

## 🧠 Agentic RAG – Summary

Agentic RAG (Retrieval-Augmented Generation with Agents) is an evolution of traditional RAG systems.

Instead of a single static retrieve-then-answer pipeline, an LLM acts as an agent that can reason about what information it needs, decide actions, and iteratively refine retrieval and answers.

## ✅ How It Differs from Classic RAG

### Traditional RAG

1. Take user question
2. Retrieve top-k docs
3. Generate answer
4. ➡ fixed, one-shot, brittle when retrieval is poor

### Agentic RAG

1. Understand task + plan
2. Act (retrieve, search web, query tools, run chains)
3. Evaluate confidence
4. Decide next step:
  - retrieve more?
  - switch sources?
  - ask clarification?
  - compute something?
5. Produce final grounded answer

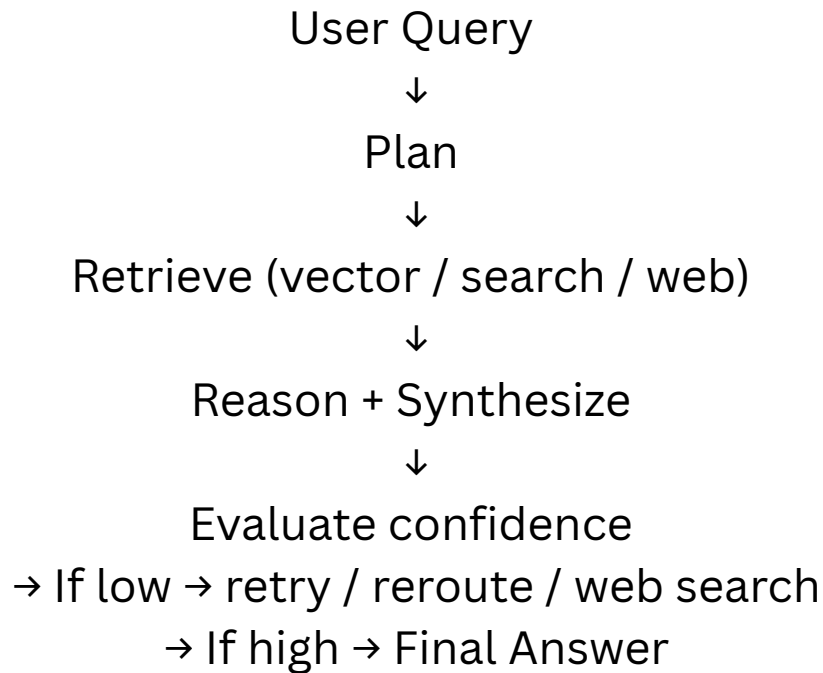
➡ dynamic, reasoning-driven, adaptive

## 🧩 Core Ideas

- Planning – model decides how to solve
- Tool Use – RAG retrieval + external APIs + calculators etc.
- Self-Reflection / Verification – confidence scoring, critique, hallucination reduction
- Iterative Retrieval – refine queries, expand search

- pipeline
-  Typical Architecture

Common node flow (like your LangGraph example):



### ★ Benefits

- Better accuracy
- Reduced hallucinations
- Handles ambiguous or missing knowledge
- More robust to poor retrieval
- Enables complex reasoning tasks

### Typical Use Cases

- long-context question answering
- research assistants
- enterprise knowledge search
- coding + docs
- multi-source evidence aggregation