International Journal of Scientific Research and Modern Technology (IJSRMT) ijsrmt.com
Volume 4, Issue 8, 2025
DOI: https://doi.org/10.38124/ijsrmt.v4i8.977

Leveraging Retrieval-Augmented Generation (RAG) and LLMs to Develop a Multi-Team Confluence Insights Dashboard

Ravikant Singh[1]

[1]Sr. Data Engineering Manager

Publication Date: 2025/08/25

Abstract

Organizations that expand their operations make Confluence their main platform for team-wide knowledge exchange. The project information becomes fragmented and outdated when multiple teams contribute to the project. The paper presents a Retrieval-Augmented Generation (RAG)– enabled Multi-Team Confluence Insights Dashboard which retrieves team documentation to generate real-time visual analytics. The architecture leverages vector databases for scalable semantic search, large language models (LLMs) for context-aware summarization, and dynamic charting for actionable insights. The key strategies include scheduled re-indexing and metadata filtering for data freshness, vector database selection for scalability and latency optimization, RAG-based constraints for transparency and control, and multi-model orchestration to ensure deterministic, reliable outputs. The solution converts unstructured Confluence content into an interactive system which provides dependable decision-ready knowledge.

Keywords: *Confluence Insights Dashboard, Vector Databases, Retrieval-Augmented Generation (RAG), Metadata Filtering, Large Language Models (LLMs), Multi-Model Orchestration, Real- Time Analytics, Semantic Search, Knowledge Transparency.*

I. INTRODUCTION

Confluence serves as a standard tool for team documentation management through which teams store project specifications and meeting notes and design documents and operational procedures. The built-in search and indexing features function properly yet they do not possess sophisticated semantic understanding or cross-document summarization or real-time analytical features. Organizations face difficulties in extracting prompt useful information from extensive distributed content repositories because of this limitation (eGain, n.d.).

Retrieval-Augmented Generation (RAG) offers a compelling method to enhance the capabilities of Large Language Models (LLMs) performance by uniting semantic retrieval with generative reasoning capabilities. The system generates output based on current and context-specific information as shown in Figure 1 through this approach which minimizes the occurrence of hallucinations and reliance on obsolete information (Zhu, 2024).

Singh, R. (2025). Leveraging Retrieval-Augmented Generation (RAG) and LLMs to Develop a Multi-Team Confluence Insights Dashboard. *International Journal of Scientific Research and Modern Technology, 4*(8), 174–179. https://doi.org/10.38124/ijsrmt.v4i8.977

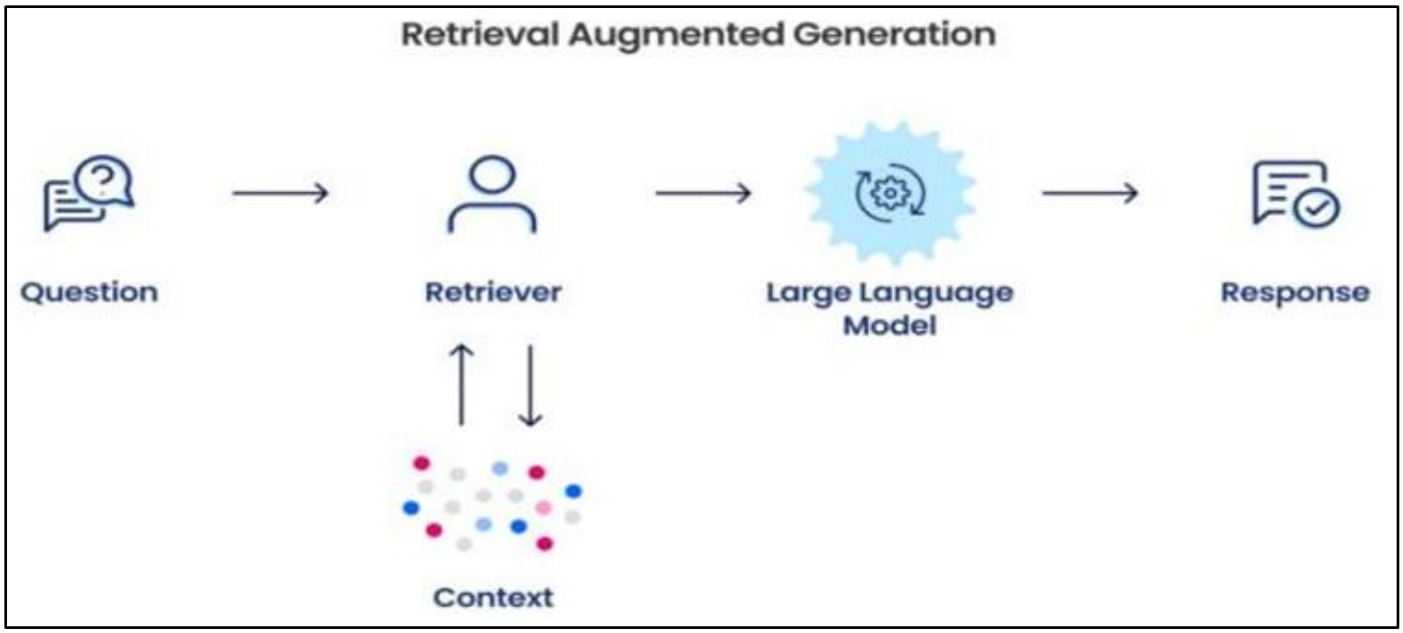

Fig 1 RAG with LLM (Tran, 2023).

- *The Proposed Multi-Team Confluence Insights Dashboard Serves as a Solution to These Problems Through:*

- The process demands the collection of documentation from various Confluence spaces into a single unified knowledge index.
- Processing and indexing data in a vector database for efficient semantic search.
- RAG pipelines function as a system for generating context-specific summaries.
- Visual analytics system development should concentrate on building tools which enable fast evidence-based decision-making capabilities.

The solution merges RAG with vector databases and interactive visualization frameworks to transform static Confluence documentation into an intelligent dynamic transparent decision- support platform (Zhu, 2024).

II. SYSTEM OVERVIEW

The Multi-Team Confluence Insights Dashboard utilizes a modular framework which combines data ingestion from Confluence with semantic processing and retrieval-augmented summarization and visual analytics. The modular design structure provides both scalability and maintainability and flexibility for upcoming enhancements (Zhang & Elhamod, 2025).

The high-level architecture and data flow of Figure 2 demonstrates the following stages.

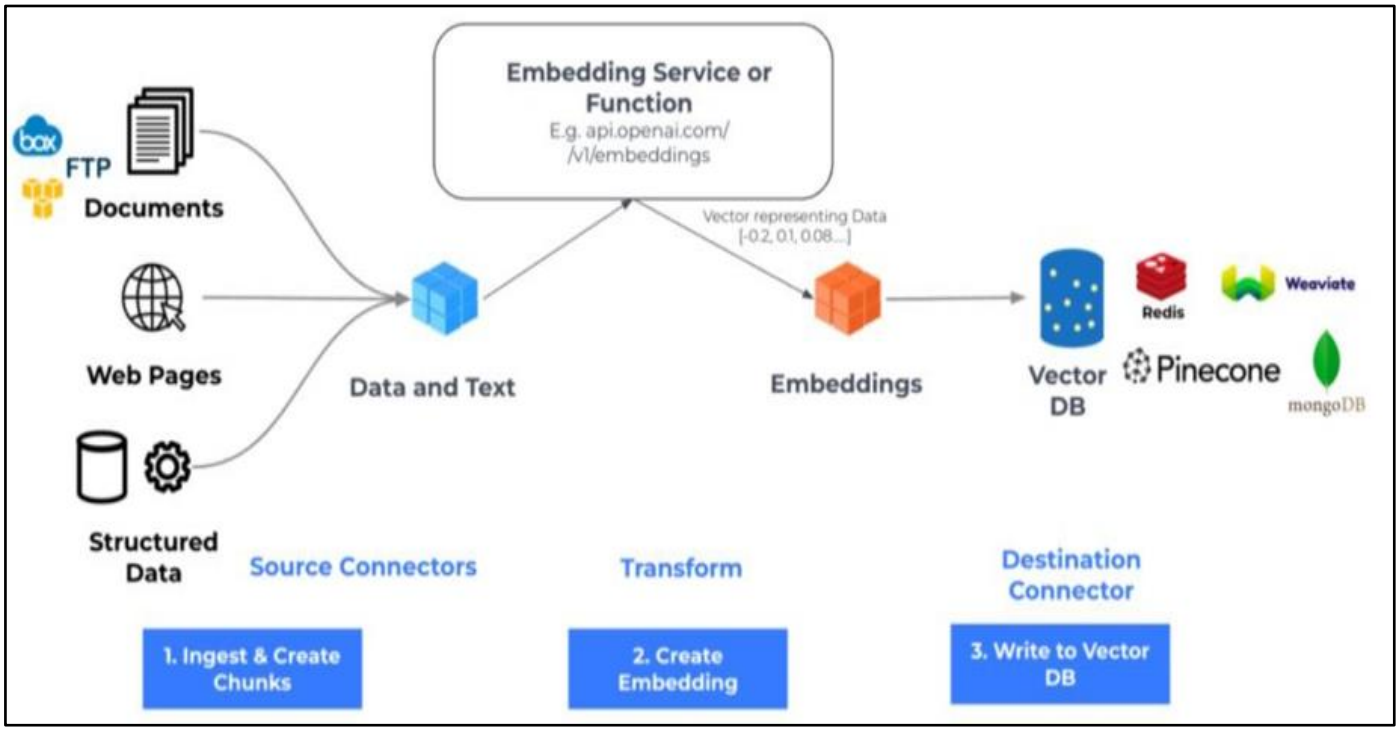

Fig 2 High-Level Architecture and Dataflow (Nexla., 2023).

➢ *Data Ingestion:*

The Connectors use the Confluence REST API to obtain pages and meeting notes and attachments from various team spaces. The system allows users to run the ingestion process either on a schedule or manually at any time to maintain current data (Nexla., 2023).

➢ *Preprocessing:*

The text cleaning process removes non-informative elements while standardizing formatting in extracted content. The system applies metadata tags including author information and date stamps and team affiliations and tags to each document for efficient retrieval and filtering purposes. The system removes unneeded content during this processing stage (Nexla., 2023).

➢ *Vectorization:*

The cleaned documents receive semantic vector representation through the application of high- quality embedding models. The vector database Pinecone, Chroma or FAISS stores these vectors for performing quick and precise similarity searches across extensive datasets (Nexla., 2023).

➢ *RAG Pipeline:*

The system retrieves relevant content chunks from the vector database after receiving a query. The LLM receives chunks of information through Retrieval-Augmented Generation processing to produce summaries and insights and recommendations that draw from the retrieved context (Zhang & Elhamod, 2025).

➢ *Visualization Layer:*

The dashboard displays the generated insights through an interactive interface. Stakeholders can analyze trends and compare team metrics through dynamic visualizations which include charts and timelines and heat maps and other interactive tools to access original content.

The modular pipeline maintains continuous refresh of Confluence information which undergoes contextual analysis before delivering visual outputs that support organizational decision-making (Zhang & Elhamod, 2025).

III. METHODOLOGY

The Multi-Team Confluence Insights Dashboard development depends on a systematic approach that guarantees precise data and stable system operation with complete transparency of generated outputs (eGain, n.d.). The methodology consists of four essential components which include data freshness and noise reduction and vector database selection and LLM knowledge reversibility and control and multi-model orchestration for reliability (Shah, 2024).

➢ *Data Freshness & Noise Reduction:*

RAG-based insights need fresh and pertinent data to operate dependably. This is achieved through:

- Scheduled Re-Indexing: Automated refresh cycles update the vector database at regular intervals, ensuring newly created Confluence content is discoverable and outdated entries are archived or removed.
- The system indexes content only from Confluence spaces that have been authorized and from authors or pages which satisfy quality standards to stop the spread of unimportant or low-quality information (Shah, 2024).
- The retrieval system filters documents based on attributes including date and project tags and workflow stage to retrieve only relevant and current materials (Shah, 2024).

➢ *Vector Database Selection:*

The selection of vector database determines both the operational speed and flexibility of the system. The choice process depends on the following factors.

- Scalability: The capacity to store and search millions of semantic vectors without performance degradation.
- Latency: Low response times to support interactive querying and real-time analytics.
- Integration: Seamless compatibility with RAG frameworks such as LangChain or Llama Index (Shah, 2024).
- Candidate Vector Databases: Pinecone provides complete management features along with high scalability and optimized performance for enterprise-level low-latency retrieval capabilities.

The open-source lightweight Chroma platform serves as a developer-friendly solution for on- premises and smaller deployments. Self-hosted FAISS solution provides high-performance capabilities that work well with customized retrieval pipelines (Shah, 2024).

➢ *LLM Knowledge Reversibility & Control:*

The output needs to be auditable and controlled while maintaining the impossibility of reversing LLMs to reveal their training data.

- RAG Constraining: Restricts the LLM to work only with document chunks retrieved from query intent to minimize the possibility of hallucination.
- Prompt Engineering: The LLM receives structured prompts which direct it to produce brief summaries with documented sources while maintaining its responses within predetermined content parameters.
- Retrieval Filters: Applying metadata-based restrictions ensures that responses are drawn only from trusted, current sources (Masoudifard, 2024).

➢ *Multi-Model Orchestration for Reliability:*

Using multiple specialized models enhances both the accuracy of results and their consistency and determinism.

- Retriever Model: Optimized for semantic recall (e.g., BAAI/bge-large, OpenAI text- embedding-ada-002).

- Summarizer Model: Fine-tuned LLM focused on generating concise, contextually accurate summaries without hallucinations.
- Validator Model: Cross-verifies factual accuracy by comparing generated outputs against retrieved source material.
- Charting Engine: Transforms structured insights into visual analytics using libraries such as Plotly, D3.js, or Apache ECharts.

The dashboard delivers timely accurate and transparent insights through this multi-layered methodology which enables decision-makers to trust the information they receive (Masoudifard, 2024).

IV. BENEFITS

The Multi-Team Confluence Insights Dashboard provides multiple benefits which overcome the standard search and reporting restrictions of Confluence.

- Cross-Team Knowledge Integration: The system unites documentation from various spaces to show organizational knowledge through a single interface.
- Real-Time Insights: The RAG system provides immediate access to summaries and analytics which updates automatically to avoid using outdated or incomplete data (EDB, 2024).
- Improved Transparency: All generated insights include source citations and metadata, enabling users to trace information back to the original document (EDB, 2024).
- Reduced Cognitive Load: Users receive synthesized dashboards and visual analytics instead of having to read through hundreds of Confluence pages.
- Enhanced Decision Velocity: Executives and project managers can use data to make quick decisions because they do not need to conduct manual search for extended periods (EDB, 2024).
- Customizable Analytics: Stakeholders can use dynamic charting to customize their views by team or project or timeline for deeper analysis.

V. CHALLENGES & CONSIDERATIONS

The proposed system provides substantial advantages, yet multiple challenges need resolution to achieve long-term success.

- Data Privacy & Compliance: The system needs to hide sensitive information or remove restricted content from its database to fulfill GDPR and HIPAA requirements (Arooj. (2025, 2025).
- Model Drift: The process of embedding models and summarization LLMs needs periodic retraining or updates to preserve accuracy and team terminology alignment.
- Metadata Consistency: The accuracy of retrieval decreases when Confluence metadata and tagging are inconsistent, so teams need to standardize their metadata.
- Dashboard Usability: Complex visualizations that are too detailed will confuse users, so dashboards need to strike a balance between depth and clarity.
- Cross-Team Terminology Alignment: Different teams use different terminology to describe similar concepts which need normalization during preprocessing or retrieval operations.
- System Scalability: The system needs to maintain low-latency performance while handling larger vector databases when integrating additional Confluence spaces (Zilliz, 2024).

VI. FUTURE SCOPE

RAG enhances the performance of LLMs by integrating external, up-to-date information, thus addressing some of the inherent limitations of static LLMs such as hallucinations and outdated knowledge (Huang & Huang, 2024).

- *Future Developments of RAG Systems in Dashboards Could Focus on Several Key Areas:*

- *Improved Data Integration and Accuracy:*

By using RAG, dashboards can dynamically incorporate external data sources to provide more accurate and reliable insights. This integration can help in generating contextually relevant insights by utilizing real-world data, thus supporting more informed decision-making within teams (Gao et al., 2023).

- *Mitigation of Hallucinations:*

One significant challenge for RAG and LLMs is dealing with hallucinations during text generation. Future research could focus on refining techniques to detect and correct such hallucinations, thereby enhancing the trustworthiness of generated insights (Zhang & Zhang, 2025).

- *Benchmarking and Robustness Improvements:*

Developing benchmarks like the Retrieval-Augmented Generation Benchmark (RGB) to evaluate LLMs' performance in aspects such as noise robustness and information integration can help identify and surmount current limitations. This enables better tuning of RAG for specific tasks and domain requirements (Chen et al., 2024).

- *Advanced Multi-Agent Systems / Agentic AI:*

AI agents are trending across industries as autonomous assistants capable of reasoning and initiating actions. These agentic systems are poised to become integral to enterprise workflows. Enterprises are developing strategic roadmaps to transition toward agentic AI, focusing on knowledge management, system integration, precision retrieval, and prompt engineering (Hummel, 2025).

- *Multi-Agent & Adaptive RAG Architectures:*

The multi-agent reconfigurable RAG system enables real-time dynamic orchestration to manage quality, cost, and latency according to real- world SLAs. Emerging agentic RAG models are transforming passive retrieval

systems into proactive reasoning agents capable of multi-step decompositions, hybrid retrieval, and hybrid modality processing (Iannelli, 2025).

VII. CONCLUSION

In conclusion, leveraging RAG and LLMs to develop a Multi-Team Confluence Insights Dashboard presents notable advantages and challenges. The integration of RAG methods allows dynamic incorporation of up-to-date external information, thereby improving the integrity and consistency of LLM responses by mitigating their inherent static limitations (Huang & Huang, 2024). This combination enhances the dashboard's capability to deliver precise, domain-specific insights by balancing the generative prowess of LLMs with the real-time retrieval strengths of RAG. The deployment of RAG ensures that the generated content is factual and up-to-date, overcoming the common issue of generating plausible but incorrect responses (Chen et al., 2024; Zhu et al., 2025). A key innovation in this approach is the use of scalable and pluggable virtual tokens within RAG frameworks, which preserves the general generation capabilities of LLMs while allowing for fine-tuning specific to retrieval contexts. This ensures that the LLMs maintain their core functionalities without necessitating parameter modifications, which can affect their broader utility (Zhu et al., 2025). Moreover, focusing retrieval efforts through the use of reflective tags and modular frameworks can reduce noise, thereby optimizing retrieval for high- quality and reliable results (Jin et al., 2024; Yao & Fujita, 2024). Though, challenges remain, particularly in avoiding retrieval's introduction of irrelevant or low-quality data, which can degrade the quality of output. Continued advancements and research are needed to refine the interaction between LLMs and RAG frameworks to fully leverage their combined potential, particularly in multi-team environments where diverse information needs and data sources exist (Jin et al., 2024; Zhang & Zhang, 2025).

REFERENCES

[1]. eGain. (n.d.). Confluence knowledge management: 9 signs you've outgrown it. eGain. Retrieved August 25, 2025, from https://www.egain.com/confluence-knowledge-management-limitations/.

[2]. Zhu, Y., Huang, Z., Dou, Z., & Wen, J.-R. (2024). One token can help! Learning scalable and pluggable virtual tokens for retrieval-augmented large language models. arXiv. https://doi.org/10.48550/arXiv.2405.19670.

[3]. Tran, H. (2023, September 20). Which is better, retrieval augmentation (RAG) or fine- tuning? Both. Snorkel AI. https://snorkel.ai/blog/which-is-better-retrieval-augmentation- rag-or-fine-tuning-both/.

[4]. Zhang, R., & Elhamod, M. (2025). Data-to-dashboard: Multi-agent LLM framework for insightful visualization in enterprise analytics. In Proceedings of the 2nd Workshop on Agentic AI for Enterprise (ACM SIGKDD). https://arxiv.org/pdf/2505.23695.

[5]. Nexla. (2023). Retrieval-augmented generation (RAG) tutorial & best practices. Nexla. https://nexla.com/ai-infrastructure/retrieval-augmented-generation/.

[6]. Shah, M., Muralidhar, R., & Fort, N. (2024, October). Retrieval augmented generation options and architectures on AWS (AWS Prescriptive Guidance). Amazon Web Services. https://docs.aws.amazon.com/pdfs/prescriptive-guidance/latest/retrieval-augmented- generation-options/retrieval-augmented-generation-options.pdf.

[7]. Masoudifard, A., Sorond, M. M., Madadi, M., Sabokrou, M., & Habibi, E. (2024). Leveraging Graph-RAG and prompt engineering to enhance LLM-based automated requirement traceability and compliance checks. arXiv. https://arxiv.org/pdf/2412.08593.

[8]. EDB Team. (2024, October 8). What is retrieval-augmented generation (RAG)? How to build a RAG app with pgvector. Enterprise DB. https://www.enterprisedb.com/blog/rag- app-postgres-and-pgvector?lang=en.

[9]. Zilliz. (2024, May 17). Scaling vector databases to meet enterprise demands. Medium. https://medium.com/@zilliz_learn/scaling-vector-databases-to-meet-enterprise-demands-24416ae803e1.

[10]. Arooj. (2025, April 8). Optimizing RAG for sensitive data & privacy compliance. Chitika. https://www.chitika.com/optimizing-rag-sensitive-data-privacy/.

[11]. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. https://doi.org/10.48550/arxiv.2312.10997.

[12]. Chen, J., Lin, H., Han, X., & Sun, L. (2024). Benchmarking Large Language Models in Retrieval-Augmented Generation. Proceedings of the AAAI Conference on Artificial Intelligence, 38(16), 17754–17762. https://doi.org/10.1609/aaai.v38i16.29728.

[13]. Zhang, W., & Zhang, J. (2025). Hallucination Mitigation for Retrieval-Augmented Large Language Models: A Review. Mathematics, 13(5), 856. https://doi.org/10.3390/math13050856.

[14]. Iannelli, M., Kuchipudi, S., & Dvorak, V. (2025). SLA management in reconfigurable multi-agent RAG: A systems approach to question answering (Version 2) [Preprint]. arXiv. https://arxiv.org/abs/2412.06832.

[15]. Hummel, M. (2025, August 25). How enterprises can transition their knowledge and systems for Agentic AI. TechRadar Pro. https://www.techradar.com/pro/how-enterprises-can-transition-their-knowledge-and-systems-for-agentic-ai.

[16]. Huang, Y., & Huang, J. (2024). A Survey on Retrieval-Augmented Text Generation for Large Language Models. https://doi.org/10.48550/arxiv.2404.10981.

[17]. Jin, J., Zhu, Y., Dong, G., Zhang, Y., Yang, X., Zhang, C., Zhao, T., Yang, Z., Dou, Z., & Wen, J.-R. (2024). Flash RAG: A Modular Toolkit for Efficient Retrieval-Augmented Generation Research. https://doi.org/10.48550/arxiv.2405.13576.

[18]. Yao, C., & Fujita, S. (2024). Adaptive Control of Retrieval-Augmented Generation for Large Language Models Through Reflective Tags. Electronics, 13(23), 4643. https://doi.org/10.3390/electronics13234643.

[19]. Zhu, Y., Wen, J.-R., Huang, Z., & Dou, Z. (2025). One Token Can Help! Learning Scalable and Pluggable Virtual Tokens for Retrieval-Augmented Large Language Models. Proceedings of the AAAI Conference on Artificial Intelligence, 39(24), 26166– 26174. https://doi.org/10.1609/aaai.v39i24.34813.