Analytics Vidhya

Why LangGraph?

Dipanjan Sarkar

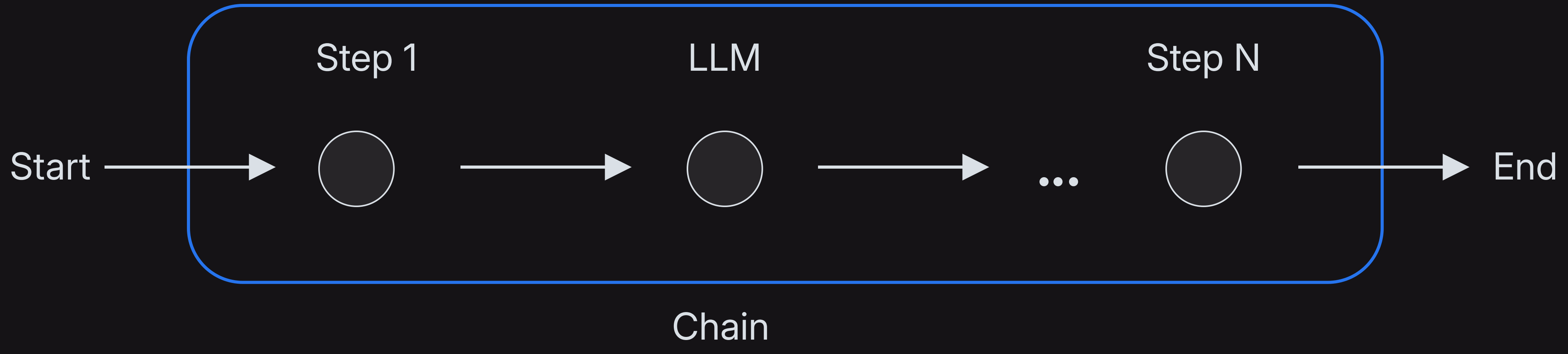
Head of Community & Principal AI Scientist at Analytics Vidhya

Google Developer Expert - ML & Cloud Champion Innovator

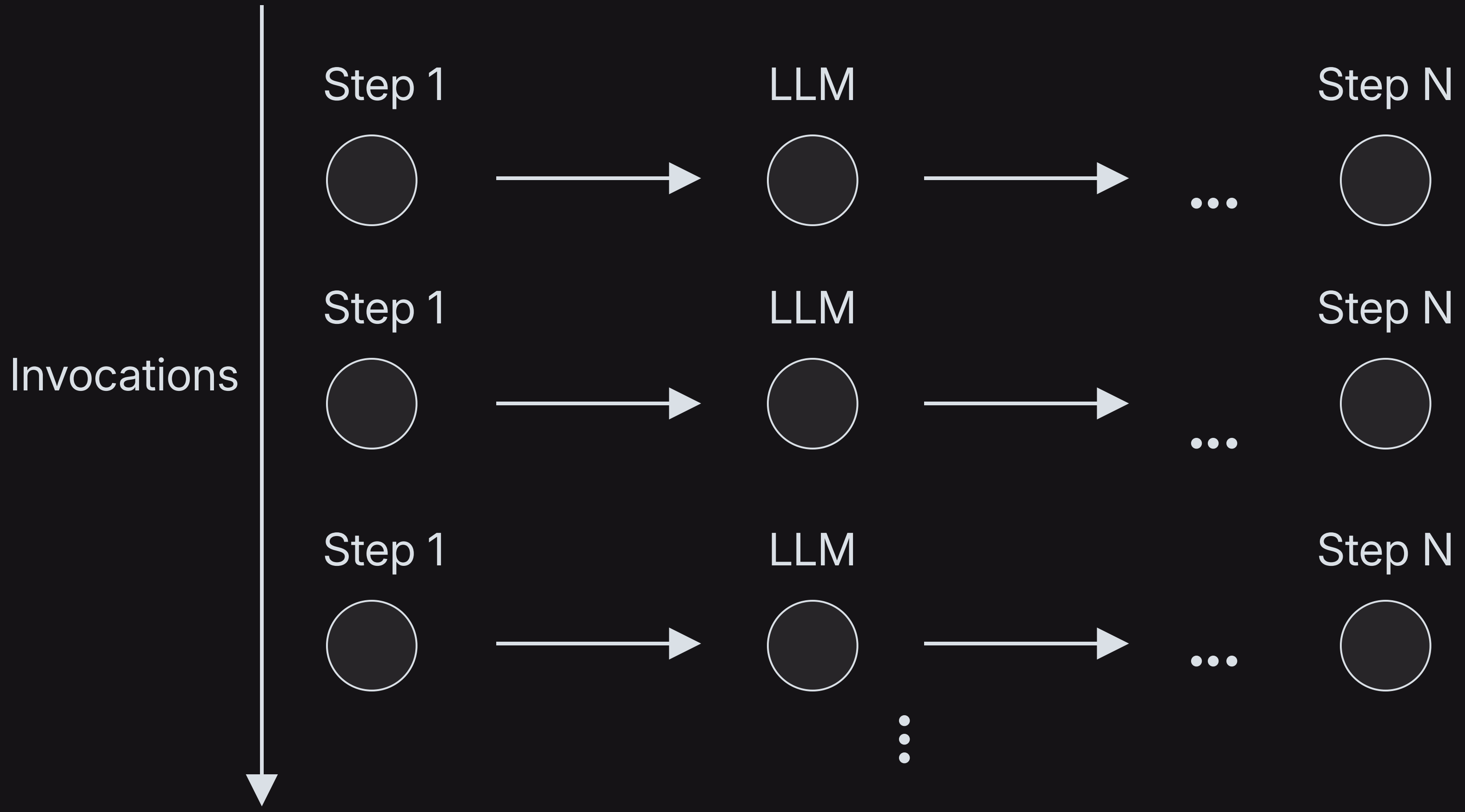
Published Author

AI Workflows use Chains

Follows a sequential fixed order of steps



AI Workflows use Chains

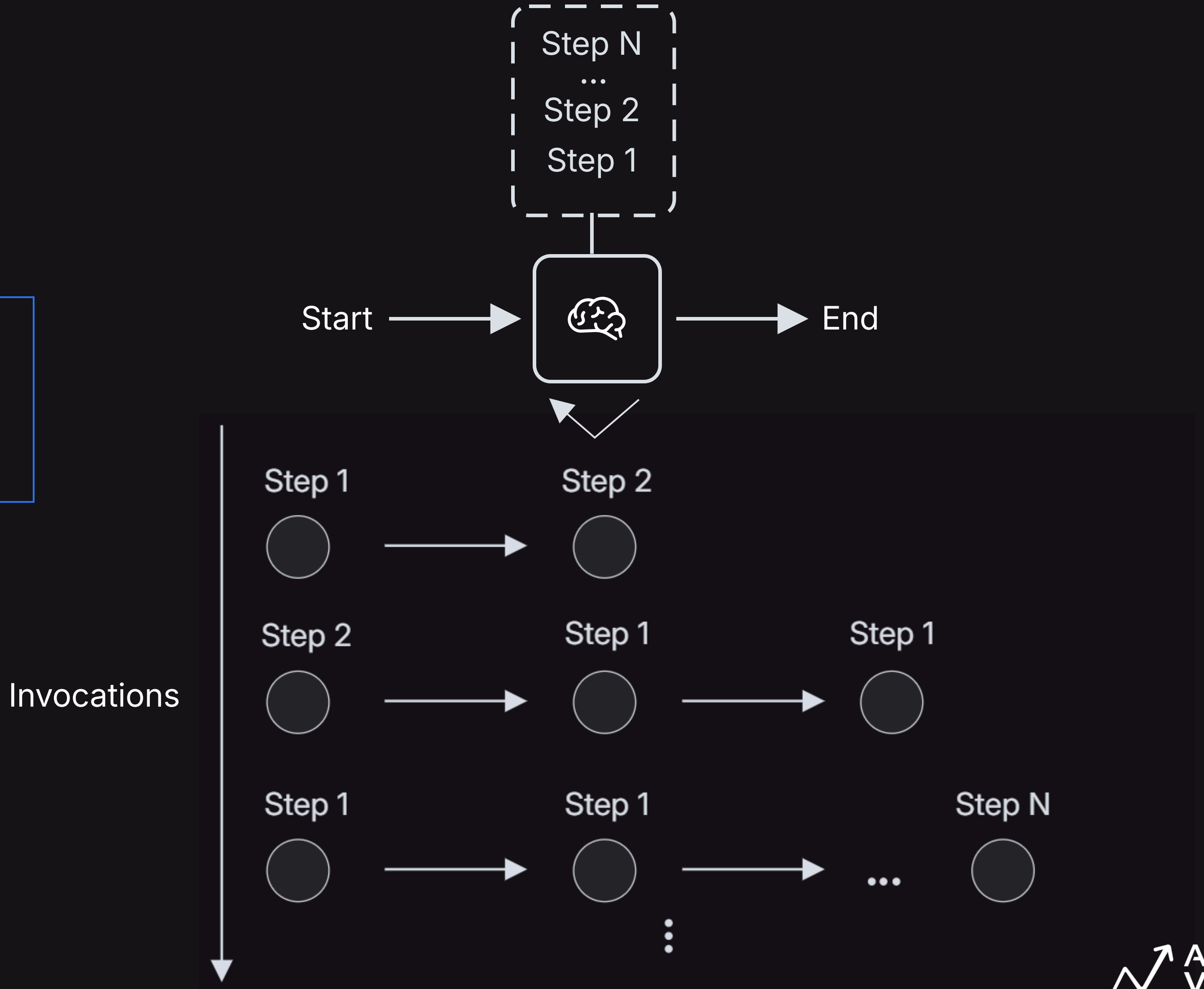


Workflows are reliable as the same fixed steps are executed in order

Analytics Vidhya

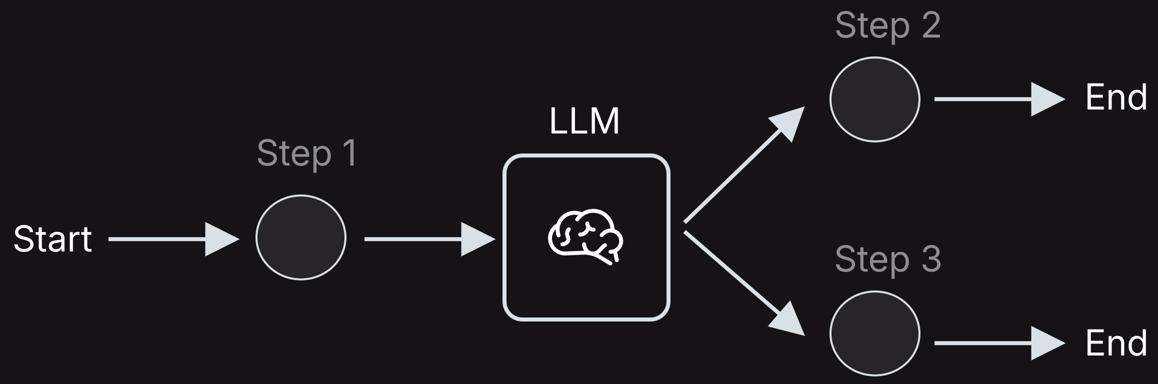
Agentic AI Systems Rely on the LLM to Control the Flow

Autonomous complex systems would rely on the LLM to reason and decide the execution flow of steps

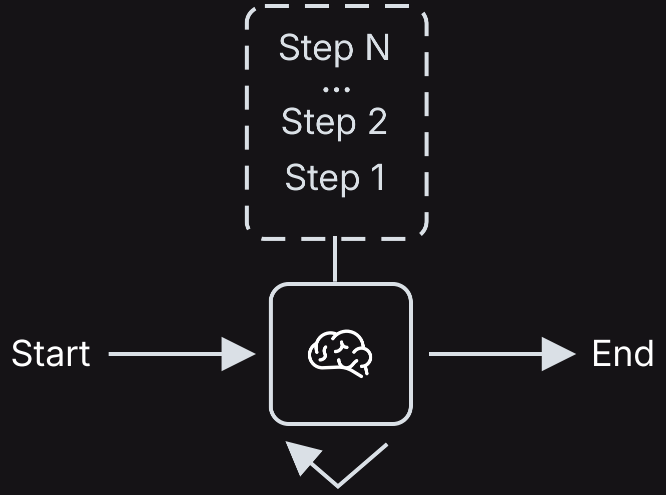


Analytics Vidhya

There are a Variety of Agentic AI Systems



Router

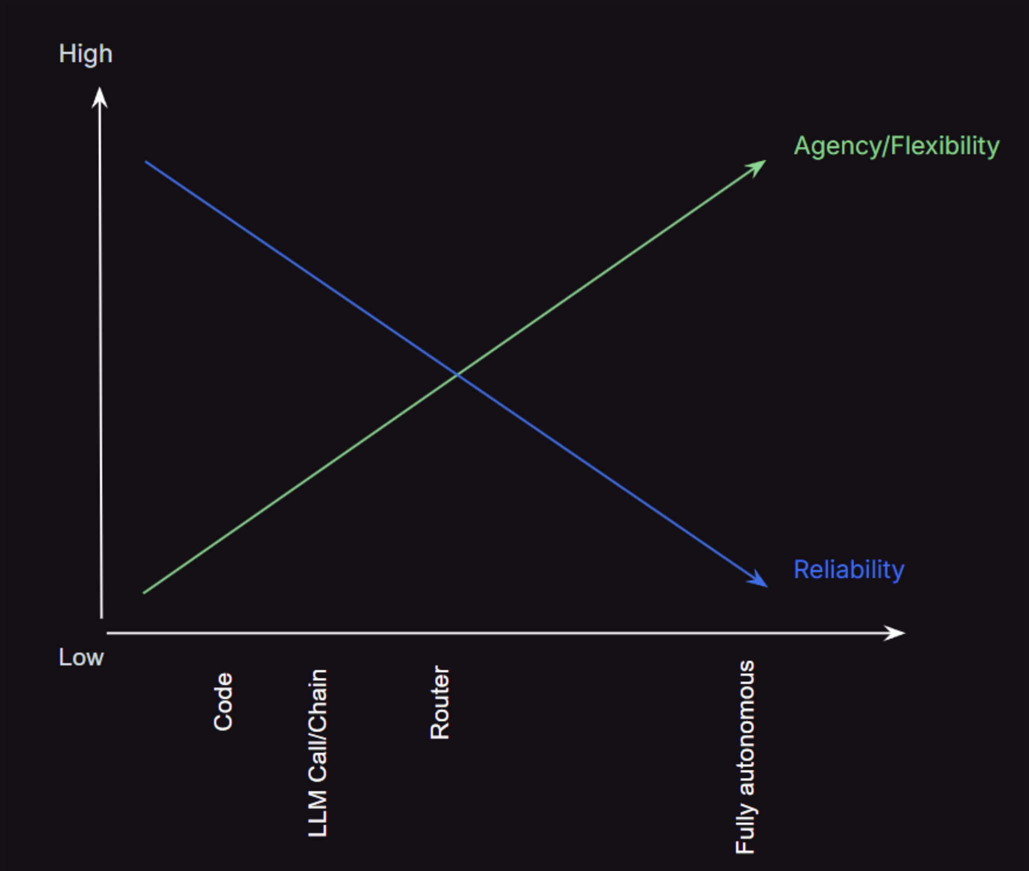


Fully Autonomous

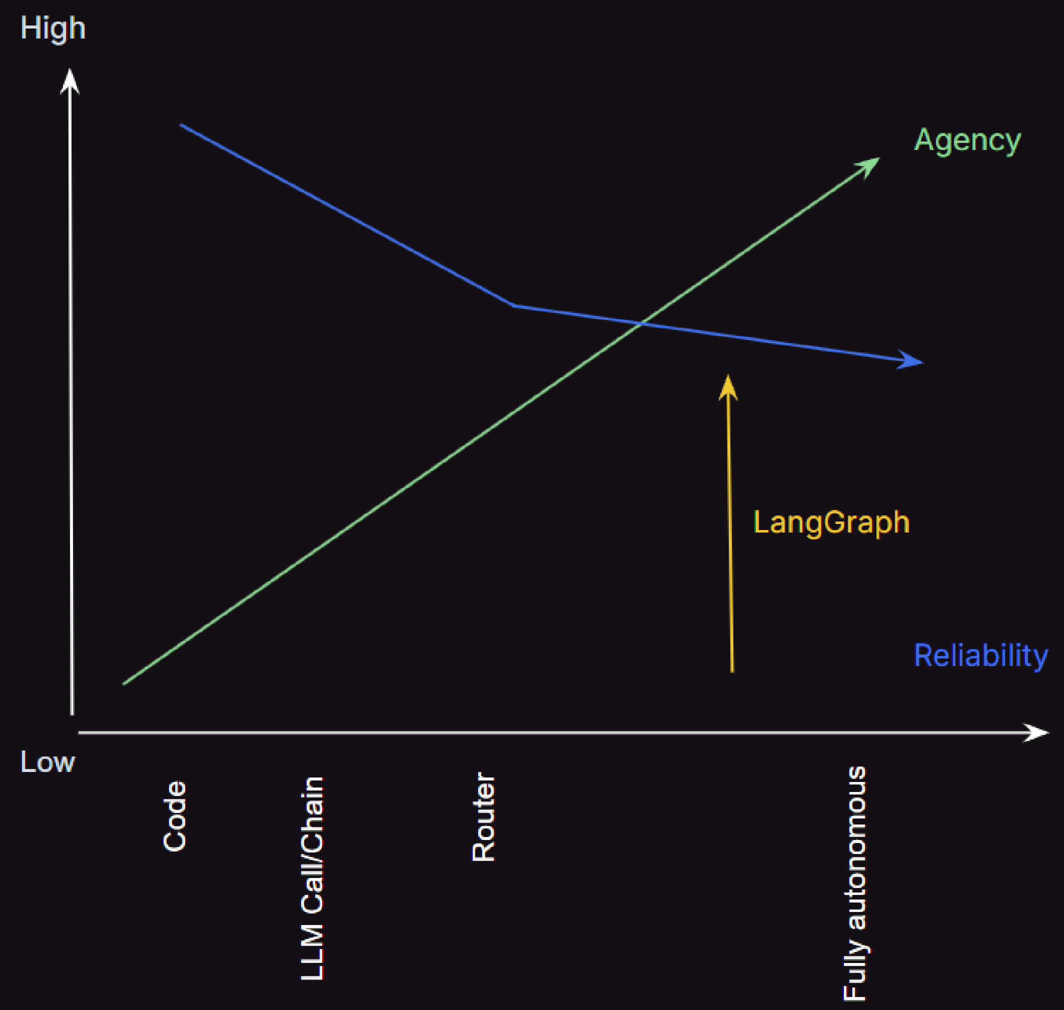
Practical Challenges with Agentic AI Systems

- More autonomy and reliance on LLMs leads to more flexibility but a drop in reliability

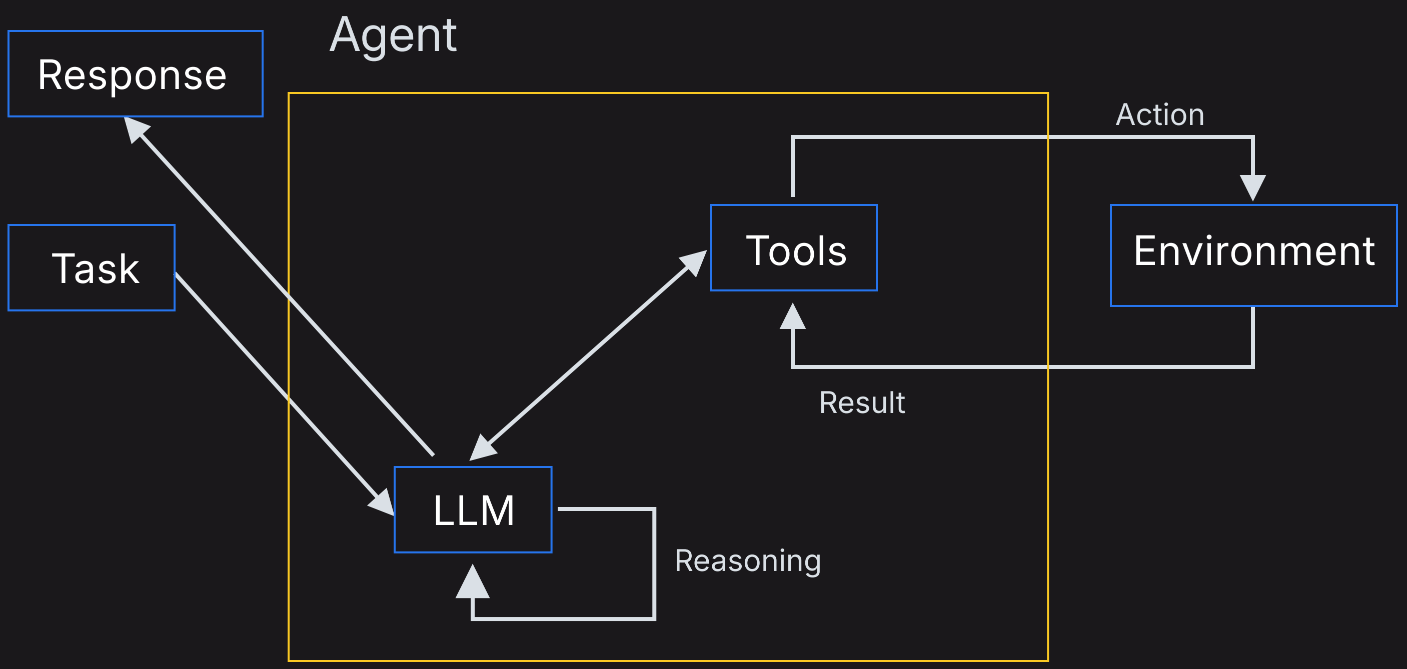
- Less autonomy and reliance on manual workflows leads to less flexibility but an increase in reliability



LangGraph Helps you Improve Reliability of Agentic Systems



Standard Agentic AI System Workflow



- This agentic workflow consists of a large language model (LLM) and tools
- The ReAct methodology is used to run the agent:
    - If the agent says to take an action (i.e. call tool), relevant tools are run and results are returned to the agent
    - If the agent did not ask to run tools, the response output is sent back to the user

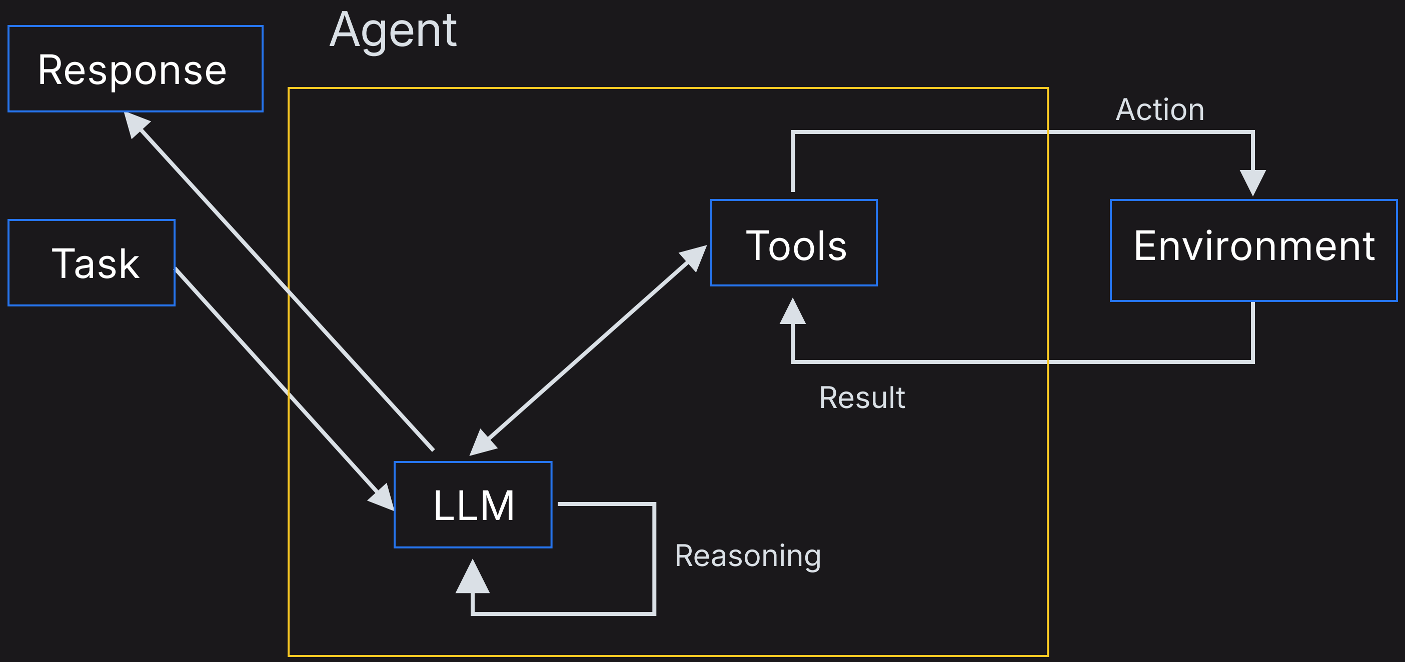
Analytics Vidhya

Why LangGraph?

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

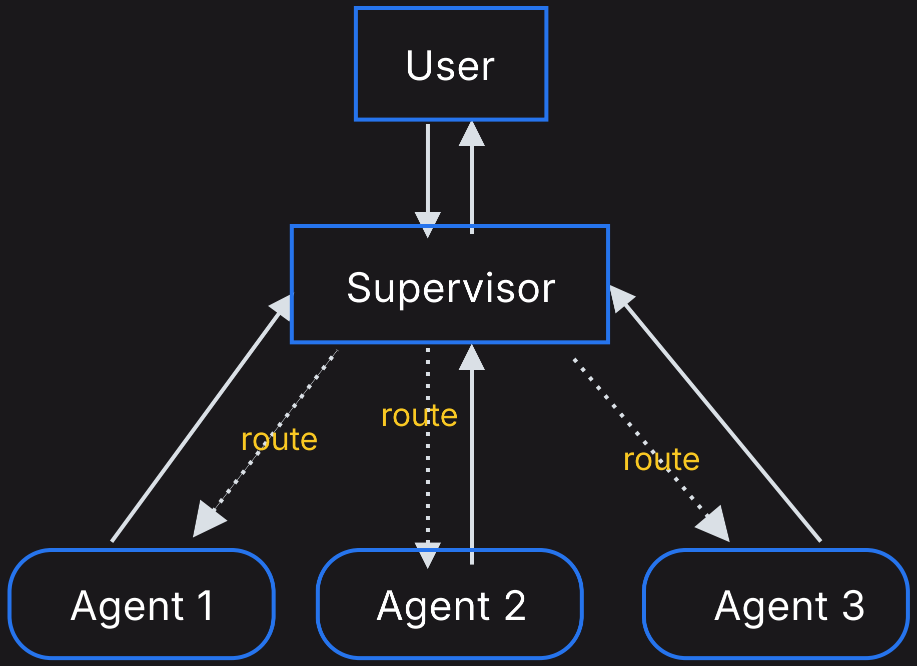
- **Controllability:**
  Offers low-level control which increases reliability in agentic systems.
- **Human-in-the-Loop:**
  Built-in persistence layer enhances human-agent interaction patterns.
- **Streaming First:**
  Supports streaming of events and tokens, providing real-time feedback to users.

Standard Agentic AI System Workflow



- Complex LLM applications often use cycles during execution.
- These cycles often use the LLM to reason about what to do next in the cycle, like Chain of Thought.
- This can essentially be thought of as running an LLM in a for-loop.
- These types of systems are commonly called AI agents.
- LangGraph is the perfect framework to model these agents as graphs, and LangChain recommends it.

Multi-Agent AI System Workflow



- Multi-Agent workflows involve multiple independent agents powered by LLMs connected in a specific way
- Each agent can have its prompt, LLM, tools, and other custom code to collaborate with the other agents if needed
- LangGraph makes each agent a node in the graph. The control flow is managed by edges, and they communicate by adding to the graph's state

Analytics Vidhya

Thanks