FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Architectural design for the integration of Federated Learning Strategies: the NOUS Project use case

António Oliveira Ferreira

Mestrado em Engenharia Informática e Computação

Supervisors: Prof. Ademar Aguiar (FEUP)

Co-supervisor: Prof. Hugo Paredes (UTAD)

July 19, 2025

Architectural design for the integration of Federated Learning Strategies: the NOUS Project use case

António Oliveira Ferreira

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 19, 2025

Abstract

In 2016, Federated Learning (FL) was introduced as a new privacy-preserving distributed machine learning paradigm, functional especially in edge computing environments. However, this paradigm does not take into consideration the growing necessity to guarantee that Human-in-the-Loop (HITL) methodologies are robustly integrated into AI systems, as standardized integration approaches remain absent. Therefore, to bridge this gap, this thesis proposes a comprehensive, modular reference architecture for introducing HITL strategies into FL workflows at the edge, using the NOUS project (an EU initiative for next-generation cloud services) as a use case.

Our work begins with a systematic literature review and gap analysis to characterize the current practices and identify the core challenges in this research domain. From this foundation, we derive architectural high-level goals alongside functional and non-functional requirements. From this, we then present the HITL-FL framework, which utilizes C4 models (System Context, Container, Component). This framework includes a dedicated Human Oversight & Interaction Hub, secure annotation interfaces, task routing, ethical validators and operational loops for active learning and model governance.

We validate the architecture through three pillars: systematic compliance analysis against our requisites; mapping using the NOUS project and its Use Case #2 (Energy Prediction & Data Lifecycle Management); and qualitative expert reviews with NOUS architects. Validation results showcase strong support for comprehensive human oversight and alignment with trustworthiness, privacy, security, and usability requirements while also uncovering limitations and design challenges for future work.

Through this process, this research delivers a foundational blueprint for building human-centric, transparent, federated AI systems that foster a responsible collaboration between humans and AI in complex edge ecosystems.

Keywords: Federated Learning, Human-in-the-Loop, Software Architecture, Edge Computing

Resumo

Em 2016, Federated Learning (FL) foi introduzido como um novo paradigma de machine learning distribuído, que permite a preservação da privacidade e que é útil especialmente em ambientes de edge computing. No entanto, este paradigma não tem em consideração a necessidade crescente de garantir a integração robusta de metodologias Human-in-the-Loop (HITL) em sistmas IA, sendo que métodos referência de integração não existem. Sendo assim, para preencher esta lacuna, esta tese propõe uma arquitetura abrangente e modular para a introdução de estratégias HITL em workflows de FL na edge, utilizando o projeto NOUS (uma iniciative europeia para a próxima geração de serviços cloud) como um caso de aplicação.

O nosso trabalho começa com uma literature review sistemática e uma análise de falhas na literatura para caracterizar as práticas atuais e identificar os principais desafios presentes nesta área de investigação. A partir desta base, objetivos arquiteturais de alto-nível, juntamente como requisitos funcionais e não-funcionais, são definidos. De seguida, nós apresentamos a HITL-FL framework, utilizando a notação de modelos C4 (System Context, Container, Component). Esta framework inclui um Human Oversight & Interaction Hub, interfaces para anotação segura, task routing, validação ética e ciclos operacionais para active learning e model governance.

Nós validamos a arquitetura através de três pilares: uma análise de conformidade sistemática, contra os requisitos definidos; uma mapeamento utilizando o projeto NOUS e o seu 2º caso de uso (Energy Prediction & Data Lifecycle Management); e revisões de expecialistas externos com arquitetos do NOUS. Resultados da validação demonstrão um forte apoio da arquitetura quanto à sua implementação de oversight humano e alinhamento com vários requisitos de confiabilidade, privacidade, segurança e usabilidade, enquanto ainda detalha limitações e desafios de design para o futuro.

Através deste proceso, este trabalho apresenta uma base funcional para construir sistemas federados, centrados no humano e transparentes, que contribuem para uma colaboração responsável entre humanos e IA, em ecossistemas complexos de edge.

Palavras-chave: Federated Learning, Human-in-the-Loop, Arquitetura de Software, Computação Edge

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) [54] provide a global framework to achieve a better and more sustainable future for all. It has 17 goals to address the world's most pressing issues, such as poverty, inequality, climate change, environmental degradation, peace and justice. Taking this into account, to demonstrate the broader societal relevance of the proposed HITL-FL system, this section focuses on the **three** SDGs to which the thesis most directly contributes, considering UNESCO's vision of engineering for sustainable development [53]: **SDG 7 (Affordable and Clean Energy)**, **SDG 9 (Industry, Innovation and Infrastructure)** and **SDG 17 (Partnerships for the Goals)**.

SGD	Target	Contribution	Performance Indicators and Metrics
7	7.2	In the NOUS use case (Chapter 6), FL, with human oversight, enhances short-term demand and production forecasts for the power company, facilitating better integration of renewable energy, for example.	Forecast error reduction (%). Increase in renewable utilization (%) enabled by improved forecasts. Reduction in peak plant dispatch events attributable to better prediction.
	7.3	Edge-based FL reduces latency and compute overhead in forecasting pipelines.	Reduction in ML pipeline energy consumption. Improvement in response time for forecast updates. Decrease in network data transfer versus centralized processing.
9	9.1	Resilient distributed analytics architecture promotes infrastructure robustness.	Uptime or reliability improvements in pilot deployments. Reduction in communication overhead in edge FL rounds.
	9.5	Novel HITL-FL methods advance technological capabilities.	Count of new methods/prototypes developed and tested. Number of collaborative R&D outputs (publications, open-source modules).
17	17.6	Federated approach fosters multi-stakeholder collaboration without sharing raw data.	Number of distinct organizations in FL processes. Qualitative assessment of collaboration effectiveness.
	17.7	Architecture scaffolds trust in partnerships via defined governance and audit roles.	Documented new or strengthened partnerships formed. Reduction in time/effort for data collaboration agreements.

Acknowledgements

First of all, I would like to thank my two supervisors for their support throughout the past year of working on my thesis. To Professor Ademar Aguiar, for helping me develop a dissertation theme, for introducing me to the NOUS project, and for all the valuable tips and guidance throughout this process. To Professor Hugo Paredes, who took me in at INESC TEC and guided my work in the NOUS project, always making time to meet with the team every week and to respond to every doubt I had. This thesis wouldn't have been possible without them. This work is carried out as part of the NOUS project, which has received funding from the European Union's Horizon Europe research and innovation program under Grant Agreement 101135927.

To Professors Filipe Correia, Marco Oliveira and Carlos Duarte, who I had the opportunity to work with in the NOUS project and were a great help for this thesis.

To my parents, who have been a great source of support not only throughout this past year, but also during the last five years of university. You always pushed me to be the best version of myself and taught me that one should always give 100% on everything they do. And if something doesn't work out as we expected, there's always a second chance to try it. To my mother, who was my biggest supporter and was always worried about my future. To my father, who always takes the rational perspective in every situation, helping me be grounded and calm. Now, it is the end of my years as a student, but I am sure that I'll continue to make you proud.

To my friends at JuniFEUP, who, over the past three years, were my home away from home. You gave me the chance to step out of my comfort zone and be exposed to many challenges and opportunities that shaped the person that I am today. I have learned a lot, and I promise that I will try to be an active alumni.

Finally, to my colleagues and friends whom I made during the course. To Lia and Tomás, for working with me on many projects and helping me manage my anxieties and worries about completing the work on time and achieving good grades. To Nessy and Bruna, who made me forget about the stress of the course and were always ready to make me laugh with their jokes and antics. To Kiko, who stood beside me throughout this whole journey, from Projeto FEUP until this thesis, where we worked side by side, day after day, and for becoming the brother I never had. This may be the end of our time at FEUP, but I'm sure that in some way or another, we all will still meet in the future. Don't forget: ***vamos falando*** and ***vai correr tudo bem***.

Thank you to all of you.

António

"It always seems impossible until it's done."

Nelson Mandela

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
DSR	Design Science and Research
DDD	Domain-Driven Design
E2F2C	Edge-to-Fog-to-Cloud
FL	Federated Learning
FR	Functional Requisites
FTL	Federated Transfer Learning
GDPR	General Data Protection Regulation
HCAI	Human-Centered Artificial Intelligence
HCI	Human Computer Interaction
HFL	Horizontal Federated Learning
HITL	Human-in-the-loop
HPC	High-Performance Computing
IaaS	Infrastructure-as-a-Service
ML	Machine Learning
MLaaS	Machine Learning-as-a-Service
NFR	Non-functional Requisites
PaaS	Platform-as-a-Service
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PLM	Personalized Local Model
RIA	Research and Innovation Action
SDG	Sustainable Development Goals
SLM	Shareable Local Model
SOTA	State of the Art

Chapter 1

Introduction

1.1 Context

Over the past decade, artificial intelligence (AI) developments have taken significant leaps, leading to AI-enabled systems in different domains of daily life becoming mainstream and completely overhauling each individual's way of life. With the evolution in machine learning techniques and an increase in data availability and computing power, as evidenced by Rausch and Dustdar [44], there is a growing need to not only maximize resource allocation but also ensure efficient and secure data processing within the Edge-to-Fog-Cloud (E2F2C) continuum.

Federated learning (FL), first proposed by Google in 2016 [23][22] consists of a machine learning approach where its main defining point is the utilization of distributed datasets for training models, thereby allowing the collaboration between different organizations to build a shared model while each keeping their data locally. This concept has a significant positive impact on various aspects, with data privacy and security being the key highlights, as it enables the leveraging of data from multiple sources for machine modeling. Due to this privacy-focused way, only model updates can be shared, and all data is kept decentralized.

At the same time, as Cronholm and Göbel stated [13], although it is important to recognize the technical benefits and advantages the widespread usage of AI has brought to society, concerns regarding the incomprehensibility of the models used, embedded bias against particular societal groups and overall data and privacy issues have turned the technological-centered design of these AI models insufficient, leading to the necessary inclusion of human knowledge and competencies in these artificial intelligence systems.

Against this backdrop of evolving AI needs, NOUS [38], an EU-funded project under the Horizon Europe program, is a pioneer in imagining, designing and deploying state-of-the-art Cloud Services in the age of data spaces, high-performance computing (HPC) and edge computing. The goal of this project is to introduce an innovative distributed architecture [15] and a federated learning framework for the European Cloud Service, in which computational and data storage resources can be utilized from edge devices, HPC network supercomputers and Quantum Computers. As a

result, an unprecedented opportunity is unlocked for businesses, research institutions and individuals alike. In the realm of edge computing components, the NOUS project focuses on enhancing computational capabilities and efficiency within the E2F2C continuum.

1.2 Motivation

While FL methodologies have proven to reduce the interaction gap between edge devices and traditional cloud services, significant opportunities remain for optimizing and enhancing these frameworks. Therefore, as AI systems become increasingly complex and the potential of human expertise to strengthen and complement AI decision-making processes quickly gains traction, particularly in edge computing scenarios, a critical opportunity is highlighted: designing a system that incorporates Human-in-the-Loop (HITL) approaches into FL frameworks for edge computing.

1.3 Problem Identification

As the integration of federated learning has shown significant advancements in edge computing environments, there is still an evident lack of standardized and robust approaches for integrating HITL methodologies in these types of frameworks. While FL intrinsically supports privacy, and HITL promotes increased accuracy, interpretability and ethical alignment of AI models, the unified and practical combination of both these concepts, especially in dynamic edge environments, remains unaddressed, mainly architecturally. Other related issues will also need to be considered in this architectural unification, such as the scalability of human input, including relevant and unbiased expert inputs and integration of continuous monitoring. Considering this, within the context of NOUS' edge computing framework, there is the opportunity to design and implement an architecture that seamlessly integrates federated learning methods and HITL approaches. Drawing on the architectural principles outlined by Valipour et al. [58], the inclusion of HITL can potentially establish a foundation for standard practices in reliable AI development in edge computing environments.

Addressing this gap is the central focus of this thesis. The core problem is to propose a foundational architecture where the benefits of HITL, like enhanced reliability, safety and trustworthiness, as supported by HCAI principles [49], can be integrated within a decentralized, privacy-preserving FL system. Thus, embedding human oversight ensures that AI does not have 100% of the decision-making power, particularly in fully self-operating systems, in which AI exhibits limitations. Additionally, as FL's privacy is being preserved, the goals of the NOUS project will be achieved, thereby strengthening the EU's position in this global technology landscape.

1.4 Research Questions

To further understand the problem identified in section 1.3 and to guide the pursuit of this work's set of objectives, it is possible to define the following research questions:

RQ1. What are the key architectural components and strategies necessary for implementing HITL approaches, integrated with federated learning, in edge computing environments?

RQ2. What are the best methods and standards for designing and implementing architectures that enhance human agent oversight integration, thereby guaranteeing trustworthiness, ethical validation and accountability?

RQ3. How can data privacy, integrity, security and performance be maintained when implementing federated learning and HITL approaches?

1.5 Hypothesis

Considering the problem identified and the RQs, this work formulated the following hypothesis:

The implementation of an architecture design made up of components, patterns and good practices that integrate federated learning with human-in-the-loop approaches at the edge will lay the groundwork for highly adaptable and trustworthy AI systems in the NOUS edge computing framework without compromising data privacy or system performance.

This hypothesis addresses the main challenge of designing an architecture that effectively integrates human participation and oversight with federated learning processes while balancing the demands for trustworthiness against key system requirements such as performance, reliability, scalability, and interoperability within the constraints of NOUS' framework. This architecture plans to create a standardized solution to this complex balancing task.

1.6 Objectives

The main objectives of this thesis are as follows:

Objective 1: **Reference Architecture Design** - provide a *reference architecture* that tightly integrates HITL methods with FL in resource-constrained edge environments, taking the NOUS project as a guiding real-world use case, by defining components and specific workflows to embed human oversight within FL operations effectively.

Objective 2: **Requirement Elicitation** - derive a comprehensive set of functional (FR) and non-functional requirements (NFR), covering security, privacy, performance, scalability, maintainability and auditability, that the HITL-FL system must satisfy.

Objective 3: **Architectural Validation** - validate the proposed architecture to ensure its credibility and practicality, by employing a multi-faceted validation process.

Objective 4: **Transferable Knowledge Contribution** - distill and document design principles, architectural patterns and practical lessons, learned throughout the research process,

that can be reused and benefit the development of other federated ecosystems that integrate human expertise.

1.7 Planned Contributions

To achieve the proposed objectives (Section 1.6), this thesis aims to deliver the following contributions, also aligned with the hypothesis (Section 1.5) and RQs (Section 1.4):

- **Reference HITL-FL Architecture** - a modular, extensible architectural framework for integrating HITL mechanisms into FL in edge environments (Chapter 4). This includes C4-based system context, container and component view diagrams, with a clear separation of concerns and secure communication patterns.
- **Aligning Design Principles, Requirements and Architectural Patterns** - a catalog of functional and non-functional requirements and distilled design principles linked to key architectural patterns, showing how the architecture fulfills these requisites (Sections 4.2.1, 4.2.2 and 4.3.5).
- **HITL Integration Strategies and Workflows** - concrete mechanisms and workflows for demonstrating how human expertise is incorporated without compromising FL's privacy and performance (Chapter 5).
- **NOUS Mapping** - a mapping and discussion of how the reference architecture can be applied within the NOUS project environment and, particularly, for energy-prediction use cases (Section 6.4).
- **Evaluation via Expert Review** - validation of the proposed system through expert interviews, where insights on strengths, limitations and potential improvements are collected to assess the architecture's feasibility and usability (Section 6.5).
- **Broader Impact Considerations** - discussion of how the architecture supports trustworthy AI goals, privacy preservation, and potential contributions to collaborative edge-FL deployments (Chapter 6).
- **Design Lessons Learned** - reflections on trade-offs, challenges and best practices discovered during the design and evaluation process that contribute to future research (Chapter 7).

These contributions collectively address the identified gap in architecturally unifying HITL methodologies with FL frameworks in dynamic edge settings while balancing privacy, security, performance and trustworthiness.

1.8 Research Methodology

Regarding the chosen methodology for addressing the previously identified problem (Section 1.3), find an answer for the RQs (Section 1.4) and achieve the defined objectives (Section 1.6), the **Design Science & Research (DSR)** [8] methodology was adopted in this thesis.

DSR is a research paradigm, rooted in engineering and the sciences of the artificial, that focuses on developing innovative solutions to address practical problems. As outlined by vom Brocke *et al.* [8], DSR aims to "enhance technology and science knowledge bases via the creation of innovative artifacts that solve problems and improve the environment in which they are instantiated". The outcomes of DSR include innovative artifacts, which in this case can be the proposed architectural framework, and design knowledge that specifies the purpose and functions of the artifact, such as why and how it addresses the identified problem. This concept involves a rigorous process made up of various steps, which include artifact design, development and evaluation to produce contributions that are useful in a specific application context.

DSR's application is well-suited for this thesis since it directly aligns with its core objectives. The central aim of this work is to design and propose a reference architectural framework, which represents an innovative artifact for integrating HITL within FL systems. This process is fundamentally a problem-solving exercise, as it seeks to address the identified gaps in current FL and HITL integration practices.

With this in mind, DSR provides a structured approach to guide this research. The process begins with thoroughly exploring and understanding the problem space through a literature review and a comprehensive SOTA analysis. Consequently, this process forms the foundation for developing the HITL-FL framework, constituting this research's primary artifact. The usability of this artifact is then demonstrated through its mapping to a real-world scenario. More specifically, to the NOUS project and one of its use cases. Additionally, it is also evaluated against defined architectural requisites and design problems, to assess its potential effectiveness, as well as being the center of review in external expert interviews. This iterative process of problem understanding, design, and evaluation is characteristic of the DSR process, ensuring that this thesis proposes a novel solution that contributes to practical and validated design knowledge, thereby finding answers to the research questions proposed.

1.9 Document Structure

This document is structured into eight different chapters, each focusing on a specific process or task of this research:

Chapter 1 ***Introduction*** Presents the context, motivation and objectives. Also formalizes the research questions, hypothesis, the objectives, the planned contributions and the DSR methodology approach.

Chapter 2 ***Background*** Reviews fundamentals of AI/ML, FL, HCAI, HITL, software architecture and the NOUS project.

Chapter 3 ***Literature Review & State of the Art*** Synthesizes current research and pinpoints open gaps.

Chapter 4 ***HITL-FL Architecture*** Introduces design principles, requirements and the complete C4 model of the proposed system.

Chapter 5 ***HITL Integration Strategies & Mechanisms*** Details concrete HITL interactions, mechanisms and workflows (active learning, ethical validation and governance).

Chapter 6 ***Architectural Validation & Discussion*** Reports requirement fulfillment, NOUS mapping, expert feedback, strengths, trade-offs and threats to validity.

Chapter 7 ***Conclusions & Future Work*** Summarizes findings, reflects on objective satisfaction and outlines future research directions.

Chapter 2

Background

2.1 Introduction

This chapter presents a comprehensive review of the core concepts and topics that will be explored throughout the subsequent chapters, to contextualize and provide a better understanding to the reader. Bearing this in mind, this chapter will introduce and discuss various key concepts like AI and ML in Section 2.2; the main principles, processes and motivations concerning FL in Section 2.3; the definition and functions of HCAI, and how it related with the philosophy with HITL, throughout Section 2.4, the fundamentals of Software Architecture in section 2.5 and an overview and context of the NOUS project, in Section 2.6, which serves as the key application case. The objective of each section is to give the reader the ability to understand the topics and fully consider the challenges and contributions that this work details.

2.2 Artificial Intelligence & Machine Learning

Artificial Intelligence can be described, at a higher level, as the broad field of Computer Science dedicated to developing systems that simulate human intelligence and perform tasks autonomously [35].

AI encompasses various subfields, with Machine Learning (ML) being a pivotal one. This branch enables computers to learn from data and perform tasks without explicit programming [43] by using algorithms to build models. This process involves focusing on features, tasks and model selection.

By identifying patterns and relationships within large databases, ML models are trained to make predictions on new data. Common ML tasks include classification, which assigns predefined labels to data; prediction, which forecasts continuous numerical values or future outcomes; and clustering, which groups similar data points into clusters [6]. Concerning the typical ML process, it follows a workflow from feature generation to model selection and uncertainty quantification [4]. This data-driven learning capability of ML is fundamental for many modern AI applications and a core subject for understanding specialized approaches, such as Federated Learning.

2.3 Federated Learning

FL is a decentralized collaborative machine learning paradigm that enables multiple distributed clients to actively collaborate in training a global shared prediction model, without exposing individual data [22, 23]. In other words, while still maintaining data privacy. Unlike traditional ML, FL keeps data local, only sharing model updates like gradients or learned model parameters to a central server for aggregation of a model that is formed by the collective knowledge of all the participants. Considering this, it is established that the core principles of FL revolve around data decentralization, collaborative model improvement and enhanced privacy preservation.

Key motivations for adopting FL cover a range of subjects. Primarily, FL enhances privacy by design and provides a way for privacy preservation, as data breaches during transit or at a central repository are mitigated. This factor is particularly important when dealing with sensitive data in healthcare or finance. FL also enables access to distributed data sources that otherwise would be kept in isolation due to privacy or regulatory concerns. With this in mind, another great motivation is reducing data movement and sharing overhead, which can be costly. Finally, FL also seeks to help organizations meet and follow regulatory compliance, such as the General Data Protection Regulation (GDPR) rules in Europe [16], by minimizing direct data handling.

FL can be categorized into different types, based on how data is distributed across clients [61]. The FL types are:

- **Horizontal Federated Learning (HFL)**: sample-based FL, where the datasets share the same feature space but have different user samples. This is the more traditional method of FL.
- **Vertical Federated Learning (VFL)**: used when datasets share duplicate sample IDs but have different feature spaces. It is also called feature-based FL. Each party contributes different features for the same individuals/clients.
- **Federated Transfer Learning (FTL)**: when both samples and feature space differ regarding datasets. This method uses knowledge from a source domain to improve learning in a target domain under federated settings.

With these data distribution models in mind, as Shanmugam *et al.* [47] expresses, federated learning architectures also typically adhere to two types of architectural models: client-server, which minimizes communication overhead by selecting a coordinator that receives updates from the clients, and peer-to-peer, that requires more computational resources for data encryption.

Regarding the task of aggregating client model updates to create a global shared model, several aggregation algorithms have been created to combine client updates, with Federated Averaging (FedAvg) being the most popular one, where the server averages the model parameters received from clients, often weighted by the amount of data each client used for training [31]. However, centralized approaches outperform FedAvg with non-IID data [37], like FedProx [27], which adds a proximal term to the local objective function, or FedDyn, which aims to handle this type of data

by dynamically updating local models. Other recent advancements include Hierarchical Federated Averaging (HierFAVG) and Federated Matched Averaging (FedMA), which improve model aggregation techniques.

Regarding a typical FL architectural process, it often involves various steps. Initially, a central server initializes a global model and distributes it to a selection of participating clients. Then, each client trains the received model on its local data for several iterations, producing local model updates (model weights or gradients), which are sent back to the central server. Privacy-preserving techniques often enhance the communication of these updates. The central server aggregates the received updates and produces an improved global model. This model is finally sent back to clients for the next round of training. This process is repeated until the model converges or a predefined number of rounds is completed.

Despite its advantages, FL faces various limitations and challenges, such as dealing with non-IID data distribution, which can degrade model performance and convergence speed; ensuring communication security so that model inversion attacks, membership inference attacks, data poisoning or privacy leaks are prevented from happening; and dealing with communication overhead and systems heterogeneity (like varying computational power, network connectivity and storage on client devices). However, addressing these challenges is essential for the practical adoption of FL.

2.4 Human-Centered AI

Human-centered artificial intelligence (HCAI) is an approach that joins the areas of AI and human-computer interaction (HCI) [11]. The primary focus of this approach is to develop AI systems that not only empower and enable human users but also prioritize ethical considerations and user agency.

More specifically, HCAI is an approach to the process of designing, developing and deploying AI systems that emphasizes human values and capabilities, ethical considerations and user-centric design, often incorporating techniques for explainable AI principles, throughout the entire AI life cycle [11, 48]. The ethical concerns are a core tenet of HCAI, as this paradigm addresses and ensures that the systems are fair, transparent and have privacy protection [56], instead of focusing solely on algorithmic performance or efficiency. HCAI also emphasizes user agency to give individuals meaningful control over AI systems, thereby promoting an effective AI collaboration [56].

Capel and Brereton's critical review of the HCAI literature yields the "Map of the field of Human-Centered AI" shown in Figure 2.1 [11]. It reveals four major quadrants: Explainable & Interpretable AI, Human-Centered Design Methods, Humans Teaming with AI and Ethical AI, plus a center area of Interaction with AI. The authors use a horizontal axis ranging from AI-led to human-values-led work and a vertical axis progressing from AI under design to AI situated in use, helping to situate the various human-AI interactive techniques within this research landscape.

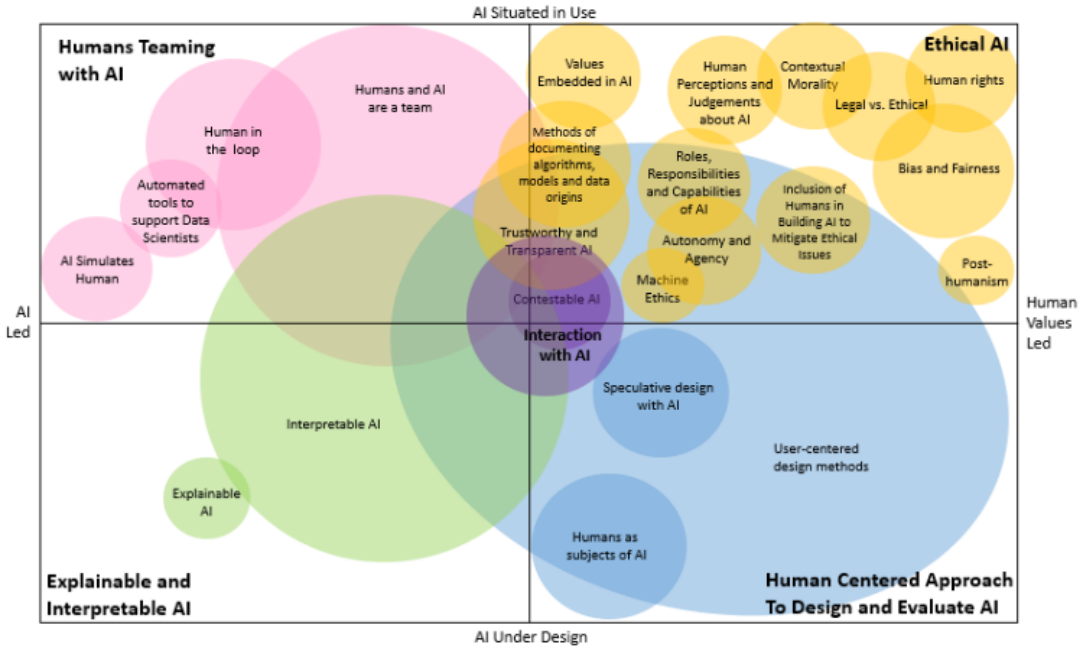

Figure 2.1: "Map of the field of Human-Centered AI," from Capel and Brereton [11] (their Figure 1). The horizontal axis runs from AI led to human values led research; the vertical axis from AI under design to AI situated in use.

Human-in-the-loop

With these principles in mind, HITL techniques emerge as a cornerstone for realizing HCAI. By embedding human oversight, feedback and intervention points directly into ML workflows, HITL ensures that user agency, ethical governance and transparency are not afterthoughts but integral design elements, guiding AI development toward more reliable, trustworthy, unbiased and easy-to-understand outcomes, fully aligning with HCAI's goals. According to Figure 2.1, HITL is situated just to the left of the central "Interaction with AI" zone, representing a more AI-led yet human-centered approach design that brings humans directly into model creation and refinement [11].

There are several methods through which human oversight can be included in ML systems:

- **Active Learning** - where the ML model queries the human expert to label points for which it is most uncertain. Consequently, the model can learn from a smaller set of high-value labeled data, thereby reducing the overall labeling effort and improving accuracy [24]. This method is crucial since labeled data is often scarce or expensive.
- **Human Oversight & Monitoring** - humans regularly review the ML system's performance, outputs and decisions, as they constantly monitor model drifts, identify biases and anomalous behavior, and validate system outputs. This constant analysis serves as a safeguard against errors.

- **Human Feedback** - humans can provide feedback on model predictions and outputs, which is then used to fine-tune models, adjust parameters or guide reinforcement learning agents toward the desired behavior. This feedback can either be corrective or preferential.
- **Human Expertise for Rule Definition or Feature Engineering** - experts can define explicit rules that control the model's behavior. Furthermore, these human experts can also transform, select or create input features relevant to the learning task, thereby improving the model's performance.
- **Human Decision Authority** - in critical applications, especially in scenarios where errors have high consequences, the human expert acts as the final decision-maker, while the AI takes on the role of intelligent assistant, providing the human with recommendations or analyzes.

As mentioned earlier, it is essential to note that the integration of HITL provides significant benefits to machine learning systems. Besides the improved performance and robustness, ML systems showcase increased trustworthiness, accountability and transparency in the acts of decision-making [24]. Beyond these benefits, HITL formalizes human engagement, defining *when*, *how* and *which* experts intervene [11], thereby strengthening ethical alignment and interpretability by incorporating human values and contextual insights into model refinement.

However, implementing HITL systems faces various methodological, technical and ethical challenges, including, for example, data quality, bias and user engagement [59]. Latency is also a concern when implementing human oversight, as it may not be suitable for all real-time applications. Finally, the cost associated with human expert time, effort and scalability, so human input does not scale linearly with the data size or number of required decisions, is also a challenge associated with HITL.

2.5 Fundamentals of Software Architecture

Software Architecture has emerged as a crucial aspect of software engineering, which focuses on the fundamental organization of a software system, made up of its components, structures, their relationships to each other, properties and the environment that surrounds them, as well as the principles that guide its design and evolution [5, 15]. It represents the high-level structure of the system, playing an essential role in ensuring that overall objectives and quality attributes are achieved. In other words, an architecture is much more than a blueprint, as it encompasses the principal design decisions and development efforts that shape a system, whether it concerns its interfaces, structural elements, behavior or overall composition into progressively larger subsystems [18].

As software becomes increasingly important in systems, a well-designed architecture becomes crucial for meeting the overall requirements and objectives [21], offering a common language for understanding a system's design. A well-thought-out architecture provides a basis for managing

complexity by decomposing a large, multidimensional system into smaller, more manageable parts with clear responsibilities and interactions. A sound architecture is also important for achieving the desired quality attributes, also known as non-functional requirements, as well as for controlling and achieving system evolution and maintenance.

Several key good practices enhance utility and clarity for end-users when developing a solid and robust software architecture. These include:

- Clearly articulate the main goals and constraints.
- Use consistent and well-defined terminology and notation.
- Use multiple views to provide different perspectives and levels of detail, taking into consideration different stakeholder concerns.
- Document key design decisions and the justifications behind them.
- Use concrete examples or reference architectural implementations.

2.5.1 Architectural Views & Models

To effectively describe software architectures, various architectural views and models are employed. An architectural view is the representation of a system from the perspective of a related set of concerns [18]. Different views can have different focuses, such as structure, process allocation or development organization.

A reference approach for visualizing software architecture is the C4 Model, created by Simon Brown [9]. The C4 Model is divided into four different hierarchical levels, each describing a system in various degrees of detail. The four levels of detail are:

- **System Context** (Level 1)
- **Containers** (Level 2)
- **Components** (Level 3)
- **Code** (Level 4)

This architectural approach makes creating, understanding and maintaining diagrams much easier, as the four levels of detail allow for reaching a wider audience according to their needs and level of understanding. This model will be utilized later in this thesis (Chapter 4) to present the proposed HITL-FL architecture.

2.5.2 Architectural Patterns

Software architecture often leverages established architectural patterns, which are general, reusable solutions for common architectural problems within a specific context in software design [10]. Examples of these patterns include Client-Server, Model-View-Controller (MVC), Microservices,

Pipes and Filters and Layered Architecture. Understanding how to apply appropriate architectural patterns is essential to guarantee a software system's quality, robustness and maintainability. With this in mind, as this research aims to propose a new architectural solution to serve as a reference architecture, it utilized established architectural concepts and principles commonly found in such patterns, particularly concerning distributed systems and modular design.

2.5.3 Architectural Attributes/Requisites

Finally, functional and non-functional architectural attributes are crucial components in software development. On one hand, the functional attributes define a system's behavior (the *"WHAT"* of a system). On the other hand, non-functional attributes define the quality requirements that a system should possess (the *"HOW"* of a system, as they influence and restrict a system's behavior) [12]. Taking this into consideration, there are many common non-functional attributes: **Security**, **Reliability**, **Availability**, **Efficiency**, **Performance**, **Scalability**, **Fault-Tolerance**, **Usability**, **Portability**, **Adaptability** and **Dependability**.

Other internal non-functional quality attributes exist, such as **Complexity**, **Implementability**, **Maintainability**, **Trust Management/Trustworthiness** and **Reputation**. The primary difference between external and internal attributes is that external quality attributes have a direct impact on the system's users. In contrast, internal ones are invisible to the user but essential to developers.

Furthermore, it is essential to consider that architectural patterns (showcased in Section 2.5.2) are also capable of addressing both functional and non-functional requirements simultaneously, thereby enabling trade-off analyzes and easier decision-making [17].

2.6 NOUS Project

The NOUS [1] project, titled "A catalyst for EuropeaN ClOUd Services in the era of data spaces, high-performance and edge computing," is a Research and Innovation Action (RIA) funded under the Horizon Europe Program (Call: HORIZON-CL4-2023-DATA-01), and serves as the primary real-world context and application domain for the architectural research developed and presented in this work [33].

The NOUS Project is highly relevant to this research. On the one hand, it creates a rich, complex and industrially significant environment where the challenges of integrating HITL methodologies with FL strategies are directly pertinent. On the other hand, its use cases, like Use Case #2 concerning Energy Prediction and Data Lifecycle Management, which will be later explored in Chapter 6, offer concrete scenarios for applying and evaluating the proposed HITL-FL architecture.

The project's core goal is to create an architecture for a European Cloud service that will act as an Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) provider. This service aims to use computational and data storage resources in various environments, such as edge devices,

[1] acronym derived from the Greek word for "intellect" or "mind" and the French word for "We"

supercomputers via the HPC Network and quantum computers. NOUS also seeks to make use of edge computing and decentralization paradigms to advance Europe's ability to process vast amounts of data and dominate the cloud market, thereby empowering the EU's digital sovereignty.

Some key concepts and technologies are central to NOUS, such as cloud architecture, HPC integration, quantum computing interfaces, edge computing, distributed/federated concepts, AI, ML, blockchain and robust data management within Data Spaces. With this in mind, NOUS' pipeline includes three different types of components:

- **Computational components (#Compute)**: responsible for executing computations by connecting to the HPC Network and creating an interface between Quantum Computers and HPC.
- **Edge components (#Edge)**: focused on communicating with edge devices (such as IoT sensors or other devices of the same type) to enhance the efficiency and computational capabilities within the E2F2C continuum. Additionally, these components also oversee the development of a FL framework.
- **Data Storage components (#Data)**: responsible for dealing with the various steps of the data life cycle, like data storage, management, sharing, standardization and exchange.

The NOUS project also addresses several challenges, including the effective use of existing HPC and Quantum Computers to solve complex problems, the promotion of edge computing services and the seamless connection and communication with European Data Spaces. Furthermore, this project recognizes the importance and faces the challenge of integrating seamless human intervention and HITL strategies into its systems, which includes designing interfaces and workflows that enable human oversight, validation and correction [14].

2.6.1 NOUS Architecture

The NOUS architecture can be perceived as the unifying blueprint that binds all the technical and governance concepts discussed throughout this work. Its design balances two complementary objectives: to keep data ownership and human oversight close to the points where the data is created and used; and to offer flexible, reliable computing power that can run any system on any device, whether it is tiny edge devices, supercomputers, or even quantum hardware. By separating these two concerns while still connecting them through shared trust, discovery and policy services, NOUS demonstrates how organizations can collaborate and provide a path toward a federated, cloud-to-edge data space, where each doesn't surrender control over their data or models.

Figure 2.2 presents the NOUS architecture. Rather than a monolithic stack, NOUS is structured, at a high level, into two major participants: the **Data Space Participant** on the left and the **Data Space Infrastructure Participant** on the right, whose interactions are governed by a neutral layer of **Common Administration Services** in the middle.

Concerning the left side of the architecture, the Data Space Participant represents an entity, such as an organization, an individual user,or an edge device, that owns, provides or consumes

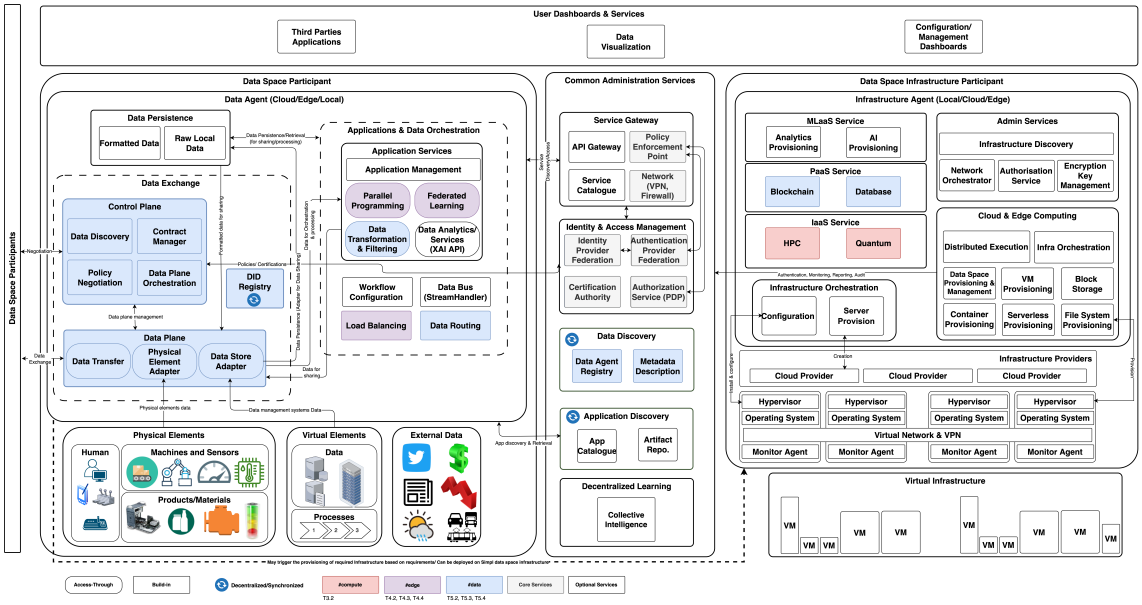

Figure 2.2: NOUS' Architecture

data or application services within the data space. Its core functionalities are realized through the **Data Agent (Cloud/Edge/Local)**, which encapsulates three key capabilities:

- **Data Persistence** - manages the storage of **Raw Local Data** and **Formatted Data**.
- **Data Exchange** - allows secure and policy-compliant data exchange. This includes a **Control Plane** (for **Data Discovery**, **Contract Management** and **Policy Negotiation**) and a **Data Plane** (for actual **Data Transfer**, supported by **Physical Element Adapters** and **Data Store Adapters**). A **DID Registry** is also utilized for managing decentralized identifiers and associated policies or certifications.
- **Applications & Data Orchestration** - hosts and manages participant-specific applications and workflows for data processing. In this case, key services include **Application Management**, support for **Parallel Programming** and **Federated Learning**, **Data Transformation & Filtering** capabilities and **Data Analytics/Services** (which may expose **XAI APIs**).
- **Workflow Configuration & Data Flow Management** - includes services for **Workflow Configuration**, **Load Balancing** and **Data Routing**, as well as a **Data Bus (StreamHandler)** which manages real-time data streams.

It is also important to note that the Data Agent ingests data from **Physical Elements** (such as Humans, Machines, Sensors and Products/Materials) and can also integrate **Virtual Elements** (internal data stores and processes) and **External Data Sources** (such as public cloud data feeds and social media).

Regarding the right side of the architecture, the Data Space Infrastructure Participant is responsible for providing the computational, storage and service infrastructure required by the Data

Space Participants (and the overall data space). Similarly to the left side, its core capabilities are offered via the **Infrastructure Agent (Local/Cloud/Edge)**, which offers:

- **MLaaS (Machine Learning as a Service)** - provides **Analytics Provisioning** and **AI Provisioning**. In other words, it offers environments for training and deploying AI models.
- **PaaS (Platform as a Service)** - offers platform services, such as the **Blockchain** and **Databases**.
- **IaaS (Infrastructure as a Service)** - delivers core computational resources, such as **High-Performance Computing (HPC)** clusters and **Quantum Computing** capabilities.
- **Admin Services** - essential services such as **Infrastructure Discovery, Network Orchestration, Authorization Service** and **Encryption Key Management**.
- **Cloud & Edge Computing Infrastructure** -manages **Distributed Execution** environments, **Infrastructure Orchestration** and provisioning for various compute and storage paradigms (like **Data Space Provisioning & Management, VM Provisioning, Container Provisioning, Serverless Provisioning, File System Provisioning** and **Block Storage**).

These services operate on top of a foundational **Virtual Infrastructure** layer, which is comprised of virtual networks/VPNs, operating systems, hypervisors and monitoring agents.

The central layer is a trusted intermediary, a man-in-the-middle, that provides essential shared services governing all interactions and contributing to interoperability within the data space. It includes:

- **Service Gateway** - an entry point for accessing services, incorporating an **API Gateway**, a **Policy Enforcement Point (PEP)** and **Network Security (VPN, Firewall)**.
- **Identity & Access Management (IAM)** - manages digital identities and access rights through the **Identity Provider Federation**, **Authentication Provider Federation**, a **Certification Authority** and an **Authorization Service (PDP, Policy Decision Point)**.
- **Data Discovery** - enables the discovery of available data sources via a **Data Agent Registry** and **Metadata Descriptions**.
- **Application Discovery** - allows the discovery and retrieval of applications and software artifacts through an **App Catalogue** and **Artifact Repository**.
- **Decentralized Learning (Support Services)** - includes services that support various types of decentralized and collective intelligence paradigms.

Furthermore, above these three layers, there are **User Dashboards & Services (Data Visualization, Configuration/Management Dashboards, Third-Party Applications)** that sit on top of, and interact with, both the Data Space Participant and Data Space Infrastructure Participant, representing user-facing interfaces and applications that consume the data within the NOUS Ecosystem.

2.6.2 Illustrative Use Cases

With the objective of testing, demonstrating and validating the developed technologies in real-world contexts, the NOUS project uses four distinct use cases, each addressing different technological challenges:

1. **Perception of Connected Vehicles using Camera Data (TIM-POLITO)**, which aims to improve road and traffic safety by integrating roadside and in-vehicle cameras to enhance the perception abilities of connected vehicles, leveraging NOUS services for data processing and communication.

2. **Energy Prediction and Energy Data Lifecycle Management (PPC)**, which wants to enhance the accuracy of energy prediction and manage the energy data lifecycle by using the NOUS framework for computational tasks, standardization of data and FL.

3. **Crisis Management and Civil Protection Platform (CS Group)**, which wants to improve the Crimson system, a European reference platform for crisis management, by utilizing the NOUS framework for robust data management and analytics.

4. **Scientific Data HPC Storage and AI Analytics (NCSR "Demokritos")** focuses on using NOUS tools and services for scientific and technological purposes, particularly in HPC environments and AI-driven statistics for scientific discoveries.

These use cases provide diverse settings to evaluate aspects and features of the NOUS project.

Chapter 3

Literature Review & State of the Art

3.1 Introduction

This chapter showcases a comprehensive analysis of the current state of the art concerning the FL architectural strategies with HITL integration in edge computing environments. It aims to establish a knowledge foundation of the existing scientific advancements, identify critical gaps and pinpoint crucial challenges that serve as motivation for the work of this thesis.

This process is grounded in the findings of a systematic literature review. The chapter begins by briefly summarizing, in Section 3.2, the developed review process to retrieve relevant academic publications and articles for this work. Following that, in Section 3.3, the current landscape of knowledge regarding the main domain research fields is the main focus, as categorization and analysis of these themes are conducted to identify pertinent findings and limitations. More specifically, the examination of the advancements and gaps in areas like direct HITL and FL collaboration, applications of FL in smart systems, ethical and responsible AI engineering, trustworthy and privacy-preserving FL, and the optimization of FL systems. This chapter concludes by articulating the core problem at hand that this thesis seeks to address, which is the need for a standardized, human-focused architectural solution for HITL in FL frameworks.

3.2 Summary of the Literature Review Process

The state of the art presented in this chapter is the result of a systematic literature review to identify and synthesize the main academic articles that are useful for this research. This review process followed the guidelines proposed by PRISMA 2020 [41] to ensure transparency and replicability by other researchers. It included the process of searching, retrieving, screening and selecting documents pertinent to the core topics of this research.

To accomplish this task, a detailed strategy was developed by employing a set of keywords, to form a query for searching publications, spanning from three conceptual domains: **Federated Learning** (encompassing terms related to distributed AI and privacy-preserving machine learning), **Human-Centered Artificial Intelligence** (including keywords for human oversight, HITL

and trustworthy AI) and **Software Architecture** (covering architectural design, patterns and principles for AI systems). The created search query was the following:

("federated learning" OR "federated machine learning" OR "privacy-preserving distributed machine learning" OR "collaborative machine learning" OR "distributed AI") AND ("human in the loop" OR "HITL" OR "HIL" OR "human oversight" OR "interactive machine learning" OR "human-AI teaming" "human teaming with AI" OR "human-AI collaboration" 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 ML" OR "trustworthy AI" OR "responsible AI" OR "ethical AI") AND ("architecture" OR "system architecture" OR "AI architecture" OR "architectural design" OR "design principles" OR "architecture patterns" OR "design patterns" OR "reference architecture" OR "architecture style" OR "architectural styles")

The initial search yielded 219 publications retrieved from four different academic databases, including IEEE Xplore, ACM Digital Library, Scopus and Web of Science. Following the subsequent screening process, conducted using Rayyan [45], where duplicates were removed, documents were screened against a sequential predefined inclusion and exclusion criteria (defined in Table 3.1) and full-text access was inquired, a total of 24 research publications were identified. Figure 3.1 summarizes the systematic flow of this literature review process for obtaining the base set of articles that make up the state of the art, which characterizes the current landscape and advancements in these fields.

Phase	Inclusion/Exclusion Criteria
1	Search engine results after executing a search query
2	Date of publication from 2018 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 "Federated Learning", "Active Learning", "Smart Cities", "Responsible AI", "Mobile Computing", "Architecture", "Distributed AI" or "Human-In-The-Loop"
11	Not published in the field of "Quantum Computing", "Computer Vision", "Environmental AI", "AI Models" or "Websecurity"

Table 3.1: Inclusion/Exclusion Criteria for Literature Review

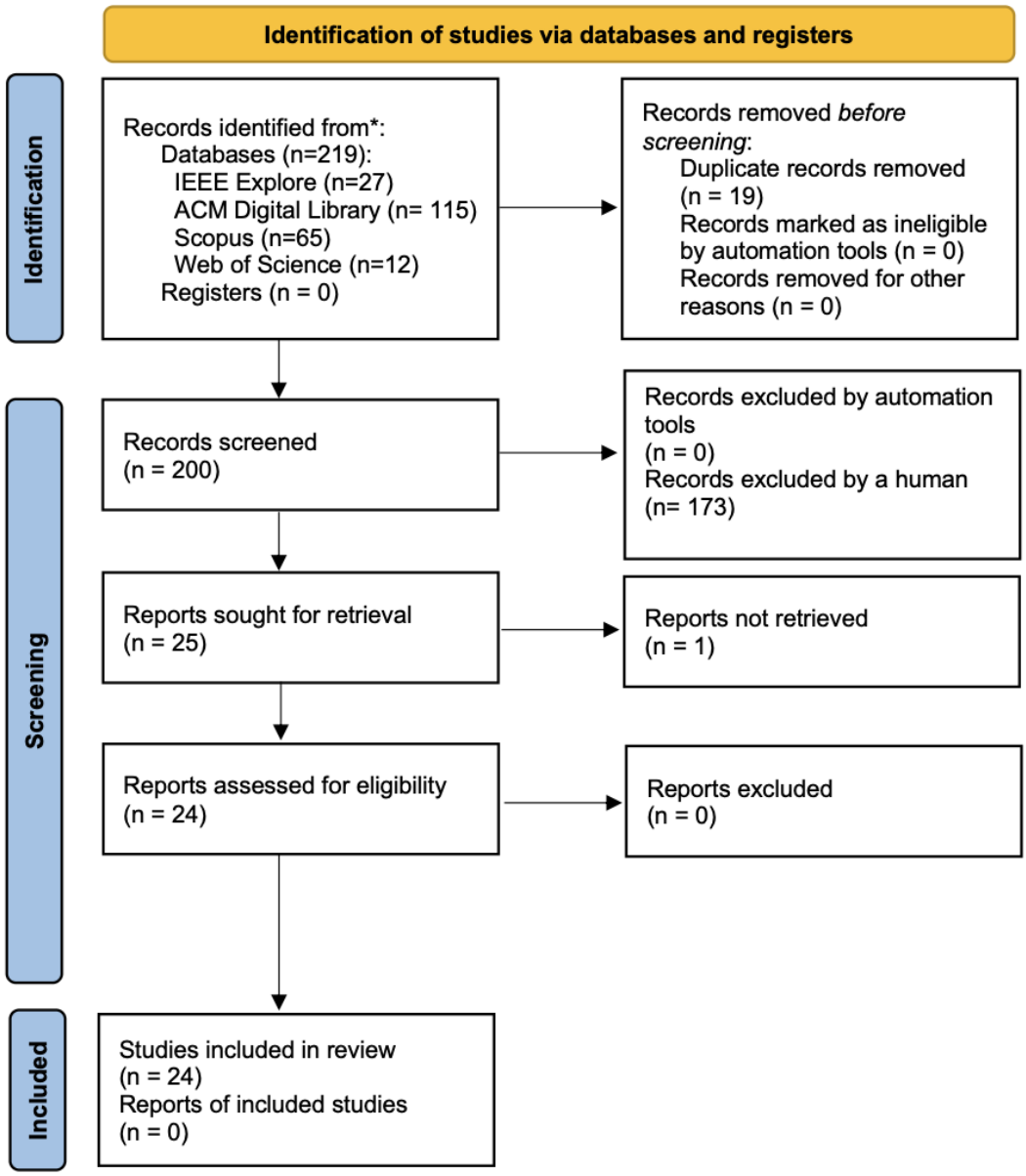

Figure 3.1: Prisma 2020 process followed in this research

3.3 Current Landscape & Key Findings

For the analysis process, the selected 24 articles were divided into five different clusters, based on their primary focus. Table 3.2 defines each cluster and its respective key focus. This research domain, as shown in Graph 3.2, is diverse, with the area of privacy-preserving FL and trustworthy AI exhibiting a greater concentration of studies. However, there is also substantial work in applications, ethical considerations, optimization and HITL approaches. Therefore, this overall distribution highlights not only the technical foundation of FL (privacy, decentralization, efficiency and

applications), but also human-focused considerations (HITL, ethics and trustworthiness), underscoring the relevance of bridging these two areas.

Human-in-the-Loop Federated Learning & AI Collaboration	Instances that showcase how HITL is included into federated learning systems, by exploring human-AI interactions, decision-making augmentation and user collaboration.
Applications of Federated Learning in Smart Systems	Federated learning's real-world applications in healthcare and smart cities, showcasing how federated models interact with smart environments.
Ethical AI & Responsible AI Engineering	Ethical considerations, legal frameworks and responsible AI engineering that ensure human involvement in AI decision-making align with ethical and regulatory standards, and ensure bias mitigation.
Trustworthy & Privacy-Preserving Federated Learning	Addressing security challenges, privacy concerns and fairness in federated learning, with a particular emphasis on the blockchain, anonymity and security protocols.
Optimization, Efficiency & Scalability in Federated Learning	Methods to enhance the efficiency, scalability and resource allocation of federated learning models, including active learning, uncertainty measurement and simulation.

Table 3.2: Article categories and respective key focus

From a systematic analysis of the 24 relevant research publications, it is possible to conclude that significant developments have been revealed across multiple dimensions, particularly in terms of architectural approaches for FL implementations that prioritize privacy and security. Moreover, advancements in HITL design approaches that integrate human oversight into AI systems, as well as innovations for improved resource utilization, scalability and overall system performance, have also been made. However, critical gaps remain, particularly concerning the unified integration of the individual areas or clusters. Although each cluster may touch on the intersection of some relevant fields, such as the standardized approaches for integrating HITL methods with FL, or the development of trustworthy AI systems that balance human oversight with privacy and efficiency demands in edge environments, a prominent gap remains.

The following subsections give an in-depth analysis of each cluster, with an explicit focus on current achievements, identified gaps and solutions that align with the research objectives of this work.

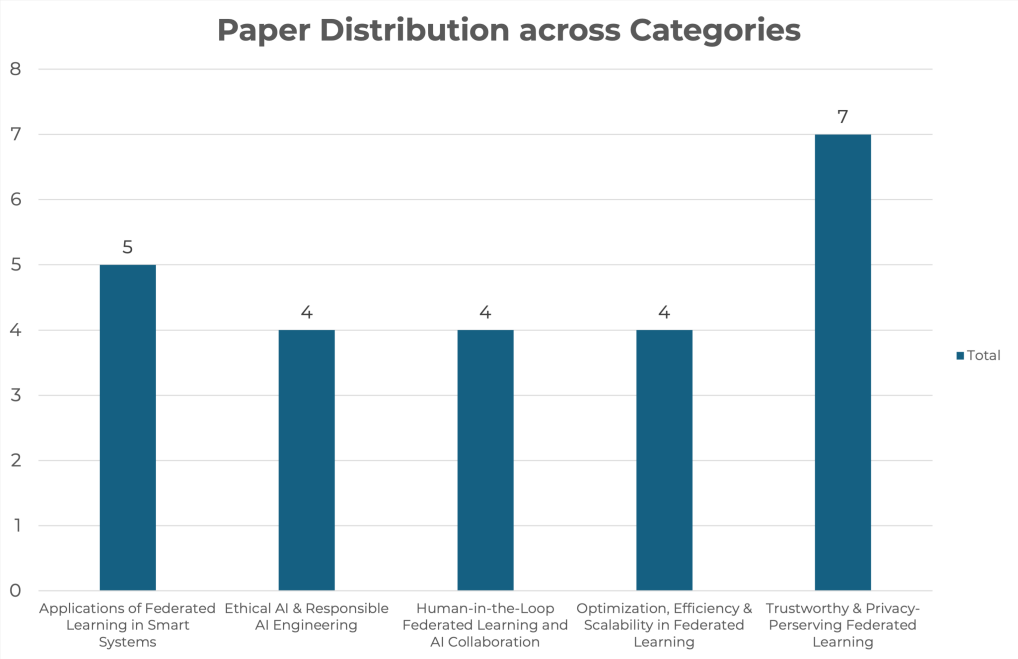

Figure 3.2: Graph showcasing Article Distribution across different Domains

3.3.1 Human-in-the-Loop Federated Learning & AI Collaboration

This research mainly focuses on integrating HITL approaches into FL architectures. The chosen papers for this cluster explored various methods for leveraging human feedback to enhance AI decision-making, thereby reflecting the multifaceted nature of human-AI collaboration that was outlined by Puerta-Beldarrain *et al.* [42]. Specific examples of these methods include experience sharing and HITL optimization for robot navigation, as presented by Moradi *et al.* [34], applying semi-supervised learning (active learning and label propagation) within a FL framework for the recognition of human activity [40] and automation of regulatory tagging by employing active learning with LLMs, presented by Al-Turki *et al.* [52]. These examples demonstrate how incorporating human intervention into FL systems, which already keep data decentralized while allowing models to learn collectively across edge distributed environments, as shown by Presotto *et al.* [40], often with privacy preservation as the key goal [42], can help refine AI models [40, 52], improve classification accuracy, even with minimal labeled data [40], and ensure that the outputs generated by AI are aligned with real-world requirements [40, 52].

Despite these advancements, limitations need to be considered. In large networks, reliance on user input may not always be practical or scalable. As a consequence, potential issues may arise, such as delays in responses [40] or the introduction of biases and even malicious feedback from human participants, which can impact the overall effectiveness of a system, as explicitly addressed by [34]. Additionally, current approaches also face challenges in scalability and computational overhead on edge devices and overall system scalability.

3.3.2 Applications of Federated Learning in Smart Systems

Recent research indicates that FL has become a significant paradigm in smart systems, particularly in areas such as healthcare [1, 36, 51] and smart cities [57]. This approach enables decentralized model training without exposing raw data, which is an important benefit in privacy-preserving FL architectures [36, 51]. The effective implementation of FL in various real-world applications, such as healthcare digital twins [19], trustworthy smart city environments [57], interactive health monitoring [1] and lung nodule segmentation on CT scans [51] showcase how data security can be ensured while also achieving high model performance, sometimes even surpassing the performance of centralized models, that were trained under strict privacy constraints, like Tenescu *et al.* [51] stated. Additionally, as explored by Joo *et al.* [19], FL systems incorporating blockchain technology offer integrity guarantees for the global aggregation model, protecting against security attacks.

However, current FL frameworks still struggle with challenges. Heterogeneous data distributions across clients [19, 36, 51] can lead to suboptimal model updates and computational and communication overheads, with varying impacts depending on the chosen FL algorithm (like FedDyn or Model Soup [51]), can limit real-world deployment, especially in environments where resources are scarce [1, 36]. Furthermore, while blockchain can aid in security, FL systems still face another challenge: the inability to fight against more sophisticated data privacy attacks that require further protective measures, beyond just basic integrity checks [19]. The integration of HITL is another area that still needs more development, as although systems proposed by Alawadi *et al.* [1] show potential, there is still a lack of standardized mechanisms for incorporating HITL in real-world federated learning deployments. However, active learning is an increasingly explored approach for efficiently incorporating human feedback. For instance, Tenescu et al. [51] showcased the effectiveness of active learning in labeling data by selecting meaningful subsets, thereby improving model performance.

3.3.3 Ethical AI & Responsible AI Engineering

There have been significant advancements in recent years in the field of ethical and responsible AI, with a strong focus on obtaining a tangible AI governance with risk management and transparency frameworks, as Lu *et al.* stated [29, 30]. Additionally, there has been a growing need for AI systems to be designed with accountability mechanisms to follow legal standards such as the EU AI Act, as noted by Urquhart *et al.* [55]. Efforts have also been directed towards integrating risk assessment processes [29, 30, 55] and human control mechanisms [29, 30, 55] into governance systems to improve their trustworthiness, sometimes framed within specific legal contexts [55]. Other key system characteristics like explainability, transparency and privacy preservation have also been explored, as there have been recent developments in architectures that introduce techniques for more interpretable outputs for stakeholders [29], ensure data protection [29, 30] in

decentralized edge computing environments and address the challenge of creating high-quality labeled datasets through methods that allow the collaboration of humans with AI, like the DataScope proposed by Alsaid *et al.* [2].

With this in mind, there is still an evident lack of standardized enforcement mechanisms in AI governance frameworks, which can lead to inconsistencies in AI accountability across different domains [29, 30]. Besides that, due to evolving AI models, anticipating emerging risks can be a struggle and explainability techniques can be too complex and challenging for non-expert users to interpret [55]. Finally, concerning the integration of human oversight mechanisms, their success depends almost exclusively on the expertise and clarity of intervention protocols [55].

3.3.4 Trustworthy & Privacy-Preserving Federated Learning

As modern systems introduce new privacy risks or aggravate already existing ones [25], current research, like the work by Yang *et al.* [60] or Lo *et al.* [28], explored the enhancement of the trustworthiness, security and privacy-preserving capabilities of FL systems. Blockchain technology is one of the most explored solutions, proposed for its potential to mitigate the risks of model tampering and stopping malicious updates, through secure aggregation and consensus protocols, like Yang *et al.* [62] showcased in 2023, and to increase auditability through various mechanisms like the inclusion of immutable data-model provenance registries [28]. Beyond core FL training, techniques like using trusted third parties in secure data handling are explored, as Rimez *et al.* [46] demonstrated. Furthermore, architectural solutions focus on securing the network infrastructure itself by introducing features like containerized client environments or selecting secure protocols like MQTT [7], for healthcare deployments. To protect sensitive data, while maintaining learning performance, researchers also consider mechanisms like multi-party computation and homomorphic encryption, often described in the context of border privacy-preservation [60]. Regarding fairness concerns, techniques like weighted data sampling and other data bias mitigation methods are used [28].

However, significant challenges remain in introducing privacy-preserving FL systems. On the one hand, computational overhead and increased computational complexity occur from integrating blockchain or differential privacy [60, 62]. On the other hand, ensuring trustworthiness requires continuous human monitoring and oversight. While systems like SPATIAL [39] propose architectures with various features, such as AI sensors and dashboards, to combat these issues, this type of monitoring increases system complexity [39]. Additionally, there is an overall lack of interpretability in FL systems and current federated techniques still only address a small subset of AI privacy risks.

3.3.5 Optimization, Efficiency & Scalability in Federated Learning

Current research in this category has focused on bettering the low efficiency and scalability characteristic of FL systems by optimizing client selection, reducing computational overhead, and improving deployment feasibility. Smart client selection is a prime example of this research, where

uncertainty-based selection methods, such as those proposed by Li *et al* [26], aim to reduce training costs and maintain model performance. Another key example is FL simulation engines, which allow pre-deployment testing to assess resource efficiency and model performance, as demonstrated by Arroyo Galende *et al.* [3]. Another strategy that has also emerged is multi-objective optimization, which, as explored by Kang *et al.* [20] can balance trade-offs between privacy, utility and efficiency. Finally, federated active learning should also be mentioned. For instance, as Mittal *et al.* [32] showcased, besides allowing the integration of human feedback, this method reduces labelling efforts, thereby showcasing success in decentralized environments, particularly in disaster responses.

However, these approaches still struggle with limitations that need to be considered. Examples of these challenges include working in highly dynamic and heterogeneous environments where the availability of clients and data varies, often encountered in emergency response contexts [32] or large-scale deployments like smart grids [26]. Furthermore, there is often the inability to fully replicate real-world network constraints, like in simulation-based evaluations [3]. Many methods also rely on predefined constraints, like optimization frameworks [20] or simulation setups [3], that do not capture the complexity of real-world scenarios. Finally, there is a lack of HITL mechanisms that can dynamically adjust overall operational strategies based on real-time conditions.

3.4 Conclusion

The current state of the art reveals significant advancements in current FL systems, as well as notable gaps, particularly in the integration of HITL methodologies. As FL architectures have explored methods of enhancing privacy, scalability and robustness, incorporating human oversight still remains a crucial challenge, as it is beneficial for refining model accuracy, interpretability and ethical alignment with real-world requirements. Furthermore, other challenges persist with computational overhead, scalability and the overall practicality of leveraging real-time human feedback in diverse and dynamic environments. With this in mind, these gaps in the research literature present both a challenge and an opportunity to fill them with a human-focused architectural solution that responds to such limitations but preserves privacy, efficiency and trustworthiness.

Chapter 4

HITL-FL Architecture

4.1 Introduction

This chapter outlines the proposed reference architecture framework for integrating HITL methodologies within FL systems. As previous chapters have shown, promising efforts have been made to evolve FL systems' privacy, scalability, efficiency and trust. Regarding HITL, progress has also been demonstrated in enhancing AI decision-making and leveraging human opinions and feedback. However, the unified integration of human oversight approaches with FL frameworks remains a critical gap, as there is a lack of standardized methods that incorporate robust HITL approaches in this type of federated systems, in resource-constrained or dynamic edge environments [34], capable of maintaining and taking into account factors like privacy, scalability and trustworthiness.

Notable challenges like mitigating potential latencies or biases introduced by human feedback, ensuring continuous human monitoring to guarantee trustworthiness, or even guaranteeing that human oversight is not too complex for non-experts, as mentioned in the challenges of sections 3.3.1, 3.3.4 and 3.3.3, respectively, from chapter 3, and formulated as design problems that are going to be addressed further in this chapter, make this integration non-trivial. Additionally, it is essential to keep in mind the clear objective that this architecture is trying to achieve, which is to not only ease the inclusion of human expert oversight, but also ensure that this expert feedback contributes meaningfully to a system's accuracy, ethical principles and overall trustworthiness, allowing its users to use the system confidently, while simultaneously maintaining FL's core characteristic privacy benefits.

It is also important to note that this reference conceptual architecture will serve as a means to extend the proposed NOUS architecture. It will do so by not only mapping containers, components and insights to existing elements within NOUS, but also by creating new microservices and components to facilitate the integration of HITL. The mapping of the architecture will be discussed in the subsequent Chapter 6.

Considering this, these challenges are the main motivation behind developing the HITL-FL architecture presented in this chapter. There is a need for a structured architectural solution that

effectively addresses how human expert feedback can be integrated into FL frameworks, thereby creating a human-centric, reliable and secure AI system. To achieve this, this proposed reference architecture is designed with the following high-level goals in mind:

1. To propose a standardized and reference architectural framework for integrating HITL mechanisms into FL systems.
2. To enhance the trustworthiness, transparency and accountability of FL systems by including human oversight, expert validation and an auditable feedback process, ensuring that the human user has the confidence to interact with the system.
3. To guarantee privacy and security by design, allowing human interaction without compromising the core principles of FL.
4. To ensure that the system can manage an increasing client workload and demand for human expert interaction and oversight, without the decline of its performance.
5. To empower human experts with decision-making and influence over AI decisions during the FL process, from model validation to ethical governance, thus preventing AI from making all the decisions.

These goals aim to achieve an architectural design with clear responsibilities and interaction patterns for HITL-FL to integrate FL techniques with human-centric AI. This architecture will also be a fundamental tool to respond to the research questions presented in Chapter 1.

To elaborate on the proposed architecture, this chapter is structured as follows:

- **Section 4.2: Design Principles & Architectural Requisites** outlines the core design principles that characterized the architectural development, as well as key functional and non-functional requisites that the architecture aims to satisfy and will be the base for later validation.
- **Section 4.3: Architectural Overview** presents the architecture in question using the C4 model notation, providing a multi-layer understanding of this framework through a diagram of the system context (level 1), another one of its decomposition into its key containers (level 2) and component diagrams that further details selected containers (level 3). It also details the core HITL-FL design problems and architectural patterns used.

4.2 Design Principles & Architectural Attributes

To design the conceptual HITL-FL architectural framework, a set of core design principles was derived from the high-level goals proposed in Section 4.1 and defined following the overall objectives to be achieved through this architecture. Furthermore, to ensure that the proposed framework showcased fundamental functional capabilities and quality characteristics, attributes (functional and non-functional) were also considered. Both of these sets of guidelines form the basis for understanding and validating the architecture.

4.2.1 Design Principles

The fundamental design principles that influenced the development of the architecture and its design choices are detailed in Table 4.1. These are not presented in a ranked order.

Table 4.1: Core Design Principles for the HITL-FL Architecture

Principle	Description & Rationale for HITL-FL Context
Human-Centrality	Prioritizing human oversight and control in systems, specifically in AI systems, thereby ensuring that humans act as the primary actors, their capabilities are expanded and their values are reflected in the system. This principle aligns with the need for trustworthy AI as highlighted by the HCAI philosophies [49].
Modularity & Separation of Concerns	Structuring an architecture into distinct coupled components and containers, each with their responsibilities and interfaces. As a consequence, factors such as maintainability and the addition or update of features, due to the evolution of systems, are ensured for complex HITL-FL frameworks [36].
Privacy & Security by Design	Integrating privacy-preserving technologies and other security mechanisms from the outset, such as the inclusion of secure communication channels, data encryption and access control for HITL mechanisms, is crucial to upholding the core privacy principles of an FL system.
Transparency Assurance	Designing a system to offer its users a clear visibility of its processes, model states and rationale behind AI and human decisions. Logging key events, enhancing accountability and making debugging easier are mechanisms that create auditability and achieve the task of transparency.
Efficiency & Scalability	Designing a system that takes into account the performance and implications of HITL, thereby aiming for efficient integration of human oversight in the system, making it able to accommodate growing numbers of clients, data volumes and other expert interactions.
System Interoperability	Promoting interoperability capabilities, between different components not only within the proposed standard system, but also with elements in broader ecosystems (like the NOUS project).

4.2.2 Architectural Attributes

As stated in the introduction of Section 4.2, a set of key architectural attributes was defined, during the development process, derived from the insights of the papers of the literature review, to ensure that the proposed architecture addresses and achieves its identified goals. These attributes were categorized into two groups: functional and non-functional. They will also serve as criteria used for validating the architecture's design in Chapter 6.

4.2.2.1 Functional Attributes

The proposed HITL-FL architecture shall support the functional attributes:

FR.001: Federated Model Training

- **Priority:** Critical
- **Description:** Clients/Users shall be able to train local models (shareable and personalized) on their private data, as well as update and fine-tune them based on the aggregated server model.
- **Motivation:** To enable model learning from decentralized private data sources while preserving data privacy and allowing for generalized and user-specific model improvements.

FR.002: Local Data Storage

- **Priority:** High
- **Description:** Training data, including private user data and locally generated labels, shall be stored on a local storage system accessible only to the respective client.
- **Motivation:** To ensure clients can access their data directly and that it remains private, thereby enabling local model training and personalization.

FR.003: Secure Model Update Exchange

- **Priority:** Critical
- **Description:** Clients shall securely send model updates, such as gradients or weights, along with anonymized metadata, to the central server. Consequently, the server sends model updates back to each client.
- **Motivation:** To protect the confidentiality and integrity of AI model parameters and associated metadata during transit between clients and the central server, leading to the prevention of eavesdropping or tampering.

FR.004: Centralized Model Aggregation

- **Priority:** Critical
- **Description:** The central server shall aggregate the model updates from the various clients to create an improved global model.
- **Motivation:** To combine learnings from decentralized datasets into a single, more robust and generalized global model.

FR.005: Model Trustworthiness and Selection

- **Priority:** Critical
- **Description:** The central server shall evaluate the aggregated global model against defined performance benchmarks and perform trustworthiness checks. As a result, the server shall select the most suitable model (either the current improved version or a stable prior version) and distribute it to the clients.

- **Motivation:** To ensure that only high-quality, reliable and trustworthy models are distributed to clients and to prevent the propagation of poorly performing or potentially harmful models.

FR.006: Global Model Predictions and Uncertainty Outputs

- **Priority:** High
- **Description:** The central server shall utilize the aggregated model to generate predictions on a representative dataset, outputting a list of predictions along with their respective uncertainty and confidence scores.
- **Motivation:** To provide data points (predictions with associated confidence or uncertainty scores) that can give more context to active learning strategies and help human experts identify samples requiring review.

FR.007: Human Expert Oversight and Control

- **Priority:** Critical
- **Description:** Human Experts shall view the aggregated model state, performance metrics and contextual summaries (through a dashboard), allowing them to securely provide input on the development process, such as approving or rejecting the model, mandating retraining, redefining fairness constraints, flagging data, requesting rebalancing or adjusting system configuration, for example.
- **Motivation:** To enable meaningful human intervention in the AI tasks, ensure models align with human values, ethical considerations and domain expertise, and provide mechanisms for correcting and guiding the AI system.

FR.008: Active Learning Selection and Annotation

- **Priority:** High
- **Description:** The system shall identify the most informative predictions for human experts to review and annotate. Besides sending the metadata of each prediction to a dashboard, secure annotation requests are made through a gateway to the appropriate experts.
- **Motivation:** To optimize human expert time by reviewing the most impactful samples, improving model performance and data quality efficiently.

FR.009: Expert Feedback Integration

- **Priority:** Critical
- **Description:** Labeled data and high-quality feedback shall be utilized to refine the central, aggregated, global model.
- **Motivation:** To incorporate valuable human knowledge and corrections directly into the model improvement process. Additionally, enhancing model accuracy, robustness and alignment with expert understanding.

FR.010: Ethical Validation and Alerting

- **Priority:** Medium
- **Description:** The system shall detect deviations from ethical guidelines or performance degradations and send alerts.
- **Motivation:** To proactively identify and flag potential ethical issues, biases or performance problems, enabling human intervention and ensuring the system operates within acceptable ethical boundaries.

FR.011: Auditable Blockchain Logging

- **Priority:** High
- **Description:** The system shall log cryptographic hashes and records of validated model versions, timestamps, human expert inputs, validation results and key operation events to a blockchain.
- **Motivation:** To provide a secure, transparent and tamper-proof audit trail of critical system activities and artifacts, and support accountability, traceability and trust in the system's operations and evolution.

FR.012: Historical Data Rollback

- **Priority:** Medium
- **Description:** The central server shall query the blockchain for previous model performances, validate the current model state and review past expert decisions to facilitate a model rollback.
- **Motivation:** To enable recovery from incorrect or faulty model states by reverting to a previously known good version, thereby enhancing system resilience and reliability.

FR.013: End-User Expert Review and Annotation

- **Priority:** High
- **Description:** End-user experts shall review the outputs from the personalized model and provide corrections and feedback to update the data stored locally.
- **Motivation:** To further enhance the performance and relevance of personalized models, direct user feedback and corrections at the local level, thereby improving individual user experience and model utility.

FR.014: Local Label Propagation

- **Priority:** Medium
- **Description:** Client systems shall automatically propagate existing expert-annotated and validated labels to similar unlabeled local data.

- **Motivation:** To increase the amount of labeled data available locally with minimal additional human effort, and improve the quality of local and personalized models.

FR.015: System Tradeoff Analysis

- **Priority:** Medium
- **Description:** The system shall load and present to human experts on the dashboard the results of offline or periodic quantitative tradeoff analysis, enabling them to make more informed decisions.
- **Motivation:** To provide Human Experts with crucial data and insights regarding system-level tradeoffs, enabling them to make more informed decisions about system configuration, operational constraints and policy settings.

4.2.3 Non-Functional Attributes

This architecture is also defined by the following non-functional attributes:

NFR.001: Security

- **Priority:** Critical
- **Description:** Ensure privacy and confidentiality of sensitive data (client data and model updates), model integrity and authorized access to system functions, and prevent data attacks (like data poisoning).
- **Motivation:** To protect the privacy of users, maintain data integrity, prevent unauthorized model manipulation or access, and ensure compliance with regulations of data protection, with the overall aim of fostering user trust and ensure system privacy and reliability.

NFR.002: Trustworthiness & Explainability

- **Priority:** Critical
- **Description:** Ability for the users to trust and effectively interact with a system, particularly one that is perceived to make reliable, ethical and transparent complex decisions in sensitive environments, fostering the confidence from all users.
- **Motivation:** To ensure human experts can oversee, validate and overall interact confidently with the AI system, thereby leading to more ethical, robust and accepted AI outcomes, allowing all users to trust the system's decisions and behaviors.

NFR.003: Auditability

- **Priority:** High
- **Description:** Provide a clear, immutable record of not only all events and process steps, but also model versions, metadata and key expert decisions [30], so that at all times, there is a record of what happened or who did what, enabling the process debug and build trust.

- **Motivation:** To enable effective debugging, traceback capabilities and ensure accountability for each action and decision taken, leading to trust being built by providing a transparent and verifiable history of operations.

NFR.004: Usability

- **Priority:** High
- **Description:** Ability for Human Experts to interact with the system and their respective allocated components, so as to understand model behavior, perform validation and provide feedback easily.
- **Motivation:** To maximize the effectiveness of HITL processes, ensure high-quality expert input, reduce expert fatigue and allow the operation of the HITL system.

NFR.005: Adaptability & Maintainability

- **Priority:** High
- **Description:** The ease with which the system can be modified and corrected to incorporate new FL algorithms, expert tasks, ethical guidelines or integrate with other external systems.
- **Motivation:** To guarantee the long-term viability and relevance of the system, allowing it to evolve with changing requirements, technologies and ethical standards at a low cost and with little performance effort for ongoing support for new updates.

NFR.006: Scalability

- **Priority:** Medium
- **Description:** The system's ability to handle a growing number of clients, data volume, complexity of models and interactions.
- **Motivation:** To support the system's growth, allowing it to behave with an increasing load without degradation in performance, user experience or operational stability.

NFR.007: Performance

- **Priority:** Medium
- **Description:** The system shall showcase adequate responsiveness regarding the time needed for an aggregation round or for an expert task to be completed, as well as sufficient throughput (number of clients supported for example).
- **Motivation:** To ensure, although "latency" is inherently introduced by HITL [62], timely model updates, efficient human workflows and the prevention of bottlenecks.

Besides the presented quality attributes, other attributes could be considered, but are of less relevance to the architecture in question, such as, for example, **Reliability and Availability** (asynchronous nature of HITL processes, the nature of federated learning and implicit standard deployment practices like load balancing and redundancy) and **Fault-Tolerance** (blockchain's rollback

mechanism, asynchronous operations' buffering against components that are down and the nature of FL that tolerates client dropouts),

4.3 Architectural Overview

This section of the chapter provides a detailed description of the proposed HITL-FL reference architecture.

It is important to note that the design process for creating this architecture was entirely iterative and divided into *four* distinct phases. Starting from a base horizontal FL architecture, showcased by Yang *et al.* [61], each phase had the main task of collecting all the insights from several papers, selected in the literature review process (detailed in Chapter 3), and add new modules, rules, guidelines and interactions that would increasingly add more detail to the architecture and point it towards the desired goal.

However, the created architecture was designed using a personalized notation. Therefore, with the aim of effectively communicating its design structure at different levels of abstraction, the **C4 model** notation, created by Simon Brown [9], was adopted, allowing the creation of a narrative approach that utilizes diagrams representing a progressively increasing zoom into detail.

For this work, three out of the four C4 levels were used to describe the architecture:

- **Level 1 (System Context Diagram)** - provides the highest level view of the HITL-FL system, representing it as a black box that interacts with external users/actors and other systems. Its objective is to understand its scope and relationships with its surrounding environments.
- **Level 2 (Container Diagram)** - zooms into the HITL-FL system, decomposing it into its major deployable units (containers). This level of detail outlines the high-level technological choices, the primary responsibilities of each container, and their relationships with each other and other containers.
- **Level 3 (Component Diagrams)** - selected key containers are further decomposed into the components that form them, thereby detailing the specific modules and how they collaborate.

Taking this into consideration, the following subsections introduce each of these diagrams, their elements and their respective relations, for an easy and comprehensive understanding of the architecture. Moreover, Section 4.3.4 subsequently outlines how this architectural framework design addresses the core design problems, central to the effective integration of human oversight within FL systems, and Section 4.3.5 mentions how reference architectural patterns were applied to this architecture.

4.3.1 System Context Diagram

The C4 model begins with the first level, the **System Context diagram**, where a high-level view of the architecture is presented. From Figure 4.1, the main system in question, the Federated

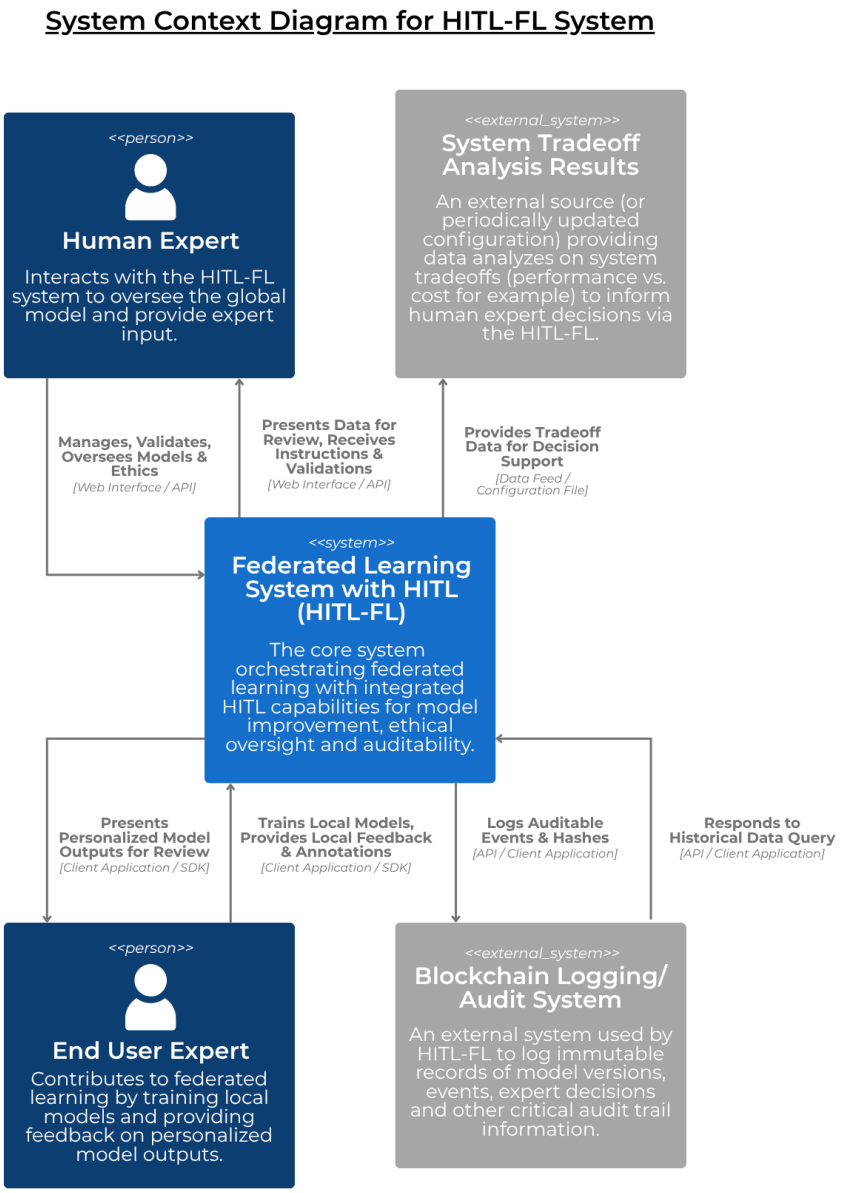

Figure 4.1: System Context Diagram for the HITL-FL System

Learning System with HITL (HITL-FL), is considered as a single black box that interacts with system users and other external systems. Besides that, the interactions between the central system and the other participants are also showcased.

4.3.1.1 The HITL-FL System

It corresponds to the central core system that is being designed. Its primary responsibility is to coordinate the federated learning process while integrating HITL capabilities. More specifically, this system is responsible for managing the distributed model training, aggregating client updates to improve the global central model, including human oversight, ensuring that ethical policies are upheld and contributing to maintaining a record of all operations and metadata [30].

4.3.1.2 External Actors

The core system interacts with two main types of human actors:

- **Human Expert** - this actor represents humans who possess the domain knowledge to interact with the HITL-FL system, overseeing the performance of the global model, validating its outputs, analyzing its metrics, providing user inputs, and making critical decisions regarding the central global model refinement and ethical compliance [3]. Therefore, their primary interaction is to manage, validate, and oversee models and ethics (through dedicated interfaces). In return, the system presents the aggregated model outputs and state, as well as performance metrics and contextual summaries, to the human experts for subsequent review and instructions.
- **End User Expert** - this actor represents client-side users who participate in the local federated learning process and have the expertise relevant to reviewing their local data and model performance. They contribute by training local models and providing feedback and annotations on personalized model outputs. The system, in turn, presents the outputs of the local, personalized model for their review and subsequent data labeling [40].

4.3.1.3 External Systems

Finally, the HITL-FL also relies on and interacts with the following external systems:

- **System Tradeoff Analysis Results** - represents an external source or a periodically updated configuration that provides crucial trade-off data for decision support by **Human Experts**. The results from an offline or periodic quantitative analysis, which explores the potential trade-offs between key objectives such as privacy leakage, utility loss and performance versus cost across various system configurations, are loaded into the HITL-FL system [20]. The data given by this system is often visualized in the form of, for example, tradeoff curves or possibility frontiers, as stated by Kang *et al.* [20], thereby allowing the **Human Experts** to make informed decisions regarding the selection of specific operational system configurations or the definition of acceptable performance thresholds (like minimum utility thresholds or maximum latency/cost budgets).
- **Blockchain Logging/Audit System** - an external blockchain system utilized by the HITL-FL system for maintaining an immutable and verifiable record of various types of information like timestamps, historical performance and fairness metrics, trusted model state hashes, human expert decisions and triggered actions [25, 28, 62], relevant contextual summaries and other key operational events (aggregation rounds, errors, HITL gateway interactions, client-side local active learning metric summaries and parameter changes) [55]. It is essential to note that while the HITL-FL system logs various events and hashes them to this external system, thereby creating an ethical black box [29, 30] for auditability, it can

also query the system to obtain historical data for model rollback, performance analysis or accountability investigations.

This System Context Diagram establishes the scope of the HITL-FL system and the key external entities it depends on or serves, setting the stage for a more detailed decomposition in the subsequent Container Diagram.

4.3.2 Container Diagram

Taking a deeper look into the Federated Learning System with HITL (HITL-FL) (from Figure 4.1), the Container Diagram in Figure 4.2 showcases a high-level view of the various containers that make up the building blocks of the system in question. In this case, each container represents a modular runnable unit (like a data store, a distinct application or a collection of related services) that implements this system's functionalities [36]. The goal of this level is to identify the major technological components and their respective responsibilities and interactions within the overall HITL-FL ecosystem.

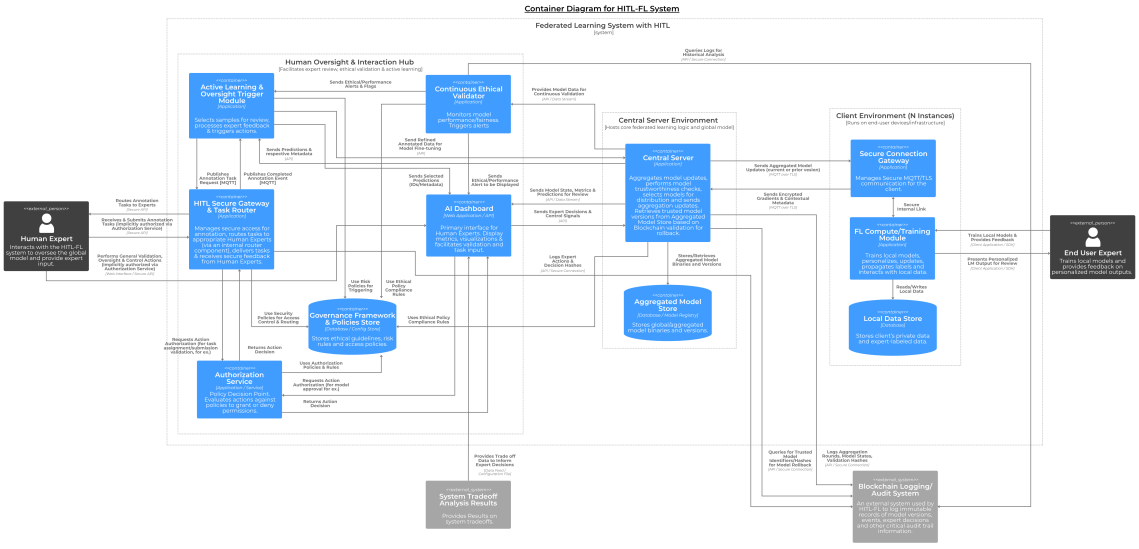

Figure 4.2: Container Diagram for the HITL-FL System

As shown in Figure 4.2, the HITL-FL system is divided into three logical environments/hubs, each encapsulating a set of related containers responsible for performing specialized tasks aligned with the hub's overall focus. These containers showcase both intra-environment interactions, with containers that are in the same environment, and inter-environment interactions, where communication between different logical hubs occurs. The three logical environments are as follows:

- **Client Environment (N Instances)** - represents the distributed edge environments where data processing and the training of the models happen.
- **Central Server Environment** - hosts the core FL orchestration logic and the global model.

- **Human Oversight & Interaction Hub** - groups containers that are responsible for facilitating human expert review, ethical validations and active learning processes.

In the following Sections 4.3.2.1 to 4.3.2.3, these environments will be presented in more detail, as their containers and interactions will be showcased.

4.3.2.1 Client Environment (N Instances)

This environment simulates a client's edge instance [34], where local data is stored, model training is performed and data processing, such as personalized fine-tuning, local model updates and label propagation, occurs, all while ensuring data privacy. The key components of this environment are shown in Table 4.2.

Table 4.2: Summary of Containers within the Client Environment

Container Name	**Type**	**Primary Functionality**
Secure Connection Gateway	Application	Manages Secure MQTT/TLS client-server communication.
FL Compute/Training Module	Application	Trains local models, personalizes, updates, propagates labels and interacts with local data.
Local Data Store	Database	Stores client's private data and expert-labeled data.

The containers within the client environment collaborate to manage local data and model training:

- **FL Compute/Training Module** - consists of the core computational unit on the client side, being responsible for training local models on the client's private data, fine-tuning these models based on the characteristics of the local data, integrating feedback from the **End User Expert,** process received global model updates from the central server and propagate labels within its local dataset to increase labeled data (and thereby increase the quality assurance of the predictions). This container directly interacts with the **Local Data Store** for accessing and storing data, as well as with the **End Human Expert** for iterative refinement of the local models.

- **Secure Connection Gateway** - works as a middleman for the secure communication channels, using MQTT over Transport Layer Security (TLS) [7, 60], between the various individual client instances and the **Central Server Environment.** On the one hand, it handles all outgoing data, including model gradients and weights, as well as anonymized contextual metadata (such as data source characteristics, timestamps, the quantity of local data points used for updates and other relevant data) [46]. On the other hand, it handles the incoming aggregated global models from the central server, which are then passed to the **FL Compute/Training Module** for further processing.

- **Local Data Store** - securely stores all client-specific data. Examples of this data include raw private training data, expert-labeled data and generated expert annotations. It provides the data for the **FL Compute/Training Module** to perform local training and data personalization/fine-tuning.

One important aspect that needs to be highlighted is the encryption of the shared model updates and the anonymization of the contextual metadata shared with the central server [46], which is performed by the containers in this environment. To protect data content, techniques such as differential privacy may be applied to the gradients before they are sent. In addition to this, the transmission of data, performed by the **Secure Connection Gateway** container as previously mentioned, occurs over an MQTT communication channel secured by TLS [7, 60], which ensures confidentiality and integrity during data transit and helps prevent gradient leakage attacks [60]. This preoccupation with privacy contributes to answering Research Question RQ3.

4.3.2.2 Central Server Environment

From the diagram showcased in Figure 4.2, the **Central Server Environment** can be perceived as the central backend hub for the HITL FL system, as it is primarily responsible for handling the FL process, managing the aggregated global model and its inputs, and processing the insights and other outputs from the **Human Oversight & Interaction Hub.** Essentially, it hosts the logic for model aggregation, validation and distribution. The key containers of this environment are listed in Table 4.3.

Table 4.3: Summary of Containers within the Central Server Environment

Container Name	Type	Primary Functionality
Central Server	Application	Aggregates model updates, performs model trustworthiness checks, selects models for distribution and sends aggregation updates. Retrieves trusted model versions from the Aggregated Model Store based on Blockchain validation for rollback.
Aggregated Model Store	Database/Model Registry	Stores global/aggregated model binaries and versions.

The containers within this hub collaborate as follows:

- **Central Server** - the core application responsible for the orchestration of the FL cycle. It is responsible for receiving encrypted model gradients and contextual metadata, from the various clients (via their **Secure Connection Gateways**), and creating an improved global model, resulting from the aggregation of the sent model updates. Moreover, this container also has the task of performing model trustworthiness and ethical checks against predefined governance guidelines and policies (stored in the **Governance Framework &**

Policies Store of the **Human Oversight & Interaction Hub**) [30], to determine and select the most suitable model version to distribute back to the involved client instances. This model can be either the newly improved version or a validated, trusted version stored in the **Aggregated Model Store** for rollback purposes and accessed by querying the blockchain for trusted model hashes/identifiers [19]. The central server also provides crucial information to the **Human Oversight & Interaction Hub**, specifically to components like:

  - **AI Dashboard** - provides the aggregated model state, global performance ethics, contextual summaries and samples/predictions for human review [42].
  - **Continuous Ethical Validator** - provides current model performance metrics and state for continuous validation.
  - **Active Learning & Oversight Trigger Module** - provides predictions and respective metadata (uncertainty scores, confidence scores and client-made contextual flags) [36, 51].

  This sharing of information provides an easy way for human experts to review the model's performance, offer their insights, make decisions and intervene when necessary [3]. Finally, this container also logs various information to the **Blockchain Logging/Audit System**, highlighting the cryptographic hash of the validated aggregated model, the timestamp, validation results, key performance and fairness metrics, relevant contextual summaries, records of human expert decisions [25] and triggered actions (model approval/rejection decisions, for example), and reasons if provided, expert ID and relevant component version identifiers, and key operational events (like aggregation rounds, errors, **HITL Secure Gateway & Task Router** interactions, client-side local active learning metric summaries and parameter changes).

- **Aggregated Model Store** - functions as a central repository for the global model versions. It has the task of storing the binaries of the aggregated global model and its previous versions. The **Central Server** is the only container with which it interacts to save new global model versions throughout the FL training rounds, after aggregation, and to retrieve specific model versions that are either for distribution to clients or for rollback to a previously stable, valid state.

Focusing now more on the interaction between the **Central Server** with the blockchain external system introduced in this proposed architecture, the principal objective of logging large amounts of information concerning the model, human expert decisions and other metadata to the **Blockchain Logging/Audit System**, is to guarantee that the system achieves accountability and transparency [62], allowing to create a trace of all the processes that took place in each FL round. With this in mind, logging is crucial for auditability (an aspect mentioned as essential in Research Question RQ2), whether it is for debugging purposes, tracing specific events, or explaining to the user the decisions made by the system.

4.3.2.3 Human Oversight & Interaction Hub

This environment comprises a collection of interconnected containers designed to integrate all aspects of human oversight, ethical validation and active learning within the HITL-FL system. In other words, it can be seen as the primary interface for **Human Experts** to interact with the system, as it orchestrates the various processes that leverage their knowledge and judgment. The main containers in this hub are showcased in Table 4.4.

Table 4.4: Summary of Containers within the Human Oversight & Interaction Hub

Container Name	Type	Primary Functionality
AI Dashboard	Web App/API	Primary interface for Human Experts. Displays metrics, visualizations and facilitates validation and task input.
Active Learning & Oversight Trigger Module	Application	Selects samples for review, processes expert feedback and triggers actions.
Continuous Ethical Validator	Application	Monitors model performance/fairness. Triggers alerts.
HITL Secure Gateway & Task Router	Application	Manages secure access for annotation, routes tasks to appropriate Human Experts [1], delivers tasks and receives secure feedback [46].
Governance Framework & Policies Store	Database/ Config Store	Stores ethical guidelines, risk rules, and access policies.
Authorization Service	Application/ Service	Policy Decision Point. Evaluates actions against policies to grant or deny permissions.

The containers within the Human Oversight & Interaction Hub collaborate extensively to enable an effective human-AI partnership:

- **AI Dashboard** - serves as the primary interface for interacting with **Human Experts** [39]. It is tasked with displaying the aggregated model state, global performance metrics and contextual summaries from the **Central Server.** Furthermore, it also interacts with the **Active Learning & Oversight Trigger Module,** receiving the IDs and metadata of the selected predictions for human review, and the **Continuous Ethical Validator,** which forwards ethical and performance alerts to be displayed in the dashboard [39, 42]. The primary objective of this dashboard is to provide human experts with a means to oversee and analyze the current state of the model, as well as to offer their own input and influence how AI behaves [42]. More specifically, this dashboard allows users to input their own decisions and trigger control signals.

  It is also important to note that these actions should be accompanied by their respective reason codes/text justifications (to ensure accountability and transparency [62]). To guide expert actions, ethical policy compliance rules from the **Governance Framework & Policies Store** are also utilized [25, 30], as well as tradeoff data from the external **System Tradeoff**

Analysis Results, which are displayed. The decisions made via the dashboard are communicated back to the central server for further processing and logged to the **Blockchain Logging/Audit System** [25, 55].

- **Active Learning & Oversight Trigger Module** - one of the central containers for optimizing and including expert oversight and responding to system events. It is responsible for receiving a list of predictions generated by the global aggregated model on a centrally-held representative dataset, along with their respective uncertainty and confidence scores [26], and corresponding contextual flags from the **Central Server.** Based on this, it selects the most informative samples/predictions that require review from the human experts [36], thereby initiating annotation requests (via the **HITL Secure Gateway & Task Router**), by publishing task requests (via MQTT [7]). Afterward, it processes and sends the verified user feedback and annotations received from the experts (like corrected labels or validated predictions from the representative dataset samples) back to the **Central Server** for further model fine-tuning, helping it learn from areas of prior uncertainty or error. Moreover, it receives ethical and performance triggers or alerts from the **Continuous Ethical Validator,** enabling human experts to take action and review the situation.

- **Continuous Ethical Validator** - dedicated to the ongoing monitoring of the FL global model's performance and fairness against predefined ethical policy compliance rules [29], stored in the **Governance Framework & Policies Store** [25, 30]. From the **Central Server,** this container receives the current model's performance metrics and state while also querying historical performance data and logs from the **Blockchain Logging/Audit System.** Suppose deviations from predefined ethical guidelines, significant performance degradation, potential biases, model drift or other configured alert conditions are detected. In that case, it triggers ethical/performance alerts that are sent to the **Active Learning & Oversight Trigger Module** for further action [36].

- **HITL Secure Gateway & Task Router** - works as a middleman between the **Active Learning & Oversight Trigger Module** and the humans that interact with the system. Its main task is to manage this secure and efficient interaction for specific functions [46], such as annotation. When it receives annotation task requests from the active learning container, it routes these tasks to the most suitable experts [1], using matching strategies (based on expert specialty and workload) performed by an internal router component, and manages secure access for them. Consequently, it receives and validates the submitted annotations and feedback from the experts (in terms of proof and anonymity) [46], before forwarding them back to the active learning container or logging information to the **Blockchain Logging/Audit System.** One of the main concerns of this container is to maintain and ensure the security and anonymity of these data transactions.

- **Governance Framework & Policies Store** - acts as a centralized repository for all governance-related information [25, 30]. It stores ethical guidelines and compliance benchmarks, risk

rules and access policies (GDPR rules for example), which are queried by various containers of this hub (such as the **AI Dashboard,** the **Continuous Ethical Validator** or the **Authorization Service**) to inform their operations and guarantee compliance.

- **Authorization Service** - functions as a PDP, as it receives requests to authorize specific actions (from the **AI Dashboard** for model approval or from the **HITL Secure Gateway & Task Router** for task assignment validation) [25]. Additionally, it evaluates these requests against the policies stored in the **Governance Framework & Policies Store** to grant or deny permissions accordingly [25, 30].

4.3.3 Component Diagrams

While the container diagram presented in Section 4.3.2 provides a more detailed view of the HITL-FL system, in terms of the deployable units within it, a deeper understanding of the internal structure and functionalities of specific key containers is necessary.

To achieve the desired level of detail, Component Diagrams (level 3 of the C4 Model) were utilized. These diagrams are best suited for decomposing the selected containers into their components, which in turn represent a logical grouping of related functionality, such as a module or a set of classes, within that container.

This section delves into the component diagrams for three critical containers, whose internal complexity and importance for the HITL-FL system warrant further attention. These containers are:

1. **Active Learning & Oversight Trigger Module** (from the Human Oversight & Interaction Hub) - pivotal for efficient human intervention and response.
2. **Central Server** (from the Central Server Environment) - orchestrates the main process of FL.
3. **FL Compute/Training Module** (from the Client Environment) - central to local data processing and model training.

This detailed view is crucial to clarify how the specific functionalities attributed to these containers are performed internally.

4.3.3.1 Component Diagram for the Active Learning & Oversight Trigger Module

The **Active Learning & Oversight Trigger Module** container is a key central part of the Human Oversight & Interaction Hub, since it handles the whole process of active learning from selecting data for human review [36], managing the feedback loop and responding to ethical/performance alerts [51]. The components and respective interactions are shown in Figure 4.3, and a summary of them is provided in Table 4.5.

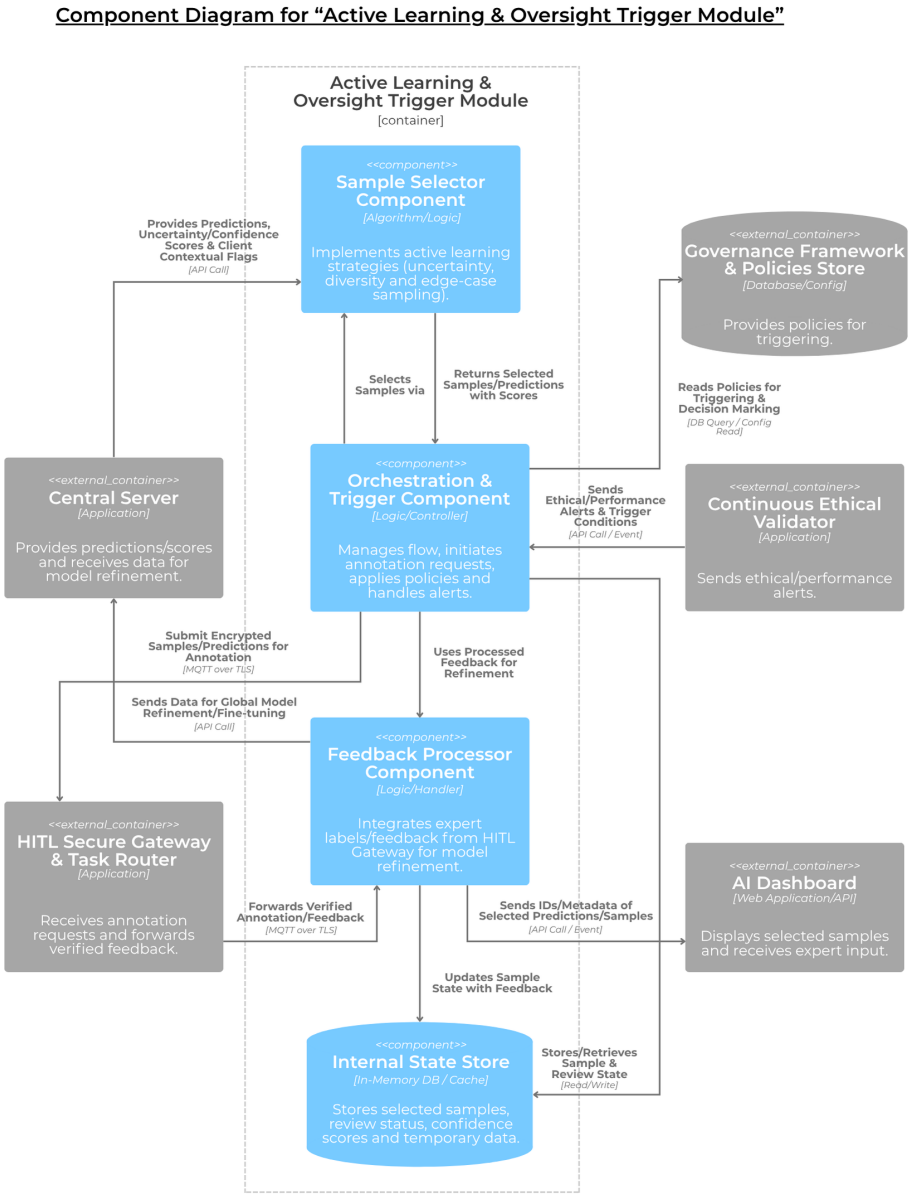

Figure 4.3: Component Diagram for the Active Learning & Oversight Trigger Module Container

The primary components within the Active Learning & Oversight Trigger Module and their detailed functionalities are as follows:

- **Sample Selector Component** - implements various active learning strategies (such as uncertainty sampling, diversity sampling and edge-case sampling) to identify the most informative data points or model predictions that human experts need to review. This component analyzes the predictions, uncertainty and confidence scores, as well as the client contextual flags received from the external **Central Server,** to apply the referred algorithms and select samples/predictions. It then passes these, along with their respective scores, to the **Orchestration & Trigger Component.**

Table 4.5: Summary of Components within the Active Learning & Oversight Trigger Module

Component Name	Type/Stereotype	Primary Functionality
Sample Selector Component	Algorithm/Logic	Implements active learning strategies (uncertainty, diversity and edge-case sampling).
Orchestration & Trigger Component	Logic/Controller	Manages flow, initiates annotation requests, applies policies and handles alerts.
Feedback Processor Component	Logic/Handler	Integrates expert labels/feedback from HITL Gateway for model refinement.
Internal State Store	In-Memory DB/Cache	Stores selected samples, review status, confidence scores and temporary data.

- **Orchestration & Trigger Component** - acts as the command center of this container by managing the end-to-end workflow of the active learning and human oversight responses. On the one hand, it receives selected samples from the **Sample Selector Component.** It initiates the annotation process by sending encrypted samples and predictions to the external **HITL Secure Gateway & Task Router** [46], which enables them to reach the **Human Experts** and the IDs/metadata of the selected predictions and samples to the **AI Dashboard.** On the other hand, it also applies policies for triggering annotation requests and decision-making that are read from the **Governance Framework & Policies Store** [25, 30]. Regarding incoming ethical/performance alerts from the external **Continuous Ethical Validator,** it handles them and, based on the policies, also initiates an annotation request for the human experts [36]. The subsequent feedback received from the **Feedback Processor Component** will be used to inform future actions and refinement cycles.

- **Feedback Processor Component** - processes and integrates the expert labels and feedback received from the external **HITL Secure Gateway & Task Router** after verification. It utilizes expert inputs, processes them by formatting or transforming them as needed for model refinement, and then sends the refined data to the external **Central Server** for global model fine-tuning. It also updates the status of the reviewed samples in the **Internal State Store** based on the received feedback.

- **Internal State Store** - provides temporary, often in-memory, storage for data that is crucial to the ongoing active learning process. This cache stores information such as the selected samples awaiting human review, the current review status, the confidence and uncertainty scores of the predictions, and any other transient data for managing the HITL workflow. This data store can be thought of as the shared storage that tracks the items being processed by this container, which in turn enables better reliability in the event of a system failure and facilitates auditability.

One key aspect that should be highlighted and clarified is the workflow within this container. It begins with the **Sample Selector Component** identifying candidates who deserve to be reviewed by **Human Experts** based on data provided by the **Central Server** and by applying the mentioned active learning algorithms. These candidates are then passed to the **Orchestration & Trigger Component**, which collaborates with external systems:

- Sends the IDs and metadata of these samples to the **AI Dashboard** for display and to aid with human analysis.
- Dispatches annotation requests via the **HITL Secure Gateway & Task Router**.
- Consults the **Governance Framework & Policies Store** for trigger/decision-making rules.
- Reacts to triggers or alerts sent by the **Continuous Ethical Validator**.

Once it receives its feedback, the **Feedback Processor Component** takes over, updates the **Internal State Store** and sends the refined data to the **Central Server** for further fine-tuning of the aggregated model.

4.3.3.2 Component Diagram for Central Server

The **Central Server** container, from the Central Server Environment, is the central orchestrator of the FL life cycle. This group of modules serves as the key point for aggregating client model updates, ensure model trustworthiness, select the most appropriate model version for client distribution, integrate and process the human oversight, and log all the steps and actions that occurred to external logging systems (in this case to the Blockchain Logging/Audit System). Similarly to Section 4.3.3.1, the internal components of this container are illustrated in Figure 4.4 and summarized in Table 4.6.

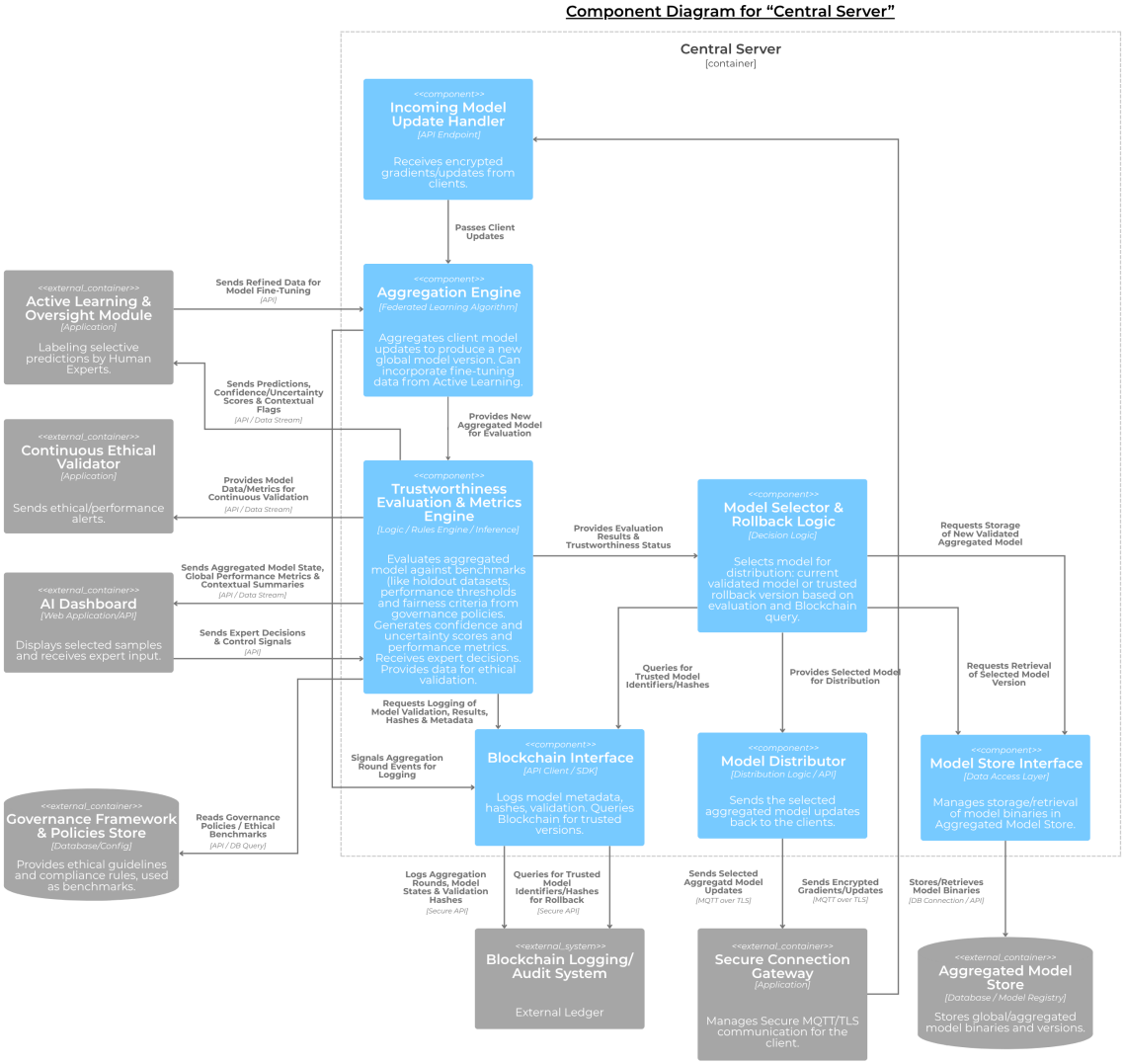

Figure 4.4: Component Diagram for the Central Server Container

The primary components within the **Central Server** container and their detailed functionalities are as follows:

- **Incoming Model Update Handler** - serves as the entry point for client model updates. It receives model gradients or parameter changes, which are then passed to the **Aggregation Engine** for further processing.
- **Aggregation Engine** - the core FL component that aggregates the model updates from the clients, received via the **Incoming Model Update Handler**, to produce a new candidate version of the global model. This new version can also incorporate fine-tuning received from the external **Active Learning & Oversight Module** to better enhance the quality and performance of the aggregated model, which is then provided to the **Trustworthiness Evaluation & Metrics Engine** for assessment [3].

Table 4.6: Summary of Components within the Central Server Container

Component Name	**Type / Stereotype**	**Primary Functionality**
Incoming Model Update Handler	API Endpoint	Receives encrypted gradients/updates from clients.
Aggregation Engine	Federated Learning Algorithm	Aggregates client model updates to produce a new global model version. Can incorporate fine-tuning data from Active Learning.
Trustworthiness Evaluation & Metrics Engine	Evaluation Engine	Evaluates aggregated model against benchmarks (holdout datasets, performance thresholds, fairness criteria from governance policies). Generates confidence and uncertainty scores, performance metrics; receives expert decisions and provides data for ethical validation.
Model Selector & Rollback Logic	Decision Logic	Chooses between the current validated model or a trusted rollback version (based on evaluation results and a Blockchain query).
Model Distributor	Distribution Logic / API	Sends the selected aggregated model updates back to clients.
Blockchain Interface	API Client / SDK	Logs model metadata, hashes and validation events; queries the Blockchain for trusted versions.
Model Store Interface	Data Access Layer	Manages storage and retrieval of model binaries in the Aggregated Model Store.

- **Trustworthiness Evaluation & Metrics Engine** - evaluates the newly aggregated global model predictions (using holdout datasets, for example) against predefined performance benchmarks, performance thresholds, and fairness criteria extracted from governance policies and ethical benchmarks, queried from the **Governance Framework & Policies Store**. This component also generates uncertainty and confidence scores [26], as well as performance metrics. In addition, it provides the data for the ethical validation and monitoring of the **Continuous Ethical Validator**, logs aggregation rounds, model states and validation hashes to the internal **Blockchain Interface**, and sends the results of its evaluation and trustworthiness status to the **Model Selector & Rollback Logic** component. Finally, this component sends the aggregated model state, global performance metrics, and contextual summaries to the **AI Dashboard** while also receiving expert decisions and control signals from it.

- **Model Selector & Rollback Logic** - based on the evaluation results and trustworthiness status from the **Trustworthiness Evaluation & Metrics Engine**, this component selects the model version that should be distributed to all the clients. This version could be either the newly aggregated model, created in the most recent FL round, or a previously validated and trusted version stored in the external **Aggregated Model Store** (accessed via the **Model Store Interface**), in case a rollback is necessary [19]. To retrieve the previous model versions from the data store, this component queries the **Blockchain Logging/Audit System** through the **Blockchain Interface** to obtain trusted model identifiers or hashes. If a new model version is selected, this component requests its storage in the **Aggregated Model**

Store. The chosen model version is then passed to the **Model Distributor**.

- **Model Distributor** - sends the selected aggregated model version back to the participating clients via the external **Secure Connection Gateway**.
- **Blockchain Interface** - handles interactions with the external **Blockchain Logging/Audit System**, logging model metadata, validation results, hashes of trusted models and aggregation round events. Furthermore, it also receives the results of the queries to the blockchain to retrieve the trusted model identifiers/hashes to support the **Model Selector & Rollback Logic component**.
- **Model Store Interface** - manages the reading and writing of model binaries to the external Aggregated Model Store. It handles the requests for storing new validated aggregated models from the **Model Selector & Rollback Logic**, as well as the queries for retrieving specific model versions, also from the mentioned component.

The overall flow within this container begins with the **Incoming Model Update Handler** receiving client updates and passing them to the **Aggregation Engine**. This component produces the new candidate for the aggregated global model using FL aggregation algorithms like FedDyn (or FedAvg) [51]. However, before being sent to the clients, it needs to undergo a thorough assessment by the **Trustworthiness Evaluation & Metrics Engine**. By using guidance policies [30], it outputs its evaluation results and the model's trustworthiness status to the **Model Selector & Rollback Logic component**, which makes the final decision on which model version to send to the clients. Suppose a new or current model is deemed suitable. In that case, the model selector component will request its storage in the external **Aggregated Model Store** via the **Model Store Interface** and pass the selected version to the **Model Distributor** for dissemination to the clients. Throughout this process, the **Blockchain Interface** logs critical information, including details of the finally selected/stored model, aggregation round events and other metadata, to the external ledger. Additionally, the **Trustworthiness Evaluation & Metrics Engine** sends various performance data and metric summaries to external containers, including the **AI Dashboard**, the **Continuous Ethical Validator** and the **Active Learning & Oversight Module**.

4.3.3.3 Component Diagram fro FL Compute/Training Module

The **FL Compute/Training Module** container, located within the Client Environment of the HITL-FL system, serves as the engine responsible for all local machine learning operations. More specifically, it handles the model training on private data, enables personalization based on local context or user feedback, processes updates from the global model and manages local data augmentation techniques, such as label propagation. As illustrated in Figure 4.5, its internal component architecture is designed to perform efficient local training and handle user interaction. Additionally, Table 4.7 summarizes the components that comprise this container.

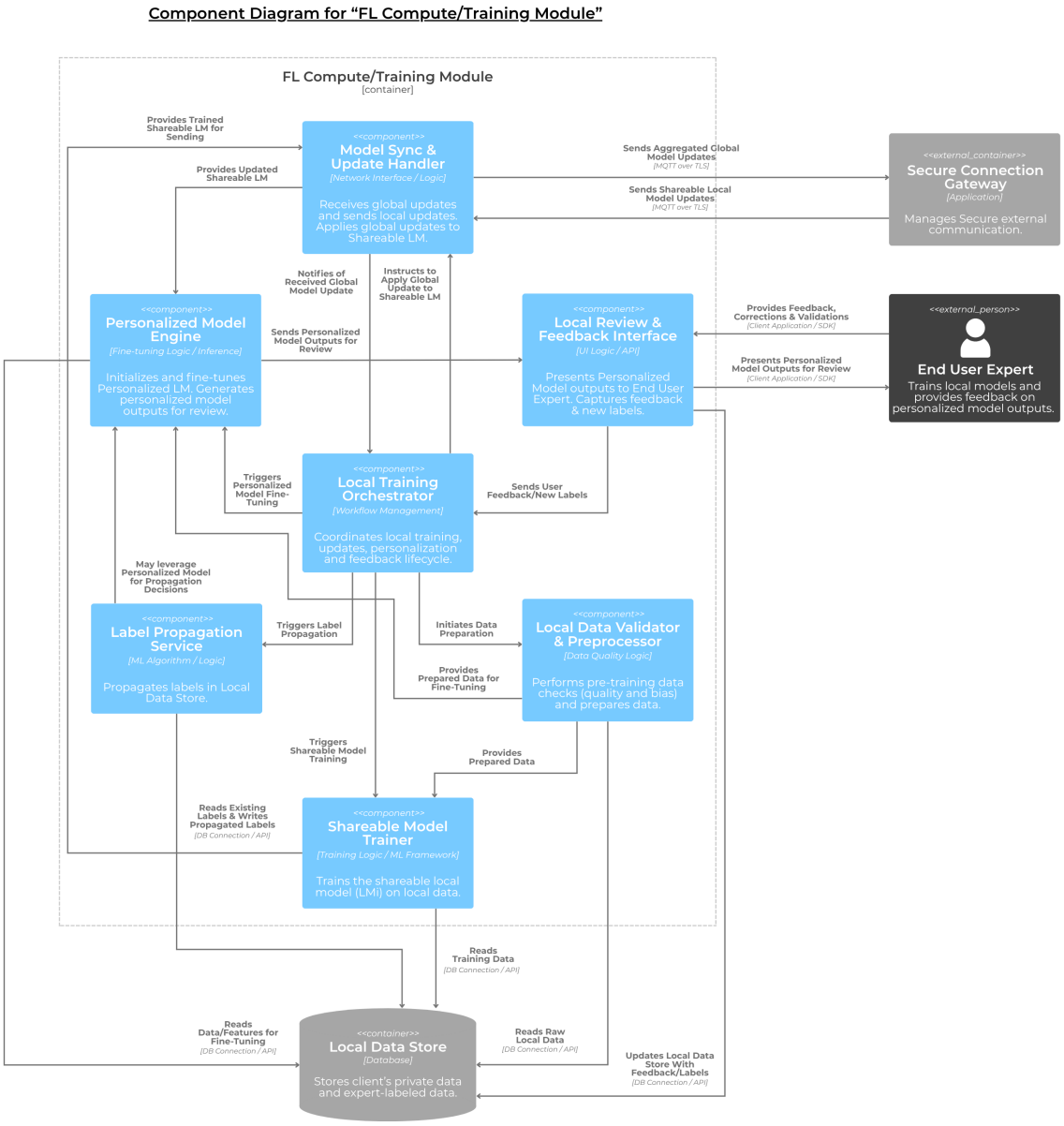

Figure 4.5: Component Diagram for the FL Compute/Training Module Container

An important aspect that needs to be clear is that, within each client's FL Compute/Training Module, a crucial architectural decision was made to maintain two different local models, balancing global knowledge contribution with local data fine-tuning [40]. These two models are the Shareable Local Model (SLM) and the Personalized Local Model (PLM):

- The **Shareable Local Model** can be seen as each client's contribution to the global learning process. It is initially trained on the client's private data, but in subsequent FL rounds, after the central server distributes an updated version of the aggregated global model, it is directly updated using these global insights. The client model updates are derived from training this SLM, which are then securely transmitted to the central server for aggregation.

Table 4.7: Summary of Components within the FL Compute/Training Module Container

Component Name	Type / Stereotype	Primary Functionality
Model Sync & Update Handler	Network Interface / Logic	Receives global updates, sends local updates and applies global updates to Shareable LM.
Personalized Model Engine	Fine-tuning Logic / Inference	Initializes and fine-tunes Personalized LM and generates personalized model outputs for review.
Shareable Model Trainer	Training Logic / ML Framework	Trains the shareable local model (LMi) on local data.
Local Training Orchestrator	Workflow Management	Coordinates local training, updates, personalization and feedback life cycle.
Label Propagation Service	ML Algorithm / Logic	Propagates labels in Local Data Store.
Local Data Validator & Preprocessor	Data Quality Logic	Performs pre-training data checks (quality and bias) and prepares data.
Local Review & Feedback Interface	Feedback Interface / UI Logic / API	Presents personalized model outputs to the End User Expert. Captures feedback and new labels.

- The **Personalized Local Model** aims to make the model more tailored to each client's unique data characteristics. After the SLM has been updated with the latest global knowledge, the PLM is initialized/reset based on this enhanced SLM. It is then further fine-tuned exclusively using the client's local data. The outputs and predictions generated by this refined PLM are subsequently presented to the End User Expert for review, allowing the model to adapt more precisely to the client's local context than the global model alone could. The feedback and verified annotations provided by the **End User Experts** are used to update the local data store. This enriched local data can then be used in subsequent training rounds of the Shareable Local Model, thereby influencing future FL aggregation rounds. As the SLM becomes better adapted to the (expert-refined) local context, its contributions lead to improvements in the global aggregated model's performance.

This approach enables the system to leverage the collective intelligence captured in the global model via the SLM while delivering highly relevant personalized experiences through the PLM, continuously refined by local user feedback.

The main components within the **FL Compute/Training Module** and their detailed functionalities are as follows:

- **Model Sync & Update Handler** - the primary component for model-related communication with the **Secure Connection Gateway**. Its primary responsibility consists of receiving the updated aggregated models from the central server and subsequently instructing the **Shareable Model Trainer** and the **Personalized Model Engine** to update both of their local model versions with the new knowledge (especially the Shareable LM). In contrast, it gathers all the client model updates from the **Shareable Model Trainer** (such as gradients, model weight parameters, etc.) and sends them externally to the central server. Moreover, it also notifies the **Local Training Orchestrator** that it received global model updates to coordinate further local actions.

- **Personalized Model Engine** - focuses on maintaining the PLM tailored to the specific client's data and/or **End User Expert**'s preferences [40]. It initializes and fine-tunes the PLM, often starting from the updated SLM provided by the **Model Sync & Update Handler**. In terms of data, it uses local data from the **Local Data Store** and generates predictions for user feedback to the **Local Review & Feedback Interface**. It sends the personalized model outputs to the **Local Review & Feedback Interface**, where they are analyzed and reviewed by the **End User Expert** [40].

- **Shareable Model Trainer** - trains the Shareable Local Model using the client's local data that is stored in the **Local Data Store**. This shareable model is the one whose updates contribute to the aggregated global federated model [40]. It also receives instructions to update its base model from the **Local Training Orchestration** whenever a new version of the global central model reaches the **Model Sync & Update Handler**.

- **Local Training Orchestrator** - is the core component of this container, as it coordinates the overall sequence of local operations related to training, updates, personalization and feedback life cycle. From the **Model Sync & Update Handler**, it receives new global model updates, while from the **Local Review & Feedback Interface**, it receives feedback from the end user human expert. Based on these triggers, it initiates actions such as triggering the **Shareable Model Trainer**, triggering personalized model fine-tuning in the **Personalized Model Engine**, initiating data preparation via the **Local Data Validator & Preprocessor** and activating the **Label Propagation Service**.

- **Label Propagation Service** - aims to expand the local dataset by automatically propagating existing labels (those provided by the **End User Expert** or high-confidence predictions from the personalized model) to unlabeled data samples that have similar features and characteristics stored in the **Local Data Store** [40]. This process, triggered by the **Local Training Orchestrator**, helps to increase the volume of labeled data available for local training with minimal additional manual effort from humans. As a consequence, it reads existing labeled data and writes newly propagated labels back to the **Local Data Store**.

- **Local Data Validator & Preprocessor** - performs pre-training data checks before the actual data is used for training or fine-tuning. This process includes assessing the data quality, identifying potential biases and ensuring that the data is in the correct format. It then makes the data suitable for consumption by the **Shareable Model Trainer** or the **Personalized Model Engine**. It is also important to note that the **Local Training Orchestrator** initiates this preparation process and the **Local Data Store** is used to retrieve raw data.

- **Local Review & Feedback Interface** - bridges the **Personalized Model Engine** with **End User Experts**, presenting the outputs of the personalized model for their review and validation [40]. It also captures the feedback, corrections or new labels provided by the expert, which is then passed to the **Local Training Orchestrator** and the **Personalized Model Engine** to influence subsequent model fine-tuning or data updates in the **Local Data Store**.

Concerning the main flow of the FL local training process, as stated before, the **Local Training Orchestrator** plays a central role in coordinating it, as it responds to external model updates (received via the **Model Sync & Update Handler** from the **Secure Connection Gateway**) and local user feedback (from the **Local Review & Feedback Interface** connected to the **End User Expert**). Due to these events, the central orchestrator commands the **Local Data Validator & Preprocessor** to prepare data from the **Local Data Storage**, thereby triggering training in both the **Shareable Model Trainer** and fine-tuning in the **Personalized Model Engine**. This may, in turn, invoke the **Label Propagation Service** to enrich the local dataset [40]. The **Personalized Model Engine** provides its outputs for review through the **Local Review & Feedback Interface**. At the same time, the model updates from the **Shareable Model Trainer** are sent to the central server via the **Model Sync & Update Handler**. All data-centric operations ultimately read from or write to the **Local Data Store** (which is external to this container but part of the **Client Environment**).

4.3.4 Core HITL-FL Design Problems

During the design process, several fundamental challenges drove the development of the HITL-FL reference architecture. Taking this into consideration, these challenges can be translated into core design problems that outline how the proposed architecture, as showcased in Sections 4.3.1 through 4.3.3, provides a solution through its specific components and mechanisms. This subsection enumerates and details these problems by presenting the challenges each represents and their respective solutions, as follows:

1. **How to effectively and safely incorporate human expertise oversight into a decentralized and privacy-sensitive machine learning paradigm (Federated Learning)?**
    - **Challenge:** Since FL operates on distributed data, bringing humans into the loop requires mechanisms that are secure, privacy-preserving, auditable and that provide meaningful interfaces for expert interaction without centralizing all data.
    - **Solution:**
        a. Dedicated components such as an AI Dashboard, Active Learning Module, HITL Secure Gateway and Expert Task Router.
        b. Secure and anonymized data/metadata exchange for expert review.
        c. Clear roles for Human Experts and End User Experts.
2. **How to ensure trustworthiness, ethical alignment and accountability in an evolving, collaboratively trained AI system?**
    - **Challenge:** Models trained via FL need to remain ethical, fair and accountable. For this, continuous monitoring, validation and a transparent record of decisions are required.
    - **Solution:**

   a. Human expert validation and override capabilities.
   b. Blockchain Logging/Audit System for immutable records of model states, validations and expert decisions.
   c. Continuous Ethical Validator and Governance Framework & Policies Store.
   d. AI Dashboard for monitoring and reporting.
   e. Mechanism to showcase System Tradeoff Analysis Results on the AI Dashboard.

3. **How to manage the complexity of collaboration between AI and humans at scale, including task management, security and varying expert specializations?**

   - **Challenge:** Coordinating many human experts, routing tasks appropriately according to each expert's specialty [1], while ensuring secure access and managing diverse feedback is complex.
   - **Solution:**
     a. Expert Task Router for routing review tasks to the right specialists.
     b. HITL Secure Gateway for authentication and secure feedback submission.

4. **How to make the HITL process efficient by focusing expert attention on the most critical or informative instances?**

   - **Challenge:** Human expert time dedicated to interacting with the system should be minimal and effective. Reviewing all data or all model outputs is often infeasible.
   - **Solution:**
     a. Active Learning & Oversight Trigger Module that selects informative samples or predictions for review.

5. **How to maintain data privacy and model confidentiality throughout the training and HITL processes?**

   - **Challenge:** Sending model updates, contextual data for review and expert annotations all create potential privacy risks if not handled carefully.
   - **Solution:**
     a. Encryption of model updates.
     b. Anonymization of contextual metadata during annotation requests.
     c. Consideration of differential privacy techniques.
     d. Secure communication channels (TLS over MQTT).
     e. HITL Secure Gateway ensuring anonymity where needed.

6. **How to ensure the integrity and verifiability of the global model and the decisions influencing its development?**

- **Challenge:** In a distributed system with multiple actors (clients, server, experts), ensuring that the model hasn't been tampered with and that decisions are accurately recorded is vital.
- **Solution:**
  a. Blockchain Logging/Audit System for cryptographic hashes of models, expert decisions and key events.
  b. Model Trustworthiness Check & Selection on the central server.

7. **How to balance global model improvement with the desire for personalized model performance for individual clients/users?**

   - **Challenge:** A global FL model might not be optimal for every individual client's specific data distribution or needs.
   - **Solution:**
     a. Distinction between Shareable Local Models (contributing to the global model) and Personalized Local Models (fine-tuned for individual clients).
     b. Local review and refinement loops for personalized models by End User Experts.

8. **How to ensure the HITL-FL system can effectively manage and process a growing number of participating clients and volumes of model updates, as well as a proportionally increasing demand for human expert interaction and oversight, without degrading performance and responsiveness?**

   - **Challenge:** As federated learning scales with more clients and complex models, the system needs to efficiently manage the proportionally increasing demand for human expert interaction and central coordination without creating bottlenecks.
   - **Solution:**
     a. Modular and independently scalable HITL services designed as distinct microservices for independent scaling.
     b. Asynchronous HITL Workflows by decoupling expert interaction timing from real-time model aggregation to buffer tasks and manage expert workload flexibly.
     c. Expert Task Router for efficient distribution to specialized experts and Active Learning Module for selecting only critical samples, managing task volume.
     d. Leveraging FL's inherent client scalability and designing central server components for deployment on scalable infrastructure.

Addressing these design problems has been the primary driver in creating and shaping the HITL-FL architecture, as to ensure that each component can be seen as a targeted solution for a specific challenge, rather then just a collection of features. With this in mind, by tackling several

challenges, namely secure expert incorporation, privacy-preserving interfaces, trustworthiness, auditable logging, active learning and scalable expert workflows, the proposed system is capable of delivering a robust framework for human-centric, trustworthy AI. More specifically, architectural elements such as the dedicated **Human Oversight & Interaction Hub**, the **Continuous Ethical Validator**, the **Governance Framework & Policies Store**, the **Blockchain Logging/Audit System**, the **Active Learning & Oversight Trigger Module** and other mechanisms work to create a system that is not only functional but also aligns with the principles of HCAI.

4.3.5 Core Architectural Patterns

From an architectural pattern perspective, it is also important to mention and highlight the patterns that realize the design principles showcased in Section 4.2 and aid in implementing solutions for the HITL problems presented in the previous section.

According to its architectural overview (Section 4.3), it is evident that the HITL-FL system employs a microservices style, where core functionalities are separated into different containers that communicate via APIs and event-driven messaging. Additionally, this architecture also showcases a layered style, with a clear separation between client-side components and central-server orchestration, for example. These layers are also connected to the client-server pattern, which underpins the FL rounds, while secure gateways act as proxies for all external communication. Finally, repository-like database/state stores and other cache modules were also included, all of which are responsible for managing both transient and persistent data.

These applied patterns focus on promoting and achieving various quality attributes essential to this system, such as modularity, scalability and maintainability.

To sum up, this chapter presents a detailed and problem-driven architecture of the HITL-FL framework. Due to its clear separation of concerns and responsibilities, dedicated HITL functionalities and emphasis on trustworthiness, security and auditability, this architecture proposed a standardized approach to embedding human expertise within decentralized learning paradigms. The specific strategies, mechanisms and workflows through which HITL is implemented within this architecture will be explored in detail in Chapter 5.

Chapter 5

HITL Integration Strategies & Mechanisms

5.1 Introduction

Having conducted a detailed architectural overview of the HITL-FL system's structure, this chapter focuses on the various mechanisms and strategies that enable the integration of HITL capabilities within this architecture. Considering this, while Chapter 4 described the building blocks that make up this reference architecture, this chapter addresses the specific strategies, mechanisms and workflows that contribute to an effective integration of human oversight.

The primary aim is to provide a deeper understanding of how human intelligence is leveraged to enhance a model's performance, ensure its ethical alignment and maintain system trustworthiness. To achieve this objective, this chapter will first cover the fundamental types of human intervention, then detail the key HITL mechanisms, and finally illustrate how all these elements work together by describing several HITL workflows and feedback loops from the proposed architecture that warrant highlighting. In addition, this emphasis on detailing the HITL strategies integrated into the system, also looks to answer research question RQ1.

This chapter is structured to provide a comprehensive understanding of the HITL integration within the HITL-FL architecture:

- **Section 5.2: Types of Human Intervention** revisits the core methods of involving human oversight in ML systems, mapping them to specific intervention opportunities with the proposed architecture.
- **Section 5.3: HITL Mechanisms** details the core enablers of HITL, including AI dashboards, annotation interfaces, task routing logic, alerting systems and feedback integration pipelines, while simultaneously explaining their realization through specific architectural components.
- **Section 5.4: HITL Workflows/Loops** sequentially illustrates the dynamic operation of these mechanisms by detailing three process flows/loops:

- **Active Learning Loop**
- **Model Oversight and Governance Loop**
- **Ethical Validation Loop**

5.2 Types of Human Intervention

Human intervention in ML systems, as stated in Section 2.4 of Chapter 2, can be categorized into several distinct methods that have unique roles, responsibilities and contribute to the effectiveness and trustworthiness of the overall system, allowing users to intervene and participate in the ML process, thereby not allowing AI to have the last say when it comes to decision making. These methods are listed as the following:

- **Human Oversight & Monitoring** - continuous monitoring and review of a system's performance and decisions, as well as model outputs, thereby ensuring alignment with established objectives and ethical standards [30].
- **Human Expertise for Rule Definition or Feature Engineering** - domain experts define explicit rules, select or craft input features, and also verify outputs against expectations to ensure that the model's behavior aligns with real-world constraints.
- **Human Feedback** - inputs that directly influence the model's training, refinement and adjustment processes.
- **Active Learning** - direct human inputs that are triggered by system alerts, anomalies or other expert evaluations, thereby leading to corrective measures and updates.
- **Human Decision Authority** - humans assume the final decision-making authority, especially in critical scenarios where the implications of possible errors are grand.

Building on these generic methods of intervention, the HITL-FL system especially leverages three primary types of human intervention, each with their own advantages, that are going to be described in the Sections 5.2.1 to 5.2.3.

5.2.1 End User Expert Feedback & Annotation

The **End User Expert** interacts within the **Client Environment** of the HITL-FL system, with the personalized local models that are trained locally on their devices. The focus of this intervention is on reviewing the outputs from these personalized models due to low confidence [32] or high-uncertainty predictions [40], potential bias indicators or factors that require the expert's own review and contextual knowledge. Based on their assessment, these experts correct errors and provide feedback and annotations, such as data labels. The particular advantage of this intervention is the influence it has on directly enhancing the local dataset quality and influencing the subsequent training cycles.

Table 5.1 details the containers/components involved in the enabling the integration of **End User Expert** feedback and annotations.

Table 5.1: Mapped Containers and Components for End User Expert Intervention

Container/Component	Description
Local Review & Feedback Interface	Allows End User Experts to review personalized outputs.
Local Data Store	Stores local data with user annotations and corrections.
Label Propagation Service	Uses annotations for further local data enrichment.
Personalized Model Engine	Uses annotations for model fine-tuning.

The **Local Review & Feedback Interface** makes the bridge between the expert and the system, as it is responsible for presenting the PLM outputs and capture the user's feedback and other responses. The **Local Data Store** stores the data with the users' feedback and the **Label Propagation Service** automatically propagates the newly added user labels to local data points that have similar characteristics [40]. The Personalized Model Engine also utilizes end-user insights to fine-tune its performance.

5.2.2 Human Expert Oversight via AI Dashboard

One of the ways **Human Experts** can interact with the central global system is through the continuous monitoring and assessment of the model's performance, fairness and ethical compliance verification [30]. To complete this task, these experts utilize the **AI Dashboard** [39, 42], which provides a comprehensive overview into the overall status and behavior the current aggregated global federated model. These experts have the ability to make crucial decisions, including:

- Approving or rejecting the model for the current FL round.
- Triggering retraining with modified parameters/data filters.
- Refining fairness constraints/policy thresholds.
- Requesting a model revision or a deeper dive analysis.
- Flagging the system for data rebalancing if biases are identified or other corrective actions are required.
- Selecting/Adjusting system configuration, according to system tradeoffs.
- Updating annotation guidelines.

Table 5.2 explicitly describes the containers and components involved in supporting the global monitoring and control actions from the **Human Experts**, through the **AI Dashboard** [25, 39, 55].

Table 5.2: Mapped Containers and Components for Human Expert Oversight via AI Dashboard

Container/Component	Description
AI Dashboard	Serves as the primary interface for Human Experts, as it displays important information regarding the aggregated model state and its performance metrics. It also facilitates expert input for decision-making and triggering control actions [39, 55].
Central Server	Provides the AI Dashboard with the necessary data and receives the experts' control signals/decisions from the AI Dashboard to influence the FL process.
Authorization Service	Validates the actions initiated by Human Experts via the AI Dashboard against defined policies.

Human Experts primarily interact with the **AI Dashboard**, which receives comprehensive data from the **Central Server**. Whenever an expert makes a decision, this action is first validated by the **Authorization Service** container against authorization policies and rules. The validated decision is then communicated to the **Central Server** to effect the change in the FL process [34].

5.2.3 Human Expert Participation in Active Learning

Active Learning enables human intervention by identifying the most uncertain or informative samples that require human review [42]. The **Human Experts** actively engage by labeling data points that are selected based on uncertainty/confidence scores, edge-case identification, diversity and other specific active learning algorithms. By performing this targeted annotation, the global aggregated model is enhanced, as it can integrate human expertise and, as a consequence, improve the quality and relevance of the training data [51].

The components/containers that are involved in the active learning done by **Human Experts** are detailed in Table 5.3.

Fundamentally, the active learning process, enabled by the mentioned architectural modules, allows the creation of a powerful relationship between the FL system and the human experts [51]. While the AI identifies areas of possible uncertainty or those that deserve analysis [26], the experts provide the necessary knowledge to resolve these ambiguities [36]. In this case, the **Orchestration & Trigger Component** ensures that this collaboration between containers and components is managed effectively. At the same time, the **HITL Secure Gateway & Task Router** routes the necessary information to the specific points of interest, all while ensuring the security of the exchange. The **Internal State Store** and **Feedback Processor Component** also have an essential task, as they store the data used in this process and enable the integration of human expert input, respectively.

Container/Component	Description
Sample Selector Component	Identifies informative samples requiring human annotation.
Orchestration & Trigger Component	Coordinates and manages the active learning annotation workflow and requests to Human Experts.
Feedback Processor Component	Integrates the expert-annotated data back into global model refinement processes.
HITL Secure Gateway & Task Router	Forwards predictions and samples, in an encrypted secure way, for the Human Experts to perform their annotation task. It also deals with sending the annotations and user feedback, in a secure way, back to the Active Learning & Oversight Module.
Internal State Store	Temporarily holds selected data points, their annotation status and expert feedback for tracking and auditing purposes.

Table 5.3: Mapping of Human Expert Participation in Active Learning

5.3 HITL Mechanisms

To employ the various types of human intervention outlined in Section 5.2, the proposed HITL-FL architecture introduces several key HITL mechanisms (some already mentioned in the previous section). These mechanisms are tools that facilitate the critical cycle of emerging model/system information, capturing valuable human inputs and channeling this feedback back to the central FL system [42].

In the following subsections from 5.3.1 to 5.3.5, five primary categories of HITL mechanisms will be explored, namely:

1. **AI dashboards**, for visualization and control.
2. **Annotation interfaces**, for detailed data labeling.
3. **Task routing logic**, for efficient workload distribution.
4. **Alerting workflows**, for proactive human engagement.
5. **Pipelines for feedback integration**, for systematic model improvement.

In addition to describing these mechanisms, the way they are instantiated through specific and relevant containers/components of the proposed architecture, presented in Section 4.3, will also be highlighted.

5.3.1 AI Dashboard

A centralized AI dashboard serves as the main mechanism for easing comprehensive human-global-model interaction within distributed machine learning systems, like FL ones[39].

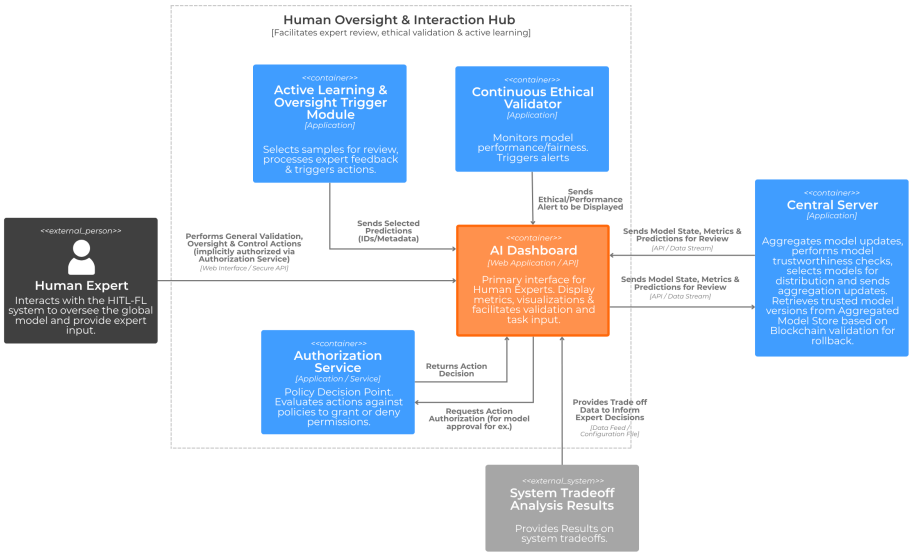

Figure 5.1: AI Dashboard in the Human Oversight & Interaction Hub, alongside its interactions with other containers

Its primary purpose is to provide users with a consolidated view of the system's overall status and performance. This task involves aggregating and displaying key telemetry, model quality metrics and data/concept drift trends. Furthermore, it also exposes various model artifacts, such as global model parameters, sample predictions and feature importance scores, for human review through a unified control panel.

In the proposed HITL-FL architecture, this pivotal role is performed by the **AI Dashboard** container, situated within the **Human Oversight & Interaction Hub** (as shown in Figure 5.1). This dashboard directly pulls from the **Central Server** the aggregated model state, global metrics, performance evaluations and contextual summaries. Additionally, it displays the selected predictions/samples and contextual flags highlighted by the **Active Learning & Oversight Module** container, as well as the ethical/performance alerts from the **Continuous Ethical Validator**. The dashboard also presents the data from the external **System Tradeoff Analysis Results** to aid experts in their review and decision-making processes [39].

More specifically, through this consolidated interface, human experts can analyze performance metrics (such as accuracy, precision and recall), explainability visualizations (from SHAP/LIME) [57], fairness audits (AIF360) [57], and synthesize contextual information via dashboard visualizations. Crucially, they evaluate the system's current operating point, assessing whether the achieved balance between objectives (privacy, utility and efficiency) is acceptable.

Interactive visualization tools (for dimensionality reduction and parallel coordinates) are also used as a way to explore client contributions, model representations and uncertainty clusters, allowing the dashboard to become an active exploration tool for experts to find patterns, analyze clustered low confidence samples or potentially biased instance groups and make informed oversight decisions [2].

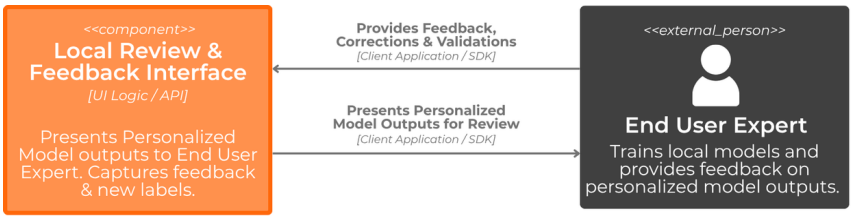

Figure 5.2: Local Review & Feedback Interface in the HITL-FL System

As a consequence, as outlined in Section 5.2, experts can take various oversight control actions, including approving or rejecting a candidate version of the global model, trigger retraining or request modifications to the model parameters, adjusting fairness constraints, policy thresholds [25, 34], or ways the models are being trained for future rounds [55]. However, these actions are first validated against predefined policies by the **Authorization Service** container. Once authorized, the expert decisions and their respective justifications are communicated back to the **Central Server**.

5.3.2 Annotation Interfaces

Effective annotation interfaces are crucial for capturing the input from human experts, since their primary role is to enable users to assign new labels to unlabeled data, correct existing misclassifications or add rich contextual feedback. To maximize the quality of this process, these interfaces need to minimize the friction for the annotator by presenting data clearly and intuitively. Furthermore, they also need to guarantee secure data transfers and provide in-context relevant information to help users make informed decisions or add high-quality, consistent labels.

The proposed HITL-FL architecture presents two complementary types of annotation interfaces tailored to different contexts and user roles, as depicted by their interactions with the Human Expert and the End User Expert, illustrated in Figures 5.2 and 5.3:

- **Local Review & Feedback Interface** (Client Environment) - this interface, located within the **FL Compute/Training Module** (Section 4.3.3.3) is designed for **End User Experts** to interact with their personalized local models. This component presents the outputs from these models directly on the client device, thereby allowing the users to provide immediate corrections, give new labels or offer feedback based on their knowledge of the local context

Figure 5.3: HITL Secure Gateway & Task Router in the HITL-FL System

and own expertise. As a consequence, this feedback is then directly written back to the client's **Local Data Store**.

- **Remote Annotation Interface** (via the HITL Secure Gateway) - although not physically showcased as a component or a container in the proposed HITL-FL architecture, this interface capability is primarily enabled by the **HITL Secure Gateway & Task Router** container within the **Human Oversight & Interaction Hub**, which focuses on delivering encrypted annotation tasks to the **Human Experts** [46]. The gateway presents the data to the assigned expert through a secure web form or a dedicated annotation tool [1]. The expert's annotations and feedback are validated (for completeness) by the gateway before being securely routed back, often as encrypted data [46], to the **Feedback Processor Component** (within the **Active Learning & Oversight Trigger Module**) for integration into the global model refinement process. This centralized interface ensures that expert annotations on globally relevant data are securely managed, consistently processed and effectively utilized.

Both interface types are designed to enable the incorporation of human oversight into the FL system, thereby improving data quality and, consequently, model accuracy and reliability.

5.3.3 Task Routing

In large-scale HITL systems, where numerous human experts with varying specializations and availability might be involved, a dedicated task routing mechanism is essential.

The primary role of this system is to intelligently and efficiently assign each annotation, validation or review task to the most appropriate human expert or group [1]. Factors such as the expert's domain knowledge, current workload to ensure timely completion, security or access control constraints and historical performance are considered during this routing process. Therefore, effective task routing is key to maximizing the quality of human input and the overall throughput of the HITL process.

Within the proposed HITL-FL architecture, this critical functionality is performed as an internal component of the **HITL Gateway & Task Router** container. It first authenticates human experts to ensure authorized access. Then, the internal component applies its matching strategies to assign incoming annotation requests from the **Active Learning & Oversight Trigger Module**. This matching logic considers expert specialty (for example, ensuring a medical image is reviewed by a radiologist), current availability or workload (to balance distribution) and potential quality scores from past contributions.

In addition to finding the appropriate expert or group of experts, the HITL gateway is also responsible for enforcing anonymity policies by stripping identifying information before task presentation or feedback submission [46]. It also utilizes security policies for access control and routing from the **Governance Framework & Policies Store** [30].

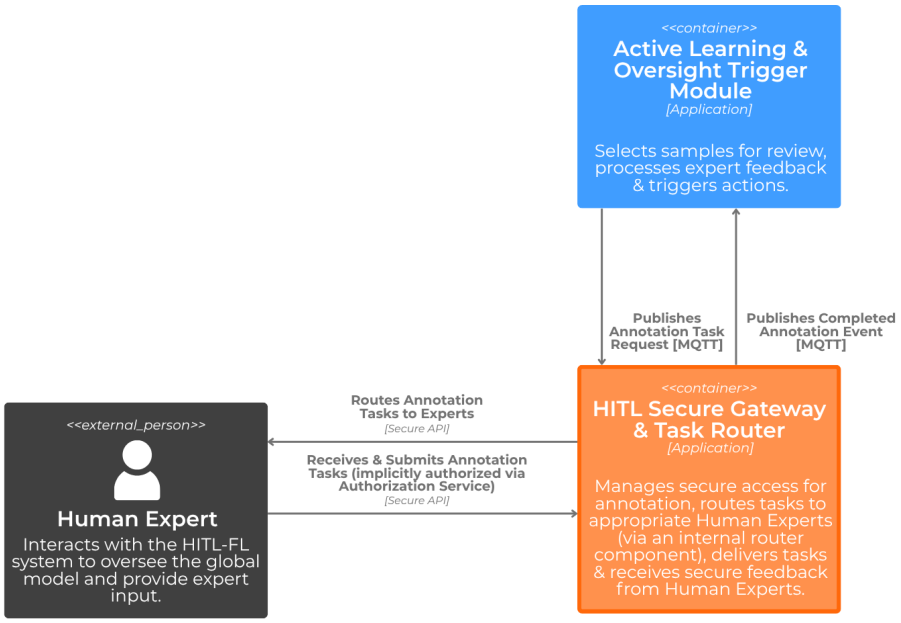

Figure 5.4: Interaction Flow for Task Routing and Annotation via the HITL Secure Gateway & Task Router

After the expert completes the task, this container receives the verified outputs (like annotations and validation decisions), ensures their integrity and securely forwards them to the appropriate upstream component within the **Active Learning & Oversight Trigger Module**. This process is illustrated in Figure 5.4.

5.3.4 Alerting & Review Triggers

Alerting workflows and review triggers are fundamental HITL mechanisms that are designed to engage human experts when an AI system encounters conditions that require critical expert review or exhibit anomalous behavior [52]. Their main purpose is to ensure opportune human intervention, thereby preventing the system from operating outside of predefined acceptable performance parameters, ethical guidelines or fairness constraints. These conditions include, for example, significant model performance degradation, the detection of potential biases, the exceeding of established thresholds, evidence of data/concept drift or other system anomalies that automated processes cannot resolve independently.

In the proposed HITL-FL architecture, this alerting capability is orchestrated by the cooperation between the **Continuous Ethical Validator** and the **Active Learning & Oversight Module** containers (more specifically its **Orchestration & Trigger Component**), both within the **Human Oversight & Interaction Hub**.

The **Continuous Ethical Validator** is responsible for constantly monitoring each new aggregated version of the global central model, received from the **Central Server**. It evaluates these models against defined ethical guidelines, compliance rules and performance benchmarks that are stored in the **Governance Framework & Policies Store** container [29]. This validation process

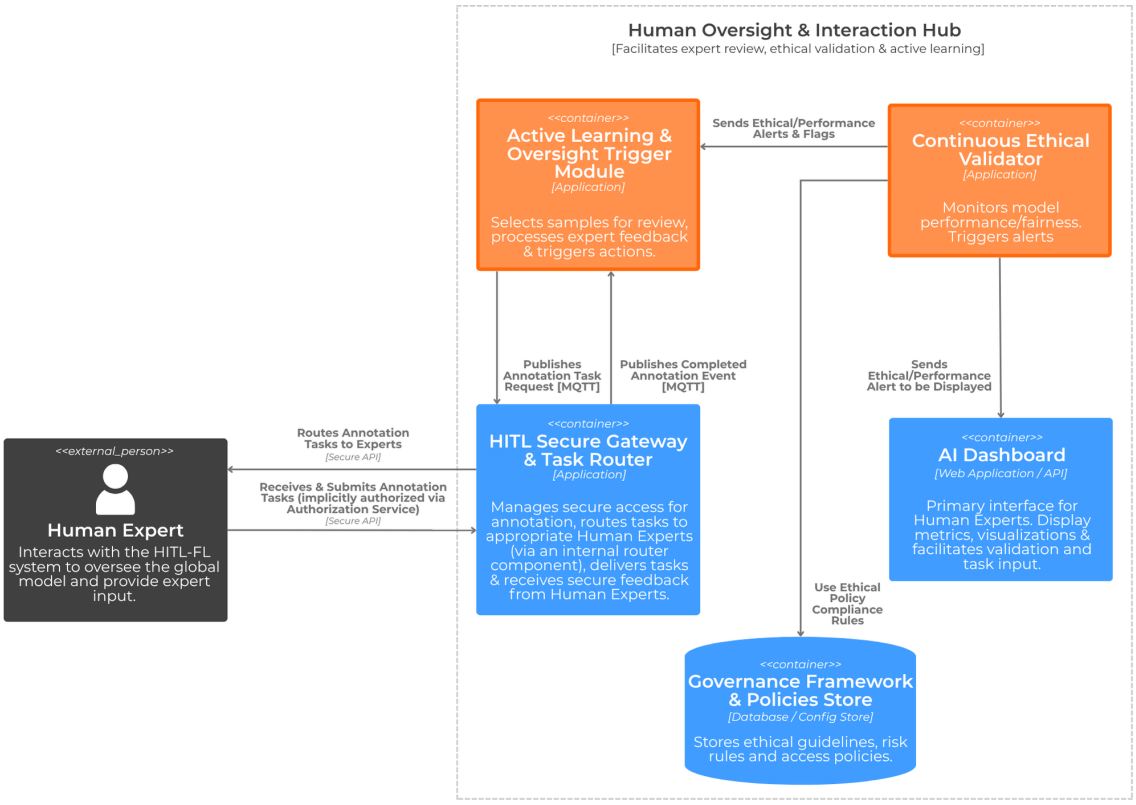

Figure 5.5: Alerts & Triggering Capability in the HITL-FL System

can involve, for example, the constant checking of fairness metrics, bias indicators or significant drops in accuracy results.

Whenever any violation or concerning trend is detected, the **Continuous Ethical Validator** sends ethical/performance alerts and flags via an API call or event bus [29]. This alert is consumed by the **Orchestration & Trigger Component** and, based on the degree of severity of the alert, it initiates a response, which involves:

- Creating high-priority review or annotation tasks.
- Routing these urgent tasks to appropriate **Human Experts**, via the **HITL Secure Gateway & Task Router container**.
- Sending a prominent alert banner on the **AI Dashboard** container to immediately draw the attention of overseeing experts [39].

This mechanism ensures that potential issues are not overlooked and that human expertise is leveraged to address them, thereby maintaining the system's overall trustworthiness [36]. These critical alert events and expert interventions are also logged to the **Blockchain Logging/Audit System** for comprehensive traceability.

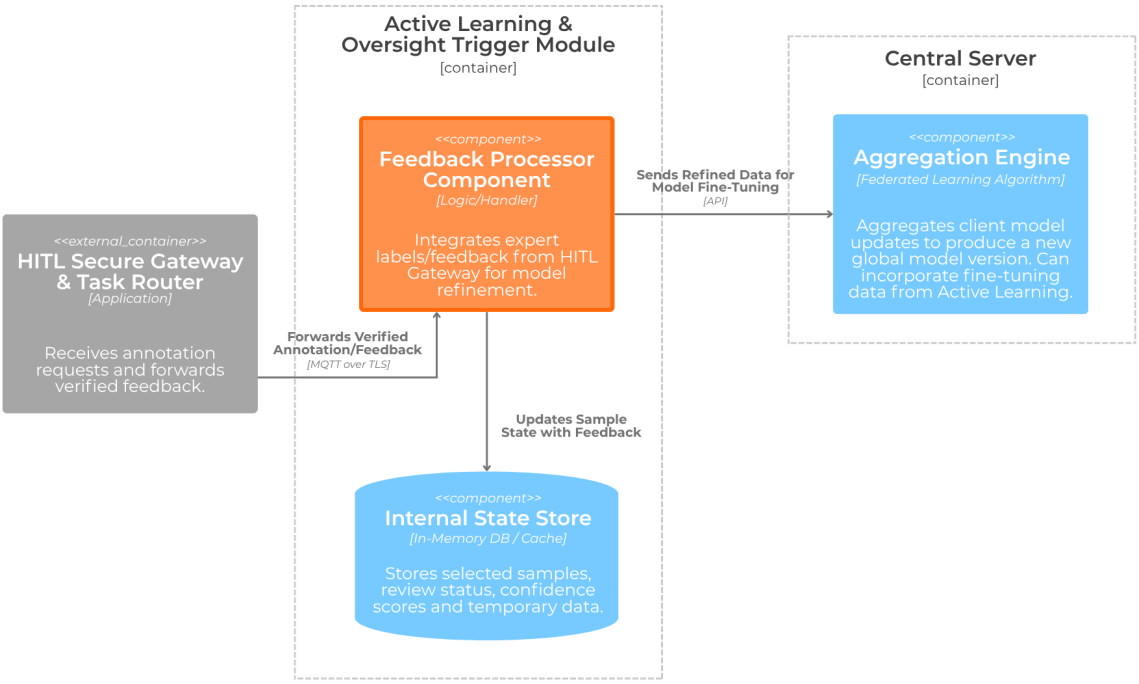

Figure 5.6: HITL-FL Feedback Integration Logic

5.3.5 Feedback Integration Logic

A robust feedback integration pipeline is essential for effectively incorporating validated human input back into the FL system. Its main functions are capturing validated annotations/corrections from human experts, processing the input (which may include format checks, data cleaning or the application of privacy-enhancing techniques [60], if required by policy) and then routing the feedback for immediate global model retraining or fine-tuning, for local model personalization at the client level or for its inclusion in governance records and auditable trails of expert input.

In the proposed HITL-FL system architecture, this critical role is performed by the **Feedback Processor Component** that is located within the **Active Learning & Oversight Trigger Module** (as showcased in Figure 5.6). After receiving verified human annotations and feedback forwarded by the **HITL Secure Gateway & Task Router**, this component updates the status of the corresponding sample or task in the **Internal State Store** component. Subsequently, it prepares and pushes the refined expert data to the **Central Server** container, where this data is then used by the **Aggregation Engine** to fine-tune the current aggregated global model or is incorporated into the representative dataset for the next federated learning round, thereby improving the global model with expert knowledge. Furthermore, primarily for auditability purposes, the **Feedback Processor Component** ensures that a cryptographic hash representing the accepted annotation is logged to the external **Blockchain Logging/Audit System**, via the **Central Server**.

This closed-loop process is crucial to the adaptive learning capability of the HITL-FL system, as it enables it to continuously improve based on targeted human expertise and consequently improve its reliability and accuracy.

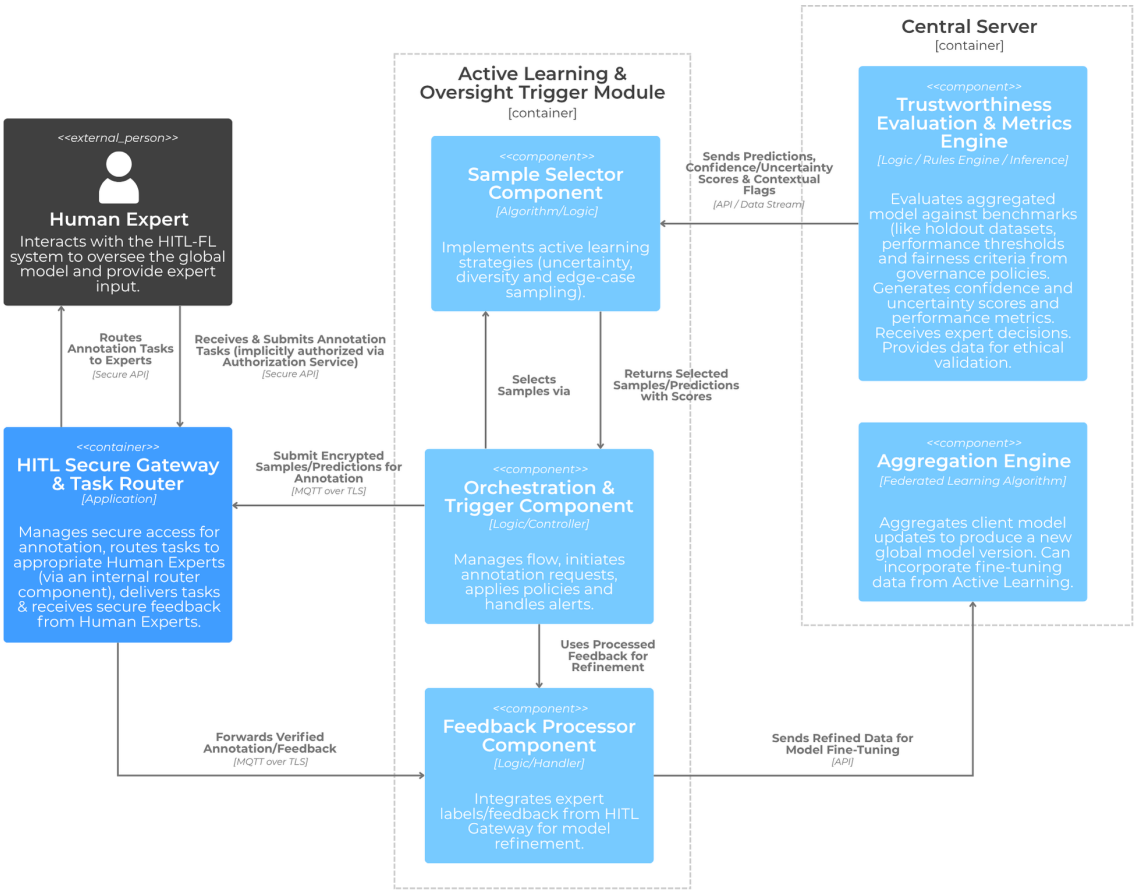

Figure 5.7: HITL-FL Active Learning Loop

5.4 HITL Workflows/Loops

Having presented the core HITL mechanisms, this section now focuses on showcasing how the various containers and components interact through several key dynamic workflows. With this in mind, three critical HITL workflows/loops will be showcased.

5.4.1 Active Learning Loop

In the proposed HITL-FL system, the Active Learning Loop, illustrated in Figure 5.7, is a core mechanism of HITL that iteratively improves the performance and quality of the global aggregated model's outputs. For this, the process identifies specific data samples from a centrally held representative proxy dataset, for which the global model could produce uncertain predictions or where **Human Expert** input on these sample-prediction pairs would be most valuable. Consequently, the expert-curated knowledge, presented in the form of validated labels or contextual feedback-/insights for specific instances, is then integrated back into the model refinement process. With this in mind, it is possible to state that this loop involves a precise orchestration of several key architectural components and expert actions, as detailed below:

Step 1 – Global Model Inference and Uncertainty Surfacing
The first phase of this loop involves assessing the current global model's performance and identifying areas of potential weakness.

- **Responsible Entity** - the **Central Server container**, specifically its **Trustworthiness Evaluation & Metrics Engine** component.
- **Action** - following each federated aggregation round, the **Central Server** takes the newly formed global aggregated model and runs it against a centrally held, representative proxy dataset. However, it is essential to note that this proxy dataset is distinct from any private client data used during the federated training process.
- **Generated Output** - for each sample within this proxy dataset, the **Central Server** generates not only the model's predictions but also uncertainty or confidence scores (derived from softmax probabilities in classification tasks or variance if using model ensembles) and any relevant, anonymized contextual flags that might have been aggregated from clients during the FL process [46].
- **Purpose** - the main objective of this step is to produce predictions and their associated confidence/uncertainty scores, which will subsequently allow the system to identify areas where the model is unsure and highlight where it does not understand well, or where human expert help would be most useful for improvement.

Step 2 – Intelligent Sample Selection
Once the global model's predictions and associated uncertainties are made available, the next phase focuses on applying active learning algorithms to select which of these instances should be prioritized for human expert review. This selection process is key to optimizing the use of expert time.

- **Responsible Entity** - the **Active Learning & Oversight Trigger Module container**, specifically its **Sample Selector Component**.
- **Action** - the **Central Server** transmits the list of predictions and data points, along with their corresponding uncertainty/confidence scores and any relevant contextual flags to the **Active Learning & Oversight Trigger Module**. Within this container, the **Sample Selector Component** applies predefined active learning strategies. These strategies include algorithms, such as uncertainty sampling, which prioritizes samples with the lowest model confidence, diversity sampling, which selects samples that differ from those already well-understood by the model, or edge-case detection, which identifies unusual/boundary instances.
- **Generated Output** - a subset of the proxy dataset samples that were identified by the active learning strategies as being the most informative for human review, meaning they are most likely to contribute significantly to improving the global model's understanding and performance if correctly labeled or validated by an expert.

- **Purpose** - by prioritizing expert review efforts on the predictions/samples that offer the highest potential learning gain for the model, this HITL-FL system aims to maximize the impact of limited human resources and accelerate model improvement.

Step 3 – Secure Task Creation and Dispatch
The following step involves creating annotation requests and securely dispatching them to the human experts. This phase ensures that sensitive information is handled appropriately and that tasks are initiated in a traceable manner.

- **Responsible Entity** - the **Orchestration & Trigger Component**, located in the **Active Learning & Oversight Trigger Module** container)
- **Action** - the **Orchestration & Trigger Component** receives the informative predictions/samples and respective metadata selected by the **Sample Selector Component**. Before dispatching them to the **Human Experts**, de-identification, feature masking and other anonymization techniques are applied to ensure that the expert review process respects data confidentiality and privacy regulations [1].
- **Generated Output** - once the most informative data samples from the proxy dataset, along with the global model's corresponding predictions and uncertainty scores for those samples [26], have undergone any necessary privacy-preserving transformations, the component prepares these for expert review, publishing annotation task requests as MQTT events [7]. These event messages encapsulate:
  - The data sample itself.
  - The model's original prediction for that sample.
  - The model's confidence/uncertainty score associated with that prediction.
  - Any additional contextual information or specific instructions relevant to the annotation task.

  These task requests are destined for the **HITL Secure Gateway & Task Router** container, which will then route them to the appropriate **Human Expert**'s annotation interface.
- **Purpose** - the goal of this step is to initiate the human review process in a secure, auditable and event-driven manner, ensuring that the data is appropriately protected before being exposed for expert review and that tasks are dispatched to the correct downstream components for routing to human experts.

Step 4 – Task Routing and Presentation to Expert
After the secure annotation tasks are created and dispatched, the system must make sure that they are routed to the most suitable human expert and subsequently presented in a clear manner. This step bridges the automated system with the human cognitive process.

- **Responsible Entity** - the **HITL Secure Gateway & Task Router container**.

- **Action** - the **HITL Secure Gateway & Task Router** subscribes to the annotation task request MQTT events, published by the **Active Learning & Oversight Trigger Module**. Upon receiving this request, its internal expert task router component performs a sequence of critical actions:
  (a) Authenticates the source of the request (typically the **Active Learning & Oversight Trigger Module**) to ensure legitimacy.
  (b) Applies matching algorithms to assign the annotation task to the most appropriate available **Human Expert**. This match logic considers various factors, such as the expert's documented specialization, current workload and availability (to ensure timely completion) and any access control policies or security constraints defined in the **Governance Framework & Policies Store** [25, 30] (validated via the **Authorization Service**).
  (c) Securely presents the complete annotation task to the assigned **Human Expert** through a dedicated and secure annotation interface (like a web form or specialized tool facilitated by this gateway).
- **Purpose** - ensure that each review task is directed to a qualified Human Expert, who can provide the most valuable input, and manage this process securely and efficiently, controlling access to task data and balancing the workload across the available experts.

Step 5 – Expert Review and Annotation
This step represents the principal human contribution within the Active Learning Loop. Once the annotation task is routed and presented, the assigned human engages with the system to provide their own specialized knowledge or judgment.

- **Responsible Entity** - the **Human Expert**.
- **Action** - the assigned expert interacts with the dedicated annotation interface, where they can analyze the complete task information, including the data sample requiring review (an image, text snippet or data record), the model's original prediction for that sample, the associated confidence/uncertainty score and any pertinent contextual details or specific instructions. Based on this information, the human expert performs one or more of the following critical actions [55]:
  – **Correcting Misclassifications/Errors** - if the model's prediction is inaccurate, the expert provides the correct label, classification or output value.
  – **Validating Predictions** - confirm the model's prediction, especially if it was made with low confidence but is, in fact, correct [32].
  – **Provide Labels or Annotations** - adding more detailed attributes, bounding boxes, segmentation or nuanced contextual information.
  – **Identifying Ambiguity or Novel Edge Cases** - flag samples that are inherently ambiguous, represent novel scenarios not well-covered by the model's current training or that are important edge cases.

  - **Flagging Data Quality Issues** - identify and report any corruption, artifacts or inconsistencies found in the input data itself.
  - **Providing Contextual Justifications** - experts can give reasoning behind their annotations or decisions.
- **Generated Output** - the primary output from this step is the expert's input, which includes their definitive labels, corrections, validations, richer annotations and provided justifications. This output is then securely submitted back to the system through the **HITL Secure Gateway & Task Router** container.
- **Purpose** - the fundamental aim of this step is to directly leverage human intelligence and domain-specific knowledge to improve the model and include human oversight.

Step 6 – Feedback Processing and Preparation
The system must now process the feedback given by the human experts to ensure its integrity and auditability, and prepare it for its integration into the model cycle. This step can be characterized as the bridge between raw input data and actionable insight.

- **Responsible Entity** - the **Feedback Processor Component**, in the **Active Learning & Oversight Module**.
- **Action** - the process begins when the **HITL Secure Gateway & Task Router** container receives the expert's submission and subsequently publishes a completed annotation event, transmitted via MQTT, that contains the expert's curated feedback (labels, corrections, justifications, etc.) [7]. The **Feedback Processor Component** subscribes to and consumes this event. Upon receipt, it performs several key processing actions, including further sanitization and consistency checks. It facilitates the logging of a cryptographic hash or a comprehensive summary of the accepted annotation and expert decision to the external **Blockchain Logging/Audit System**, via the **Central Server**.
- **Generated Output** - the primary output of this component is refined data, which is verified, processed, structured and ready to be integrated for the improvement of the aggregated global model.
- **Purpose** - to transform raw human expert input into a clean, validated and auditable dataset for effective integration into the model improvement, ensuring that the expert's knowledge is reliably incorporated.

Step 7 – Global Model Refinement
The final step in the Active Learning Loop closes the cycle by incorporating the processed human expertise back into the global federated model. It is the end product of the HITL effort, transforming expert insights into measurable model improvements.

- **Responsible Entity** - The **Central Server** container, primarily through its **Aggregation Engine**.

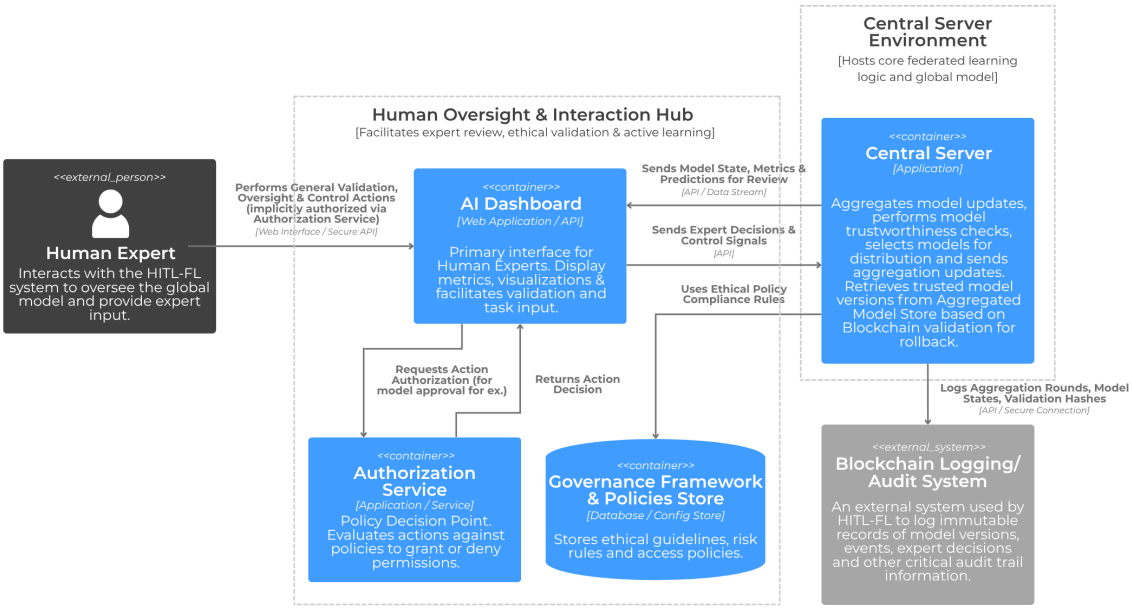

Figure 5.8: Model Oversight & Governance Loop in HITL-FL System

- **Action** - the **Feedback Processor Component** transmits the refined and processed data back to the **Central Server**. The **Aggregation Engine** then utilizes this high-quality, expert-validated dataset to perform a fine-tuning of the current global aggregated model. This process allows the model to learn directly from the instances where it was previously uncertain or incorrect, effectively internalizing the provided human knowledge.
- **Generated Output** - the integration of this expert feedback yields two significant benefits:
  - **Direct Model Improvement** - the most immediate outcome is the enhancement of the accuracy, robustness and generalization capabilities of the global aggregated model.
  - **Enhanced Future Active Learning** - patterns found in expert corrections/annotations can be used to refine the strategies employed by the **Sample Selector Component**, in subsequent active learning cycles.
- **Purpose** - to finalize the active learning cycle by guaranteeing that human expertise is directly translated into a more innovative, more reliable and accurate global central aggregated model. Furthermore, it enables the system to adapt and improve based on a real-world understanding that goes beyond what can be learned from raw data alone by incorporating oversight and expertise from human users.

This detailed breakdown of the Active Learning Loop illustrates the collaborative interplay between automated processes and targeted human expertise, orchestrated by the HITL-FL architecture to drive continuous model improvement.

5.4.2 Model Oversight & Governance Loop

This feedback loop, showcased in Figure 5.8, empowers the **Human Expert** to conduct a comprehensive review of the global aggregated model, ensuring that the system's quality, trustworthiness, accountability, reliability and ethical alignment are influenced by the human experts. This loop involves the following steps:

Step 1 – Information Aggregation & Presentation
The basis of effective oversight is access to comprehensive and understandable information concerning the global model's state.

- **Responsible Entities** - primarily the **Central Server container** and the **AI Dashboard container**.
- **Action** - on the one hand, the **Central Server**, following each aggregation round and its internal trustworthiness evaluation, compiles key information about the latest global aggregated model. This includes overall performance metrics (such as accuracy or precision), fairness metrics across predefined subgroups, drift detection indicators, contextual summaries of contributing data (if available and anonymized), and comparisons with previous model versions. On the other hand, the **AI Dashboard** pulls and integrates this comprehensive data from the server. Additionally, it also further integrates data from other containers and external systems, such as alerts or status updates from the **Continuous Ethical Validator** container, summary statistics about ongoing or recent annotation tasks from the **Active Learning & Oversight Trigger Module** container (providing context on model weaknesses being addressed) and results from periodic **System Tradeoff Analysis Results** system [20].
- **Generated Output** - the **AI Dashboard** presents this multifaceted information to authorized **Human Experts** using various visualizations, trend charts and summaries. These are designed with HCAI and eXplainable AI [60] principles in mind to enable users to make informed decisions.
- **Purpose** - provide human experts with a complete, understandable and actionable overview of the global model's current state, performance, ethical standing and operational context, enabling them to provide insightful oversight.

Step 2 – Expert Review & Assessment
Human Experts perform a strategic evaluation of the global model.

- **Responsible Entity** - the **Human Expert**.
- **Action** - using the **AI Dashboard**, **Human Experts** can review the aggregated information. Their assessment and oversight focuses on the overall characteristics and behavior of the global model rather than individual samples. This evaluation includes:
  - **Performance sufficiency.**
  - **Fairness and bias.**

  - **Ethical alignment.**
  - **Robustness and stability.**
  - **Risk Assessment.**
  - **Alignment with strategic goals.**
- **Purpose** - apply human judgment, domain expertise and insights to assess the global model's performance and its adherence to established quality, ethical and operational standards.

Step 3 – Strategic Decision-Making & Control Actions
Based on their assessment, experts initiate actions to govern the model's lifecycle and evolution.

- **Responsible Entities** - the **Human Expert** (acting via the AI Dashboard), with validation by the **Authorization Service container**.
- **Action** - Human experts start strategic control actions through the **AI Dashboard**. However, before commitment, the **Authorization Service** is consulted to ensure the expert has the necessary permissions.
- **Generated Output** - these decisions are translated into commands sent from the **AI Dashboard** to the **Central Server** (for model lifecycle actions) or interfaces with the **Governance Framework & Policies Store** (for policy updates).
- **Purpose** - enable experts to interact and influence the FL process, enforce quality and ethical standards, mitigate risks and ensure that the HITL-FL system evolves in alignment with overarching goals.

Step 4 – System Response and Auditing
The system then acts upon authorized expert decisions and records/logs them.

- **Responsible Entities** - the **Central Server** container, **Governance Framework & Policies Store** and the external **Blockchain Logging/Audit System**.
- **Action** - the **Central Server**'s relevant components (the **Trustworthiness Evaluation & Metrics Engine**, **Model Selector & Rollback Logic** and **Aggregation Engine**) implement the authorized decisions received from the **AI Dashboard**. Regarding the **Governance Framework & Policies Store**, it persists any updated policies, making them active for future operations. Finally, all significant expert decisions, control actions, policy updates and resulting system responses are immutably logged to the **Blockchain Logging/Audit System**. This log includes details such as who made the decision, when it was made and any justifications provided.
- **Intended Purpose** - ensure that the system implements expert governance actions and that a complete and verifiable audit trail of these strategic interventions is maintained, contributing to the transparency and auditability of the system.

This loop highlights the role of **Human Experts** in guiding the overall direction and quality of the FL models developed through the HITL-FL system, complementing all refinements of the Active Learning Loop, presented in Section 5.4.1.

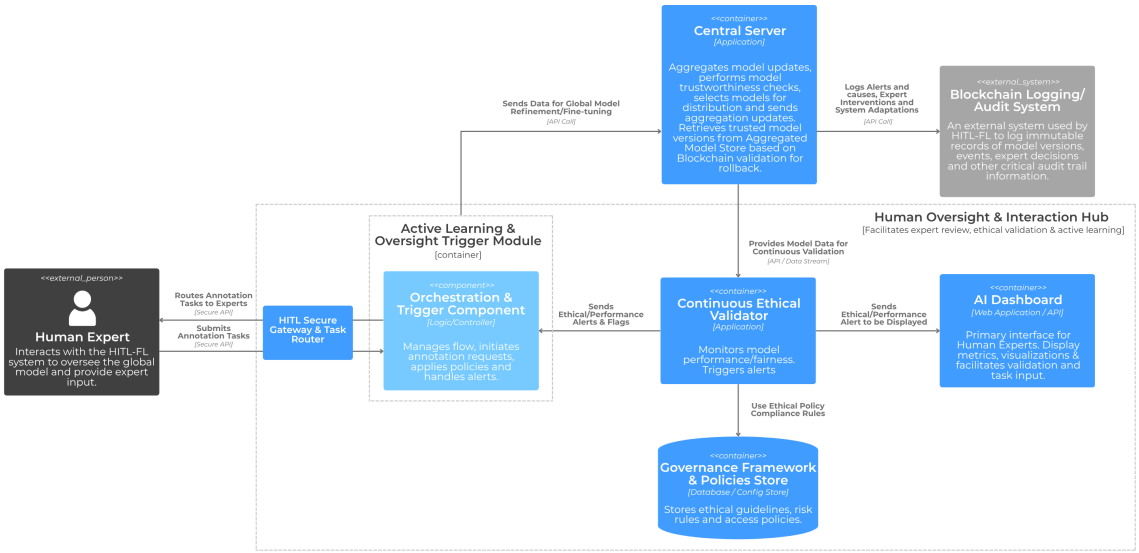

Figure 5.9: Ethical Validation Loop in HITL-FL System

5.4.3 Ethical Validation Loop

This workflow, presented in Figure 5.9, serves as a continuous safeguard for monitoring the global aggregated model to ensure that it adheres to predefined ethical guidelines, fairness criteria and performance benchmarks. Its goal is to detect and flag potential issues early, enabling timely intervention by **Human Experts** through the **Active Learning & Oversight Trigger Module** to maintain the integrity and trustworthiness of the AI. The loop is described in the following phases:

Step 1 – Model Acquisition for Validation
The loop begins with the system obtaining the latest model for inspection.

- **Responsible Entity** - the **Continuous Ethical Validator**.
- **Action** - following each successful federated aggregation round on the **Central Server**, and after a new global aggregated model is produced, the **Continuous Ethical Validator** is triggered/scheduled to retrieve this latest model version for assessment.
- **Purpose** - ensure that every newly generated global model iteration undergoes systematic ethical and performance inspection.

Step 2 – Policy-Driven Assessment
The retrieved model is evaluated against established standards and policies.

- **Responsible Entities** - the **Continuous Ethical Validator container** that utilizes policies from the **Governance Framework & Policies Store container**.
- **Action** - the **Continuous Ethical Validator** evaluates the global model against a set of rules, metrics and thresholds defined within the **Governance Framework & Policies Store**. This assessment includes, for example:

  - Calculating fairness metrics like disparate impact and equal opportunity difference, across defined sensitive groups.
  - Running specific tests to detect potential learned biases.
  - Verifying that performance indicators such as accuracy, precision and recall on specific data slices meet minimum acceptable levels.
  - Monitoring for significant concept drift or deviations in data distributions/model behavior compared to baselines, which might indicate a degrading ethical stance or performance.
  - Checking adherence to ethical directives or behavioral constraints.
- **Input for Action** - policies and benchmark criteria are read directly from the **Governance Framework & Policies Store**.
- **Purpose** - systematically audit the model against established ethical principles and quality standards.

Step 3 – Alert Generation & Dissemination
If issues are detected, the system generates and communicates an alert.

- **Responsible Entity** - the **Continuous Ethical Validator container**.
- **Action** - if the **Continuous Ethical Validator**'s assessment showcases that the model violates a predefined policy, a fairness/performance threshold or shows a significant problematic drift, it generates an ethical/performance alert. This alert contains details about the specific policy violated, relevant metric values or the assessed severity.
- **Generated Output** - the alert is then securely forwarded to downstream components within the **Human Oversight & Interaction Hub**.
- **Purpose** - promptly notify the appropriate **Human Expert**, and other system components, when a potential ethical or performance issue is detected, initiating further action.

Step 4 – Notification & Information Surfacing to Experts
Human experts are made aware of the detected issues.

- **Responsible Entity** - the **AI Dashboard** container.
- **Action** - one of the recipients of the ethical/performance alert is the **AI Dashboard**. The alert information is showcased to **Human Experts**, often highlighted with an indication of urgency or severity. The dashboard may also provide access to additional contextual information or diagnostic tools to help experts investigate the flagged issue.
- **Purpose** - ensure **Human Experts** are immediately made aware of critical ethical or performance issues, enabling them to investigate thoroughly.

Step 5 – Triggering Automated or HITL Responses
The system initiates actions in response to the alert.

- **Responsible Entity** - the **Active Learning & Oversight Trigger Module container** (specifically, its **Orchestration & Trigger Component**).
- **Action** - the alert is also sent to the **Orchestration & Trigger Component**. Based on the severity of the alert, and guided by predefined rules (informed by the **Governance Framework & Policies Store**), this component can initiate an active learning alert to be reviewed by the **Human Experts**.
- **Intended Purpose** - to enable rapid and automated responses to detected ethical/performance issues.

Step 6 – Feedback, Adaptation & Auditing
The loop closes with expert intervention, system adaptation and comprehensive logging.

- **Responsible Entities** - **Human Experts**, the **Central Server container** and the external **Blockchain Logging/Audit System**.
- **Actions** - the **Human Experts**, informed by the alerts on the **AI Dashboard** and potentially assigned specific review tasks, investigate the issues and intervene, providing targeted annotations for problematic data samples/predictions. The **Central Server** then acts upon these authorized expert decisions, such as triggering a model rollback or start retraining with new parameters. Furthermore, all alerts and their root causes, expert interventions and subsequent system adaptations are immutably logged to the **Blockchain Logging/Audit System** for full traceability and accountability.
- **Purpose** - ensure that detected issues are not only flagged but are effectively addressed through a combination of human oversight and system adaptations, thereby making sure that the remediation process is auditable and transparent.

This flow serves as a proactive mechanism for ensuring the responsible development of AI models within the HITL-FL system while also guaranteeing that ethical considerations and performance standards are not just one-time checks but an ongoing, integral part of the model life cycle.

Ultimately, this chapter has demonstrated how the proposed architecture translates into a functional system that supports an effective human-AI collaboration. More specifically, the described strategies, mechanisms, and workflows are designed not only to leverage human intelligence and enhance model performance and trustworthiness, but also to ensure that these processes are secure, auditable, and manageable at scale. In the subsequent chapter, the overall HITL-FL architecture will be validated, assessing its practicality and effectiveness against the previously defined attributes, as well as its mapping to the NOUS project and use case application.

Chapter 6

Architectural Validation & Discussion

6.1 Introduction

One key aspect of proposing a new architecture is to present a comprehensive validation and critical discussion of its various components, interactions, benefits, limitations and potential implications. With this in mind, the primary purpose of this chapter is to rigorously evaluate the conceptual soundness, practical applicability, potential effectiveness and overall utility of the proposed architectural framework. Therefore, this validation process aims to assess how well the architecture meets its defined objectives and addresses the challenges inherent in integrating human oversight into FL systems.

The HITL-FL architecture, with its defined C4 model views (System Context, Containers and Components), as well as its solutions to core design problems, as presented in Chapter 4, forms the subject of this evaluation.

Considering this, to detail the validation process and its outcomes, this chapter unfolds like this:

- Section 6.2: **Methodology** details the three strategies chosen for the validation of the proposed system.
- **Section 6.3: Architectural Attributes Fulfillment Analysis** systematically evaluates how the proposed architecture meets the previously defined functional and non-functional requisites.
- **Section 6.4: NOUS Mapping** provides a detailed demonstration of the architecture's application to NOUS overall architecture and Use Case #2, illustrating its practical deployment and the integration of HITL functionalities within this specific context.
- **Section 6.5: Expert Review through Qualitative Interviews** presents the key findings from interviews conducted with NOUS project members.
- **Section 6.6: Discussion of Validation Results** synthesizes the findings from all validation activities, highlighting the architecture's identified strengths and limitations/trade-offs.

- **Section 6.7: Threats to Validity** critically examines potential limitations of the validation study itself.

6.2 Methodology

Regarding the chosen methodology for validating the HITL-FL system, it combines analytical evaluation with practical demonstrations and expert feedback. This approach consists of three primary pillars:

1. **Architectural Requisites Fulfillment Analysis** - a systematic analysis of how the proposed architecture's design addresses the functional and non-functional requisites established in Section 4.2 of Chapter 4.
2. **NOUS Mapping & Demonstration** - the mapping of the HITL-FL architecture to the concrete proposed architecture (NOUS' architecture, Section 2.6.1), alongside its application to a real-world scenario (Use Case #2, Energy Prediction and Data Lifecycle Management) to illustrate its practical instantiation and potential operational flows.
3. **Qualitative Expert Review** - the gathering and analysis of insights from external experts who have not participated in the development of the architecture, gathered through semi-structured interviews that were based on established qualitative research protocols.

By triangulating these three validation strands, the validation process can ensure a comprehensive and robust assessment of the HITL-FL architecture's completeness, practicality and alignment with its stated design goals.

6.3 Architectural Attributes Fulfillment Analysis

To initiate the architecture validation process, a key method employed is the systematic evaluation of how the proposed architectural design of the HITL-FL system addresses the defined functional and non-functional attributes, as presented in Section 4.2.2 of Chapter 4. These attributes establish the core capabilities and characteristics that the architecture is designed to deliver. With this in mind, this section is dedicated to analyzing how each attribute is fulfilled by showcasing the various architectural containers, components, interactions and design choices that contribute to meeting each specific goal.

6.3.1 Functional Attributes Fulfillment

As it was presented previously, in Section 2.5 of Chapter 2, the functional attributes define the capabilities that the HITL-FL system must support to integrate human oversight within FL frameworks effectively. Taking this into consideration, the following analysis details how each functional requisite is addressed in the proposed architecture and how well they are fulfilled.

FR.001: Federated Model Training

- **Requisite** - clients/users shall be able to train local models on their privately owned data, as well as fine-tune them based on the central aggregated global model.
- **Fulfillment** - this task is directly addressed by the FL Compute/Training Module container, showcased in Figure 4.5. In this container, local model training is performed by the Shareable Model Trainer component. In addition, the Model Sync & Update Handler updates the shareable local model with each new version of the central global model, and the Personalized Model Engine component performs the local refinement for adaptation to the user context and environment.

FR.002: Local Data Storage

- **Requisite** - training data shall be stored on a local storage system, accessible only to the respective client.
- **Fulfillment** - the Local Data Store database container, showcased in Figure 4.2 as part of the Client Environment, provides this capability. The data stored in it is accessed by various components within the FL Compute/Training Module for model training, updating, and personalization.

FR.003: Secure Model Update Exchange

- **Requisite** - clients shall securely send model updates to the central server which, subsequently, sends the model updates back to each client.
- **Fulfillment** - on the client side, this action is performed by the Secure Connection Gateway of the Client Environment, presented in Figure 4.2, while on the central server side, the central model updates are sent by the Model Distributor component (presented in Figure 4.4, present in the Central Server container.

FR.004: Centralized Model Aggregation

- **Requisite** - the central server shall aggregate model updates from various clients to create a newly improved global model.
- **Fulfillment** - the Aggregation Engine, detailed in Figure 4.4, is the one responsible for receiving the client updates from the Incoming Model Update Handler and applying the appropriate aggregation algorithms.

FR.005: Model Trustworthiness and Selection

- **Requisite** - the central server shall evaluate the aggregated global model to select the most suitable model version to send back to the clients.
- **Fulfillment** - this is a combination of two components from the Central Server container, both showcased in Figure 4.4: the Trustworthiness Evaluation & Metrics Engine,

which performs the model evaluation against the predefined performance benchmarks and checks its trustworthiness, and the Model Selector & Rollback Logic component that, considering the results from the analysis, selects the most appropriate version to send to the clients (whether it is the newly aggregated version or a validated stable previously stored version).

FR.006: Global Model Predictions and Uncertainty Outputs

- **Requisite** - the central server shall use the aggregated model to generate predictions on a representative dataset, outputting a list of predictions along with their respective uncertainty and confidence scores.
- **Fulfillment** - the Trustworthiness Evaluation & Metrics Engine, illustrated in Figure 4.4, performs this task on a centrally-held proxy dataset. The resulting predictions and their associated uncertainty/confidence scores are then transmitted to the Active Learning & Oversight Trigger Module container (Figure 4.2).

FR.007: Human Expert Oversight and Control

- **Requisite** - human experts shall view the aggregated model state and other metrics, and provide input on the development process (approve/reject, retrain, adjust policies, etc.).
- **Fulfillment** - the AI Dashboard container, within the Human Oversight & Trigger Hub (Figure 4.2), is the main interface for the Human Experts to review and analyze the state of the model and its metrics, since it receives and presents the information from the Central Server, the Active Learning & Oversight Trigger Module and the Continuous Ethical Validator. Regarding expert decisions/control actions, before they are transmitted to the Central Server, they are validated by the Authorization Service.

FR.008: Active Learning Section and Annotation Workflow

- **Requisite** - the system shall identify the most informative predictions for human experts to review and annotate.
- **Fulfillment** - the Active Learning & Oversight Trigger Module contains the Sample Selector Component, illustrated in Figure 4.3, which applies active learning strategy algorithms, like uncertainty sampling or edge-case sampling, to identify the predictions and data samples that deserve human expert attention.

FR.009: Expert Feedback Integration

- **Requisite** - labeled data and high-quality feedback shall be utilized to refine the central, aggregated, global model.
- **Fulfillment** - this is ensured by the Feedback Processor Component, within the Active Learning & Oversight Trigger Module (Figure 4.3). This refined data is then sent to the Central Server container for its Aggregation Engine to fine-tune and include the expert-labeled data into future training rounds.

FR.010: Ethical Validation & Alerting

- **Requisite** - the system shall detect deviations from ethical guidelines or performance degradations, and send alerts.
- **Fulfillment** - the Continuous Ethical Validator, present in the Human Oversight & Interaction Hub (Figure 4.2), constantly monitors the models' state and performance, against defined policies stored in the Governance Framework & Policies Store. Whenever a performance/ethical deviation occurs, not only are alerts generated and sent to the AI Dashboard, for Human Experts to become aware, but also to the Orchestration & Trigger Component of the Active Learning & Oversight Trigger Module (Figure 4.3), for task generation and human analysis.

FR.011: Auditable Blockchain Logging

- **Requisite** - the system shall log cryptographic hashes and metadata records to a blockchain.
- **Fulfillment** - the external Blockchain Logging/Audit System (Figures 4.1 and 4.2) is tasked with this purpose. Various containers log critical information like the model state hashes, human expert decisions, important metadata and other key round events to this blockchain system. For example, the Central Server makes use of its Blockchain Interface (Figure 4.4) to log information, thereby ensuring a secure and immutable audit trail.

FR.012: Historical Data Query and Model Rollback

- **Requisite** - the central server shall query the blockchain for previous performances, validate the model state and review past expert decisions to facilitate a model rollback.
- **Fulfillment** - the Model Selector & Rollback Logic of the Central Server, showcased in Figure 4.4, interacts with its Blockchain Interface component to query historical trusted model identifiers/hashes, as to enable informed rollback decisions.

FR.013: End-User Expert Local Review and Refinement

- **Requisite** - end-user experts shall review the outputs from the personalized model and provide corrections and feedback to update their local data.
- **Fulfillment** - within the FL Compute/Training Module container, the Local Review & Feedback Interface (Figure 4.5) presents outputs of the personalized model to the End User Expert. The feedback received will update the Local Data Store and influence the Personalized Model Engine component.

FR.014: Local Label Propagation

- **Requisite** - client systems shall automatically propagate existing expert-annotated and validated labels to similar unlabeled local data.

- **Fulfillment** - the Label Propagation Service component, illustrated in Figure 4.5, performs the automatic label propagation to unlabeled data that is similar to the end-user expert annotated ones. It reads from and writes to the Local Data Store to enrich the local datasets with minimal additional human effort.

FR.015: System Tradeoff Analysis

- **Requisite** - the system shall load and present to human experts on the dashboard the results of offline or periodic quantitative tradeoff analysis.
- **Fulfillment** - in Figure 4.1, it is possible to evidence the external System Tradeoff Analysis Results system, which performs offline periodic data tradeoff analysis that are consumed and displayed by the AI Dashboard container, to better inform the Human Experts regarding system configuration and operation parameters, as well as helping them decide with a greater understanding [20].

From these detailed descriptions, it is possible to conclude that all functional requirements that describe the main functions and responsibilities of the HITL-FL system are being fully met by all its interacting components.

6.3.2 Non-Functional Attributes Fulfillment

Concerning the non-functional quality attributes, as presented in Section 2.5 of Chapter 2, they represent the behavioral characteristics of a system that determine whether it can meet its defined needs or goals. Taking this into account, this section reviews the previously defined non-functional attributes (Section 4.2.3) in terms of their implementation and the extent to which they are fulfilled.

NFR.001: Security

- **Requisite** - ensure privacy and confidentiality of sensitive data (client data and model updates), model integrity and authorized access to system functions, and prevent data attacks.
- **Fulfillment** - this attribute is one of the main foundations behind this system, and it is addressed through multiple layers and components:
  - **Federated Learning Principles of Data Privacy and Confidentiality** - the FL paradigm itself, supported by the FL Compute/Training Module and Local Data Store (Figure 4.5), ensures that the raw client data remains decentralized.
  - **Mechanisms of Encryption and Anonymization in Transmitted Client Data** - when the Secure Connection Gateway sends the client model updates to the Central Server (Figure 4.2), privacy-enhancing techniques, like differential privacy [60], are considered to protect data contributions further. At the same time, encryption is used for data in transit. On a related note, metadata (data source characteristics, timestamps and other relevant data) that are sent in conjunction with the client model

gradients/model parameter changes is also anonymized to ensure that only data that can be shared with the Central Server is sent through. Additionally, the Secure Connection Gateway utilizes TLS over MQTT [7, 60], which helps ensure confidentiality and data integrity during transit, as well as prevent data leakage attacks.

- **Secure Annotation Tasks** - to initiate the active learning task, the Central Server sends prediction-data sample pairs to the Active Learning & Oversight Module. To guarantee security, these pairs undergo privacy-preserving transformations (like de-identification and feature masking [1], which remove any critical private data fields and ensure that human experts only see the essential) [46]. The transformed data is then prepared for the expert. The human expert annotations and feedback, when submitted back through the HITL Secure Gateway & Task Router, are also encrypted with secure cryptographic proof/tag (ensuring integrity and non-repudiation) [60], which will be then used by the HITL gateway to verify proof and forward the securely verified expert oversight to the active learning module. It is also important to note that the transmission of data between the Active Learning & Oversight Trigger Module and the HITL Secure Gateway & Task Router occurs by TLS over MQTT [60].
- **Authentication** - during the active learning loop, the HITL Secure Gateway & Task Router has many responsibilities, including handling authentication [29]. This process enables Human Experts to participate in the annotation/labeling task of the active learning workflow, as the secure gateway, through its internal matching component, forwards the annotation task to the most suitable expert.
- **Authorization Access** - the Authorization Service (Figure 4.1), acts as Policy Decision Point, evaluating actions against defined user roles and permissions, which are in accordance with the Governance Framework & Policies Store. This container governs access to the AI Dashboard and HITL Secure Gateway & Task Router functionalities for Human Experts. In this context, IAM principles are assumed for identity management.
- **Secure Model Aggregation** - the Aggregation Engine performs the task of model aggregation, illustrated in Figure 4.4. For this process to be secure, a protocol like the MPC-based ones could be used [60], as it prevents the server from seeing individual client updates. Another way to ensure a secure aggregation is to filter malicious client updates before the model metrics are presented to the human experts, which makes their oversight much more trustworthy.
- **Model Integrity** - the external blockchain system stores cryptographic hashes of validated model versions, allowing for integrity verification.

NFR.002: Trustworthiness & Explainability

- **Requisite** - users shall trust and effectively interact with a system, particularly one that is perceived to make reliable, ethical and transparent complex decisions in sensitive environments, fostering the confidence of all users.

- **Fulfillment** - guaranteeing that Human Experts and, by extension, end-users who interact with this particular system, have the confidence not only in its outputs, but also in interacting with it, was one of the main objectives that influenced and marked the design of the HITL-FL system. This focus is addressed in the following containers/components:
  - **HITL Paradigm** - the overall inclusion of human oversight and judgment into the AI life cycle already helps make the whole AI process trustworthy and easier to understand to the users, as their own input is included.
  - **AI Dashboards** - within the Human Oversight & Interaction Hub, there is an AI Dashboard container (Figure 4.2) that serves as the focal point for showcasing the overall state of the central global model to Human Experts. Through interacting with the dashboard, which is perceived to be designed with HCAI and XAI principles, users are exposed to the model's performance metrics, explainability visualizations, fairness audits and contextual information, thereby gaining a better understanding of the current state of the AI model. Complementary to this, users also have access to risk assessments (by critically evaluating potential impacts and safety tradeoffs provided by the external System Tradeoff Analysis Results system), client contributions, model representations, and uncertainty clusters, allowing the dashboard to become an active exploration tool [2]. In consequence, model behavior becomes more transparent to Human Experts.
  - **Ethical Validation** - performed by the Continuous Ethical Validator (4.2), in conjunction with the Governance Framework & Policies Store, the system constantly monitors the model's ethical alignment and performance to avoid not only ethical and performance drifts but also to provide transparency into the decision-making criteria. The policies by which the ethical validator container evaluates the model's performance are derived from or in accordance with the GDPR European laws, for example.
  - **Expert Justifications** - whenever user experts are allowed to contribute to the AI life cycle by making decisions through the AI Dashboard or labeling data and providing feedback through active learning processes, these actions are also accompanied by their justifications or reason codes, thereby adding another layer of explainability to the feedback loop.

NFR.003: Auditability

- **Requisite** - provide a clear, immutable record of not only all events and process steps but also model versions, metadata, and key expert decisions so that at all times, there is a record of what happened and who did what, enabling process debugging and building trust.
- **Fulfillment** - in this system, the external Blockchain Logging/Audit System is responsible for storing and securing an immutable log of all critical events, data artifacts and decisions during each FL round. This blockchain system receives information from:

  - **The Central Server** - through its Blockchain Interface (Figure 4.4), the server logs to the blockchain key information, such as cryptographic hashes of validated versions of the global model and their states (that includes performance and fairness metrics, relevant contextual summaries, triggered control actions, from the AI Dashboard, provided justifications and aggregated model version identifiers). It also logs other operational events, such as timestamps, aggregation rounds, errors encountered and model parameter changes. This minute logging enables audit and traceability.
  - **HITL Secure Gateway & Task Router** - it logs records of Human Expert decisions and action hashes, alongside their provided justifications, and other HITL gateway interactions.

  This blockchain external system also enables containers/components, such as the Continuous Ethical Validator or the Model Selector & Rollback Logic (present in Figures 4.2 and 4.4), to query for historical logs and analysis, or trusted model identifiers and hashes, respectively.

NFR.004: Usability

- **Requisite** - ability for Human Experts to interact with the system and their respective allocated components to understand model behavior, perform validation and provide feedback easily.
- **Fulfillment** - the designed HITL-FL system integrates specific containers and modules that perform this specific task:
  - **AI Dashboard** - showcased in Figure 4.2, the dashboard provides a dedicated interface for Human Experts to review the model's current state through clear visualizations and interactive visualization tools. One of its main goals is to ensure that expert oversight is effectively included and that their comprehension does not pose any issues.
  - **Expert Task Router** - as part of the HITL Secure Gateway & Task Router container (Figure 4.2), this internal component is responsible for utilizing matching strategies based on expert specialty, workload, and performance to allocate annotation tasks to the most suitable human experts.
  - **Most Relevant Prediction/Sample Selection** - the Active Learning & Oversight Trigger Module contains the Sample Selector Component (Figure 4.3), which is tasked with selecting the most informative and critical prediction-sample pairs to be reviewed, annotated [52] and labeled by Human Experts, thereby enhancing the perceived efficiency of this process.

NFR.005: Adaptability & Maintainability

- **Requisite** - the ease with which the system can be modified and corrected to incorporate new FL algorithms, expert tasks, ethical guidelines or integrate with other external systems.

- **Fulfillment** - one of the key concerns when designing the HITL-FL system was the clear separation of responsibilities throughout the various modules and services that comprise the system. Taking this into consideration, this design choice can be seen manifested in the following architectural characteristics:
  - **Modular Container/Component Design** - it is evident from Figures 4.2, 4.3, 4.4 and 4.5, which represent the Container and Component diagrams of the HITL-FL system, that there is a precise distribution and distinction of responsibilities through the various modules of the system, interconnected via well-defined interfaces (APIs or event communication). This modularity ensures that whenever there is a need to update or replace one of these services, it does not affect other components or containers.
  - **The Governance Framework & Policies Store** - as presented in Figure 4.2, allows for the update of ethical rules, fairness thresholds, and other policies without hassle or significant changes to the orchestration logic.
  - **HITL Adaptability** - the mechanisms that allow the integration of human oversight, by themselves, contribute to the system's adaptability to evolving understanding or changing data characteristics.

NFR.006: Scalability

- **Requisite** - the system's ability to handle a growing number of clients, data volume, complexity of models and interactions.
- **Fulfillment** - various design choices were implemented when designing the HITL-FL system to enable the system to continue operating stably as growth occurs across multiple dimensions, such as client numbers, data volumes, model complexity, or the scale of human expert interactions. These choices are as follows:
  - **Inherent FL Scalability** - one of the foundational aspects of FL is that its client model training is decentralized (as shown in Figure 4.5 and 4.2), which enables the system to scale with an increasing number of participating clients. The Central Server handles and aggregates client model updates, as this container does not deal with raw client data. Therefore, it avoids the central data ingestion bottleneck.
  - **Scalable and Modular Services** - within the HITL-FL system, as presented in the previous non-functional attribute, modularity is one of the key characteristics that make it up. As a consequence, individual services can be scaled horizontally (for example, by deploying more instances) based on their specific load if the volume of expert interactions increases [36].
  - **Asynchronous Workflows** - within the Human Oversight & Interaction Hub (Figure 4.1), MQTT is used for communication between the Active Learning & Oversight Trigger Module and the HITL Secure Gateway & Task Routing [7]. The usage of this

event-driven communication allows the central model aggregation and evaluation cycles to decouple themselves from the human interaction stages. As a consequence, the FL process is not blocked by potentially variable HITL processing times. In addition, client-server communication also utilizes MQTT, as it supports multiple clients to connect/disconnect and asynchronously submit their model updates [7]. Consequently, it also enables the central server to process these updates effectively without requiring tightly synchronized communication, thereby supporting a large and dynamic client base.

NFR.007: Performance

- **Requisite** - the system shall demonstrate adequate responsiveness regarding the time required for an aggregation round or an expert task to be completed, as well as sufficient throughput (like the number of clients supported).
- **Fulfillment** - related to Scalability (NFR.006), the system was designed to ensure that, although HITL inherently introduces *latency*, the model presents timely updates, responsive expert interfaces, and the avoidance of bottlenecks. With this in mind, the following characteristics of the system are highlighted:
  - **Optimized Aggregation Algorithms** - in the Aggregation Engine, showcased in Figure 4.4, the FedDyn algorithm can be utilized, as it is one of the most effective FL aggregation algorithms. However, if there are bandwidth issues, other algorithms, such as FedAvg or the Model Setup approach [51], can be used.
  - **Targeted Expert Involvement** - the active learning module ensures that Human Experts only work on and are exposed to relevant and informative prediction-data sample pairs, thereby making the overall labeling and feedback process more effective in terms of model improvement.
  - **Efficient Communication** - the usage of MQTT for event-driven communication offers various performance benefits. Due to its publish-subscribe model, the network overhead can be reduced, and responsiveness can be improved, whether for delivering model updates or annotation/review task notifications, especially when compared to traditional synchronous polling for frequent API calls.

Through this detailed analysis, it is possible to verify that the proposed HITL-FL architecture is designed to comprehensively address all defined functional requirements, as well as to exhibit the desired non-functional quality attributes, thereby laying a strong foundation for a trustworthy, secure and effective system. However, it is also essential to consider that there is still future work that could be done to address these attributes further and fulfill them, especially in terms of scalability and performance, for example, which, although relevant to the system, are not one of the critical focuses of this work.

6.4 NOUS Mapping

To further evaluate the practical applicability of the proposed HITL-FL reference architecture, this section is focused on presenting its mapping to the NOUS' architecture and its second use case (Use Case #2 - Energy Prediction and Energy Data Lifecycle Management). Due to its ambitious goals concerning data sovereignty, distributed AI and the need for trustworthy systems, as stated in Chapter 2, in Section 2.6, the NOUS project serves as the main real-world context to instantiate and explore the proposed architectural concepts.

6.4.1 HITL-FL System & NOUS

Considering the NOUS architecture presented in Section 2.6.1 of Chapter 2, this section outlines the practical implementation of the HITL-FL system within the existing components and services of the NOUS project infrastructure. This exercise aims to not only show how the conceptual elements of the HITL-FL system can align with its concrete NOUS counterparts, but also highlight the ways this proposed architecture extends the currently proposed NOUS ecosystem to allow for the integration of human oversight, one of the project's objectives, as stated in Section 2.6.

6.4.1.1 Mapping the Conceptual HITL-FL Architecture onto NOUS

Descriptions from A. to K. illustrate how each major logical grouping or key container/component of the HITL-FL architecture is mapped to its NOUS counterpart.

A. FL Compute/Training Module

Model Sync & Update Handler
NOUS Counterpart - Data Agent → Applications & Data Orchestration → Federated Learning.
Details - opens a TLS channel through the Service Gateway (VPN / Firewall); sends encrypted gradients to the central server and listens for global-model downloads.

Local Training Orchestrator
NOUS Counterpart - same FL Component, working with Data Agent → Workflow Configuration + Data Bus (Streamhandler).
Details - starts either shareable-model training or personalized fine-tuning when triggers appear (like new labels).

Shareable Model Trainer
NOUS Counterpart - same FL Component, running on local AI-Provisioning resources (CPU, GPU, or HPC).
Details - reads Raw Local Data; trains the shareable local model (no fine-tuning); hands the resulting gradients to the Model Sync & Update Handler.

Personalized Model Engine
NOUS Counterpart - same FL Component, invoked after expert feedback.

Details - fine-tunes the personalized copy of the model with the expert's decisions; writes predictions + confidence scores to the Formatted Data.

Label Propagation Service
NOUS Counterpart - new microservice under Data Agent → Applications & Data Orchestration.
Details - reads expert labels from Data Store Adapter; applies domain logic to propagate labels to similar unlabeled samples; writes propagated labels back into Formatted Data for the next training cycle.

Local Review & Feedback Interface
NOUS Counterpart - UI widget inside the FL Component.
Details - displays model outputs to the end-user expert and captures corrections sent to the Personalized Model Engine.

Local Data Validator & Pre-Processor
NOUS Counterpart - data Agent → Applications & Data Orchestration.
Details - checks data quality, removes outliers, and prepares batches before any training or fine-tuning run.

End User Expert
NOUS Counterpart - Common Administration Services → Identity & Access Management (IAM) → Roles of "local_data_contributor" or "personalized_model_reviewer".
Details - interacts primarily with the Local Review & Feedback Interface to provide feedback on personalized models.

The client-side functionalities of the HITL-FL system, encapsulated within the FL Compute/-Training Module container, are mapped to corresponding services within the NOUS Data Space Participant's Data Agent. As detailed above, core federated learning operations, including local data orchestration and end-user interactions, utilize the NOUS components, such as the Federated Learning service, Workflow Configuration, Data Bus, and AI-Provisioning for local computing, while ensuring that local data remains private and secure.

B. Secure Communication Layer

Secure Connection Gateway (MQTT / TLS broker)
NOUS Counterpart - Common Administration Services → Service Gateway (API Gateway exposes endpoints, Policy Enforcement Point (PEP) checks every call, Network uses VPN and Firewall); Identity & Access Management → Certification Authority; Infrastructure Agent → Admin Services → Encryption Key Management.
Details - sets up the TLS channel that all FL traffic uses; both gradient uploads and model downloads go through the same API path, protected by the VPN + firewall and filtered by the PEP. The Certification Authority and Key-Management supply TLS certificates and keys, and protect cryptographic material (gradients and models).

The HITL-FL architecture relies on a dedicated Secure Connection Gateway at each client node to manage the channels for communicating with the central server. In the NOUS framework, this layer is realized through its Service Gateway, VPN and key management services, which ensure that all FL traffic between clients and the central server remains encrypted, authenticated and policy-controlled.

C. Active Learning & Oversight Trigger Module

Sample Selector Component
NOUS Counterpart - Data Space Participant → Data Agent → Applications & Data Orchestration → Federated Learning + Data Analytics / XAI API.
Details - calls the XAI API to pull each sample's prediction + confidence; ranks samples by uncertainty and saves the chosen IDs in a small in-memory list.

Orchestration & Trigger Component
NOUS Counterpart - new microservice deployed in the Data Agent → Applications & Data Orchestration.
Details - reads policy rules from the Governance Policies Store; decides which samples need human expert review; publishes annotation task requests as MQTT events, consumed by the HITL Secure Gateway & Task Router; actions are authorized via IAM and critical triggers may consult the central Authorization Service.

Feedback Processor Component
NOUS Counterpart - Service Gateway (for secure callbacks) → Data Agent → Data Persistence → Data Store Adapter.
Details - subscribes to MQTT events for completed annotations; processes received expert labels; writes verified labels to Formatted Data for training components.

Internal State Store
NOUS Counterpart - deployed next to the Data Agent.
Details - tracks workflow status for each sample; avoids extra load on disk-backed storage; cleared after every aggregation round.

The Active Learning & Oversight Trigger Module is a crucial component of the HITL-FL architecture, responsible for intelligently selecting data for human review, orchestrating the annotation workflow and processing user feedback. In this specific case, it utilizes NOUS' analytics and orchestration components to automate the selection process of uncertain samples for human review and to handle user feedback. More specifically, for example, the Sample Selector leverages NOUS' XAI and Data Analytics APIs to rank uncertain samples for review. At the same time, the Orchestration and Trigger logic is perceived as a new NOUS microservice that interprets governance policies to determine which samples require human annotation and triggers those annotation tasks through IAM-controlled MQTT events. Another component to highlight is the Internal State

Store, located within the Data Agent, which tracks the workflow status of each sample throughout this cycle.

D. Central-Server (Global-Orchestration) Layer

Incoming Model Update Handler
NOUS Counterpart - Data Space Infrastructure Participant → Infrastructure Agent → AI Provisioning → Federated Learning service.
Details - receives clients' encrypted gradients through the Service Gateway; IAM authenticates and authorizes the sender; FL service decrypts before aggregation.

Aggregation Engine (federated-learning algorithm)
NOUS Counterpart - Federated Learning service inside AI Provisioning.
Details - combines all client gradients (with FedDyn or FedAvg) [51] to build a new global model; runs on the central CPU/GPU cluster managed by the Infrastructure Agent.

Trustworthiness Evaluation & Metrics Engine
NOUS Counterpart - new microservice deployed in the Infrastructure Agent that reads models from PaaS Service → Database (aggregated-model store).
Details - tests freshly aggregated model on a held-out set; calculates accuracy and fairness scores; stores results for audit and policy checks.

Model Selector & Rollback Logic
NOUS Counterpart - new microservice in the Infrastructure Agent, calls Common Admin Services → Policy Enforcement Point (PEP) and PaaS Service → Blockchain.
Details - examines trust metrics, applies policy rules (from Governance Policies Store); marks model as "approved" if policy met, else rolls back to last blockchain-approved version.

Model Distributor
NOUS Counterpart - Federated Learning service plus Service Gateway (API Gateway + VPN).
Details - pushes approved global model over TLS to every Data Agent topic (e.g., '/nous/models/new'); clients pull automatically.

Model Store Interface
NOUS Counterpart - PaaS Service → Database (aggregated-model registry).
Details - stores every approved or rolled-back model binary, with metadata and version tag.

Blockchain Interface
NOUS Counterpart - PaaS Service → Blockchain (NOUS-hosted ledger).
Details - records model hashes, approval decisions, expert IDs and aggregation-round events; queries the ledger for trusted fallback models.

This mapping illustrates the overall workflow that occurs in the Central Server. Gradients that arrive through the Service Gateway are aggregated in the AI Provisioning service (Federated

Learning service) and evaluated by the new trust-metrics engine. The resulting model is then either approved or rolled back, with every decision and key metadata logged on the blockchain. Finally, the approved model is stored in the database (Aggregated Model Store) and distributed back to all client nodes over the same secure channel via the Service Gateway.

E. Governance Framework & Policies Store

Governance Framework & Policies Store
NOUS Counterpart - Policy Repository / Administration Point (PAP), implemented as a specialized PaaS Service → Database or dedicated Configuration/Policy Management Service.
Details - stores policy directives such as fairness thresholds and role-based permissions; application components query this store for operational rules and thresholds; the HITL-FL Authorization Service (system's PDP) consults this store for policies and roles, enforced by PEPs in gateways or application interceptors.

The Governance Framework & Policies Store is conceptualized within NOUS as a central Policy Administration Point (PAP). This store is critical as it holds the definitions for operational rules, ethical guidelines, fairness thresholds and access control policies. Various components throughout the HITL-FL architecture rely on querying this repository to guide their decision-making processes and ensure compliant system behavior.

F. AI Dashboard & Human-Expert Interfaces

HITL Secure Gateway & Task Router
NOUS Counterpart - Common Administration Services → Service Gateway (API Gateway + VPN / Firewall).
Details - a new microservice behind NOUS's standard Service Gateway; subscribes to MQTT events for 'review-this-sample' tasks; uses internal Task Router to assign tasks to Human Experts (via IAM roles); presents tasks via its annotation interface; experts submit annotations back through this Gateway, which are published as MQTT events.

AI Dashboard
NOUS Counterpart - User Dashboards & Services → Data Visualization / Config-&-Management Dashboards.
Details - runs as a web application in the dashboard layer; allows Human Experts to oversee model performance, fairness and monitoring; all actions travel through the Service Gateway for logging, IAM enforcement and oversight.

Human Expert
NOUS Counterpart - Identity & Access Management → Roles ("expert_labeler", "model_auditor", "trust_reviewer", for example).
Details - logs in via single-sign-on; uses annotation page (served by HITL Gateway) to add or

correct labels; uses AI Dashboard to inspect metrics, validate model quality and issue control actions.

As detailed, the Human Expert, authenticated and role-managed by NOUS' Identity & Access Management, interacts with the system in two different ways, depending on the task at hand. For detailed, sample-specific tasks like annotation, the HITL Secure Gateway & Task Router is conceptualized in the NOUS framework as a new microservice that leverages NOUS' Service Gateway infrastructure, providing a secure and dedicated communication channel for human oversight. For model oversight and monitoring, the Human Experts interact with the AI Dashboard that is integrated within NOUS' User Dashboards & Services. This dual-interface approach ensures that different types of expert interaction are managed efficiently and securely.

G. Local Data Store

Local Data Store (on-device repository for all training and feedback data)
NOUS Counterpart - Data Space Participant → Data Agent → Data Persistence: Raw Local Data bucket, Formatted Data bucket, Data Store Adapter.
Details - holds all node data: raw inputs, expert labels, propagated labels, predictions, personalized model artifacts; FL Component reads raw files for training, writes predictions as formatted records; Label Propagation Service updates labels; Streamhandler broadcasts new rows; only encrypted gradients and high-level metrics ever leave the store.

As detailed above, the HITL-FL architecture's Local Data Store maps to a structured data persistence layer within the NOUS Data Agent. This includes distinct storage for raw and formatted data, accessed via a uniform Data Store Adapter. This setup ensures that all client-side operations, from initial training on raw data by the FL Compute/Training Module to the storage of expert-annotated labels and the application of the label propagation process, are underpinned by a local and secure data repository, with strict controls on what information leaves the client environment.

H. Continuous Ethical Validator

Continuous Ethical Validator
NOUS Counterpart - new microservice that plugs into: Data Space Infrastructure Participant → PaaS Service → Database; Common Administration Services → Policy Enforcement Point (PEP); User Dashboards & Services → AI Dashboard.
Details - downloads fresh global model after each aggregation round; runs tests: accuracy, fairness metrics, concept-drift; compares with thresholds from Governance Framework & Policies Store; raises "ethical/performance alert" via Service Gateway to AI Dashboard and Orchestration & Trigger Component; writes evaluation summary to database for traceability.

The mapping detailed above positions the Continuous Ethical Validator as a new, proactive microservice within the NOUS ecosystem. It interacts with NOUS' PaaS database to access aggregated models, consults policy enforcement points for evaluation thresholds (derived from the

Governance Framework & Policies Store) and forwards alerts through NOUS' Service Gateway to both the AI Dashboard, for human awareness, and the Orchestration & Trigger Component, for automated or HITL-mediated responses. This ensures continuous, policy-driven, ethical and performance oversight for every global model iteration.

I. Blockchain Logging / Audit System Interface

Blockchain Interface (within Central Server)
NOUS Counterpart - Data Space Infrastructure Participant → PaaS Service → Blockchain.
Details - signs each significant event, writes it as a transaction to NOUS blockchain; queried by Model Selector & Rollback Logic for trusted model version or audit trail.

The blockchain interface component of the HITL-FL architecture directly leverages NOUS' PaaS Blockchain service. This interface acts as a dedicated wrapper, responsible for recording critical system events, model states and expert decisions onto the NOUS ledger. It also enables functionalities like trusted model rollback by the Model Selector & Rollback Logic through queries to the blockchain, thereby ensuring a high degree of transparency and verifiability for the entire HITL-FL process.

J. Aggregated Model Store

Aggregated Model Store
NOUS Counterpart - Data Space Infrastructure Participant → PaaS Service → Database.
Details - stores every aggregated, approved, or rolled-back global model binary with version tags and metadata; used by Trustworthiness Evaluation & Metrics Engine and Model Distributor to access/download models.

The HITL-FL Aggregated Model Store, as detailed in this section, is realized using NOUS' PaaS Database services, thereby functioning as a central model registry. This store is critical for storing all versions of the global aggregated model, including those approved or resulting from rollbacks, complete with their associated metadata. It serves as the authoritative source for the Trustworthiness Evaluation & Metrics Engine during its model assessment and for the Model Distributor when disseminating models to clients, ensuring consistency and access to validated model artifacts.

K. Authorization Service

Authorization Service
NOUS Counterpart - new microservice within the Common Administration Services; acts as Policy Decision Point (PDP); leverages IAM for identities and roles; reads policies from Governance Framework & Policies Store.
Details - centralizes fine-grained authorization logic for HITL operations; evaluates permissions

for critical actions (approve model, submit annotation, trigger retraining) based on policies and roles; AI Dashboard and HITL Secure Gateway & Task Router query this service before executing sensitive operations.

The Authorization Service within the HITL-FL architecture is presented as a new microservice, part of NOUS's Common Administration Services that acts as a central Policy Decision Point (PDP). It leverages NOUS' existing Identity & Access Management (IAM) for user identities and roles and consults the Governance Framework & Policies Store for specific authorization rules.

6.4.1.2 NOUS Architectural Extension

The proposed HITL-FL system, as observed in Section 6.4.1.1, leverages the various services and modules offered by the NOUS reference architecture. However, it also significantly extends the foundational capabilities of this data space framework.

While NOUS provides essential infrastructure, like common administration services and basic data processing blocks (Section 2.6.1), the HITL-FL architecture introduces new microservices/-components, specialized workflows and new applications of these offered services as to address the unique requirements that are inherent to the development of a trustworthy, ethical and auditable AI system, that integrates human oversight. The key architectural extensions are detailed as follows:

1. **Introduction of a specialized Human Oversight & Interaction Hub** - the HITL-FL architecture delineates a dedicated Human Oversight & Interaction Hub as a collection of integrated application-layer services. This hub, comprising the AI Dashboard (for expert oversight and XAI), the Active Learning & Oversight Trigger Module, the Continuous Ethical Validator, the HITL Secure Gateway & Task Router, the Governance Framework & Policies Store and the Authorization Service represents a substantial extension beyond the generic user dashboard and data visualization services initially described in NOUS, as it provides specific application logic and interfaces to manage the lifecycle of HITL processes, tailored for AI model development and governance.

2. **Formalization and Operationalization of Ethical AI and Trustworthiness Mechanisms** - a core contribution of the HITL-FL architecture is the explicit integration of ethical AI principles. The Continuous Ethical Validator, in conjunction with the Governance Framework & Policies Store and the XAI techniques in the AI Dashboard, introduces concrete mechanisms for embedding dynamic ethical assessments, fairness evaluations, and compliance checks directly within the federated learning workflow.

3. **Integration of Advanced Active Learning Strategies for Federated Learning** - the Active Learning & Oversight Trigger Module optimizes the utilization of expert oversight and enhances the efficiency of the HITL process in an FL context. While NOUS may support general "Data Analytics," this specific application of active learning techniques to guide human intervention in a distributed learning environment is an extension to be highlighted.

4. **Purposeful Application of Blockchain Technology for Comprehensive AI Auditing and Governance** - NOUS offers Blockchain as a PaaS component. The HITL-FL architecture explicitly applies this technology to establish an immutable and transparent audit trail (like an "Ethical Black Box"). This ledger records critical artifacts and events, including model version hashes, expert decisions, validation outcomes and operational metadata, which enhances the audibility, accountability and verifiability of the HITL processes, cementing itself as a significant extension of NOUS' generic blockchain service.

In conclusion, the HITL-FL architecture extends NOUS not by just replacing its core services but by building a specialized layer upon it, where novel components and specific data flows are able to transform NOUS into a targeted platform capable of governing AI systems that have a strong focus on human collaboration, ethical considerations, security and auditable trust.

6.4.2 NOUS Use Case (Energy Prediction and Energy Data Lifecycle Management)

As presented in Chapter 2, the second NOUS use case is entitled *Energy Prediction and Energy Data Lifecycle Management (PPC)*, and has the goal of leveraging the NOUS framework to enhance PPC's [1] (a major energy company in Greece) capabilities in two key areas:

1. **Energy Prediction Tasks** - improve the accuracy and efficiency of short-term energy production/demand forecasting.
2. **Energy Data Lifecycle Management** - improve the management, security, integrity and confidentiality of PPC's energy-related data.

This use case, due to its nature and surrounding themes, was chosen to map and validate the HITL-FL proposed architecture within the context of a real-world scenario.

PPC's primary goal is to improve its operations and services by becoming more data-driven, as dealing with the available data across multiple systems, often in isolation, has been a persistent issue. Additionally, its energy prediction tasks are too expensive and do not reach the desired accuracy. That brings us to the NOUS project, which provides PPC the opportunity to assist with the many challenges associated with energy predictions. The NOUS Framework can be used to adopt new technologies, including:

- Advanced ML models, such as Quantum ML, for more accurate energy forecasting.
- A unified technological framework that integrates these advanced AI capabilities, improves data handling and streamlines processes.

Concerning the challenges addressed by NOUS, they include:

- **Limited Accuracy & Timeliness** - improving the precision and speed of energy forecasts.

[1] https://www.ppcgroup.com/en/ppc/

- **Data Management Complexity** - enhancing the way data is collected, processed and utilized from various sources, like PV Parks and weather stations (which can be accessed via APIs).
- **Integration of Advanced Technologies** - providing access to and support for computationally intensive models and distributed learning techniques.

Given the collaborative nature of the NOUS project, one of its primary defining characteristics is its involvement with multiple European organizations and partners, each of which provides specific services that interact with one another. In the case of Use Case #2, numerous partners contribute to it, each performing a particular role, as highlighted below:

- **PPC (Use Case Provider)** - the energy company at the center of the use case, which provides the real-world problem (energy prediction and data lifecycle management), the necessary data and acts as the primary beneficiary of the developed solutions. Their data scientists and system operators are key human experts involved in the HITL processes.
- **AETHON (Coordinator)** - leads and coordinates this use case. AETHON also contributes with an "auto-standardiser" tool for enhancing data consistency and quality.
- **NCSRD** - responsible for developing advanced ML models, including DL approaches and exploring the potential of Quantum ML, specifically for the energy prediction task. These models will leverage HPC resources.
- **ITML** - develops and implements the FL framework. This framework allows models to be trained on data from PPC's various sources (like stations and factories) without needing to centralize the raw sensitive data.
- **POLITO** - focuses on integrating HPC and Quantum computing resources necessary for training advanced models. Additionally, it provides Blockchain solutions for ensuring data integrity, creating auditable trails and supporting secure data lifecycle management.
- **AEGIS** - provides cybersecurity solutions for the NOUS framework (specifically for data exchange within the use case).
- **UNIMORE** - contributes to the data lifecycle management by focusing on developing standardized processes for managing production data.
- **Other NOUS Partners** - entities like ARCTUR (for HPC integration) and NETCOMPANY (common data storage solutions) also contribute to providing infrastructure and services for the considered use case.

To better understand the overall functioning and flow of this use case, a data flow diagram was created, as showcased in Figure 6.1.

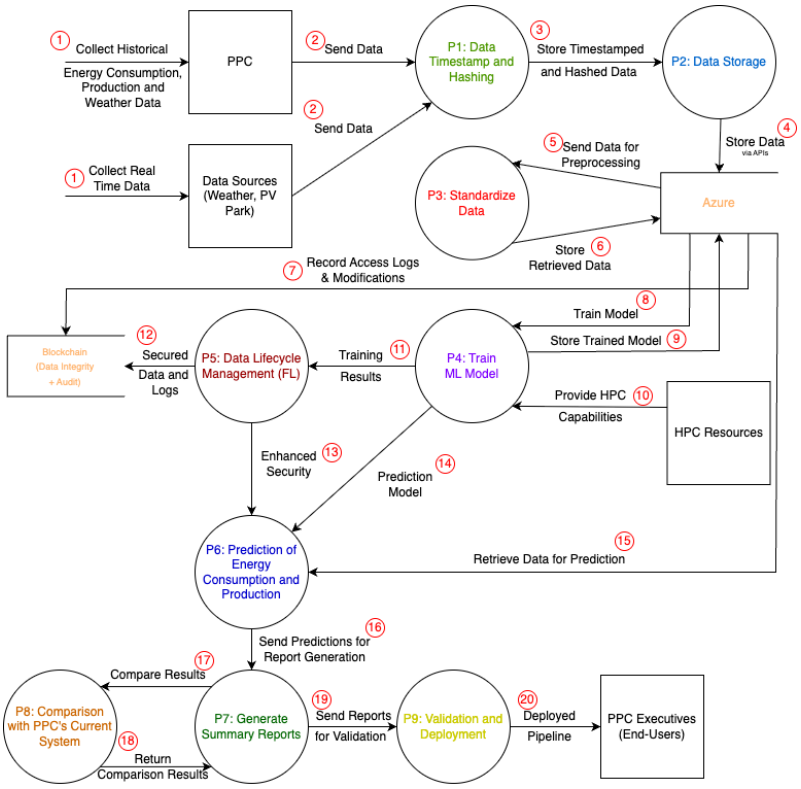

Figure 6.1: Use Case #2 Data Flow Diagram

The end-to-end pipeline of Use Case #2 begins with the PPC and other external sensors/APIs, such as weather stations, PV parks and market feeds, which collect historical and real-time measurements (**1**). Every incoming record is then immediately forwarded to the Timestamp & Hashing service, which appends an immutable timestamp and cryptographic hash (**2** and **3**). In parallel, the raw data streams are persisted in a cloud data service, such as Azure, and handed to AETHON's Auto-Standardizer for standardization, which includes processes like schema harmonization and basic data cleaning (**4**, **5**, and **6**). All-access logs and modifications are recorded on a Blockchain to ensure data integrity through a tamper-proof audit (**7**).

Once curated, the data feeds the model-training branch, where NCSRD uses the standardized data to train ML or quantum-enhanced models, thereby leveraging the HPC resources provided by ARCTUR/POLITO for heavy computation (**8** and **10**). As a consequence, the resulting artifacts (like the trained model) are stored back in the data storage (**9**). Subsequently, the training results feed the other branch, the lifecycle-management branch, where blockchain services and ITML's FL framework keep sensitive data local while exchanging only model parameters (**11** and **12**).

The trained predictor is deployed to the Prediction module, which pulls fresh data from Azure, the cloud data storage (**15**), and generates short-term forecasts of energy demand and production (**14** and **16**). With these forecasts, executive-level summary reports are produced, which are then benchmarked against PPC's existing system (**17** and **18**). From these results, metrics and any detected drifts, the Validation and Deployment stage is reached, where the loop is closed by PPC

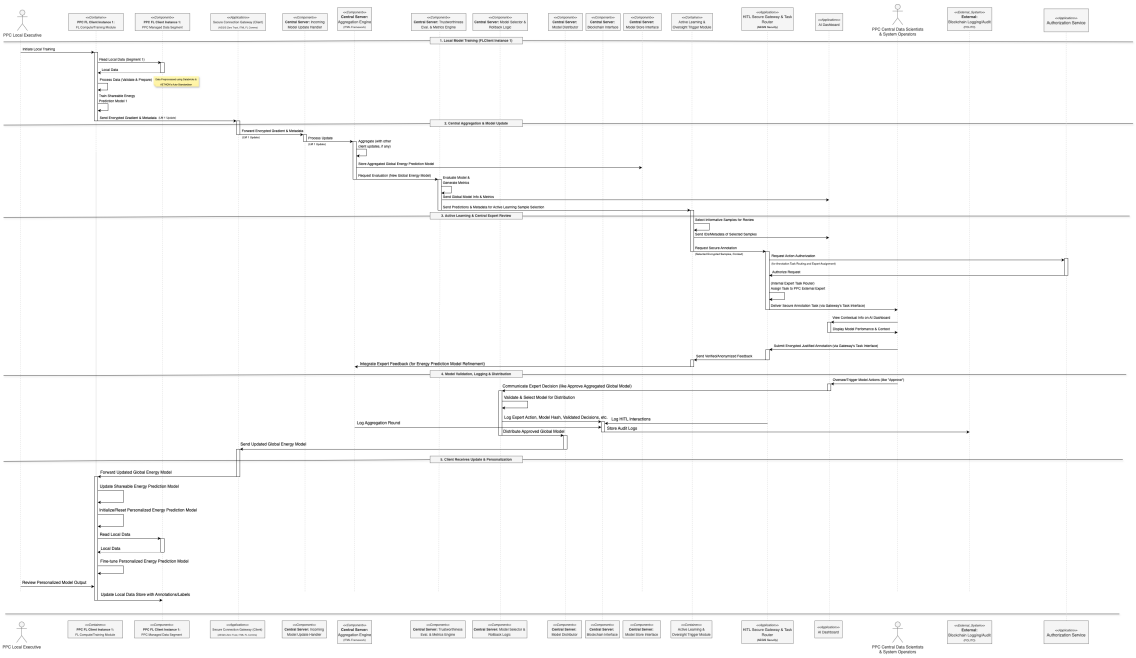

Figure 6.2: NOUS Use Case #2 Sequence Diagram - End-to-End Federated Learning Round with Centralized Active Learning and Local Personalization (Scenario A)

Executives validating the deployed pipeline and results, or ordering a rollback to previous model versions and orchestrating the CI/CD pipeline for subsequent iterations (**19** and **20**).

6.4.3 Application to NOUS Use Case #2

As stated in Section 6.4.2, to further validate the conceptual HITL-FL architecture in a practical setting, this section presents its application to the NOUS Use Case #2 (Energy Prediction and Data Lifecycle Management). This process is done through three sequence diagrams (Figures 6.2, 6.3 and 6.4), each representing distinct scenarios, to illustrate the operational flow and the integration of HITL functionalities as thought out by the proposed architecture. It is also important to note that, in each diagram, the components and containers of the conceptual diagram were mapped to the partners of the use case.

6.4.3.1 Scenario A(End-to-End Federated Learning Round with Centralized Active Learning and Local Personalization)

Figure 6.2 consists of Scenario A, which illustrates a comprehensive end-to-end FL cycle. This scenario showcases the dynamic interplay between local client operations, central server aggregation, HITL active learning process involving central experts and subsequent local model personalization, which were all orchestrated by the proposed HITL-FL architecture.

The sequence begins with the **PPC Local Executive** initiating local model training within a PPC FL Client Instance's FL Compute/Training Module. This module accesses its designated

PPC Managed Data Segment (representing PPC's local data preprocessed by **AETHON's Auto-Standardiser** and **Databricks**), trains a shareable energy prediction model and securely transmits the encrypted model update via the client-side **Secure Connection Gateway** (leveraging **AEGIS security** and **ITML communication protocols**).

Upon arrival at the **Central Server** environment, the update is processed by the **Incoming Model Update Handler** and passed to the **Aggregation Engine** (reflecting **ITML's FL framework**). Here, itis aggregated with other client updates to form a new global **Aggregated Energy Prediction Model**, which is then stored via the **Model Store Interface**. The **Trustworthiness Evaluation & Metrics Engine** subsequently assesses this global model, generating performance metrics and uncertainty scores. This vital information is forwarded:

- Metrics are sent to the **AI Dashboard** for central expert visibility.
- Predictions/uncertainty scores are relayed to the **Active Learning & Oversight Trigger Module**.

The central active learning loop is then initiated, which identifies samples for expert review. While sample metadata is displayed on the **AI Dashboard** for broader context, the secure annotation task itself is managed by the **HITL Secure Gateway & Task Router**. This gateway assigns the task to a **PPC Central Data Scientist**. The expert receives the task and submits their feedback via the gateway's secure interface, subsequently enabling the active learning module to provide refined data for global model improvement by the **Aggregation Engine**.

Following this, and after an authorized approval decision by the central expert via the **AI Dashboard**, the **Model Selector & Rollback Logic** within the **Central Server** prepares the updated global model for distribution. Key decisions and model states are logged to the **Blockchain Logging/Audit System (POLITO)** for auditability before the Model Distributor sends the approved model back to the **PPC FL Client Instances**. These instances then update their local models, with the **PPC Local Executive** fine-tuning a personalized version and providing feedback that enriches the local data for future cycles using the local label propagation.

6.4.3.2 Scenario B (Local Expert Refinement of a Personalized Energy Prediction Model)

Figure 6.3 illustrates Scenario B, which details the client-side active learning process for refinement of the personalized energy prediction model within a **PPC FL Client Instance**. This scenario assumes a personalized model (LM_P1) is already in place and has generated predictions. The primary focus is on leveraging local expert knowledge to enhance both the model's local performance and the quality of the underlying data segment, which is used to improve the performance of the central aggregated model in further FL rounds.

The refinement cycle is initiated by the **PPC Local Executive**, who triggers a review of the personalized model's outputs. The **FL Compute/Training Module** then presents these predictions, highlighting instances that may be of low confidence [32], exhibit potential fairness issues or were flagged by the local executive's domain knowledge. Subsequently, the executive then

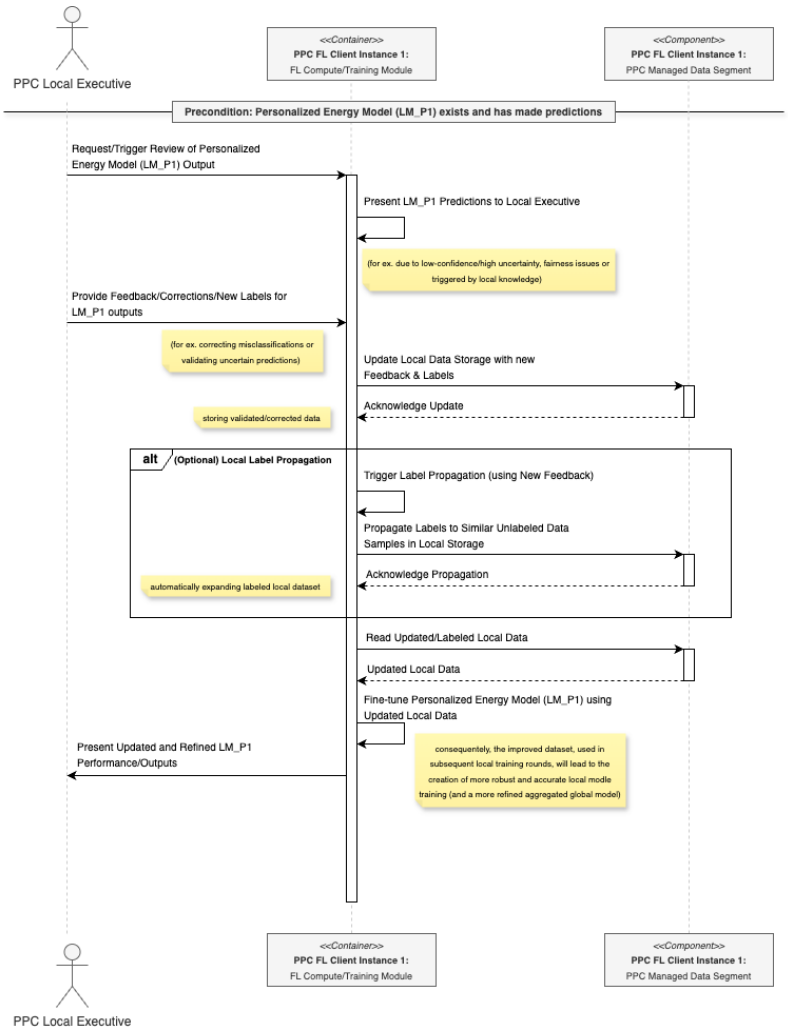

Figure 6.3: NOUS Use Case #2 Sequence Diagram - Local Expert Refinement of a Personalized Energy Prediction Model (Scenario B)

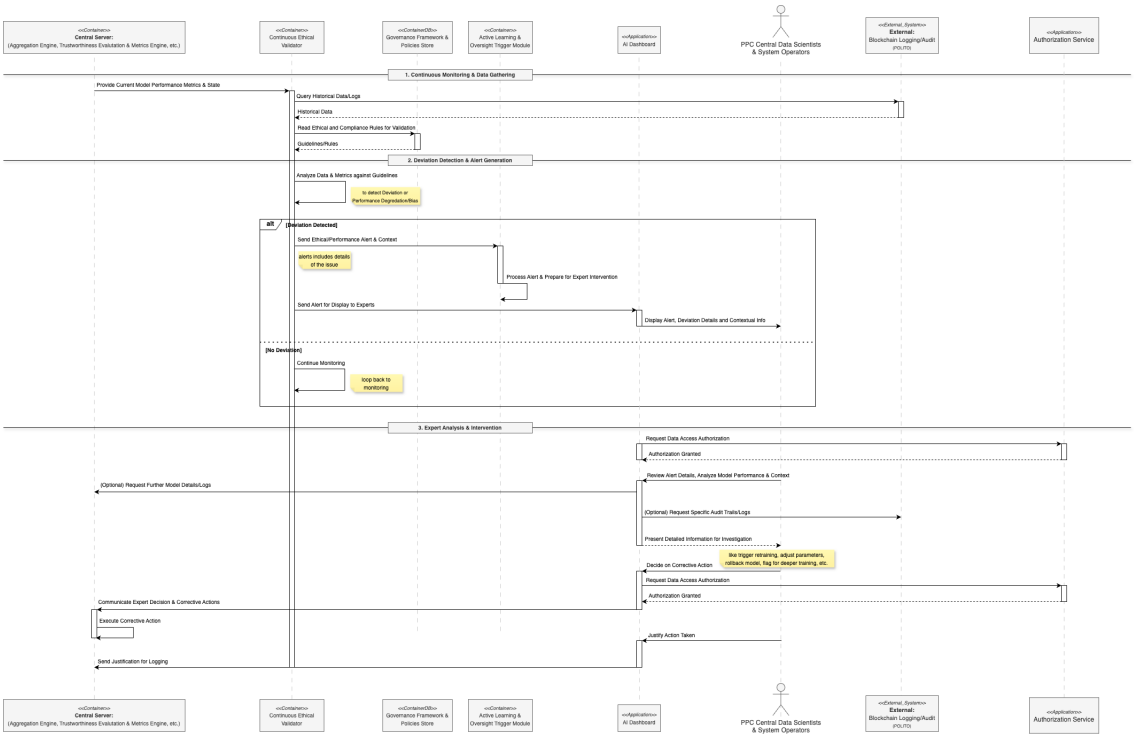

Figure 6.4: NOUS Use Case #2 Sequence Diagram - Ethical Validator Alert and Central Expert Intervention (Scenario C)

provides feedback, corrections or new labels for these outputs, which are then used to update the client instance's **PPC Managed Data Segment**, ensuring that the local data store reflects the most accurate and validated understanding.

Finally, this process utilizes the newly validated labels and the current personalized model to automatically expand the labeled dataset by propagating labels to similar, previously unlabeled samples stored locally. With this enriched and more accurate local dataset, the **FL Compute/-Training Module** improves the predictive performance of the PPC client instance and, by extension, contributes to higher-quality updates in subsequent global FL rounds. The performance of the newly refined LM_P1 can then be presented back to the local executive for further assessment.

6.4.3.3 Scenario C (Ethical Validator Alert and Central Expert Intervention)

Figure 6.4 depicts Scenario C, showcasing the proactive ethical and performance monitoring capabilities of the HITL-FL system within the use case in question. The **Continuous Ethical Validator** continually gathers data, including current model performance and state from the **Central Server**, historical data from the **Blockchain Logging/Audit System** and predefined ethical compliance rules from the **Governance Framework and Policies Store**. Upon analyzing this data, if a deviation from guidelines or significant performance degradation is detected, the validator issues an alert. This alert is simultaneously sent to the **Active Learning & Oversight Trigger Module** to

prepare for potential expert intervention and to the **AI Dashboard** for immediate display to the **PPC Central Data Scientists & System Operators**.

Once alerted, the **PPC Central Data Scientists & System Operators** utilize the **AI Dashboard** to review the deviation details and analyze the model's performance and context. During this investigation, they may optionally request further model details from the **Central Server** or specific audit trails from the blockchain, with these data access requests being subject to approval from the **Authorization Service**. After forming a judgment, the expert decides on a corrective action (such as triggering retraining, adjusting parameters or rolling back the model). However, before this action is executed, the **AI Dashboard** again consults the **Authorization Service** to confirm the expert's authority. If authorized, the decision and corrective actions are communicated to the **Central Server** for execution, and a justification for the action taken is sent for logging to the **Blockchain Logging/Audit System**, ensuring a transparent and auditable intervention process.

From these three scenarios, it is clear that the proposed HITL-FL architecture can be applied to this use case, thereby validating it. The mapping illustrates how the proposed architecture's structure and defined workflows fulfill the key functional and non-functional attributes by enabling HITL functionalities, such as targeted expert feedback, local model personalization, and proactive ethical oversight. This showcases the architecture's capacity to address the specific demands of the NOUS energy use case while adhering to its core design principles.

6.5 Expert Review through Qualitative Interviews

To complement the validation process and provide a better insight into the proposed HITL-FL architecture, a qualitative expert review process was coordinated, which involved conducting semi-structured interviews with other architects. The primary objective is to include external individuals not directly involved in the conceptualization of this specific HITL-FL architecture to obtain an unbiased perspective on the perceived strengths and weaknesses of the proposed system, as well as identify potential areas for future refinement that might not be evident from a purely self-assessment standpoint.

In the context of this thesis, two interviews were conducted, both with members of the NOUS project who didn't partake in the development of this architecture but were familiarized with the overall context in which this work resides.

6.5.1 Interview Protocol

To ensure a systematic approach to the qualitative expert review, a semi-structured interview protocol was developed, guided by the principles outlined in the ACM SIGSOFT Empirical Standards for qualitative research, particularly concerning the conduct of surveys and interviews [50].

The interview protocol was structured to cover several areas, allowing for a comprehensive discussion of the proposed HITL-FL architecture while also exploring emergent themes and in-depth insights from the interviewees. The main sections of the protocol included:

1. **Purpose of the Interview and Study Context** - outline the research objectives and the role of the interview in the validation process.
2. **Interviewee and Interviewer Information** - documenting interviewee selection rationale and interviewer details, including a reflection on potential biases and mitigation strategies.
3. **Ethical Considerations and Consent** - detail aspects such as voluntary participation, confidentiality, data anonymization, recording procedures and consent, data usage limitations, and the interviewee's right to review their transcript.
4. **Pre-Interview Preparation for the Interviewee** - list the architectural documents and diagrams provided to the interviewees in advance for them to analyze. This included a summary of the HITL-FL architecture, its C4 model diagrams, the defined Functional and Non-Functional Requisites, the identified Core HITL-FL Design Problems and the sequence diagrams for the NOUS Use Case #2 mapping.
5. **Interview Guide** - a set of open-ended questions organized thematically to explore overall impressions of the architecture, its effectiveness and practicality in integrating HITL mechanisms, its contributions to trustworthiness and ethical alignment, its ability to deal with privacy and security, the methods to ensure scalability and efficiency, its overall application within the NOUS use case and, finally, general feedback (perceived strengths, weaknesses, potential trade-offs and suggestions for future work or improvement).
6. **Concluding the Interview** - procedures for closing the interview.

The full, detailed interview protocol employed for this study is provided in Appendix A

6.5.2 Data Collection

The expert review data was collected with semi-structured interviews that lasted between 60 to 90 minutes. All interviews were audio recorded at the participants' request to capture their responses and perspectives accurately. Notes were also taken to highlight some of the shared themes and interesting points that developed during the interview.

Following each interview, the audio recordings were transcribed word for word to produce a detailed textual record. To ensure the fidelity and accuracy of this transcribed data, a member-checking process was employed, where each interviewee was provided with a transcript of their respective interview and allowed to review it for any inaccuracies, misinterpretations or to offer clarifications.

6.5.3 Key Findings

From the conducted expert interviews, various topics of discussion emerged, highlighting the overall strengths of the HITL-FL architecture and revealing gaps and opportunities for improvement. The main findings from these interviews are presented in Sections 6.5.3.1 to 6.5.3.6, each representing a specific theme.

6.5.3.1 Architectural Clarity/Readability

The overall legibility of the architectural diagrams was a key point of positive feedback throughout the interviews. They were described as *"very well designed"* and *"clear and easy to interpret"*, which reinforced their effectiveness in communicating the system's structure and responsibilities.

However, even though the diagrams were perceived as being solid and easy to understand, an effort to maintain visual consistency could enhance their comprehension, since there are instances where the same element shifts positions across different diagrams. According to one interviewee, *"keeping this visual consistency would help (...) maybe you could rearrange the lines a bit so that they are closer together and do not cross the diagram as much."* These small adjustments in visual coherence could improve the experience of navigating across the different architectural views.

6.5.3.2 Modularity & Maintainability

In both interviews, modularity was also recognized as one of the strengths of the presented framework, with a clear division of responsibilities among its various components/modules. As one interviewee noted, *"something you did that was very good was you actually modularized the components, and I appreciate that because in that sense, if one of them fails, then the other one may not fail,"* which underscores the clear separation of tasks and the subsequent reduction of severe consequences whenever a failure occurs.

From a maintainability approach, there is also no immediate red flags. As expressed by one of the individuals *"I haven't seen anything that specifically calls the maintainability of your current design into question."* The same interviewee added that having *"different components with different responsibilities"* positions the system to adapt well to future changes.

Regarding ways to potentially improve this modular structure, two methods stood out. One was following a microservices approach, where *"each microservice must have its own database."* The other method was creating a domain model of this system, as an exercise to identify possible containers or components that, if they *"don't have domain entities"*, could be merged with another module.

6.5.3.3 Blockchain Integration & Auditability

This was one of the topics where, even though both interviewees acknowledged the importance of introducing and maintaining an immutable log, they also expressed their concerns and skepticism concerning its practical implementation costs.

One interviewee reflected on their prior experience with blockchain, stating: *"I know it's good for auditability, but I'm wondering if it's a bit overkill (...) it was expensive, it consumes a lot of energy, added a lot of additional complexity."* The same individual also warned that *"the operations will certainly be slower"* when practically every transaction must be logged.

Another interviewee echoed this skepticism in a more direct way: *"I'm a bit skeptical on the blockchain component in there, I must confess. If an alternative could be digitally signing all the actions that are happening within a system and storing them somewhere (...) I see that as fulfilling*

the same goal (...) in a much cheaper way." The suggestion to substitute a full blockchain for lightweight cryptographic signatures highlights a potential trade-off between absolute immutability and operational efficiency.

Putting these concerns aside, the value of the blockchain was not fully dismissed, as one interviewee said "*Knowing that trust is such an important quality attribute (...) then it does make sense to use the blockchain, since architecture is the art of trade-offs."* Therefore, with this in mind, since one of the project's primary objectives is to have a demonstrable provenance, the added complexity can be justified (considering that the performance and energy costs are mitigated).

6.5.3.4 Logging Strategy & Data Preservation

While an immutable audit trail is central to accountability, the interviews made clear that reliability also depends on a more nuanced logging strategy. For example, one interviewee mentioned that a single point of failure can eliminate critical evidence (*"If one of them [components] fails and that one is not connected to the blockchain to log, you lose all the logs"*).

To avoid this risk, redundant local stores can be included at the container level (*"these components (...) should have their own database that they can access to dump the logs"*). In parallel, a shared facility for runtime diagnostics was also encouraged, as one interviewee stated: *"I worked in products where you have a central logging system where multiple apps logged into there, so that could still be useful to understand what happened and why it happened."*

6.5.3.5 Authentication & Authorization

Another critical gap identified during the interview was the lack of explicit authorization components in the design at the time of the interview. One interviewee admitted to being surprised by its absence (*"Another thing I found strange was that you didn't have an authentication or an authorization component (...). Authorization is connected to permissions and important for auditability"*). However, after the interview, taking into account the reasons provided by the individual, an additional component for performing authorization tasks (Authorization Service) was added to HITL-FL's architectural system, as showcased in Section 4.3.2.

Additionally, the same individual emphasized that, since the architecture already recognizes two types of external actors, these users should have distinct privileges, allowing actions to be attributed to each specific user. As a result, authentication should be the system's primary focus in order to determine who is authorized to do what and who did what. This latter concept, however, is already being introduced in the system, as expert decisions are already being logged with an identification that states the respective expert, and the HITL Secure Gateway handles authentication of the Human Experts.

6.5.3.6 Domain-Driven Design Principles

One interviewee suggested clearer component naming based on Domain-Driven Design (DDD) [63] principles, rather than generic infrastructure labels. As they reflected: *"I wondered a couple of*

times while watching your presentation of the architecture, whether we could design this architecture using principles of Domain-Driven Design, and perhaps if we did (...) it would be more visible within the architecture, the main domain concepts."

From this perspective, labels such as Central Server could appear too vague. The interviewee asked directly, *"Is it possible that there's a more descriptive name than 'central server'?"* Furthermore, the same individual added that making concepts like "global model," "local model" or data flowing between nodes much clearer would *"help one read the architecture and the diagrams to understand the system's"* intent and boundaries more quickly.

Overall, these findings emphasize both the strengths of the current architectural strategy and areas that require refinement to ensure practical effectiveness, robustness and clarity. Other themes were also discussed. In particular, the approval of the privacy and security techniques used in the system, as well as the well-thought-out application of the architecture to the NOUS' use case. However, the other topics described above occupied a bigger portion of the interviews and, therefore, deserved to be highlighted.

6.6 Discussion of Validation Results

The previous sections have detailed the validation approaches employed to assess the proposed HITL-FL reference architecture. This process involved a systematic analysis of its fulfillment of predefined architectural attributes (Section 6.3), a practical demonstration through its mapping to NOUS Use Case #2 (Section 6.4), and the collection of qualitative insights via expert interviews (Section 6.5).

This section focuses on providing a discussion of the architecture's overall strengths (in Section 6.6.1) and its inherent limitations and trade-offs (in Section 6.6.2).

6.6.1 Strengths

By performing the comprehensive validation process, several key strengths of the proposed HITL-FL reference architecture were identified.

6.6.1.1 Comprehensive HITL Integration

One of the main strengths of the proposed architecture, consistently highlighted throughout the validation process, is its comprehensive and systematic approach to integrating HITL capabilities. More specifically, it is how the architecture embeds human oversight at multiple levels. Centrally, the dedicated **Human Oversight & Interaction Hub** (Section 4.3.2 of Chapter 4), comprising key containers such as the AI Dashboard, the Active Learning & Oversight Trigger Module and the HITL Secure Gateway & Task Router, provides a platform for Human Expert interaction to influence the global model. Concurrently, the architecture also supports localized oversight within each **Client Environment** through mechanisms that enable End User Experts to provide direct feedback and label data, which enhances personalized models (Section 4.3.2.1 of Chapter 4).

Furthermore, the architecture's strength in this area is also reinforced. On the one hand, the explicit definition and operationalization of three distinct yet potentially interconnected HITL workflows (as detailed in Chapter 5): the Active Learning Loop, the Model Oversight & Governance Loop and the Ethical Validation Loop. On the other hand, the successful mapping of three scenarios in the second NOUS Use case, as explained in Section 6.4, demonstrates the architecture's ability to mediate a range of tasks, from target data annotation to continuous monitoring and analysis of ethical performance. Finally, the fulfillment of numerous functional requisites, such as FR.007 (Human Expert Oversight and Control), FR.008 (Active Learning Selection and Annotation Workflow) and FR.010 (Ethical Validation and Alerting) (Section 6.3.1), further attests to the architecture's robust support for a broad spectrum of HITL functionalities.

6.6.1.2 Modularity & Separation of Concerns

During the expert reviews, one of the main strengths highlighted by both interviewees was the effective modularization and clear separation of concerns throughout the independently functioning modules and components of the architecture. The feedback highlighed the practical benefit of modularization, particularly in reducing the impact of isolated component failures. On another note, it is also important to consider that, due to its modularity, the architecture is robust enough to handle future system updates, thereby enabling easier management, debugging and potential enhancements (throughout the architecture's lifecycle).

6.6.1.3 Focus in Ethical AI & Governance

Another distinct strength of the proposed reference architecture, identified during the validation process, is its dedicated focus on ethical AI and governance through specialized containers, such as the Continuous Ethical Validator and the Governance Framework & Policies Store. This strength was especially emphasized by one interviewee, who acknowledged its value and importance to the architecture (*"The ethical validator is a really important concern"*). The Governance Framework & Policies Store guarantees ethical assurance by effectively encapsulating policies and guidelines for ethical oversight. These components allow the architecture to align with broader human-centered AI principles. As the interviewees stated, the inclusion of dedicated components for providing ethical guidelines or benchmarks is vital for effectively operationalizing ethical oversight.

6.6.1.4 Robustness in Addressing Design Problems

Finally, the last strength that should be highlighted in this architecture is the solutions found to address all the Design Problems presented in Section 4.3.4, which were conceptualized during the architectural design. Each problem highlighted challenges that were fully addressed by the integration of specific components, containers and services, allowing for the effective integration of human oversight in FL frameworks. For instance, implementing an ethical validator or integrating

structured logging via blockchain directly addresses issues of trustworthiness and transparency. At the same time, microservices and modular designs enhance scalability and resilience.

These identified strengths distinguish this architecture, as they highlight its unique value and position it to effectively address the complex demands of integrating human oversight within FL frameworks.

6.6.2 Limitations & Trade-Offs

Although the validation process highlighted significant strengths of the HITL-FL architecture, it is also essential to acknowledge the various limitations and challenges inherent to the system, as well as the design trade-offs made. These are listed below:

6.6.2.1 Implementation Complexity, Overhead & External Systems Dependency

From the presented HITL-FL architecture in Chapter 4, it is evident that it is composed of numerous interconnected containers and components that are designed to support its various functionalities. However, as a consequence, it introduces additional complexity that hinders the deployment and maintenance of such a multi-component system in a production environment like NOUS, which could require significant effort and careful orchestration of its services.

Furthermore, the integration of specific containers or systems, like the Continuous Ethical Validator or the Blockchain Logging/Audit System, although they can provide substantial benefits in terms of trustworthiness and accountability, they could also introduce performance overhead, as to support the impact that these features can have on the overall system latency and resource consumption could lead to the necessity of carefully benchmarking and guaranteeing optimization in a full-scale deployment.

The blockchain, since it is an external system to the proposed HITL-FL framework, it is possible to affirm that it is not under the direct control of the HITL-FL system. Therefore, the performance, availability and reliability of this external blockchain component can significantly impact the proposed system's overall effectiveness and robustness. On top of that, this dependency on external systems also extends to the System Tradeoff Analysis Results, whose overall performance and responsibility can directly influence the quality of decision support provided to human experts.

6.6.2.2 Real-time Constraints/Latency

The integration of human feedback, especially for tasks like model validation or active learning, can introduce latency into the HITL-FL system, as noted in the SOTA (Section 3.3.1). Considering this, while the proposed architecture supports asynchronous HITL workflows (via MQTT communication for annotation tasks) to decouple human interaction from real-time model aggregation where possible, scenarios requiring synchronous expert approval (such as before distributing a critical model update) may impact the overall iteration speed of the FL process. This represents a trade-off between the rapidness of expert intervention and the output of the learning system.

6.6.2.3 Scalability of Human Expertise & Interaction

Finally, a fundamental challenge that should be underscored in any HITL system, including the proposed architecture, is the scalability of human expert involvement. While mechanisms like the Active Learning & Oversight Trigger Module and the internal expert task router of the HITL gateway are designed to mitigate this, the system's ability to scale effectively with an increasing number of clients, data requiring review or the complexity of expert tasks depends on the availability/capacity of human experts.

The expert interviews (Section 6.5) also raised this as a practical concern, suggesting that managing large teams of reviewers requires robust workflow management tools beyond what is explicitly detailed at the architectural level. Therefore, this limitation highlights a trade-off between the depth of human oversight desired and the practical constraints of human resources.

Overall, these limitations and trade-offs do not undermine the contributions of the proposed architecture, but rather provide a realistic assessment of the considerations for the future development and implementation of the HITL-FL system.

6.7 Threats to Validity

While the validation process used in this research was designed to be comprehensive, including diverse methods of validating, such as architectural requirement fulfillment analysis, use case mapping and qualitative expert reviews, it is also essential to acknowledge the various potential limitations that could put into question the validity of the findings and the consequential conclusions drawn. These threats are considered across several dimensions:

6.7.1 Construct Validity

Construct validity refers to how accurately the concepts used in this work represent the ideas they are meant to evaluate. In this research, construct validity threats can be expressed in the following ways:

- **Architectural Requirements Definition** - the defined functional and non-functional attributes (Section 4.2) were derived from the literature, the goals of HITL-FL integration and the specific context of NOUS. However, there is a possibility that these requisites might not fully capture all critical aspects or that their interpretation during the fulfillment analysis could vary.
- **Expert Interview Interpretation** - the analysis of qualitative data from the expert interviews depends especially on the researcher's interpretation. This, therefore, introduces a potential for bias since the conclusions derived from interview responses could vary based on subjective perspectives.

- **Conceptual Nature of the Artifact** - the primary artifact subject to analysis is a conceptual architecture that is validated analytically rather than empirically through real-world implementation. Therefore, the validation process assesses potential rather than proven operational performance.

6.7.2 Internal Validity

Internal validity typically concerns causal relationships. However, possible limitations related to the design and the rigor with which it was evaluated need to be mentioned:

- **Researcher Bias in Design** - since the researcher is the sole designer of the HITL-FL architecture, its assumptions may skew how the design's strengths are perceived, resulting in bias.
- **Consistency of Evaluation Criteria** - even though predefined evaluation criteria were established, the qualitative nature of specific assessments (such as the expert's evaluations of usability or adaptability) may introduce subjective interpretations.

6.7.3 External Validity

External validity is the degree to which the proposed architecture and its findings can be generalized beyond the specific context of this study. Relevant threats in this work include:

- **Single Use Case Mapping** - the architecture was only mapped to NOUS' second use case (Section 6.4), which focuses on one domain (energy). Therefore, its effectiveness in integration might vary, as additional mappings would enhance confidence in broader applicability.
- **Limited Sample Size for Expert Interviews** - the qualitative expert reviews included a small number of NOUS project architects. Considering this, even though their insights were contextually rich, this limited number may affect the generalizability of findings. Therefore, to achieve saturation, additional experts who focus on other related knowledge domains may provide alternative views.
- **Context Specificity of NOUS** - the proposed solution reflects a NOUS-specific technical infrastructure, goals and constraints. However, to adapt the system to other frameworks/-contexts, further unplanned adaptation may be necessary for deployment.

Although this research carefully validated the proposed architecture, the presented limitations underscore the importance of cautious interpretation, as future research will help confirm the generalizability and effectiveness of the proposed solutions.

In conclusion, this chapter evaluated the proposed HITL-FL architecture through a multi-faceted validation approach that combines architectural requisite fulfillment analysis, practical NOUS project mapping and qualitative expert interviews. From this process, it is possible to

affirm that the architecture addresses the previously defined design challenges since it supports human oversight mechanisms and aligns comprehensively with its defined FRs and NFRs. Additionally, key strengths such as its capacity for effective HITL integration and clear division of responsibilities were consistently highlighted, underscoring its relevance and potential. At the same time, however, issues such as the complications associated with implementing the system in a real-world context and aspects of the current validation itself were also highlighted, thereby laying the groundwork for future work, as discussed in Chapter 7.

Chapter 7

Conclusions & Future Work

7.1 Introduction

This chapter begins by revisiting the initial research objectives, RQs and hypothesis to assess their fulfillment. Subsequently, it also outlines several promising directions for future investigation and practical application, building upon the contributions and insights gained from this work.

The two main focuses of this chapter are detailed in their own dedicated sections:

- **Section 7.2: Objective Satisfaction** summarizes how the objectives and each RQ were addressed and comments on the hypothesis.
- **Section 7.3: Future Work** proposes concrete extensions and research tasks that can strengthen and broaden the value of the HITL-FL architecture.

7.2 Major Achievements

This thesis set out to design and validate a reference architecture that integrates HITL mechanisms with FL in edge environments without compromising its trustworthiness, performance, ethical compliance and privacy, using the NOUS project as a guiding context. First, table 7.1 provides a mapping from each RQ (Section 1) to the sections and artifacts that answered it.

Having demonstrated how each research question aligns with the artifacts of this work, the four core objectives (Section 1.6) have also been met by the end of this thesis:

- **Objective 1: (Reference Architecture Design)** was accomplished in Chapter 4, where the C4-based HITL-FL architecture was presented and its components and workflows were detailed.
- **Objective 2: (Requirement Elicitation)** is met in Chapter 6, Section 6.3, through the derivation of functional and non-functional requirements that guided the design of the architecture.

Table 7.1: Research question coverage

RQ	Where addressed / key evidence
RQ1	The key architectural components are detailed in Chapter 4 (like the System Context, Container and Component diagrams, Section 4.3), defining the structural elements. Chapter 5 is entirely dedicated to detailing and showcasing not only the main types of upstander interaction, but also the HITL mechanisms and feedback loops/workflows where human oversight is fully integrated into the proposed system.
RQ2	The best methods and standards are reflected in the Design Principles (Section 4.2) such as Human-Centricity, Transparency and Auditability. These are implemented through dedicated architectural elements of the HITL-FL system (Section 4.3.2.3). In the proposed architecture, containers and external systems like the Ethical Validator, the Blockchain Logging/Audit System and the various related policies and guidelines stored in the Governance Framework & Policies Store ensure that the architecture in question follows through with the non-functional requirements related to this domain (Section 6.3.2. Additionally, expert interviews confirmed the relevance of this approach and the methods to introduce it into the system.
RQ3	Data privacy and security are addressed as by-design attributes, guided by Design Principles like Privacy & Security (Section 4.2). Mechanisms include client data anonymization, encrypted communication (through the Secure Connection Gateway), secure annotation workflows (the HITL Secure Gateway & Task Router container) and immutable audit trails via the Blockchain Logging/Audit System. The fulfillment of NFRs related to performance (Section 6.3.2) details and further supports these aspects. Performance is considered through efficient HITL mechanisms (like active learning and semi-supervised learning) and the support for asynchronous workflows (Sections 5.3 and 5.4).

- **Objective 3: (Architectural Validation)** was realized in Chapter 6, via a multi-faceted validation process (compliance analysis, use-case mapping, expert reviews).
- **Objective 4: (Transferable Knowledge Contribution)** is delivered throughout Chapters 4, 5 and 6, where design principles, patterns and lessons learned for future federated-HITL systems are distilled as a reference for future implementations of the integration of HITL with FL systems.

Together, these mappings confirm that the thesis hypothesis is supported and that all planned objectives have been achieved. Through compliance analysis, use-case mapping and expert review, this work has established a modular, privacy-preserving architecture with built-in human oversight mechanisms, thereby demonstrating that such a design can underpin trustworthy, adaptable AI at the edge without compromising security, privacy or performance.

7.3 Future Work

While this work has presented many contributions to the design of FL architectures with HITL, several avenues for future work that can further deepen its validation, broaden its applicability and

enhance its practical utility are also important to explore, such as:

- **Broader Expert Engagement** - conduct additional interviews to achieve thematic saturation, especially with specialists in FL, HITL and other NOUS domains.
- **Architectural Refinements** - incorporate expert-suggested enhancements into the architecture, such as microservice decomposition with per-service logging, more explicit authentication modules and clearer domain-driven naming.
- **Alternative Audit Mechanisms** - evaluate lighter audit solutions, such as digitally signed append-only logs or certificate-based ledgers, as substitutes for full blockchain. Additionally, compare energy and performance costs to evaluate the possible improvement. This topic was also discussed during the expert interviews.
- **Full C4 Model of NOUS** - develop a complete C4 representation of the existing NOUS architecture to verify a better and easier-to-read mapping of the proposed architecture with NOUS'.
- **Cross-Domain Validation** - map and test the architecture in additional NOUS use cases and non-NOUS federated settings to confirm the architecture's adaptability in real-world scenarios.
- **Automated Governance Tooling** - expand the Continuous Ethical Validator into a configurable rules engine, enabling domain experts to author and update fairness or risk policies without code changes.
- **Prototype Implementation** - transition from a conceptual reference to a deployable prototype, by initiating the process of implementing the various components and modules that comprise the HITL-FL system for a prototype that can later be used for usability studies with target expert users, quantitative performance evaluation of HITL mechanisms, latency measurement and more.
- **Enhanced XAI Integration** - explore integrating more advanced XAI techniques directly into the HITL workflows to provide experts with deeper insights into why a model is uncertain or why an ethical flag was raised, therefore leading to more informed expert interventions.

Ultimately, this work presented a detailed architectural framework that bridged the gap between the advancements made in FL and the ever-growing necessity for including human oversight in AI. Through its design and validation, this work provided a tangible step towards realizing AI systems that are not only efficient and privacy-preserving but also accountable and that align with human values in complex ecosystems, such as the NOUS project.

References

[1] Sadi Alawadi et al. “A Federated Interactive Learning IoT-Based Health Monitoring Platform”. In: July 2021, pp. 235–246. ISBN: 978-3-030-85081-4. DOI: 10.1007/978-3-030-85082-1_21.

[2] Areen Alsaid and John Lee. “The DataScope: A Mixed-Initiative Architecture for Data Labeling”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 66 (Oct. 2022), pp. 1559–1563. DOI: 10.1177/1071181322661356.

[3] Borja Arroyo Galende, Juan Mata Naranjo, and Silvia Uribe Mayoral. “Scalable and Portable Federated Learning Simulation Engine”. In: eSAAM ’23. Ludwigsburg, Germany: Association for Computing Machinery, 2023, pp. 10–14. ISBN: 9798400708350. DOI: 10.1145/3624486.3624488. URL: https://doi.org/10.1145/3624486.3624488.

[4] Stephanie Kay Ashenden et al. “Chapter 2 - Introduction to artificial intelligence and machine learning”. In: *The Era of Artificial Intelligence, Machine Learning, and Data Science in the Pharmaceutical Industry*. Ed. by Stephanie Kay Ashenden. Academic Press, 2021, pp. 15–26. ISBN: 978-0-12-820045-2. DOI: https://doi.org/10.1016/B978-0-12-820045-2.00003-9. URL: https://www.sciencedirect.com/science/article/pii/B9780128200452000039.

[5] Len Bass, Paul Clements, and Rick Kazman. “Software Architecture In Practice”. In: Jan. 2003. ISBN: 978-0321154958.

[6] Qifang Bi et al. “What is Machine Learning? A Primer for the Epidemiologist”. In: *American Journal of Epidemiology* 188.12 (Oct. 2019), pp. 2222–2239. ISSN: 0002-9262. DOI: 10.1093/aje/kwz189. eprint: https://academic.oup.com/aje/article-pdf/188/12/2222/32614486/kwz189.pdf. URL: https://doi.org/10.1093/aje/kwz189.

[7] Antonio Boiano et al. “A Secure and Trustworthy Network Architecture for Federated Learning Healthcare Applications”. In: Oct. 2024, pp. 124–129. DOI: 10.1109/WiMob61911.2024.10770536.

[8] Jan vom Brocke, Alan Hevner, and Alexander Maedche. “Introduction to Design Science Research”. In: Sept. 2020, pp. 1–13. ISBN: 978-3-030-46780-7. DOI: 10.1007/978-3-030-46781-4_1.

[9] Simon Brown. *The C4 model for visualising software architecture. Context, Containers, Components, and Code*. 2021.

[10] Frank Buschmann and Kevlin Henney. “Pattern-oriented software architecture”. In: (Jan. 1993).

[11] 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.

[12] Jane Cleland-Huang et al. “The Twin Peaks of Requirements and Architecture”. In: *IEEE Software* 30.2 (2013), pp. 24–29. DOI: 10.1109/MS.2013.39.

[13] Stefan Cronholm and Hannes Göbel. “Design Principles for Human-Centred {AI}”. In: *30th European Conference on Information Systems - New Horizons in Digitally United Societies, {ECIS} 2022, Timisoara, Romania, June 18-24, 2022* (2022).

[14] Panagiotis Dimitrakis. *D2.1 Status quo, gap analysis, requirements and project evaluation framework | Pending Approval*. May 2005. DOI: 10.5281/zenodo.15393605. URL: https://doi.org/10.5281/zenodo.15393605.

[15] David Garlan and Mary Shaw. “An introduction to software architecture”. In: *Advances in software engineering and knowledge engineering*. World Scientific, 1993, pp. 1–39.

[16] *General Data Protection Regulation (GDPR) – Legal Text*. https://gdpr-info.eu/. Last Accessed: 2025-06-23.

[17] Neil Harrison and Paris Avgeriou. “Pattern-Driven Architectural Partitioning: Balancing Functional and Non-functional Requirements”. In: *2007 Second International Conference on Digital Telecommunications (ICDT’07)*. 2007, pp. 21–21. DOI: 10.1109/ICDT.2007.65.

[18] “IEEE Recommended Practice for Architectural Description for Software-Intensive Systems”. In: *IEEE Std 1471-2000* (2000), pp. 1–30. DOI: 10.1109/IEEESTD.2000.91944.

[19] Yunsang Joo et al. “Blockchain and Federated Learning Empowered Digital Twin for Effective Healthcare”. In: *Human-centric Computing and Information Sciences* 14 (Sept. 2024). DOI: 10.22967/HCIS.2024.14.051.

[20] Yan Kang et al. “Optimizing Privacy, Utility, and Efficiency in a Constrained Multi-Objective Federated Learning Framework”. In: *ACM Trans. Intell. Syst. Technol.* 15.6 (Dec. 2024). ISSN: 2157-6904. DOI: 10.1145/3701039. URL: https://doi.org/10.1145/3701039.

[21] Paul A. Kogut. “The Software Architecture Renaissance”. In: 2009. URL: https://api.semanticscholar.org/CorpusID:12126062.

[22] Jakub Konečný et al. *Federated Learning: Strategies for Improving Communication Efficiency*. 2017. arXiv: 1610.05492 [cs.LG]. URL: https://arxiv.org/abs/1610.05492.

[23] Jakub Konečný et al. *Federated Optimization: Distributed Machine Learning for On-Device Intelligence*. 2016. arXiv: 1610.02527 [cs.LG]. URL: https://arxiv.org/abs/1610.02527.

[24] Sushant Kumar et al. “Applications, Challenges, and Future Directions of Human-in-the-Loop Learning”. In: *IEEE Access* 12 (2024), pp. 75735–75760. DOI: 10.1109/ACCESS.2024.3401547.

[25] Hao-Ping (Hank) Lee et al. “Deepfakes, Phrenology, Surveillance, and More! A Taxonomy of AI Privacy Risks”. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI ’24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703300. DOI: 10.1145/3613904.3642116. URL: https://doi.org/10.1145/3613904.3642116.

[26] Pengfei Li et al. “Uncertainty Measured Active Client Selection for Federated Learning in Smart Grid”. In: *2022 IEEE International Conference on Smart Internet of Things (SmartIoT)*. 2022, pp. 148–153. DOI: 10.1109/SmartIoT55134.2022.00032.

[27] Tian Li et al. “Federated Optimization in Heterogeneous Networks”. In: *Proceedings of Machine Learning and Systems*. Ed. by I. Dhillon, D. Papailiopoulos, and V. Sze. Vol. 2. 2020, pp. 429–450. URL: https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf.

[28] Sin Kit Lo et al. “Toward Trustworthy AI: Blockchain-Based Architecture Design for Accountability and Fairness of Federated Learning Systems”. In: *IEEE Internet of Things Journal* 10.4 (2023), pp. 3276–3284. DOI: 10.1109/JIOT.2022.3144450.

[29] Qinghua Lu et al. “Responsible AI Pattern Catalogue: A Collection of Best Practices for AI Governance and Engineering”. In: 56.7 (Apr. 2024). ISSN: 0360-0300. DOI: 10.1145/3626234. URL: https://doi.org/10.1145/3626234.

[30] Qinghua Lu et al. “Towards a roadmap on software engineering for responsible AI”. In: CAIN ’22. Pittsburgh, Pennsylvania: Association for Computing Machinery, 2022, pp. 101–112. ISBN: 9781450392754. DOI: 10.1145/3522664.3528607. URL: https://doi.org/10.1145/3522664.3528607.

[31] H. Brendan McMahan et al. *Communication-Efficient Learning of Deep Networks from Decentralized Data*. 2023. arXiv: 1602.05629 [cs.LG]. URL: https://arxiv.org/abs/1602.05629.

[32] Viyom Mittal, Mohammad Jahanian, and K. K. Ramakrishnan. “FLARE: federated active learning assisted by naming for responding to emergencies”. In: *Proceedings of the 8th ACM Conference on Information-Centric Networking*. ICN ’21. Paris, France: Association for Computing Machinery, 2021, pp. 71–82. ISBN: 9781450384605. DOI: 10.1145/3460417.3482978. URL: https://doi.org/10.1145/3460417.3482978.

[33] 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. URL: https://doi.org/10.1145/3685651.3686660.

[34] Morteza Moradi, Mohammad Moradi, and Dario Calogero Guastella. “Experience Sharing and;Human-in-the-Loop Optimization for;Federated Robot Navigation Recommendation”. In: *Image Analysis and Processing - ICIAP 2023 Workshops: Udine, Italy, September 11–15, 2023, Proceedings, Part II*. Udine, Italy: Springer-Verlag, 2023, pp. 179–188. ISBN: 978-3-031-51025-0. DOI: 10.1007/978-3-031-51026-7_16. URL: https://doi.org/10.1007/978-3-031-51026-7_16.

[35] Fabio Morandín-Ahuerma. “?What is Artificial Intelligence?” In: *Int. J. Res. Publ. Rev* 3.12 (2022), pp. 1947–1951.

[36] D. Surendiran Muthukumar and P. Deepa Lakshmi. “A Systematic Analysis on Active Federated Learning Network in Healthcare Industry”. In: *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*. 2024, pp. 1341–1347. DOI: 10.1109/ICDICI62993.2024.10810760.

[37] Adrian Nilsson et al. “A Performance Evaluation of Federated Learning Algorithms”. In: DIDL ’18. Rennes, France: Association for Computing Machinery, 2018, pp. 1–8. ISBN: 9781450361194. DOI: 10.1145/3286490.3286559. URL: https://doi.org/10.1145/3286490.3286559.

[38] *NOUS - Empowering Europe's Data Future*. https://nous-project.eu/. Last Accessed: 2024-10-09. Sept. 2024.

[39] Abdul-Rasheed Ottun et al. "The SPATIAL Architecture: Design and Development Experiences from Gauging and Monitoring the AI Inference Capabilities of Modern Applications". In: July 2024. DOI: 10.1109/ICDCS60910.2024.00092.

[40] Riccardo Presotto, Gabriele Civitarese, and Claudio Bettini. "Semi-supervised and personalized federated activity recognition based on active learning and label propagation". In: *Personal Ubiquitous Comput.* 26.5 (June 2022), pp. 1281–1298. ISSN: 1617-4909. DOI: 10.1007/s00779-022-01688-8. URL: https://doi.org/10.1007/s00779-022-01688-8.

[41] *PRISMA 2020 statement - PRISMA statement*. https://www.prisma-statement.org/prisma-2020. Last Accessed: 2024-11-4.

[42] Maite Puerta-Beldarrain et al. "A Multifaceted Vision of the Human-AI Collaboration: A Comprehensive Review". In: *IEEE Access* 13 (2025), pp. 29375–29405. DOI: 10.1109/ACCESS.2025.3536095.

[43] Anushree Raj. "A Review on Machine Learning Algorithms". In: *International Journal for Research in Applied Science and Engineering Technology* 7 (June 2019), pp. 792–796. DOI: 10.22214/ijraset.2019.6138.

[44] Thomas Rausch and Schahram Dustdar. "Edge intelligence: The convergence of humans, things, and AI". In: *Proceedings - 2019 IEEE International Conference on Cloud Engineering, IC2E 2019*. Institute of Electrical and Electronics Engineers Inc., June 2019, pp. 86–96. ISBN: 9781728102184. DOI: 10.1109/IC2E.2019.00022.

[45] *Rayyan - Intelligent Systematic Review - Rayyan*. https://www.rayyan.ai/. Last Accessed: 2024-11-1.

[46] Dany Rimez, Axel Legay, and Benoît Macq. "Ensuring Data Security and Annotators Anonymity Through a Secure and Anonymous Multiparty Annotation System". In: *Novel and Intelligent Digital Systems: Proceedings of the 4th International Conference (NiDS 2024)*. Ed. by Phivos Mylonas, Dimitris Kardaras, and Jaime Caro. Cham: Springer Nature Switzerland, 2024, pp. 620–631. ISBN: 978-3-031-73344-4.

[47] Lavanya Shanmugam, Ravish Tillu, and Manish Tomar. "Federated Learning Architecture: Design, Implementation, and Challenges in Distributed AI Systems". In: *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)* 2 (2 Sept. 2023), pp. 371–384. DOI: 10.60087/jklst.vol2.n2.p384.

[48] Ben Shneiderman. "Bridging the Gap Between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-centered AI Systems". In: *ACM Trans. Interact. Intell. Syst.* 10.4 (Oct. 2020). ISSN: 2160-6455. DOI: 10.1145/3419764. URL: https://doi.org/10.1145/3419764.

[49] Ben Shneiderman. "Human-Centered Artificial Intelligence: Reliable, Safe I& Trustworthy". In: *International Journal of Human-Computer Interaction* 36 (6 2020). ISSN: 15327590. DOI: 10.1080/10447318.2020.1741118.

[50] *Standards | Empirical Standards*. https://www2.sigsoft.org/EmpiricalStandards/. Last Accessed: 2025-06-17.

[51] Andrei Tenescu et al. “Lung Nodule Segmentation Using Federated Active Learning”. In: PETRA ’23. Corfu, Greece: Association for Computing Machinery, 2023, pp. 17–21. ISBN: 9798400700699. DOI: 10.1145/3594806.3594850. URL: https://doi.org/10.1145/3594806.3594850.

[52] Dhoyazan Al-Turki et al. “Human-in-the-Loop Learning With LLMs for Efficient RASE Tagging in Building Compliance Regulations”. In: *IEEE Access* 12 (2024), pp. 185291–185306. DOI: 10.1109/ACCESS.2024.3512434.

[53] UNESCO. *Engineering for sustainable development: executive summary*. https://unesdoc.unesco.org/ark:/48223/pf0000375634.

[54] United Nations. *THE 17 GOALS | Sustainable Development*. https://sdgs.un.org/goals.

[55] Lachlan D. Urquhart, Glenn McGarry, and Andy Crabtree. “Legal Provocations for HCI in the Design and Development of Trustworthy Autonomous Systems”. In: *Nordic Human-Computer Interaction Conference*. NordiCHI ’22. Aarhus, Denmark: Association for Computing Machinery, 2022. ISBN: 9781450396998. DOI: 10.1145/3546155.3546690. URL: https://doi.org/10.1145/3546155.3546690.

[56] Usman Ahmad Usmani, Ari Happonen, and Junzo 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. DOI: 10.1109/HORA58378.2023.10156761.

[57] Sapdo Utomo et al. “Federated Trustworthy AI Architecture for Smart Cities”. In: *2022 IEEE International Smart Cities Conference (ISC2)*. 2022, pp. 1–7. DOI: 10.1109/ISC255366.2022.9922069.

[58] Mohammad Hadi Valipour et al. “A brief survey of software architecture concepts and service oriented architecture”. In: *Proceedings - 2009 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009*. 2009. DOI: 10.1109/ICCSIT.2009.5235004.

[59] Jiangtao Wang, Bin Guo, and Liming Chen. “Human-in-the-loop Machine Learning: A Macro-Micro Perspective”. In: *ArXiv* abs/2202.10564 (2022). URL: https://api.semanticscholar.org/CorpusID:247026110.

[60] Qiang Yang. “Toward Responsible AI: An Overview of Federated Learning for User-centered Privacy-preserving Computing”. In: 11.3–4 (Oct. 2021). ISSN: 2160-6455. DOI: 10.1145/3485875. URL: https://doi.org/10.1145/3485875.

[61] Qiang Yang et al. “Federated machine learning: Concept and applications”. In: *ACM Transactions on Intelligent Systems and Technology* 10 (2 2019). ISSN: 21576912. DOI: 10.1145/3298981.

[62] Zhanpeng Yang et al. “Trustworthy Federated Learning via Blockchain”. In: *IEEE Internet of Things Journal* 10.1 (2023), pp. 92–109. DOI: 10.1109/JIOT.2022.3201117.

[63] Chenxing Zhong et al. “Domain-Driven Design for Microservices: An Evidence-Based Investigation”. In: *IEEE Transactions on Software Engineering* 50.6 (2024), pp. 1425–1449. DOI: 10.1109/TSE.2024.3385835.

Appendix A

Annex

A.1 Interview Protocol

Thesis Title: Architectural Design for the Integration of Federated Learning Strategies: The NOUS Project Use case
Interviewer: António Oliveira Ferreira (Master's Student, FEUP)
Date: [Insert Date]
Interviewee: [Name & Role]
Location: [TO BE DEFINED]
Duration: Approximately 60-90 minutes

1. Purpose of the Interview and Study Context

This interview is part of a Master's dissertation in Informatics and Computing Engineering at FEUP, focusing on proposing a reference architectural framework for integrating Human-in-the-Loop (HITL) methodologies within Federated Learning (FL) systems, specifically in the context of the NOUS project. The objective of this interview is to gather feedback and insights from architects who did not participate in the elaboration of the architecture, so as to validate it and identify its strengths, potential limitations, and areas for improvement.

2. Interviewee & Interviewer Information

2.1 Interviewee Selection and Description

2.1.1 Selection Interviewees are selected based on their knowledge of architectural design, thereby ensuring that the feedback is provided by an individual who is informed and has a deep understanding of the strategic objectives.

2.1.2 Description

- **Role in the NOUS Project:**

- **Years of experience in software architecture:**
- **Specific responsibilities within NOUS:**

2.2 Interviewer Description and Reflection on Biases

2.2.1 Interviewer António Oliveira Ferreira, a Master's student in Informatics and Computing Engineering at FEUP, who is currently writing a thesis that focuses on Federated Learning Architectural Solutions for integrating Human-in-the-Loop approaches.

2.2.2 Possible Biases As the designer of the proposed architecture, it is essential to consider the natural tendency to view this design favorably. Therefore, to avoid this from happening, I commit to:

- Actively listening to all feedback without defensiveness.
- Asking for specific examples and justifications for opinions.
- Considering alternative explanations for findings during the analysis phase.
- Following the interview, analyzing the main issues, improvements, and constraints stated to improve the architecture.

3. Ethical Guidelines & Consent

- **Voluntary Participation:** Participation in this interview is voluntary and the interviewee can withdraw at any time, without any consequences.
- **Confidentiality and Anonymity:** The identity of the participants will be anonymized.
- **Recording:** This interview will be audio-recorded to ensure accuracy in transcribing responses. Notes will also be taken and the recording will be deleted after the thesis is completed and graded.
- **Data Usage:** The data collected will be used only for the purpose of this Master's thesis and other academic articles.
- **Right to Review:** The interviewee will have the opportunity to review the transcript of the interview to ensure accuracy.
- **Time Commitment:** The interview is expected to last approximately 60-90 minutes.

4. Pre-Interview Preparation for the Interviewee

The following materials will be provided to the interviewee prior to the interview:

- A summary of the proposed HITL-FL architecture.

- The C4 model diagrams (System Context, Container, and relevant Component diagrams).
- A list of the defined Functional and Non-Functional Requisites.
- A list of the identified HITL-FL Design Problems and their proposed solutions.
- The data flow diagram for NOUS Use Case #2 (Energy Prediction and Data Lifecycle Management) with and without HITL elements, as well as the main sequence diagrams illustrating the architecture's application to this use case.

5. Interview Guide

The questions below are designed to be open-ended to encourage comprehensive and nuanced responses.

5.1 Introduction and Warm-up (5-10 minutes)

1. Thank you for agreeing to this interview. To start, could you briefly describe your current role and main responsibilities within the NOUS project?
2. What is your experience with Federated Learning and Human-in-the-Loop systems in distributed or edge computing environments?

5.2 Overall Architecture Impressions (Focus on C4 Diagrams)

1. After reviewing the provided C4 diagrams (System Context, Container, and Component diagrams), what are your initial overall impressions of the proposed HITL-FL architecture? *Probes: Is the overall structure clear? Are there any parts that are confusing or unclear? Does it align with your understanding of NOUS's architectural philosophy?*
2. How well do you think the architecture's modularity and component organization support its stated goal of being adaptable and maintainable for future changes or new integrations?

5.3 Human-in-the-Loop (HITL) Integration (Focus on Design Problems 1, 3, 4)

1. The architecture proposes specific types of human intervention (e.g., Oversight, Validation, Feedback). How effectively do you believe these HITL mechanisms are integrated into the Federated Learning process? *Probes: Do the proposed "Human Oversight & Interaction Hub" components (AI Dashboard, Active Learning Module, Secure Gateway, Expert Task Router) make sense in practice?*
2. The goal is to focus expert attention efficiently. Do you think the active learning selection mechanism (identifying critical/informative instances for review) is a promising approach for the NOUS context? What potential challenges do you see in its implementation?

5.4 Trustworthiness, Ethical Alignment, and Accountability (Design Problems 2, 6)

1. Trustworthiness and ethical alignment are central to this architecture. How effectively do you believe mechanisms like the Continuous Ethical Validator and Blockchain Logging/Audit System contribute to these goals? *Probes: Do you see the blockchain's role as practical for recording model states, validations, and expert decisions in a NOUS-like environment? What are the implications for performance or overhead?*

2. Do you think the architecture provides sufficient transparency and explainability for human experts to understand AI decisions and behaviors?

5.5 Privacy and Security (Design Problem 5)

1. Given the sensitive nature of data in FL and NOUS, how confident are you that the proposed architecture adequately addresses data privacy and model confidentiality throughout the training and HITL processes? *Probes: Are the proposed mechanisms (encryption, anonymization and secure channels) sufficient, or are there additional considerations from a NOUS security perspective?*

5.6 Use Case Mapping

1. Reflecting on the application to NOUS Use Case #2 (Energy Prediction and Data Lifecycle Management), how practical and beneficial do you find the integration of HITL in this specific scenario? *Probes: Are there any specific steps or interactions in the sequence diagrams that you believe would be particularly challenging or beneficial in a real-world deployment?*

2. Considering NOUS's edge computing and distributed nature, how well do you believe this architecture would scale to a growing number of clients, data volumes, or model complexities? *Probes: Are there any potential bottlenecks in the proposed interactions or data flows that might limit its performance at scale?*

5.7 General Feedback and Future Work (10-15 minutes)

1. What do you see as the biggest overall strengths of this proposed HITL-FL architecture?

2. What are its biggest weaknesses, potential trade-offs, or areas where you think future research or development should focus?

3. Are there any crucial components, interactions, or considerations that you feel might be missing from this architecture, or that could be improved upon?

6. Concluding the Interview

- Thank the interviewee for their valuable time and insights.

- Reiterate the confidentiality and data usage policies, as well as the member checking of the transcript.
- Offer a summary of the findings to the interviewee after it is completed.

7. Post-Interview Process

- **Transcription:** The audio recording will be transcribed "verbatim."
- **Member Checking:** The transcript will be sent to the interviewee for review and correction to ensure accuracy and provide an opportunity for clarification.
- **Qualitative Data Analysis:** The validated transcripts will undergo analysis to:
  - Evaluate how the architecture fulfills or fails to fulfill the Functional Requisites, Non-Functional Requisites, Design Problems, and HITL mechanisms, among other aspects.
  - Identify new, emergent themes or insights that were not initially anticipated, ensuring the analysis is not solely constrained by the pre-defined categories.
- **Chain of Evidence:** Findings will be directly supported by verbatim quotations from the interviewees.
- **Discussion and Integration:** The findings will be integrated and discussed in the validation chapter of the thesis.