Designing and Evaluating an Agentic RAG System for a Domain-Specific Research Archive: A Case Study on the KDC Aviation Knowledge Base

STEN DEN HARTOG, Hogeschool van Amsterdam, The Netherlands

The Knowledge Development Centre (KDC), a research collaboration between LVNL, Schiphol Group, and KLM, performs operational research to secure Schiphol's future position as a major hub in European airspace. The 89 technical reports this research has produced lack a semantic query interface, and their size and technical depth make it difficult to synthesise findings across documents, hindering research efficiency.

This thesis presents the KDC Research Assistant, a domain-specific RAG system developed across two design iterations. The final system is a locally-deployable reasoning agent with extended thinking that orchestrates four tools: internal corpus search using hybrid BM25 and dense retrieval with cross-encoder reranking, KDC research agenda access, aviation authority databases (Eurocontrol/SESAR via CORDIS and OpenAlex), and external academic literature, with source references maintained for human-in-the-loop verification. The agent evaluates retrieved context quality and issues multiple searches before committing to an answer, enabling synthesis and gap analysis that a single-step pipeline cannot perform.

Retrieval is evaluated using a 35-query BEIR-style benchmark with expert-annotated relevance judgements. A four-configuration ablation study shows the hybrid+rerank pipeline achieves nDCG@10 = 0.894 and Recall@20 = 1.000, a 30.7 percentage-point gain over the BM25 baseline. Generation quality is evaluated with RAGAS across five independent judge runs, for both the API development configuration and a fully local configuration: the local model matches the API configuration on faithfulness (0.655 vs 0.665) and scores higher on answer relevancy, at the cost of lower context recall, supporting deployment without external API calls.

Additional Key Words and Phrases: Retrieval-Augmented Generation, Information Retrieval, Aviation Knowledge Management, BEIR Evaluation, Hybrid Retrieval, Agentic RAG

Contents

Abstract 1
Contents 1
1 Introduction 2
1.1 Context 2
1.2 Problem Statement 2
1.3 Related Work and Knowledge Gap 3
1.4 Proposal 4
2 Background 4
2.1 Relevant Literature and State of the Art 4
2.2 Stakeholder Analysis 6
2.3 Current Situation 7
2.4 AI Considerations 7
3 Methodology 7
3.1 Design Science Research Approach 7
3.2 Requirements 8
3.3 System Design and Iterations 9

Author's Contact Information: Sten den Hartog, Hogeschool van Amsterdam, Master Applied Artificial Intelligence, Amsterdam, The Netherlands, sten.denhartog.sdh@gmail.com.

3.4 Corpus Data Analysis 13
3.5 Data 14
3.6 Model Architecture 15
4 Results 16
4.1 Retrieval Evaluation Results 16
4.2 Generation Quality: RAGAS Evaluation (Iteration 2) 17
4.3 Stakeholder Evaluation 18
5 Discussion 19
5.1 Evaluation of Results and Architecture Choices 19
5.2 External Validity: SciFact Benchmark 20
5.3 Limitations 21
5.4 Reliability, Validity, and Generalisability 22
5.5 Ethical and Societal Reflection 22
6 Conclusion 22
6.1 Conclusion and Requirements Evaluation 22
6.2 Future Work and Recommendations 24
6.3 Ethical Responsibility Statement 24
A System Diagrams 25
B Interface Screenshots 28
References 28

1 Introduction

1.1 Context

The Knowledge Development Centre (KDC) is a research collaboration between LVNL (Luchtverkeersleiding Nederland), Schiphol Group, and KLM. Its mission is to perform the operational research that keeps Schiphol relevant and competitive as a major hub in European airspace, covering air traffic management, airport operations, noise modelling, and related topics. Over time, this work has produced 89 research reports on a broad range of air traffic management (ATM) topics: runway occupancy, trajectory prediction, approach procedures based on the Global Navigation Satellite System (GNSS), air traffic controller workload, air traffic flow management (ATFM), and taxiing optimisation, among others.

These reports represent a significant institutional knowledge asset. Aviation operational decisions on separation standards, procedure design, capacity planning, and safety tool deployment benefit directly from synthesising insights across multiple studies. Researchers and operational specialists at LVNL and partner organisations regularly need to ask questions such as: *"What methods have been used to predict taxi times at Schiphol?"*, *"Which studies identify gaps in GNSS approach validation?"*, or *"How are noise models connected to trajectory optimisation research?"* These questions cross document boundaries and require semantic understanding, not keyword matching.

1.2 Problem Statement

The current state of the KDC knowledge base is a collection of PDF documents on SharePoint. This structure presents three interconnected problems. First, **discovery** requires knowing which documents exist; practitioners cannot retrieve

relevant studies without prior knowledge of the archive. Second, **synthesis** is manual: identifying connections and contradictions across studies requires reading multiple documents in full. Third, **access speed** is limited. Extracting insights across documents is challenging when those documents are in-depth research reports, ranging from 16 to 224 pages and written in dense technical language; finding one specific result takes substantial reading time even when the right document is known.

These problems are not unique to KDC: finding and reusing knowledge held in internal document repositories is a long-standing, well-documented challenge for organisations in general [3, 13], from research institutes to regulatory authorities and engineering firms. At KDC they are made worse by the specialised vocabulary of aviation and ATM: domain-specific acronyms (e.g. AMAN, Arrival Manager; TCAS, Traffic Collision Avoidance System) and regulatory terminology reduce the effectiveness of general-purpose search tools.

1.3 Related Work and Knowledge Gap

At its core, this is an information access problem: a bounded collection of unstructured technical documents must answer natural-language questions that range from specific facts to cross-document synthesis. Three solution directions exist for such problems. Classical keyword search is simple and avoids generation errors, but it cannot bridge the paraphrase gap that dominates natural-language research questions, and it cannot synthesise across documents. Fine-tuning a language model on the corpus internalises its content, but 89 documents is far too little training data, the model would need retraining as the archive grows, and a fine-tuned generative model offers no citation grounding. The third direction, Retrieval-Augmented Generation (RAG) [17], keeps the documents outside the model: it grounds large language model (LLM) outputs in retrieved document passages, enabling cited, verifiable answers at query time without retraining on domain data. A parallel AI-opportunity study at LVNL [30] independently recommended a retrieval-augmented knowledge system as the single most valuable application across departments, which corroborates this choice.

General-purpose RAG systems have been applied to open-domain question answering [16] and enterprise document collections, but applying them to a specialist domain introduces challenges the general literature does not fully address: (1) specialised vocabulary (aviation acronyms, regulatory terminology) causes problems for embedding models trained on general text, reducing retrieval quality; (2) researchers in specialised domains need more than factual lookup: they need synthesis, gap identification, and relationship discovery across studies; (3) evaluation of domain RAG is methodologically weak, with the most common approach (RAGAS [10]) using an LLM as judge, which produces non-reproducible scores.

BEIR [33] provides a deterministic alternative: a benchmark protocol using human-annotated relevance judgements and rank-based metrics (nDCG@10), validated across 18 heterogeneous retrieval datasets. Kamalloo et al. [15] extend BEIR with reproducibility infrastructure and statistical comparison methods for multi-dataset retrieval evaluation. Applying BEIR-style evaluation to a proprietary domain corpus fills both the reproducibility gap and provides a principled methodology for comparing retrieval architectures.

Two gaps motivate this work, framed as application contributions rather than claims of methodological novelty: (1) hybrid BM25+dense+reranker architectures are well studied on general benchmarks but rarely evaluated on the small, acronym-dense, grey-literature corpora typical of domain knowledge centres; and (2) the application of domain RAG to aviation ATM research archives, with their specialised vocabulary and regulatory context, has not been reported.

1.4 Proposal

This thesis presents the **KDC Research Assistant**: a hybrid RAG system designed for multi-type interaction with the KDC archive, developed across two design iterations. The system is evaluated using a BEIR-compatible benchmark of 35 queries spanning five query types, comparing four retrieval configurations in an ablation study. The central research question is:

> *How can a RAG system be designed and evaluated to enable reliable, grounded interaction with a domain-specific aviation research archive?*

where *reliable* means that retrieved passages accurately represent the content of source documents, measurable via nDCG@10 following the BEIR evaluation methodology [33], and *grounded* means that generated answers are attributable to specific retrieved passages rather than drawing on parametric LLM knowledge, following the Attributable to Identified Sources framing of Rashkin et al. [23].

This question is decomposed into four sub-questions: SQ1 (Literature): which retrieval architectures and chunking strategies suit domain-specific technical archives? SQ2 (Design): how should the pipeline be configured to serve KDC practitioners' diverse information needs? SQ3 (Evaluation): what methodology enables reproducible retrieval measurement on a proprietary corpus? SQ4 (Results): to what extent does each component (BM25, dense, RRF, reranker) contribute, and what are the failure modes?

The contributions of this work are: (1) a hybrid RAG architecture adapted for a specialised aviation corpus, with a five-mode query interface in Iteration 1 and an agentic reasoning orchestrator in Iteration 2; (2) a reproducible BEIR-style evaluation methodology with expert-validated relevance judgements for proprietary domain archives; (3) empirical evidence comparing four retrieval configurations, establishing the marginal value of each component over a BM25 baseline; and (4) RAGAS generation quality measured across five independent judge runs, for both the API and a fully local configuration.

2 Background

2.1 Relevant Literature and State of the Art

2.1.1 Retrieval-Augmented Generation. RAG [17] pairs a retriever with an LLM generator. The retriever selects relevant passages from a document store; those passages are passed as context to the LLM, which generates a grounded, citable answer. This keeps factual knowledge outside the model weights, so the knowledge base can be updated without retraining. Gao et al. [12] describe three components every RAG system shares: indexing (parsing, chunking, embedding), retrieval (finding the most relevant chunks), and generation (prompting an LLM with the retrieved context).

2.1.2 Retrieval Architectures. **Sparse retrieval (BM25).** BM25 [26] ranks documents by word frequency, adjusted for document length. It works well on exact-match queries, which matters here because aviation acronyms like "GNSS" or "AMAN" are rarely paraphrased in source documents. On BEIR [33], BM25 remains a strong zero-shot baseline: several fine-tuned neural retrievers underperform it on the majority of the 18 datasets (the late-interaction model ColBERT, for example, beats BM25 on only 9 of 18), making it a strong non-AI baseline.

Dense retrieval. Models like DPR [16] and SBERT [24] embed both the query and each passage into vectors, then find the closest passages by cosine similarity. This handles semantic paraphrase (a query about "controller cognitive load" can match passages about "workload"), which BM25 cannot. This system uses BAAI/bge-large-en-v1.5, the English large model of the BGE family [36], a bi-encoder trained specifically for retrieval; at the time of selection it ranked

among the top English retrieval embedders on the MTEB benchmark [19]. The choice follows the project constraints as much as the benchmark: at 335M parameters it runs locally on CPU or a modest GPU, leaving the workstation GPU free for the local reasoning model (REQ-ORG-1), and its permissive licence allows organisational use. Comparable open models (E5-large, GTE-large) offered no decisive advantage, larger top-ranked embedders would not fit the hardware budget, and proprietary embedding APIs (Cohere Embed, OpenAI text-embedding-3) were excluded by REQ-ORG-1, which forbids transmitting corpus content to external services. The embedding landscape evolves quickly: on the successor benchmark MMTEB [9], newer compact open models now outrank BGE-large in the same size class, and re-evaluating the embedder is identified as future work (Section 6.2).

Hybrid retrieval and RRF. Reciprocal Rank Fusion [4] merges ranked lists from multiple retrievers without needing to normalise their scores: $\mathrm{RRF}(d) = \sum_{r \in R} \frac{1}{k + r(d)}$, where R is the set of retrievers, $r(d)$ is the rank of document d in retriever r's list, and $k = 60$ is a smoothing constant that limits the influence of top-ranked items. Combining BM25 and dense retrieval with RRF gets the best of both: exact-match precision from BM25, semantic recall from the embeddings.

Cross-encoder reranking. A cross-encoder [20] reads the query and a candidate passage together in a single input and scores their relevance in one forward pass, so the model can attend to every query token in relation to every passage token. This is more precise than a bi-encoder but too slow to run over a full index, which motivates the two-stage design: a fast first stage (BM25 + dense retrieval) casts a wide net, and the cross-encoder rescores only the top candidates. On BEIR, BM25 + cross-encoder is the strongest zero-shot configuration across 18 datasets [33].

2.1.3 Document Chunking Strategies. Chunk size and boundaries substantially affect retrieval quality [12]. Fixed-size splitting at token limits is simple but cuts mid-sentence. Semantic chunking splits at embedding-distance thresholds, but boundary detection degrades when the embedding model is not calibrated to the target domain vocabulary. Sentence-window chunking groups a fixed number of consecutive sentences with a stride smaller than the window, producing overlapping chunks that always begin and end at sentence boundaries; Kamalloo et al. [15] find it competitive on BEIR long-document datasets. Contextual retrieval [1] prepends a document-level summary to each chunk before embedding, reducing retrieval failure by 35–49%. The chunking strategy selection and parameter justification for this system are presented in Section 3.4.

2.1.4 Language Model Selection for Domain RAG. The LLM in a RAG pipeline generates a grounded answer from retrieved passages. Model choice affects hallucination risk, answer quality, and compliance with infrastructure requirements: large proprietary models (GPT-4o, Claude) produce the highest quality but require external API calls that may transmit sensitive document content, while open-weight models such as Llama 3.1 [7] and DeepSeek-R1 [5] can run locally via Ollama, satisfying data residency requirements. Some of these are *reasoning models*: they produce an explicit step-by-step reasoning trace before the final answer, building on chain-of-thought prompting [35], which showed that intermediate reasoning steps improve performance on complex, multi-step problems. Iteration 2 uses this trace as a transparency mechanism (Section 3.3.3). Ji et al. [14] classify hallucination into intrinsic (contradicts retrieved source) and extrinsic (adds unsupported content); grounded RAG reduces but does not eliminate the extrinsic variant. The specific model selection requirements and deployment scenarios for this system are justified in Section 3.

2.1.5 Agentic RAG and Tool Use. The pipeline described so far is fixed: it retrieves once and then generates. A parallel line of work gives the language model control over the process. ReAct [37] interleaves reasoning steps with actions, so the model can plan, call a tool, observe the result, and re-plan before answering. Toolformer [29] shows that a model can learn to call external tools such as search engines and fold the results into its output. Self-RAG [2] lets the model

decide whether retrieval is needed at all and judge the retrieved passages before using them. Recent work groups these ideas under *agentic RAG* [32]: a reasoning model orchestrates retrieval tools over multiple steps and checks whether the evidence is sufficient before answering. Iteration 2 adopts this design (Section 3.3.4), with the fixed Iteration 1 pipeline as one of the tools.

2.1.6 RAG Evaluation. RAGAS [10] evaluates generation quality (faithfulness, answer relevance, context precision/recall) using an LLM as judge, enabling evaluation without human annotation. However, the stochastic nature of LLM sampling produces non-reproducible scores across runs, and judge model alignment with specialised domain concepts is uncertain. BEIR [33] evaluates retrieval quality independently from generation using nDCG@10 as the primary metric, a rank-discounted measure of relevance that penalises relevant documents appearing lower in the result list. BEIR's human-annotated qrels provide deterministic, reproducible scores, making it methodologically preferable for research-grade evaluation.

2.1.7 Knowledge Representation. Knowledge graphs store explicit entity–relationship triples that vector search cannot represent: that two studies address the same runway, or that one finding contradicts another, is absent from cosine similarity. Microsoft's GraphRAG [8] demonstrated that graph enrichment improves RAG on cross-document queries. In this project, a graph was built in Iteration 1 as a corpus exploration tool; its role and the decision to retire it as a retrieval mechanism in Iteration 2 are discussed in Section 3.3.4.

2.1.8 Retrieval Failure Modes. Retrieval fails in two directions. False positives inject misleading context: Shi et al. [31] term this *context-conflicting hallucination*, where the model produces a cited answer consistent with a wrongly retrieved passage. False negatives produce incomplete synthesis: for multi-document queries, a missing relevant passage may produce misleadingly confident answers that omit contradictory evidence. Domain vocabulary mismatch is a source of both: embedding models trained on general text may misrepresent aviation acronyms (e.g. grouping "AMAN" with "Amazon" by character overlap rather than domain semantics), motivating the BM25 component as an exact-match safety net [14].

2.1.9 Multimodal Document Retrieval. KDC reports contain runway diagrams, trajectory plots, noise contour maps, and measurement tables that a text-only pipeline silently discards. Two retrieval strategies have been published: caption-based retrieval, which passes extracted figures to a vision-language model and indexes the resulting descriptions alongside text chunks [12]; and page-level visual retrieval (ColPali [11]), which embeds each PDF page as an image without a text extraction step. The current prototype is text-only; a multimodal extension path using Microsoft MarkItDown [18] for document conversion is proposed in Section 6.2, motivated by a stakeholder request to include the Aeronautical Information Publication (AIP).

2.2 Stakeholder Analysis

Three stakeholder groups were identified through the project brief and a prototype demonstration in April 2026:

KDC researchers (primary users) need to efficiently retrieve methodology details, findings, and citation sources across the full corpus. Key requirements elicited: cited answers, synthesis across multiple papers, gap identification, and domain-appropriate terminology handling.

LVNL operational specialists need to identify relevant research for operational decision-making. Requirements include: policy implication framing, traceability of recommendations to source material, and local deployment (no external API for sensitive operational data).

KDC management / Schiphol Group need strategic research overviews. Requirements include: identification of KDC's knowledge development strategy, linking studies to operational outcomes, and iterative search refinement.

Feedback from the April 2026 prototype demonstration (Koos Noordeloos, KDC company supervisor)[1] identified four requirements: iterative search refinement, a "policy implications" query mode, linking studies to concrete operational outcomes, and contextualisation within KDC's broader research strategy. These informed the second design iteration (Section 3).

2.3 Current Situation

Prior to this project, the KDC archive was accessible only as a collection of PDF files on SharePoint. Practitioners relied on manual reading, filename-based search, and personal recall for knowledge retrieval. No semantic query interface, no cross-document synthesis capability, and no structured knowledge representation existed. The KDC archive had not previously been used as a corpus for any RAG or information retrieval system.

2.4 AI Considerations

Technical. LVNL operates safety-critical infrastructure and has strict requirements for software tools used operationally. The system is designed for fully local deployment using Ollama (open-source LLM serving), eliminating external API dependencies. The embedding model (BAAI/bge-large-en-v1.5) runs locally via sentence-transformers. This constraint directly influences architecture choices: ColBERT's large index and cloud-oriented infrastructure are less appropriate than a local BM25+CE stack.

Societal. Aviation ATM systems affect flight safety and passenger welfare. An AI system that provides incorrect or hallucinated information about separation standards or approach procedures could, if operationally misused, contribute to unsafe decisions. The system is explicitly scoped as a *research discovery tool* (finding relevant studies), not an operational safety tool. Every answer includes citations to source passages, enabling human verification.

Ethical. Hallucination is the primary ethical risk: the LLM generating claims unsupported by retrieved context. Mitigation measures include strict citation grounding (every answer must cite a retrieved passage), low LLM temperature (0.0 for factual modes), a system prompt that explicitly instructs the model not to draw on parametric knowledge outside the retrieved context, and an explicit in-interface disclaimer that the tool is an AI system whose answers can contain mistakes and must always be verified against the cited sources (Figure 7). Future work should add automated citation verification (checking that cited claims appear verbatim in the retrieved passages).

3 Methodology

3.1 Design Science Research Approach

The project follows the Design Science Research (DSR) methodology [22], which structures artefact-producing research as six activities: Problem Identification, Define Objectives, Design and Development, Demonstration, Evaluation, and Communication. DSR is appropriate here because the research question concerns *how* an artefact should be designed, not whether a phenomenon exists. Two design iterations structure the project:

- **Iteration 1**: Core hybrid RAG pipeline with a five-mode Streamlit interface and knowledge graph. Demonstrated to KDC supervisors in April 2026; evaluated with the BEIR-style benchmark (Section 3.4).

[1] Named individuals are identified with their documented consent.

- **Iteration 2**: Agentic RAG: a locally-deployable reasoning model with extended thinking, orchestrating four tools: internal corpus, KDC research agenda, aviation authorities (Eurocontrol/SESAR), and external academic literature. Motivated by evaluation evidence, stakeholder feedback, and academic supervision. Described in Section 3.3.3.

3.2 Requirements

3.2.1 Legal, Ethical, and Organisational Requirements.

- **REQ-ORG-1**: The system must support fully local deployment without external API calls, to comply with LVNL's data security requirements for operational environments.
- **REQ-ETH-1**: All generated answers must cite the source passages used, enabling human verification and preventing unattributed hallucination.
- **REQ-ETH-2**: The system must be scoped as a research discovery tool, not an operational decision support system, and must clearly communicate this scope to users.
- **REQ-ORG-2**: The system must integrate the KDC Research Agenda 2026–2030 as a queryable source, at the explicit request of the KDC company supervisor, so that findings can be situated against KDC's strategic priorities.
- **REQ-TRANS-1**: In the agentic configuration, the system must expose its full reasoning trace alongside the final answer (motivated in Section 3.3.3).

3.2.2 Functional and Technical Requirements.

- **REQ-FUNC-1**: The system must support five query modes: Factual Q&A, Synthesis, Gap Analysis, Idea Generation, and Connection queries (Iteration 1; superseded by agent-driven mode selection in Iteration 2).
- **REQ-FUNC-2**: Retrieval must support at least 89 documents without performance degradation.
- **REQ-FUNC-3**: The system must accept new documents via a web interface without requiring code changes.
- **REQ-FUNC-4**: Response time must be under 30 seconds for factual queries on standard hardware.
- **REQ-FUNC-5**: The system must support iterative query refinement within a session (KDC stakeholder request, April 2026).
- **REQ-FUNC-6**: The system must be able to contextualise individual findings against the KDC research agenda (KDC stakeholder request).
- **REQ-FUNC-7**: The system should frame answers in terms of operational or policy consequences, not only academic conclusions (LVNL operational specialists).
- **REQ-TECH-1**: The retrieval pipeline must include a non-AI (BM25) baseline configuration for comparative evaluation.
- **REQ-TECH-2**: Retrieval evaluation must be reproducible: rank-based metrics (nDCG@10) computed against human-annotated relevance judgements, with no stochastic LLM judge in the primary metric.
- **REQ-TECH-3**: The system must be deployable via Docker Compose without manual environment configuration, ensuring consistent deployment across LVNL workstations and demonstration environments.

The requirements listed above are the subset directly addressed in this design study. A full requirements specification, including traceability to stakeholder feedback sessions and supporting source citations for each requirement, is maintained in the project GitLab wiki [6].

3.3 System Design and Iterations

3.3.1 Iteration 1: Core Architecture. The first iteration built the three-layer system shown in Figure 3 (Appendix A). The architecture follows the BM25+CE pattern, which BEIR [33] identifies as the strongest zero-shot configuration across 18 diverse retrieval datasets.

Layer 1: RAG pipeline. PDFs are parsed with PyMuPDF (page numbers preserved for citations), split into 12-sentence overlapping chunks (stride=6), and embedded with BAAI/bge-large-en-v1.5 into ChromaDB. A BM25Okapi index is built over the same chunks. At query time, both retrievers return top-20 candidates; RRF merges the lists; the cross-encoder (ms-marco-MiniLM-L-6-v2) reranks the merged top-20 and passes the top-5 to the LLM.

Layer 2: Knowledge graph. An LLM extracts entity triples at ingestion time and persists them as a networkx graph. Connection-mode queries traverse the graph; a visualisation provides a corpus-level topic overview. Extraction quality is a bottleneck: the LLM produced overlapping nodes ("air traffic management" and "air traffic control" as distinct entities), making traversal less reliable than retrieval-based methods. The graph is treated as a prototype exploration feature (Figure 6); its retrieval contribution is not measured in the evaluation. It is retired as a retrieval mechanism in Iteration 2 (Section 3.3.4).

Layer 3: Application. A four-page Streamlit app provides: a chat interface (Page 1), a knowledge graph visualiser (Page 2), a document browser (Page 3), and a bulk upload interface (Page 4). A query router assigns each query to one of five modes, which controls the retrieval strategy and LLM prompt template:

- **Factual Q&A**: a direct question with a specific answer (e.g., *"What measurement method was used in study X?"*).
- **Synthesis**: a question that requires combining information from multiple documents (e.g., *"What approaches have KDC researchers used to predict taxi times?"*).
- **Gap Analysis**: a question about what has not yet been studied (e.g., *"What aspects of GNSS approach procedures are missing from KDC research?"*).
- **Idea Generation**: a question asking for research directions based on existing work (e.g., *"What follow-up research looks promising for ATFM?"*).
- **Connection**: a question about relationships between topics or studies (e.g., *"How are noise modelling studies connected to trajectory optimisation research?"*).

Interface design decisions. The five-mode picker (Figure 5, Appendix B) makes the query type an explicit user parameter rather than inferring it automatically. An auto-classification approach was prototyped and rejected: misroutes are silent (a Gap Analysis question routed to Factual Q&A returns a plausible but wrongly-framed answer), and experienced KDC researchers already know what kind of answer they need before typing. In Iteration 2, the agent replaces the static mode picker: it determines strategy per query based on reasoning, which is more flexible but less transparent to the user about what retrieval approach is being used.

Error handling at the interface boundary covers three failure modes. A retrieval timeout or ChromaDB connection failure is caught at the service layer and surfaced as a descriptive error message in the chat window rather than a stack trace, so the user knows what happened and can retry. A document with a parsing failure during ingestion (corrupted PDF, DRM-protected file) is skipped and logged; the upload interface reports which files succeeded and which were skipped. A query that returns zero passages above the relevance threshold is handled by returning an explicit "no relevant passages found" response, preventing the LLM from hallucinating an answer from parametric knowledge.

User control mechanisms beyond the mode picker include: a conversation reset button; a source panel in the sidebar showing the PDFs currently in the corpus; and a document browser page where users can inspect which documents

are indexed, providing transparency over what the system actually knows. In agent mode (Iteration 2), the reasoning trace and tool call log are displayed inline as collapsible status blocks (Figure 7), making the agent's decision process inspectable rather than treating it as a black box (REQ-TRANS-1).

Iterative improvement from user feedback. The interface evolved directly in response to the April 2026 stakeholder session, through three changes. First, Koos Noordeloos's request to incorporate the KDC Research Agenda 2026–2030 as a searchable source led to the *search_kdc_agenda* tool, so internal findings can be positioned against KDC's research priorities. Second, stakeholder comments about single-step retrieval not capturing the full context of multi-document questions motivated the shift to an agentic loop in which the model can issue multiple sequential searches and synthesise across them. Third, the knowledge graph, while visually engaging, received feedback that its value was unclear relative to the chat interface, which motivated its retirement as a retrieval mechanism.

3.3.2 Iteration 1: Evaluation Framework. After the April 2026 demonstration, two gaps were identified: (1) the planned RAGAS evaluation was rejected as non-reproducible, and (2) the chunking parameters were not grounded in corpus data. The evaluation framework was added to Iteration 1: a BEIR-style benchmark, a corpus EDA pipeline, and bulk ingestion support. The evaluation pipeline is three scripts: `eda_corpus.py` (corpus statistics and plots), `qrels_builder.py` (retrieve top-20 per query for annotation), and `beir_runner.py` (compute nDCG@10, MRR, Recall@k across all four retrieval modes). Results are reported in Section 4.

3.3.3 Iteration 2: Motivation for Agentic Architecture. Three independent sources of evidence point to the same limitation of the Iteration 1 design: the static, single-shot retrieval pipeline does not fit the complex, multi-document reasoning tasks that KDC practitioners most frequently need.

Evaluation evidence. The BEIR ablation study shows that Synthesis queries (nDCG@10 = 0.859) and Idea Generation queries (0.821) are the lowest-performing modes in the hybrid+rerank configuration. These are precisely the modes that require multi-document reasoning: a Synthesis answer cannot be grounded in a single passage, and Idea Generation must span the corpus for patterns and unexplored directions. The performance gap relative to Gap Analysis (0.950) and Connection queries (0.938) reflects a structural limitation of single-step retrieval. Recall@20 = 1.000 across all query types confirms that all relevant documents are reachable; the pipeline cannot reason across them once retrieved.

Stakeholder feedback. The April 2026 prototype demonstration with Koos Noordeloos and Evert Westerveld (KDC/LVNL) identified three improvement requests that a deterministic pipeline cannot meet: (1) iterative query refinement, where users should be able to narrow or expand a query without restarting the session (REQ-FUNC-5, high priority); (2) contextualisation of individual study findings against KDC's documented research agenda (REQ-FUNC-6); and (3) framing of answers in terms of operational consequences rather than academic conclusions (REQ-FUNC-7). These requirements describe a system that must maintain context across steps, access structured external knowledge, and reason about research strategy.

Academic feedback. The thesis supervisor identified that BEIR metrics alone do not capture generation faithfulness; the lab supervisor recommended engagement with the state of the art in agentic reasoning. Both concerns are addressed by an agentic architecture with RAGAS generation evaluation [10] as a complement to BEIR, run five times to control for score variance.

Design decision. These three sources of evidence motivate a shift from a deterministic pipeline to an **agentic architecture**: a reasoning-capable LLM that orchestrates tools, evaluates retrieved context quality, and iterates before committing to an answer. The Iteration 1 retrieval pipeline becomes the primary tool available to the agent; BEIR results remain valid as the baseline for internal search.

Table 1 compares the two architectures across the criteria most relevant to KDC.

Table 1. Deterministic (Iteration 1) versus agentic (Iteration 2) architecture

Criterion	Deterministic (It. 1)	Agentic (It. 2)
Query handling	Single retrieval step; fixed prompt per mode	Multi-step planning; agent decides strategy per query
Iterative refinement	Not supported	Native via agentic loop
External knowledge	Internal corpus only	Internal + KDC agenda + CORDIS / Eurocontrol + Semantic Scholar / OpenAlex
Transparency	Document citations (title + page)	Citations + full reasoning trace
Failure handling	Generates regardless of context quality	Re-queries if retrieved context is insufficient
Evaluation	BEIR (deterministic)	BEIR (retrieval) + RAGAS ≥5 runs (generation)
Latency	2–4 seconds	15–30 seconds (acceptable for research use [21])
Local deployment	Fully local via Ollama	Local reasoning models (DeepSeek-R1, Qwen3) viable via Ollama

Transparency as a first-class requirement. An agentic system brings a transparency obligation that the deterministic design does not have. In Iteration 1, document citations provide traceability. In a multi-step agentic system, citations remain necessary but are not enough: researchers must also be able to inspect the agent's reasoning (which tools it called, why it searched again, and how it weighed evidence across sources). In an organisation with a safety-adjacent research mission, opaque automated reasoning is an institutional risk. This motivates **REQ-TRANS-1** (Section 3): the system must expose the full reasoning trace alongside the final answer.

REQ-TRANS-1 directly constrains model selection. OpenAI o-series models (o3, o4-mini) do not expose their chain of thought via the API. Two alternatives satisfy the requirement: (1) Claude with extended thinking (Anthropic), which streams thinking tokens separately from the response; and (2) open-source reasoning models such as DeepSeek-R1 [5] and Qwen3, which emit reasoning in explicit `<think>` blocks and run locally via Ollama, simultaneously satisfying REQ-ORG-1 (local deployment). Model selection for Iteration 2 is developed in Section 3.3.4.

3.3.4 Iteration 2: Agentic Architecture Design. The Iteration 2 system is organised around a reasoning-capable LLM acting as an orchestrator. Instead of executing a fixed retrieve-then-generate sequence, the orchestrator interprets each query, decides which tools to invoke and in what order, judges whether the retrieved evidence is sufficient, and iterates before committing to an answer. The Iteration 1 pipeline is not discarded; it becomes one of the tools the orchestrator can call, which preserves the BEIR-validated retrieval capability as the baseline for the internal search function.

Tool selection. Four tools are made available to the orchestrator, and the selection is grounded in stakeholder evidence and the KDC research context rather than technical convenience.

The first, *internal retrieval*, wraps the Iteration 1 hybrid pipeline and searches the KDC thesis corpus. It is the primary tool, invoked first unless the question is explicitly strategic or regulatory. A parallel LVNL study [30] found that a retrieval-augmented knowledge system was the single application valued across all five interviewed departments, explicitly recommending its development as a student pilot; the present system is that pilot.

The second, *research agenda access*, reads the KDC Research Agenda 2026–2030, a structured strategy document that defines five research clusters and records both active and proposed studies. Access to it allows the system to situate a query against KDC's stated priorities and to support gap-oriented questions. This tool was included at the explicit request of the KDC company supervisor (Koos Noordeloos, April 2026): "Verwerken van research Agenda in het prototype" (REQ-ORG-2).

The third, *aviation authorities search*, queries the CORDIS EU R&D database (which indexes all projects funded by the Single European Sky ATM Research Joint Undertaking (SESAR JU) and Clean Aviation) and OpenAlex filtered by Eurocontrol's Research Organization Registry identifier [25]; a broader keyword search also covers EASA and ICAO publications in OpenAlex. Results are tagged `source=authority` and are never persisted. This tool fills a gap the academic search tool cannot: Eurocontrol operational concepts, SESAR JU deliverables, and EASA rulemaking opinions are grey literature that does not appear in academic citation databases, yet gap analysis questions such as "what does the international ATM community consider a research priority for trajectory-based operations?" need exactly this regulatory and programme-level context.

The fourth, *external literature search*, queries Semantic Scholar and OpenAlex for peer-reviewed abstracts, allowing internal findings to be compared against the broader academic field. The system prompt encodes a three-level hierarchy: internal corpus first, authorities for regulatory context, external literature for academic evidence.

All external sources follow a fetch-cite-discard pattern: results are retrieved at query time and discarded after the query, avoiding copyright concerns from persistently embedding third-party publications. Every result carries a provenance tag (`internal`, `authority`, or `external`), and authority and external claims include source URLs for verification.

Retirement of the knowledge graph. The knowledge graph built in Iteration 1 is retired as a retrieval mechanism. It was intended to serve Connection and Gap Analysis queries through entity traversal, but the reasoning orchestrator takes over that role by reasoning over retrieved passages directly. This is more flexible than a fixed graph, because the notion of a research gap is query-dependent and cannot be fully captured by a static set of extracted entities and relations, and it avoids the entity-extraction quality limitations of the Iteration 1 graph. The graph visualisation is retained only as an optional corpus-overview feature.

Model selection. The embedding model and cross-encoder reranker run locally, as in Iteration 1. The orchestrating reasoning model is constrained by REQ-TRANS-1: it must expose its reasoning trace. For development and evaluation, this study uses Claude with extended thinking (Anthropic API), which streams thinking tokens separately from the response. For production deployment at LVNL, REQ-ORG-1 requires fully local operation; the recommended path is DeepSeek-R1 [5] or Qwen3 via Ollama, since both emit reasoning in explicit `<think>` blocks. Both configurations are evaluated with the same RAGAS protocol in Section 4.2.

Containerised deployment. The application is packaged with Docker Compose (REQ-TECH-3): a demonstration configuration uses the API reasoning model; a local production configuration additionally runs Ollama. Document data is mounted as a volume, not built into the image, since the corpus contains embargoed material.

Figure 4 (Appendix A) shows the nominal execution sequence and four identified failure modes in the Iteration 2 system.

The internal-retrieval tool in Iteration 2 is the Iteration 1 hybrid pipeline unchanged: the agentic layer adds planning, multi-step retrieval, and cross-source synthesis on top of it rather than modifying the retriever itself. Three retrieval-level enhancements were considered but not implemented in this study (structure-aware chunking, contextual embeddings, and multimodal indexing); each is evaluated and deferred in Sections 4 and 6.2.

3.4 Corpus Data Analysis

Before finalising the chunking strategy, an exploratory data analysis was conducted on the 89 KDC PDFs. The EDA measured five properties: (1) token and word length distributions, to determine whether fixed-size chunking is viable; (2) distribution shape and outliers, to detect bias from unusually long documents; (3) information density (Type-Token Ratio, numeric token ratio), to identify table-heavy passages that embed poorly as prose; (4) near-duplicate detection, to prevent retrieval bias from repeated boilerplate text; and (5) natural breakpoints (section header detection), to evaluate whether structure-aware chunking would improve chunk coherence.

Table 2 presents selected corpus statistics obtained after ingestion.

Table 2. KDC Corpus Statistics (89 documents)

Metric	Mean	Median	P95
Pages per document	81.0	79.0	139
Words per document	27,404	25,174	49,745
Chunks per document (raw)	362	316	688
Words per chunk	158	159	189
Type-Token Ratio	0.10	–	–
Numeric token ratio	0.10	–	–
Total chunks in index (post filter): 23,039			

Beyond the summary statistics in Table 2, the *shape* of the length distribution matters for retrieval. Figure 1 shows that most reports cluster between 15k and 35k words, with a right tail of three outliers more than two standard deviations above the mean (*Assessing AI Opportunities for LVNL*, *Effects of Flexible Use of Airspace*, *Effects of wind and trajectory uncertainty in a 4D trajectory management interface*). These outliers produce proportionally more chunks and are over-represented in retrieval results; future work should investigate length-normalised scoring.

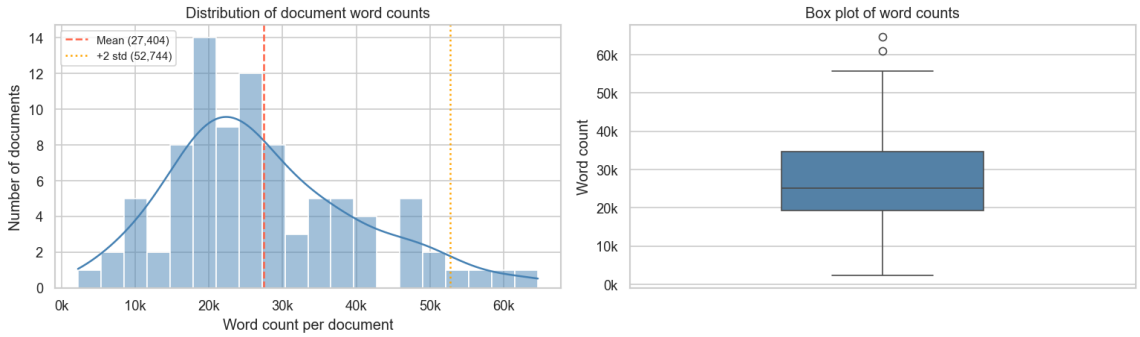

Fig. 1. Word count distribution across 89 KDC documents. Left: histogram with KDE curve; dashed lines mark the mean (27,404 words) and the +2 standard deviation threshold (52,744 words). Right: box plot showing the same distribution. Three outliers above 52k words are visible.

Figure 2 shows the chunk size and count distributions. Average chunk length clusters tightly around 158 words (120–200 range), confirming that the 12-sentence window produces consistently sized passages regardless of document

length. Chunk count per document follows the same right-skewed shape as word count, peaking around 200–300 chunks.

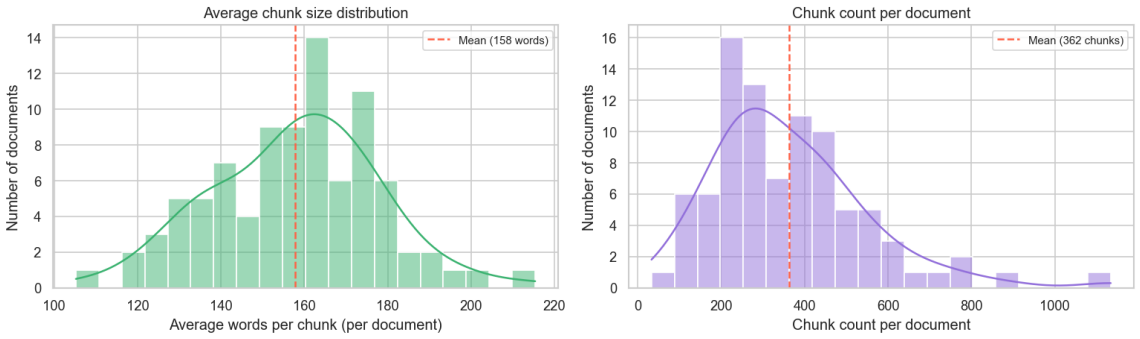

Fig. 2. Chunk statistics across 89 documents. Left: average words per chunk, mean 158 words. Right: total chunks per document, mean 362. The tight word-per-chunk distribution confirms consistent chunking across documents of varying length.

Section header detection was the decision criterion for structure-aware chunking: if more than 70% of documents contain detectable headers, section-boundary chunking would be considered. Detection succeeded in 98.9% of documents, yet structure-aware chunking was not adopted. Detected headers define sections of highly variable length (a one-paragraph conclusion and a twenty-page results section would both become single-section units), and inconsistent chunk lengths degrade embedding quality because a 30-word chunk and a 2,000-word chunk are not directly comparable in the same similarity space [12]. Sentence-window chunking with fixed parameters produces the consistent chunk sizes confirmed by the EDA (Figure 2); structure-aware chunking remains future work if paired with length normalisation or hierarchical indexing such as RAPTOR [28]. Based on these findings, the chunking strategy was set to 12-sentence windows with a 6-sentence stride (50% overlap), yielding 23,039 chunks after filtering passages below 50 tokens (from 32,249 raw chunks).

The mean Type-Token Ratio of 0.10 indicates moderate lexical diversity, consistent with technical reports that repeat domain terminology (AMAN, GNSS, ATFM). The numeric token ratio of 0.10 confirms a non-trivial density of numeric content (tables, measurements), which embeds less reliably and is a minor retrieval limitation for highly quantitative passages.

3.5 Data

The corpus comprises 89 PDF research reports produced by the KDC between approximately 2015 and 2024, covering the following topic areas: air traffic control workload, runway occupancy and separation, GNSS-based approach procedures, taxi time prediction, trajectory prediction, aircraft noise modelling, air traffic flow management (ATFM), weather integration in ATM, and multi-airport coordination.

All documents are in English. Documents range from short technical notes to full research reports. The corpus was ingested using PyMuPDF for text extraction, NLTK for sentence tokenisation, and BAAI/bge-large-en-v1.5 for embedding. The BM25 index was built over the same chunks using the rank-bm25 library. The knowledge graph was populated with entities extracted by GPT-4o during ingestion.

3.6 Model Architecture

3.6.1 Architecture Overview. The retrieval pipeline follows the BM25+CE pattern identified as the BEIR top performer [33]. The full pipeline is:

(1) **Query embedding**: The query is prefixed with the BGE instruction prefix and embedded with BAAI/bge-large-en-v1.5.
(2) **Dense retrieval**: Top-20 chunks by cosine similarity from ChromaDB.
(3) **Sparse retrieval**: Top-20 chunks by BM25Okapi score.
(4) **RRF fusion**: RRF($k = 60$) merges and deduplicates the two ranked lists.
(5) **Cross-encoder reranking**: cross-encoder/ms-marco-MiniLM-L-6-v2 rescores the top-20 fused results; top-5 are passed to the LLM.
(6) **Generation**: The LLM generates an answer with inline citations to source passages. For production this is a local reasoning model served via Ollama (REQ-ORG-1); an API model is used for development and evaluation.

The retrieval depth parameters follow standard two-stage retrieval practice rather than per-query tuning. The first-stage cutoff of 20 candidates per retriever is the conventional reranking pool size in the BEIR cross-encoder protocol [33]; the evaluation confirms it is sufficient here (Recall@20 = 1.000, so no relevant document is lost before reranking). The top-5 passages passed to the LLM are bounded by context budget: five passages of ≈158 words (Table 2) fit comfortably within the generation prompt while keeping latency low. These values were fixed a priori from these constraints, not optimised against the benchmark, which avoids tuning to the small evaluation set.

3.6.2 Configuration Comparison. Table 3 summarises the four configurations evaluated in the ablation study.

Table 3. Retrieval configurations in the ablation study

Configuration	First stage	Fusion	Reranker
BM25-only	BM25	–	–
Dense-only	BGE dense	–	–
Hybrid (no rerank)	BM25 + BGE	RRF	–
Hybrid + rerank	BM25 + BGE	RRF	Cross-encoder

3.6.3 Evaluation Metrics. Following BEIR methodology [33], three rank-based metrics are computed against manually annotated qrels:

nDCG@10 (primary): Normalised Discounted Cumulative Gain at rank 10. Rewards systems that place highly relevant documents (grade 2) higher than partially relevant ones (grade 1), with logarithmic rank discounting. nDCG@k = $\frac{\text{DCG@k}}{\text{IDCG@k}}$, where DCG@k = $\sum_{i=1}^{k} \frac{\text{grade}(i)}{\log_2(i+1)}$. Here grade($i$) is the relevance grade (0–2) of the document at rank i, and IDCG@k is the DCG of the ideal ranking (all relevant documents ordered by descending grade), so the score is normalised to $[0, 1]$.

MRR (Mean Reciprocal Rank): The reciprocal rank of the first relevant (grade ≥ 1) document, averaged across queries. Captures how quickly the system surfaces at least one useful result.

Recall@k ($k \in \{5, 20\}$): The fraction of relevant documents that appear within the top-k results. Particularly important for Synthesis and Gap queries that require multiple source documents.

Qrels were annotated by KDC domain experts grading the top-20 retrieved documents per query on a three-point scale (0 = not relevant, 1 = partially relevant, 2 = highly relevant), using document title, page number, and chunk preview as context.

4 Results

4.1 Retrieval Evaluation Results

Table 4 presents the BEIR-style evaluation results for the four retrieval configurations on the 35-query benchmark.

Table 4. BEIR evaluation results across four retrieval configurations (35 queries)

Configuration	nDCG@10	MRR	R@5	R@20
BM25-only (baseline)	0.587	0.786	0.418	0.649
Dense-only	0.760	0.957	0.546	0.765
Hybrid (no rerank)	0.765	0.981	0.571	0.787
Hybrid + rerank	**0.894**	**0.924**	**0.683**	**1.000**

The full hybrid+rerank configuration achieved nDCG@10 of 0.894, representing a 30.7 percentage-point absolute improvement over the BM25-only baseline (0.587). The ablation reveals three distinct gains. First, moving from BM25-only to dense-only retrieval yields the largest single step: +17.3 pp in nDCG@10. Semantically rephrased queries (e.g., "cognitive load" for passages discussing "workload") cannot be matched by BM25's term overlap; dense embeddings capture these synonyms effectively. Second, adding BM25 to the dense pipeline via RRF hybrid fusion adds only +0.5 pp on aggregate; this small overall figure hides a targeted gain on acronym queries, analysed in Table 8. Third, the cross-encoder reranking step adds +12.9 pp, the second-largest single gain. The cross-encoder attends jointly to the full (query, passage) pair, resolving ambiguities that the first-stage bi-encoder cannot.

Recall@20 reaches **1.000** for the full system: every document annotated as relevant in the benchmark appears within the top 20 results. This indicates that the retrieval bottleneck is ranking precision, not recall coverage: the system finds all relevant material but must order it correctly. The MRR of 0.924 confirms that a highly relevant result appears within the first two positions on average.

Table 5. Results by query type: hybrid+rerank configuration (35 queries total)

Query type	n	nDCG@10	MRR	R@20
Factual	10	0.898	0.950	1.000
Synthesis	8	0.859	0.854	1.000
Gap Analysis	5	**0.950**	**1.000**	1.000
Idea Generation	5	0.821	0.900	1.000
Connection	7	0.938	0.929	1.000

4.1.1 Results by Query Type. Recall@20 is 1.000 across all query types (Table 5), confirming that the retrieval ceiling is not the bottleneck for any mode; performance differences across types are attributable to ranking precision rather than coverage gaps.

Gap Analysis achieves the highest nDCG@10 (0.950) and perfect MRR (1.000). Research gaps are expressed in crisp, consistent domain language in KDC reports (e.g., "no study has addressed…", "future research should…"), making

them well-matched to the dense embedding space. Connection queries also score highly (0.938), as cross-document entity relationships are anchored by shared terminology. Synthesis queries (0.859) show slightly lower precision: multi-document synthesis requires retrieving passages from several different reports, and partial credit (grade 1) annotations are more common in this category. Idea Generation scores lowest (0.821): relevance for speculative future-work content is inherently subjective, and one idea query can have several reasonable answers, making strict rank ordering difficult.

4.2 Generation Quality: RAGAS Evaluation (Iteration 2)

BEIR metrics measure whether the right documents are retrieved, not whether the agent's final answer is accurate or relevant. RAGAS [10] fills this gap by using an LLM as an automated judge to score four aspects of answer quality. The protocol was applied to two configurations: the API development configuration (Claude with extended thinking) and a fully local configuration (qwen3:8b served via Ollama on consumer hardware). To control for the stochasticity of the judge model, each configuration's 35 agent outputs were scored by five independent RAGAS runs; Table 6 reports mean and standard deviation per configuration.

Table 6. RAGAS generation quality of KDCAgent (Iteration 2), 35 queries, 5 judge runs, mean ± std per configuration

Metric	API (Claude)	Local (qwen3:8b)
Faithfulness	0.665 ± 0.081	0.655 ± 0.027
Answer relevancy	0.724 ± 0.071	0.905 ± 0.013
Context precision	0.736 ± 0.019	0.790 ± 0.011
Context recall	0.628 ± 0.061	0.431 ± 0.033

Each metric captures a distinct aspect of generation quality; the API configuration is used as the running example.

Faithfulness (0.665 ± 0.081) measures whether every claim in the agent's answer is supported by the retrieved passages; claims that introduce information not present in any retrieved passage are penalised. A score of 0.665 means that roughly two thirds of the agent's statements are directly grounded in retrieved evidence. The rest reflect parametric knowledge or inferences beyond what the retrieved text explicitly states, a known characteristic of reasoning models, which synthesise and extrapolate rather than quote verbatim.

Answer relevancy (0.724 ± 0.071) measures whether the answer actually addresses the question that was asked. It is computed by generating hypothetical questions from the answer and measuring how well they match the original query. A score of 0.724 indicates that the answers are substantially on-topic, with room for improvement on broad synthesis and idea-generation queries, where the agent tends to provide wider context than strictly necessary.

Context precision (0.736 ± 0.019) measures the fraction of the retrieved passages that are actually relevant to the query; a low score means the agent feeds irrelevant context to the generation step. A score of 0.736 indicates that approximately three quarters of the passages the agent retrieves and uses are genuinely relevant; the low variance across judge runs shows the metric is stable, matching the BEIR finding that ranking precision is the remaining bottleneck.

Context recall (0.628 ± 0.061) measures whether the retrieved passages contain all the information needed to produce the reference answer; here the agent's retrieval captures roughly 63% of that evidence. The gap is partly structural: for synthesis and gap analysis queries the reference answer draws on many documents, and even with Recall@20 = 1.000 at the document level, not every relevant passage fits in the context passed to the model. Retrieving more passages per tool call would raise this score at the cost of longer prompts and higher latency.

Local configuration. The fully local model matches the API configuration on faithfulness (0.655 vs 0.665) and scores higher on answer relevancy and context precision, but lower on context recall (0.431 vs 0.628). The pattern is consistent with its behaviour: the local agent typically issued a single retrieval round (1.6 tool calls per query on average), so it grounds its answers in less evidence, while its more direct answers are rewarded by the relevancy metric. For factual queries the local configuration is competitive; for broad synthesis queries the API model's wider evidence gathering remains an advantage. The production-recommended larger model of the same family (qwen3:32b) was not separately measured.

Taken together, both configurations produce answers that are substantially grounded, on-topic, and supported by relevant context. The faithfulness gap is the most actionable finding: tightening the system prompt to require explicit citation for every factual claim would improve trustworthiness without architectural changes.

4.2.1 Comparison to Published Domain RAG Results. Interpreting RAGAS scores in isolation is difficult: no standardised benchmark exists and published scores depend strongly on task design. The closest published reference point for a specialised technical domain is the telecom QA study of Roychowdhury et al. [27]. Table 7 places the KDCAgent results next to theirs.

Table 7. RAGAS faithfulness in context: this work versus the telecom-domain QA study of Roychowdhury et al. [27]

System / condition	Task	Faithfulness
KDCAgent (this work)	Multi-document synthesis, aviation ATM	0.665
Telecom QA, correct retrieval [27]	Single-passage factual QA	0.88–0.94
Telecom QA, wrong retrieval [27]	Single-passage factual QA	0.55–0.78

Two observations follow. First, published faithfulness depends heavily on retrieval correctness: Roychowdhury et al. report 0.88–0.94 when the retrieved context is correct and 0.55–0.78 when it is wrong, where a low score signals that the generator answered from outside the retrieved context. The same study found faithfulness generally agrees with manual expert evaluation, which supports its use as a quality indicator. Second, the tasks differ: their setup scores short factual answers against a single retrieved passage, while the KDCAgent produces multi-document syntheses in which the reasoning model deliberately combines and extrapolates across passages, so per-claim grounding is lower.

4.3 Stakeholder Evaluation

Iteration 1 (April 2026). The Iteration 1 prototype was demonstrated to the KDC company supervisor (Koos Noordeloos) in April 2026. Qualitative feedback was positive: the multi-mode query interface was considered intuitive for research use; cited answers with page numbers were valued for traceability; the knowledge graph visualisation provided a novel overview of the corpus. Improvement requests included: support for iterative query refinement, a "policy implications" mode that frames answers in terms of operational consequences, and contextualisation of individual studies within KDC's broader research strategy. These requirements drove the Iteration 2 design (Section 3.3.4).

Iteration 2 (May 2026). A second stakeholder validation session was conducted with KDC/LVNL representatives to validate the Iteration 2 prototype. The session was deliberately comparative: Iteration 1 was shown first as a baseline, then the Iteration 2 agent interface, with a walkthrough of what changed and why in response to the April 2026 feedback; stakeholders then posed their own research questions.

Stakeholder responses were positive overall. The integration of internal and external knowledge sources was specifically highlighted: an answer that situates KDC internal findings against Eurocontrol or SESAR programme-level context within a single query was seen as a meaningful improvement. The reasoning trace (extended thinking visible as a collapsible status block) was found to increase confidence in the system's answers: seeing which tools the agent called and in what order made the provenance of the answer transparent. Cited answers with explicit source URLs for external and authority results were valued for verification.

Stakeholders also probed the system's out-of-scope handling with a deliberately irrelevant query (a request for a cake recipe). The system declined, explained its purpose, and named the knowledge domains it can address. This follows from the system prompt and the agent's grounding in tool-returned evidence: with no relevant passages to cite, the reasoning model declines rather than inventing an answer from parametric knowledge. Stakeholders found this appropriate and noted that reliable refusal on out-of-scope queries is as important as answer quality in a professional research context.

One concrete feature request emerged from the session: adding the **Aeronautical Information Publication (AIP)** as a knowledge source. The AIP is the official document series (published by LVNL under International Civil Aviation Organization (ICAO) Annex 15) describing procedures, airspace structure, navigation aids, and approach charts for Dutch airspace and Schiphol, and is directly relevant to gap analysis and operational context queries. However, AIP documents are heavily procedural and carry much of their information in visual form (aerodrome charts, approach diagrams, sector maps). The current text-only pipeline would discard that content entirely, producing an index that is systematically incomplete for procedural queries. This finding directly motivates the multimodal recommendation in Section 6.2.

5 Discussion

5.1 Evaluation of Results and Architecture Choices

Each component of the pipeline adds measurable value. The full system (nDCG@10 = 0.894) is 30.7 percentage points above the BM25 baseline (0.587), and that gap breaks down cleanly across three steps.

BM25 to dense retrieval: +17.3 pp. This is the biggest jump. Users query this system with natural-language questions, not keyword strings. BM25 misses paraphrase (a query about "controller cognitive load" does not match passages that say "workload"); dense embeddings do. This confirms the core design premise: specialised vocabulary in the corpus creates a mismatch that keyword search alone cannot fix.

Dense to hybrid (RRF): +0.5 pp overall, but the aggregate hides a clean per-subset signal. The Background hypothesised that BM25 would help on acronym queries, where exact-token matching matters. To test this, the 35 queries were partitioned by whether they contain a domain-specific acronym (ATC, GNSS, ATFM; "KDC" excluded as it appears in nearly every query). Table 8 reports the decisive dense → hybrid comparison.

Table 8. Effect of adding BM25 (dense → hybrid, no rerank), disaggregated by acronym presence

Subset	Dense	Hybrid	Δ nDCG@10
With acronym ($n = 10$)	0.656	0.719	**+0.063**
Without acronym ($n = 25$)	0.801	0.783	−0.018
Overall ($n = 35$)	0.760	0.765	+0.005

The hypothesis is confirmed. Adding BM25 raises acronym-query nDCG@10 by +0.063 and lowers non-acronym queries by −0.018; the two effects nearly cancel in the pooled mean (+0.005), which is why the aggregate looks negligible. BM25's contribution is therefore a targeted lift on a specific failure mode of dense retrieval (acronym matching), not a generic improvement. This is also a methodological point: aggregate metrics on a small benchmark can mask structurally different effects across subsets, so per-query-type reporting is necessary. The full per-mode breakdown is in the project repository [6].

Hybrid to hybrid+rerank: +12.9 pp. The cross-encoder is the second-biggest gain. It reads the query and each passage together, so it can distinguish a passage that genuinely answers the question from one that only shares a few words with it. For example, a passage about "taxi time prediction" in the context of delay modelling ranks above one that mentions the phrase only in a literature overview.

Recall@20 = 1.000. Every relevant document appears in the top-20 candidates that the cross-encoder sees. This means the first-stage retrieval is not losing anything; all remaining errors happen in the reranking step. The right direction for future improvement is reranking precision, not broader retrieval.

The choice of BM25+CE (hybrid+rerank) as the primary architecture is grounded in BEIR empirical evidence [33]: across 18 heterogeneous retrieval benchmarks, BM25+CE achieved the highest average nDCG@10 among configurations that do not require domain-specific fine-tuning data. This directly mirrors our deployment constraints: the KDC corpus is too small for bi-encoder fine-tuning, and no labelled training pairs exist.

5.2 External Validity: SciFact Benchmark

Tuning chunking and retrieval choices on a single 89-document archive carries a clear risk: the configuration may be overfitted to this corpus rather than reflecting a generally sound design. To test this directly, the same four retrieval configurations, with identical model weights and parameters, were evaluated on the SciFact dataset [34] from the BEIR benchmark: 5,183 biomedical abstracts, 300 scientific claim queries, binary relevance judgements. Table 9 reports the cross-dataset comparison.

Table 9. Cross-dataset comparison: KDC domain corpus vs. SciFact (BEIR), same model weights, four retrieval configurations

Configuration	KDC nDCG@10	SciFact nDCG@10
BM25-only	0.587	0.630
Dense-only	0.760	0.746
Hybrid (no rerank)	0.765	0.726
Hybrid + rerank	**0.894**	0.703

Two findings stand out.

Dense retrieval generalises; the reranker does not. BGE-large-en-v1.5 achieves nDCG@10 = 0.746 on SciFact without any domain adaptation, close to its KDC score of 0.760 and well above BM25 on both corpora. The cross-encoder reranker tells the opposite story: on KDC it adds +12.9 percentage points over the hybrid baseline; on SciFact it reduces nDCG@10 by 2.3 pp relative to dense-only. The reranker used here (MiniLM-L-6 fine-tuned on MS MARCO) was trained on web search queries and Bing passages. KDC queries are natural-language research questions over technical prose, which is close enough in distribution to MS MARCO that the reranker generalises. SciFact queries are scientific claims (e.g. "Protein X is essential for process Y") evaluated against biomedical abstracts: a substantially different distribution. The model has no signal for claim-evidence matching in that domain and introduces noise rather than signal. This is

consistent with the finding by Thakur et al. [33] that no single configuration wins across all 18 BEIR datasets: BM25+CE is the best on average across diverse retrieval tasks, but on individual biomedical datasets dense-only often outperforms it.

The KDC gain is real, not an artefact. The reranker's failure on SciFact actually strengthens the KDC result: the +12.9 pp gain on KDC is not a consequence of the benchmark being too easy or the qrels being biased toward the reranker's output. It reflects a genuine distribution match between the MiniLM training data and the KDC query-passage pairs. A domain-specific reranker fine-tuned on aviation or technical report data would likely improve both corpora; the current zero-shot configuration is already near optimal for the KDC domain.

Hybrid fusion is neutral to negative on out-of-distribution data. RRF adds BM25 signal to dense retrieval. On KDC, BM25 provides a small but consistent benefit (+0.5 pp) as a safety net for rare acronyms. On SciFact, adding BM25 reduces nDCG@10 by 2.0 pp, because BM25 term overlap on biomedical vocabulary adds noise rather than complementary signal. This confirms that BM25's contribution is corpus-dependent: it helps when the corpus vocabulary matches query terminology closely enough for exact-match to be informative.

5.3 Limitations

Corpus and annotation scope. The evaluation uses 89 documents and 35 queries (the full extent of the current KDC archive). Relevance judgements were produced by KDC domain experts grading query–document pairs from top-20 results across all four configurations. This scope is sufficient for the ablation study but a larger query set would improve statistical power.

Generation quality. The RAGAS evaluation (Section 4.2) covers the 35-query benchmark used for retrieval evaluation. The faithfulness score of 0.665 indicates that approximately one third of agent claims are inferred beyond what the retrieved passages explicitly state. RAGAS uses an automated LLM judge (GPT-4o-mini), which introduces its own alignment uncertainty: the judge's concept of faithfulness may not fully correspond to domain expectations in aviation research, where inference from regulatory standards or implicit domain knowledge may be considered appropriate by practitioners. A manual citation audit over a representative query subset would provide a complementary, domain-grounded estimate of faithfulness.

Chunking strategy. The sentence-window chunker does not exploit document structure: a chunk that spans the boundary between a Methods section and a Results section may contain sentences from two different argumentative contexts. The EDA confirmed 98.9% header detection, which means structure-aware splitting is technically feasible; the reason it was not adopted (variable section lengths degrading embedding consistency) is a design trade-off, not a constraint. A future iteration using hierarchical indexing (e.g., RAPTOR [28]) could address this without incurring the length-inconsistency penalty.

Speculative language in source documents. During the April 2026 demonstration, a sustainability query retrieved a document that mentioned the topic only in a speculative future-impact sentence in its conclusion. The embedding model scored the chunk relevant because the words co-occur with query terms, with no signal that the mention is peripheral: a context-conflicting hallucination [31]. Two responses: (1) editorial: researchers should avoid aspirational language in conclusions; (2) technical: contextual embeddings [1] ground each chunk in a document-level summary, reducing false positives from peripheral mentions. This failure directly motivates the contextual embedding recommendation in Future Work (Section 6.2).

Knowledge graph extraction quality. The LLM-based entity extractor produced semantically overlapping nodes, making graph traversal less reliable than retrieval-based methods; the graph is retained only as a corpus-overview feature and is excluded from the quantitative evaluation.

5.4 Reliability, Validity, and Generalisability

Internal reliability: retrieval evaluation uses deterministic BEIR metrics; generation evaluation runs five independent RAGAS passes on fixed agent outputs (low SDs 0.019–0.081 confirm stability). **Construct validity**: nDCG@10 is the standard primary metric in BEIR and TREC evaluations, capturing the operational requirement that relevant documents appear at the top of results. **External validity**: results are specific to the KDC corpus, but BM25+CE superiority across BEIR's 18 datasets provides external evidence that findings generalise to similar domain corpora; the evaluation framework itself is transferable.

5.5 Ethical and Societal Reflection

The primary ethical risk is hallucination in a safety-adjacent domain: a researcher acting on a hallucinated finding could indirectly influence an operational recommendation. The RAGAS faithfulness score of 0.665 shows this is not hypothetical: roughly a third of agent statements go beyond what the retrieved passages explicitly state. Citation grounding (REQ-ETH-1) reduces but does not eliminate this risk.

A second, subtler risk is **over-trust (automation bias)**: a fluent, well-cited answer invites users to accept it without checking the sources, especially under time pressure. Four design choices push against this: every claim carries a clickable citation so verification is one step away; the agent's reasoning trace is exposed so users can see *how* an answer was reached (REQ-TRANS-1); the system is explicitly scoped and labelled as a research-discovery tool, not a decision system (REQ-ETH-2); and a permanent disclaimer in the interface states that the assistant is a language model that can make mistakes and that findings must be verified against the cited documents (Figure 7). A residual **bias** risk also remains: both the embedding model and the reasoning model carry the priors of their general-domain training, which may under-rank minority or older findings in the archive; this is not measured here and is noted as a limitation. Societal benefits include accelerating aviation safety research by enabling practitioners to cross-reference 89 reports efficiently, potentially surfacing connections and gaps that time constraints would otherwise prevent.

6 Conclusion

6.1 Conclusion and Requirements Evaluation

This thesis presented the design, implementation, and evaluation of a domain-specific RAG system for the KDC aviation research archive across two DSR iterations. Iteration 1 established a hybrid BM25+dense+cross-encoder pipeline with a five-mode query interface and knowledge graph exploration tool. Iteration 2 replaced the static pipeline with an agentic reasoning orchestrator and four tools, retiring the knowledge graph as the agent subsumes its functionality with more flexible reasoning over retrieved passages.

Four contributions address the research question. First, the hybrid retrieval architecture achieves nDCG@10 = 0.894 and Recall@20 = 1.000 on the domain benchmark (+30.7 pp over BM25). Second, RAGAS generation quality across five judge runs: faithfulness 0.665, answer relevancy 0.724, context precision 0.736, context recall 0.628. Third, two validated query interfaces (five-mode Iteration 1 and agentic Iteration 2) covering Factual, Synthesis, Gap Analysis,

Idea Generation, and Connection queries. Fourth, a reproducible BEIR-style evaluation framework (deterministic qrels, four-configuration ablation, SciFact external comparison) transferable to other domain-specific RAG deployments.

Table 10 evaluates the requirements from Section 3 against the delivered system. The full origin-and-fulfilment matrix, with code-level evidence for each requirement, is maintained in the project GitLab wiki [6].

Table 10. Requirements fulfilment. F = fulfilled, P = partial, N = not implemented (deferred to future work).

Requirement	Status	Evidence / note
REQ-ORG-1 (local deployment)	F	Local agent and Docker compose implemented; generation quality measured on a local configuration (Table 6).
REQ-ORG-2 (research agenda)	F	*search_kdc_agenda* tool.
REQ-ETH-1 (citations)	F	Enforced in pipeline and agent prompts.
REQ-ETH-2 (scope)	F	Out-of-scope refusal validated (Section 4.3).
REQ-TRANS-1 (reasoning trace)	F	Thinking trace streamed in the UI.
REQ-FUNC-1 (five modes)	F	Iteration 1 mode picker; superseded by the agent.
REQ-FUNC-2 (≥89 docs)	F	89 documents indexed, 23,039 chunks.
REQ-FUNC-3 (web upload)	F	Bulk upload page.
REQ-FUNC-4 (<30 s)	P	Pipeline 2–4 s; agent 15–30 s.
REQ-FUNC-5 (iterative refinement)	F	Multi-turn agent loop.
REQ-FUNC-6 (agenda contextualisation)	P	Agent can use the agenda tool; no dedicated surfacing.
REQ-FUNC-7 (policy framing)	N	Policy-implications mode deferred (Section 6.2).
REQ-TECH-1 (BM25 baseline)	F	Included in the ablation.
REQ-TECH-2 (reproducible eval)	F	Deterministic BEIR + human qrels.
REQ-TECH-3 (Docker)	F	`docker-compose.local.yml`.

Of fifteen requirements, twelve are fulfilled, two partial, and one deferred. The partial and deferred items (agent latency, agenda surfacing, and a policy-framing mode) form a coherent agenda for production hardening rather than gaps in the core design.

6.2 Future Work and Recommendations

Five directions are identified for future development.

Contextual embeddings. Prepending a short document-level context summary to each chunk before embedding (Anthropic's contextual retrieval [1]) directly addresses the speculative-language retrieval failure identified in the Limitations section: a chunk whose embedding is grounded in the document's actual topic is less likely to match queries about peripheral mentions. Anthropic report a 35–49% reduction in retrieval failure rate relative to standard chunking. The implementation cost is 89 LLM calls at index time (one per document), cached after first ingestion. It is the most directly actionable retrieval improvement available.

Multimodal ingestion and the AIP. The stakeholder validation session (Section 4.3) identified the AIP as a high-value source that the current system cannot adequately ingest: its procedural content lives largely in aerodrome charts and approach diagrams that a text-only pipeline discards. KDC thesis reports themselves also contain trajectory plots, noise maps, and runway occupancy diagrams that the current index does not capture.

A promising near-term solution is Microsoft MarkItDown [18], a document conversion library that parses PDFs into structured Markdown and extracts embedded images as separate files. Each image can then be described by a vision model using a domain-specific prompt (figure type, the aviation concept shown, key quantitative findings), and the descriptions embedded alongside text chunks as `chunk_type: "figure"` entries. This slots into the existing ingestion pipeline without changes to the retrieval or generation layers. The current prototype is text-only; multimodal extension is the primary technical priority for a production deployment.

Iterative query refinement. Stakeholder feedback identified iterative search as a priority: researchers often need to narrow or reframe a query based on what an initial answer reveals. A multi-turn interface with the agent retaining context from earlier tool calls would further reduce the friction of exploratory research.

Policy implications mode. A dedicated query mode framing retrieved findings in operational or regulatory terms would serve LVNL specialists who need research translated into actionable recommendations rather than academic summaries.

Domain-adapted reranker and embedder refresh. As shown in the SciFact comparison (Section 5.2), the MiniLM reranker's effectiveness is distribution-dependent: it works well on the KDC corpus but fails to generalise to biomedical scientific claims. A cross-encoder fine-tuned on aviation or technical documentation query-passage pairs would improve robustness and would be the recommended path if a labelled training set becomes available through extended KDC use. The embedding model deserves the same periodic scrutiny: on the MMTEB benchmark [9], compact open models such as Qwen3-Embedding-0.6B now rank above BGE-large within the same size class while remaining locally deployable. The reproducible BEIR framework built here makes such a swap directly measurable: re-run the four-configuration ablation with the candidate embedder and compare nDCG@10 against the current baseline.

6.3 Ethical Responsibility Statement

The KDC Research Assistant is designed as a research discovery tool, not a safety-critical operational system. Deployment of AI-generated answers in an aviation operational context without human expert review would be inappropriate and is not recommended. The hallucination mitigation measures described in Section 2.4 represent best-effort safeguards

for a research prototype; a production deployment would require additional validation, a formal risk assessment, and explicit user guidance on the system's limitations.

A System Diagrams

The two diagrams referenced in the main text are reproduced here at full size. Each is placed on its own landscape page.

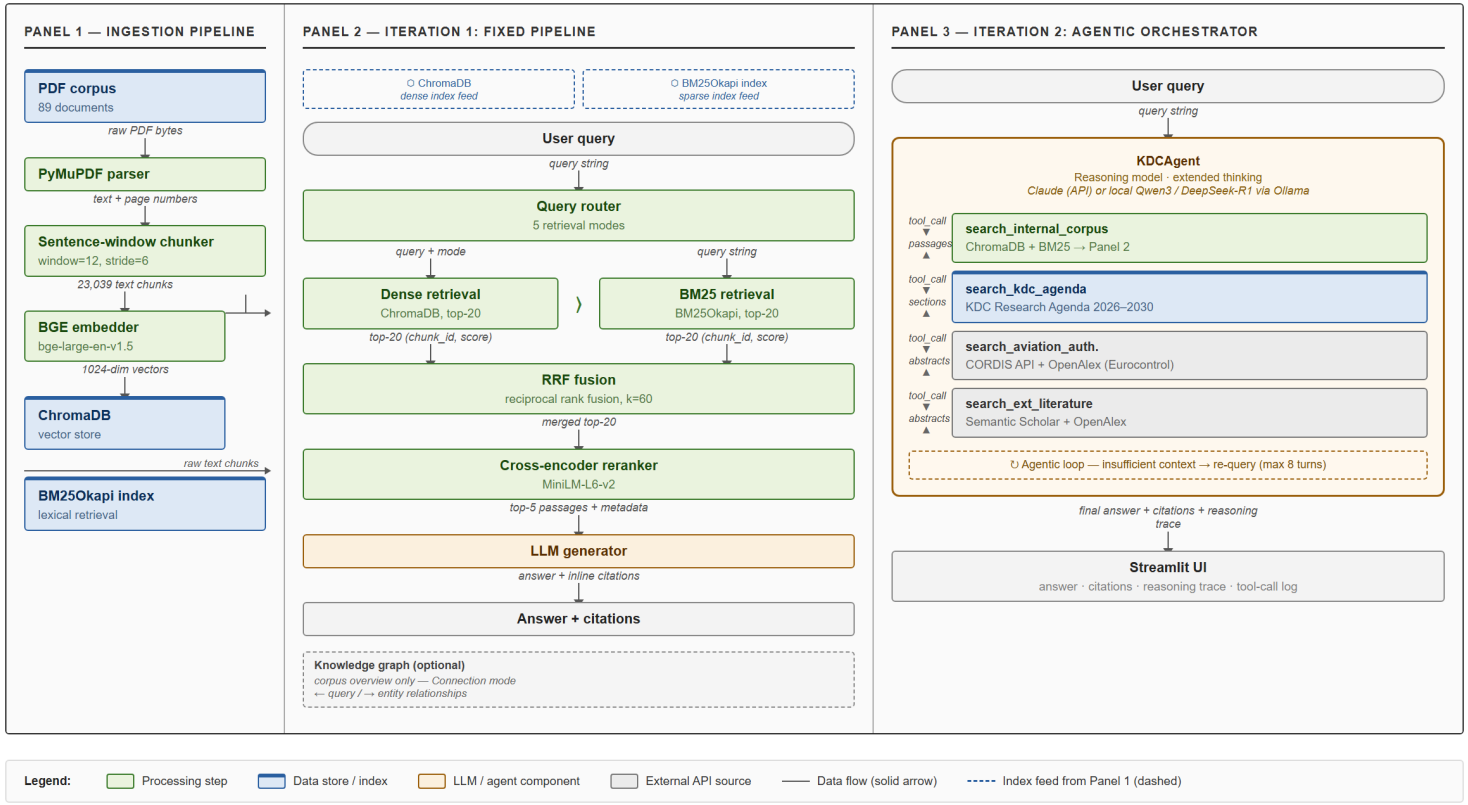

Fig. 3. System architecture. Ingestion (left) builds two indexes from the PDF corpus. At query time (right), dense and BM25 retrieval run in parallel, RRF merges their ranked lists, a cross-encoder reranks the top 20, and the LLM generates a cited answer from the top 5. Panel 3 shows the Iteration 2 agentic orchestrator, which wraps the Iteration 1 pipeline as one of four tools.

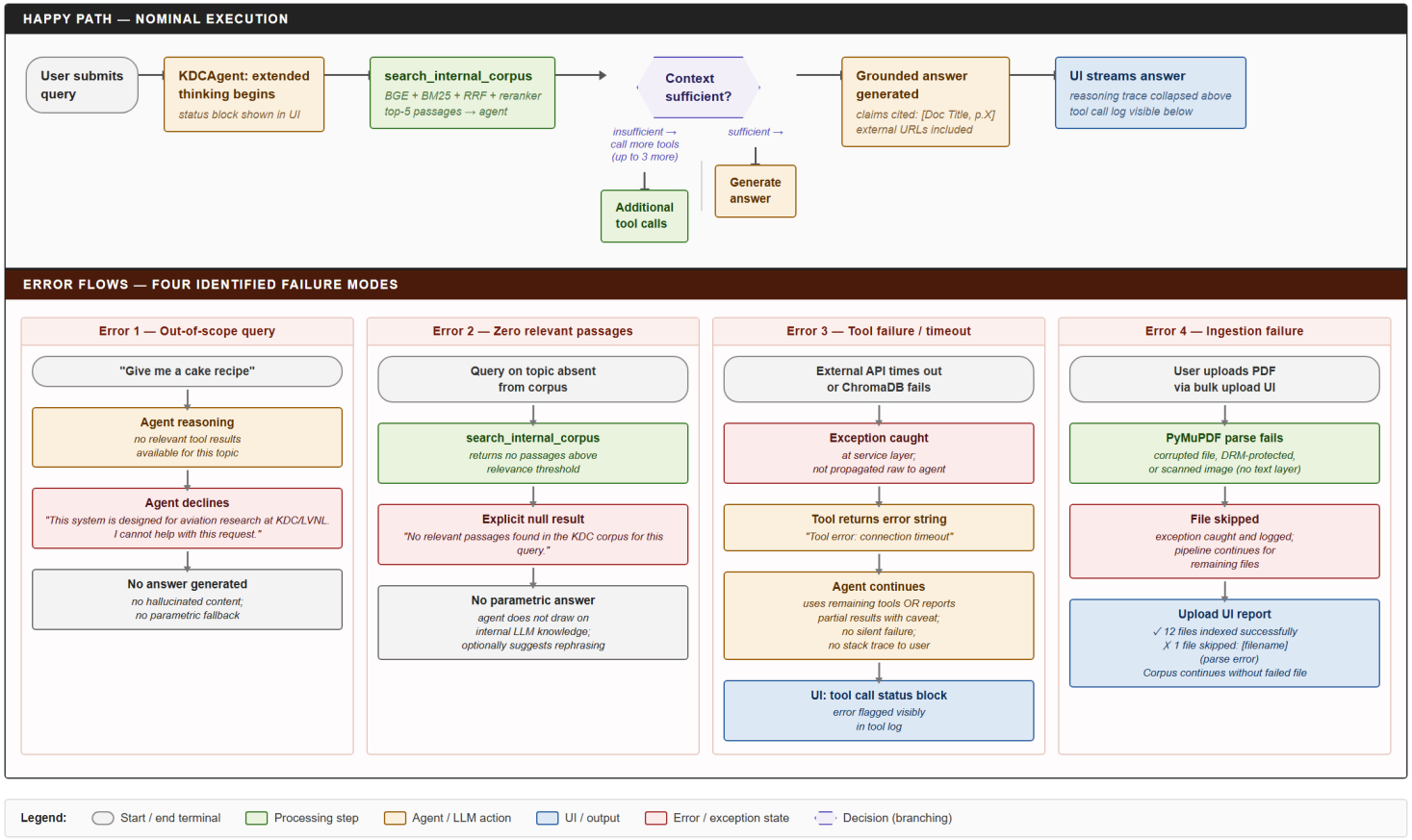

Fig. 4. Happy path and error flows for the KDC Research Assistant (Iteration 2). Top lane: the user's query triggers extended thinking; the agent calls *search_internal_corpus* first and evaluates context quality; if insufficient, it calls up to three additional tools before generating a grounded, cited answer. Bottom lane: four handled failure modes: out-of-scope queries produce an explicit refusal; absent corpus topics return a declared null; tool failures are caught at the service layer and reported in the tool-call log; PDF ingestion failures skip the offending file and report per-file status in the upload UI.

B Interface Screenshots

Screenshots of the two delivered interfaces. The full set, including the document library, upload, and landing pages of both iterations, is maintained in the project GitLab wiki (Interface Design page) [6].

References

[1] Anthropic. 2024. *Introducing Contextual Retrieval.* Technical Report. Anthropic. https://www.anthropic.com/news/contextual-retrieval Technical blog post.

[2] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv preprint arXiv:2310.11511* (2023). https://arxiv.org/abs/2310.11511

[3] Paul H. Cleverley and Simon Burnett. 2019. Enterprise Search: A State of the Art. *Business Information Review* 36, 2 (2019), 60–69. doi:10.1177/0266382119851880

[4] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 758–759. https://doi.org/10.1145/1571941.1572114

[5] DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948* (2025). https://arxiv.org/abs/2501.12948

[6] Sten den Hartog. 2026. KDC Research Assistant: Project Repository and Wiki. HvA GitLab. https://gitlab.fdmci.hva.nl/maai/2025-2026-sem2/smart-asset-management/sten-den-hartog-maai-portfolio

[7] Abhimanyu Dubey et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024). https://arxiv.org/abs/2407.21783

[8] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130* (2024). https://arxiv.org/abs/2404.16130

[9] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, et al. 2025. MMTEB: Massive Multilingual Text Embedding Benchmark. In *International Conference on Learning Representations (ICLR).* https://arxiv.org/abs/2502.13595

[10] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2024. RAGAS: Automated Evaluation of Retrieval Augmented Generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations.* Association for Computational Linguistics, 150–158. https://arxiv.org/abs/2309.15217

[11] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. ColPali: Efficient Document Retrieval with Vision Language Models. *arXiv preprint arXiv:2407.01449* (2024). https://arxiv.org/abs/2407.01449

[12] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* (2023). https://arxiv.org/abs/2312.10997

[13] David Hawking. 2004. Challenges in Enterprise Search. In *Proceedings of the 15th Australasian Database Conference (ADC 2004) (CRPIT, Vol. 27).* 15–24. http://david-hawking.net/pubs/hawking_adc04keynote.pdf

[14] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12 (2023), 1–38. https://doi.org/10.1145/3571730

[15] Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2024. Resources for Brewing BEIR: Reproducible Reference Models and Statistical Analyses. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 1431–1440. doi:10.1145/3626772.3657862

[16] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, 6769–6781. https://arxiv.org/abs/2004.04906

[17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 9459–9474. https://arxiv.org/abs/2005.11401

[18] Microsoft Corporation. 2024. *MarkItDown: Convert files and office documents to Markdown.* https://github.com/microsoft/markitdown Open-source document conversion library.

[19] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 2014–2037. https://arxiv.org/abs/2210.07316

[20] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. arXiv:1901.04085 https://arxiv.org/abs/1901.04085

[21] Koos Noordeloos and Evert Westerveld. 2026. Prototype demonstration feedback session. KDC/LVNL internal feedback session, April 2026.

[22] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24, 3 (2007), 45–77. https://doi.org/10.2753/MIS0742-1222240302

[23] Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring Attribution in Natural Language Generation Models. *Computational Linguistics* 49, 4 (2023), 777–840.

doi:10.1162/coli_a_00486

[24] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3982–3992. https://arxiv.org/abs/1908.10084

[25] Research Organization Registry. 2024. Eurocontrol — ROR entry. https://ror.org/02e0nhh54 ROR ID: 02e0nhh54.

[26] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[27] Sujoy Roychowdhury, Sumit Soman, H. G. Ranjani, Neeraj Gunda, Vansh Chhabra, and Sai Krishna Bala. 2024. Evaluation of RAG Metrics for Question Answering in the Telecom Domain. In *ICML 2024 Workshop on Foundation Models in the Wild*. https://arxiv.org/abs/2407.12873

[28] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. In *International Conference on Learning Representations*. https://arxiv.org/abs/2401.18059

[29] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv preprint arXiv:2302.04761* (2023). https://arxiv.org/abs/2302.04761

[30] Coen Schinkel. 2026. Artificial Intelligence for LVNL: Assessing AI Opportunities for Dutch Air Traffic Control. Graduation thesis, Amsterdam University of Applied Sciences, in collaboration with LVNL/KDC.

[31] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Shi, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. Retrieval-Augmented Generation: A Survey on Trustworthiness. *arXiv preprint arXiv:2403.16971* (2024). https://arxiv.org/abs/2403.16971

[32] Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V. Vasilakos. 2025. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. *arXiv preprint arXiv:2501.09136* (2025). https://arxiv.org/abs/2501.09136

[33] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-Shot Evaluation of Information Retrieval Models. In *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. https://arxiv.org/abs/2104.08663

[34] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7534–7550. https://arxiv.org/abs/2004.14974

[35] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2201.11903

[36] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 641–649. doi:10.1145/3626772.3657878

[37] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2210.03629

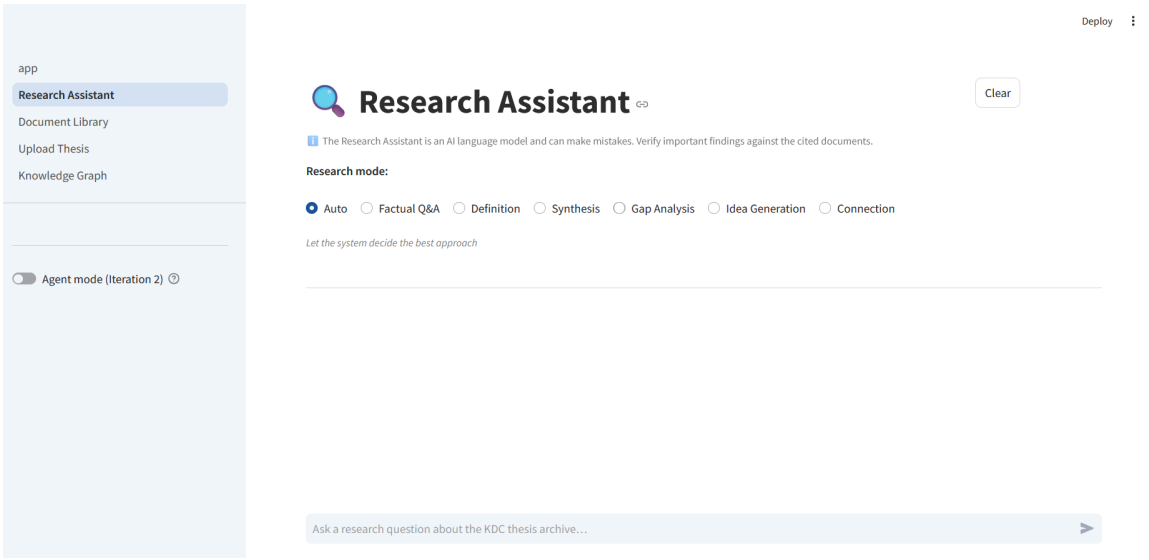

Fig. 5. Iteration 1 Research Assistant. The research-mode picker makes the query type an explicit user choice (with automatic routing as the default), the caption under the title states that the assistant is a language model that can make mistakes, and the sidebar offers the agent-mode toggle and knowledge graph page.

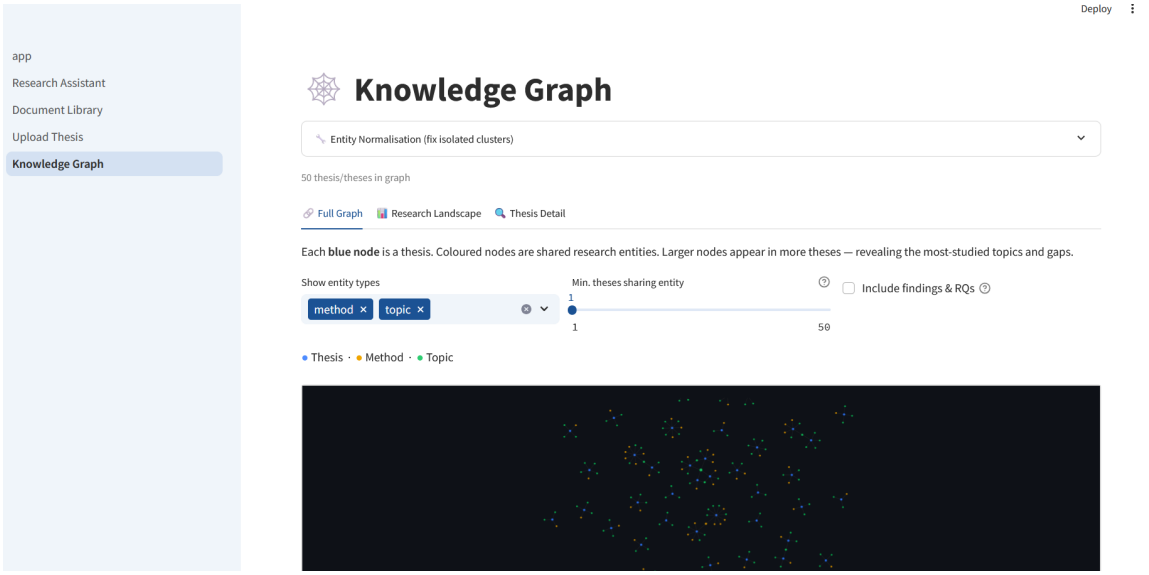

Fig. 6. Iteration 1 knowledge graph explorer: entity relationships extracted from the corpus, retained as a corpus-overview feature and retired as a retrieval mechanism in Iteration 2.

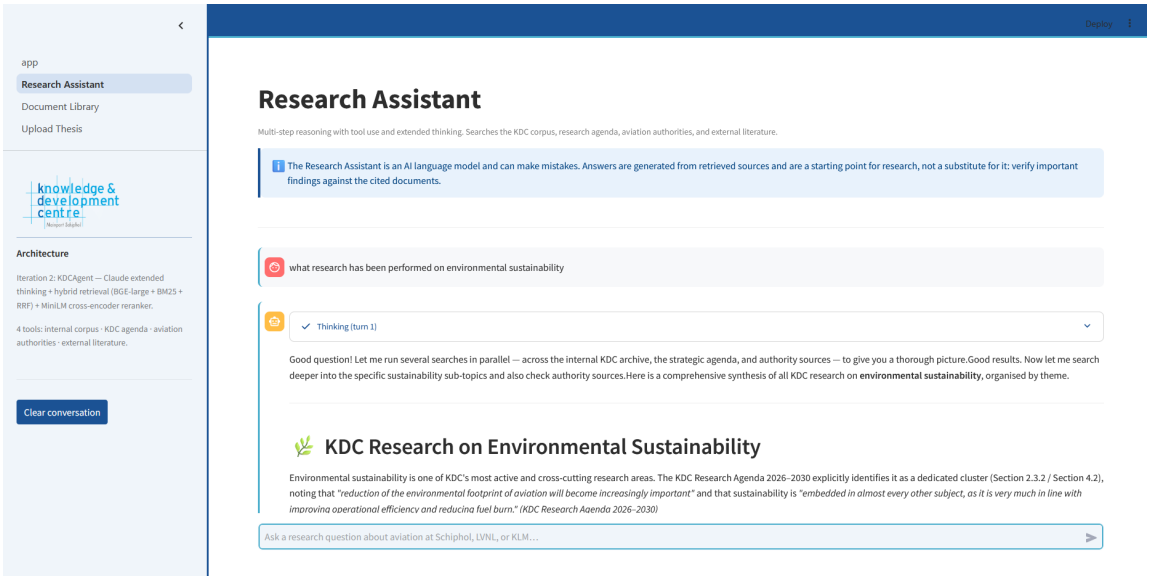

Fig. 7. Iteration 2 KDCAgent interface in the KDC house style. From top to bottom: the AI-limitations disclaimer banner, the user query, the collapsible extended-thinking trace (REQ-TRANS-1), and the generated answer grounded in the KDC Research Agenda.