

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Architectural design for Human-Centered AI Systems: the NOUS Project use case

João Maldonado



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Ademar Aguiar (FEUP)

Co-supervisor: Prof. Hugo Paredes (UTAD)

July 21, 2025

Architectural design for Human-Centered AI Systems: the NOUS Project use case

João Maldonado

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. André Restivo

Referee: Prof. Ângelo Martins

Referee: Prof. Ademar Aguiar

July 21, 2025

Abstract

The proliferation of Artificial Intelligence into high-stakes domains, from financial services to civil protection, has transformed “meaningful human oversight” from an ethical ideal into a pressing legal and technical mandate. While principles for responsible AI abound, practitioners lack concrete, validated blueprints for architecting systems that are simultaneously scalable, auditable, and responsive to human judgment. This dissertation confronts this critical gap by presenting a reference architecture for integrating Human-in-the-Loop mechanisms into complex AI systems.

Grounded in the EU Horizon NOUS project and leveraging Design Science Research, this work translates abstract requirements into a formal, C4-modelled architecture. The core contribution is a modular framework comprising a Decision Guard for real-time policy enforcement, an Expert Review Queue for scalable oversight, an Explainability Service for transparent reasoning, and a Feedback Manager for continuous model improvement. These components are designed to operate asynchronously, preserving the performance of the core data pipeline while creating verifiable, ledger-anchored audit trails for every human intervention.

The architecture’s feasibility and soundness are validated through its application to the CRIMSON crisis-management use case, demonstrating how it satisfies demanding functional and non-functional requirements without compromising security or real-time operation. By delivering a deployable and extensible blueprint, this research moves beyond theory to provide an engineered, practical foundation for building the next generation of trustworthy, Human-Centered AI systems.

Keywords: Human-in-the-Loop, Software Architecture, Artificial Intelligence, Explainable AI, Human Oversight

Resumo

A proliferação da Artificial Intelligence em domínios de alto risco, desde serviços financeiros à proteção civil, transformou a “supervisão humana” passando de um ideal ético a uma obrigação legal e técnico premente. Embora os princípios para uma Responsible AI abundem, os profissionais carecem de referências concretas e validadas para arquitetar sistemas que sejam simultaneamente escaláveis, auditáveis e responsivos ao julgamento humano. Esta dissertação enfrenta esta lacuna crítica ao apresentar uma arquitetura de referência para a integração de mecanismos de Human-in-the-Loop em sistemas de Artificial Intelligence complexos.

Fundamentado no projeto EU Horizon NOUS e utilizando a metodologia de Design Science Research, este trabalho traduz requisitos abstratos numa arquitetura formal, modelada com o C4 model. A contribuição central é um framework modular que inclui um Decision Guard para a validação face a regras em tempo real, uma Expert Review Queue para supervisão escalável, um Explainability Service para raciocínio transparente, e um Feedback Manager para a melhoria contínua de modelos. Estes componentes foram desenhados para operar de forma assíncrona, preservando a performance da pipeline de dados principal, ao mesmo tempo que criam registos auditáveis, verificáveis e ancorados numa ledger para cada intervenção humana.

A viabilidade e solidez da arquitetura são validadas através da sua aplicação ao caso de uso de gestão de crises CRIMSON, demonstrando como esta satisfaz requisitos funcionais e não-funcionais exigentes sem comprometer a segurança ou a operação em tempo real. Ao fornecer uma referência implementável e extensível, esta investigação vai além da teoria para oferecer uma base de engenharia prática para construir a próxima geração de sistemas de Human-Centered AI confiáveis.

Palavras-chave: Human-in-the-Loop, Software Architecture, Artificial Intelligence, Explainable AI, Human Oversight

Contribution to Sustainable Development

The architectural framework in this thesis contributes to the UN Sustainable Development Goals (SDGs) [37] by providing an engineering foundation for responsible AI. The table below outlines this alignment, mapping thesis contributions to specific SDG targets.

Table 1: Alignment of thesis outcomes with selected UN SDG targets.

SDG	Target	Contribution	Performance Indicators / Metrics
9	9.1	The standardised HITL architecture provides a blueprint for resilient digital infrastructure across European data spaces.	• Nodes adopting the architecture • Mean uptime increase • Deployment lead-time reduction
	9.5	This research advances human-centered AI engineering through new design principles, patterns, and open artefacts.	• Peer-reviewed outputs • Documented cross-project reuse
11	11.5	The CRIMSON use case demonstrates a mechanism to improve early warning and decision-support for disaster response.	• Decision latency vs. baseline • False-negative rate in detection • Qualitative outcome improvements
	11.b	The architecture embeds integrated risk-management hooks that align with local disaster-reduction strategies.	• Municipalities integrating nodes • Compliance gap closure time
16	16.6	Immutable ledgers and fine-grained provenance logs are designed to create effective and transparent AI decision pipelines.	• % of decisions with traceable audit • Average audit-retrieval time
	16.7	Role-aware dashboards and escalation workflows are designed to enable responsive and participatory human oversight.	• Operator interventions per inference • SME satisfaction/usability scores

Acknowledgements

I would like to thank my two supervisors for their support and guidance throughout the last year. To Professor Ademar Aguiar, thank you for introducing me to the NOUS project and for sharing your passion for software architecture with me. To Professor Hugo Paredes, who accompanied me in my work week after week, sharing a lot of knowledge and providing incredible guidance. To Professors Filipe Correia, Marco Oliveira and Carlos Duarte, whom I had the privilege of working on the NOUS project. I am grateful for their constant availability and willingness to help.

To the friends I made during these stressful but unforgettable five years, especially Lia and Tomás, whom I worked with many times.

To my second family, JuniFEUP, where I met the most inspiring and hardworking people. Thank you, Vasco, Serra, Carol, Bruna, Inês, and many, many others, for sharing these unbelievable three years. Thanks to Dani, Duarte, Ribeiro, Rita, Renata, and Mário for this last year.

To a very special friend, António, who I shared this whole process with since Projeto FEUP until the Dissertation, it was a fantastic ride.

To my great “Férias” friends, Ângelo, Bessa, Chaves, Gonçalo, Iglésias, Melo and Sebastião, a great thanks and an apology for the weekends and birthdays I was not available.

To my family—my mom, dad, and brother (and my dog!)—who supported me unconditionally every single day. You saw my worst days, pushed me to keep going, and made all of this possible.

To Migui, thank you for all the patience and for never letting me give up.

Here, I say goodbye to FEUP, where I lived the best years of my life so far, and a new door opens; we will see where it takes me, but I am sure FEUP will forever be home.

Francisco Maldonado

“Open your mind, get up off the couch, move.”

Anthony Bourdain

Contents

1	Introduction	1
1.1	Context and Motivation	1
1.2	Problem Statement	1
1.3	Research Questions	2
1.4	Hypothesis	2
1.5	Planned Contributions	2
1.6	Objectives	3
1.7	Research Methodology	3
1.8	Thesis Structure	5
2	Background	6
2.1	Artificial Intelligence & Machine Learning	6
2.2	Human-Centered AI	6
2.3	Software Architecture	8
2.3.1	Principles of Software Architecture	8
2.3.2	Architectural Views and Models	9
2.3.3	Quality Attributes	9
2.3.4	Architecture Validation and Evaluation	9
2.4	The NOUS Project	9
2.4.1	Objectives and Vision	10
2.4.2	Core Technological Pillars	10
2.4.3	Architectural Overview	11
2.4.4	Use Cases	11
3	Literature Review & State-of-the-art	12
3.1	Introduction	12
3.2	Literature Review Process	12
3.3	Discussion	14
3.3.1	Design-time Pattern & Reference Architectures	14
3.3.2	Ethical Guardrails & Safety Layers	16
3.3.3	Interactive Learning & Feedback Loops	17
3.3.4	Operational Trust Dashboards & Federated Oversight Platforms	18
4	Human-in-the-loop Architecture	20
4.1	Introduction	20
4.2	NOUS Architecture Current State	20
4.3	Architecture Standardization	21
4.3.1	Conversion Process and Assumptions	22

4.3.2	Key Architectural Decisions and Assumptions	23
4.3.3	C4 Level 1: System Context	23
4.3.4	C4 Level 2: Container View	23
4.4	Design Drivers for HITL Integration	25
4.4.1	Design Principles	25
4.4.2	Architectural Attributes	25
4.5	HITL Architecture	30
4.5.1	Design problems addressed	30
4.5.2	HITL container view	30
4.5.3	Decision Guard	32
4.5.4	Expert Review Queue	33
4.5.5	Explainability Service	34
4.5.6	Feedback Manager	36
4.5.7	Cross-cutting stores and ledger links	37
4.5.8	End-to-end sequence	37
5	Validation	39
5.1	Introduction	39
5.2	Validation against Project Requisites	40
5.2.1	Functional Attributes Fulfilment	40
5.2.2	Non-Functional Attributes Fulfilment	41
5.3	Expert Review through Qualitative Interviews	45
5.3.1	Interview Method and Protocol	45
5.3.2	Data Collection	45
5.3.3	Key Findings	45
5.3.4	Adjustments Following the Interviews	46
5.4	Validation via Use-Case Mapping	47
5.4.1	Introduction & Scope	47
5.4.2	Main Scenario Synopsis	47
5.4.3	Actor-Container Mapping	48
5.4.4	End-to-End Sequence (HITL Overlay)	48
5.4.5	Coverage of Alternative Flows	51
5.4.6	Validation Verdict & Residual Risks	52
6	Conclusions & Future Work	53
6.1	Major Achievements	53
6.2	Future Work	54
A	Annex	60
A.1	Interview Protocol	60
A.1	Purpose of the Interview and Study Context	60
A.2	Interviewee and Interviewer Information	61
A.2.1	Interviewee	61
A.2.2	Interviewer & Bias Reflection	61
A.3	Ethical Considerations & Consent	61
A.4	Interview Guide	61
A.5	Closing the Interview	62
A.6	Post-Interview Steps (researcher-internal)	62

List of Figures

2.1	“Map of the field of Human-Centered AI,” reproduced from Capel and Brereton [8] (their Figure 1). Illustrating the four major quadrants of research.	7
3.1	Prisma 2020 process followed in this research	14
3.2	Article Distribution across the defined clusters	16
4.1	Original Conceptual NOUS Architecture Diagram	21
4.2	C4 Level 1 – System Context for NOUS (Baseline)	24
4.3	C4 Level 2 – Container View of NOUS (Baseline)	24
4.4	HITL containers integrated into the NOUS node (green = new).	31
4.5	Internal structure of the Decision Guard container.	32
4.6	Internal structure of the Expert Review Queue.	34
4.7	Internal structure of the Explainability Service.	35
4.8	Internal structure of the Feedback Manager.	36
4.9	End-to-end sequence diagram for a flagged decision requiring human review, showing interactions between the HITL containers.	37
5.1	Baseline CRIMSON message flow before HITL integration (steps S1–S10). . . .	48
5.2	Sequence diagram for Use-Case 3 with HITL intercepts.	50

List of Tables

1	Alignment of thesis outcomes with selected UN SDG targets.	iii
3.1	Inclusion/Exclusion Criteria for Literature Review	13
3.2	Cluster name and respective focus	15
4.1	Summary of Baseline NOUS Containers (C4 Level 2).	25
4.2	Core Design Principles for the HITL Architecture.	26
4.3	Design problems introduced by Human-in-the-Loop requirements.	30
5.1	Container-attribute coverage matrix (✓ = directly supports).	44
5.2	CRIMSON actors and events mapped to HITL containers.	49
5.3	UC-3 alternative scenarios and architectural responses.	51
6.1	Research-question coverage	53

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ATAM	Architecture Trade-Off Analysis Method
BDVA	Big Data Value Association
DLT	Distributed Ledger Technologies
DP	Design Problem
DSR	Design Science Research
EU	European Union
FR	Functional Requisite
GDPR	General Data Protection Regulation
HCAI	Human-Centered Artificial Intelligence
HITL	Human-in-the-loop
HPC	High-performance Computing
IaaS	Infrastructure as a Service
IML	Interactive Machine Learning
IoT	Internet of Things
ML	Machine Learning
MLaaS	Machine Learning as a Service
NFR	Non-Functional Requisite
PaaS	Platform as a Service
RQ	Research Question
SaaS	Software as a Service
SME	Subject Matter Expert
UI	User Interface
VM	Virtual Machine

Chapter 1

Introduction

1.1 Context and Motivation

Artificial intelligence (AI) now curates news feeds, pilots vehicles, triages medical cases, and brokers financial trades. Such automation amplifies human capability; at its worst, it can ossify hidden biases and generate opaque, unaccountable outcomes. Mainstream deployment has shifted the discussion from *whether* AI should be used to *how* it can be used responsibly and transparently. Two converging lines of evidence motivate this thesis.

First, software engineering research demonstrates that machine-learning (ML) components introduce unique forms of “*hidden technical debt*” that accumulate unless data dependencies, feedback loops, and configuration drift are explicitly managed [31]. Second, the human-computer interaction community argues that trustworthy AI must keep people meaningfully “in the loop” through intervention tools, monitoring dashboards, and interfaces that surface model uncertainty [33]. These academic insights are rapidly becoming mandatory practice: the EU Artificial Intelligence Act, for example, elevates “appropriate human oversight” in high-risk systems from a best practice to a legal obligation [13].

Against this backdrop, the *NOUS* project, an EU Horizon initiative federating cloud, edge, and high-performance computing resources, adopts *Human-Centered AI* as a core principle. NOUS seeks to guarantee the ethical advancement of automated processes by embedding *human-in-the-loop* (HITL) mechanisms across its architecture. This dissertation addresses the architectural implications of that ambition, proposing a concrete design to realise it.

1.2 Problem Statement

Integrating human judgment into distributed AI platforms is both necessary and challenging. Designers must reconcile two competing forces: the efficiency and scalability demanded by modern cloud–edge pipelines, and the responsive flexibility that allows humans to monitor, override, and improve algorithmic decisions. While existing literature offers isolated patterns, it stops short of a coherent, end-to-end architectural blueprint. Such a blueprint must show *how* these patterns

integrate under realistic performance, security, and data-protection constraints. The absence of a validated, holistic reference architecture for HITL in complex, federated systems is the principal gap this thesis aims to fill.

1.3 Research Questions

To guide this investigation, the following research questions are posed:

RQ1: What architectural components and interaction patterns are needed to effectively support human-in-the-loop strategies across heterogeneous applications like those in the NOUS ecosystem?

RQ2: How can a system architecture facilitate the continuous improvement of AI models based on human feedback and intervention, without compromising the system’s scalability or data privacy?

1.4 Hypothesis

Integrating responsible-AI components and patterns into the NOUS platform will yield a reference architecture that improves the system’s safety, transparency, and ethical alignment by enabling rich human-in-the-loop interactions without prohibitive performance penalties.

In essence, the hypothesis posits that architectural foresight is the key to making human-centered AI principles both practical and scalable, turning trustworthy AI from a concept into an engineering reality.

1.5 Planned Contributions

Four complementary contributions are proposed in response to the research questions and objectives outlined earlier:

C1: Comprehensive state-of-the-art analysis. A PRISMA-guided literature review distils the current body of knowledge on Human-in-the-Loop and responsible-AI architecture, identifying recurring patterns, open gaps, and emerging best practices.

C2: Standardised architectural notation for the NOUS platform. Leveraging the C4 model, the dissertation formalises the NOUS cloud–edge ecosystem in a consistent, tooling-friendly notation that clarifies component boundaries, responsibilities, and deployment contexts.

C3: Reference integration of HITL mechanisms. Building on the analysis, a modular blueprint embeds explainability services, decision-override points, and feedback channels directly into the NOUS control, data, and infrastructure planes, turning “meaningful human oversight” into an explicit quality attribute.

C4: Multi-faceted validation of the proposed architecture. The design is exercised in the CRIMSON crisis-management use case and evaluated through architect interviews, and step-by-step use-case mapping, yielding a risk catalogue and mitigation guidelines.

Together, these contributions transform the abstract goal of human-centered AI into a concrete, deployable, and verifiable architecture for large-scale federated systems.

1.6 Objectives

To test this hypothesis, the research pursues four primary objectives:

- (i) **Characterise** the design space for HITL in large-scale, distributed AI systems, identifying the key architectural levers that influence safety, transparency, and user agency.
- (ii) **Specify** a reference architecture that embeds monitoring hooks, override points, and feedback channels while respecting the scalability and data-protection constraints imposed by the NOUS platform.
- (iii) **Instantiate** key parts of this architecture via use-case mapping, exercising fundamental NOUS components.
- (iv) **Evaluate** the resulting instantiation through a multi-faceted approach, including expert walk-throughs and a formal architecture analysis, to assess its fitness for purpose.

1.7 Research Methodology

To construct and validate a solution to the identified problem, this thesis adopts the **Design Science Research (DSR)** methodology [18]. DSR is a research paradigm focused on generating knowledge by creating and evaluating novel, purposeful artefacts. This approach is exceptionally well-suited for this work, as its central goal is to produce a reference architecture to solve the practical problem of integrating human oversight into complex AI systems. The research process is structured according to the six-stage DSR model proposed by Peffers et al. [25], which is materialised in this thesis as follows:

- (1) **Problem Identification and Motivation.** The research problem and the solution's value are defined and grounded in the literature and the state of the art. This stage is addressed in this chapter and the next.
- (2) **Definition of Objectives.** A set of concrete objectives for the artefact is derived from the problem statement. These objectives form the basis for the architectural requirements and the subsequent evaluation criteria.

(3) **Artefact Design and Development.** This is the core constructive phase of the research. The process begins by selecting and synthesising established HITL patterns from the literature into a coherent architectural blueprint. To make the design concrete and communicable, it is translated from conceptual diagrams into the formal C4 model notation. This translation makes interface assumptions explicit and clarifies deployment boundaries. The design is not a single, static event; rather, it is refined through an iterative process, including weekly design clinics with the NOUS technical board, ensuring continuous feedback and early validation.

(4) **Demonstration.** The utility of the designed artefact is demonstrated by applying it to a concrete, practical scenario. The selected vehicle for this demonstration is the Crisis Management (CRIMSON) use case, which is introduced as part of the NOUS project overview in Section 2.4.4.

(5) **Evaluation.** The artefact is evaluated to measure how well it achieves its objectives. A multi-faceted evaluation strategy is employed to triangulate findings from several complementary techniques:

- **Validation against Requisites:** Both Functional and Non-Functional Requisites are restated and the rationale behind why they are being fulfilled is presented.
- **Architect Interviews:** The C4 architecture diagrams are presented to NOUS project architects in structured walkthroughs to assess the design's feasibility, scalability, and potential risks.
- **Use-Case Mapping:** A step-by-step walkthrough is conducted to map the operational flow of the CRIMSON use case (introduced in Section 2.4.4) onto the proposed architecture. This verifies that the design provides all the necessary functional hooks and interaction points required for a practical HITL workflow.

Feedback from these activities is then used to perform a final design refinement before the artefact is fixed for reporting.

(6) **Communication.** The final stage involves the dissemination of the research contributions. This dissertation serves as the primary vehicle for communicating the problem, the artefact, its design rationale, and its evaluation to the academic and practitioner communities.

It is important, however, to acknowledge the inherent limitations of this methodological approach within the scope of this project. The evaluation is confined to a single NOUS use case and a limited time frame, meaning longer-term challenges like model drift and the generalizability of the architecture to other sectors remain open questions. To bolster validity despite these constraints, the research relies on combining different methods (interviews, trade-off analysis, and use-case mapping) to corroborate findings. This approach mitigates, though does not eliminate, potential bias and strengthens the confidence in the evaluation's conclusions.

1.8 Thesis Structure

The remainder of this dissertation is structured to follow the research journey outlined above.

- **Chapter 2** provides the necessary background, reviewing the state of the art in human-centered AI, software architecture for ML systems, and existing HITL patterns.
- **Chapter 3** details the systematic literature review process undertaken to ground this research in the most recent and relevant scholarly work.
- **Chapter 4** presents the central contribution: the design and rationale of the proposed reference architecture. It details the new components, their interactions, and the design decisions made, using the C4 model for documentation.
- **Chapter 5** describes the evaluation of the architecture. It reports the findings from the architect interviews, and the use-case mapping, and discusses how the design addresses the core research questions.
- **Chapter 6** concludes the dissertation, summarising the contributions, acknowledging the limitations of the work, and proposing promising avenues for future research.

Chapter 2

Background

This chapter surveys the theoretical and technological landscape that underpins the research presented in this dissertation. It introduces the core concepts, frameworks and initiatives that later chapters will build upon and, in a final synthesis, maps these foundations to the thesis research questions and contributions.

2.1 Artificial Intelligence & Machine Learning

Artificial Intelligence encompasses computational methods that enable machines to perform tasks such as perception, reasoning and decision making by employing techniques from search, logic and statistical learning [30]. Over the past decade, AI systems have incorporated built-in mechanisms for interpretability and fairness, aligning model behaviour with ethical and sustainability objectives.

Machine Learning, a main branch of AI, focuses on algorithms that improve through data-driven experience. It spans supervised, unsupervised and reinforcement learning paradigms, underpinning applications from recommendation engines to medical diagnosis. Recent advances in deep neural architectures and automated training pipelines have markedly lowered the barrier to deploying robust, scalable ML solutions [19].

2.2 Human-Centered AI

Human-Centered AI (HCAI) has emerged as a critical paradigm that seeks to align the development and deployment of AI systems with human values and needs. Rather than focusing purely on technical performance, HCAI prioritises qualities like transparency, fairness, and human oversight. However, “human-centered” is a broad term that encompasses a wide range of research goals and methodologies.

To better understand this diverse field, this thesis adopts the framework proposed by Capel and Brereton in their comprehensive mapping of the HCAI research landscape [8]. As depicted in Figure 2.1, they organise HCAI research along two primary axes. The horizontal axis represents a

spectrum from work that is **AI Led** (focusing on algorithmic techniques) to work that is **Human Values Led** (focusing on ethics and societal impact). The vertical axis distinguishes between AI systems that are **Under Design** (developed in controlled, lab-like environments) and those that are **Situated in Use** (studied and interacting with real-world contexts).

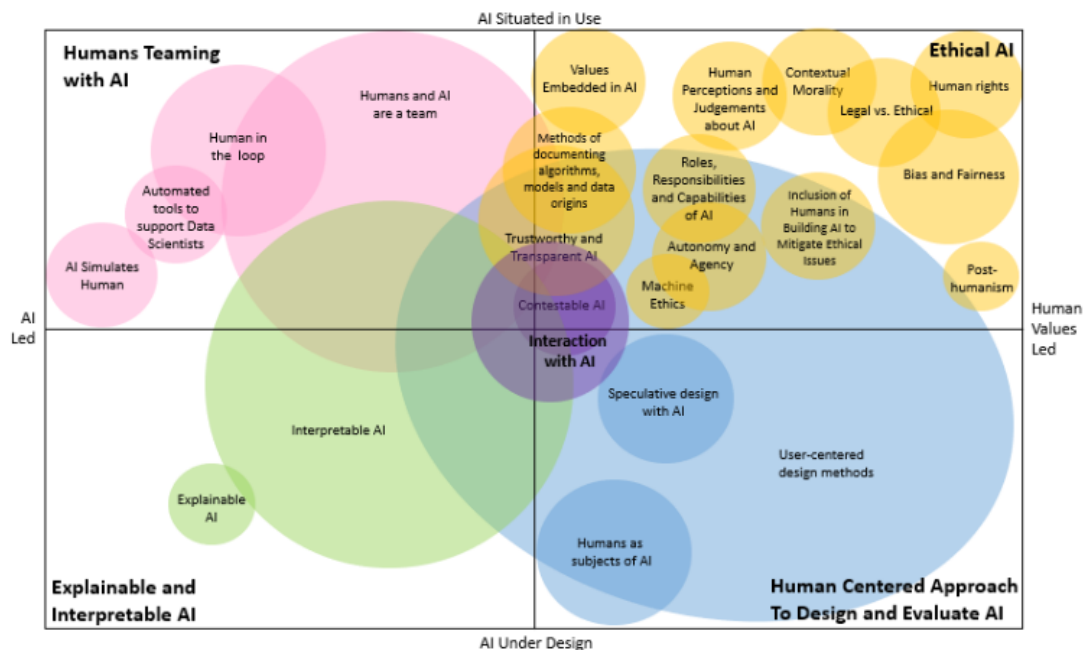


Figure 2.1: “Map of the field of Human-Centered AI,” reproduced from Capel and Brereton [8] (their Figure 1). Illustrating the four major quadrants of research.

These axes define four key quadrants of HCAI research:

- **Explainable and Interpretable AI:** This quadrant (lower-left) contains technically-focused work on making “black box” models more transparent. Explainability aims to reveal *how* a model reached a decision [27], while interpretability focuses on ensuring humans can comprehend and contest those explanations.
- **Human-Centered Design Approaches:** This quadrant (lower-right) focuses on applying established user-centered design methods to the creation and evaluation of AI systems, ensuring they are useful, usable, and aligned with user needs from the outset.
- **Humans Teaming with AI:** This quadrant (upper-left) explores the synergy between humans and AI systems. A particularly important implementation of this is the **Human-in-the-Loop (HITL)** paradigm. In a HITL system, human experts are not merely consumers of AI output but are integrated as essential components within the system’s operational cycle. They actively intervene to guide, correct, and validate the AI’s performance, especially in situations that are ambiguous, novel, or have high stakes.

The “loop” refers to the crucial feedback mechanism where human input, such as corrected labels, verifications of machine output, or rankings of generated content, is systematically

captured and fed back to retrain or fine-tune the AI model. This creates a continuous cycle of improvement, enabling the model to become more accurate, robust, and aligned with the expert’s nuanced understanding over time [3]. Common HITL patterns include:

- *Data Annotation*, where humans provide ground-truth labels for training supervised models.
- *Active Learning*, a more efficient approach where the system intelligently queries humans for labels only on the most uncertain or informative data points, thus maximizing the value of the expert’s time [32].
- *Verification and Override*, where a human must review and approve an AI’s proposed decision before it is executed, providing a critical safeguard in safety-critical applications.

Ultimately, HITL architecture aims to create a symbiotic relationship that leverages the scalability and computational power of machines alongside the contextual awareness, domain expertise, and common-sense reasoning of human experts.

- **Ethical AI:** This quadrant (upper-right) addresses the societal and ethical implications of AI. This includes research on algorithmic fairness to prevent discriminatory outcomes [16], as well as broader concerns about privacy, accountability, and the preservation of human autonomy.

The architectural work of this dissertation is situated at the intersection of these domains. It draws heavily from the **Humans Teaming with AI** and **Explainable and Interpretable AI** quadrants. The goal is not just to design an abstract algorithm, but to define a concrete software architecture that brings these principles into practice. Therefore, this work spans the vertical axis, creating an architecture that is conceived **Under Design** but is explicitly intended to support AI systems that are **Situated in Use**, enabling the robust, auditable, and human-steered AI that the HCAI paradigm demands.

2.3 Software Architecture

2.3.1 Principles of Software Architecture

Software architecture defines the fundamental organisation of a software system, encompassing its components, their interactions and the guiding principles for design and evolution [15]. It captures the major design decisions that determine overall structure, interfaces and behaviour and provides a common language for stakeholders. Core principles include the separation of concerns to partition responsibilities, high cohesion and low coupling to promote modularity, abstraction to hide internal details behind interfaces, and anticipation of change to accommodate future requirements [6].

2.3.2 Architectural Views and Models

Architecture is described through multiple views to address diverse stakeholder concerns. A practical and widely adopted approach is the C4 model, which organises architecture into four hierarchical diagrams: System Context, Containers, Components and Code, each offering an appropriate level of detail for different audiences [7]. This hierarchy ensures clarity when communicating architecture to both technical and non-technical stakeholders.

2.3.3 Quality Attributes

Quality attributes, or non-functional requirements, specify how well a system must perform rather than what it must do. Attributes such as performance, scalability, reliability, security and maintainability guide architectural decisions and are evaluated through metrics and scenarios to verify that system qualities are met [6]. Balancing these often competing attributes is a central challenge of architectural design.

2.3.4 Architecture Validation and Evaluation

Before committing to full implementation, an architecture must be treated as a hypothesis and subjected to rigorous evaluation. Two complementary methods are especially valuable:

- **Architect Interviews:** Structured interviews with the system’s architects and other key stakeholders elicit the rationale behind major design decisions, expose hidden assumptions and capture concerns that may not appear in documentation. These qualitative insights help evaluators to tailor scenarios and to uncover potential risks that purely document-driven methods might miss [10].
- **Use-Case Mapping:** Also called change-case analysis, this technique organises representative functional and quality attribute scenarios. By walking through each use case such as a peak-load transaction or a security breach simulation, evaluators can observe how the architecture accommodates real-world operations and anticipated changes, identifying both robustness and brittleness in the design [6].

2.4 The NOUS Project

The human-in-the-loop architecture proposed in Chapters 4 and 5 needs a demanding, multi-stakeholder environment to prove its worth. NOUS, an open, federated European cloud that spans edge devices, HPC clusters and emerging quantum resources, provides exactly that laboratory. The platform’s focus on data-sovereign, auditable workflows and its real-world crisis-management use case (see 2.4.4) supply the constraints, actors and performance targets against which the architecture is evaluated.

Funded under Horizon Europe (Call HORIZON-CL4-2023-DATA-01), NOUS aims to define and validate an open, federated European cloud platform that seamlessly spans edge devices, HPC infrastructures, and quantum resources.

2.4.1 Objectives and Vision

NOUS pursues three complementary objectives:

- **Federated Compute Fabric:** Integrate the European HPC network and quantum processors into a unified IaaS/PaaS offering, enabling dynamic offloading and co-execution of workloads across classical and quantum hardware.
- **Edge–Cloud Continuum:** Deploy decentralised compute agents on IoT and edge nodes to support local data processing, federated learning and real-time analytics, reducing latency and preserving data sovereignty.
- **Data Space Orchestration:** Provide robust components for data storage, exchange, standardisation and governance within European Data Spaces, leveraging DLT primitives, such as Blockchain, for auditability and policy enforcement.

These objectives align with Europe’s strategic need for digital sovereignty, fostering an open-source blueprint for next-generation cloud services.

2.4.2 Core Technological Pillars

NOUS is structured around three interoperable layers (see Fig. 4.1):

Compute Layer (#Compute)

- *HPC Integration:* High-throughput data transfer and secure interface with pan-European supercomputers.
- *Quantum Offload:* Runtime circuits generation, qubit encoding schemes and bidirectional I/O channels for quantum accelerators.
- *Security Overlay:* Darknet-based network segmentation and Zero-Trust modules for end-to-end confidentiality.

Edge Layer (#Edge)

- *Edge Agents:* Lightweight containers on IoT and mobile devices supporting federated learning and parallel skeleton execution.
- *Resource Orchestration:* Hierarchical load-balancing and policy-driven task placement across the Edge–Fog–Cloud continuum.

Data Layer (#Data)

- *Data Lifecycle Services*: Storage, metadata management, and blockchain-anchored audit logs for raw and processed datasets.
- *Auto-Standardiser*: Microservice translating domain-specific formats via active ML to harmonised schemas.
- *Data Exchange Fabric*: Control plane for discovery, contract negotiation and DID registry; data plane with adapters for physical and virtual sources.

2.4.3 Architectural Overview

At its core, NOUS delineates two principal participants (Data Space Participant and Infrastructure Participant) bridged by a suite of *Common Administration Services* (identity, policy, discovery, gateway). This modular decomposition ensures that the data remains under local governance while compute services scale elastically across heterogeneous resources. The NOUS architecture is analysed in further detail in 4.2.

2.4.4 Use Cases

To validate the platform, NOUS implements four exemplar scenarios:

1. *Connected Vehicle Perception (MASA)*: Fusion of roadside and in-vehicle camera feeds via edge inference and HPC-powered analytics.
2. *Energy Forecasting Lifecycle (PPC)*: Federated model training across smart meters, with auto-standardisation for IEC/SAREF datasets.
3. *Crisis Management (CRIMSON)*: Scalable data processing and real-time dashboards for civil protection systems.
4. *Scientific HPC Storage & AI*: High-performance archiving and explainable AI on large experimental datasets.

Chapter 3

Literature Review & State-of-the-art

3.1 Introduction

No matter the subject, before any work is developed, if the final result is intended to be of high quality and contribute to the existing knowledge gaps, the literature should be analysed and deeply understood.

Understanding the importance of reproducibility, comprehensibility, and bias in this process, a systematic review approach [35] was taken. This method can be defined as a research technique whose objective is identifying and collecting all available research in a particular field, to provide an unbiased and reliable synthesis of the existing evidence, more specifically, according to the PRISMA 2020 guidelines.

This chapter summarises a systematic literature review and presents its conclusions.

3.2 Literature Review Process

The first step in conducting a systematic review is to develop a set of search terms to query multiple databases for relevant literature. These keywords were organised into three distinct domains: **Artificial Intelligence**, **Software Architecture** and **Human-in-the-loop**. The resulting query is presented below.

(ai OR “artificial intelligence” OR ml OR “machine learning” OR algorithm*) AND (“architecture” OR “software architecture” OR “system architecture” OR “architectural design” OR “design principles” OR “architecture pattern” OR “architecture patterns” OR “architectural patterns” OR “design pattern” OR “design patterns” OR “reference architecture” OR “system design”) AND (“human-in-the-loop” OR “human in the loop” OR hitl OR “humans in the loop” OR “human oversight” OR “human feedback” OR “human intervention” OR “human agent” OR “human agents” OR “interactive machine learning” OR “human-AI collaboration” OR “human-AI teaming” OR “expert in the loop” OR “mixed-initiative” OR “active learning” OR “participatory design” OR “co-creation” OR “user-centered design” OR “human-centered AI” OR “human-centered artificial intelligence” OR

“human-centered machine learning” OR “human-centered ML” OR “responsible AI” OR “ethical AI” OR “trustworthy AI”)

An initial corpus of 1,159 records was retrieved from IEEE Xplore, ACM Digital Library and Scopus. Rayyan [26] was used both for automatic duplicate finding and removal, and for screening against the inclusion and exclusion criteria listed in Table 3.1. After assessing full-text access, 20 publications satisfied all eligibility requirements. The PRISMA flow diagram in Figure 3.1 depicts this step-by-step selection process, which yielded the final set of articles forming the state of the art in the field.

Phase	Inclusion/Exclusion Criteria
1	Search engine results after executing a search query
2	Date of publication from 2019 to 2025 (inclusive)
3	Published in English or Portuguese
4	Not duplicated publications
5	Not retracted publications
6	Not work-in-progress publications
7	Must be an Article, Conference Paper, or Proceeding
8	Must not be Surveys or Literature Reviews
9	Must not be Summaries of Workshops, Seminars, Lectures or Tutorials
10	Published in the field of “AI Systems”, “Human Centered AI”, “Human-in-the-loop” or “Software Architecture”
11	Not published in the field of “Model Architecture”, “Autonomous Systems” or “AI-powered Coding Assistants”

Table 3.1: Inclusion/Exclusion Criteria for Literature Review

The 20 selected studies were classified into four groups, each reflecting its dominant research focus; Table 3.2 outlines these clusters and their associated domains. The distribution reveals a pronounced skew towards architectures that channel human feedback directly into learning loops: 7 out of 20 articles fall into the Interactive Learning & Feedback Loops cluster, underscoring the community’s current emphasis on continuously adapting models rather than treating oversight as a static afterthought. Design-time Pattern & Reference Architectures is the second largest group (6 articles), showing strong foundational work on reusable blueprints, while Operational Trust Dashboards & Federated Oversight Platforms (5 articles) reflect growing interest in real-time monitoring at scale. Only two articles concentrate on Ethical Guardrails & Safety Layers, suggesting that formalised and provable veto channels remain an emerging research frontier compared with the more pragmatic concerns of learning efficacy and deployment operations.

The following subsections explore every cluster in detail, emphasising accomplishments, knowledge gaps, and proposed strategies that align with the goals of this work.

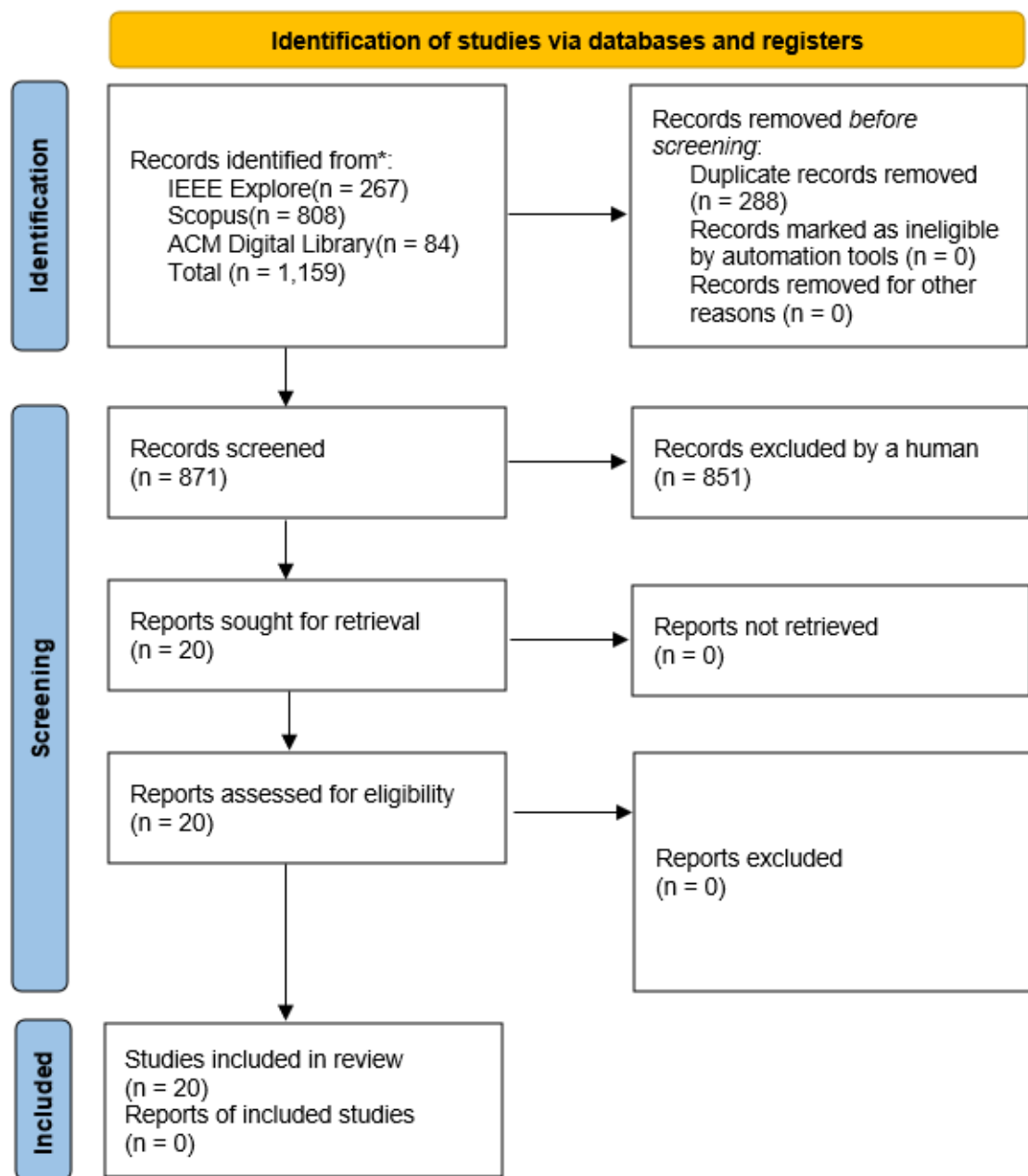


Figure 3.1: Prisma 2020 process followed in this research

3.3 Discussion

3.3.1 Design-time Pattern & Reference Architectures

Across the six contributions, a convergent thread is the elevation of human intervention from ad hoc UI widgets to first-class architectural elements planned at design time. **Andersen and Maalej's** catalogue [4] of ten Human-in-the-loop patterns, from Active Learning to Thumbs-Up/Down feedback, is laid out as a pipeline graph so architects can decide where domain experts inject labels, pause execution, or replay inference logs before any code is written. **Xu et al.** [42]

Design-time Pattern & Reference Architectures	Reusable catalogues and blueprints that spell out which architectural layers, components, and data flows are required to embed human-in-the-loop capabilities from the very first design decision, giving practitioners a common vocabulary across application domains.
Ethical Guardrails & Safety Layers	Middleware add-ons and layered "ethical firewalls" that enforce formal norms, trigger risk-based escalation, and guarantee a human veto in high-stakes contexts, ensuring AI actions remain within legal, ethical, and safety boundaries.
Interactive Learning & Feedback Loops	Architectures whose core logic directly channels human feedback into model updates through active learning, reinforcement learning from human feedback, or low-code interactive workflows enable continuous, data-efficient improvement without complete redeployment.
Operational Trust Dashboards & Federated Oversight Platforms	End-to-end service stacks that surface live trust metrics, audit trails, and kill-switches, often across edge-to-cloud or federated topologies, so operators and domain experts can monitor, override, or retrain models in production.

Table 3.2: Cluster name and respective focus

extend that idea with a three-layer HCAI framework whose "hybrid intelligence" and "meaningful human control" approaches supply meta-patterns such as an AI-mode switcher and a multimodel decision maker, guaranteeing override paths are wired into runtime contracts. **Usmani et al.** [38] refine those hooks at the interaction layer, insisting that explanation engines be embedded on the core service bus so every prediction is accompanied by a queryable "why/how", sustaining user agency rather than only dashboards.

At system scale, **Rožanec et al.** [29] propose an Industry 5.0 reference architecture that aligns BDVA layers with an Active Learning service, an Ethical Digital Twin, and a Continuous Ethical Validator; the twin simulates operator overrides while the validator enforces confidence thresholds, turning oversight into an architectural concern instead of an operational patch. **Lu et al.** [20] add governance scaffolding through their Responsible-AI-by-Design pattern collection (e.g., *Ethical Sandbox*, *Bill of Materials*, and *Co-versioning Registry*) so auditors can trace how and when

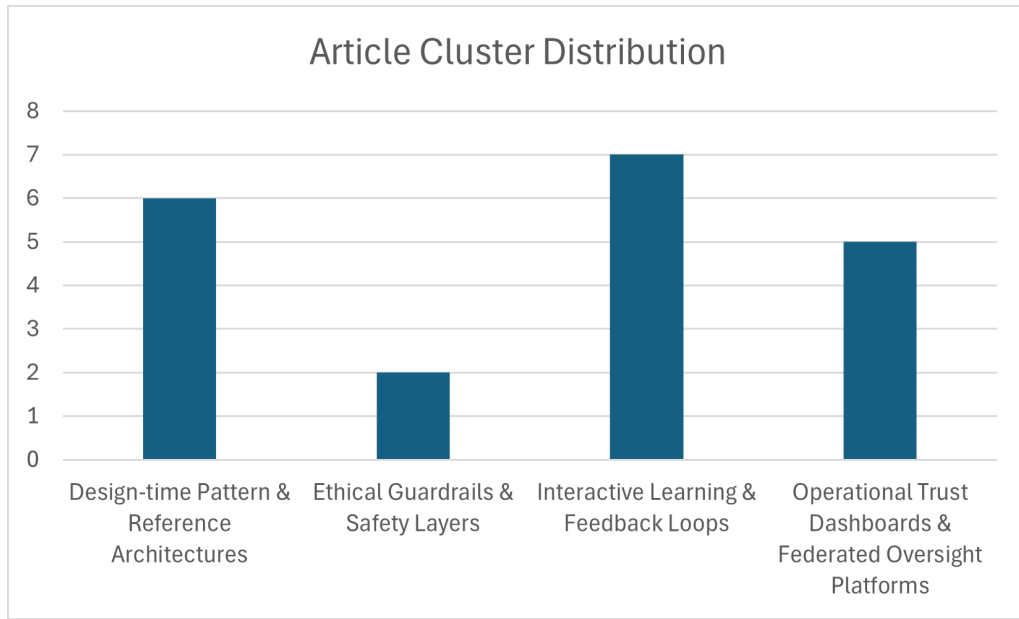


Figure 3.2: Article Distribution across the defined clusters

humans intervened across code, data, and model versions. The companion roadmap links these artefacts to multi-level ceremonies and sector-level standards, ensuring oversight patterns remain actionable throughout MLOps pipelines [21].

Collectively, these works push HITL from sporadic annotation to verifiable design primitives: pattern-encoded intervention points, mode-switching mediators, and twin-based simulation environments that let teams reason about latency, auditability, and risk allocation before deployment. Persistent gaps remain around cross-vendor interoperability and the cognitive load of maintaining rich provenance stores. Still, the direction is clear: responsibility in AI begins with design-time pattern libraries and reference models that keep humans where they matter: inside the control loop.

3.3.2 Ethical Guardrails & Safety Layers

Across both studies, architectural foresight is positioned as the key lever for translating HITL rhetoric into enforceable practice. **Comes *et al.*** [11] decompose crisis response AI into a 4 step control pipeline: sensing, analytics, choice and execution, and advocate "adjustable autonomy" sliders at every boundary so that fire-commanders, volunteers, or city officials can suspend, reinterpret, or override machine recommendations the moment uncertainty or moral overload spikes.

Thurzo [36], working in the medical education domain, hardens the same intuition behind a formal Ethical Firewall Architecture. Ethical constraints are encoded as temporal logic proofs, sealed on a cryptographic ledger, and patrolled by a Bayesian risk sentinel that either self-corrects the agent or escalates to a chartered Ethical AI Officer when proofs wobble.

Together, these articles sketch a defence-in-depth blueprint. A soft perimeter of contextual human sense-making wrapped around a hard core of code-level guardrails. Routine, low-risk micro decisions traverse the automated fast lane while ambiguous cases are siphoned into escalatory

channels calibrated to time pressure and resource scarcity.

3.3.3 Interactive Learning & Feedback Loops

Across the seven studies, architectural choices converge on a central principle: keeping the learning core close to the data while exposing explicit "hooks" for human intervention. These systems move beyond static models, creating continuous feedback loops that are increasingly integral to the architecture itself.

Component-based toolkits provide the foundational modularity for these loops. **Françoise et al.** [14] introduce *Marcelle*, which decomposes the entire Interactive Machine Learning (IML) workflow into reactive UI pipeline components. This compositional approach allows designers or clinicians to build custom interfaces by assembling blocks for data capture, training, and visualisation, enabling them to pause any stage and inspect features. Building on this, the architecture from **Alawadi et al.** [1] formalises this for a federated health context. It uses an *Active Learning broker* to escalate only low-confidence events from on-device models to a queue for remote medical experts. Their corrections are then federated back, creating a scalable, privacy-preserving architecture for expert-in-the-loop validation.

Other works focus on enhancing the core decision-making cycle with richer human input. The history-aware MAPE-K loop from **Parra-Ullauri et al.** [24] introduces a crucial architectural pattern for auditability, logging every self-adaptation step in a temporal graph store. This gives administrators explicit effectors to replay, query, and patch decisions, transforming opaque learning into a transparent, negotiated process. At a more granular level, **Wu et al.** [41] demonstrate a mechanism for real-time reinforcement learning where a Hug-DRL actor-critic directly fuses human steering overrides with policy gradients, achieving faster convergence by treating human input as a first-class signal in the learning process. In a similar vein, **Woods et al.** [40] wrap a core AI model (MuZero) inside an "Intelligent Decision Tool". This architectural wrapper provides a suite of human-facing services, including adversarial explanations and criticality gates, ensuring that operators are only prompted to intervene when their judgment is most valuable.

At the scale of industrial systems, these feedback loops must be both robust and resource-aware. The **Havrilla et al.** [17] *trIX* framework addresses this challenge for large language models. It provides a unified architecture to manage the significant memory and compute costs of Reinforcement Learning from Human Feedback (RLHF), abstracting the underlying complexity and making feedback loops feasible for massive models. Finally, the BICTSeMS intrusion detection blueprint from **Roponena et al.** [28] offers a practical example, coupling analyst approval gates with active learning filters. This design ensures that low-confidence alerts are automatically routed for semi-supervised retraining, feeding a curated playbook and closing the loop in a production security environment.

Collectively, these systems show a clear architectural evolution from sporadic annotation toward continuous, latency-aware feedback loops. They achieve this by externalising decision

provenance, logging interventions for auditability, and lowering the cognitive cost of human oversight through intelligent routing and explanation services. Yet common limitations persist: annotation fatigue in sustained guidance scenarios, domain-specific tuning of criticality metrics, storage overheads for history graphs, and the still-significant hardware footprint for large-scale RLHF. Addressing these architectural constraints, particularly in federated systems, remains the frontier for state-of-the-art HITL research.

3.3.4 Operational Trust Dashboards & Federated Oversight Platforms

Across the five studies, a convergent idea emerges: production AI now ships with a second control plane, an always-on "trust dashboard" that instruments every inference path and exposes levers for federated human oversight. These architectures provide continuous, in-situ governance, moving beyond sporadic, post-hoc audits.

A common pattern is the use of a microservice-based gateway to wrap the core AI functionality. **Cerulli et al.** [9] exemplify this by augmenting a Clinical Decision Support System gateway with dedicated *Situation-Identification* and *Model-Management* microservices. Each service emits confidence scores and patient-level provenance, enabling clinicians to interrogate, relabel, or roll back predictions before committing to a decision. Similarly, **Ottun et al.** [23] propose *SPATIAL*, a microservice mesh where virtual AI-sensors feed a central trust index. This architecture allows human operators to manage robustness-privacy trade-offs live, with every action recorded on an immutable provenance ledger for accountability.

This dashboard concept extends naturally to distributed and federated environments. **Alberti et al.** [2] apply it to an edge-to-cloud continuum, where every containerised deployment registers with a "knowledge cockpit." This cockpit provides operators with widgets that track model drift, energy consumption, and override rates, enabling factory technicians to hot-swap faulty models in under 90 seconds. At the city scale, **Utomo et al.** [39] introduce a *Federated Trustworthy-AI broker*. This architectural component acts as a governance gatekeeper, withholding global model aggregation until local dashboards confirm that fairness and bias metrics are within pre-set municipal thresholds, thereby embedding human agency directly into the federated consensus protocol.

Finally, these platforms are designed with role-specific interfaces to manage cognitive load. In the clinical auditing realm, **Barnard et al.** [5] extend the MLighter tool with a *Medical-MLOps loop*. SHAP-based explainers stream data to a role-specific panel, presenting concise "traffic-light" alerts for ward staff while providing deep logs for QA scientists. This allows frontline users to flag incoherent episodes, which in turn automatically triggers pipeline tests, closing the loop between operational oversight and model maintenance.

Collectively, these architectures demonstrate a clear shift from periodic auditing to continuous, multi-tenant governance. Trust metrics are sensed in-situ, surfaced through role-aware dashboards, and federated across organizational boundaries before automatic retraining is permitted. Harmonizing these diverse trust signals, preventing dashboard fatigue, and formalizing the escalation logic for intervention remain key open research frontiers for operational HITL platforms.

Conclusion As this review shows, human-centered AI cannot simply be added to an existing pipeline; it must be embedded as a set of first-class architectural concerns. Across the 20 recent studies we see four common patterns: (i) design-time pattern libraries that articulate intervention hooks before any code has been written, (ii) interactive learning loops that allow the low-confidence cases to be routed back to humans without causing downtime, (iii) operational trust dashboards that surface live risk & drift metrics and allow for hot swaps, and (iv) formal guardrail layers combining (provable) constraints with autonomy sliders. All four patterns address RQ1 and RQ2 by showing how modular services (e.g., ethical digital twins, active-learning brokers) can work together to keep humans in the decision loop at sub-second latency. Closing these gaps, particularly in federated, edge-to-cloud systems like NOUS, will be essential for turning principled HCAI blueprints into potential, auditable, scalable platforms.

Chapter 4

Human-in-the-loop Architecture

4.1 Introduction

This chapter details the design of a reference architecture for integrating HITL mechanisms into the NOUS platform. The objective is to present a concrete and extensible architectural solution for human-centered AI systems.

The chapter is structured as follows. First, Section 4.2 reviews the current state of the NOUS architecture to provide context. Section 4.3 describes the standardisation of this architecture, translating it into the C4 model to create a clear and layered baseline. Next, Section 4.4 defines the Design Drivers for HITL integration, outlining the core design principles and key quality attributes (non-functional requirements) the final design must address. Finally, Section 4.5 presents the proposed HITL Architecture, detailing the new components and explaining how they integrate into the system to enable effective human oversight.

4.2 NOUS Architecture Current State

To design and integrate HITL mechanisms, it is first essential to establish an unambiguous understanding of the existing system architecture. The NOUS project is defined by a comprehensive architectural vision, depicted in Figure 4.1.

This diagram provides a valuable, high-level overview of the NOUS ecosystem. It successfully identifies the major conceptual blocks, such as the *Data Space Participant*, *Data Space Infrastructure Participant*, and the *Common Administration Services* that bridge them. It also illustrates the vast array of services involved, from data persistence and exchange to infrastructure provisioning, including advanced capabilities like federated learning, HPC, and quantum computing.

However, while conceptually rich, this diagram presents several challenges for detailed architectural work:

1. **Lack of Standard Notation:** The diagram uses a custom, bespoke notation. While effective for initial project communication, it lacks the formal semantics and clear conventions of

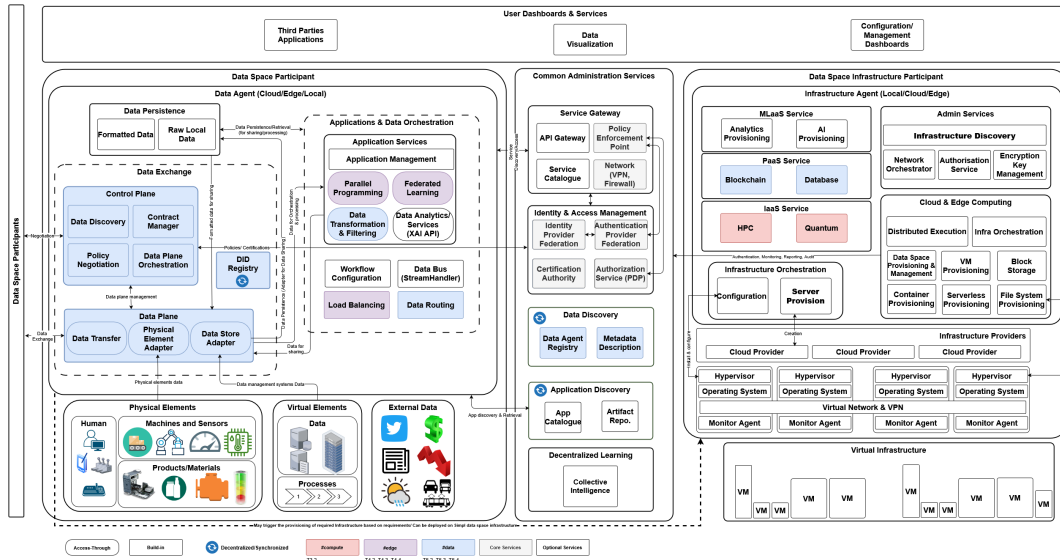


Figure 4.1: Original Conceptual NOUS Architecture Diagram

established architectural modelling methods. This ambiguity makes it difficult to reason precisely about component boundaries, responsibilities, and interfaces.

2. **Mixed Levels of Abstraction:** The diagram combines very high-level concepts (e.g., *Data Space Participants*) with low-level implementation details (e.g., VMs and Hypervisors) within a single view. This mixing of abstractions makes it challenging to understand the system from a specific stakeholder's perspective, such as a developer who needs to integrate a new service or an operator who needs to understand deployment units.
3. **Unclear Integration Points:** For this work, it is challenging to identify the precise points where HITL mechanisms could be integrated. The diagram does not explicitly show the runtime relationships or data flows between deployable software elements in a way that supports this kind of detailed design.

To overcome these limitations and create a solid foundation for architectural extension, a more structured representation is required. The **C4 model**, as introduced in Section 2.3.2, provides a solid solution. Its hierarchical approach allows for a progressive zoom from a high-level system context down to specific containers and components using a simple, clear, and consistent notation.

Therefore, the following section details translating the NOUS architecture into the C4 model. This standardisation will provide the clarity and precision necessary to effectively design, communicate, and reason for integrating new Human-in-the-Loop capabilities.

4.3 Architecture Standardization

As established, a standardised architectural model is necessary to provide a clear baseline for HITL integration. This section details the process and results of converting the conceptual NOUS

diagram into the C4 model notation. This translation provides a concrete, layered view of the system, focusing on its key structural elements and their interactions, which is essential for the subsequent design of HITL mechanisms.

4.3.1 Conversion Process and Assumptions

The conversion from the original diagram to the C4 model was not a direct one-to-one mapping. It required interpretation to distil deployable software systems and containers from the conceptual blocks presented in the source material. The process was guided by three primary sources:

- The original NOUS architecture diagram (Figure 4.1) for the overall structure and component inventory.
- The official documentation for the Simpl-Open architecture [34], which provides foundational concepts for European Data Spaces upon which NOUS is built.
- The specific requirements and workflows detailed in the NOUS project proposal and use case documentation, which helped clarify the runtime interactions between different services.

All diagrams were authored in PlantUML using the open-source `c4-plantuml` library. The source files are version-controlled alongside this thesis and are rendered automatically during the $\text{\LaTeX}$ build process, guaranteeing synchronisation between the documented architecture and its source code.

The conversion followed a systematic, five-step process:

1. **Identify System Boundary:** The “NOUS Data-Space Node” itself was defined as the single *System of Interest* for the C4 model at Level 1.
2. **Extract Logical Groupings:** At Level 2, the high-level blocks from the original diagram, specifically the “Data Agent” and “Infrastructure Agent”, were re-cast as C4 *Container Boundaries* to serve as logical groupings for related containers.
3. **Map Services to Containers and External Systems:** Individual services within the original diagram were modelled as distinct C4 *Containers*. Key foundational services like the DLT/Ledger, which serve a cross-participant trust function, were modelled as *External Systems*.
4. **Define Relationships:** Arrows from the source diagram were translated into explicit, labelled C4 relationships, clarifying the purpose and direction of interactions.
5. **Iterate with Stakeholders:** The resulting C4 models were validated for fidelity and terminological accuracy in two design-review sessions with the project supervisors.

4.3.2 Key Architectural Decisions and Assumptions

The translation process required making several explicit architectural decisions (D) and assumptions (A) to create a coherent model.

D1 – Single System Boundary At Level 1, the **NOUS Data-Space Node** is modeled as a single, opaque system. This adheres strictly to C4 principles by focusing on the system’s external interactions before revealing internal structure at Level 2.

D2 – Agent Boundaries as Logical Groups At Level 2, the *Data Agent* and *Infrastructure Agent* are treated as logical C4 container boundaries, not as monolithic applications. This reflects a microservices architecture and aligns with the Simpl-Open concept of a modular agent [34].

D3 – Ledger as an External System The **DLT / Ledger** is modelled as an external system that the NOUS node interacts with. This decision reflects its likely real-world deployment as a shared, consortium-level service that provides a common trust fabric for all participants, rather than a service owned and operated by a single node.

A1 – Collapsed Runtimes For clarity at Level 2, edge and cloud execution environments are abstracted into a single `Distributed Execution Runtime` container. Specific deployment variants will be detailed later as part of the HITL architecture design.

4.3.3 C4 Level 1: System Context

The System Context diagram is the highest level of abstraction, illustrating how the NOUS platform fits into its operating environment. It shows the platform as a single system, surrounded by the user roles (personas) and external systems that interact with it. This view is crucial for establishing the system’s scope and its primary boundaries. Figure 4.2 presents the System Context for the baseline NOUS platform.

This view establishes the system’s operational scope. Two primary personas interact with the node: an **End-User** (such as a data scientist) and an **Ops-Engineer**. The node exchanges data and negotiates contracts with peer **Data-Space Participants** and provisions hardware from external **Infrastructure Providers**. This high-level view clarifies the system’s boundaries and external dependencies.

4.3.4 C4 Level 2: Container View

Zooming inside the system boundary, Figure 4.3 decomposes the NOUS node into its major containers and shows its interaction with the external DLT service. The containers are grouped under the logical *Data Agent* and *Infrastructure Agent* boundaries.

This view reveals the main deployable units of the system, summarised in Table 4.1.

This standardised container view clarifies that key components like the `Control Plane` and `App Orchestration` write attestations to an external **DLT / Ledger**, reinforcing the system’s

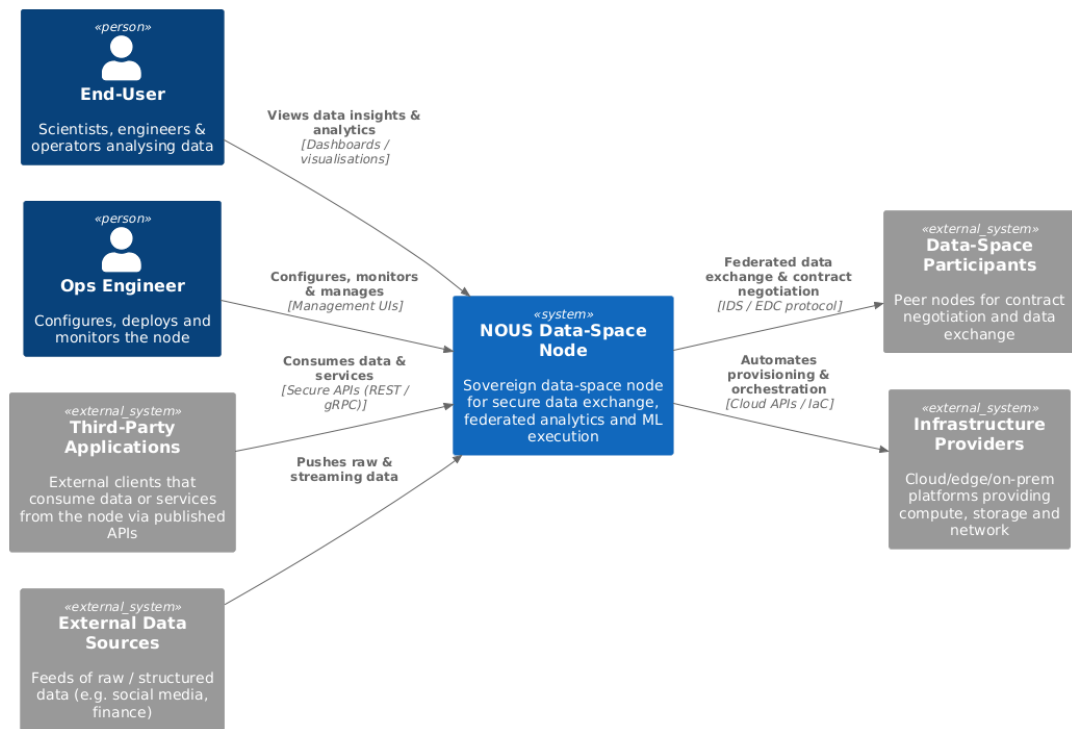


Figure 4.2: C4 Level 1 – System Context for NOUS (Baseline)

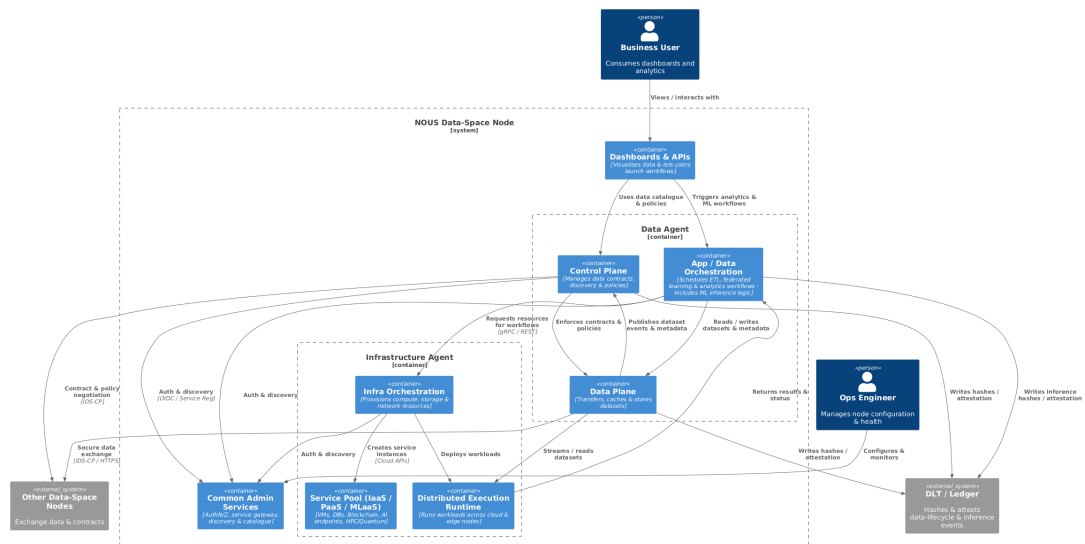


Figure 4.3: C4 Level 2 – Container View of NOUS (Baseline)

Table 4.1: Summary of Baseline NOUS Containers (C4 Level 2).

Container Name	Primary Functionality
Dashboards & APIs	Primary user-facing interface for configuration, monitoring, and future HITL oversight tasks.
Control Plane	Manages data discovery, policies, and contracts; aligns with Simpl-Open’s control plane concept [34].
Data Plane	Handles the secure, policy-governed transfer and caching of datasets.
App / Data Orchestration	Schedules complex workflows, such as ETL pipelines and federated learning jobs.
Infra Orchestration	Provisions and manages resources from the Service Pool.
Distributed Execution Runtime	Abstraction layer over compute resources (e.g., Kubernetes, VMs) that executes orchestrated workloads.
Common Admin Services	Provides cross-cutting functionalities including AuthN/Z, a service catalogue, and an API gateway.
Service Pool	Represents the pool of provisioned IaaS/PaaS/MLaaS resources (e.g., VMs, HPC nodes, Quantum accelerators).

reliance on a shared trust infrastructure. It provides the clear, actionable baseline required for integrating the HITL architecture.

4.4 Design Drivers for HITL Integration

Before extending the baseline architecture with HITL capabilities, it is crucial to establish the set of principles and requirements that will guide the design. These “design drivers” ensure that the resulting architecture is not only functional but also robust, secure, and aligned with the core goals of human-centric AI. This section first outlines the fundamental design principles that influenced the architectural approach, followed by a detailed list of the functional and non-functional attributes the system must satisfy.

4.4.1 Design Principles

The following core principles were defined to guide the high-level design choices for integrating HITL mechanisms into the NOUS architecture. They are presented in Table 4.2.

4.4.2 Architectural Attributes

In addition to the guiding principles, a set of specific architectural attributes (requisites) was defined to ensure the proposed architecture addresses its identified goals. These were categorised into functional attributes (what the system does) and non-functional attributes (how the system behaves). These attributes will serve as the basis for the architectural validation in Chapter 5.

Table 4.2: Core Design Principles for the HITL Architecture.

Principle	Description & Rationale
Human Centricity	Prioritises meaningful human oversight and control, ensuring that human experts are the ultimate arbiters in the AI lifecycle. This aligns with the need for trustworthy AI as promoted by HCAI philosophies [33].
Modularity & Separation of Concerns	Structures the architecture into distinct, loosely coupled components and containers. This is essential for managing the complexity of HITL workflows and ensures the system is maintainable and extensible [6].
Security & Privacy by Design	Integrates security and privacy-preserving mechanisms from the outset. Given the federated nature of NOUS and the sensitivity of HITL interactions, secure communication channels and data protection are non-negotiable.
Transparency & Auditability	Ensures that all significant AI decisions, human interventions, and data flows are traceable. This is achieved through comprehensive logging and provides a clear, immutable record required for debugging, accountability, and building trust.
Efficiency & Scalability	Designs a system that can accommodate a growing number of users, data volumes, and HITL interactions without a degradation in performance. This involves supporting asynchronous workflows and scalable components to avoid bottlenecks.
System Interoperability	Promotes interoperability not only between internal components but also with the broader NOUS ecosystem and other external data spaces. This ensures that the HITL capabilities can be leveraged within a federated environment.

4.4.2.1 Functional Attributes

The proposed HITL architecture shall support the following functional attributes:

FR.001: Lifecycle Oversight and Control Human experts must possess the capability to actively monitor, validate, and influence the AI model development and operational lifecycle within the NOUS ecosystem. This includes, but is not limited to, viewing model states and performance metrics, approving or rejecting model versions for deployment, triggering or guiding retraining processes with potentially modified parameters or data subsets, adjusting fairness constraints, and having clear interfaces to issue these control actions. *Motivation: A core tenet of the NOUS project is to incorporate Human-Centered practices [12], ensuring human experts are empowered with tangible control mechanisms. To achieve trustworthy AI, the architecture must provide tools for experts to be the ultimate arbiters in the AI lifecycle, particularly in complex data spaces, moving beyond simple passive monitoring.*

FR.002: Targeted and Efficient Human Expert Feedback for Model Refinement The system must support mechanisms to intelligently identify and prioritise data samples, model predictions, or operational events that would most benefit from human expert review and annotation. This involves routing these specific instances to the appropriate experts for feedback. *Motivation: The NOUS project is designed to handle “intensive and complex computing algorithms” on “vast amounts of data” [22]. To make human oversight practical and scalable within such a system, expert attention must be directed efficiently. This requisite addresses the need for optimising limited human resources by focusing their input where it has the highest impact on model performance, fairness, or ethical alignment, thereby promoting the “scientific excellence” [22] that is central to the project’s vision.*

FR.003: Comprehensive and Verifiable Auditability of AI Processes and Human Interventions All significant events within the AI lifecycle, including model training iterations, validation results, human expert decisions, policy changes, and data access logs, must be recorded in a secure, immutable, and verifiable manner. *Motivation: The NOUS project architecture is designed to “track the entire data’s lifecycle within a dataset” [22] to ensure transparency and monitor data flows. By using Distributed Ledger Technologies, it supports auditable and immutable records, which are essential for addressing legal challenges, ensuring GDPR compliance, and providing a transparent “ethical black box” for accountability and debugging.*

FR.004: Secure and Effective Integration of Expert-Curated Knowledge The system must provide secure and reliable pathways for human-provided labels, corrections, contextual insights, and other forms of feedback to be integrated back into the AI model training and refinement processes, both at the global federated level and for local model personalisation. *Motivation: To directly leverage human expertise to improve AI models beyond what can be learned from raw data alone, the NOUS architecture integrates innovative data-driven, edge*

computing, and federated learning services [12]. The quality and trustworthiness of these services depend on accurately incorporating human-validated knowledge, which enhances both the central models and personalised user experiences.

FR.005: Proactive Ethical Validation and Alerting for AI Systems The architecture must incorporate mechanisms for continuously monitoring AI models against predefined ethical guidelines, fairness criteria, and performance benchmarks. Deviations or potential risks must trigger timely alerts to relevant human experts. *Motivation: The NOUS project is committed to building a trustworthy data infrastructure that adheres to the European Commission’s “Ethics Guidelines for Trustworthy AI” [22]. This commitment requires more than post-hoc analysis; the system must be designed to proactively identify and flag potential ethical harms, biases, or significant performance degradations, enabling swift intervention to maintain the ethical integrity of its services.*

FR.006: Robust Reconciliation of Divergent Feedback When multiple operational users or domain experts supply annotations, overrides, or approval signals within the same workflow, the system shall record provenance, detect inconsistencies, and apply configurable reconciliation strategies, such as majority voting, weighted trust scores, or role-based precedence, to ensure the knowledge fed back into the AI models remains coherent, auditable, and reliable. *Motivation: In live deployments (e.g. crisis-response centres, energy-grid control rooms, or connected-vehicle fleets) users with different roles and situational perspectives may evaluate the same AI output simultaneously. Their feedback can legitimately conflict. The architecture must therefore manage this heterogeneity at runtime so that disagreement strengthens, rather than degrades, model quality.*

FR.007: Real-Time Human Override The architecture must support synchronous intervention points that allow authorised experts to inspect and, if necessary, veto, an AI inference before the action is finalised (e.g. blocking a data transfer, stopping an automated actuation). *Motivation: A core requirement for Trustworthy AI is enabling meaningful human agency and oversight [12]. Safety-critical workflows, in particular, cannot rely solely on post-hoc review and require mechanisms for real-time intervention.*

4.4.2.2 Non-Functional Attributes

The proposed HITL architecture shall exhibit the following non-functional attributes. These have been directly derived and synthesised from the official NOUS project non-functional specifications [12], ensuring the design aligns with the platform’s core quality, security, and compliance goals.

NFR.001: Security & Privacy The system must implement a Zero Trust security model, ensuring data privacy and protection through robust encryption, strict access control, and secure communication channels. It must maintain data integrity, using mechanisms such as those provided by Distributed Ledger Technologies (DLT). *Motivation: Given the federated nature of NOUS and the potential sensitivity of data reviewed by humans, a defence-in-depth*

security posture is non-negotiable. This protects user privacy and data integrity throughout the HITL workflow, preventing unauthorised access or data leakage.

NFR.002: Trustworthiness & Traceability The system must ensure comprehensive auditability of all AI processes and human interventions. Every significant event must be recorded in immutable, version-controlled logs to guarantee traceability. This includes ensuring that AI practices are fair and avoid bias. *Motivation: To build operator trust and satisfy regulatory demands, the system's behaviour and the impact of human intervention must be transparent and verifiable. Traceability provides a concrete mechanism for accountability, debugging, and demonstrating compliance.*

NFR.003: Usability & Accessibility Interfaces for human experts must be intuitive and efficient, minimising cognitive load and expert fatigue. The system should provide comprehensive documentation and support resources, ensuring it is accessible to all intended users, including those with disabilities. *Motivation: The effectiveness of any HITL system hinges on the quality and efficiency of expert input. A seamless and accessible user experience maximises engagement and the accuracy of human-provided feedback.*

NFR.004: Performance & Scalability The HITL architecture must be performant and scalable, capable of handling growing volumes of data, users, and computational load with low latency and high throughput. This is especially critical in edge computing environments. *Motivation: To ensure that human-in-the-loop mechanisms do not become a bottleneck in the overall system, the architecture must remain effective and responsive as the operational scale of the NOUS ecosystem grows.*

NFR.005: Adaptability & Maintainability The architecture must feature a modular, extensible, and open-source design to facilitate easy upgrades, modifications, and community contributions. It should be supported by detailed logging and automated CI/CD pipelines to simplify maintenance. *Motivation: The fields of AI and HCAI are rapidly evolving. The architecture must be adaptable to accommodate new explainer algorithms, expert workflows, or ethical guidelines with minimal disruption, guaranteeing the long-term viability of the system.*

NFR.006: Interoperability & Compliance The system must ensure seamless data integration and interaction between different components and external data spaces through standardised APIs and data exchange protocols. It must demonstrate strict adherence to European regulations and standards, particularly GDPR. *Motivation: As a cornerstone of a European Data Space, the NOUS platform must not be a silo. The HITL capabilities must be interoperable by design, facilitating federated oversight while ensuring full and transparent compliance with the relevant legal frameworks.*

4.5 HITL Architecture

This section turns the design drivers of Section 4.4 into a concrete, deployable *HITL layer* for the NOUS node. It extends the “as-is” container baseline (Figure 4.3) with five new containers and two shared stores, all confined within the same *Participant Trust Zone* dashed boundary introduced in Figure 4.2.

4.5.1 Design problems addressed

Table 4.3 summarises the six concrete problems that arise when we introduce Humans-in-the-Loop. They were distilled from the functional (FR) and non-functional (NFR) requisites in Section 4.4. Each subsequent design element explicitly references the problems it tackles.

Table 4.3: Design problems introduced by Human-in-the-Loop requirements.

ID	Problem statement
DP-1	Actionable explainability & uncertainty. Humans need rationales, confidence scores, and error bounds to make informed interventions.
DP-2	Trustworthy bidirectional feedback. Subject-matter expert (SME) corrections must be versioned, durable, and auditable.
DP-3	Real-time vs. review-time. Some decisions may block for humans, others must not.
DP-4	Scalable human effort. SME time is scarce; the system must prioritise the highest-value items.
DP-5	Governance, security, provenance. Every override must be signed, least privileged, and GDPR-compliant.
DP-6	Safe continuous model integration. New labels must improve models without breaking reproducibility; rollback and A/B testing are mandatory.

4.5.2 HITL container view

Figure 4.4 embeds the five *green* HITL containers inside the existing Data-Agent / Infrastructure-Agent boundaries. Grey boxes show unchanged baseline containers for orientation.

To clearly distinguish between different types of interactions and address potential ambiguity, the diagram uses two distinct line styles. This visual separation is a deliberate architectural choice to highlight the system’s structure:

- **Solid lines** represent the **Primary Workflow**, showing the sequential, value-adding pipeline that processes the core data artefact from the initial AI inference to the final, human-validated feedback.
- **Dashed lines** denote **Supporting & Control Interactions**. These are essential but orthogonal activities, such as user-initiated commands, configuration lookups, resource management, and out-of-band audit logging.

This convention makes the primary data flow easy to follow while still showing the necessary dependencies and control signals that enable it.

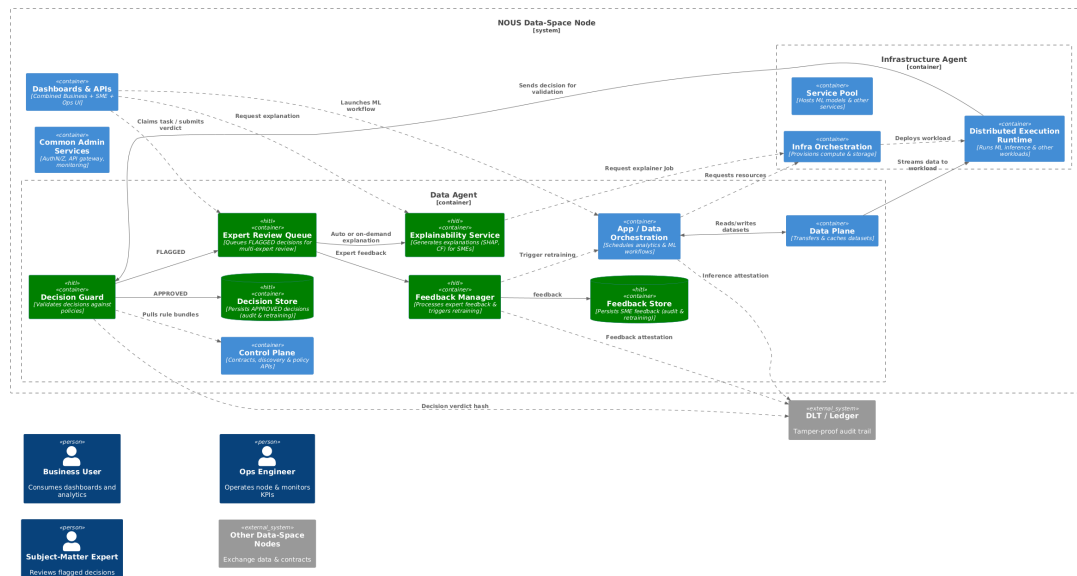


Figure 4.4: HITL containers integrated into the NOUS node (green = new).

The HITL architecture introduces five specialised containers and two data stores. They work in concert to create a robust, auditable, and scalable human oversight process. We explain their roles by following the path of an AI inference from execution to human feedback.

Decision Guard An AI inference first arrives at the **Decision Guard**. This container acts as a synchronous, real-time gateway. Its primary role is to enforce pre-defined policies (e.g., confidence thresholds, business rules) to separate high-confidence, automated decisions from ambiguous ones that require human scrutiny. This directly addresses the need to balance real-time performance with review-time analysis (**DP-3**) and provides an initial layer of governance (**DP-5**).

Expert Review Queue Decisions flagged by the Guard are sent to the **Expert Review Queue**. This container manages the asynchronous workflow for human experts. It buffers flagged items, prioritises them to ensure experts’ time is spent on the most critical or uncertain cases (**DP-4**), and orchestrates the collation of expert feedback.

Explainability Service To empower experts to make informed judgments, the **Explainability Service (XAI)** provides on-demand rationales for the AI’s decisions. When an expert reviews a task from the queue, this service can generate and provide explanations (e.g., SHAP values, counterfactuals). This is critical for actionable explainability (**DP-1**). By operating asynchronously, it avoids blocking the review process.

Feedback Manager Once an expert submits their verdict (e.g., correcting a label), the feedback is routed to the **Feedback Manager**. This is the final component in the loop. It is responsible for consolidating the expert feedback, ensuring its durability and provenance by logging attestations to the DLT Ledger, and triggering downstream actions like model retraining. This completes the loop, satisfying the need for trustworthy bidirectional feedback (**DP-2**) and safe continuous model integration (**DP-6**).

Cross-Cutting Stores and Ledger The entire process is supported by two new persistent stores: the **Decision Store** for an auditable log of approved decisions, and the **Feedback Store** for SME-provided labels. All critical events are anchored with cryptographic hashes to the external **DLT / Ledger**, creating an immutable provenance chain that underpins system-wide governance (**DP-5**).

4.5.3 Decision Guard

Figure 4.5 shows the internal components of the Decision Guard. The diagram follows the convention established in Section 4.5.2, with solid lines for the primary workflow and dashed lines for supporting interactions.

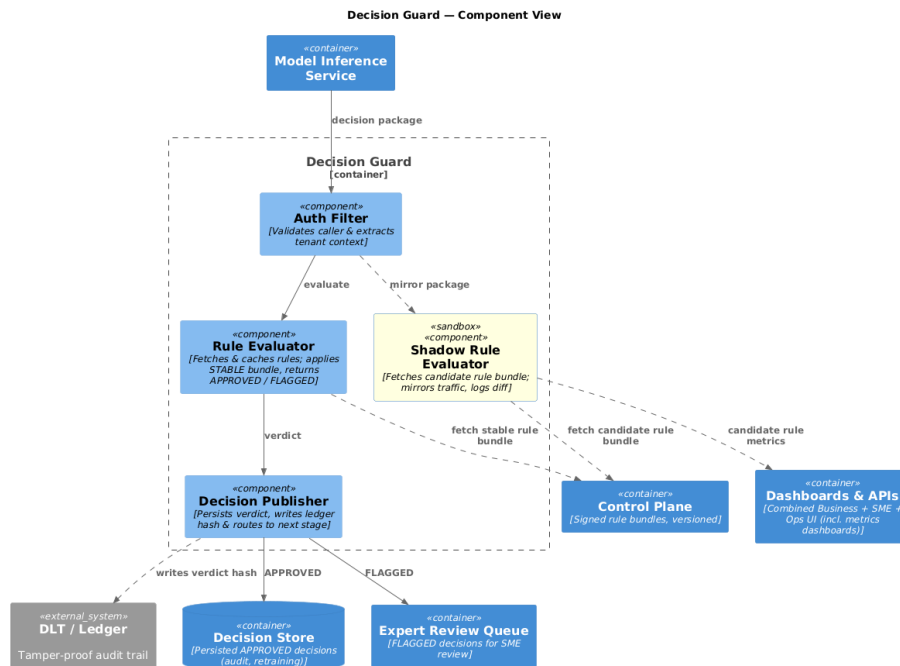


Figure 4.5: Internal structure of the Decision Guard container.

Role. First gateway for every inference; enforces synchronous policies and decides whether human review is required.

Internal anatomy

Auth Filter

Verifies caller identity and tenant context, dropping unauthorised requests at the perimeter.

Rule Evaluator

Applies the *stable* synchronous policy bundle retrieved from the Control Plane; returns either APPROVED or FLAGGED.

Shadow Rule Evaluator

Executes the *candidate* bundle on mirrored traffic, logging verdict deltas for safe A/B testing.

Decision Publisher

Persists a hash of every verdict on the DLT Ledger, writes APPROVED events to the Decision Store, and enqueues FLAGGED events in the Expert Review Queue.

Design-problem coverage. *DP-3* (real-time path) is satisfied by keeping the Auth Filter + Rule Evaluator chain synchronous, enabling FR-007's human override. *DP-5* (governance) is met through signed policy bundles and immutable ledger attestations, while the Shadow Evaluator contributes to *DP-6* (safe continuous evolution).

4.5.4 Expert Review Queue

Figure 4.6 shows the internal components of the Decision Guard. The diagram follows the convention established in Section 4.5.2, with solid lines for the primary workflow and dashed lines for supporting interactions.

Role. Buffers flagged items for human review, **ordering** them for SMEs based on a configurable **prioritisation policy**.

Internal anatomy**Task Broker**

Stores every FLAGGED decision and pre-fetches a draft rationale from the Explainability Service.

Task Prioritiser

Fetches, caches, and applies a pluggable **prioritisation policy** from the Control Plane. This component is responsible for **ordering** the work queue. By default, it can use a simple First-In, First-Out (FIFO) policy, which treats all tasks with equal priority. For advanced use cases, operators can deploy policies that use variables like model uncertainty or business impact to surface the most critical tasks first.

Consensus Engine

Aggregates multiple SME reviews into a single `FINAL_VERDICT` via majority vote (default) or future reputation-weighted voting.

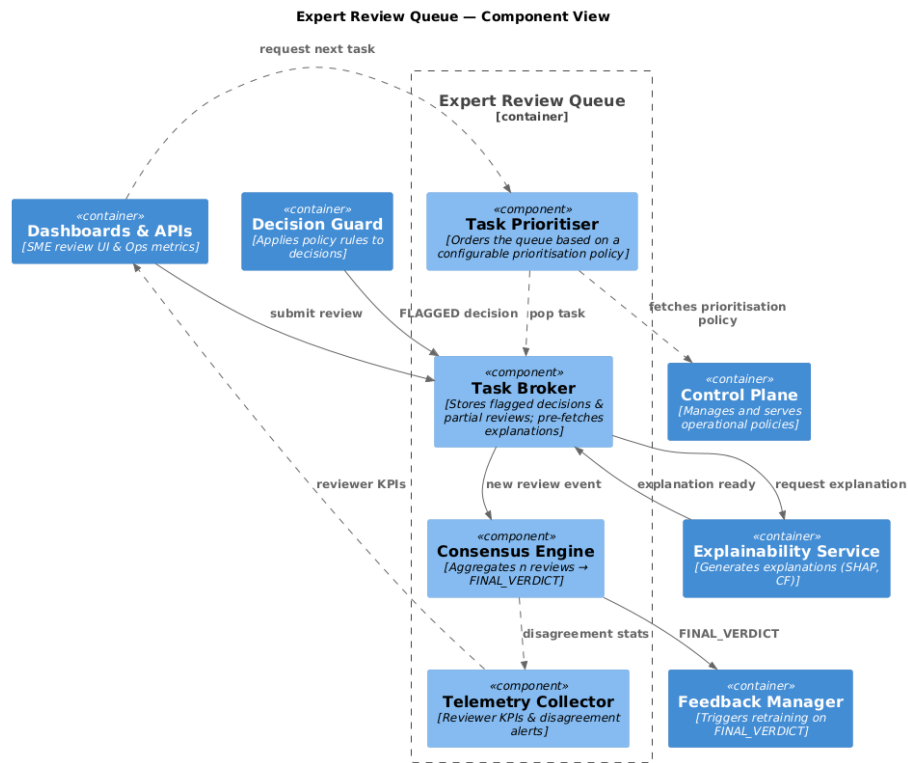


Figure 4.6: Internal structure of the Expert Review Queue.

Telemetry Collector

Publishes reviewer KPIs (latency, disagreement rate) to Ops dashboards and alerts.

Design-problem coverage. *DP-2* (durable bidirectional feedback) is met because every review is persisted immediately in the Broker and re-played if failures occur. *DP-4* (scalable human effort) is addressed by the configurable nature of the **Task Prioritiser**. In high-load scenarios, an operator can deploy a sophisticated prioritisation policy to ensure that limited human attention is focused on the most critical tasks.

4.5.5 Explainability Service

Figure 4.7 shows the internal components of the Decision Guard. The diagram follows the convention established in Section 4.5.2, with solid lines for the primary workflow and dashed lines for supporting interactions.

Role. Generates local and global explanations, uncertainty metrics, and recommended next questions for SMEs.

Internal anatomy

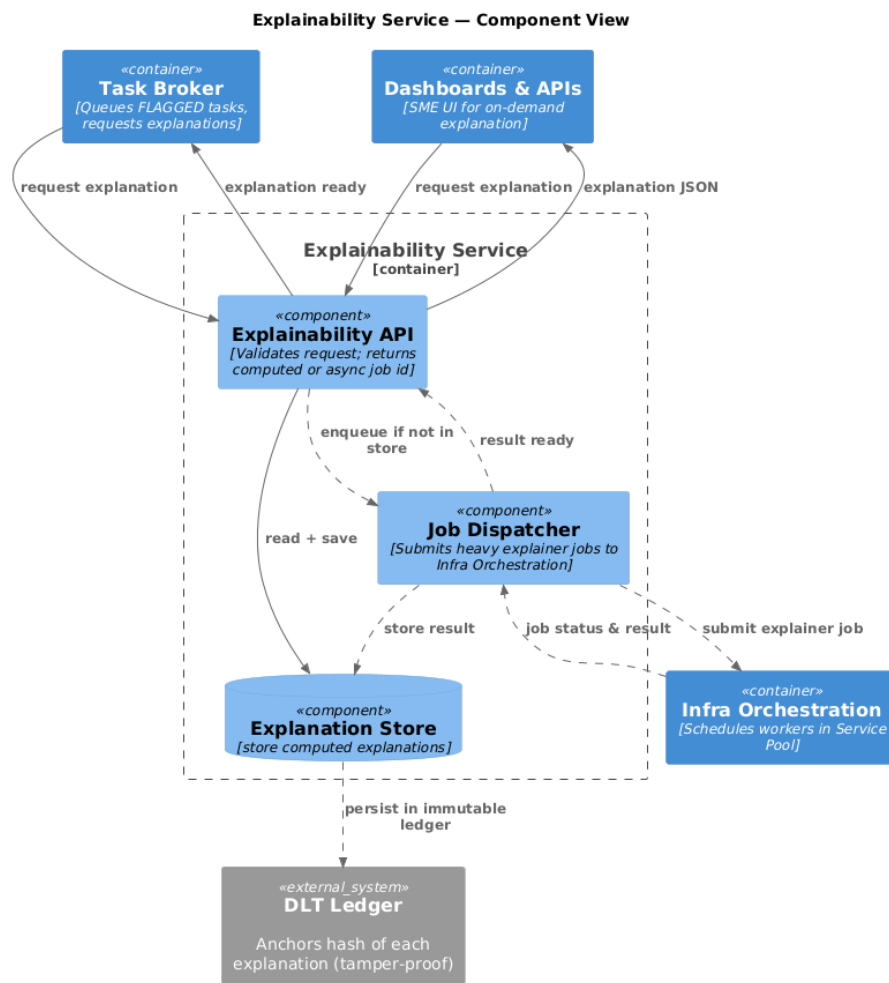


Figure 4.7: Internal structure of the Explainability Service.

Explainability API

Front door that validates quotas, serves computed rationales, or schedules a new explanation job.

Explanation Store

Stores explanations, aiming to reduce both computational resource usage and time needed to receive an explanation.

Job Dispatcher

Packages long-running SHAP/counterfactual jobs, submits them to Infra Orchestration, and streams status back to the API.

Design-problem coverage. Addresses *DP-1* (actionable explainability) directly; the Store+async path also reduces SME wait time, indirectly helping *DP-4* (scalable effort).

4.5.6 Feedback Manager

Figure 4.8 shows the internal components of the Decision Guard. The diagram follows the convention established in Section 4.5.2, with solid lines for the primary workflow and dashed lines for supporting interactions.

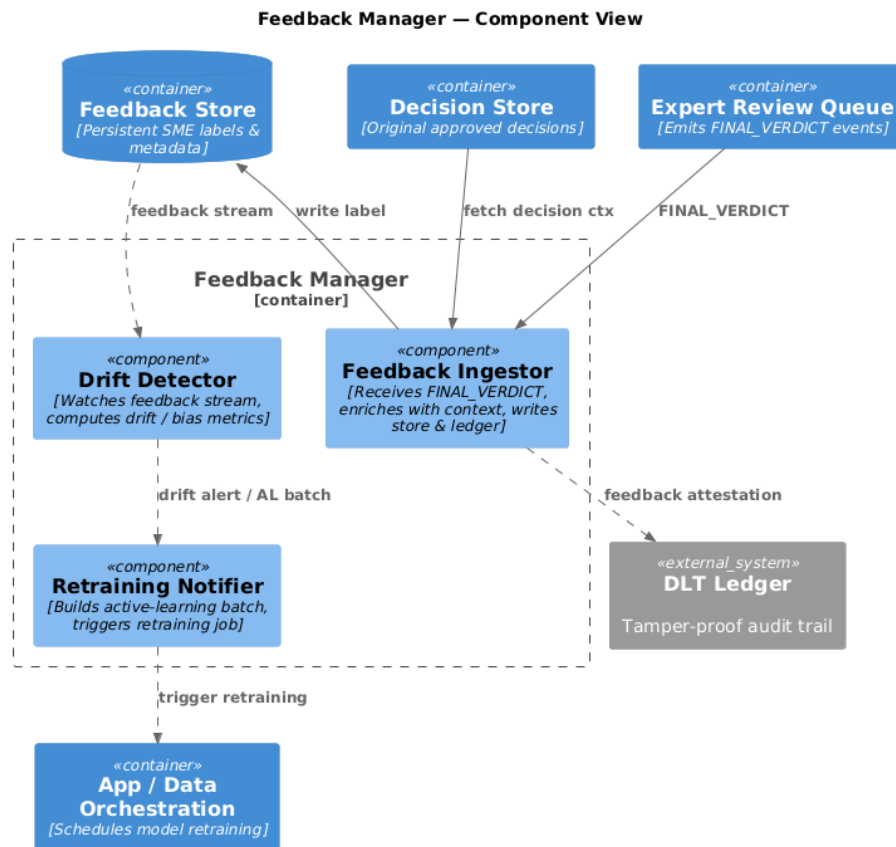


Figure 4.8: Internal structure of the Feedback Manager.

Role. Consolidates SME verdicts, updates the Feedback Store and issues retraining.

Internal anatomy

Feedback Ingestor

Enriches each `FINAL_VERDICT` with original decision context; writes to the Feedback Store and logs an attestation hash on the DLT Ledger.

Drift Detector

Streams new labels, computes drift or bias metrics against the Decision Store baseline, and raises alerts when thresholds are breached.

Retraining Notifier

Bundles the most informative samples (active-learning batch) and triggers a retraining workflow through App/Data Orchestration.

Design-problem coverage. *DP-2* (feedback durability) is met via the append-only Feedback Store; *DP-5* (provenance) via ledger hashes; *DP-6* (safe continuous model integration) through automated drift detection and controlled batch-size retraining triggers.

4.5.7 Cross-cutting stores and ledger links

- **Decision Store** – append-only log of APPROVED decisions, queried by auditors and the Feedback Manager.
- **Feedback Store** – mutable, versioned SME labels; primary source for drift detection and active-learning batches.
- **DLT Ledger** – hashes emitted by Decision Guard and Feedback Manager form an immutable provenance chain (DP-5).

4.5.8 End-to-end sequence

To illustrate how the designed components interact over time, this section describes the primary end-to-end workflow for a decision that requires human intervention. The sequence, depicted in Figure 4.9, begins after an ML model in the `Service Pool` has been executed by the `Distributed Execution Runtime` and ends after the expert’s feedback has been processed.

The process unfolds across three main phases: Decision Gating, Human Review, and Feedback Processing.

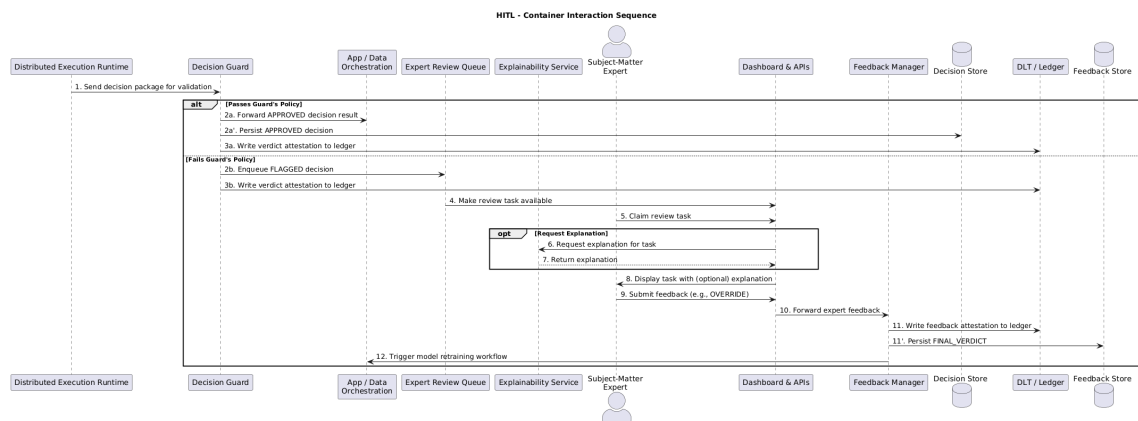


Figure 4.9: End-to-end sequence diagram for a flagged decision requiring human review, showing interactions between the HITL containers.

Phase 1: Decision Gating (Steps 1-3) The sequence is initiated when the `Distributed Execution Runtime` sends a decision package, including the inference result and its confidence score, to the **Decision Guard**. The Guard evaluates this package against a set of synchronous policies.

- If the decision is approved, the result is forwarded directly to the `App / Data Orchestration` to complete the workflow (the “happy path”, not fully detailed in this diagram).
- If the decision is flagged, it is sent to the **Expert Review Queue**.

In both cases, the `Decision Guard` writes an immutable cryptographic attestation of its verdict to the `DLT / Ledger`, fulfilling the auditability requirement (DP-5).

Phase 2: Human Review (Steps 4-9) Once a decision is enqueued, it becomes available for review by a SME. The SME interacts with the system via the **Dashboard & APIs** container, which acts as the user-facing interface for the HITL process. The expert can claim a task from the queue and, if necessary, request a rationale from the **Explainability Service** to make an informed judgment (DP-1). After evaluating the case, the SME submits their final verdict, such as an override with a corrected label, back through the dashboard.

Phase 3: Feedback Processing (Steps 10-12) The final feedback from the SME is not sent back to the queue but is routed to the **Feedback Manager**. This component is responsible for the final steps of the loop: it writes a verifiable attestation of the expert’s feedback to the `DLT / Ledger` (DP-2, DP-5) and then triggers a new workflow in the `App / Data Orchestration`, such as initiating a model retraining job with the newly validated data (DP-6). This closes the loop, ensuring that human insight is used to continuously and safely improve the AI system.

Chapter 5

Validation

5.1 Introduction

This chapter presents the validation of the HITL reference architecture proposed in Chapter 4. As a critical step in the DSR paradigm, the designed artefact must be rigorously evaluated to demonstrate its utility, soundness, and fitness for purpose. To ensure a comprehensive and unbiased assessment, this evaluation employs a multi-faceted strategy, triangulating evidence from three distinct validation streams. This approach not only verifies the design but also surfaces the critical architectural trade-offs inherent in balancing human oversight with system performance.

The validation is structured as follows:

1. **Validation against Project Requisites:** The architecture is first systematically checked against the full set of functional and non-functional requirements established in Chapter 4. This analysis provides a foundational verification that the design formally satisfies all its specified goals.
2. **Expert Review through Qualitative Interviews:** The design is then subjected to critique by senior NOUS architects who were not involved in its creation. This qualitative review serves to mitigate author bias, stress-test the architecture’s clarity and feasibility, and uncover potential risks and hidden assumptions.
3. **Validation via Use-Case Mapping:** Finally, the architecture’s dynamic behaviour is demonstrated by mapping it to the CRIMSON crisis-management use case. This walkthrough validates the end-to-end operational flow, including exception handling, and confirms the architecture’s viability in a realistic, high-stakes scenario.

Collectively, these methods provide robust evidence for the architecture’s soundness, delivering a conceptually validated blueprint for building the next generation of trustworthy, human-centered AI systems.

5.2 Validation against Project Requisites

The HITL layer has been woven into the NOUS node with one goal in mind: every design problem introduced in Table 4.3 must map back to at least one functional or non-functional requisite from Chapter 4, and the implemented mechanisms must close that gap in practice. The narrative below walks through the coverage in detail.

5.2.1 Functional Attributes Fulfilment

FR.001: Lifecycle Oversight and Control

- **Requisite** - Human experts need live insight into every lifecycle phase and must be able to steer training, deployment, rollback and retirement of models.
- **Fulfilment** - The *Dashboard & APIs* container is the single pane of glass for oversight (Figure 4.4). It surfaces state from the Decision Guard, Expert Review Queue and Feedback Manager, giving operators a unified view. Control actions flow back through signed API calls that the Auth Filter in the Guard validates before execution. Retraining or rollback commands are forwarded downstream by the Feedback Manager's *Retraining Notifier*.

FR.002: Targeted Expert Feedback

- **Requisite** - Scarce human attention must be focused where it improves model quality most, not wasted on low-value samples.
- **Fulfilment** - the *Decision Guard* forwards to the queue only those decisions whose confidence score falls below an adjustable threshold; high-confidence, low-risk items flow straight through the automated path. Inside the queue, the **Task Prioritiser** component refines the ordering based on a configurable **prioritisation policy**. By default, this can be a simple First-In, First-Out queue, but operators can deploy more advanced policies. This ensures that even within the low-confidence set, the most informative cases can be surfaced to the top, focusing scarce human attention where it is most valuable. Pre-fetching explanations from the XAI service ensures SMEs open every task with full context already at hand.

FR.003: Comprehensive Auditability

- **Requisite** - Every decision, policy change and human verdict must be provably traceable for years. Tampering must be detectable.
- **Fulfilment** - Two containers emit hashes to the external DLT: the Decision Guard records its APPROVED/FLAGGED verdict, while the Feedback Manager records each Final Verdict. The same hashes index stored payloads in the Decision and Feedback Stores, allowing auditors to replay the full chain.

FR.004: Secure Integration of Expert Knowledge

- **Requisite** - SME corrections must travel back into the training loop without exposure or loss.
- **Fulfilment** - The *Consensus Engine* ensures that only reconciled labels leave the queue. The Feedback Ingestor writes them to an append-only, versioned store and stamps a ledger hash, then packages an active-learning batch that the Retraining Notifier ships to App/Data Orchestration for the next run.

FR.005: Proactive Ethical Validation

- **Requisite** - The node must catch ethical breaches or unfair outcomes early, not after the fact.
- **Fulfilment** - Two tiers operate. The *Rule Evaluator* in the Decision Guard applies hard synchronous constraints. In parallel, the Drift Detector in the Feedback Manager streams fresh labels and raises alerts when bias metrics exceed thresholds. Shadow evaluation of candidate rule bundles provides a safe playground for tighter policies.

FR.006: Conflicting Expert Feedback Handling

- **Requisite** - When SMEs disagree, the system must converge on a reliable label, preserving the dissenting opinions for review.
- **Fulfilment** - The Consensus Engine gathers multiple reviews, chooses the outcome via majority vote (or configurable quorum) and stores both the final verdict and the individual votes. Disagreement statistics surface on the dashboard so that governance teams can refine policies or reviewer pools.

FR.007: Real-Time Human Override

- **Requisite** - Authorised experts must be able to veto a live inference before downstream action occurs.
- **Fulfilment** - When the Guard flags a decision, processing halts while the item sits in the queue. SMEs can *override* the decision via the dashboard, which triggers an immediate response path that either blocks the action or substitutes the SME-provided decision.

5.2.2 Non-Functional Attributes Fulfilment

The non-functional quality attributes define how the HITL layer must behave rather than what it must do. Each attribute, derived from the official NOUS project specifications, is examined here in terms of (a) its underlying intent and (b) the concrete architectural mechanisms, containers, or workflows that realise it, as depicted in the architecture diagrams.

NFR.001: Security & Privacy

- **Requisite** - The system must implement a Zero Trust security model, ensuring data privacy and protection through robust encryption, strict access control, and secure communication channels.
- **Fulfilment** - The design achieves this through a layered approach:
 - *No New Ingress Points.* All external traffic is routed through the existing **Common Admin Services** container, which includes the API Gateway. The HITL containers are not directly exposed, limiting the attack surface as shown in [Figure 4.4](#).
 - *Strict Authentication & Authorisation.* The **Auth Filter** component within the **Decision Guard** intercepts every incoming inference request, validating caller identity and tenant context before any processing occurs ([Figure 4.5](#)). This enforces least-privilege access at the perimeter of the HITL workflow.
 - *Immutable Audit Trail.* Both the **Decision Guard** and the **Feedback Manager** write cryptographic attestations of their actions to the external **DLT / Ledger**, creating a tamper-proof record for security audits, as detailed in the end-to-end sequence ([5.2](#)).

NFR.002: Trustworthiness & Traceability

- **Requisite** - The system must ensure comprehensive auditability of all AI processes and human interventions. Every significant event must be recorded in immutable, version-controlled logs to guarantee traceability and fairness.
- **Fulfilment** - Traceability and trust are woven into the data flow:
 - *End-to-End Provenance Chain.* The sequence begins with an inference, is validated by the **Decision Guard**, reviewed via the **Expert Review Queue**, and concluded by the **Feedback Manager**. Each step logs a hash to the **DLT / Ledger**, creating a verifiable chain connecting the initial data to the final human verdict.
 - *On-Demand, Asynchronous Explanations.* The **Explainability Service** provides rationales (e.g., SHAP values) to build SME trust. Its **Job Dispatcher** offloads heavy computation, preventing blocking and ensuring the UI remains responsive ([Figure 4.7](#)).
 - *Conflict Resolution.* The **Consensus Engine** within the **Expert Review Queue** aggregates feedback from multiple experts, providing a trustworthy mechanism for resolving disagreements before a final verdict is committed ([Figure 4.6](#)).
 - *Persistent and Enriched Feedback.* The **Feedback Ingestor** component enriches the final verdict with the original decision context before persisting it in the **Feedback Store**, ensuring that all data for future audits or model retraining is complete and traceable ([Figure 4.8](#)).

NFR.003: Usability & Accessibility

- **Requisite** - Interfaces for human experts must be intuitive and efficient, minimising cognitive load and expert fatigue, while being accessible to all intended users.

- **Fulfilment** - The design prioritises the expert's workflow:
 - *Unified Interface.* All SME interactions are consolidated within the **Dashboard & APIs** container, providing a single point of entry and avoiding the cognitive load of switching between different tools.
 - *Intelligent Task Prioritisation.* The **Task Prioritiser** component within the **Expert Review Queue** applies a configurable policy to order the work queue. This allows operators to surface the most critical tasks first, ensuring expert time is spent on high-impact items (Figure 4.6).
 - *Reduced Wait Time.* The **Task Broker** proactively requests an explanation from the **Explainability Service** when a task is created. This pre-fetching means the rationale is often ready before the SME even claims the task, eliminating frustrating wait times.

NFR.004: Performance & Scalability

- **Requisite** - The architecture must be performant and scalable, capable of handling growing volumes of data, users, and computational load with low latency and high throughput.
- **Fulfilment** - Scalability is achieved via several architectural patterns:
 - *Asynchronous Processing.* By placing the **Expert Review Queue** after the synchronous **Decision Guard**, the system decouples real-time decision gating from off-line human review. This ensures that a backlog of reviews does not impact the performance of the primary application workflow.
 - *Stateless, Replicable Services.* The core processing containers, **Decision Guard**, **Expert Review Queue**, and **Explainability Service**, are designed to be stateless, allowing them to be horizontally scaled to handle increased load.
 - *Elastic Compute for Heavy Jobs.* The **Explainability Service** uses a **Job Dispatcher** to offload long-running explanation generation tasks to the **Infra Orchestration** service, which can provision dedicated resources from the **Service Pool** without impacting the API's responsiveness.

NFR.005: Adaptability & Maintainability

- **Requisite** - The architecture must feature a modular, extensible, and open-source design to facilitate easy upgrades, modifications, and community contributions.
- **Fulfilment** - The design supports safe and rapid evolution:
 - *Modular Container Boundaries.* The HITL functionality is encapsulated in five self-contained containers with versioned APIs (Figure 4.4). A single container (e.g., the **Feedback Manager**) can be updated or replaced without requiring a rebuild of the entire system.
 - *Hot-Swappable Policies.* The **Rule Evaluator** in the **Decision Guard** fetches policy bundles from the **Control Plane**. This allows operators to roll out new rules by simply deploying a new policy version, without any code change or service restart.

- *Safe Rule Validation.* The **Shadow Rule Evaluator** component runs candidate policies on mirrored production traffic, allowing new logic to be safely validated against real-world data before it is promoted to the stable, blocking path (Figure 4.5).
- *Targeted Observability.* Components like the **Telemetry Collector** in the **Expert Review Queue** emit specific metrics on reviewer performance and disagreement rates, simplifying monitoring and targeted process improvement.

NFR.006: Interoperability & Compliance

- **Requisite** - The system must ensure seamless data integration with external data spaces and demonstrate strict adherence to European regulations (e.g., GDPR).
- **Fulfilment** - The architecture is designed for a federated ecosystem:
 - *Standardised Interfaces.* The **Dashboard & APIs** serves as the primary integration point for human-facing systems, while the **Control Plane** handles policy and contract negotiation with peer Data Space nodes, promoting system-level interoperability.
 - *Auditability for Compliance.* The immutable log on the **DLT / Ledger** provides a verifiable audit trail required for demonstrating compliance with regulations like GDPR, proving who accessed data, when, and for what purpose.
 - *Explicit Policy Enforcement.* The **Decision Guard** serves as a technical enforcement point for compliance rules, capable of blocking data processing that would violate configured data sovereignty or usage policies.

Table 5.1: Container-attribute coverage matrix (✓ = directly supports).

Container	Security & Privacy	Trust/Traceability	Usability	Scalability	Adaptability	Interop/Compliance
Decision Guard	✓	✓		✓	✓	✓
Expert Review Queue	✓	✓	✓	✓	✓	
Explainability Service		✓	✓	✓	✓	
Feedback Manager	✓	✓		✓	✓	✓
Dashboard & APIs	✓		✓	✓		✓
Decision Store	✓	✓		✓	✓	✓
Feedback Store	✓	✓		✓	✓	✓

Synthesis The analysis confirms that each non-functional attribute is backed by specific, observable mechanisms within the HITL architecture. **Security and Privacy** are enforced through a Zero Trust model from the ingress point inwards. **Trustworthiness and Traceability** are delivered through a verifiable, DLT-anchored provenance chain and explicit conflict-resolution components. **Usability** is achieved by designing workflows that minimise expert cognitive load, while **Scalability** is ensured through asynchronous processing and stateless services. Finally, **Adaptability** is supported by modularity and safe-change mechanisms like shadow testing, and **Interoperability**

and Compliance are addressed via standardised interfaces and auditable policy enforcement. Accordingly, the architecture can be deemed sound on its behavioural qualities and ready for further validation.

5.3 Expert Review through Qualitative Interviews

A third validation stream complemented requisite checking, ATAM and use-case walkthroughs with a *qualitative interview survey* involving two senior NOUS architects who had **no prior involvement** in the HITL design. Their independent perspective was therefore well-suited to stress-test the assumptions, diagrams and cross-cutting decisions presented in Chapter 4.

5.3.1 Interview Method and Protocol

Both sessions followed the semi-structured guide in Appendix A, which complies with the ACM SIGSOFT empirical standard for architect interviews. The protocol concentrated on three topics that emerged during pilot runs: (i) diagram integrity across C4 levels; (ii) soundness of the new HITL containers; and (iii) adequacy of the security–usability hooks. Each interview lasted ≈ 60 min, was audio-recorded, transcribed verbatim and member-checked.

The two architects occupy distinct layers of the NOUS stack providing complementary viewpoints on scalability, governance and user workflows.

5.3.2 Data Collection

All baseline and HITL diagrams (System, Container, Component, and Sequence) were screen-shared at the start of each call to serve as the primary artifacts for discussion. The sessions were audio-recorded and the resulting transcripts were member-checked for accuracy. These transcripts, alongside the interviewer’s observational notes, form the primary data set for the qualitative analysis.

5.3.3 Key Findings

Perceived clarity of the architectural narrative. Both interviewees welcomed the migration from the project’s ad-hoc “wall-of-boxes” drawing to a C4 hierarchy, yet they still felt that the *System Context* view should “*treat the node as a single system first and only zoom in later*”. That advice confirmed Decision D1 in the standardisation exercise (4.3.2) and motivated the tighter system boundary seen in Figure 4.2.

Arrow semantics and visual load. Interviewee B cautioned that the container diagram “*continues to look busy with too many arrows and readers may assume dependency instead of data-flow*”. This highlighted a critical ambiguity in the diagrams, where essential but distinct types of interactions, such as the primary data pipeline versus a configuration lookup, were visually indistinguishable, increasing cognitive load.

Cache versus Store debate. A vigorous thread centered on the so-called *Explanation Cache*. Both architects argued that the element behaved more like durable storage than an ephemeral cache: “*I’m not sure a cache is what you want, maybe it’s just a database*”. Similar doubts arose for the Decision and Feedback caches. The consensus was to rename them **Store** and to render them with database icons to avoid misleading stakeholders.

Ledger write strategy. Persisting *every* event to the consortium ledger was flagged as both costly and a potential single point of failure: “*If you write everything to the blockchain, it will become a single point of failure*”. The architects recommended batching low-risk events and hashing the batch root. This mirrors patterns used in high-performance DLT deployments that are designed to handle a high volume of Transactions Per Second.

Observability and log aggregation. Both architects encouraged a tiered logging approach: component-local logs aggregated into a shared store to simplify forensic audits.

5.3.4 Adjustments Following the Interviews

Four concrete changes were implemented before publication:

1. **System Context redraw.** The top-level diagram now depicts the *NOUS Data-Space Node* as a single bounded system with external actors, reducing cognitive load and aligning with C4 principle D1.
2. **Store iconography and nomenclature.** *Explanation Cache*, *Decision Cache* and *Feedback Cache* were renamed to *Stores*; all three are shown with the standard database glyph.
3. **Refined diagrammatic conventions.** To address the ambiguity between data-flow and dependency, all HITL architecture diagrams in Chapter 4 were updated to use a consistent, two-style line convention. Solid lines now exclusively represent the primary data-processing workflow, while dashed lines denote supporting interactions such as control signals, configuration lookups, and audit logging. This visual separation of concerns clarifies the architectural intent.
4. **Selective ledger attestation.** To mitigate the risk of the DLT becoming a performance bottleneck, the design was updated to recommend a more selective attestation strategy. The architecture now supports batching lower-priority events, reducing the frequency of ledger writes while ensuring that all high-value verdicts and critical actions are still recorded individually for maximum provenance.

Summary. Although based on just two experts, the interviews provided high-leverage insights. The resulting diagram tweaks and persistence strategy bring the HITL layer closer to production viability without sacrificing auditability or human oversight guarantees.

5.4 Validation via Use-Case Mapping

5.4.1 Introduction & Scope

Use-Case #3, code-named **CRIMSON**, models a multi-agency crisis-management platform that coordinates first responders during natural or man-made disasters.

Its declared aim is to “*enhance the capacities of a cloud-based crisis management platform by integrating NOUS for improved data lifecycle management*” [22] and to “*handl[e] the access and traceability of vast amounts of data exchanged during major crises*” [22].

CRIMSON is therefore an ideal stress-test for the HITL layer: it mixes time-critical automation (AI anomaly detection, live drone streams) with moments where a human crisis-manager must override or endorse machine recommendations.

This validation maps every step of the CRIMSON main flow to the containers, stores and workflows introduced in Chapter 4. The goal is to show that:

1. each *actor* or external system has a clear entry point into the NOUS node;
2. every *decision* or *data artefact* required by the use case is either produced or consumed by a HITL component; and
3. all functional and non-functional requisites remain satisfied while the scenario unfolds.

The scope is limited to the interactions that occur *inside* a single NOUS node during one crisis event; cross-node data spaces, long-term archival and post-mortem analytics are outside of this analysis’s scope.

5.4.2 Main Scenario Synopsis

The authoritative main flow lists ten high-level steps. They are restated here verbatim to establish the baseline scenario before the integration of the HITL components:

- S1. A crisis event is reported.**
- S2. The Crisis Manager logs into the platform.**
- S3. Field Operators collect and send real-time data (text, images, video, GPS).**
- S4. Data are transmitted to the cloud server.**
- S5. The AI Analytics System detects anomalies in video streams.**
- S6. Drones and CCTV provide live monitoring.**
- S7. The Crisis Manager analyses data and issues strategic decisions.**
- S8. Emergency Responders receive orders and act.**
- S9. The blockchain logs all actions and communications.**

S10. After the crisis, an After-Action Report is generated.

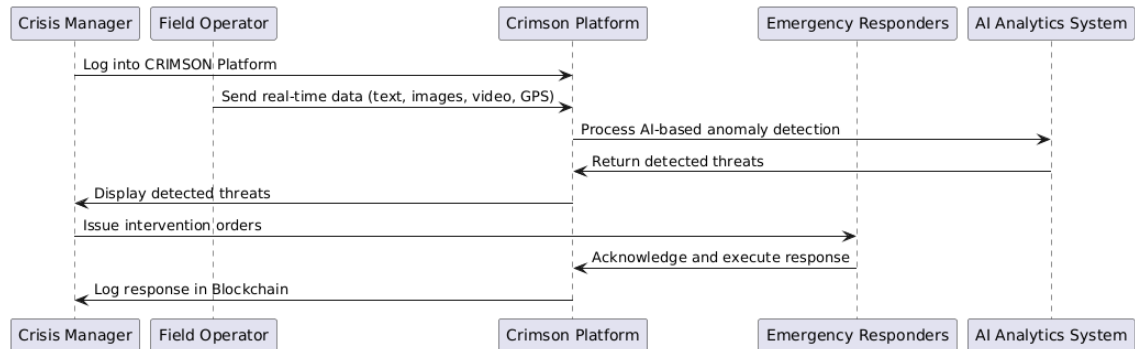


Figure 5.1: Baseline CRIMSON message flow before HITL integration (steps S1–S10).

Figure 5.1 presents the original call flow without the HITL containers. The subsequent sections introduce the actor–container mapping and the updated sequence diagram that show how the new layer fits into this baseline.

5.4.3 Actor-Container Mapping

Table 5.2 links the principal CRIMSON actors and main-flow events to the concrete runtime elements defined in Chapter 4. The right-hand column highlights the *incremental value* contributed by the five HITL containers.

Rows 3-7 illustrate where the HITL layer actively enriches or governs the baseline flow, ensuring that scarce human expertise is applied precisely where the machine is least confident and that all interventions remain traceable and reversible. The mapping also confirms there are *no new ingress points*: every external actor still enters through existing baseline gateways, preserving the node’s security perimeter.

5.4.4 End-to-End Sequence (HITL Overlay)

Figure 5.2 superimposes the five green HITL containers on top of the original CRIMSON message timeline. The labels **H1-H7** mark the precise interception points introduced by the extension; these identifiers are reused in the narrative that follows.

Message flow.

- H1. Sensor data ingested.** Video and telemetry arrive via the *Data Plane* and are routed to the *Distributed Execution Runtime* for inference (baseline).
- H2. Guard verdict issued.** The **Decision Guard** checks each inference. *APPROVED* events pass straight to the dashboard; *FLAGGED* events are published to the **Expert Review Queue**. A hash of the verdict is anchored on the external ledger.

Table 5.2: CRIMSON actors and events mapped to HITL containers.

#	Actor / event	Runtime realisation (base-line path)	Added HITL contribution
1	Crisis event is reported	<i>Dashboards & APIs</i> → Control Plane	Pure baseline step; no HITL interception required.
2	Crisis Manager logs into the platform	Common Admin Services (AuthN/Z)	Login token later authorises <i>real-time override</i> actions (FR-007).
3	Field Operator streams text/media/GPS	Data Plane → Distributed Execution Runtime	Subsequent model inferences enter the Decision Guard , which computes confidence and flags uncertain cases (FR-002).
4	AI Analytics System issues an anomaly detection	Distributed Execution Runtime	Decision Guard writes APPROVED/FLAGGED hash to DLT and enqueues low-confidence detections in the Expert Review Queue .
5	Drones / CCTV deliver live video feeds	Data Plane → Runtime (stream inference loop)	Same interception as Row 4; heavy explanation jobs are off-loaded to the Explainability Service .
6	Crisis Manager validates the threat or requests a rationale	<i>Dashboards & APIs</i> container	Receives pre-fetched explanation from XAI store; verdict recorded by the Queue's Consensus Engine and passed to the Feedback Manager .
7	Crisis Manager issues intervention orders	Dashboard → App/Data Orchestration	Feedback Manager hashes the final verdict + order to the DLT, then triggers downstream orchestration; fulfils FR-004 and FR-003.
8	Emergency Responders acknowledge and act	Orchestration → External responders	No direct HITL processing, but responders' acknowledgements are hashed in the Decision Store for provenance.
9	Blockchain logs every critical action	External <i>DLT / Ledger</i>	Hashes emitted by Decision Guard & Feedback Manager create an immutable audit chain, reinforcing NFR-001 (Security) and FR-003 (Auditability).

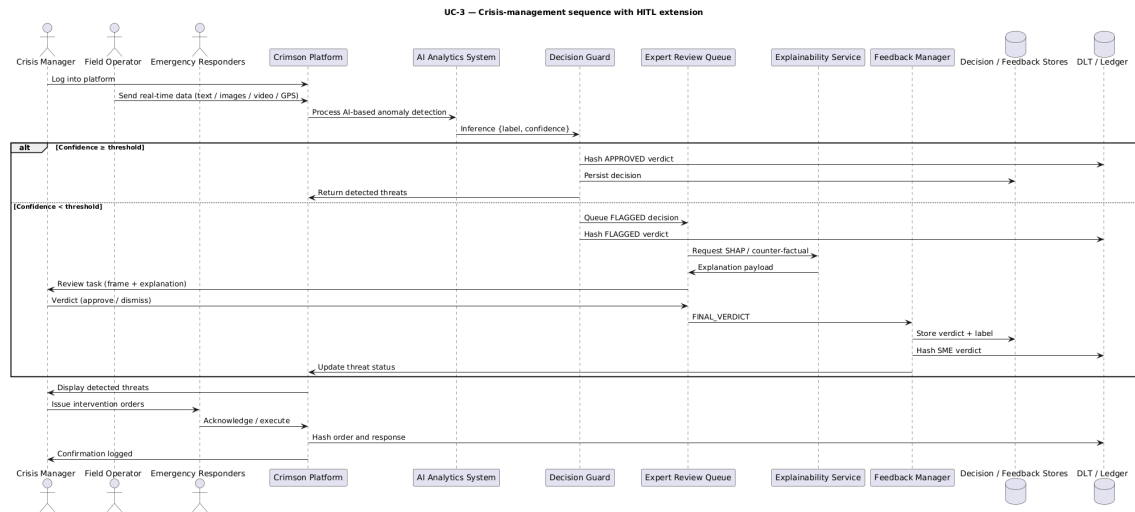


Figure 5.2: Sequence diagram for Use-Case 3 with HITL intercepts.

- H3. Task brokered.** The Queue’s *Task Broker* stores the flag and requests an explanation from the **Explainability Service**. Computed rationales return immediately; otherwise a GPU job is spawned.
- H4. SME review.** The Crisis-Manager dashboard pulls the task, displaying the frame, the explanation and metadata. The manager can *Confirm*, *Dismiss* or *Request Info*.
- H5. Consensus + ledger.** Once the configurable quorum is reached, the Queue’s *Consensus Engine* emits a `FINAL_VERDICT`. The **Feedback Manager** stores the verdict, links it to the original decision hash, and writes a new ledger record.
- H6. Operational action.** Confirmed threats generate an `Issue Order` to Emergency Responders; dismissed detections suppress the automated alert. Either path now carries a provenance link back to the SME verdict.
- H7. Learning loop.** The Feedback Manager adds SME-labelled frames to the *Feedback Store*, recalculates drift metrics, and if thresholds are crossed triggers an active-learning batch for retraining in the next orchestration cycle.

Latency profile. Only the FLAGGED branch incurs the SME round-trip; all APPROVED detections experience the same real-time latency as in the baseline. Guards, Queue and XAI jobs are horizontally scalable, so the worst-case tail latency is dictated mainly by SME availability, addressed in Section 5.4.5 when we examine backlog overload.

This sequence confirms that the HITL layer *adds* explainability, auditability and human veto without altering the existing ingress pattern or the critical-path timing of high-confidence detections.

5.4.5 Coverage of Alternative Flows

The CRIMSON specification lists six alternative or exception scenarios. Table 5.3 explains how the HITL-enabled NOUS node satisfies each one without breaking the guarantees established for the main flow.

Table 5.3: UC-3 alternative scenarios and architectural responses.

Alt. scenario	Handling in HITL-enhanced node
A. Limited connectivity	Field device buffers frames in a local store. Once the connection is restored, frames are replayed through the Decision Guard . Ledger timestamps preserve original capture time, so auditors can correlate late-arriving data with real-world events.
B. AI processing failure	If the inference container crashes, the Guard receives no detection and therefore raises <i>FLAGGED null-result</i> tasks. SMEs label these frames manually; the Feedback Manager pushes the labels to retraining, repairing the model.
C. Drone / CCTV malfunction	Sensor-health metrics are ingested by the Data Plane. This scenario is handled by the baseline architecture's standard health monitoring capabilities, which the HITL layer integrates with.
D. Unauthorised access attempt	Requests without a valid short-lived JWT never reach the HITL layer; the API Gateway blocks them. A ledger hash records the failed attempt, and the Security Dashboard issues an alert. No new ingress was introduced by HITL, so the original perimeter stands.
E. Data overload (burst traffic)	This scenario is handled by the core design principles of the HITL layer. The Expert Review Queue acts as an asynchronous buffer, insulating the system from the traffic spike. Its internal Task Prioritiser can be configured to reorder the queue, preventing high-priority events from being delayed. This mechanism, combined with the designed scalability of the stateless HITL containers, directly addresses the performance requirements (NFR.004) without affecting the latency of the primary APPROVED decision path.
F. Conflicting SME orders	All SME votes are retained. If the Consensus Engine cannot reach quorum, the task escalates to a senior reviewer; both the conflict and the resolution hash are written to the ledger, meeting auditability and governance requirements.

Observation. None of the alternative flows introduces a new trust boundary or bypasses the Decision Guard. Instead, each exception is surfaced to SMEs through the same queue-and-dashboard

workflow, keeping security and provenance controls intact while allowing rapid human intervention.

5.4.6 Validation Verdict & Residual Risks

Verdict. The walkthrough demonstrates that the HITL layer:

- *Preserves real-time operation* - high-confidence detections still cross the node in the same sub-second path as before; only low-confidence cases incur SME latency (FR-002, NFR-003).
- *Closes the governance gap* - every inference, human verdict and downstream order is immutably chained on the consortium ledger, providing a replayable evidence trail for the eight-year retention period required by CRIMSON (FR-003, NFR-001).
- *Adds explainability without code intrusion* - the Explainability Service is fully decoupled from the model runtime; AI containers remain unchanged, satisfying the “no vendor lock-in” clause of the project objectives (NFR-002, NFR-005).
- *Keeps the security perimeter intact* - no new north-south ingress, no additional credential stores; all traffic is protected by the node’s existing service mesh (NFR-001).

Residual risks. Three open issues were identified:

R1. GPU saturation for on-demand explanations.

A burst of flagged video frames could exhaust the shared GPU pool, delaying XAI jobs and, by extension, SME verdicts. *Mitigation* - enforce per-tenant GPU quotas and fall back to a faster but coarser explainer when the queue exceeds the 95th percentile.

R2. Ledger write outage.

If the external DLT is unreachable, the Guard and Feedback Manager cannot push hashes, risking a break in the provenance chain. *Mitigation* - both containers now buffer hashes locally and flush asynchronously; a drill is scheduled for the next release sprint to verify loss-free replay.

R3. SME backlog under prolonged crises.

A multi-day incident could push the queue beyond the 200-task threshold, where usability degrades. *Mitigation* - the Task Prioritiser supports dynamic policy adjustments. For instance, operators can tune the confidence threshold that flags items for review, effectively reducing task inflow by trading recall for throughput in a controlled, auditable manner.

Conclusion. All mandatory functional and non-functional requisites are met for Use-Case #3. The remaining risks have well-defined mitigation tactics and do not undermine the architectural soundness. The HITL extension is therefore **validated as fit for deployment in the CRIMSON scenario**, and the same mechanisms can be reused, with minor policy tuning, for the other NOUS pilot use cases.

Chapter 6

Conclusions & Future Work

6.1 Major Achievements

This thesis set out to architect and conceptually validate a human-in-the-loop reference architecture for the NOUS federated cloud-edge platform, showing that rich human oversight can be embedded without undermining performance, privacy or scalability. Table 6.1 first maps each research question to the chapters, sections and artefacts that provided an answer at the conceptual level. A separate list then reviews how the four objectives introduced in Section 1.6 were satisfied.

Table 6.1: Research-question coverage

RQ	Where addressed / key evidence
RQ1:	<i>Architectural components and patterns.</i> Chapter 4 introduces the System-Context, Container and Component views that define the new <i>Decision Guard</i> , <i>Expert Review Queue</i> , <i>Explainability Service</i> , <i>Feedback Manager</i> and extended <i>Dashboard & APIs</i> . Earlier, Chapter 3 distils a four-cluster taxonomy of HITL patterns from a PRISMA-guided review, providing the design rationale for those components.
RQ2:	<i>Continuous-improvement workflow and quality attributes.</i> Design principles such as Human-Centricity, Transparency and Privacy (Section 4.4.1) translate into architectural mechanisms: an asynchronous feedback channel, inference with pseudonymised metadata, and provable audit records on the NOUS ledger (Sections 4.5.6 and 5.2.2).

Objective satisfaction. All four objectives have been met, although at a conceptual rather than implementation level:

- (i) **Characterise the design space**

Achieved in Chapter 3, where the literature survey pinpoints recurring oversight patterns, open gaps and ethical guard-rails, thereby framing the architectural requirements.

- (ii) **Specify a reference architecture**

Delivered in Chapter 4 via C4 diagrams and interface contracts that make HITL responsibilities explicit and tooling-friendly.

- **(iii) Instantiate the architecture**

Realised conceptually through detailed sequence diagrams and scenario maps of the CRIMSON crisis-management use case (Section 2.4.4), demonstrating feasibility without a running prototype.

- **(iv) Evaluate the design**

Addressed in Chapter 5 through architect interviews, design-review workshops and use-case mapping, which collectively produced a risk catalogue and mitigation guidelines.

Hypothesis assessment. The evidence above supports the hypothesis posed in Section 1.4: responsible-AI patterns can be integrated architecturally to provide scalable, human-centered oversight without incurring prohibitive performance penalties- *in principle*. Empirical validation is deferred to future implementation efforts.

6.2 Future Work

Although this thesis delivers a conceptually validated blueprint for embedding human-in-the-loop oversight in the NOUS cloud–edge ecosystem, several follow-on activities are required to harden the design, broaden its scope and translate it into operational value. Key directions include:

- **Broader expert engagement.** Conduct additional semi-structured interviews until thematic saturation is reached, explicitly targeting (i) domain leads from other NOUS work packages and (ii) external practitioners with first-hand HITL experience in safety-critical AI. The goal is to triangulate assumptions, uncover blind spots and gather domain-specific requirements that did not surface in the initial review cycle.
- **Cross-use-case validation and ATAM workshops.** Re-enact the architecture walkthrough with at least two further NOUS demonstrators and run Architecture Trade-off Analysis Method (ATAM) sessions to examine quality-attribute tensions across latency, explainability, privacy and cost. This will stress-test the design’s adaptability before code is written.
- **Prototype implementation and benchmarking.** Progress from C4 diagrams to a deployable minimum-viable prototype. Containerise the *Decision Guard*, *Expert Review Queue* and *Feedback Manager*, then measure end-to-end latency, resource footprint and fault tolerance on representative edge hardware.
- **Human-centered interface design.** Co-design operator dashboards that visualise model uncertainty, escalation queues and audit trails. Follow up with usability testing and cognitive-load assessment to ensure interventions are intuitive under time pressure.

- **Role-Based and Multi-Faceted Explanations.** Evolve the *Explainability Service* to provide explanations tailored to different user personas. This would involve developing distinct explainer modules: a technical, feature-based explainer for SMEs; a policy-based explainer for compliance officers that links to the specific rule violated in the *Control Plane*; and an impact-based explainer for business analysts. Implementing this would advance the system towards providing truly comprehensive transparency for all stakeholders.
- **End-to-end C4 baseline for NOUS.** Complete a system-wide C4 model of the existing NOUS infrastructure so that the proposed HITL extensions can be traced unambiguously to their hosting contexts and communication paths.
- **HITL in model training and labelling.** Build a continuous data-curation pipeline where the *Feedback Manager* surfaces high-uncertainty or high-impact samples to subject-matter experts for annotation; track inter-annotator agreement, feed curated data back into training loops, and experiment with active- and semi-supervised learning strategies to minimise labelling effort while maximising model improvement.

Pursuing these strands will transform the present architecture from a conceptual framework into a rigorously evaluated, user-centered and production-ready platform for responsible AI across the NOUS federation.

Bibliography

- [1] S. Alawadi et al. “A Federated Interactive Learning IoT-Based Health Monitoring Platform”. In: *Communications in Computer and Information Science* 1450 (2021), pp. 235–246. DOI: [10.1007/978-3-030-85082-1_21](https://doi.org/10.1007/978-3-030-85082-1_21).
- [2] E. Alberti et al. “AI Lifecycle Zero-Touch Orchestration within the Edge-to-Cloud Continuum for Industry 5.0”. In: *Systems* 12 (2024). DOI: [10.3390/systems12020048](https://doi.org/10.3390/systems12020048).
- [3] Saleema Amershi et al. “Power to the People: The Role of Humans in Interactive Machine Learning”. In: *Proceedings of the 2014 Conference on Human Factors in Computing Systems*. 2014.
- [4] J.S. Andersen and W. Maalej. “Design Patterns for Machine Learning-Based Systems With Humans in the Loop”. In: *IEEE Software* 41 (2024), pp. 151–159. DOI: [10.1109/MS.2023.3340256](https://doi.org/10.1109/MS.2023.3340256).
- [5] P. Barnard et al. “Responsible MLOps Design Methodology for an Auditing System for AI-Based Clinical Decision Support Systems”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 15383 (2025), pp. 217–236. DOI: [10.1007/978-3-031-80889-0_16](https://doi.org/10.1007/978-3-031-80889-0_16).
- [6] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. 2nd. Addison-Wesley, 2003.
- [7] Simon Brown. *The C4 Model for Visualising Software Architecture*. <https://c4model.com>. 2018.
- [8] Tara Capel and Margot Brereton. “What is Human-Centered about Human-Centered AI? A Map of the Research Landscape”. In: *Conference on Human Factors in Computing Systems - Proceedings*. 2023. DOI: [10.1145/3544548.3580959](https://doi.org/10.1145/3544548.3580959).
- [9] R. Cerulli et al. “A Service Oriented Architecture for Clinical Decision Support Systems Based on Artificial Intelligence”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 14779 (2025), pp. 161–171. DOI: [10.1007/978-3-031-78241-1_15](https://doi.org/10.1007/978-3-031-78241-1_15).
- [10] Paul Clements et al. *ATAM: Method for Architecture Evaluation*. Tech. rep. Software Engineering Institute, Carnegie Mellon University, 2001.

- [11] T. Comes. “AI for crisis decisions”. In: *Ethics and Information Technology* 26 (2024). DOI: [10.1007/s10676-024-09750-0](https://doi.org/10.1007/s10676-024-09750-0).
- [12] Panagiotis Dimitrakis. *D2.1 Status quo, gap analysis, requirements and project evaluation framework | Pending Approval*. May 2025. DOI: [10.5281/zenodo.15393605](https://doi.org/10.5281/zenodo.15393605).
- [13] European Parliament and Council of the European Union. *Regulation laying down harmonised rules on artificial intelligence and amending certain Union legislative acts (Artificial Intelligence Act)*. Available via EUR-Lex. Text adopted by the Council on 21 May 2024. Awaiting publication in the Official Journal of the European Union. Procedure file 2021/0106(COD). May 2024.
- [14] J. Françoise, B. Caramiaux, and T. Sanchez. “Marcelle: Composing Interactive Machine Learning Workflows and Interfaces - UIST 2021 - Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology”. In: (2021), pp. 39–53. DOI: [10.1145/3472749.3474734](https://doi.org/10.1145/3472749.3474734).
- [15] David Garlan and Mary Shaw. “An introduction to software architecture”. In: *Advances in Software Engineering and Knowledge Engineering* 2 (1993), pp. 1–39.
- [16] Moritz Hardt, Eric Price, and Nathan Srebro. “Equality of Opportunity in Supervised Learning”. In: *Advances in Neural Information Processing Systems*. 2016.
- [17] A. Havrilla et al. “trlX: A Framework for Large Scale Reinforcement Learning from Human Feedback - EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings”. In: (2023), pp. 8578–8595. DOI: [10.18653/v1/2023.emnlp-main.530](https://doi.org/10.18653/v1/2023.emnlp-main.530).
- [18] Alan Hevner et al. “Design Science in Information Systems Research”. In: *Management Information Systems Quarterly* 28 (Mar. 2004), pp. 75–.
- [19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep Learning”. In: *Nature* 521.7553 (2015), pp. 436–444.
- [20] Q. Lu et al. “Responsible-AI-by-Design: A Pattern Collection for Designing Responsible Artificial Intelligence Systems”. In: *IEEE Software* 40.3 (2023), pp. 63–71. DOI: [10.1109/MS.2022.3233582](https://doi.org/10.1109/MS.2022.3233582).
- [21] Q. Lu et al. “Towards a Roadmap on Software Engineering for Responsible AI”. In: *Proceedings - 1st International Conference on AI Engineering - Software Engineering for AI, CAIN 2022* (2022), pp. 101–112. DOI: [10.1145/3522664.3528607](https://doi.org/10.1145/3522664.3528607).
- [22] Alberto Montero Fernández et al. “A catalyst for European cloud services in the era of data spaces, high-performance and edge computing: NOUS”. In: *Proceedings of the 4th Eclipse Security, AI, Architecture and Modelling Conference on Data Space. eSAAM '24*. Mainz, Germany: Association for Computing Machinery, 2024, pp. 1–9. ISBN: 9798400709845. DOI: [10.1145/3685651.3686660](https://doi.org/10.1145/3685651.3686660).

- [23] A. -R. Ottun et al. “The SPATIAL Architecture: Design and Development Experiences from Gauging and Monitoring the AI Inference Capabilities of Modern Applications”. In: *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)* (2024), pp. 947–959. ISSN: 2575-8411 VO -. DOI: [10.1109/ICDCS60910.2024.00092](https://doi.org/10.1109/ICDCS60910.2024.00092).
- [24] J. Parra-Ullauri et al. “History-aware explanations: Towards enabling human-in-the-loop in self-adaptive systems - Proceedings - ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems, MODELS 2022: Companion Proceedings”. In: (2022), pp. 286–295. DOI: [10.1145/3550356.3561538](https://doi.org/10.1145/3550356.3561538).
- [25] Ken Peffers et al. “A design science research methodology for information systems research”. In: *Journal of Management Information Systems* 24 (Jan. 2007), pp. 45–77.
- [26] Rayyan - Intelligent Systematic Review - Rayyan. <https://www.rayyan.ai/>. Last Accessed: 2024-04-15.
- [27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ““Why Should I Trust You?” Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016.
- [28] E. Roponen et al. “Use Cases and Design of an Intelligent Intrusion Detection System”. In: *Baltic Journal of Modern Computing* 10 (2022), pp. 689–709. DOI: [10.22364/bjmc.2022.10.4.06](https://doi.org/10.22364/bjmc.2022.10.4.06).
- [29] J.M. Rožanec et al. “Human-centric artificial intelligence architecture for industry 5.0 applications”. In: *International Journal of Production Research* 61 (2023), pp. 6847–6872. DOI: [10.1080/00207543.2022.2138611](https://doi.org/10.1080/00207543.2022.2138611).
- [30] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd. Prentice Hall, 2010.
- [31] D. Sculley et al. “Hidden Technical Debt in Machine Learning Systems”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcdf2674f757a2463eba-Paper.pdf.
- [32] Burr Settles. *Active Learning Literature Survey*. Tech. rep. University of Wisconsin–Madison, 2009.
- [33] Ben Shneiderman. *Human-Centered AI*. Oxford University Press, Jan. 2022. ISBN: 9780192845290. DOI: [10.1093/oso/9780192845290.001.0001](https://doi.org/10.1093/oso/9780192845290.001.0001). eprint: https://academic.oup.com/book/41126/book-pdf/50987951/9780192659996_web.pdf.
- [34] Simpl Programme Team. *Simpl-Open Functional and Technical Architecture Specifications*. Technical Report. Version 0.2 (draft). published 19 December 2024. Brussels: European Commission, Dec. 2024. URL: <https://simpl-programme.ec.europa.eu/system/files/2024-12/Simpl-Open%20Functional%20and%20Technical%20Architecture%20Specifications.pdf>.

- [35] *Systematic Review - an overview* | ScienceDirect Topics. <https://www.sciencedirect.com/topics/social-sciences/systematic-review>.
- [36] A. Thurzo. “Provable AI Ethics and Explainability in Medical and Educational AI Agents: Trustworthy Ethical Firewall”. In: *Electronics (Switzerland)* 14 (2025). DOI: [10.3390/electronics14071294](https://doi.org/10.3390/electronics14071294).
- [37] United Nations. *THE 17 GOALS | Sustainable Development*. <https://sdgs.un.org/goals>.
- [38] U. A. Usmani, A. Happonen, and J. Watada. “Human-Centered Artificial Intelligence: Designing for User Empowerment and Ethical Considerations”. In: *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (2023), pp. 1–7. ISSN: VO -. DOI: [10.1109/HORA58378.2023.10156761](https://doi.org/10.1109/HORA58378.2023.10156761).
- [39] S. Utomo et al. “Federated Trustworthy AI Architecture for Smart Cities - ISC2 2022 - 8th IEEE International Smart Cities Conference”. In: (2022). DOI: [10.1109/ISC255366.2022.9922069](https://doi.org/10.1109/ISC255366.2022.9922069).
- [40] W. Woods et al. “Combining AI control systems and human decision support via robustness and criticality - Proceedings of SPIE - The International Society for Optical Engineering”. In: 13058 (2024). DOI: [10.1117/12.3016311](https://doi.org/10.1117/12.3016311).
- [41] J. Wu et al. “Toward Human-in-the-Loop AI: Enhancing Deep Reinforcement Learning via Real-Time Human Guidance for Autonomous Driving”. In: *Engineering* 21 (2023), pp. 75–91. DOI: [10.1016/j.eng.2022.05.017](https://doi.org/10.1016/j.eng.2022.05.017).
- [42] W. Xu and Z. Gao. “Enabling Human-Centered AI: A Methodological Perspective”. In: *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)* (2024), pp. 1–6. ISSN: VO -. DOI: [10.1109/ICHMS59971.2024.10555771](https://doi.org/10.1109/ICHMS59971.2024.10555771).

Appendix A

Annex

A.1 Interview Protocol

Thesis title	<i>Architectural Design of Human-Centered Artificial-Intelligence Systems</i>
Method	Semi-structured qualitative interview survey with expert architects
Interviewer	Francisco Maldonado (MSc student, FEUP)
Date	[insert]
Interviewee	[insert name & role]
Location	[insert]
Duration	≈ 60–90 min

A.1 Purpose of the Interview and Study Context

This interview supports a Master’s dissertation that investigates **software-architecture patterns enabling effective Human-in-the-Loop (HITL) oversight of AI systems**. Your experience as an architect will help to:

1. Identify architectural components and patterns that sustain meaningful human oversight in AI applications.
2. Clarify how real-world feedback can be channelled into continuous model refinement.
3. Capture best practices for the technical and user-facing aspects of HITL solutions.

A.2 Interviewee and Interviewer Information

A.2.1 Interviewee

Selection. Two practising architects from the NOUS project were invited, each responsible for different layers of the platform, to capture distinct viewpoints on security, scalability and human-oversight workflows.

To capture at session start

- Role / focus area
- Years of architecture experience
- Typical responsibilities

A.2.2 Interviewer & Bias Reflection

Interviewer: Francisco Maldonado, MSc candidate specialising in human-centered AI. As author of the reference architecture I may favour my design; therefore the session encourages critique, probes assumptions and records contradictory evidence.

A.3 Ethical Considerations & Consent

Participation is voluntary; you may withdraw at any time. The conversation will be audio-recorded solely for transcription accuracy and deleted after thesis completion. You may review and edit the transcript; published material will be anonymised.

A.4 Interview Guide

A.4.1 Warm-up (≈ 5 min)

- Briefly describe your current role and architecture challenges.
- How often do AI/ML elements feature in your work?

A.4.2 Diagram Integrity (System & Container Views) (≈ 15 min)

- First impressions of baseline versus extended views?
- Do arrows read as data-flows or dependencies? Would a different notation help?
- Which boxes or lines could be removed at this level?

A.4.3 Traceability of Information (≈ 10 min)

- For highlighted elements: inherited, inferred or new?
- How should provenance be documented in the report?

A.4.4 HITL Component Deep-Dive (≈ 20 min)

- Decision Guard - policy bundles, confidence thresholds, real-time path
- Expert Review Queue - priority logic, SME load, scalability
- Explainability Service - security boundary, GPU pool, caching
- Feedback Manager - verdict persistence, blockchain hash, retraining trigger

A.4.5 Cross-Cutting Concerns (≈ 10 min)

- a) Security & privacy - authentication, access-control, PII handling
- b) Observability & auditability - logging topology, evidence for compliance
- c) Usability - dashboard workflow, front-end patterns for reviewers

A.4.7 Wrap-Up Reflections (≈ 5 min)

- a) Which elements offer most immediate value?
- b) Which appear risky, over-engineered or unproven?
- c) Emerging patterns or tools worth watching?

A.5 Closing the Interview

Thank you for your time. A transcript will be provided for validation, and a brief summary of findings is available on request.

A.6 Post-Interview Steps

1. Transcribe recording.
2. Member-check with interviewee.
3. Thematic analysis linking requirements, diagrams and feedback.
4. Maintain chain-of-evidence with verbatim quotes.
5. Integrate insights into Chapter 5.