Designing an AI-Enhanced Interface for Cognitive Support in High-Stakes Interrogations

Jonathan Ahlström & André Roxhage

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY 2026

Designing an AI-Enhanced Interface for Cognitive Support in High-Stakes Interrogations

Jonathan Ahlström & André Roxhage

LUND
UNIVERSITY

Designing an AI-Enhanced Interface for Cognitive Support in High-Stakes Interrogations

Published by
Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Supervisor: Kirsten Rassmus-Gröhn,
kirsten.rassmus-grohn@design.lth.se
Co-supervisor: Mats Dahl, mats.dahl@psy.lu.se
Examiner: Jonas Borell, jonas.borell@design.lth.se

Abstract

Interrogation leaders in high-stakes investigative settings must convert long interview sessions into accurate written protocols under time pressure, stress, and cognitive bias. While AI has the potential to streamline this work, generic large-model assistants risk replacing one form of cognitive effort with another by inviting passive acceptance of generated output. To address this, we designed, developed, and evaluated FENRIR, a locally hosted AI product that supports post-interview documentation. We then assessed how well FENRIR fits the workflow of interrogation leaders at an authority that conducts such investigations. We followed a user-centered design process and developed FENRIR in cooperation with a domain expert and interrogation leaders from the studied organization. Participants agreed that transcript traceability of AI-based analysis was essential, and reported that AI-assisted analysis shifted their effort from generation to verification rather than removing effort altogether. Building on this, we argue for further work on so-called frictional design, interface patterns that make verification a required step of the workflow rather than one the user can skip, so that the human remains the final decision-maker, particularly in high-stakes investigative settings.

Keywords: Investigative interviewing, AI, Local LLM, Frictional design, Human-in-the-loop, Cognitive load, Interrogation protocol drafting

Sammanfattning

Förhörsledare i utredande verksamhet med höga krav måste omvandla långa förhör till korrekta skriftliga protokoll under tidspress, stress och kognitiva bias. AI kan effektivisera dessa processer. Generiska språkmodeller riskerar dock att ersätta en form av kognitiv belastning med en annan, genom att inbjuda till passivt accepterande av genererat innehåll. För att möta detta designade, utvecklade och utvärderade vi FENRIR, en produkt med lokalt körda AI-analyser som stödjer protokollskrivandet efter förhör. Utvärderingen undersökte hur väl produkten passar in i arbetsflödet hos förhörsledare vid en myndighet som genomför sådana utredningar. Arbetet följde en användarcentrerad designprocess, och produkten utvecklades i samarbete med en domänexpert och förhörsledare från den studerade organisationen. Deltagarna var eniga om att spårbarhet till transkriptet är avgörande när AI stödjer protokollskrivning vid utredande förhör. Deltagarna rapporterade också att en analys med stöd av AI förflyttade deras arbete från manuell analys till verifiering av genererat innehåll. Utifrån detta argumenterar vi för fortsatt forskning kring så kallad friktionsbaserad design, gränssnittsmönster som gör verifiering synlig, så att människan förblir den slutgiltiga beslutsfattaren, särskilt i utredande sammanhang med höga insatser.

Nyckelord: Investigative interviewing, AI, Local LLM, Frictional design, Human-in-the-loop, Cognitive load, Interrogation protocol drafting

Acknowledgements

A big thank you to our supervisor Mats Dahl for the many discussions, the encouragement, and the quick, thoughtful feedback whenever we needed it. We are also grateful to Daniel Grahn, whose technical discussions and ideas helped shape the architecture of FENRIR, and to Daniel Malmgren for his steady support throughout. To our academic supervisor, Kirsten Rassmus-Gröhn, thank you for the guidance through this thesis. We also owe a great deal to the interrogation leaders from the studied organization who gave their time to take part in meetings and shared valuable feedback. Finally, a heartfelt thank you to our friends, families, and partners for all the encouragement.

Contents

List of Acronyms

AI Artificial Intelligence

FENRIR Forensic Evaluation and Neutralization of Reasoning, Inference, and Response

GDPR General Data Protection Regulation

LLM Large Language Model

OODA Observe, Orient, Decide, Act

PIR Priority Intelligence Requirement

RAG Retrieval-Augmented Generation

RPD Recognition-Primed Decision

RQ Research Question

SAT Structured Analytic Techniques

SDG Sustainable Development Goal

SIR Specific Information Requirement

UCD User-Centered Design

UI User Interface

UX User Experience

1 Introduction

Every operational decision rests on the quality of the intelligence behind it. When reporting is slow or inaccurate, planning stalls and personnel face unnecessary risk. Yet the interrogators who turn raw interview material into actionable assessments work under severe cognitive constraints. High-stakes interviewing demands precise recall and resistance to bias, all from a mind already strained by stress and continuous multitasking. This thesis presents FENRIR, an AI-enhanced interface for post-interview analysis that surfaces extracted information and assists in drafting intelligence protocols. The interface deliberately keeps the human interrogator as the final decision-maker through purposeful interface friction. Addressing this problem requires reasoning across several domains at once, since the work spans a structured design process, a system architecture for offline AI support, the technical development of the prototype, and the psychological grounding that explains why such support is needed in the first place.

1.1 Background

This section situates the thesis in the intelligence workflow followed by an authority that conducts high-value interrogations and the shift toward information-gathering, rapport-based interviewing. It then summarizes the cognitive constraints and biases that shape post-interview analysis, clarifying why manual processing becomes a bottleneck. Next, it reviews how AI-based support and its associated risks have been discussed in relation to such analytic work, and considers what this implies for the interrogator's role. Finally, it presents the FENRIR system and identifies the research gap that motivates the questions addressed in this thesis.

1.1.1 Operational Context: Intelligence & Interrogation

In an authority that conducts high-value interrogations, the primary objective of intelligence work is to build a predictive understanding of the operational environment by systematically collecting and analyzing information. The purpose is to remove uncertainty so that decision-makers can move from reactive responses to proactive measures. Because the resulting assessments are time-sensitive, the velocity of this cycle is critical. Assessments must be delivered before the information becomes obsolete or the operational window closes. The process is driven by Priority Intelligence Requirements (PIRs). PIRs are specific critical questions formulated during the direction phase, and they guide what information must be collected to reduce operational uncertainty. The workflow is formally described as an iterative cycle of planning, collection, processing, and dissemination (see Figure 1.1). In practice, however, these phases overlap and run in parallel, so interrogation leaders must manage immediate tactical demands while maintaining longer-term strategies. This thesis is situated in the processing part of that workflow, specifically the post-interview work in which raw interview material is processed into balanced assessments and communicated through standardized intelligence protocols. These protocols structure extracted information to answer the initial PIRs, ensuring decision-makers receive actionable intelligence.

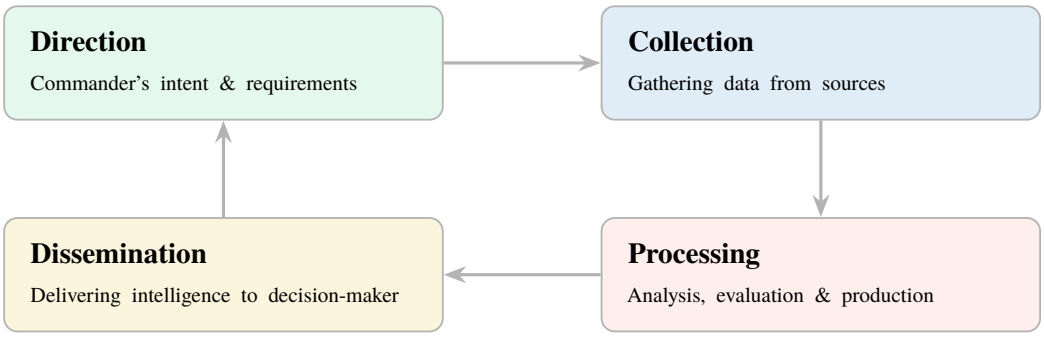

Figure 1.1: The Intelligence Cycle phases: Direction, Collection, Processing, and Dissemination.

Beneath this organizational cycle, the interrogator's own reasoning can be framed through the *Observe*, *Orient*, *Decide*, *Act* (OODA) loop originally developed by US Air Force Colonel John Boyd (Richards, 2020). Of its four

phases, Orient is the central one, since it is here that new information is integrated with previous experience, analytical synthesis, and cultural traditions to shape how evidence is interpreted (Richards, 2020). In the post-interview scope of this thesis, FENRIR therefore targets the interrogator's Orient phase over transcribed material, rather than the live dynamic between interrogator and source. This framing matters because Orient is also where confirmation bias exerts its strongest influence, what Richards (2020) calls "incestuous amplification", a point developed further in the following subsection where human biases will be discussed.

Furthermore, interrogation practice has moved away from accusatorial, confession-oriented approaches, often associated with mid-twentieth-century police models such as the Reid method and other pressure-based techniques, toward investigative interviewing that prioritizes open-ended questions and careful information elicitation (Oxburgh et al., 2023; Tribbels & Michels, 2025; Vrij et al., 2017). This shift is not only contemporary. Conclusions from World War II show that non-authoritarian questioning can gain more reliable and useful intelligence than oppression, while also reducing the risk of misinformation and forced confessions (Oxburgh et al., 2023; Vrij et al., 2017). Over time, critiques of confession-driven tactics, including evidence linking such practices to increased risk of false confessions, further reinforced the case for information-gathering approaches in both forensic and security-critical contexts (Gudjonsson, 2012; Tribbels & Michels, 2025). Modern frameworks such as ORBIT (Observing Rapport-Based Interpersonal Techniques) and PEACE (Planning and Preparation, Engage and Explain, Account, Closure, Evaluation) take this historical knowledge into account by emphasizing non-authoritative interaction, transparency, and structured reflection, with the aim of increasing information gathering and reducing risks such as false confessions.

1.1.2 The Human Problem: Cognitive Constraints

The transition toward information-gathering interrogation methods places significant demands on the human, as the quality of the resulting intelligence is fundamentally limited by the cognitive capacity of the interrogator. The interrogation environment requires the operator to manage multiple parallel tasks: active listening, the formulation of strategic follow-up questions, the assessment of credibility, and the simultaneous documentation of the interaction. This multitasking creates a severe bottleneck within the working memory, of-

ten leading to a reduction in the interrogator's ability to follow best practices (Hanway et al., 2021; Sweller, 2011). When cognitive resources are reduced, interrogators are more likely to fall back on intuitive but suboptimal behaviors, for instance interrupting the subject or inadvertently repeating questions, that risk contaminating the evidence.

Furthermore, the operational environments in which this kind of investigative interviewing often occurs are characterized by high physiological and psychological stress. Such conditions further degrade the accuracy of human memory and perception. Research by Artwohl (2002) indicates that acute stress can lead to significant perceptual distortions, including tunnel vision and auditory exclusion, which may cause critical details of an interview to be missed entirely. At a neurobiological level, extreme stress triggers the release of cortisol, which has been shown to impair hippocampal function and impair memory recall (Vrij et al., 2017). Consequently, stress affects the cognitive function of both the interrogator and the source. To mitigate these perception distortions, interrogators must aggregate and cross-reference data across multiple interviews. Reconstruction of a scenario from these fragmented and often contradictory sources places substantial demands on the interrogator's processing capacity.

Beyond memory constraints, the analysis of the interrogation material is affected by multiple cognitive biases. Human perception is not a neutral recording of reality but is filtered through pre-existing mental models. As Heuer (1999) notes, interrogators tend to "perceive what they expect to perceive". This fosters confirmation bias, where mindsets cause an observer to dismiss contradictory evidence or do not give it the proper weight (Primer, 2009). Similar to this is anchoring bias, where initial impressions influence and impact all subsequent interpretations, making it difficult to adjust to new information (Khamlichi & Benahmed, 2025). In the analytic workflow, these tendencies are compounded by common pitfalls such as underestimating the complexity of the task or being overly certain in one's own judgments, even when the underlying data is ambiguous (Belton & Dhami, 2020).

A potential method to mitigate these biases is the Structured Analytic Techniques (SAT), methods developed to externalize reasoning and counteract confirmation and anchoring effects (Heuer, 1999; Primer, 2009). Within this tradition, the Analysis of Competing Hypotheses is the most comprehensive technique endorsed by Western intelligence communities (Whitesmith, 2020). However, SAT's empirical efficacy is contested, and Whitesmith (2020) reviews a debate in which critics argue that analysts fall back on experience-based heuristics rather than formal techniques. These

methods aim at a broader quality criterion, analytic rigour, which Zelik et al. (2018) defines not as procedural compliance but as observable behaviours such as hypothesis exploration, information validation, stance analysis, and explanation critiquing. The design of the prototype does not take a stance for or against SAT, but instead operationalizes elements of the tradition by surfacing competing interpretations using AI.

Finally, the modality through which information is processed influences the cognitive load experienced during analysis. While the combination of audio and visual recordings provides the richest data, they also introduce significant sensory noise that can overwhelm an interrogator's processing capacity. Recent findings suggest that isolating verbal content (e.g., by audio-only analysis) can reduce this cognitive burden, allowing for a more focused evaluation of verbal content and improving the accuracy of assessments (Giorgianni et al., 2025). By addressing these human limitations through technological intervention, there is a potential to bridge the gap between human intuition and the rigorous standards required for high-stakes investigative analysis.

1.1.3 AI Support and Its Risks for Cognitive Offloading

Artificial intelligence has the potential to reduce the cognitive load in complex mental processes (Yan, 2025). In this thesis, AI support is not framed as a replacement for the interrogator but as a cognitive extension of the interrogator's mind (Hollan et al., 2000). Although externalizing these processes helps maintain attention and consistency, research suggests that excessive *cognitive offloading* can have negative effects on long-term memory and learning (Grinschgl & Neubauer, 2022). Yan (2025) further argue that to support complex tasks, AI must evolve from a passive tool into a "cognitive teammate." This concern aligns with research on Naturalistic Decision Making, where expert judgment depends on recognizing patterns and mentally simulating courses of action (Klein, 1999), a process that passive acceptance of AI output can short-circuit. Therefore, the system must be designed to guide attention without enforcing conclusions, ensuring that the human remains in the loop.

However, the same support that reduces cognitive burden also risks introducing automation bias, where decision makers disregard or fail to search for contradictory information, favoring a computer-generated solution that is accepted as correct (Cummings, 2017). When assistance is presented as reliable and paired with plausible explanations, users may shift from active interpreta-

tion to passive monitoring, which can reduce verification intensity and make errors harder to detect (Romeo & Conti, 2025). The design challenge is therefore not only to provide helpful suggestions, but to maintain the interrogator's agency and accountability throughout the workflow.

To address this, the interface must be designed for human-in-the-loop decision-making, where the interrogator remains the final decider and the system's role is limited. This study therefore uses friction to require active confirmation of flagged claims, making uncertainty visible and supporting alternative interpretations so that the interrogator is prompted to iteratively check rather than simply accepting AI generated content (Romeo & Conti, 2025).

1.1.4 FENRIR: Forensic Evaluation and Neutralization of Reasoning, Inference, and Response

FENRIR (Forensic Evaluation and Neutralization of Reasoning, Inference and Response) is being developed by Damalytics, a company founded by researchers from Lund University's Department of Psychology, in collaboration with the studied organization. Its central aim is to operationalize advanced Large Language Models (LLMs) and behavioral science methods to enhance information processing and support human decision-making during security and investigative interviews. In other words, the system is designed as a cognitive support layer that reduces error and bias under conditions of time pressure, stress, fatigue, and cognitive overload while maintaining accountability at all stages.

Phases of Support

FENRIR is designed to support the interview process across three phases: pre-interview planning, live execution, and post-interview evaluation. This thesis focuses exclusively on the third phase, in which AI-assisted analysis supports post-interview documentation, semantic analysis of interviewee responses, and structured assessment of interviewer performance.

Thesis Scope within FENRIR

FENRIR is designed to span the full interview lifecycle, from pre-interview planning through post-interview analysis. This thesis focuses exclusively on the post-interview phase, where transcribed material is processed, analyzed, and documented by the interrogator.

Development Roadmap

FENRIR is conceived as a modular system developed incrementally, starting with a limited set of robust functions before adding analytical complexity as empirical validation and operational testing progresses. The initial development focus is on the analysis of recorded interviews and the formatting of basic reports, with later phases moving toward more advanced interaction analysis and, eventually, real-time decision support. In this thesis, we align with that trajectory by focusing on text-based transcript processing and interrogator-facing review, rather than real-time support during live interviews.

Security Considerations

Data security is a foundational design principle of FENRIR. Mechanisms such as Federated Learning, a distributed approach that trains models without centralizing raw data, enable model improvement without exposing raw interview data across environments (J. Liu et al., 2022). The technical implementation of these security mechanisms falls outside the scope of this thesis.

1.1.5 Research Gap

FENRIR has the potential to provide the secure architectural foundation necessary to modernize intelligence workflows. However, how to design the interaction between humans and AI is yet to be explored for the specific purpose of intelligence analysis. Existing related research advances two adjacent directions. First, agentic systems have been proposed to conduct interrogations themselves. For example, Chkroun and Azaria (2024) introduce an Autonomous Interrogator and Deception Identifier that uses LLMs to run a text-based interrogation in a game setting, while also highlighting that computer-led interrogation is controversial and raises concerns about empathy, ethics, and trust. Subsequently, digitization efforts target the manual workload that surrounds interrogation documentation. Garcia et al. (2025) present INTU-AI,

a decision-support system that automates transcription, summarization, and report generation and operates offline, while acknowledging data availability and validation constraints.

Furthermore, although text-based analysis can reduce sensory noise and improve accuracy (Giorgianni et al., 2025), AI-driven support layers introduce distinct operational risks. There is a critical gap in understanding how to design interfaces that provide meaningful cognitive support, such as bias-mitigating flags, while simultaneously avoiding the automation bias risk described in Section 1.1 (Romeo & Conti, 2025), and thereby preserving attributes of analytic rigour such as hypothesis exploration and explanation critiquing (Zelik et al., 2018). Existing frameworks often treat AI as a passive tool rather than a "distributed cognition" partner or socio-cognitive teammate that requires active human-in-the-loop engagement through intentional interface friction (Grinschgl & Neubauer, 2022; Yan, 2025). This thesis addresses these limitations by designing and evaluating a system architecture that employs *frictional design* to maintain verification intensity within a secure, high-stakes investigative workflow.

1.2 Purpose

The purpose of this thesis is to design and evaluate a product tailored to support the analysis and drafting of transcribed interrogation protocols within an authority that conducts high-value interrogations. To ensure operational security and data integrity, the research explores a system architecture for offline deployment, using locally hosted LLMs. By integrating AI-driven support layers for semantic extraction, structured information processing, and analytical summaries, the work aims to reduce the cognitive load of the analysis process and improve decision making, protocol quality, and time efficiency. This study investigates how such a system can reduce the risk of human error and cognitive bias while improving operational efficiency. Furthermore, the work explores the interaction between the human interrogator and the AI system, with a specific focus on mitigating automation bias and ensuring that the interrogator remains active and in control through purposeful interface design. Finally, this research provides a scientifically grounded framework to support future research and methodological refinement by both the studied organization and the Department of Psychology.

1.2.1 Research Questions

This thesis addresses a primary research question, supported by multiple secondary questions:

1. How can an AI-supported analytic workflow be designed to reliably extract and structure relevant information from transcribed interrogations into protocols, while preserving human oversight and operational validity?
    (a) For which analytic tasks can FENRIR assist in order to reduce the interrogator's work load?
    (b) How should AI-generated output be surfaced in the user interface so that the interrogator remains the accountable decision-maker?
    (c) How do interrogators assess the workflow and operational usefulness of AI-extracted information when integrated into the process?

1.3 Scope & Limitations

The scope of this thesis is limited to the processing phase of the intelligence cycle, specifically focusing on the analysis of text-based transcripts. Although a project team member contributed an offline audio transcription capability that was integrated into the evaluated system, this thesis focuses on the analysis pipeline and does not evaluate transcription accuracy. The broader FENRIR system utilizes Federated Learning and adheres to security standards such as ISO/IEC 27001 to maintain data sovereignty. However, the technical implementation of these architectures, such as network isolation and encrypted model aggregation, is outside the scope of this project. Instead, this thesis simulates a secure, standalone offline environment populated by local LLMs. This allows the research to focus specifically on validating the interface's functionalities and cognitive support. Due to the sensitivity of the studied domain, the details of the underlying operations, including any real transcripts, remain classified. Real interrogation transcripts can only be accessed on designated classified workstations, and they therefore never entered the development environment used for this thesis. The authors instead generated a set of fabricated transcripts using AI, which were then refined manually to support development, testing, and evaluation throughout the project. Multiple fabricated

transcripts were produced in this way at different stages, ranging from early prototyping to the prompt calibration described later in this thesis. Their construction drew on the authors' prior practical experience in security-critical investigative settings to keep the material realistic, while preserving properties relevant for the interrogator's task, including operational detail and a length and narrative complexity comparable to a real interrogation protocol. Furthermore, the tool is specifically tailored for users in a single language community within a high-stakes investigative context, which means that linguistic nuances or reporting standards from other domains or languages are not addressed.

1.4 Sustainability Goals

Within these boundaries, the project also considers its broader societal responsibilities. The United Nations Sustainable Development Goals (SDGs) (United Nations, n.d.) represent a global framework of 17 objectives within Agenda 2030 designed to enable a sustainable future. This thesis aligns its design and process with three specific goals: SDG 9 (Industry, Innovation and Infrastructure), SDG 10 (Reduced Inequalities), and SDG 16 (Peace, Justice and Strong Institutions). In alignment with SDG 9, transparency is prioritized by documenting the system architecture, underlying assumptions, and technical limitations. In addition, the performance of FENRIR is evaluated so that the results can be reviewed. Regarding SDG 10, the design aims to support accessible interaction and reflects on potential exclusion resulting from interface choices or model behavior, while explicitly delimiting our scope to users in a single language community. Furthermore, the tool is designed to address inequality by aiming to mitigate cognitive pitfalls such as anchoring bias, where initial impressions might otherwise distort the analysis. By prompting alternative interpretations, the system is designed to promote a fairer assessment process independent of the user's initial assumptions. Finally, in line with SDG 16, the product is intended to strengthen trust and institutional safeguards. We aim to transparently communicate the capabilities and uncertainties of the system. Recognizing that high stress and cognitive load negatively affect human memory and information retrieval, the tool is designed to serve as a support mechanism that could increase the diagnostic value of the evidence gathered. Also, by excluding classified data, clear data governance protocols for processing, storage, and informed consent are followed. Validating whether these design intentions translate into measurable outcomes remains

a long-term goal beyond the scope of this thesis.

1.5 Ethical Considerations

The research design and conduct of this thesis were guided by the four fundamental principles of good research practice: reliability, honesty, respect, and accountability (Vetenskapsrådet, 2024). These principles serve as a framework for addressing the potential ethical challenges of developing AI support for sensitive, high-stakes investigative applications.

1.5.1 Participant Protection & Interrogator Autonomy

In terms of methodological ethics, the study prioritizes the principle of respect. Informed consent is important for all activities. Domain experts and usability participants are provided with clear information on the purpose of the project and their right to withdraw at any stage without consequence. The research aims to ensure that the tool's design does not lead to over-reliance or a reduction in the interrogator's autonomy. By transparently communicating both the capabilities and the uncertainties of the system, the project seeks to create a relationship where the human remains the accountable decision-maker.

1.5.2 Data Sovereignty, Integrity & Accessibility

The technical architecture reflects the principles of accountability and reliability, particularly concerning data governance. Regarding the high sensitivity of the project's context, it targets an offline deployment architecture using locally hosted models. This ensures data sovereignty and eliminates the risks associated with transmitting potentially classified information to external cloud providers. By keeping data processing entirely local, the system supports compliance with strict security requirements. Finally, regarding accessibility and fairness, as a pilot study focused on a specific group of interrogation leaders, the scope for accessibility implementation is limited. Future iterations will need to address these aspects more to ensure that the tool serves a diverse user base effectively.

1.6 Thesis Outline

The remainder of this thesis follows the structure of the design process it documents. Chapter 2 establishes the theoretical background, covering large language models and their failure modes, naturalistic decision making, and the design methodology that frames the work. Chapter 3 describes the design process, reporting the method and the result of each phase from initial field research through requirements engineering to iterative prototyping. Chapter 4 presents the resulting FENRIR system, detailing the offline architecture, the analysis pipeline, and the interface that supports post-interview documentation. Chapter 5 reports the evaluation, in which interrogation leaders and adjacent operational specialists assessed the prototype through a guided walkthrough. Chapter 6 interprets the findings, answers the research questions, reflects on the methodology, and outlines directions for future work. Chapter 7 concludes the thesis.

2 Theoretical Background

This chapter outlines the theoretical foundations of the thesis, bridging the technical architecture of LLMs with the user-centered design methodology used to develop the interrogator interface.

2.1 Large Language Models

This section establishes the technical context that constrains and enables the system developed in this thesis. It summarizes the relevant capabilities and failure modes of LLMs, and introduces interaction patterns, such as prompt design, agentic workflows, and Retrieval Augmented Generation, that make AI assistance more controllable and auditable in analysis tasks. The goal is to clarify the relevant principles that guide the system architecture.

2.1.1 Large Language Models

LLMs represent a fundamental shift in natural language processing, moving from task-specific modeling to generalized probabilistic generation. Unlike traditional classifiers, LLMs are trained on large volumes of text to predict sequential data, enabling them to generate coherent text across diverse domains without task-specific training (Minaee et al., 2024). These models approximate semantic understanding by statistically predicting the next word fragment in a sequence.

Generative Capabilities

In the context of intelligence workflows, LLMs facilitate the transition from rigid data processing to flexible information synthesis. Y. Zhang et al. (2025) demonstrate that these architectures excel at abstract summarizing, capable

of condensing complex narratives into core insights rather than simply extracting sentences. Furthermore, Z. Zhang et al. (2025) highlight their ability for generative information extraction, where unstructured interview data are parsed into structured entities and relationships. This capability allows for the automated transformation of raw text into analyzable formats, supporting intelligence tasks.

Reliability Limitations

Despite their utility, the integration of LLMs is constrained by inherent reliability issues. The most significant challenge is hallucination, where the models generate plausible but incorrect output. Huang et al. (2025) distinguish between intrinsic and extrinsic hallucinations. Intrinsic contradict source inputs and extrinsic hallucinations generate details that are unverifiable. In the context of FENRIR, an intrinsic hallucination might attribute a statement to the wrong speaker in an interrogation transcript, while an extrinsic hallucination might fabricate a location or operational detail that was never mentioned in the source material. Another limitation is the potential inconsistency of the output that could risk reducing the forensic validity. Y. Li et al. (2025) demonstrate that LLMs frequently can be inconsistent in their judgments while remaining confident. This instability complicates the reproducibility required for intelligence documentation.

Finally, processing extensive interrogation protocols requires managing context window limitations. While modern architectures support long input sequences, retrieval accuracy often becomes weaker in the middle of extended documents, a phenomenon known as ”lost in the middle” (N. F. Liu et al., 2024). To produce each word, the model relies on attention, an internal scoring step that decides which parts of the prior text matter most. T. Li et al. (2024) further argue that efficient long-context learning remains a bottleneck, since this scoring grows less reliable over longer interactions and performance degrades as token counts increase.

2.1.2 Prompt Engineering & Agentic Workflows

Although LLMs provide raw semantic processing capabilities, their integration into reliable systems requires structured guidance. This section examines how Prompt Engineering optimizes model instructions and how Agentic Workflows extend these interactions into autonomous, goal-oriented pro-

cesses.

Prompt Engineering

An LLM responds to whatever text it receives as input, which means the wording, ordering, and framing of that input largely determines the quality, reliability, and format of the response (Maaz et al., 2025). A prompt is that text, and prompt engineering is the practice of crafting it so the model reliably performs the intended task (Maaz et al., 2025). The unit the model actually processes is a token, a word fragment produced by tokenization that the model generates one at a time (Brown et al., 2020; Minaee et al., 2024). The number of tokens a model can handle at once is limited by its context window, which makes concise and well-structured prompts practically important (N. F. Liu et al., 2024).

Prompt Engineering refers to the design of instructions that guide the output of an LLM to help with a specific task, audience, and set of constraints (Maaz et al., 2025). Three techniques recur across the literature and in the FENRIR implementation. The system prompt sets the overall behavioral rules for the conversation, defining role, tone, output format, and constraints before any user input arrives. Few-Shot Prompting uses examples to illustrate the desired format and quality of the response, rather than describing them in the abstract (Brown et al., 2020). Chain-of-Thought prompting is a related technique in which intermediate reasoning steps are elicited to improve performance on multi-step reasoning tasks (Wei et al., 2022). In this project, Prompt Engineering is defined as the optimization of an instruction to an LLM.

Agentic Workflows & Retrieval-Augmented Generation

A single well-designed prompt can handle a specific task, but analytical work on an interrogation transcript involves several such tasks chained together. Agentic workflows extend single prompt-responses to a goal-oriented process in which an LLM iteratively plans, invokes tools, and refines structured outputs (Yan, 2025). This distinguishes an agent, a system that acts within a bounded environment, from a conventional chatbot that primarily generates conversational responses.

One common mechanism in agentic pipelines is Retrieval-Augmented Generation (RAG), where a retrieval component first selects relevant passages from a larger dataset and the generator then conditions its output on that retrieved evidence (Lewis et al., 2020). Conceptually, RAG implements

a pattern that supports tasks by grounding responses in explicit source material, which is particularly valuable when working with transcripts where traceability and selective evidence use are central to the analysis (Lewis et al., 2020).

These underlying failure modes collectively reframe the design challenge. Where the technical limitations of hallucination, output inconsistency, and degraded recall over long contexts undermine the reliability of any raw model output, the interface wrapping those capabilities must compensate through deliberate structural choices rather than assumed user vigilance. Consequently, the design problem extends beyond feature selection to building for trust, legibility, and active verification, placing human-centered methodology at the center of the development process.

2.2 Naturalistic Decision Making

This section establishes how experienced interrogators actually decide under the conditions typical of operational intelligence work such as time pressure, incomplete data, and consequential stakes. Naturalistic Decision Making is the research tradition that studies such decisions in the field rather than in a laboratory, and the Recognition-Primed Decision (RPD) model is its framework (Klein, 1999). He developed RPD through field studies of firefighters, naval commanders, and intensive-care nurses rather than of intelligence analysts, the underlying conditions that shape those decisions are the same ones that characterize operational intelligence work.

In the RPD model, an expert recognizes a situation as typical of a familiar pattern, and that recognition simultaneously activates a plausible course of action. Rather than generating and comparing alternatives, the expert mentally simulates whether the default action will work and iterates only when the simulation flags a problem (Klein, 1999). Pattern recognition and mental simulation are thus the two cognitive capabilities that expertise develops in such operational settings. RPD describes the interrogator's internal cognitive process, whereas distributed cognition, introduced in Chapter 1, frames the interrogator and FENRIR together as a wider cognitive system.

This distinction carries a direct design consequence. Klein (1999) warns that using purely analytical frameworks on experts risks "paralysis by analysis", in which rigid rational procedures decrease experiential judgment. FENRIR's frictional design aims to support RPD by protecting the recognition-and-simulation loop, surfacing uncertainty and competing

interpretations rather than replacing expert judgment with forced analytical comparison.

2.3 Design Methodology

This section establishes the methodological framework that structures the design and development of the system. It motivates the use of the Double Diamond process and User-Centered Design (UCD) principles to ensure that technical solutions are grounded in verified user needs rather than assumed requirements. The goal is to clarify the iterative process of Discovery, Definition, Development, and Delivery.

2.3.1 Double Diamond and User-Centered Design

The design process of this thesis follows the Double Diamond framework, developed by the Design Council. It maps the divergent and convergent stages of the design process into four phases: Discover, Define, Develop, and Deliver (see Figure 2.1) (Design Council, 2024). The Discover phase focuses on research to understand the underlying problem space without relying on initial assumptions. This is followed by the Define phase, where gathered insights are synthesized to frame the specific design challenge. In the Develop phase, the process enters a second divergent stage to explore various potential concepts and solutions. Finally, the Deliver phase converges on testing and refining these solutions to ensure that they effectively address the defined problem.

Central to this workflow is the principle of UCD, which emphasizes an iterative process optimized for the end-user's needs and context of use (Sharp et al., 2019). To operationalize these goals, we follow the definitions provided by ISO 9241-11, which describes usability as the "outcome of use" in terms of effectiveness, efficiency, and satisfaction in a specified context ("Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts", 2018).

2.3.2 Design Activities

This thesis utilizes various methods from Interaction design tailored to the project's goals. The theory below details these chosen activities, structured

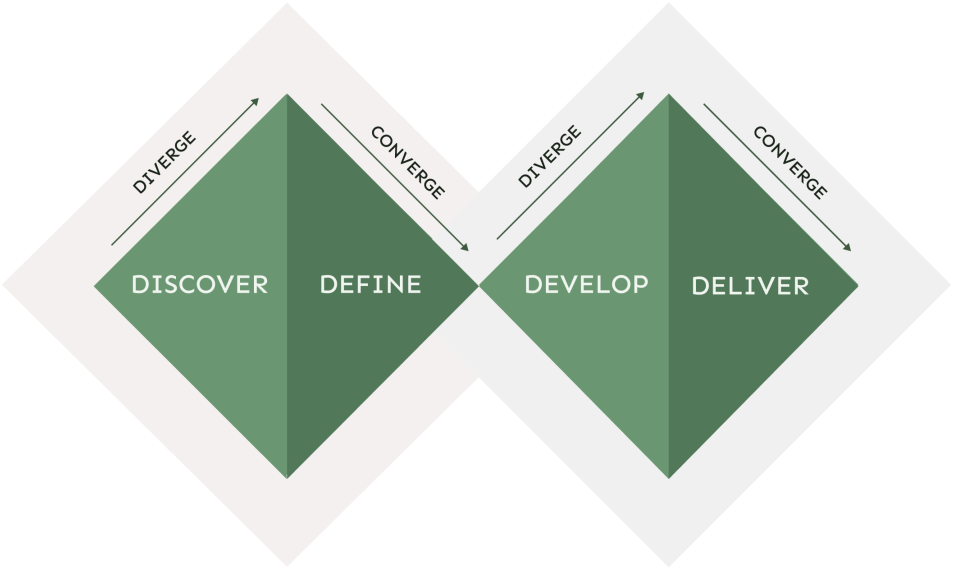

Figure 2.1: The Double Diamond Design Process: Discover, Define, Develop, and Deliver.

within the Double Diamond framework. By integrating User-Centered Design principles into each phase, the process ensures that the final system is driven by empirical user needs rather than assumptions.

Discover: Context & User Needs

The first divergent phase focuses on broadening the understanding of the problem space. Data gathering is the foundational step to ground later development in the actual needs and context of end-users. In addition to reading previous research from the area, semi-structured interviews can also be used. Such interviews offer a balance between consistency and flexibility. Using a set of questions while allowing open-ended exploration, this can reveal user goals, environmental constraints, and underlying motivations that might be missed with stricter protocols (Sharp et al., 2019).

Define: Requirements & Constraints

Following the initial exploration, the collected qualitative data is processed to extract meaningful insights. Thematic analysis is a foundational method for identifying, analyzing, and reporting themes within the data, offering a systematic way to organize and describe the dataset in detail (Braun & Clarke, 2006). These themes directly inform the definition of the scope and goals of users and the system. Based on these insights, Requirements Engineering translates user needs into system specifications. This process involves defining both functional requirements, which describe what the system should do, and non-functional requirements, which specify the constraints and quality attributes, such as user experience and performance, under which the system must operate. To ensure that all user needs are addressed and linked to system features, techniques within Requirement Engineering are used, such as prioritization and task-goal mapping. One widely adopted prioritization technique is MoSCoW analysis, which classifies requirements into four tiers: Must have, Should have, Could have, and Won't have. This structured method allows design teams to balance user expectations against project constraints without losing sight of lower-priority features entirely. The feature prioritization is presented in Appendix A. This traceability ensures that every design decision is justified by a specific stakeholder need, effectively narrowing the focus to high-value interventions (Lauesen, 2002).

Develop: Prototyping & Co-Design

In the second divergent phase, the focus shifts to exploring potential solutions through iterative cycles. Prototyping allows for the exploration and validation of requirements of an interactive model. Low-fidelity prototyping, such as paper prototyping, is a rapid and cost-effective method for verifying early design concepts and workflows without the distraction of visual details (Snyder, 2003).

A complementary activity within this phase is co-design, in which domain experts participate as co-creators rather than as passive evaluators. Sanders and Stappers (2008) define co-design as collective creativity applied across the whole span of a design process, repositioning users as experts of their own experience whose situated knowledge cannot be accessed through observation alone. In this role, the researcher shifts from translator to facilitator, providing structure that enables domain knowledge to surface and directly shape the evolving design.

As the design matures, high-fidelity prototypes are developed to test specific interactions, visual feedback, or complex user flows (Sharp et al., 2019). Recent advances in generative AI have enabled ”prompt-to-design” workflows, in which designers describe interface components in natural language and receive interactive high-fidelity output within seconds. Tools following this pattern (e.g., Figma Make, Lovable, Claude Design) substantially compress the iteration cycle, allowing more design alternatives to be tested within a fixed project timeline (Figma, 2025).

Deliver: Evaluation & Validation

The final convergent phase tests the solutions to ensure that they meet the defined user needs. Usability evaluations are conducted to strictly validate the system against the initial requirements. Formative evaluation occurs iteratively throughout the design process to identify and fix usability issues as they arise (Sharp et al., 2019). In contrast, a summative evaluation is typically performed at the end of the development cycle to validate the overall effectiveness, efficiency, and satisfaction of the final product against the initial research questions (“Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts”, 2018; Rubin & Chisnell, 2008).

In addition to usability metrics, the Deliver phase also validates the system’s broader utility within its operational context. While verification ensures that the system is built correctly according to specifications, validation ensures that the right system is built by addressing actual user needs (Sharp et al., 2019). Consequently, this phase serves as the final check where the design is tested against the reality of the users’ workflow.

Together, these two bodies of theory reveal both the opportunity and the obligation that shape this project. LLMs offer genuine capability for extraction and synthesis, yet their tendency toward hallucination and inconsistency means that raw model output cannot be trusted as a finished analytic product. The Double Diamond process and UCD principles provide the framework for designing within those constraints, ensuring that every interface decision is traceable to a verified user need. Taken together, these foundations motivate the core design principle of this work. The system must create deliberate friction that keeps the analyst actively engaged in verification, so that AI support enhances rather than displaces human judgment.

3 Design Process

Building on the theoretical foundations established earlier, this chapter traces the design process that moved the project from an open problem statement to a working prototype. The narrative follows the first three phases of the Double Diamond, Discover, Define, and Develop. Empirical findings in each phase led to the next design decision and triggered multiple design iterations.

The chapter includes both the method and the result within each phase and their design activities to explain that causal chain. Discover maps the operational reality of interrogation leaders conducting investigative interviews and surfaces the cognitive bottlenecks that motivate later design decisions. Define summarizes those findings into requirements and guiding design principles. Develop translates the principles into concrete interaction patterns through parallel design and implementation work. The result that emerges from this process is described in the next chapter, "The Resulting System", and its evaluation follows in Chapter 5.

3.1 Phase 1: Discover

The design process began with a divergent exploration of the problem space, aimed at understanding the operational realities of high-stakes investigative analysis without relying on technical assumptions. This phase focused on mapping the current workflows, constraints, and cognitive challenges faced by interrogators to ground subsequent design decisions in empirical needs.

3.1.1 Domain Research

To establish a foundational understanding of the domain, a broad contextual analysis was conducted from academic literature, expert consultations, and artifact review.

Contextual Analysis & Expert Consultation

Initial research focused on multiple domains: investigative interviewing, cognitive psychology, and human-AI collaboration. This theoretical grounding was refined through continuous dialogue with Mats Dahl, the project lead for FENRIR at Lund University. These discussions clarified the operational boundaries, specifically the role of PIRs as the central mechanism for directing collection. Furthermore, the technical constraints of the FENRIR architecture, specifically the requirement for offline and local LLM deployment, were identified as important requirements.

Artifact Review

To ground the design in current output standards, the authors reviewed five interrogation protocols from previous training exercises in the studied organization together with their corresponding audio and video recordings. The review revealed how protocols vary in form and detail. The analysis revealed that naming conventions for persons and locations were generally consistent and abbreviation usage was largely standardized, yet formatting and level of detail varied considerably between interrogators. Some protocols had long descriptions of the interrogation session, while others condensed them into a more summarized versions. Also, a few reports contained only sparse information relative to what the recordings actually revealed. These patterns suggested that protocol quality depended more on individual writing habits than on a shared reporting standard. Due to the classified nature of the material, specific protocol content cannot be disclosed. Overall, the review reinforced a finding that emerged across the contextual analysis: the core challenge was not transcription speed alone, but the cognitive load of simultaneously managing a social interaction, recalling specific details, and transforming raw observations into a coherent written narrative.

3.1.2 Interview with an Interrogation Leader

To ground the system design in a realistic context, a semi-structured interview was conducted with an experienced interrogation leader with a background in investigative interviewing. The respondent had a forensic background in both police work and high-stakes investigative settings, including training in tactical questioning and interrogation management. The session lasted approximately 35 minutes and was conducted by the authors. One led the primary

questioning while the other focused on asking follow-up questions to explore emergent topics in depth.

The audio of the interview was recorded and later transcribed using KlangAI, a GDPR-compliant service. The resulting text was then analyzed using structured qualitative coding, where the material was organized to identify recurring patterns in the workflow description. The primary objective was to capture objective descriptions of the workflow rather than subjective interpretations.

Findings from the Interview

To move from individual quotes to higher-level insights, the coded statements were thematically grouped using an affinity diagram. As shown in Figure 3.1, this process identified six areas, Cognitive Limitations, Subjectivity and Bias, Protocol Creation Bottleneck, Distinction between Data and Analysis, Fragmented Intelligence Flow, and the interrogator's own Feature Ideas for future tools.

The analysis reveals that the intelligence workflow is constrained by human working memory. The respondent explicitly identified a capacity limit of approximately five to six active PIRs. During a session, the interrogator creates a mental filter based on these priorities, and information that falls outside this scope is often discarded immediately or forgotten within minutes of leaving the room. This ”tunnel vision” could be explained as a necessary coping mechanism for high-pressure interaction, but results in the loss of intelligence that might be valuable to the broader operational picture.

Another interesting topic was the clash between the raw interaction and the final intelligence report. The respondent described a ”performance drive”, an internal pressure to deliver high-quality intelligence, which can unconsciously lead to the smoothing of ambiguities. Once a report is written, it becomes the functional truth for the analysis cell. Furthermore, because interrogators rarely receive feedback on the accuracy of their reports due to security reasons, they lack a mechanism to calibrate their own reports.

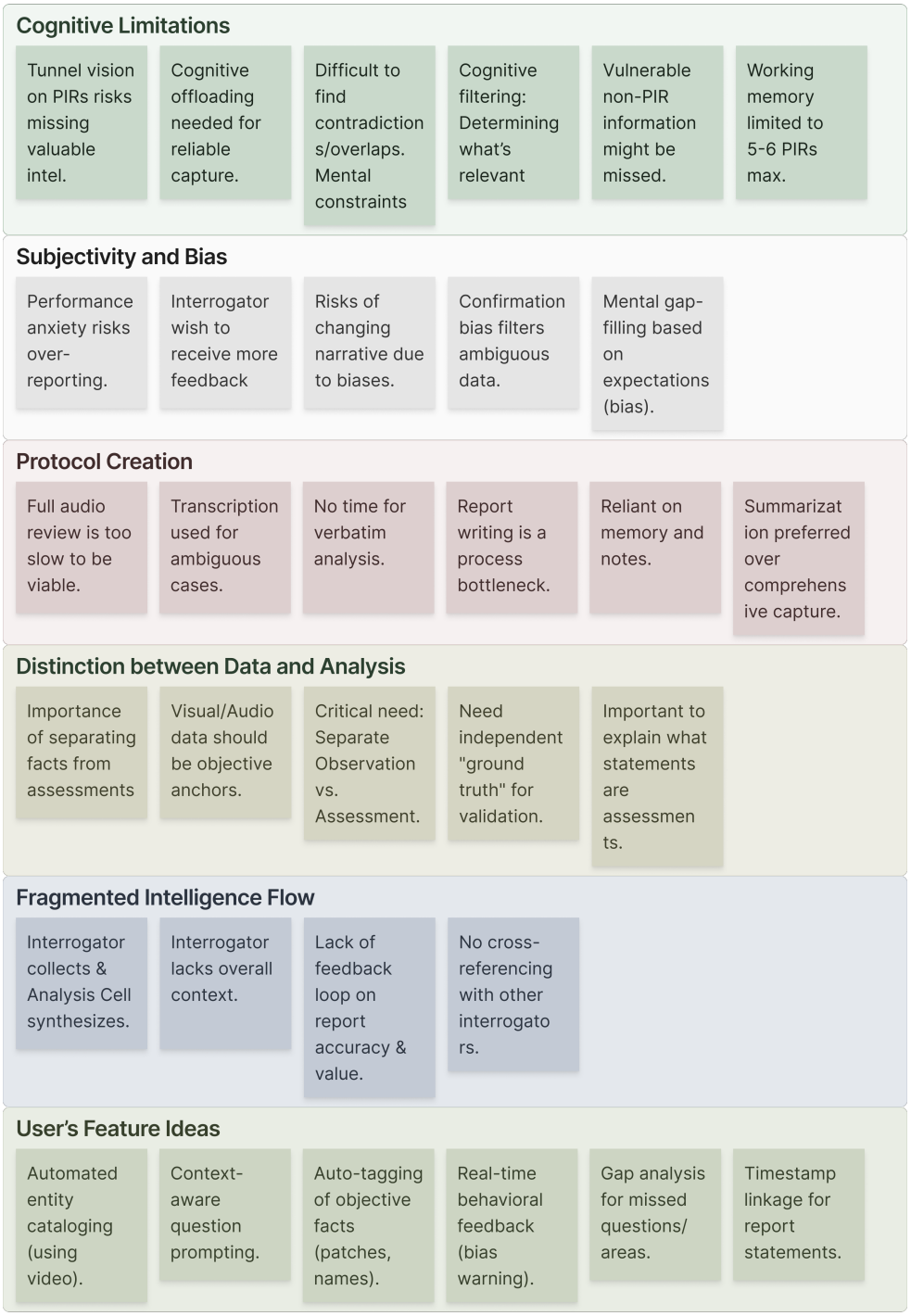

Figure 3.1: Affinity diagram showcasing key themes from the discover-interview.

Current Workflow

The interview mapped the standard interrogation process into five distinct phases, highlighting where cognitive load peaks:

1. **Preparation:** The interrogator has approximately five to six PIRs. These serve as the primary mental focus during the session.

2. **Execution:** During the interrogation, the operator relies on pen and paper to write notes while simultaneously managing the relationship and strategy. The cognitive capacity is almost entirely dedicated to maintaining the conversation and tracking the PIRs.

3. **Immediate Reporting:** Directly after leaving the room, urgent intelligence is verbally communicated to the Chief Interrogation Leader. This phase is also when the interrogator applies a mental "filter" that discards information felt to be irrelevant to the PIRs, freeing up working memory.

4. **Documentation:** Formal writing involves writing protocols, a *person form* (information about the person) and an *interrogation report* (intelligence data). This phase relies often on memory and the notes taken, as re-listening to the full audio is considered time-consuming.

5. **Submission:** The protocols are sent to the analysis cell, effectively finalizing the interrogator's interpretation of events. Ready for new tasks or new PIRs to conduct additional interrogations.

Design Implications

These findings were triangulated against the artifact review and the continuous expert dialogue with Mats Dahl, whose research experience from previous training exercises in the studied organization provided an independent perspective on the operational constraints. The combined insights directly informed the requirements for FENRIR. The system must not only speed up documentation, but act as an objective true layer that exists independently of the interrogator's memory. Specifically, the interface must support the process by offloading parts of the tasks. For example, by automatically tracking low-level details, such as timeline, names, units, and equipment, the human operator can focus on more demanding tasks such as answering the PIRs or

finding additional intelligence. In addition, the system must bridge the gap between the raw transcript and the formal report by helping the user verify the conclusions against specific evidence. This process mitigates the risk of human factors while increasing the nuance of the report beyond standard PIRs.

3.2 Phase 2: Define

The Define phase translated insights from Discover into concrete goals, features, and requirements for the system being built. The goal was to formalize what the system should achieve, which constraints were important in the operational context, and how each proposed feature could be traced back to a user goal. This step was essential because LLM-based systems can easily appear capable while still producing output that is difficult to verify in practice (see Chapter 2). Consequently, the phase focused on user goals, requirements engineering, and an initial conceptual workflow that would guide later prototyping.

3.2.1 User Goals

The user goals were defined using insights from the interview and previous research in the area. This led to three identified main stakeholder. The interrogation leader as the primary user, intelligence analysts as the secondary users who consume the protocols, and the studied organization as the tertiary organizational stakeholder.

Primary Users: Interrogators

The interrogators are responsible for transforming the interview into *interrogation reports* under strict time constraints. Consequently, the primary value of the system is to help the interrogator verify and structure claims before disseminating them.

- Support in managing large amounts of information without forgetting details or missing inconsistencies.
- Active assistance in identifying potential cognitive biases (e.g., confirmation bias) during the analysis phase.

- Drastic reduction in the time required to convert a raw transcript into a structured, formal protocol (*person form & interrogation report*).
- Tools to identify vague statements that require follow-up (control questions) to ensure the interrogation meets legal and operational standards.

Secondary Users: Analysts

Intelligence analysts in the studied organization consume the submitted protocols to make assessments across sources and continue the intelligence cycle. The role of FENRIR is to improve protocol quality and intelligence findings, and therefore reduce the need for analysts to second-guess what was said and why a claim was made.

- Receive high-quality structured protocols that require minimal follow-up clarification.
- Trust that the protocol content accurately reflects the spoken dialogue without distortion.
- Extract key intelligence entities efficiently, without unnecessary reliance on raw transcripts.

Tertiary Users: The Studied Organization

At the organizational level, the studied organization is the system owner. It defines the operational boundaries, the security measures, and the accountability requirements under which interrogation documentation is produced and disseminated.

- Deploy a tool that integrates securely with existing workflows and increases throughput in interrogation units.
- Ensure that documentation practices hold up under scrutiny by reducing human error and supporting ethically grounded, human-centric automation.

3.2.2 The System's Layers of Support

The three *layers of support* were introduced as an abstract model of the parts of the analysis process. The model's framing emphasizes gradual understanding

from what is explicitly present in the material to what it may imply in a broader context. Importantly, the layers are not separate stages with strict boundaries, see Figure 3.2. They are overlapping perspectives used iteratively. Narrative conclusions depend on meaningful interpretation, and meaningful interpretation depends on reliable semantic anchors. In practice, interrogators move back and forth between these perspectives while refining their understanding.

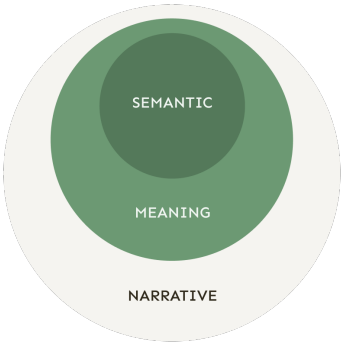

Figure 3.2: Venn diagram of the layers of support

Semantic Layer: Establishing a Factual Foundation

The *semantic layer* focuses on what can be treated as directly observable in the transcript material, such as people, places, events, and time references. The purpose is to establish a factual foundation before the interpretation begins. By making these anchors explicit, the interrogator can keep track of details that are otherwise easy to miss during or after an interview.

Meaning Layer: Interpreting with Caution

The *meaning layer* focuses on how statements should be understood, not only what was said. Here, the interrogator examines ambiguity, tone, implicit claims, and possible bias in both the source material and the analysis. The purpose is to support nuanced conclusions that remain traceable to semantic building blocks while also accounting for what is interpreted between the lines.

Narrative Layer: Building the Operational Picture

The *narrative layer* focuses on assembling the full context using all the material. This includes identifying consistencies, contradictions, information gaps, and potential follow-up questions related to the PIRs. Consequently, the layer model functions as an analytical structure for judgment: semantic anchors preserve traceability, meaning checks protect interpretive quality, and narrative explains the broader picture.

3.2.3 Requirements Engineering

To translate the findings and user goals into implementable specifications, requirements engineering principles were applied to distinguish between what the system must do (functional requirements), what information it must handle (data requirements), and what quality constraints the system must hold (non-functional requirements) (Lauesen, 2002). After a long brainstorming session and iterative revisits to our requirements specifications, the result was a list of requirements on the system. The resulting requirements did not consider the time constraint and other resources of this project and used a wishful way of thinking. The requirements were grouped into four categories.

System Functional Requirements Specify the core workflows the system must execute end to end, from transcript ingestion to protocol export.

Feature Functional Requirements Detail the user-facing capabilities that operate on top of those workflows, such as the editor, analysis layers, and traceability features.

Data Requirements Define how session data, AI outputs, and traceability metadata must be stored and linked throughout an analysis session.

Non-Functional Requirements Define quality constraints on usability, learnability, reliability, and ethical use, ensuring a cohesive and trustworthy experience.

System Functional Requirements

To ensure operational viability, the initial set of requirements defined the core capabilities necessary for the tool to function within the intended workflow. These included the ability to import and parse transcript formats, ensuring

that dialogue was correctly attributed to speakers. An important requirement was the integration with an LLM for extraction of evidence and summarization, together with clear state indicators to manage user expectations during processing. To support data hygiene and reporting, the system was required to allow session clearing and provide an export function. Safety mechanisms were also brainstormed, including hallucination disclaimers, manual override capabilities for all AI outputs, and version control options such as undo and redo functionality to prevent accidental data loss.

Feature Functional Requirements

The capabilities provided to the user were detailed in three categories. The first category, the *protocol editor*, was required to provide a text environment that automatically suggests AI-generated text segments while maintaining a split-view to allow reading the raw transcript. The second category covered the analysis layers, which focused on semantic and interpretive analysis through entity extraction such as underlining weapons or locations from the script, identification of leading questions from the interrogator, and detection of vague statements. The third category required the system to distinguish between observed facts and assessments. Finally, *traceability* was included for visual linking, ensuring that every claim in the protocol could be verified against the raw transcript.

Data Requirements & Non-Functional Requirements

Integrity and handling of session data were captured as data requirements to ensure that the user could iteratively work with the material without losing context. Specifically, the system must temporarily store the generated protocol draft, together with the traceability metadata needed to preserve an evidence trail during a session, such as evidence links and processing metadata. Non-functional requirements were defined to translate the user goals presented above, particularly *cognitive offloading*, verification, and *bias mitigation*, by constraining how the interface communicates, guides, and help decision-making. Usability requirements focused on reducing cognitive load through controlled and layered interface, supporting non-linear navigation between transcript and protocol writing, and maintaining user forgiveness through reversible actions. Learnability was planned to be achieved by progressively display explanatory help to use the interface. Reliability requirements addressed trust in the final protocol by enforcing a clear separation

between AI-generated verbatim quotes and summarized text, and by ensuring that extracted content was presented in a way that could be verified against the source. Finally, the primary ethical constraint formalizes a human-in-the-loop approach, the system ensures human agency by requiring explicit confirmation of all AI suggestions before they are committed to the protocol.

3.3 Phase 3: Develop

3.3.1 Conceptual Design

The conceptual design activity translated the requirements into an operational workflow that could be implemented and evaluated. At this stage, the main challenge was not identifying additional functions, but structuring interaction so the interrogator could maintain analytical focus under time pressure. Consequently, the design work centered on a non-linear process with iterative steps: read and/or listen to the transcript, analyze using active PIRs, highlight interesting parts, write the protocol, and export for dissemination. This process model then became the decision framework for which features were prioritized.

3.3.2 Feature Prioritization

Based on this workflow, potential features were mapped to the requirements and prioritized using the MoSCoW method (Must, Should, Could, Won't). The prioritization reflects both project constraints and the practical sequence of interrogator work. Before the prioritization was finalized, the resulting requirement set was reviewed with Mats Dahl, the project lead for FENRIR at Lund University, to confirm that the priorities aligned with operational needs. The complete feature-to-requirement mapping is provided in Appendix A. The following subsections summarize the motivation behind each priority tier.

Must-Have Features

The Must-Have features focused on establishing a reliable, efficient workspace. To satisfy the goal of *cognitive offloading* and the requirement for *data reliability*, the system included a *text slicer* to automatically structure raw transcripts, and an *audit log* to ensure every AI action was recorded for

auditability purposes. To meet the critical goal of *efficiency*, an *entity extractor* was prioritized to rapidly identify key intelligence such as names, times and locations. To maintain *trust*, these features were bound by *traceability* requirements, ensuring every extracted entity was linked to its source text.

Should-Have Features

The Should-Have features addressed the more complex goals of *quality assurance* and *bias mitigation*. These included a *bias & quality layer* to satisfy requirements for identifying vague statements and potential leading questions. While essential for the system's long-term value as a "neutral observer," these were prioritized second to the foundational data handling features.

Could-Have Features

Advanced narrative features, such as timeline visualizations or multi-perspective summaries, were classified as Could-Have. While they mapped to the goal of "understanding the bigger picture," they required a level of semantic reliability that could only be guaranteed after the Must and Should layers were fully operational.

Finally, one requirement was treated as a special methodological case. For the purpose of evaluating LLM capability against human-written protocols, the system also needed the ability to generate a fully autonomous draft protocol. This function was not meant to replace interrogator review in operational use; instead, it served as a controlled baseline for comparison in the evaluation phase.

3.3.3 Low- and Mid-Fidelity Prototyping

Low-fidelity prototyping was used at this stage to test interaction logic before any implementation commitment. The test targeted whether a proposed workflow could satisfy the core requirements from the Define phase: *cognitive offloading*, end-to-end *traceability*, and human-in-the-loop decision-making. Visual refinement was not the aim at this stage. Two independent low-fidelity wireframes were produced and then compared, and compatible elements were merged into a mid-fidelity wireframe, see Figure 3.3.

Two key design decisions emerged from this process. First, a configuration setup flow into sequential steps: case name, transcript upload, and read-

ing of transcript and importantly the interrogator initial note-taking. This enforced a preparation sequence aim to prevent the user from skipping directly to AI output. Second, a three-column workspace layout placed AI-generated analysis findings on the left, a transcript in the center, and contextual AI chat on the right. This arrangement allowed the source document and helping tools remain visible throughout editing.

These decisions were reviewed with the project supervisor and a potential user. The supervisor confirmed the setup flow's sequencing logic, the potential user noted that the approach was operationally meaningful as interrogators must form an independent reading before any AI output appears. The workflow structure was confirmed, but questions of visual clarity, information density, and three-column usability remained open. These unresolved questions drove the transition to a high-fidelity implementation.

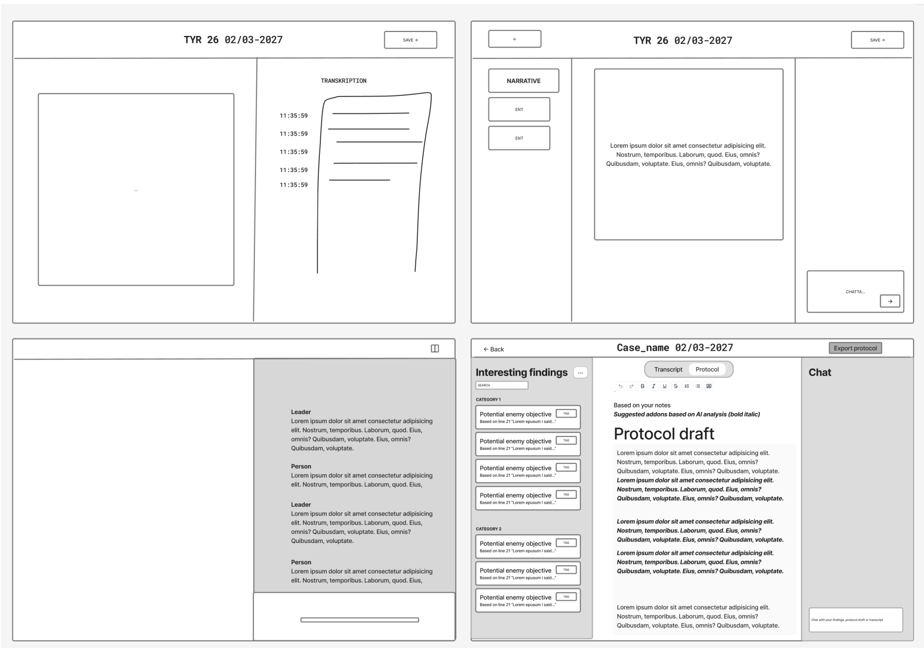

Figure 3.3: Low and mid fidelity screens

3.3.4 High-Fidelity Prototyping

High-fidelity prototyping included iterative work using Lovable, an AI-powered application builder, to generate interfaces from natural-language prompts derived directly from the mid-fidelity wireframes. This enabled a rapid loop of prompt, render, review, and refine with less manual frontend coding. Lovable produced the structural shell of the *three-panel workspace*, the setup configuration, and the *findings panel* layout before any real data were wired up, serving as scaffolding for the coding implementation phase. The coding phase began with planning the full stack architecture, before implementing it, also with help from AI-assisted tools such as Claude Code, which accelerated both learning new principles and building the system. The resulting working prototype served as the basis for the design iterations and co-design session described in the following subsections.

Design Iterations

The working prototype at this stage included a sequential setup configuration and a *three-panel analysis workspace*, with a *findings panel* on the left, a *transcript view* and *protocol editor* in the center, and a *chat panel* on the right. At this stage, the *findings panel* rendered all AI output as a flat list. Two design challenges and one team contribution emerged during implementation.

Analysis Latency on Local Hardware. The most consequential design shift addressed the latency that full AI analysis introduced into the user flow. Because no data leaves the interrogator's machine, runtime depends entirely on local hardware, which varies across the computers used in operational settings. On a mid-range personal computer, a complete analysis required 20-60 minutes depending on model choice. Low-tier hardware also degraded interface responsiveness while the pipeline executed. This left the interrogator blocked immediately after an interrogation, exactly when short-term memory is sharpest and time pressure is highest. The solution restructured delivery so that an intelligence summary and an initial protocol draft appeared within approximately one minute. Deeper perspective findings were then loaded progressively in the background, allowing the interrogator to start reviewing and editing without waiting. To absorb hardware variability, three *workflow modes* were introduced so the interrogator could choose a practical depth-speed trade-off. Their detailed behavior is described in Section 4.5.6.

Flat Findings List. The first working prototype rendered every analytical perspective as one long, repetitive list of *findings*. Navigating this list meant scanning through structurally identical text blocks to locate relevant information. That scanning cost is cognitively demanding for interrogators already operating under time pressure. The diagnosis, however, was not purely visual. Each analytical perspective produces a qualitatively different kind of insight. *Timeline reconstruction*, for example, orders events along temporal gaps. *PIR assessment*, by contrast, structures answers to each intelligence requirement. These *findings* are therefore structurally distinct, and each suggests a different visual treatment. The redesign introduced tab-based organization with visualization components matched to each content type. This change preserved the analytical character of each output and reduced the overhead of moving between perspectives.

Team Contribution: Audio Transcription. During the implementation phase, an IT architect joined the project team and independently designed and built an audio transcription capability. The feature converts audio recordings into formatted transcripts using locally hosted speech-to-text and speaker separation models, specifically Whisper large-v3, maintaining the system's offline constraint. It was integrated into the setup wizard as an alternative audio file input alongside transcript text upload. The thesis authors incorporated the resulting interface elements but did not design the underlying pipeline logic. The design process documented in this thesis, from interviews through requirements to design decisions, focuses on the analysis pipeline and the interrogator-facing interface. The audio transcription feature extends the system functionalities without altering the analytical workflow, and is acknowledged here transparently as a team contribution rather than part of the authors' own design work.

3.3.5 Co-Design Session

Where the previous prototyping and coding phases validated interaction logic and interface clarity, the co-design session tested a different dimension, whether the system's analytical scope and output structure were accurate from a domain perspective.

Session Overview

To better understand the protocol structure and refine the system's analytical scope against operational expertise, a single co-design session was held with two senior interrogation leaders from the studied organization present together. The session lasted approximately one hour. It was structured as a section-by-section walkthrough of the interrogation protocol template to establish what AI could and could not contribute to each field.

Protocol Structure Requirements

The interrogators described a mandatory three-part separation that protocols must enforce: an objective account of what was said and done, the interrogator's own comments clearly delimited with start and end markers, and any assessments marked as a separate block. Only content in the comments block may include the interrogator's interpretation, and only the assessment block may contain conclusions. Any AI-generated summary that included these categories into a single narrative would violate the reporting standard regardless of how accurate its content was. The prototype's initial protocol draft did not enforce this separation, and the session made clear that this was the most critical structural fix needed before evaluation.

Intelligence Requirement Hierarchy

The session also clarified the intelligence requirement hierarchy used in practice. The initial design had modeled only Priority Intelligence Requirements. The session revealed that interrogators operate within a multi-level structure: PIRs define broad collection goals, Specific Information Requirements (SIRs) define thematic subsets, and Essential Elements of Information (EEIs) are targeted questions answerable from individual statements. The interrogators noted that they typically carry broad PIRs into a session to stay agile in the room, while the AI analysis benefits from access to all three levels in order to map extracted findings at the appropriate granularity.

AI Scope and Evaluation Boundaries

The session narrowed AI support to two protocol sections, the session description and the intelligence extraction section, because both derive directly from the transcript and require no behavioral observation beyond the audio

record. The session description provides factual framing of who was interviewed, when, and where, anchoring the protocol in verifiable context. The intelligence extraction section, captures substantive answers tied to the active PIRs, SIRs, and EEIs. Concentrating implementation effort on these two sections sharpened the analytical scope of the prototype.

4 The FENRIR System

The preceding chapter described how FENRIR was shaped through discovery, definition, and development. This chapter describes the parts that result from those decisions. This includes a complete interrogator workbench that carries a case from an audio recording to a transcript, and finally to an intelligence protocol. The system integrates the cognitive requirements surfaced in the expert interview, the three-layer analytical framework, the PIR hierarchy refined during co-design, and the decisions made across iterative prototypes. The Evaluation chapter that follows then assesses how this system performs in practice.

4.1 The Landing Page

When the interrogator opens FENRIR, a *landing page* serves as the entry point for both new and returning users. The header displays a ”Local” badge, confirming that all data remains on the interrogator's machine. Below the header, a *case gallery* presents existing projects as a card grid. Each card shows the case name, a status indicator for the current pipeline state (for example “Analyzing”), and the creation date. Hovering over a card reveals a button for showing a dropdown menu with options to enter or delete the project. Search and sort controls let the interrogator locate a specific case quickly, which is essential when multiple investigations run concurrently. To start a new project, the interrogator clicks *New Protocol*, which opens the setup wizard described in the next section.

Above the gallery, a *carousel* of five animated videos introduces the workspace capabilities before the interrogator encounters them. The cards preview the *analysis workspace*, the structured *findings panel*, evidence traceability through inline citations, the *protocol editor*, and the chat interface. Each card pairs a video with a short explanation of the section's purpose,

reducing the discovery burden for new users by simplifying the system's complexity. Experienced interrogators can scroll past the carousel directly to their case list, so the onboarding material does not interfere with routine workflows.

4.2 Guided Project Setup

Before any AI output appears, FENRIR routes the interrogator through four sequential wizard steps. Step one captures project metadata such as case name and interrogation date. Step two handles data ingestion in two parts. First, the interrogator provides the source material, either a text transcript or an audio recording. When audio is selected, the system transcribes it locally using speech-to-text and speaker separation. While the audio is being transcribed, a live preview appears below the upload area. Segments with speaker labels and timestamps populate the preview as transcription completes. Then in the second part of the data-ingestion step, the interrogator enters intelligence requirements as questions and sub-questions (PIR and SIR) established in the co-design session. These are the prioritized questions that will be answered during the AI analysis. In step three, the interrogator can optionally write prior knowledge, behavioral observations, and working hypotheses in a text input. Finally, in step four, the interrogator selects one of three workflow modes (*quick start*, *standard*, or *full analysis*) before starting the pipeline.

The setup's sequential structure is designed to mitigate anchoring bias, which was raised during the expert interview. The *initial thoughts* step is optional, but placing it before any AI output appears serves two purposes. First, it counteracts anchoring bias, since an interrogator who has already committed their own reading of the interrogation is less likely to defer uncritically to the system's later findings. Second, it lets the interrogator offload working memory onto the page, recording prior knowledge, observations, and hypotheses before the cognitive demand of reviewing AI output begins. When the interrogator chooses to write notes, these are injected into the analysis workflow.

4.3 The Analysis Workspace

The *analysis workspace* organises the work into three columns, shown in Figure 4.1. The *findings panel* sits on the left, the interrogation transcript and *protocol editor* occupy the centre, and the *contextual chat* sits on the right. Intelligence analysis demands context from multiple sources at once. The interrogator must compare findings and ideas against the original transcript while drafting the protocol. The modular layout addresses context-switching costs directly. Panels can be resized, collapsed, or split so the interrogator can, for example, view the transcript alongside the *protocol editor*. All panels remain visible by default, and the layout adapts as the task shifts from reviewing and verifying findings to editing the protocol draft.

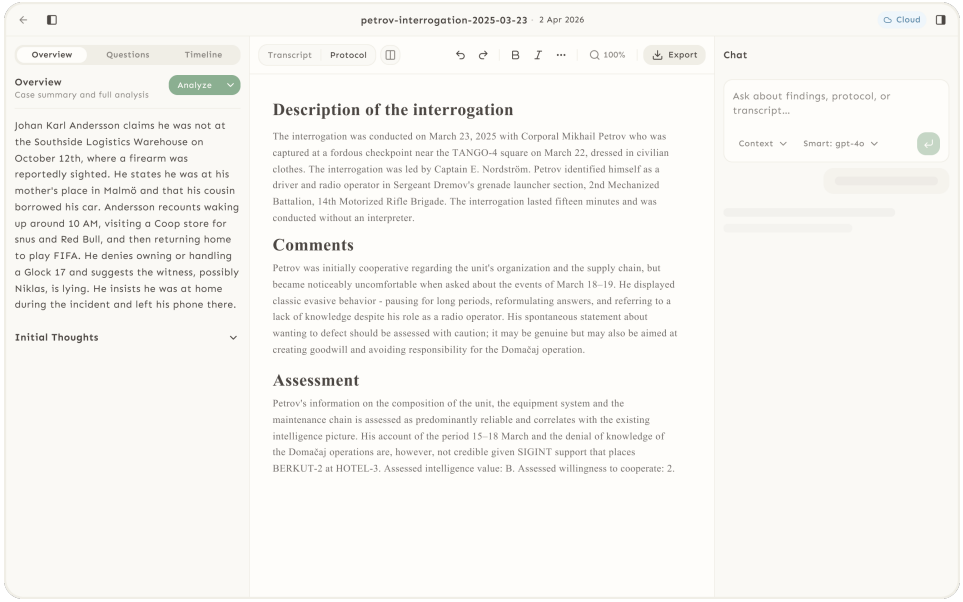

Figure 4.1: The FENRIR analysis workspace.

When the project originates from an audio recording, the *transcript view* includes an integrated *audio bar*. The bar shows where in the recording the current passage is being spoken, and lets the interrogator re-listen to specific parts.

Early prototypes displayed all perspective output as a uniform list of text blocks. This uniform layout created a wall of text that was cognitively demanding to scan under time pressure. It also flattened the analytical structure produced by the pipeline, hiding the distinctions between event reconstruction,

PIR coverage, and gap analysis. The solution was tab-based organisation with a visualisation matched to each content type.

The *findings panel* itself is structured around a tab layout in the left column, with each tab opening a different view onto the same underlying analysis (see Figure 4.2). The three implemented tabs sit side by side, the interrogator switches between them with a single click, and the active tab fills the panel below the layout. This keeps the centre column free for the transcript and *protocol editor*, and the right column free for chat, while still giving the analytical output a dedicated home.

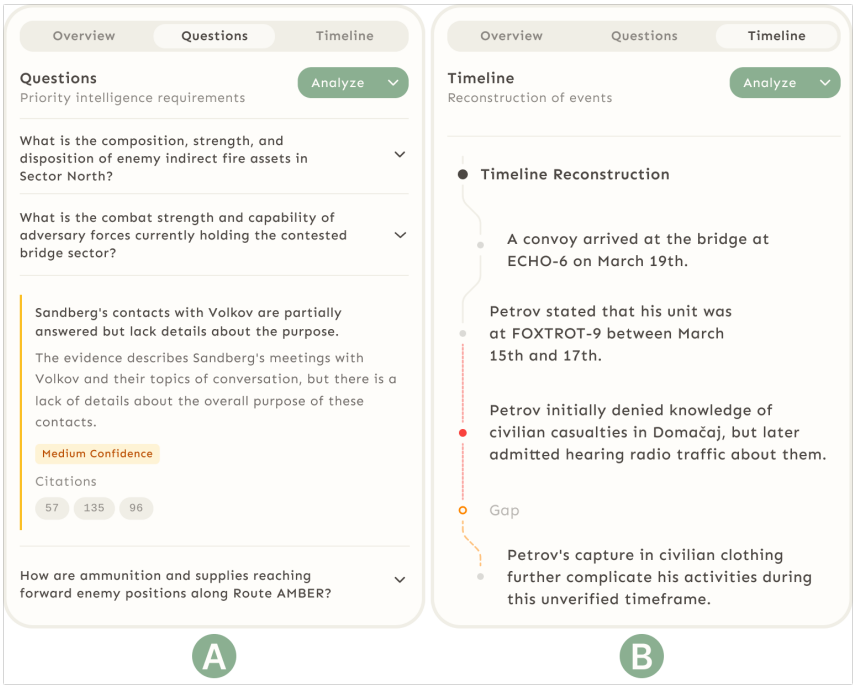

Figure 4.2: Questions tab and timeline tab in the findings panel.

The *overview tab* presents a summary alongside the interrogator's own notes from the setup. The *questions tab* groups findings by PIR, highlighting what is known, partially known, and unanswered for each intelligence requirement (see Figure 4.2a). The *timeline tab* presents chronological event reconstruction, showing sequence and time gaps (see Figure 4.2b). Each tab serves a distinct analytical concern, not a generic list.

Within each tab, findings arrive progressively as analytical perspectives complete. An analytical perspective is a single AI processing stage that turns the extracted evidence into a typed set of findings, and FENRIR implements three of them (*timeline reconstruction*, *PIR analysis*, and *gap analysis*). Individual *finding cards* use an accordion pattern. The headline and confidence level are visible by default, while motivation and source evidence are revealed when the interrogator expands the card. A collapsible progress bar at the bottom of the *findings panel* exposes the analysis progress. Critically, every finding includes inline citations that specify a transcript line. Clicking a citation highlights those lines in the *transcript view*.

The three workflow modes, *quick start*, *standard*, and *full analysis*, shape how progressively these findings appear and how soon the interrogator can begin editing. These modes are further described in Section 4.5.6. In the *findings panel* header, a *perspective dropdown* allows re-runs of individual analyses. This is especially important in *standard* mode, where perspectives can start before entity extraction is complete, so early findings may not yet include the richer structured context. Once the entity database has finished building, the interrogator can re-run any perspective without restarting the full pipeline, and the updated context can reveal connections that the initial analysis missed.

A toolbar at the top of the centre panel offers three view modes. *Transcript only* and *protocol only* each expand a single document to the full width of the panel, while *split view* places the transcript and the *protocol editor* side by side. The *split view* lets the interrogator verify every AI-drafted claim against the source passage without any navigation, and therefore addresses the co-design requirement that the interface must let the interrogator verify protocol statements by tracing them back to the original text. A text editor renders the AI-drafted protocol following the interrogation-protocol section structure agreed on in co-design. The draft is a starting point and includes empty sections and AI-suggestions based on the transcript and findings. The interrogator edits freely, and all changes are autosaved. One current limitation is that AI-generated text and interrogator-written text are not visually distinguished within the editor. Introducing a visual differentiation between AI-drafted and human-revised content was identified as an important future feature and is revisited in Section 6.1.2.

The *chat panel* provides a natural-language interface for querying the case evidence directly. The interrogator selects which context the model draws from (transcript, findings, protocol, or a combination) and whether to use the *fast model* or *smart model* depending on urgency or hardware capabilities.

Responses are grounded through Retrieval-Augmented Generation (RAG, see Section 4.5.3) over the transcript chunk store and the structured entity database, so the model answers from the actual case data rather than general knowledge. Every response includes inline citations specifying the transcript line numbers on which the answer is based. Clicking a citation highlights the corresponding passage in the *transcript view* and links the claim to the relevant finding in the left panel when applicable; see Figure 4.3a-b. This is the same traceability mechanism used in the *findings panel*. The interrogator can probe a specific claim, ask for alternative interpretations, or request feedback on the protocol draft. While the *findings panel* answers predefined intelligence questions, the chat supports follow-up questions that arise during the interrogator's review.

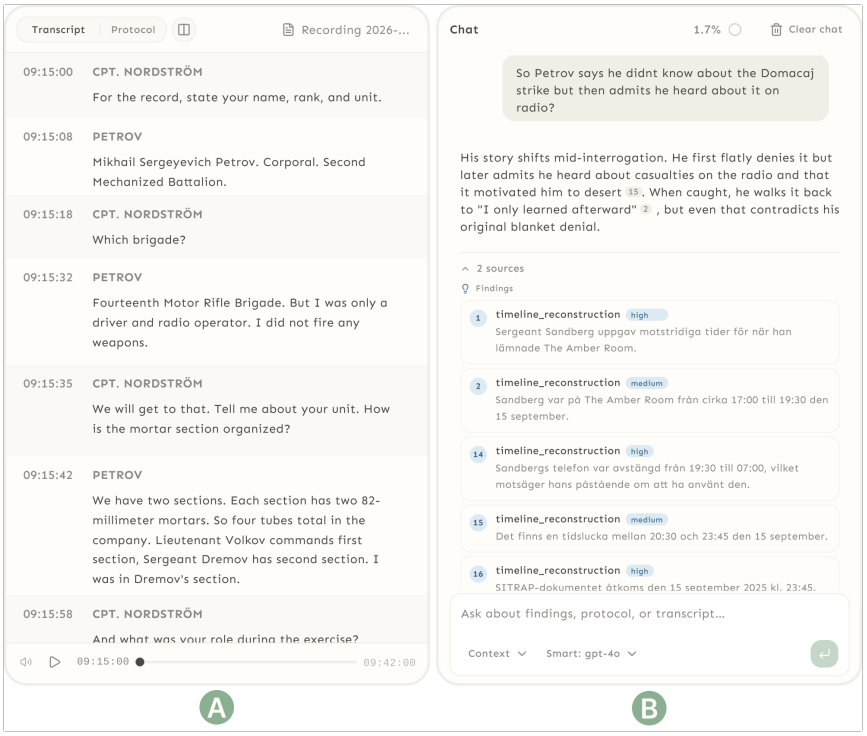

Figure 4.3: Traceability from the chat panel to the transcript view.

4.4 Interface Design Choices

Several visual design choices reflect the specific conditions under which the system will be used. The interface is designed for desktop-first use and offers a dark mode option, acknowledging that analytical sessions may occur in low-light operational environments. The UI is bilingual. The target users' local language is the primary interface language, with English available as a secondary option to support international collaboration without localization friction. All interactive components are built on Shadcn UI[1] and Radix primitives[2], providing a consistent and accessible design system. Animations are used rarely and intentionally. For example, a notification toaster informs the interrogator when an AI analysis completes, rather than forcing them to wait for a result to appear.

4.5 System Architecture and Pipeline Design

The requirements established during Section 3.2 placed four constraints on the system architecture.

- **Offline execution**, so that sensitive interrogation data never leaves the interrogator's machine.
- **Traceability** from every claim back to the specific transcript line that produced it.
- **Progressive output delivery**, so the interrogator can begin working before the full analysis completes.
- **Cognitive load reduction** during protocol drafting.

The resulting design rests on three pillars. A multi-model strategy routes each task to a purpose-matched model, balancing analytical depth against throughput on a single local Graphics Processing Unit (GPU). A three-phase

[1] Shadcn UI is an open-source collection of React components built on Radix primitives. See https://ui.shadcn.com.

[2] Radix primitives are unstyled, accessible UI building blocks. See https://www.radix-ui.com/primitives.

pipeline mirrors the analytical framework introduced in Section 3.2, moving from *ingestion* through *knowledge construction* to *intelligence analysis*. A dual-store architecture links vector embeddings to structured entity records, enabling end-to-end *traceability*. Progressive delivery then lets the interrogator work in parallel with the ongoing analysis.

4.5.1 System Topology

Every component of FENRIR executes offline on the interrogator's own machine. No network connection is required at any point during usage, ensuring that sensitive interrogation data never leaves the local environment. The architecture includes a web-based frontend that communicates with a Python backend, which in turn dispatches requests to a locally hosted server. The Next.js[3] application serves the interrogator workbench. The FastAPI[4] pipeline orchestrator listens for requests, and Ollama[5] hosts all language models. Figure 4.4 shows how these components connect.

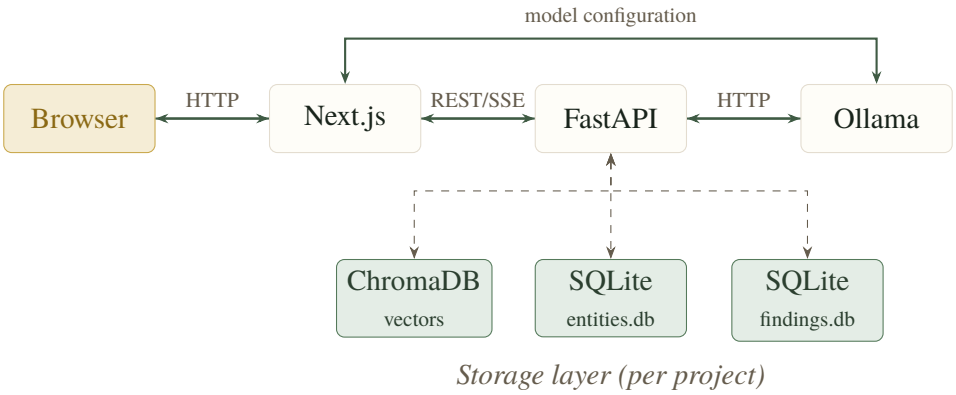

Figure 4.4: FENRIR system topology.

Data isolation is enforced at the project level. Each project receives its own directory containing all associated stores, such as vector embeddings, structured entity databases, and session metadata. Projects do not share state, and an interrogator can create, analyze, and delete projects independently without

[3]Next.js is a React-based framework for building web applications. See https://nextjs.org.
[4]FastAPI is a Python web framework for building APIs. See https://fastapi.tiangolo.com.
[5]Ollama is a local runtime for hosting and serving large language models on the user's own hardware. See https://ollama.com.

affecting other investigations. Table 4.1 summarizes the technology choices at each layer.

Layer	Technology
Frontend	Next.js, React, TypeScript, Shadcn UI, Tailwind, TipTap
Backend	FastAPI, Python
Storage	ChromaDB, SQLite
LLM	Ollama

Table 4.1: Technology stack overview.

4.5.2 Traceability Chain

Traceability is the first design commitment that this architecture enforces. In intelligence work, every claim requires a traceable source. This enables links to each analytical finding back through the processing layers to the specific transcript lines that produced it. This implements the *traceability* requirement from the Define phase (Section 3.2). In the interface, clicking a citation in the *findings panel* or *chat panel* navigates the interrogator to the corresponding transcript line. The chain spans four layers, each connected to the next through explicit references:

1. **Transcript lines:** The raw input, with each line retaining its original line number, timestamp, and speaker label after sanitization.

2. **Chunks:** Segments of transcript lines stored in the vector database. Every chunk records which source lines it contains, preserving the bridge to the original document.

3. **Entities and events:** Structured records produced by the extraction phase. Each entity mention links to the specific chunk from which it was extracted. Events reference their participant entities and inherit the same chunk-level traceability.

4. **Findings:** Analytical perspectives generate typed observations with a summary, detailed reasoning, confidence level, and references to the chunks, entities, and events that produced them. Findings contribute as the evidentiary basis for the protocol writing.

4.5.3 Multi-Model Strategy

Rather than routing all tasks through a single language model, FENRIR assigns each processing step to a purpose-matched model, that is, a model selected for the specific demands of that step. A small, fast model handles text normalization, while a larger reasoning model drives the analytical perspectives. This decision follows directly from the offline constraint. A large reasoning model delivers the analytical depth required for summary and perspective generation, but its inference speed is impractical for the many smaller tasks the pipeline must perform repeatedly, such as text normalization and speaker identification. A small model handles those mechanical tasks efficiently, yet it lacks the capacity for nuanced analytical reasoning. Splitting responsibilities across models balances quality against throughput within the limits of a single local GPU. Table 4.2 lists the four model roles and their default assignments. All assignments are configurable through the system settings, so interrogators can adapt the pipeline to the hardware available in their deployment context.

A related architectural choice concerns how models access transcript content. Local models typically offer limited context windows, so the full transcript may not fit in a single prompt. FENRIR therefore uses Retrieval-Augmented Generation (RAG), where relevant text passages are retrieved per query rather than passed in full. The trade-offs between RAG and long-context approaches are discussed in Chapter 6.

Role	**Default Model**	**Tasks**
Fast model	Llama 3.2 3B	Text normalization, speaker ID fallback
Extraction model	Ministral 3 8B	Per-chunk entity and event extraction
Smart model	GPT-OSS 20B	Summary, analysis, perspectives, drafting
Embedding model	Nomic-Embed-Text	Chunk embeddings, similarity search
Speech Recognition	Whisper large-v3	Audio-to-text transcription
Diarization	pyannote.audio	Speaker segmentation

Table 4.2: Default model roles.

4.5.4 Ingestion and Knowledge Construction

Semantic similarity search matches a query to text segments by comparing their meaning rather than their exact wording, using vector embeddings to rank segments by closeness in meaning. This retrieves text segments that relate to a query, but intelligence analysis demands structured answers. For example, who was present, what objects were mentioned, where and when events occurred, and how these elements connect. The pipeline therefore builds a complementary structured knowledge layer through two sequential phases before any analytical reasoning begins. Each phase depends strictly on the preceding one. When the input is an audio recording rather than a text transcript, an optional transcription pre-stage runs first. Whisper large-v3[6] handles speech-to-text, and pyannote.audio[7] performs speaker separation, producing the formatted transcript that the three-phase pipeline then consumes. Figure 4.5 shows the full pipeline dataflow.

The *ingestion* phase transforms the transcript into a searchable knowledge base that preserves who said what and when. Each line of the transcript is first tagged with its line number, timestamp, and speaker identity. When the formatting alone is not enough to determine the speaker, the *fast model* infers the tags based on surrounding context. The tagged lines then need to be divided into meaningful segments before they can be searched. Simply splitting the text at fixed intervals would cut segments mid-sentence. The pipeline therefore uses a three-step process for creating the segments from the raw transcript.

1. Consecutive lines from the same speaker are grouped into turns.
2. Unusually long turns are split at sentence boundaries so that no segment exceeds a practical length.
3. Adjacent segments whose content overlaps are merged back together.

Each resulting segment is converted into a vector embedding that captures its meaning, and is stored in ChromaDB[8]. This is what enables retrieval and semantic search in RAG.

[6] Whisper is an open-source automatic speech recognition model from OpenAI. See https://github.com/openai/whisper.

[7] pyannote.audio is an open-source toolkit for speaker diarization in Python. See https://github.com/pyannote/pyannote-audio.

[8] ChromaDB is an open-source vector database for storing and searching text embeddings. See https://www.trychroma.com.

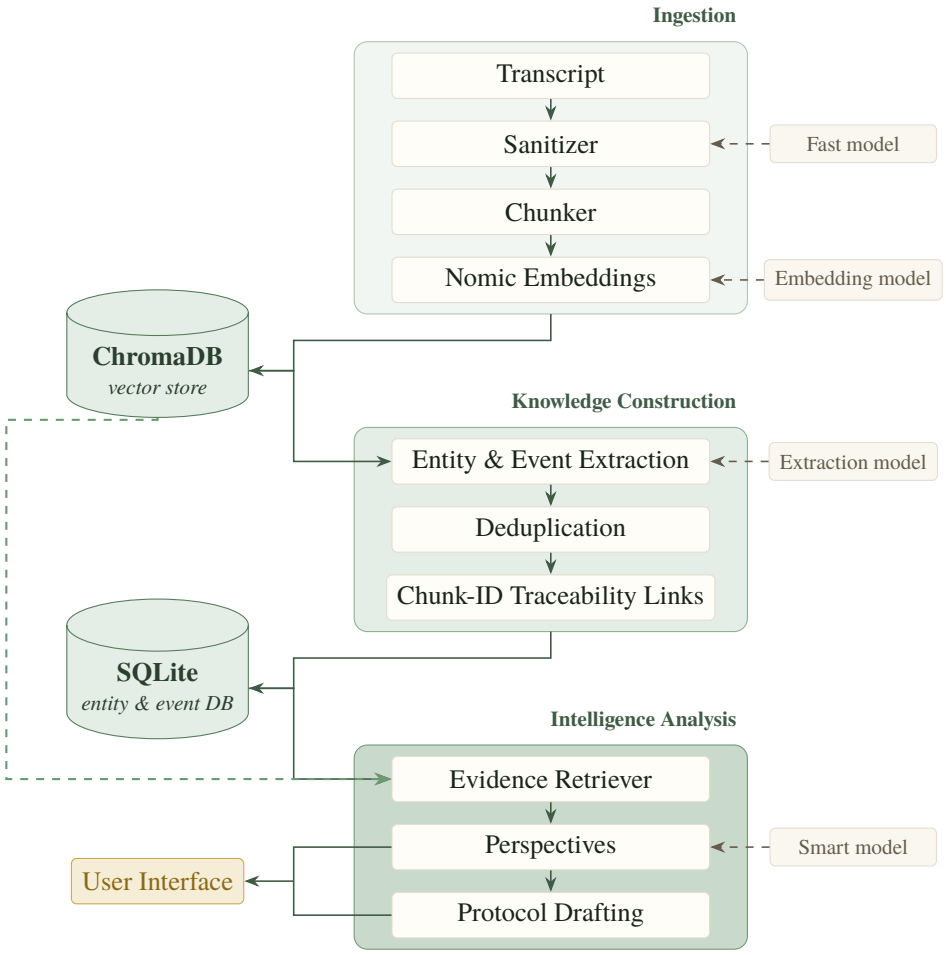

Figure 4.5: Three-phase pipeline data flow.

The second phase, *knowledge construction*, processes every stored chunk through a dedicated *extraction model*. The model identifies people, locations, objects, organizations, and the events that connect them. Each event is assigned to one of two temporal layers. The interrogation part of the transcript captures what happened during the interview itself, and the narrated part captures events the subject describes from the past. This distinction matters for later *timeline reconstruction*. A *deduplication* pipeline then merges overlapping mentions across chunks into standardized, cross-referenced records stored in SQLite. The resulting knowledge layer maps directly to the *semantic layer* of the analytical framework (Section 3.2), establishing the factual foundation that subsequent analysis builds on.

These two stores form FENRIR's dual-store architecture and serve complementary roles. ChromaDB provides semantic similarity search, returning the most relevant transcript segments for a given query. SQLite holds the structured entity and event records produced by the *knowledge construction*. Every entity mention stores the identifier of the chunk from which it was extracted, connecting the two stores. The *evidence retriever* traverses this link in three steps. First, it queries ChromaDB for the transcript segments most semantically relevant to the query. Second, it follows the chunk identifiers into SQLite and retrieves every entity mentioned in those segments. Third, it expands the result with every event tied to those entities, including events whose source text did not rank among the top retrieval results. This expansion beyond what semantic search alone would return becomes critical when analytical perspectives need to reason across scattered references to the same entity. A trade-off is that RAG's probabilistic retrieval may still miss relevant information.

4.5.5 Intelligence Analysis and Protocol Drafting

The *intelligence analysis* phase transforms structured evidence into findings that directly address the interrogator's intelligence requirements. For each Priority Intelligence Requirement (PIR), the *evidence retriever* assembles an *evidence pack* containing the most relevant transcript records from ChromaDB together with all related entities and events from SQLite. Phase dependencies are enforced. The analysis cannot begin until both the vector store and the entity database are populated, ensuring every perspective operates on a complete evidentiary foundation.

Then, four analytical perspectives process these *evidence packs*. *Timeline*

reconstruction orders extracted events chronologically, identifies missing periods in the sequence, and flags conflicts where events contradict each other. *PIR analysis* maps retrieved evidence to each intelligence requirement, assessing what is known, partially known, or unanswered. *Gap analysis* complements *PIR analysis* by focusing on unanswered questions and partial answers. Together, these two perspectives produce a nuanced view of both what the transcript establishes and what it does not.

To illustrate, consider a PIR such as "Identify any external contacts the subject coordinated with in the past month." *Timeline reconstruction* first orders the relevant events the subject describes, surfacing two meetings at a café and a phone call placed the following day. *PIR analysis* then maps these events to the requirement, classifying the contact at the second meeting as known and the phone-call recipient as partially known because only a first name was recorded. *Gap analysis* finally flags the unidentified recipient as an open question and notes that the subject avoided naming a third person referenced in passing. The interrogator can probe both findings further in chat or in follow-up interviews.

4.5.6 Progressive Delivery and Workflow Modes

During project configuration, the interrogator chooses between three workflow modes that balance analytical depth against available time. The three modes are *quick start*, *standard*, and *full analysis*. The design iterations described in Section 3.3.4 revealed that output delivery must be configurable, because time pressure and hardware demands vary widely between workflows.

All modes share the same progressive-delivery mechanism. The architecture delivers a summary and protocol draft within approximately one minute so the interrogator can begin writing in the workspace. In parallel, deeper analysis runs in the background, first building the vector and entity databases and then generating analytical perspectives whose results are progressively revealed in the interface. The collapsible progress bar and notification toasters described in Section 4.3 keep the interrogator informed of this background work without interrupting the editing flow. Results persist at each phase, enabling state recovery if the connection drops or if the interrogator leaves the workspace.

Quick start skips entity extraction and the analytical perspectives, which can still be triggered on demand. The resulting draft is functional but lacks the entity context and cross-referenced findings that deeper analysis provides.

Standard mode closes this gap progressively. The interrogator begins editing immediately while extraction and perspectives run in the background, and findings appear in the workspace as each perspective completes.

Full analysis prioritizes completeness over speed and suits situations where time pressure is low, such as overnight processing. The mode runs the full pipeline before the interrogator begins editing, including entity extraction and every analytical perspective. The interrogator then opens a complete draft instead of watching findings appear progressively.

Full analysis originated from a methodological need. The initial evaluation plan was to compare AI-generated protocols against human-written ones, which required a pipeline capable of producing a complete draft without user intervention. That plan later shifted, and the reasoning is addressed in the Discussion chapter's Methodological Limitations (Section 6.2). *Full analysis* remained the mechanism for observing the full pipeline and inspecting the output of each AI stage. As a consequence, the human-in-the-loop principle established in the preceding chapters (see Section 3.2) does not apply inside *full analysis*. No checkpoints interrupt processing, and the draft is not shaped by interrogator decisions along the way. The exact evaluation process is described in Chapter 5.

4.5.7 Prompt Design and Calibration

The pipeline architecture determines how data flows between processing stages. Prompt engineering determines what intelligence the models actually produce from that data. Every model in FENRIR operates within parameter constraints, none matching the reasoning capacity of cloud-hosted LLMs. Prompts must therefore compensate by imposing strict structure and clear output schemas.

Four design principles direct every prompt in the system.

1. Every finding must reference the specific chunks, entities, and events from which it was derived. A claim without a source reference is discarded.

2. Prompts mirror the distinction that the interrogation protocol makes between documented facts and analytical interpretation. *Observational findings* explain what the transcript states, while *interpretive findings* explain what is understood. For example, an interpretive finding might infer that two named individuals are coordinating, based on overlapping

references in the transcript. Mixing the two in a single finding is not allowed.

3. Models generate findings only when the evidence provided supports them.

4. Draft-writing prompts prefer leaving empty paragraphs over fabricating content, ensuring that every drafted passage rests on real intelligence findings.

Prompt quality was evaluated across eight dimensions that reflect the requirements of an interrogation protocol in the studied organization. These include factual accuracy, completeness, analytical usability, structure according to the protocol template, separation of observation from assessment, naming conventions, correct use of format markers, and section coherence. These dimensions were inspired by the interview conducted during the research phase of the design process and by the co-design session with domain experts. Each dimension was scored on a five-point Likert scale by an LLM with instructions on how to judge the pipeline's results.

An automated calibration system used these scores to iteratively improve prompts. Each iteration generated a prompt variation based on prompt engineering principles. It then ran the full pipeline against five fabricated test transcripts with known ground truth and evaluated the resulting protocols. Changes that improved the score were committed, while regressions were rolled back. The LLM's memory was then reset and the calibration restarted. The calibration ran for approximately 60 iterations. OpenAI's GPT-5.3 generated both the prompt variations and the evaluation judgments, while the pipeline itself ran on a mix of local and online models due to time constraints. All test transcripts used during calibration were fabricated and contained no real or classified information, so the offline constraint applied exclusively to live operational data. This approach carries a known limitation. The same model family generated the outputs and judged their quality, introducing self-preference bias. To partially mitigate this, the authors manually reviewed pipeline results and prompt changes at checkpoints every ten iterations, verifying that score improvements corresponded to genuine quality gains. The most effective improvements came from adding concrete output examples to prompts, which gave the local models clearer targets for format and structure. Adjusting formality and sentence-level guidance for the target-language prompts also produced consistent gains.

5 Evaluation

This chapter marks the Deliver phase of the Double Diamond, closing the process opened in the Design Process chapter. Its aim is to examine whether the FENRIR prototype, described in the preceding chapter, integrates into the operational workflow of practitioners from related areas. The evaluation is therefore scoped to product fit, meaning how the system aligns with the research questions and the user goals that guided its design. This framing shaped the methodological choice of a guided walkthrough, since operational integration is best surfaced through expert reflection on the workflow as a whole. The Method section that follows details the participants, study design, procedure, and analytic approach. The Results section then reports the findings, organised into themes. The main themes concern source traceability as a precondition for trust, a demand for multi-session scope beyond the prototype, and the *workflow fit* gained after an interrogation against the verification overhead it introduces.

5.1 Method

The final phase of the design process evaluated whether the prototype of FENRIR integrated meaningfully into the operational workflow of interrogators. The evaluation focused on product fit, which means how the system was received and where it aligned with the initial user goals and the research questions. The study therefore adopted a guided walkthrough of the prototype, and a subsequent semi-structured interview. Lastly, the resulting transcripts were analyzed with a reflexive thematic analysis that combined inductive and deductive coding (Braun & Clarke, 2006). An initial pilot session was conducted to refine the walkthrough script and the interview guide before the main evaluation.

5.1.1 Recruitment

The evaluation combined two sampling strategies. The initial intent was to recruit practising interrogators, but the limited availability and the project's timeframe made this unfeasible. The sample was therefore broadened to include four practitioners from the studied organization who work within the same operational domain but are not themselves interrogators. They were recruited through a convenience sample drawing on the authors' professional network. Their backgrounds covered adjacent operational specialisations within the studied organization. All four are engineers, and each was familiar with AI tools or had worked specifically with interrogations. Although none had served as an interrogation leader, all were familiar with the broader high-stakes operational context, which broadened the analytical value of the sample. The fifth participant was a domain expert with extensive practical experience in structured interviewing and protocol writing in security-critical settings. The combination allowed the evaluation to capture both broad operational reactions and deep domain-specific critique.

In addition, one pilot session was conducted prior to the main evaluation to refine the walkthrough and the interview guide. The pilot participant is a domain expert with extensive practical experience in structured interviewing and protocol writing in security-critical settings, alongside prior experience in police work, and currently works in information security while pursuing doctoral studies in psychology. The participant is also a stakeholder in the FENRIR project, and this relationship is acknowledged here to maintain transparency regarding its potential influence on the evaluation.

5.1.2 Study Design and Scenario

The study was designed as an evaluation that combined a guided product walkthrough with a semi-structured interview. A guided walkthrough, rather than hands-on use, was chosen as a consequence of the limited availability of practitioners. Also, the methodology was chosen because the primary objective was to assess operational fit, not task performance or learnability. This allowed the authors to present the full workflow in a controlled sequence, including features that would otherwise require long analysis runs to appear.

Participants were asked to follow along during the walkthrough, think aloud, and interrupt with questions. An interrogation scenario, centered on the questioning of a captured individual, was used throughout all sessions. A transcript and a set of Priority (PIR) and Specific Information Requirements

(SIR) were written to provide the scenario with additional operational context. This kept the session free from classification concerns while still anchoring the demonstration in a realistic high-stakes investigative context. One practical deviation from the operational target was the language model configuration. FENRIR is designed to run entirely offline on local hardware. To keep the walkthrough within a practical length and to avoid long idle periods, the demo was run against an online OpenAI endpoint. Because the study evaluates product fit rather than output quality, this change did not affect the result of the investigation. Participants were informed of the substitution at the start of the session. One further session-level condition should be noted: the domain expert viewed the demo on a mobile phone, which bounds the evidentiary weight of their reactions to visual UI elements such as typography, colour coding, and the *timeline rendering*.

5.1.3 Materials

The evaluation relied on four parts. The first was the FENRIR prototype described in Chapter 4, deployed on the authors' machine and shared with participants through the screen-share function on Google Meet. The prototype represented the state of the system at the end of the Develop phase. The second part was a fabricated interrogation transcript created using Claude Opus 4.6 and a prompt retained on file by the authors. The third part was a set of PIRs and SIRs also generated. These AI generated parts were then manually adjusted to match the fictional scenario. Together, these inputs produced the same analysis output across sessions, limiting content variability as a confounding factor. The fourth part was the interview guide used during the main evaluation. A pilot session preceded the main study and informed several changes to the guide. Specifically, a short contextual briefing on interrogation work and PIRs was added for the participants, the workflow theme was reframed to better answer the research questions, and the questions on human control were expanded to cover accountability between AI-generated and interrogator-written content in the protocol.

5.1.4 Procedure

Each session lasted approximately 65 minutes (range: 37-84 minutes) and used the following structure. It opened with a short introduction in which the authors presented the study's purpose, clarified that the session was a design

evaluation rather than a test of the participant, and obtained informed consent and consent for recording. Participants were told that they could stop the session and request deletion of the material at any time, and that all responses would be treated anonymously. The responsibility of the sessions was divided between the two authors: one led the interview, while the other navigated the prototype and asked follow-up questions when relevant.

The walkthrough guided participants through FENRIR in the order that an operator would use it, moving through the *project overview*, the *setup wizard*, the *three-panel workspace* with transcript and *protocol editor*, the *findings panel* and its perspectives, and the *chat assistant*. At each step the authors briefly explained the design rationale, and participants were encouraged to think aloud and interrupt with questions or reactions. After the walkthrough, a semi-structured interview was held. Questions were organized around five themes, covering *workflow fit* and cognitive load, human control and trust, AI quality and information coverage, feature relevance, and critical feedback, with follow-up questions used as needed to explore emerging topics in more depth. The complete interview guide, reproduced in the original Swedish with the fictional walkthrough scenario, is given in Appendix C. The session closed with an open invitation for additional comments and a short debrief.

5.1.5 Analysis

The recorded sessions were analysed using both inductive and deductive reflexive thematic analysis following the six-phase process described by Braun and Clarke (2006). In practice, this meant coding each transcript with two approaches, one that tested statements against themes defined in advance by the interview guide, and one that remained open to patterns emerging directly from the data. A purely deductive coding would have forced emergent observations into existing categories. In contrast, a purely inductive coding would have discarded the structure the guide had deliberately imposed. Combining the two allowed the authors to align the analysis with the research questions while leaving space for unanticipated findings.

After each session, the recording was transcribed using KlangAI, a GDPR-compliant transcription service, and stored on encrypted local drives for the duration of the analysis. As a first pass, both authors relistened to each recording and took independent notes, which were then revisited to find recurring patterns and to anchor codes in direct quotations. Coding was guided by the predefined areas of the interview guide while remaining open to patterns

emerging outside these framings. New themes were then constructed by clustering related codes and refined iteratively against the transcript. Once each transcript had been analyzed, the themes were summarized into a shared document of findings that formed the basis for the results reported below. All interviews were conducted in Swedish, the participants' first language, and the participant quotations presented in the Results section are the authors' English translations, reviewed against the original transcripts for fidelity.

5.2 Results

Thematic analysis produced five deductive themes organized around the interview guide and emergent themes that surfaced outside its scope. All five participants treat source traceability as an important requirement for trusting any AI-generated claim. Four of the five raise architectural requirements that exceed the prototype's single-transcript scope. All five found FENRIR to be beneficial for offloading the cognitive load after an interrogation while identifying a shift in analytical effort to verify AI-generated content. The participants are anonymized throughout the following sections. EX denotes the domain expert, and R1-R4 the four participants. The pilot participant is excluded from the reported findings since the subject is a stakeholder in the FENRIR project, and the pilot session is retained in the Method section as a guide-refinement step. All interviews were conducted in Swedish, and the participant quotations have been translated into English.

5.2.1 Workflow Fit and Cognitive Load

The primary cognitive-load benefit was identified directly after the interrogation, as every participant found the *initial thoughts* step to be an important feature for keeping the human in the loop and to mitigate anchoring bias. All five participants found FENRIR's clearest benefit in the translation of a completed mental picture into a structured written protocol. EX frames the most important requirement as time compression, stating that "the most important thing is actually time, every corner we can cut, everything that shaves off time will be good" (EX). R4 found the same benefit one level down, in information organization rather than transcription itself, because "information presented in a very clear and structured way allows me to focus on the analysis itself rather than on organizing the information" (R4).

Four participants identify verification fatigue as the new primary cognitive cost as a consequence of AI assistance shifting rather than eliminating the analytical effort. R1 points this out by asking whether the interrogator ”saves more work than is created through the need to quality-assure it” (R1). For R4, relief sits in the AI’s organization of information, but a new task is introduced in the need for an assessment stage, because he would want to confirm ”what this assessment is based on” (R4) before acting on any AI-generated conclusion. EX diverges from this concern and does not bring up any verification anxiety regarding cognitive load.

All five participants supported the *initial thoughts* note-writing step, but for different reasons. R2 values it for its *frictional design*, because it ”forces someone to think for themselves” (R2). EX reframes the step as a working-memory offloading, a place to quickly write off things that must not be forgotten before the draft arrives. R1 and R3 propose extending the step with more structure, for example by splitting the free-text field into separate inputs for known facts, judgements, and assessments, and by allowing the interrogator to attach supporting material such as adjacent interrogation protocols or reconnaissance reports. R3 draws the line most categorically between *objective extraction* and evaluative authorship, arguing that the evaluative sections ”should be a human who draws those conclusions” (R3) because the interrogator carries embodied evidence, such as body language, tone, and stress, that the model does not.

5.2.2 Human Control and Trust

All five participants treat the ability to trace an AI-generated claim to a specific transcript line as a must have feature for acting on it. This is the strongest convergence in the dataset. R4 labels the requirement with great importance, anchoring it in the stakes of intelligence work, because ”we can take decisions that are a matter of life or death, then I want to confirm things” (R4). EX frames the requirement as a professional norm rather than a usability preference. A few other aspects surface across three participants that go beyond the current citation and traceability features. R1 demands more from citations asking ”why do we believe what he says?” (R1), a question that is rather about source credibility than traceability. R2 identifies that displaying the AI’s reasoning steps is a missing feature in the current prototype, so that errors can be located in the middle of the reasoning rather than only in the resulting output. R4 asks for hovering text in a draft to expose both the PIR findings and the

transcript lines behind an assessment, demanding that the interaction remain smooth and fast to be worth the effort.

The second question concerns whether AI-written drafts should remain visually distinct on the exported protocol itself, which none of the participants thought. However, R4 nuances it as an additional verification option, because ”this very vital information, on which we will base our entire battle plan, is written by AI. Then I can contact the source handler and really confirm these details” (R4). The remaining participants place the value of visual distinction inside the drafting view rather than in the exported protocol. R2 explains that color-coded authorship during drafting helps because ”you really take ownership of the text” (R2), yet the same marker on the exported protocol risks harming how the recipient interpret its credibility, because ”those on the other side think no, that was just AI, half the document” (R2). R3 separates the two contexts directly, reserving the visible distinction for the interrogator's own review and treating the protocol's author signature as claiming total authorship. EX says that once the interrogator signs the protocol, the distinction between AI-written and interrogator-written text no longer matters for the exported document, because the signature makes the interrogator the author of it. The human is always responsible for the end result, regardless of how and what tool was used to create the protocol, that is the underlying consensus from every participant.

The consensus coexists with a practical gap that is itself a finding. Participants acknowledge that they would prioritize their review rather than in detail verify all claims. R4 states that he will be responsible for all content but in practice will review sources only for the most important parts. What drives the prioritization is pointed out by R1 that frames the problem as a cost-benefit concern, asking whether the interrogator ”saves more work than is created through the need to quality-assure it” (R1), and R2 adds a fatigue dimension, because ”when you are tired and have not slept you become dumber” (R2), which amplifies the gap under the operational conditions in which a protocol is actually drafted.

5.2.3 AI Quality and Information Coverage

The clearest finding in this theme is a mismatch between how FENRIR grades an intelligence finding and the formal grading scale that interrogators use professionally. FENRIR labels each AI-generated claim as low, medium, or high confidence, reflecting only the confidence of a claim based on the single tran-

script. The interviews reveal that a different probability scale should be used and that two or more independent sources are needed to claim something as likely. R1, R3, and R4 raise the mismatch. R4 describes the granularity of the formal scale he is trained on, explaining that interrogators apply a four-level vocabulary in which the highest level requires two independent observations to be claimed (R4).

Three participants frame hallucination as a known LLM risk. But in contrast, R1 brings up the human error premise with ”there is no natural law saying the interrogator must be right” (R1). R4 reflects on the concern by stating the need for always verifying specifics such as coordinates, names, and times. R2 anchors the same framing in prior personal experience of LLM output, accepting that errors occur and utilizing verification to mitigate them.

5.2.4 Feature Priorities and the Chat as Retrieval Aid

Setting aside the *protocol editor*, which every participant treats as the foundation for protocol creation, the *PIR-questions panel* is highlighted as the most important component of FENRIR. R3 motivates that the PIR a critical part of all intelligence gathering regardless of method. R4 extends the same motivation by saying that the questions tab is the specific information his commander wants. R2 pairs the PIR-panel with the timeline instead of isolating either one, treating the two together as ”quite critical for getting an overview” (R2) of a session. EX is the one participant who highlights a different component, pointing out the chat as the highest-value feature for supporting protocol writing.

Three participants frame the chat as retrospective information retrieval rather than an analytical reasoning aid. R4 describes it as ”a kind of Ctrl+F” (R4) for locating where in the transcript a given question can be answered, and EX also predicts the chat will support the user’s memory, with the interrogator first drafting from memory and only turning to the chat when a specific detail needs to be recovered. R1 diverges by treating the chat as a fallback rather than a standalone feature, arguing that a sufficiently strong *findings panel* would make it largely redundant, so the chat earns its place by compensating for gaps in the analysis rather than by adding independent value.

5.2.5 Critical Feedback and Operational Prerequisites

Critical feedback at the UX layer was sparse, and the concerns that surfaced were issues that the walkthrough could not itself evaluate. The more developed architectural critiques exceeded the feature-level frame of the interview guide and are treated as separate emergent themes in the sections that follow. EX separates the visible functionality from requirements beyond what is presented in the demo, arguing that ”if we only look at the functionality of what we are seeing now, it can be used operatively now. That is not the problem. The difficulties would be in whether there are any security-related gaps” (EX). R4 anchors a related concern in the hardware that an interrogator carries, wondering whether a laptop at a field deployment could run the models fast enough to be useful. The two concerns operate at different layers, infrastructure security and compute capacity, but both locate the deployment question beyond the UX where the evaluation was scoped.

R1 raises requirements of a different kind, methodological rather than technical. Before deploying FENRIR in practice, R1 argues that the organization needs ”some kind of systematic method for working with this tool, how does it integrate into a larger work methodology?” (R1). The remaining substantive critiques exceeded the scope of the interview guide's feature-level structure. They are therefore addressed as emergent themes in Section 5.2.6.

5.2.6 Themes Beyond the Interview Guide

Several concerns surfaced spontaneously during the sessions and fell outside the deductive structure of the interview guide. These emergent themes reach past the interview guide as well as the current architectural scope of the system.

Demand for Multi-Source Scope

Four of the five participants independently raised architectural requirements outside the prototype's scope. The following findings and ideas reach beyond the current version of the prototype. EX asked for a shared PIR master file that can be reused across interrogators rather than re-entered in each session, so that ”I have a master file where I can write everything once, and then everyone can import it” (EX). R3 wished for conflict detection between multiple

interrogations, because ”identifying conflicts in people’s accounts or stories, that is spontaneously what I think would be of great value” (R3). R2 proposes being able to attach reports from adjacent interrogations, so that conflicts can be detected across parallel timelines rather than only within a single transcript. R3 raised a similar idea but extended it to reports of a different kind, such as those from drone operators or signals intelligence.

AI Error Mitigation and Feedback Loops

Participants respond to AI unreliability with three mechanisms. They want errors highlighted in place, traced back to their source, and corrected without leaving the workflow. The responses converge around transparency during reasoning, and around the ability to re-run and correct the analysis. R2 names visibility of the AI’s reasoning as the one missing feature of the current prototype. The prototype currently exposes the model’s conclusions in the *output panel*, but not the intermediate reasoning that produced them. R4 proposes a correction loop routed through the *chat panel*, where the interrogator tells the system ”this is not correct, this is how it actually is, and it updates the content” (R4). R1 argues that, depending on where the root problem sits, the interrogator may need to re-run the full pipeline because errors risk propagating across different parts of the analysis. The feature he points at is therefore a selective re-execution of the affected pipeline components rather than a re-run of individual findings. Together, the participants express a need for feedback loops at both a high and a low level, with the ability to re-run either the full pipeline or specific parts of it.

A related concern was raised about the risk that contained AI errors spread to future sessions. R4 observes that ”the more AI steps, the more risk of errors along the way. It is a bit like the telephone game” (R4). He locates the risk in the chain of reporting across multiple interrogations rather than within a single analysis. When AI errors remain invisible, they can propagate into future analyses and carry major consequences. The observation points to a need for mechanisms that mitigate AI errors in a wider architectural context.

Information Density as a Design Constraint

Another emerging theme concerns the need for denser AI-generated titles. R1 argues that the current finding titles are too long and risk creating unnecessary noise for the user, since many findings are presented at once. During the walkthrough, R1 read aloud a sentence from a finding written in a descriptive

way. He then suggested that the line should be condensed to a single bullet point using the interrogators' compact technical shorthand instead (R1). R1 identifies the intelligence service as the primary reader and reasons that the interrogator's role is to grasp a large amount of information and condense it into a readable protocol without producing a wall of text. R2 converges on this topic and frames the architectural problem, because ”too little and you lose credibility, too much and you just do not read it” (R2).

Traceability Versus Anonymisation at Export

Section 5.2.2 highlights traceability as the main trust prerequisite, since it lets the interrogator link a protocol claim back to the originating transcript line. At export, however, this link works against the obligation of source protection, because the same connection can expose the identity of the interrogated person. EX raises this tension during the walkthrough, saying that ”I do not want the source's name to appear anywhere, but rather a contact number” (EX), and explains the need of source protection, meaning that the interrogated person remains anonymous upon protocol export. This is an architectural finding that highlights a mismatch in how the prototype is developed and the requirement of having source protection originating from international human rights.

6 Discussion

This chapter interprets the evaluation in three movements. It first answers each research question and then responds to the main research question. The chapter then reflects on the methodology and examines the ethical implications that arose during the development of FENRIR. The discussion ends with translating the evaluation's themes into concrete directions for future work.

6.1 Answering the Research Questions

This section answers the research questions that framed the study. The main research question asks how an AI-supported analytic workflow can be designed to reliably extract and structure relevant information from transcribed interrogations into protocols while preserving human oversight and operational validity. Three sub-questions refine that goal. RQ 1.1 asks for which analytic tasks FENRIR can assist in order to reduce the interrogator's work load. RQ 1.2 asks how AI-generated output should be surfaced in the user interface so that the interrogator remains the accountable decision-maker. RQ 1.3 asks how interrogators assess the workflow and operational usefulness of AI-extracted information when integrated into the process.

6.1.1 Analytic Tasks and Cognitive Load Redistribution

The participants described that cognitive load can be reduced when using FENRIR, especially in the retrieval and structuring of information needed for writing a protocol. However, topics emerged discussing whether the load was redistributed rather than removed, due to the additional need to verify AI-generated information. EX framed the benefits mainly as the potential to

shorten analysis time, while another participant explained that AI assistance freed attention for other tasks.

As AI takes over otherwise manual work, new tasks appear for the user. The need to verify AI-generated information is the predicted cost of that redistribution. Four of the five participants raised the added need for quality assurance, and R1 articulated this as a cost-benefit question of whether the workflow saves more work than it creates. Cognitive-offloading research expects this pattern. When interrogators hand tasks to a tool, they later remember the material less well and judge their own grasp of it less accurately (Grinschgl & Neubauer, 2022). Recent work on AI coding assistants, though drawn from a different domain, suggests the mechanism plausibly transfers. It finds that verification effort is measurable as a distinct behavioral load that accumulates across tasks and partially mediates increases in stress and fatigue (Fan et al., 2026). This positions verification as a real cognitive cost rather than a costless byproduct of AI assistance. Whether the net load on the interrogator is higher, lower, or comparable to fully manual analysis remains undetermined, since the evaluation did not include this. R2 speculated that tired interrogators check AI output less carefully, no matter how disciplined they try to be. This marks the point where verification effort becomes verification fatigue, and the automation-bias literature treats this fatigue-driven acceptance as the core safety risk (Cummings, 2017). The saving the workflow offers is therefore conditional.

The AI analysis sees only the semantic layer of the interrogation, whereas embodied cues such as body language, tone, and stress remain part of the interrogator's subjective experience. Objective extraction and PIR-mapped findings were identified as having the greatest potential. Participants discussed that some evaluations and conclusions should remain a human task, since not every aspect of an interrogation is verbal. Whether verification fatigue harms interrogators' skill in the long term is a speculative concern that the evaluation cannot conclude on.

6.1.2 Surfacing AI Output for Accountable Decision-Making

RQ 1.2 asks how AI output should be surfaced so that the interrogator remains the accountable decision-maker. Traceability emerged as one of the most important features and was framed as a must-have for intelligence work. Three participants went further and demanded visibility into the model's reasoning

chain, not only its sources. Citations show where a claim came from, whereas reasoning shows how the model arrived at its conclusions. The participants' opinions explain that citation alone is not enough, because a faithful quotation can still sit on top of flawed reasoning. The confidence labels the prototype attaches to findings raise a related question. FENRIR claims low, medium, or high confidence, whereas the formal grading scale used by interrogation leaders in operational practice is more granular and reserves its highest level for claims backed by two independent observations. This is a definitional mismatch in how confidence is expressed to the reader.

The *initial thoughts* step, where the interrogator writes their ideas before any AI analysis appears, is designed to offload the user's memory and to mitigate anchoring bias. This feature was greatly appreciated by the participants, as it was an opportunity to ground their own fundamental understanding of the interrogation before seeing any AI analysis. However, since the evaluation was simply a walkthrough and not a realistic usability test, the real impact of the feature is difficult to conclude.

When inspecting AI output takes more effort than accepting it, interrogators default to acceptance and the interface ends up producing the automation bias it was meant to counter. The protocol draft that is currently generated would benefit from an accept or discard functionality to increase the friction to confirm generated content. This was a feature identified early in the design process, but was not included in the implementation.

All participants agreed that the interrogator is solely responsible for the resulting protocol. During drafting, adding visible markers that separate AI-generated from interrogator-written text could help the interrogator in the writing and verification process. However, on export, all content is formatted uniformly and the user becomes the author of it. Keeping the authorship distinction on the exported document was seen as a risk to its perceived credibility in the eyes of the reader. The implication is that the tool for writing the protocol does not matter as long as the author takes full responsibility for it. EX's observation about replacing the source's name with a generic contact number shows that traceability has multiple important aspects. Transcript links that build trust during drafting must be removed on export, because international human rights require the interrogated person to remain anonymous in the released protocol. This issue can be easily addressed and occurred due to a gap in the requirements specification.

6.1.3 Interrogator Assessment of Workflow and Operational Usefulness

RQ 1.3 asks how interrogators assess the workflow once AI-extracted information is integrated into their process. The participants discussed the *protocol editor* and the *PIR-questions panel* as the two essential parts of the workflow. The theory of Recognition-based decision making and the Orient phase of the OODA loop explain why. Both features help the interrogator match the case against remembered mental patterns to simulate the draft writing process, which is the core of expert judgment in Orient (Klein, 1999; Richards, 2020). The *chat panel* plays a different role, since it behaves like a search bar the interrogator falls back on after recognition has already started, more similar to the Observe step than to Orient. R1 stated that the *chat panel* could be redundant once the *findings panel* held enough generated information. The two panels therefore drive the interrogator's judgment work, while the *chat panel* complements rather than drives the workflow. However, exactly how an interrogator will use the product is difficult to assess without a real test in a realistic scenario.

An operational critique concerned text density. Participants read the AI output as noise compared with the compact-text norm that they are trained to use. AI can generate coherent and fluent summaries (Y. Zhang et al., 2025). However, most titles were long because they aimed to be grammatically correct, and when displayed in a list they created a wall of text. R1 gave an example where a long descriptive sentence was condensed using a domain-specific formatting technique used in operational contexts. Information density therefore becomes a significant design goal to improve readability and reduce cognitive load. The walkthrough cannot determine how the workflow holds in realistic scenarios with time pressure, which makes it difficult to judge operational usefulness and this remains speculative.

6.1.4 Designing for Reliable Extraction with Human Oversight

The main research question asks how such a workflow can be designed to reliably extract and structure relevant information from transcribed interrogations into protocols while preserving human oversight and operational validity. A potential result of FENRIR is a task allocation that keeps objective extraction on the AI side and leaves evaluation and judgment to the interrogator, who

remains the final decision-maker. A frictional interface has the potential to keep the human in the loop even though many of the tasks are carried out by AI. The *initial thoughts* step and the chain of traceability make verification a deliberate step in the workflow. Alongside these, the formal rules used by interrogation leaders for separating fact, comment, and assessment shape both the AI drafts and the UI layout. AI assistance therefore has the potential to aid protocol writing while keeping source traceability visible at every step. How the workflow is perceived in use, and whether it saves time under operational conditions, remains unknown within the limits of the evaluation methodology. The design drew on knowledge and requirements distilled from a domain expert in investigative interviewing, but deeper analysis across more participants and realistic scenarios would enrich future iterations. The project should therefore be explained as a functioning prototype rather than a finished product, with methodological reflections and future directions developed in the sections that follow.

6.2 Methodological Limitations

Late in the project, the evaluation shifted from a full usability study to a product-fit evaluation, which narrowed its scope. The Double Diamond places Deliver as the phase where a focused method tests whether the developed prototype actually fits the user's work (Design Council, 2024). Because the study ran as a guided walkthrough it was only possible to evaluate the overall impressions and most results remain speculative regarding the potential of the workflow. Claims about efficiency and learnability under operational pressure therefore fall outside what this evaluation can support. The methodological consequence is that the study addresses conceptual fit instead of real operational use. Interviewing participants while the prototype remained on screen, and without alternative designs to compare against, may have framed responses around the choices they saw rather than ones they might otherwise have imagined. The authors reflected on this risk during the sessions and explicitly invited critique and alternative ways of doing the same work. The framing constraint, however, cannot be eliminated by the interview guide alone.

The small sample size reflects the difficulty of gathering domain-specific users. Practicing interrogators were unavailable due to operational schedules and classification constraints. The other four participants therefore carried general operational context, but not the specific investigative-interviewing

practice of the primary users. The co-design activity treated domain experts as co-creators because their situated knowledge is only accessible when they participate in the design work (Sanders & Stappers, 2008). The session was originally planned to help prepare a first evaluation plan that would review AI output against real interrogation protocols from the studied organization. That plan later changed to a guided walkthrough, but the insights gained from the co-design session remained valuable for other parts of the design work.

The pilot of the main evaluation is also a stakeholder in the FENRIR project and participated in an early design interview during the design process. This dual role carries a known methodological risk, because an informant already invested in the project tends to frame their observations in ways that support it, which could bias both the design choices drawn from the early interview. The risk is also affected by the fact that the supervisor who holds knowledge in the domain is also a stakeholder, so fully independent domain feedback was not available within the team. The authors argue that the design decisions and directions were grounded in external sources rather than only from the interview and supervisor's feedback. The stakeholder participant was retained only as a pilot in the main evaluation, with their session excluded from the reported findings and used only to refine the walkthrough and interview guide.

Combining inductive and deductive coding kept interpretation flexible during analysis. The combination of a small sample and a walkthrough-based design keeps the certainty of any conclusion low. The themes therefore function as exploratory in purpose to identify where the design concept holds and where more work is needed. Their primary value is to sharpen the questions and design decisions that future iterations of FENRIR will need to test under more operational conditions. The evaluation of output-quality against human-written protocols and a realistic usability test were moved to a future planned study where ecological validity can be tested. However, since this was scoped as a study beyond the time frame of the authors work, it could not be included in the thesis. The strongest claim this reflection carries is that the Deliver of the prototype was purposed mainly for evaluating product-fit.

The project's broad scope including research, design process, system implementation, and evaluation meant a trade-off in the depth in certain areas. The authors chose to build rather than a mock a prototype, and that choice narrowed some design activities. Design iterations were fewer, no separate UX evaluation was conducted, and the co-design follow-ups that would have tested alternative layouts with practicing interrogation leaders were not scheduled. The working prototype, however, enabled a realistic walkthrough where

participants evaluated the fully functioning product from transcript upload to protocol draft. A coded prototype also raises the cost of change compared with low-fidelity alternatives, because each adjustment requires implementation rather than a sketching a mock up interface. Late insights about user needs must then compete with the rework they imply, which limits how much Requirements Engineering can iterate.

6.3 Future Work

A meaningful finding from the evaluation is that the interrogator should write down their own interpretations from the interrogation before any AI output appears. Two participants proposed extending the current free-text *initial thoughts* step with a more structured input form. This could be done with the purpose to separate fact from subjective comments. A future version could give the option to link the notes to specific PIRs. For example, this would give the possibility to comment or add facts about what is already known about a PIR based on what they interpreted during the interrogation or gained from other sources. To extend this further, the interrogator could also import previous interrogation protocols or intelligence protocols of other kinds. That extended context would then feed the initial AI analysis.

A refinement to the architecture concerns how interrogators correct AI output once it has been generated. Low-level gaps surfaced in two places during the evaluation. The prototype did not display the model's reasoning, so an error can only be caught in the final output. It was proposed a correction loop through the *chat panel*, where the interrogator states the correction in natural language and the relevant finding or protocol segment updates accordingly. Both additions could be integrated into the current interface. The chain-of-thought could be displayed near its corresponding finding. At the architectural level, if the fault sits in an early pipeline stage, correcting the resulting finding does not address the root problem that the finding derived from. A selective re-analysis such as per-finding or diagnosing mechanism, could address such problems. To summarize, the AI analysis can never be perfect, so the product must have a feedback loop designed to adjust based on user feedback and adjust accordingly.

A new design concept completely independent of the current workflow is a guided drafting process that sits between analysis and text writing. Once the AI finishes its analysis, the system would display a short sequence of contextual questions before any text is generated. The questions would be spe-

cific to each case, derived from the AI's findings and the interrogator's initial thoughts. For example, the system could ask the interrogator which two conflicting statements the protocol should treat as most probable, what findings should be prioritized into the draft and which held back, which confidence label to apply to a given claim, and whether a weak conclusion belongs as assessment or comment. The answers become the scaffold the AI analysis is based on, so the resulting draft creates output that the interrogator has already committed to. This new design concept is an alternative way to make a *frictional design* of the protocol writing workflow.

The previous concept could be combined with an accept-or-decline feature to further increase friction to confirm AI-generated content. As the AI produces the protocol segment by segment, the interrogator could see each text segment alongside the transcript line and the finding its based on, and confirm or reject it before the next segment is written. The interaction would resemble a walkthrough of the draft as it is composed, with traceability visible during the writing, rather than having to verify a fully completed draft in retrospect. This interaction would add quality assurance into the user flow, which probably would be more time efficient than the current setup. Another aspect of this concept is that instead of generating all sorts of perspectives and findings, which the current prototype does, this workflow could tailor the analysis based on the authors decisions and demands.

Participants asked for more features beyond the version shown during the demo. They requested shared PIR master files between projects, conflict detection across interrogations, and inclusion of multiple adjacent reports. Using the Distributed Cognition theory, it can be interpreted that the interrogator and the FENRIR system will work as one unit over a longer period of time and several interrogations (Hollan et al., 2000). As noted in earlier explanations that at least two independent sources are needed to confirm a claim as likely, the possibility to include multiple sources of information is a must have feature to enhance the system's analytical capacity. However, this additional ability has risks when protocols include disjunct information. This could result in propagating errors between sessions. For example, two independent interrogators might draw completely different conclusions, then when the AI uses the two it must determine which to accept and discard. Such a multi source context for AI will create alternative truths and increase the complexity in the analytical effort.

What this thesis delivered is a working prototype rather than a final system. A *full pipeline* run takes roughly 7 minutes for transcription and 2 hours for analysis on a consumer computer for an 18 minute long transcript with

the default model stack. These timings refer to the most complete *full analysis* mode, which includes full and final protocol drafting. The lighter *quick start* and *standard* modes would complete faster and would likely be the preferred choice in practice. The pipeline is sequential by default, with a single thread so that only one project analysis runs at a time on the shared local hardware. Entity extraction and the creation of the relational database is the most time consuming part. No empirical benchmarking of pipeline performance or formal output quality evaluation was carried out during the thesis, since the evaluation scope was product-fit rather than system performance. The timings reported above are observations from development runs. A thorough benchmarking and optimization effort is therefore needed. The current architecture that was developed is therefore not concluded as the final one and more architectural work is needed. Finally, the prototype suggests the workflow can exist on a local machine, but it does not yet suggest that the system meet the operational response times needed.

7 Conclusion

The primary research question asked how a workflow can reliably extract information from transcribed interrogations while keeping the interrogator accountable for the resulting protocol. The design process resulted in a functional prototype to answer this. The evaluation summarizes that the biggest potential gain for the protocol writing workflow is to let AI assist in retrieving and structuring of information. The interface was designed with the goal to be deliberately frictional, so verification is a step the interrogator has to take rather than a step that they can skip. However, more work remains to achieve a complete product with the human always in the loop. The professional norms of separating fact, comment, and assessment that interrogation leaders follow shape both the AI drafts and the layout the interrogator reads. Under these conditions, AI output becomes material the interrogator inspects and verifies.

The tasks FENRIR takes on are semantic. It extracts entities and events, reconstructs timelines, maps findings to PIRs, and produces a first protocol draft. Interpretation and authorship remain with the interrogator. Output reaches the interrogator through inline traceability, an *initial thoughts* step, and a retrieval-grounded chat using semantic search and an entity database. Participants in the evaluation treated the *protocol editor* and the *PIR-questions panel* as the most important features for the workflow. They also named three specific gaps. The prototype labels confidence as low, medium, or high, which does not map onto the more granular grading scale that interrogation leaders use professionally. The AI-generated texts were sometimes too long and descriptive for an information dense interface. And the system has no scope beyond a single interrogation per project, which could matter when the same source is interviewed more than once.

The evaluation was scoped to product fit, with one fabricated transcript and five participants of whom only one was a domain expert in investigative interviewing, so claims about cognitive offloading and time saving remain perceptual until a field study can test them. From the evaluation, a few directions

are worth pursuing. A multi-session architecture could let the system detect conflicts across interrogations with the multiple sources. Better readability in AI-generated content by calibrating and evaluating prompts. Finally, a realistic study with interrogators on offline hardware could test the claims this walkthrough could not. The underlying argument is that verifying AI output should cost the interrogator less effort than accepting it, and that signing the intelligence product must remain the interrogator's act.

References

Artwohl, A. (2002). Perceptual and memory distortion during officer-involved shootings. *FBI L. Enforcement Bull.*, *71*, 18.

Belton, I. K., & Dhami, M. K. (2020). Cognitive biases and debiasing in intelligence analysis. In *Routledge handbook of bounded rationality* (pp. 548–560). Routledge.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77–101.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Chkroun, M., & Azaria, A. (2024). Autonomous agents for interrogation. *2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI)*, 686–693.

Cummings, M. L. (2017). Automation bias in intelligent time critical decision support systems. In *Decision making in aviation* (pp. 289–294). Routledge.

Design Council. (2024). *The double diamond*. Retrieved January 29, 2026, from https://www.designcouncil.org.uk/our-resources/the-double-diamond/

Fan, G., Liu, D., Pan, L., & Zhang, R. (2026). When help hurts: Verification load and fatigue with ai coding assistants. *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3772318.3791176

Figma. (2025). *Introducing figma make: A new way to test, edit, and prompt designs*. Retrieved January 29, 2026, from https://www.figma.com/blog/introducing-figma-make/

Garcia, J. P., Grilo, C., Domingues, P., & Miragaia, R. (2025). Intu-ai: Digitalization of police interrogation supported by artificial intelligence. *Applied Sciences*, *15*(19), 10781.

Giorgianni, D., Vrij, A., Leal, S., & Deeb, H. (2025). The effect of the interviewer's cognitive load on the quality of the forensic interview. *The European Journal of Psychology Applied to Legal Context*, *17*(2), 101–110.

Grinschgl, S., & Neubauer, A. C. (2022). Supporting cognition with modern technology: Distributed cognition today and in an ai-enhanced future. *Frontiers in Artificial Intelligence*, *5*, 908261.

Gudjonsson, G. H. (2012). *Investigative interviewing*. Willan.

Hanway, P., Akehurst, L., Vernham, Z., & Hope, L. (2021). The effects of cognitive load during an investigative interviewing task on mock interviewers' recall of information. *Legal and Criminological Psychology*, *26*(1), 25–41.

Heuer, R. J. (1999). *Psychology of intelligence analysis*. Center for the Study of Intelligence.

Hollan, J., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *7*(2), 174–196.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, *43*(2), 1–55.

Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. (2018).

Khamlichi, N., & Benahmed, N. (2025). Anchoring bias: A literature review on cognitive mechanisms and decision-making impact. *IJTM International Journal of Trade and Management*, *2*(4), 195–205.

Klein, G. A. (1999). *Sources of power: How people make decisions* (2nd ed.) [PDF is the 1999 second paperback edition of the 1998 first edition]. MIT Press.

Lauesen, S. (2002). *Software requirements: Styles and techniques*. Pearson Education.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, *33*, 9459–9474.

Li, T., Zhang, G., Do, Q. D., Yue, X., & Chen, W. (2024). Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Li, Y., Miao, Y., Ding, X., Krishnan, R., & Padman, R. (2025). Firm or fickle? evaluating large language models consistency in sequential interactions. *arXiv preprint arXiv:2503.22353*.

Liu, J., Huang, J., Zhou, Y., Li, X., Ji, S., Xiong, H., & Dou, D. (2022). From distributed machine learning to federated learning: A survey. *Knowledge and information systems*, *64*(4), 885–917.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, *12*, 157–173.

Maaz, S., Palaganas, J. C., Palaganas, G., & Bajwa, M. (2025). A guide to prompt design: Foundations and applications for healthcare simulationists. *Frontiers in Medicine*, *11*, 1504532.

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Oxburgh, G., Myklebust, T., Fallon, M., & Hartwig, M. (2023). *Interviewing and interrogation: A review of research and practice since world war ii*. Torkel Opsahl Academic EPublisher.

Primer, A. T. (2009). Structured analytic techniques for improving intelligence analysis. *CIA Center for the study of intelligence*.

Richards, C. (2020). Boyd's OODA loop. *Necesse*, *5*(1), 142–165.

Romeo, G., & Conti, D. (2025). Exploring automation bias in human–ai collaboration: A review and implications for explainable ai. *AI & Society*. https://doi.org/10.1007/s00146-025-02422-7

Rubin, J., & Chisnell, D. (2008). *Handbook of usability testing: How to plan, design, and conduct effective tests*. John Wiley & Sons.

Sanders, E. B.-N., & Stappers, P. J. (2008). Co-creation and the new landscapes of design. *CoDesign*, *4*(1), 5–18. https://doi.org/10.1080/15710880701875068

Sharp, H., Rogers, Y., & Preece, J. (2019). *Interaction design: Beyond human-computer interaction* (5th). John Wiley & Sons.

Snyder, C. (2003). *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann.

Sweller, J. (2011). Cognitive load theory. In *Psychology of learning and motivation* (pp. 37–76, Vol. 55). Elsevier.

Tribbels, S., & Michels, M. (2025). Validity and effectiveness of interrogation techniques: A meta-analytic review. *Military Psychology*, *37*(2), 127–137.

United Nations. (n.d.). United nations sustainable development goals. Retrieved January 20, 2026, from https://sdgs.un.org/2030agenda

Vetenskapsrådet. (2024). *God forskningssed 2024* (tech. rep.) (Retrieved February 3, 2025). Vetenskapsrådet. Stockholm. https://www.vr.se/analys/rapporter/vara-rapporter/2024-10-02-god-forskningssed-2024.html

Vrij, A., Meissner, C. A., Fisher, R. P., Kassin, S. M., Morgan III, C. A., & Kleinman, S. M. (2017). Psychological perspectives on interrogation. *Perspectives on Psychological Science*, *12*(6), 927–955.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824–24837.

Whitesmith, M. (2020). *Cognitive bias in intelligence analysis: Testing the analysis of competing hypotheses method*. Edinburgh University Press.

Yan, L. (2025). From passive tool to socio-cognitive teammate: A conceptual framework for agentic ai in human-ai collaborative learning. *arXiv preprint arXiv:2508.14825*.

Zelik, D. J., Patterson, E. S., & Woods, D. D. (2018). Measuring attributes of rigor in information analysis. In *Macrocognition metrics and scenarios* (pp. 65–84). CRC Press.

Zhang, Y., Jin, H., Meng, D., Wang, J., & Tan, J. (2025). A comprehensive survey on automatic text summarization with exploration of llm-based methods. *Neurocomputing*, 131928.

Zhang, Z., You, W., Wu, T., Wang, X., Li, J., & Zhang, M. (2025). A survey of generative information extraction. *Proceedings of the 31st International Conference on Computational Linguistics*, 4840–4870.

Appendices

A Prioritization of Features

Priority	Feature	Layer	Description	Primary User Goal
MUST	*fully autonomous protocol*	SYS	System generates documentation (Person Form & Interrogation Report) without human intervention.	Effectiveness (used to evaluate human vs AI protocol quality in thesis)
MUST	*text slicer*	SEM	Segments data by topic shifts to preserve context.	Import & process files, avoid "blank page."
MUST	*entity extractor & resolver*	SEM	Pulls hard facts and links aliases (e.g., linking "The Boss" to "Andersson"). Semantic tag: automatic tagging of names, locations, weapons, organizations, equipment, unit identifiers.	Identify key intel (Names, Weapons, Aliases, Equipment, Units).
MUST	*audit log*	SYS	Records metadata (model, prompt, timestamp) for every AI output to maintain a chain of custody.	Trust & transparency, scientific best practices.
MUST	*prompt separator*	SEM	Keeps instructions, background facts (PIR), and raw transcript data isolated to prevent model confusion.	Prevent AI distortion, isolate raw data.
MUST	*semantic search*	SEM	Uses the vector database to find concepts and feelings rather than just keywords.	Fast extraction of concepts/feelings.
SHOULD	*interviewer behavior feedback*	MEA	Feedback on questioning style to mitigate confirmation bias (e.g., leading questions).	Mitigate confirmation bias & effective questioning.
SHOULD	*time stamping*	SEM	Linking events to specific timestamps or relative timeframes (e.g., "Two hours after the meeting"). Behavioral patterns	Automatic protocols with timestamps.
SHOULD	*fairness check*	MEA	Ensure that AI's summary has neutral language. Statements must include argumentations supporting the statement and the potential risk.	Compliance with ethics, neutral language.
SHOULD	*implicit vs. explicit*	MEA	Differentiating between what was stated (Observation) and what the AI/Human is inferring (Assessment).	Identify vague statements, distinguish fact from interpretation.

Priority	Feature	Layer	Description	Primary User Goal
SHOULD	*undo/redo*	SYS	System functionality to revert or repeat edits.	Ensure ease of use. Easy to revert mistakes.
COULD	*timelines*	MEA	Converts text into timelines to highlight overlaps or gaps. Ensure that unsure parts are clearly communicated.	Highlight important events, overlap with other's narrative or gaps in the story
COULD	*ask mode*	NAR	Acts as a critical AI partner, generates viewpoints and arguments.	Human-AI co-operation
COULD	*sentiment & stress analysis*	MEA	Identifying shifts in tone when specific entities (weapons/names) are mentioned.	Identify tone shifts regarding specific entities.
COULD	*ask questions to user*	NAR	Agent asks: "can we assume they are the same person?" if aliases appear across pages.	Manual resolution of assumptions/conflicts.
COULD	*deeper re-analyzer*	SEM	Allows user to trigger a high-res "second look" (Discuss or Re-generate). Maybe as inline text-highlight (Action 1: Discuss, Action 2: Re-analyze).	Detailed analysis of specific parts.
COULD	*multi-perspective summarizer*	NAR	Creates neutral briefs focused on arguments and counter arguments	Create neutral briefs (Timeline vs. Discrepancy).
COULD	*automation bias guardrails*	SYS	System must highlight why it flagged something as a contradiction to prevent blind trust.	Legal safety, maintain Human-in-the-loop.
COULD	*layer filter*	SYS	Ensuring controlled cognitive load by filtering the UI. Sequential process (SEM, MEA, NAR, SYS) (…then create protocol)	Ground truth to settle AI/Human disputes.
WON'T	*audio playback engine*	SYS	Raw audio file playback for context or manual corrections to transcript.	Ground truth to settle AI/Human disputes.

B FENRIR System Screenshots

The following figures present screenshots from the final FENRIR prototype, illustrating the key screens encountered during a typical analysis session.

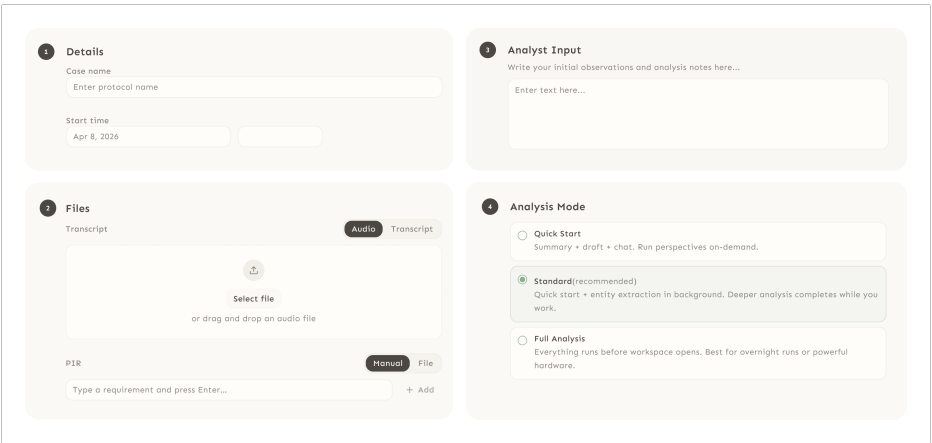

Figure B.1: New protocol creation wizard

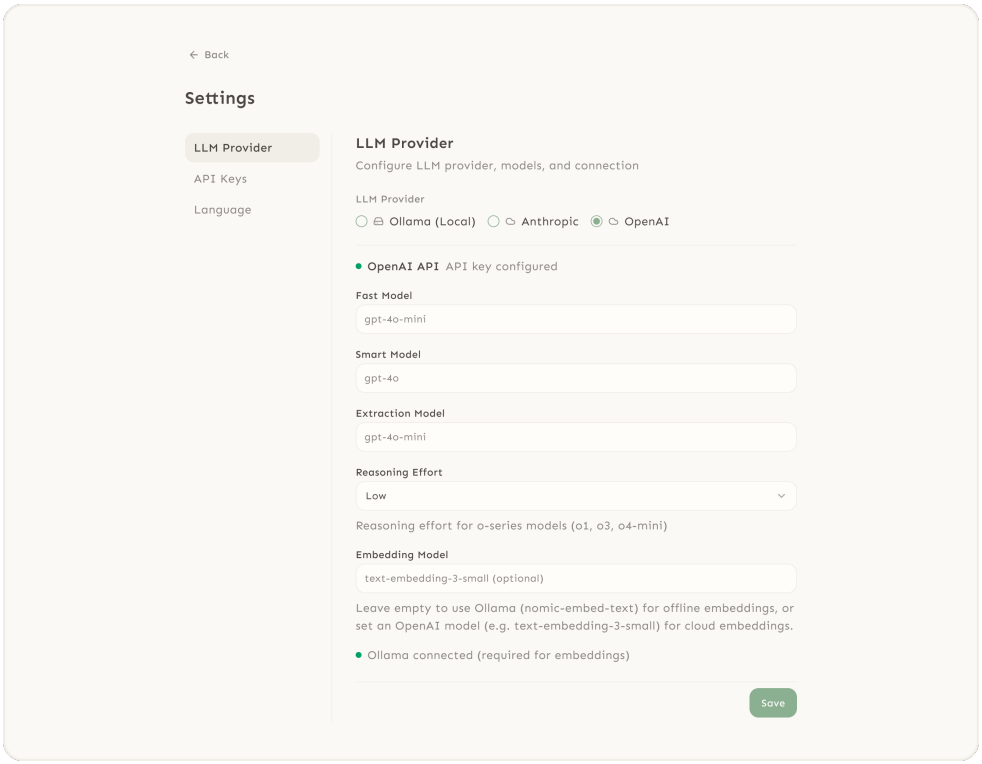

Figure B.2: Settings page for LLM provider configuration.

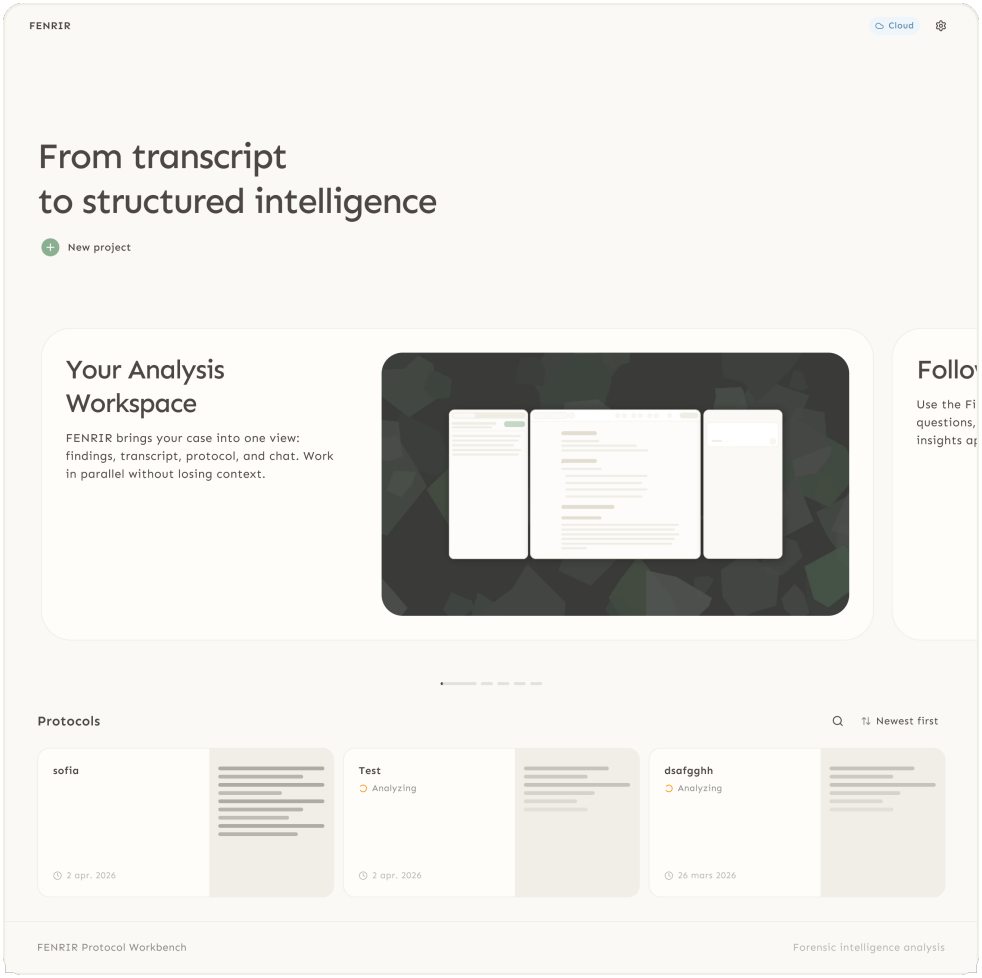

Figure B.3: FENRIR onboarding page

C Interview Guide

Anonymisation note: The interview guide below is in Swedish, like the original, but has been anonymized. Explicit institutional and operational terminology elsewhere in the guide has been generalised so that the studied authority is not identifiable.

Förutsättningar

Observation under genomgången. Innan intervjun noteras vad deltagaren frågar om spontant, vad som verkar förvirra eller störa dem, vad de verkar ignorera, och deras reaktioner när olika funktioner visas upp.

Inspelning och anteckningar. Sessionen spelas in (med samtycke). En person leder intervjun, en person skriver anteckningar och hanterar tekniken. Inspelningen transkriberas och analyseras med induktiv tematisk analys. Om deltagaren inte vill bli inspelad förlitar vi oss på anteckningar. Total intervjutid är ca 35-40 minuter per deltagare.

Tekniskt. Transkriptet `demo-transcript.txt` är ett fabricerat exempeltranskript. Språkmodellen körs online via OpenAI för att minimera friktion och ge snabbare laddning under demot, med en modell av motsvarande intelligens som den offline-baserade GPT-OSS 20B. Detta påverkar inte utvärderingen då studien inte bedömer AI-outputens kvalitet. Modeller (OpenAI API-namn): `o3-mini` (reasoning_effort: "high") för sammanfattning, analys, perspektiv och drafting; `gpt-4.1-mini` för per-chunk entity/event-extraktion; `gpt-4.1-nano` för textnormalisering och talaridentifiering som fallback; Nomic-Embed-Text (lokal via Ollama) för embedding.

Introduktion (ca 3 min)

Innan inspelningen startar:

"Tack för att ni tar er tid. Vi är två masterstudenter vid LTH som skriver

examensarbete om hur AI-stöd kan utformas för att hjälpa analytiker med protokollskrivning. Vi har byggt en prototyp och vill idag visa det för er och sedan ställa frågor om hur väl det passar ert faktiska arbetsflöde.

Det här är en designutvärdering, inte ett test av er. Det finns inga rätta eller felaktiga svar. Vi är intresserade av era ärliga reaktioner, både positiva och negativa.

Intervjun tar ca 35-40 minuter: först en genomgång av systemet, sedan frågor.

Vi vill gärna spela in samtalet för att kunna transkribera det efteråt. Inspelningen används bara inom examensarbetet och raderas efter analys. Ni kan avbryta när som helst och be oss radera materialet. Är det okej att vi spelar in?"

Produktdemo (ca 15-30 min)

1. Startsida: projektöversikt. Inled med "Då börjar vi. Du får väldigt gärna tänka högt, avbryta oss och ställa frågor när du vill." Ge deltagaren bakgrund: ett förhör är ett samtal i syfte att hämta in relevant information från en källa, byggt på systematisk informationsinhämtning snarare än konfrontation eller tvång. Förhörsledaren arbetar utifrån ett antal **Priority Intelligence Requirements** (PIR), specifika informationsbehov formulerade av beställaren inför förhöret, som styr vilken information som ska samlas in och bildar ryggraden i dokumentationen. Resultatet sammanfattas i ett **protokoll**, det formella underlag som förs vidare till den mottagande funktionen, med en objektiv beskrivning av vad som sades samt förhörsledarens egna kommentarer och slutsatser markerade som sådana. FENRIR är ett stöd för att skriva detta protokoll: förhörsledaren laddar upp transkriptet, skriver in sina PIR:er, och systemet kör en AI-analys.

Visa startsidan inkl. onboarding (korten med respektive video), sedan listan av befintliga projekt.

"Det vi ska visa er idag är FENRIR, ett verktyg vi byggt för att stödja förhörsledare i protokollskrivningen efter ett förhör. Man laddar upp sitt transkript och sina informationsbehov, och systemet analyserar materialet med hjälp av AI-modeller som körs helt lokalt på datorn, ingen data lämnar maskinen.

Att använda AI-modeller på datorn offline tar lång tid, dock kommer vi under detta demo använda online-metod för att påskynda presentationen. Därför kan delar av AI-analyserna i ett verkligt offline-scenario ta längre tid än det uppfattas i detta demo.

> Detta är startsidan. Varje projekt motsvarar ett förhör: ett transkript med tillhörande protokoll. Man kan söka, sortera och se status för varje projekt."

Visa att projekt har olika statusar (klart, pågående, etc.).

2. Skapa nytt projekt: wizard-flöde. Klicka "Nytt projekt" och gå igenom wizarden steg för steg, med inledningen "Nu ska vi visa hur man skapar ett nytt projekt."

Steg 1, Detaljer. "Man börjar med att ange namn och datum/tid för förhöret."

Steg 2, Filer. "Här laddar man upp intervjun som en ljudfil eller en transkriberad textfil. Man kan förhandsgranska det innan man går vidare. PIR:er eller SIR skrivs in direkt i gränssnittet: man kan lägga till och ta bort." *Designmotivering:* "PIR:erna styr hela analysen, de avgör vilken evidens som hämtas och vilka fynd som genereras. Genom att strukturera dem redan här kan systemet koppla varje fynd tillbaka till ett specifikt informationsbehov." En fiktiv scenariotext med tillhörande PIR och SIR användes live i wizarden under demot för att ge deltagaren konkret kontext; scenariot och dess informationsbehov återges inte här av anonymiseringsskäl.

Steg 3, *analytikernoteringar*. "Innan AI-analysen startar skriver analytikern sina egna initiala observationer: bakgrundsinformation, hypoteser, farhågor." *Designmotivering:* "Det här steget är avsiktligt, tanken är att man ska bilda sig en uppfattning innan AI-output visas för att minska risken för förankringsbias, men också ge förhörsledaren möjlighet att skriva ner alla idéer och tankar om de kommer direkt från förhöret."

Steg 4, Analysläge. "Man väljer mellan tre lägen: *snabbstart* för tidskritiska situationer, där analysen är färdig på ca 1 minut utan djupa perspektiv; *standard*; och *fullständig analys*." *Designmotivering:* Systemet kör lokala LLM-modeller helt offline, ingen data lämnar datorn. Lokala modeller är långsammare än molntjänster, så tre lägen låter analytikern välja avvägningen mellan hastighet och analysdjup utifrån tidspress.

3. Arbetsytan. Visa de tre panelerna i tur och ordning.

Centerpanel, transkript och protokoll. Visa transkriptet med talare och tidsstämplar, byt till protokollvyn, och visa delad vy med transkript och protokoll sida vid sida. "Det här är *protokolleditorn*, en texteditor likt Microsoft Word eller Google Docs där man kan redigera det första utkast som AI:n genererat." *Designmotivering:* delad vy gör att analytikern kan redigera protokollet med transkriptet synligt bredvid.

Vänsterpanel, *findings*. "Här visas alla fynd som AI:n har genererat, strukturerade efter flikar: *översikt* med sammanfattning och användarens noteringar; *frågor* där man ser alla PIR och senare fynd per PIR; och *tidslinje* med rekonstruerad tidslinje från förhöret." *Designmotivering:* fynden genereras progressivt, man ser dem dyka upp allteftersom varje analysperspektiv blir klart.

Högerpanel, *chatt*. "*Chattpanelen* låter analytikern ställa frågor mot transkriptet, fynden och protokollet. Man ser källhänvisningar i svaret som man kan

klicka på.” *Designmotivering:* chatten ger analytikern ett sätt att ställa riktade frågor utan att manuellt söka igenom hela transkriptet.

4. Analysperspektiv (kortfattat). “Systemet kan köra fyra analysperspektiv: *tidslinjerekonstruktion*, kronologisk ordning av händelser; *entitetsrelationer*, nätverk av personer, platser, objekt och deras kopplingar; *PIR-analys*, direkt koppling mellan fynd och informationsbehov; och *gapanalys*, vad som saknas, vad som är delvis besvarat, vad som inte berörs. Eftersom dessa analyser kräver mycket datorkraft varierar de i tidsåtgång, därför kan man köra enskilda perspektiv om man vill fördjupa sig i en åt gången eller är tidspressad.”

5. Protokollgenerering. Gå tillbaka och välj ett utkast av typen Full pipeline. Förklara att detta har kört alla analysperspektiv och med de resulterande fynden skapat ett mer gediget protokollutkast. “Allt som AI:n producerar är ett utkast. Analytikern redigerar, tar bort och lägger till innan dokumentet exporteras.”

Demoavslut. Innan intervjufrågorna: “Har ni några spontana frågor eller reaktioner innan vi går vidare till intervjufrågorna?”

Intervjufrågor (ca 20-25 min)

Semistrukturerade frågor: huvudfrågor följt av valfria fördjupande frågor beroende på svar.

Tema 1: Arbetsflödespassning och kognitiv belastning

1. Om ni tänker på en förhörsledares vanliga arbetsflöde, förberedelse, genomförande av förhör, omedelbar rapportering, dokumentation, inlämning: finns det analytiska moment i den kedjan där ett verktyg som FENRIR skulle kunna påverka er arbetsminnes- eller koncentrationsbelastning, antingen genom att avlasta eller tillföra belastning? *Fördjupning:* I vilka faser bedömer ni att AI-extraherad information skulle vara operativt användbar, respektive inte användbar? Finns det steg som måste förbli helt manuella, och varför?

2. Vad tänker ni om att skriva egna noteringar innan man ser AI-output (Steg 3 i wizarden)? *Fördjupning:* Hur upplevs det steget?

Tema 2: Mänsklig kontroll och tillit

1. Källhänvisningarna i fynden pekar tillbaka till specifika rader i transkriptet. Hur ser ni på spårbarhet till källor som en AI använder? *Fördjupning:* Skulle ni lita på ett fynd som saknar källhänvisning?

2. I protokollet, hur ser ni på möjligheten att skilja mellan vad som är AI-genererat respektive skrivet av användaren? *Fördjupning:* Förhörsledaren är ansvarig för det som skickas till den mottagande verksamheten, hur ser ni på

ansvarsfördelningen för det som skrivs när man nyttjar AI till att effektivisera sådana processer?

Tema 3: AI-kvalitet och informationstäckning

1. Hur ser ni på tillförlitligheten i AI:ns analyser, att den exempelvis kan presentera analyser som förhörsledaren inte håller med om?
2. Om ni upplever att det finns en sådan risk, vad skulle ni behöva se i systemet för att hantera den?
3. Hur ser ni på hur informationen är strukturerad, uppdelningen i *översikt*, *frågor* och *tidslinje*, samt kopplingen mellan fynd och PIR? Stödjer strukturen ert sätt att tänka, eller krockar den?

Tema 4: Funktionsrelevans

1. Vilken del av de olika funktionerna vi presenterat är viktigast och minst viktig för att stödja protokollskrivandet? Motivera. *Fördjupning:* Finns det någon funktion som kändes onödig eller distraherande? Något som ni inte förstod syftet med? Finns det något ni hade önskat ha istället?
2. *Chatten* låter er ställa frågor mot materialet. Hur ser ni på den funktionen jämfört med att själv söka i transkriptet? *Fördjupning:* I vilka situationer skulle ni använda den, och i vilka inte?

Tema 5: Kritisk feedback

1. Finns det något som skulle behöva ändras för att ett verktyg som det här skulle kunna användas operativt? Om ja, vad väger tyngst?
2. Finns det funktionalitet som ni förväntar er som inte fanns? *Fördjupning:* Om ni arbetar med flera förhör av samma eller olika källor, hur skulle ett verktyg som detta behöva hantera relationen mellan förhör (t.ex. ett projekt per källa med underförhör, eller tvärsökning mellan förhör)?

Avslutning

> "Tack så mycket: det här har varit väldigt värdefullt. Finns det något mer ni vill tillägga som vi inte har berört?"

Informera om att inspelningen kommer transkriberas och analyseras tematiskt, och att de är välkomna att kontakta oss om de har ytterligare tankar efteråt.

D Declaration of AI Use

We, Jonathan Ahlström and André Roxhage, hereby acknowledge the use of several generative AI tools throughout our project and thesis work. In accordance with the examination guidelines, the tools were used strictly as assistants and not as primary authors or problem solvers. Specifically, Claude, Claude Code, Cursor, and occasionally Google Gemini were used to assist with brainstorming, idea generation, and coding by providing structural suggestions and code snippets. During the initial phases, Notion AI was utilized to assist with project management and task structuring. For research and data handling, NotebookLM helped organize and synthesize our reference materials, while KlangAI was used specifically for transcribing audio data. For the evaluation phase, Claude Opus 4.6 was used to generate the fictional interrogation transcript and the accompanying Priority Intelligence Requirements (PIRs) and Specific Information Requirements (SIRs) that served as input material during the walkthrough sessions; these artefacts were manually reviewed and adjusted by us to fit the scenario and to avoid classification concerns. We confirm that all AI-provided code, transcriptions, generated evaluation material, and structural suggestions were thoroughly reviewed, adapted, and verified by us. Generative AI was not used to write the raw text of this report. Its role in the writing process was strictly limited to linguistic refinement and improving readability. All core ideas, analyses, results, and conclusions are entirely our own, and we have personally read and verified all referenced sources.

E Work Distribution

During the full project and writing of this thesis, the authors have participated in all phases of the project with an equal distribution of work.

E.1 Time Work Project management

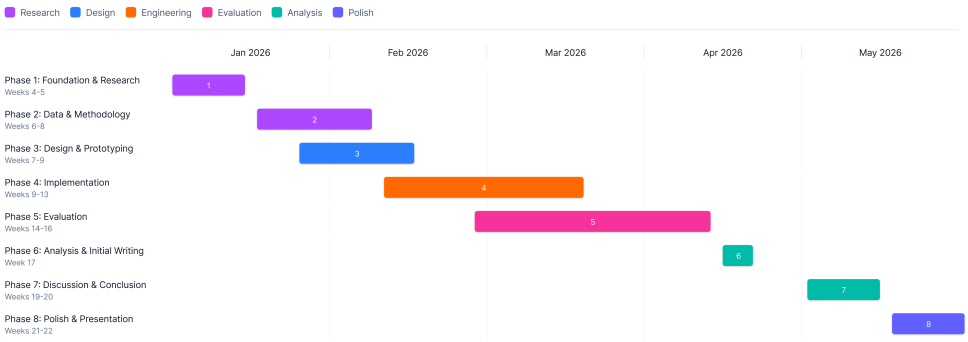

Figure E.1: Gannt-chart showing the different project phases.