International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Authorization-Aware Multi-Agent Retrieval-Augmented Generation for Secure Enterprise Knowledge Access

Guide: Prof. Vaishnavi Patharwat

Chinmay Kundan Ahir[1]

Aditya Ganesh Kumbhoje[2]

Prathamesh Mohanlal Mahajan[3]

Pranav Bhivaji Gawade[4]

Mrunal Atul Jagtap[5]

Department of Artificial Intelligence and Machine Learning,
Zeal Polytechnic, Pune, Maharashtra, India

ABSTRACT

Retrieval-augmented generation (RAG) enables language models to answer enterprise questions using external documents; however semantic retrieval and multi-agent delegation create authorization risks when access checks occur only after search or answer generation. This study presents a literature-informed architecture for authorization-aware multi-agent RAG in which a policy decision point evaluates current user and resource attributes, whereas a policy enforcement point restricts the eligible candidate set before content reaches rerankers, agents, or a large language model. This design combines role-based and attribute-based access control, semantic embeddings, vector search, least-privilege delegation, provenance-aware auditing, and policy revalidation at protected access boundaries. A focused survey identifies risks including semantic overfetch, cross-domain synthesis leakage, stale permissions, and delegation escalation. The paper defines authorization-correctness conditions and proposes an evaluation protocol that compares authorization-first retrieval with a retrieve-then-filter baseline. The metrics include unauthorized-context exposure, authorized Recall@k, answer relevance, latency, revocation propagation time, delegation violations, and audit completeness. This study is a survey and design contribution; no benchmark results are claimed because the proposed prototype has not yet been empirically evaluated.

Index terms—*Access control, enterprise knowledge, multi-agent systems, retrieval-augmented generation, semantic search.*

I. INTRODUCTION

Enterprise organizations maintain policies, technical documentation, contracts, project records, and internal reports across multiple repositories. Retrieval-augmented generation (RAG) combines document retrieval with language model generation so that answers can be grounded in external information rather than relying only on model parameters [1]. Embedding-based retrieval supports semantic matching between a natural-language query and document chunks [2]. These capabilities are useful for enterprise knowledge assistants, but relevant content must be authorized before it is used by downstream AI components.

Traditional access-control systems often enforce permissions at the application or database boundary. In RAG, document chunks can pass through vector search, reranking, summarization, agent tools, and language model context windows. If a system retrieves restricted chunks and filters them only from the final answer, sensitive information may already have reached intermediate components. A recent systematization of knowledge on authorization in multi-agent RAG discusses failure modes including semantic overfetch, cross-domain synthesis leakage, and delegation escalation [3]. These risks motivate treating authorization as a precondition for defining the semantic retrieval candidate set.

This study surveys relevant foundations and proposes an authorization-aware architecture for secure enterprise knowledge access. It does not claim that authorization-aware retrieval is new. Its contribution is a practical integration design and evaluation protocol covering candidate-set restriction, delegated-agent scope, permission changes, provenance, and auditing.

II. BACKGROUND AND RELATED WORK

A. Retrieval-Augmented Generation

Lewis et al. describe RAG as a combination of a parametric generator and non-parametric memory accessed through retrieval [1]. In an enterprise implementation, documents are extracted and chunked, an embedding model maps each chunk to a vector, and a vector index returns semantically similar candidates. Sentence-BERT enables efficient comparison of sentence embeddings using cosine similarity [2]. These methods do not independently determine whether a requesting user may access a chunk.

B. Access-Control Models

Role-based access control (RBAC) associates permissions with organizational roles. The NIST RBAC model formalizes role hierarchies and constraints [4]. Attribute-based access control (ABAC) evaluates the attributes of the subject, resource, requested operation, and potentially the environment against the policy [5]. RBAC supports job-based permissions, whereas ABAC can express contextual rules involving department, classification, project membership, or time. A policy decision point (PDP) evaluates a request, and a policy enforcement point (PEP) enforces the decision. RBAC and ABAC describe policy models, whereas PDP/PEP describe decision and enforcement responsibilities.

C. Authorization in Agentic RAG

Namboothiri's systematization of authorization in multi-agent RAG organizes risks and mitigation approaches

around Authorization-First Retrieval (AFR): authorization constrains the semantic candidate space before retrieved content reaches downstream model contexts [3]. Vector databases have the capability to implement metadata or payload filters during search processes. For instance, Qdrant provides documentation on conditions that limit matching points based on payload attributes [6]. Such filters are useful implementation mechanisms, but metadata should not be the sole authority for current permissions when policies can change.

D. Gap and Scope

Existing work covers access-control models, filtered vector search, and authorization-first formulations. The practical gap addressed here is their integration across the full multi-agent path: policy evaluation, candidate restriction, reranking, delegated-agent scope, permission changes, provenance, and audit records. This is a proposed design and evaluation protocol, not a claim of a completed production system.

III. THREAT MODEL AND DESIGN PRINCIPLES

The system assumes authenticated users and enterprise resources with stable identifiers and descriptive metadata. Protected assets include document chunks, derived summaries, intermediate agent outputs, tool results, and the final responses. Users may submit adversarial or ambiguous queries, attempt cross-document inference, or exploit agents with broader permissions than those of the requesting user. The design does not assume that a language model can reliably enforce an access policy by itself. OWASP also identifies sensitive information disclosure and excessive agency as risks in LLM applications [7].

Four principles guide the design: deny by default; authorize before exposure to downstream learned components; inherit and attenuate permissions during delegation; and preserve source provenance. These principles reflect least privilege and authorization-correctness concerns discussed in [3]. Let U denote users, D document chunks, and t the time of a request. The permitted set for user u is:

$$A(u,t) = \{ d \in D \mid PDP(u,d,t) = Permit \} \quad (1)$$

For query q, let E(q) and E(d) be the query and document embeddings. Cosine similarity is:

$$sim(q,d) = \frac{E(q)\cdot E(d)}{\|E(q)\|\|E(d)\|} \quad (2)$$

The system ranks only the chunks in the permitted set:

$$R_k(q,u,t) = TopK \text{ over } d \in A(u,t) \text{ of } sim(q,d) \quad (3)$$

For agent a_i assigned task τ_i, the effective permissions are limited to the intersection of the user permissions and task scope:

$$perm(a_i) = perm(u) \cap scope(\tau_i) \quad (4)$$

This constraint prevents a sub-agent from acquiring permissions that the initiating user lacks and narrows access to the assigned task.

IV. PROPOSED ARCHITECTURE AND WORKFLOW

The architecture separates offline document preparation from online query processing. During ingestion, the source files are parsed, normalized, chunked, and associated with resource identifiers and access metadata. A sentence-embedding model generates vectors for the chunks, which are stored with metadata in a vector database. This prepares searchable content but does not grant access to the user.

At query time, the system authenticates the user and obtains the current identity and attribute context. The PDP evaluates the applicable RBAC/ABAC rules, and the PEP ensures that only permitted resources are eligible for semantic search. A vector-store filter may implement this restriction efficiently, while a trusted policy service remains responsible for the authoritative decision. A reranker, if used, receives only authorized candidates. The language model then generates a response with source references.

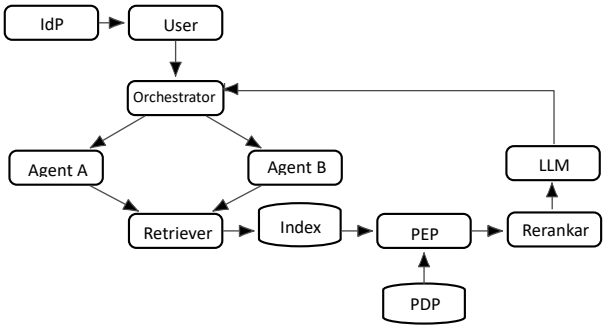


Fig. 1. Architecture of the authorization-aware multi-agent RAG system [3].

Figure 1 represents the overall architecture. The identity provider (IdP) authenticates the user, and the orchestrator coordinates Agent A and Agent B. The agents submit information requests to the retriever, which searches the index. The policy decision point (PDP) evaluates the applicable policy and supplies its decision to the policy enforcement point (PEP). The PEP is the enforcement boundary: retrieved candidates are checked before their content is passed to the large language model (LLM). The agents must not receive unauthorized chunks through tool results or intermediate summaries; delegation is limited to the initiating user's permissions and the assigned task scope.

The workflow has two operational phases. During offline ingestion, documents are parsed, split into chunks, embedded, and stored in the index with resource identifiers and security metadata; this happens before a user submits a query. During online query processing, the system authenticates each request, evaluates current policy, retrieves permitted candidates, enforces the decision, and generates an answer. “Offline” therefore means document preparation, not that the system is disconnected from the network.

Audit records should include a request identifier, pseudonymous principal identifier, policy decision reference, retrieved resource identifiers, agent/tool actions, and final answer provenance. Logs should not duplicate unnecessary sensitive text and must themselves have access controls and retention rules.

V. METHODOLOGY AND EVALUATION PLAN

A. Survey Method

This paper uses a focused literature-based design method. The source set covers foundational RAG, sentence embeddings, RBAC, ABAC, authorization in multi-agent RAG, and vector-store filtering [1]–[6]. The sources were compared for retrieval ordering, policy evaluation, delegation, and auditability. The synthesis informs the architecture and evaluation plan. This is not an exhaustive systematic review.

B. Prototype Components

A prototype can use Python, FastAPI, Qdrant, a BGE or Sentence Transformers embedding model, a cross-encoder reranker, a language model, and PostgreSQL for policy and audit metadata. React or HTML/CSS may provide the interface. These are candidate components, not reported results. Current policy evaluation belongs in a trusted service; vector filters provide an additional enforcement layer.

C. Baseline and Test Scenarios

The baseline retrieves top-k candidates by semantic similarity and then removes unauthorized results before constructing the answer. The proposed variant evaluates policy first and restricts the candidate space before any content reaches rerankers or agents. Both variants should use the same corpus, embedding model, language model, query set, and hardware. Test scenarios should include a query whose most similar chunk is restricted; permission revocation between requests; cross-department synthesis; an agent attempting an out-of-scope tool call; and a prompt asking the model to reveal restricted information. Relevance labels must be separated from authorization labels.

D. Evaluation Metrics

Unauthorized-context exposure counts restricted chunks or derived artifacts reaching any downstream model context. Authorized Recall@k measures relevant chunks retrieved from the permitted set. Answer relevance should be assessed with a human-labelled rubric or fixed evaluator, with source grounding checked separately. Latency should be measured end-to-end and by stage. Revocation propagation time measures delay between a policy update and enforcement. Delegation violation rate measures out-of-scope agent actions, and audit completeness measures the fraction of required events recorded.

$$Exposure\ (\%) = 100 \times N_u / max(1, N_m) \quad (5)$$

N_u is the number of unauthorized information units reaching downstream model contexts; N_m is the total number reaching those contexts. For an authorization-correct implementation, N_u should be zero in the defined test set.

$$Authorized\ Recall@k = |Rel(q,u) \cap R_k(q,u)| / |Rel(q,u)| \quad (6)$$

Rel(q,u) denotes relevant chunks that the user is permitted to access. Queries with no authorized relevant chunks should be reported separately rather than assigned an artificial recall value.

$$T_r = t(enforcement\ observed) - t(policy\ update) \quad (7)$$

VI. DISCUSSION

Authorization-first retrieval can reduce the risk of exposing restricted content to downstream models, but strict filtering may reduce recall when metadata or policy mappings are incomplete. Evaluation must therefore measure security and retrieval utility together. The PDP should use current policy state and fail closed when authorization cannot be confirmed; each chunk and derived artifact should preserve provenance. Delegated agents must inherit and narrow the user's permissions. The proposed design remains to be implemented and benchmarked, so no performance gains are claimed.

VII. CONCLUSION

This paper presented a survey-informed architecture for authorization-aware multi-agent RAG. Its central requirement is to restrict the retrieval candidate set using current authorization decisions before content reaches rerankers, agents, or a language model. The design combines access-control policy, least-privilege delegation, provenance, and audit logging. Future work is to implement the prototype and measure security, relevance, and latency against a retrieve-then-filter baseline.

ACKNOWLEDGMENT

The authors acknowledge the Department of Artificial Intelligence and Machine Learning, Zeal Polytechnic, Pune, for academic support.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 9459–9474.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

[3] R. Namboothiri, "SoK: Authorization in multi-agent retrieval-augmented generation systems," preprint, Jan. 2026, doi: 10.5281/zenodo.18346863. [Online]. Available: https://rohithnamboothiri.com/research/sok-authorization-rag

[4] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: Towards a unified standard," in Proc. 5th ACM Workshop on Role-Based Access Control, 2000, pp. 47–63, doi: 10.1145/344287.344301.

[5] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST Special Publication 800-162, updated Aug. 2019, doi: 10.6028/NIST.SP.800-162.

[6] Qdrant, "Filtering," Qdrant Documentation. [Online]. Available: https://qdrant.tech/documentation/search/filtering/. Accessed: Oct. 28, 2026.

[7] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications," OWASP GenAI Security Project, 2025. [Online]. Available: https://genai.owasp.org/llm-top-10/.