



AI Agents

by Igor Slinko & Maria Tigina

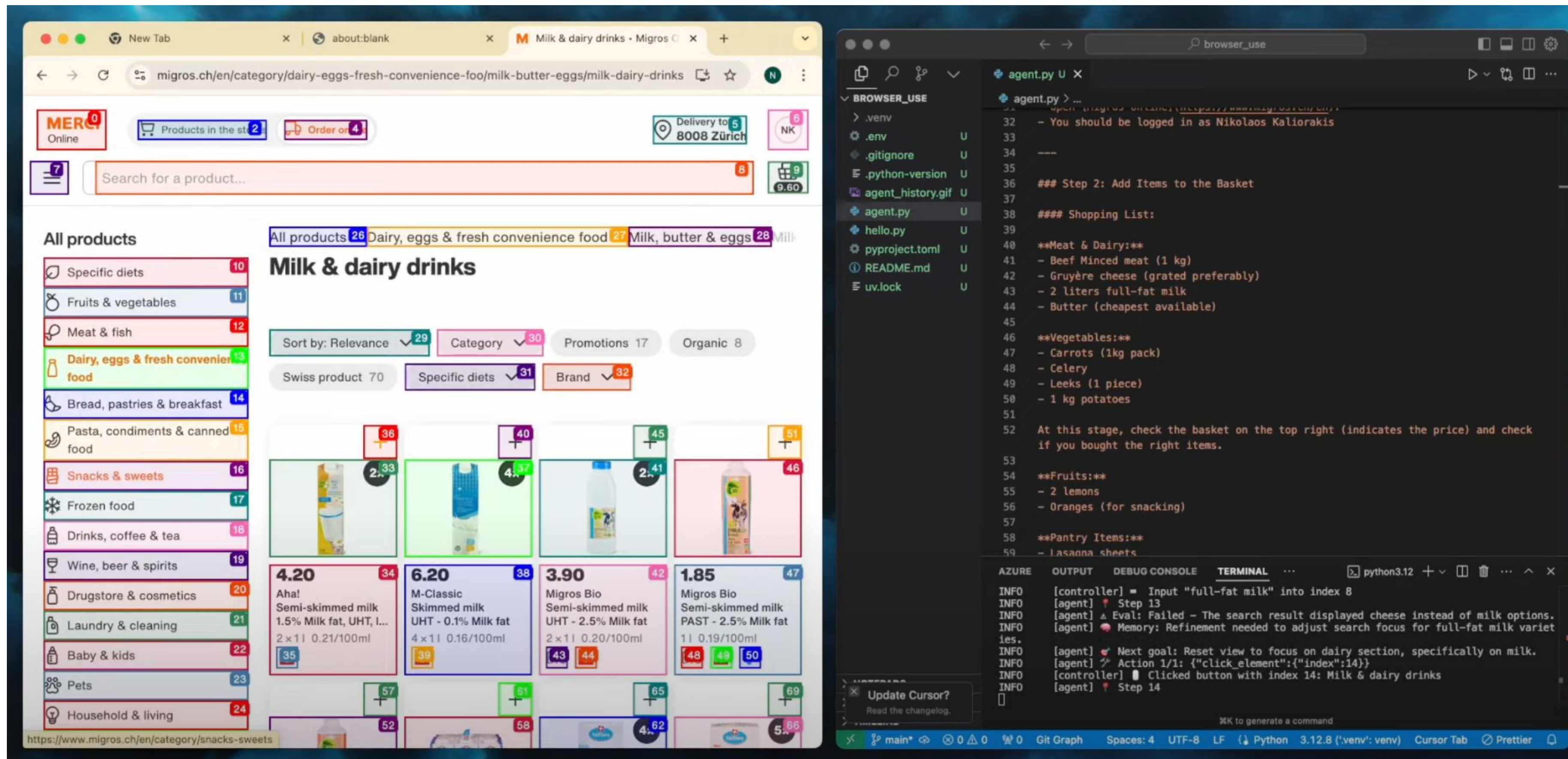
AI Agents & Planning lab @ JetBrains Research

Plan for today

1. Agents around us...
2. What is an Agent?
3. Ways to build an Agent?
4. Workshop: How to build an agent using LangGraph?
5. Homework

Browser Use Agent

- Easy open-source way to build agent with browser access
- <https://github.com/browser-use/browser-use>



Operator by OpenAI

- Same but costs 200\$ / month

Computer Use by Anthropic

- Not only browser but whole computer

<https://leaderboard.steel.dev/>

Steel.dev®

Browser Agent Leaderboard

See how various AI browser agents stack up based on their accuracy in completing web-based tasks on the WebVoyager benchmark.

LEADERBOARD			
RANK	MODEL	WEB VOYAGER	ORGANIZATION
1	BROWSER USE  NEW	89.1%  SOTA	Browser Use
2	OPERATOR NEW	87%	OpenAI
3	KURA NEW	87%	Kura
4	SKYVERN 2.0  NEW	85.85%	Skyvern
5	PROJECT MARINER	83.5%	Google
6	PROXY	82%	Convergence AI
7	AGENT-E	73.1%	Emergence AI
8	RUNNER H 0.1	67%	H Company
9	WILBUR	60.6%	Academic Research
10	WEBVOYAGER 	59.1%	Academic Research
11	COMPUTER USE	52%	Anthropic

Coding Agents

- SWE-Bench & SWE-Agents
- <https://www.swebench.com/>

Leaderboard

Lite Verified Full Multimodal						
Model	% Resolved	Org	Date	Logs	Trajs	Site
🥇 Isoform	55.00	i	2025-01-14	✓	✓	🔗
🥈 Blackbox AI Agent	49.00	-	2024-12-20	✓	✓	🔗
🥉 Gru(2024-12-08)	48.67	🏆	2024-12-08	-	-	🔗
Globant Code Fixer Agent	48.33	G	2024-11-27	✓	✓	🔗
devlo	47.33	💬	2024-11-22	✓	✓	🔗
NEW 🏆✅ SWE-agent 1.0 (Claude 3.5 Sonnet + o1)	45.33	🖥️	2025-02-12	✓	✓	🔗
🏆 Kodu-v1 + Claude-3.5 Sonnet (20241022)	44.67	🤖	2024-12-07	✓	✓	🔗
🏆✅ OpenHands + CodeAct v2.1 (claude-3-5-sonnet-20241022)	41.67	🙌	2024-10-25	✓	✓	🔗
🏆 PatchKitty-0.9 + Claude-3.5 Sonnet (20241022)	41.33	🌐	2024-12-20	✓	✓	-
🏆 Composio SWE-Kit (2024-10-30)	41.00	⚡	2024-10-30	✓	✓	🔗
🏆 Agentless-1.5 + Claude-3.5 Sonnet (20241022)	40.67	I	2024-12-02	✓	✓	🔗
OpenCSG Starship Agentic Coder + GPT 4 (0806)	39.67	-	2025-01-13	✓	✓	🔗
Bytedance MarsCode Agent	39.33	📈	2024-09-12	✓	✓	🔗

Coding Agents

- Cursor Agent, JetBrains Junie Agent

The screenshot displays the JetBrains Junie AI agent interface. On the left, a code editor shows a Python file named `main.py` with the following content:

```
1 from langchain_openai import ChatOpenAI
2 from browser_use import Agent
3 import asyncio
4 from dotenv import load_dotenv
5
6 # Load environment variables (make sure you have OPENAI_API_KEY
7 load_dotenv()
8
9 async def main(): 1 usage
10     try:
11         # Initialize the AI agent with the task to build a route
12         agent = Agent(
13             task="Build a route from Limassol to Paphos in Google
14             llm=ChatOpenAI(model="gpt-4"), # Using GPT-4 for b
15         )
16
17         # Execute the browser automation task
18         await agent.run()
19
20     except Exception as e:
21         print(f"An error occurred: {str(e)}")
22
23 if __name__ == "__main__":
24     asyncio.run(main())
25
```

On the right, the Junie panel shows a task: "Create a browser example for routing in Google Maps." Below the task is a "Stop" button and a "Plan" section. The plan consists of five steps:

- ✓ 1. Review the current code structure and dependencies
- ✓ 2. Fix any obvious errors in the current code
 - ✓ - Fix the model name typo in main.py
 - ✓ - Add proper error handling
- ✓ 3. Enhance the example to properly demonstrate the route building functionality
 - ✓ - Add descriptive comments
 - ✓ - Ensure proper imports
- ✓ 4. Test the implementation
- ✱ 5. Verify the changes work as expected
 - - Ensure .env file exists with OPENAI_API_KEY
 - - Document the requirements for users

Below the plan, there is a section titled "Open requirements.txt" with the text: "The project has been initialized for browser automation, and the requirements have been identified. The next step is to ensure all necessary dependencies for the project are in place." and a "Click to expand" link.

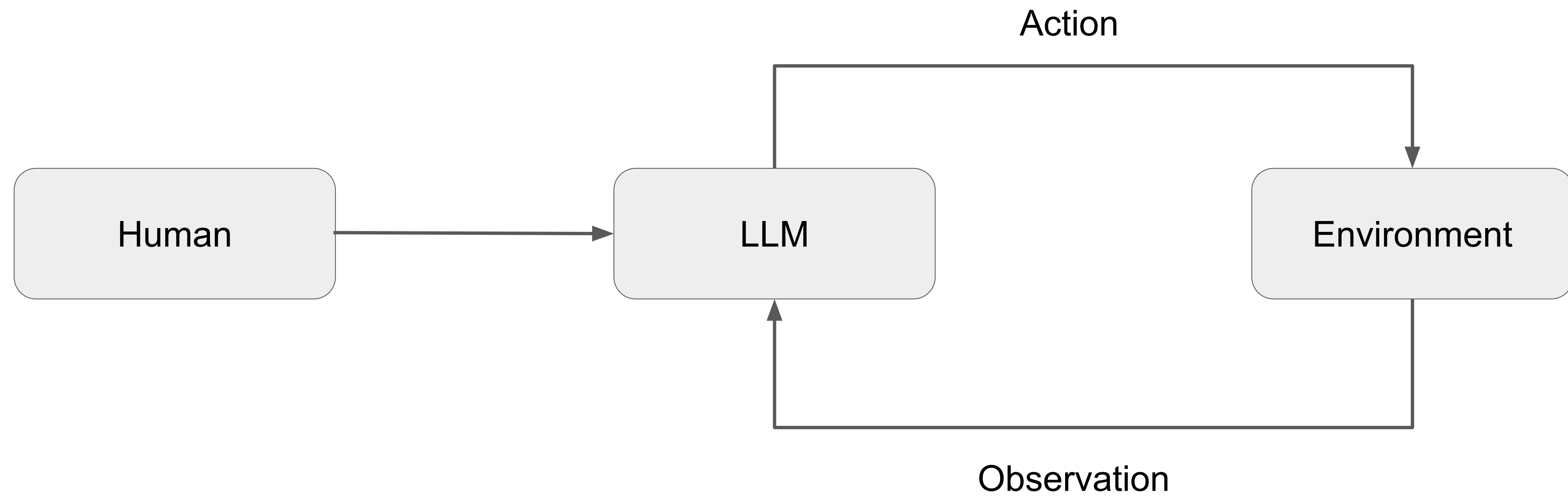
At the bottom right of the Junie panel, there is a dark blue button with a white downward arrow.

What is an Agent?

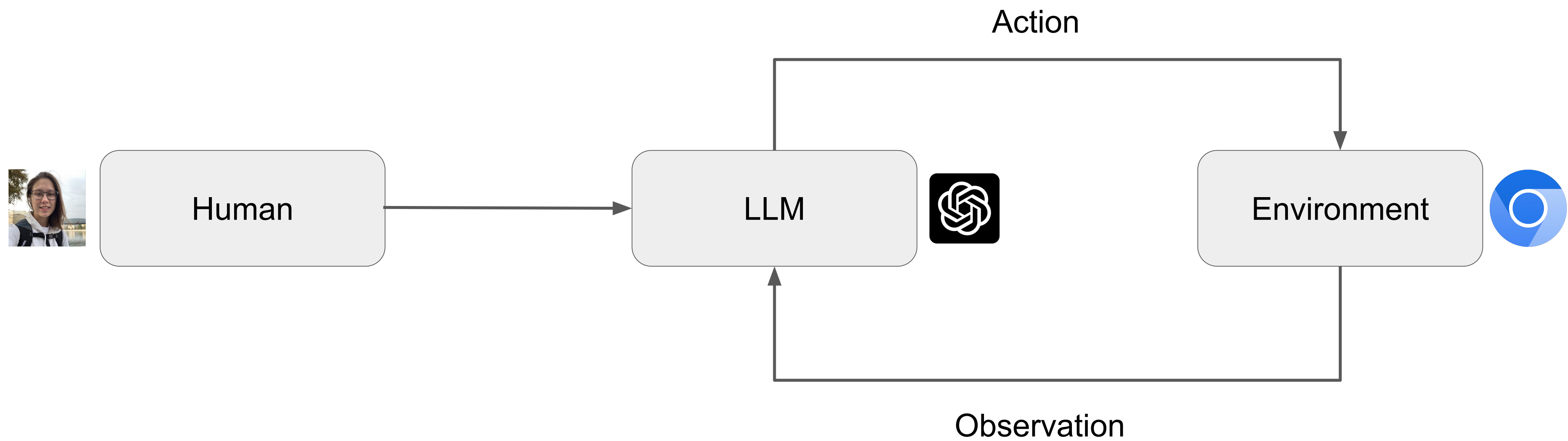
1. AI systems that are more complex and go beyond a single prompt
2. AI systems capable of operating independently of humans, and/or controlling your device

We will mostly cover (1) but not (2)

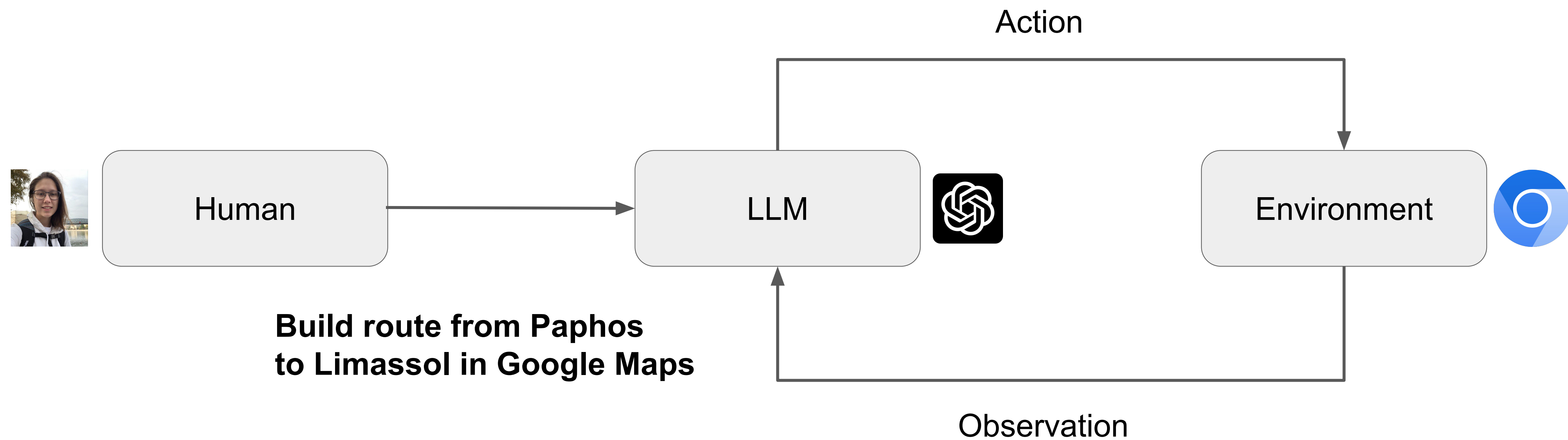
What is an Agent?



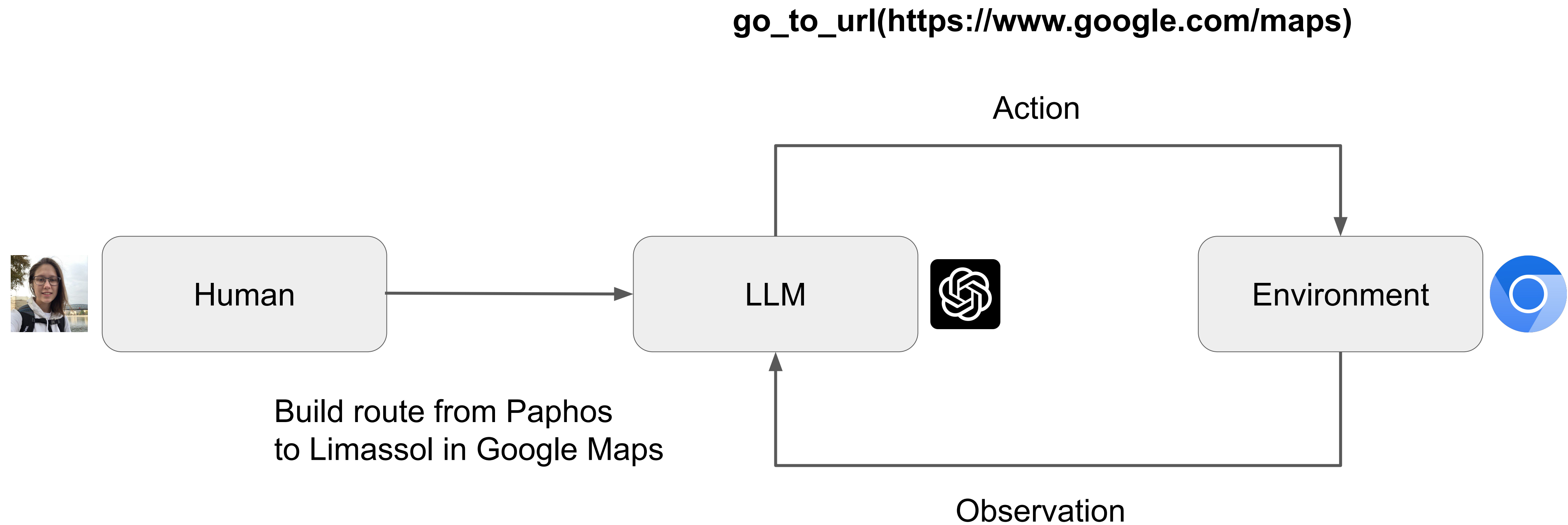
What is an Agent?



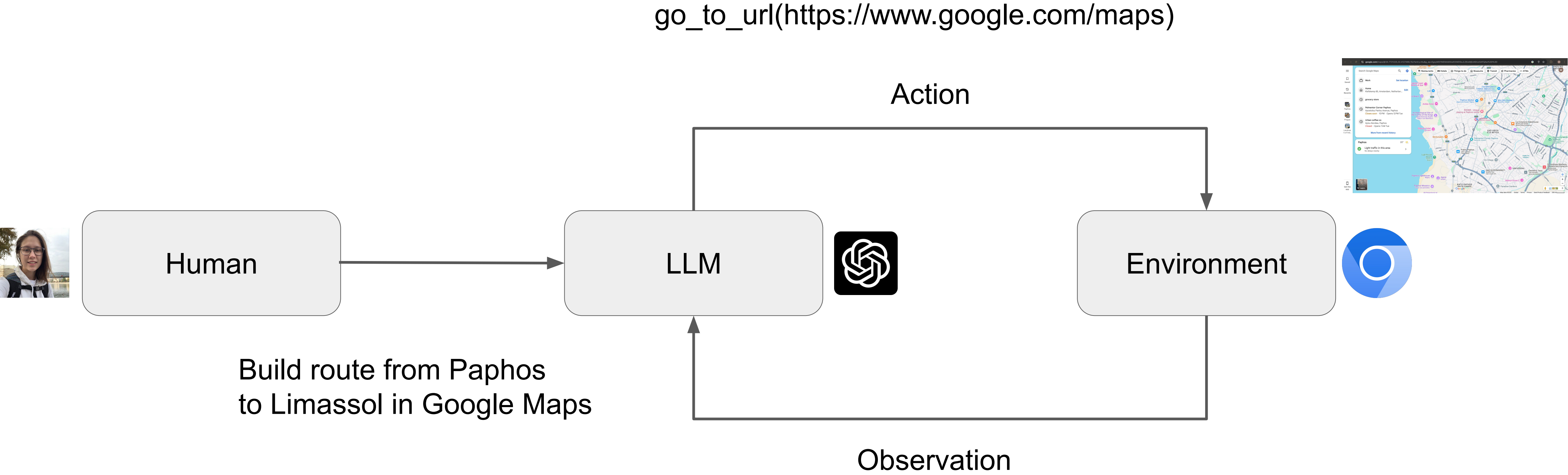
What is an Agent?



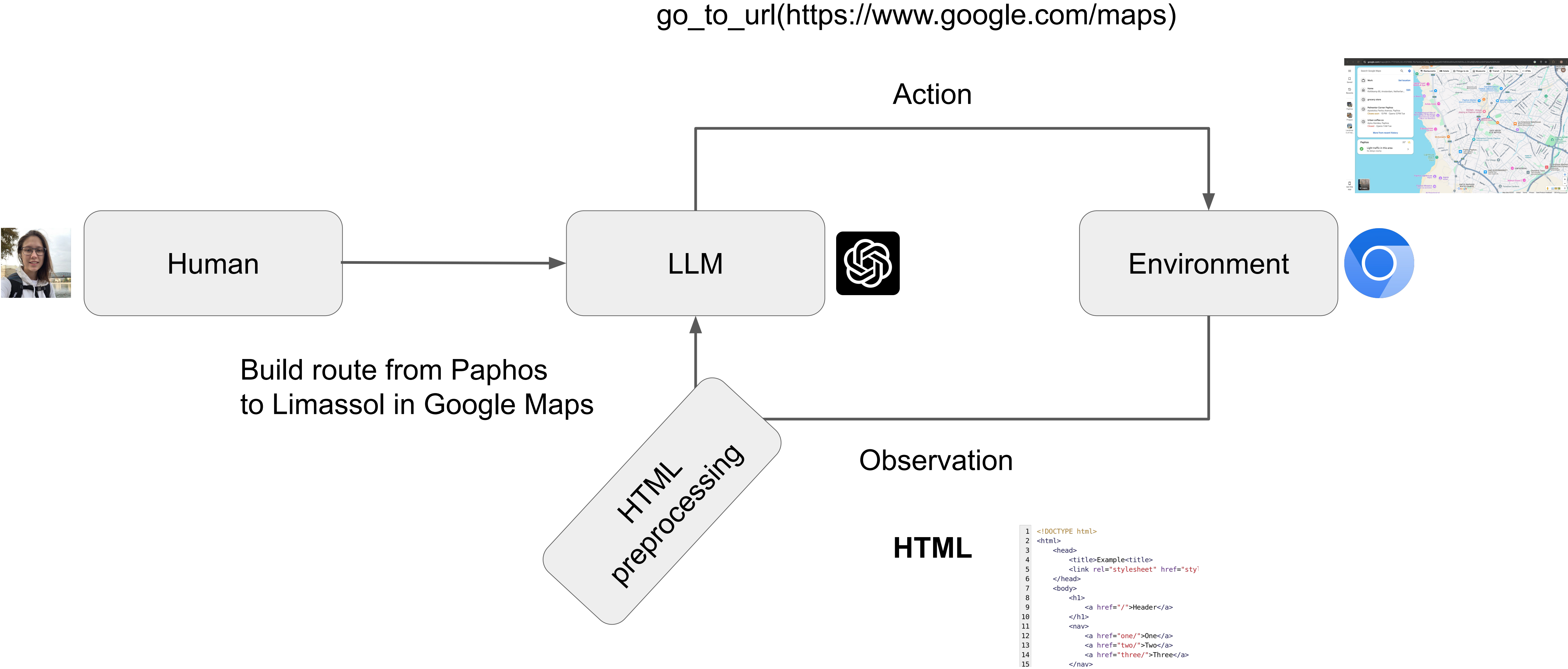
What is an Agent?



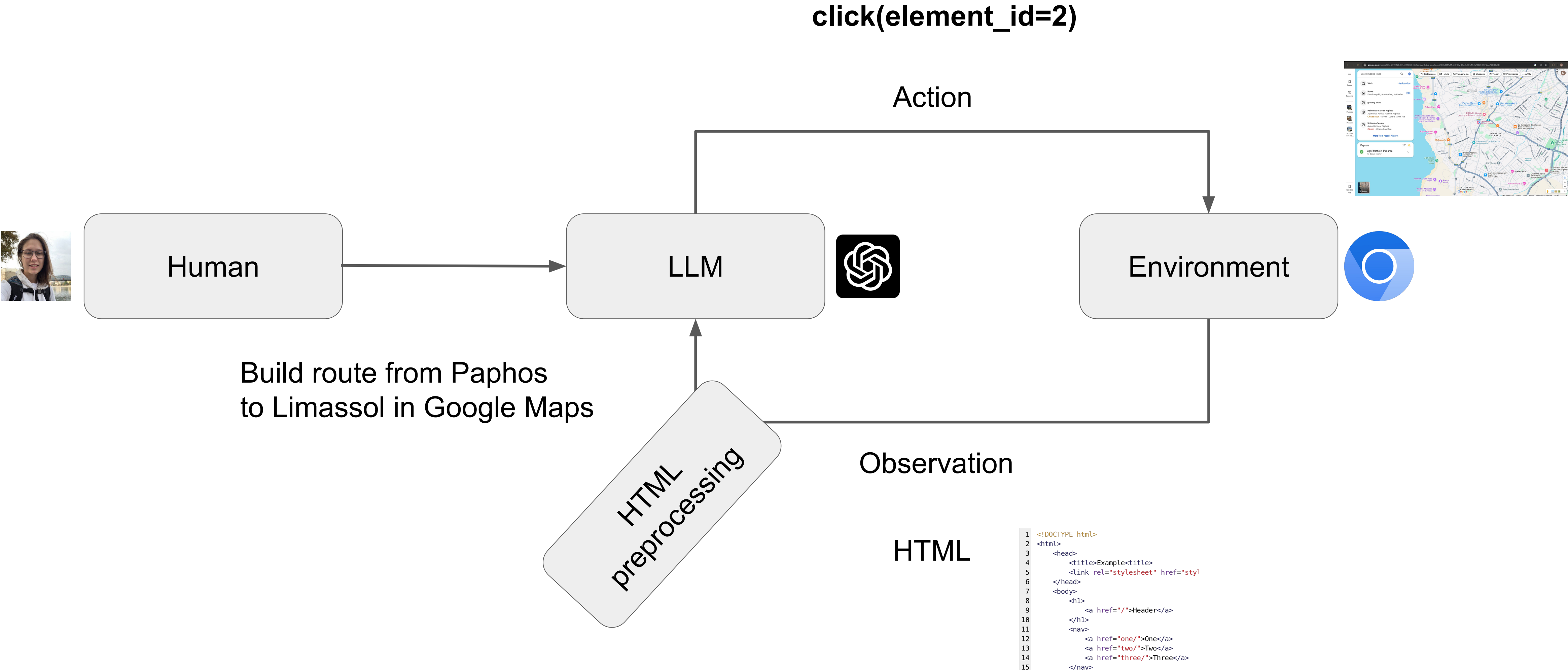
What is an Agent?



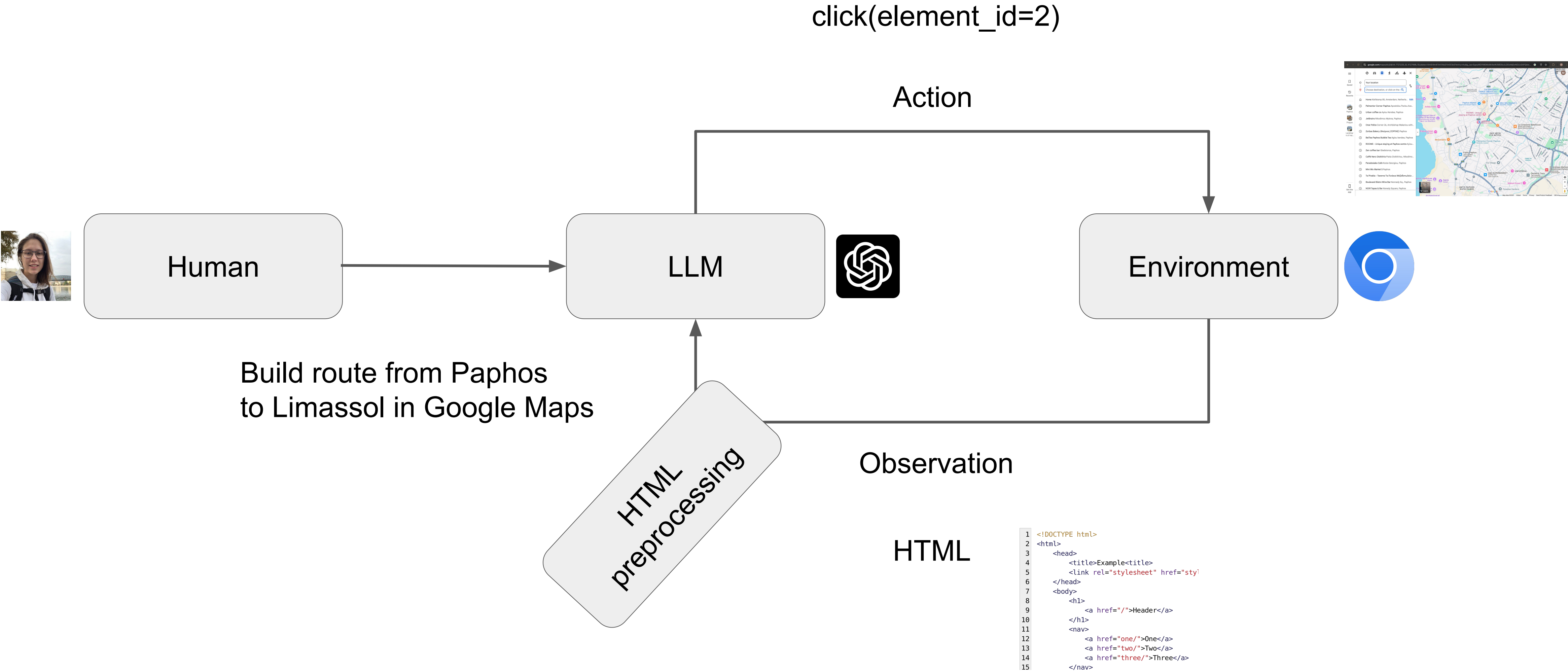
What is an Agent?



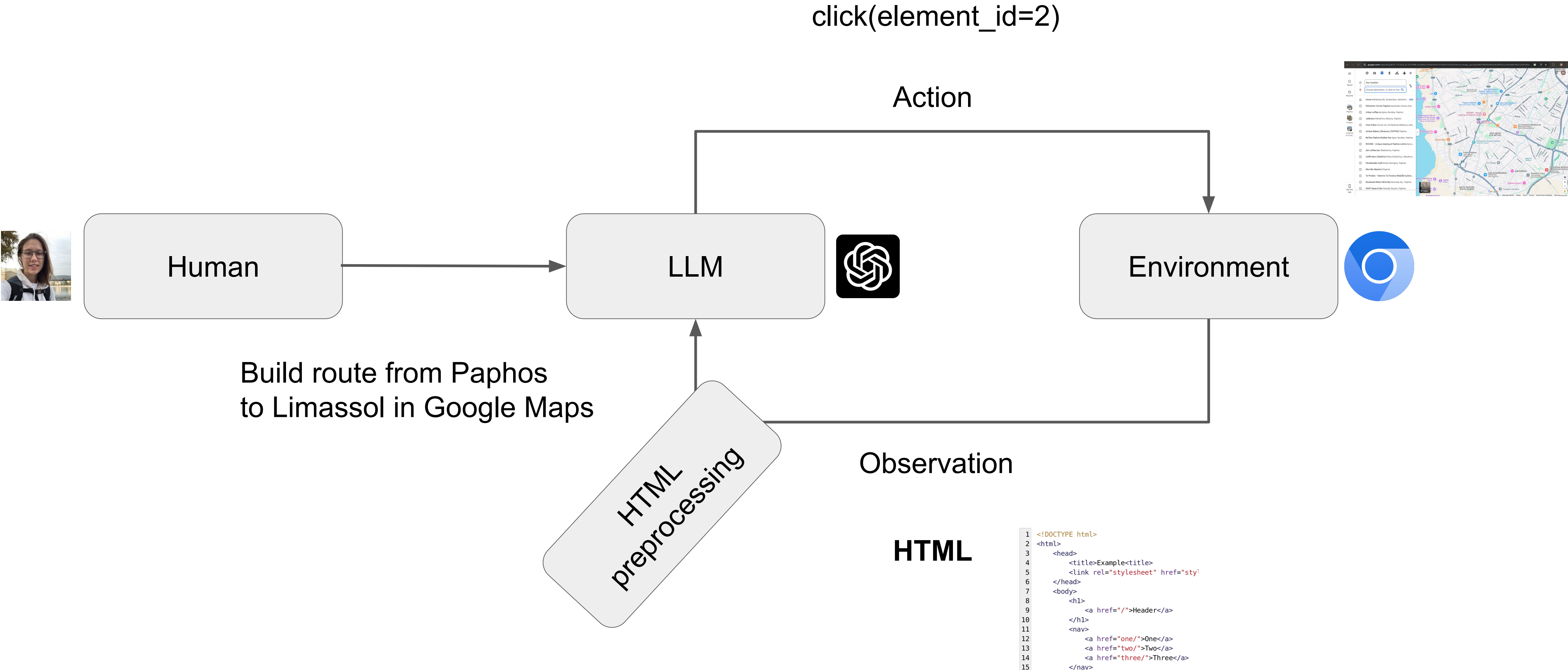
What is an Agent?



What is an Agent?



What is an Agent?



Need?

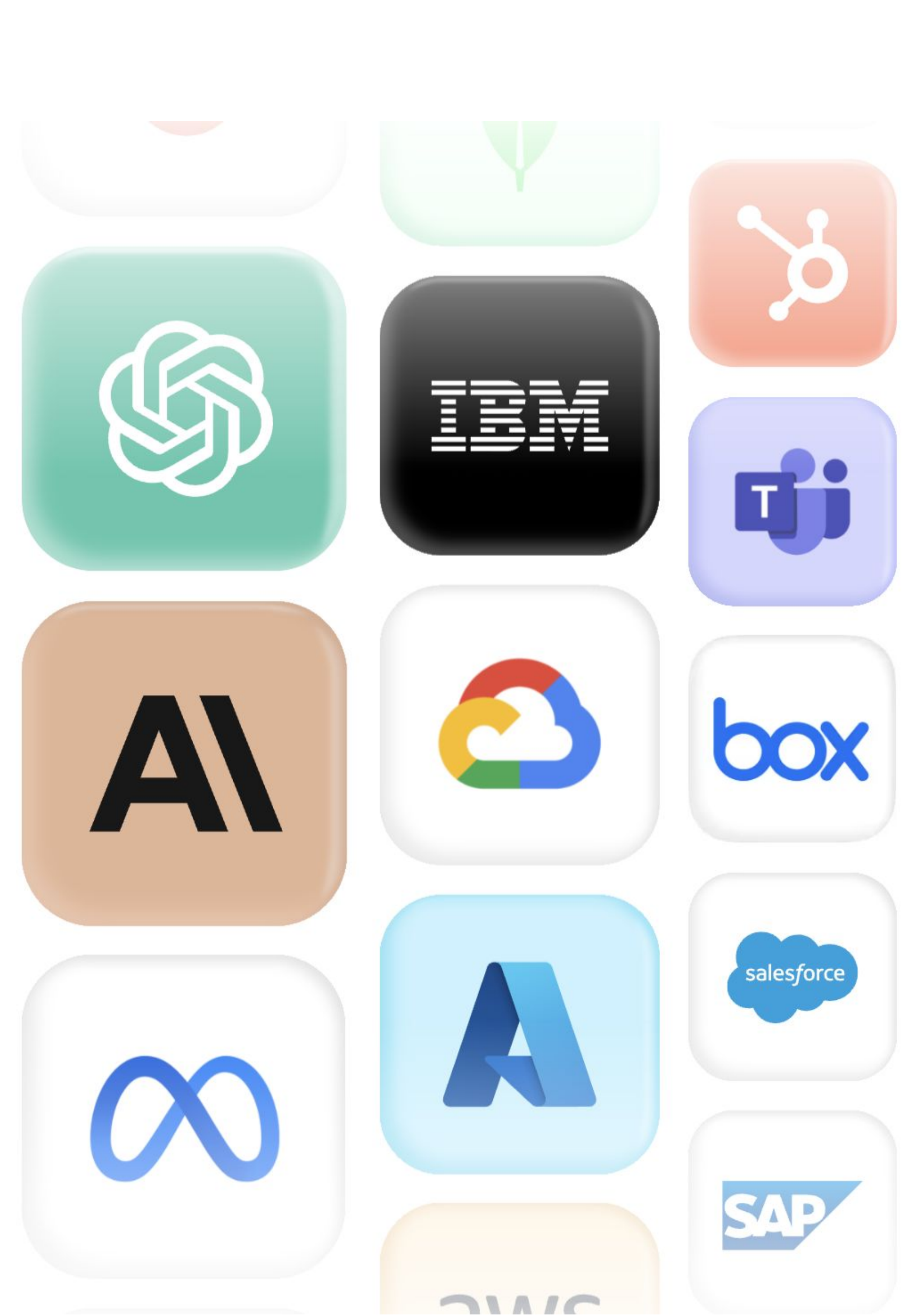
- Multi-task intent based systems
- Tasks where you want to provide authority to LLMs
- Application similar to finite state machines, behavior trees
- Robots

Hype?

- Production level use cases are usually finite
- Simplicity is always the best
- “Agents” = “LLM Systems designed for a particular task/set of tasks”
- More like LLM powered automation

Tools you can use to build Agentic Systems

- [LangGraph](#) (from LangChain)
- [CrewAI](#)
- [AutoGen](#) (from Microsoft)
- ...



Workshop: How to build an agent using LangGraph?



<https://github.com/Slinkolgor/Agents101>