing from the human expert's identification, it would be really valuable for the Pilot. A detailed report of KPI evaluation will be presented in the upcoming Deliverable D6.6 – *Socio-technical Evaluation of XR5.0 Systems v2* and in the final report of the project.

The model also achieved a relatively low false-positive rate as illustrated in Table 4, with only nine false-positive predictions. However, an area that requires improvement is the real-time processing ratio, which needs to be enhanced in order to transition from the current laboratory environment to a fully operational XR environment.

Table 3 – Evaluation Metrics for Outlier Detection using NSAI

#	Size	Frame Interval	Frames	Processing Time	Real outliers	AI outliers	Total Predictions	Confidence
1	739.311MB	30	728	88.2 secs	25	30	720	68.5%
2	639.051MB	30	624	62.4 secs	25	29	620	70.9%

Table 4 – NSAI Evaluation Results

Video	Precision (%)	F1 Score (%)	Processing Speed (fps)	Real-time Ratio	Inspection Reduction (%)
1	83.3	98.4	8.16	0.28	95.1
2	86.2	98.9	9.94	0.33	94.7
TOTAL/AVG	84.7	98.6	9.05	0.3	94.9

2.1.7 Discussion and Lessons Learned

The development and initial evaluation of the CBM-based CCTV inspection component within the XR5.0 project yielded several important lessons that will inform future iterations of the system. First, the Neurosymbolic approach proved effective in bridging the gap between automated detection and operator trust as reported in D6.5 – *Socio-technical Evaluation of XR5.0 Systems v1*: the concept-level explanations were qualitatively coherent and well-received during preliminary evaluation, reinforcing the value of interpretability as a first-class design requirement rather than an afterthought in industrial AI deployments. Second, the scarcity of labelled inspection data emerged as the most consequential practical constraint, confirming that active learning is not merely a desirable enhancement but a necessity for this domain — small defect datasets alone are insufficient to train a robust classifier, and the human expert must be structurally embedded in the model improvement lifecycle from the outset. Third, the evaluation results demonstrated that the system achieves high recall, successfully identifying all outliers tagged by human experts, but that precision (84.7%) and real-time processing capacity (real-time ratio ≈ 0.30) remain areas requiring targeted improvement before the solution can transition from a laboratory-grade tool to a fully operational field deployment within an XR streaming environment which will be reported in upcoming deliverables D4.4 and D6.6. Finally, it is important to note that human-in-the-loop systems must be treated

as necessary and complementary to the AI component, meaning that the tools that enable expert participation are equally critical to the system's long-term value and adaptability.

3. XR-ENABLED GENERATIVE AI

3.1 XR-enabled GenAI – LLM Chat Engine with Customizable Agents

This section presents the final version of the Generative AI integration within XR environments, building upon the foundation established in D4.1. The system, developed by project partner INNOV, leverages an LLM Chat Engine, centred around Retrieval-Augmented Generation (RAG), to provide as a service intelligent and domain-aware assistance to industrial users.

3.1.1 Role and Functionality

The system consists of two core components, the LLM Chat Engine and a Vector Database for integrating use-case specific knowledge. Users interact with the system's API sending natural language queries to the LLM Chat Engine. The engine retrieves relevant content from ingested technical documentation via RAG and returns grounded, real-time guidance back into the XR environment. Alongside these conversational capabilities, the system handles the uploading, processing, and storing of technical documents, such as user manuals, technical datasheets, maintenance reports etc., which form the knowledge base from which the chat engine draws its responses.

Expanded Access and Management Interfaces

The system is accessible through two complementary interfaces. The first is the RESTful API, which serves as the primary integration point for XR applications and devices. It exposes endpoints for chat interactions, document upload and management, system prompt and custom pilot configuration, and user and session handling. Figure 9 shows an overview of the available API endpoints. Admin-level endpoints are not shown for security reasons.

The second interface is a web-based dashboard. This provides operators and administrators with a direct and accessible means of managing the system without requiring API integration. Through the dashboard, users can upload and organise technical documentation, configure the behaviour of their AI assistant, manage user access and permissions, and interact directly with the chat engine. Figure 10 shows the dashboard landing page.

The dashboard includes a conversational chat interface that mirrors the query experience available through the API integration (Figure 11). This allows engineers and operators to test and refine the system's responses against ingested documentation in a straightforward way.

The management of the knowledge base is also streamlined through the UI, as users can directly view all ingested documents, the pilot they belong to, and associated metadata such as document type and the summary of available documents (Figure 12).

Together, the two interfaces (API and Web) facilitate both plug-n-play integration with external systems (e.g., XR devices) and simple management of the LLM agents.

Updated User and Pilot Management

The system includes a user management layer, supporting registration, login, and role-based access control. Each user is associated with one or more pilots, with their permissions determining which pilots they can access and interact with. In addition, users with the appropriate permissions can create custom pilots tailored to more specific use cases they wish to configure or test. Administrator users are responsible for managing accounts and granting appropriate access levels to other users. All user and pilot management operations are available through both the dashboard and the API. Users with appropriate permissions can configure the agent instructions (i.e., system prompt) of their pilot, allowing the behaviour of the chat engine to be adapted to their requirements (Figure 13).

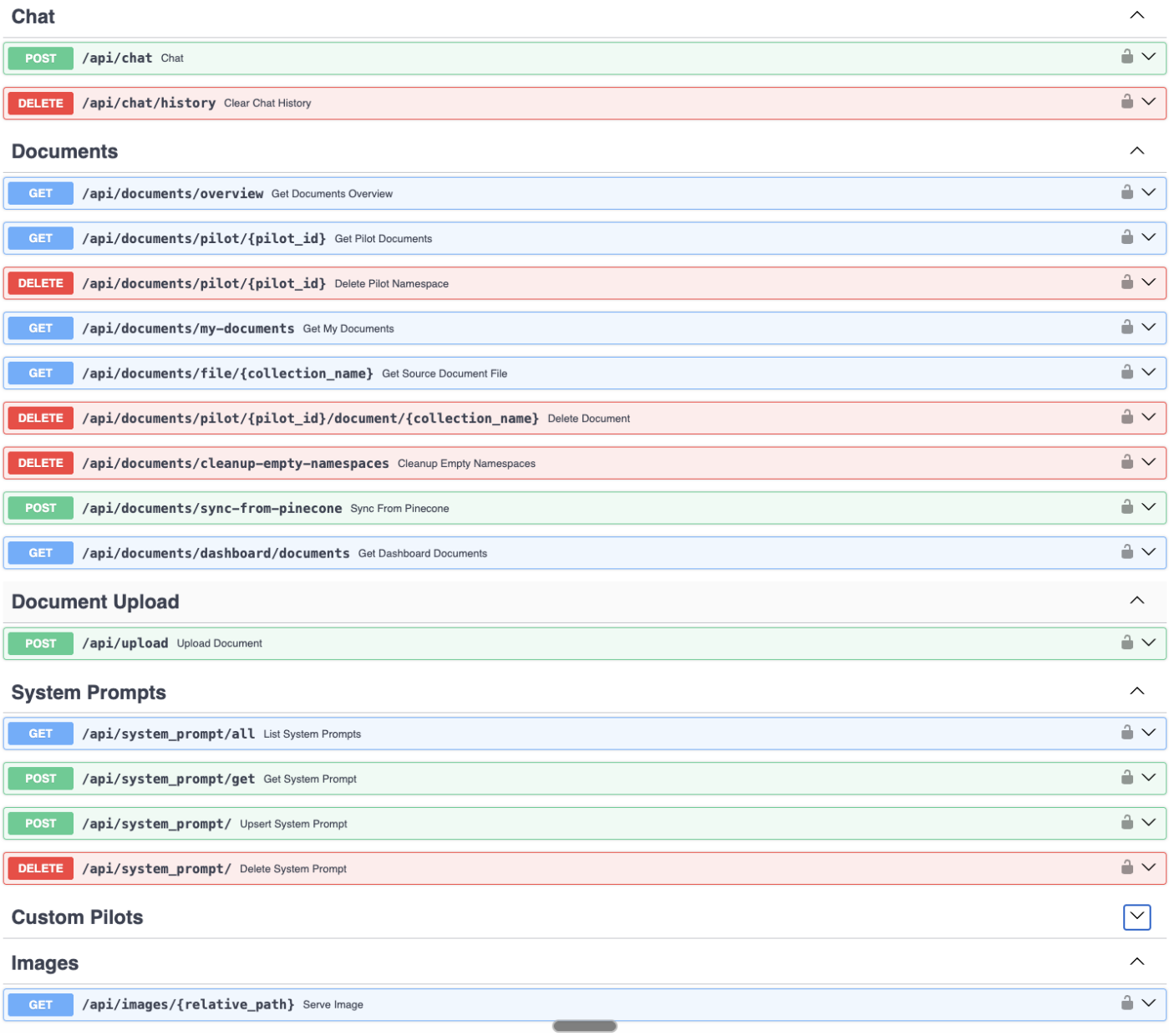

Figure 9 – Overview of Available API Points

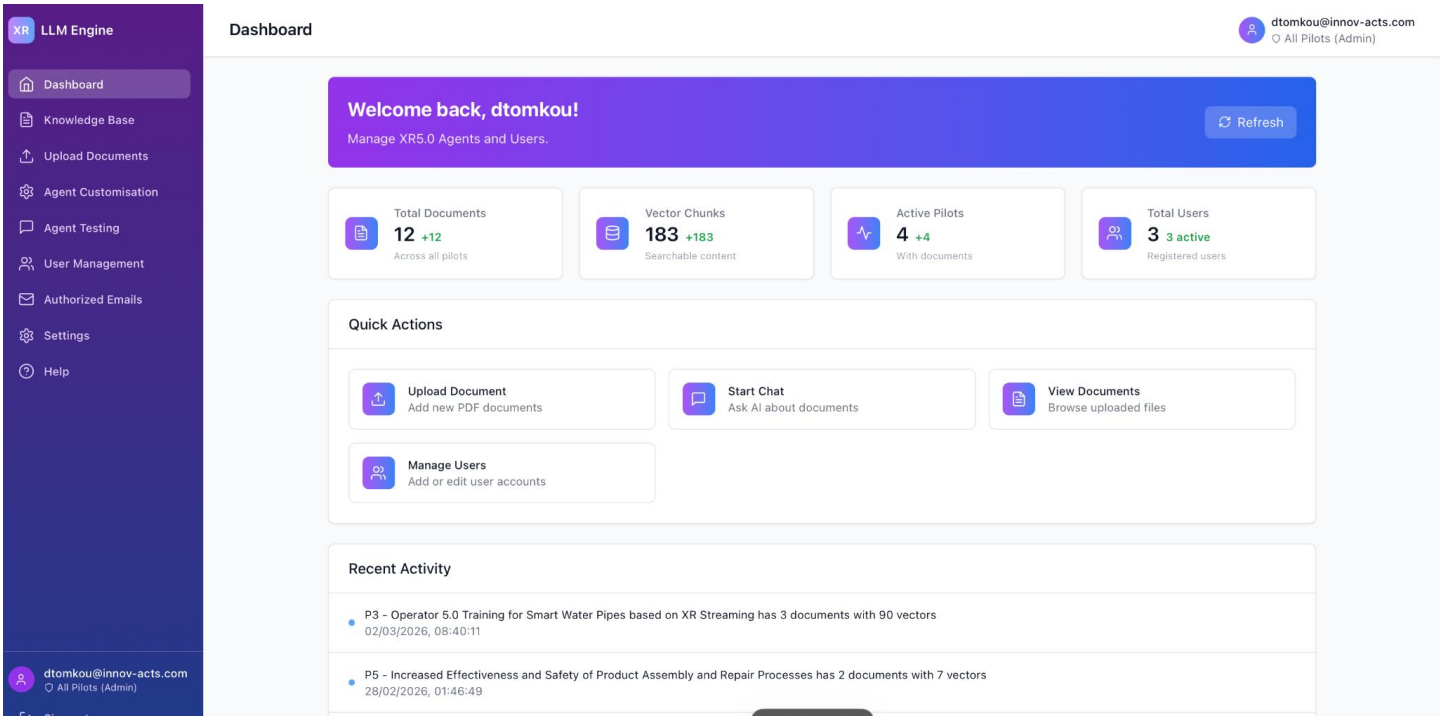

Figure 10 – Landing Page of UI Dashboard

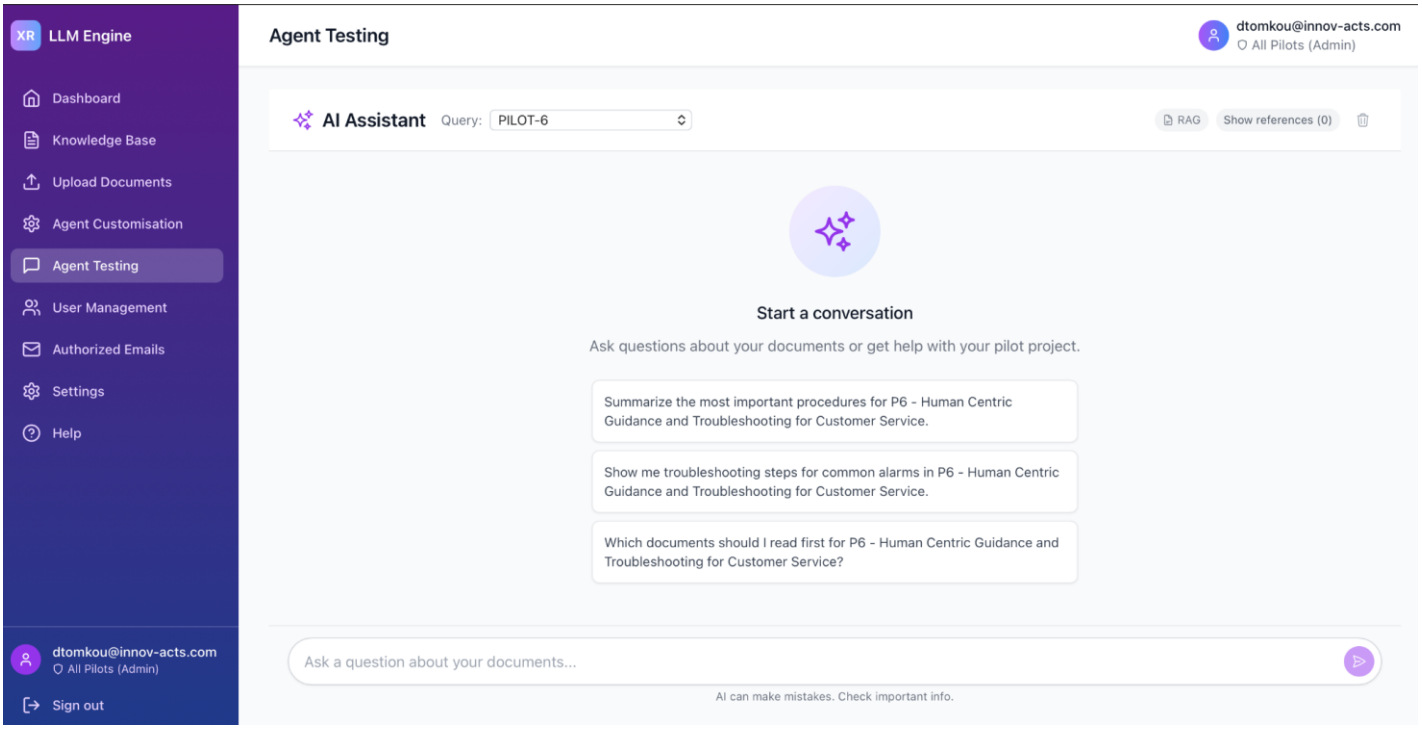

Figure 11 – Chat Interface of UI Dashboard

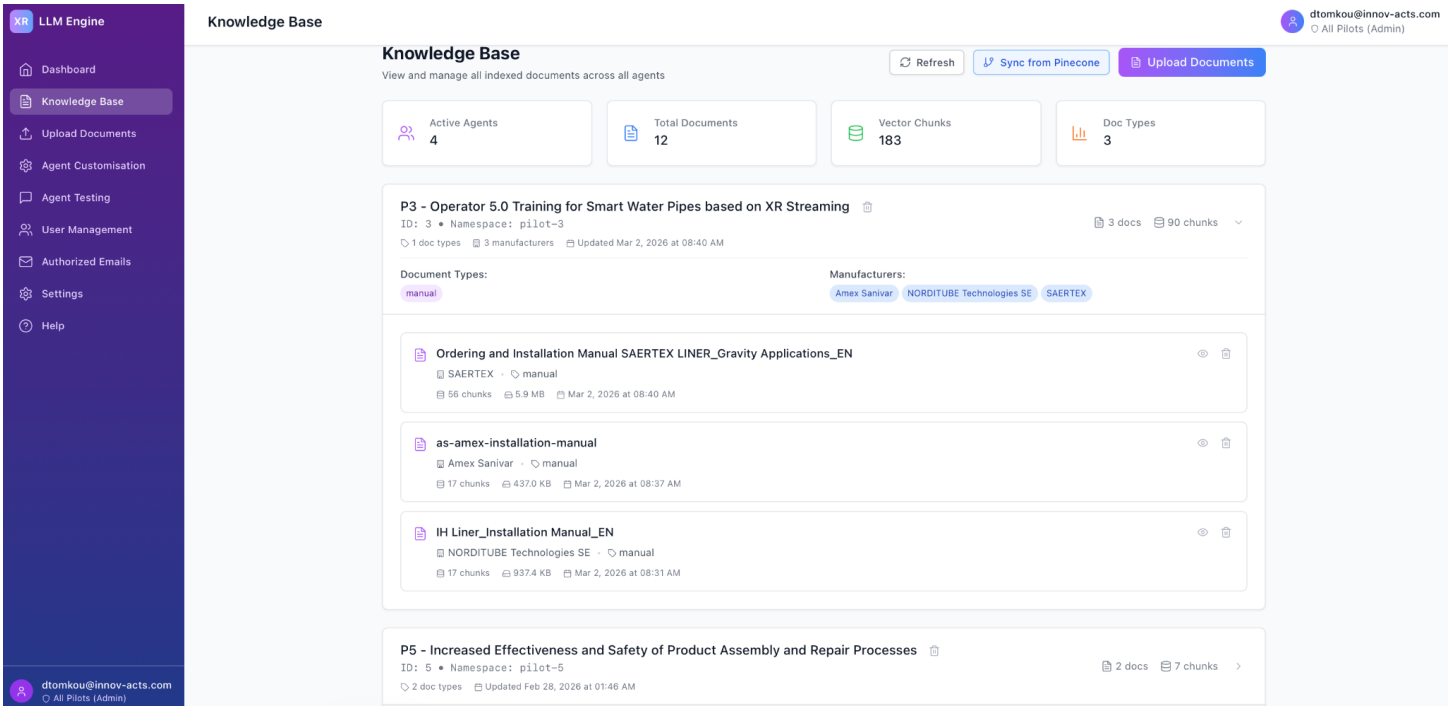

Figure 12 – Knowledge Base Overview Page

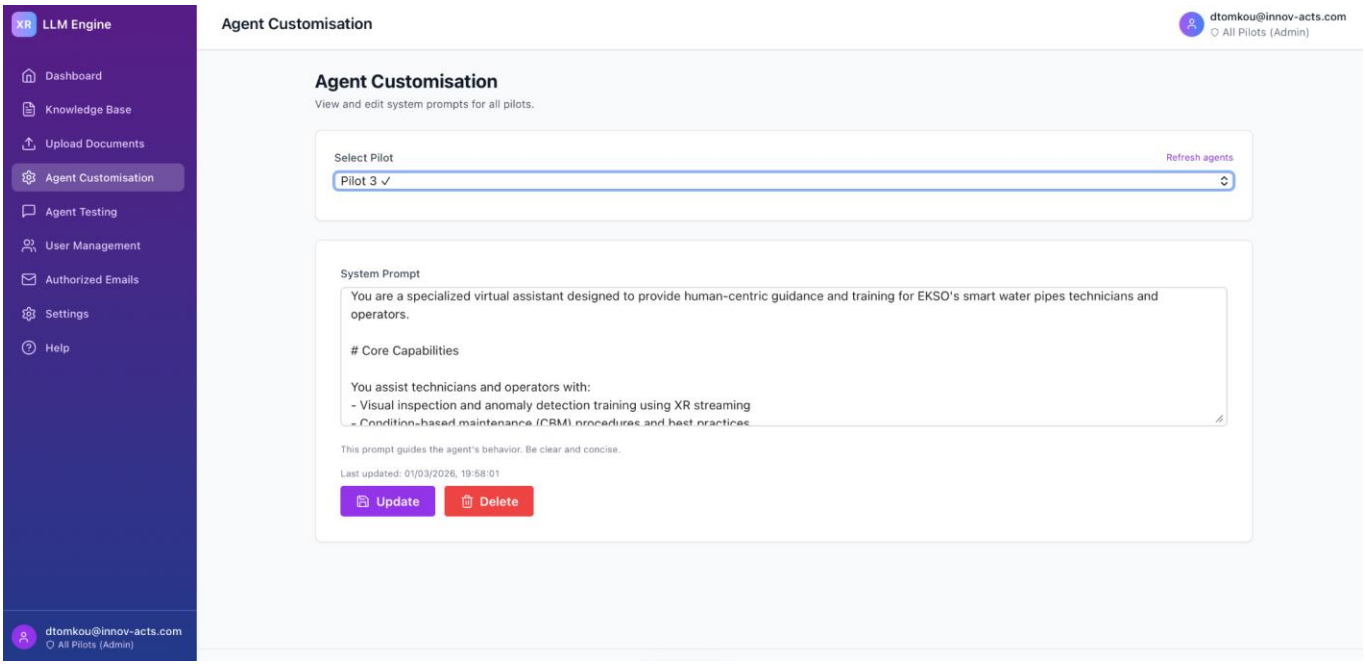

Figure 13 – System Prompt Management Page

Multimodal Document Understanding and Ingestion

The document processing capabilities introduced in D4.1, covering the upload and storage of technical PDFs, have been significantly extended. The pipeline now preserves the full logical structure of a document including its sections and headers. Moreover, it extracts and interprets tables and analyses images to produce descriptions of their visual content. The descriptions of extracted images are made available to the chat engine to reference and explain relevant figures naturally during interactions with the user. This preprocessing pipeline improves the accuracy and relevance of content retrieved during chat interactions.

Additionally, a summary of each document is produced and stored alongside its metadata. This summary is displayed in the dashboard to give users a quick overview of each document's content and is also used at query time to help the retrieval agent identify the most relevant documents for a given question.

Extended Conversational Capabilities

A major functional advancement is that the chat engine now delivers responses that go beyond text, incorporating relevant technical images, tables, and direct access to source PDFs alongside each generated answer. For industrial training and maintenance scenarios, where procedures are frequently accompanied by technical diagrams and explanatory images, this represents a meaningful improvement over the previous version. By showing the source PDFs, users are able to trace every piece of guidance back to the original technical documentation and inspect it in full (Figure 14). This transforms the interaction from a purely textual exchange into a multimodal and grounded response where users can verify the information they receive. The quality and relevance of responses is further supported by the multi-stage query processing pipeline, which is described in detail in the system architecture section.

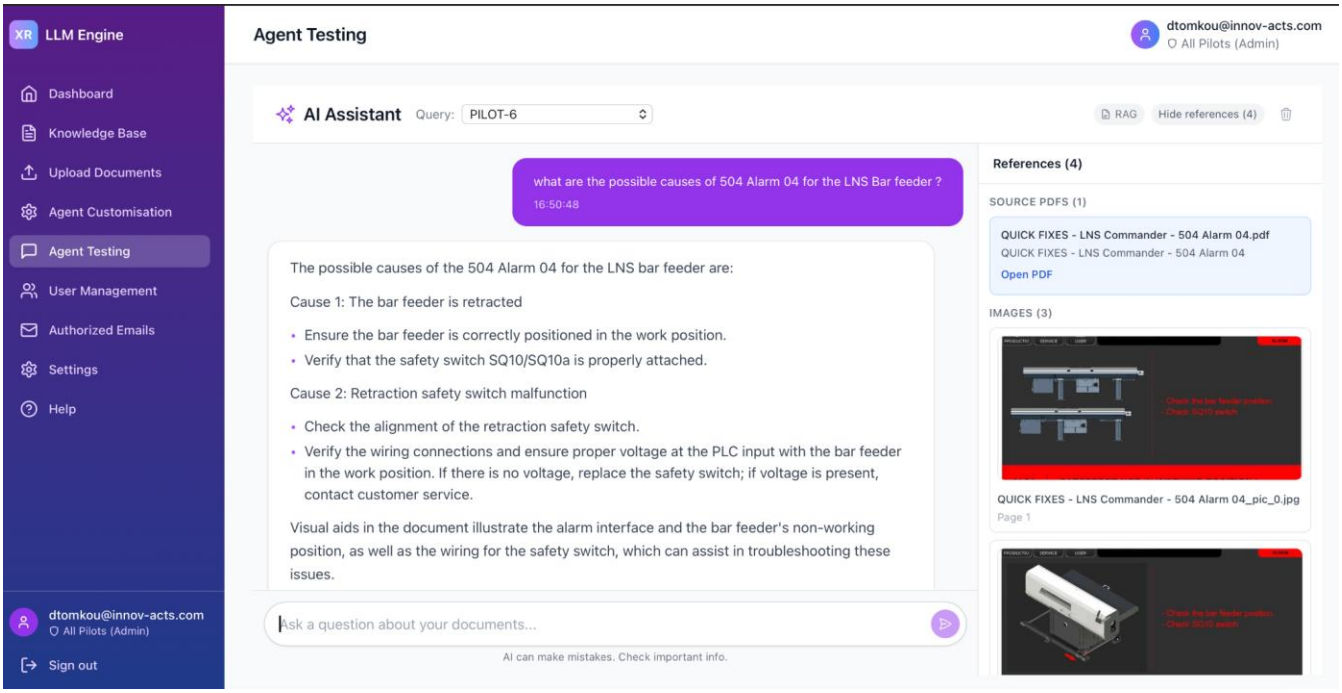

Figure 14 – Example of Conversational Capabilities

3.1.2 System Architecture

While the foundational components, the LLM Engine, Vector Database, and RESTful API, remain central, the architecture has been restructured to introduce stricter separation of concerns, a more robust and scalable solution. Figure 15provides an updated high-level overview of the architecture.

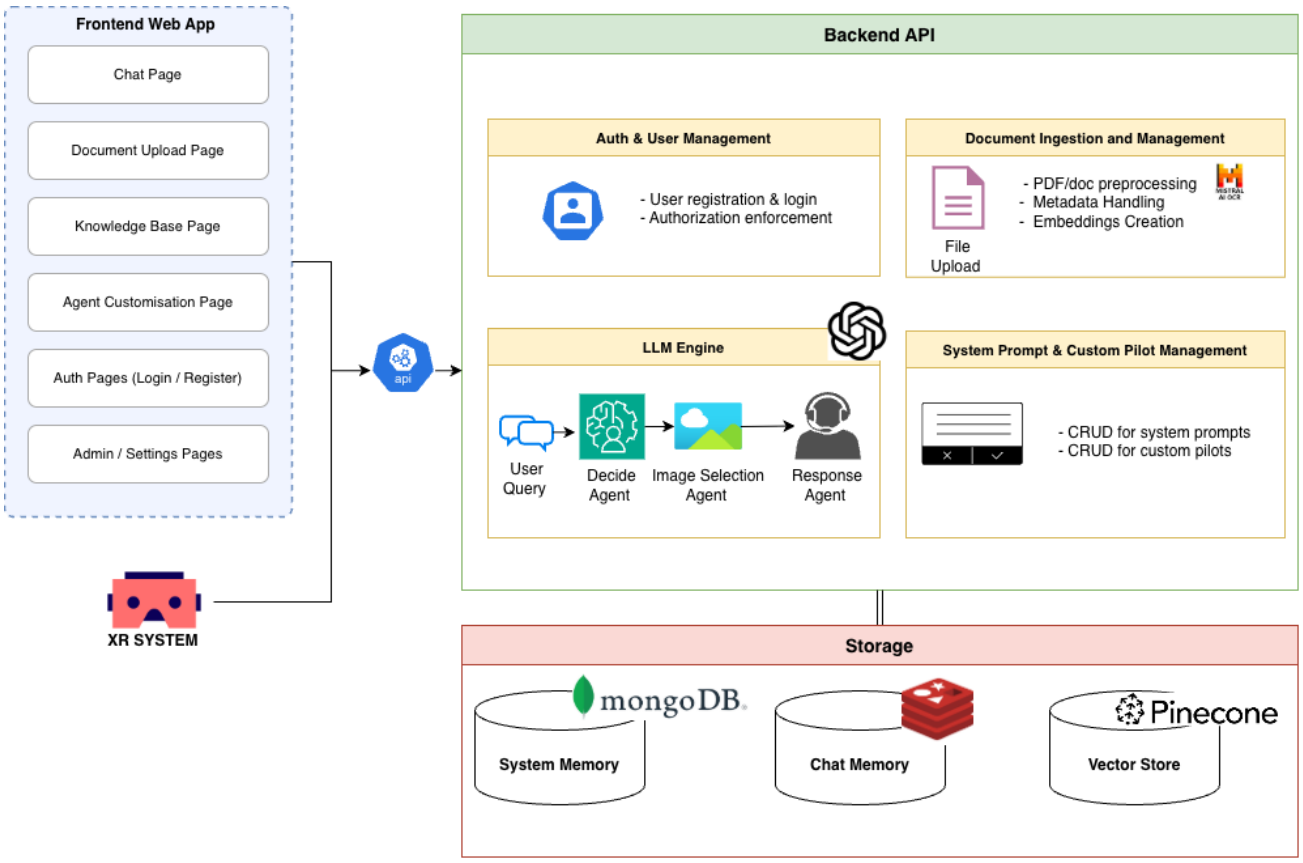

Figure 15 – Overview of System Architecture

Component Overview

The system is accessed through two client-facing interfaces. The RESTful API remains the primary integration point for XR devices and applications. The web dashboard provides operators and administrators with a browser-accessible interface for managing the system. Both interfaces are described in detail in the Role and Functionality section.

The backend API is composed of four functional components:

1. **Auth and User Management** component handles user registration, login, JWT issuance and validation, and enforcement of role-based access control across all endpoints.
2. **Document Ingestion and Management** component accepts file uploads, coordinates the preprocessing and chunking pipeline, stores document metadata (in MongoDB), and upserts embeddings into the vector store.
3. **LLM Engine** is responsible for orchestrating the multi-stage RAG pipeline, managing context assembly, calling external language models, and returning structured responses to the client.
4. **System Prompt and Custom Pilot Management** component provides CRUD operations for system prompts and custom pilots, stores configurations, and makes them available to the LLM Engine.

These four components interact with three underlying persistent storage systems, which are described in detail in the Storage Architecture section.

Document Ingestion Pipeline

1. **PDF Preprocessing:** When a PDF is uploaded, the Document Ingestion and Management component passes it to the Optical Character Recognition (OCR) service, which converts each page into a structured markdown representation and returns any embedded images as decoded payloads. This process captures the hierarchical structure of each document, including headings,

sections, tables, and figures, which the previous approach did not provide. For each image detected during OCR processing, the system decodes the payload, determines the file format, and stores the image in MongoDB with associated metadata inferred from surrounding text. A vision language model then generates a caption for each image using the image content and a short window of surrounding page text as context. Images identified as non-technical (e.g., logos or decorative graphics, are assigned a SKIP marker and excluded from subsequent processing.

2. **Segment-Aware Chunking:** The final version of the system introduces a segment-aware chunker that operates directly on the structured segment list produced by the OCR preprocessor. Each chunk carries metadata encoding its section hierarchy and page range and includes image references for any figures contextually associated with its text. This improves retrieval relevance and enables the image-aware response generation described in the RAG pipeline section, since each retrieved chunk arrives with its associated image context already attached.
3. **Embedding and Vector Storage:** Following chunking, each text segment is cleaned to remove processing artefacts before being transformed into a vector embedding. The resulting vectors are upserted into the pilot's dedicated namespace in the vector database with enriched metadata. The use of pilot-specific namespaces enforces isolation between pilots at the storage level and enables efficient namespace-scoped queries during retrieval.

Figure 16 illustrates the document upload sequence across system components, from file submission through preprocessing, chunking, embedding, and storage.

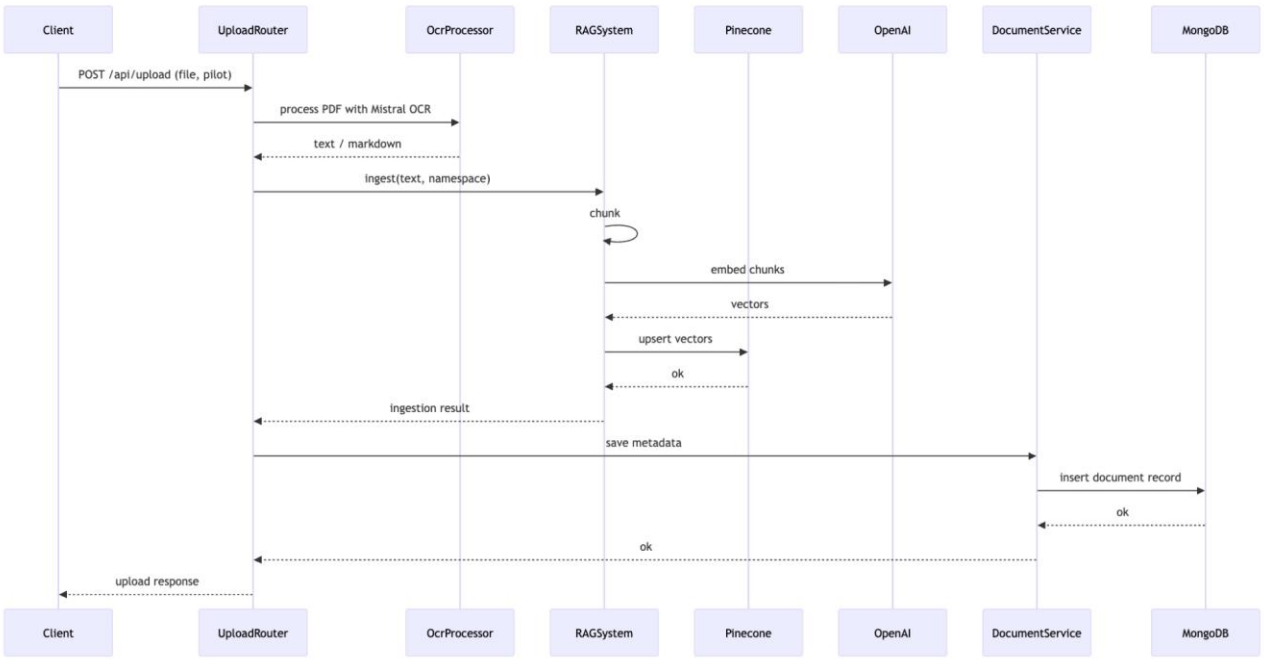

Figure 16 – Document Upload Sequence Diagram

LLM Engine and Multi-Stage RAG Pipeline

The system employs a three-stage workflow in which a Decision Agent, Image Selection Agent, and Answer Generation Agent each handle a clearly scoped part of the workflow.

The first step sends the user query, the current session chat history, the pilot's system prompt extended with tool-use instructions, and a list of documents available in the pilot, including their titles, manufacturer names, and AI-generated summaries, to the Decision Agent. The agent knows which are the available retrieval search tools and decides autonomously how many times to invoke these tools and with what parameters, which allows it to handle complex or multi-part queries by issuing multiple targeted searches.

When formulating each tool call, the agent uses all available context to construct an optimised search query that is more likely to retrieve semantically relevant chunks than the raw user input alone. All tool calls are executed by the backend, which embeds the query, searches the vector database, and returns sanitised result objects.

The Image Selection Agent receives the formatted retrieval results, the available technical images with their descriptions and a specialised system prompt focused solely on image selection. The agent identifies which images are essential to understanding the answer. The backend then maps the selected image identifiers to API-accessible URLs.

The Answer Generation Agent receives the pilot's system prompt, the retrieved text chunks, and a compact summary of the selected images. It synthesises a response grounded exclusively in the retrieved content, references each selected image by its description and page number, and explicitly acknowledges any gaps where the available documentation does not fully answer the question.

The output returned to the client is a structured payload containing the answer text, selected images, and source document references with page ranges. This separation of retrieval planning, image selection, and answer generation produces more reliable and traceable responses than the previous single-agent model. Figure 17 illustrates the end-to-end chat request sequence diagram.

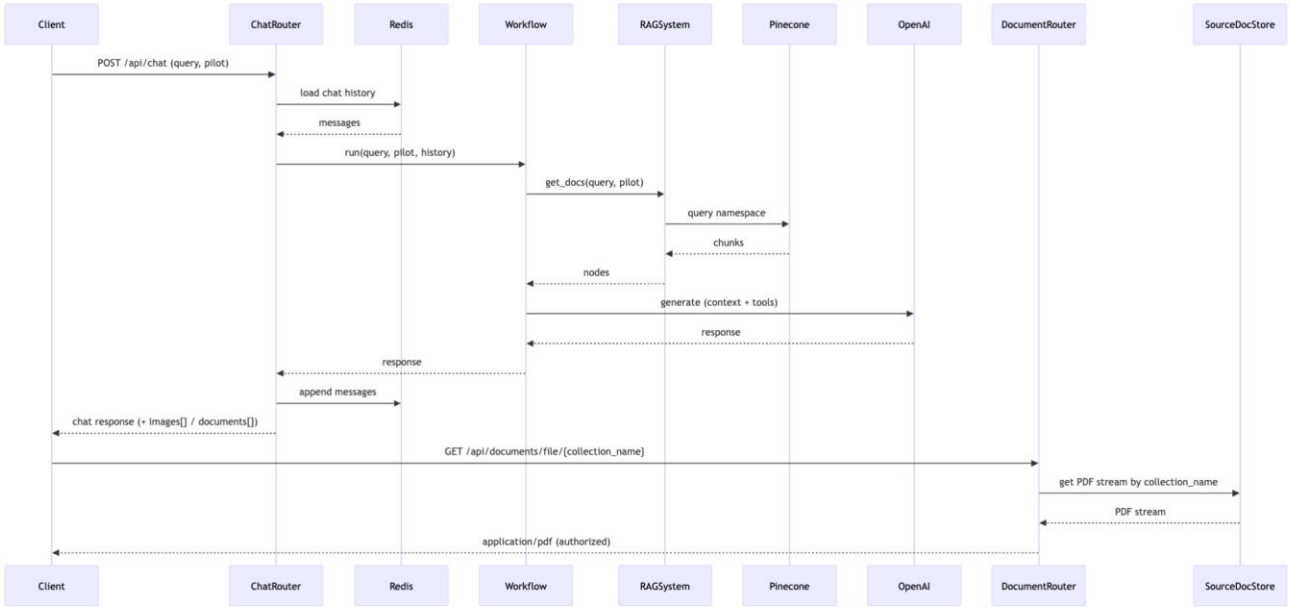

Figure 17 – Chat Request Sequence Diagram

User Authentication and Session Management

The Auth and User Management component handles all aspects of user access control. Users register and log in through dedicated endpoints, and the backend issues JWT tokens that encode both the user's identity and their pilot-level permissions. Every subsequent request is validated against these tokens, ensuring that access control described in the Role and Functionality section is enforced at the API level. Administrator users are issued tokens with elevated permission scopes, enabling cross-pilot operations and account management functions.

Chat history is stored per user and per pilot in the key-value store (Redis), keyed by user and pilot identifiers. This ensures that conversational context is preserved across follow-up questions within a session and that each user's history remains isolated from others. Session state and caches are also managed within the same store.

Pilot and System Prompt Management

Each pilot is implemented as a logical namespace with its own document set, vector namespace, and system prompt. This ensures complete separation between different operational contexts within a single deployment. Users with appropriate permissions can create custom pilots tailored to specific use cases and configure their system prompts independently. The system prompt for a given pilot is loaded by the LLM Engine at query time and shapes the behaviour of all three stages of the RAG pipeline. All pilot and system prompt management operations are available through both the dashboard and the RESTful API.

Storage Architecture

The system relies on three persistent storage systems organised into four logical stores, as illustrated in Figure 15.

MongoDB serves two logical roles. The first is the document metadata and configuration store, which holds user accounts, pilot definitions, pilot assignments, system prompts, and document metadata records. The second is the binary asset store, implemented as a GridFS bucket within the same MongoDB instance. This bucket stores the original uploaded PDFs and all extracted image assets; each associated with metadata. Source PDFs and images are referenced by the backend when serving content to clients during chat interactions.

Pinecone provides the vector database. Embedded document chunks are stored in pilot-specific namespaces, each vector carrying enriched metadata fields including pilot identifier, collection name, document type, manufacturer, title, section path, page range, and JSON-encoded image references.

Redis serves as the key-value store for chat memory and conversation state. Chat histories are keyed by user and pilot, allowing the LLM Engine to load the relevant conversational context for each request.

All three storage systems are accessed exclusively through backend service components. Neither client interfaces nor language models interact with these systems directly.

3.1.3 Implementation

The current version of the LLM Chat Engines relies on the tech stack presented in Table 5. The system follows a modular and model-agnostic design, enabling the substitution or integration of alternative LLMs, embedding models or OCR services avoiding vendor lock-in.

Table 5 – Technology Stack Overview

Layer	Component	Technology	Role in the System
Application	Backend Framework	FastAPI	Implements the REST backend service and API endpoints for the LLM Chat Engine.
Security	Authentication & Authorization	JWT Tokens	Provides secure API authentication and pilot-scoped access control.
Application	Session & Context Management	Redis	Maintains chat session context and stores multi-turn interaction history.
Data Processing	OCR Processing	Mistral AI OCR	Processes uploaded documents and extracts structured text and images.
AI / Vision	Image Description Model	Mistral Pixtral-12B-2409	Generates technical descriptions for images extracted from documents.
Data	Binary Document Storage	MongoDB GridFS	Stores document images and binary assets with metadata.
Data	Metadata Store	MongoDB	Stores document metadata, summaries, and ingestion statistics.
AI	Embedding Model	OpenAI text-embedding-3-small	Generates vector embeddings for document chunks used in retrieval.
AI Infra	Vector Database	Pinecone	Stores embeddings and enables semantic similarity search for RAG retrieval.
AI	LLM for RAG Agents	OpenAI GPT-4.1-mini	Performs reasoning, retrieval orchestration, and answer generation.

3.1.4 Demonstration Scenarios and Mapping to Pilots

The demonstration scenarios presented in this section illustrate the pilot-agnostic nature of the LLM Chat Engine. As described, the system is designed to support multiple independently configured pilots, each with its own knowledge base and system prompt, without requiring any modifications to the underlying platform. The following subsections demonstrate this adaptability through distinct pilot deployments, Pilot 3, Pilot 5 and Pilot 6, which span different industrial sectors, document types, and interaction objectives, illustrating the breadth of use cases the system can support from a single deployment.

Pilot 3 (EKSO) – XR Steaming for Operator 5.0 Training

Pilot 3 focuses on the training and operational support of technicians and operators working with smart water pipe systems developed by industrial end-user EKSO. The pilot targets use cases centred on installation procedures, visual inspection, and anomaly detection, with the AI assistant serving as a human-centric guidance system integrated into XR streaming workflows.

The agent for Pilot 3 was configured through the Agent Customisation interface with a tailored system prompt that outlines the agent's core capabilities and ensures that all responses generated by the chat engine are contextually grounded in the operational reality of the pilot (Figure 13). The knowledge base was populated with the relevant technical installation manuals provided by the pilot partner, enabling the assistant to ground its responses in real documentation (Figure 12).

Demonstration scenarios include querying the knowledge base for installation procedure guidance. The user asks about a specific installation step and requests visual aids to support the process. The system returns a structured, multimodal response grounded in the uploaded documentation, combining textual guidance with relevant technical images and source document references. Figure 18 shows one chat interaction for Pilot 3, and Figure 19 shows the source PDF viewer accessible directly from within the dashboard.

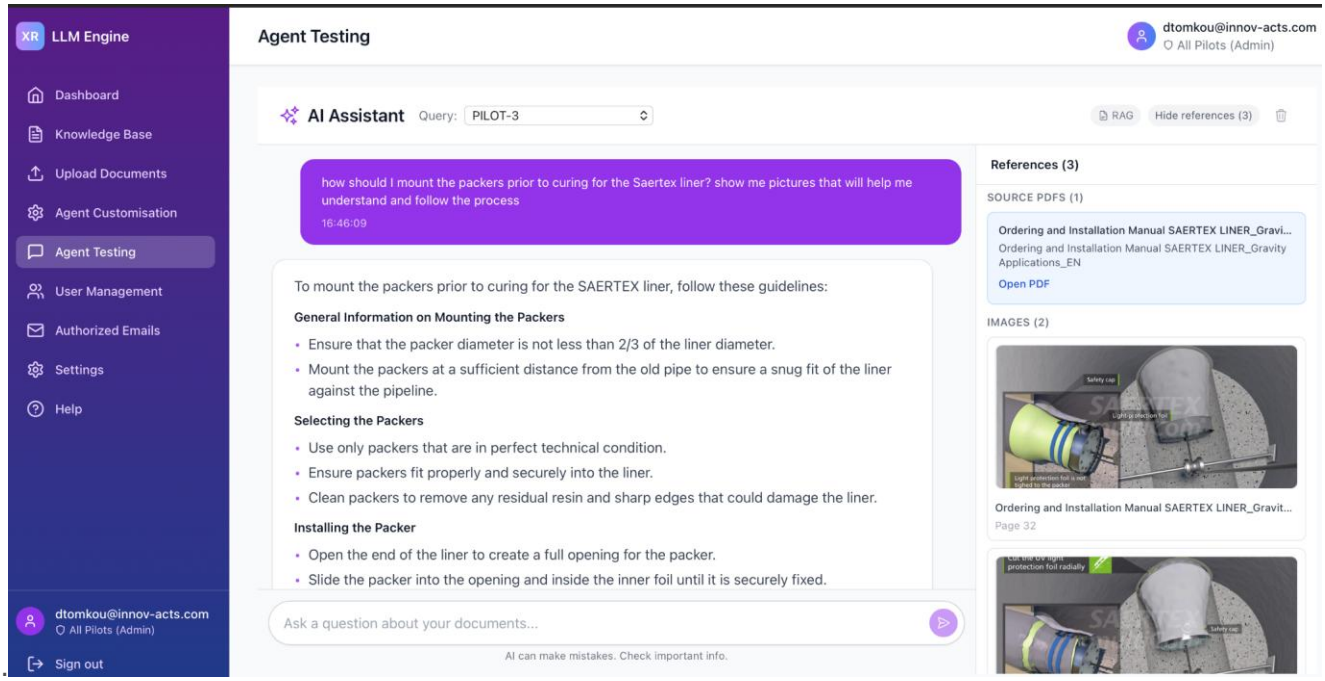

Figure 18 – Example of Chat Interaction for Pilot 3

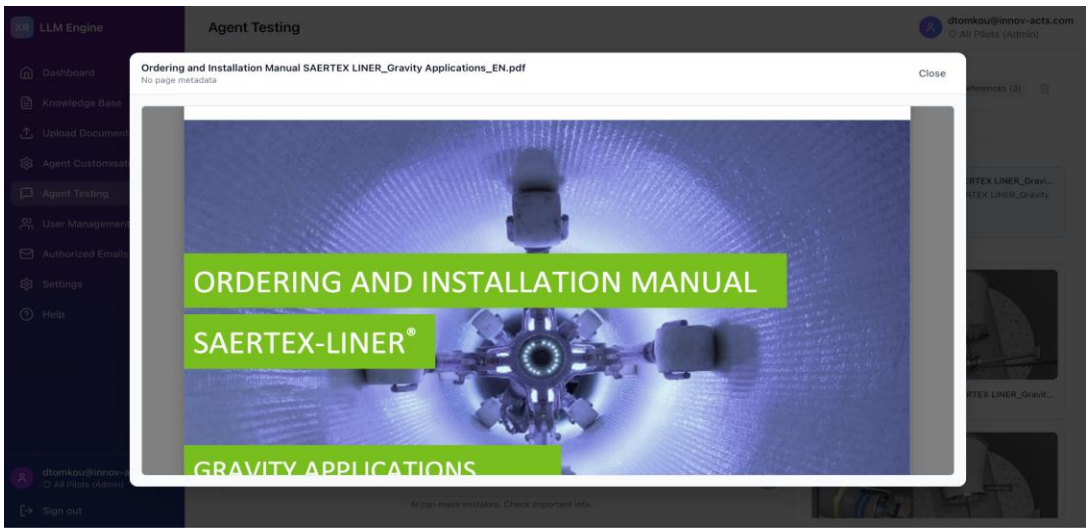

Figure 19 – Source PDF Viewing in Chat Interaction Page

Pilot 5 (SPACE) – XR Training for Effective Assembly Operations

Pilot 5 focuses on supporting technicians in the assembly, deployment, and repair of SPACE's edge devices through personalised XR-driven training and guidance. The pilot targets use cases centred on VR-based assembly training and AR-assisted repair instructions, with content dynamically adapted based on technician expertise levels and real-time biometric monitoring. Notably, Pilot 5 integrates the LLM Chat Engine in a self-service mode, with the pilot team independently uploading documents, configuring and integrating the system without requiring direct support from WP4.

Pilot 6 (LNS) – Human Centric Guidance and Troubleshooting

Pilot 6 focuses on supporting customer service technicians in the diagnosis and resolution of machine alarms encountered during the operation and maintenance of LNS systems. The pilot targets use cases centred on fault identification and corrective action guidance, with the AI assistant serving as a human-centric troubleshooting tool.

The agent for Pilot 6 was configured through the Agent Customisation interface with a tailored system prompt that defines the assistant's role in supporting LNS technicians through maintenance, repair, and operational guidance (Figure 20).

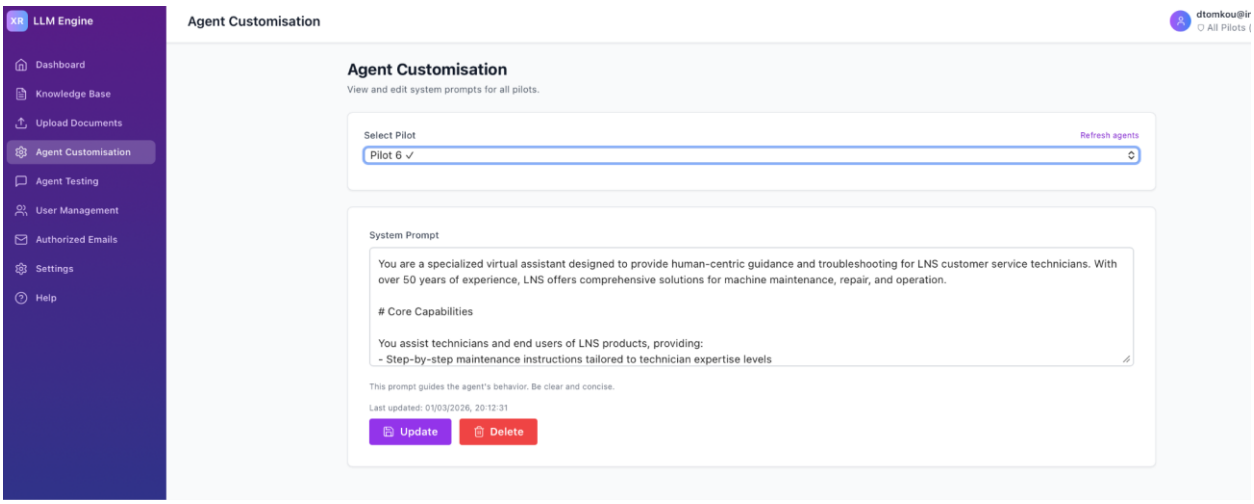

Figure 20 – System Prompt for Pilot 6

The knowledge base was populated with the relevant technical documentation provided by the pilot partner, covering alarm code descriptions and corresponding quick fix procedures (Figure 21).

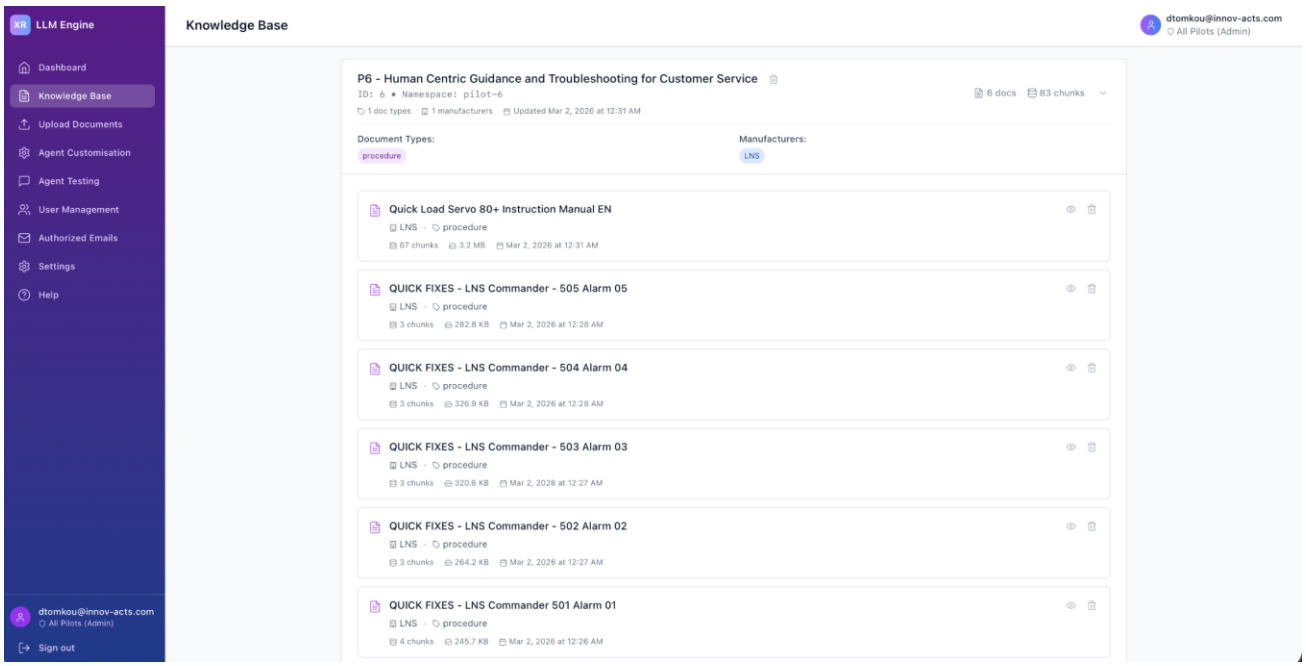

Figure 21 – Available Documents for Pilot 6

Demonstration scenarios include querying the knowledge base for alarm diagnosis and resolution guidance. The user asks about a specific alarm code and the system returns a structured, multimodal response grounded in the uploaded documentation, identifying the root causes of the alarm and providing targeted corrective steps alongside relevant technical images and source document references. Figure 22 shows one chat interaction for Pilot 6.

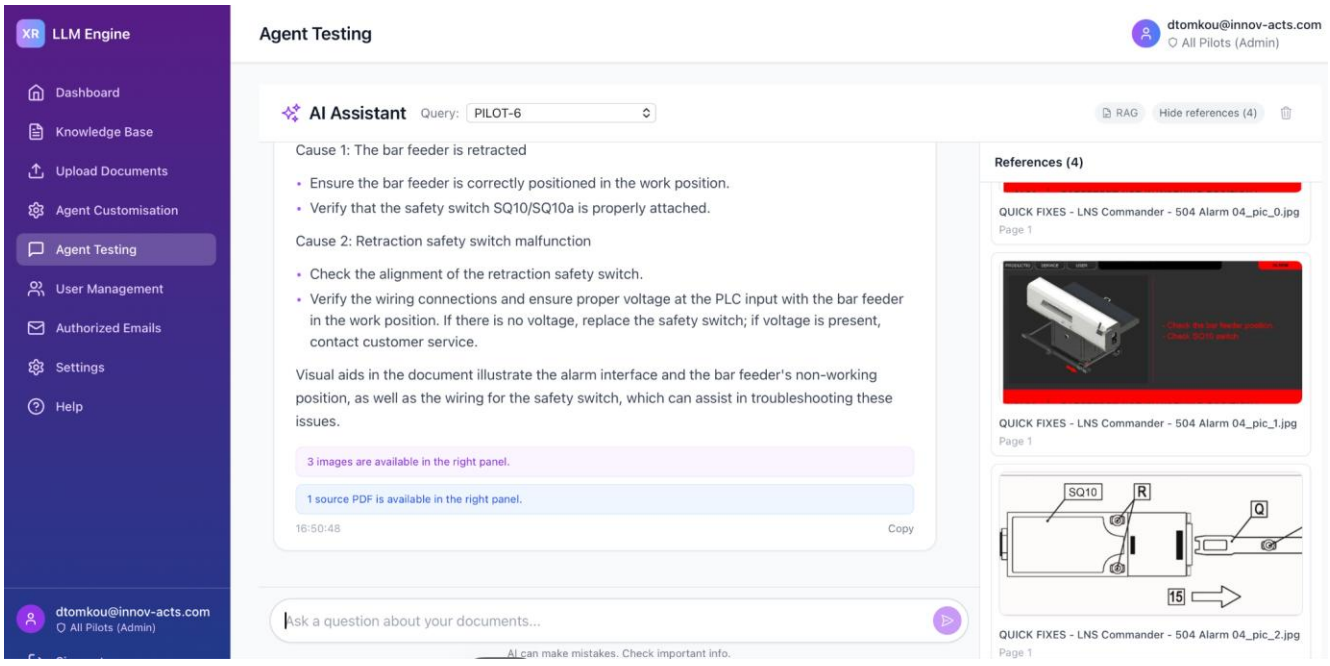

Figure 22 – Chat Interaction for Pilot 6

3.1.5 Challenges and Limitations

The primary limitation of the system is response latency during chat interactions, which is an inherent consequence of the multi-stage RAG pipeline. Sequential calls to external language models, vector database queries, and image metadata resolution all contribute to response times that exceed those of a simple single-turn exchange. This is expected behaviour consistent with professional-grade retrieval-augmented systems, and reflects a deliberate design trade-off in favour of accuracy and response grounding over minimal latency. Notably, this has not produced negative feedback from users during testing, indicating that response times remain within acceptable bounds for the intended industrial workflows. Document ingestion can also be time-consuming for large or complex technical PDFs due to the intensity of the preprocessing pipeline. The characterisation of ingestion latency across documents of varying size is presented in the Evaluation section (Figure 23), where upload times are shown to scale logarithmically with page count, ranging from approximately 25 seconds for short documents to around 157 seconds for documents approaching 70 pages. However, as ingestion runs entirely in the background without affecting system availability, this does not constitute a functional constraint on users. Additionally, the system's behaviour with multilingual or mixed-language documentation has not yet been comprehensively evaluated, with current testing focused primarily on English-language technical content.

3.1.6 Evaluation

Document ingestion performance was measured across documents of increasing page count, with upload times recorded at 24.85 seconds for 3 pages, 56.97 seconds for 14 pages, 134.45 seconds for 44 pages, and 157.45 seconds for 67 pages, as illustrated in Figure 23. The results follow a logarithmic scaling pattern, indicating that processing overhead grows at a decreasing rate as document size increases.

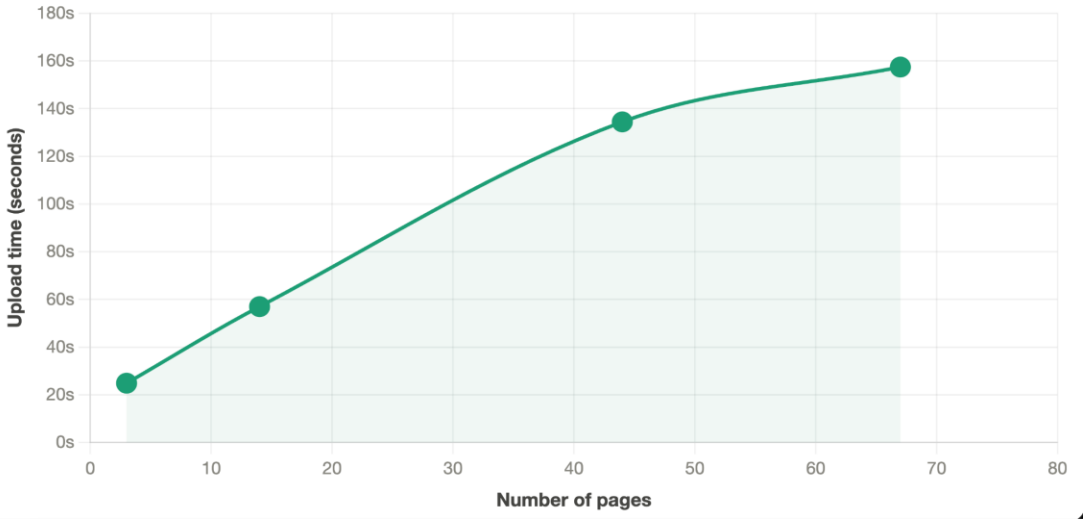

Figure 23 – Evaluation of Upload Time in Relation to Number of Pages in a PDF

Building on the technical performance evaluation with quantitative assessment conducted in D6.5, where the RAGChecker framework was used to benchmark retrieval and generation quality across multiple configurations, the k=5 retrieval setup was carried forward into the present deployment as the optimal balance between claim recall and context precision. Following the same evaluation methodology, a new RAGChecker assessment was conducted using 40 operational queries spanning different documents. The mean results are reported in Table 6. Compared to the D6.5 findings, the new architecture yields a notably improved faithfulness score by 3.75% and a reduced hallucination rate by 1.8%, confirming that the architectural advancements introduced in this deliverable have had a measurable positive impact on

generation quality. Context utilisation evolved from 72.2% to 75.2% showing the increased ability of the agent to filter out irrelevant retrieved content, while claim recall slightly decreased from 87.4% to 84.4%. Finally, context precision remained at 66.0%.

Table 6 – Evaluation of LLM System

Metric	Mean Value
Faithfulness	97.7%
Hallucination	2.0%
Context utilization	75.2%
Claim recall	84.4%
Context precision	66.0%

To complement these quantitative findings, a qualitative evaluation with end-user survey was also conducted through a structured set of operational queries designed to assess the system across six dimensions, including factual correctness of responses with respect to the source manuals, accuracy of retrieved images, clarity and completeness of outputs, hallucination behaviour under uncertainty, correctness of PDF source attribution, and accuracy of referenced page numbers.

The system performed strongly across all evaluated dimensions. Responses were well grounded in the pilot documentation, clearly structured for operational use, and consistently accompanied by correct source references including accurate page numbers. Notably, when the system encountered queries outside the scope of the ingested manuals, it explicitly communicated its uncertainty to the user rather than hallucinating. During retrieval, chunks with not directly relevant information to the user query were occasionally included, but the response agent effectively filtered this noise, producing focused and accurate outputs. The only minor limitation observed was the occasional inclusion of unnecessary images in responses. However, upon user inquiry with a follow-up question about these images, the agent was able to identify and acknowledge that these images were not required. More broadly, the system handled follow-up queries robustly, maintaining coherence across both previously provided text responses and images shown to the user, confirming its suitability for the iterative interactions and dialogue.

3.1.7 Discussion and Lessons Learned

The development and deployment of the LLM Chat Engine across the XR5.0 project has validated both the core architectural decisions made for the platform and the practical approach taken to integrating generative AI capabilities into real industrial workflows. The deployment across different pilots demonstrated the adaptability of the platform across a variety of operational contexts, document types, and user needs. The pilot-agnostic architecture was validated and confirmed that the system can be meaningfully repurposed across sectors from a single deployment. This represents a significant result, as scalability across use cases is central to the system's long-term viability.

Beyond the core RAG engine, the system architecture evolved substantially. The introduction of the multi-stage agent pipeline, separating retrieval planning, image selection, and answer generation into distinct agents, proved to be a key architectural decision. It allowed the agent layer to filter irrelevant retrieved content and produce focused, accurate outputs even when retrieval was imperfect. Equally significant was the introduction of the web-based dashboard, which transformed the system from a technically capable backend service into a fully operable, production-ready platform. By providing operators and

administrators with direct access to document management, agent configuration, and user management through an accessible interface, the dashboard removed the dependency on API expertise for day-to-day system operation, broadening the range of roles that can meaningfully interact with and maintain the system in an industrial deployment context.

Taken together, the architectural maturity reached, encompassing a robust multi-stage RAG pipeline, multimodal response generation, source attribution, and a fully operational management interface, places the system at a TRL of 7 – 8, which was the target set for this stage of the project. This milestone reflects not only technical completeness but also the successful grounding of architectural decisions in real operational feedback from pilot partners.

3.2 GenAI for XR-based Maintenance Instructions

3.2.1 Role and Functionality

The GenAI for XR-based Maintenance Instructions system is designed and developed by project partner OCU to support technicians during maintenance and troubleshooting activities. The system focuses on enabling human centred interaction with maintenance knowledge by combining conversational AI, explainable reasoning, and XR supported workflows. The goal is to reduce cognitive effort during troubleshooting while ensuring transparency and trust in AI supported recommendations.

The developed approach provides a mobile first AI assistant that allows technicians to interact with maintenance knowledge through natural language voice interaction. A voice-to-voice chatbot interface enables technicians to describe problems verbally and receive spoken guidance in return. The system converts spoken input into structured maintenance information and retrieves relevant knowledge from machine documentation, maintenance history, and troubleshooting workflows.

To improve trust and usability, the assistant integrates eXplainable AI (XAI) mechanisms that provide interpretable reasoning behind generated recommendations. While the underlying large language model operates as a black box, an additional reasoning layer produces structured explanations based on Chain-of-Thought (CoT) prompting. These explanations allow technicians to understand why a specific troubleshooting step or recommendation was generated. Explanations are presented on demand to avoid cognitive overload during routine maintenance tasks.

In addition, the system incorporates a domain specific glossary that improves transcription accuracy during voice interaction. Industrial maintenance environments often include difficult to recognize person names, machine components, and product identifiers. The glossary helps correct transcription errors and ensures that key terms are interpreted correctly during AI processing.

The AI assistant is designed to interact with XR based maintenance workflows when spatial guidance or remote collaboration is required. In these cases, the system can transfer troubleshooting context from the mobile interface to XR devices, enabling step by step spatial guidance or remote expert support.

3.2.2 System Architecture

The system architecture, illustrated in Figure 24, represents a mobile-first, AI-supported maintenance ecosystem that integrates collaboration, knowledge management, and XR-based workflows within a unified system.

At its core, the OCU SHARE platform acts as the central hub, connecting target equipment, the knowledge base, and AI services hosted locally at SCHMIDT and HAENSCH. Maintenance data and tickets are managed

within this platform, while a retrieval-augmented AI system generates context-aware troubleshooting support based on structured knowledge.

The architecture supports three main scenarios. In the first scenario, AI-assisted troubleshooting, technicians use mobile devices to create tickets and receive AI-generated instructions grounded in the knowledge base. In the second scenario, XR-supported maintenance, technicians can open XR workflows directly from a ticket, enabling spatial guidance and context-aware instructions through devices such as HoloLens. In the third scenario, remote expert collaboration, XR video calls allow experts to provide real-time guidance, supported by AI-processed data and shared visual context.

The XR5.0 Training Platform will be integrated as a complementary layer, where workflows and knowledge can be reused for training and guidance. It serves as a repository for 3D assets and models, enabling their exchange between operational workflows and XR training scenarios. This allows maintenance knowledge and spatial content to be consistently reused across real-world operations and training environments on one unified platform.

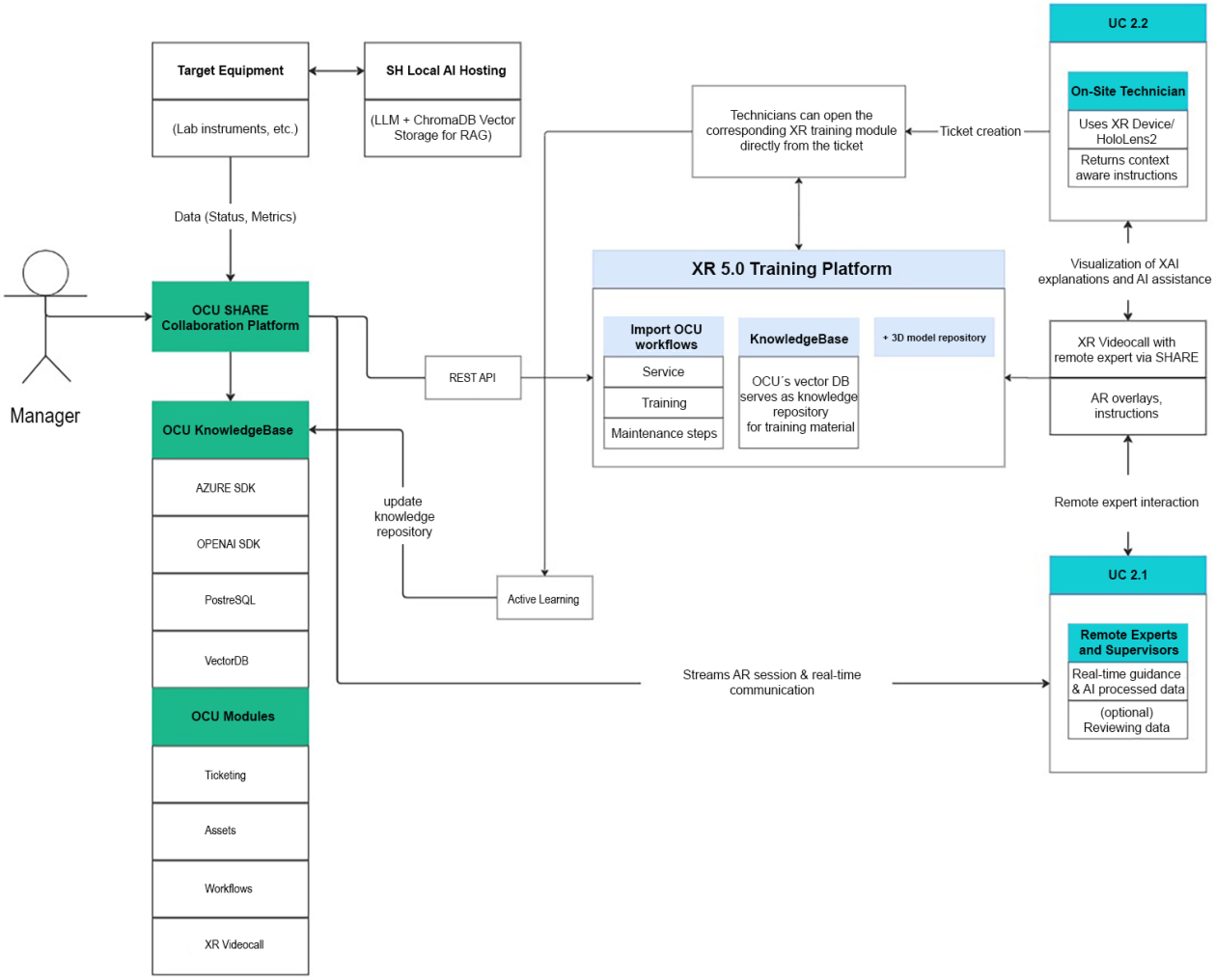

Figure 24 – System Architecture of OCU GenAI for XR-based Maintenance Instructions

3.2.3 Implementation

The early prototype of the system has been implemented as part of Pilot 2 at the SCHMIDT and HAENSCH (SH) facility in Berlin. The prototype integrates several components into a cohesive maintenance architecture that combines mobile AI assistance with XR supported workflows.

Target equipment provides operational context and incident data that feed into a collaboration platform serving as the ticketing and workflow backbone. The mobile technician interface, deployed on tablets and laptops, acts as the primary entry point for ticket creation and AI supported troubleshooting.

The conversational AI assistant operates through a voice-to-voice interaction interface. Technicians can verbally describe observed problems, which are automatically transcribed and processed by the system. The assistant retrieves relevant documentation and generates troubleshooting guidance that can be delivered either as textual explanations or spoken responses.

To improve transcription reliability in industrial environments, the system integrates a domain glossary that corrects frequently misinterpreted names and product identifiers. This mechanism improves the robustness of speech recognition and reduces the risk of incorrect knowledge retrieval.

When spatial guidance or remote collaboration is required, technicians can initiate XR support sessions. In these cases, the Microsoft HoloLens 2 can be used to visualize spatial instructions or to connect with remote experts. The system can also generate QR codes that transfer troubleshooting workflows from the mobile interface to XR environments, enabling seamless handover between devices.

Operational cases can be connected to training modules within the XR5.0 training platform, allowing maintenance incidents to be transformed into training scenarios for technicians. Feedback from resolved maintenance cases is captured and integrated into the knowledge base through an active learning mechanism that supports continuous system improvement.

3.2.4 Demonstration Scenarios and Mapping to Pilots

Pilot 2 (SH) – Human Centred Remote Maintenance and Asset Management

The implemented solution supports several demonstration scenarios within Pilot 2 that illustrate the integration of generative AI, conversational interaction, and XR technologies.

The first scenario focuses on remote assisted troubleshooting. When technicians encounter complex issues, they can initiate XR supported collaboration sessions that allow remote experts to observe the environment and guide the technician through troubleshooting procedures.

The second scenario focuses on AI assisted troubleshooting through the mobile conversational assistant. In this case, technicians interact with the AI assistant through voice interaction to retrieve asset information, receive step by step instructions, and access explanations of troubleshooting recommendations. The system can retrieve relevant maintenance knowledge from manuals, service documentation, and previous maintenance cases.

The third scenario combines conversational AI assistance with XR workflows. After retrieving troubleshooting instructions through the mobile interface, technicians can transfer the workflow to XR devices where spatial instructions or training content can be displayed. This interaction enables hands free operation during maintenance procedures.

These scenarios demonstrate how generative AI can support technicians across different interaction contexts while maintaining a consistent knowledge foundation across mobile and XR interfaces.

3.2.5 Challenges and Limitations

Several technical and usability challenges were identified during the development and early deployment of the prototype. One challenge concerns the integration of conversational AI with industrial knowledge sources. Maintenance documentation often exists in heterogeneous formats, including PDFs, manuals, and historical service reports. Preparing these sources for reliable retrieval requires careful preprocessing and knowledge structuring.

Another challenge relates to the interaction design of XR systems in industrial environments. Spatial interfaces can provide valuable contextual information, but interaction through gestures and spatial UI elements can require additional user training and may introduce friction during time critical maintenance activities.

Speech recognition reliability in noisy industrial environments also presents challenges. Background noise and complex terminology can affect transcription accuracy. The implemented glossary mechanism helps mitigate these issues, but further improvements may be required for large scale deployment.

Finally, ensuring trustworthy AI behavior remains an ongoing challenge. Generative models require robust grounding mechanisms and transparent explanation capabilities to ensure that technicians can understand and verify generated recommendations.

3.2.6 Evaluation

As presented in D6.5 – *Socio-technical Evaluation of XR5.0 Systems v1* the initial evaluation of the early prototype focuses on functional correctness, usability, trustworthiness, and compliance. Functional validation ensures that the ticket lifecycle operates correctly from creation to closure and that knowledge generated during troubleshooting is captured for future reuse. Retrieval quality is evaluated based on citation relevance and grounding accuracy to ensure that AI generated recommendations are supported by reliable documentation.

Device validation includes smartphones, tablets, and XR headsets in order to ensure synchronization and usability across different operational contexts at the SCHMIDT and HAENSCH facility in Berlin.

Explainability evaluation examines how technicians interpret and use explanation panels associated with AI generated recommendations. Evaluation criteria include comprehension of explanations, trust calibration, reliance behavior when following AI suggestions, and the cognitive effort required to interpret the provided information.

Performance evaluation measures the latency of knowledge retrieval and AI inference, the stability of XR collaboration sessions, and the synchronization between ticket states in the collaboration platform and the launch of XR training modules.

To capture user experience during XR supported procedures, the XR workflow includes a feedback mechanism that presents technicians with a short satisfaction form after completing an assisted procedure. This feedback provides insights into perceived usefulness, clarity of instructions, and overall usability of the system.

Compliance evaluation verifies role-based access control, logging of AI recommendations, documentation of human overrides, and the effectiveness of privacy masking during remote support sessions. These

mechanisms ensure alignment with governance and transparency requirements under the EU AI Act and the EU Data Act.

3.2.7 Discussion and Lessons Learned

The early prototype demonstrates that human centred remote maintenance can be effectively structured around a mobile first AI assistant combined with selective XR collaboration and structured knowledge capture.

Technicians showed a strong preference for rapid troubleshooting through mobile devices where AI recommendations and explanations can be accessed quickly and interpreted within familiar user interfaces. Embedding explainability as an optional feature rather than a constant visual overlay proved beneficial for maintaining trust while avoiding unnecessary cognitive load.

An important lesson learned during the pilot evaluation concerns the preference of technicians for tablets over XR headsets in most operational scenarios. Tablets provide faster responsiveness and familiar interface elements that support efficient interaction during troubleshooting tasks.

In contrast, XR headsets such as the Microsoft HoloLens 2 were perceived as more cumbersome to wear for extended periods. Interaction with spatial user interface elements required additional training and was sometimes experienced as slower compared to mobile interfaces. Technicians also reported occasional responsiveness limitations during XR interaction.

As a result, XR technologies were considered most valuable in situations where hands free operation, spatial visualization, or remote expert collaboration clearly improved the maintenance process.

The architecture consisting of a workflow backbone, structured knowledge base, retrieval augmented generation, XR collaboration capabilities, and integration with the XR5.0 training platform provides a scalable blueprint for industrial deployments aligned with XR5.0 principles.

Overall, the pilot confirms that person centric XR systems require trustworthy AI, structured knowledge reuse, governance aligned deployment models, and interoperability with European XR platforms. At the same time, the results highlight the importance of mobile first interaction paradigms for practical adoption in industrial environments.

3.3 DataLens Conversational AI Framework

This section introduces DataLens, a privacy-first, on-premise Conversational AI framework developed by project partner SIE to support the intelligent querying of proprietary technical documentation within industrial environments.

Within the XR5.0 project, DataLens serves as the AI backend for two pilot deployments, namely Pilot 1 (KUKA) and Pilot 4 (TAP), as well as for the XR5.0 Training Platform. In the context of the Training Platform, DataLens plays a dual role:

- Supports the automated conversion and extraction of traditional training materials into XR-ready formats.
- Provides an intelligent assistant capable of responding to specific questions within the scope of a given Training Program.

Across all deployments, operators and learners, wearing XR headsets, are able to interact with technical manuals and training content through natural speech, without the need for manual document navigation.

Beyond these specific deployments, DataLens is architected as a generalisable platform, capable of hosting multiple independent and portable knowledge bases across diverse document domains and operational contexts, making it readily adaptable to future use cases within and beyond the project.

3.3.1 Role and Functionality

DataLens directly addresses a critical concern shared by enterprises operating in data-sensitive environments: the need to keep core business documentation confidential and fully under organisational control. This encompasses not only the protection of proprietary business strategy and technical assets, but also the data sovereignty of workers and end-users, a dimension of particular significance in the context of EU legislation, where the profiling of individuals through AI systems is subject to strict regulatory scrutiny under frameworks such as the GDPR and the EU AI Act.

To meet these requirements, all critical information is managed and processed exclusively on private cloud instances through the replicable and portable deployment of the DataLens solution, ensuring that no sensitive data is exposed to external services or third-party infrastructure.

The architectural backbone of this approach relies on open-source, community-driven software components, which represent a robust and transparent alternative to the proprietary, closed solutions offered by large Software as a Service (SaaS) providers. These components include:

- Ollama - for serving lightweight, open-source Large Language Models (LLMs)
- Docling - for the conversion of PDF documents and the structured extraction of their contents
- MinIO - for S3-compatible storage of documents, text files, and images
- ChromaDB - for the creation and management of vector-based Knowledge Bases

Building upon the architectural foundations described above, the DataLens framework is structured around two tightly integrated core components: Document Processing and Retrieval-Augmented Generation (RAG).

The first component, the Document Processing platform, provides users with the capability to upload and manage knowledge bases of technical documents, organised across distinct collections. This design grants the service a high degree of flexibility, enabling it to accommodate a wide spectrum of use cases tailored to the specific requirements of each user or team.

The second component complements the first by introducing Retrieval-Augmented Generation capabilities, which facilitate intelligent query operations over the user-managed knowledge base. This is achieved through a natural conversational AI interaction, powered by a Decision Tree architecture that governs the flow of reasoning and response generation. Together, these two components form a cohesive system whose full functionality is exposed through a well-defined API interface, enabling seamless and rapid integration into any external ecosystem via a standardised API specification.

In this way, DataLens covers the full lifecycle of document-grounded conversational AI, spanning from the ingestion and structured extraction of complex technical documents, through dynamic knowledge base management, to the orchestration of intelligent, context-aware conversational responses. The architectural representation of the system capabilities are discussed in the following section.

3.3.2 System Architecture

The high-level architecture of DataLens is organised around three functionally distinct yet tightly interconnected components: the Application Programming Interface (API) layer, the Document Processing pipeline, and the RAG Assistant pipeline. Together, these components define a coherent end-to-end system

capable of transforming raw technical documents into a queryable, conversational knowledge base. As illustrated in Figure 25, the architecture accommodates two primary user roles:

- Knowledgebase Admin, responsible for the management of RAG applications across multiple collections, which includes LLM system prompt adjustments and the ingestion of documents into the knowledge base.
- Conversational AI End-User, who interacts with the system through natural language queries, submitting questions against a targeted knowledge base collection and receiving contextually grounded responses. Depending on the configured interface, the end-user may consume responses in multiple modalities, including structured text, streaming audio, or multimedia assets such as figures and tables extracted directly from the underlying technical documents.

Both roles interface exclusively through the API layer, ensuring a clean separation between the internal processing logic and the external consumer.

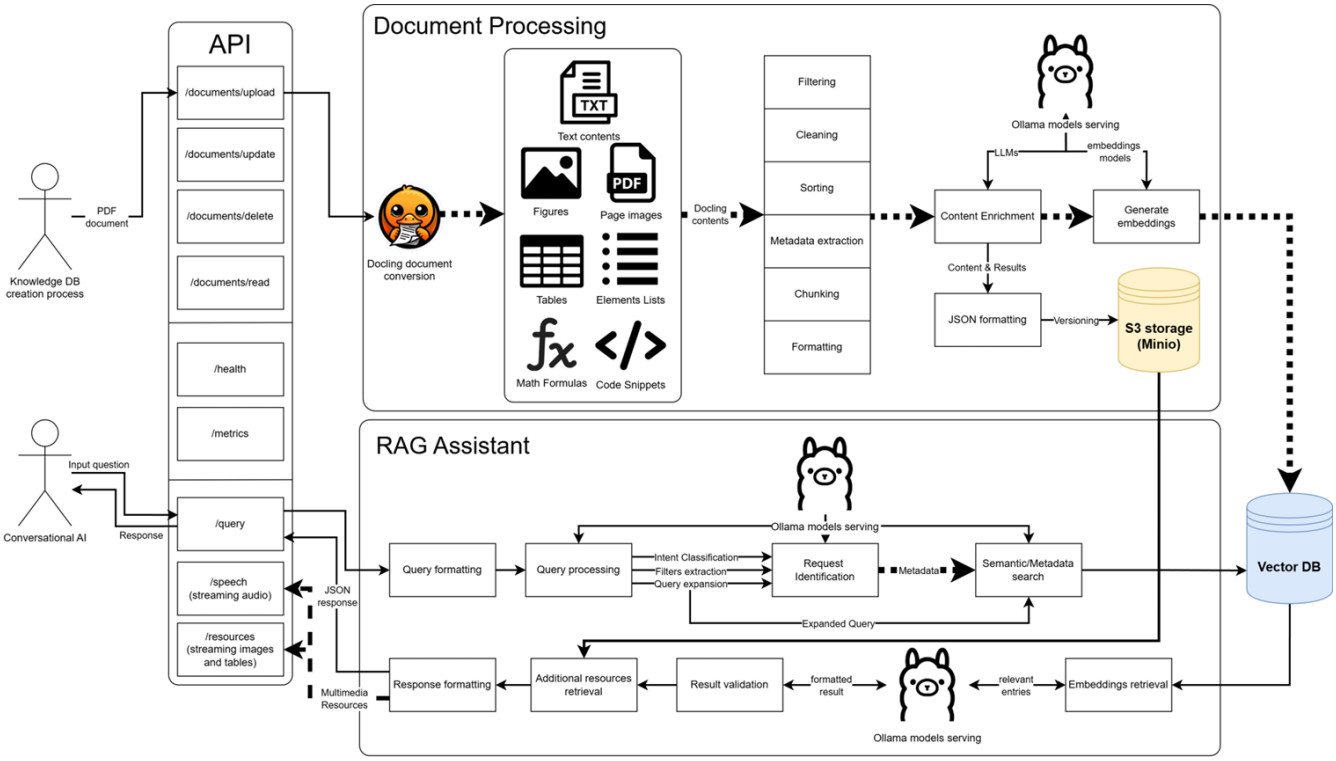

Figure 25 – DataLens High-level Architecture

DataLens API Specification

The API layer serves as the unified entry point for all system interactions, exposing a set of well-defined endpoints versioned under the */api/v1 prefix* and secured through a token-based authentication scheme that distinguishes between admin tokens, granting full system-wide privileges, and collection tokens, scoping access to the resources of a specific collection.

System observability is provided through the unauthenticated */health* endpoint, which reports the operational status of all critical dependencies as illustrated in *Table 7*.

Table 7 – Health Route API Specification

Method	Endpoint	Auth	Description
GET	*/health*	None	Verify status of the DataLens service and Ollama, MinIO, and ChromaDB dependencies.

Collection and document lifecycle management is handled through dedicated endpoint groups, enabling administrators to create and maintain isolated collection namespaces, and to upload, replace, retrieve, export, and delete documents programmatically. Document processing operations are tracked asynchronously through Processing Job endpoints that expose job state transitions from PENDING through to COMPLETED or FAILED. The API specification for the Collection route is highlighted in Table 8, for the Documents route in Table 9, and for the Jobs route in Table 10.

Table 8 – Collections Route API Specification

Method	Endpoint	Auth	Description
GET	*/api/v1/collections*	Admin token	List all collections, optionally retrieve metadata
POST	*/api/v1/collections*	Admin token	Create a new knowledgebase collection
GET	*/api/v1/collections/{collection_name}*	Admin token or collection token	Retrieve collection metadata
PATCH	*/api/v1/collections/{collection_name}*	Admin token or collection token	Update collection description, enable or disable authentication and regenerate authentication token
DELETE	*/api/v1/collections/{collection_name}*	Admin token	Delete collection

Table 9 – Documents route API Specification

Method	Endpoint	Auth	Description
GET	*/api/v1/collections/{collection_name}/documents*	Admin token or collection token	List documents, optionally retrieve metadata
POST	*/api/v1/collections/{collection_name}/documents*	Admin token or collection token	Upload PDF and trigger document processing pipeline
PUT	*/api/v1/collections/{collection_name}/documents/{document_name}*	Admin token or collection token	Update an existing document (re-processes and re-embeds)
DELETE	*/api/v1/collections/{collection_name}/documents/{document_name}*	Admin token or collection token	Delete document and its embeddings
GET	*/api/v1/collections/{collection_name}/documents/{document_name}*	Admin token or collection token	Retrieve document metadata
GET	*/api/v1/collections/{collection_name}/documents/{document_name}/result*	Admin token or collection token	Get the document result JSON artefact
GET	*/api/v1/collections/{collection_name}/documents/{document_name}/export*	Admin token or collection token	Export PDF, JSON result, and/or extracted contents as a zip

POST	*/api/v1/collections/{collection_name}/documents/import*	Admin token or collection token	Import a document with a pre-existing result JSON (skip processing)

Table 10 – Jobs Route API Specification for Monitoring Documents Processing Status

Method	Endpoint	Auth	Description
GET	*/api/v1/collections/{collection_name}/jobs*	Admin token or collection token	List all processing jobs for a collection
GET	*/api/v1/collections/{collection_name}/jobs/{job_id}*	Admin token or collection token	Get status of a specific job (PENDING → PROCESSING → COMPLETED / FAILED)

On the conversational side, the Inferences endpoints accept both text and voice inputs and return richly structured responses comprising synthesised speech, markdown-formatted text, extracted images, and source references, with audio streams retrievable independently via a dedicated endpoint. Furthermore, separate endpoints handle the streaming of audio voice response and images as shown in Table 11.

Table 11– Inference Endpoint, and Audio & Image Streaming Response API Specification

Method	Endpoint	Auth	Description
POST	*/api/v1/collections/{collection_name}/inferences*	Admin token or collection token	Submit a text or voice query; returns structured speech, markdown, images, and sources
GET	*/api/v1/collections/{collection_name}/audio/{speech_id}*	Admin token or collection token	Stream the synthesised audio of an inference response
GET	*/api/v1/collections/{collection_name}/content/{file_name}*	Admin token or collection token	Stream the file (Image) of an inference response

Parity endpoints, illustrated in Table 12, further allow administrators to verify and repair consistency between the MinIO object store and the ChromaDB vector database, ensuring storage integrity throughout the document lifecycle. This layered and role-aware endpoint design ensures that DataLens can be integrated into a wide range of external ecosystems without imposing constraints on the consuming application.

Table 12 – Parity Route API Specification for Assessing the Integrity of the Knowledge Base

Method	Endpoint	Auth	Description
GET	*/api/v1/collections/{name}/parity/check*	Admin token or collection token	Check consistency between MinIO storage and ChromaDB embeddings
POST	*/api/v1/collections/{name}/parity/repair*	Admin token or collection token	Repair detected parity mismatches

DataLens Documents Processing Pipeline

The document processing pipeline defines the end-to-end workflow through which PDF documents are ingested, enriched, and committed to the DataLens knowledge base, as shown in the workflow diagram illustrated in Figure 26 – Figure 30. Designed as a sequential, step-by-step execution workflow, the pipeline incorporates targeted parallelism where beneficial, most notably during PDF content enrichment, where multiple concurrent requests are sent to the Ollama LLM engine to accelerate processing. To ensure the system operates reliably within its available resource boundaries, a configurable throttling mechanism limits throughput based on document length, accommodating the large-scale documents commonly encountered in enterprise environments, some spanning more than 1000 pages. Document uploads are handled asynchronously, upon submission, a processing job is immediately created, and a non-blocking response is returned to the caller, allowing progress to be monitored independently via the assigned job ID. When multiple documents are submitted concurrently, they are managed through a First In, First Out (FIFO) queue, ensuring orderly and predictable processing in resource-constrained deployments. Additionally, the pipeline supports a dedicated import mechanism, which allows pre-processed documents to be re-introduced into the system by supplying a previously generated JSON file and the original PDF document. This path bypasses the full processing pipeline entirely, committing the document and its enriched metadata directly to the knowledge base, enabling portability across environments and eliminating redundant reprocessing actions for the already enriched documents.

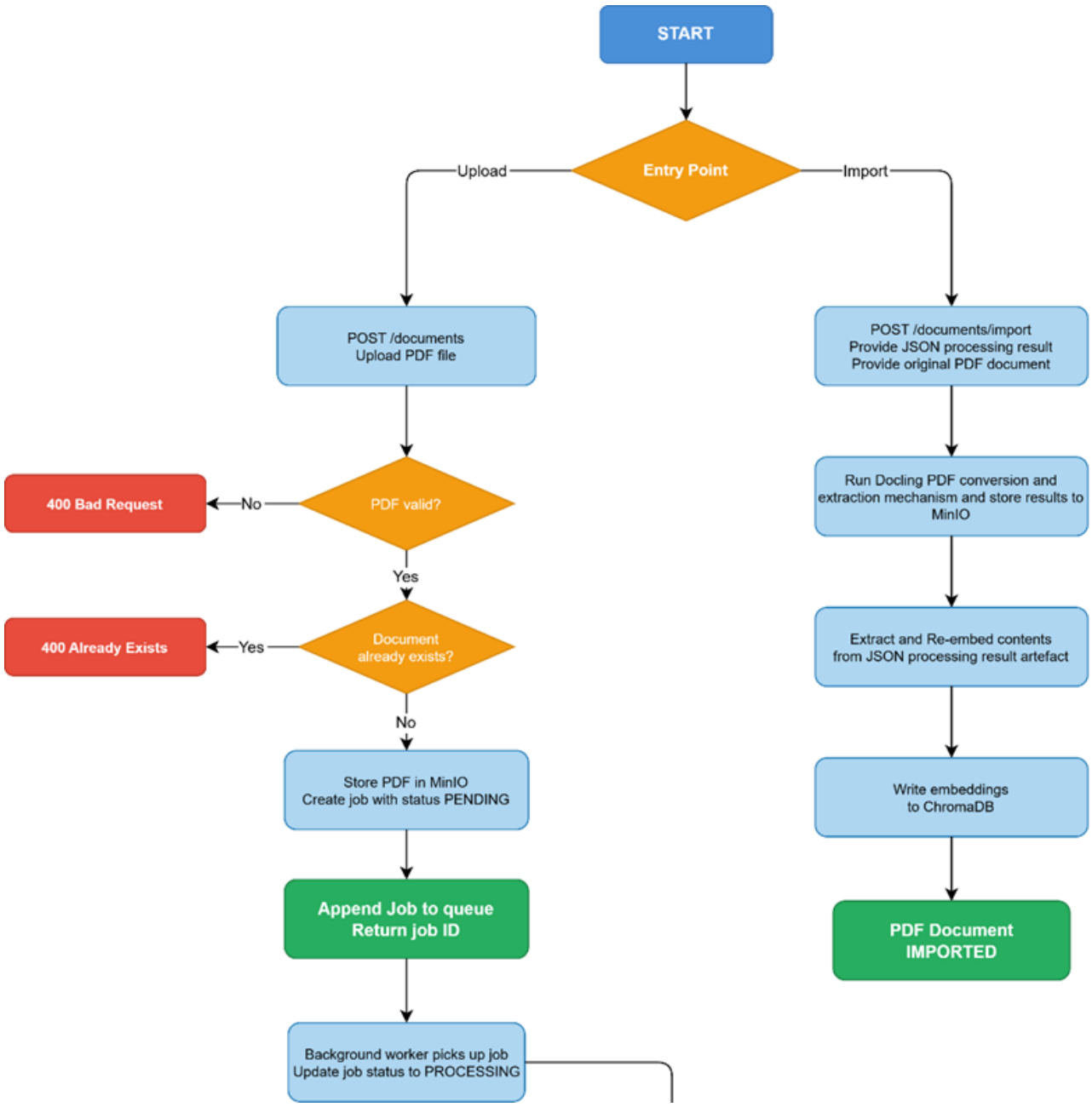

Figure 26 – Import Path, Job Submission and Queue Management

Once the background worker picks up the job, it transitions its status to PROCESSING. The PDF is then loaded and, if its length exceeds the configured system thresholds, split into smaller sub-documents to ensure processing remains within available resource limits, as illustrated in Figure 27. The pipeline then proceeds with the extraction of the document's structural and visual contents. Page images are rendered from the PDF and stored to MinIO, preserving the visual fidelity of the source document for later retrieval. The document is subsequently parsed by Docling, which performs a deep structural conversion, identifying and extracting all content elements, including text blocks, tables, figures, mathematical formulas, and code snippets. The output of this conversion is a structured, element-aware document representation, where each content element is classified by type and retained in its original structural context, serving as the direct input to the per-chunk enrichment stage described in the following section.

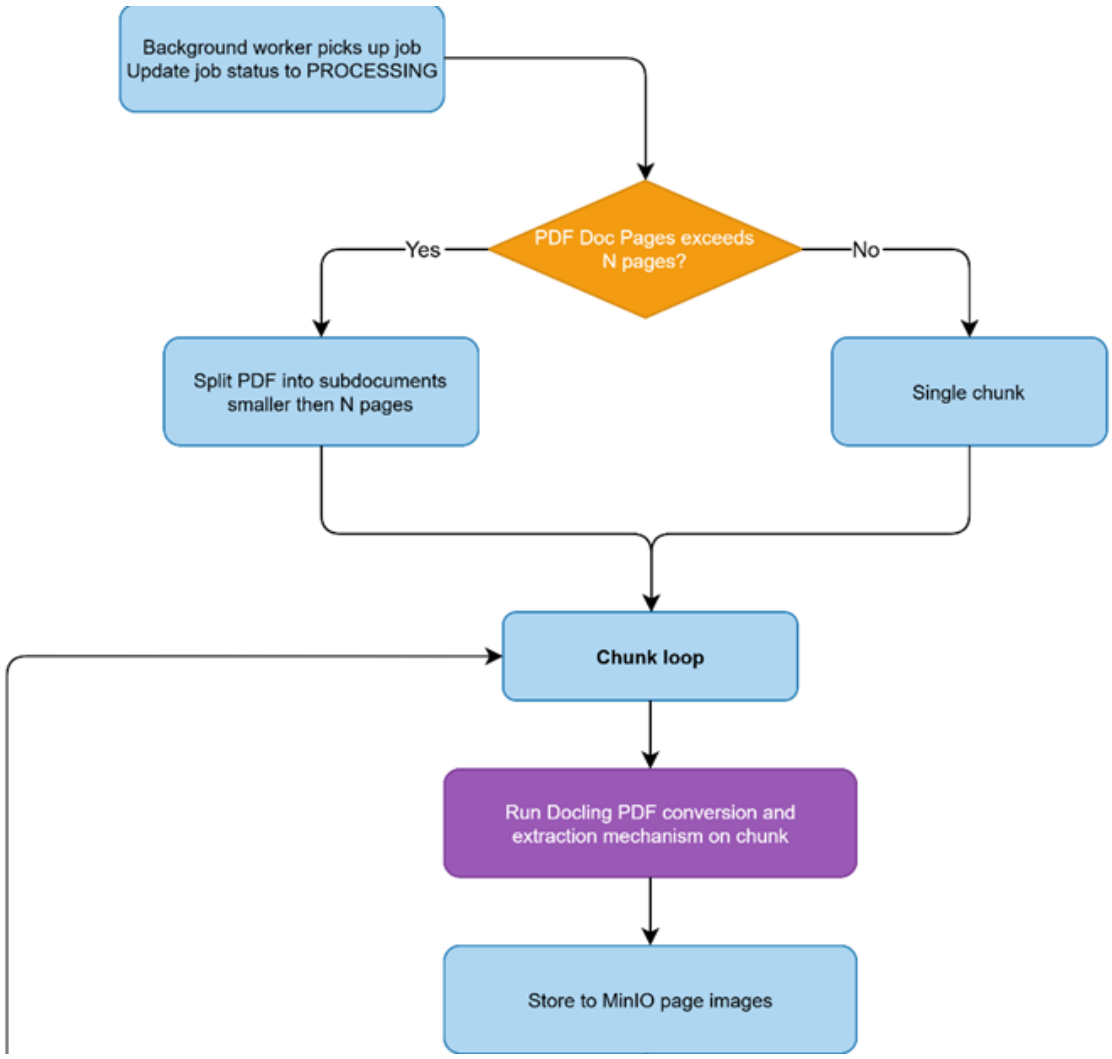

Figure 27 – Document Loading and Pages Extraction

Once the conversion completes, each chunk is subsequently iterated over and passed into the Per-Chunk Content Extraction block, where its individual content elements are processed according to their type through a series of specialised enrichment stages, as illustrated in Figure 28.

The first stage handles picture elements. If pictures have been extracted from the chunk, they are processed asynchronously through an LLM-Vision Language Model (VLM), which first classifies each image as either a meaningful content type, including technical schematics, circuit diagrams, graphs, and procedural flowcharts, or as boilerplate such as logos, decorative elements, or watermarks, in which case it is immediately discarded. For images that pass this relevance filter, the VLM generates a structured search index description capturing the figure title, visible components and labels, extracted part numbers and alphanumeric codes, and relevant technical keywords. Valid pictures then have their metadata formatted and stored to MinIO for later retrieval.

The second stage follows an analogous pattern for table elements. Extracted tables are first assessed by a dedicated classification model that distinguishes genuine technical tables from decorative or navigational elements such as revision histories and approval tables, which are immediately discarded. Tables that pass this filter are processed asynchronously through the LLM, which generates a structured search index capturing the table subject, technical parameters alongside their associated measurement units, and all specific model numbers and part identifiers present in the table rows. Tables subsequently identified as

irrelevant during LLM analysis are skipped, while valid tables have their metadata formatted and stored to MinIO.

The final stage addresses text chunks. Extracted text is split into appropriately sized chunks using a sentence and paragraph aware chunking strategy, with configurable minimum and maximum chunk size, preserving the full semantic integrity of each text unit. These chunks are processed asynchronously through the LLM, which enriches each chunk by resolving ambiguous references using the surrounding header context, generating hypothetical engineer queries the chunk answers, and extracting key data points including values, units, and tolerances. Text chunks identified as irrelevant, such as legal disclaimers, copyright notices, or table of contents entries, are discarded, while the remaining chunks have their metadata formatted and carried forward into the subsequent pipeline stage. Additionally, across all three per-chunk content extraction stages, a dedicated glossary extraction step is applied in parallel to each valid element, identifying domain specific and difficult to pronounce terminology to build a unified terminology index for the document, supporting speech communication and accessibility use cases.

Second, the Table of Contents (TOC) is extracted by identifying section titles and their corresponding page numbers from the accumulated headers and page breaks, preserving the hierarchical structure of the document as a navigational backbone that is carried forward across subsequent iterations. Once all chunks have been processed, the fully accumulated content metadata, enriched elements, document summary, and TOC are consolidated into the final document result JSON.

Once the document result JSON has been fully assembled, the pipeline enters the embedding loop, where each accumulated content element is processed sequentially, as illustrated in Figure 30. For every item, a vector embedding is generated from its enriched metadata and textual content. If the embedding is successfully produced, the item is written to ChromaDB alongside its associated metadata, making it immediately available for semantic retrieval. If the embedding generation fails for a given item, a warning is logged and the item is skipped, ensuring that a single failure does not interrupt the processing of the remaining elements. This loop continues until all items have been processed. Upon completion, the final document result JSON artefact, containing the fully enriched and structured representation of the document, is stored to MinIO as a permanent record of the ingestion output. Finally, the pipeline performs an exception check across the entire run. If no exceptions were raised during processing, the document status is marked as successfully processed. If any exception was encountered, the document status is marked as failed, ensuring that upstream systems and monitoring tools are accurately informed of the pipeline outcome. A configurable number of retry attempts is set in place for failed operations within the processing job, while recurring errors are accumulated and displayed in the processing job message. This logging mechanism ensures transparency and traceability for the collection administrator user, across the entire document processing event.

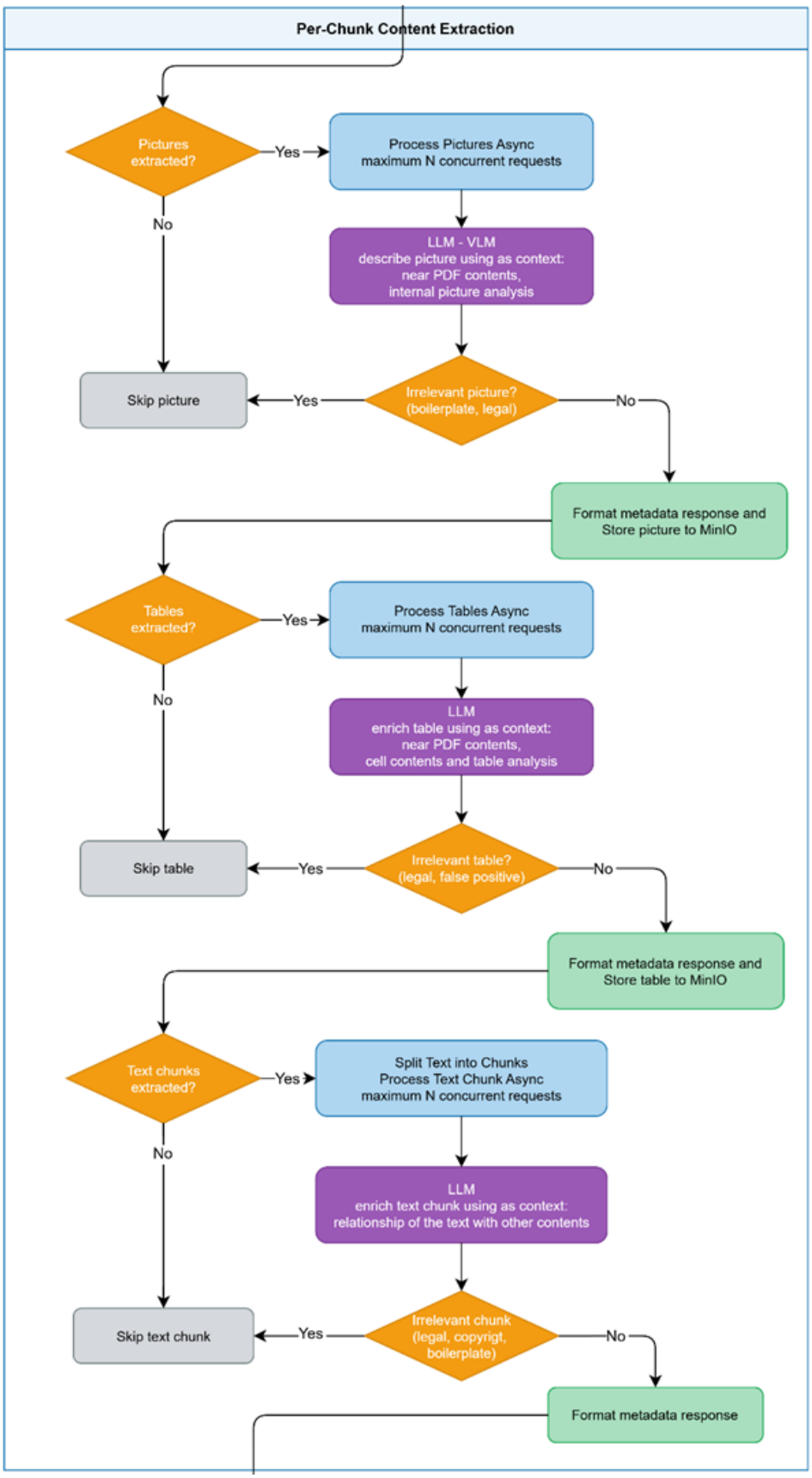

Figure 28 – Per-Chunk Content Extraction Pipeline Stages for Pictures, Tables and Text

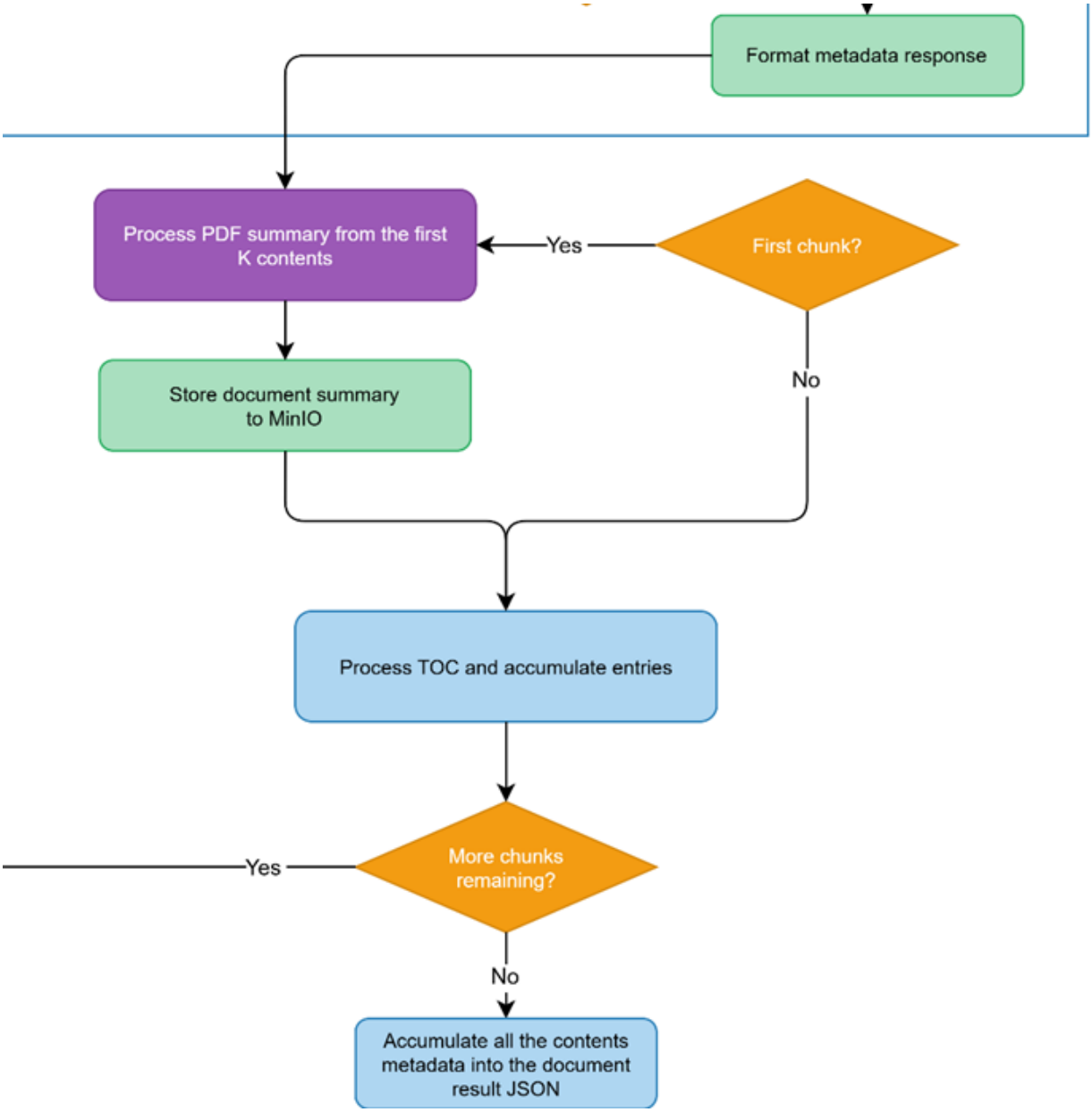

Figure 29 – Document Summary and Table of Contents Extraction Flow across Chunk Iterations

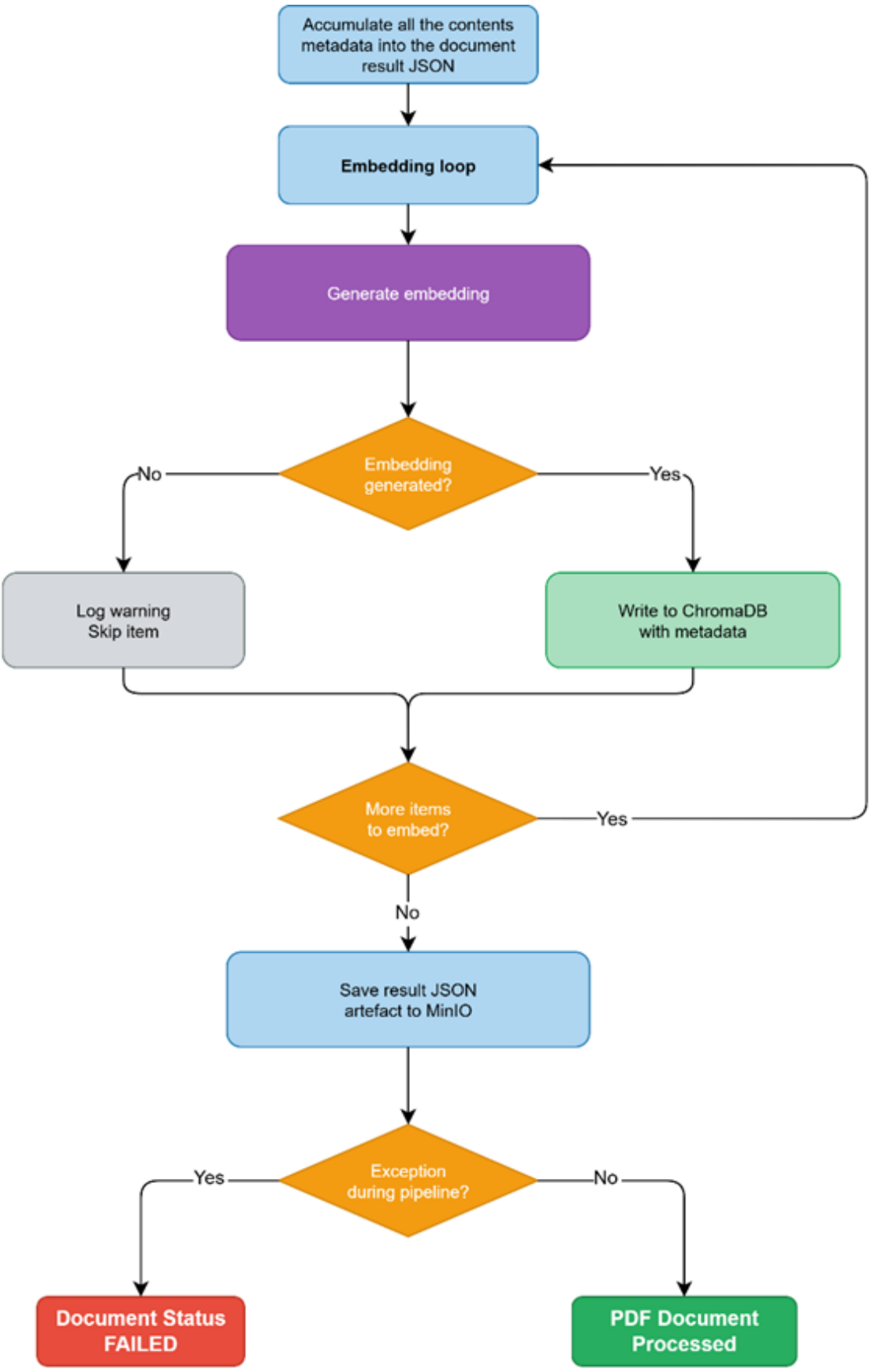

Figure 30 – Embedding Loop, Result Persistence, and Final Pipeline Status Resolution

DataLens Conversational AI Pipeline

The DataLens Conversational AI Pipeline implements a structured, decision-driven approach to answering user queries against a technical documentation knowledge base. Rather than routing all requests through a single generalised model call, the pipeline decomposes the task of query resolution into a sequence of specialised stages, each responsible for a distinct aspect of understanding, retrieval, and response generation. This design enables the system to handle a broad spectrum of interaction types, ranging from precise factual lookups and visual asset retrieval to broad summarisation requests and natural conversational exchanges, while maintaining grounded, traceable, and explainable outputs at every stage. In alignment with the requirements set forth by the EU AI Act, particularly those pertaining to transparency, human oversight, and traceability for high-risk AI systems under Articles 13 and 14, the pipeline's multi-tier reasoning and structured JSON outputs ensure that every response can be audited, attributed to its source context, and reviewed by a human operator. The pipeline accepts both text and voice audio as input, enabling integration across a wide range of use cases. Voice capabilities are provided through the external Siemens AI Voice Assistant service, which supplies text-to-speech (TTS) and speech-to-text (STT) functionality. A lightweight, configurable session mechanism preserves conversation history across interactions, with configurable parameters governing both the in-memory duration of a session and the depth of the conversation history retained, expressed as a number of turns. The complete workflow is illustrated across three figures, each covering a distinct phase of the pipeline: input handling and route classification, intent resolution and filter extraction, and knowledge base retrieval through to final response delivery.

Once an incoming request is received, the pipeline first verifies the input type, as illustrated in Figure 31. If the input is a voice audio signal, it is forwarded to the Siemens AI Voice Assistant service, which applies speech-to-text conversion to produce a text representation of the user query. Text inputs proceed directly. The pipeline then checks whether a session identifier was provided with the request. If no session exists, a new session is generated. If a session is found, the corresponding session state is loaded, providing the pipeline with the accumulated chat history, document context, and the last set of conversational suggestions offered to the user. With the session state established, the ConversationRouter LLM performs route classification, analysing the user input alongside the available session context to assign one of four route types:

- **NEW_QUERY** - indicates a new independent technical question, for which the pipeline proceeds to search the knowledge base directly.
- **FOLLOW_UP_QUESTION** - indicates that the user is elaborating on a prior topic, prompting the pipeline to retain the existing document context and search the knowledge base with the enriched query.
- **FOLLOW_UP_CONFIRMATION** - indicates that the user has confirmed a suggestion offered in the previous turn, causing the pipeline to retain the document context and reactivate the query using the last suggestion as the resolved input.
- **CHITCHAT** - indicates a non-technical conversational message such as a greeting or acknowledgement, for which a pre-generated response is returned directly, and the session history is updated without triggering any retrieval.

For all non-chitchat routes, the resolved query is forwarded to the next stage for expansion and filter extraction.

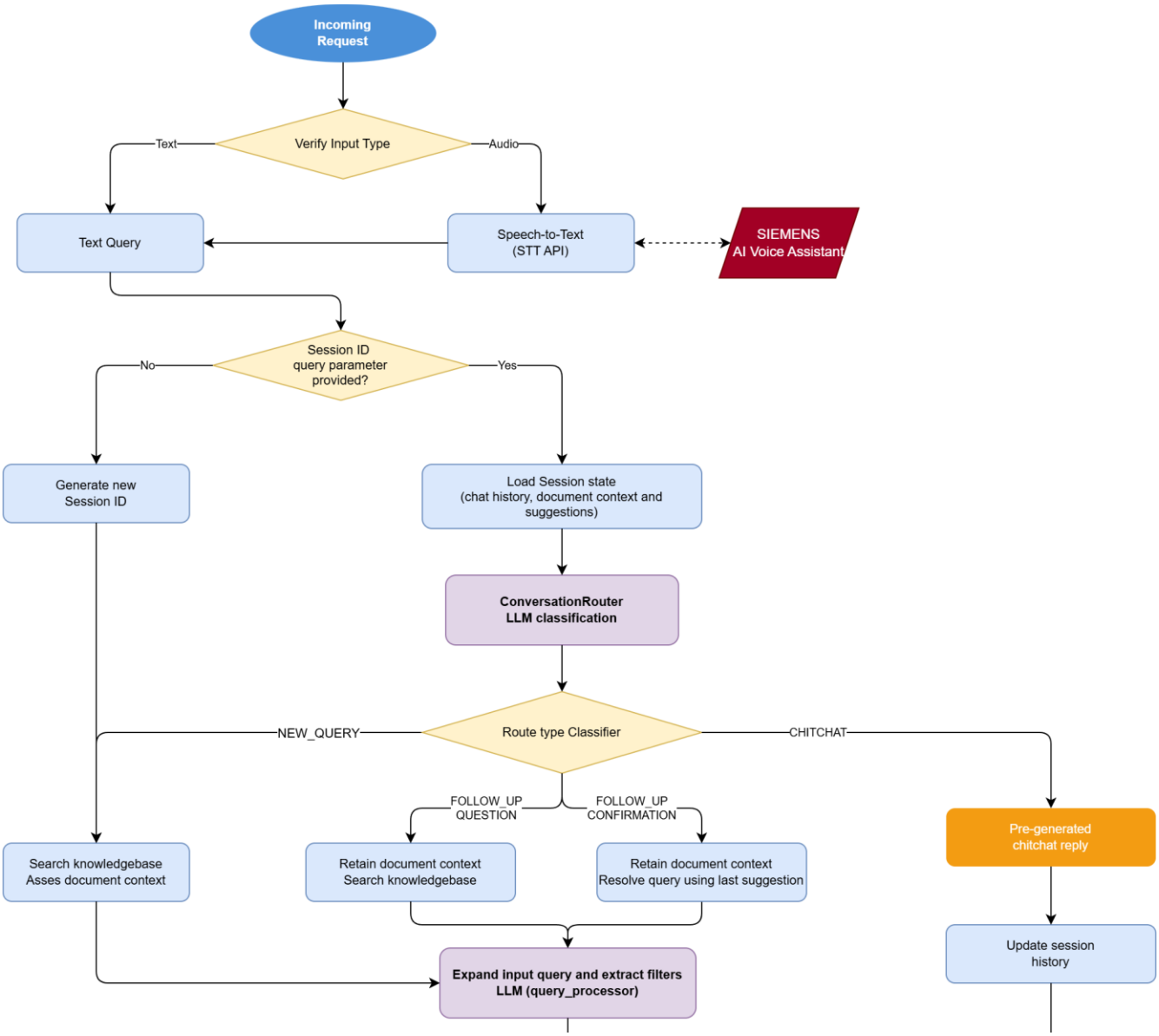

Figure 31 – Conversational AI Input Handling, Session Management, and Route Classification

Following route classification, the resolved query is passed to the query processor LLM, which performs two operations in a single step, as illustrated in Figure 32. First, it expands and rewrites the query into a search-optimised form, expanding acronyms, removing conversational filler, and enriching the query with semantically relevant terminology to improve vector retrieval accuracy. Second, it extracts structured filters from the query, identifying any explicit references to a page number, figure or table identifier, or document filename. If a fuzzy filename reference is detected, the pipeline invokes the Fuzzy Filename Resolution LLM, which compares the extracted reference against the document index and attempts to identify the exact matching filename. If a confirmed match is found, the document context is appended to the metadata filters, scoping all subsequent retrieval to that specific document. If no match can be confirmed, the filter is discarded and retrieval proceeds across the entire collection without document scoping. In parallel, the expanded query is passed to the Query Intent Classifier, which assigns one of seven

intent types, each configuring a dedicated system prompt and retrieval depth for the response generation stage:

- **GENERAL_QA** - targets narrow factual lookups such as specifications, limits, or single-step procedures.
- **SUMMARIZATION** - targets broad overviews requiring synthesis across multiple aspects of a topic or device.
- **IMAGE_LOOKUP** - targets diagrams, schematics, or photographs.
- **TABLE_LOOKUP** - targets tabular data and structured numerical content.
- **SPECIFIC_PAGE** - targets content from an explicitly referenced page.
- **TABLE_OF_CONTENTS** - retrieves the document navigation structure.
- **NA** - indicates that the query falls outside the scope of the knowledge base, for which a pre-generated out-of-scope response is returned directly without retrieval.

For all intent types except NA, the classified intent and expanded query are carried forward to the retrieval and response generation stage.

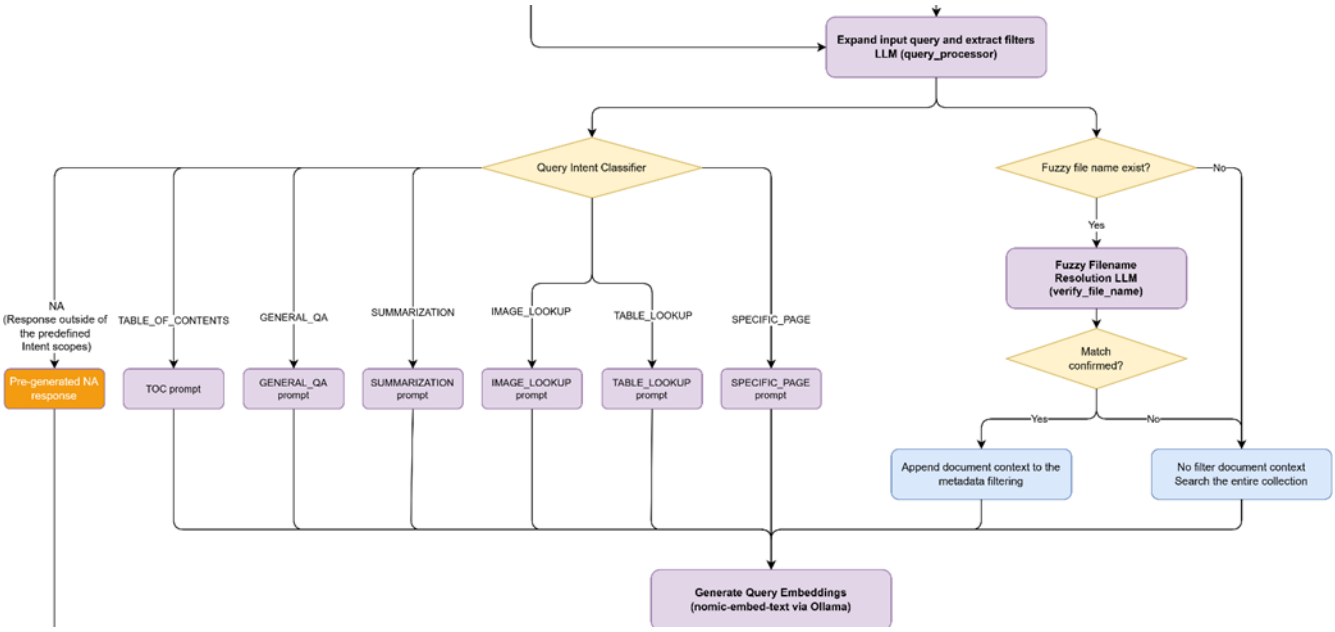

Figure 32 – Conversational AI Query Expansion, Filter Extraction, and Intent Classification

With the expanded query, resolved intent, and metadata filters established, the pipeline generates a vector embedding of the query using the configured embedding model, as illustrated in Figure 33. This embedding is used to perform a cosine similarity search against ChromaDB, retrieving the top-k most semantically relevant content chunks, where k is configured per intent to reflect the retrieval depth appropriate for each response type. If the vector search returns no results, the pipeline returns a no relevant context found response immediately. If results are found, they are assembled alongside the intent-specific system prompt and the last k turns of the conversation history, and submitted to the Response Generation LLM to produce a multi-tier structured response, composed of the following elements:

- **Speech Summary** - plain text formatted for text-to-speech delivery, providing a concise conversational answer.
- **Markdown Response** - a detailed structured answer using headers and bullets, intended for visual rendering.

- **Follow-up Suggestions** - an optional set of conversational prompts to guide the next interaction turn.
- **Referenced Content Identifiers** - a list of source identifiers providing full traceability back to the retrieved knowledge base content.
- **Image Paths** - included where the query requested visual assets such as diagrams or photographs.

Once the LLM response is validated and successfully parsed, it is formatted and enriched with any associated image paths. TTS processing is then evaluated:

- If enabled, the speech summary is forwarded to the Siemens AI Voice Assistant service, which generates an audio stream that is attached to the response payload in a streamable format alongside any images.
- Otherwise, the speech field is set to null. In the event that the LLM response fails to parse correctly, an error response is returned at this stage. Finally, the session state is updated with the current chat turn, the active document context, and the latest follow-up suggestions, and the fully assembled AskResponse is returned to the API caller.

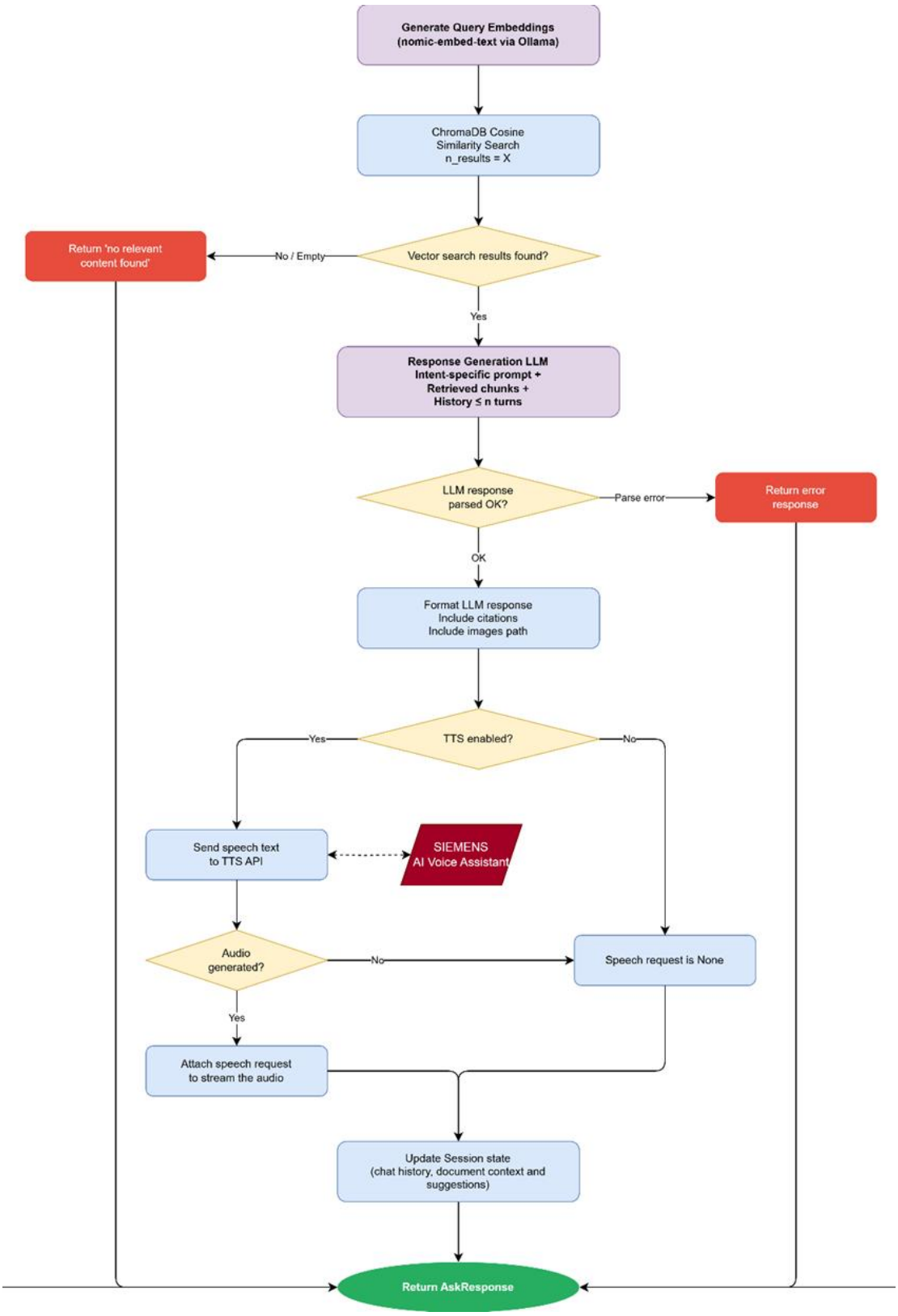

Figure 33 – Conversational AI Knowledge Base Retrieval, Response Generation, and Output Delivery

3.3.3 Implementation

The DataLens conversational AI pipeline is implemented in Python, structured according to a clean architecture pattern that separates concerns across four distinct layers: routers, services, schemas, and utilities. The FastAPI framework serves as the web layer, exposing the pipeline through a versioned REST API under the */api/v1* prefix, with all core pipeline logic encapsulated within the service layer, which orchestrates the stages described in the preceding design sections. The relationship between the 4 layers is highlighted in Figure 34.

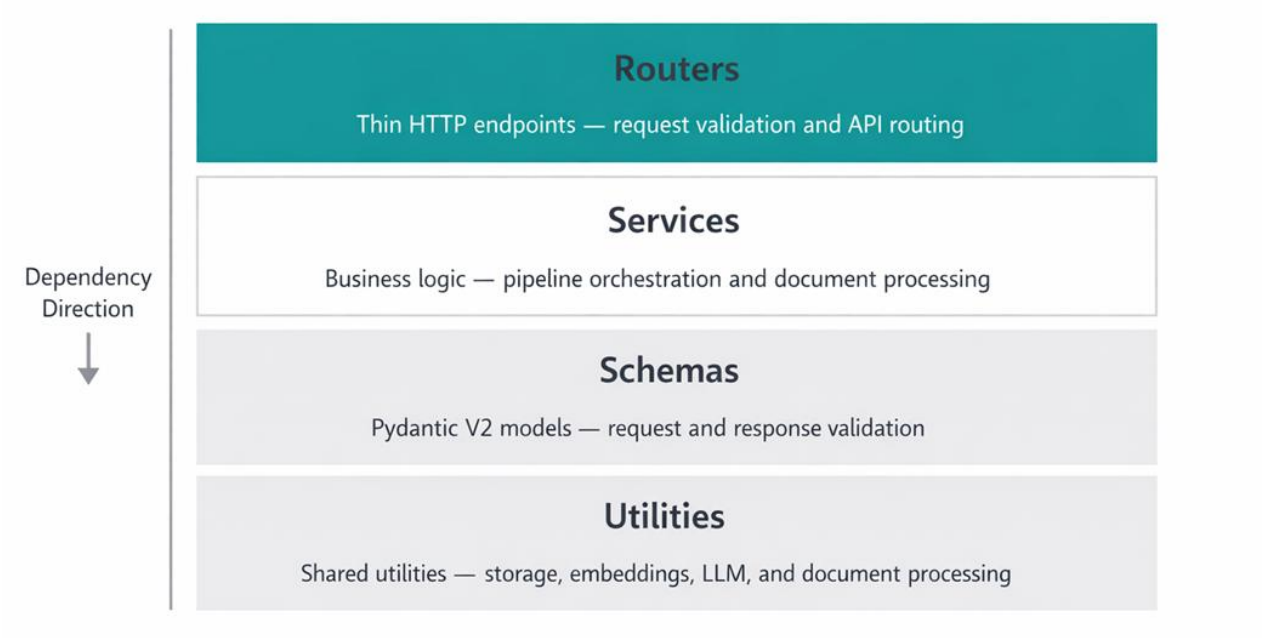

Figure 34 – Layer Architecture Diagram Illustrating the Dependency Direction from the HTTP Routing Layer down to Shared Utilities

The implementation relies on the following core technologies:

- FastAPI - provides an asynchronous HTTP layer, request validation, and API routing.
- ChromaDB - serves as the local vector database, storing and querying document embeddings via cosine similarity search.
- Ollama - provides the local LLM runtime, hosting both the language models used across the pipeline and the embedding model used for vector generation.
- Docling - handles PDF ingestion, extracting text, tables, and images from uploaded documents into a structured JSON artefact.
- MinIO - provides S3-compatible object storage for raw PDF files, processed artefacts, and extracted visual assets.

The project follows a modular directory structure, with the router layer containing thin endpoint definitions that delegate all business logic to the service layer. Supporting utilities for storage, embedding operations, and document processing are maintained separately, keeping the service layer focused and testable. LLM system prompts are externalised into a dedicated prompts directory, decoupling prompt engineering from application logic and allowing prompts to be revised without modifying source code. Pipeline behaviour is further governed by a central configuration file, which defines the embedding model, LLM model identifiers, retrieval depth values per intent type, and TTS integration settings, allowing the pipeline to be adapted to different deployment environments or model choices without requiring code changes.

Document ingestion, embedding generation, and the conversational pipeline itself follow the flows described in detail in the preceding architecture sections. Of note from an implementation perspective, all LLM nodes in the pipeline use structured output where applicable, ensuring that classification results, extracted filters, and expanded queries are returned in a predictable, parseable format rather than as free-form text, which significantly improves pipeline reliability. Session state is maintained in memory across conversation turns, storing the chat history, the active document context, and the latest follow-up suggestions, enabling conversational coherence across multiple exchanges without requiring a persistent database for session data.

A central aspect of the implementation is the structured JSON response schema returned by the conversational pipeline endpoint, which has been deliberately designed to serve the requirements of XR applications operating in immersive, hands-free environments. The schema is presented in the Code 1:

```
{
  "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

The pipeline integrates with the Siemens AI Voice Assistant service for TTS output. When TTS is enabled and a valid speech summary is present in the generated response, the summary is forwarded to the external service, which returns a synthesised audio stream made available for streaming retrieval via a dedicated audio endpoint. Speech-to-text input follows the reverse path, with the transcribed text entering the pipeline at the input handling stage as a standard text query.

Finally, the infrastructure supporting the application is progressively being containerised, and the results will be available for deployment through Docker containers, allowing for portability across environments and simplifying the setup of dependent services such as MinIO and ChromaDB.

3.3.4 Demonstration Scenarios and Mapping to Pilots

The DataLens platform was demonstrated across two distinct pilot contexts, each representing a different industrial domain and user base. Rather than serving as a generic document retrieval tool, DataLens was instantiated in each pilot as a tailored knowledge assistant, adapted to the specific operational requirements, security constraints, and training objectives of the respective partner. The following subsections describe the demonstration scenarios and their mapping to Pilot 1 (KUKA) and Pilot 4 (TAP).

Pilot 1 (KUKA) – Rapid Human Centric AI-Enabled Product Design

Within the context of KUKA's Use Case 1.2 (*Virtual Commissioning and Generative AI in Robotics*) DataLens was deployed as KUKABot, a RAG-powered conversational knowledge assistant built on top of a dedicated knowledge base collection instance, serving KUKA's internal technical manuals and engineering documentation. KUKABot provides real-time access to KUKA technical documentation through natural language queries within an XR application, leveraging large language models to retrieve relevant documents from its vector database. The system processes user queries by pre-processing the input, retrieving semantically relevant resources using vector similarity search, and generating structured responses that include formatted answers, audio descriptions via text-to-speech, detailed Markdown content, optional reference images, and source metadata.

The demonstration scenario centred on the processing and querying of a knowledge base covering robotic systems and commissioning procedures. The knowledge base comprised five technical documents provided by KUKA, spanning equipment datasheets, operating manuals, and electrical design schematics. Users interacted with the ingested content through a natural language conversational interface, posing queries for component specifications and receiving contextually accurate responses grounded in the uploaded documentation. A total of 40 interactions were recorded across the full range of supported query capabilities, encompassing summarisation, question and answer, image lookup, table lookup, and page lookup. The outcomes of these interactions are evaluated in detail in Section 3.3.6.

Pilot 4 (TAP)– Personalised Aircraft Maintenance Training

Within the context of Pilot 4, DataLens was deployed in conjunction with the XR-Enabled AI Assistant, a comprehensive enterprise-grade utility platform providing speech-to-text, text-to-speech, and conversational AI capabilities through a unified RESTful API. Together, the two components form an integrated voice-driven RAG pipeline: the XR-Enabled AI Assistant handles the full audio processing layer by capturing user voice input, transcribing it via speech-to-text, and synthesising the final response back into audio via text-to-speech, while DataLens serves as the document intelligence backbone, retrieving contextually relevant information from the ingested knowledge base and grounding the conversational AI responses in verified source documentation. The combined solution operates as a middleware layer, abstracting the complexity of the underlying AI models while delivering scalable, GPU-accelerated voice processing capabilities tailored for XR applications.

The demonstration scenario was conducted in the context of aircraft maintenance training, specifically targeting Aircraft Maintenance Technicians involved in procedures related to the Wing Anti-Ice Valve. Technical documentation describing the WAIV installation and maintenance procedure was ingested into a TAP-specific DataLens knowledge base. Users interacted with the system through voice and gesture, submitting natural language queries such as:

"What are the steps involved in the WAIV installation?"

The XR-Enabled AI Assistant processed the voice input via the speech-to-text pipeline, forwarded the transcribed query to the DataLens conversational retrieval engine, and returned a contextually grounded response — both in text and synthesised audio — derived from the ingested technical documentation.

In alignment with the broader XR5.0 Training Programme, the TAP pilot demonstrates how DataLens enables the automatic conversion and selection of document content through natural language queries. Rather than requiring technicians to manually navigate dense technical manuals, the system surfaces the most relevant procedural information on demand, which is then rendered within XR environments as immersive, step-by-step training experiences. This approach supports both on-the-job and off-the-job training contexts, allowing Aircraft Maintenance Technicians to rehearse complex maintenance procedures in controlled XR simulations before applying them in real-world scenarios. The outcomes of the demonstration and the associated user evaluation are presented in detail in Section 3.3.6.

3.3.5 Challenges and Limitations

Despite the promising results demonstrated across the DataLens pilot deployments, several challenges and limitations were identified that warrant consideration for future iterations and broader adoption of the solution.

A primary challenge concerns the performance gap between open-source and closed-source language models. While DataLens is architected around open-source LLMs to preserve data privacy and enable deployment on local private cloud instances, the current open-source ecosystem remains incrementally behind the state-of-the-art capabilities offered by closed-source commercial alternatives. That said, the open-source community is experiencing an unprecedented surge in development activity, with increasingly capable models being released at a rapid pace. This trajectory suggests that the performance gap is narrowing, and that specialised task execution on smaller, locally deployable models will increasingly deliver results comparable to their closed-source counterparts.

This directly ties into the second challenge: infrastructure requirements and hardware overhead. Deploying and operating LLM-based systems locally demands significant computational resources, particularly GPU support. The hardware investment required to sustain performant inference, especially for resource-intensive tasks such as summarisation or multi-turn conversational retrieval, represents a meaningful barrier for organisations without dedicated AI infrastructure.

A further operational challenge relates to the maintenance burden associated with a rapidly evolving dependency landscape. The AI and LLM ecosystem have undergone a fundamental shift in release cadence, moving from traditional monthly or annual software update cycles to weekly and, in some cases, daily releases. Keeping all system components, including model versions, inference frameworks, vector database libraries, and API integrations, aligned and mutually compatible requires continuous engineering effort and introduces the risk of regressions or breaking changes between updates.

XR integration complexity and latency represent an additional layer of challenge specific to the deployment contexts explored within the XR5.0 programme. Delivering responsive AI-assisted experiences within XR environments requires the resolution of a multitude of interdependent pipeline steps, spanning document ingestion and preprocessing, vector embedding, semantic retrieval, response generation, and, where applicable, text-to-speech synthesis. Each step introduces latency, and the cumulative effect across the full pipeline can meaningfully impact the perceived responsiveness of the system, which is particularly critical in immersive XR contexts where interaction fluidity is essential to user experience.

Finally, accuracy and response consistency remain an ongoing concern in long-running conversational sessions. LLM-based systems can exhibit behavioural drift over extended interactions, whereby the model's response style, tone, and factual grounding become inconsistent relative to earlier exchanges within the same session. This phenomenon can manifest as hallucinations, where the model generates plausible but factually incorrect responses, or as personality shifts that undermine user trust and reduce the reliability of the system as an authoritative source of technical information. Mitigating these effects requires robust prompt engineering, session management strategies, and, where appropriate, validation layers to ensure response quality is maintained throughout the interaction lifecycle.

3.3.6 Evaluation

The DataLens framework was evaluated across both pilot deployments through a combination of interaction logging and structured user feedback collection as reported in D6.5 – *Socio-technical Evaluation of XR5.0 Systems v1*. In Pilot 1 (KUKA), a total of 40 interactions were recorded and analysed across the full range of supported query capabilities, namely summarisation, question and answer, image lookup, table lookup, and page lookup, yielding mixed results. Retrieval performance was strongest for queries containing explicit technical terminology, where the query expansion and vector similarity search mechanisms consistently surfaced relevant knowledge base entries. However, four critical issues were identified that materially impacted overall pipeline performance: aggressive silence detection thresholds causing premature query submission, transcription inaccuracies captured from unnatural speech patterns adopted by users unfamiliar with voice-driven interfaces, extended response times attributable to hardware resource constraints and dynamic model loading overhead, and instances of LLM hallucination when user queries fell outside the scope of the ingested knowledge base. In Pilot 4 (TAP), evaluation was conducted through a structured questionnaire for pilot end-users (as discussed in detail in D6.5) combining multiple-choice and open-ended items, supplemented by verbal feedback collected during live interaction with the system. The system accurately retrieved and presented the sequential steps of the WAIV installation procedure in response to natural language queries, and the response was independently validated by a senior Aircraft Maintenance Technician as correct and clearly communicated.

Across both pilots, the evaluation confirmed the technical viability of DataLens as a privacy-preserving, on-premise conversational AI solution for industrial document querying, while surfacing a concrete set of limitations that inform the development roadmap. The document processing pipeline demonstrated reliable performance on well-formed, born-digital technical documentation, and the structured multimodal response format combining synthesised speech, Markdown-formatted content, and source references was well received across both operational contexts. The critical issues identified in Pilot 1 provide an actionable basis for immediate improvement, with near-term priorities targeting silence detection calibration, query correction mechanisms, and dedicated hardware provisioning, and longer-term efforts directed towards multi-layer response validation, enhanced conversational coherence, and the integration of eXplainable AI principles into the response generation pipeline.

3.3.7 Discussion and Lessons Learned

The two pilot deployments of DataLens demonstrated the practical feasibility of deploying a privacy-preserving, on-premise conversational AI framework within real industrial and operational environments, validating the core architectural decisions underpinning the platform, namely the modular document processing pipeline, the RAG-based retrieval engine, and the multimodal response generation layer. Both pilots confirmed that the system is capable of delivering accurate, contextually grounded responses to natural language queries over complex technical documentation, with the TAP deployment in particular illustrating the value of voice and gesture-driven document access in safety-critical training contexts where hands-free interaction represents a meaningful operational advantage.

At the same time, the Pilot 1 findings highlighted a set of challenges that extended beyond configuration and hardware provisioning, ultimately concluding in the need for a more sophisticated RAG architecture capable of handling the full complexity of real-world industrial queries. This insight directly informed the evolution of the DataLens conversational pipeline, which in its current version incorporates a Conversation Route Classifier, a User Intent Classifier, a query expansion module, and a set of additional filters supporting the resolution of figure and table references, page number lookups, and fuzzy file name matching. Taken together, the lessons learned across both pilots reinforce the importance of grounding architectural decisions in real operational feedback, ensuring that the system evolves in close alignment with the needs, constraints, and expectations of the industrial environments it is intended to serve.

4. AI MODELS FOR TRUSTED HUMAN-AI COLLABORATION

4.1 Post-hoc XAI Models

eXplainable Artificial Intelligence (XAI) techniques are able to enhance industrial visual inspection systems by making their AI components transparent and interpretable. Modern AI and Machine Learning solutions are composed of thousands if not millions of parameters that affect the final classifications and predictions. Although powerful, these systems often lack interpretability, making it challenging for operators to understand their reasoning, especially when errors occur. XAI solutions address this limitation by providing clear and interpretable insights into how decisions are made. The primary aim of incorporating XAI into visual inspection systems is to reduce the cost and time associated with manual inspections while allowing workers to focus on more meaningful, less repetitive tasks. The generated explanations increase human confidence in AI-driven decisions and empower workers and operators to handle complex inspection tasks that rely on human evaluation. The Post-hoc XAI suite developed by project partner UPRC provides explanations in terms of attribution scores and heatmaps, offering insights into the importance of each individual input feature for a specific classification or prediction.

4.1.1 Role and Functionality

The XAI suite for XR5.0 represents a significant advancement in human-centric explainability for black-box AI models, providing a multi-layered approach to interpretability. It is implemented as a stand-alone module that can be integrated in multiple use cases and provides a separate API that has been validated in controlled environments of previous EU funded projects. For the purposes of the XR5.0 project the Post-hoc XAI suite is composed of two individual XAI methods which were selected to match the use cases and requirements of the project.

- Local Interpretable Model-agnostic Explanations (LIME) is part of the XAI suite since it is able to support multiple data modalities, including vectors, images, and text. It is model-agnostic and interacts directly with the model's predict() method. Considering the amount of data that was processed and its lightweight nature for standard use, LIME proved to be robust and agile in providing relevant explanations.
- Gradient-weighted Class Activation Mapping (Grad-CAM) is also part of the XAI suite since it is specifically designed for Convolutional Neural Networks (CNN). By leveraging gradients flowing through the network, Grad-CAM is able to generate visual explanations to *"peer through the black-box"* and enhance human trust and understanding. Furthermore, considering that many use cases involved around the area of Computer Vision in image classification or object detection tasks, Grad-CAM provided useful insights since it highlights image regions that contribute most to the model's predictions

4.1.2 System Architecture

The Post-hoc XAI suite is composed of four main components:

- User interface.
- Data injections and storage.
- Model training.
- Inference service.

All the components have been implemented following a microservices approach that communicates using a REST API to ensure interoperable communication and serve as a data-agnostic platform. The use of containerization encapsulates dependencies and runtime configurations, enabling consistent deployment, efficient scalability, and straightforward updates. The original user interface has been developed using Streamlit, but for the purposes of the XR5.0 project, the XAI suite explanations can be served in Immersive Environments (XR Apps) in an ad-hoc manner. The data ingestion and storage service, implemented as a Flask microservice, manages dataset validation, cleaning, and secure storage, ensuring maintainability and

scalability. The model training service, also Flask-based, handles resource-intensive training processes in an isolated environment, enabling experimentation and updates without impacting other system components. The inference service provides a scalable mechanism for generating predictions via REST endpoints, with the flexibility to adapt to increased demands or enhancements independently.

Due to the nature of the identified use case's constraints and requirements, all XAI suite explanations are provided to the end users through the Immersive Environments (XR app) in an ad-hoc setting. Due to the nature of the XR apps and the identified use cases, an ad-hoc setting for the explanations was preferred in order to avoid overwhelming operators with multiple types of information (classifications, predictions and explanations in one Immersive Environment) at the same time, thus leading to confusion. Therefore, as presented in Figure 35, the end user has the ability to request for additional explanations/clarifications of a specific AI prediction. Depending on the nature of the use case (underlying dataset) the system identifies which XAI method (LIME or Grad-CAM) is more suitable to generate additional explanations before they are provided within the Immersive Environment.

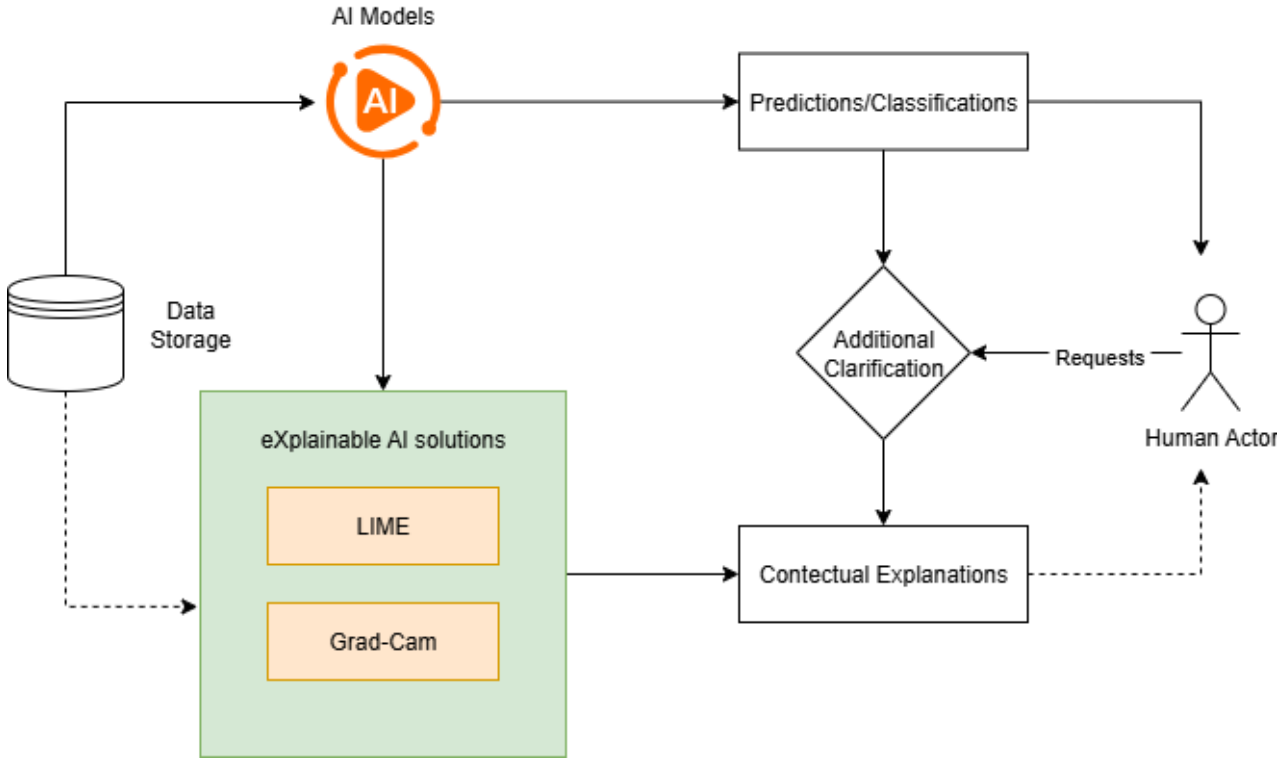

Figure 35 – High-level Architecture of Contextual Explanations through Post-hoc XAI Suite

4.1.3 Implementation

The Post-hoc XAI suite has been implemented in python and is containerised via Docker Compose, with Kubernetes manifests provided for production environments. The Open CV library is exploited for the image processing and Matplotlib is being utilised for the generation of visualisation outputs. Although an interface is also provided using Streamlit, the explanations are also served within the Immersive Environments (XR App). The data ingestion and storage as well as the model training are Flask-based services which are exposed through a REST API. Table 13 presents the different API endpoints that are publicly exposed for the utilization of the Post-hoc XAI suite.

Table 13 – API Specification and Public Endpoints

Method	Endpoint	Auth	Input	Output	Description
GET	*/health*	None	request	status	Returns service health status and model loading state.
POST	*/gradcam/{image_path}*	Auth token	AI prediction output	"image":"base64_encoded_image_string"	GradCAM heatmap visualization as base64 image. Visual heatmaps showing important regions.
POST	*/ lime / {image_path}*	Auth token	AI prediction output.	"Image":"base64_encoded_image_string"	Returns LIME segment-based explanation as base64 image.

4.1.4 Demonstration Scenarios and Mapping to Pilots

Pilot 3 (EKSO)– XR Steaming for Operator 5.0 Training

The Post-hoc XAI suite is currently being integrated in the demonstration scenario of Pilot 3 which prioritizes on predictive maintenance for smart water pipes. For the purposes of Pilot 3 the Post-hoc XAI suite serves as an add-on of the Neurosymbolic AI (NSAI) component to provide additional explanations should they be requested by the end users (operators). The XAI suite processes video data converted into frame-by-frame images to provide explanations on the anomalies that have been identified by the NSAI component in smart water pipes CCTV inspection videos. Due to the nature of the use case the exploitation of Grad-CAM has been prioritized to provide pixel by pixel analysis and generate heatmaps that highlight the most important regions of a frame. As stated above, in order to avoid overwhelming and confusing the end users with multiple visualisations, the generation of the explanations from the post-XAI suite occur in an ad-hoc manner. Therefore, the operator is allowed to request the explanation for a particular model output and receive the corresponding explanations.

Furthermore, the Post-hoc XAI suite is currently being evaluated on its additional value for the purposes of Pilot 4, which focuses on aircraft maintenance training and aircraft engine part identification using AI models.

4.1.5 Challenges and Limitations

Data availability and quality has been a significant challenge in terms of obtaining sufficient amounts of data that are also representative for the training and evaluation of the solution. Although Pilot 3 owner has been very active to provide additional datasets upon request, exploiting the maximum information out of these data has been a considerable bottleneck. Most of the data provided were unlabeled videos which had to be processed and carefully evaluated using cross reference reports that were generated by the Pilot owner. Furthermore, the proper designing and integration with the eXtended Reality application, was another major challenge faced. The risk of overwhelming the end users (operators) with multiple types of visualization that refer to different aspects of the XR app, was seriously considered and addressed. To that

end, the ad-hoc usage of the component was preferred to ensure user understanding and adaptability of the system.

4.1.6 Evaluation

The Post-hoc XAI suite was not evaluated during the first user evaluation period for the purposes of Pilot 3, since it was not yet integrated to the Immersive Environment (XR app). Due to modifications of the XR App and its migration from an XR device to handheld devices to better serve the Pilot owner's needs, the XAI suite was not part of the first user evaluation period. During the second user evaluation period the XAI suite will be evaluated using quantitative methods (e.g. user feedback and questionnaires) but also using its rigorous evaluation framework which contains quantitative metrics (e.g. accuracy, concept alignment, etc.).

4.1.7 Discussion and Lessons Learned

eXplainable Artificial Intelligence methods provide powerful tools to further enhance human knowledge and understanding of complex AI solutions and their decision making processes. Post-hoc XAI methods allow human experts to *"peer through the black-box"* and provide additional information regarding AI's complex decision making processes, thus enhancing human trust and allowing human operators to handle multiple tasks at the same time resulting in productivity increase. System interoperability is of paramount importance since the seamless flow of data and information between different components, ensures system reliability and timely predictions and explanations.

However, integrating XAI solutions in Immersive Environments is not a trivial task. Overwhelming end users (operators) with distinct types of information within the XR app creates the risk of user disengagement. Careful system design is required from the perspective of the XR app as well as the XAI solution, to provide context aware responses that are relevant to specific tasks and at the same time easy to interpret within the Immersive Environment in order to enhance human cognition and understanding.

4.2 AI Glass-box Models for Computer Vision

In many computer vision systems, deep neural networks operate as black-box models, producing predictions without revealing the reasoning behind them. While such models can achieve high performance, their lack of transparency makes them difficult to audit, validate, or correct. In safety-critical domains such as infrastructure inspection, this lack of interpretability can limit operational trust.

Glass-box models address this limitation by making the decision process fully interpretable. In a glass-box model, the internal reasoning process is explicitly visible and understandable to human users. Each decision can be traced to a set of human-interpretable features or concepts, allowing experts to understand why the model produced a particular prediction.

4.2.1 Role and Functionality

Within the proposed system, the glass-box model corresponds to the symbolic layer of the Concept Bottleneck Models architecture. For example, a high activation of concepts such as surface deformation and structural irregularity may increase the probability of the Crack class, while discoloration and texture degradation may increase the probability of Corrosion. Because each contribution is explicitly represented, domain experts can inspect the reasoning process, identify potential errors, and intervene by modifying concept values. This capability supports human oversight, making the model suitable for environments where transparency and accountability are required.

4.2.2 System Architecture

The glass-box reasoning component operates as the second stage of the Neurosymbolic inspection pipeline. The architecture separates perception from reasoning, allowing each component to specialize in a different task. The pipeline consists of three main stages:

1. **Visual Perception Layer:** The first stage processes raw CCTV frames using a neural vision backbone. This component extracts high-level visual representations from the image.
2. **Concept Representation Layer:** The extracted visual features are mapped to a predefined set of semantic concepts. These concepts represent interpretable visual characteristics relevant to infrastructure inspection, such as surface texture irregularity, structural deformation, discolouration, or moisture patterns.
3. **Glass-Box Reasoning Layer:** The concept vector is passed to a transparent classifier that performs the final prediction. This classifier uses a sparse linear model where each concept has an associated weight for each inspection category.

The final output of the system includes:

- A predicted inspection class {Normal, Crack, Corrosion, Leakage}.
- A concept activation vector representing the detected visual attributes.
- A reasoning trace showing how each concept contributed to the final decision.

Because the reasoning layer operates directly on interpretable concepts, the decision path from input to output remains fully transparent. This separation of perception and reasoning is a defining characteristic of **Neurosymbolic AI**, enabling both strong visual recognition and interpretable decision-making.

4.2.3 Implementation

The reasoning layer is implemented as part of the overall Neurosymbolic system, where the symbolic component is responsible for producing the final predictions based on the detected concepts. In this architecture, the model does not rely solely on raw visual features but instead performs classification using an intermediate set of human-interpretable concepts.

The training process is therefore structured around a predefined set of concepts that reflect the requirements of the inspection task and the types of visual patterns the system is expected to recognize. These concepts act as the semantic bridge between the neural perception component and the symbolic reasoning layer.

To ensure that the selected concepts are meaningful and relevant to the problem domain, domain expert knowledge is incorporated into the concept definition process. Experts identify the visual characteristics they would normally examine when evaluating an image and determining whether an anomaly is present. These expert-defined cues are then formalized as model concepts and integrated into the training procedure.

For the needs of Pilot 3 and to better evaluate the glassbox models, a user interface was developed. This user interface is connected to the API and is able to present to the user all the detected outliers along with the concepts identified by the model and allows the CCTV inspector to provide feedback for those predictions.

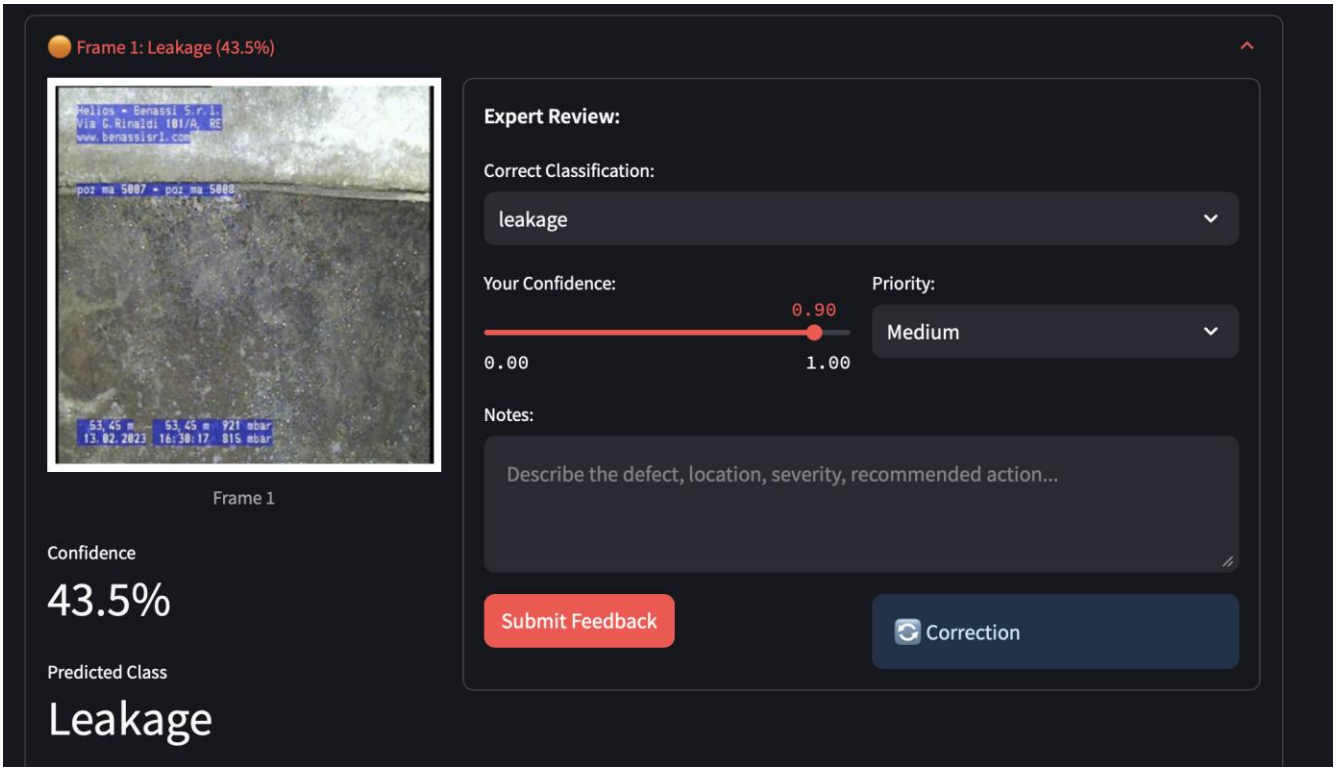

Figure 36 – Identified outlier on Image along with Confidence of the Model with a Feedback Option

Figure 36 presents a representative output of the Concept Bottleneck Model (CBM) applied to a frame extracted from a CCTV pipeline inspection video. The model classifies the frame as **Leakage** with a confidence of 43.5%, and simultaneously produces a ranked explanation of the semantic concepts that drove this decision. Specifically this frame depicts a concrete pipe interior surface exhibiting visible surface dampness and textural irregularity, consistent with early-stage or diffuse leakage — a condition characterised by moisture ingress rather than a discrete, easily localised water flow.

The CBM's symbolic bottleneck layer decomposes the prediction into interpretable concept contributions, visualised as a horizontal bar chart (Figure 37). The two dominant positive contributions — *Irregular, pitted marks* (contribution ≈ +1.45) and the absence of a generic anomaly signal (*NOT Type of Anomaly:*, contribution ≈ +0.62) — indicate that the model associates the observed surface degradation pattern with the textural signatures characteristic of leakage-induced material deterioration. A secondary positive contribution from *Unusual wet spots or puddles* (+0.12) further corroborates the leakage hypothesis by capturing the presence of localised moisture accumulation on the surface.

Conversely, the model's reasoning is appropriately tempered by several contradicting concepts. The strong negative contribution of the Exposed *root system* (≈ −0.35) reflects the absence of biological intrusion, a feature more characteristic of severe, long-standing defects. Similarly, *NOT Water path interruption* (≈ −0.18) and *Surface damage* (≈ −0.17) reduce confidence by signalling that the frame does not exhibit the structural indicators typically associated with high-severity leakage events. These negative contributions explain the relatively moderate confidence score (43.5%), reflecting genuine ambiguity between early-stage leakage and surface-level moisture without structural breach.

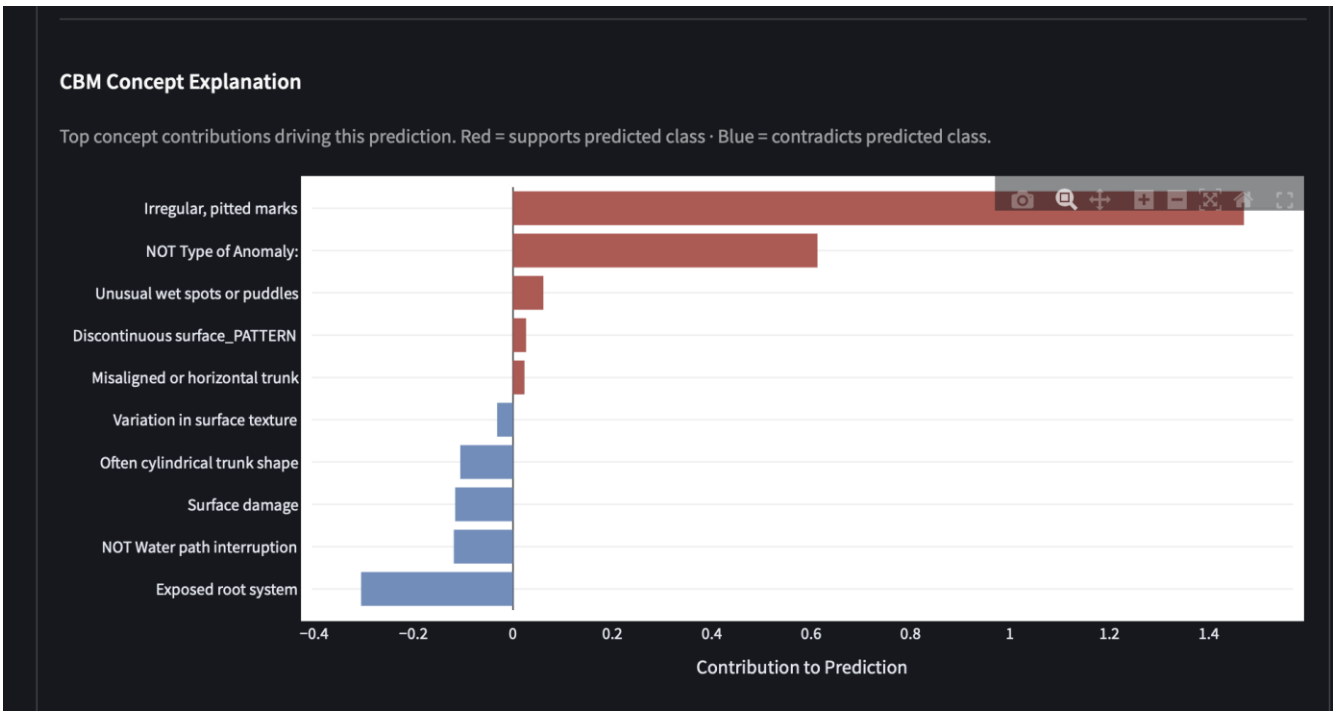

Figure 37 – Various Concepts and their Contributions to the Final Model Prediction of a Certain Frame

This example illustrates a key advantage of the Neurosymbolic CBM architecture over black-box classifiers. Rather than presenting the operator with a bare classification label, the system provides a transparent reasoning trace: the inspector can see precisely which visual properties of the frame the model detected, how strongly each contributed to or contradicted the classification, and where residual uncertainty originates.

4.2.4 Demonstration Scenarios and Mapping to Pilots

Pilot 3 (EKSO) – XR Steaming for Operator 5.0 Training

For the purposes of Pilot 3 Concept Bottleneck Models were integrated with the Neurosymbolic AI component to enhance human understanding by creating AI models that are transparent and easy to interpret. The Glass Box models were presented in the evaluation of the solution (Neurosymbolic AI) for the purposes of Pilot 3. Each model prediction had its concept contribution explanation that resulted in that prediction. The domain expert consulted the model's reasoning and how it reached a certain prediction. Through the dedicated user interface, the CCTV inspector was able to review each flagged frame alongside the ranked concept contributions, gaining insight not only into what the model predicted but also into the visual evidence that supported or contradicted that prediction. This transparency allowed the expert to make more informed decisions when reviewing borderline cases, such as frames where the model assigned moderate confidence scores, and to provide targeted corrective feedback where the model's reasoning was found to be misaligned with operational ground truth. The scenario therefore demonstrated the practical value of glass-box reasoning in an inspection context, establishing a foundation for a continuous human-AI collaboration loop in which expert knowledge and model transparency reinforce one another.

4.2.5 Challenges and Limitations

The quality of the model's explanations is fundamentally bounded by the quality of the concept vocabulary used during training. In our example we have used both an LLM that acted as expert and identified the most common concepts for the different outlier types (e.g. Leakage, crack) as well as the expert's knowledge coming directly from people working for EKSO. Concepts must be defined and validated by domain experts

prior to training; if the concept set is incomplete, ambiguous, or poorly aligned with the visual signatures of the defect classes, the bottleneck layer will produce explanations that are technically accurate at the model level but operationally misleading. As illustrated in the example above, concepts such as *"NOT Type of Anomaly"* carry limited semantic clarity for a non-technical operator, suggesting that concept engineering remains an iterative, labour-intensive process requiring close collaboration between AI researchers and infrastructure inspection specialists.

Also, the CLIP backbone was pre-trained on large-scale natural image datasets and may not fully capture the low-level visual features specific to pipeline inspection environments, which are characterised by constrained lighting, cylindrical geometry, and surface materials (concrete, clay, PVC) absent from natural images. Performance may degrade when the system is deployed on inspection videos from pipelines with materially different characteristics than those represented in the training data, requiring periodic re-evaluation and potential full retraining in new operational contexts.

4.2.6 Evaluation

A dedicated evaluation framework tailored to the specific operational requirements of the CCTV inspection pilot was not developed within the current phase of the project. The definition of a rigorous evaluation protocol – encompassing representative ground-truth annotations, standardised test sets across pipe material types and defect severities, inter-annotator agreement metrics, and domain-specific performance thresholds – constitutes a necessary next step that falls beyond the scope of the present implementation cycle. As a result, no quantitative assessment of the model's concept-level explanation quality has been conducted against a held-out labelled dataset reflective of the pilot environment. This evaluation process will be held in two steps. First, the concepts identified by the model will be assessed by a human expert which will confirm if indeed those concepts are present in the specific frame the model has labeled as outlier or not. The second step will assess the weights the model attributes to each concept and how these weights drive the model's final prediction. Again, the human expert will evaluate if he also uses such features when he identifies an outlier and if the weight of each concept has the same importance as one human expert would attribute in his outlier detection process.

4.2.7 Discussion and Lessons Learned

The integration of glass-box reasoning through Concept Bottleneck Models into the CCTV inspection pipeline demonstrated that Neurosymbolic AI can serve as a viable and practically meaningful approach to interpretable outlier detection in infrastructure monitoring contexts. What someone can deduct from this work is that interpretability is not simply a technical property of the model but depends equally on the quality of the concept vocabulary, the availability of domain expertise and the usability of the interfaces through which experts interact with the system. The use of both a Large Language Model and direct input from EKSO specialists to define the concept set represents a promising hybrid approach to concept engineering. Looking ahead, the primary areas for improvement are threefold: first, refining the concept vocabulary to ensure all concepts carry unambiguous, actionable meaning for non-technical operators; second, investigating domain-adaptive fine-tuning of the visual backbone to reduce the performance gap introduced by the distribution mismatch between natural image pretraining and pipeline inspection footage; and third, completing and validating the active learning feedback loop so that expert knowledge can be continuously and systematically incorporated into the model, transforming the glass-box component from a static classifier into a progressively improving, human-aligned reasoning system.

4.3 Human-centric GPT-based Interaction using CoT

This section provides an overview of human-centric interaction with General Pre-trained Transformer (GPT)-based systems through the application of Chain-of-Thought (CoT) reasoning. This is a prompting paradigm in which a language model decomposes a complex query into a sequence of intermediate reasoning steps before arriving at a final answer. Within the XR5.0 project, CoT serves as the central mechanism for grounding eXplainable AI (XAI) in LLM-based conversational interactions, and is operationalised through two distinct implementations: DataLens, contributed by project partner SIE, and GenAI for XR-based Maintenance Instructions, contributed by project partner OCU. Each approach is described in depth in Section 3.3 *DataLens Conversational AI Framework* and Section 3.2 *GenAI for XR-based Maintenance Instructions*, respectively. This section focuses on how each approach applies CoT as a mechanism for XAI transparency within human-centric interaction.

4.3.1 Role and Functionality

Chain-of-Thought reasoning addresses one of the fundamental interpretability challenges of LLM-based systems, namely the opacity of the inference process. Rather than producing a response in a single, unobservable inference pass, CoT-enabled models externalise their reasoning as a structured trace of logical steps, each of which can be inspected, attributed, and evaluated independently. This staged reasoning approach improves both the accuracy and the interpretability of responses generated by LLMs, particularly in knowledge-intensive and domain-specific settings such as industrial maintenance and technical documentation access.

The CoT mechanism is directly relevant to the requirements set forth by the EU AI Act, particularly under Articles 13 and 14, which mandate transparency and human oversight for high-risk AI systems. In industrial environments where AI-generated guidance may inform safety-critical decisions, such critical specifications, wiring procedures, or maintenance sequencing, the ability to trace a response back through its full reasoning chain is a regulatory and operational necessity. By making the model's intermediate reasoning observable rather than presenting only a final answer, CoT provides the structural foundation for traceable, auditable, and human-interpretable AI behaviour that is compliant within these regulations.

Within the XR5.0 project, this principle is operationalised through two distinct but complementary approaches, as presented in the following subsections.

Chain-of-Thought in DataLens

DataLens implements a hybrid XAI approach that combines two complementary reasoning layers. The first is a system-defined decision tree, which provides a structured and predictable template for resolving incoming user queries through a sequence of specialised system prompts and metadata filters applied at each stage of the pipeline. The second is the LLM's internal thinking mechanism, which navigates and resolves each individual node of the decision tree, producing fine-grained reasoning at every inference step. Together, these two layers ensure that the system's behaviour is both structurally auditable through the deterministic decision tree, and semantically interpretable through the model's captured internal reasoning.

At every stage of the end-to-end interaction, the reasoning produced by each LLM inference node is captured and incrementally appended to a cumulative, Markdown-formatted reasoning trace, which is returned as a dedicated field in the API response. This design elevates the reasoning trace from an internal debugging artefact to a first-class output, making the full decision path from user query to the generated response, transparent and accessible to both human operators and downstream application layers. A detailed description of the complete response workflow, including all pipeline stages and their interactions, is provided in subsection 3.3.2 *System Architecture* of the DataLens Conversational AI Pipeline.

Chain-of-Thought in GenAI for XR-based Maintenance Instructions

The GenAI for XR-based Maintenance Instructions system developed by OCU applies Chain-of-Thought (CoT) prompting to enable transparent and human-centric interaction between technicians and AI-assisted maintenance systems. The approach integrates conversational AI with retrieval-augmented knowledge access to support technicians during troubleshooting and maintenance procedures. When a technician submits a request through the conversational interface, typically through a voice-to-voice interaction on a mobile device, the system retrieves relevant documentation and maintenance knowledge from a structured repository and processes the query using a large language model. Instead of generating a response as a single opaque output, the system applies Chain-of-Thought prompting to decompose the request into intermediate reasoning steps that reflect the logic behind the generated troubleshooting guidance.

The reasoning trace generated by the model is used to provide eXplainable AI (XAI) functionality within the user interface. Rather than presenting explanations continuously, the system exposes these reasoning steps through optional explanation panels that technicians can access when additional clarification is required. This design supports trust and interpretability while avoiding unnecessary cognitive load during routine troubleshooting tasks. The reasoning chain also allows technicians and supervisors to understand how a recommendation was derived from underlying documentation, which is particularly relevant when maintenance guidance must be verified against technical manuals or operational procedures.

Within the broader XR5.0 architecture, the CoT-based reasoning mechanism is integrated with the OCU knowledge base and the SHARE collaboration platform. Retrieved documentation, maintenance workflows, and historical troubleshooting cases serve as grounding sources for the reasoning process. The resulting explanations can be visualized within mobile troubleshooting interfaces. This integration ensures that generative AI recommendations remain transparent, verifiable, and aligned with the human-centric design principles required for industrial maintenance environments.

4.3.2 System Architecture

Chain-of-Thought in DataLens

The XAI properties of the DataLens system are highlighted directly from its pipeline structure. Rather than being applied in the final stage, explainability is shaped at each stage of a sequential, multi-step workflow in which every inference decision is scoped, constrained, and captured before the next stage begins. The pipeline follows an n-stage CoT methodology, where each stage contributes a discrete and traceable reasoning step toward the final response, as illustrated in Figure 38.

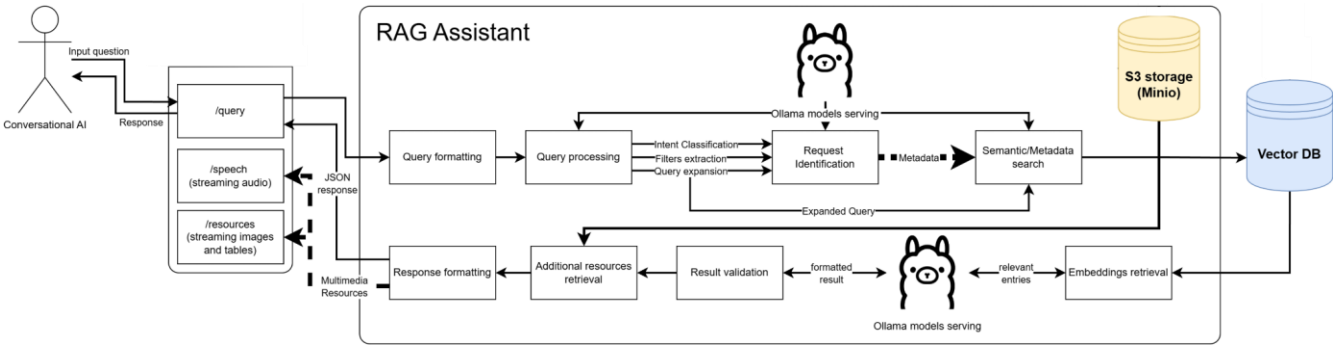

Figure 38 – DataLens RAG CoT Pipeline High-level Architecture

1. Route Classification - The pipeline is initiated by a route classification step in which the LLM determines the nature of the incoming interaction. Four route types are defined: *NEW_QUERY*, *FOLLOW-UP_CONFIRMATION*, *FOLLOW-UP_QUESTION*, and *CHITCHAT*. This classification establishes the conversational context and governs which downstream stages are activated, ensuring that the system's behaviour is conditioned on an explicit, logged, and human-readable routing decision from the outset.
2. Query Expansion and Filter Extraction - The user's input is enriched and optimised for semantic search through a query expansion step, which reformulates the query to improve retrieval relevance. In parallel, filtering parameters are identified and extracted to narrow the search space over the available knowledge base. Both the expanded query and the extracted filters are captured as intermediate reasoning artefacts, providing a transparent record of how the system interpreted and prepared the user's intent before retrieval.
3. Query Intent Classification - The optimised query is passed through an intent classifier that determines the appropriate response generation path. Six intent types are defined: *SUMMARIZATION*, *QA_RESPONSE*, *IMAGE_LOOKUP*, *TABLE_LOOKUP*, *PAGE_LOOKUP*, *TOC_LOOKUP*. Six distinct intents are defined, each associated with a dedicated system prompt and a specific set of LLM optimisation parameters. This stage is a key XAI contributor, where the classified intent is explicit and logged, and the selection of a specific system prompt and parameter configuration means that the conditions under which the final answer is generated are fully attributable and inspectable.
4. File Name Resolution - Prior to response generation, the system identifies the most relevant technical manual or engineering datasheet from which the answer should be derived. This is achieved by matching the optimised query against document-level summaries to determine the most relevant source. By making source attribution an explicit pipeline stage rather than an implicit retrieval side-effect, the system ensures that the provenance of every response is traceable and verifiable.
5. Final Response Generation and Formatting - The final response is generated using the system prompt, LLM parameters, and retrieved context established in the preceding stages. The output is formatted according to the response schema defined in the API, including the Markdown reasoning trace that aggregates the captured reasoning from all prior stages into a single, structured, and human-readable artefact returned alongside the final answer.

To illustrate the pipeline in practice, Code 2 presents a reasoning trace generated by the system in response to the user query *"Generate me a summary of the KUKA Robot arm featured within the system"*. The trace exposes the full sequential reasoning chain from route classification and query expansion, through intent resolution and filename matching, to the final response generation logic. Thus, providing a concrete and human-readable record of every inferential decision made by the system across all five stages.

```
## User input query:
create me a summary about the robot arm featured in the system

## Conversation Routing:
Route: NEW_QUERY
We need to route. The user says: \"create me a summary about the robot arm featured in the system\". This is a new technical question. No LAST_SUGGESTIONS. So route NEW_QUERY. resolved_query: \"summary robot arm featured in the system\". needs_search true. chitchat_response null.
Resolved query: summary robot arm featured in the system

## Processing user query:
We need to parse user query: \"summary robot arm featured in the system\". They want a summary of robot arm featured in the system. So intent: SUMMARIZATION. Expanded query: \"robot arm system features summary\". Remove filler. No page number. filename_fuzzy maybe \"robot arm\" or \"system\". Provide JSON.

## Resolving filename:
We need to match target_ref \"robot arm\" to doc index. The doc index has two entries: KUKA_KR_10_R1100-2 and DataSheet MID RANGE DISTANCE SENSORS Sick DT35-B15251. The target is \"robot arm\". The KUKA doc is about an industrial robot, which is a robot arm. The sensor doc is about a sensor. So match KUKA_KR_10_R1100-2. Return filename string exactly. So output: KUKA_KR_10_R1100-2.

## Generating final response:
Semantic search results: [('KUKA_KR_10_R1100-2_image_6', 0.7549066543579102), ('KUKA_KR_10_R1100-2_image_7', 0.7691254019737244), ('KUKA_KR_10_R1100-2_summary', 0.7831407189369202), ('KUKA_KR_10_R1100-2_table_1', 0.808718740940094), ('KUKA_KR_10_R1100-2_image_3', 0.810034990310669), ('KUKA_KR_10_R1100-2_table_2', 0.8225975036621094)].

We need to produce JSON with relevant_contents_id list, speech_response, follow_up_question, markdown_response.
User query: \"robot arm system features summary\". They want a summary of robot arm system features. We have data: summary, table 1 (technical data), table 2 (axis motion range), images (workspace graphic, technical drawing). We should cite IDs. Provide summary of features: payload, reach, repeatability, axes, mounting options, footprint, weight, operating temperature, protection rating, controller, cycle time, workspace graphic, axis limits.

We need to include relevant IDs: KUKA_KR_10_R1100-2_summary, KUKA_KR_10_R1100-2_table_1, KUKA_KR_10_R1100-2_table_2. Also maybe images: KUKA_KR_10_R1100-2_image_3 (workspace graphic). But summary may not need images. But we can mention workspace graphic (image 3). So relevant IDs: [\"KUKA_KR_10_R1100-2_summary\",\"KUKA_KR_10_R1100-2_table_1\",\"KUKA_KR_10_R1100-2_table_2\",\"KUKA_KR_10_R1100-2_image_3\"].

Speech: 3 sentences max, conversational. Provide summary: \"The KUKA KR 10 R1100-2 is a six-axis industrial robot with a 10 m reach, 10 kg payload, high repeatability, etc.\" Mention key specs. Also mention workspace graphic.

Follow-up question: maybe ask if they'd like details on mounting options or axis limits. Provide suggestion: \"Would you like to see the detailed axis limits or mounting configurations?\" Ensure not referencing IDs already shown? They have seen IDs? Not specified. So we can ask.

Markdown: structured with headers: Overview, Mechanical Specifications, Electrical & Control, Workspace & Mounting, Safety & Protection, Key Applications. Use bullet points, bold key terms. Cite references: (See Table 1), (See Table 2), (See Image 3). Provide citations.

Let's craft.
```

Code 2 – DataLens Output Reasoning Example

Chain-of-Thought in GenAI for XR-based Maintenance Instructions

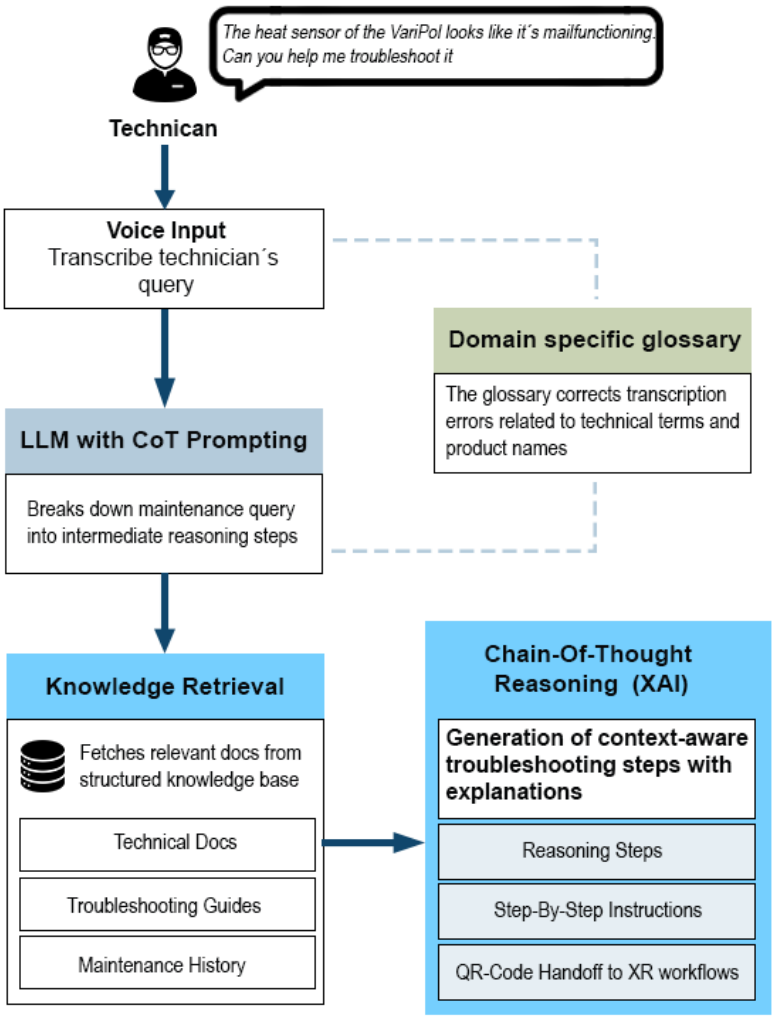

Figure 39 – High-level System Architecture of CoT in GenAI for XR-based Maintenance Instructions Implementation

The system architecture in Figure 39 follows a human-centric AI pipeline that combines voice interaction, retrieval-augmented generation, and explainable reasoning.

A technician initiates the process through voice input, which is first transcribed and then refined using a domain-specific glossary. This glossary corrects transcription errors related to technical terms, product names, and person names, ensuring accurate interpretation of the query before further processing. The refined input is processed by a large language model using Chain-of-Thought prompting, which decomposes the query into intermediate reasoning steps. Relevant information is retrieved from a structured knowledge base, including technical documentation, troubleshooting guides, and maintenance history. Based on this grounded information, the system generates context-aware troubleshooting instructions. In parallel, a structured reasoning trace is produced and made available through optional explanation panels, allowing users to access detailed explanations when needed. The final output is delivered as clear, step-by-step guidance, ensuring that AI recommendations are traceable, interpretable, and grounded in verified knowledge, supporting effective and trustworthy human-AI collaboration.

4.3.3 Implementation

Chain-of-Thought in DataLens

The CoT pipeline within DataLens is implemented through a set of specialised system prompts and LLM query parameters, each scoped to a specific conversational route or query intent, collectively enabling traceable and predictable system behaviour. All pipeline configuration that includes route definitions, intent classifiers, system prompts, and parameter sets are unified under a single configuration file, *rag_runtime.yaml*, which serves as the central artefact governing the conversational logic of the system. This design choice not only simplifies development and debugging but also ensures that extending the framework with new routes or intents remains a contained and low-friction operation, directly supporting the long-term adaptability of DataLens across evolving XR5.0 use cases.

Chain-of-Thought in GenAI for XR-based Maintenance Instructions

The CoT-based interaction is implemented through a pipeline combining conversational AI, retrieval-augmented generation, and explainability within the OCU platform. Technician input is captured via a voice-to-voice interface, transcribed, and improved using a domain-specific glossary to handle technical terminology. Prompt templates enforce stepwise reasoning, guiding the model to retrieve relevant knowledge from the OCU Knowledge Base and generate structured troubleshooting instructions. The reasoning trace is produced alongside the final answer and made available through optional explanation panels to balance transparency and cognitive load. The system integrates with the SHARE platform linking AI outputs to maintenance intelligence. An active learning loop captures feedback and continuously improves retrieval quality and reasoning performance. Generated workflows can also be transferred to XR environments for spatial guidance.

4.3.4 Demonstration Scenarios and Mapping to Pilots

Pilot 1 (KUKA) – Rapid Human Centric AI-Enabled Product Design

Chain-of-Thought in DataLens: DataLens originated as KUKABot, a privacy-driven local solution developed in the initial phase of the project to address the specific use cases of the KUKA Pilot. Through iterative development and successive testing phases, including formal Pilot Evaluations, the framework was progressively refined into the CoT-based pipeline described in this chapter, offering more confident, better-grounded responses applicable across a broad range of technical documentation. A comprehensive account of the demonstration scenarios and their mapping to the XR5.0 pilots is provided in Section 3.3.4 *Demonstration Scenarios and Mapping to Pilots* of the *DataLens Conversational AI Framework* section.

Pilot 2 (SH) – Human Centered Remote Maintenance and Asset Management

Chain-of-Thought in GenAI for XR-based Maintenance Instructions: The system is demonstrated in Pilot 2 at SCHMIDT and HAENSCH, focusing on AI-assisted troubleshooting and XR-supported maintenance. The first scenario combines XR collaboration with remote expert support, where AR overlays and annotations during a video-call deliver insights and assist real-time decision-making. In the second scenario technicians use a mobile interface with voice interaction to retrieve troubleshooting instructions and explanations grounded in technical documentation. The explanations are derived via the described Chain-Of-Thought pipeline. In the third scenario, user generated workflows are transferred via QR-Code handoff from the mobile interface to XR devices, enabling spatial guidance and hands-free interaction. These scenarios illustrate how generative AI and XR complement each other across different maintenance contexts.

4.3.5 Challenges and Limitations

Chain-of-Thought in DataLens

The primary challenge identified within the DataLens CoT pipeline relates to response latency, where the complexity of the multi-stage reasoning process inevitably increases the time required to produce a response beyond the approximately 10-second exchange cadence characteristic of natural human conversation. This overhead is a direct consequence of the decision tree traversal imposed on the LLM at each pipeline stage, compounding inference time across route classification, intent resolution, and retrieval before a final response is generated. Extensive model evaluation was conducted throughout the testing phases to identify the most suitable backbone for the workflow, with results indicating that *gpt-oss* offered the most balanced and stable performance across the pipeline stages. Other reasoning-capable models, while theoretically well-suited to a CoT approach, exhibited a tendency to over-elaborate their internal thinking process, consuming a disproportionate share of the token generation budget without a corresponding improvement in response quality. This behaviour is further compounded by the current absence of a configurable thinking budget for reasoning models served via Ollama, leaving token allocation an open limitation with no viable mitigation at the time of writing.

Chain-of-Thought in GenAI for XR-based Maintenance Instructions

Key challenges include balancing explainability and performance. CoT reasoning improves transparency but increases response latency, which can affect time-critical tasks. Speech recognition in noisy environments remains a limitation, although glossary integration improves accuracy. XR usability also presents challenges, including device ergonomics, interaction complexity, and responsiveness. Additionally, the quality of retrieval-augmented generation depends on the availability and structure of maintenance documentation, which can vary across real-world scenarios.

4.3.6 Evaluation

Chain-of-Thought in DataLens

The evaluation of the DataLens CoT pipeline was conducted across the XR5.0 pilot testing phases, with assessment focused on the accuracy of conversational routing, intent classification correctness, and the relevance and grounding of generated RAG responses against their source technical documentation. Results demonstrated that the CoT structure consistently produced more reliable and traceable outputs compared to earlier single-prompt approaches, with the pipeline's reasoning trace providing a transparent basis for identifying and correcting failure cases during iterative refinement. Response latency was acknowledged as a measurable trade-off, however the improvement in response confidence and document grounding was deemed to outweigh the overhead introduced by the multi-stage reasoning process across the evaluated use cases.

Chain-of-Thought in GenAI for XR-based Maintenance Instructions

Evaluation in Pilot 2, as documented in D6.5 – *Socio-technical Evaluation of XR5.0 Systems v1,* focused on functionality, usability, trust, and compliance. Results show that CoT-based explanations improve understanding and trust in AI recommendations, especially when presented on demand. Retrieval quality was generally high when sufficient documentation was available, and glossary-based corrections improved transcription accuracy. While response latency increased due to reasoning steps, performance remained acceptable for most use cases. User feedback indicated strong usability for mobile workflows, while XR was valued mainly for spatial guidance and remote support. Compliance evaluation confirmed support for traceability, logging, and role-based access aligned with EU AI Act requirements.

4.3.7 Discussion and Lessons Learned

The development and deployment of Chain-of-Thought (CoT) in the services yielded several key lessons that extend beyond the immediate scope of the framework. In the case of the DataLens, the transition from a single-prompt architecture to a structured CoT pipeline demonstrated that explainability and reliability in RAG systems are best achieved through deliberate decomposition of the reasoning process rather than reliance on model capability alone. The centralisation of configuration within *rag_runtime.yaml* proved to be a particularly valuable design decision, significantly reducing integration friction during iterative development. Furthermore, the model evaluation process reinforced that raw reasoning capability does not necessarily translate to pipeline suitability, and that stability, predictability, and token efficiency are equally critical selection criteria when deploying LLMs within structured conversational workflows.

In the case of GenAI for XR-based Maintenance Instructions, the implementation shows that explainability is most effective when provided on demand, supporting trust without increasing cognitive load. A key finding is the strong preference for mobile-first interaction. Tablets offer faster responsiveness and familiar interfaces, while voice interaction simplifies access to information. XR devices were perceived as less practical for continuous use due to ergonomics and interaction complexity, but valuable in specific scenarios such as hands-free operation and remote collaboration. The integration of a domain glossary highlighted the importance of handling technical terminology for reliable AI performance. Overall, the system demonstrates that effective human-centric AI requires a balance between explainability, usability, and performance, with mobile-first design complemented by targeted XR support. The concrete evaluation metrics will be specified in D4.4 – *XR Visualisations of AI and XAI Outcomes v2* and they will be evaluated in D6.6 – *Socio-technical Evaluation of XR5.0 Systems v2*.

5. CONCLUSIONS

5.1 Summary

Deliverable D4.2 – Advanced AI Paradigms for Human-AI Collaboration v2 concludes the activities of WP4 in the research and developing of robust AI solutions for the purposes of the XR5.0 project and presents the final advancements in the development, implementation, and integration of advanced AI technologies for human-AI collaboration within XR-enabled industrial environments. Building upon the results reported in Deliverable D4.1, this deliverable consolidates the progress achieved during the second phase of WP4 and demonstrates the maturity of the proposed AI paradigms through their implementation, integration within the XR5.0 platform architecture, and validation across multiple industrial pilots as reported in Deliverable D6.5 – *Socio-technical Evaluation of XR5.0 Systems v1*. The latter is based on a structured, survey-driven assessment framework designed to systematically capture user feedback across all pilots while enabling in-depth analysis of pilot-specific innovations.

The work carried out in WP4 focuses on strengthening the synergy between Artificial Intelligence (AI) and Extended Reality (XR) technologies to support augmented intelligence in Industry 5.0 scenarios. The developed AI solutions aim to enhance human decision-making, improve transparency and trust in AI systems, and enable seamless collaboration between human operators and intelligent systems.

Within Task T4.1, trustworthy AI models and infrastructures were developed to support transparent and interpretable human-AI collaboration. The work introduced eXplainable AI (XAI) methods, including Post-hoc explainability techniques such as LIME and Grad-CAM, which provide visual and feature-level explanations for AI predictions. In addition, glass-box reasoning models were implemented to expose intermediate reasoning steps in AI decision-making processes. Complementing these approaches, human-centric GPT-based interaction using Chain-of-Thought (CoT) reasoning enables conversational AI systems to provide structured reasoning traces, allowing users to better understand how AI-generated responses are derived. These methods enhance user trust, support human oversight, and contribute to compliance with emerging regulatory frameworks such as the EU AI Act.

Task T4.2 focused on the development of XR-enabled human-AI collaboration through Neurosymbolic AI (NSAI) and Active Learning (AL) techniques. The proposed neurosymbolic framework combines deep neural perception with symbolic reasoning to produce interpretable AI decisions. The implemented Concept Bottleneck Models (CBMs) introduce a symbolic reasoning layer that expresses predictions through human-understandable concepts, enabling domain experts to inspect and influence the reasoning process. Additionally, active learning mechanisms allow experts to provide feedback that continuously improves model performance, establishing a human-in-the-loop learning cycle particularly suited to industrial contexts with limited labelled data.

Within Task T4.3, the work focused on the integration of Generative AI technologies to support XR-enabled knowledge access and human-centric assistance systems. Conversational AI frameworks were developed to enable intuitive interaction with technical documentation, maintenance procedures, and operational knowledge. These systems rely on Retrieval-Augmented Generation (RAG) pipelines, customizable AI agents, and multimodal interaction mechanisms to provide contextualized guidance and troubleshooting support. Through natural language queries, voice interaction, and XR interfaces, operators can access complex information more efficiently and interact with industrial knowledge bases in a more intuitive way.

The AI technologies developed within WP4 were subsequently integrated and demonstrated across the XR5.0 industrial pilots as presented in Table 14, validating their applicability in real operational contexts.

Table 14 – Mapping of XR5.0 Pilots to WP4 AI Technologies

	WP4 AI Technologies	Description of Role	Technical Partner
Pilot 1 (KUKA)	GenAI, RAG-based conversational AI (DataLens)	Privacy-preserving conversational AI assistant enabling natural language querying of technical documentation with structured, explainable responses	SIE
	XAI (CoT reasoning)		
Pilot 2 (SH)	GenAI conversational assistant	Mobile-first AI assistant for maintenance troubleshooting, knowledge retrieval, and XR-assisted remote collaboration with logically traceable textual explanations	OCU
	XAI (CoT reasoning)		
Pilot 3 (EKSO)	NSAI (CBM)	AI-assisted inspection and anomaly detection with interpretable reasoning, expert feedback loops	UPRC
	Active Learning (human-in-the-loop)		
	XAI (Post-hoc, CBM)		
	GenAI agent (LLM Chat Engine, RAG-based)	AI-assisted conversational access to inspection knowledge	INNOV
Pilot 4 (TAP)	GenAI Assistant (DataLens)	Conversational AI for maintenance training and documentation access	SIE
	NSAI under evaluation	NSAI being evaluated for object recognition and training support	UPRC

Pilot 5 (SPACE)	NSAI (Reasoning Context & Contextual Models)	Personalised XR training using biometric (stress) monitoring and adaptive content delivery	UPRC
	GenAI agent (LLM Chat Engine, RAG-based)	Conversational AI for training support	INNOV
Pilot 6 (LNS)	GenAI agent (LLM Chat Engine, RAG-based)	AI-based troubleshooting assistant for diagnosing machine alarms and retrieving corrective procedures	INNOV

In **Pilot 1 (KUKA)**, the **DataLens conversational AI framework by SIE** was deployed as a privacy-preserving, on-premise knowledge assistant enabling operators to query technical documentation through natural language interactions within XR environments. In this pilot, **conversational AI capabilities** and **Chain-of-Thought explainable reasoning mechanisms** support technicians in accessing engineering documentation and training content through intuitive voice and text-based interfaces.

In **Pilot 2 (SH)**, **generative AI solutions** were applied through a **mobile-first conversational AI assistant by OCU** that supports maintenance troubleshooting and operational guidance. This system enables technicians to retrieve relevant knowledge from maintenance documentation and historical service records through natural language interactions, while also incorporating **Chain-of-Thought reasoning mechanisms** to provide structured and traceable responses. In addition, the solution supports XR-assisted collaboration with remote experts when necessary.

In **Pilot 3 (EKSO)**, **Neurosymbolic AI, Active Learning** and **XAI solutions by UPRC** as well as **a generative AI agent by INNOV** were applied to support AI-assisted inspection and anomaly detection in smart water pipe infrastructure through CCTV inspection analysis. In this scenario, the Neurosymbolic framework enables interpretable classification of infrastructure anomalies, while XAI mechanisms provide operators with visual explanations and reasoning traces that support decision-making during inspection processes. Active learning further enables domain experts to provide feedback on model predictions, supporting continuous improvement of the system and adaptation to real operational conditions. The generative AI agent provides conversational access to inspection documentation and operational knowledge, enabling operators to retrieve relevant information and guidance through natural language interaction.

In **Pilot 4 (TAP), a generative AI assistant** based on the **DataLens framework by SIE** was deployed to provide conversational access to maintenance documentation and training material, enabling users to retrieve contextual information through natural language interaction and enhancing training effectiveness through AI-supported guidance. The **Neurosymbolic AI framework is being evaluated for integration** in Pilot 4, where AI-based object recognition and classification models can assist in aircraft maintenance training scenarios to support the identification of aircraft engine components within XR environments and enable more effective operator training through AI-supported guidance.

In **Pilot 5 (SPACE)**, the NSAI framework's **Reasoning Context and Contextual Models component by UPRC** was integrated to support personalised XR-based training and task guidance. The system leverages wearable devices to monitor operator stress levels during training and assembly tasks, enabling the identification of learning gaps and the dynamic adaptation of training content to individual user needs. In addition, Pilot 5 incorporates the **LLM Chat Engine by INNOV** in a self-service configuration, allowing the pilot team to independently manage documentation and system integration, supporting VR-based assembly training and AR-assisted repair workflows.

Finally, **Pilot 6 (LNS)** demonstrated the application of **generative AI-based conversational assistant by INNOV** for industrial machine troubleshooting, where AI agents assist technicians in diagnosing machine alarms and retrieving relevant corrective procedures from technical documentation.

The work presented in this deliverable demonstrates the feasibility and benefits of AI/XR symbiosis for augmented intelligence, where advanced AI technologies augment human capabilities rather than replace them. The integrated approaches contribute to the broader vision of Industry 5.0, where AI systems are designed to be human-centric, trustworthy, resilient, and collaborative, empowering workers and supporting more efficient and intelligent industrial processes.

5.2 Mitigation of Challenges and Limitations

The work carried out in WP4 revealed several technical and operational challenges, many of which were addressed through the implemented methodologies. In Task T4.1, challenges related to the limited interpretability of black-box models, cognitive overload in XR environments, and the need for traceable reasoning were addressed through the integration of post-hoc XAI methods, context-aware explanation strategies, glass-box models, and Chain-of-Thought (CoT) reasoning.

In Tasks T4.1 and T4.2, the development of concept-based and Neurosymbolic models revealed challenges in concept engineering and model generalizability, which were partially addressed through iterative refinement with domain experts and pilot-specific adaptations. In Task T4.2, data scarcity and domain adaptation limitations were mitigated through Active Learning and human-in-the-loop feedback mechanisms, while real-time performance constraints were partially addressed through optimisation of inference pipelines.

In Task T4.3, challenges related to the reliability and grounding of Generative AI outputs, including hallucinations and inconsistent responses, were addressed through the adoption of Retrieval-Augmented Generation (RAG) architectures, multi-stage retrieval pipelines, and intent-aware query routing. Additional challenges concerning computational requirements, system complexity, and deployment were mitigated through modular, microservices-based architectures and on-premise solutions such as DataLens. Finally, usability challenges identified across pilots—particularly the limited suitability of XR for routine tasks—were addressed through multimodal interaction design, combining mobile, voice, and XR interfaces depending on the operational context. Overall, while certain limitations such as real-time performance, scalability, and cross-domain generalisation remain partially open, the approaches developed in WP4 provide robust and practical solutions for advancing human-centric AI in industrial XR environments.

These mitigation efforts underline that the successful integration of AI and XR technologies depends not only on the development of advanced algorithms, but also on careful attention to human-centric design, robust and scalable system architectures, and sustained collaboration with domain experts.

5.3 Lessons Learned

The development, implementation, and pilot deployment of the advanced AI paradigms within WP4 provided valuable insights regarding the design, integration, and practical deployment of human-centric AI systems in industrial XR environments.

A key lesson concerns the importance of transparency and interpretability in AI-assisted industrial workflows. The integration of eXplainable AI (XAI), glass-box models, and Chain-of-Thought (CoT) reasoning mechanisms demonstrated that operators are more likely to trust and effectively interact with AI systems when they can understand the reasoning behind predictions and recommendations. At the same time, the deployment of XAI within XR environments highlighted the need for careful interface design, as presenting too many explanations or visualisations simultaneously can lead to cognitive overload for users. Context-aware and on-demand explanations were therefore identified as a more effective strategy for delivering interpretability in immersive environments. These findings were validated across multiple industrial pilots, as further detailed in Deliverable D6.5, *Socio-technical Evaluation of XR5.0 Systems v1*.

Another important lesson relates to the role of human expertise in the AI lifecycle. The Neurosymbolic AI and Active Learning approaches developed in WP4 confirmed that human-in-the-loop mechanisms are essential in industrial contexts where labelled datasets are limited and domain expertise plays a central role. In particular, the Neurosymbolic inspection framework demonstrated that combining neural perception with symbolic reasoning enables interpretable predictions while allowing experts to intervene in the reasoning process. However, the development of concept-based models also highlighted the importance of concept engineering, as the quality and clarity of the concept vocabulary directly influence the interpretability and usefulness of the explanations provided by the system.

The project also revealed several data-related challenges. In multiple pilot scenarios, the availability of labelled data was limited, particularly for rare anomalies or failure conditions. This limitation reinforced the importance of active learning and expert feedback loops for progressively improving model performance. Furthermore, the use of pretrained vision models trained on general datasets may introduce domain adaptation challenges when applied to specialised industrial inspection environments, requiring further refinement and evaluation when deployed in new operational contexts.

From an architectural perspective, the implementation of Retrieval-Augmented Generative AI systems demonstrated the effectiveness of combining document retrieval mechanisms with large language models for knowledge access and troubleshooting support. The pilot deployments showed that conversational AI systems can significantly improve the accessibility of technical documentation. At the same time, the evaluation results from D6.5 highlighted the importance of structured retrieval pipelines, query routing mechanisms, and intent classification in order to improve the accuracy and reliability of generated responses. The adoption of modular architectures and multi-stage RAG pipelines proved essential for producing grounded responses and maintaining traceability to source documentation.

The pilot demonstrations also provided important insights regarding user interaction modalities in industrial environments. While XR technologies offer strong benefits for spatial visualization, immersive training, and remote expert collaboration, technicians often preferred mobile or tablet-based interfaces for routine troubleshooting tasks due to their familiarity, responsiveness, and ease of use. XR devices were perceived as particularly valuable in scenarios where hands-free interaction, spatial guidance, or remote collaboration clearly improved the maintenance or training process.

Finally, the project highlighted several practical deployment considerations related to AI infrastructure and system integration. Generative AI systems and conversational assistants require significant computational resources and careful system orchestration to ensure acceptable response latency. Additionally, the rapidly evolving ecosystem of AI frameworks and models introduces maintenance challenges, requiring continuous monitoring of dependencies and model updates. These observations emphasize the importance of modular system architectures, containerized deployments, and scalable infrastructures when integrating advanced AI capabilities into complex industrial platforms such as XR5.0.

The experience gained within WP4 also highlights the importance of aligning technical developments with real operational conditions and user requirements. In this context, the results of WP4 contribute to the

ongoing development of AI-enabled XR solutions in line with Industry 5.0, where AI systems are intended to support and augment human capabilities while ensuring transparency, trust, and responsible deployment.