**Part 1**

**Event Manager Architecture**



**Part 2**

**Overview of Retrieval-Augmented Generation(RAG)**

**Stage 1 Document Preparation & Indexing**

* Load Docs
* Chunk Docs(LLMs have limited context window)
* Embed Chunks(word-embeddings)
* Load into VectorDB(VectorStoreIndex).

**Stage 2 Query Processing & Docs Retrieval**

* Input Query
* Retrieve Query-relevant Docs from VectorDB
* Prompt Template with Context.

**Stage 3 Response Generation**

* Prompt + Context + Query
* LLM Inference Generate Response

**Evolution of RAG Paradigms**



**Most Notable Limitations**

* Contextual integration: Limited contextual awareness lead to inconsistent results
* Multi-step reasoning: Iterative or multi-hop reasoning across multiple steps
* Scalability and latency issues: Large volume data, increasing computation and response time

**Graph RAG**

A graph-based approach that enables global sensemaking over the entirety of large text corpus. GraphRAG builds upon prior work on advanced RAG strategies. GraphRAG leverages summaries over large sections of the data source as a form of self-memory, which are later used to answer queries.These summaries are generated in parallel and iteratively aggregated into global summaries.Uses hierarchical indexing to create summaries.

Traditional RAG(vector RAG) only retrieves semantically similar chunks, but fails on global/sensemaking questions and no understanding of entity relationships or structure for accurate results on large corpus which solves using GraphRAG.

**Graph RAG Pipeline**



**Graph RAG has some limitations:**

* Limited Scalability: The reliance on graph structures can restrict scalability, especially with extensive data sources.
* Data Dependency: High-quality graph data is essential for meaningful outputs, limiting its applicability in unstructured or poorly annotated datasets.
* Complexity of Integration: Integrating graph data with unstructured retrieval systems increases design and implementation complexity.

**Agentic RAG**

Introducing autonomous agents capable of dynamic decision-making and workflow optimization. Unlike static systems, Agentic RAG employs iterative refinement and adaptive retrieval strategies to address complex, real-time, and multi-domain queries.

**Key characteristics**

Autonomous Decision-Making: Agents independently evaluate and manage retrieval strategies based on query complexity.

Iterative Refinement: Incorporates feedback loops to improve retrieval accuracy and response relevance.

Workflow Optimization: Dynamically orchestrates tasks, enabling efficiency in real-time applications.

**Challenges**

Coordination Complexity: Managing interactions between agents requires sophisticated orchestration mechanisms.

Computational Overhead: The use of multiple agents increases resource requirements for complex workflows.

Scalability Limitations: While scalable, the dynamic nature of the system can strain computational resources for high query volumes.

**Core Principles and Background of Agentic Intelligence**

* LLM: Defined Role and Task
* Memory (Short-Term and Long-Term)
* Planning and Reflection

1. Reflection is a foundational design pattern in agentic workflows, enabling agents to iteratively evaluate and refine their outputs
2. Planning is a key design pattern in agentic workflows that enables agents to autonomously decompose complex tasks into smaller, manageable subtasks.

* Tools Vector Search, Web Search, APIs, etc.):

1. Tool Use enables agents to extend their capabilities by interacting with external tools, APIs, or computational resources

* Multi-agent collaboration is a key design pattern in agentic workflows that enables task specialization and parallel processing.

**Taxonomy of Agentic RAG Systems**

1. Single-Agent Agentic RAG: Router

* Serves as a centralized decision-making system where a single agent manages the retrieval, routing, and integration of information

1. Multi-Agent RAG

* Represents a modular and scalable evolution of single-agent architectures, designed to handle complex workflows and diverse query types by leveraging multiple specialized agents

1. Hierarchical Agentic RAG

* Systems employ a structured, multi-tiered approach to information retrieval and processing, enhancing both efficiency and strategic decision-making

1. Corrective RAG

* Introduces mechanisms to self-correct retrieval results, enhancing document utilization and improving response generation quality

1. Adaptive Retrieval-Augmented Generation (Adaptive RAG)

* Enhances the flexibility and efficiency of LLMs by dynamically adjusting query handling strategies based on the complexity of the incoming query. Unlike static retrieval workflows, Adaptive RAGemploys a classifier to assess query complexity and determine the most appropriate approach, ranging from single-step retrieval to multi-step reasoning, or even bypassing retrieval altogether for straightforward queries

1. Agent-G: Agentic Framework for Graph RAG Agent-G

* Introduces a novel agentic architecture that integrates graph knowledge bases with unstructured document retrieval. By combining structured and unstructured data sources, this framework enhances retrieval-augmented generation systems with improved reasoning and retrieval accuracy. It employs modular retriever banks, dynamic agent interaction, and feedback loops to ensure high-quality outputs

1. GeAR: Graph-Enhanced Agent for Retrieval-Augmented Generation GeAR

* Introduces an agentic framework that enhances traditional Retrieval-Augmented Generation systems by incorporating graph-based retrieval mechanisms. By leveraging graph expansion techniques and an agent-based architecture, GeAR addresses challenges in multi-hop retrieval scenarios, improving the system’s ability to handle complex queries

1. Agentic Document Workflows (ADW)

* Extend traditional Retrieval-Augmented Generation (RAG) paradigms by enabling end-to-end knowledge work automation. These workflows orchestrate complex document-centric processes, integrating document parsing, retrieval, reasoning, and structured outputs with intelligent agents. ADW systems address limitations of Intelligent Document Processing (IDP) and RAG by maintaining state, coordinating multi-step workflows, and applying domain-specific logic to documents.

**Conclusion**

The evolution of RAG systems from single-agent routers to complex multi-agent and graph-enhanced frameworks reflects the growing need for more intelligent and adaptive information processing. While single-agent RAG offers simplicity, multi-agent and hierarchical approaches provide better modularity, scalability, and strategic coordination. Adaptive RAG introduces flexibility by tailoring retrieval strategies based on query complexity, improving both efficiency and response quality. Corrective RAG enhances factual accuracy by introducing self-correction mechanisms. Graph-based agentic systems like Agent-G and GeAR significantly boost reasoning capabilities by combining structured knowledge graphs with unstructured data retrieval. Overall, these advancements pave the way for more robust, domain-aware, and automated knowledge workflows.