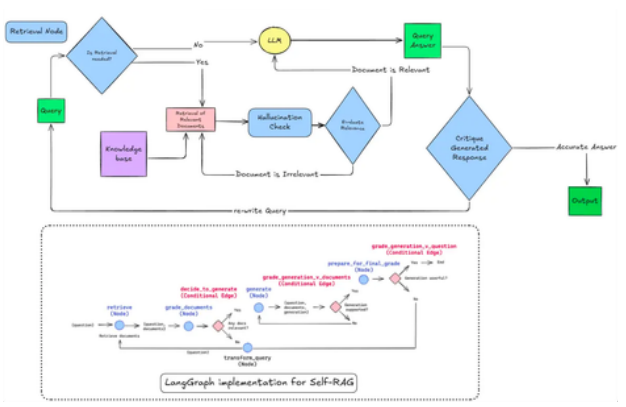
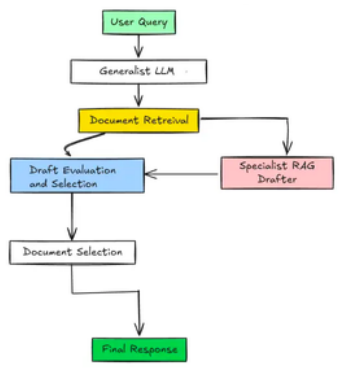
7 Agentic RAG System Architectures to Build AI Agents

Analytics Vidhya

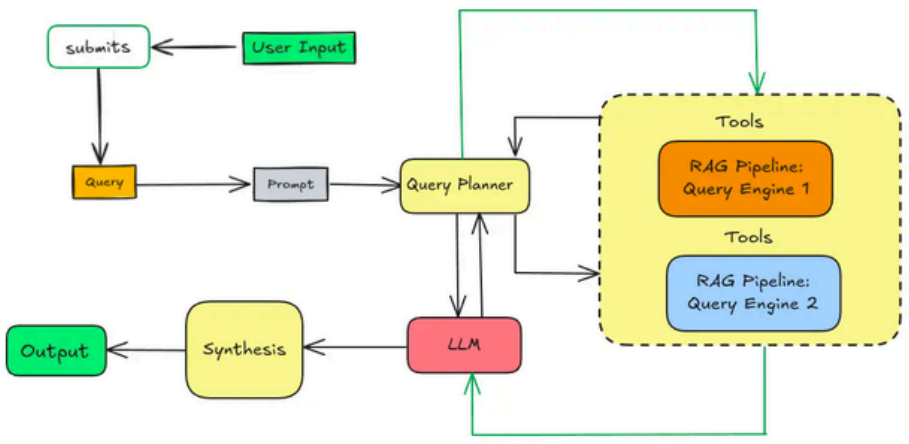
Self-reflective RAG



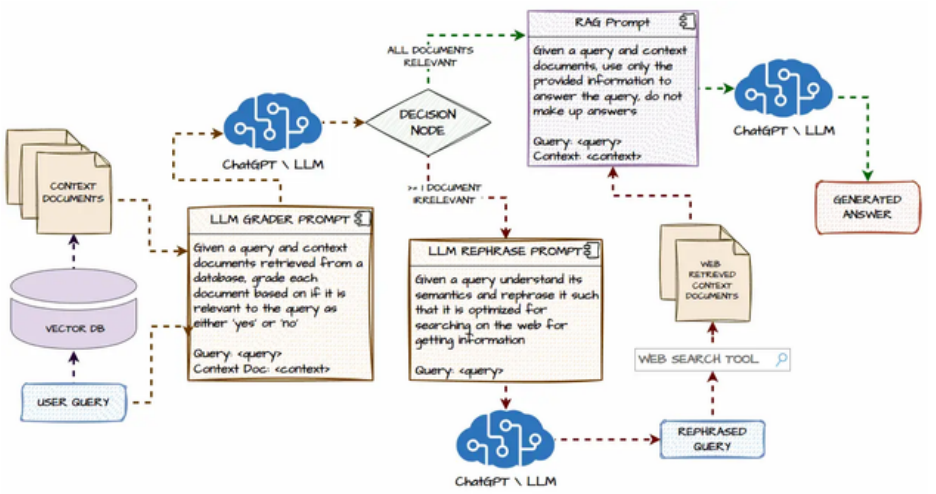
Speculative RAG



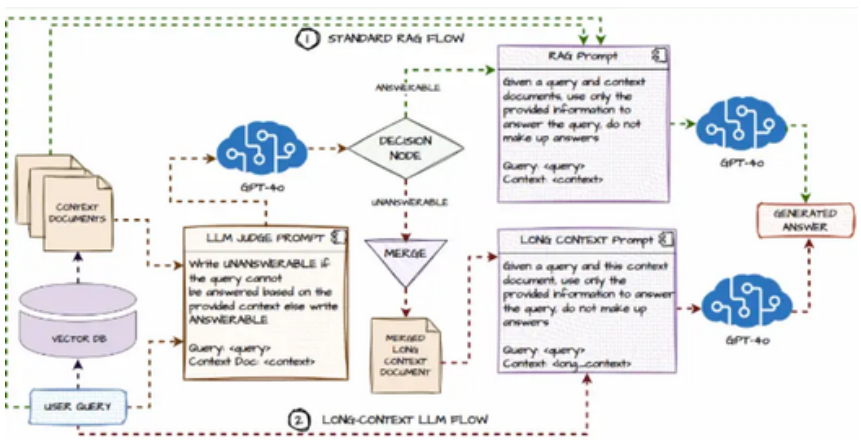
Query Planning Agentic RAG



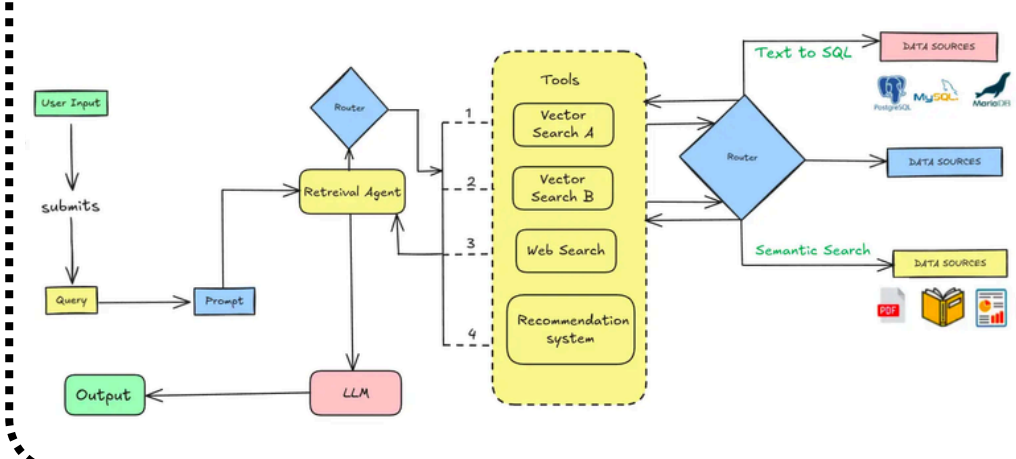
Agentic Corrective RAG



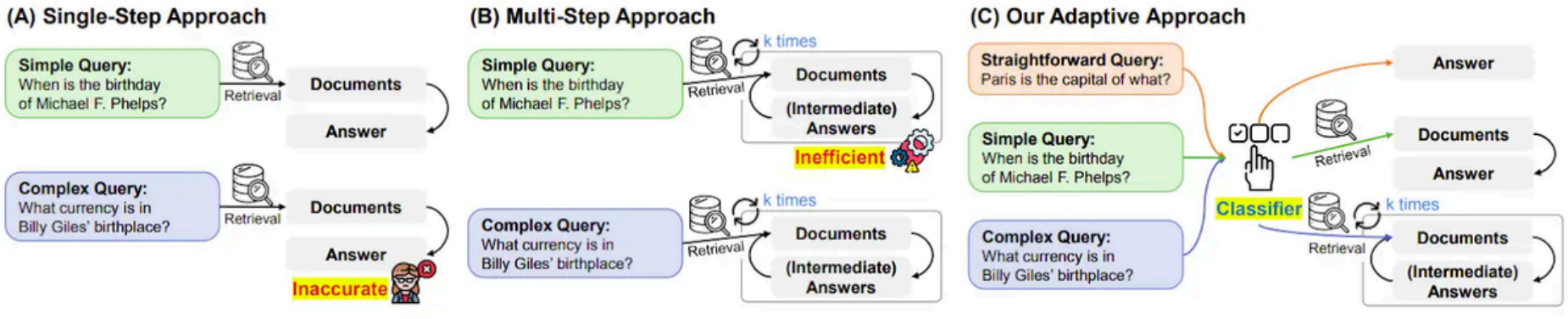
Self Route Agentic RAG



Agentic RAG Routers



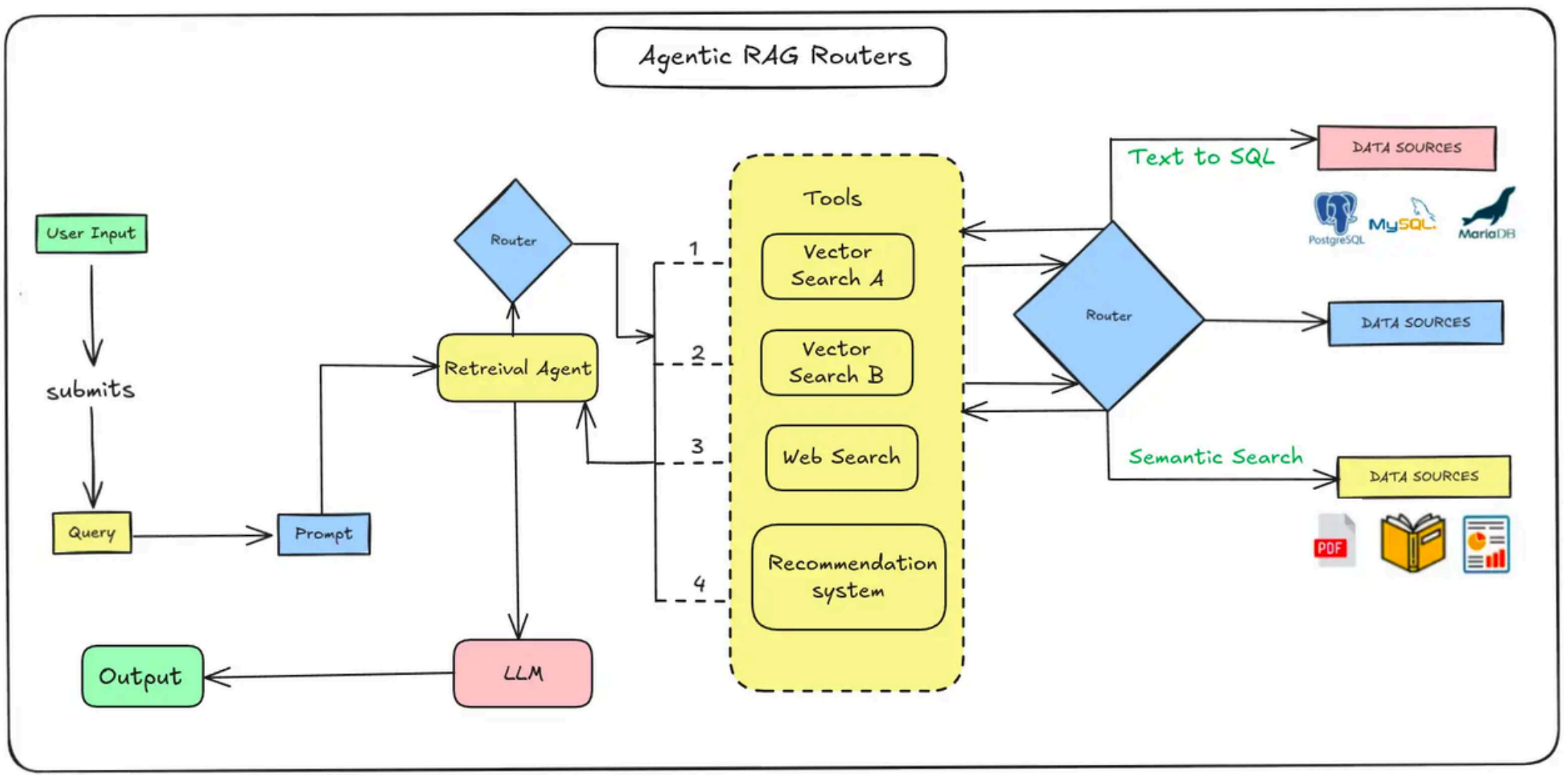
Adaptive RAG



Agentic RAG Routers

- Agentic RAG Routers are systems designed to dynamically route user queries to appropriate tools or data sources, enhancing the capabilities of Large Language Models (LLMs). The primary purpose of such routers is to combine retrieval mechanisms with the generative strengths of LLMs to deliver accurate and contextually rich responses.

- This approach bridges the gap between the static knowledge of LLMs (trained on pre-existing data) and the need for dynamic knowledge retrieval from live or domain-specific data sources. By combining retrieval and generation, Agentic RAG Routers enable applications such as:
  - Question answering
  - Data analysis
  - Real-time information retrieval
  - Recommendation generation



The architecture shown in the diagram provides a detailed visualization of how Agentic RAG Routers operate. Let’s break down the components and flow:

User Input and Query Processing

- User Input: A user submits a query, which is the entry point for the system. This could be a question, a command, or a request for specific data.

- Query: The user input is parsed and formatted into a query, which the system can interpret.

Retrieval Agent

The Retrieval Agent serves as the core processing unit. It acts as a coordinator, deciding how to handle the query. It evaluates:

- The intent of the query.
- The type of information required (structured, unstructured, real-time, recommendations).

Router

A Router determines the appropriate tool(s) to handle the query:

- **Vector Search**: Retrieves relevant documents or data using semantic embeddings.
- **Web Search**: Accesses live information from the internet.
- **Recommendation System**: Suggests content or results based on prior user interactions or contextual relevance.
- **Text-to-SQL**: Converts natural language queries into SQL commands for accessing structured databases.

Tools

The tools listed here are modular and specialized:

- **Vector Search A & B**: Designed to search semantic embeddings for matching content in vectorized forms, ideal for unstructured data like documents, PDFs, or books.
- **Web Search**: Accesses external, real-time web data.
- Recommendation System: Leverages AI models to provide user-specific suggestions.

Data Sources

The system connects to diverse data sources:

- **Structured Databases**: For well-organized information (e.g., SQL-based systems).
- **Unstructured Sources**: PDFs, books, research papers, etc.
- **External Repositories**: For semantic search, recommendations, and real-time web queries.

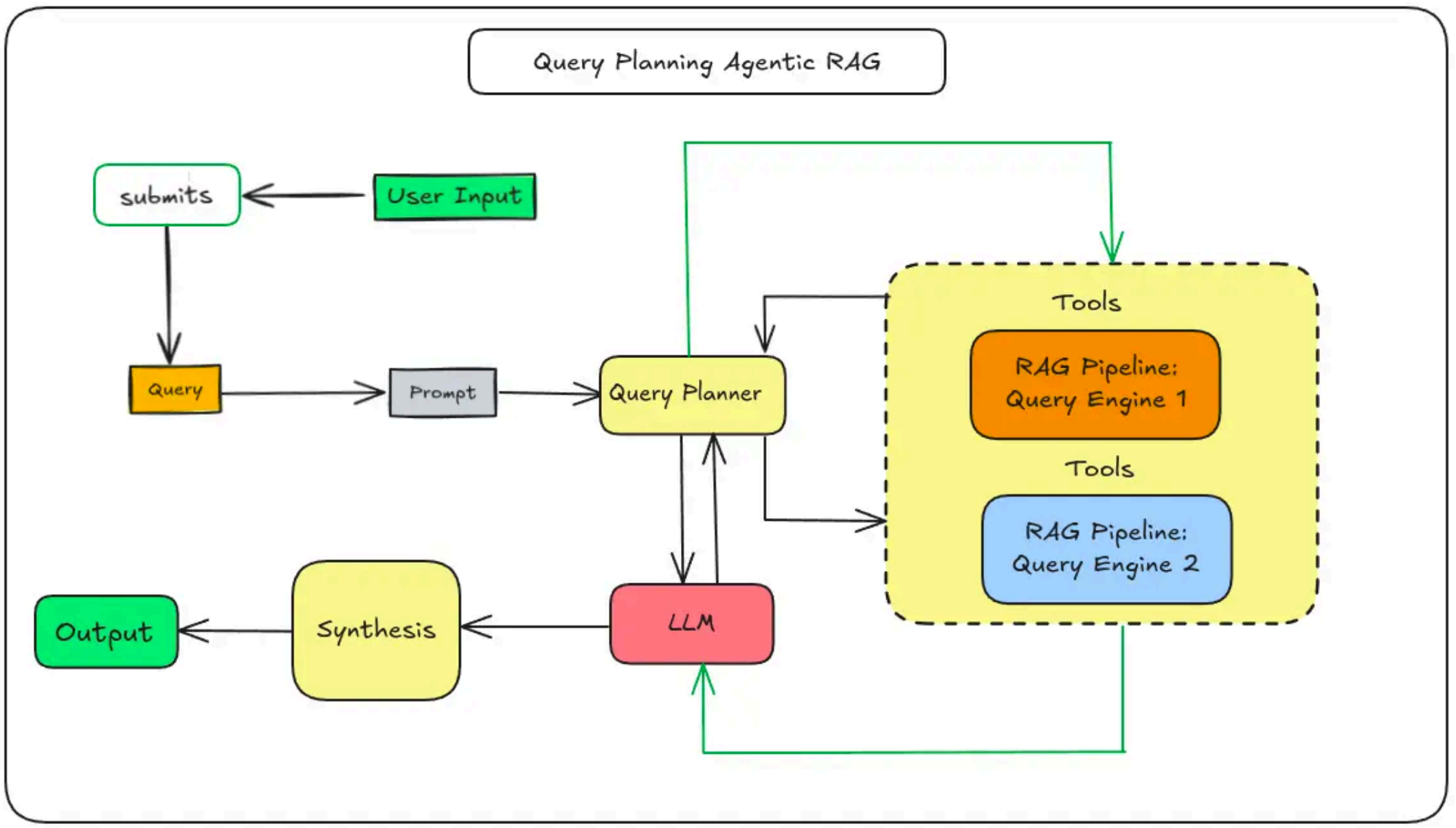
LLM Integration: Once data is retrieved, it is fed into the LLM:

- The LLM synthesizes the retrieved information with its generative capabilities to create a coherent, human-readable response.

Output: The final response is sent back to the user in a clear and actionable format.

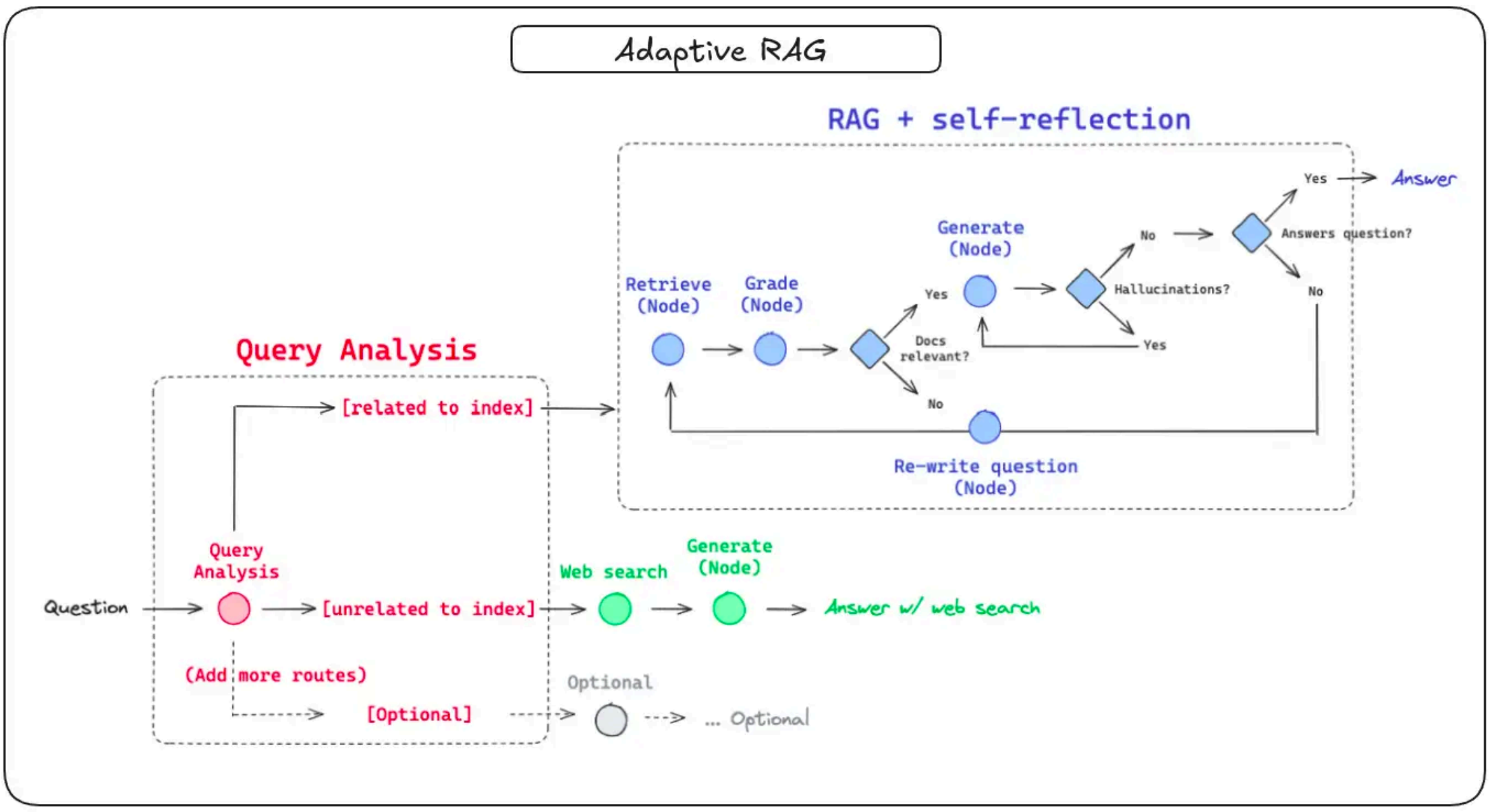
Query Planning Agentic RAG

- Query Planning Agentic RAG (Retrieval-Augmented Generation) is a methodology designed to handle complex queries efficiently by leveraging multiple parallelizable subqueries across diverse data sources. This approach combines intelligent query division, distributed processing, and response synthesis to deliver accurate and comprehensive results.



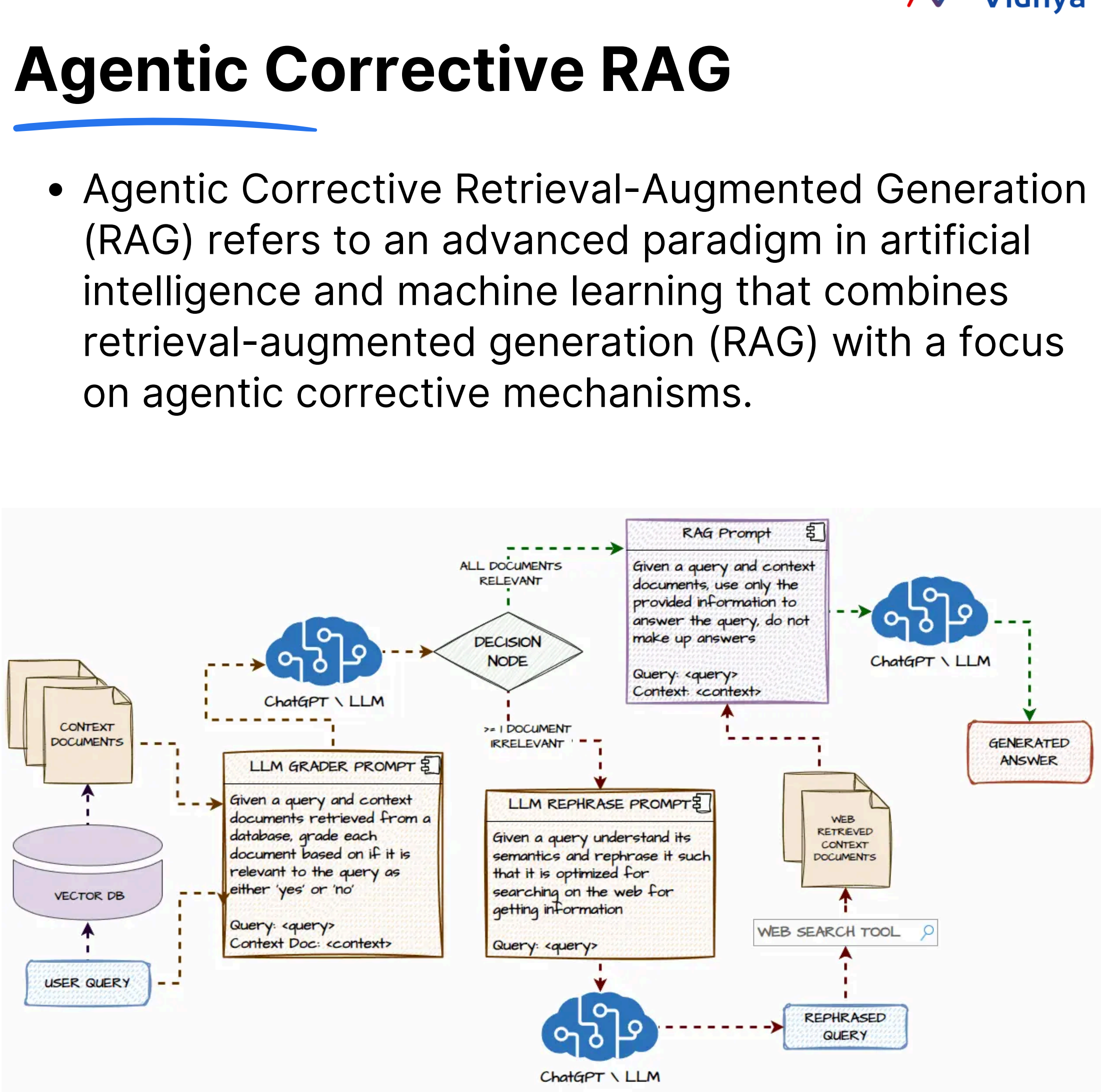
Adaptive RAG

- Adaptive Retrieval-Augmented Generation (Adaptive RAG) is a method that enhances the flexibility and efficiency of large language models (LLMs) by tailoring the query handling strategy to the complexity of the incoming query.



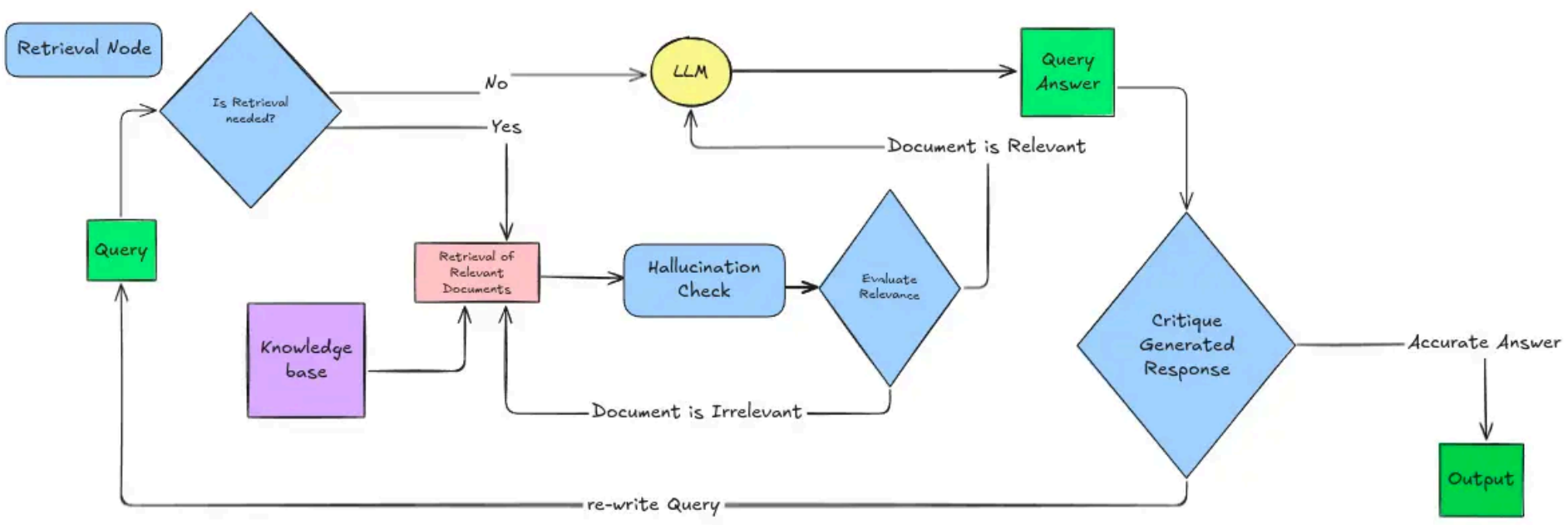
Agentic Corrective RAG

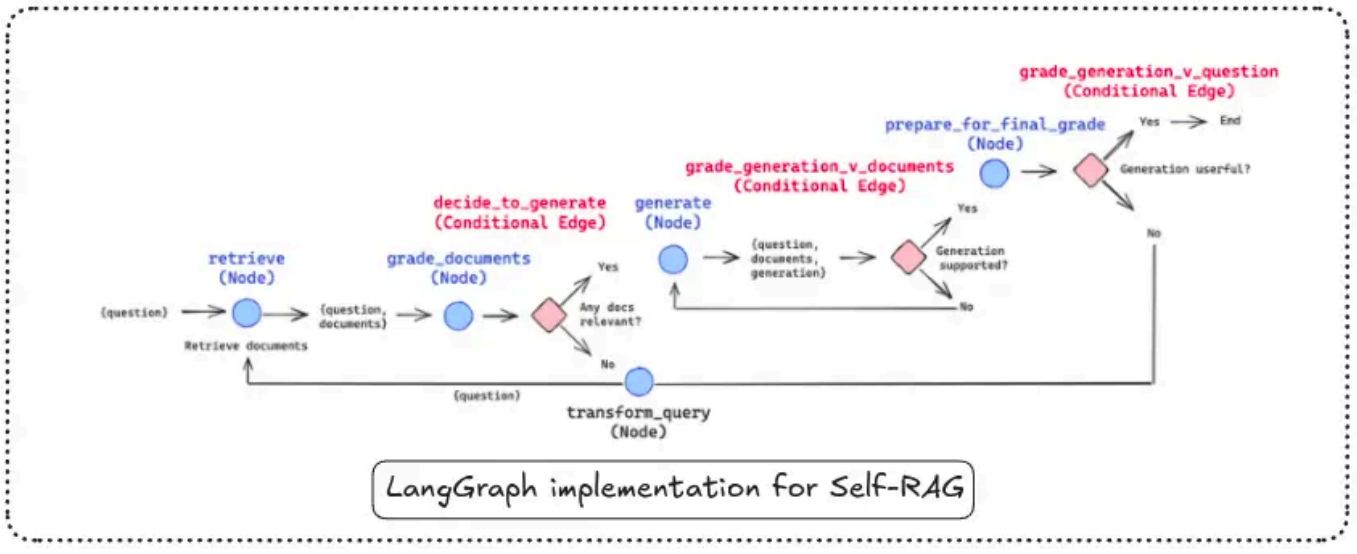
- Agentic Corrective Retrieval-Augmented Generation (RAG) refers to an advanced paradigm in artificial intelligence and machine learning that combines retrieval-augmented generation (RAG) with a focus on agentic corrective mechanisms.



Self-Reflective RAG

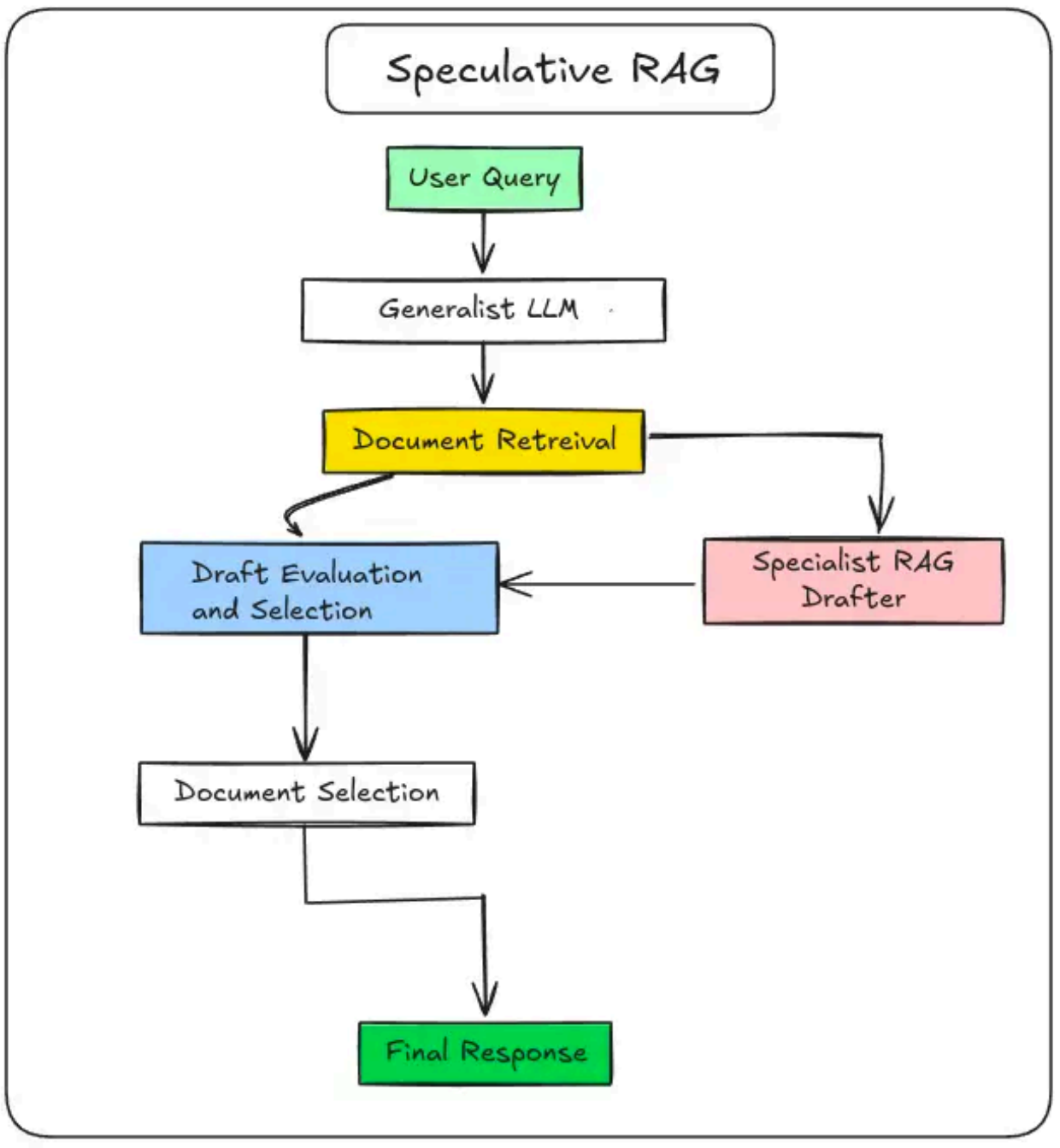
- Self-reflective RAG (Retrieval-Augmented Generation) is an advanced approach in natural language processing (NLP) that combines the capabilities of retrieval-based methods with generative models while adding an additional layer of self-reflection and logical reasoning. For instance, self-reflective RAG helps in retrieval, re-writing questions, discarding irrelevant or hallucinated documents and re-try retrieval. In short, it was introduced to capture the idea of using an LLM to self-correct poor-quality retrieval and/or generations.





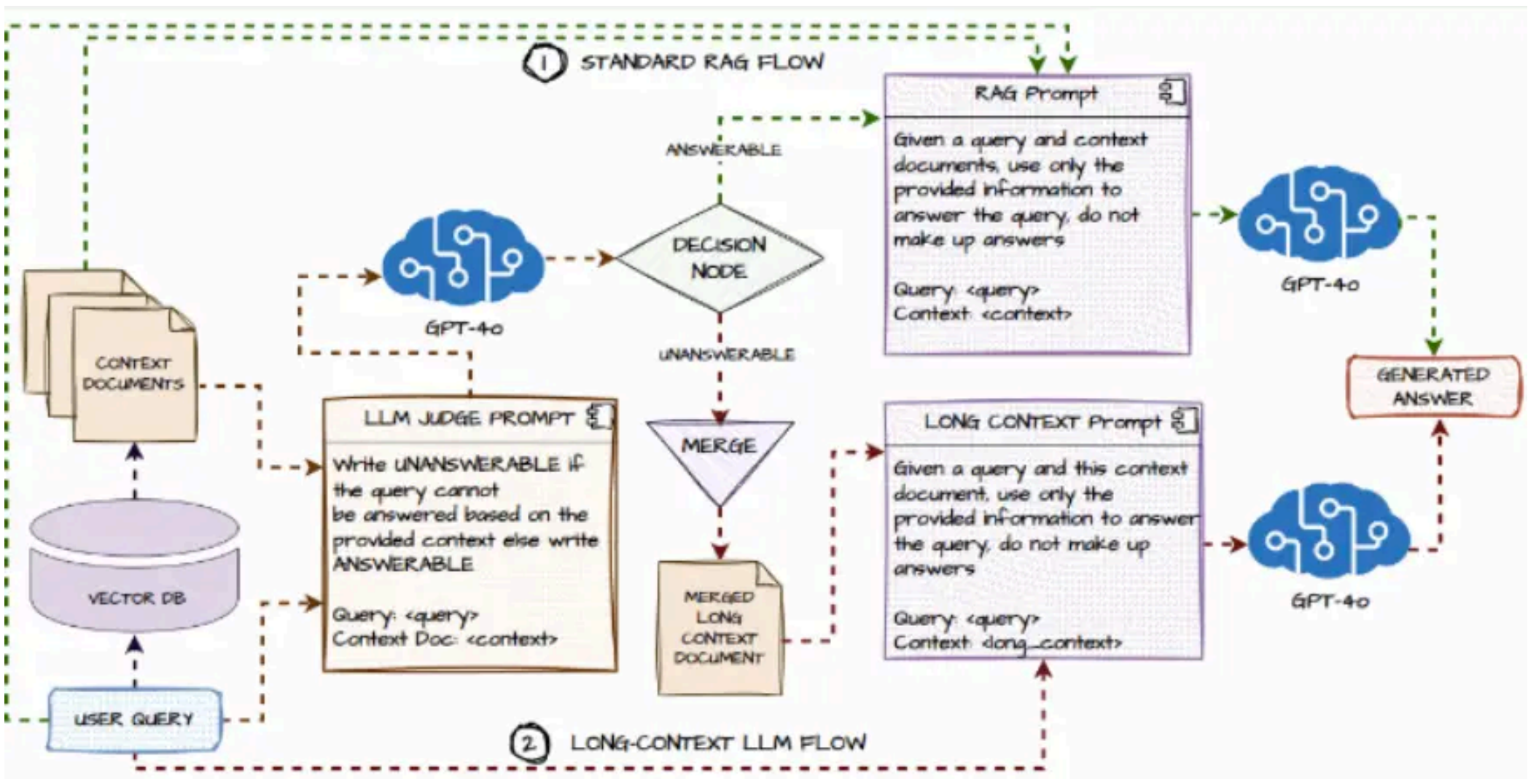
Speculative RAG

- Speculative RAG is a smart framework designed to make large language models (LLMs) both faster and more accurate when answering questions. It does this by splitting the work between two kinds of language models:
  - A small, specialized model that drafts potential answers quickly.
  - A large, general-purpose model that double-checks these drafts and picks the best one.

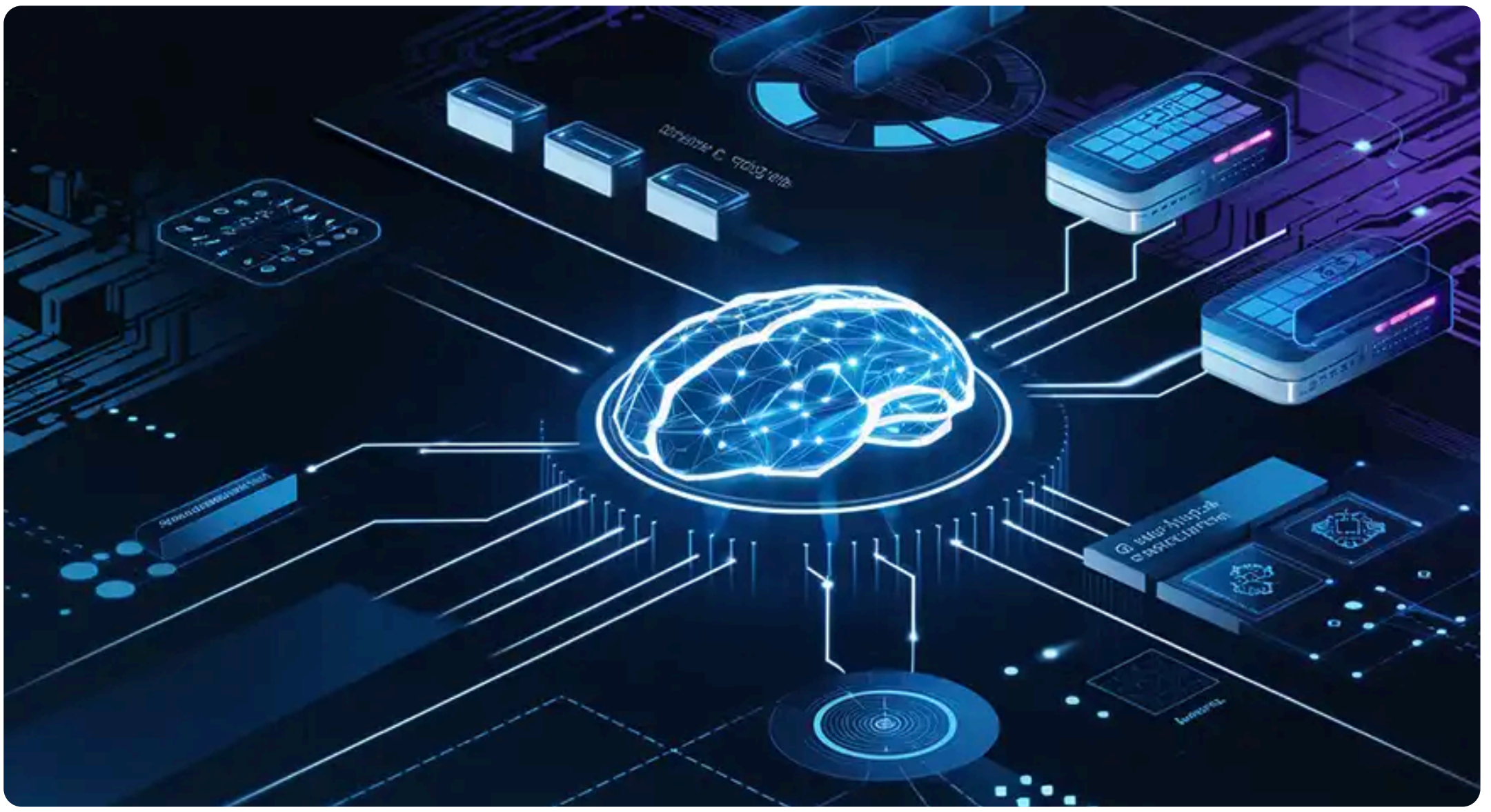


Self Route Agentic RAG

- Self Route is a design pattern in Agentic RAG systems where Large Language Models (LLMs) play an active role in deciding how a query should be processed. The approach relies on the LLM's ability to self-reflect and determine whether it can generate an accurate response based on the context provided. If the model decides it cannot generate a reliable response, it routes the query to an alternative method, such as a long-context model, for further processing.



For more information, please visit this article



Advanced AI Agents Best of Tech RAG

7 Agentic RAG System Architectures to Build AI Agents

Agentic RAG System Architectures: Explore dynamic frameworks merging RAG and AI Agents to enhance decision-making, retrieval, and more.