Agentic RAG

1) BASIC Agentic RAG

```
__start__
   |
   v
retriever
   |
   v
responder
   |
   v
__end__
```

State

LLM

Query
   |
   v
START

Document
   |
   v Embedding

Retriever ⇄ Vectorstore

   | Context
   v
Responder   [LLM + context]
   | O/P
   v
END

2) Agentic RAG With ReAct

↓ Query

```
__start__
   |
   v
react_agent
   |
   v
__end__
```

AI Agent

Binding.

react_agent → LLM

LLM — Tool 1 → Retriever

LLM — Tool 2 → Wikipedia

LLM — Tool 3 → Web Search

BRAIN

SELF REFLECTION

Retriever + [LLM] ── SELF REFLECTION ──→

Prompt

Context + Output ──→ [LLM] ── YES → O/P → Done
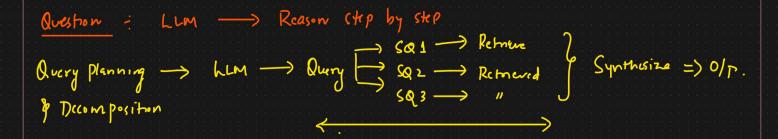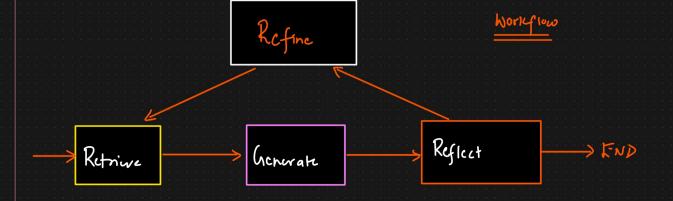   ── No →

✦ Query Planning & Decomp Vs COT
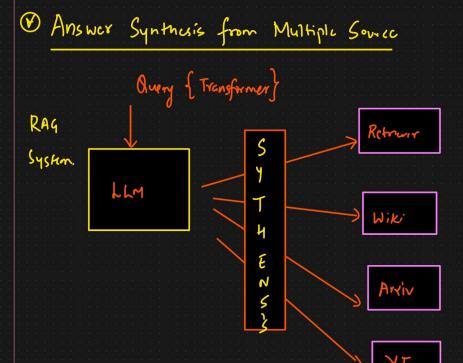
🔍 1. Conceptual Difference

Aspect	Chain-of-Thought (CoT)	Query Planning & Decomposition
🧠 Purpose	Let the LLM reason step-by-step	Break a complex query into structured sub-queries
👥 Style	Natural language reasoning path	Explicit sub-queries or formal question segments
📘 Inspiration	Human-like scratchpad thinking	Structured task planning or modular Q&A
💡 Agent Behavior	Think → Retrieve → Think → Answer	Plan all → Retrieve all → Answer once

Question : LLM ⟶ Reason step by step

Query Planning ⟶ LLM ⟶ Query
& Decomposition

- SQ1 ⟶ Retrieve
- SQ2 ⟶ Retrieved
- SQ3 ⟶ "

} Synthesize ⟹ O/P.

←———————————————→

```
[PLAN Query] ⟶ [Subquestion ⇓ Retrievir] ⟶ [Combine Context] ⟶ O/P.
```

✦ Iterative Retrieval

<u>Workflow</u>

```
        [Refine]
        ↙      ↖
[Retrive] ⟶ [Generate] ⟶ [Reflect] ⟶ END
```

✔ Answer Synthesis from Multiple Source

Query { Transformer }

RAG System.

```
         Query { Transformer }
              │
              ▼
        ┌──────────┐        S                ┌──────────┐
        │          │        Y         ──────▶│ Retrevr  │
        │   LLM    │──────  T                └──────────┘
        │          │        H
        └──────────┘        E         ──────▶┌──────────┐
                            N                │  Wiki    │
                            S                └──────────┘
                            }
                                      ──────▶┌──────────┐
                                             │  Arxiv   │
                                             └──────────┘

                                      ──────▶┌──────────┐
                                             │   YT     │
                                             └──────────┘
```

⊛ Autonomous RAG
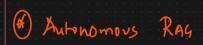
🤖 What is Autonomous RAG?

Autonomous RAG is a Retrieval-Augmented Generation system where the LLM (or agent) is capable of reasoning, planning, acting, reflecting, and improving — on its own — without manual control over each step.
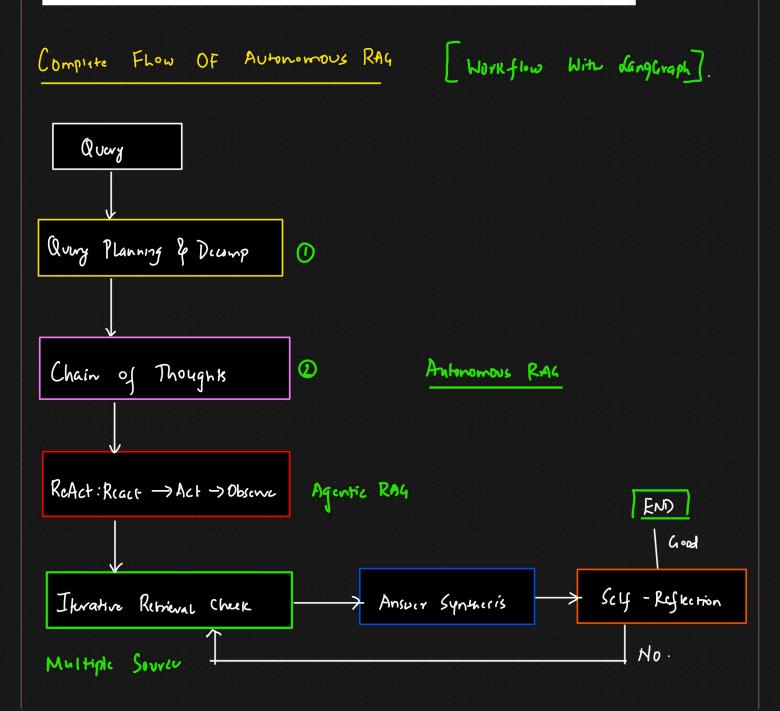
It combines:

🧠 Agentic reasoning (like ReAct or LangGraph agents)
🔁 Self-reflection & self-correction
🛠 Dynamic tool selection
📚 Multi-source retrieval

ReAct	COT	Query PLANNING DECOMPOSE	Itrieval Statrgies

SELF REFLECTION

📦 Core Components of Autonomous RAG

Component	Role
🕐 Planner Agent	Breaks complex queries into sub-questions
🔧 Tool Selector	Chooses between Wikipedia, ArXiv, vector DBs, APIs, etc.
📚 Retriever	Executes tool calls to retrieve relevant documents
📐 Synthesizer	Uses LLM to generate the final answer
☐ Reflector	Verifies whether context or answer is good enough
🔁 Retry Loop	Refines and retries if reflection fails
🧠 Memory (opt.)	Stores feedback, logs bad queries, or improves prompts/tools

Complete Flow OF Autonomous RAG   [Workflow With LangGraph].

```
┌─────────┐
│  Query  │
└─────────┘
     │
     ▼
┌──────────────────────┐
│ Query Planning & Decomp │  ①
└──────────────────────┘
     │
     ▼
┌──────────────────┐
│ Chain of Thoughts │  ②        Autonomous RAG
└──────────────────┘
     │
     ▼
┌──────────────────────────┐
│ ReAct : React → Act → Observe │    Agentic RAG              ┌─────┐
└──────────────────────────┘                                 │ END │
     │                                                        └─────┘
     ▼                                                            │ Good
┌──────────────────────┐      ┌──────────────────┐      ┌──────────────────┐
│ Iterative Retrieval Check │──▶│ Answer Synthesis │──▶│ Self - Reflection │
└──────────────────────┘      └──────────────────┘      └──────────────────┘
     ▲                                                            │
     │                                                            │ No.
Multiple Source └───────────────────────────────────────────────┘
```

🧠 Agentic RAG vs Autonomous RAG

Concept	Agentic RAG	Autonomous RAG
✴️ Definition	A RAG system that uses an **agentic approach** — where an LLM **reasons, plans, and acts** using tools	A RAG system that **operates independently**, with **full self-management** of planning, retrieving, reflection, and improvement
🎯 Focus	Structured reasoning and tool use (ReAct, LangGraph, etc.)	Complete autonomy in task execution, retry, and learning
⚙️ Behavior	Think → Act → Observe → Answer	Think → Act → Reflect → Retry → Learn → Answer
🔄 Retry Logic	Optional — usually static agent plans	Built-in retry/refine strategies (context + answer reflection)
🪞 Self-Reflection	May include it optionally	Core feature: reflects on retrieval & answers before finalizing
🛠️ Tool Use	Uses tools via agents (e.g., Wikipedia, SQL, ArXiv)	Selects and adapts tools dynamically based on reasoning
🧠 Planner	Often present (manual or LLM-generated plans)	Always present — triggers multi-step workflows adaptively
💡 Learning Loop	Not always present	May log feedback, improve over time