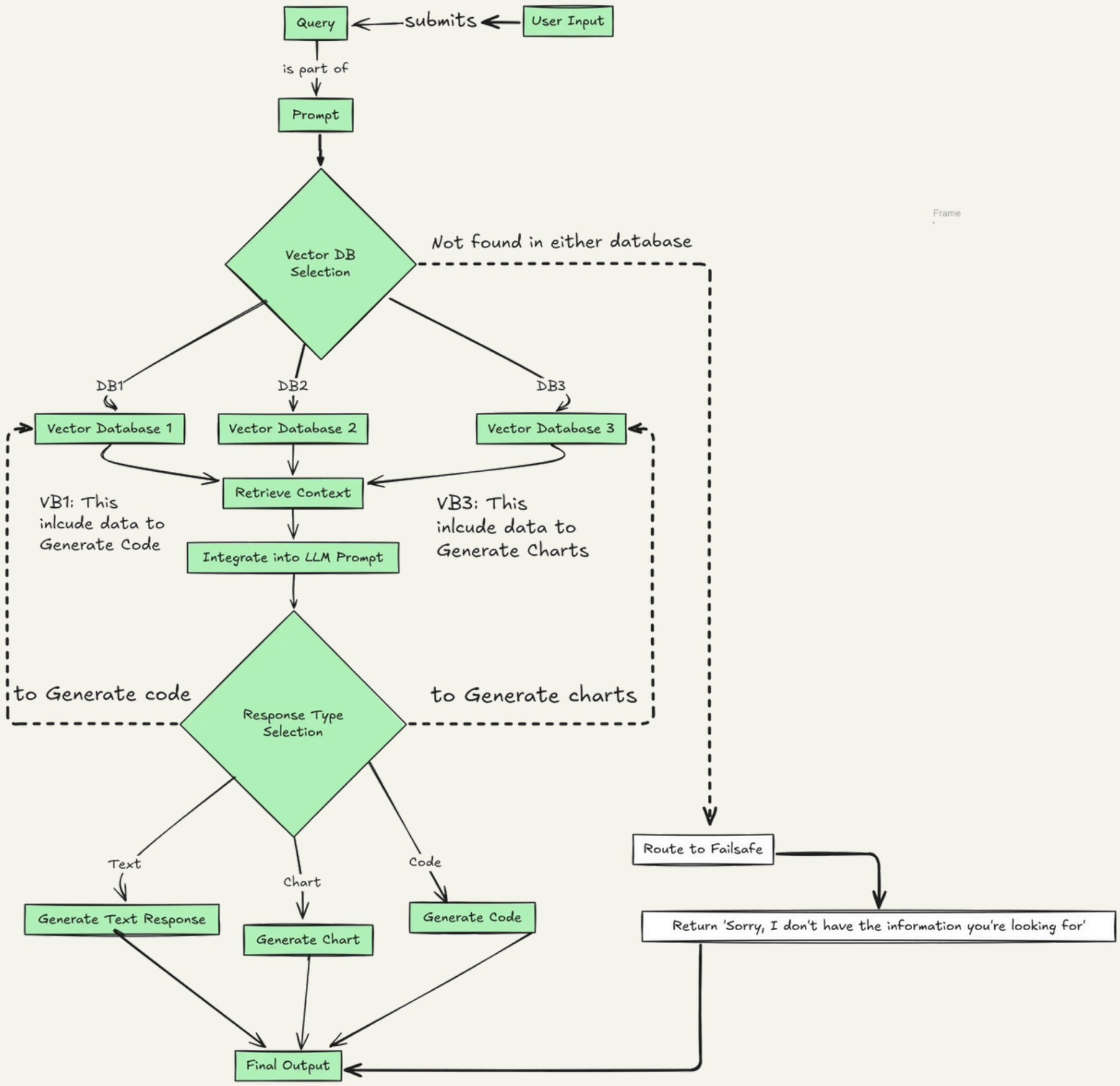
Guide to 7 Popular Agentic RAG System Architectures

Dipanjan (DJ)

Agentic RAG Workflow

The agent will decide the vector database most relevant to the query.

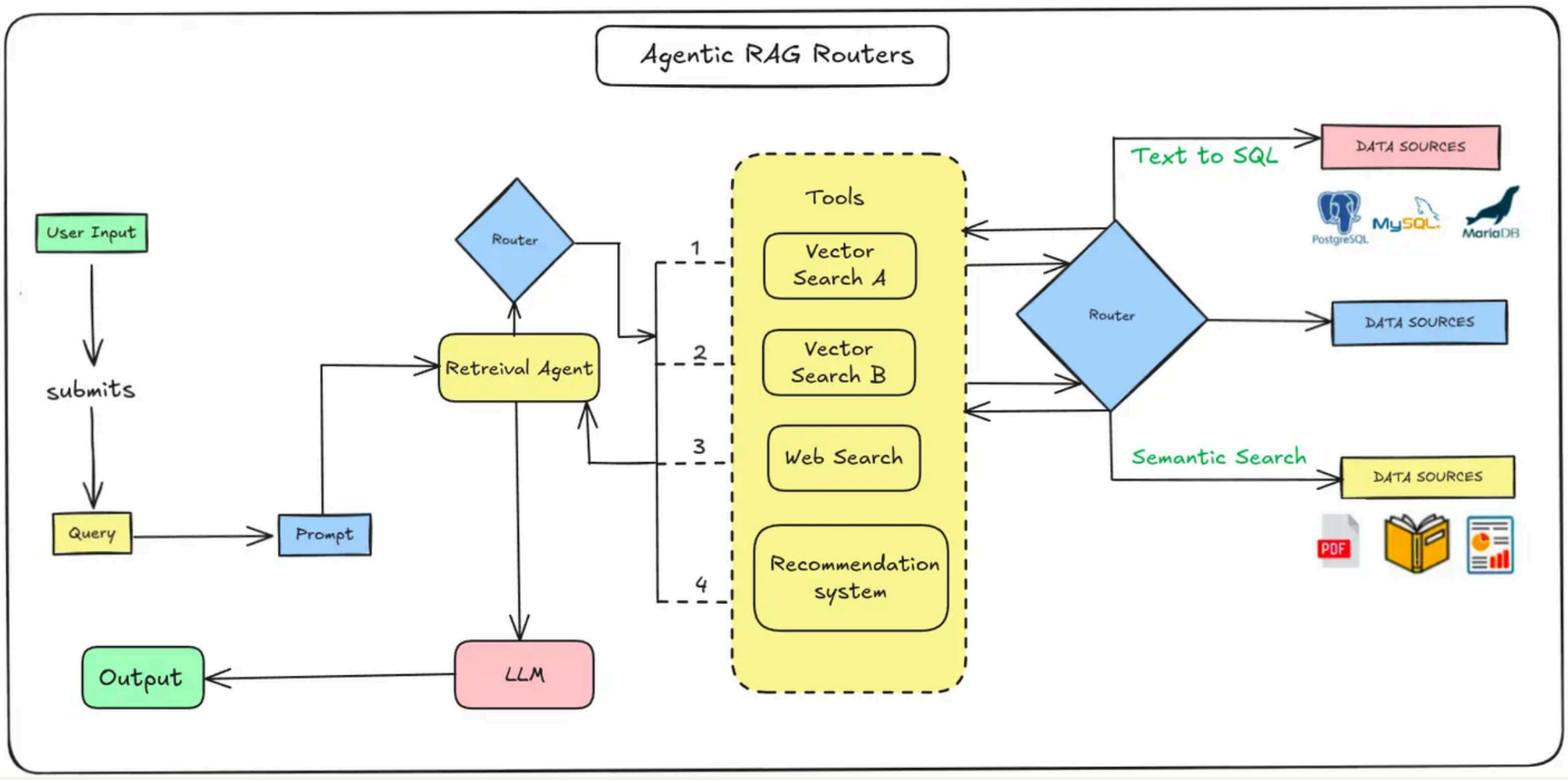


- **Agentic RAG is a combination of AI Agents and RAG Systems**
- **Agentic RAG Systems have various workflows depending on the use-case**
- **In this workflow we first create various vector databases based on specific document types and domains**
- **Based on the user query the LLM will reason and route to the relevant Vector DB**
- **Context documents are retrieved and the standard RAG flow is executed after that as usual**
- **Very useful when you have documents related to different domains, departments**

Source: 7 Agentic RAG System Architectures to Build AI Agents

Created by: Dipanjan (DJ)

1. Agentic RAG Routers

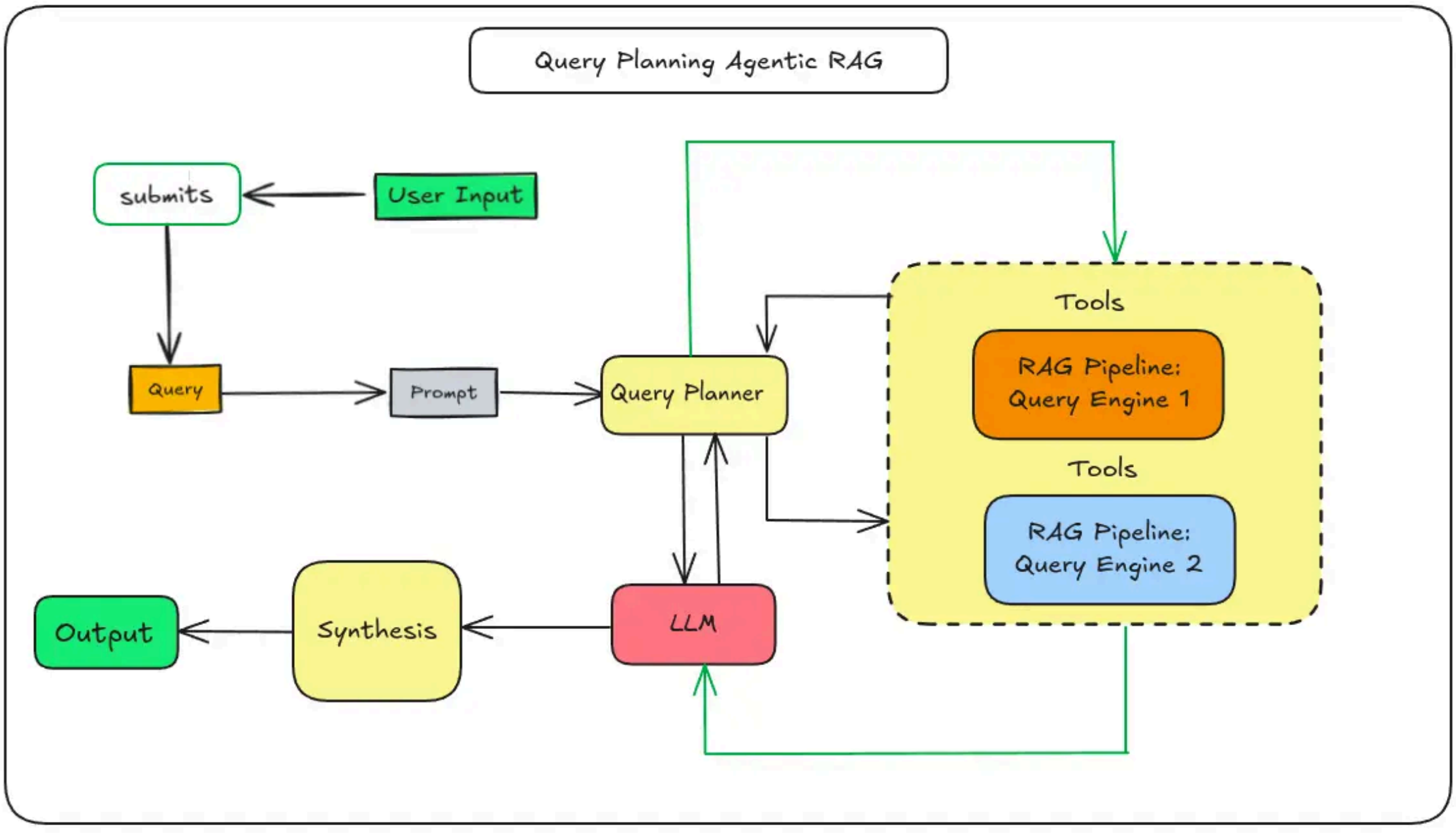


- **Agentic RAG Routers are systems designed to dynamically route user queries to appropriate tools or data sources, enhancing the capabilities of Large Language Models (LLMs)**
- **The primary purpose of such routers is to combine retrieval mechanisms with the generative strengths of LLMs to deliver accurate and contextually rich responses**
- **There are various types of Agentic RAG Routers:**
  - **Single Agentic RAG Router:** One unified agent responsible for all routing, retrieval, and decision-making tasks.
  - **Multiple Agentic RAG Routers:** Multiple agents, each handling a specific type of task or query.
- **Multi-Agentic RAG Routers are useful where the system employs multiple retrieval agents, each specializing in a specific type of task. For example:**
  - **Retrieval Agent 1** might handle SQL-based queries.
  - **Retrieval Agent 2** might focus on semantic searches.
  - **Retrieval Agent 3** could prioritize recommendations or web searches.

Source: 7 Agentic RAG System Architectures to Build AI Agents

Created by: Dipanjan (DJ)

2. Query Planning Agentic RAG

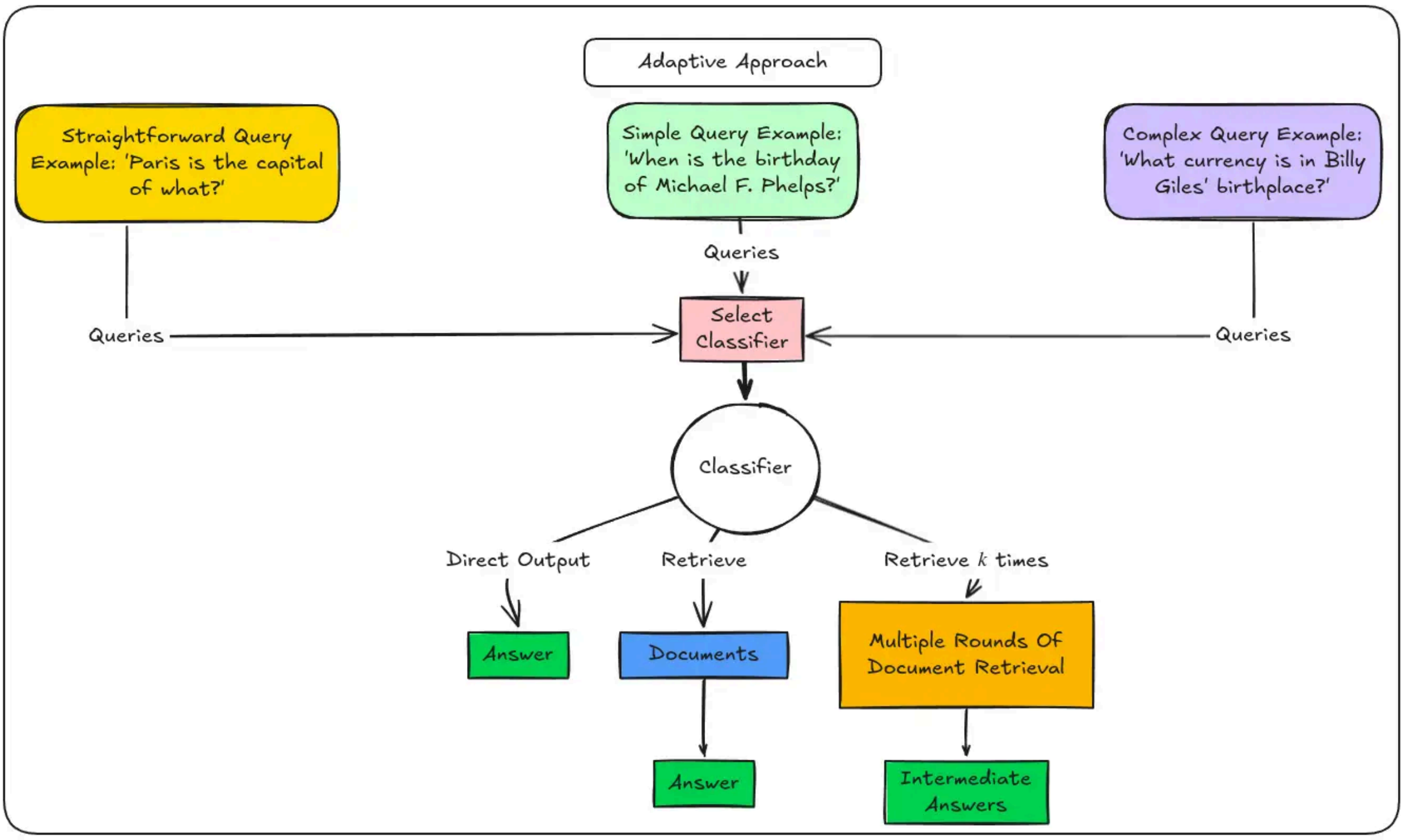


- **Query Planning Agentic RAG (Retrieval-Augmented Generation) is a methodology designed to handle complex queries efficiently by leveraging multiple parallelizable subqueries across diverse data sources**
- **This approach combines intelligent query division, distributed processing, and response synthesis to deliver accurate and comprehensive results**
- **The Query Planner is the central component orchestrating the process. It:**
  - **Interprets the query provided by the user**
  - **Generates appropriate prompts for the downstream components**
  - **Decide which tools (query engines) to invoke to answer specific parts of the query**
- **Very useful to break down complex queries step by step to retrieve results for each query to get to the final result**

Source: 7 Agentic RAG System Architectures to Build AI Agents

Created by: Dipanjan (DJ)

3. Adaptive RAG

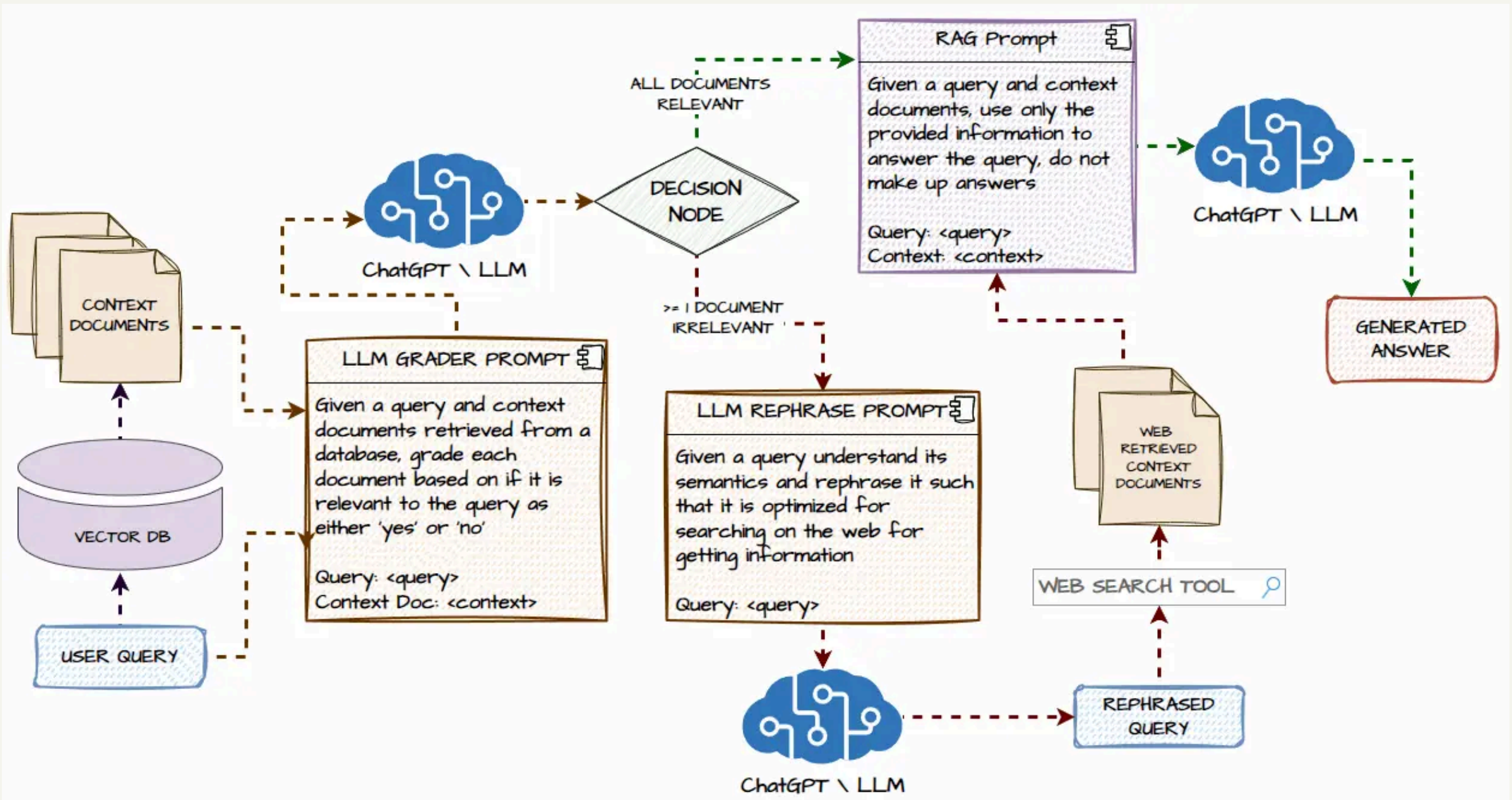


- **Adaptive Retrieval-Augmented Generation (Adaptive RAG) is a method that enhances the flexibility and efficiency of large language models (LLMs) by tailoring the query handling strategy to the complexity of the incoming query**
- **Adaptive RAG is a better version of routing RAG where it dynamically chooses between different strategies for answering questions—ranging from simple single-step approaches to more complex multi-step or even no-retrieval processes—based on the complexity of the query.**
- **This selection is facilitated by a classifier, which analyzes the query's nature and determines the optimal approach.**
- **Classifier Role:**
  - A smaller language model predicts query complexity
  - It is trained using automatically labelled datasets, where the labels are derived from past model outcomes and inherent patterns in the data
- **Dynamic Strategy Selection:**
  - For simple or straightforward queries, the framework avoids wasting computational resources
  - For complex queries, it ensures sufficient iterations through multiple retrieval steps

Source: 7 Agentic RAG System Architectures to Build AI Agents

Created by: Dipanjan (DJ)

4. Agentic Corrective RAG



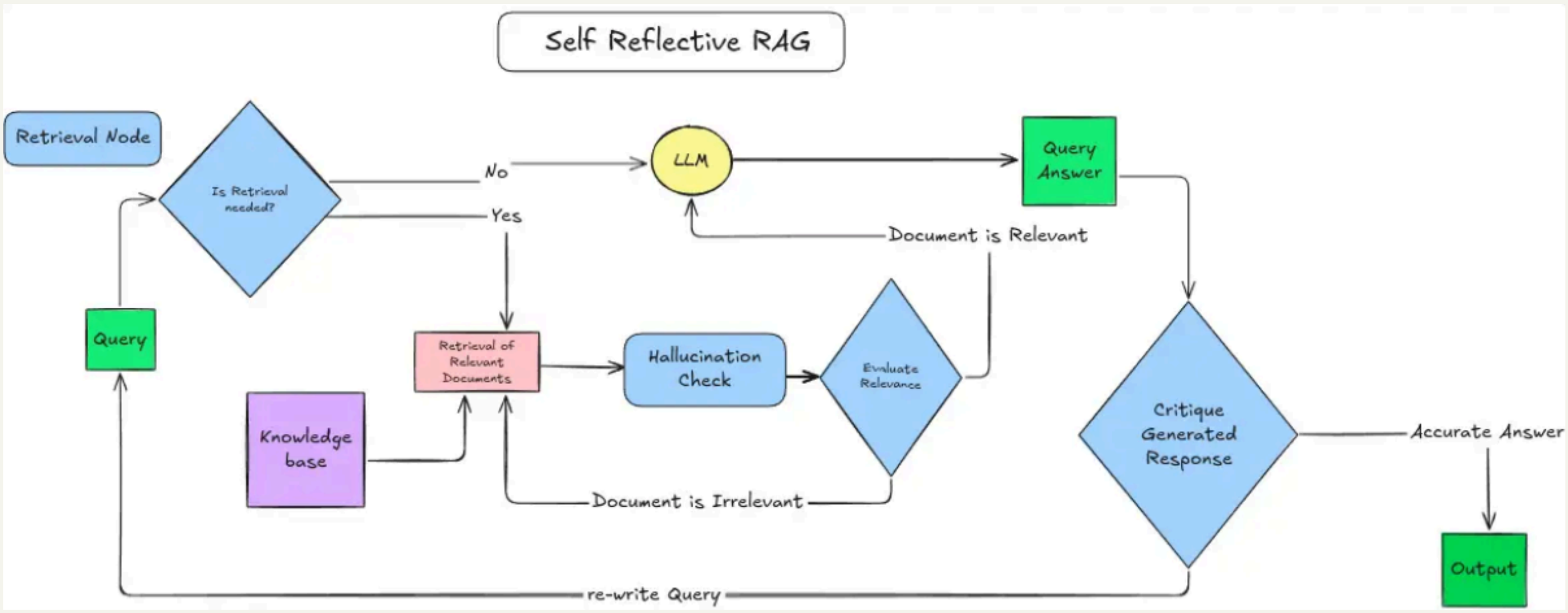
The Agentic Corrective RAG Architecture enhances Retrieval-Augmented Generation (RAG) with corrective steps for accurate answers:

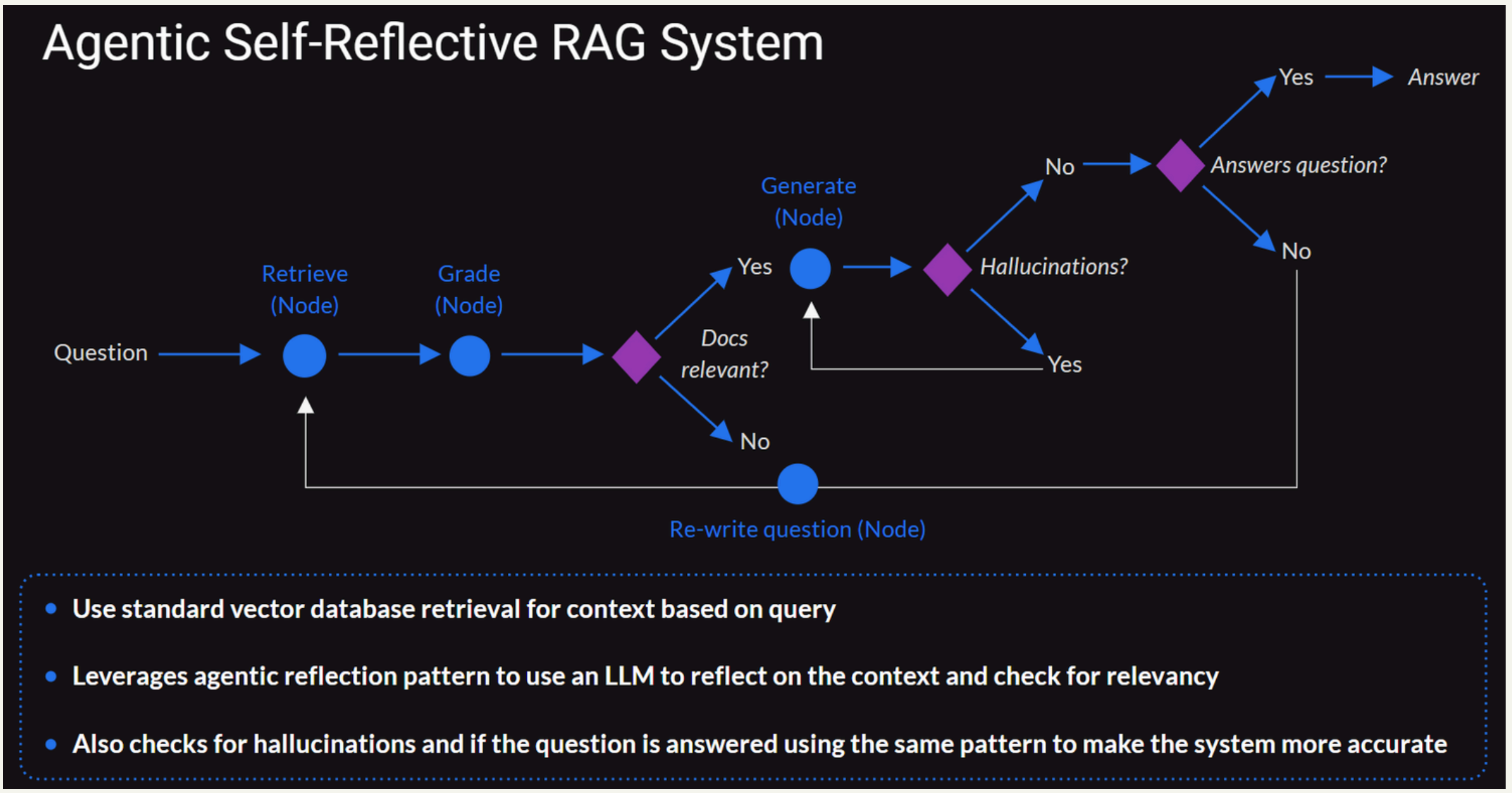
- **Query and Initial Retrieval:** A user query retrieves context documents from a vector database.
- **Document Evaluation:** The LLM Grader Prompt evaluates each document's relevance (yes or no).
- **Decision Node:**
  - **All Relevant:** Directly proceed to generate the answer.
  - **Irrelevant Documents:** Trigger corrective steps.
- **Query Rephrasing:** The LLM Rephrase Prompt rewrites the query for optimized web retrieval.
- **Additional Retrieval:** A web search retrieves improved context documents.
- **Response Generation:** The RAG Prompt generates an answer using validated context only.

Source: 7 Agentic RAG System Architectures to Build AI Agents

Created by: Dipanjan (DJ)

5. Self-Reflective RAG



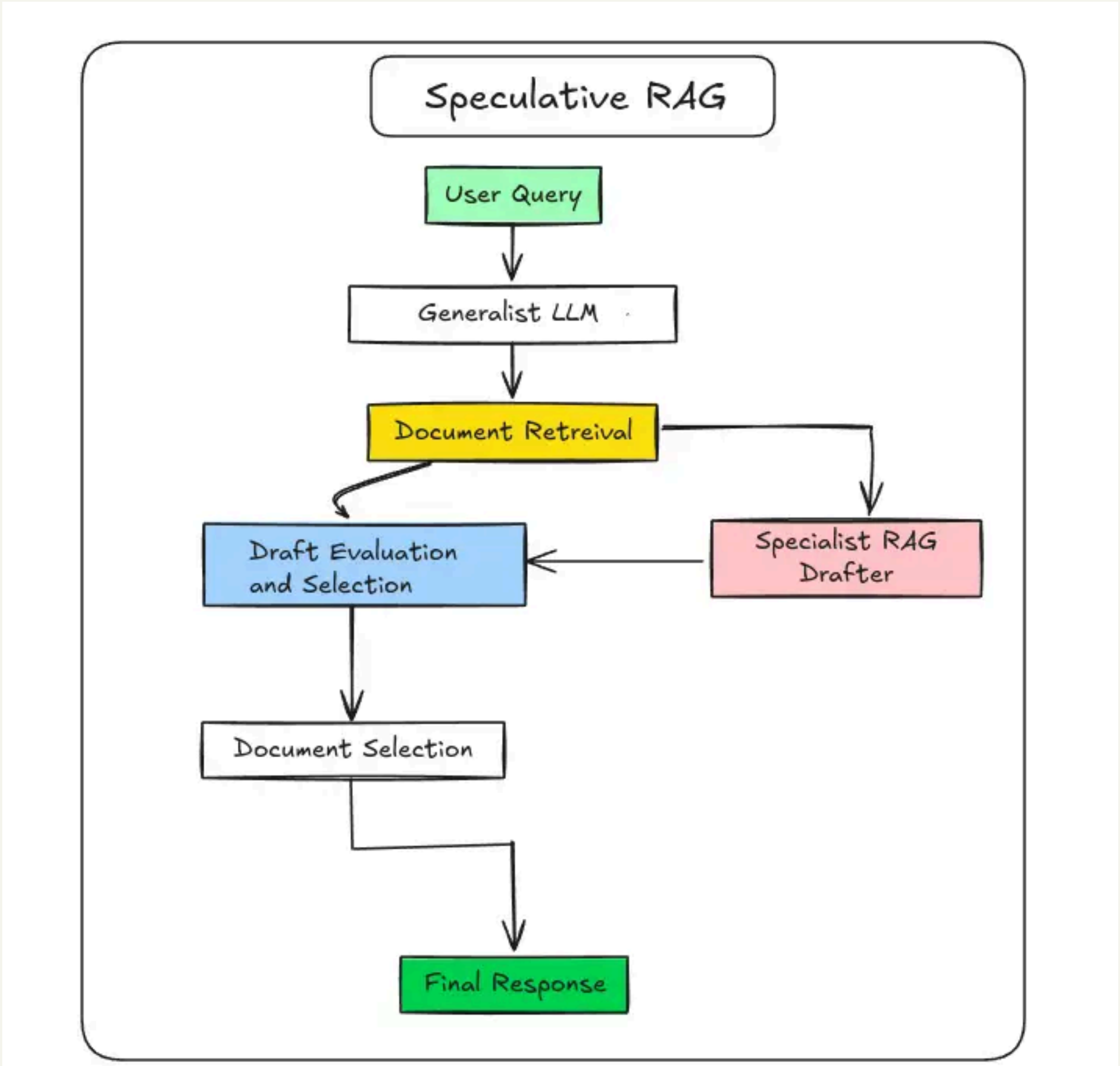


- **Self-reflective RAG (Retrieval-Augmented Generation) is an advanced approach that combines the capabilities of retrieval-based methods with generative models while adding an additional layer of self-reflection and logical reasoning.**
- **Self-reflective RAG helps in retrieval, re-writing questions, discarding irrelevant or hallucinated documents and re-try retrieval.**

Source: 7 Agentic RAG System Architectures to Build AI Agents

Created by: Dipanjan (DJ)

6. Speculative RAG

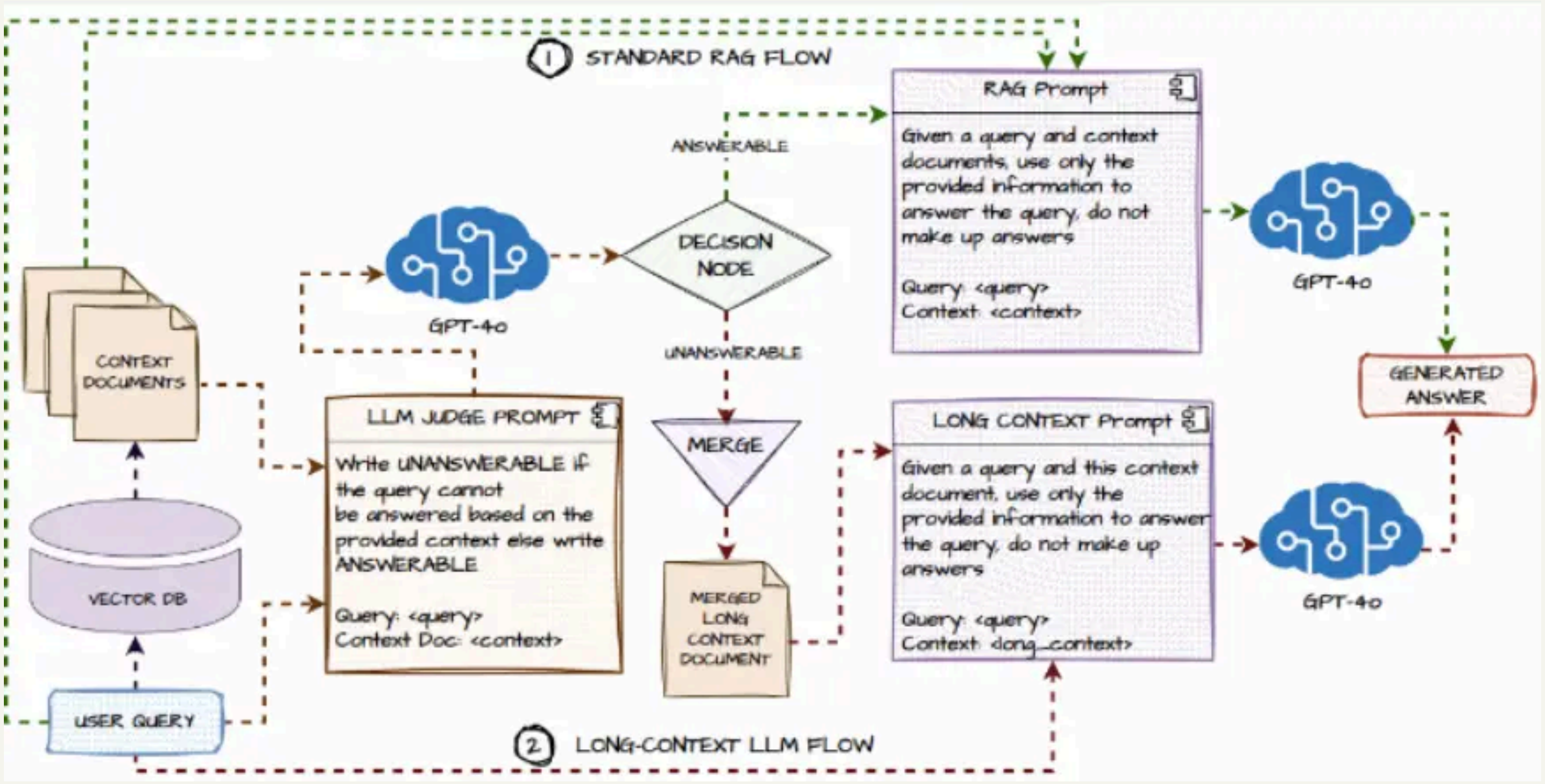


- **Speculative RAG is a smart framework designed to make large language models (LLMs) both faster and more accurate when answering questions. It does this by splitting the work between two kinds of language models:**
  - **A small, specialized model that quickly create multiple drafts of possible answers and includes reasoning for each draft (like saying, "This answer is based on this source").**
  - **A large, general-purpose model that double-checks these drafts and picks the best one as the final response.**

Source: 7 Agentic RAG System Architectures to Build AI Agents

Created by: Dipanjan (DJ)

7. 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**
- **This is a hybrid approach which combines Retrieval-Augmented Generation (RAG) and Long Context (LC) LLMs.**
- **Key components of Self Route:**
  - **Decision-making by LLMs:** Queries are evaluated to determine if they can be answered with the given retrieved context.
  - **Routing:** If a query is answerable, response is generated immediately. Otherwise, it is routed to a long-context model with the full context documents to generate the response.
  - **Efficiency and Accuracy:** This design balances cost-efficiency (avoiding unnecessary computation cost and time) and accuracy (leveraging long-context models only when needed).

Source: 7 Agentic RAG System Architectures to Build AI Agents

Created by: Dipanjan (DJ)

Detailed Article

7 Agentic RAG System Architectures to Build AI Agents

Pankaj Singh
Last Updated : 07 Jan, 2025

26 min read

For me, 2024 has been a year when I was not just using LLMs for content generation but also understanding their internal working. In this quest to learn about LLMs, RAG and more, I discovered the potential of AI Agents—autonomous systems capable of executing tasks and making decisions with minimal human intervention. Going back to 2023, Retrieval-Augmented Generation (RAG) was in the limelight, and 2024 advanced with Agentic RAG workflows, driving innovation across industries. Looking ahead, 2025 is set to be the "Year of AI Agents," where autonomous systems will revolutionize productivity and reshape industries, unlocking unprecedented possibilities with the Agentic RAG Systems.

These workflows, powered by autonomous AI agents capable of complex decision-making and task execution, enhance productivity and reshape how individuals and organisations tackle problems. The shift from static tools to dynamic, agent-driven processes has unlocked unprecedented efficiencies, laying the groundwork for an even more innovative 2025. Today, we will talk about the types of Agentic RAG systems. In this guide, we will go through the architecture of types of Agentic RAG and more.

Table of contents

CHECK OUT THE DETAILED ARTICLE HERE

Created by: Dipanjan (DJ)