

Introduction to LangGraph

Instructor

Lucas Soares

AI Engineer at Otovo

Specialist in LLM Applications & Computer Vision

Instructor at O'Reilly Media

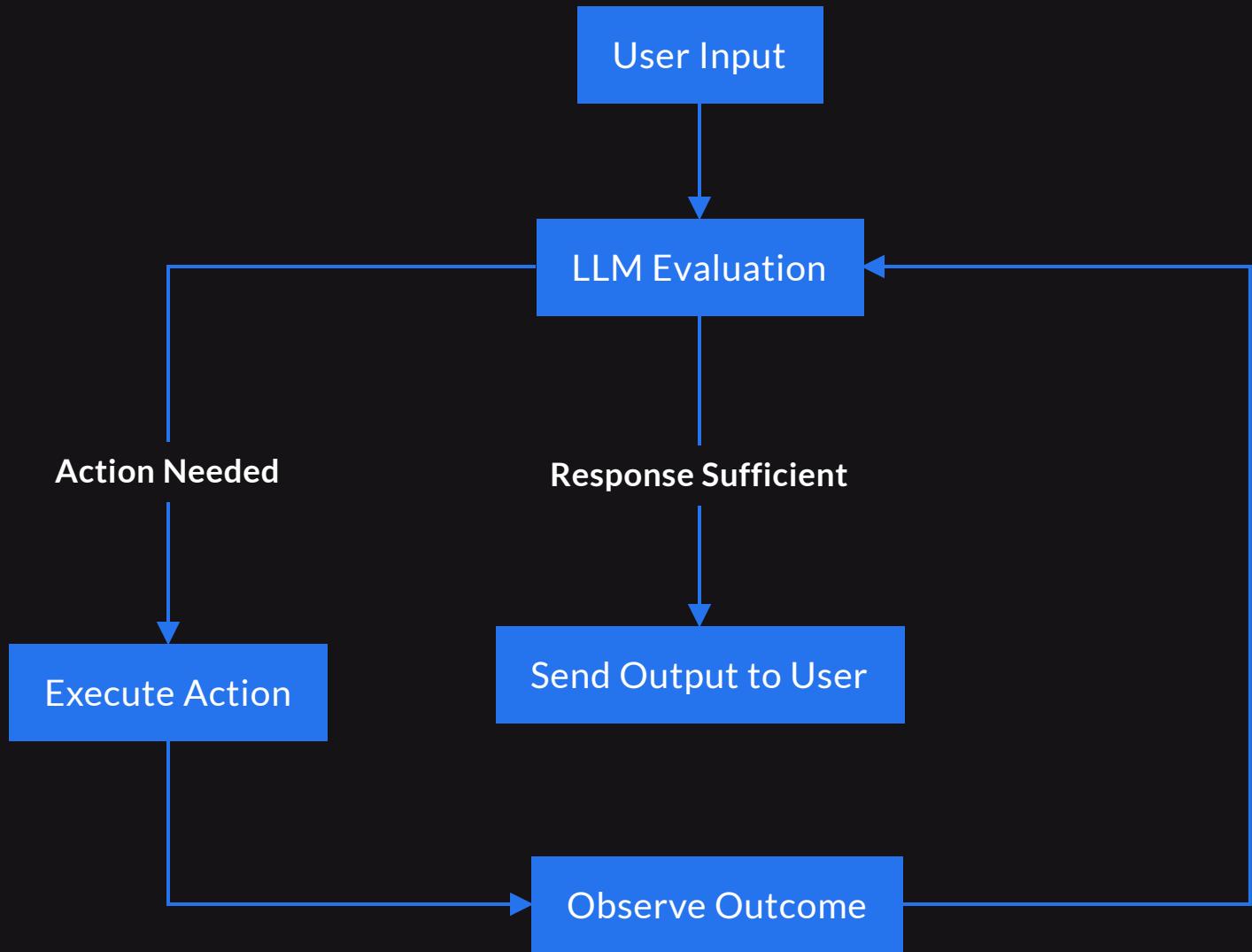
Technical Writer & Content Creator



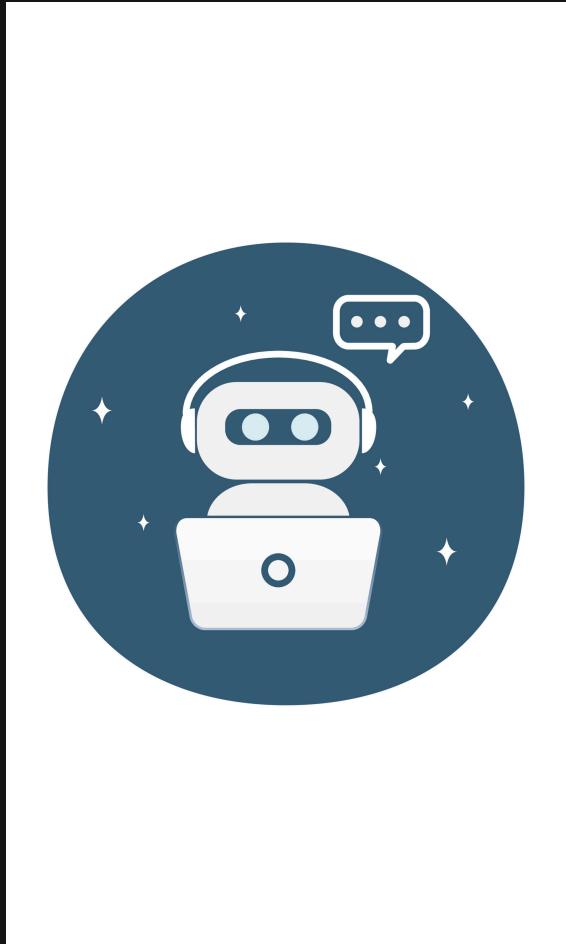
Understanding Agentic Systems

- **Agentic Systems:**
Systems that utilize LLMs to manage the control flow of applications
- **Key Functions:**
Routing decisions, tool selection, and evaluating output sufficiency
- **Agent Loop:**
Continuous decision-making process that enables agents to solve complex tasks

The Agent Loop



Practical Use Case: Customer Support Agent



Scenario:

An LLM-powered customer support agent.

User Input:

Customer asks about order status.

LLM Decision:

Determines if it can provide the status directly or if it needs to fetch data from the database.

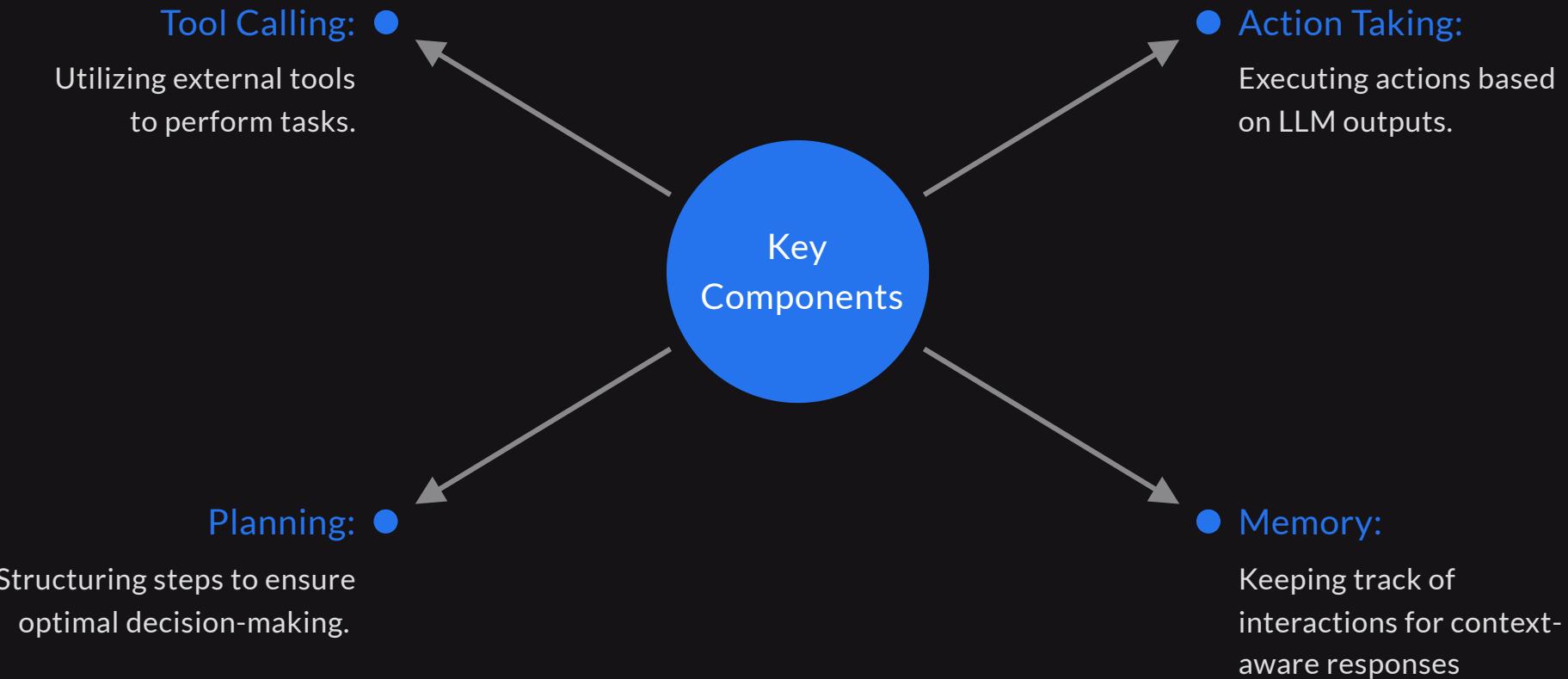
Action Taken:

If data fetch is needed, the agent queries the database and updates the user with the order status.

Advantages of LLM Agents

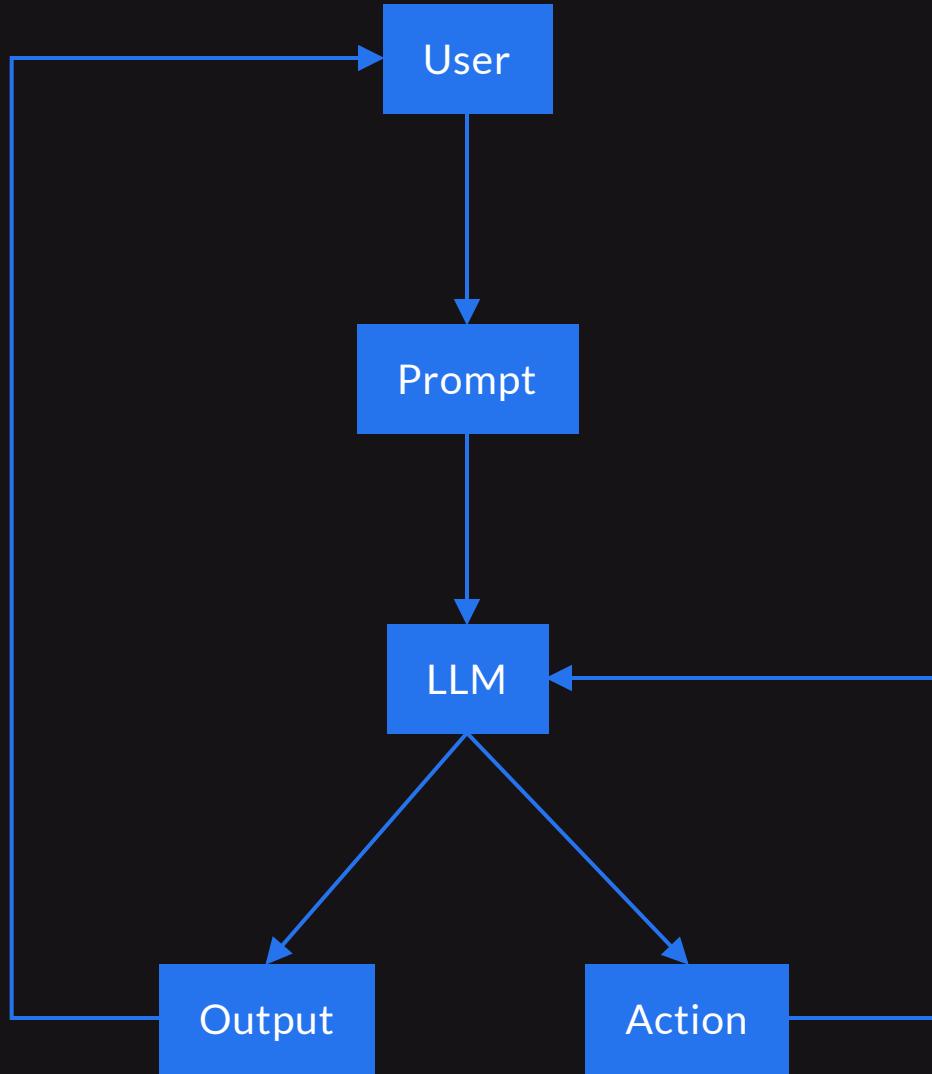
- **Flexibility:**
 - Agents can adapt to various tasks by determining the best action to take.
- **Specialization:**
 - LLM agents can be specialized with tools to perform niche tasks.
- **Multi-Agent Collaboration:**
 - Specialized LLM agents can collaborate to perform complex tasks.

Key Components of Agentic Systems



Agent as Graphs

Workflows build with agents
are usually structured as
graphs



Why LangGraph?

LangGraph is designed for building agentic applications with some core principles:

- **Controllability:**

Offers low-level control which increases reliability in agentic systems.

- **Human-in-the-Loop:**

Built-in persistence layer enhances human-agent interaction patterns.

- **Streaming First:**

Supports streaming of events and tokens, providing real-time feedback to users.

Thank You
