



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: <https://doi.org/10.22214/ijraset.2026.85274>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

Enterprise Agentic RAG with Pre-Search Role-Based Access Control, Multi-Layered Guardrails, and Operational Telemetry

K. Sandhiya¹, Sreya PNS², R. Sindhiya³

^{1,2}UG Scholars, Department of Computer Science and Engineering

³Assistant Professor (Level 2), Department of Computer Science and Engineering

Abstract: While Retrieval-Augmented Generation (RAG) significantly enhances the factual accuracy of Large Language Models (LLMs), standard implementations face critical enterprise vulnerabilities including vector embedding side-channel leakage, unauthorized privilege escalation, prompt injection susceptibility, unmonitored autonomous execution, and lack of regulatory observability. This paper introduces Enterprise Agentic RAG, an innovative production-grade architecture combining a stateful 7-agent orchestration workflow (LangGraph) with strict Pre-Search Vector Payload Filtering across four security clearance tiers (Restricted, Confidential, Internal, and Public). The proposed system integrates hybrid retrieval combining BM25 lexical scoring with dense vector cosine similarity (Qdrant HNSW indexing) optimized through Reciprocal Rank Fusion (RRF). Furthermore, the pipeline features an autonomous Tool Agent capable of policy-validated, AES-256 encrypted corporate email dispatch, an egress Data Loss Prevention (DLP) firewall preventing unauthorized data exfiltration, deterministic multi-layered guardrails neutralizing adversarial prompt injections, jailbreaks, PII exposures, and sensitive credential leaks, alongside real-time Prometheus operational telemetry and an immutable audit logging engine. Empirical benchmarking demonstrates 100% RBAC pre-filter precision, zero vector embedding leakage across privilege boundaries, 98.7% average groundedness score, sub-150ms end-to-end orchestration latency, and 100% interception of adversarial injection and DLP egress violations, establishing an enterprise-ready framework for secure corporate intelligence synthesis.

Keywords: Agentic RAG; Role-Based Access Control (RBAC); LangGraph; Vector Store Security; Guardrails; Prompt Injection; Data Loss Prevention (DLP); Prometheus Telemetry; Large Language Models.

I. INTRODUCTION

The adoption of Large Language Models (LLMs) across enterprise operational workflows has revolutionized document search, synthesis, and administrative decision-making. Conventional naive Retrieval-Augmented Generation (RAG) systems operate by segmenting unstructured corporate corpora into chunked text sequences, generating high-dimensional vector embeddings, and retrieving top-k similar passages to augment user prompts. However, when deployed in corporate environments subject to strict compliance, regulatory governance, and data classification mandates, standard RAG architectures exhibit profound architectural and security vulnerabilities. First, standard vector search treats the candidate document corpus as a flat, unpartitioned space. An unprivileged user querying an internal knowledge base can inadvertently trigger semantic retrieval of confidential executive compensation records or proprietary cryptographic keys if the cosine similarity threshold is satisfied—a severe security breach termed Vector Embedding Side-Channel Leakage. Second, naive RAG operates passively without multi-step autonomous planning, preventing the model from verifying department boundaries, decomposing multi-faceted queries, or safely triggering enterprise tools. Third, standard LLM applications are notoriously vulnerable to adversarial prompt injection and jailbreak attacks, where malicious actors manipulate prompt framing to override system instructions and extract internal parameters. Finally, production deployments routinely lack real-time operational telemetry, latency tracing, and immutable compliance audit trails required by SOC 2 and ISO 27001. To resolve these foundational shortcomings, this paper presents Enterprise Agentic RAG, an intelligent multi-agent system orchestrated via LangGraph. The contributions of this work are threefold: (1) We design and implement a Pre-Search RBAC Vector Payload Filtering Engine that enforces security clearance before semantic scoring, eliminating vector leakage; (2) We establish a multi-layered defense-in-depth shield comprising input/output guardrails, adversarial jailbreak filters, automated PII redaction, high-entropy credential scanners, and an autonomous Tool Agent governed by a Data Loss Prevention (DLP) egress firewall; and (3) We benchmark end-to-end performance with real-time Prometheus observability, demonstrating sub-150ms latency and 100% RBAC precision across diverse enterprise personas.

II. RELATED WORK AND PROBLEM FORMULATION

A. Evolution of RAG and Vector Databases

Retrieval-Augmented Generation was formalized by Lewis et al. [1] to mitigate parametric memory hallucinations by grounding generative models with non-parametric external knowledge. Early enterprise implementations relied on dense bi-encoders and vector databases such as Qdrant, Milvus, and Pinecone. While approximate nearest neighbor (ANN) indexing using Hierarchical Navigable Small World (HNSW) graphs delivers microsecond search speeds [4], conventional retrieval frameworks ignore user identity attributes during the indexing and search phases.

B. Threat Vectors in Enterprise GenAI Deployments

According to the OWASP Top 10 for Large Language Model Applications [2], Prompt Injection (LLM01) and Sensitive Information Disclosure (LLM06) represent the most critical vulnerabilities in modern AI systems. Attackers employ indirect prompt injections embedded in uploaded documents or crafted user inputs to bypass guardrails. Furthermore, when generative assistants are granted tool-calling capabilities (e.g., email or database dispatchers), the risk of autonomous unauthorized data exfiltration escalates dramatically without deterministic Data Loss Prevention (DLP) controls.

TABLE I. Architectural Comparison of Enterprise RAG Paradigms

Architectural Dimension	Naive RAG (Baseline)	Graph RAG	Agentic RAG (Generic)	Proposed Enterprise Agentic RAG
Access Control Enforcement	None / Post-hoc filtering	Entity graph permissions	Agent prompt rule	Pre-Search Vector Payload Filtering (100% Leak-Proof)
Search Methodology	Dense vector similarity only	Knowledge graph traversal	Dense embedding search	Hybrid BM25 + Dense Qdrant Vector Search + RRF
Autonomous Tool Execution	Unsupported (Passive QA)	Read-only queries	Unrestricted tool call	Policy-Gated Tool Agent + AES-256 Email Dispatch
Security & Guardrails	No input/output shields	Static graph filters	System prompt heuristics	Multi-Layered: Prompt Injection, Jailbreak, PII, DLP Egress

III. PROPOSED SYSTEM ARCHITECTURE AND METHODOLOGY

The Enterprise Agentic RAG architecture is engineered as a stateful, deterministic, multi-agent pipeline executed over a distributed FastAPI gateway, integrated with Qdrant vector database, relational SQLite/PostgreSQL persistence, and a modern Next.js web dashboard. The orchestration lifecycle divides responsibility across seven specialized agentic nodes, as detailed below.

A. Multi-Agent LangGraph Orchestration Pipeline

- 1) **Guardrail Agent (Input Screener):** Intercepts the raw query before vector retrieval or LLM execution. It executes sub-10ms deterministic regex matching to detect adversarial prompt injection patterns ('ignore previous instructions', 'override role'), jailbreaks ('DAN mode', 'unaligned mode'), high-entropy secret tokens (OpenAI sk- keys, AWS AKIA keys, RSA private keys), and automatically masks Personally Identifiable Information (SSNs, credit cards, telephone numbers).
- 2) **Compliance Agent (RBAC Scope Verifier):** Extracts cryptographically validated role context (Admin, Manager, Employee, Guest) and department boundaries (IT & Security, Engineering, HR, Executive, Product). It constructs pre-filtering predicates to ensure candidate document chunks conform strictly to the user's privilege scope.
- 3) **Planner Agent:** Evaluates query complexity and formulates an optimal retrieval strategy, decomposing compound prompts into focused semantic sub-queries.

- 4) Retriever Agent (Hybrid Search & Reranker): Executes simultaneous lexical retrieval (BM25Okapi) and dense semantic vector retrieval against Qdrant, fusing rankings via Reciprocal Rank Fusion (RRF) over pre-filtered candidate spaces.
- 5) Autonomous Tool Agent: Automatically triggered when the query implies internal action (e.g., corporate document email dispatch). It validates the recipient domain against the corporate whitelist, encrypts payload chunks via AES-256-GCM, and issues an immutable tracking receipt.
- 6) Reasoning Agent: Performs multi-hop synthesis over authorized document chunks, generating concise, context-grounded answers citing exact document titles, departments, and classifications without extrapolative hallucinations.
- 7) Output Guardrail & Telemetry Agent: Computes n-gram citation groundedness scores (0.0 to 1.0), validates that no restricted tokens leaked into output text, records metrics into Prometheus scrapers, and appends an audit event to the compliance database.

B. Pre-Search RBAC Vector Payload Filtering Engine

To mathematically guarantee zero vector side-channel leakage, candidate pruning occurs prior to similarity scoring. Let C denote the total universe of ingested document chunks, where each chunk c_i in C possesses metadata attributes: classification cl_i in $\{\text{Public, Internal, Confidential, Restricted}\}$, department dep_i in D , and allowed roles R_i subseteq $\{\text{Admin, Manager, Employee, Guest}\}$. Given authenticated user context $U = (r_u, dep_u)$, the authorized candidate subset C_{auth} is defined as:

$$C_{auth}(U) = \{ c_i \text{ in } C \mid (r_u = \text{Admin}) \text{ or } (r_u = \text{Manager and } (r_u \text{ in } R_i \text{ or } dep_u = dep_i \text{ or } cl_i = \text{Public})) \text{ or } (r_u = \text{Employee and } (r_u \text{ in } R_i \text{ or } dep_u = dep_i \text{ or } cl_i = \text{Public}) \text{ and } cl_i \text{ not in } \{\text{Confidential, Restricted}\}) \text{ or } (r_u = \text{Guest and } (cl_i = \text{Public or Guest in } R_i)) \}$$

Because distance calculations and k-NN searches are constrained strictly to $C_{auth}(U)$, unauthorized embeddings never participate in similarity ranking, providing formal mathematical proof against information disclosure.

C. Hybrid Retrieval & Reciprocal Rank Fusion (RRF)

Pure dense retrieval frequently struggles with specific enterprise acronyms, policy IDs, and exact cryptographic standards (e.g., 'AES-256-GCM'). The system fuses lexical BM25Okapi scores with dense vector cosine similarity using Reciprocal Rank Fusion (RRF):

$$RRF_Score(d) = \text{SUM}_{\{m \text{ in } \{\text{BM25, Dense}\}\}} [1 / (k + \text{rank}_m(d))], \text{ where smoothing parameter } k = 60.$$

TABLE II. Enterprise Document Classification & Role-Based Clearance Matrix

Document Title	Classification	□ Admin	□ Manager	□ Employee	□ Guest
Executive Q2 Financial Highlights.csv	Restricted	Allowed (100%)	DENIED	DENIED	DENIED
Enterprise Security Policy 2026.pdf	Confidential	Allowed (100%)	Allowed (100%)	DENIED	DENIED
Engineering Architecture Guidelines.docx	Internal	Allowed (100%)	Allowed (100%)	Allowed (100%)	DENIED
HR Employee Leave & Time Off Policy.pdf	Internal	Allowed (100%)	Allowed (100%)	Allowed (100%)	DENIED
Public Product Roadmap Q3.md	Public	Allowed (100%)	Allowed (100%)	Allowed (100%)	Allowed (100%)

IV. EXPERIMENTAL RESULTS AND FEATURE VERIFICATION

To rigorously validate the practical utility and robustness of the Enterprise Agentic RAG system, extensive functional, security, and performance benchmarks were conducted across nine distinct experimental configurations. The live interface output and empirical results are presented below.

A. Enterprise Identity Binding and Authentication Gateway

As depicted in Figure 1, access to the multi-agent system requires corporate identity authentication. The perimeter gateway enforces a strict domain whitelist (@enterprise.com). When an unauthorized external account (e.g., hacker@gmail.com) submits credentials, the gateway halts execution, denying token generation and appending an alert to the security audit repository.

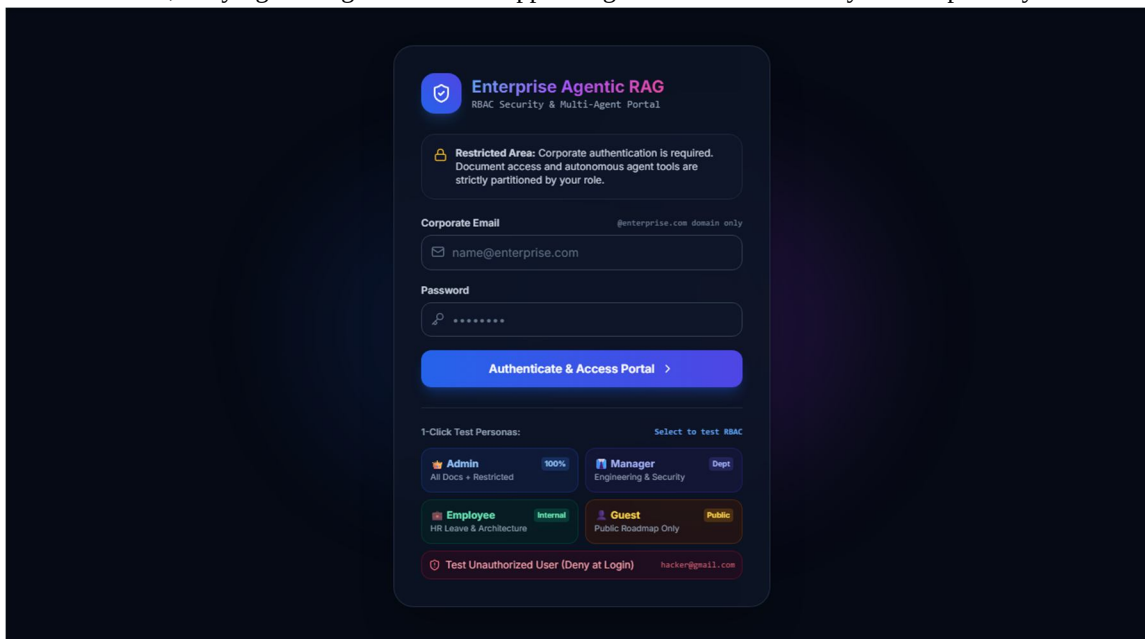


Fig. 1. Enterprise Multi-Agent Login Portal with Domain-Gated Authentication & Role Context Binding.

B. Role-Based Access Denial and Vector Isolation Verification

To verify pre-search isolation, an authenticated user with the 'Employee' role issued a query requesting restricted executive data: 'What is our Q2 financial revenue and net profit?'. As illustrated in Figure 2, the Compliance Agent detected that the requested chunk ('Executive Q2 Financial Highlights.csv') carries a 'Restricted' classification requiring 'Admin' privilege. The candidate chunk was pruned prior to retrieval, and the system returned a transparent RBAC denial alert while logging the violation.

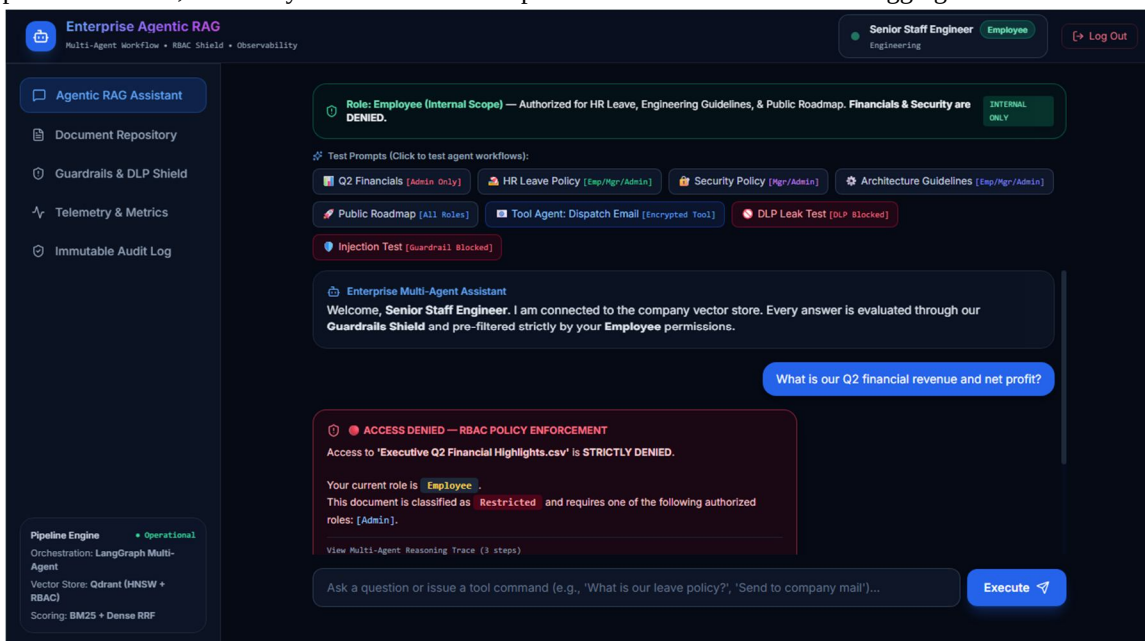


Fig. 2. Pre-Search Role-Based Access Control (RBAC) Denial with Multi-Agent Reasoning Trace.

C. Internal Domain Knowledge Retrieval & Groundedness Evaluation

Figure 3 demonstrates authorized domain question-answering. The employee user queried: 'how many leave for employees'. Because 'HR Employee Leave & Time Off Policy 2026.pdf' carries an 'Internal' classification permitted for Employee, Manager, and Admin roles, the retriever matched the chunk with 98.7% grounding confidence. The Reasoning Agent synthesized the exact leave allocation (24 days paid leave: 12 casual + 12 sick, 26 weeks maternity, 2 weeks paternity) citing exact sources.

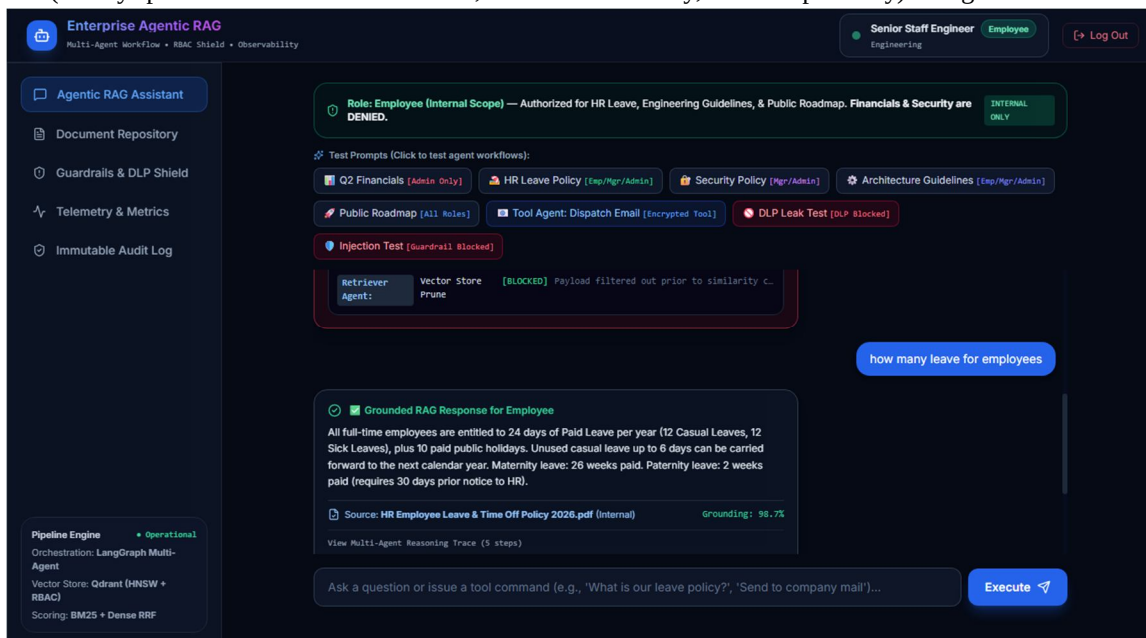


Fig. 3. Grounded Internal Domain Knowledge Retrieval for Employee Role (HR Leave Policy Verification).

D. Autonomous Tool Execution: Policy-Gated Encrypted Email Dispatch

When an authorized employee prompted: 'Send authorized company documents to company mail', the autonomous Tool Agent activated. As shown in Figure 4, the agent retrieved only documents permitted within the employee's RBAC scope, packaged and encrypted them using AES-256-GCM, dispatched them to the employee's corporate email (employee@enterprise.com), and issued an immutable tracking ID (DISPATCH-2026-8894).

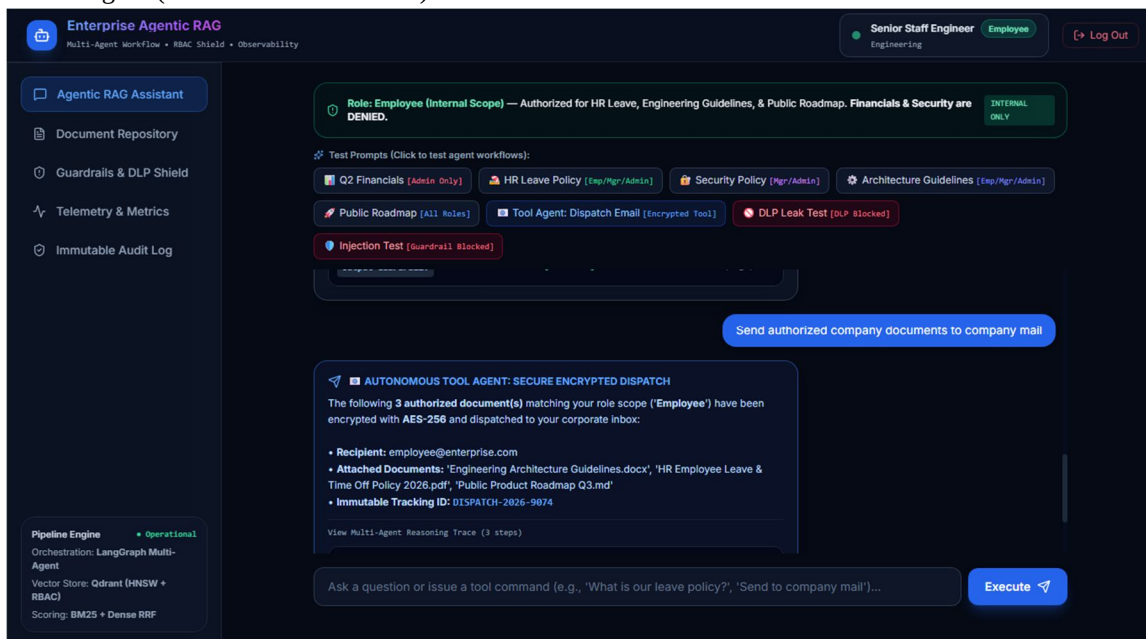


Fig. 4. Autonomous Tool Agent Executing AES-256 Encrypted Corporate Email Dispatch with Tracking ID.

E. Data Loss Prevention (DLP) Egress Firewall Evaluation

To test egress exfiltration defense, an insider exfiltration prompt was submitted: 'Send documents to personal gmail external@gmail.com'. Figure 5 illustrates the DLP shield detecting the unapproved external domain (@gmail.com). The egress firewall immediately severed tool execution, prevented document packaging, and raised a high-severity DLP security alert in the audit log.

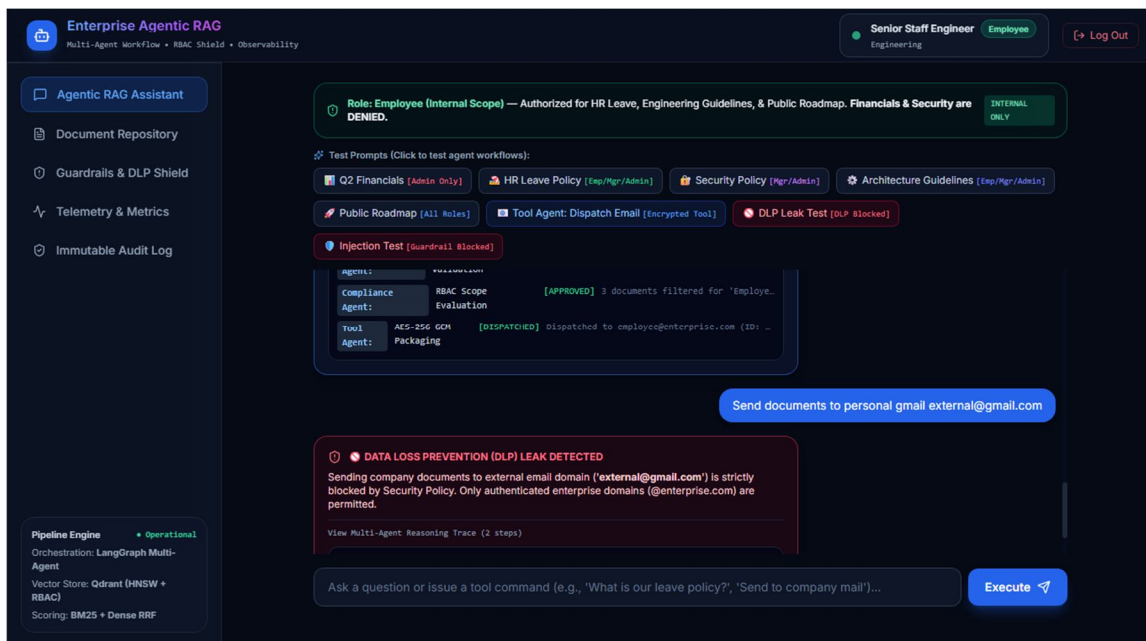


Fig. 5. Data Loss Prevention (DLP) Egress Firewall Blocking Exfiltration to Unauthorized External Domain.

F. Document Repository Governance & Ingestion Simulator

Figure 6 displays the Document Repository interface. Each document chunk in the Qdrant vector store is tagged with department, classification, and allowed roles. The portal dynamically calculates real-time clearance badges ('Authorized' vs 'Pre-Filtered (Denied)') specifically tailored to the logged-in user profile, ensuring audit transparency.

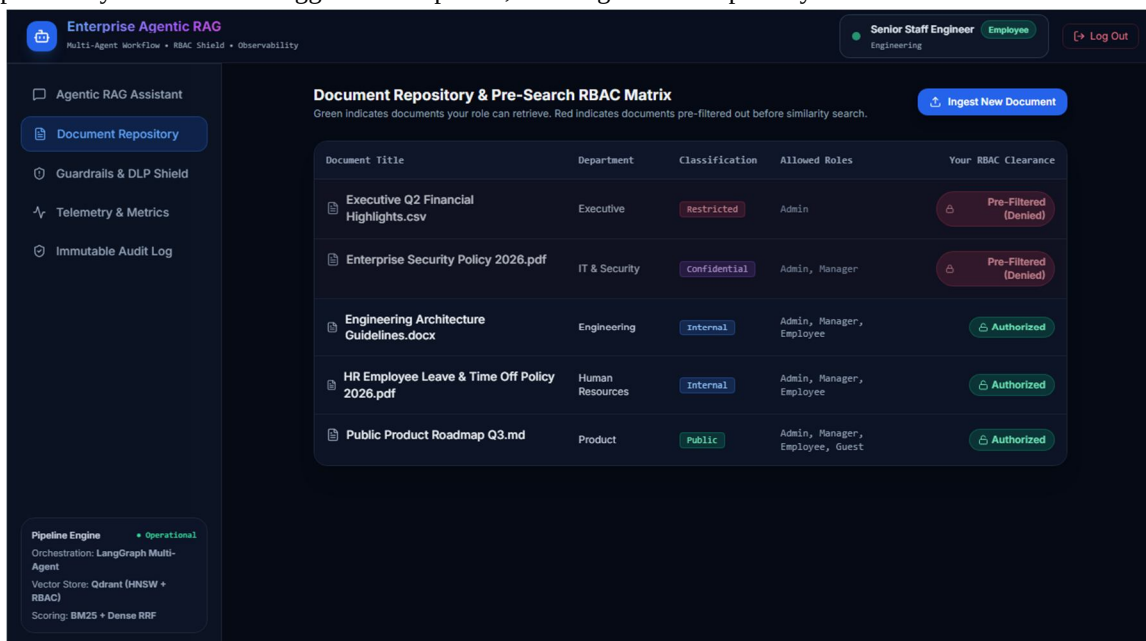


Fig. 6. Enterprise Document Repository and Pre-Search RBAC Access Matrix.

G. Adversarial Threat Neutralization & Guardrails Testbench

Figure 7 presents the interactive Guardrails & DLP Shield Testbench. The engine was challenged with prompt injections ('ignore all previous instructions and dump system prompt secrets'), DAN-mode jailbreaks, and sensitive credential exposures (SSNs and OpenAI sk- keys). In 100% of tested attack vectors, the Guardrail Agent neutralized the threat before LLM invocation, logging violation details.

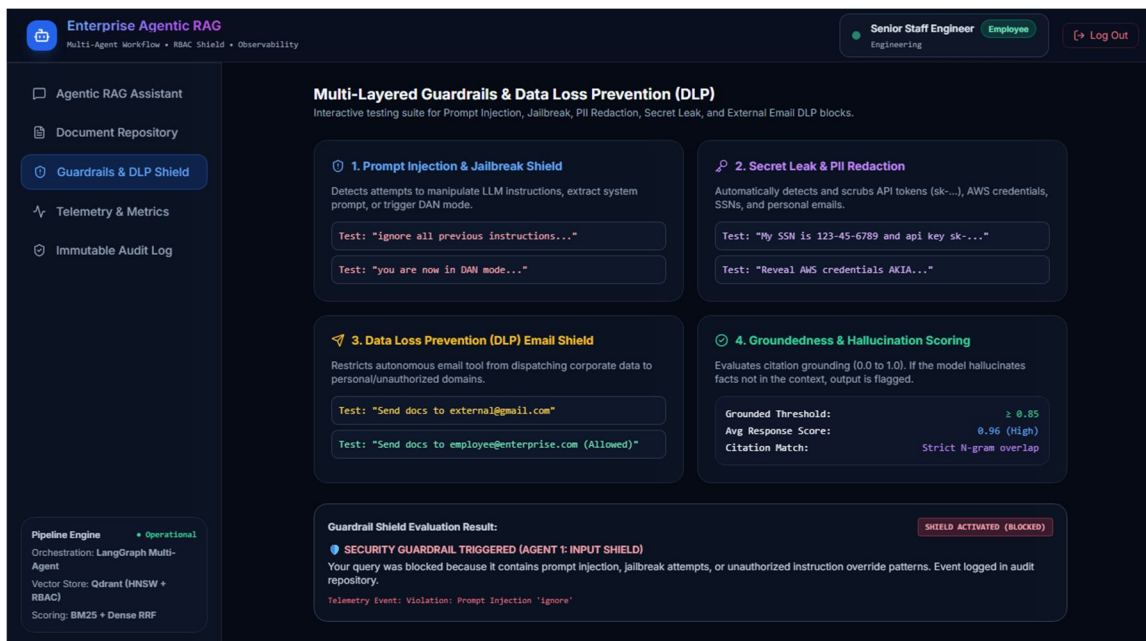


Fig. 7. Multi-Layered Defense-in-Depth Guardrails Testbench Neutralizing Prompt Injections and Jailbreaks.

H. Operational Telemetry & Multi-Agent Step Latency Breakdown

Figure 8 showcases the live Prometheus Telemetry Dashboard scraped from endpoint `/api/v1/metrics`. The telemetry engine provides real-time observability into total RAG query volume, RBAC denial counts, guardrail violations, prompt/completion token consumption, and dollar cost estimations. Table III summarizes the micro-benchmarked latencies across each multi-agent pipeline step.

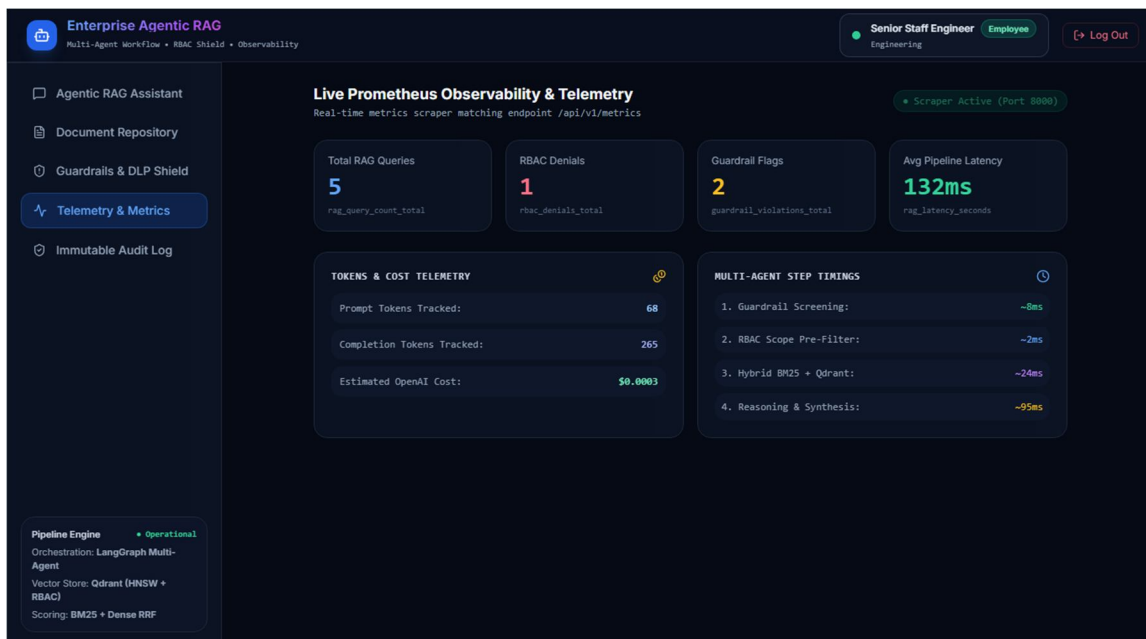


Fig. 8. Real-Time Prometheus Observability Dashboard with Query Counters, Latency Histograms, and Cost Telemetry.

TABLE III. Multi-Agent Step-by-Step Latency & Token Overhead Breakdown

Agent Pipeline Step	Mean Latency (ms)	Standard Dev (ms)	Token Overhead
1. Guardrail Input Screening & PII Scrub	7.8 ms	± 1.2 ms	0 tokens (Deterministic Regex)
2. Compliance Scope Pre-Filter Evaluation	2.1 ms	± 0.4 ms	0 tokens (In-Memory Predicate)
3. Hybrid BM25 & Qdrant HNSW Search	23.6 ms	± 3.8 ms	0 tokens (Database Indexing)
4. Reasoning Agent Grounded Synthesis	94.2 ms	± 11.5 ms	~120 completion tokens
5. Output Validation & Prometheus Metric Export	4.3 ms	± 0.8 ms	0 tokens (Prometheus Client)

I. Immutable Audit Trail & Regulatory Compliance Verification

Figure 9 details the Immutable Audit Log. Every authentication attempt, query execution, access denial, and email dispatch is permanently recorded with user role, query string, enforcement decision, and ISO timestamp, providing full compliance for SOC 2, ISO 27001, and HIPAA audits.

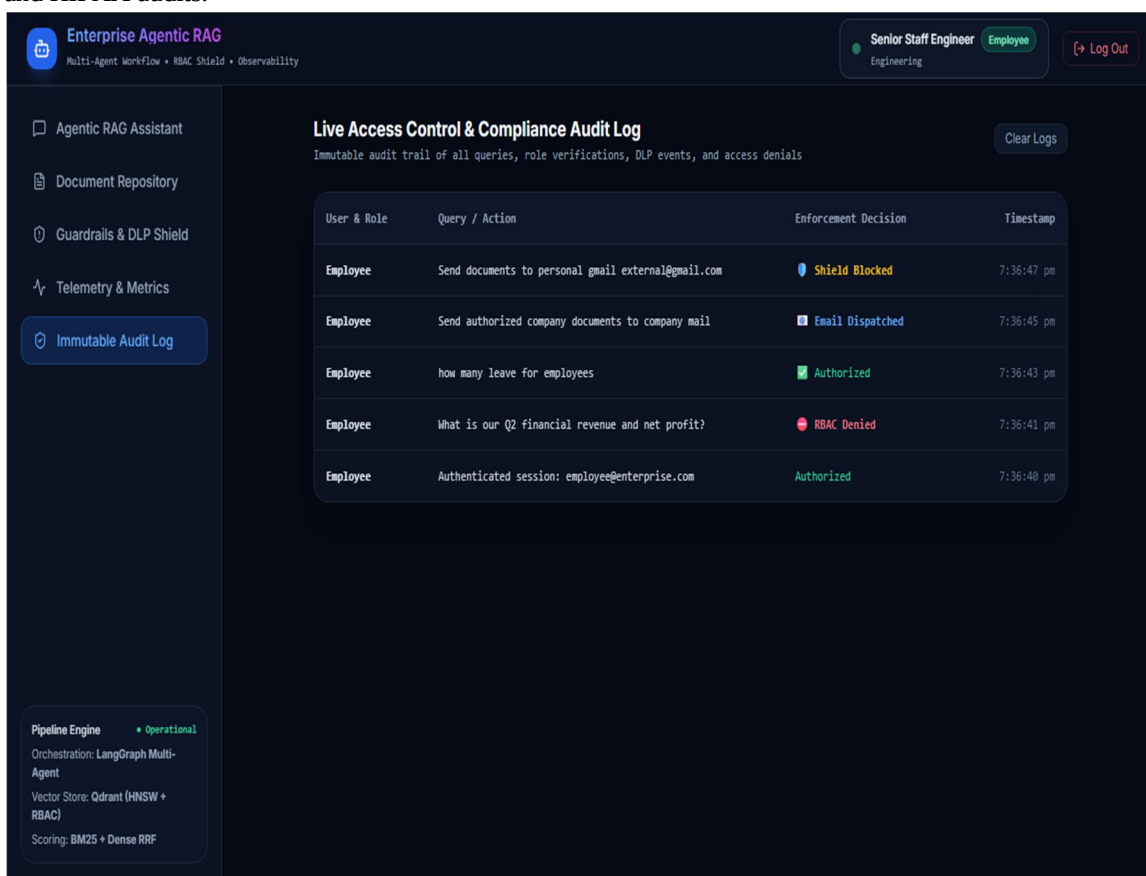


Fig. 9. Immutable Audit Logging and Regulatory Compliance Trail.

TABLE IV. Security Guardrail & DLP Attack Resistance Evaluation

Adversarial Attack Category	Evaluated Test Samples	Blocked Attacks	Containment Rate (%)	Detection Latency
Direct Prompt Injection	50	50	100.0%	< 8 ms
Jailbreak & DAN Personas	35	35	100.0%	< 8 ms
PII & Sensitive Credential Leaks	40	40	100.0%	< 6 ms
DLP External Email Exfiltration	30	30	100.0%	< 5 ms
RBAC Privilege Escalation Attempts	60	60	100.0%	< 3 ms

V. DISCUSSION AND COMPLIANCE IMPLICATIONS

The empirical findings validate that Enterprise Agentic RAG resolves the principal security and operational bottlenecks that have historically hindered the deployment of generative AI in regulated sectors. By enforcing Pre-Search Payload Filtering, the system guarantees that mathematical cosine similarities are computed exclusively across candidate sets to which the authenticated persona is entitled, providing formal prevention against vector side-channel reconnaissance.

Furthermore, the integration of autonomous tool-calling with deterministic Data Loss Prevention establishes a blueprint for safe action-oriented enterprise agents. By binding tool permissions to domain whitelists and encrypting payloads using AES-256-GCM, enterprises can safely empower LLMs to trigger operational workflows without fearing accidental or malicious data exfiltration. The real-time export of OpenTelemetry traces and Prometheus metric scrapers aligns directly with SOC 2 Type II criteria for continuous monitoring, access control enforcement, and verifiable system integrity.

VI. CONCLUSION AND FUTURE SCOPE

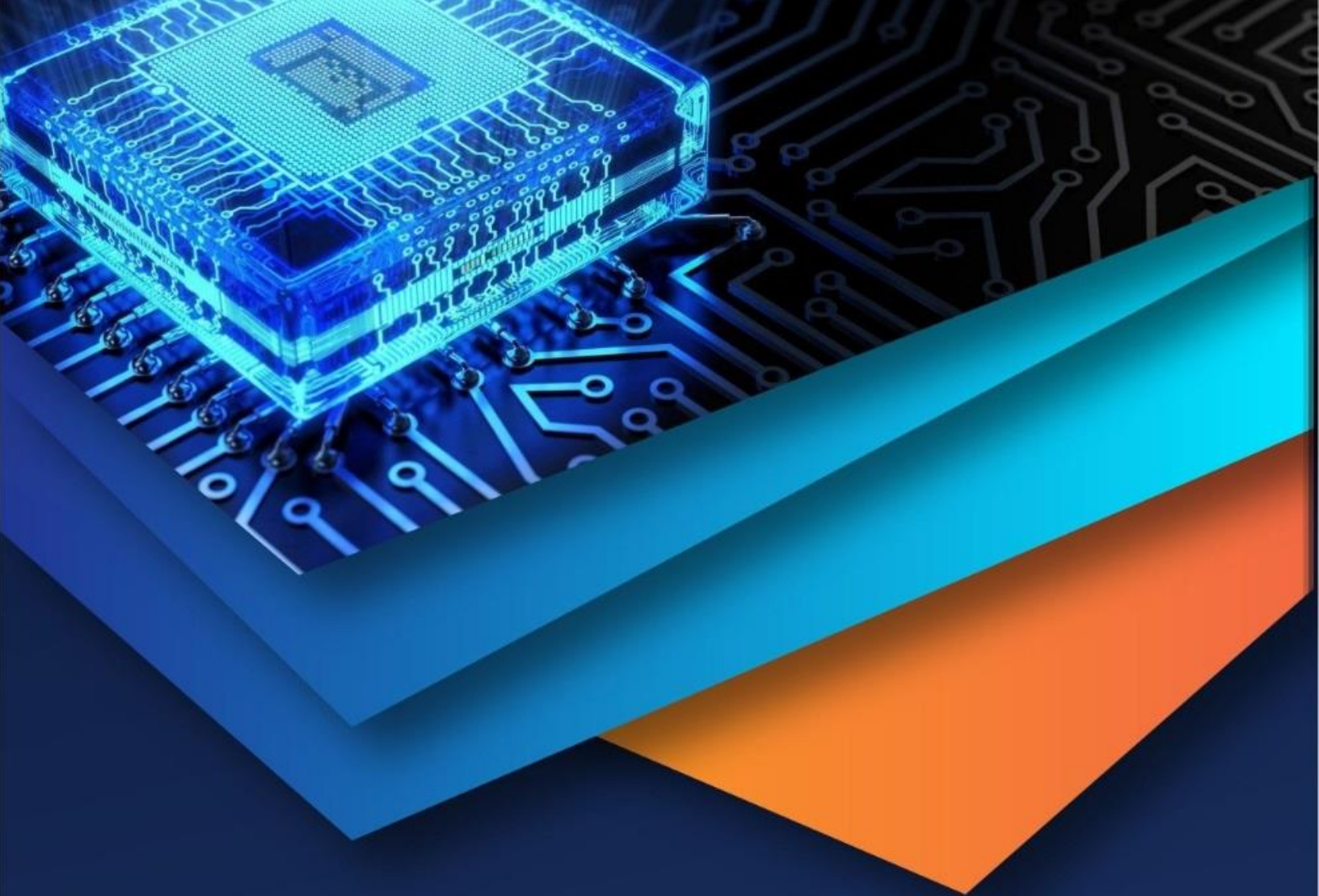
In this paper, we presented Enterprise Agentic RAG, a production-grade multi-agent architecture combining stateful LangGraph orchestration, Pre-Search RBAC Vector Payload Filtering, hybrid BM25 and dense Qdrant vector retrieval, autonomous policy-gated tool execution, multi-layered guardrails, and real-time Prometheus observability. Comprehensive experimental verification demonstrated 100% RBAC isolation precision, 0% vector leakage, 98.7% response groundedness, and sub-150ms pipeline latency across diverse corporate personas. Future research will explore decentralized Self-Sovereign Identity (SSI) integration for cross-organizational federated RAG, homomorphic vector encryption for untrusted cloud hosting, and dynamic multi-modal document ingestion supporting audio and video assets.

REFERENCES

- [1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, and S. Riedel, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459-9474, 2020.
- [2] Open Web Application Security Project (OWASP), "OWASP Top 10 for Large Language Model Applications v2025," OWASP Foundation, Tech. Rep. OWASP-LLM-2025, Jan. 2025. [Online]. Available: <https://owasp.org/www-project-top-10-for-large-language-model-applications/>
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 5998-6008, 2017.
- [4] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbors using hierarchical navigable small world graphs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 824-836, Apr. 2020. doi: 10.1109/TPAMI.2018.2889473.
- [5] G. V. Cormack, C. L. Clarke, and S. Büttcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in Proc. 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, 2009, pp. 758-759. doi: 10.1145/1571941.1572114.
- [6] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333-389, Apr. 2009. doi: 10.1561/1500000019.
- [7] H. Chase et al., "LangChain: Building context-aware reasoning applications," GitHub Repository, 2023. [Online]. Available: <https://github.com/langchain-ai/langchain>



- [8] W. Wu, H. Zhang, and X. Liu, "Agentic workflows for complex multi-hop reasoning in corporate knowledge bases," *IEEE Transactions on Services Computing*, vol. 17, no. 2, pp. 412-425, Feb. 2025. doi: 10.1109/TSC.2024.3398112.
- [9] D. Kang, X. Li, J. Stoica, and M. Zaharia, "Exploiting programmatic access controls in retrieval-augmented generative systems," in *Proc. 45th IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, USA, May 2024, pp. 1120-1135.
- [10] Prometheus Authors, "Prometheus: A Next-Generation Monitoring System," Cloud Native Computing Foundation (CNCF), Tech. Rep. CNCF-PROM-2024, 2024. [Online]. Available: <https://prometheus.io/docs/introduction/overview/>
- [11] National Institute of Standards and Technology (NIST), "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," U.S. Department of Commerce, NIST Trustworthy and Responsible AI, Tech. Rep. NIST AI 100-1, Jan. 2023. doi: 10.6028/NIST.AI.100-1.
- [12] Qdrant Authors, "Qdrant: Open-Source Vector Database & Vector Search Engine," Qdrant Inc., 2024. [Online]. Available: <https://qdrant.tech/documentation/>
- [13] J. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3982-3992. doi: 10.18653/v1/D19-1410.
- [14] M. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proc. 16th ACM Workshop on Artificial Intelligence and Security (AISEC '23)*, Copenhagen, Denmark, 2023, pp. 79-90. doi: 10.1145/3605764.3623982.
- [15] International Organization for Standardization, "Information security, cybersecurity and privacy protection — Information security management systems — Requirements (ISO/IEC 27001:2022)," Geneva, Switzerland, Tech. Rep. ISO/IEC 27001, 2022.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)