/OSEDEA

Intro to Building and Observing LangGraph Agents

Innovation and ideas made real

“Event oriented programming or microservices are too easy to debug? You want randomness in your results?

You have it with agents!”

My friend, Cedric L’Homme

The goal

Learn the concepts necessary to build agents without needing to:

- Bang your head on the desk
- Explain everything to a rubber duck
- Wonder why your agent thinks the capital of France is… “croissant”

Today’s agenda

- Who am I?
- What are Agents?
- State of Agentic System Design
- Agent Development Stack
- What is LangGraph
- What is LangFuse
- Code Demo
- QA

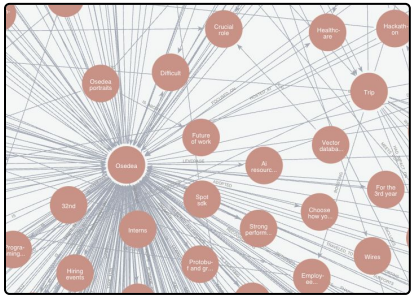
Hi! I'm Carl Lapierre

- Software developer from **Montreal, Quebec**
- 10+ years of experience in software development
- Tech Lead at Osedea, an Innovation firm in Montreal that specializes in AI & Robotics (We're hiring 😉)
- At Osedea, I specialize in agentic systems with 20+ agents designed and built across production, prototypes, and research use cases.

carllapierre

Some Street Creds

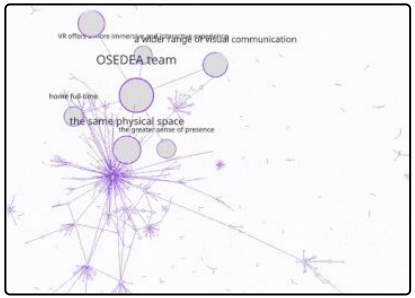
Drive Through Agent (2023)
Whisper, Coqui, LangChain, GPT-3.5-Turbo



GraphRAG (2023)
LLama Index, Neo4j, GPT-4o-mini

SPOT Draws (2024)
Whisper, Stable Diffusion, Modal, LangChain, GPT-4o-mini

Twitch Streamer (2024)
MAS, LangGraph, GPT-4o-Mini, LangFuse



Graph Explorer (2025)
LangGraph, GPT-4o-mini, LangFuse

Judy (2025)
LangGraph, OpenRouter, Langfuse

+ Enterprise Agents & MCPs
Various Technologies

Genius ERP (2025)
Semantic Kernel, GPT-4o-mini

What is an agent? (Today)

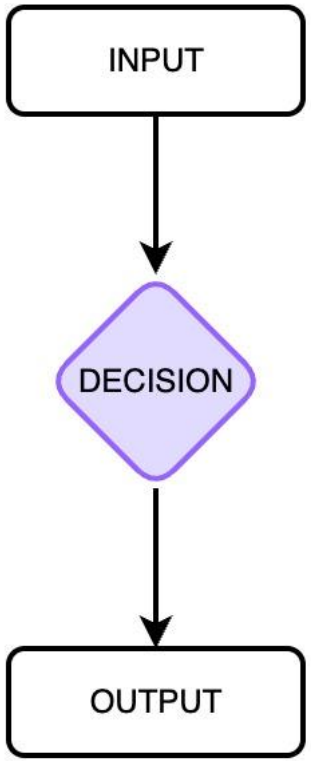
Agents are systems that use a Large Language Model (LLM) to reason on a given context and goal, and decide on the action to take in order to accomplish that goal.

They can take action by using tools, like search engines and code interpreters, in order to find information and perform tasks.

Basic Agent

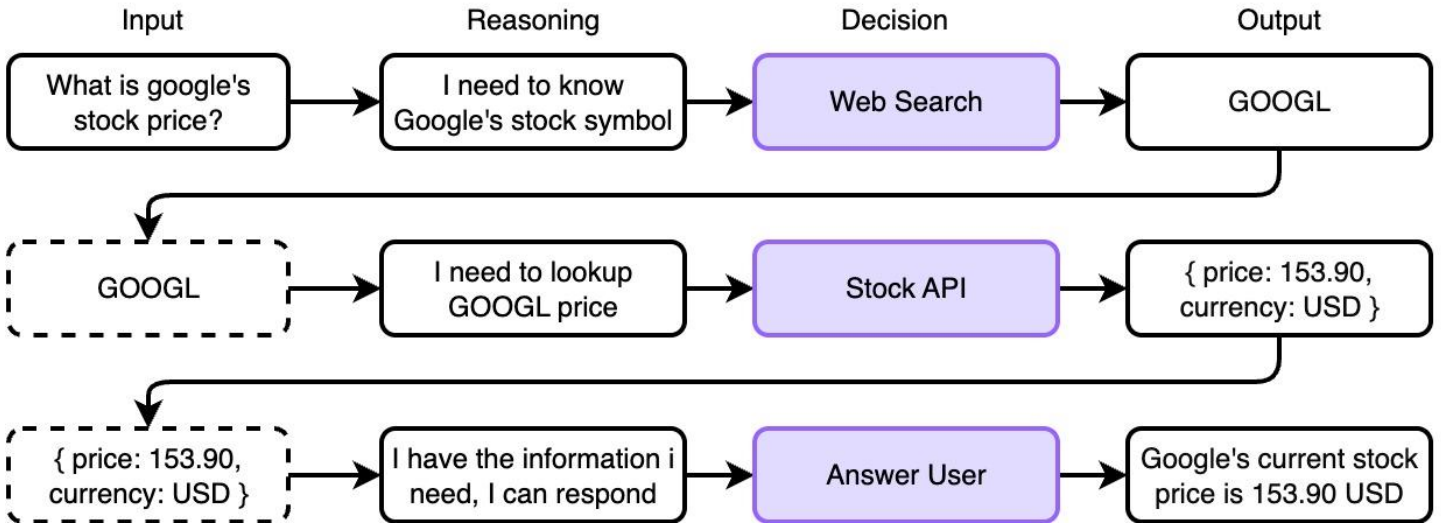
A basic agent follows a linear path from input to output.

At its core, an agent will simply make a decision based on the context it has.



Powerful Patterns

ReAct (Reason + Act) is a common agentic pattern. Instead of a single decision, the agent loops through a process of reasoning and acting until it reaches a final answer.



Agents & Workflows

Workflows

- Predefined, linear, or branching sequences of steps.
- Control-flow is hardcoded
  (e.g., “Step A → Step B → Step C”).
- Deterministic: same inputs → same outputs.

Agents

- Dynamic, LLM-directed control-flow.
  Decisions about “what to do next” are made at runtime.
- More flexible but less predictable.
- Example: an assistant deciding whether to call a tool, search, or ask a clarifying question.

A spectrum, not a dichotomy

- Simple linear workflows are themselves agents, just very constrained ones. (Constraints are GOOD!)
- As you add autonomy and decision-making, you move along the spectrum toward fully agentic systems.
- Most real systems combine both: workflows provide reliability and structure, while agentic components add adaptability.
- Together, they make up an agentic system.

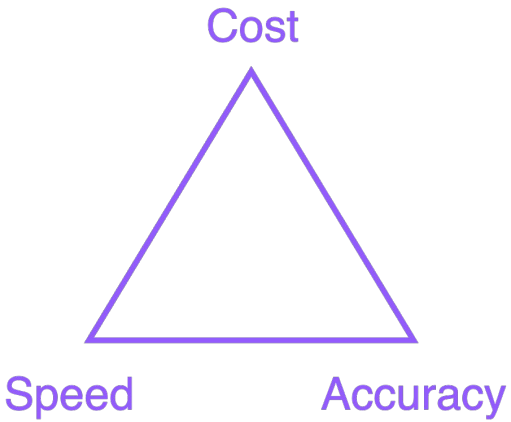
Building Agents

Agentic System Design is About Balancing Trade-Offs

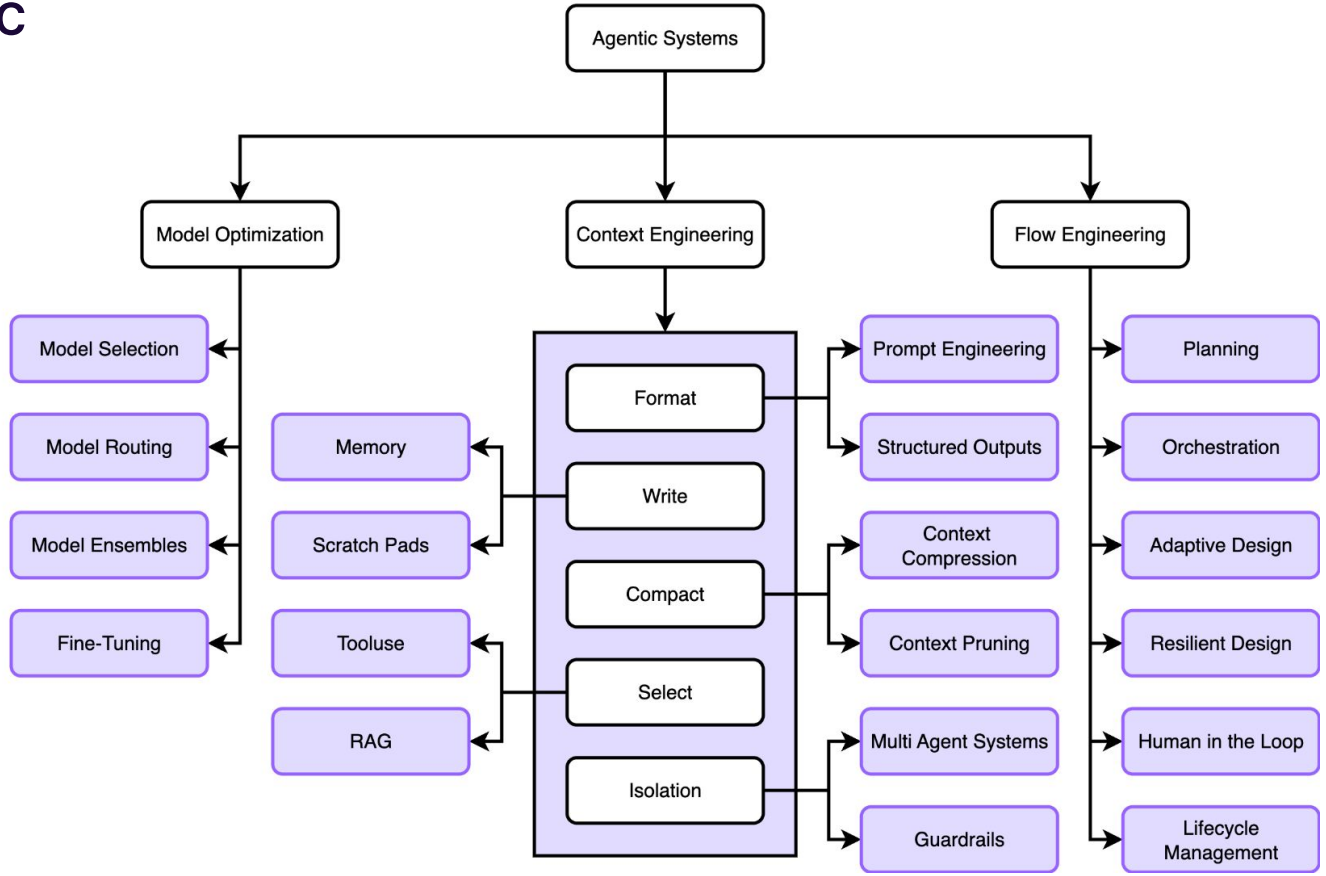
Designing robust systems requires managing three competing dimensions:

- Cost (compute, tokens)
- Performance (accuracy, reliability)
- Latency (speed, responsiveness, user experience)

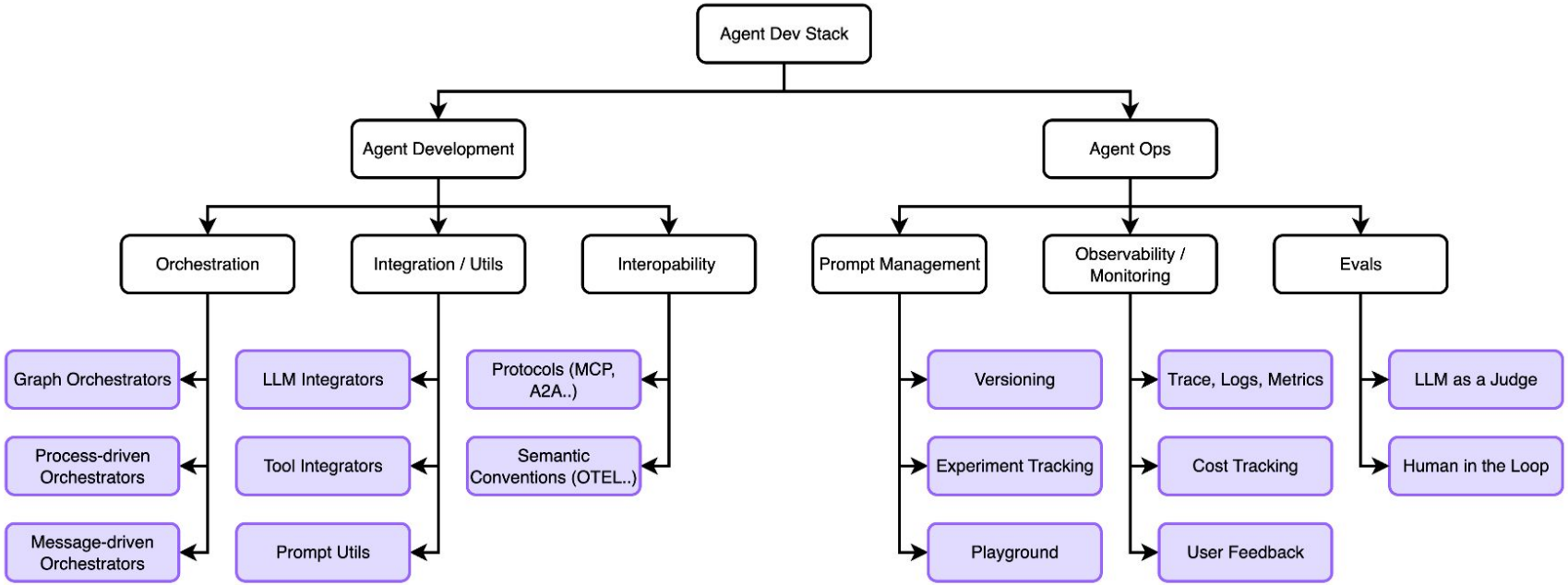
Optimizing one often comes at the expense of the others, context determines the right balance.



State of Agentic System Design

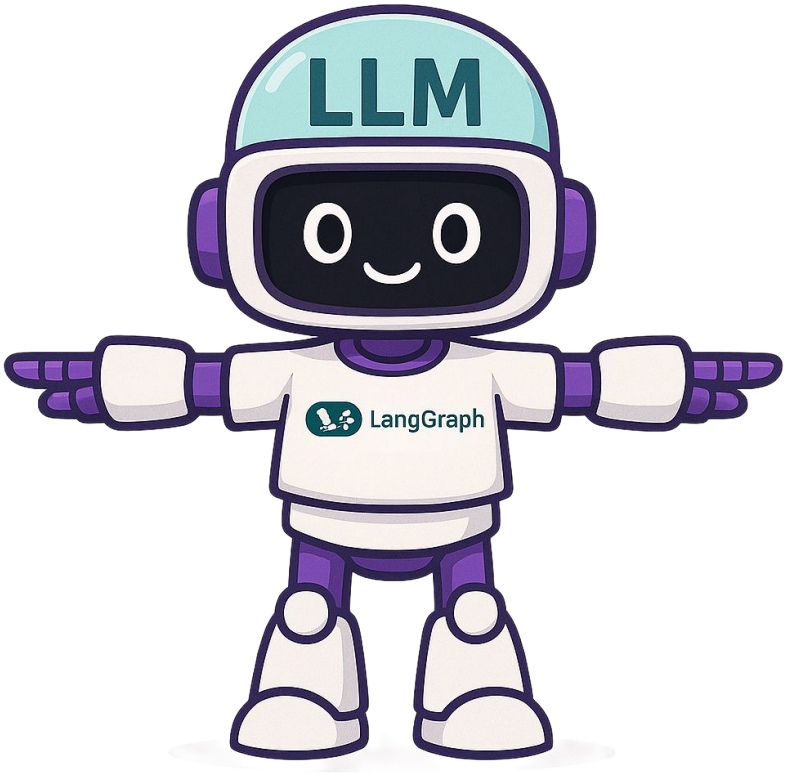


Agent Development
Stack



What is LangGraph?

LangGraph is a library from the LangChain team for building stateful, multi-agent applications. It allows you to define agent workflows as graphs, where nodes represent functions (or LLM calls) and edges represent the control flow between them.

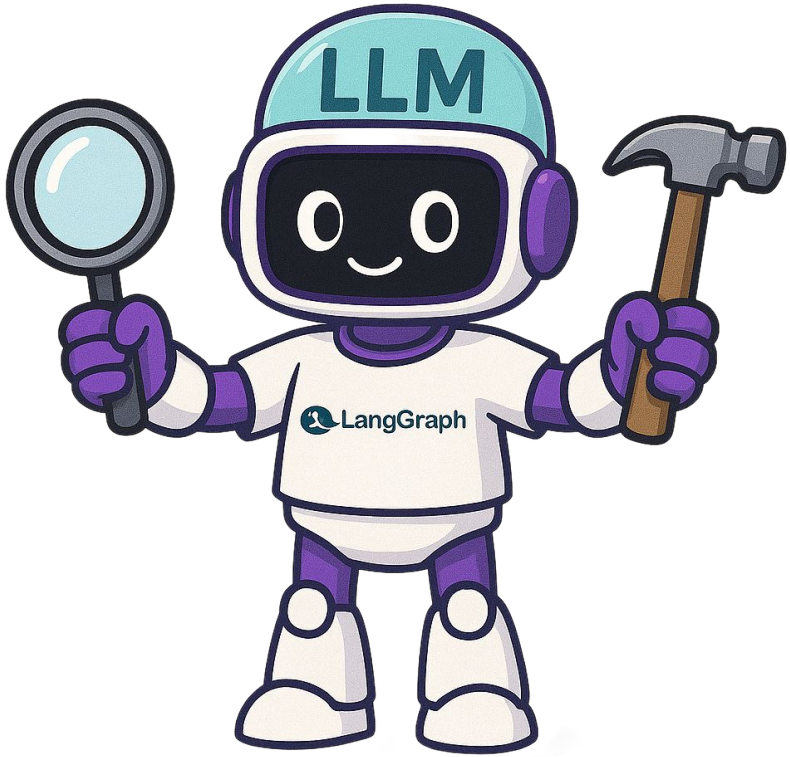


LangGraph Concepts

- **State**
  A central, persistent object that is passed between all nodes in the graph. Each node can update the state.

- **Checkpoints**
  LangGraph can automatically save the state of the graph. This allows you to resume long-running tasks, recover from failures, and inspect the full history of an agent's work.

- **Interruptions**
  The graph can be paused at any point, for example, to wait for human input or to handle an error. This is essential for human-in-the-loop workflows.



Why do we need LangGraph?

You don’t, but...

A key advantage of using a framework like LangGraph is its built-in instrumentation. The framework is designed to emit telemetry data out-of-the-box.

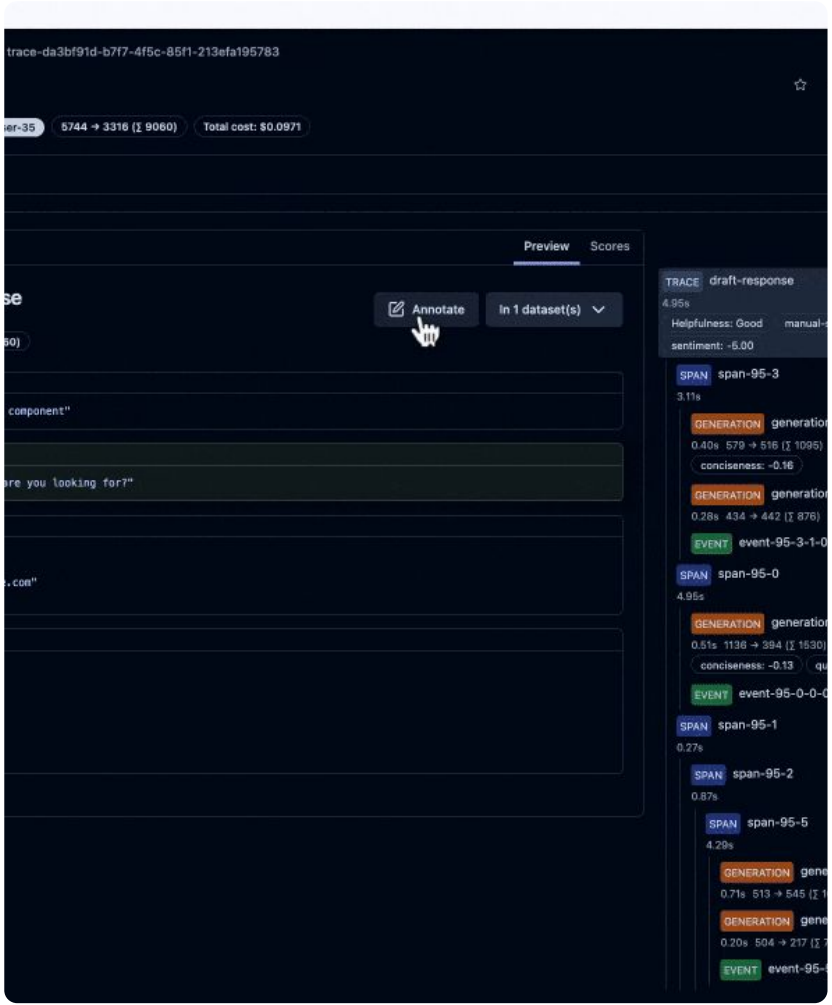
By simply adding a callback handler from a tool like Langfuse, you can capture detailed traces of your agent's execution, making complex workflows easy to observe, debug, and optimize.

LangGraph is also less intrusive than other orchestration frameworks and sits at a lower level of abstraction.

What is Langfuse?

Langfuse is an open-source observability and analytics platform specifically designed for LLM applications. It helps you:

- Trace complex chains and agentic systems.
- Debug errors and unexpected outputs.
- Evaluate the quality and performance of your application.
- Manage prompts and versions.



Let’s jump in some code

Thank you

Carl Lapierre
Tech Lead

Slides & Code

How to contact us

osedea.com
info@osedea.com
LinkedIn