LangGraph Basics

Getting Started with LangGraph

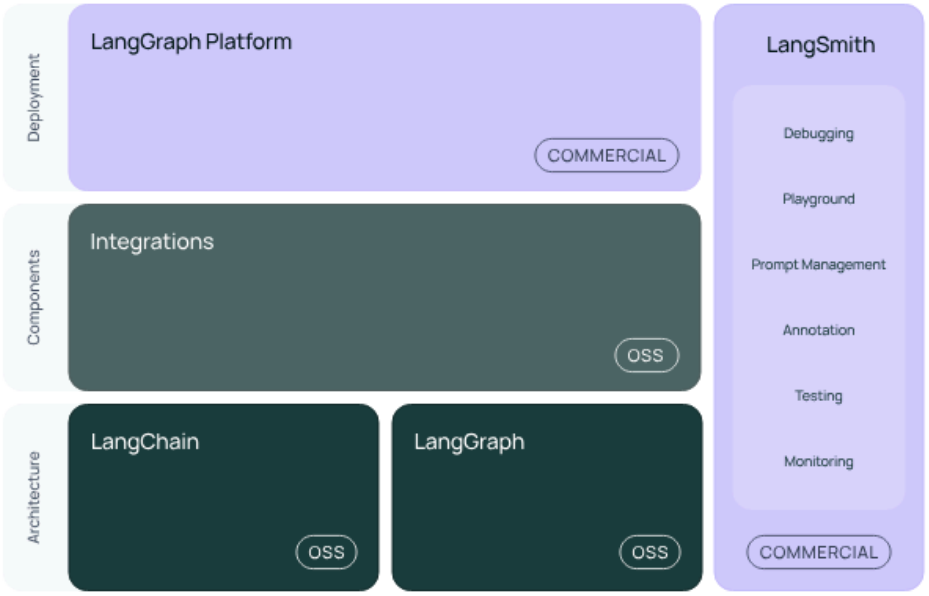
LangGraph is a library for building stateful, multi-actor applications with LLMs, used to create agents and multi-agent workflows.

LangGraph is inspired by Pregel and Apache Beam. The public interface draws inspiration from NetworkX. LangGraph is built by LangChain Inc, the creators of LangChain, but can be used without LangChain

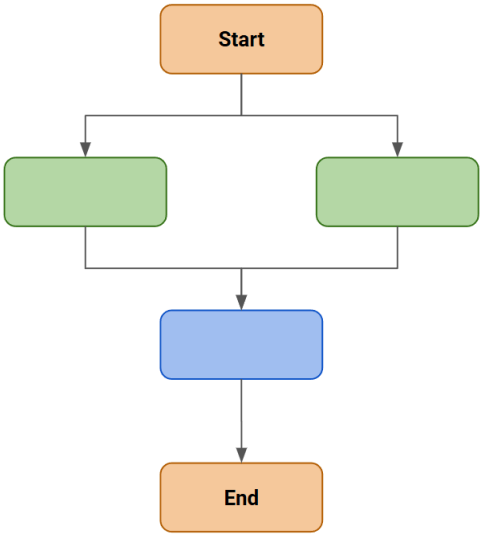
LangGraph powers production-grade agents, trusted by LinkedIn, Uber, Klarna, GitLab, and many more. LangGraph provides fine-grained control over both the flow and state of your agent applications. It implements a central persistence layer, enabling features that are common to most agent architectures.

- **Memory:** LangGraph persists arbitrary aspects of your application's state, supporting memory of conversations and other updates within and across user interactions

- **Human-in-the-loop**: Because state is checkpointed, execution can be interrupted and resumed, allowing for decisions, validations, and corrections at key stages via human input.



LangGraph



- (DAG) - Directed Acyclic Graph
- Stateful AI Agents
- Flow of information

Important Components

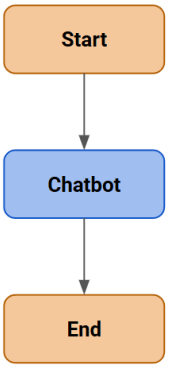
1. **Nodes:** These are the fundamental building blocks of a LangGraph workflow. Each node is typically a Python function that encapsulates a specific unit of work or computation. Nodes receive the current state as input, perform operations (e.g., calling an LLM, using a tool, manipulating data), and return an updated state.

2. **Edges:** Edges define the flow of execution within the graph by connecting nodes. They determine which node to execute next based on the current state. There are two main types:

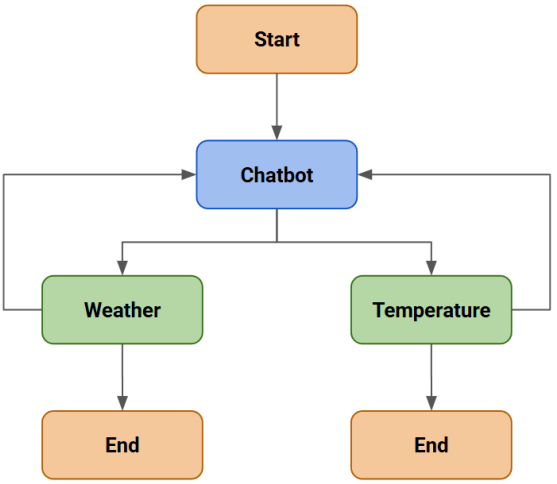
    a. **Simple Edges:** These provide direct, unconditional transitions from one node to another.

    b. **Conditional Edges:** These introduce branching logic, allowing the workflow to dynamically choose the next node based on conditions derived from the current state, similar to if-else statements.

Workflows



Simple Workflow



Complex Workflow