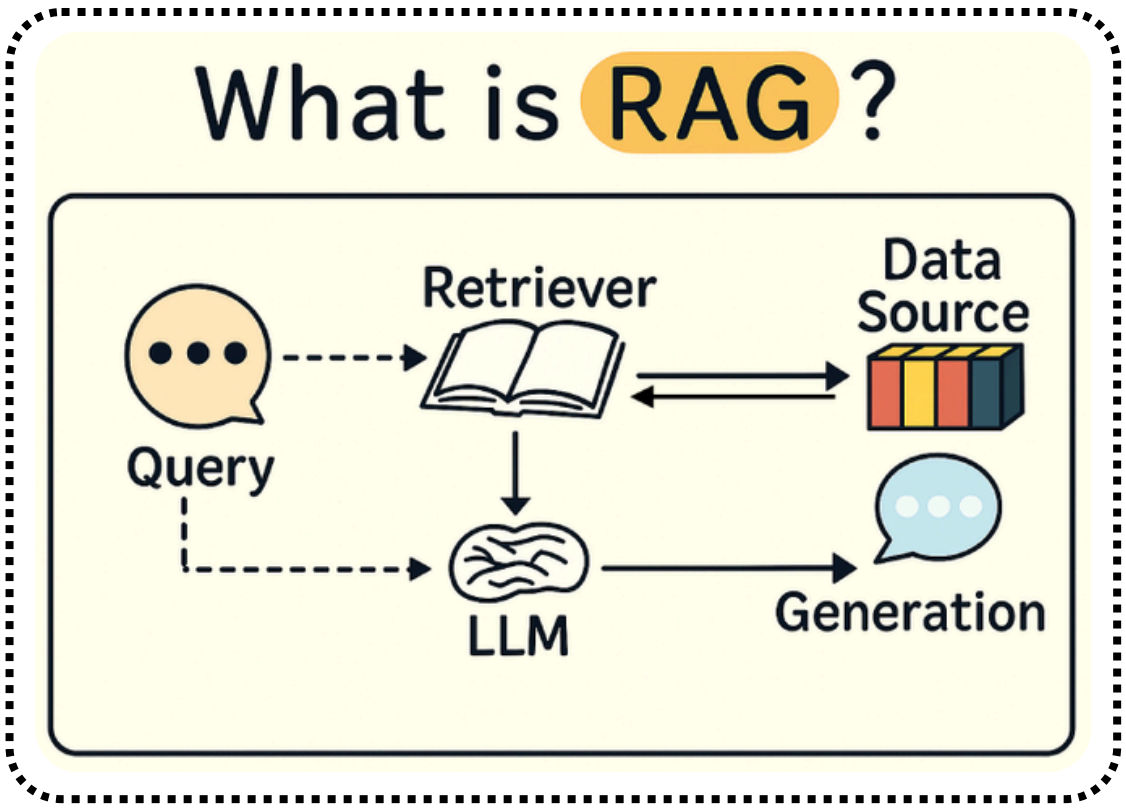
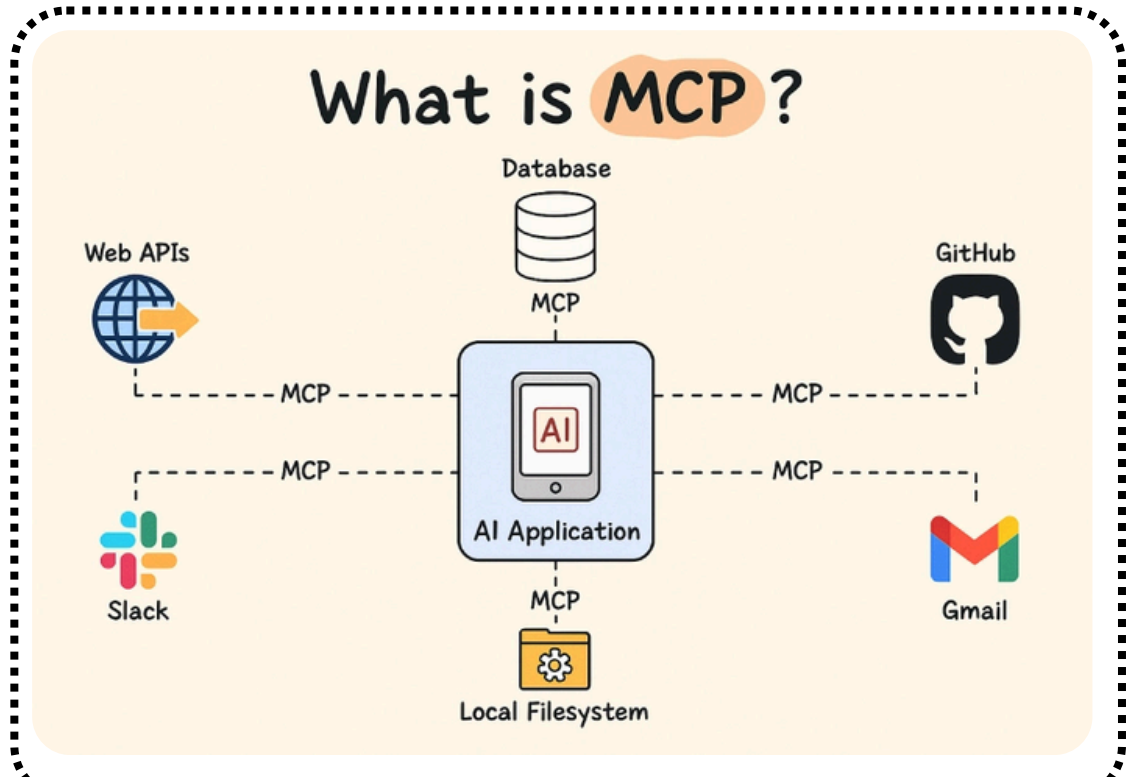
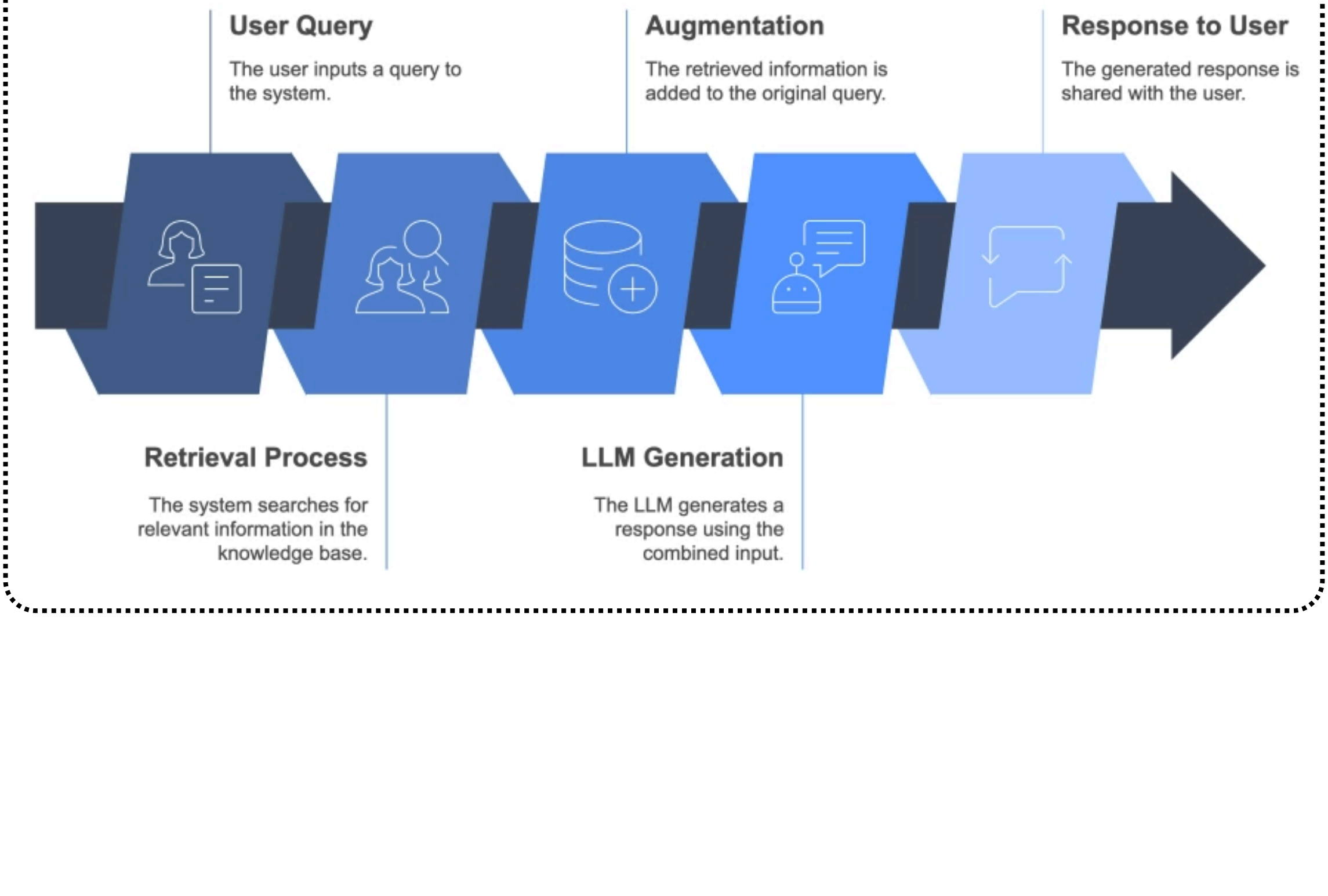
Analytics Vidhya

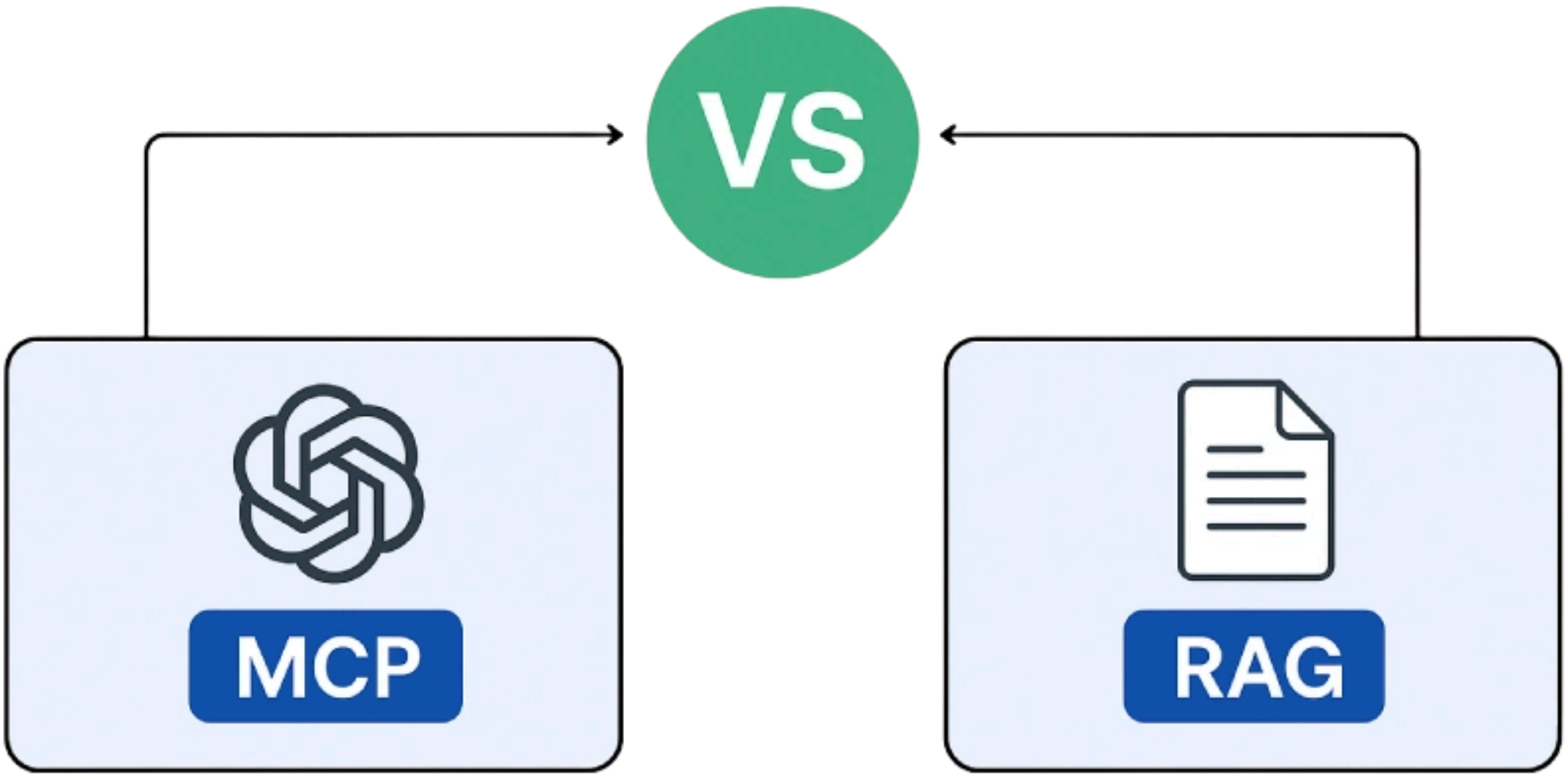
# MCP vs RAG



# MCP vs RAG: Competitors?




No, MCP and RAG are not competitors in the way they work or the tasks they perform.

As we have discussed in the previous sections, MCP and RAG perform different tasks and empower LLMs in different ways.

RAG powers LLMs with additional data while MCP grants LLMs the ability to act.

The key differences between MCP and RAG are summarised in the table below:

| Feature | RAG (Retrieval-Augmented Generation) | MCP (Model Context Protocol) |
|---|---|---|
| **Purpose** | Enhances knowledge of LLMs by retrieving relevant external data | Extends the capabilities of LLMs to use tools and perform actions |
| **Function** | Pulls info from documents, databases, or search APIs | Connects to tools, APIs, software, and real-time systems |
| **Use Case Type** | Improves response accuracy and context relevance | Enables real-world actions, tool use, and automation |
| **How It Works** | Retrieves relevant documents → augments the prompt → generates output | Uses structured tool schemas → selects tool → executes action |
| **Data Access** | Typically works with textual or vector data | Works with functional endpoints (e.g., APIs, plugins, webhooks) |
| **Execution** | Passive: Only retrieves and informs | Active: Can take actions like submitting forms or updating systems |
| **Example Task** | “What is our refund policy?” → fetches from policy doc | “Cancel my subscription” → triggers refund API |
| **Model Input Impact** | Expands the prompt with more content for better grounding | Doesn’t always expand the prompt, focuses on decision and execution |
| **Complexity** | Requires vector DB, chunking, and embedding logic | Requires tool definitions, security layers, and execution control |
| **Best Used For** | Knowledge-based Q&A, grounding, and content generation | Workflow orchestration, automation, and tool-augmented agents |

# Can MCP and RAG work together?

Yes, MCP and RAG can work together to help us design highly sophisticated AI workflows. RAG allows LLMs to pull relevant information while MCP executes tasks based on retrieved knowledge. Using these two together, we can create the following workflows:

### 1. RAG as a tool within the MCP framework

In this case, an LLM operating with MCP can have RAG as one of its tools, which it can use to fetch the required information.

Example: An MCP-powered AI system for a Marketing Campaign. It uses RAG to retrieve information regarding previous campaigns and competitor information. Then, using MCP-powered tools, it creates social media posts and schedules them across different platforms.

### 2. MCP for guiding RAG-Powered Agents

In systems involving multi-agents, each agent can have its own RAG pipeline and MCP can act as a coordinator for the system.

# Which one should you pick?


The choice between RAG, MCP, or RAG + MCP depends on the task. Each of the frameworks has its unique strengths. Here is how you can decide which approach to take:

- **RAG**: If your main goal is to improve the accuracy, relevance, and factual grounding of LLM-generated content, then “RAG” should be your choice.
- **MCP**: If your main goal is to allow your LLM to interact with external systems, perform actions, or leverage tools to complete its tasks, then “MCP” is your go-to path.
- **RAG + MCP**: If your goal is to build an intelligent, autonomous system that can better understand and act decisively, then the combination of RAG and MCP is your go-to option.