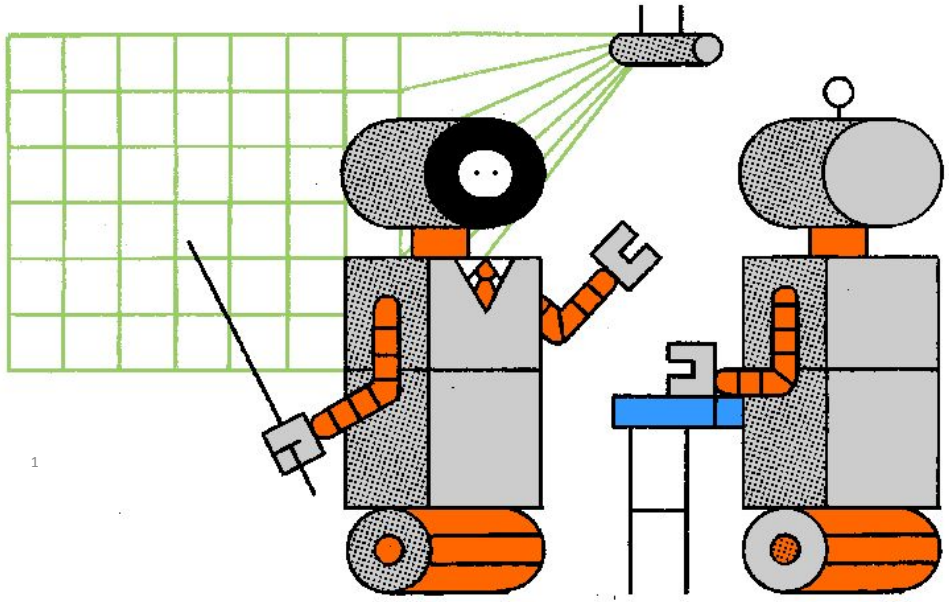
traversaal.ai

ReAct Agents with Memory

Module 4

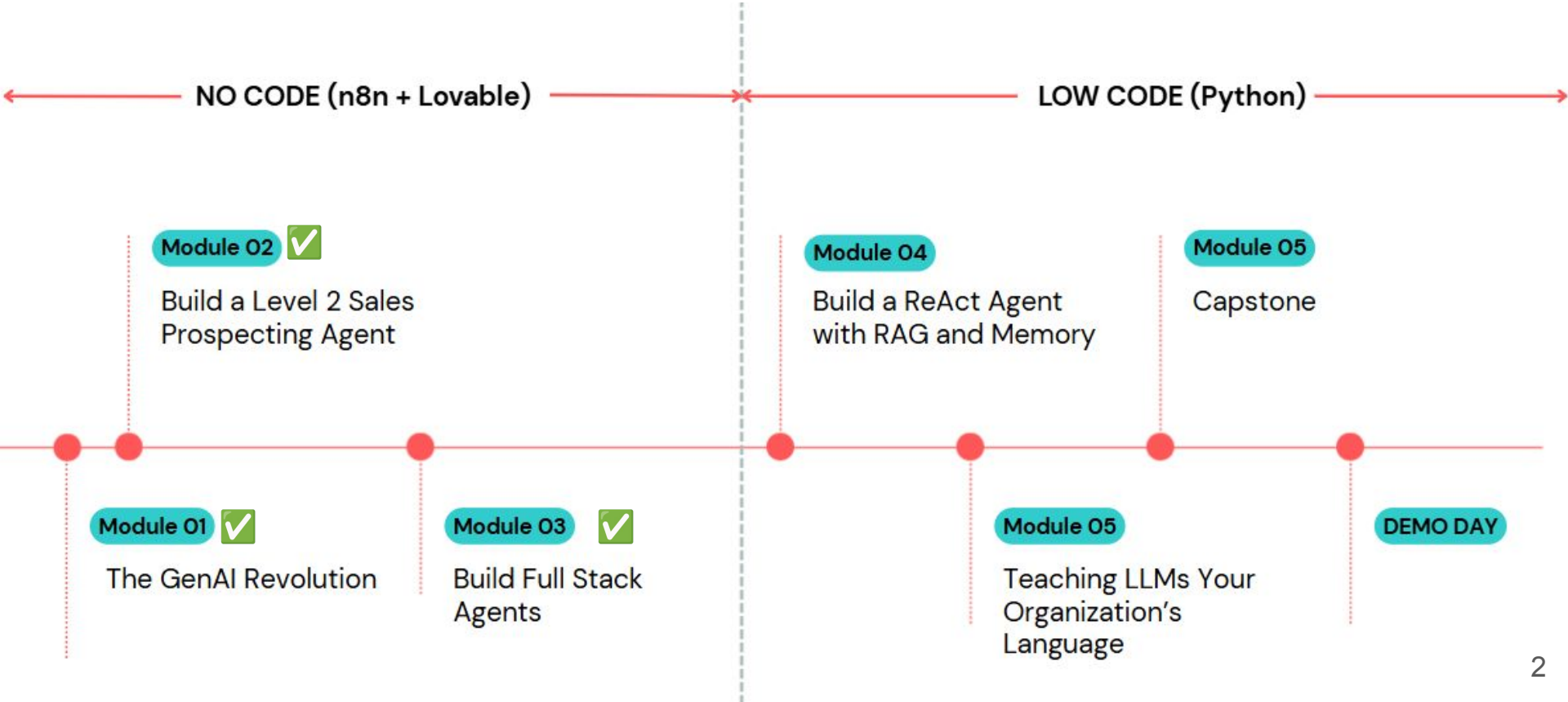


Hamza Farooq

Founder & CEO

Course OverLook

Building Gen AI Agents for Enterprise: Beyond the Hype



Recap from Module 03

- Integrating Front End Interfaces with n8n
- Model Context Protocol (MCP) in n8n
- Structuring Agent Logic and User Interactions
- Designing End-to-End Agent Workflows

Expected Outcomes for Capstone Project

🛠️ Design and develop an AI agent using course concepts

🔄 Work with tools like n8n, LangGraph, or custom stacks to connect memory, APIs

🚀 Define a use case, build key features, and deploy your solution

🤝 Brainstorm, build, and present together — just like in real AI teams

🎥 Share your solution through a 3–5 min demo and a short write-up

💡 Pick meaningful problems and apply agentic thinking to tackle them creatively

Did submit your idea for capstone project?

List your team and projects here if you have not done it already…

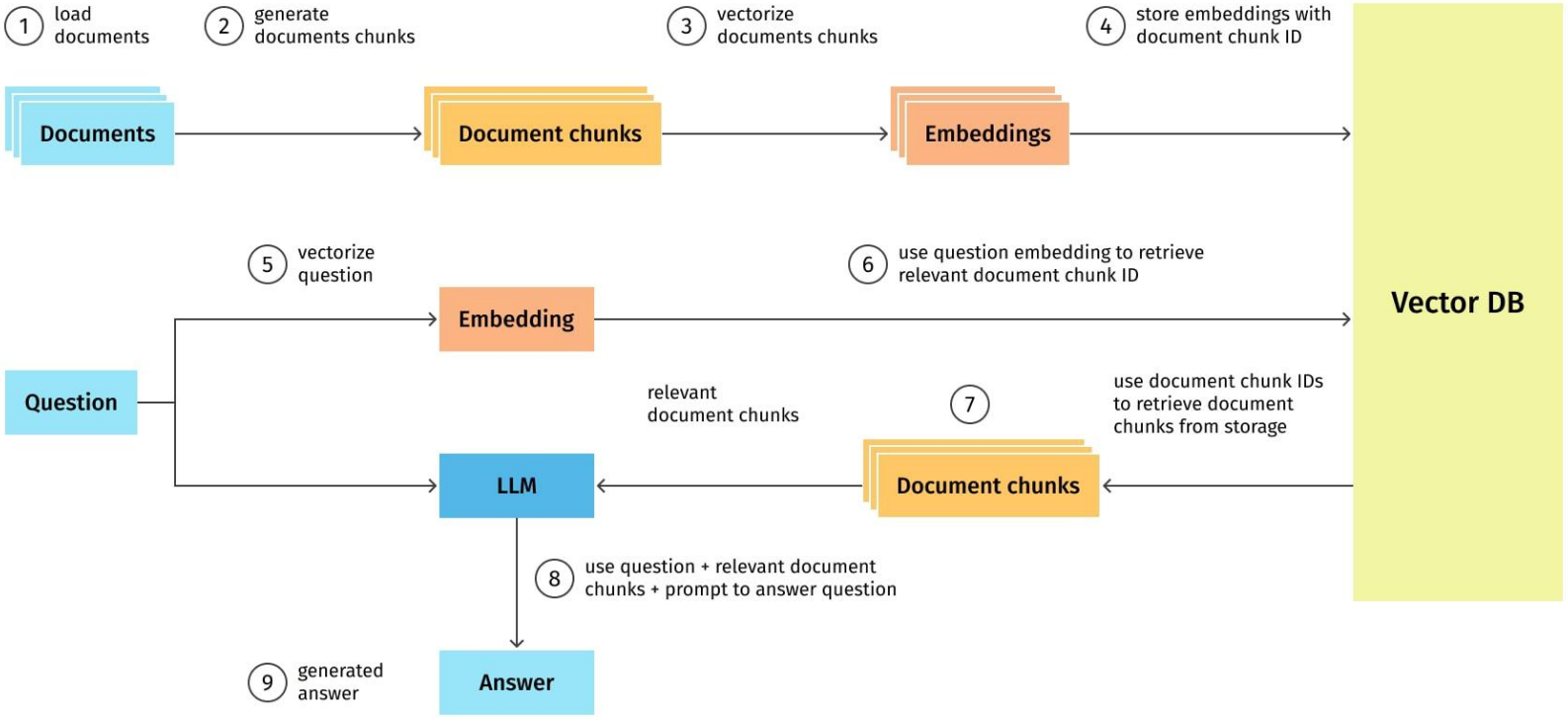
Learning Outcomes for Module 04

- Extending n8n capabilities
- Integrating APIs for Internet Search and RAG
- Introducing ReAct Agents
- Designing End-to-End Agent Workflows

01

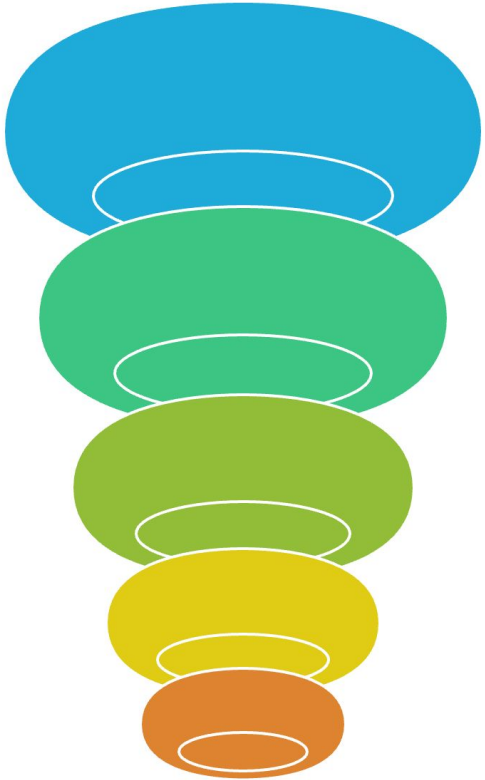
Improving RAG: RAG to Agentic RAG

RAG Architecture



00: RAG to Agentic RAG

Challenges with Basic RAG Application



Basic Accuracy

Provides basic accuracy using retrieved information

Similarity Checks

Uses basic checks that may miss relevant data

Straightforward Handling

Handles queries simply without enhancements

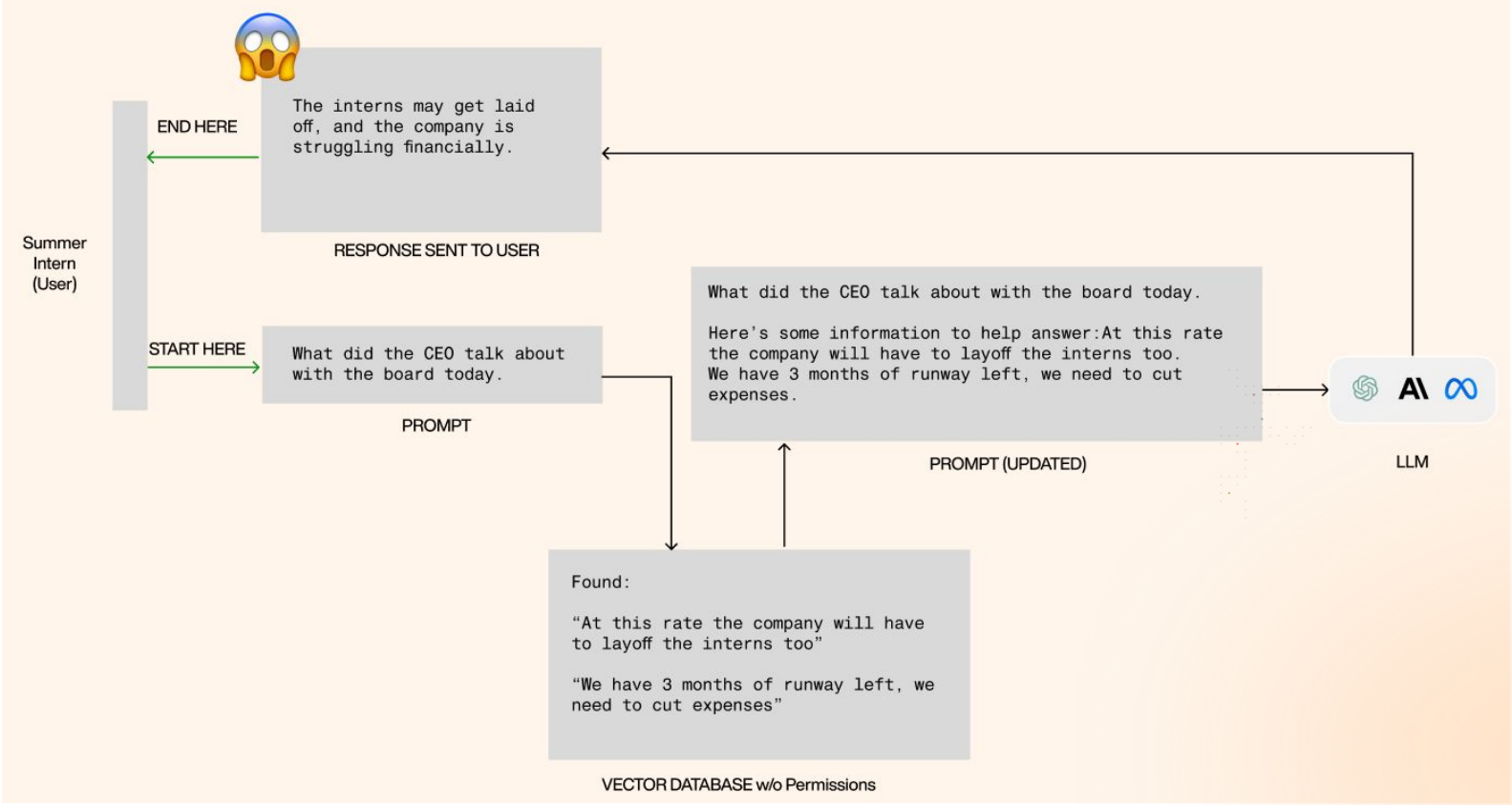
Efficiency Issues

Becomes less efficient as data size grows

Single Retrieval Pass

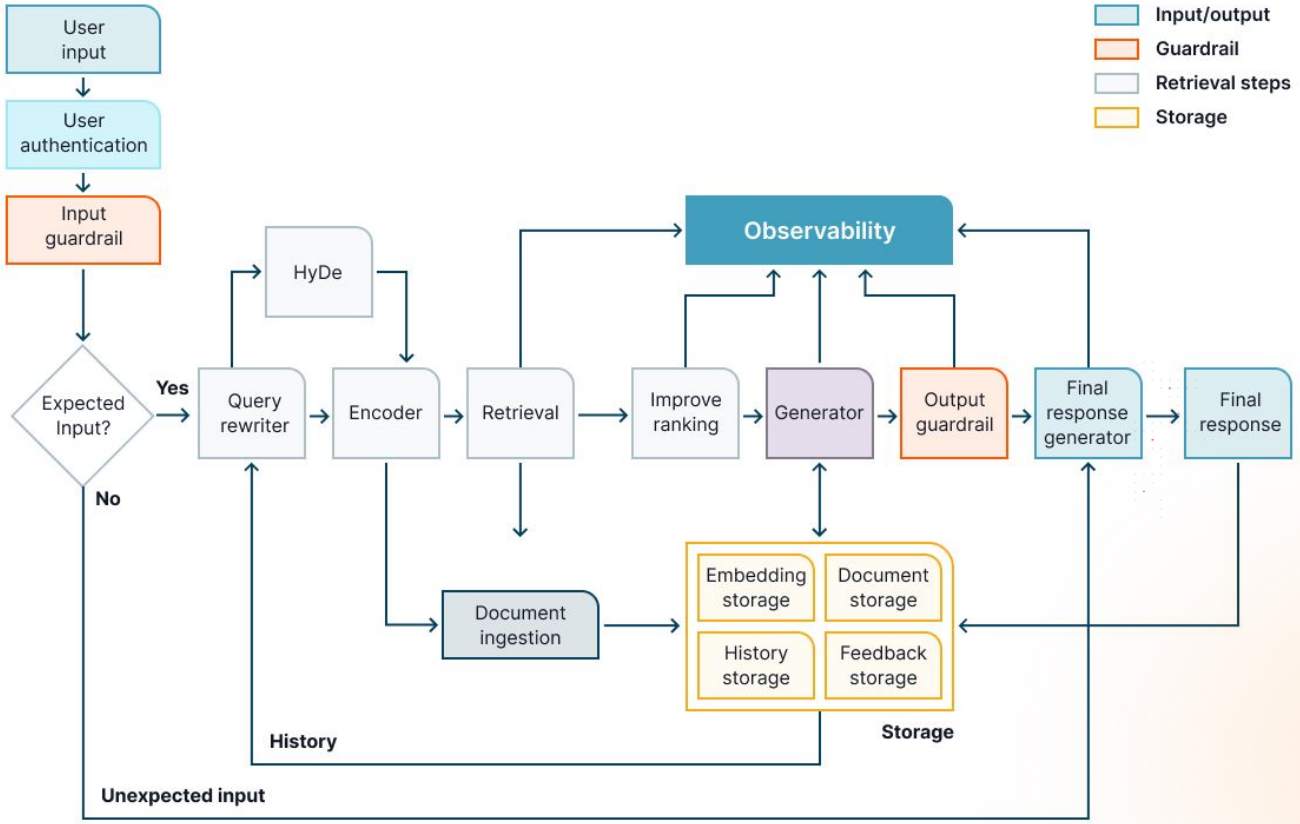
Conducts a single pass that may miss important data

Challenges with Basic RAG Application



Source

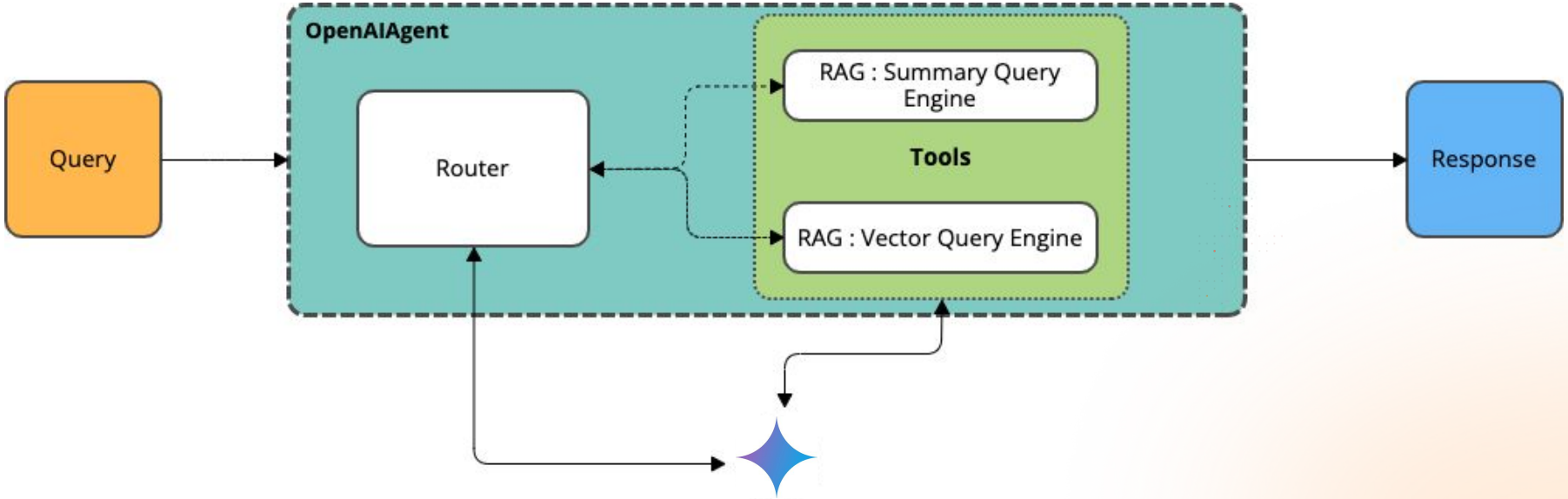
Better Solution: Enterprise and Agentic RAG



Source

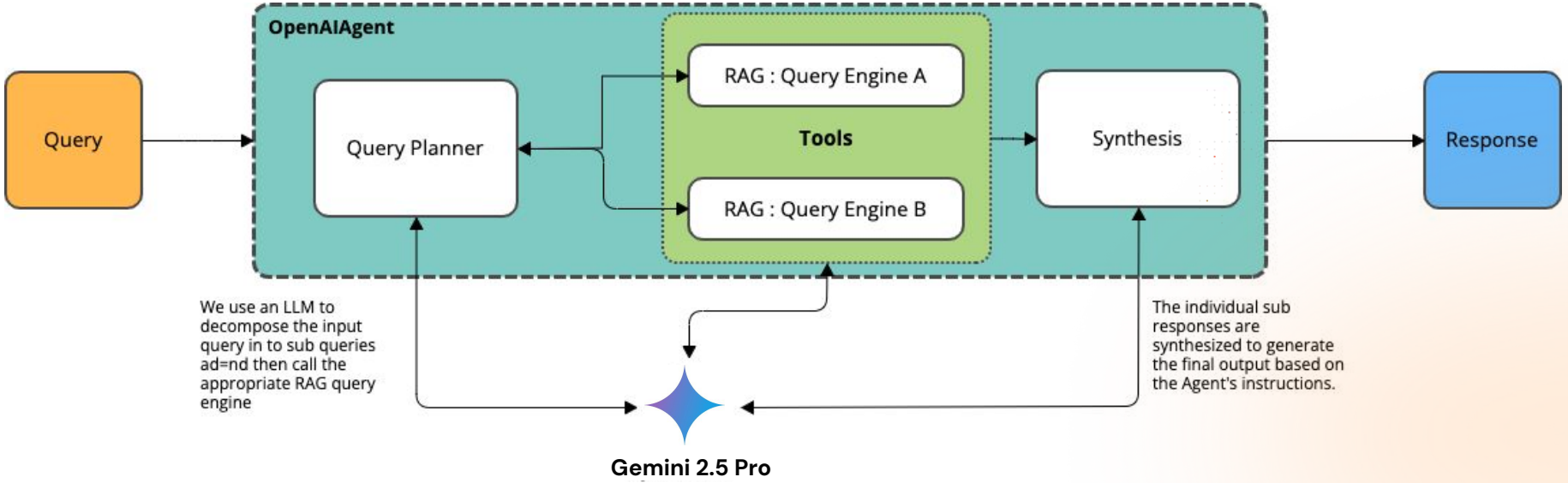
Routing

Simplest form of agentic reasoning that uses uses an LLM to pick what downstream RAG pipeline to pick



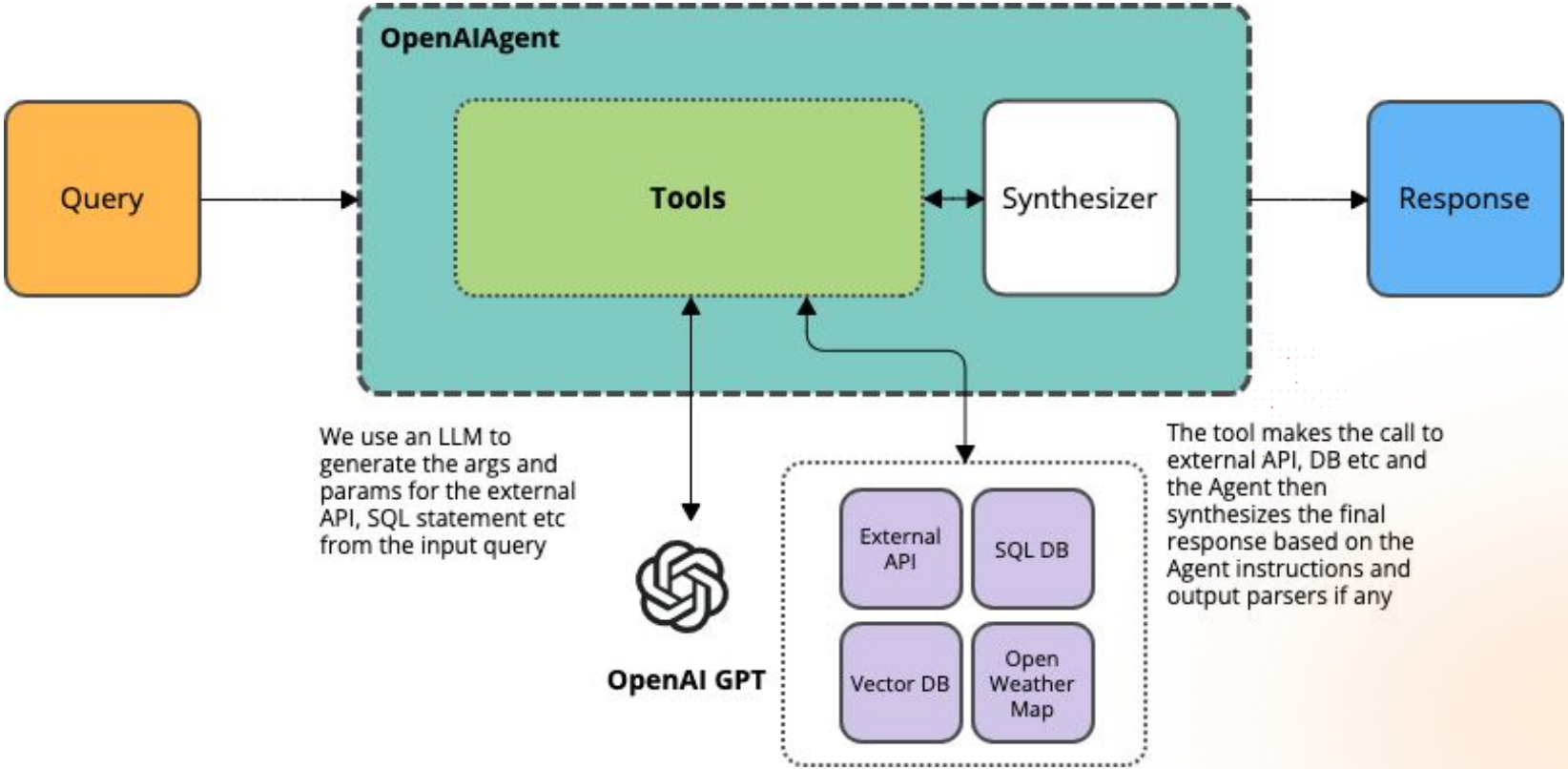
One-Shot Query Planning

Break down query into parallelizable sub-queries. Each subquery can be executed against any set of RAG pipelines. Once the results of the sub queries are generated, they are synthesized in to a final response.



Tool Use

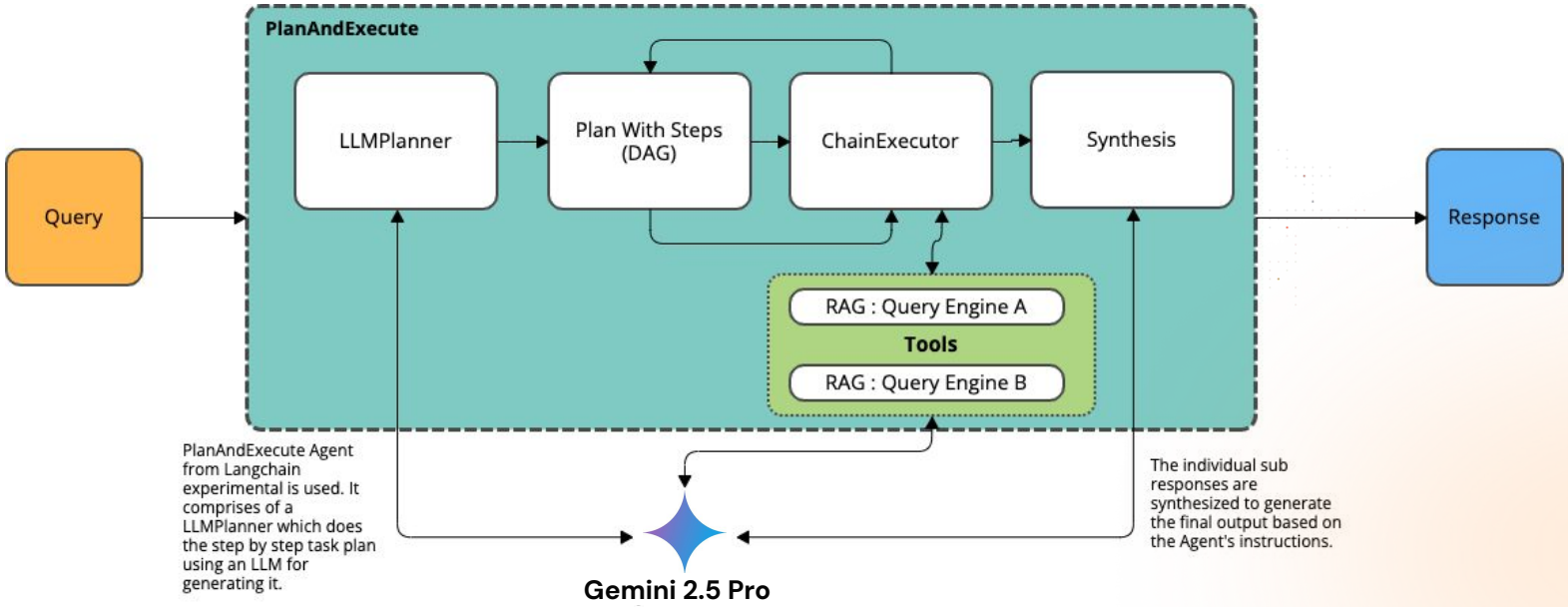
Use an LLM to call an API and Infer the parameters of that API



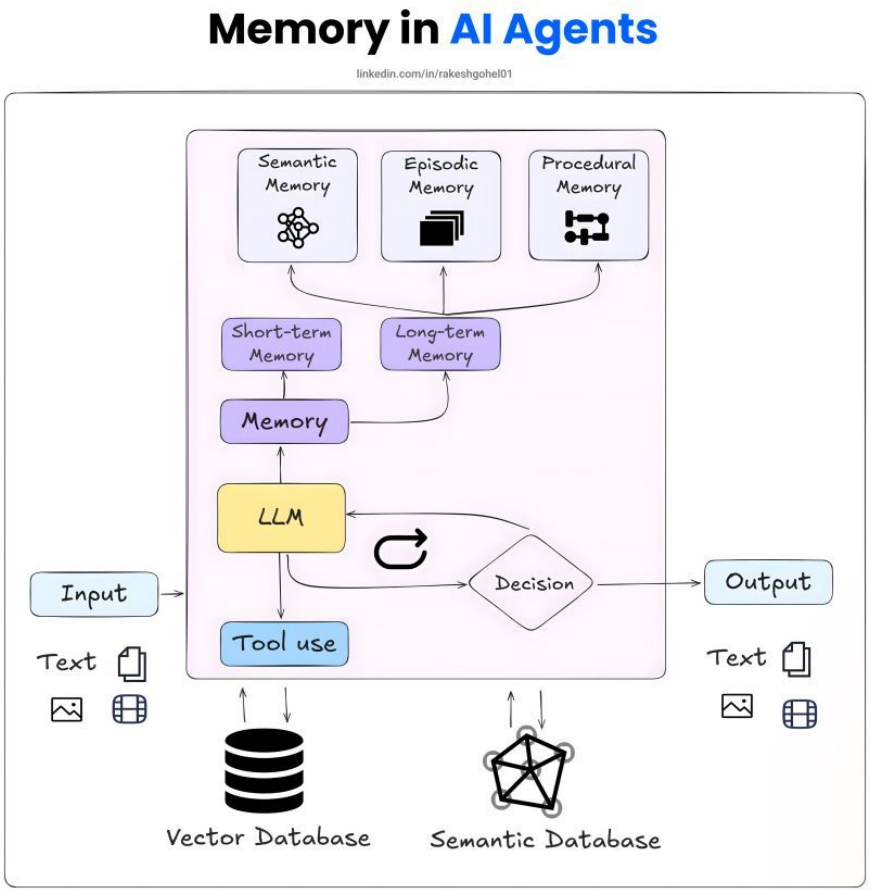
Dynamic Planning & Execution Agent

Planner: Uses an LLM to craft a step-by-step plan based on the user query

Executor: Executes each step, identifying tools needed to accomplish the tasks (outlined in plan). This iterative process continues until entire plan is executed



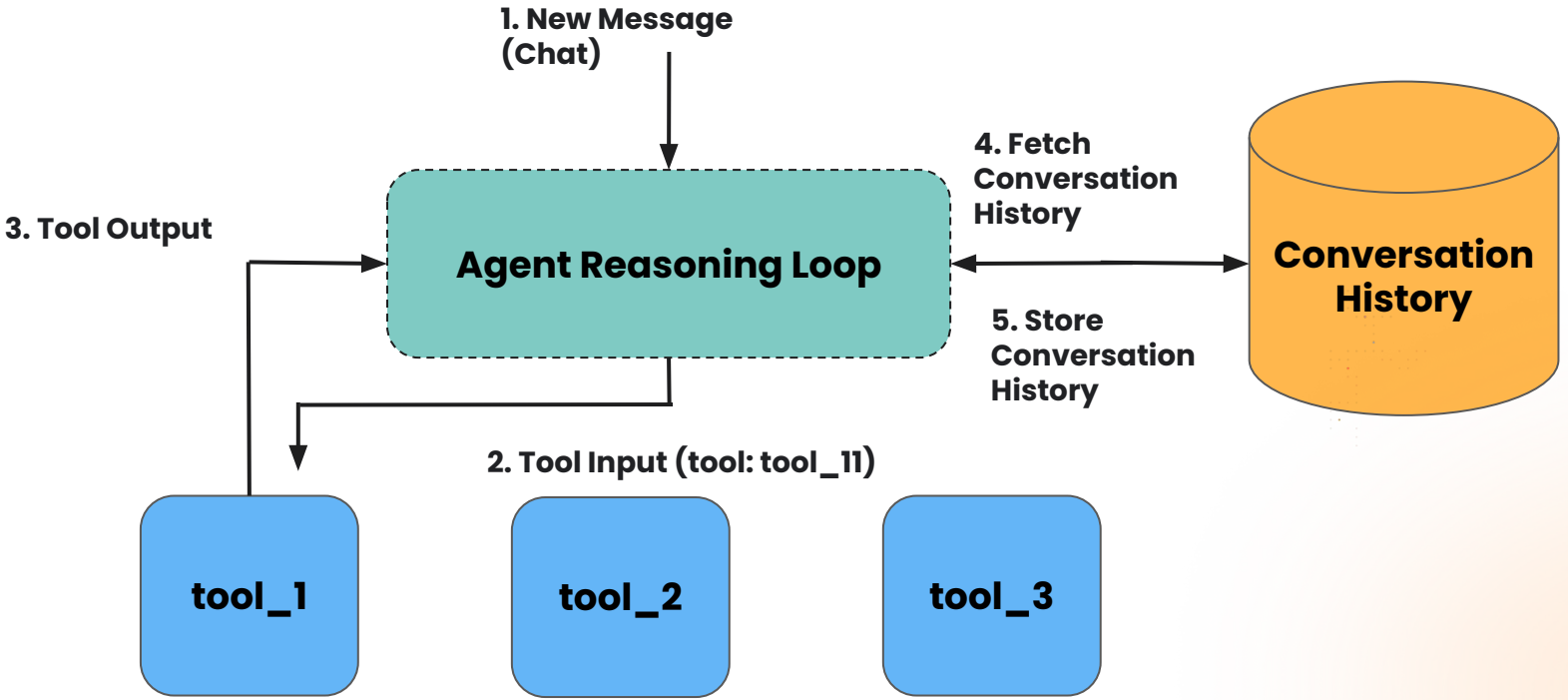
Agentic RAG



CREDITS

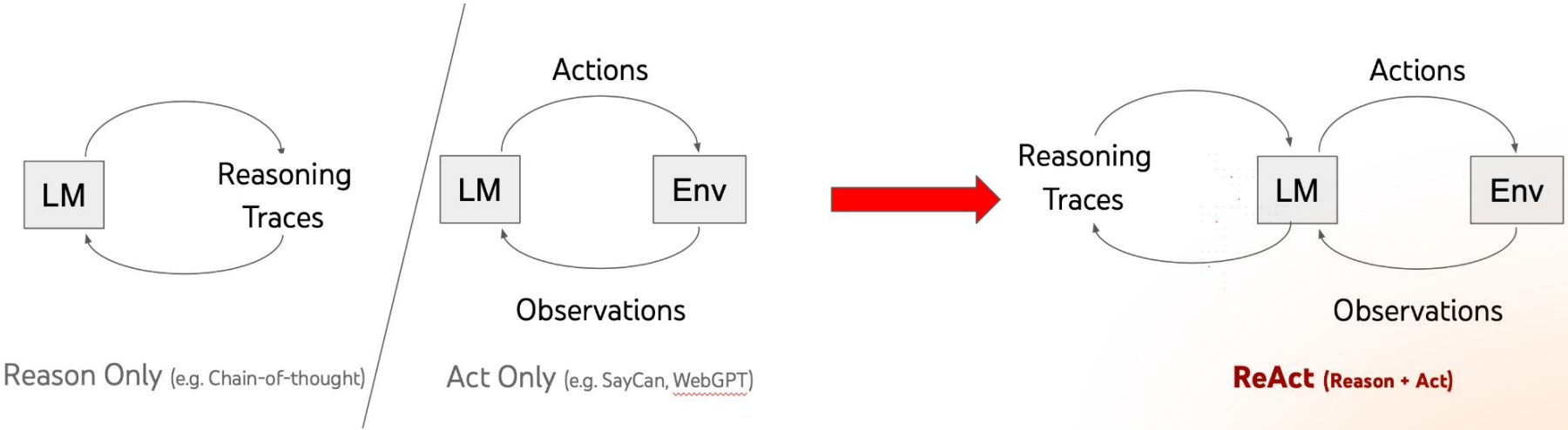
Conversation Memory

The memory is just a flat list of conversations the agent had with the user

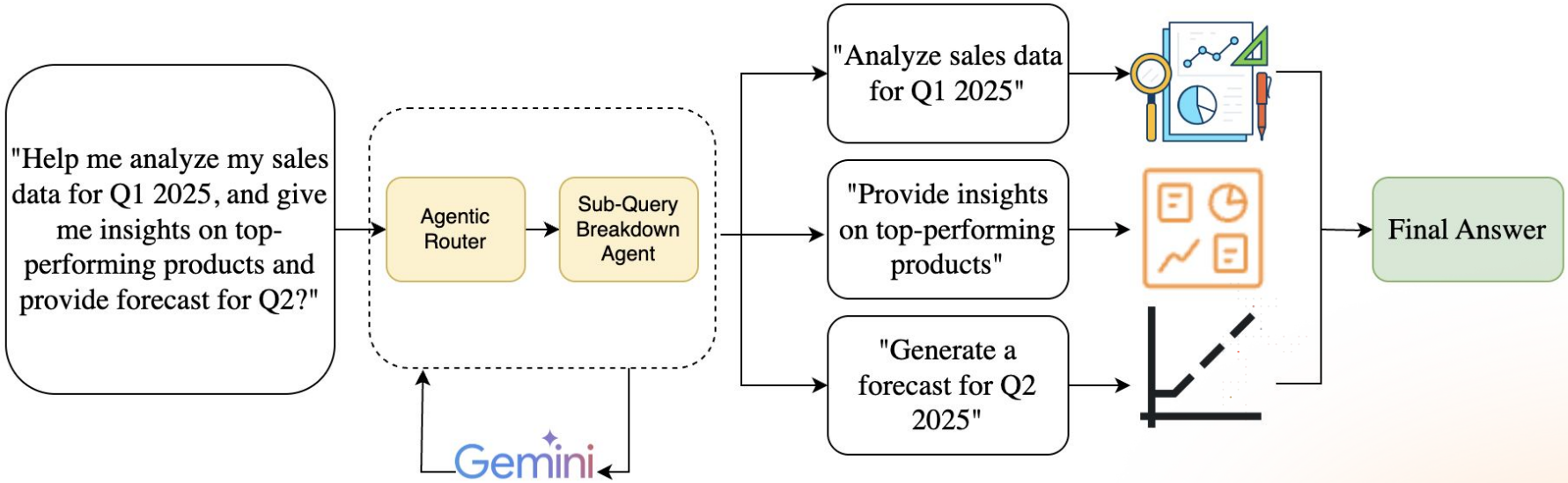


ReAct: Reasoning + Acting with LLMs

It is a superset of Routing, Query Planning and Tool Use all rolled into one. A ReAct agent can handle sequential multi part query and keep state (in memory)



Complex Query Breakdown in Agentic RAG



Agentic RAG - Demo

[HF-2]Agentic_Rag_Deep_Research_10k.ipynb - Colab

Beyond RAG

Scan to Receive Today's Content

Levels of Agentic Architectures

Level 1: Simple LLM

Level 2: LLMs with Tools

Level 3: LLMs with Reasoning

Level 4: Agent to Agent

Deterministic Workflow and Pre Defined Automation

Autonomy and Self Reflection

Levels of Agentic Architectures

Complexity

Level 4: Agent to Agent

Level 3: LLMs with Reasoning

Level 2: LLMs with Tools

Deterministic Workflow and Pre Defined Automation

Autonomy and Self Reflection

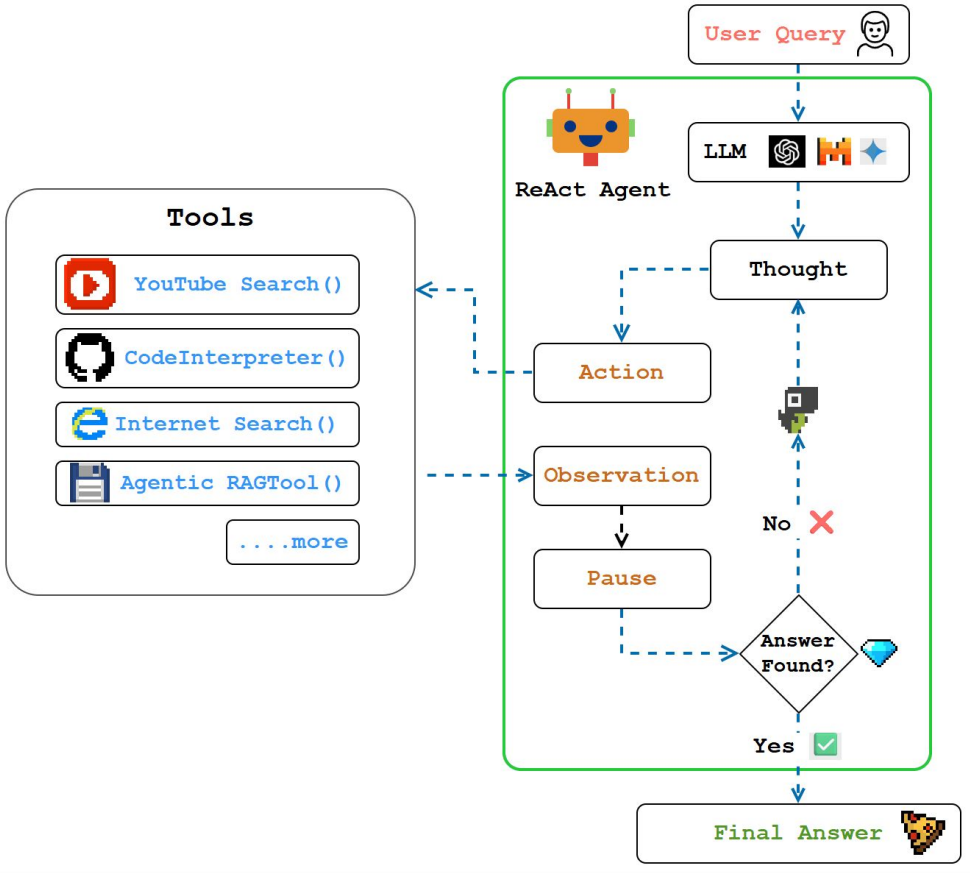
Level 1: Simple LLM

02

Level 3: Agents with Reasoning and Thinking Capability

ReAct Agent

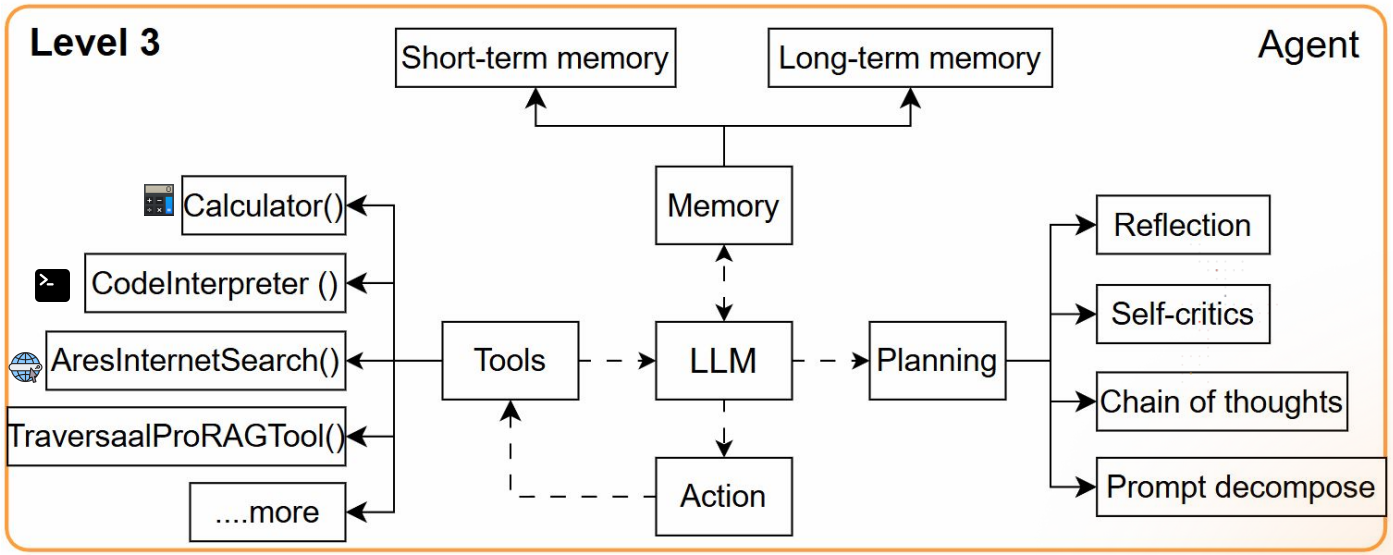
A ReAct agent is an AI agent that uses the "**reasoning and acting**" (ReAct) framework. It combines chain of thought (CoT) reasoning with external tool use.



Example: **AgentPro Demo**

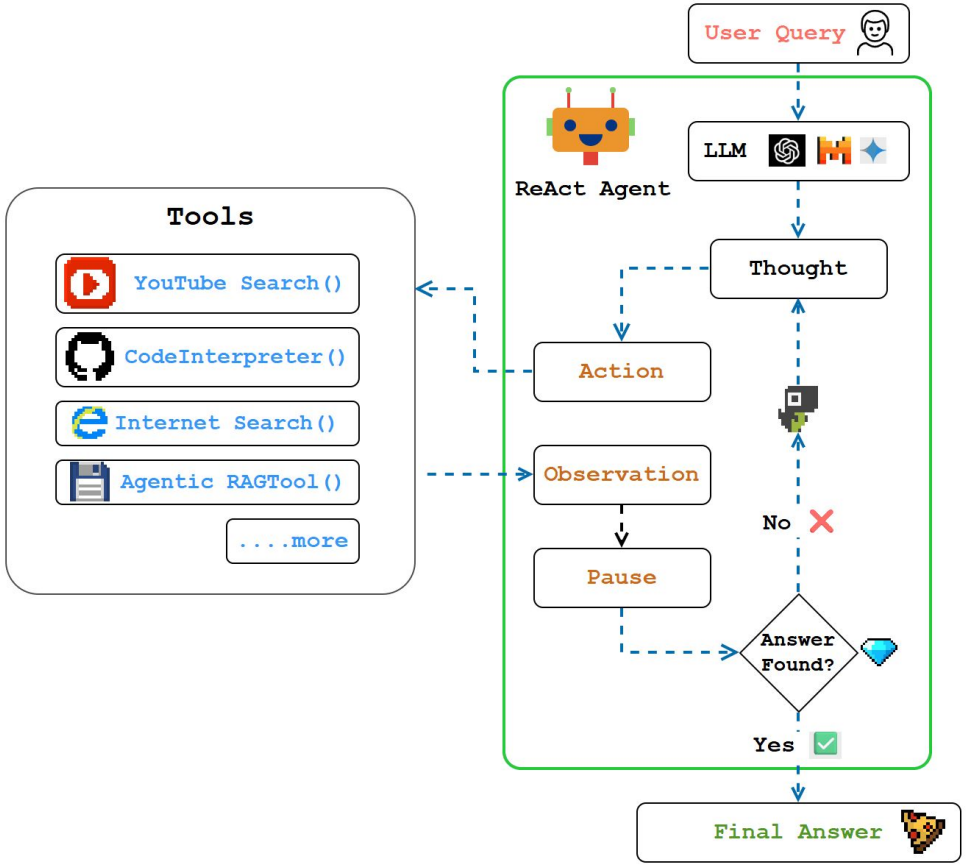
[AgentPro_Traversaal.ipynb - Colab]

Agent Architecture Levels



ReAct AgentPro - Demo

[AgentPro_Traversaal.ipynb - Colab]



AgentPro_Traversaal.ipynb - Colab

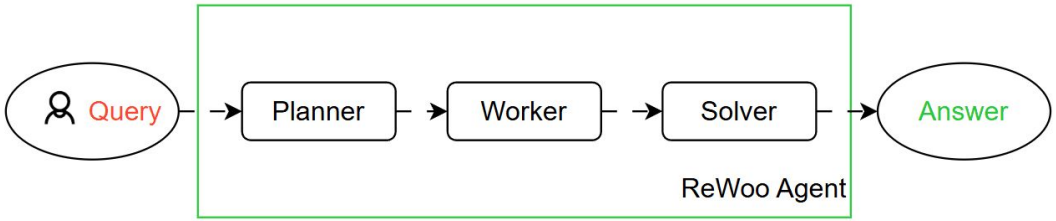
ReWOO

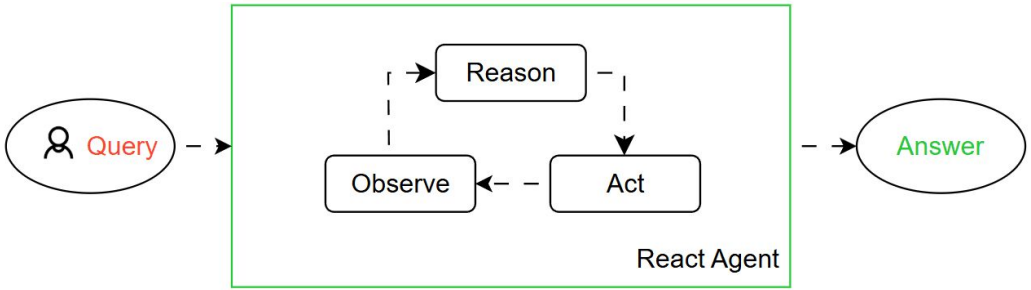
ReWOO (Reason WithOut Observation)

Structured three-component approach:

- **Planner** creates an upfront plan
- **Worker** executes all steps
- **Solver** compiles the final response

ReWOO vs ReACT





ReAct plans and adjusts as it goes step-by-step, while **ReWOO** makes all its plans upfront then executes them faster but can't change course easily.

The Power of Memory in AI Agents

- **Memory = Key to Intelligence:** Helps AI learn context, adapt, and give personalized responses.
- **Two Main Types:** Short-Term (Working Memory) and Long-Term (Procedural, Semantic, Episodic).
- **Today's Goal:** Understand how each type of memory shapes an AI agent's capabilities.

Think of memory as the key to making AI truly intelligent, because it helps them learn context, adapt, and provide personalized responses through short-term (working) memory and the three forms of long-term memory—procedural, semantic, and episodic—so let's explore how each type shapes an AI agent's capabilities.

CREDITS

Short-Term Working Memory

- **Immediate Focus**: Holds info needed for current decisions.
- **Non-Persistent:** Doesn't carry over info across different tasks or tool calls.
- **Information Hub:** Pulls relevant data from long-term memory and external sources to support real-time actions.

STM holds immediate information used for current decisions, without storing it for future tasks.
Example: When an AI assistant processes your recent speech command ("Play jazz music") to choose the correct playlist, it uses STM for those instructions—then discards them after completing the task.

CREDITS

Inside Long-Term Memory

Procedural Memory

- Stores *how* to do things (steps, algorithms, rules).
- Think of it like “muscle memory” for an AI—once it learns the steps, it can automatically perform tasks without being explicitly re-taught.

Semantic Memory

- Contains *what* knowledge: facts, concepts, and relationships.
- Helps the AI figure out logical connections—like relating places on a map, understanding word meanings, or linking facts in a knowledge graph.

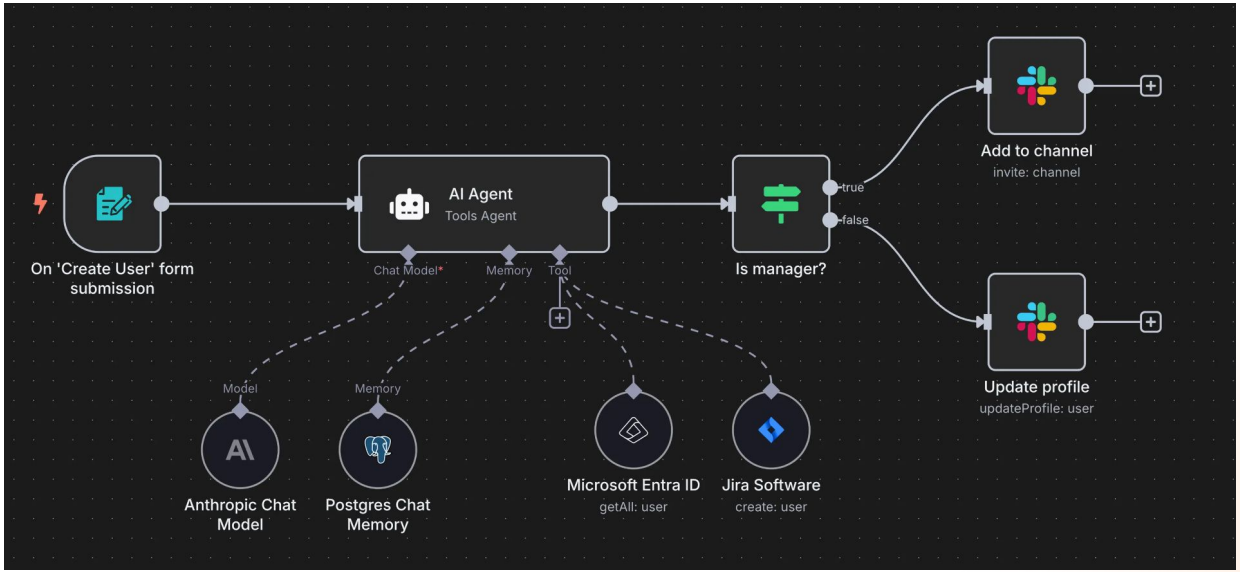
Episodic Memory

- Recalls *when* and *where* specific events happened (past user interactions, experiences).
- Lets the AI remember your preferences or previous questions to personalize current suggestions.

03: Integrating Front End Interfaces with n8n (cont..)

Recap: What is n8n?

A workflow automation tool that lets you build automated processes without writing a ton of code. Think of a digital assistant that glues your apps together, handling tasks like data syncing, alerts, and even multi-step business processes automatically.



Source

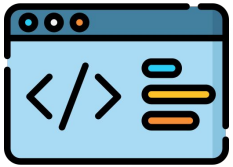
Recap: Types of Nodes

Trigger Nodes

Action Nodes

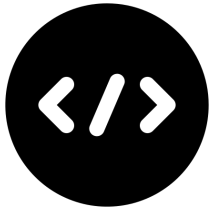
Utility Nodes

AI Nodes



Code Nodes

Types of Nodes (Advanced)



Output Parser Node

Webhook Node

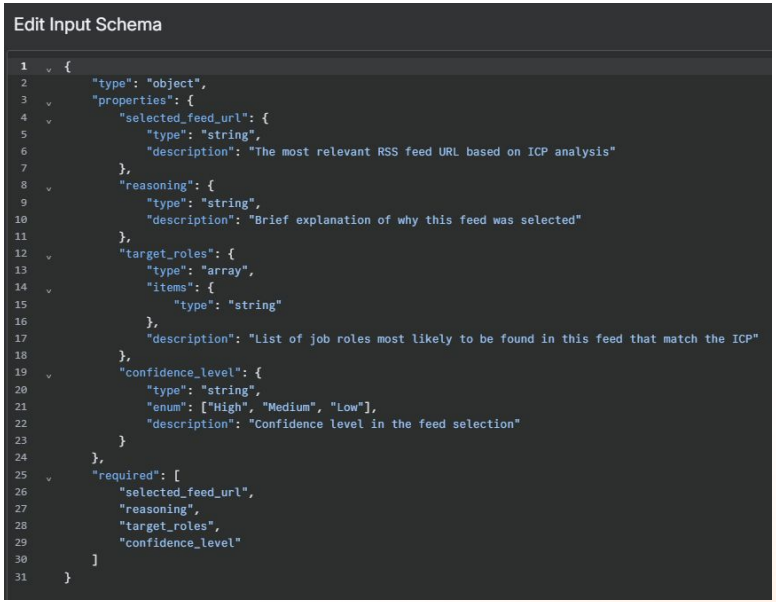
HTTP API Node

Structured Output Parser Node

This node is used within your n8n workflow to transform unstructured or semi-structured text (especially from LLM responses) into organized, machine-readable data.

How it helps:

- Prepare AI Output for Systems
- Extract Key Info
- Normalize Data

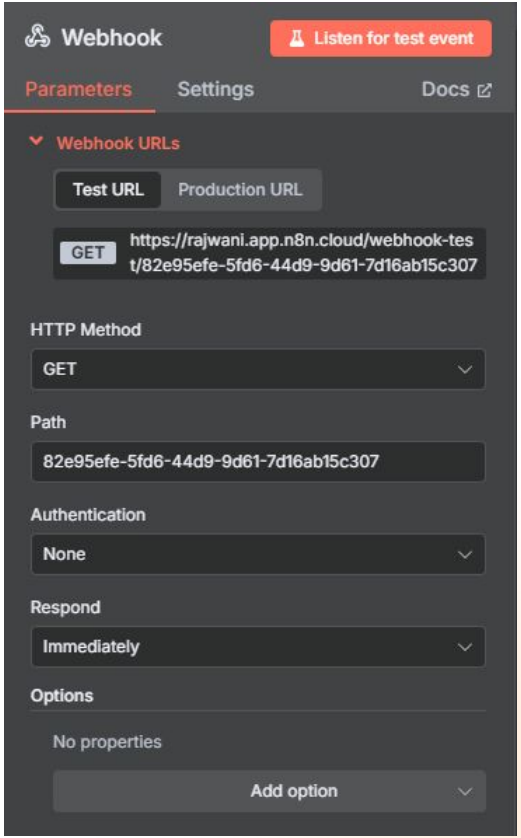


Webhook Node

A specialized Trigger Node that acts as a "listener," providing a unique URL for other services to send data to.

Key Use Cases:

- Event-Driven Automation
- Real-time Data Intake
- Integrating Custom Services: Connect n8n with any service capable of sending webhook notifications.

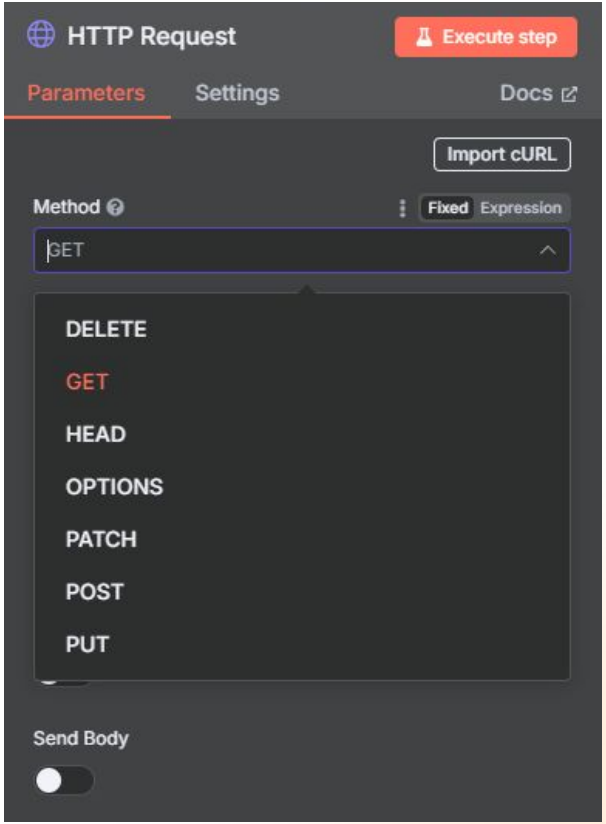


HTTP API Node

This versatile node lets your n8n workflow send and receive data from virtually any web-based API. It's your bridge to the entire internet.

Key Request Types:

- **GET**: Retrieve information (e.g., fetch product details).
- **POST**: Send new data (e.g., create a new user account).
- **PUT/PATCH**: Update existing information (e.g., modify an order status).
- **DELETE**: Remove data (e.g., delete an old record).



Thank you!

Hamza Farooq
hamza@traversaal.ai