RESEARCH-ARTICLE

Metadata-aware RAG for Enforcing Access Control and Metadata-based Filtering: Proof of Concept and Evaluation

BINGXIANG CHEN, Tampere University, Tampere, Pirkanmaa, Finland

TONI TAIPALUS, Tampere University, Tampere, Pirkanmaa, Finland

Open Access Support provided by:

Tampere University

Published: 04 February 2026

Citation in BibTeX format

RACS '25: International Conference on Research in Adaptive and Convergent Systems
November 16 - 19, 2025
Ho Chi Minh, Vietnam

Conference Sponsors:
SIGAPP

https://doi.org/10.1145/3769002.3769952
ISBN: 9798400722318

Metadata-aware RAG for Enforcing Access Control and Metadata-based Filtering: Proof of Concept and Evaluation

Bingxiang Chen
Tampere University
Tampere, Finland
bingxiang.chen@tuni.fi

Toni Taipalus
Tampere University
Tampere, Finland
toni.taipalus@tuni.fi

Abstract

Organizations are integrating Large Language Models with internal knowledge bases to enhance knowledge retrieval and decision support. However, applying Retrieval-Augmented Generation systems in domains with sensitive data presents challenges such as weak or insufficient access control, inadequate metadata filtering, and lower precision or answer relevancy in the retrieved responses. For example, without proper access control mechanisms, sensitive patient information could be inadvertently exposed to unauthorized users. Furthermore, inadequate metadata filtering can lead to the retrieval of irrelevant or incomplete information, which reduces the accuracy and relevance of the response. This paper proposes a novel framework that integrates Metadata Filtering with fine-grained Role-Based Access Control to ensure both secure and accurate retrieval. By allowing metadata to serve as both a filtering mechanism and an enforcement tool, the system narrows search results while maintaining compliance and accountability. A proof-of-concept implementation using patient profile data demonstrates its effectiveness in secure AI-driven knowledge management.

CCS Concepts

• **Information systems** → **Query languages**; *Data management systems.*

Keywords

Large Language Models, Retrieval-Augmented Generation, access control, metadata

ACM Reference Format:
Bingxiang Chen and Toni Taipalus. 2025. Metadata-aware RAG for Enforcing Access Control and Metadata-based Filtering: Proof of Concept and Evaluation. In *International Conference on Research in Adaptive and Convergent Systems (RACS '25), November 16–19, 2025, Ho Chi Minh, Vietnam.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3769002.3769952

1 Introduction

Organizations are increasingly applying Large Language Models (LLMs) to internal data to support decision-making and improve operational efficiency [4]. Retrieval-Augmented Generation (RAG) systems, which retrieve relevant documents from a knowledge base before generating a response, have gained traction for their ability to provide more accurate and up-to-date answers without retraining the model [9, 31, 35]. However, as RAG systems are introduced in sensitive domains such as healthcare, concerns around security, data privacy, and retrieval quality become more pressing [19, 30]. For example, not all healthcare data should be accessible to every user, as some records may be restricted based on professional roles or regulatory requirements.

One critical limitation is that many RAG pipelines retrieve documents solely based on textual similarity, without considering access permissions or user context. This increases the risk of exposing confidential information to unauthorized users. Moreover, systems that rely on fine-tuning LLMs are often impractical for enterprise use: they are costly to update, hard to customize for rapidly changing data, and lack mechanisms for enforcing user-specific access rules [28]. While RAG can address some of these issues by separating retrieval from model training, challenges in retrieval accuracy and access control remain [10, 31].

This work explores how metadata can help address both retrieval relevance and data protection. Metadata associated with documents, such as document type, access role, or creation time, can be used to filter search results before they are passed to the LLM. This enables role-based access control during retrieval, ensuring users only see information they are authorized to access. It also helps return results that are more contextually appropriate.

We present a framework that integrates metadata filtering with fine-grained access control in a RAG pipeline. The proposed approach constructs metadata-aware filters based on query intent and user roles, retrieves information accordingly, and uses the filtered results to generate responses. A proof-of-concept system in the healthcare domain illustrates how this method enforces strict access policies while maintaining response quality. This study expands on our earlier poster work at ACM SAC 2025 [8], providing a more complete system design and evaluation.

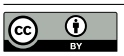
The rest of the paper is structured as follows: Section 2 reviews relevant background. Section 3 presents our system design and architecture. Section 4 describes the proof-of-concept implementation in the healthcare domain, and reports evaluation results. Section 5 discusses limitations and future work. Section 6 concludes the study.

2 Background

2.1 Generative AI

Generative AI [22] creates content such as text, images, and video based on user prompts. Unlike models that analyze existing data, generative AI learns patterns from large datasets to produce original outputs. These models can be classified into unimodal (single type of output) and multimodal (multiple types of output).

The field has evolved from proprietary, large-scale systems to more accessible, open-source tools. Advancements in models like



RACS '25, Ho Chi Minh, Vietnam

ACM ISBN 979-8-4007-2231-8/2025/11
https://doi.org/10.1145/3769002.3769952

StableLM [6], Mistral [18], and Llama2 [23] show the potential to perform comparably to leading proprietary models [26, 37].

Key components of the generative AI technology stack encompass various essential elements that work together to enable the development and deployment of AI applications. Machine learning frameworks like TensorFlow, PyTorch, and Keras provide the necessary tools and application programming interfaces for building and training generative models, while data preprocessing tools such as Apache Spark and Hadoop manage and prepare large datasets for analysis. Python is one of the primary programming languages for AI model development due to its extensive libraries and strong community support. Additionally, deployment tools and cloud platforms from providers like Amazon, Google, and Microsoft offer the computational power and scalability required to run generative AI systems efficiently. Together, these components create an environment for developing generative AI applications.

A subset of generative AI is Conversational AI, which leverages LLMs to create chatbots and virtual assistants [11]. Three approaches can be applied to integrate conversational AI with proprietary or domain-specific data: system prompts, fine-tuning, and RAGs. First, system prompts involves embedding task-specific instructions directly into the model prompts. This method guides the AI's responses to generate more accurate and contextually relevant outputs tailored to specific tasks. Second, fine-tuning refines the LLM on domain-specific data, optimizing the model for specialized tasks by adjusting the model's weights based on new data. This enhances the model's ability to understand the context and terminology pertinent to specific domains such as healthcare or technical support. Third, utilizing a RAG approach improves LLM responses by retrieving and referencing external knowledge bases. This allows access to real-time or proprietary information that the model was not initially trained on, thereby enhancing response accuracy and relevance by combining generative capabilities with real-time information retrieval.

2.2 Retrieval-Augmented Generation

RAG enhances the performance of LLMs by integrating external knowledge bases, which enables more accurate and contextually relevant responses without the need for model retraining [17]. The RAG process begins with data collection and chunking, where data are gathered and segmented into smaller, searchable chunks. These chunks are then transformed into vectors using pre-trained models, such as those based on embeddings, for efficient indexing and search. Subsequently, the vectors are stored into a vector database using the same embedding space, meaning that similar vectors logically reside close to each other. When a natural language query is submitted, the query is transformed into a search vector, which is consequently used to retrieve approximate nearest neighbor vectors. These nearest vectors (or documents associated with them) are used to provide additional context for the LLM [35].

Recent research has explored the application of RAG combined with LLMs for generating test scenarios from natural language requirements, demonstrating its effectiveness in addressing practical problems in software development [3]. This highlights the versatility of RAG beyond healthcare and oncology, reinforcing its potential to enhance LLM capabilities across various domains.

Frameworks such as LlamaIndex [21], LangChain [20], and AutoGen [24] play key roles in facilitating LLM applications. LlamaIndex enhances LLM capabilities by focusing on the ingestion, organization, and retrieval of private data, offering an interface for indexing large text datasets and integrating them into LLM workflows [33]. LangChain simplifies the development of interactive LLM applications with features that enhance data awareness and usability [20]. AutoGen automates the LLM process using multiple agents to manage complex workflows and coordinate multi-agent conversations [24].

2.3 Enhancing Retrieval Accuracy in RAG

Recent research identifies several complementary strategies for improving RAG accuracy. For example, retrieval should optimize for both relevance and coverage. Standard top-k nearest-neighbor retrieval can return redundant content, limiting recall. Coverage-aware selection methods can reduce redundancy while preserving relevance, increasing the chance of retrieving all necessary facts [27]. On the other hand, adapting the retriever to the target domain can improve both precision and recall [34]. Finally, in critical domains where factual errors can cause serious consequences, re-ranking based on both factuality and topicality has been shown to boost precision. Combining lexical or embedding-based similarity with factual accuracy scores prioritizes documents that are both relevant and factually supportive [36]. Implementing these strategies typically demands additional computational resources and careful system design.

2.4 Metadata and Its Role in AI

Metadata is data about data, providing essential attributes that help categorize, sort, and filter information. It enhances data organization, retrieval, and management by describing characteristics such as format, source, creation date, and classification criteria. Unlike primary data, which consist of factual records or observations, metadata serves as contextual information that aids in interpretation and usability [5]. For example, in document-based systems, metadata such as file names, authors, and timestamps enable more efficient data retrieval [2].

In RAG systems, metadata plays a crucial role in both preprocessing and retrieval. During preprocessing, metadata help organize and structure data, such as medical records, research papers, and clinical notes by filtering and categorizing the data before storage. This organization enhances retrieval efficiency, enabling the system to locate relevant information quickly. During retrieval, metadata refines search results, ensuring more accurate and context-specific outcomes [2]. For example, in healthcare systems, metadata can filter medical guidelines based on a patient's condition, ensuring that only relevant treatment protocols are retrieved [38].

Moreover, metadata is integral to supporting time-aware RAG. By associating metadata such as document timestamps or creation dates, systems can prioritize the most up-to-date data, ensuring that the information retrieved is both current and relevant [14]. Additionally, metadata can enhance security through fine-grained access control, ensuring that only authorized personnel can access sensitive data such as patient records in a healthcare setting. Furthermore, incorporating metadata into the LLM prompt can provide

contextual information, like the publication date of a medical study, helping the model generate more precise and relevant responses. By leveraging metadata, RAG systems can potentially improve the accuracy, security, and contextuality of information retrieval.

2.5 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) is a widely used model that assigns access permissions based on user roles, helping enforce least-privilege principles in enterprise systems [32]. In domains such as healthcare, RBAC is critical for ensuring that only authorized users can access sensitive data.

In addition to traditional role-based models, prior work has explored higher-level policy specification languages and visual frameworks to simplify access management and reduce configuration complexity. One example proposes a visual, computer-managed framework that enables flexible and intuitive access control in enterprise applications [15].

This paper adopts the simple RBAC model in the context of generative AI, applying it to domain-specific applications where sensitive data access must be tightly controlled. The approach integrates role-based constraints with metadata filtering in a RAG system to enable secure and permission-aware information retrieval.

2.6 Concerns in Applying LLMs to Domain-Specific and Enterprise Data

Using LLMs in domain-specific or enterprise data contexts such as healthcare, presents potential challenges. One issue is the dynamic nature of enterprise data, which is continuously updated. This presents difficulties for LLMs to remain accurate and up-to-date [12]. For example, patient records can change frequently, and any delay in updating these data sources could lead to outdated or incorrect results from an LLM, undermining its trustworthiness.

Furthermore, access control and permissions are crucial to ensuring that sensitive enterprise data, such as medical records can only be accessed by authorized individuals. Without mechanisms for fine-grained role-based access control (RBAC), there is a risk of unauthorized access, which can compromise the security of sensitive information [30]. This is particularly important in healthcare, where unauthorized access to patient data can have serious legal and ethical consequences [16]. Therefore, systems utilizing LLMs in enterprise data environments must incorporate precise access control and high-precision retrieval mechanisms to ensure data privacy.

For LLMs to be deployed effectively in these contexts, they need to combine dynamic adaptability, high precision in information retrieval, and trustworthy access control systems to maintain both the accuracy and security of sensitive data. Ensuring these capabilities is fundamental to leveraging AI safely and ethically in enterprise and domain-specific applications. The next sections, we present a design and proof-of-concept implementation of a system intended to address these challenges.

3 System Design

Our system integrates two core concepts: LLM metadata filtering and Role-Based Access Control (RBAC), which together ensure secure and efficient retrieval in a Retrieval-Augmented Generation (RAG) pipeline. The system is designed to filter data both semantically and according to user permissions, ensuring access to only relevant and authorized information. The system architecture is shown in Fig. 1 and consists of two main phases: ingestion and query processing.

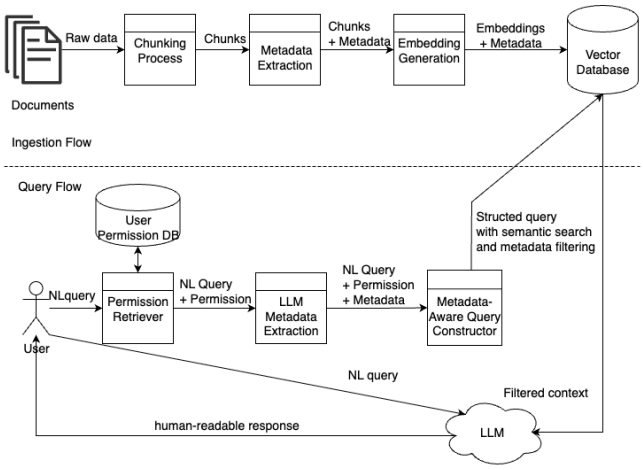


Figure 1: System Architecture

3.1 Ingestion Process

In the ingestion phase (Fig. 1, upper part), raw patient records are transformed into metadata-enriched vector embeddings. The key steps include:

(1) *Chunking*: Each input document contains multiple question–answer (QA) pairs. These are split into individual QA pairs, so that each pair forms a separate chunk for retrieval.
(2) *Metadata Extraction*: Structured metadata such as question ID (`qid`) and question type (`qtype`) is extracted from the source format (e.g., XML attributes). The `qtype` categorizes the content into medically meaningful classes such as symptoms, treatment, or billing.
(3) *Embedding Generation*: Each chunk is encoded into a vector representation that captures its semantic meaning.
(4) *Storage in Vector Database*: Embeddings are stored alongside their metadata in a vector database. This enables future retrieval based on both semantic similarity and access-aware metadata filtering.

3.2 Query Processing

The query processing phase (Fig. 1, lower part) ensures that responses are both semantically relevant and access-controlled. The system processes a query through the following steps:

(1) *Permission Retriever*: The user's role and associated access rights are retrieved from a permission database, which maps user roles to allowed `qtype` categories.
(2) *Metadata Extraction*: An LLM analyzes the user query to extract the intended information types, such as `symptoms`, `treatment`, or `exams and tests`.

(3) *Metadata-Aware Query Constructor*: Based on the user's permissions and the extracted `qtypes`, a metadata filter is constructed. If any `qtypes` in the query fall outside the user's access rights, they are excluded from the query. In such cases, a fallback filter is built using only permitted types, and the system can optionally append a warning message to inform the user about restricted access.
(4) *Vector Database Search*: The structured query, which includes both the search vector and metadata filters, is executed against the vector database. Only chunks that satisfy both semantic similarity and metadata constraints are retrieved.
(5) *LLM Response Generation*: The retrieved chunks are passed to the LLM along with the original query to generate a natural language response that respects both context and access policies.

3.3 Role-Based Access Control Overview

The access control model is based on Role-Based Access Control (RBAC). Each user is assigned a role (e.g., patient, doctor, researcher, admin), and each role has access to a predefined set of `qtypes`. These permissions determine what types of information the user is allowed to retrieve. The admin role is granted full access across all categories, while others are restricted to only those `qtypes` necessary for their responsibilities. This ensures compliance with principles such as least privilege and regulatory requirements like the HIPAA Minimum Necessary Rule [13].

4 Proof-of-Concept Implementation

We built a proof-of-concept system to validate the proposed approach in a controlled environment. The implementation uses a medical QA dataset along with simulated user roles to test access control behavior. We use `gpt-3.5-turbo` [25] for query interpretation because it balances strong language understanding with low cost, making it suitable for interactive use. For retrieval, we use `all-MiniLM-L6-v2` [1], which offers competitive semantic search accuracy with low latency and a small embedding size (384-d) that keeps the index fast and compact. The embeddings are stored in `Qdrant` [29], chosen for its efficient approximate nearest neighbor search and built-in metadata filtering, which allows access control policies to be applied directly during retrieval. This setup shows that combining metadata filtering with role-based access control can enforce security while preserving the relevance and quality of responses.

4.1 Dataset and Preprocessing

We used a filtered subset of the MedQuAD dataset [7], originally containing 47,457 question-answer (QA) pairs across 37 medical categories from 12 NIH-affiliated sources. For our experiments, we excluded three subsets with missing or incomplete answers: 10_MPlus_ADAM_QA, 11_MPlusDrugs_QA, and 12_MPlusHerbsSupplements_QA. The resulting dataset contains 16,412 QA pairs across 16 medically meaningful question types (qtypes), as shown in Table 1.

Each QA pair is treated as a chunk, with metadata extracted from XML attributes (`qid`, `qtype`). We use QA-pair granularity because each pair in MedQuAD is self-contained, with minimal dependency on other pairs. This reduces noise during retrieval and aligns naturally with the metadata categories (qtype). In preliminary trials, finer-grained sentence-level chunking fragmented context and lowered answer correctness, while coarser multi-pair chunking increased the retrieval of irrelevant information. The chunks are embedded into 384-dimensional vectors using the `all-MiniLM-L6-v2` model described earlier and stored in Qdrant along with metadata payloads, enabling metadata-constrained retrieval during querying.

Question Type	QA Pairs
Information	4540
Symptoms	2748
Treatment	2442
Inheritance	1446
Frequency	1120
Genetic changes	1087
Causes	727
Exams and tests	653
Research	395
Outlook	361
Susceptibility	324
Considerations	235
Prevention	210
Stages	77
Complications	46
Support groups	1
Total	**16,412**

Table 1: Distribution of selected question types in the filtered MedQuAD dataset

4.2 RBAC Policy Implementation

We implement role based access control (RBAC) using an `SQLite` database. Each user is assigned a role such as patient, doctor, researcher, or admin. For each role, we define a specific set of accessible question types (`qtypes`) based on realistic information sharing practices in healthcare. Access permissions are structured as follows:

(1) Patients can access general and non-sensitive categories, including information, prevention, support groups, and considerations.
(2) Doctors can access all categories available to patients, along with additional clinical types such as symptoms, treatment, and exams.
(3) Researchers can access content related to scientific inquiry, including research, genetic changes, causes, and outlook.
(4) Admins have unrestricted access to all categories.

4.3 Query Execution Workflow

When a user submits a question, the system performs the following:

(1) Extracts the relevant `qtype` from the query using GPT-3.5.
(2) Retrieves the user's role and associated access policy from the SQLite RBAC database.

(3) Filters the vector database using both semantic similarity and the set of permitted `qtypes`.
(4) Retrieves the top ranked chunks that satisfy both the similarity and access constraints.
(5) Combines the filtered context with the original query to form the input prompt for response generation.
(6) If no accessible content remains after filtering, a warning message is returned instead.

Example: A patient asks, *"What is the ICU protocol for treating acute respiratory distress syndrome?"* The system classifies the query type as `treatment`, which is restricted for the patient role when it contains advanced clinical procedures and medication dosages that could be unsafe without medical supervision. Access is denied, and the following message is returned: *"This treatment information is intended for healthcare professionals. Please consult a qualified clinician."*

4.4 System Demonstration

To validate behavior across user roles, we tested several scenarios with simulated users. Table 2 summarizes selected queries and how the system enforced access control.

Table 2: Representative queries and system behavior by user role

Query	Role	System Response
What are the symptoms of Adult Acute Lymphoblastic Leukemia?	Patient	Answer returned (qtype: `symptoms` — allowed)
What are the treatments for Age-related Macular Degeneration?	Patient	Access denied (qtype: `treatment` — not allowed)
what research (or clinical trials) is being done for Age-related Macular Degeneration?	Patient	Access denied (qtype: `research` — not allowed)
What are the treatments for Age-related Macular Degeneration?	Doctor	Answer returned (qtype: `treatment` — allowed)
what research (or clinical trials) is being done for Age-related Macular Degeneration?	Researcher	Answer returned (qtype: `outlook` — allowed)
What are the treatments for Age-related Macular Degeneration?	Researcher	Access denied (qtype: `treatment` — not allowed)

These cases demonstrate the prototype's ability to dynamically enforce role-specific access policies while maintaining the quality and relevance of allowed content. Figure 2 shows the system interface enforcing access control for different user roles.

4.5 Evaluation Setup

We evaluate the system using the filtered MedQuAD dataset introduced in Section 4, which includes 16,412 QA pairs across 16 question types (`qtype`). Each QA pair is embedded and stored in a vector database (Qdrant) along with its metadata.

Access control policies are defined for four user roles: `patient`, `doctor`, `researcher`, and `admin`. To assess the impact of metadata filtering and role-based enforcement, we compare two system variants:

- *Baseline RAG*: Retrieves relevant chunks based on semantic similarity only, without considering access control.
- *Metadata-filtered RAG (ours)*: Applies both semantic similarity and metadata constraints derived from the user's role and the extracted `qtype`.

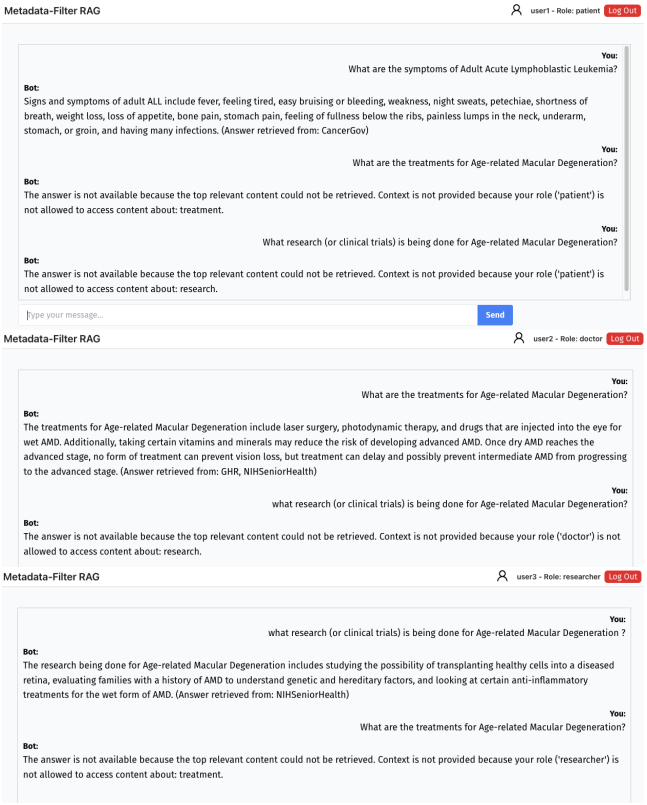


Figure 2: System demonstration of access control in different roles.

4.6 Access Control Enforcement

To evaluate access control, we submitted a set of queries from different user roles targeting various question types. For each query, we recorded whether the returned content adhered to the role's access permissions.

Table 3: Access Control Enforcement: Example-Based Comparison

Query Scenario	Baseline RAG (No Access Control)	Our System (RBAC + Metadata)
Receptionist asks: "What treatment has Liam Gonzalez received?"	Treatment info retrieved — unauthorized.	Access denied. Only basic info returned.
Patient asks: "How is Kaposi Sarcoma diagnosed?"	Full diagnostic info returned.	Access denied. Exams and tests not available to patients.
Researcher asks: "What symptoms are common in early-stage AML?"	Symptom details retrieved.	Access denied. Role does not permit access to symptom data.
Doctor asks: "What are the billing details for Layla Gomez?"	Billing info exposed.	Access denied. Financial data not accessible to doctors.

These examples demonstrate that without access control, RAG systems pose serious privacy risks. Our system mitigates this by ensuring that only authorized content is retrieved, enforcing compliance with role-based policies.

4.7 Retrieval and Answer Quality

We evaluated the impact of metadata filtering on the retrieval and answer quality using 500 randomly selected question–answer pairs

from the MedQuAD dataset, with both systems using the same embedding and language models to isolate the effect of metadata constraints. The same set of pairs was used for both the baseline and our system evaluations, with the only difference being the application of metadata-based filtering during retrieval.

Metrics. We evaluate the generated responses using metrics from the RAGAS framework[1], focusing on Context Precision (ConRel) and Answer Correctness (AnsCor). Context Precision measures the proportion of relevant statements within the retrieved context, indicating how focused and useful the retrieval is. Answer Correctness assesses the factual and semantic alignment between the generated answer and the ground truth, using a weighted scoring scheme to reflect overall response accuracy.

Results. As shown in Table 4, the metadata-filtered system outperforms the baseline in both metrics. It consistently retrieves more relevant content and produces more accurate answers. For example, when asked about symptoms, our system retrieves only context chunks labeled as `symptoms`, leading to more precise responses. In contrast, the baseline may include tangentially related but less useful content.

Both systems achieved moderate correctness scores, partly due to the detailed and lengthy nature of MedQuAD's reference answers. Generated outputs tend to be shorter and more concise, which can lower measured correctness even when key information is included.

Table 4: Evaluation results on 500 QA pairs from MedQuAD

System	Context Precision	Answer Correctness (AnsCor)
Baseline (no metadata filtering)	0.8918	0.4895
Our system (with metadata filtering)	0.9532	0.5621

5 Limitations and Future Directions

As a proof-of-concept, this work focuses on illustrating the feasibility of integrating access control into Retrieval-Augmented Generation. While effective in demonstrating core ideas, certain simplifications were made that may not fully reflect the complexity of real-world applications.

Our prototype simplifies several challenges to emphasize secure and accurate retrieval. Metadata extraction currently relies on GPT-3.5, and access control policies are illustrative rather than comprehensive. Deploying this system in real healthcare institutions may introduce additional challenges such as heterogeneous data formats, stricter regulations, and integration with existing systems. Future work could explore more efficient metadata extraction, benchmark optimized RAG components (e.g., rerankers), and evaluate responsiveness and user experience alongside retrieval accuracy on larger, real-world datasets.

6 Conclusion

This paper highlighted the importance of integrating access control into Large Language Models (LLMs) for managing sensitive organizational data. We presented a Retrieval-Augmented Generation (RAG) framework that combined fine-grained Role-Based Access Control (RBAC) with metadata filtering to ensure that retrieved content was both semantically relevant and access-compliant. By incorporating user intent and metadata constraints directly into the retrieval stage, our system improved both security and the contextual quality of LLM-generated responses.

We validated the proposed framework with a proof-of-concept implementation in the healthcare domain, using the MedQuAD dataset. Our evaluation demonstrated that metadata filtering enhanced retrieval precision and answer correctness while enforcing strict access control policies.

All code, evaluation data, and implementation scripts are publicly available on Figshare[2] and GitHub[3], supporting reproducibility and future research.

Acknowledgments

This work has been supported by FAST, the Finnish Software Engineering Doctoral Research Network, funded by the Ministry of Education and Culture, Finland.

References

[1] 2024. all-MiniLM-L6-v2 model card. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.
[2] Deepset AI. 2023. Leveraging Metadata in RAG Customization. (2023). https://www.deepset.ai/blog/leveraging-metadata-in-rag-customization Accessed: March 11, 2025.
[3] Chetan Arora, Tomas Herda, and Verena Homm. 2024. Generating Test Scenarios from NL Requirements using Retrieval-Augmented LLMs: An Industrial Study. arXiv:2404.12772 [cs.SE] https://arxiv.org/abs/2404.12772
[4] Lateef Ayinde, Muhamad Prabu Wibowo, Benhur Ravuri, and Forhan Bin Emdad. 2023. ChatGPT as an important tool in organizational management: A review of the literature. *Business Information Review* 40, 3 (2023), 137–149.
[5] Murtha Baca. 2016. *Introduction to metadata.* Getty Publications.
[6] Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, Meng Lee, Emad Mostaque, Michael Pieler, Nikhil Pinnaparju, Paulo Rocha, Harry Saini, Hannah Teufel, Niccolo Zanichelli, and Carlos Riquelme. 2024. Stable LM 2 1.6B Technical Report. arXiv:2402.17834 [cs.CL]
[7] Asma Ben Abacha and Dina Demner-Fushman. 2019. A Question-Entailment Approach to Question Answering. *BMC Bioinform.* 20, 1 (2019), 511:1–511:23. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3119-4
[8] Bingxiang Chen, John Tackman, Manu Setälä, Timo Poranen, and Zheying Zhang. 2025. Integrating Access Control with Retrieval-Augmented Generation: A Proof of Concept for Managing Sensitive Patient Profiles. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing* (Catania International Airport, Catania, Italy) *(SAC '25).* Association for Computing Machinery, New York, NY, USA, 915–919. https://doi.org/10.1145/3672608.3707848
[9] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17754–17762.
[10] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The Power of Noise: Redefining Retrieval for RAG Systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24).* Association for Computing Machinery, New York, NY, USA, 719–729. https://doi.org/10.1145/3626772.3657834
[11] Yang Deng, Lizi Liao, Wenqiang Lei, Grace Hui Yang, Wai Lam, and Tat-Seng Chua. 2025. Proactive conversational ai: A comprehensive survey of advancements and opportunities. *ACM Transactions on Information Systems* 43, 3 (2025), 1–45.
[12] José Cassio dos Santos Junior, Rachel Hu, Richard Song, and Yunfei Bai. 2024. Domain-Driven LLM Development: Insights into RAG and Fine-Tuning Practices. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24).* Association for Computing Machinery, New York, NY, USA, 6416–6417. https://doi.org/10.1145/3637528.3671445
[13] Barbara J Evans and Gail P Jarvik. 2018. Impact of HIPAA's minimum necessary standard on genomic data sharing. *Genetics in Medicine* 20, 5 (2018), 531–535.

[1] https://docs.ragas.io/en/v0.1.21/concepts/metrics/index.html

[2] https://doi.org/10.6084/m9.figshare.28830008.v3

[3] https://github.com/bingxiangch/MetaFilterRAG.git

[14] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* 2 (2023).

[15] Massimiliano Giordano and Giuseppe Polese. 2013. Visual Computer-Managed Security: A Framework for Developing Access Control in Enterprise Applications. *IEEE Software* 30, 5 (2013), 62–69. https://doi.org/10.1109/MS.2012.112

[16] Lawrence O Gostin, Laura A Levit, and Sharyl J Nass. 2009. Beyond the HIPAA privacy rule: enhancing privacy, improving health through research. (2009).

[17] Cheonsu Jeong. 2023. A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture. *Advances in Artificial Intelligence and Machine Learning* 03, 04 (2023), 1588–1618. https://doi.org/10.54364/aaiml.2023.1191

[18] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL]

[19] Jitendra Jonnagaddala and Zoie Shui-Yee Wong. 2025. Privacy preserving strategies for electronic health records in the era of large language models. *npj Digital Medicine* 8, 1 (2025), 34.

[20] LangChain Team. 2024. LangChain. https://www.langchain.com/ Accessed on 26 April 2024.

[21] LlamaIndex Team. 2024. Llamaindex. https://www.llamaindex.ai/ Accessed on 26 April 2024.

[22] Zhihan Lv. 2023. Generative Artificial Intelligence in the Metaverse Era. *Cognitive Robotics* 3 (2023), 208–217. https://doi.org/10.1016/j.cogr.2023.06.001

[23] Inc. Meta Platforms. [n. d.]. Llama-2. https://llama.meta.com/llama2/ Accessed on 18 April 2024.

[24] Microsoft Team. 2024. AutoGen. https://microsoft.github.io/autogen/ Accessed on 24 July 2024.

[25] OpenAI. 2025. GPT-3.5-Turbo model documentation. https://platform.openai.com/docs/models/gpt-3.5-turbo.

[26] C. Peng, X. Yang, A. Chen, K. E. Smith, N. PourNejatian, A. B. Costa, C. Martin, M. G. Flores, Y. Zhang, T. Magoc, G. Lipori, D. A. Mitchell, N. S. Ospina, M. M. Ahmed, W. R. Hogan, E. A. Shenkman, Y. Guo, J. Bian, and Y. Wu. 2023. A Study of Generative Large Language Models for Medical Research and Healthcare. *NPJ Digital Medicine* 6, 1 (11 2023), 210. https://doi.org/10.1038/s41746-023-00958-w

[27] Marc Pickett, Jeremy Hartman, Ayan Kumar Bhowmick, Raquib-ul Alam, and Aditya Vempaty. 2024. Better rag using relevant information gain. *arXiv preprint arXiv:2407.12101* (2024).

[28] C.S. Powers, P. Ashley, and M. Schunter. 2002. Privacy promises, access control, and privacy management. Enforcing privacy throughout an enterprise by extending access control. In *Proceedings. Third International Symposium on Electronic Commerce,*. 13–21. https://doi.org/10.1109/ISEC.2002.1166906

[29] Qdrant Contributors. 2024. Qdrant: Open-Source Vector Database and Vector Search Engine. https://qdrant.tech/documentation/.

[30] Lillian Røstad and Øystein Nytrø. 2008. Towards dynamic access control for healthcare information systems. *Studies in health technology and informatics* 136 (02 2008), 703–8. https://doi.org/10.3233/978-1-58603-864-9-703

[31] Alireza Salemi and Hamed Zamani. 2024. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2395–2400.

[32] Ravi Sandhu, Edward Coyne, Hal Feinstein, and Charles Youman. 1996. Role-Based Access Control Models. *IEEE Computer* 29, 2 (1996), 38–47.

[33] Devin Schumacher. 2023. V3CTRON| Data Retrieval & Access System for Flexible Semantic Search & Retrieval of Proprietary Document Collections Using Natural Language Queries. *Available at SSRN* (2023).

[34] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics* 11 (2023), 1–17.

[35] Toni Taipalus. 2024. Vector Database Management Systems: Fundamental Concepts, Use-Cases, and Current Challenges. *Cognitive Systems Research* 85 (June 2024), 101216. https://doi.org/10.1016/j.cogsys.2024.101216

[36] Rishabh Upadhyay and Marco Viviani. 2025. Enhancing Health Information Retrieval with RAG by prioritizing topical relevance and factual accuracy. *Discover Computing* 28, 1 (2025), 27.

[37] Gissel Velarde. 2021. Artificial Intelligence Trends and Future Scenarios: Relations Between Statistics and Opinions. In *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*. 64–70. https://doi.org/10.1109/CogMI52975.2021.00017

[38] Boyi Xu, Ke Xu, Liuliu Fu, Ling Li, Weiwei Xin, and Hongming Cai. 2016. Healthcare data analytics: using a metadata annotation approach for integrating electronic hospital records. *Journal of Management Analytics* 3 (02 2016), 1–16. https://doi.org/10.1080/23270012.2016.1141331