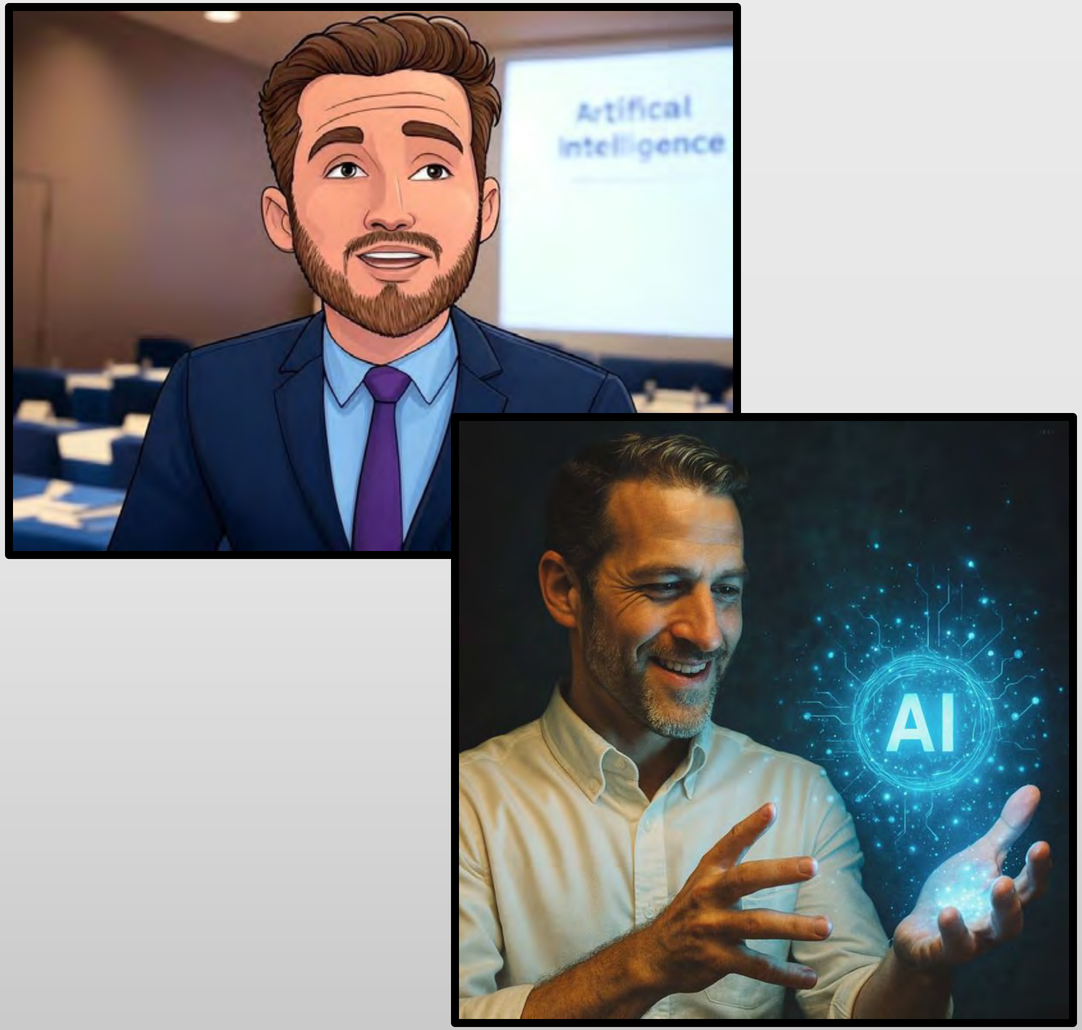
AI Agents

Getting Useful Results from AI Reasoning

Welcome

- About me
  - Background in AI
  - Biological modeling
  - Composable/CloudFit
  - Gradient Momentum
- Today
  - Building Blocks
  - Agents/Tools
  - Demos
  - Experience



What is an Agent

“Asking an LLM to do something and giving it the tools to make it happen.” – Jeremy

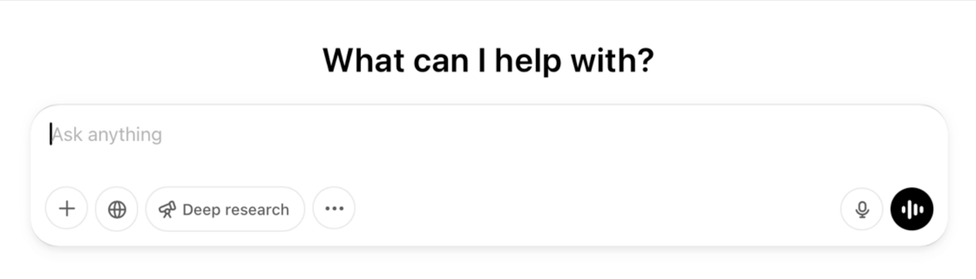
“An application that attempts to achieve a goal by observing the world and action upon it using the tools that it has at its disposal” – Google

“Applications where LLM outputs control the workflow” – Hugging Face

“Systems where LLMs dynamically direct their own processes and tool usage, maintaining control over how they accomplish tasks” – Anthropic

Building Blocks

- Prompt engineering is critical
- Importance of structure
- Importance of constraints



What can I help with?

User Input

Previous Conversation

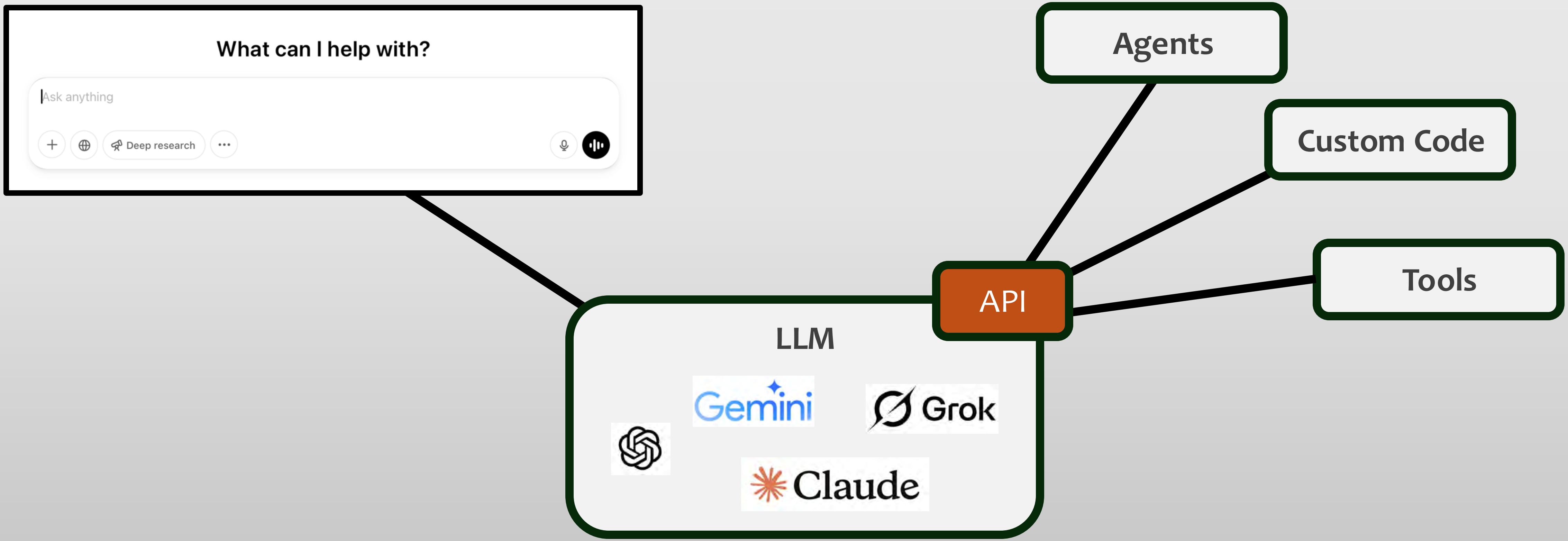
External Data

System Prompt

Demo

Structure and Constraints

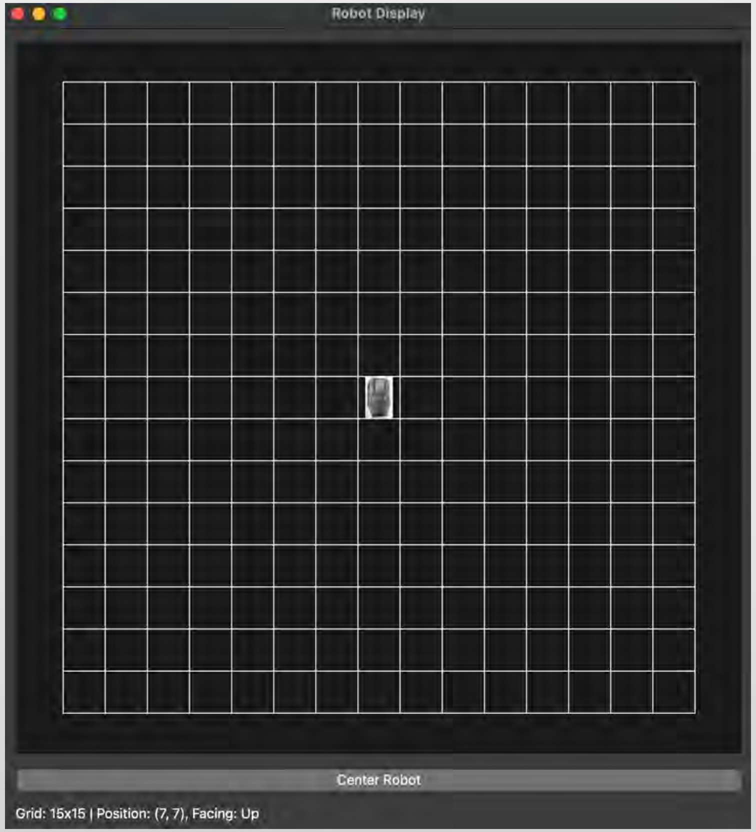
GPT as an API



What can I help with?

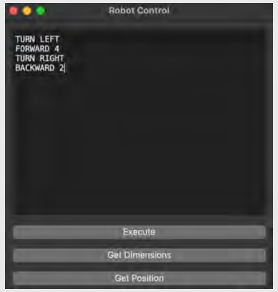
Demo

Structure and API Integration

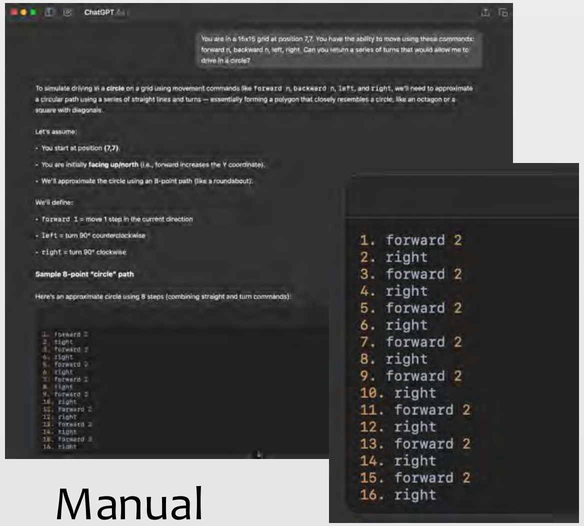


Grid: 15x15 | Position: (7, 7), Facing: Up

Manual Controller



```
TURN LEFT
FORWARD 4
TURN RIGHT
BACKWARD 2
```



You are in a 15x15 grid at position 7,7. You have the ability to move using these commands: forward n, backward n, left, right. Can you return a series of turns that would allow me to drive in a circle?

To simulate driving in a **circle** on a grid using movement commands like `forward n`, `backward n`, `left`, and `right`, we'll need to approximate a circular path using a series of straight lines and turns — essentially forming a polygon that closely resembles a circle, like an octagon or a square with diagonals.

Let's assume:

- You start at position **(7,7)**.
- You are initially **facing up/north** (i.e., forward increases the Y coordinate).
- We'll approximate the circle using an 8-point path (like a roundabout).

We'll define:

- `forward 1` = move 1 step in the current direction
- `left` = turn 90° counterclockwise
- `right` = turn 90° clockwise

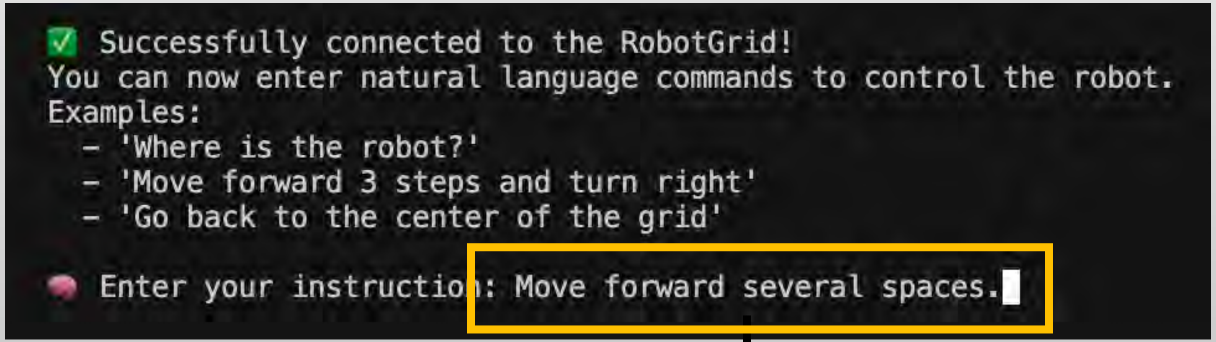
Sample 8-point "circle" path

Here's an approximate circle using 8 steps (combining straight and turn commands):

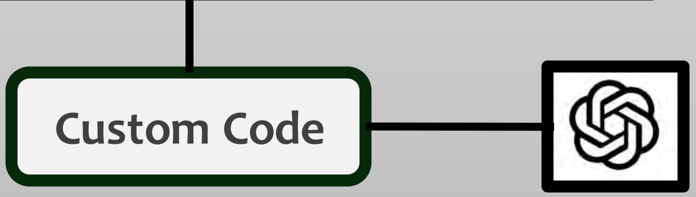
```
1. forward 2
2. right
3. forward 2
4. right
5. forward 2
6. right
7. forward 2
8. right
9. forward 2
10. right
11. forward 2
12. right
13. forward 2
14. right
15. forward 2
16. right
```

Manual Prompting

LLM Controller



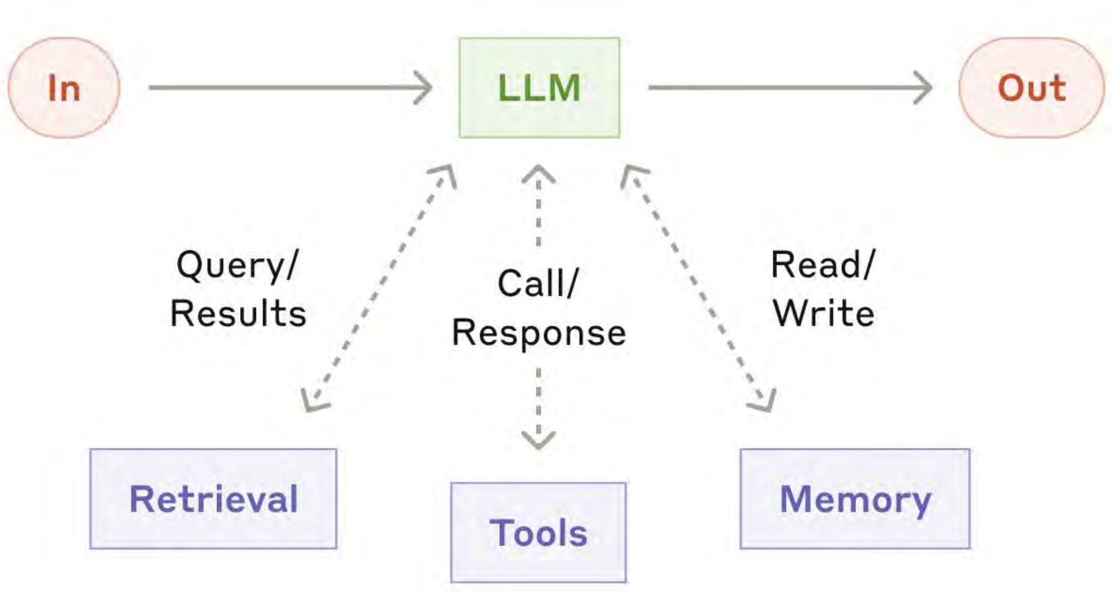
```
✅ Successfully connected to the RobotGrid!
You can now enter natural language commands to control the robot.
Examples:
  - 'Where is the robot?'
  - 'Move forward 3 steps and turn right'
  - 'Go back to the center of the grid'

🧠 Enter your instruction: Move forward several spaces.
```



Custom Code

What structure allows us to do

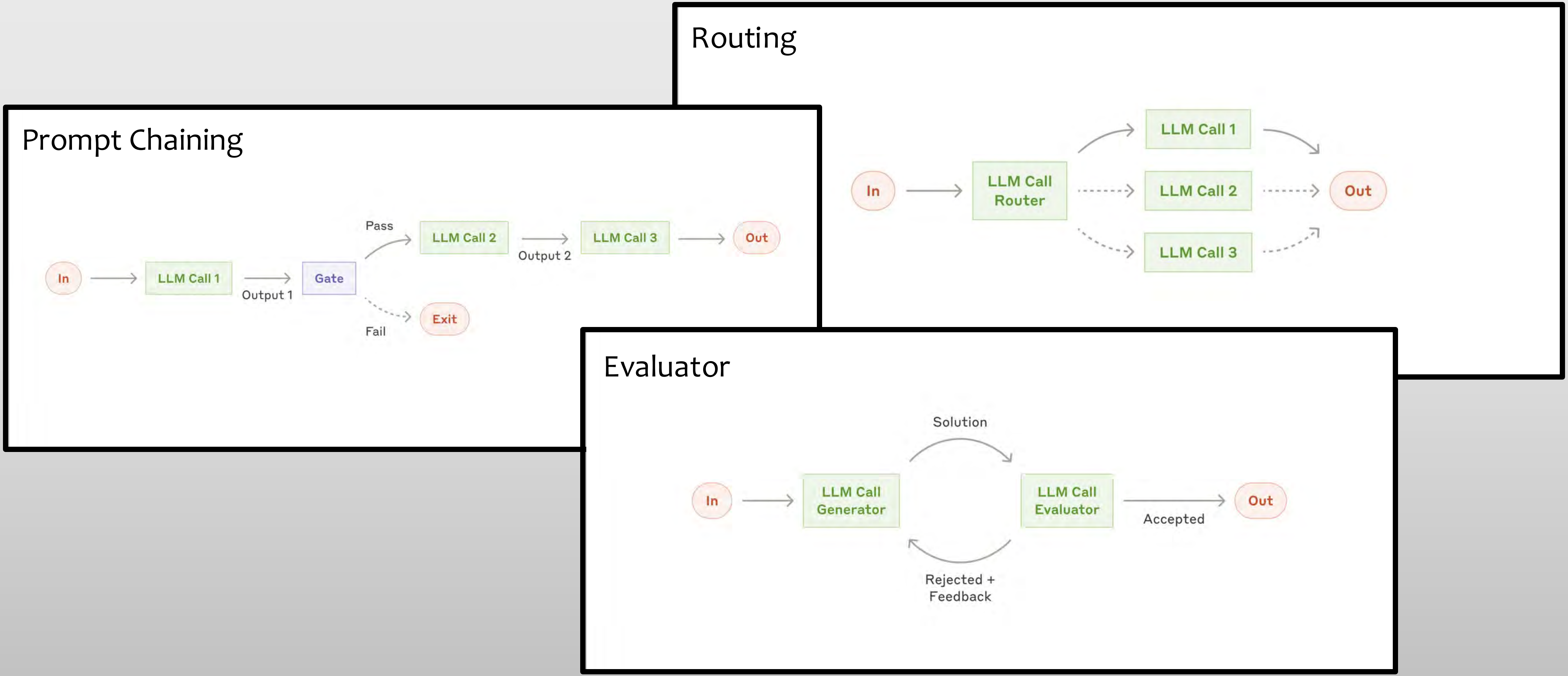
What is an Agent?



https://www.anthropic.com/engineering/building-effective-agents

AI Workflows

Prompt Chaining, Routing, Evaluator

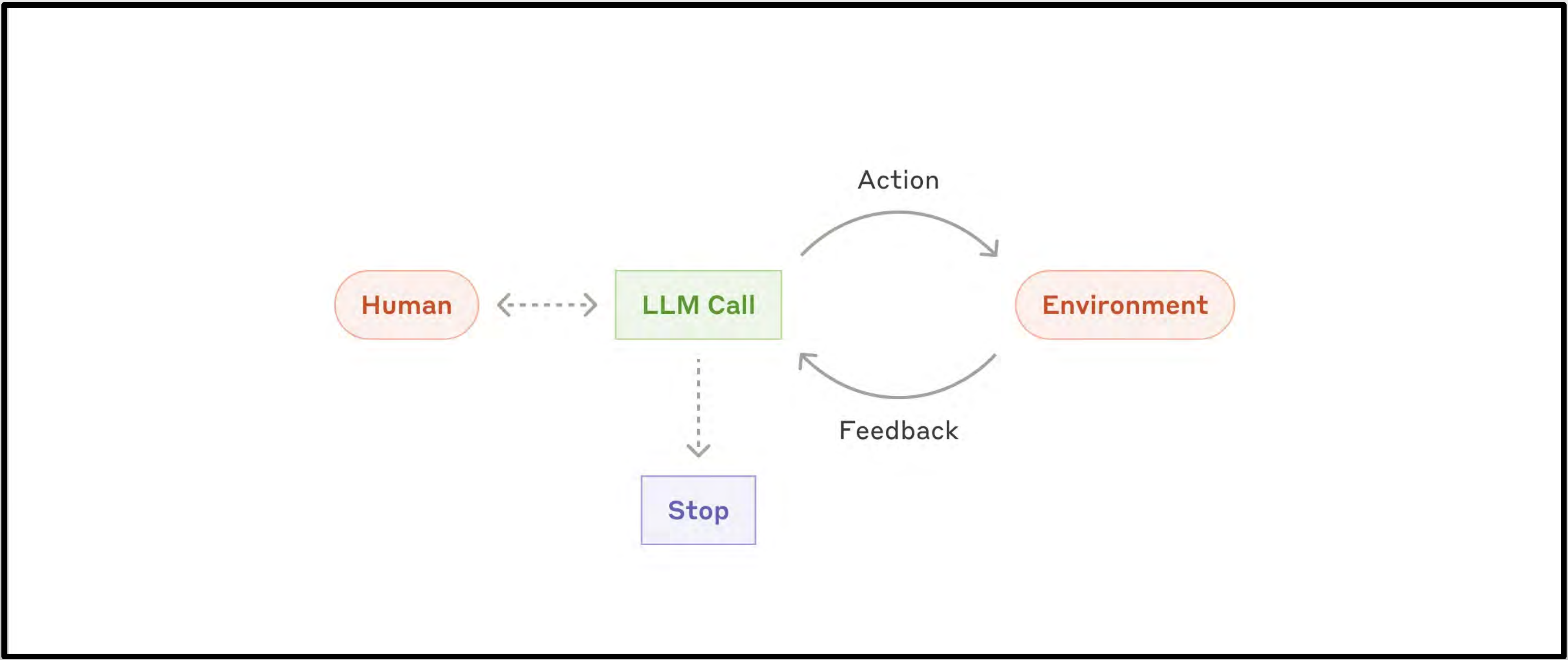


Prompt Chaining

Routing

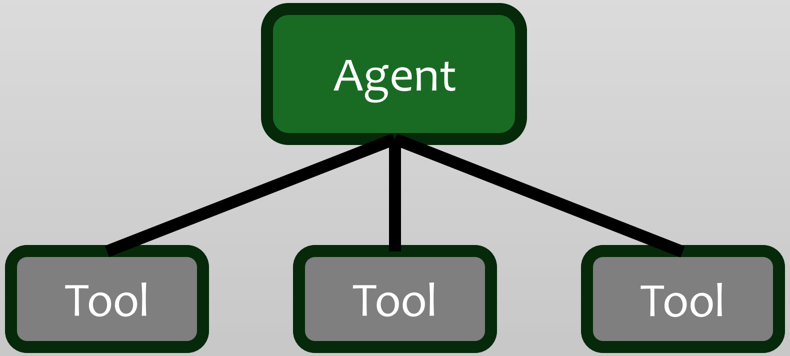
Evaluator

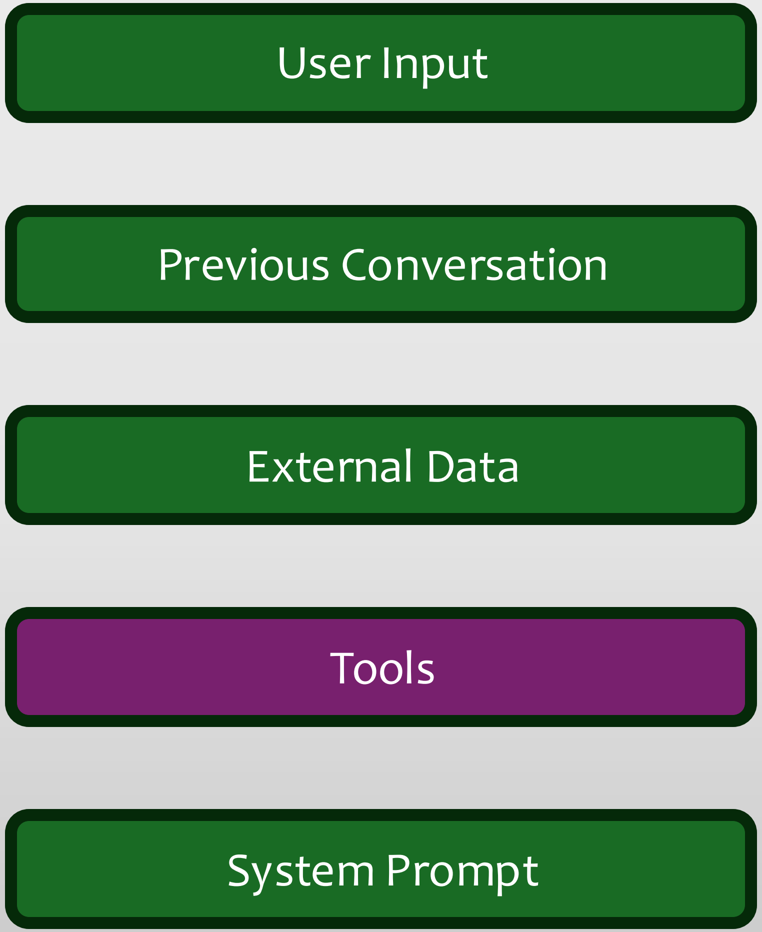
Agents



Tools

- Extends usefulness of an LLM/user interaction
- LLM has knowledge about available tools and how to use them
- Not a new concept




User Input

Previous Conversation

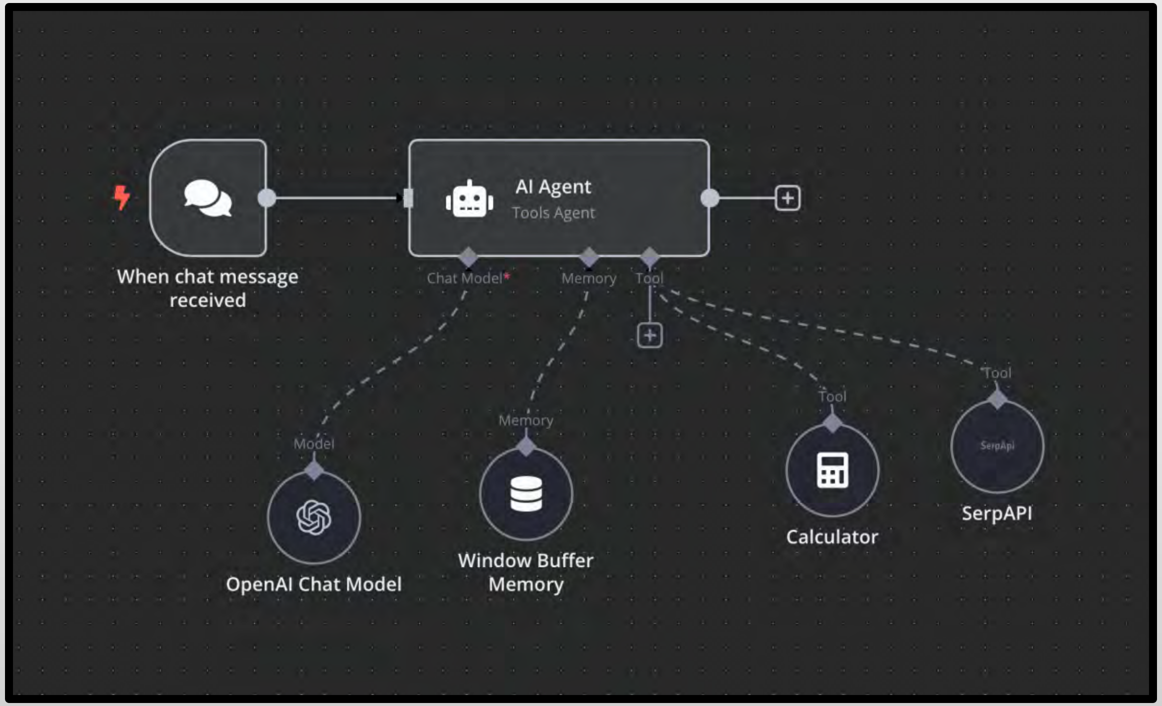
External Data

Tools

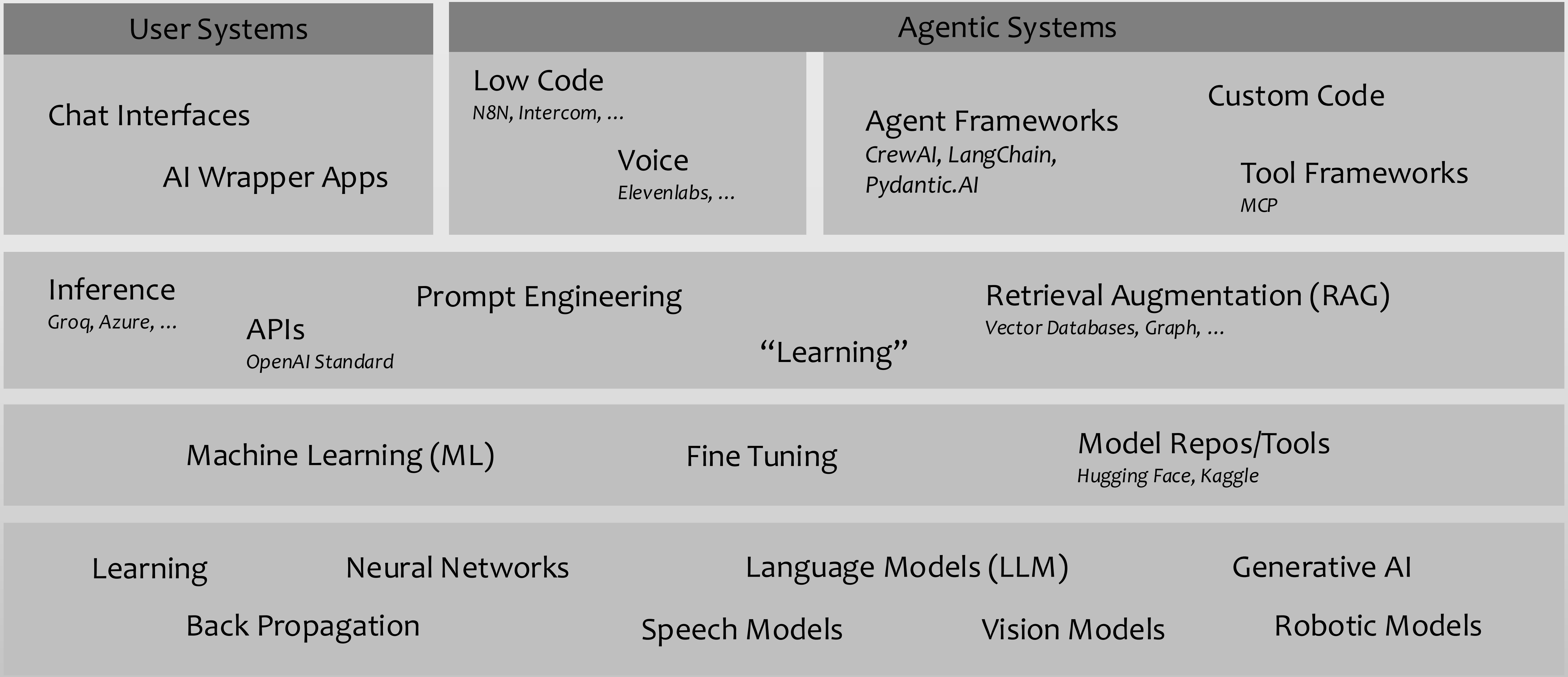
System Prompt

Demo

Low code agents with N8N



Layers and Terms



User Systems

Chat Interfaces

AI Wrapper Apps

Agentic Systems

Low Code
N8N, Intercom, …

Voice
Elevenlabs, …

Agent Frameworks
CrewAI, LangChain,
Pydantic.AI

Custom Code

Tool Frameworks
MCP

Inference
Groq, Azure, …

APIs
OpenAI Standard

Prompt Engineering

"Learning"

Retrieval Augmentation (RAG)
Vector Databases, Graph, …

Machine Learning (ML)

Fine Tuning

Model Repos/Tools
Hugging Face, Kaggle

Learning

Neural Networks

Language Models (LLM)

Generative AI

Back Propagation

Speech Models

Vision Models

Robotic Models

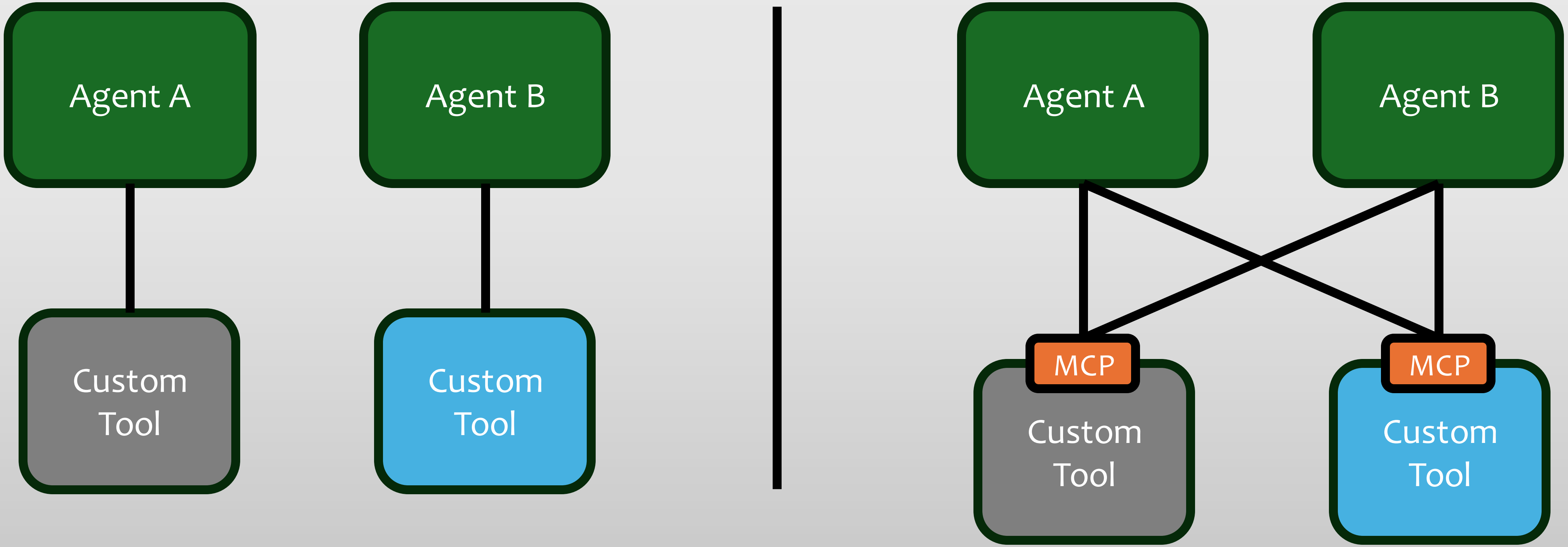
Common Tool Language: MCP

- Model Context Protocol
- A standard for agent-to-tool communication
- Self-describing
- Growing support

“Think of MCP like a USB-C port for AI applications.” - Anthropic

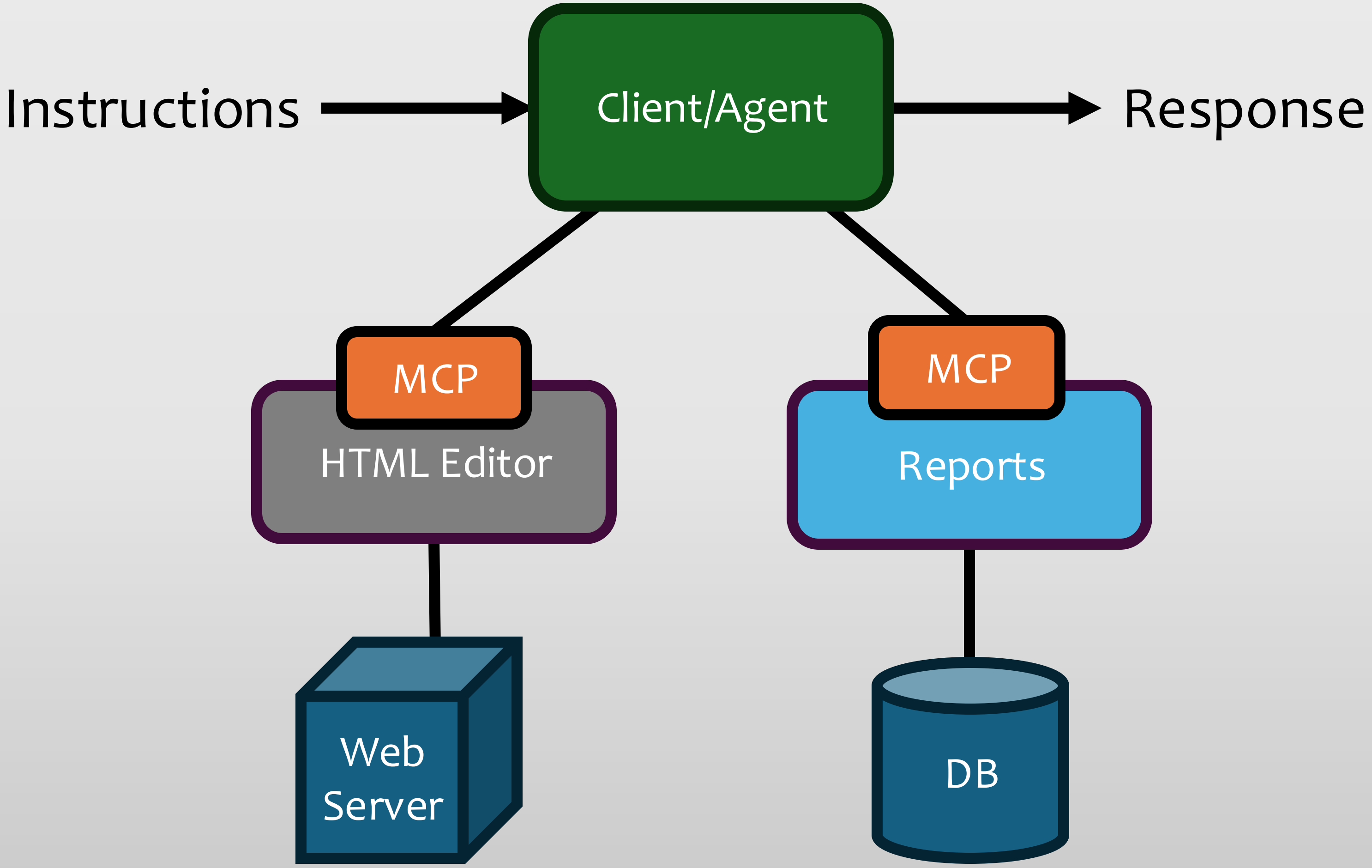
https://modelcontextprotocol.io

Common Tool Language: MCP



Demo

HTML and Report Data



Things I’ve Learned

- No code solutions can be a liability – AI speaks text-based code
- Build in pieces, not all at once
- Code generation challenges – endless loops
  - Let me fix that… adds 20 files
  - Changes best practices mid-flight
  - Check-in code often
- Establish rules/boundaries, or else the AI will get creative on you
- Using AI for automated deployments is painful – ever evolving clouds
- There are a lot of distractions
  - Chasing the latest model can be a waste of time
  - Locally hosting LLMs

Thank you

jeremy@gradientmomentum.com