Enterprise RAG Architecture

Client Type: Anglo Eastern

Service Provider: Microland

Version Information

Version number	:	0.1
Publication date	:	28 Jan 2026
Prepared by	:	Mayank Chugh
Reviewed by	:	
Approved by	:	

Revision History

Version	Date	Name	List of Modifications
0.1	28 Jan 26	Mayank Chugh	Draft

TABLE OF CONTENTS

1 Introduction

This document outlines the architecture of the Anglo-Eastern Knowledge Management System, emphasizing secure integration of diverse data sources, robust query processing, and high-quality search results.

2 System Overview

The system comprises modules for user authentication, query processing, semantic search functionality, and data storage architecture. It efficiently ingests PDFs, extracts metadata, and ranks responses based on user queries, ensuring a seamless user experience.

3 Components

3.1 User Authentication

- **Authentication Mechanism:** Utilises Active Directory (AD) to authenticate users and capture their Department ID.
- **Session Management:** Tracks user sessions and provides access based on authentication status.

3.2 Query Processing

- **Input Validation:** Ensures that user queries are well-formed and valid.
- **Non-Ethical Search Filter:** Implements mechanisms to reject inappropriate queries.
- **Memory Management:** Retains context from previous interactions to enhance subsequent queries.

3.3 Semantic Search

- **Search Engine:** Uses pre-trained LLM models to convert user queries into semantic vectors.
- **Vector Search:** Connects to a VectorDB to retrieve relevant results based on meaning.

3.4 Data Storage and Management

- **SQL Server:** Stores structured data for HR, Finance, and the overall knowledge base.
- **Blob Storage:** Stores large files such as PDFs, images, and videos, with metadata linked to SQL Server.

3.5 Data Ingestion Pipeline

- **PDF Parsing:** Employs tools for extracting text and data from PDF files.
- **Chunking:** Breaks documents into manageable pieces for processing.
- **Metadata Tagging:** Adds tags and attributes for efficient retrieval during searches.

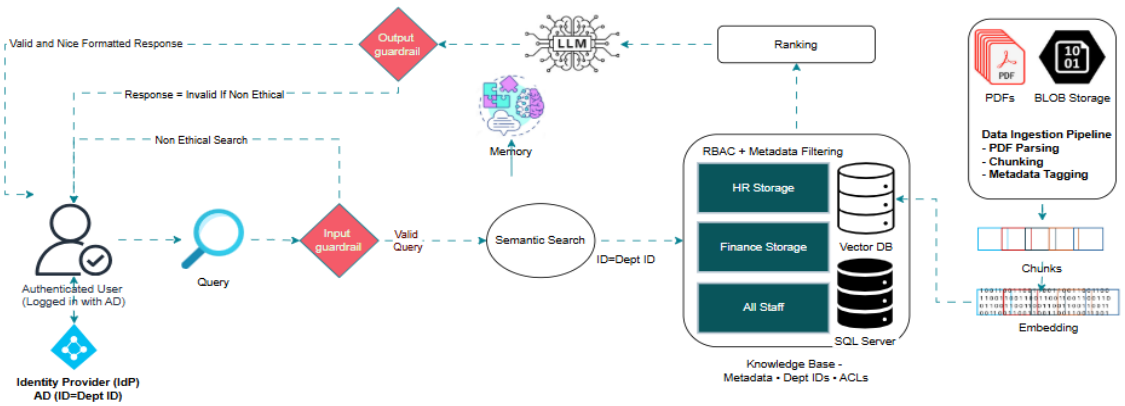
3.6 Output Handling

- **Response Formatting:** Delivers consistent, well-structured responses to user queries.
- **Ranking Mechanism:** Scores results based on relevance and user requirements.

4 RAG Architecture & Flow

1. **User submits a query:** Authenticated user submits a query through the interface.
2. **Query validation:** Systems validate the query format and check against ethical standards.
3. **Semantic search initiated:** The query is processed to generate semantic vectors.
4. **Data accessed:** Relevant data is retrieved from SQL Server and VectorDB.
5. **Results ranked:** Results are ranked based on relevance.
6. **User output:** The final output is formatted and returned to the user.

The data flow in the system is represented in the diagram below. This outlines the interactions between the user, the various processing components, and the data sources.

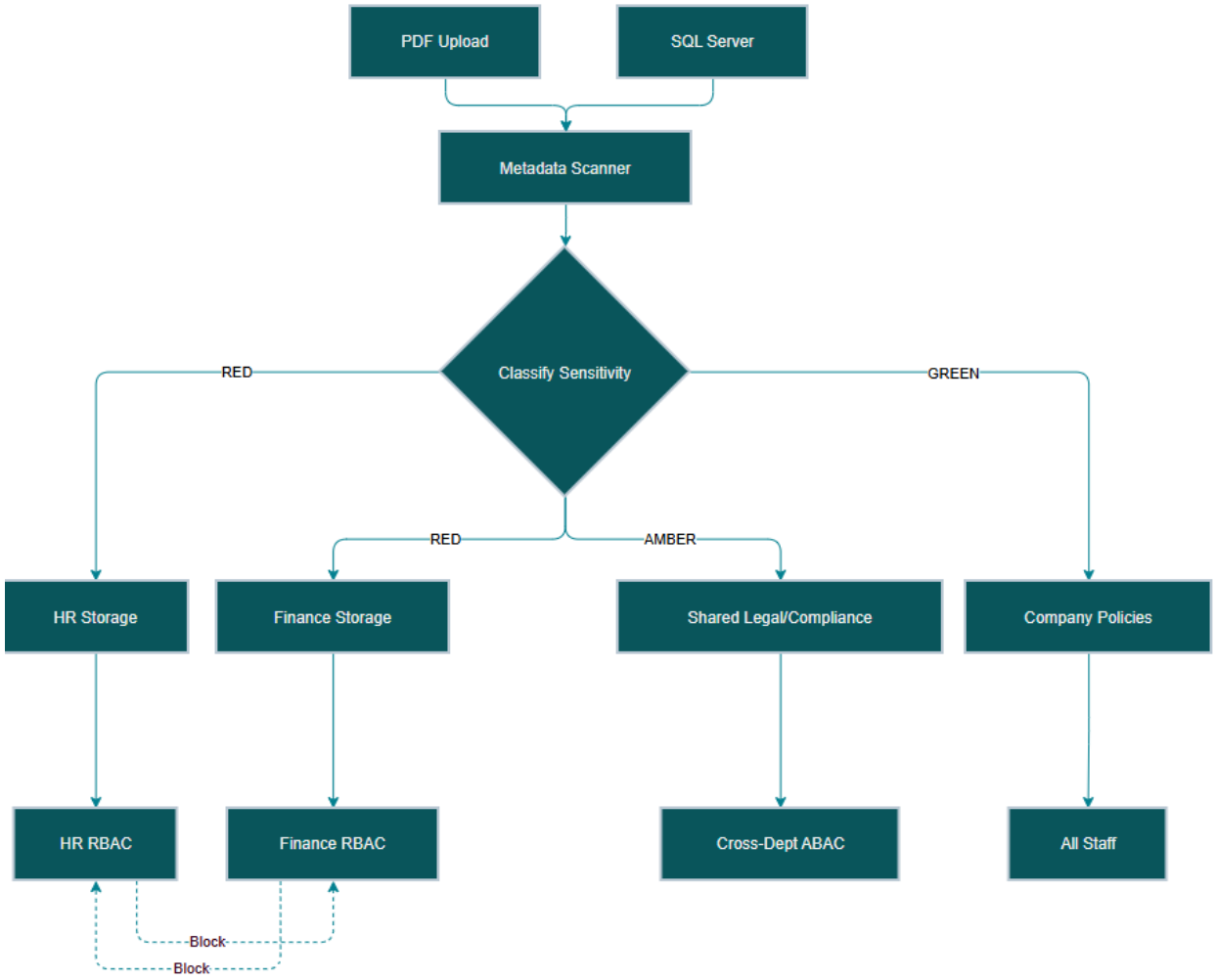


5 Security Considerations

- **Data Protection:** Ensure encryption for both data at rest and in transit.

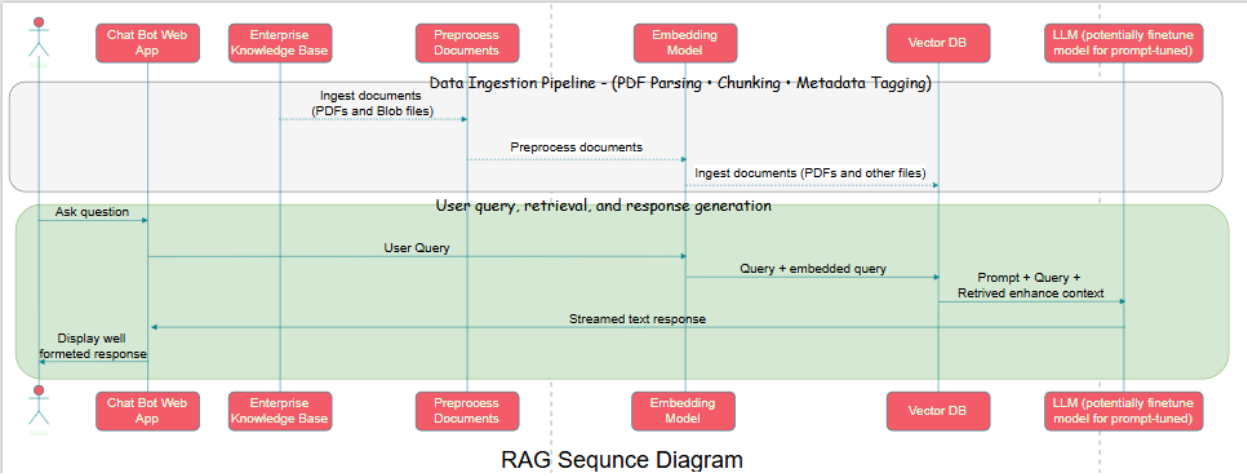
- **Access Control:** Implement strict RBAC to limit access based on user roles.
- **Compliance:** Adhere to data protection regulations (e.g., GDPR).

6 Sensitivity Classification Diagram



- **PDF Upload:** Ingests files into the system.
- **Metadata Scanner:** Extracts and classifies document metadata.
- **Sensitivity Check:** Documents are categorized based on sensitivity levels (RED, AMBER, GREEN).
    - Sensitive documents are directed to specific storage based on classification (HR, Finance, Shared Legal).

7 RAG Sequence Diagram



RAG Sequnce Diagram

- **User Query Submission:** Initiated by the user through the Chat Bot Web App.
- **Data Processing:** The system engages the embedding model and large language model to formulate an appropriate response.
- **Response Delivery:** Formats and presents the final response to the user in the chat interface.

8 Monitoring and Logging

- **Activity Logging:** Capture user interactions, successful queries, and errors.
- **Performance Monitoring:** Use monitoring tools to track system performance and uptime.

9 Testing and Validation

- **Unit Testing:** Test individual components for functionality.
- **Integration Testing:** Ensure seamless interaction between the different components.
- **User Acceptance Testing (UAT):** Validate the system against user requirements.

10 Future Enhancements

- **Finetuning and Inference Pre-Training LLM:** Explore incorporating more sophisticated LLM models for better semantic understanding.

- **Scalability Enhancements:** Assess storage and retrieval strategies as data volume grows.
- **User Feedback Integration:** Collect user feedback regularly to enhance features and usability.

11 Assumptions

- The users of the system have basic technical proficiency required to engage with the Chat Bot Web App.
- All data ingested into the system is approved by the relevant authorities for use.
- The system requires a stable internet connection for full functionality, particularly for real-time query processing and response generation.
- The organizational infrastructure, including hardware and software components, will support the performance requirements of the architecture.
- User roles and access permissions are predefined and will be managed as per the organization's

12 Risks & Mitigations

Risk	Description	Mitigation
Data Breach	Unauthorized access to sensitive data	Implement strong encryption, RBAC, and regular security audits.
System Downtime	Potential for outages that disrupt user access	Utilize redundancy strategies and implement a robust disaster recovery plan.
Inaccurate Data Retrieval	ML models may return irrelevant or incorrect results	Regularly refine models with training data and feedback loops for continuous improvement.
Compliance Violations	Risks of non-compliance with regulations like GDPR	Regular compliance audits and updates to ensure adherence to data protection laws.
User Inexperience	Users may struggle to interact effectively with the system	Provide comprehensive training and detailed user guides.
Scalability Issues	The system may not handle increased data loads effectively	Design the system with scalability in mind and conduct load testing.

- policies.

13 Conclusion

The architecture for the Anglo-Eastern Knowledge Management System is structured to ensure robust data handling, effective user interaction, and a secure environment. Regular updates and enhancements, based on performance monitoring and user discussion feedback, are essential for maintaining relevance and user satisfaction.