Multi Agentic Design

Advanced LLMs

Lara WEHBE - TheAIEngineers - April 2025

Definition

What Are Multi-Agent AI Workflows?

- A multi-agent AI workflow = multiple autonomous agents collaborating on a task.
- Each agent specializes in a skill (e.g., retrieval, reasoning, execution).
- Inspired by human workflows and organizational roles.

Core Concepts

Think of agents as mini-programs that:

- Accept instructions
- Hold memory/context
- Interact with tools or other agents

🧪 Example: MediScribe AI with triage agent → summarizer agent → diagnosis recommender

Common Roles

Think of agents as mini-programs that:

- Coordinator / Planner: breaks down tasks
- **Retriever Agent**: fetches data from RAG/vector DBs
- **Generator Agent**: calls LLMs to synthesize content
- **Executor Agent**: performs actions (e.g., API call)
- **Validator Agent**: checks or verifies output quality

✅ Pros of Multi-Agent Workflows

- Modular & composable
- Easier debugging (by agent role)
- Scalable (more agents, more parallelism)
- Reusable agents across projects
- Reflects real-world team workflows

❌ Cons & Limitations

- More complex to manage
- Difficult to debug when roles are unclear
- Requires thoughtful design (looping, retries, memory)
- Potential overhead and latency with chaining agents

When to Use Multi-Agent Systems?

✅ Use when:

- Task has natural decomposition (substeps)
- Multiple tools need to be orchestrated
- Context switching across subtasks (e.g., legal + research)
- Decision delegation improves clarity (e.g., triage systems)

When to Use Multi-Agent Systems?

❌ Avoid when:

- Task is too simple for multiple agents
- Latency is critical
- You lack clear role boundaries

🤔 RAG or Multi-Agent? When to Use What

🧱 RAG (Retrieval-Augmented Generation)

- Adds external knowledge to LLM prompts
- Best for: **answering**, **summarizing**, or **searching**
- Think of it like: “Let me go read this and respond.”

🤔 RAG or Multi-Agent? When to Use What

🧑‍💻 Multi-Agent Workflow

- Breaks task into **roles** and **decisions**
- Best for: multi-step logic, delegation, tool use
- Think of it like: “I’ll assign this task to the right teammate.”

🤔 RAG or Multi-Agent? When to Use What

⚠️ Even with 1M tokens in Gemini 2.5...

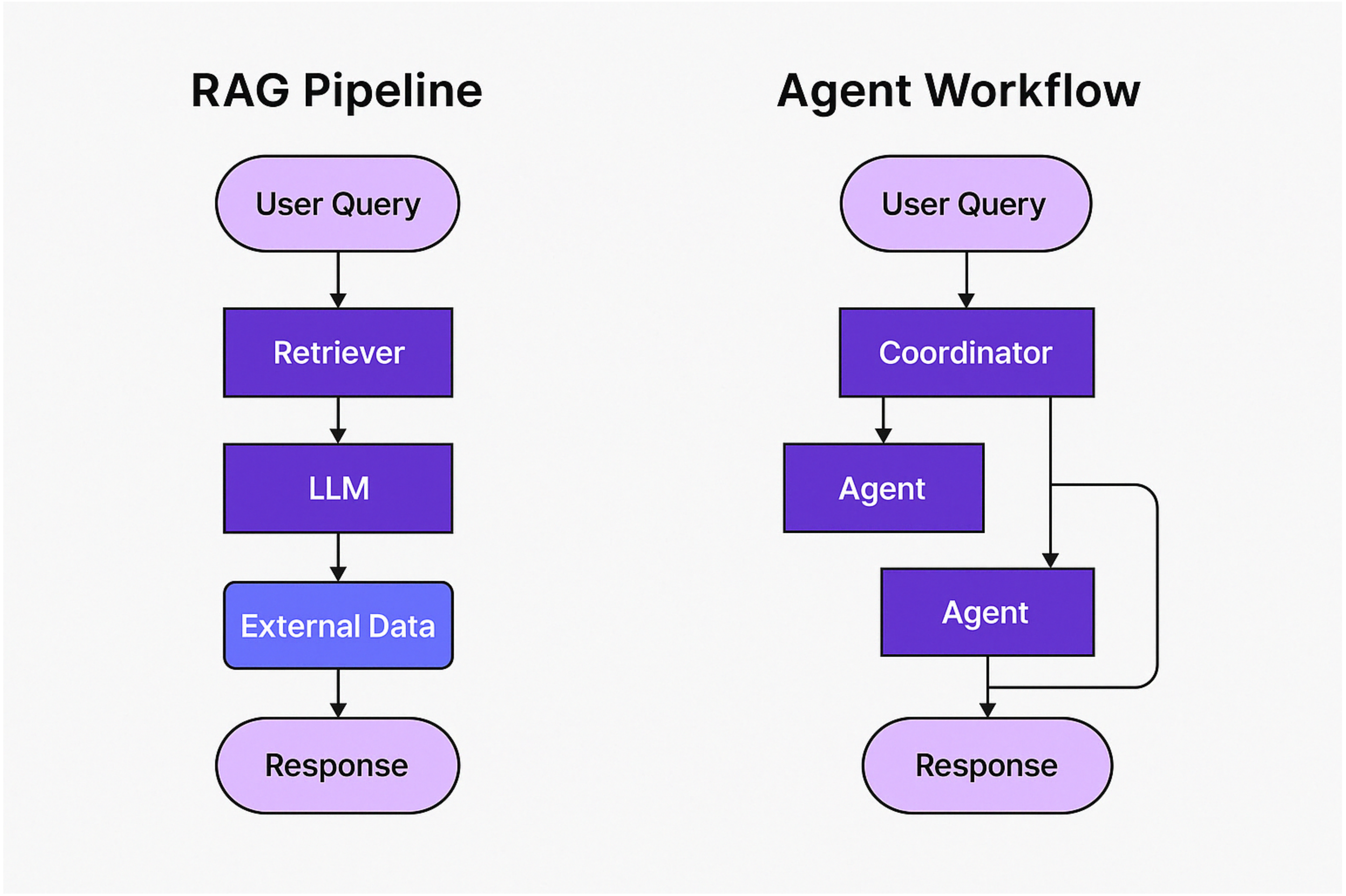
You still can't rely on context alone.

- Context ≠ Reasoning
- More context can confuse rather than help if not **structured**
- Agents reason, decide, and act — not just summarize

💡 Use RAG to **feed smart agents**, not replace them.

Design systems (patterns)

- Reflection: The LLM examines its own work to come up with ways to improve it.
- Tool Use: The LLM is given tools such as web search, code execution, or any othe function to help it gather information, take action, or process data.
- Planning: The LLM comes up with, and executes, a multistep plan to achieve a goa example, writing an outline for an essay, then doing online research, then writing a and so on).
- Multi-agent collaboration: More than one AI agent work together, splitting up tasks discussing and debating ideas, to come up with better solutions than a single ager would.


RAG Pipeline
User Query
Retriever
LLM
External Data
Response
Agent Workflow
User Query
Coordinator
Agent
Agent
Response

Example Use cases

1. Medical Pre-diagnosis
   - Intake Agent → Summarizer → Risk Evaluator → Recommender
2. Content Automation
   - Idea Generator → Research Agent → Script Writer → Editor → Publisher
3. Infrastructure-as-Code Bot
   - Requirement Interpreter → Terraform Generator → Validator → Deployment Agent
4. Customer Support
   - Router Agent → Retrieval Agent → Resolution Agent

🧪 Why CrewAI?

- Easy orchestration of agent roles
- Clean syntax for adding tools and memory
- Supports synchronous and sequential flows
- Great for demos, MVPs, and hackathons