

Enterprise RAG Architecture

A Comprehensive Enterprise Architecture Approach for Secure Retrieval-Augmented Generation Using Business Data, Vector Search, AI Models, Governance, and Enterprise Knowledge

Sanjeeve Kumar Gajadi*

*(HCL Technologies, Bengaluru, Karnataka, India)
Email: sanjeevekumar.g@hcltech.com

Abstract:

Retrieval-Augmented Generation (RAG) has emerged as a foundational architecture pattern for grounding Generative AI in trusted enterprise knowledge. This paper proposes an Enterprise RAG Architecture that integrates governed content ingestion, metadata, semantic chunking, embeddings, vector search, hybrid retrieval, re-ranking, authorization, prompt orchestration, model routing, grounding, citations, evaluation, observability, and lifecycle governance. The framework connects enterprise information from SAP and non-SAP sources with SAP Business Technology Platform, SAP Business Data Cloud, SAP Datasphere, SAP HANA Cloud, SAP AI services, APIs, and secure integration patterns. It addresses critical enterprise requirements including source-level authorization, privacy, prompt injection defense, stale-content management, retrieval quality, hallucination reduction, model governance, and Agentic RAG. The proposed architecture enables organizations to build scalable and reusable AI assistants and agents that deliver contextual, explainable, secure, and business-aligned responses across hybrid and multi-cloud environments.

Keywords — Retrieval-Augmented Generation, RAG, Enterprise AI, SAP BTP, SAP Business Data Cloud, SAP Datasphere, Vector Search, Embeddings, Generative AI, Agentic RAG, Prompt Security, AI Governance, Enterprise Knowledge

I. INTRODUCTION

Retrieval-Augmented Generation (RAG) has become a core enterprise architecture pattern for grounding Generative AI in trusted organizational information. Instead of relying solely on a foundation model's pretraining, a RAG solution retrieves relevant enterprise content at runtime and supplies that context to the model before a response is generated.

Enterprise RAG requires more than a vector database and a language model. It must integrate content ingestion, metadata, semantic chunking, embeddings, retrieval, ranking, authorization, prompt construction, model orchestration, observability, governance, and lifecycle management across structured and unstructured information.

II. BUSINESS PROBLEM

Enterprises hold knowledge across ERP systems, data warehouses, document repositories, collaboration platforms, policies, technical manuals, knowledge bases, and external sources. Employees often spend significant time searching for information, reconciling conflicting sources, and interpreting complex content.

Public Generative AI models cannot safely or reliably provide proprietary enterprise context by default. Directly fine-tuning models on frequently changing business information can also be expensive and difficult to govern. RAG addresses these challenges by retrieving current and authorized enterprise information at inference time.

III. ARCHITECTURE OBJECTIVES

The Enterprise RAG Architecture is designed to provide accurate, contextual, secure, explainable, and governed responses. It should minimize hallucination, preserve source attribution, enforce user-level access, protect sensitive content, support multilingual retrieval, and scale across business domains.

The architecture should also enable reuse. Common ingestion, embedding, retrieval, evaluation, and observability services reduce duplicated implementation and provide consistent governance across enterprise copilots and AI agents.

IV. SOURCE AND INGESTION LAYER

RAG content may originate from SAP S/4HANA, SAP BW/4HANA, SAP Business Data Cloud, SAP

Datasphere, SharePoint, file repositories, knowledge portals, service systems, product documentation, APIs, and external sources. Ingestion pipelines extract content together with metadata such as owner, source, classification, validity dates, language, and access permissions.

Pipelines should support incremental updates, delete propagation, lineage, schema evolution, deduplication, content validation, and quarantine of suspicious or low-quality sources. Sensitive sources should be excluded unless business purpose and access controls are clearly defined.

V. DOCUMENT PROCESSING AND CHUNKING

Document processing converts raw files and records into retrieval-ready units. Parsing should preserve headings, tables, sections, document hierarchy, and semantic relationships whenever possible. Chunking strategies should consider content type rather than applying one fixed length to every source.

Metadata should travel with each chunk so retrieval can apply business filters, authorization, recency, language, product, geography, and source quality. Poor chunking can fragment meaning and reduce answer quality even when the underlying model is strong.

VI. EMBEDDING AND VECTOR ARCHITECTURE

Embeddings transform text or other content into numerical representations that support semantic similarity search. Enterprise architecture should define approved embedding models, versioning, language coverage, dimensions, hosting location, cost, and security requirements.

Vector stores must support encryption, tenant isolation, backup, retention, access control, and lifecycle management. Embeddings and indexes should be treated as sensitive enterprise assets because they represent semantic information derived from source content.

VII. RETRIEVAL AND RANKING

Retrieval should combine semantic search with metadata filtering and, where useful, lexical or keyword search. Hybrid retrieval improves robustness when exact business identifiers, product codes, or technical terms are important. Re-ranking models can further improve relevance by evaluating the initial candidate set against the user's question.

Enterprise retrieval should optimize both recall and precision. Excessive context increases cost and may confuse the model, while insufficient context increases hallucination risk. Retrieval evaluation should therefore use representative business questions and labeled relevance sets.

VIII. AUTHORIZATION AND SECURITY

Authorization must be enforced before protected content is delivered to the model. The user or agent's identity should flow through retrieval so that the system only returns content the requester is authorized to access. Application-level filtering alone is insufficient for sensitive enterprise knowledge.

Security controls should include SSO, OAuth 2.0, workload identity, encryption, private connectivity, secret management, data-loss prevention, prompt injection defenses, source trust validation, and logging. Untrusted retrieved content should be treated as data, not executable instructions.

IX. PROMPT ORCHESTRATION

Prompt orchestration assembles system instructions, user questions, retrieved evidence, policy context, and tool instructions into a controlled request to the model. System prompts should define role, boundaries, citation expectations, uncertainty behavior, and prohibited actions.

Context windows should be managed efficiently through ranking, summarization, deduplication, and compression. The system should avoid silently mixing contradictory sources; where conflicts exist, responses should surface the disagreement or request clarification.

X. MODEL AND AI RUNTIME

The architecture should support approved foundation models and enterprise AI runtimes such as SAP AI Core or other governed model endpoints. Model selection may consider task quality, context length, latency, region, cost, licensing, privacy, and business risk.

Model routing can direct requests to different models based on sensitivity, complexity, or cost. Version changes should trigger evaluation because model behavior can change even when the RAG pipeline remains constant.

XI. GROUNDING AND CITATION

Grounded responses should be traceable to enterprise sources. The application should preserve document identifiers, section references, timestamps,

and source metadata so users can verify important claims. Citation improves trust and supports auditability.

The system should distinguish sourced facts from model-generated synthesis. If retrieval confidence is low or the answer is unsupported by available sources, the preferred behavior is to state uncertainty rather than fabricate content.

XII. EVALUATION FRAMEWORK

RAG evaluation should measure retrieval relevance, groundedness, faithfulness, answer completeness, citation accuracy, latency, cost, security, and user usefulness. Automated evaluation can accelerate testing, but representative human review remains important for high-impact domains.

Evaluation datasets should include common questions, ambiguous questions, adversarial prompts, permission-sensitive requests, outdated content, conflicting sources, and domain-specific terminology. Regression testing should run whenever ingestion logic, embeddings, retrieval, prompts, or models change.

XIII. OBSERVABILITY AND OPERATIONS

RAG observability should capture user identity, query metadata, retrieval results, scores, filters, model version, token consumption, latency, citations, exceptions, and user feedback. Sensitive content may require masking while preserving operational traceability.

Operational monitoring should detect stale indexes, failed ingestion jobs, authorization errors, retrieval drift, rising hallucination rates, abnormal costs, and degraded response time. Alerting and runbooks support rapid remediation.

XIV. BUSINESS DATA CLOUD INTEGRATION

SAP Business Data Cloud and SAP Datasphere can provide governed semantic business data for RAG and AI agents. Structured business information can be exposed as business data products, while unstructured enterprise knowledge is linked through metadata, catalogs, and retrieval services.

Combining semantic business data with documents and Knowledge Graphs allows RAG applications to answer questions with richer enterprise context while maintaining governance and reuse.

XV. AGENTIC RAG

Agentic RAG extends basic retrieval by allowing an AI agent to formulate sub-queries, select retrieval tools, compare evidence, call enterprise APIs, and iteratively refine an answer. This can improve complex research and decision-support tasks, but it also increases security and governance requirements.

Agents require bounded permissions, approved tools, machine identity, action logging, human approval for high-impact operations, and safeguards against prompt injection or malicious retrieved instructions.

XVI. IMPLEMENTATION METHODOLOGY

Implementation begins with a clearly defined business domain, authoritative source inventory, user population, access model, evaluation questions, and success criteria. A pilot should validate ingestion, chunking, metadata, retrieval, grounding, security, and user experience before broad deployment.

Reusable platform services for ingestion, embeddings, vector storage, retrieval, evaluation, and observability should then be standardized. Governance should assign content ownership and recertification responsibilities so stale or unauthorized knowledge does not remain indefinitely.

XVII. INDUSTRY USE CASES

Enterprise RAG supports service agents, technical support, policy assistants, procurement knowledge, engineering manuals, sales enablement, software development, finance policy interpretation, and HR knowledge. In regulated industries, stronger source attribution and access control are essential.

RAG can also support executive decision support when combined with trusted business data, process intelligence, and decision governance.

XVIII. FUTURE TRENDS

Future RAG architectures will increasingly combine multimodal retrieval, Knowledge Graphs, semantic business data, agentic search, long-context models, real-time event retrieval, and adaptive query planning. Retrieval will become more personalized while preserving enterprise authorization and privacy.

Continuous evaluation and automated governance will become central as RAG services scale across many business domains and agents.

XIX. CONCLUSION

Enterprise RAG provides a practical architecture for grounding Generative AI in current, trusted, and authorized enterprise knowledge. Its value depends on far more than vector search; strong results require high-quality content, metadata, security, evaluation, orchestration, and lifecycle governance.

A well-designed RAG platform improves answer quality, reduces hallucination, protects enterprise information, and creates a reusable foundation for copilots, decision intelligence, and agentic AI.

REFERENCES

- [1] ISO/IEC 42001:2023, Information technology — Artificial intelligence — Management system.
- [2] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), 2023.
- [3] The Open Group, TOGAF Standard, 10th Edition, 2022.
- [4] ISO/IEC 27001:2022, Information security, cybersecurity and privacy protection — Information security management systems.
- [5] ISO 31000:2018, Risk management — Guidelines.
- [6] DAMA International, DAMA-DMBOK2: Data Management Body of Knowledge, 2017.
- [7] W3C, RDF 1.1 Concepts and Abstract Syntax.
- [8] W3C, SPARQL 1.1 Query Language.
- [9] W3C, Web Ontology Language (OWL) 2 Overview.
- [10] SAP SE, SAP Business Technology Platform Documentation.
- [11] SAP SE, SAP Business Data Cloud and SAP Datasphere Documentation.
- [12] SAP SE, SAP HANA Cloud Documentation.
- [13] SAP SE, SAP Analytics Cloud Documentation.
- [14] SAP SE, SAP Integration Suite Documentation.
- [15] SAP SE, SAP AI Core and SAP AI Launchpad Documentation.
- [16] SAP SE, SAP LeanIX Enterprise Architecture Documentation.
- [17] SAP SE, SAP Signavio Process Transformation Suite Documentation.
- [18] OECD, Recommendation of the Council on Artificial Intelligence.
- [19] IEEE, Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems.
- [20] Cloud Native Computing Foundation, Cloud Native Landscape and Kubernetes Documentation.

AUTHOR BIOGRAPHY

Sanjeev Kumar Gajadi is an SAP Enterprise Architect and Lead Consultant with extensive experience in enterprise architecture, SAP Business Technology Platform, data and analytics, integration, cloud architecture, security, governance, and AI-enabled transformation. His professional interests include Enterprise RAG, Generative AI architecture, SAP BTP, business data, vector search, AI governance, and agentic AI.

ACKNOWLEDGMENT

The author acknowledges the standards, research, and enterprise architecture communities that informed this work.