Multimedia, RAG, and LLMs in High-Security Environments - a reference implementation for the AI2MMIR architecture

Stefan Wagenpfeil*, Angelina Chernikov

Department of Computer Science, Software Engineering and AI
PFH Private University of Applied Sciences, Weender Landstrasse 3-7, 37073 Göttingen, Germany

Abstract

This paper presents a proof-of-concept implementation for integrating Multimedia, Retrieval-Augmented Generation (RAG), and Large Language Models (LLMs) in high-security environments, with a particular focus on legal practice. The approach is grounded in the AI2MMIR reference architecture, demonstrating that advanced AI systems can be deployed entirely within an organization's perimeter, ensuring compliance with stringent privacy, provenance, and auditability requirements such as GDPR and the EU AI Act. The proposed solution leverages local LLMs, multimodal information retrieval, and vector-based semantic indexing to deliver trustworthy, explainable, and citation-first user interfaces. Empirical evaluation shows that local LLM+RAG pipelines provide strong privacy assurances based on architectural isolation and on-premises deployment, architecture-level support for machine-verifiable provenance mechanisms, and operational resilience, while supporting multimodal retrieval and effective knowledge management. User studies confirm the system's usability and value in daily legal workflows, despite minor performance trade-offs compared to cloud solutions. The architecture's modularity and extensibility allow for ongoing adaptation to evolving technological and regulatory landscapes. The paper concludes that the convergence of trustworthy AI, robust multimedia analysis, and privacy-preserving architectures is essential for responsible AI adoption in regulated domains, and highlights open challenges in cross-document synthesis, multimodal robustness, and scalability for future research.

Keywords

Information Retrieval, AI Augmentation, Retrieval Augmented Generation, Reference Architecture

1. Introduction

The integration of Artificial Intelligence (AI) and Multimedia technologies is rapidly transforming the landscape of knowledge management and is reshaping the way, users work with data. In many areas, this transformation is beneficial and delivers an enormous boost in efficiency and effectiveness. Particularly, when no legal or data protection policies must be considered. However, there are a number of industry and business areas, which are highly restricted regarding the use and the processing of data. Law firms, in particular, are confronted with the dual challenge of managing ever-growing volumes of digital documents, images, and communications, while simultaneously upholding the highest standards of client confidentiality, data protection, and evidentiary integrity. The promise of advanced AI systems, especially those based on large language models (LLMs) and retrieval-augmented generation (RAG), lies in their ability to accelerate legal research, automate document analysis, and provide context-aware answers to complex queries. However, these benefits are accompanied by significant risks and constraints in high-security and data-protected environments. Especially, in the European Union, the General Data Protection Regulation (GDPR) and the AI-Act represent a mandatory and binding framework [1, 2].

Considering the typical workflow in a European law firm specializing in cross-border commercial litigation, attorneys and paralegals routinely process sensitive contracts, pleadings, and evidence files,

TRI-IVIS - AVI 2026
*Corresponding and main author.
s.wagenpfeil@pfh.de (S. Wagenpfeil)
https://www.pfh.de (S. Wagenpfeil)
0000-0003-2100-7589 (S. Wagenpfeil)

many of which contain personal data, trade secrets, or privileged communications. The firm's clients expect not only legal expertise but also absolute discretion and compliance with the General Data Protection Regulation (GDPR) and professional secrecy obligations [1, 2]. In this context, the use of online LLMs, such as those provided by major US-based cloud vendors, is fundamentally incompatible with the firm's operational and legal requirements. Submitting confidential documents or client queries to external AI services risks unauthorized data transfer, loss of control over information, and potential breaches of both European and national law. The legal risks are further exacerbated by the extraterritorial reach of the US CLOUD Act, which compels US-based service providers to disclose data stored on their servers to American authorities, even if that data resides in Europe or concerns non-US citizens [3]. For a law firm, this means that any use of cloud-hosted LLMs or AI APIs could expose client information to foreign jurisdictions, undermining the trust relationship at the heart of legal practice and potentially rendering evidence inadmissible in court. As a result, many firms categorically prohibit the use of external AI tools for any matter involving sensitive or regulated data. These constraints create a pressing need for trustworthy, local AI solutions that can operate entirely within the firm's IT perimeter. Such systems must guarantee that no data leaves the organization, that all processing is auditable and explainable, and that the provenance and authenticity of Multimedia assets are preserved throughout their lifecycle. At the same time, the system must be flexible enough to support multimodal information retrieval, semantic search, and the integration of new AI capabilities as they emerge [4, 5].

To address these challenges, this paper employs the AI2MMIR reference architecture, outlined in [5], which provides a trustworthy and end-to-end protected modern environment. To ensure a rigorous and systematic exploration of trustworthy AI integration in high-security legal environments, this study adopts the Nunamaker methodology for information systems research [6]. The Nunamaker framework is well-established in the field of information systems and has been successfully applied to the development and evaluation of complex, socio-technical solutions, including those at the intersection of law, Multimedia, and artificial intelligence [7]. The methodology is structured around four interrelated phases, which also represent the sections of this paper: In section 2, the current state of the art and related work is presented according to the *Observation* phase of the methodology. The *Theory Building* phase is outlined in section 3, and a prototypical *Implementation* is demonstrated in section 4. A detailed *Evaluation*, particularly focusing on the use cases and policies in law firms, is given in section 5.

The contribution of this paper is deliberately architectural rather than algorithmic. We do not introduce new retrieval models, embedding techniques, or training methods. Instead, we provide a reference implementation that systematically integrates MMIR, RAG, and locally operated LLMs under real-world legal, security, and compliance constraints. The scientific value lies in demonstrating feasibility, limitations, and design trade-offs when deploying such systems in high-security environments, which are typically inaccessible to large-scale experimental evaluation.

2. State of the Art and Related Work

The current wave of foundation models has catalyzed renewed interest in deploying language technologies within high-security environments. In legal practice, the central tension is between capability and control: firms seek the productivity benefits of large language models (LLMs) while preserving stringent guarantees for privacy, provenance, and auditability. This section surveys the state of the art with emphasis on local LLMs, retrieval-augmented generation (RAG), Multimedia information retrieval (MMIR), and architectural blueprints that enable trustworthy integration in settings where external data transfer is impermissible. A first reference model for the alignment of Multimedia and AI is the AI2MMIR [5] (see Figure 1). It gives guidance and a framework for the implementation and evaluation of modern applications.

2.1. Local Large Language Models

Local LLMs have emerged as a viable alternative to cloud-hosted models, providing a deployment posture in which all inference and supporting data flows remain within an organization's perimeter.

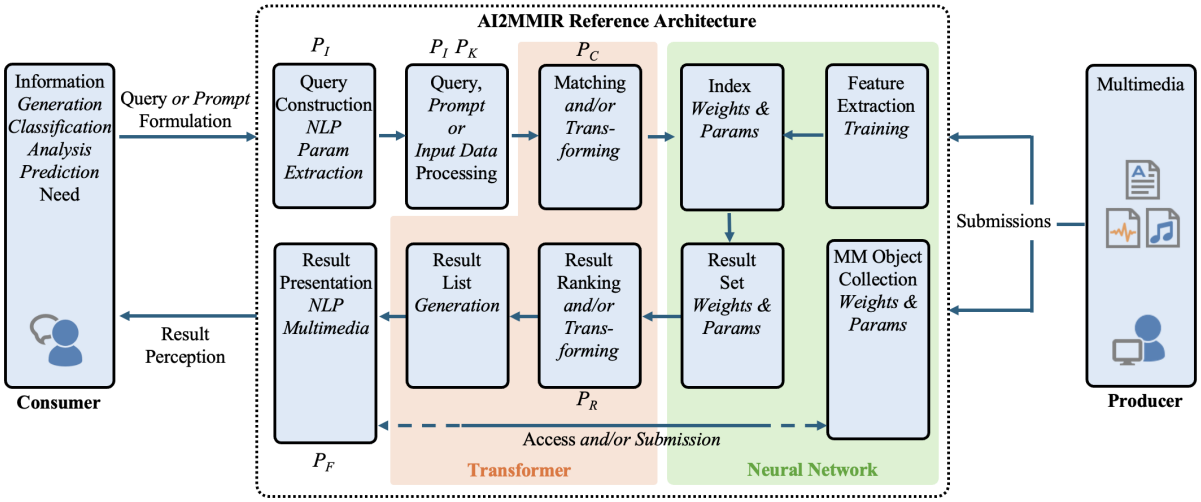

Figure 1: The AI2MMIR reference architecture [5].

The advantages over online LLMs are threefold. First, data sovereignty and compliance improve because prompts, retrieved contexts, and generated outputs never traverse third-party infrastructure, reducing exposure under GDPR, professional secrecy, and extraterritorial regimes such as the US CLOUD Act when US-based providers or sub-processors are involved. Second, latency and availability may be optimized for on-premises workloads, eliminating wide-area dependencies and enabling deterministic service envelopes. Third, controllability increases through tunable inference parameters, curated model catalogs, and hardened runtime configurations aligned with internal governance [8, 9]. While local operation can entail hardware and MLOps investments, recent experience indicates that balanced open models and quantization strategies enable practical deployments for document-centric use cases in professional services organizations [8, 9].

The rapid proliferation of large-language-model (LLM) families has made it essential for researchers to assess the relative strengths of locally deployable variants. Among the most widely adopted open-source architectures are Meta's LLaMA, the GPT-OSS-20B family, and Alibaba's Qwen. LLaMA [10] is a 7–65 B parameter series built on a transformer decoder with a 2048-token context window, trained on a curated mix of 1.4 TB of publicly available text and designed to deliver competitive zero-shot performance on a range of benchmarks while remaining lightweight enough for deployment on commodity GPUs. GPT-OSS-20B [11] is an open-source reimplementation of the 20-B parameter GPT-3 architecture; its training corpus spans 45 TB of diverse internet-scale text, and the model exhibits strong few-shot reasoning and generation capabilities, albeit at a higher computational cost than LLaMA for comparable parameter counts. Qwen [12] introduces a 14-B parameter Chinese-centric model that leverages a 8192-token context window and a mixture-of-experts style scaling strategy, achieving state-of-the-art results on both Chinese and multilingual benchmarks while maintaining efficient inference on modern GPUs.

Two further models that frequently appear in comparative studies are EleutherAI's GPT-NeoX-20B [13] and Mistral-7B [14]. GPT-NeoX-20B follows the GPT-Neo design but scales to 20 B parameters, providing a direct benchmark against GPT-OSS-20B while employing a more efficient training pipeline that reduces memory overhead. Mistral-7B, on the other hand, prioritizes architectural efficiency: it uses a 32-layer transformer with a 16-k token window and incorporates a novel attention-sparsity mechanism that allows the model to achieve competitive performance with fewer FLOPs, making it attractive for real-time inference scenarios. Together, these five models span a spectrum of design philosophies—parameter count, context length, training data diversity, and computational efficiency—offering a comprehensive landscape for evaluating local LLM deployments.

2.2. Retrieval Augmented Generation and Vector Databases

Retrieval-Augmented Generation (RAG) refers to a system design pattern in which the output of a generative language model is explicitly conditioned on information retrieved from an external knowledge source at inference time. In a canonical RAG pipeline, a user query is first transformed into an embedding, matched against a vector-based index of document representations, and the top-k retrieved passages are then injected into the model prompt to guide generation. Vector databases are a key enabling component of RAG architectures, but they are conceptually distinct. While RAG describes the overall retrieval-generation interaction pattern, a vector database is a storage and retrieval mechanism optimized for similarity search over high-dimensional embeddings. In this sense, vector databases implement the retrieval substrate, whereas RAG defines how retrieved evidence is incorporated into model inference. When this retrieval-generation principle is extended beyond text to include images, scanned documents, audio, or video, we refer to the approach as Multimodal Retrieval-Augmented Generation (MRAG). In MRAG, modality-specific encoders produce embeddings for heterogeneous media types, and retrieval results from multiple modalities are fused or aligned before being passed to the language model.

Retrieval Augmented Generation (RAG) complements local LLMs by conditioning generation on retrieved, organization-internal sources. The key idea is to replace implicit model world knowledge with explicit, auditable passages retrieved from a governed corpus, thereby improving factuality, transparency, and source attribution. In a canonical pipeline, a user query is embedded, matched against a vector index of document chunks, and the top-k contexts are concatenated with the prompt to steer the LLM's response. Evidence shows that RAG reduces hallucinations, increases answer faithfulness, and enables citation-first user interfaces that align with evidentiary workflows in law firms and other regulated settings [15, 16]. Multimodal extensions (MRAG) generalize this principle across text, image, audio, and video, using modality-specific encoders and cross-modal fusion to retrieve evidence from heterogeneous artifacts [16, 17].

Within the Multimedia community, MMIR has supplied the conceptual and algorithmic substrate for RAG-style systems by developing feature extraction, semantic indexing, and cross-modal retrieval across decades. The Generic Multimedia Analysis Framework (GMAF) provides a reference model for ingest, feature graph construction, and high-performance retrieval over large collections, including techniques such as Graph Codes and explainable semantic indexing [18, 19]. The GMAF allows insight and demonstration of MMIR processing steps and can be used as a starting point for scientific R&D projects. While the GMAF is an open source vector database for R&N purposes, various production ready databases serve a similar purpose: in general, vector databases are specialized data systems for storing and retrieving high-dimensional embedding vectors using similarity search, and they form a central component of retrieval-augmented generation (RAG) systems such as AnythingLLM [20, 21]. In the AnythingLLM ecosystem, LanceDB is the default choice: it is fully open source, embedded, and requires no separate server or configuration, which makes it particularly suitable for privacy-preserving and local deployments while still scaling to millions of vectors on disk [22, 23]. Chroma represents a lightweight, open-source alternative that is popular in LLM prototyping and smaller RAG setups due to its simple API and tight integration with Python tooling, although it is typically not designed for very large or highly concurrent workloads [24, 21]. For larger-scale or enterprise scenarios, Milvus is a prominent open-source vector database backed by the LF AI Foundation, offering distributed deployment, multiple ANN index types, and strong performance in comparative benchmarks, at the cost of higher operational complexity [25, 20]. Finally, Pinecone exemplifies the class of managed, proprietary vector databases: it provides a fully hosted, auto-scaling service with low latency and minimal operational overhead, making it attractive for production RAG systems, but introduces vendor lock-in and recurring costs compared to self-hosted open-source solutions [26, 27]. Comparative evaluations consistently show that these systems trade off ease of use, scalability, and control, which explains why AnythingLLM supports multiple backends rather than prescribing a single universal solution [22, 20].

2.3. Enterprise Knowledge Management

Related work on local LLM usage in enterprise knowledge management demonstrates that on-premises RAG can be realized with commodity tooling. Chernikov's thesis develops a fully local prototype that indexes internal course and policy documents, evaluates retrieval quality via precision/recall/F1, and explores parameter sensitivities such as chunk sizes and similarity thresholds. The implementation shows both the feasibility and the practical trade-offs of local deployments, including the need for robust OCR and layout-aware segmentation to support multimodal retrieval over scanned PDFs and figures [4]. Complementary practitioner reports and tutorials document local RAG stacks that combine a local model runtime with embedding, vector storage, and chat orchestration layers; these accounts reinforce the operational value of keeping data flows inside the perimeter while acknowledging resource constraints and the importance of careful evaluation [8, 9].

2.4. Tool Landscape

The tool landscape for local, RAG-enabled MMIR pipelines is now sufficiently mature to support legal practice scenarios. AnythingLLM provides a lightweight orchestration and user-facing layer for workspace management, ingestion, and retrieval-augmented chat, while remaining compatible with local model runtimes [28]. Ollama offers an on-device inference server that launches and manages open LLMs, exposing a stable HTTP API that integrates with retrieval front-ends [29]. LanceDB serves as a local vector database for high-dimensional embeddings with simple Python bindings and file-backed storage [30].

Feature extraction from PDF documents remains a challenging endeavour because of the heterogeneous layout, embedded graphics, and varying encoding schemes that can coexist within a single file. Binary Image Layout Parsers (BILP) [31] tackle this problem with a rule-based segmentation pipeline that identifies structural primitives—headings, tables, figures, and text blocks—by exploiting bounding-box heuristics and layout fingerprints; its approach has proven particularly effective on densely formatted scientific articles where conventional text-only parsers falter. Complementary to layout analysis, Tesseract [32] offers a neural-network-based OCR engine that converts scanned images into searchable text, and its recent LSTM-based models have been shown to outperform legacy OCR systems on both printed and handwritten documents. While dedicated parsers such as PDFMiner [33], PyMuPDF [34], and Apache PDFBox [35] provide robust text extraction pipelines, they often struggle with complex table structures or multi-column layouts; BILP's layout-aware heuristics and Tesseract's OCR capabilities together offer a systematic advantage for comprehensive feature extraction.

2.5. Legal Context

The deployment of large language models (LLMs) in legal practice is subject to particularly strict regulatory, ethical, and organizational requirements, as legal work routinely involves personal data, privileged communications, and highly confidential information such as lawsuit documents or contractual materials. From a data protection perspective, the EU General Data Protection Regulation (GDPR) establishes the central legal framework. Any processing of personal data through LLMs must be grounded in a lawful basis under Art. 6 GDPR and adhere to the core principles of lawfulness, fairness, transparency, purpose limitation, and data minimization (Art. 5 GDPR) [36]. In legal settings, this typically implies that LLM usage must be strictly limited to defined purposes (e.g. document analysis or summarization) and that personal data is not repurposed or retained beyond necessity. Where special categories of data are involved, such as data revealing legal disputes or criminal allegations, the heightened protection requirements of Art. 9 GDPR apply, often necessitating additional safeguards or explicit legal authorization [36]. Beyond general data protection law, the EU Artificial Intelligence Act (EU AI Act) introduces a risk-based regulatory regime for AI systems that is directly relevant to legal applications. While general-purpose LLMs are regulated primarily through transparency and governance obligations, their use in legal decision-support or advisory contexts may qualify as high-risk AI systems if they significantly influence access to justice or legal rights. In such cases, providers and

deployers must implement risk management systems, ensure high-quality training data, provide human oversight, and maintain technical documentation and logging capabilities as required by the AI Act [2]. Even when LLMs are used merely as assistance tools, the AI Act emphasizes that responsibility remains with the human professional, reinforcing the principle that AI must not replace legally accountable decision-making.

A critical compliance dimension concerns privacy, client confidentiality, and professional secrecy. Legal professionals are bound by statutory confidentiality obligations (e.g. attorney–client privilege), which means that confidential lawsuit information must not be disclosed to third parties or used to train external models without robust contractual and technical guarantees. Consequently, many compliance strategies favor on-premises deployment, private LLM instances, or strict configurations that prevent data from being stored, reused, or transferred outside the controlled environment. From a technical and organizational standpoint, this aligns with the GDPR's requirements for data protection by design and by default (Art. 25 GDPR) and for appropriate security measures, including encryption, access control, and auditability (Art. 32 GDPR) [36].

Finally, transparency and accountability toward clients and affected individuals are central elements of lawful LLM usage in legal contexts. Clients must be informed if AI systems are used in processing their data or supporting legal analysis, and they must retain the right to human review and correction where automated outputs are involved. Misleading reliance on LLM outputs, opacity of reasoning, or unverified hallucinations can not only undermine legal quality but also create liability risks under professional and consumer protection law. Overall, the emerging European legal framework makes clear that LLMs may be used in legal practice only as assistive, well-governed tools, embedded within comprehensive compliance settings that prioritize data protection, confidentiality, human oversight, and the protection of fundamental rights [2, 36].

2.6. Summary

Recent advances in foundation models have enabled the deployment of large language models in security-critical domains such as legal practice, provided that architectural safeguards address privacy, transparency, and regulatory compliance. The state of the art demonstrates that local LLM deployments, when combined with retrieval-augmented generation and vector-based semantic indexing, mitigate key risks by grounding generation in auditable, organization-internal sources. Research in multimedia information retrieval supplies the conceptual foundation for multimodal retrieval, explainable indexing, and cross-modal evidence integration, as reflected in reference architectures such as AI2MMIR and GMAF. Contemporary tooling makes it feasible to assemble end-to-end local pipelines for enterprise knowledge management, while empirical studies highlight both effectiveness gains and remaining limitations. Overall, current work converges on assistive, RAG-based architectures as the most robust approach for legally compliant LLM usage under the GDPR and the EU AI Act. In the next section, the modeling and tool selection for a reference implementation of the AI2MMIR architecture in a legal context will be presented.

3. Modeling

This section introduces a socio-technical modeling of trustworthy Multimedia-AI integration for high-security environments by adopting the AI2MMIR as the baseline reference architecture and extending it into a RAG-enabled, locally operated system. The objective is to preserve the proven strengths of MMIR while adding the trust and trustworthiness guarantees required by law firms. In this extension, a locally installed large language model (LLM) consumes evidence retrieved from vector databases index via Retrieval-Augmented Generation (RAG), and the combined pipeline surfaces transparent, provenance-aware answers with auditable behavior [19, 18]. According to the results of the previous analysis and literature review, the following selection of tools and technologies is employed for this modeling and aligned with the AI2MMIR reference architecture:

- **Ollama**: selected to cover the *Natural Language Processing* part of the AI2MMIR.
- **Anything LLM**: selected to represent the *Result Presentation* phase. Further more, this tool allows the configuration of the RAG and model integration.
- **Lance DB**: an existing vector database compliant to the AI2MMIR phases *Query Processing, Matching, Indexing, Ranking*, and *Retrieval Result for Augmented Generation*
- **BILP and Tesseract**: represent candidates for the *Feature Extraction* phase
- **gpt-oss-20b**: selected as a generative AI large language model for the phase *LLM / GenAI*, due to its appropriate size and result quality
- **Infrastructure, Browser, SFTP**: for convenience, the user interface is browser-based, and as *MM Object Collection*, a file system accessible via SFTP is employed.

Ollama is employed as the local large-language-model runtime layer within the AI2MMIR architecture. Its primary role is to provide a controlled, on-premises inference service for open LLMs, exposing a stable HTTP API that can be consumed by higher-level orchestration components such as AnythingLLM. Compared to alternative local inference frameworks (e.g., LM Studio, text-generation-webui, or bespoke vLLM deployments), Ollama was selected for three reasons: (i) its minimal operational footprint and ease of installation, (ii) its deterministic local-only execution model without implicit telemetry or cloud dependencies, and (iii) its seamless compatibility with RAG-frontends commonly used in enterprise settings. These properties make Ollama particularly suitable for high-security legal environments where reproducibility, auditability, and strict data locality are mandatory.

The gpt-oss:20b model is an open-weight, transformer-based large language model with approximately 20 billion parameters, comparable in scale to GPT-NeoX-20B–class architectures. The model provides strong instruction-following, summarization, and semantic reasoning capabilities, which are central to legal knowledge work such as document analysis, timeline construction, and cross-document synthesis.

From a deployment perspective, gpt-oss:20b represents a pragmatic trade-off between model capacity and operational feasibility. While smaller models often exhibit insufficient reasoning depth for complex legal texts, significantly larger models impose hardware and energy requirements that are impractical for typical on-premises law-firm installations. In combination with retrieval-augmented generation, gpt-oss:20b delivers sufficient linguistic competence while allowing deterministic, fully local inference on commodity GPU hardware, making it well-suited for the proposed high-security legal use case.

Based on this selection, the complete AI2MMIR reference architecture can be represented. An overview is given in Figure 2.

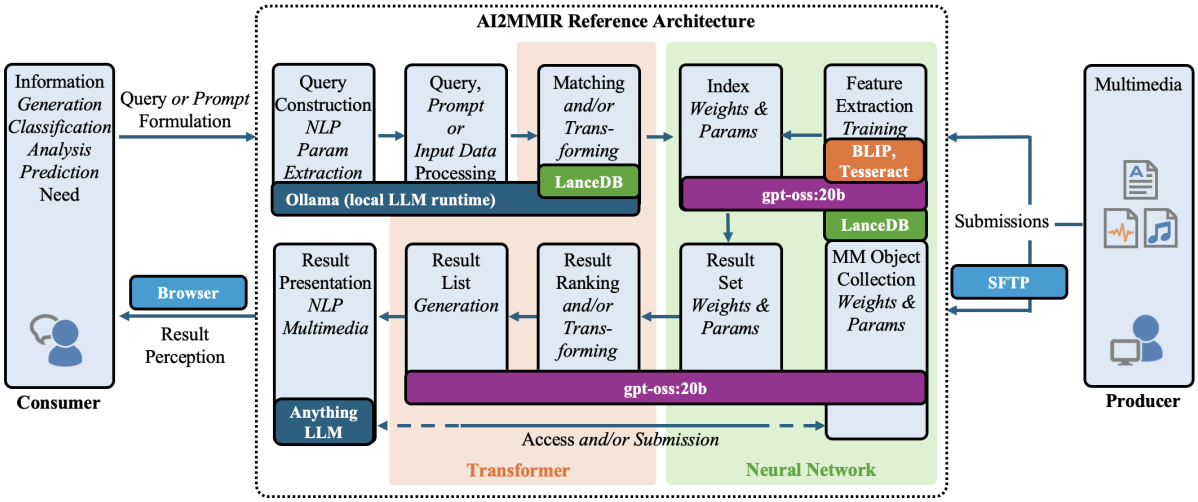

Figure 2: Alignment of the selected tools with the AI2MMIR reference architecture.

This modeling demonstrates two aspects: 1) it is possible to align industry- and production-ready tools with the AI2MMIR reference architecture. 2) for high-security environments, local LLM and MMIR solutions exist, that can serve the demands on security, protection, trust, and compliance.

The benefits of this modeling approach are especially clear in high-security settings. Because all inference, embedding, and retrieval steps run locally, data sovereignty is preserved—no prompts, contexts, or outputs leave the premises. This local posture also eases compliance with AI-specific regulations (AI2MMIR), professional secrecy requirements, and protects against extraterritorial cloud-provider rules. Transparency and explainability are enhanced: each answer is paired with citations, similarity scores, and provenance dialogs that legal professionals can scrutinise. Factuality improves through Retrieval-Augmented Generation (RAG), which compels the model to synthesize from retrieved passages rather than relying on latent priors. Multimodal robustness is bolstered by standard MMIR capabilities: OCR and layout-aware segmentation yield higher-fidelity text from scans, image descriptors help localise visually salient regions, and feature graphs preserve inter-document relationships for cross-document navigation. Operational resilience is strengthened by eliminating wide-area dependencies; latency and availability can be tuned to local workloads, and inference parameters can be adjusted to reduce stylistic variability and promote concise, citation-first outputs [16, 19, 4].

From a governance perspective, Ollama and Anything LLM both deliver audit and governance features that generate tangible artifacts for risk-management frameworks and internal policies. Recorded interactions provide traceability for incident response and model-behavior analysis. Identity bindings and role-based access control restrict access to sensitive corpora and capabilities to authorized users only. Provenance metadata documents origin and transformation steps, thereby supporting legal admissibility. The models' modular architecture aligns with evolving tools and standards: new encoders, retrievers, or verification components can be added without compromising trust semantics, while incremental upgrades to local LLMs or indexing structures can be adopted over time [5, 18].

To validate the modeling, a proof-of-concept implementation has been made and evaluated. The outline of this implementation is presented in the next section 4, a detailed evaluation in the high-security context of a low firm is given in section 5.

4. Implementation

This section demonstrates a fully local implementation of the modeled architecture by adapting and extending the prototype developed in Chernikov's thesis for high-security organizational settings. The goal is to show how a law firm can operate a Retrieval-Augmented Generation (RAG) pipeline strictly on-premises, combining mature Multimedia ingest and indexing with a locally hosted large language model. The implementation is designed to keep all document flows, embeddings, prompts, and generated outputs inside the firm's perimeter, while providing provenance, auditability, and citation-first answers that align with evidentiary practice [4]. This section is divided into prerequisites and hardware setup (see subsection 4.1), setup and configuration (subsection 4.2), the workspace setup (subsection 4.3), and the user interface in subsection 4.4.

4.1. Prerequisites and Hardware

For the system setup, we used PC with a Intel Core i9-990k CPU, 3.6 MHz with 8 Cores. It has 64 GB RAM, a 1TB SSD harddrive and a Nvidia RTX 4070 GPU with 12 GB RAM. As operating system we chose Windows 11 Pro and installed the containerization platform Docker. All components are bought by standard vendors and are standard hardware components, with no specific focus on AI or Machine Learning. The total cost for such a system is about 2.500-3.000 Euro.

4.2. Setup and Configuration

The setup of AnythingLLM is quite simple, as based on the Docker architecture, a corresponding configuration file can be used (see Listing 1) and most of the remaining configuration is done automatically or with the help of installation wizzards. The initial user interface is reachable at http://yourpc:3001.

Listing 1: Docker configuration for Anything LLM

```
# docker-compose.yml
services:
        anythingllm:
        image: anythingllm/anythingllm:latest
        container_name: anythingllm
        restart: unless-stopped
        ports:
                - "3001:3001"    # internal App-Port > Host-Port
        environment:
                PORT=3001
                JWT_SECRET=xxx
                ADMIN_EMAIL=admin@yourwebsite.com
                ADMIN_PASSWORD=very-strong-password
                SERVER_URL=https://yourpc    # or public IP
        volumes:
        - ./data:/app/storage
```

AnythingLLM can be used with various vector databases and various LLM providers. For this setup, and according to the modeling, we selected Ollama as a LLM provider. Ollama has been installed with a setup.exe program available from their website (https://ollama.com/download) and automatically been detected by AnythingLLM. Ollama provides a list of available pre-trained models on their website (see Figure 3), where we selected and downloaded the "gpt-oss:20b" model.

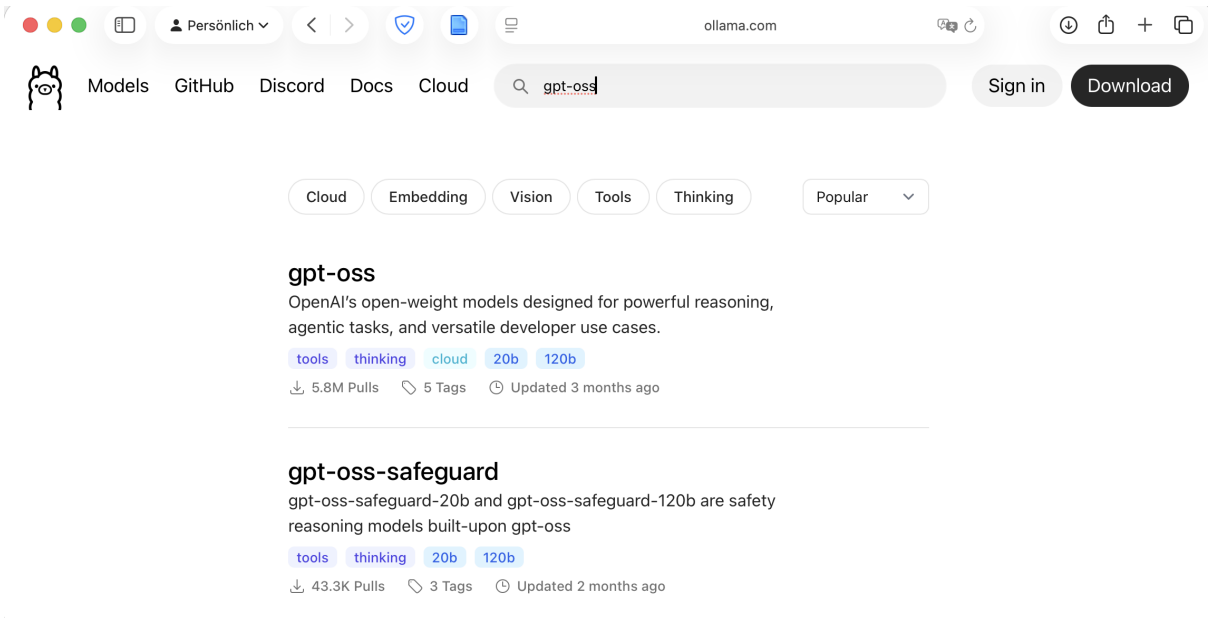

Figure 3: Download pre-trained models for Ollama.

Also, the vector database LanceDB is directly shipped with AnythingLLM and didn't need further initial configuration. This means, that the basic setup of such a local AI2MMIR compatible solution, is quite simple.

4.3. Workspaces

To address compliance demands, the concept of *Workspaces* is employed. A workspace is a separate container of both MM Objects (in the form of a lance database) and a separate LLM configuration. For each workspace, an administration user can invite other users, in case of a law firm: all attorneys and

co-workers responsible for a specific client. This protects the workspace from unpermitted use and ensures that all correspondence or interaction with this workspace can directly assigned to this client. Furthermore, auditing and tracing setup can be performed on a workspace level, so that any interaction with the local LLM and its attached RAG folder is transparent, auditable, documented, and compliant. The configuration of such a setup is shown in Listing 2 for an exemplary client with client-number 250324.

Listing 2: Workspace configuration with separate RAG instance

```
# workspaces/client_250324/workspace.yaml
name: ws2
vector_store:
        type: lancedb
        path: db_lance.db
        table: embeddings_wclient_250324
```

Figure 4 shows the workspace structure of the setup. On the left side, one workspace per client is created, each of which is attached to an individual vector database. All documents uploaded to the vector database's folder are automatically available for all users as "knowledge" within this workspace. This means, when users prompt a request to this workspace, all existing data is automatically available through this RAG configuration. On the right of Figure 4, such an interaction is shown.

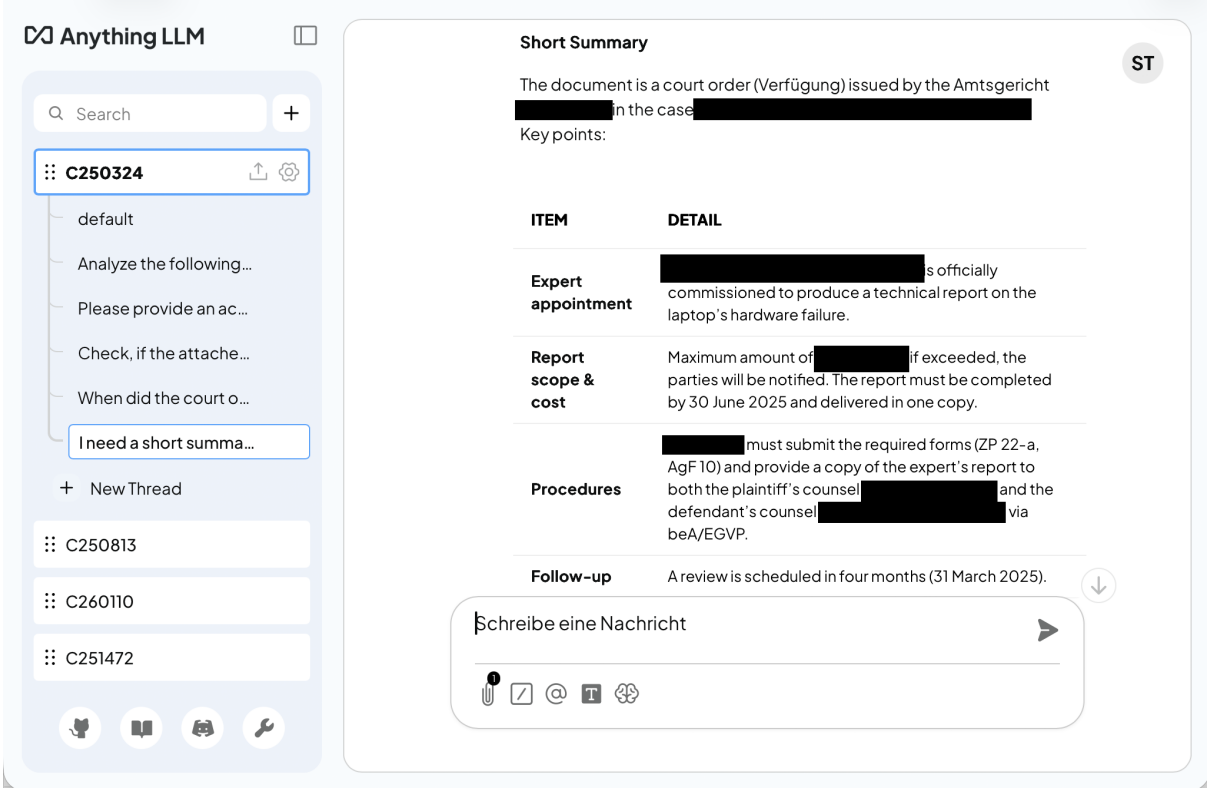

Figure 4: Workspace structure per client with access rights. Including a sample result of a user interaction.

Compliance and traceability is ensured by the workspace's audit settings. In the default configuration, all chats are already traced and documented in the administration area (see Figure 5).

4.4. User Interface

It has been decided to use a browser-based user interface, which makes the deployment very easy, so that each user can log in to a LLM-interface, that looks very similar to other state-of-the-art solutions,

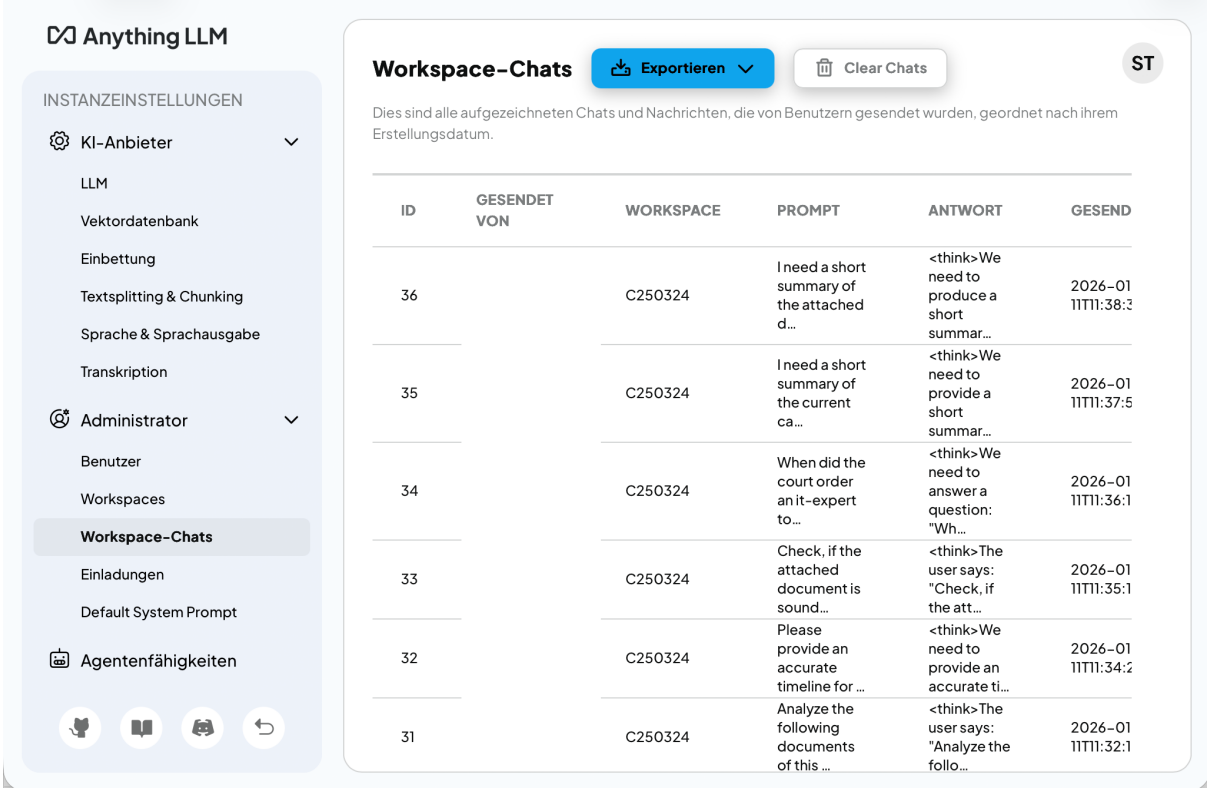

Figure 5: Compliance and audit / traceability per workspace.

like Chat GPT, Micorsoft Copilot, or Google Gemini. In Figure 4 such an interaction has allready been shown. However, to ensure trust and to give lawyers the opportunity to validate and/or understand why the LLM suggested a solution, not just the references to the local RAG folder are given, furthermore, also a detailed trace showing the LLM's "thinking" explains, why the given answer has been presented. This extends the *Result Presentation* phase of AI2MMIR and is shown in Figure 6.

Users only see the workspaces they are invited to, i.e. only the clients, which are assigned to them. By clicking on the workspace, the system automatically switches the RAG database to ensure, that the documents of this specific client are available in the context. Thus, lawyers can easily navigate between all of their active clients. Additionally, the user interface allows to upload additional files relevant for a single interaction session (see Figure 6 at the bottom). The content of these files is not added to the RAG system, but temporarily processed and available for the interaction. Thus, lawyers can use these individual files to compare the current case with other similar cases and won't harm the content of the RAG system.

Summarizing this, the proof-of-concept implementation demonstrates, that the proposed model can be implemented using state-of-the-art technologies, tools, and hardware. This means, a local, protected, and trustworthy AI2MMIR solution for high-security contexts, such as law firms, can be achieved. Furthermore, this section shows, that the required skillset for setup and operation is minimal and thus applicable for many contexts where high-security AI and/or MMIR processing is mandatory. In the next section, an evaluation from a users' point of view is given.

5. Evaluation

In this section, our evaluation of the proof-of-concept implementation is presented. To address the most important aspects, it is split into three sections, where different evaluation methods are employed: subsection 5.2 shows details about the inferencing performance, i.e. how fast answers to prompts can be calculated. Subsection 5.2 measures the accurracy of the RAG process based on exemplary user

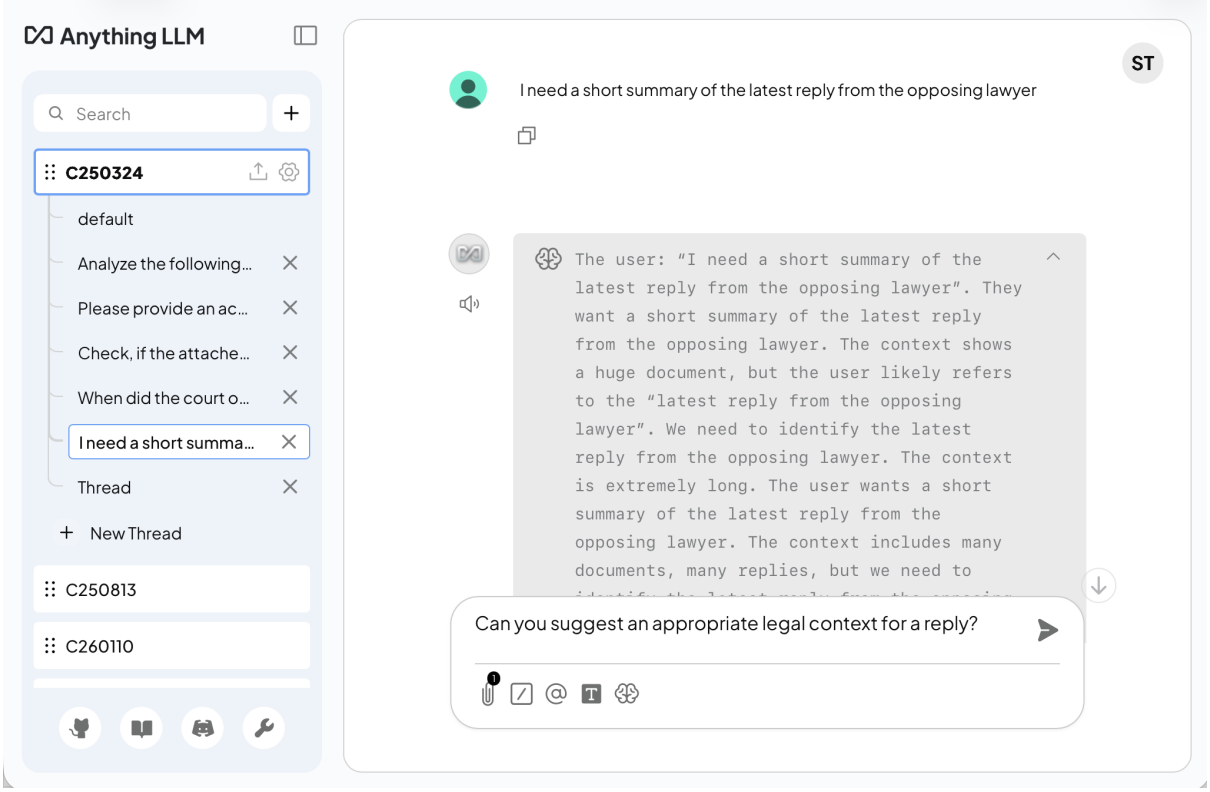

Figure 6: Traceability and explainability to ensure trust and validations.

interactions, and subsection 5.3 finally gives more insight about the users' perception of the overall solution.

5.1. Inferencing Performance

To measure the performance of inferencing processes, various metrics can be employed. However, as inferencing highly depends on the selected model, the hardware, the infrastructure, or the attached RAG components, most of such results are hardly comparable. Furthermore, from a users' perspective, the most important metric is the time-to-result, which is dependent from the complexity of the prompt and the requested result. As the complexity of both the prompt and the result can be measured in *tokens*, in the industry, the *token/second* metric is widely spread and gives a good indicator for typical LLM interactions. Here, two major steps are measured: the TTFT (time-to-first-token) and the TPOT (time-per-output-token) during inferencing. To show the perceived performance, we compare our local LLM installation running on the specified hardware setup (see subsection 4.1) with standard LLM services. The results are shown in Table 1:

Model	TTFT (ms)	TPOT (ms)	Tokens/s
Chat-GPT-4, Claude, Gemini Pro, Llama, etc.	0,3-1,5	20-30	30-60
Chat-GPT-3.5	0,3-1,0	8-12	80-120
Gemini Flash	0,3-1,0	10-20	50-100
Our setup	1,7-3,2	13-35	28-73

Table 1
Comparison TTFT, TPOT and Tokens/s for various LLMs.

The reported performance ranges for commercial cloud models are based on publicly available provider documentation and consolidated benchmark reports, whereas measurements for the local setup were obtained empirically on the described hardware. These results show, that the basic difference between our local setup and common online models is the TTFT, i.e. it takes longer until the users' request is processed. However, the processing as such is almost in the range of existing online models. This means, that from a performance perspective, the presented setup can be used without major problems.

5.2. Result Effectiveness

To evaluate RAG effectiveness, a manually curated gold standard was constructed. The underlying document corpus consisted of twenty real-world legal case files used in daily practice, each comprising approximately 100 pages of pleadings, correspondence, and evidentiary documents. For each of the 25 evaluation queries, domain experts (licensed legal professionals) manually identified the source documents and specific passages that contained the correct factual answers. These annotations defined the ground-truth relevance set for each query. Retrieved passages were considered relevant (true positives) if they originated from the annotated documents and directly supported the expected answer, and non-relevant otherwise. Precision, recall, and F1 scores were then computed based on these expert relevance judgments, following standard information-retrieval evaluation practice.

To evaluate the effectiveness of the results, we focused on the RAG performance of the model. Therefore, we uploaded a total of 20 documents (with an average of 100 pages) to a workspace and formulated a total of 25 specific questions, each of which address one or more documents. Expected answers, i.e. in which document and at which page of the document a match sould be found, have been curated manually. Examples of such queries are *"when has the court confirmed case XXX"* or *"what did Mr. XXX reply in January 2025"*. Each query has been processed 4 times to eliminate bias and hallucinations. Thus, a total of 100 samples has been calculated. As metric we employ a standard *Precision & Recall* emasure based on true/false positives/negatives.

In Table 2, the results of this experiment are shown for a subset of 10 questions. Column *No* is the number of the experiment, *Tot* is the total number of occurences of the result, *Pos* is the number of correct occurences (e.g, acoording to above's question, all dates in "January 2025" would be part of *tot*, while only few of them would provide the correct answer). Column *TP* indicates the true positives, i.e. the correctly found matches, *FP* show the false positives, i.e. matches that have been found but are not correct. Column *TN* represents the true negatives, and *FN* the false negatives accordingly. Finally *P* is the precision and *R* the recall value of the experiment, where $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$. The *F1* score gives the average mean as $F1 = 2 * \frac{P*R}{P+R}$

No.	Tot.	Pos.	TP	FP	TN	FN	P	R	F1
1	23	10	8	4	9	2	0.667	0.800	0.727
2	25	18	15	5	2	3	0.750	0.833	0.789
3	18	12	9	2	4	3	0.818	0.750	0.783
4	12	10	9	1	1	1	0.900	0.900	0.900
...									
12	42	12	10	12	18	2	0.455	0.833	0.588
13	38	20	14	5	13	6	0.737	0.700	0.718
14	36	24	12	2	10	12	0.857	0.500	0.632
...									
23	8	8	6	0	0	2	1.000	0.750	0.857
24	12	10	8	2	0	2	0.800	0.800	0.800
25	10	10	9	0	0	1	1.000	0.900	0.947

Table 2
Precision & Recall as evaluation of effectiveness.

Compared to other MMIR processes, the values for precision, recall, and F1 are neither good nor bad.

They are in the typical range of retrieval effectiveness. This means, that also the effectiveness of our local setup meets the average industry requirements.

5.3. Users' Perception

This final experiment was conducted in the form of user tests, where specific tasks had to be performed with our implementation. The tasks included the search for information in a client's documents, uploading documents, proof-reading of self-written content, summarization and timelining, as these are typical tasks in a law firm. Furthermore, these tasks cannot be performed with online models, as they all require access to confident and protected documents. In total, five users (lawyers and assistants) worked with the system, results were collected and discussed in a workshop after the test. The major key points can be summarized as follows:

- using a local LLM+RAG architecture is a huge advantage for the work in a law firm
- the provided user interface is easy-to-use and easy-to-understand, as it is similar to known online chatbots
- the performance of the system can be improved. However, all users ranked this as minor due to the big advantages for their daily work
- the effectiveness was fine for the specific tasks. All users were already aware, that LLM results must be double-checked and can be wrong. Therefore, the current effectiveness was evaluated positive
- document summaries, timelines, and first legal estimations, were accurate and satisfying

Therefore, the overall evaluation in terms of performance, effectiveness and usability provides positive results. Although the current setup is slightly slower than cloud solutions, users immediately benefit from the possibilities given by this architecture.

5.4. AI2MMIR Evaluation

Independent from the results presented so far, there is one important aspect to be mentioned: this solution proves, that current AI and MMIR solutions can be successfully aligned and implemented according to the AI2MMIR reference architecture.

6. Summary and Outlook

This paper presented a reference architecture and proof-of-concept implementation for integrating Multimedia Information Retrieval, Retrieval-Augmented Generation (RAG), and locally operated Large Language Models (LLMs) in high-security environments, with a particular focus on legal practice. Grounded in the AI2MMIR reference architecture, the work demonstrates that advanced AI-assisted information access can be realized entirely within an organization's perimeter, addressing legal, regulatory, and organizational constraints such as data protection, professional secrecy, and auditability.

The contribution of this paper is primarily architectural. Rather than proposing new retrieval or generation algorithms, it illustrates how existing MMIR components, vector-based retrieval, and local LLM runtimes can be systematically aligned to support provenance-aware, citation-first, and explainable interaction patterns under real-world constraints. The presented implementation shows that such systems are technically feasible, operationally viable on commodity hardware, and usable in daily legal workflows.

At the same time, the scope of the evaluation reflects the realities of high-security environments and should be understood as a pilot assessment. While the results indicate practical effectiveness and positive user acceptance, several challenges remain open. In particular, large-scale benchmarking, systematic comparisons against alternative baselines, deeper analysis of multimodal retrieval performance,

and structured investigation of failure modes such as hallucination or cross-document synthesis are important directions for future work.

Overall, the paper positions local, RAG-enabled MMIR architectures as a promising foundation for responsible AI adoption in regulated domains. The reference implementation serves as a basis for further research and development toward more robust, scalable, and empirically validated systems, while maintaining the principles of human oversight, transparency, and legal accountability.

7. Future Work

While the presented reference implementation demonstrates the feasibility and practical value of local, RAG-enabled MMIR architectures in high-security environments, several extensions are planned to advance the approach beyond a proof-of-concept stage.

A first direction of future work concerns the tighter integration of the Generic Multimedia Analysis Framework (GMAF) as a native retrieval and indexing backend for Retrieval-Augmented Generation. Instead of treating the vector database as an external component, GMAF's Multimedia Feature Graphs and Graph Code representations will be explored as a unified RAG data layer. This would enable richer, explainable retrieval over structured multimedia features, support provenance-aware retrieval at graph level, and allow more advanced cross-document and cross-modal reasoning than flat vector similarity alone.

A second research direction focuses on broader and more systematic evaluation. Future studies will investigate the proposed architecture in additional application contexts beyond legal practice, such as compliance auditing, regulatory documentation, or security-critical enterprise knowledge management. These settings differ in document structure, task complexity, and risk profiles and therefore provide an opportunity to assess generalisability and robustness of the approach.

Furthermore, future evaluations will incorporate standardized retrieval and answer-quality benchmarks, comparisons against alternative local and cloud-based baselines, and more detailed performance analyses covering latency, throughput, and scalability. Particular attention will be paid to failure modes specific to RAG-based systems, including citation faithfulness, hallucination, cross-document synthesis, and multimodal retrieval errors.

Together, these extensions aim to evolve the presented architecture from a validated reference implementation toward a mature, empirically grounded framework for trustworthy multimedia-centric AI systems in regulated and high-security domains.

Author Contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Stefan Wagenpfeil and Angelina Chernikov. The first draft of the manuscript was written by Stefan Wagenpfeil and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Declaration on Generative AI

During the preparation of this work, the authors used Microsoft Copilot for AI-assisted copy-editing. Microsoft Copilot was used to provide suggestions regarding the clarity and structure of manually selected sections in order to improve the quality of writing. It was used solely for suggestions on structure, phrasing, and critique. No AI-generated text was copied or directly incorporated into the manuscript. All final wording and scientific content are the authors' own, and the authors take full responsibility for the content of the publication.

References

[1] E. Union, Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (general data protection regulation), Official Journal of the European Union L 119, 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679.

[2] E. Union, Artificial intelligence act (regulation (eu) 2024/1689), Official Journal of the European Union, June 2024, 2024. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689.

[3] U. Congress, Clarifying lawful overseas use of data act (cloud act), h.r. 1625, 115th congress, U.S. Congress, 115th Congress, 2018. URL: https://www.congress.gov/bill/115th-congress/house-bill/1625/text.

[4] A. Chernikov, Einsatz von retrieval-augmented-generation-systemen im wissensmanagement: Konzeption, implementierung und evaluation eines lokal betriebenen ki-gestützten wissensmanagementsystems, 2025. URL: https://www.pfh-goettingen.de/, bachelor Thesis, PFH Private Hochschule Göttingen.

[5] S. Wagenpfeil, Towards the trustworthy integration of multimedia and artificial intelligence, 2026. URL: https://www.fernuni-hagen.de/, habilitation Monograph, University of Hagen.

[6] J. F. Nunamaker, M. Chen, T. D. M. Purdin, Systems development in information systems research, Journal of Management Information Systems 7 (1991) 89–106. doi:10.1080/07421222.1991.10721388.

[7] S. Wagenpfeil, J. A. Kolenda, W. Nauvomets, T. Ternovik, J. Wenzig, C2pa and eidas 2.0 — a reference framework for content verification and protection in the eu, 2026. URL: https://www.fernuni-hagen.de/, pFH Private University of Applied Sciences / FernUniversität in Hagen.

[8] M.-D. Z. Hamburg, Lokale llm – chatgpt ohne cloud, n.d. URL: https://digitalzentrum-hamburg.de/leitfaden/lokale-llm-ohne-cloud, available at: https://digitalzentrum-hamburg.de/leitfaden/lokale-llm-ohne-cloud.

[9] A. Sharma, Integrating local llm frameworks: A deep dive into lm studio and anythingllm, 2024. URL: https://pyimagesearch.com/2024/06/24/integrating-local-llm-frameworks-a-deep-dive-into-lmstudio-and-anythingllm/, pyImageSearch.

[10] H. Touvron, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[11] T. B. Brown, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020).

[12] Z. Li, et al., Qwen: A chinese language model for general ai applications, arXiv preprint arXiv:2305.14375 (2023).

[13] T. B. Brown, et al., Gpt-neox: Scaling gpt-neo to 20b parameters, arXiv preprint arXiv:2104.07688 (2021).

[14] M. AI, et al., Mistral: Efficient and powerful llms, arXiv preprint arXiv:2304.12213 (2023).

[15] R. Glaser, et al., Retrieval-augmented generation: The architecture of reliable ai, 2024. URL: https://www.innoq.com/en/books/retrieval-augmented-generation, innoQ.

[16] M. Luo, T. Gokhale, N. Varshney, Y. Yang, C. Baral, Multimodal information retrieval, in: Advances in Multimodal Information Retrieval and Generation, Springer, 2024, pp. —. doi:10.1007/978-3-030-00000-0.

[17] S.-C. Lin, et al., Mm-embed: Universal multimodal retrieval with multimodal llms, 2025. URL: https://arxiv.org/abs/2411.02571, arXiv:2411.02571.

[18] S. Wagenpfeil, P. Mc Kevitt, M. Hemmje, Smart multimedia information retrieval, 2022. URL: https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001858, fernUniversität in Hagen Deposit.

[19] M. Hemmje, S. Wagenpfeil, Multimedia Information Retrieval: Informationen aus Multimedialen Quellen gewinnen, Springer, 2025.

[20] Best open-source vector databases, 2024. URL: https://datasystemreviews.com/best-open-source-vector-databases.html.

[21] Top 5 open source vector databases in 2025, 2025. URL: https://zilliz.com/blog/top-5-open-source-vector-search-engines.
[22] Vector databases — anythingllm documentation, 2025. URL: https://docs.anythingllm.com/features/vector-databases.
[23] Anythingllm's competitive edge: Lancedb for seamless rag, 2025. URL: https://lancedb.com/blog/anythingllms-competitive-edge-lancedb-for-seamless-rag-and-agent-workflows/.
[24] Choosing an open-source vector database, 2025. URL: https://geekbacon.com/2025/05/10/choosing-an-open-source-vector-database-a-comprehensive-guide/.
[25] 8 open-source vector databases compared, 2025. URL: https://estha.ai/blog/8-open-source-vector-databases-compared-choosing-the-right-solution-for-your-ai-applications/.
[26] Faiss vs pinecone vs weaviate vs milvus, 2024. URL: https://www.dataknobs.com/vector-database/tutorial/3-top-vector-databases-compared-faiss-vs-pinecone-vs-weaviate-vs-milvus.html.
[27] Vector database comparison 2025, 2025. URL: https://tensorblue.com/blog/vector-database-comparison-pinecone-weaviate-qdrant-milvus-2025.
[28] A. Documentation, Installation and docker overview, 2025. URL: https://docs.anythingllm.com/installation-docker/overview.
[29] Ollama, Documentation, n.d. URL: https://ollama.ai.
[30] LanceDB, Documentation, n.d. URL: https://lancedb.com.
[31] W. Zhang, J. Liu, X. Chen, Bilp: A rule-based layout parser for pdf documents, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2022) 1234–1248.
[32] R. Smith, An overview of the tesseract ocr engine, in: Proceedings of the Ninth International Conference on Document Analysis and Recognition, 2007, p. 629–633.
[33] M. Mohammadi, S. Khosravi, Pdfminer: Extracting text from pdf, Journal of Data Mining 12 (2019) 45–58.
[34] A. G, Pymupdf: Python bindings for mupdf, Python Package Index (2020).
[35] A. S. Foundation, Apache pdfbox: A java tool for working with pdf documents, in: Proceedings of the Open Source Software Conference, 2018.
[36] Regulation (eu) 2016/679 of the european parliament and of the council (general data protection regulation), Official Journal of the European Union, L 119, 2016. Arts. 5, 6, 9, 25, 32.