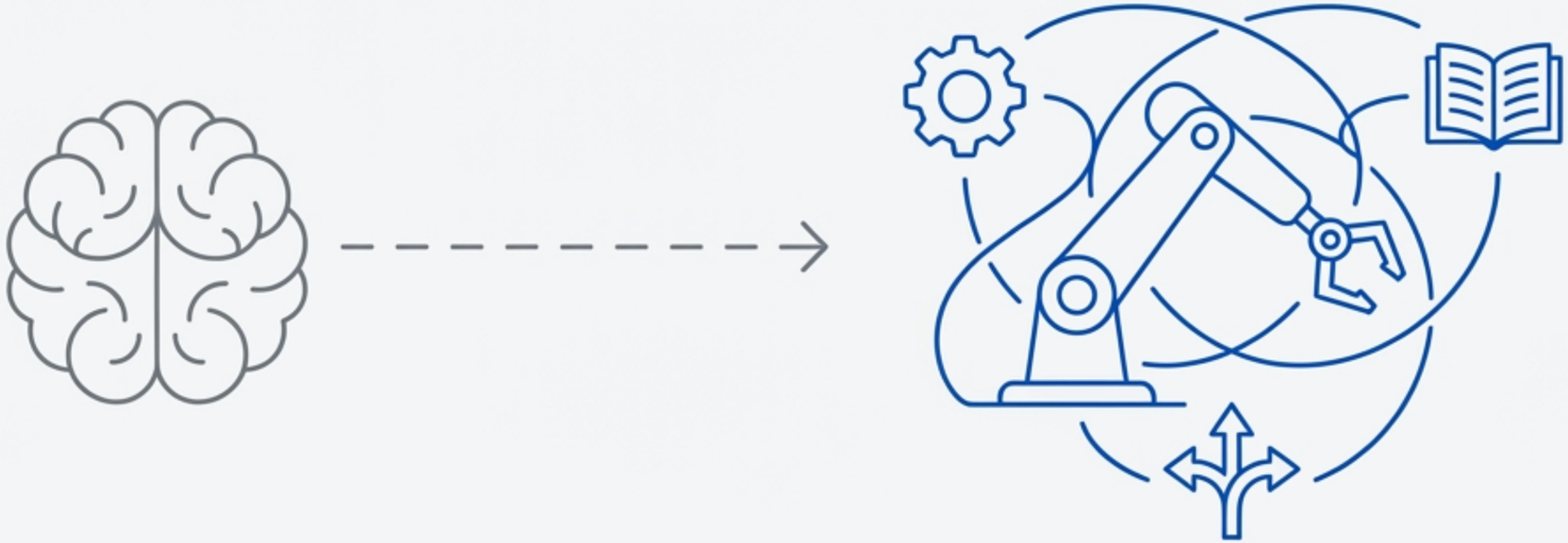
Multi-Agent Deep RAG Course

From LLMs to Autonomous Agents

Agent Foundation



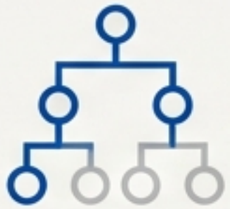
What You Will Learn in This Section

1. The Critical Distinction

Understand the fundamental difference between Large Language Models and AI Agents.

2. The Engine of Autonomy

Deconstruct the agent's core operational cycle: The Think → Tool → Think → Answer loop.



3. A Taxonomy of Agents

Explore the four primary types of AI Agents and their specific use cases.

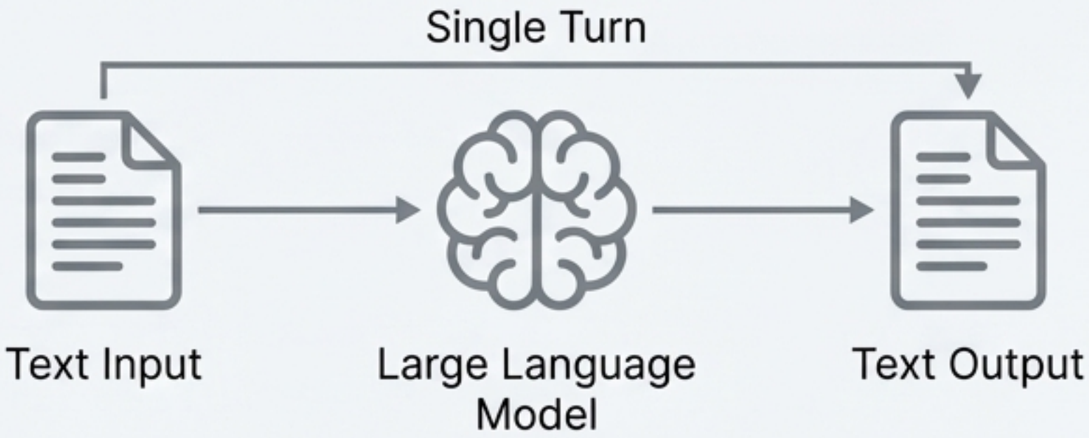
4. Your First Build

See how an agent is built using practical tools like a calculator and web search.

The Starting Point: The Large Language Model as a Passive Responder

Key Characteristics

- **Nature:** Passive responder.
- **Capability:** Text in → Text out.
- **Behaviour:**
  - Predicts the next token autoregressively.
  - Provides a single-turn response.
  - Has no memory between conversations.
  - Cannot take actions or use external tools.



Example

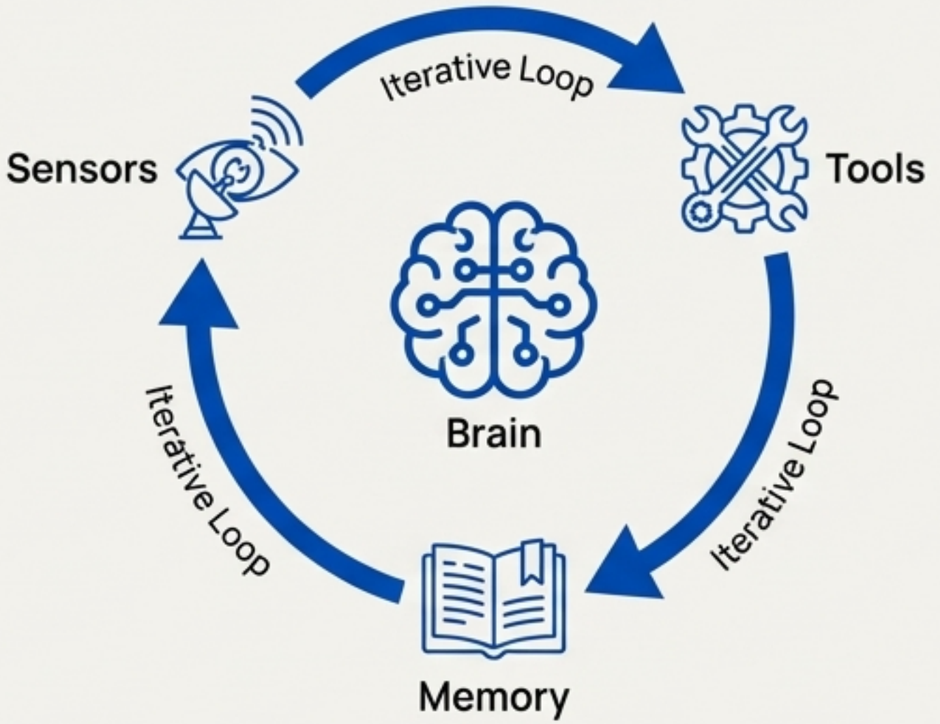
Query: What's 127 × 83?

LLM Process: Generates an answer based on pattern recognition from its training data.

The Evolution: The AI Agent as an Autonomous Actor

Key Characteristics

- **Nature**: Autonomous actor.
- **Capability**: Perceives + Acts + Reasons.
- **Behaviour**:
  - Uses an LLM as its ‘brain’ for reasoning.
  - Engages in a multi-turn, iterative process.
  - Maintains memory and context.
  - Takes actions in an environment using external tools.



Example

Query: What’s 127 × 83?
Agent Process: Decides to use a calculator tool to compute the exact answer.

The Core Insight: An Agent Augments a Reasoning Engine with Action Capabilities

An Agent = LLM (brain) + Memory + Tools + Planning/Reflection

---

The LLM provides the reasoning, while the agent framework provides the ability to act.

Anatomy of an Agent: The Core Components of Action

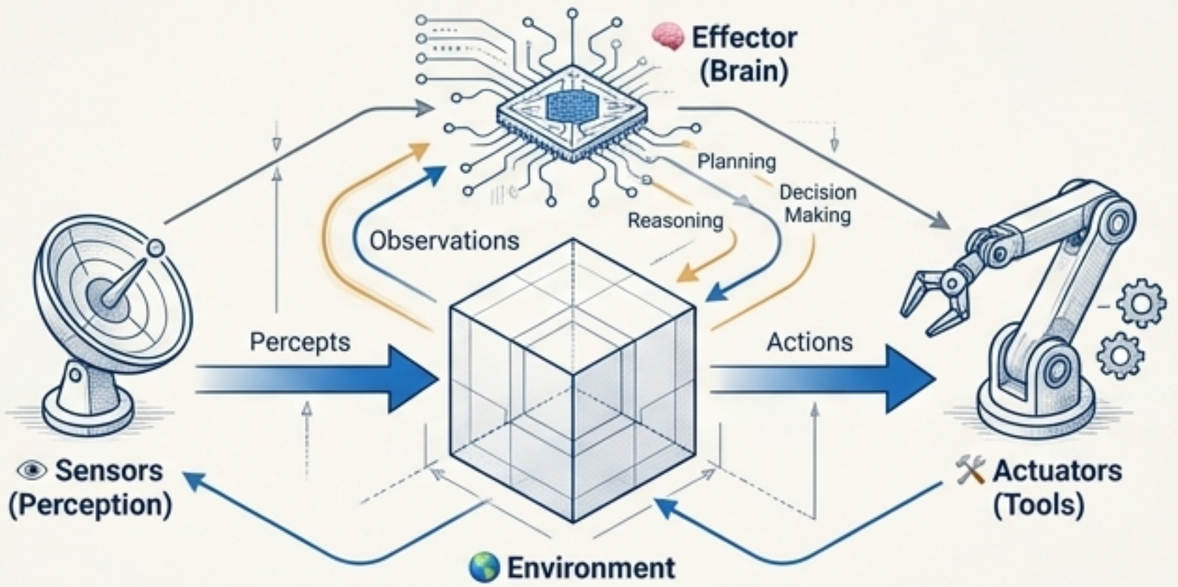
"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators." — Russell & Norvig, *AI: A Modern Approach*

🧠 Effector (Brain)

The reasoning LLM that decides actions, processes observations, and plans next steps.

📡 Sensors (Perception)

Multimodal inputs like user queries, context, tool outputs, and environmental state.



🛠️ Actuators (Tools)

External APIs, databases, file systems, web search, calculators, and code execution environments.

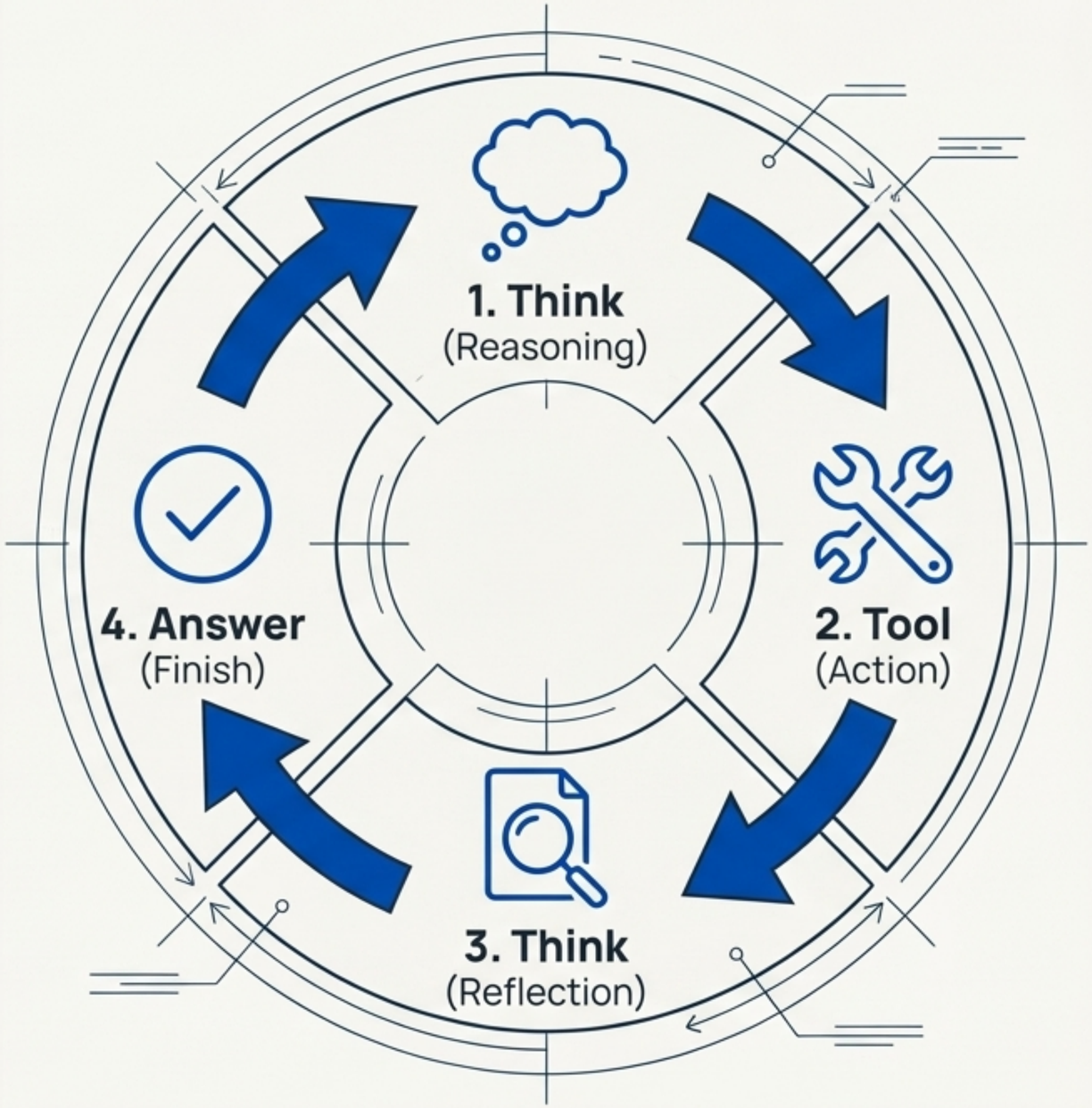
🌍 Environment

The digital or physical world where the agent operates, including user interactions, data sources, and other agents.

The Agent's Engine: The ReAct Pattern Powers the Loop of Thought and Action

Agents follow the **ReAct pattern** (Reasoning + Acting).

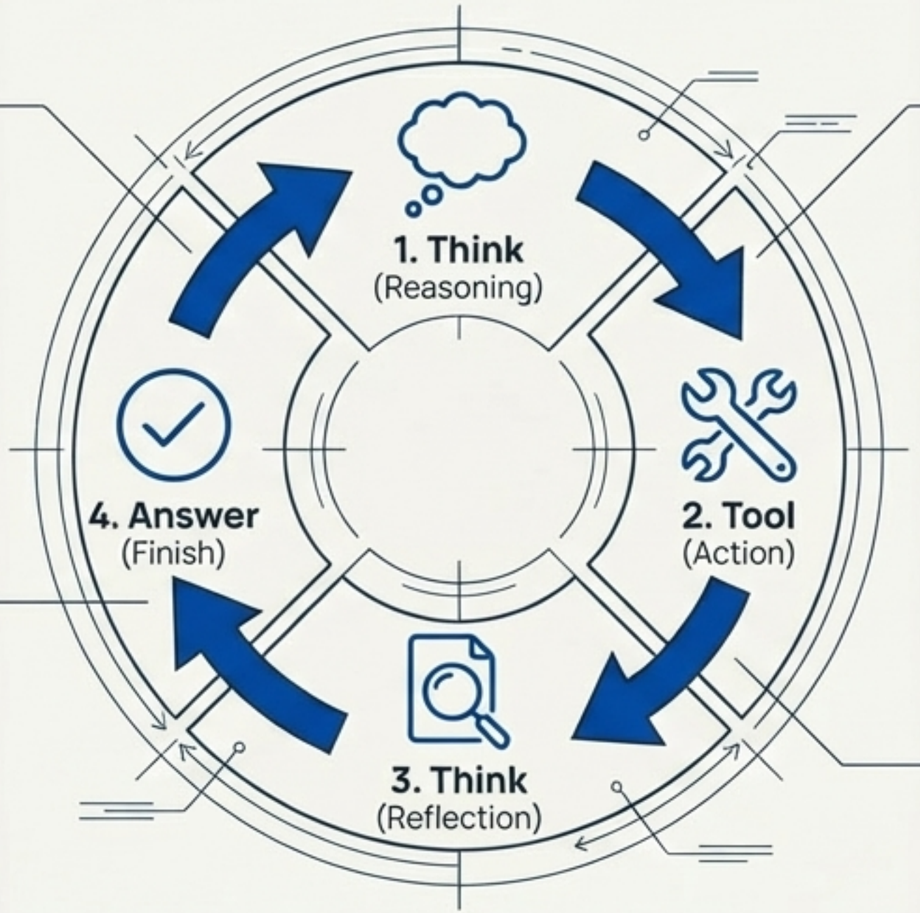
- **Iterative**: The loop can repeat multiple times.
- **Dynamic**: The agent decides when to stop based on goal completion.
- **Autonomous**: No human intervention is needed during the loop (unless configured).



ReAct in Action: Solving a Multi-Step Query

User Query: What is 127 × 83 and is it a prime number?

1. 🤔 **Think (Reasoning):** The LLM analyses the query and determines it needs to perform a calculation first.



2. 🛠️ **Tool (Action):** The agent selects and executes the calculator tool.

```
Action: calculator(127 × 83)
Observation: 10,541
```

3. 🔍 **Think (Reflection):** The agent observes the result and reasons about the next step.

```
Reasoning: "Now I need to check if 10,541 is prime"
```

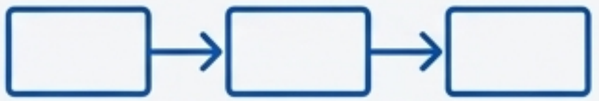
4. ✅ **Answer (Finish):** After a potential second tool use (e.g., a primality test function), the agent synthesises the information into a final response.

```
Result: "127 × 83 = 10,541. Based on my analysis, it is a prime number."
```

A Taxonomy of Agents: Four Patterns for Different Tasks

While ReAct is the most common pattern, specialised agents are designed for more complex, diverse, or critical tasks. The choice of agent architecture directly impacts performance and reliability.

ReAct Agent



Plan-Execute Agent

Multi-Tool Agent

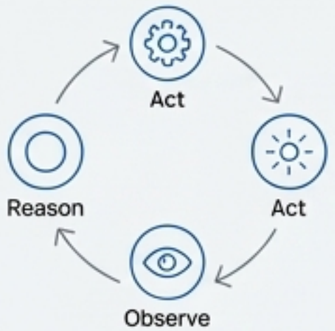
Self-Corrective Agent

Agent Architectures: Foundational and Strategic Patterns

ReAct Agent

Pattern

Reason → Act → Observe



Key Trait

Alternates between reasoning and acting; simple and effective.

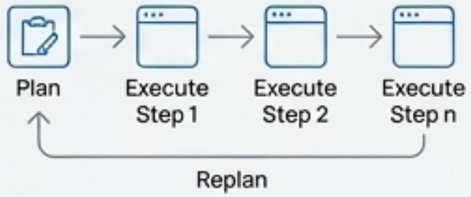
Best Use Case

Question answering, data retrieval, simple calculations.

Plan-Execute Agent

Pattern

Plan → Execute Steps → Replan



Key Trait

Creates a complete plan upfront and executes sequentially; can revise the plan.

Best Use Case

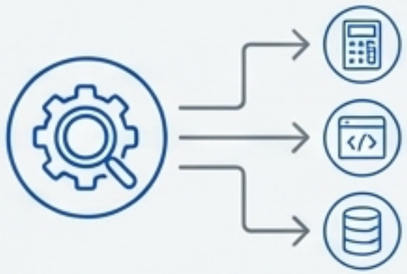
Multi-step research, report generation, complex workflows.

Agent Architectures: Specialised and High-Reliability Patterns

Multi-Tool Agent

Pattern

Tool Selection → Parallel Execution



Key Trait

Accesses multiple specialised tools and dynamically selects the right ones, sometimes in parallel.

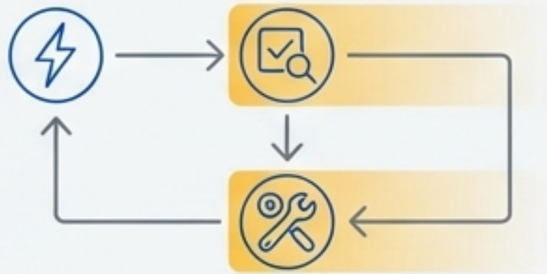
Best Use Case

Data analysis, content creation, system automation.

Self-Corrective Agent

Pattern

Act → Validate → Correct → Re-execute



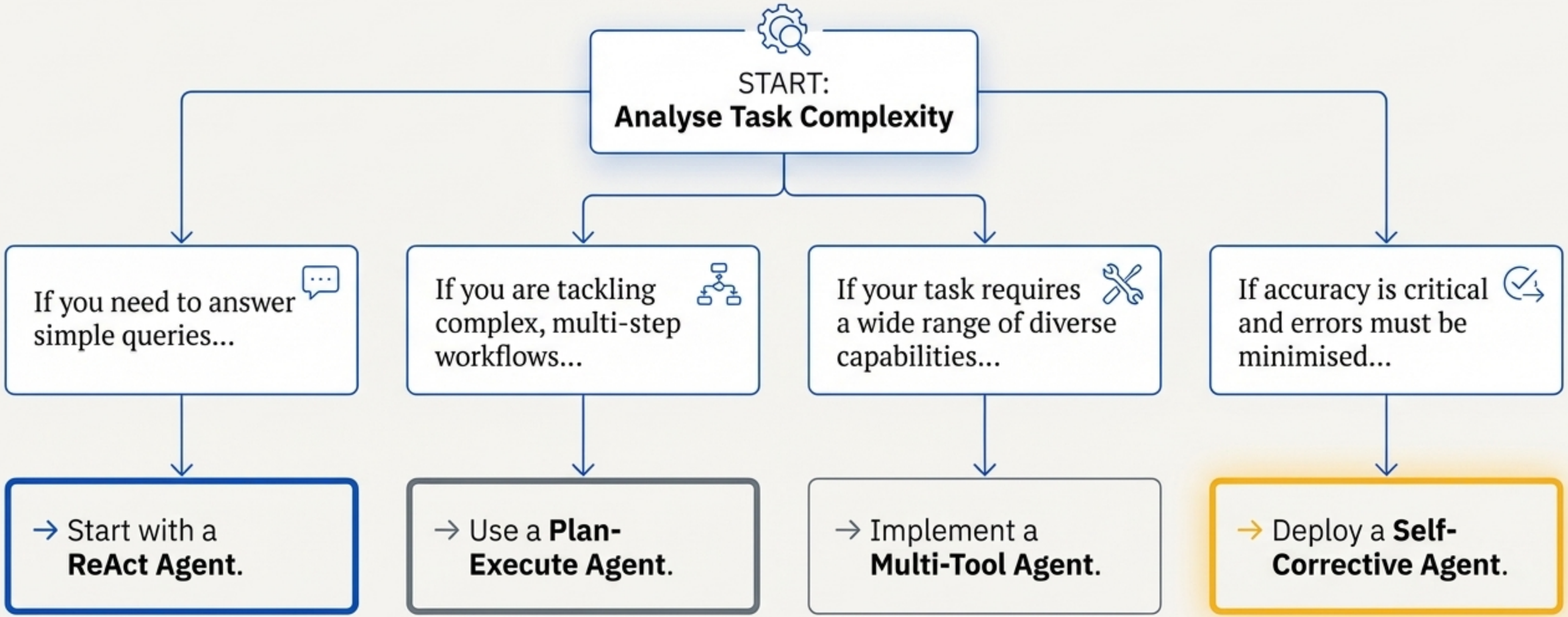
Key Trait

Validates its own outputs, detects and fixes errors, and iteratively improves responses.

Best Use Case

Code generation, SQL queries, critical decision-making.

How to Choose the Right Agent for Your Task

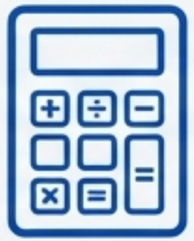


From Theory to Practice: Building an Agent with Calculator and Search Tools

Demo Objectives

- ✓ Create a simple ReAct agent with LangChain.
- ✓ Integrate two distinct tools: a calculator and a web search API.
- ✓ Observe the agent's autonomous decision-making process.
- ✓ See the Think → Tool → Think → Answer loop in a real execution trace.

The Tools We'll Use



Calculator Tool

For precise mathematical computations when a query involves complex calculations or large numbers.

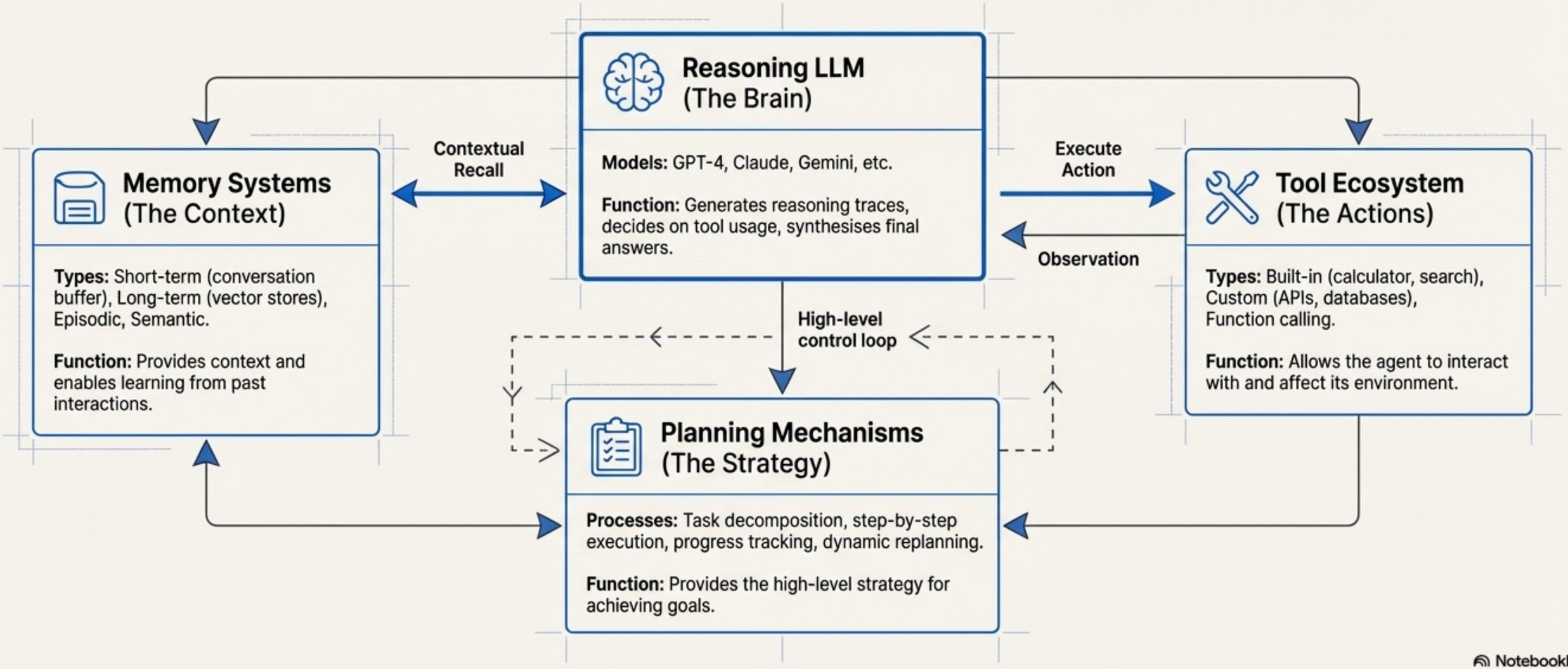
```
e.g., "What is 2847 × 392?"
```

Search Tool

For real-time information retrieval when a query involves current events, news, or factual lookups.

```
e.g., "Who won the latest Nobel Prize?"
```

The Complete Agent Blueprint: Integrating Components for Autonomy

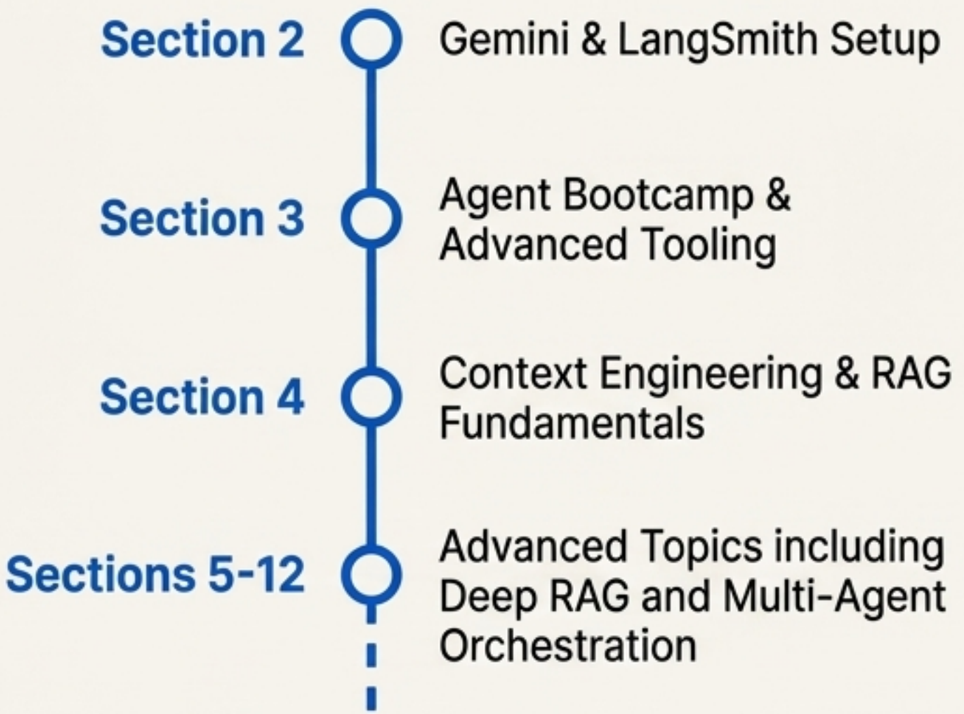


Foundation Built: Key Takeaways and Your Path Forward

Section Summary

- **LLM vs Agent:** Agents are not just LLMs; they are autonomous systems that augment LLMs with memory, tools, and planning.
- **The Agent Loop:** The iterative ReAct pattern (Think → Tool → Think) is the core engine of agent autonomy.
- **Agent Types:** Different agent architectures (ReAct, Plan-Execute, etc.) are suited for different tasks.
- **Core Components:** An agent perceives (Sensors), reasons (Effector/Brain), and acts (Actuators/Tools).

What's Next in the Course



Ready to build? Let's proceed to Section 2. 🚀