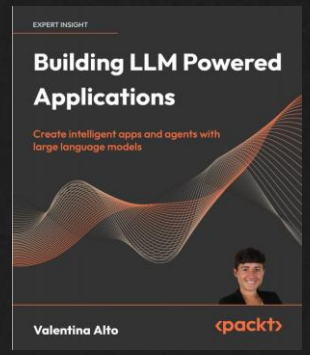
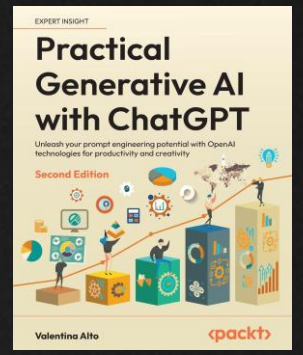
About Valentina Alto

After completing her Bachelor's degree in Finance, Valentina Alto pursued a Master's degree in Data Science in 2021. She began her professional career at Microsoft as an Azure Solution Specialist, and since 2022, she has primarily focused on working with data and AI solutions in the Manufacturing and Pharmaceutical industries. Valentina collaborates closely with system integrators on customer projects, with a particular emphasis on deploying cloud architectures that incorporate modern data platforms, data mesh frameworks, and applications of Machine Learning and Artificial Intelligence. Alongside her academic journey, she has been actively writing technical articles on Statistics, Machine Learning, Deep Learning, and AI for various publications, driven by her passion for AI and Python programming.





21-06-2025 | Confidential

AI AGENTS FOUNDATIONS – Day 1 | Part 2

By Valentina Alto

Agenda

01 Kick-Off and Introduction

02 From Content Generation to Agentic AI

03 Introduction to AI Agents

04 Demo Time

05 Introduction to Agentic Protocols: MCP and A2A

06 Conclusion

01

Kick-Off and Introduction

Rapid Advancements of Generative AI

Text generation

Creating human-like text, including articles, poetry, stories, and even computer code

November 2022

RAG

More context-aware, capable of maintaining coherent and relevant multi-turn conversations.

Early 2023

Multi-modality

Understanding and generating data in different format including images, audio and video.

Mid 2023

AI Agents

Generating sophisticated function or tool calls that can trigger actions in external systems

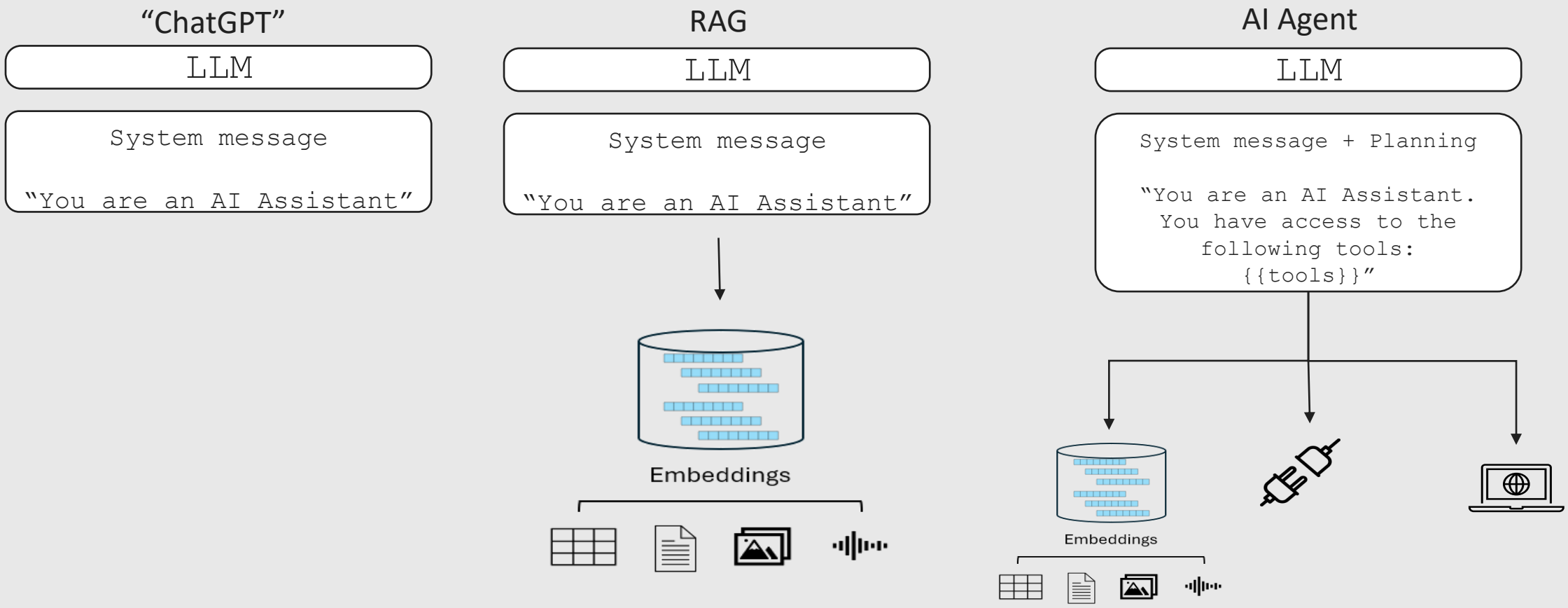
We are here!

Mid 2024

Multi-agents

Letting multiple AI agents interacting among each other in an autonomous way

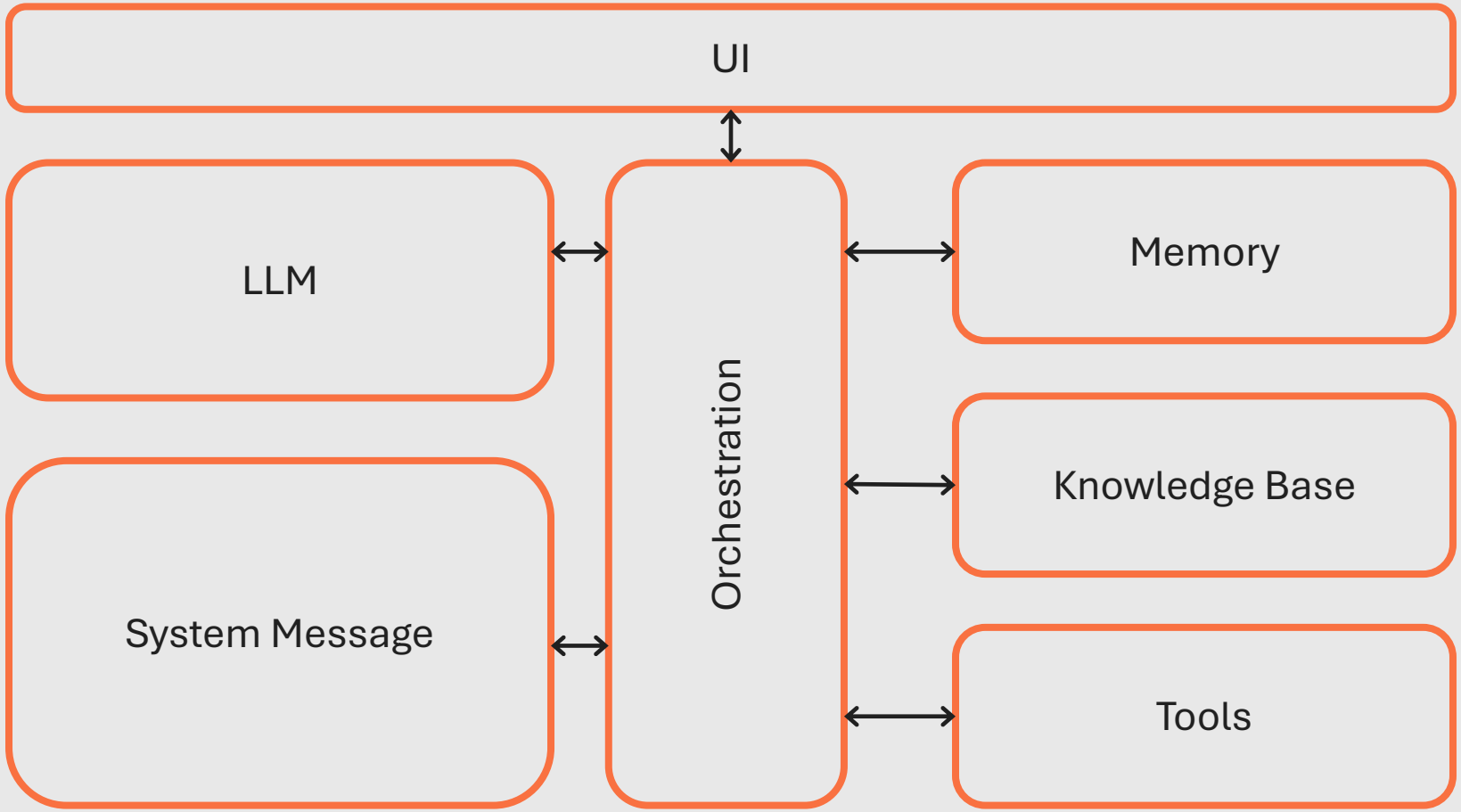
AI Workflows



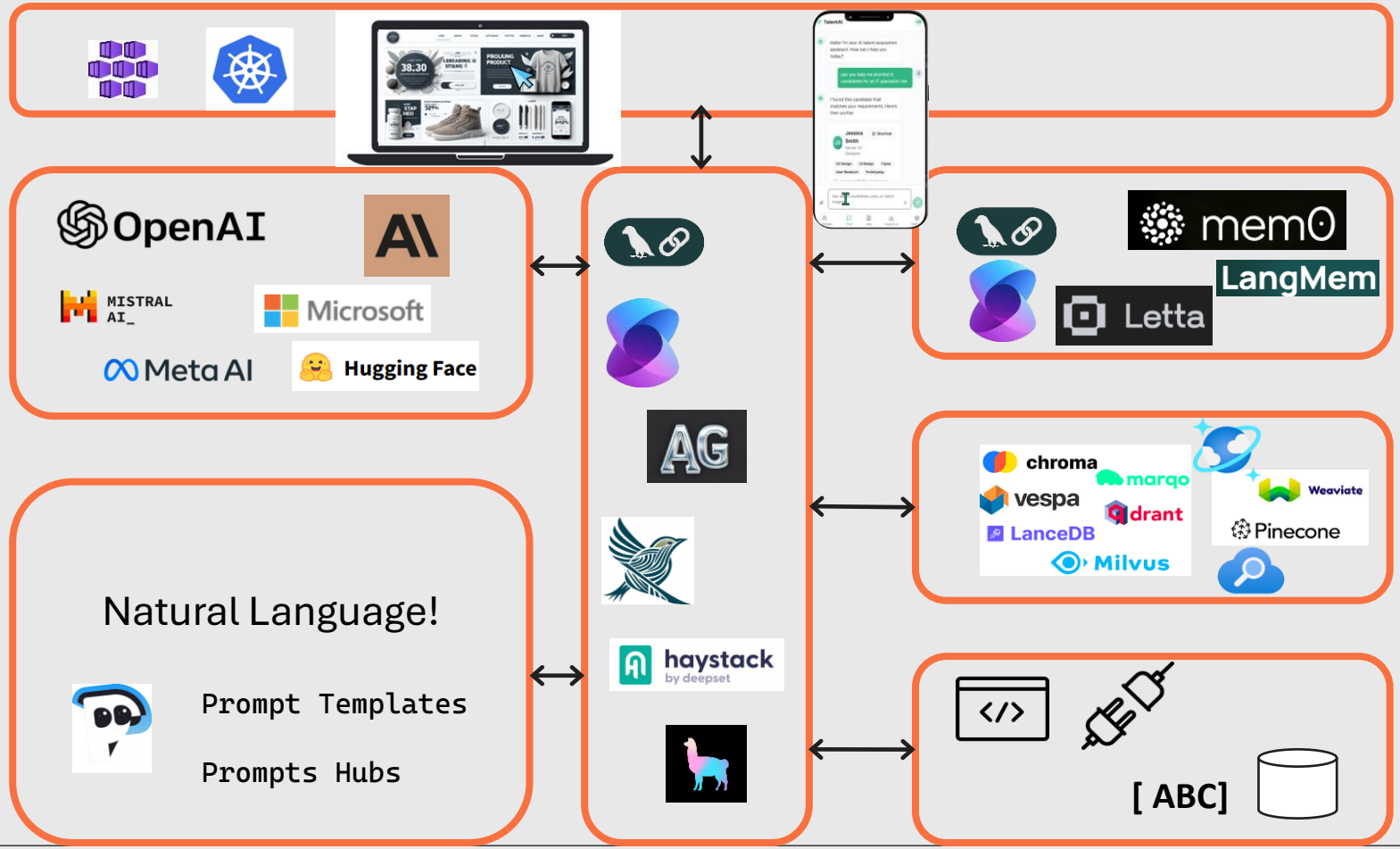
02

What is an agent?

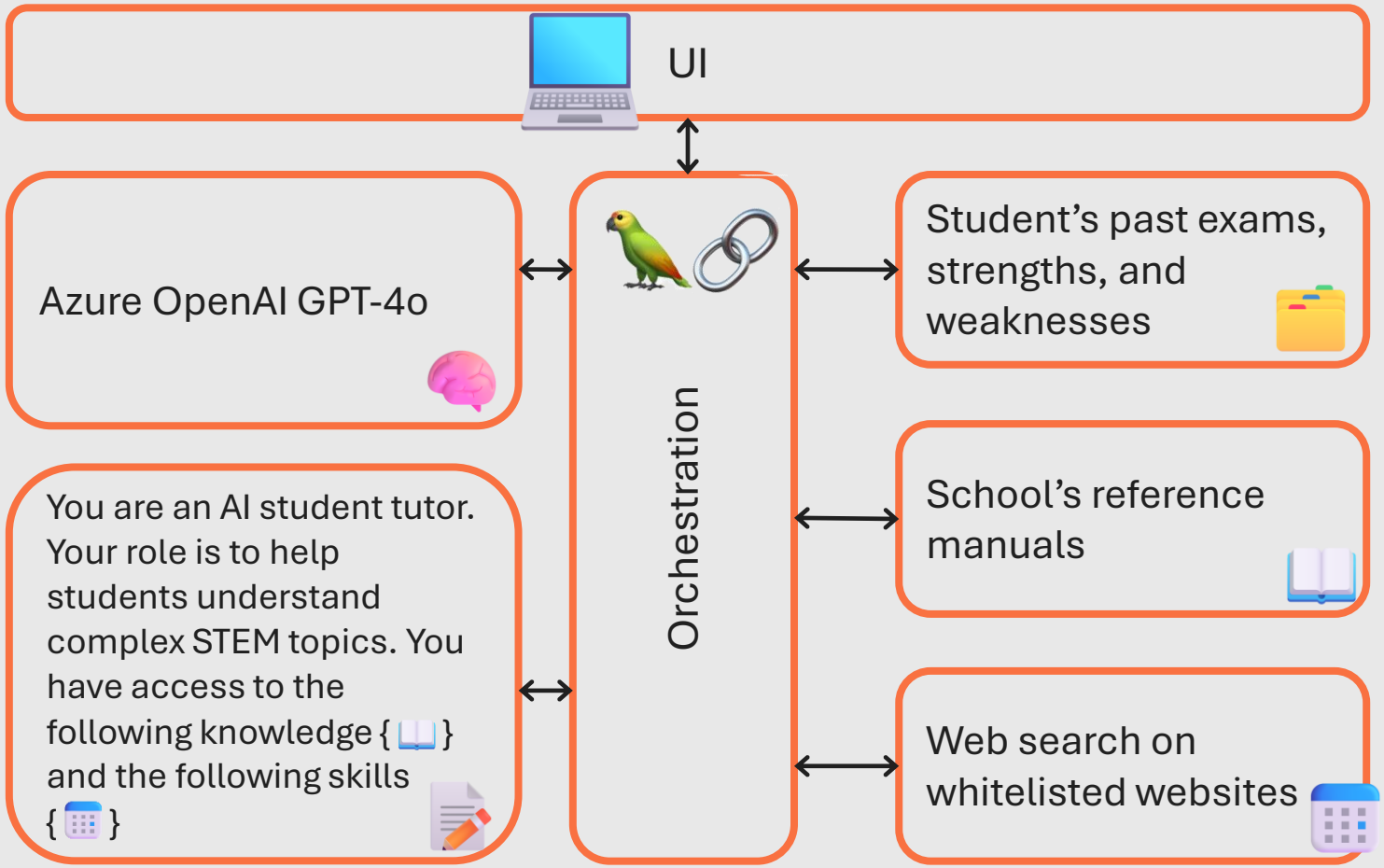
Anatomy of an AI Agent



Anatomy of an AI Agent



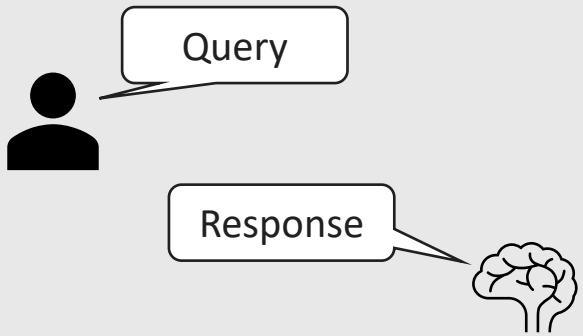
Example: AI-Powered Student Tutor

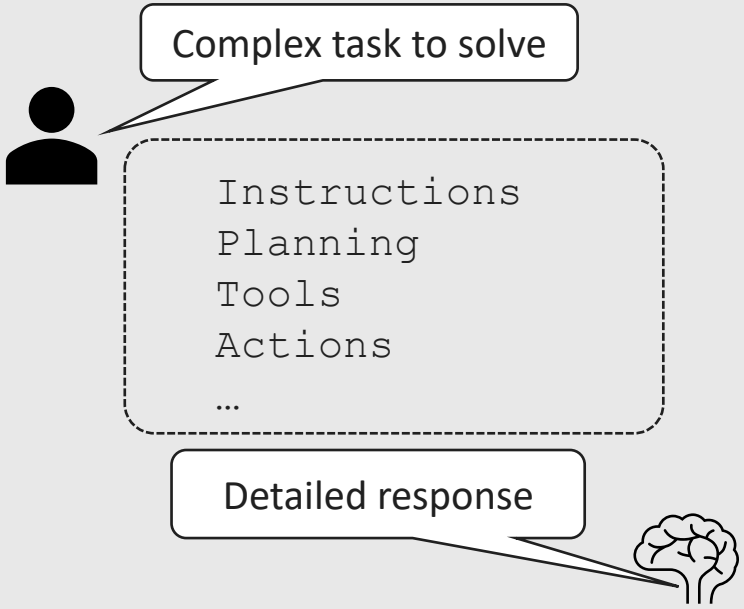


Paradigm Shift

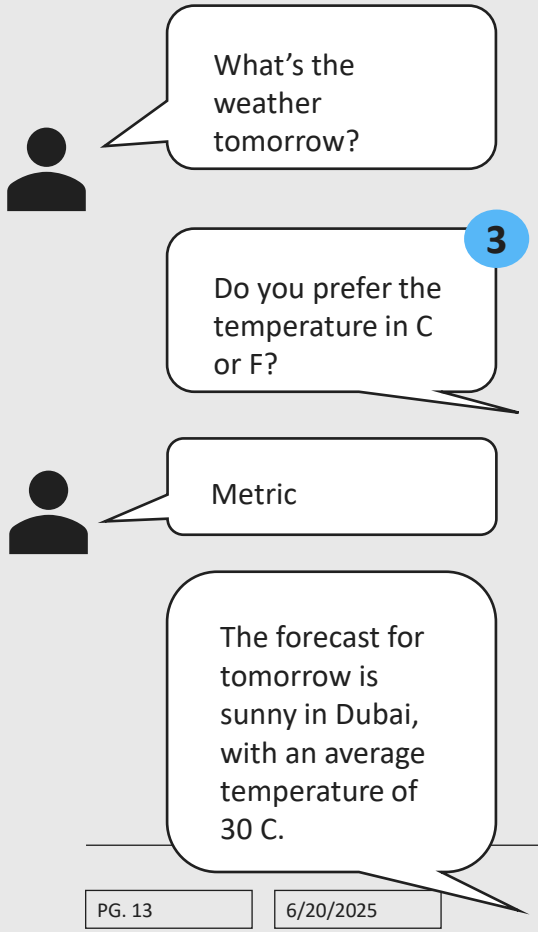
From instantaneous, generic conversations…

…to highly specialized, complex tasks including autonomous decisions





Architecture



Task Decomposition

Plan

I need to invoke the weather tool

I need the "city" and "metric" parameters. To retrieve the city, I can sue the location tool.

To retrieve the unit, I'll ask the user

I can now invoke the weather tool

Now I have the answer

"You are a helpful AI Assistant"

1 4 **Weather Tool**

get_weather (city, unit)

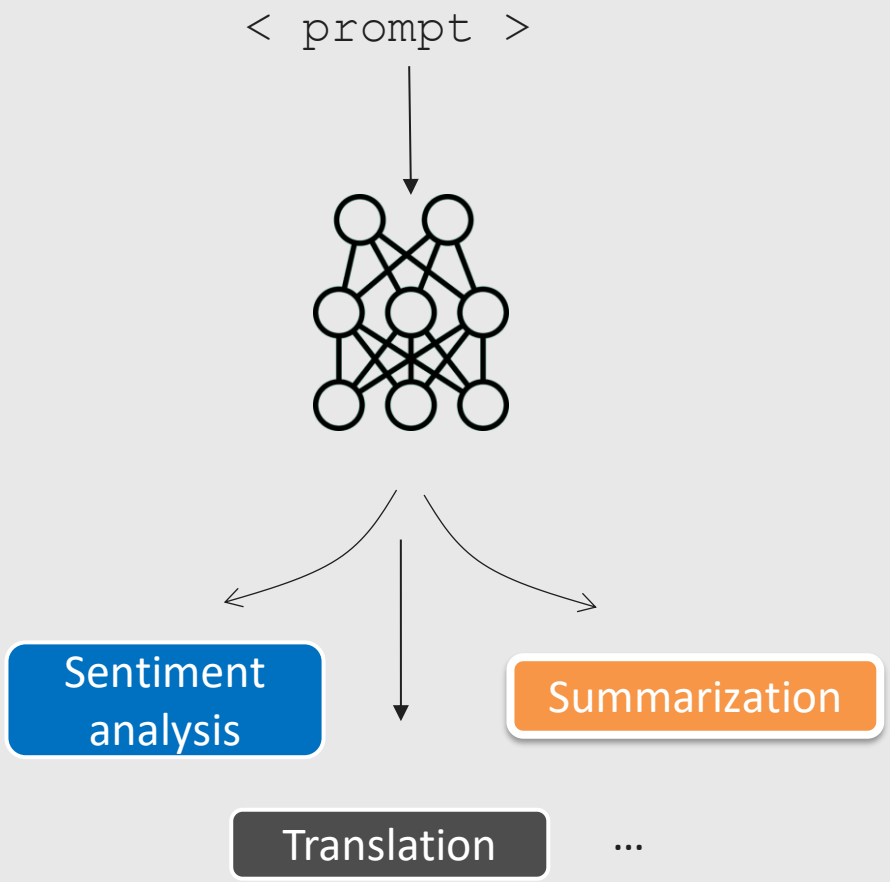
"This is a tool to get the weather"

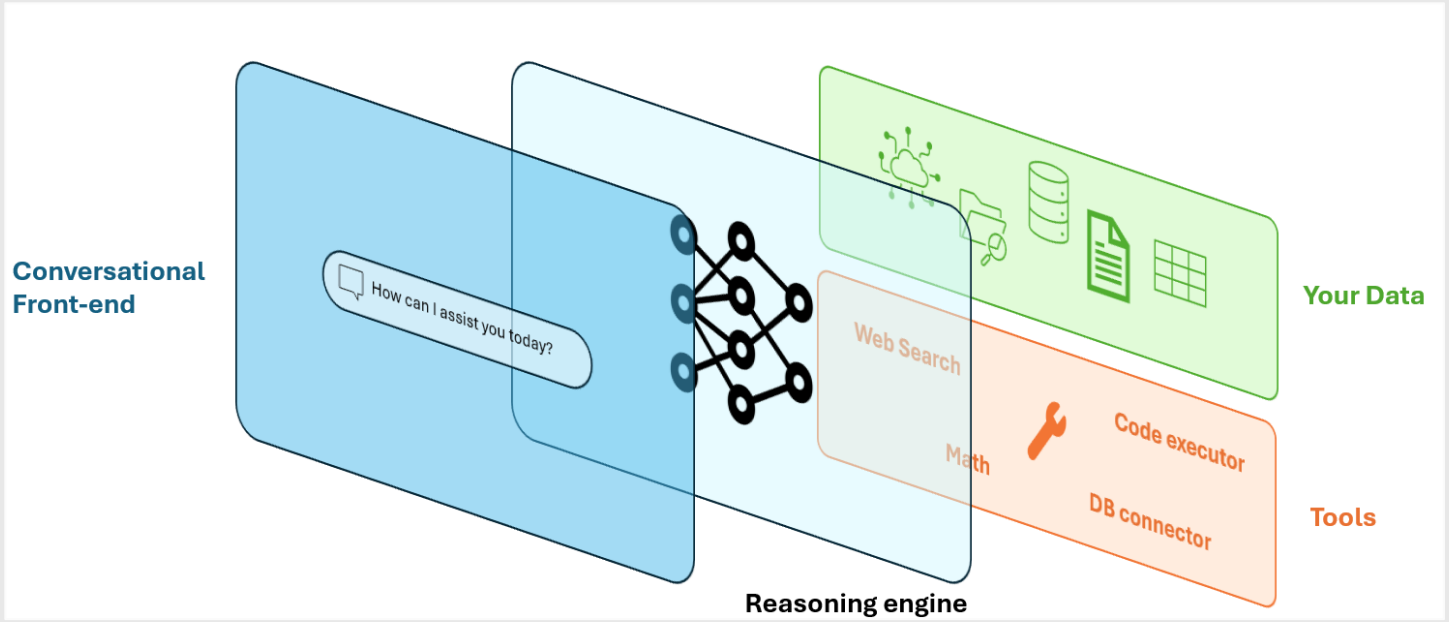
2 **Location Tool**

get_location ()

"This is a tool to get the current location"

Large Language Model





System Message

```
You are a helpful Civil Law Legal assistant.You
receive queries from taxpayers. Refer to the
Italian civil law system.
```

```
Always answer, providing a reference to the
Codice Civile.
If the answer refers to another Legal domain, do
not answer.
```

```
Always refer to the user with polite and
empathetic style, making sure not to use an
over-complicated language.
```

```
Before giving the response, review it and assess
whether it might be potentially misleading. In
that case, respond "I cannot help with that".
```

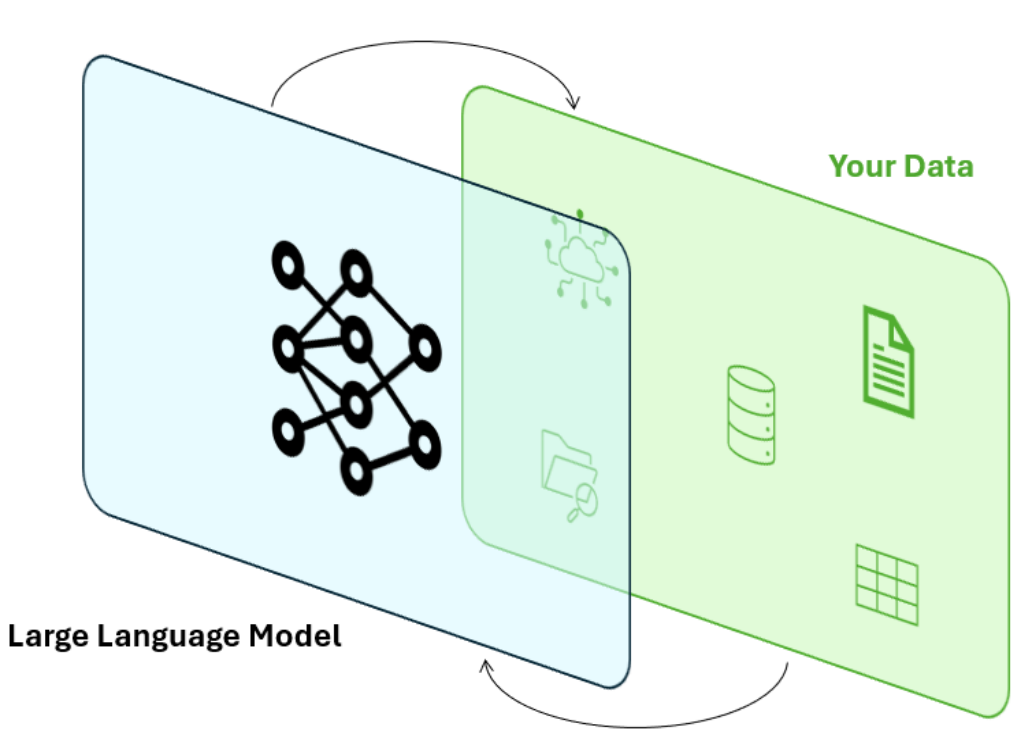
Setting the goal and scope

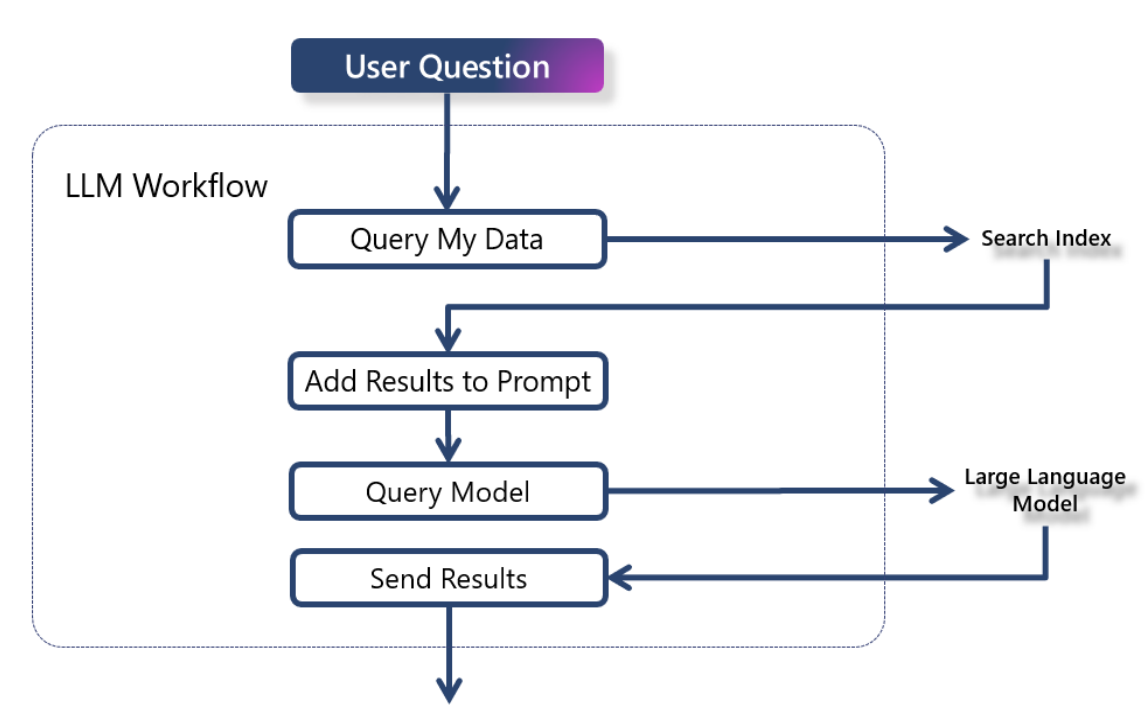
Grounding the model to relevant documents and preventing hallucination

Setting the style of conversation

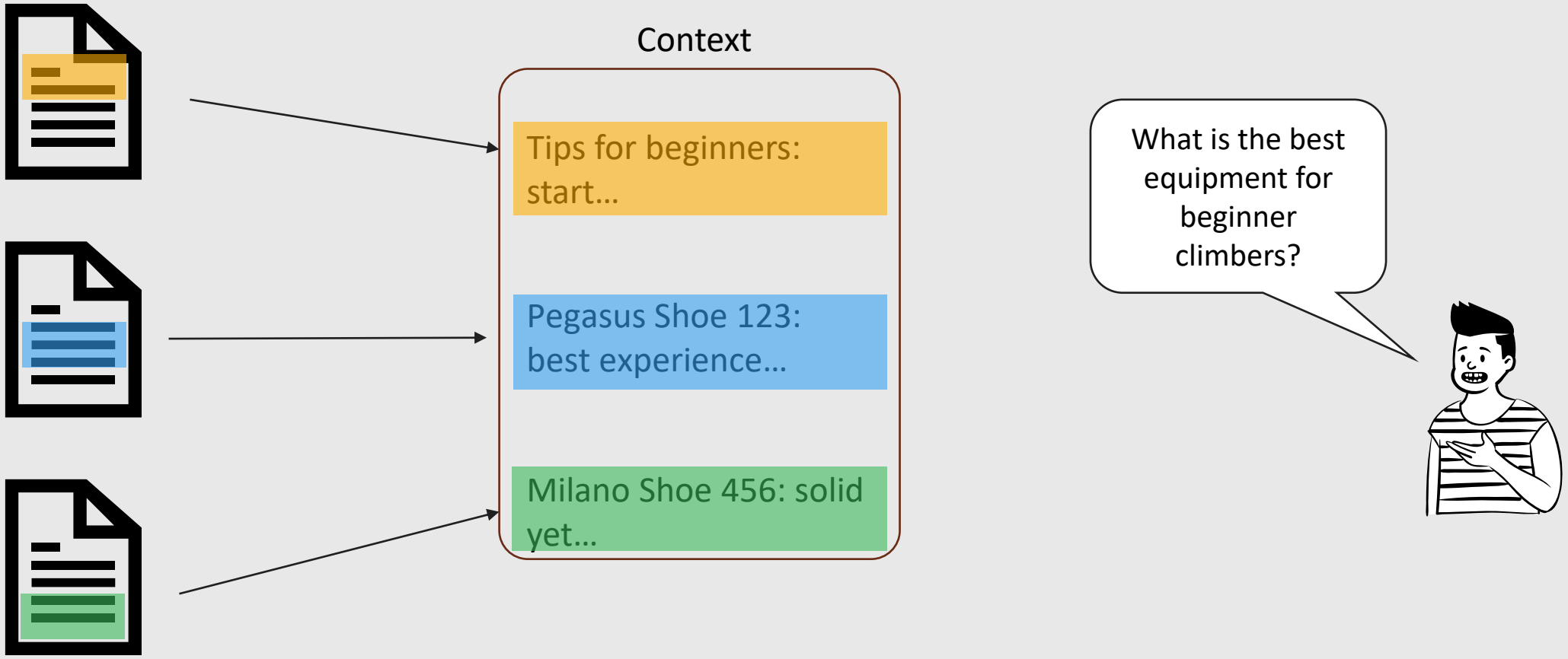
Incorporating responsible AI features

Knowledge – 1/8

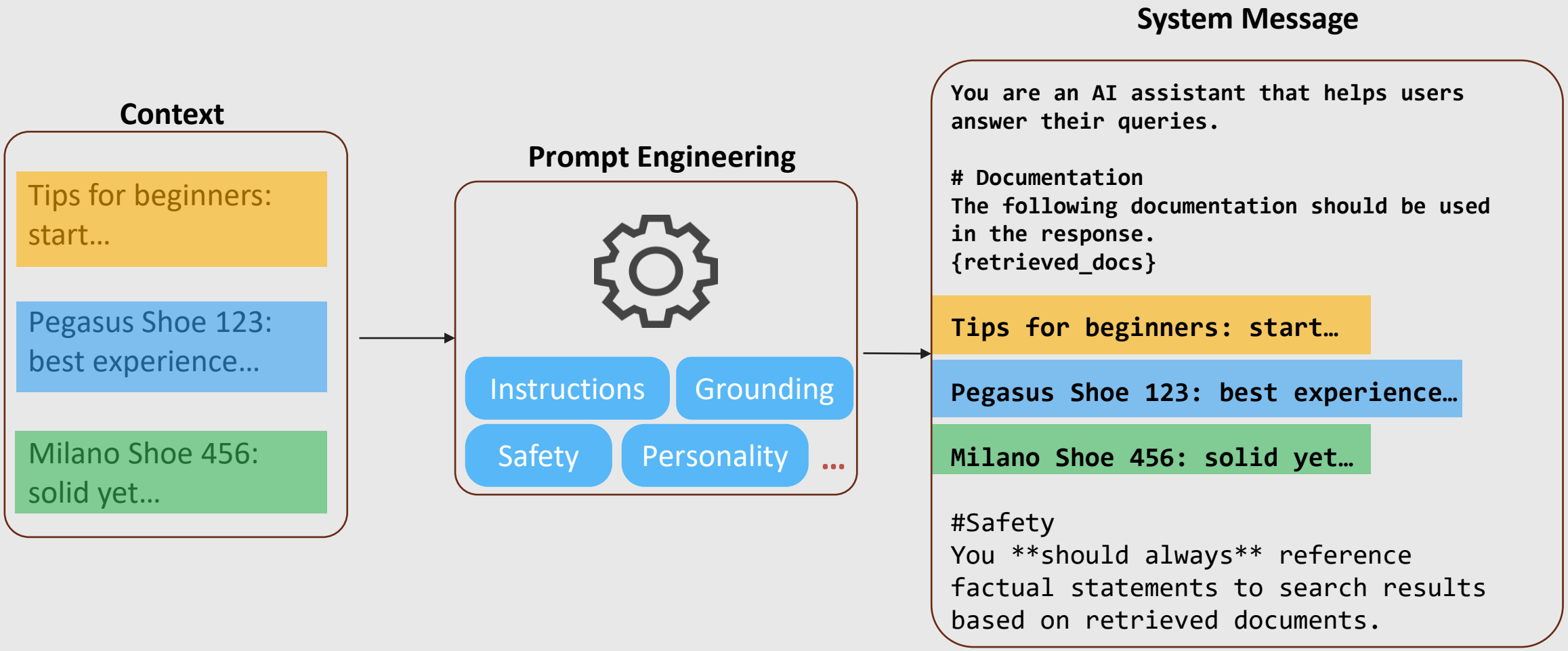




Knowledge – 2/8



Knowledge – 3/8



Knowledge – 4/8

System Message

```
You are an AI assistant that helps users
answer their queries.

# Documentation
The following documentation should be used
in the response.
{retrieved_docs}

Tips for beginners: start…

Pegasus Shoe 123: best experience…

Milano Shoe 456: solid yet…

#Safety
You **should always** reference
factual statements to search results
based on retrieved documents.
```

User's query

What is the best equipment for beginner climbers?

Generative Model (e.g. GPT-4

"According to the catalogue, if you are about to start climbing…"

Knowledge – 5/8

"cat" → Embedding model → { 0.8 , 0.9 , -0.3 , -0.2 }

An embedding is a way of representing high-dimensional, non-numeric data, such as words or sentences, in a lower-dimensional space, such as vector.
A text embedding can capture the semantic and syntactic features of the text, such as meaning, context, and similarity.

King-Man+Woman ≈ Queen

Knowledge – 6/8

Agent	Transcription
A	“Hello I’m calling for…”
B	“Am I speaking with …”

→ Embedding →

Agent	Transcription	Embedding
A	“Hello I’m calling for…”	[1 4 … 4], […]
B	“Am I speaking wi th…”	[2 3 … 9], […]

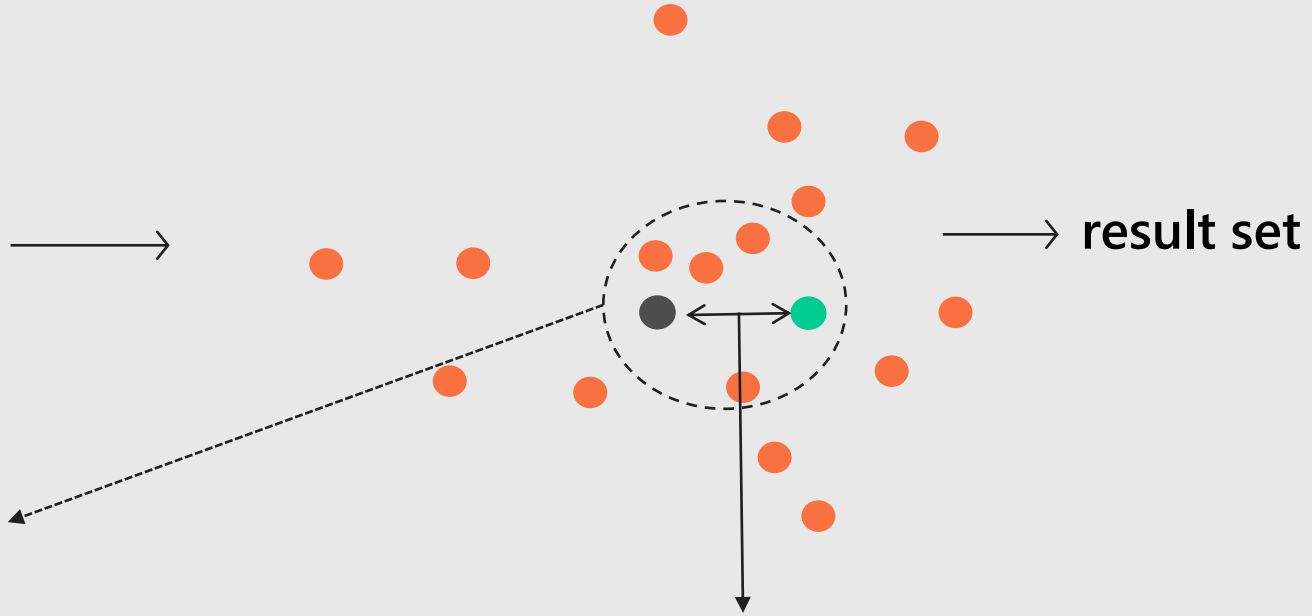
Knowledge – 7/8

user input

"What is a neutron star?"

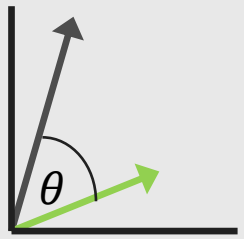
→ **embedding** →

[13 33 34 13 ...]



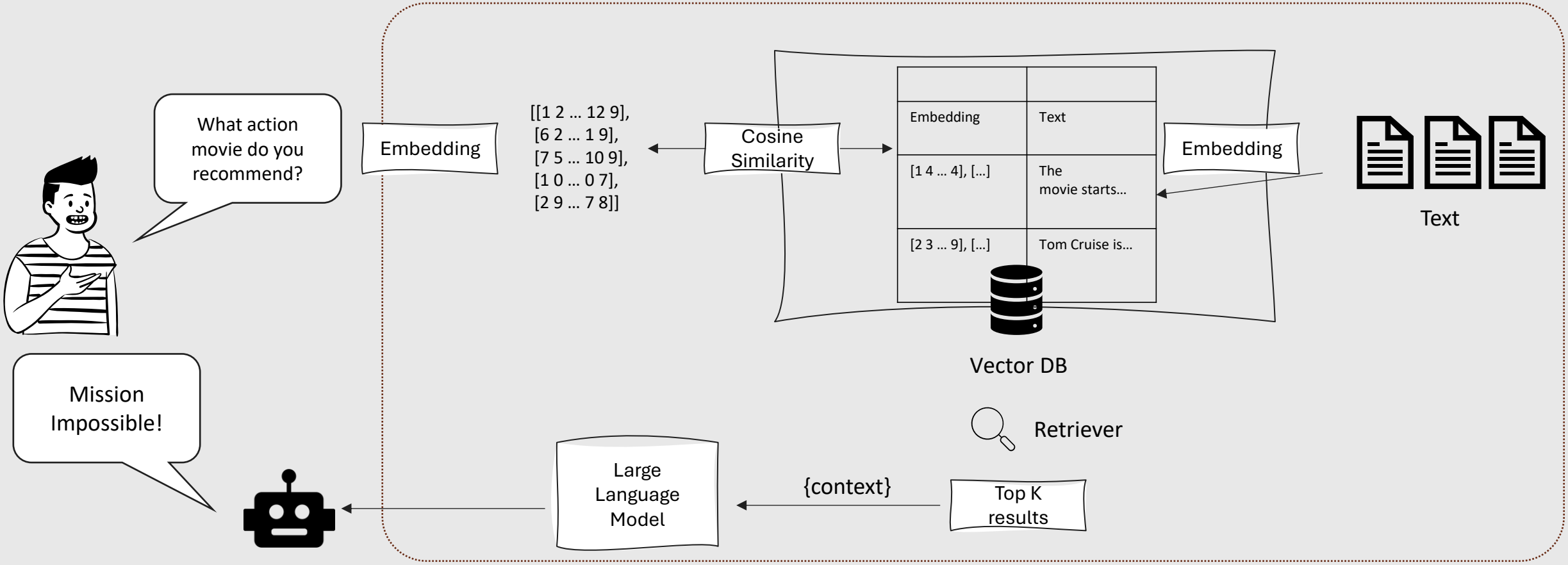
→ **result set**

K-nearest Neighbours with **Cosine similarity** distance



$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Knowledge – 8/8



Tools – 1/3

Tools allows AI Agents to interact with and take actions in the external world.

Name

Description

Content

!!!

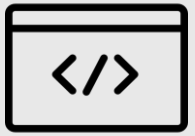
This is key! From descriptions, LLMs decide whether or not to invoke the tool.

Pre-built connectors

```
import CRMConnector
Connector = CRMConnector(cliend_id, api_key, …)
```

Custom functions

```
def get_weather(location, metric):
            …
            return …
```



Semantic Skills

```
class SemanticSkills:
            @kernel_function(name= “summarizer”,
            description=“summarize a list of docs”
```

[- - -]

Knowledge Bases

```
retriever = vector_db.as_retriever()
tool = Tool(retriever, name, decscription)
```

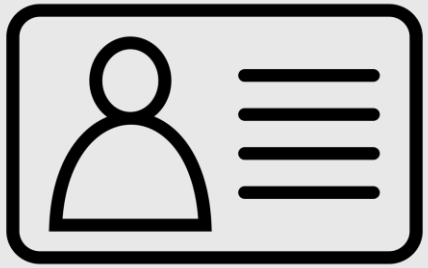
Tools – 2/3

```python
def get_stock_price(ticker: str) -> float:
    """Fetch the latest stock price for a given ticker symbol from Yahoo Finance"""
    stock = yf.Ticker(ticker)
    return stock.history(period="1d")["Close"].iloc[-1]
```

Name

Description

Content



Tools – 3/3

According to sentence X, court decision…

What's the price of MSFT stock today?

Invoking get_stock_price…

"..according to article XX…"

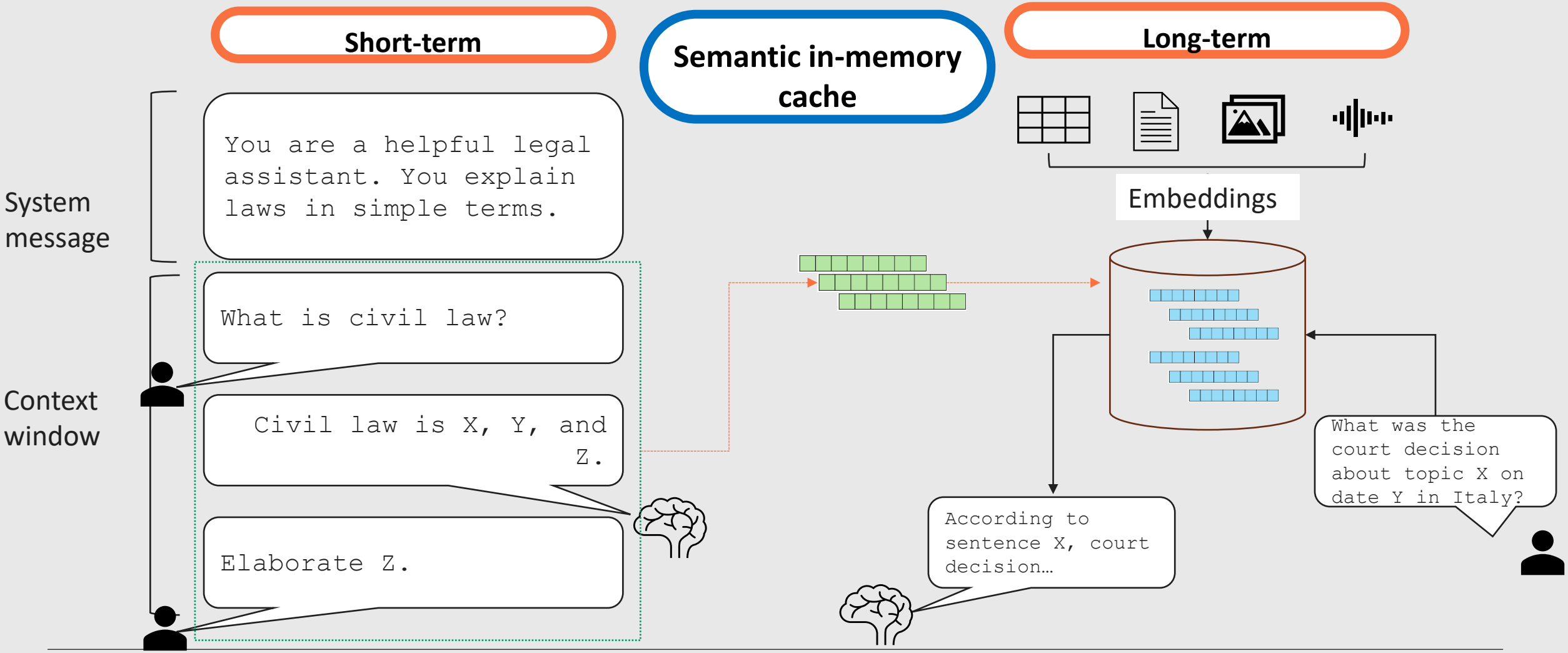
I now know the final answer.


```
{
    "name": " get_stock_price ",
    "description": "Fetch the stock prices
given a ticker."
    "parameters": {..}
}
```

You are a helpful Financial Assistant assistant.
You have access to the following tools:
{{tools}}

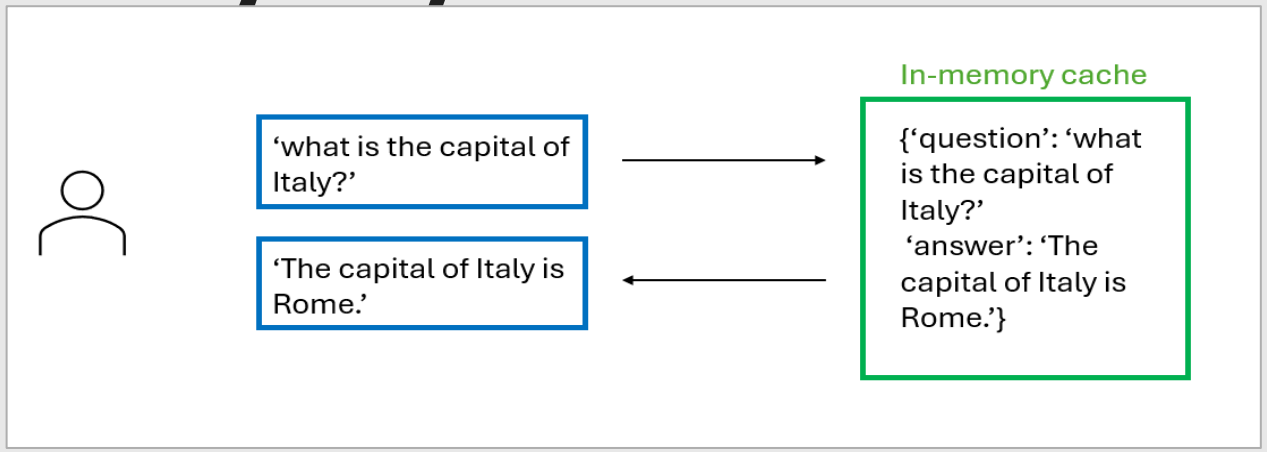
Thought: I need a tool to solve this question.
Let's see which tool can address the query.

Memory – 1/2

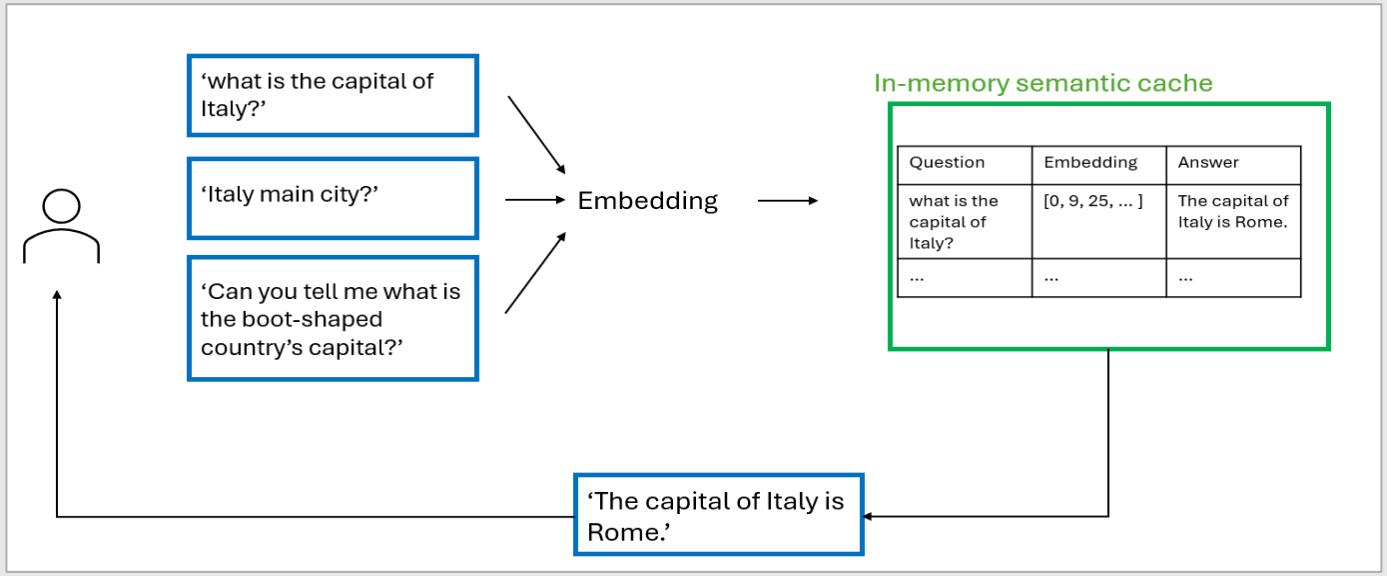

Short-term
Semantic in-memory cache
Long-term
System message
Context window
You are a helpful legal assistant. You explain laws in simple terms.
What is civil law?
Civil law is X, Y, and Z.
Elaborate Z.
Embeddings
What was the court decision about topic X on date Y in Italy?
According to sentence X, court decision…

Memory – 2/2

Key-word in-memory cache



Semantic in-memory cache



Orchestrator

LangChain

Semantic Kernel

LangGraph

AutoGen AG

....and many more!

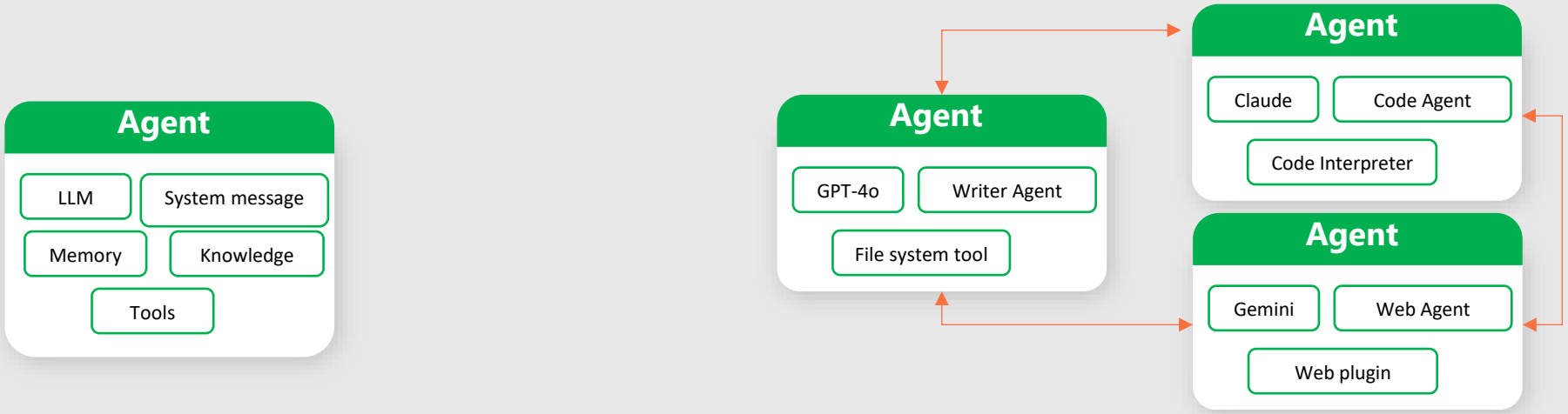
AI orchestrators provide the building blocks for designing, structuring, and managing intelligent workflows—enabling developers to compose multi-step reasoning, tool use, and agent collaboration into modular, scalable applications.

- 🧱 Abstractions for LLM Workflows
- 🔌 Pre-Built Integrations
- 🧠 Memory and State Management
- 🕹️ Multi-agent Collaboration
- 🧪 Debugging, Testing, and Evaluation Tools
- 🧩 Extensibility

03

What Happens If We Put Multiple Agents into the Same Room?

Scaling Your AI Agent



From single agent.....

- Too many tools, too hallucinations
- Growing context
- Hard to handle dynamic tasks

...to multi-agent

- Manageability
- Predictability
- Flexibility

Agents' ID Card


```
web_agent = InitializeAgent(
    name="web_agent",
    system_message="""
  You are an expert web surfer.
  Given the user's query, you find the best updated content from the web.

  """,
    description="agent that scrapes the web and retrieves up to date information",
  tools = [web_scraping_tool, captcha_tool]
)
```

Name

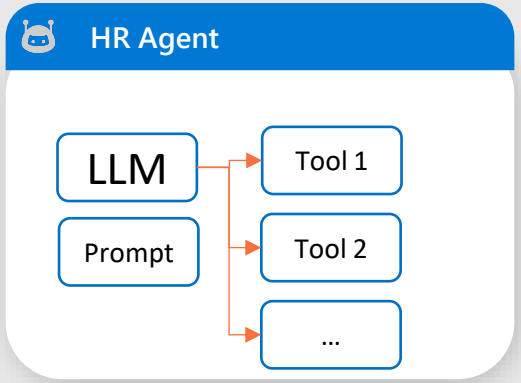
System message

Description

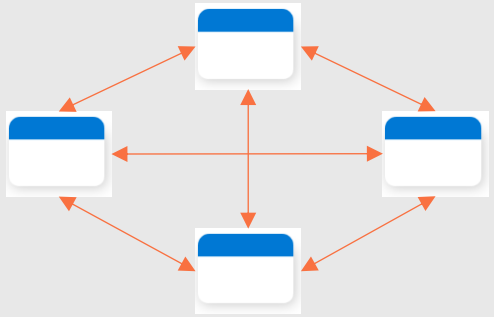
Tools

Agents Orchestration and Communication Styles

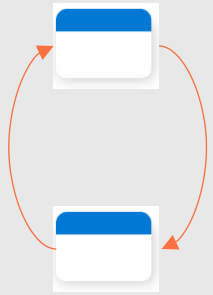
Single Agent



Network

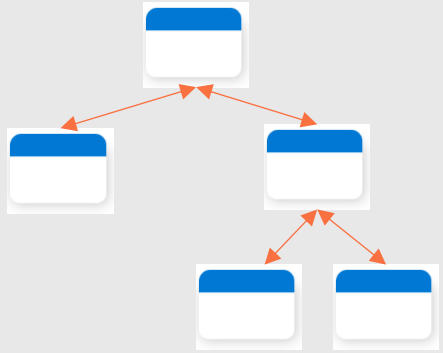


Reflection

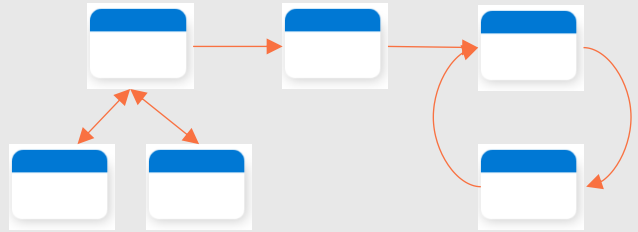


Sequential

Hierarchical



Hybrid



08

Questions

04

Design Principles and Patterns

Key Principles to Keep in Mind

Allow them to plan and execute the best strategy

Autonomy

Breakdown and simplify complexity

Abstraction

Breakdown into smaller, reusable components.

Modularity

Key Principles to Keep in Mind

Allow them to plan and execute the best strategy

Autonomy

Breakdown and simplify complexity

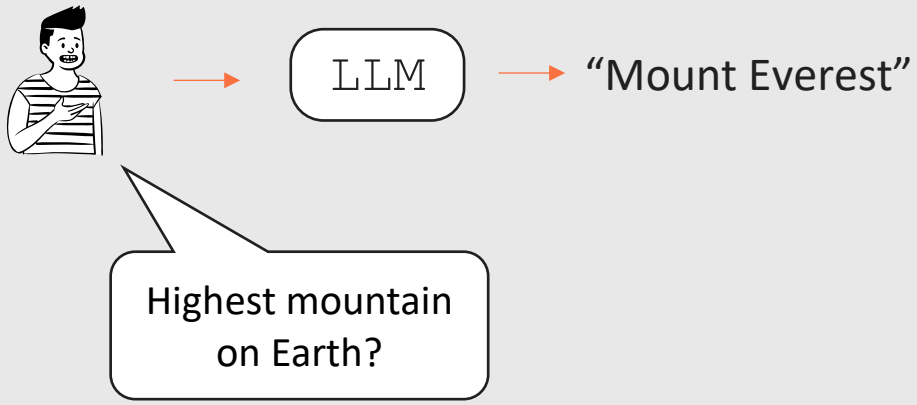
Abstraction

Breakdown into smaller, reusable components.

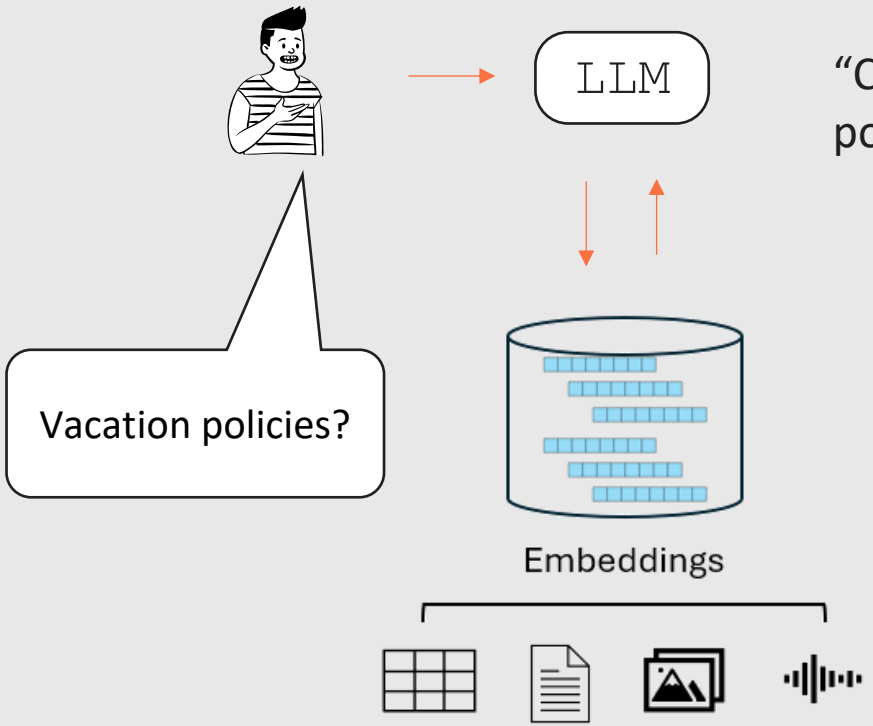
Modularity

Levels of Autonomy

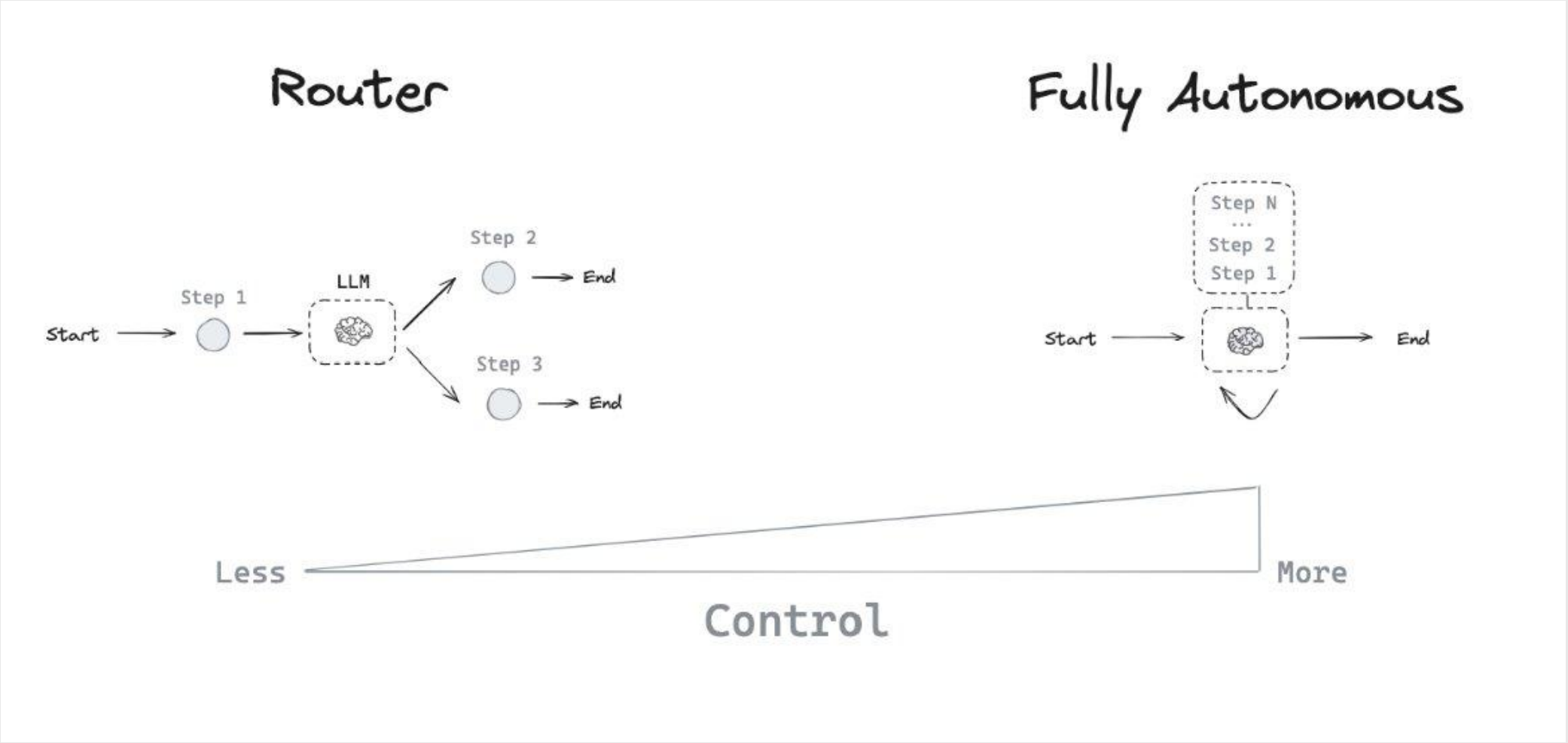
LLM API Call



RAG

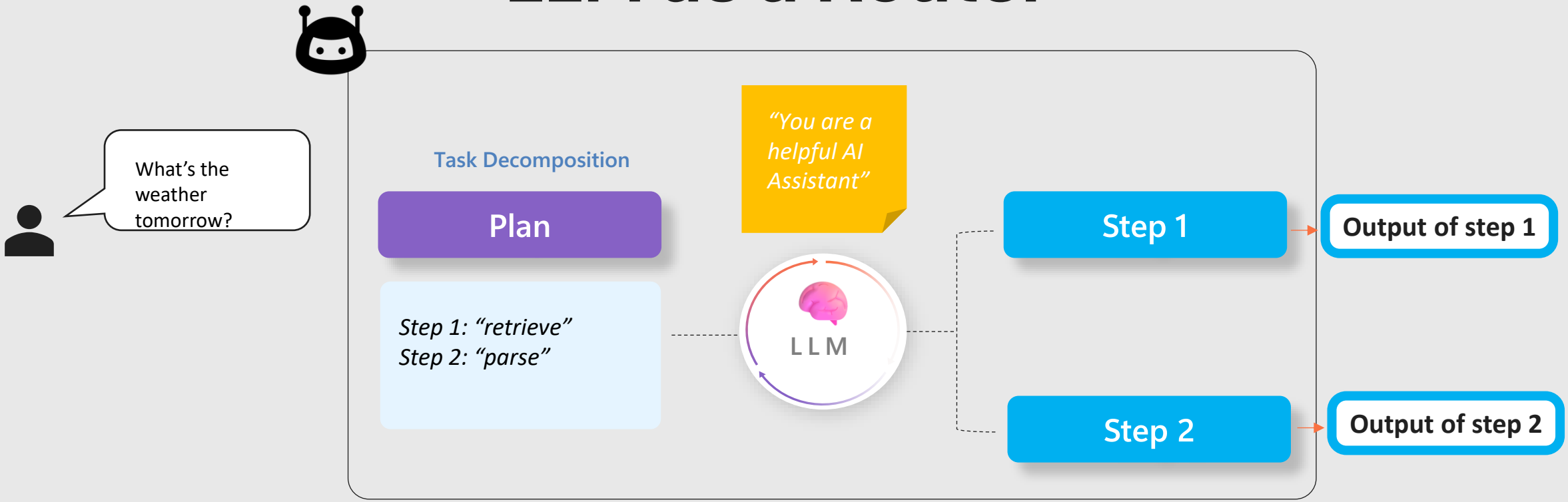


Autonomy = Degree of Freedom



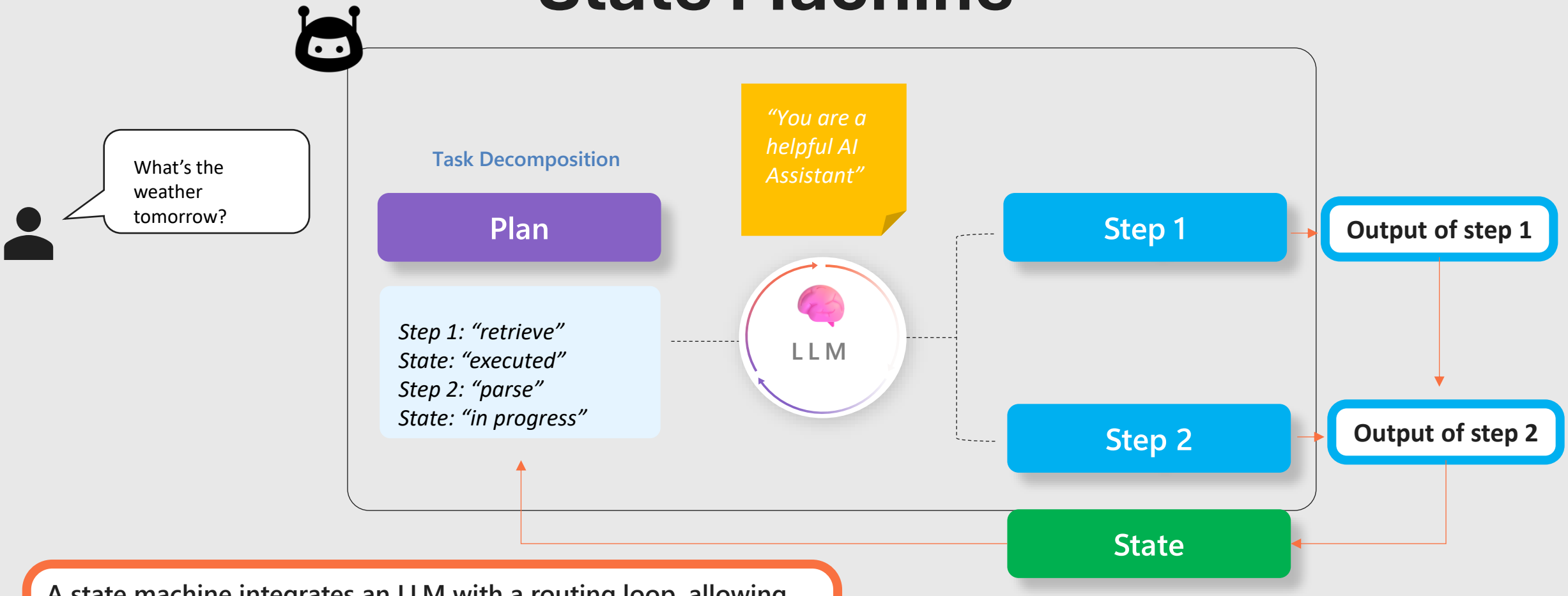
Source: https://langchain-ai.github.io/langgraph/concepts/agentic_concepts/

LLM as a Router



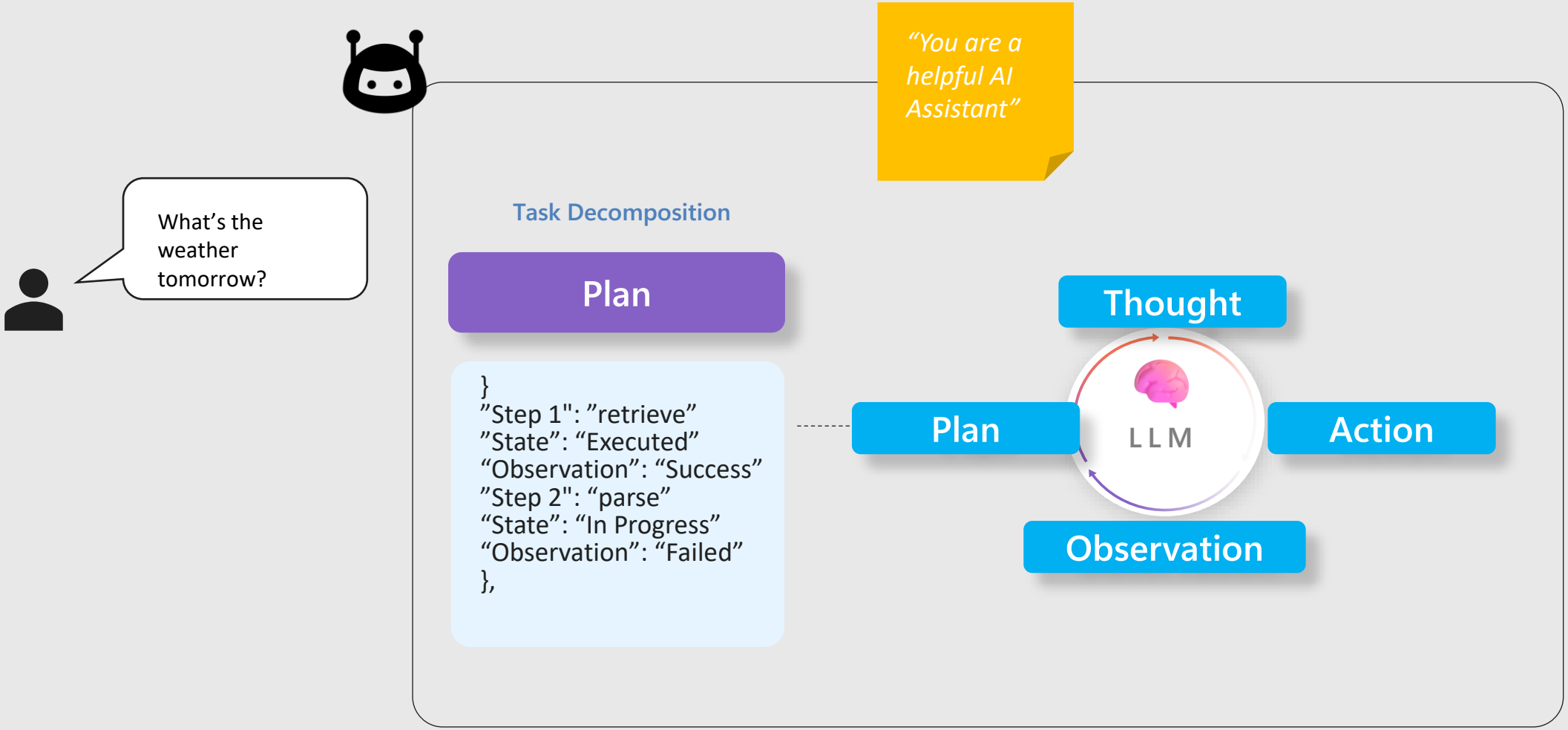
A router allows an LLM to select a single step from a specified set of options. It makes a single decision to produce a specific output. No multi-turn.

State Machine

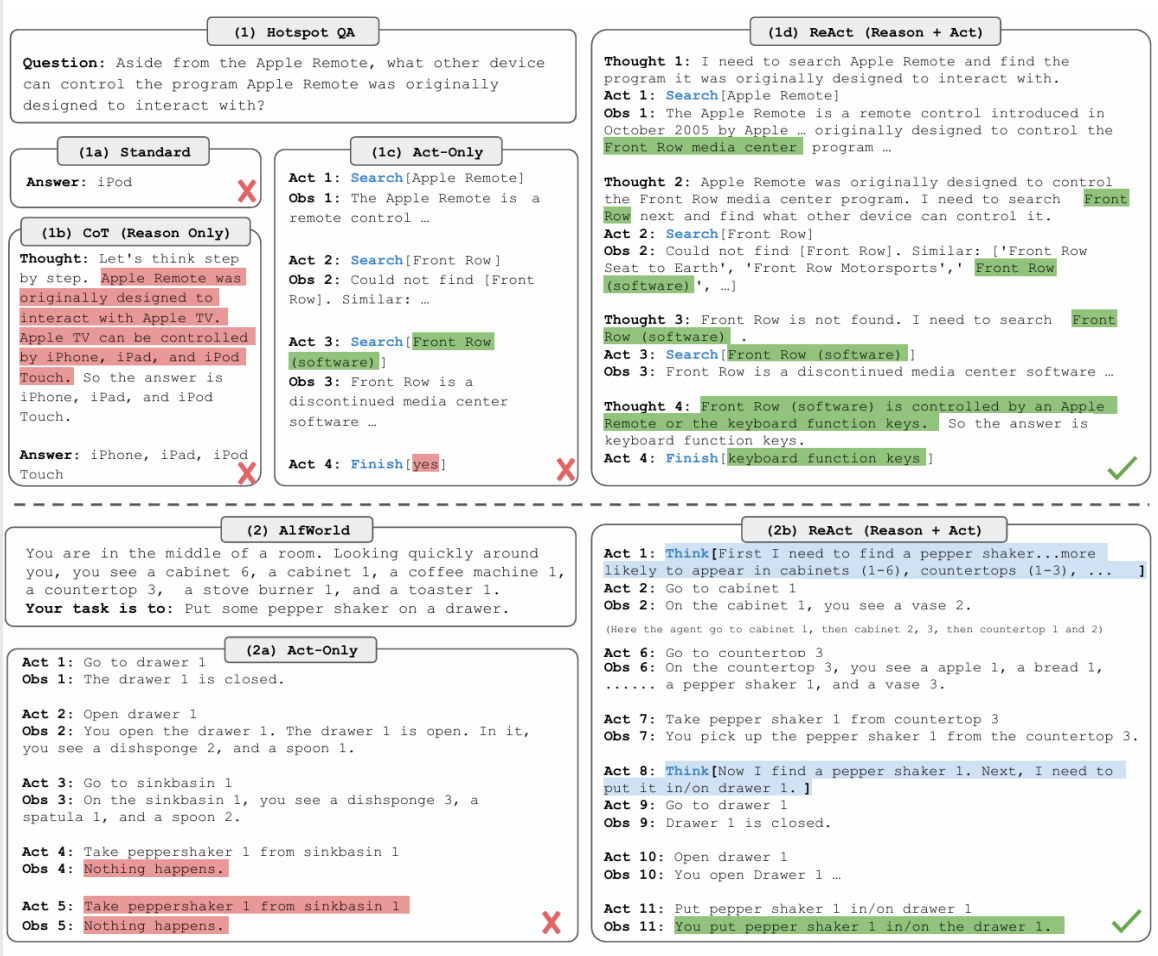


A state machine integrates an LLM with a routing loop, allowing for multi-step routing. The concept of State is part of the LangGraph ecosystem.

ReAct – Reason and Act



ReAct – Paper

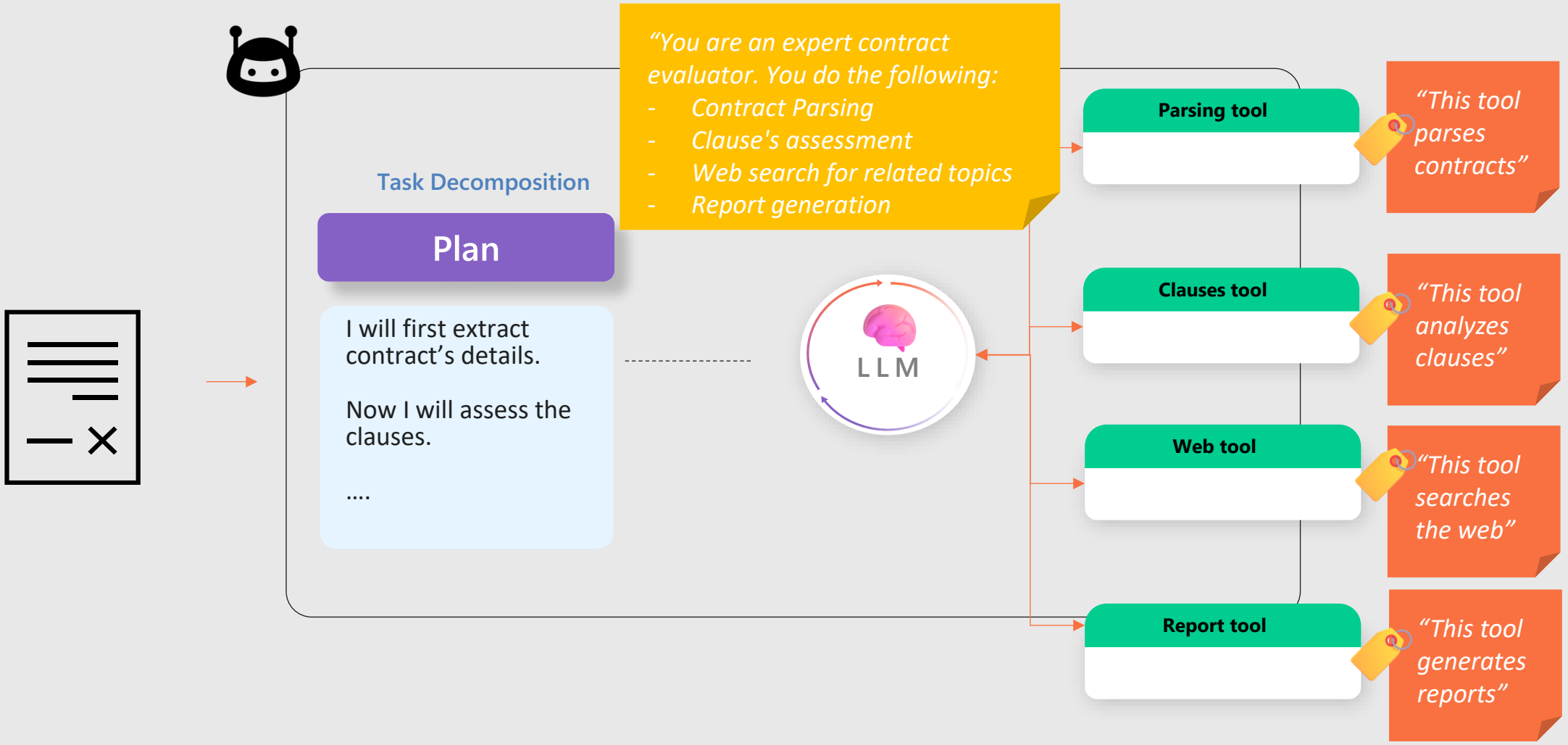


"While large language models (LLMs) have demonstrated impressive capabilities across tasks in language understanding and interactive decision making, their abilities for reasoning (e.g., chain-of-thought prompting) and acting (e.g., action plan generation) have primarily been studied as separate topics.

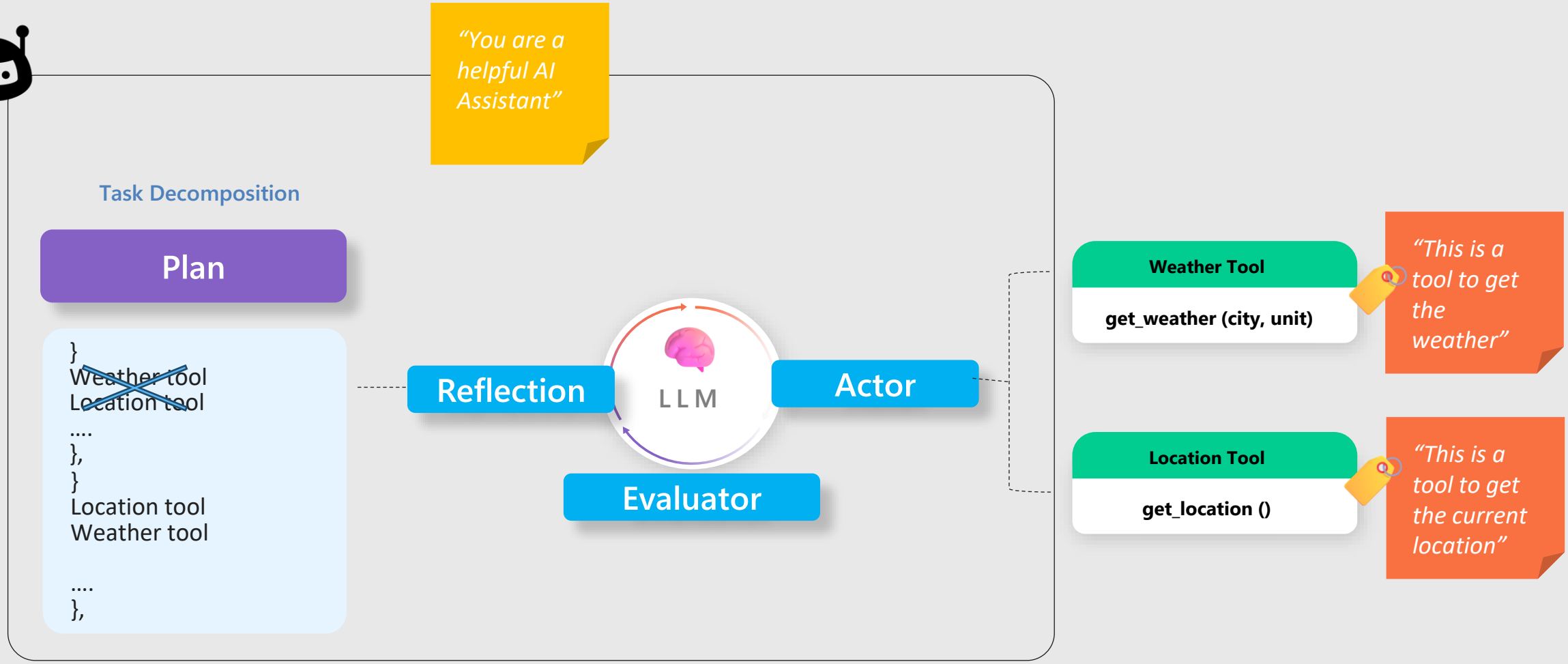
[…] reasoning traces help the model **induce, track, and update action plans** as well as handle exceptions, while actions allow it to **interface with external sources**, such as knowledge bases or environments, to gather additional information. "

Source: https://arxiv.org/abs/2210.03629

Structured Planning

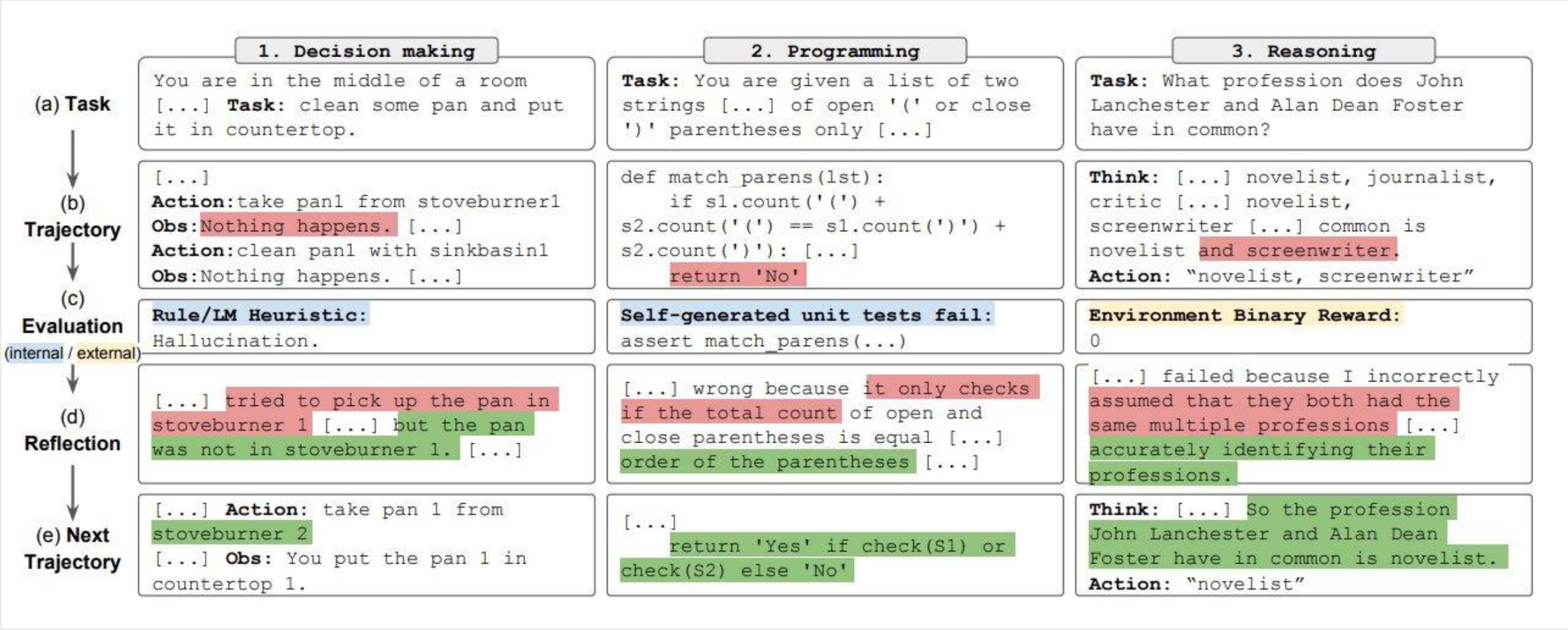


Self-Reflection

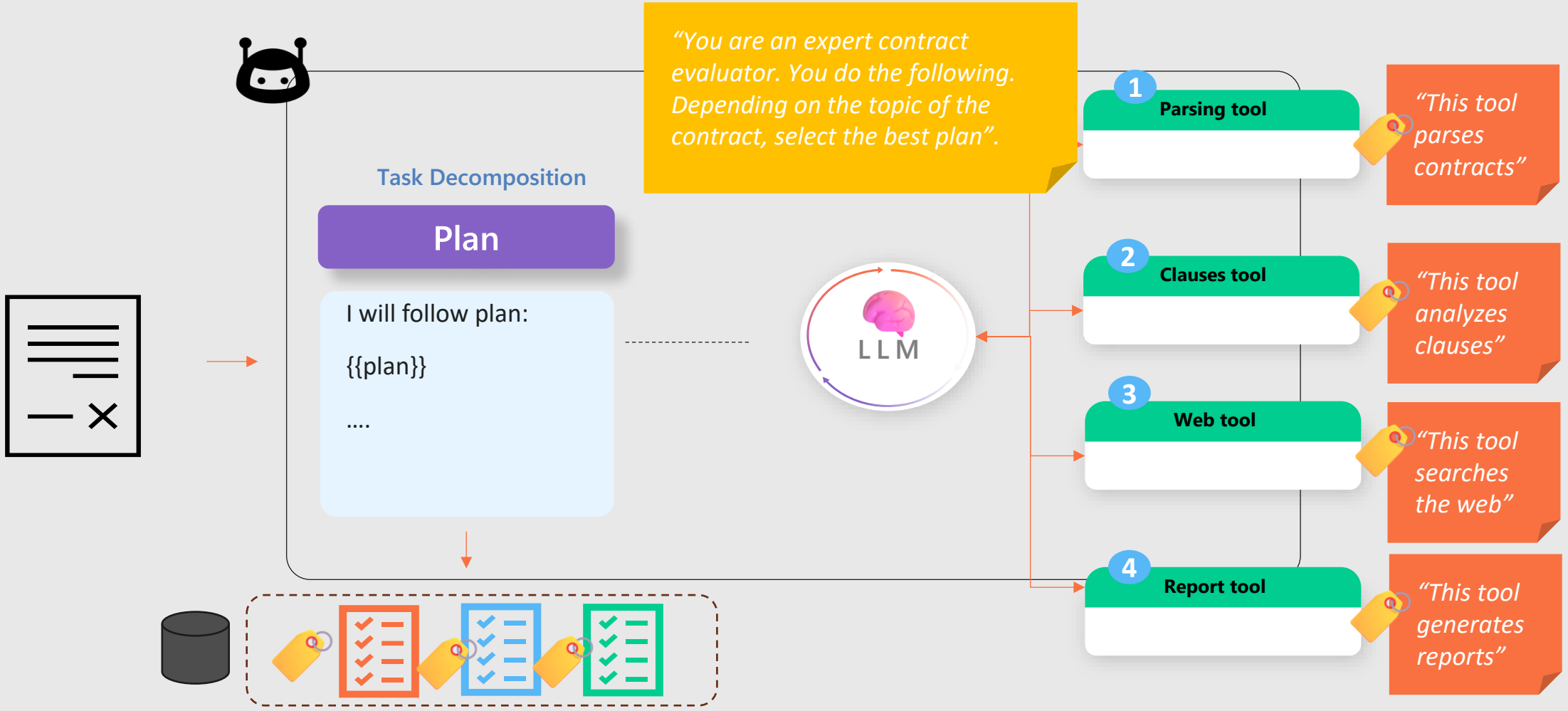


Source: https://arxiv.org/pdf/2303.11366

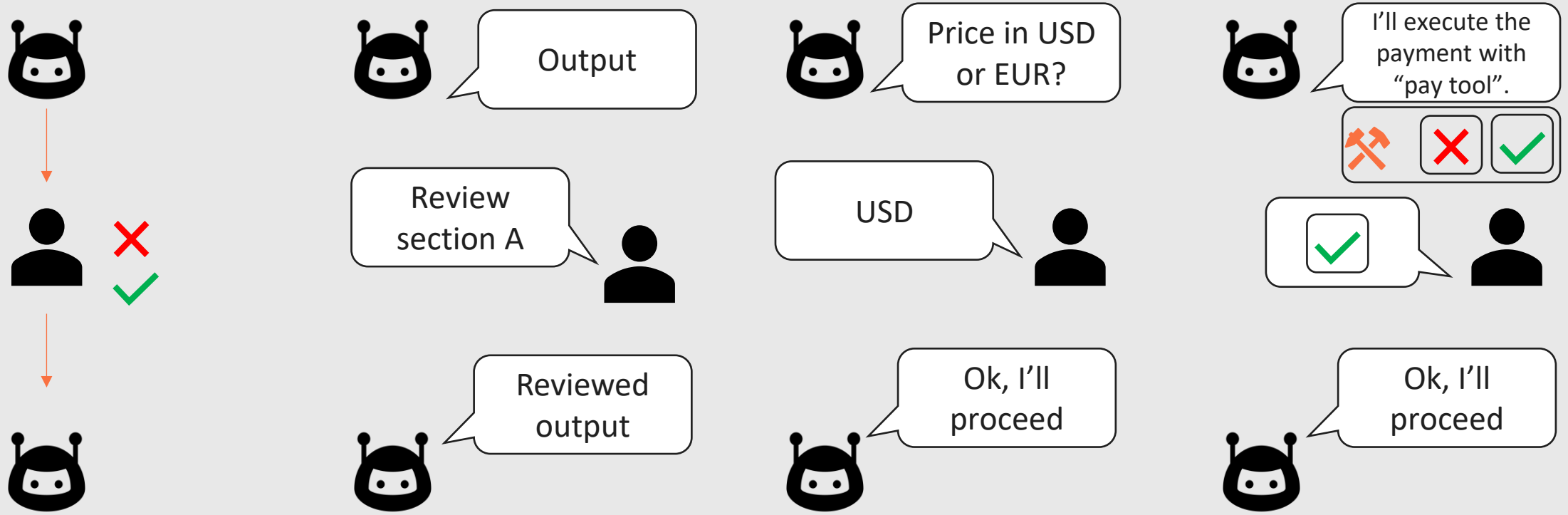
Self-Reflection – Paper



Plan Selection



Human in the Loop



Key Principles to Keep in Mind

Allow them to plan and execute the best strategy

Autonomy

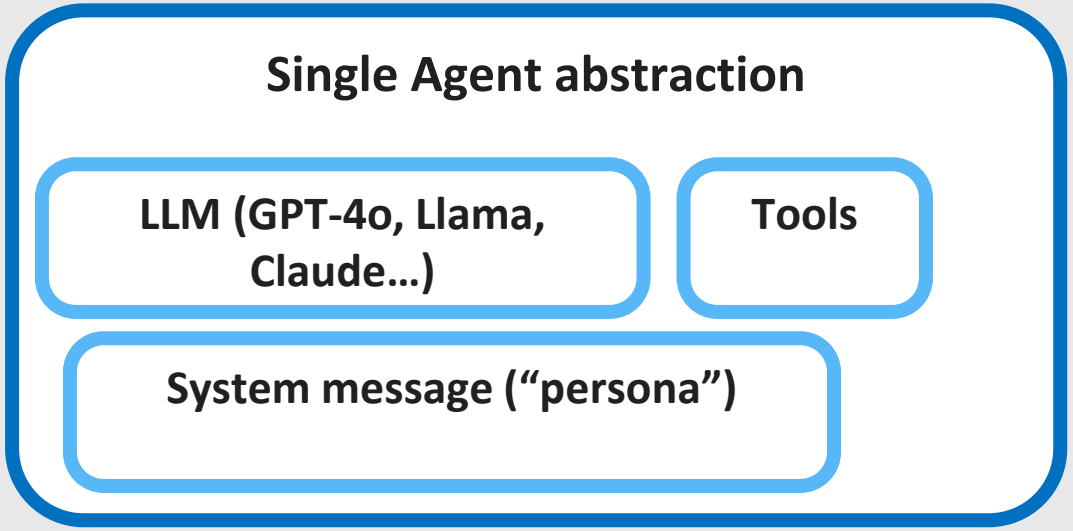
Breakdown and simplify complexity

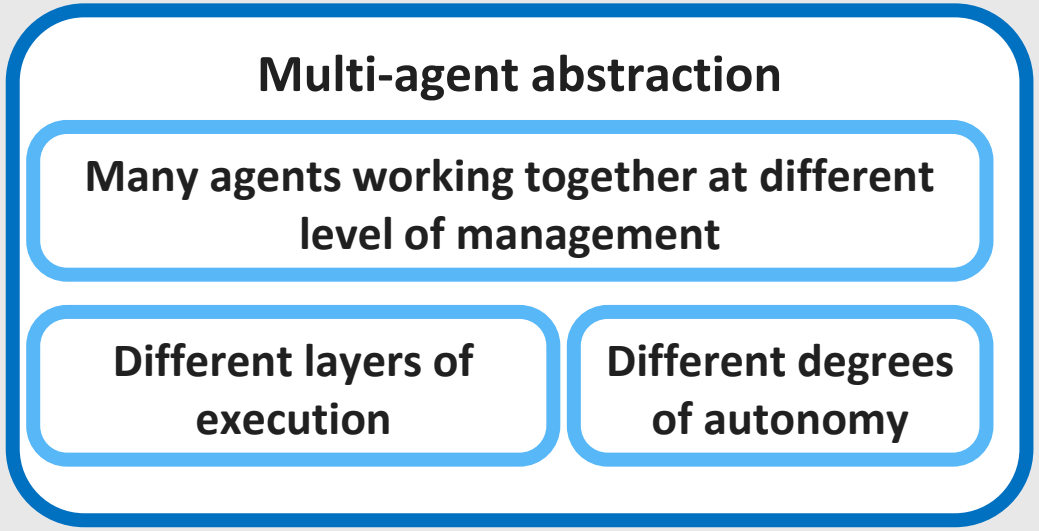
Abstraction

Breakdown into smaller, reusable components.

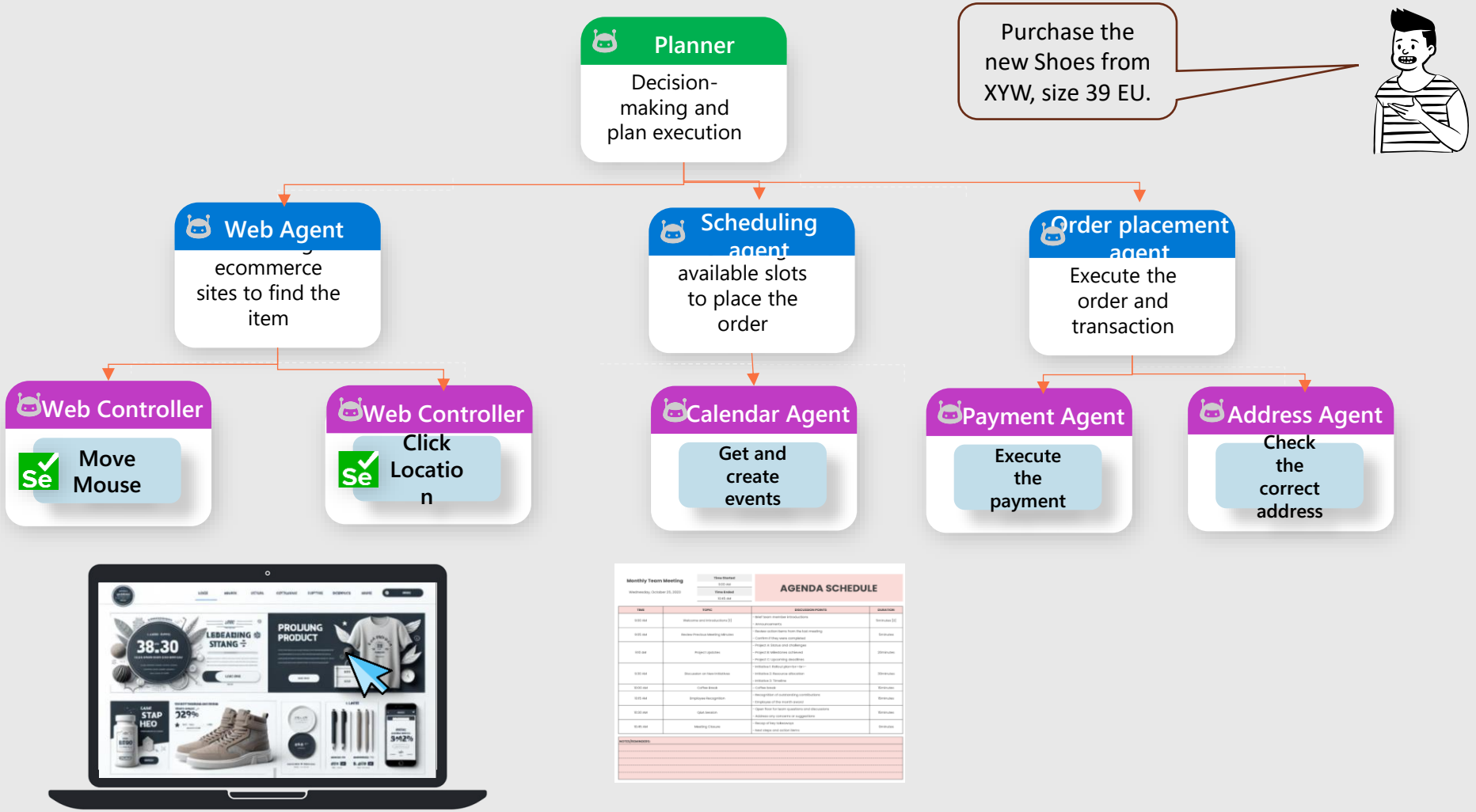
Modularity

Agent Abstraction – First-class Citizen





Modularity and Abstraction




Operator
Find me a family friendly campsite at Joshu

Key Principles to Keep in Mind

Allow them to plan and execute the best strategy

Autonomy

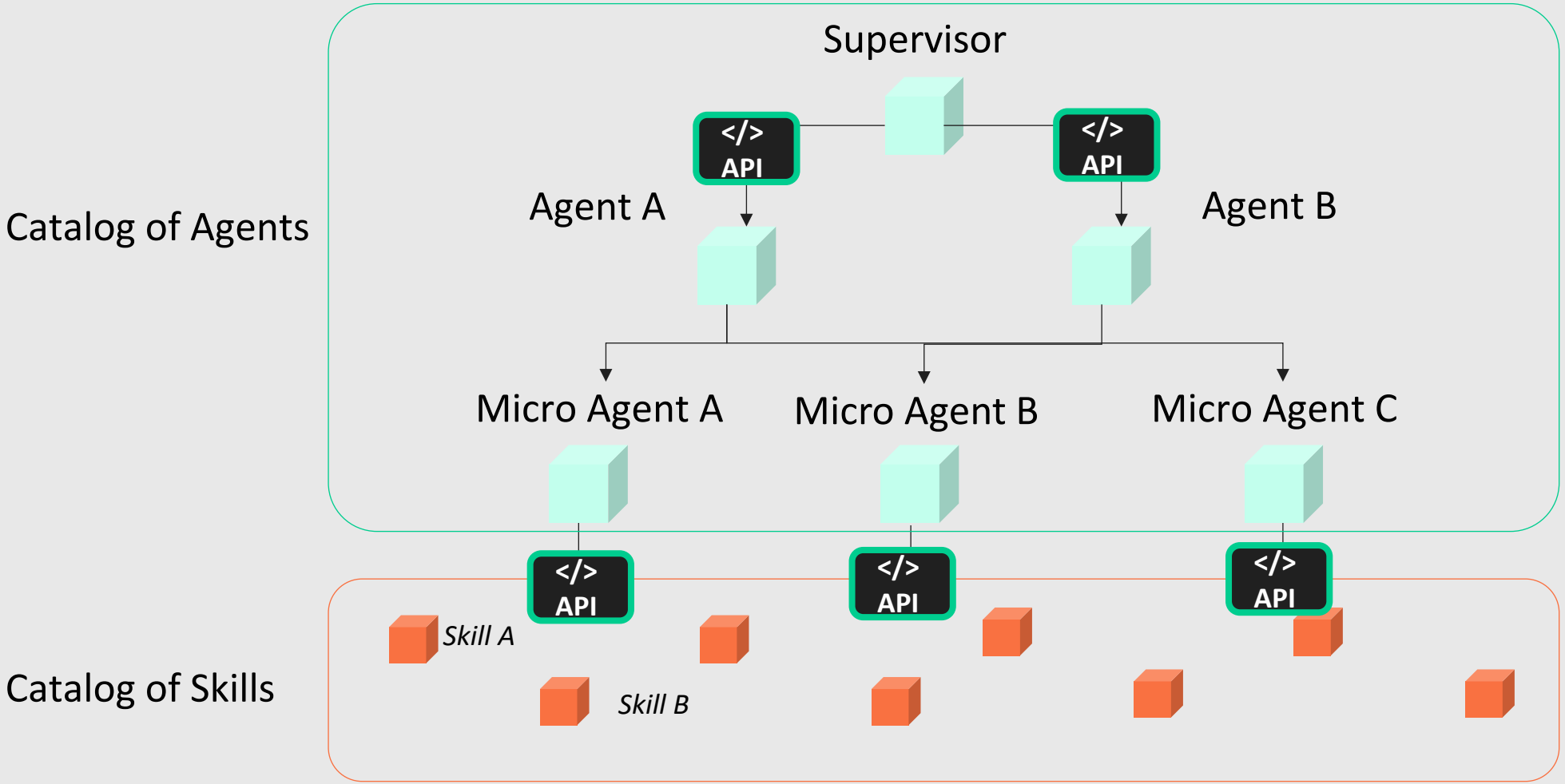
Breakdown and simplify complexity

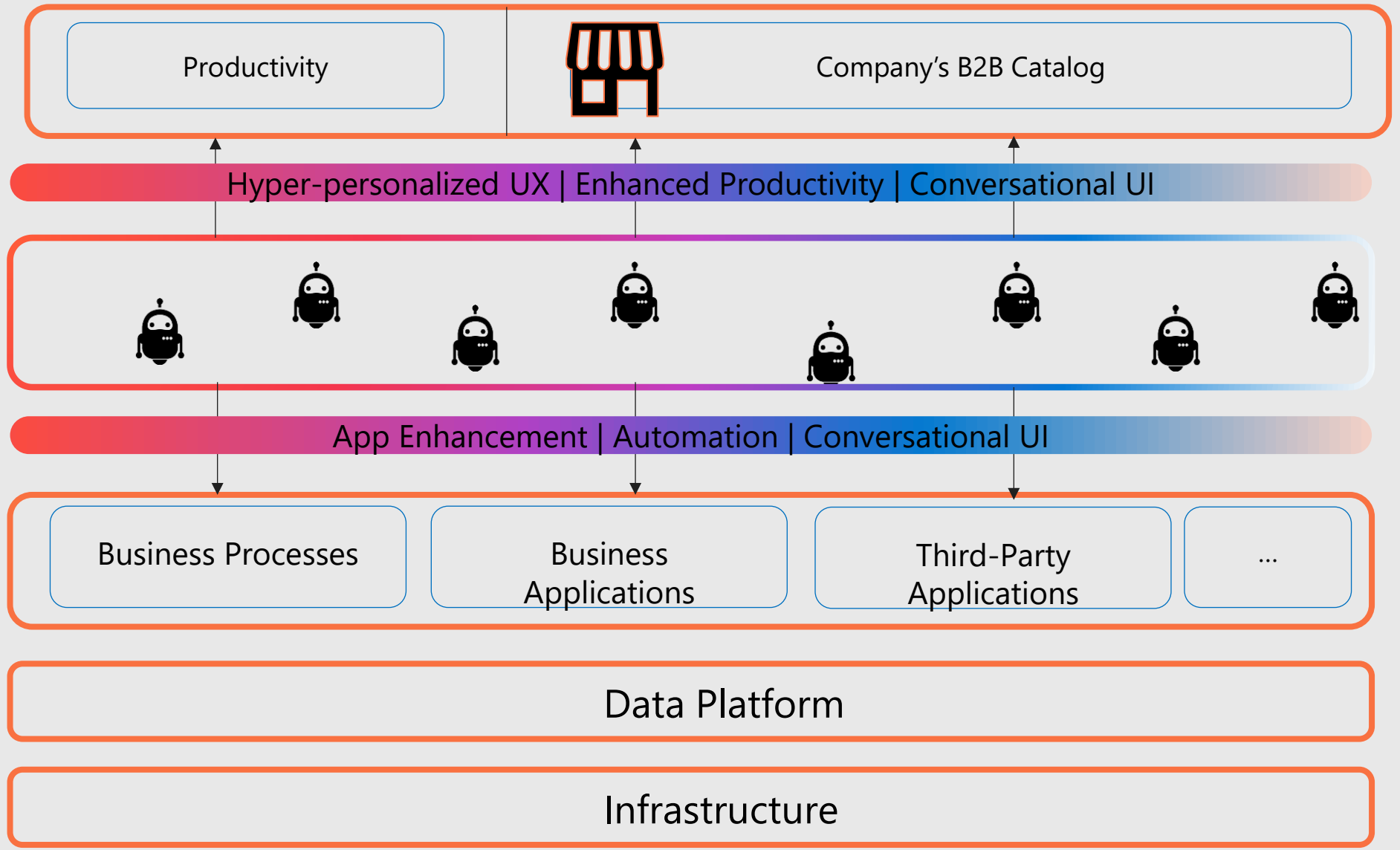
Abstraction

Breakdown into smaller, reusable components.

Modularity

Microservices Go Hand in Hand with Modularity and Abstraction




Engagement Platform (internal and external)
Agent Orchestration
Agentic State
Agent Orchestration
Enterprise Applications
Productivity
Company's B2B Catalog
Hyper-personalized UX | Enhanced Productivity | Conversational UI
App Enhancement | Automation | Conversational UI
Business Processes
Business Applications
Third-Party Applications
...
Data Platform
Infrastructure

Questions

07

Concept Demo: Build your first "mini agent"

ReAct Agent

Financial Advisor

Read and optimize user's portfolio based on market trends.

Read_portfolio_tool

Custom Tool

Web_search_tool

Pre-built Tool

AI Agents Open Protocols

AI Agents Open Protocols

Model Context Protocol

Agent to Agent

NLWeb

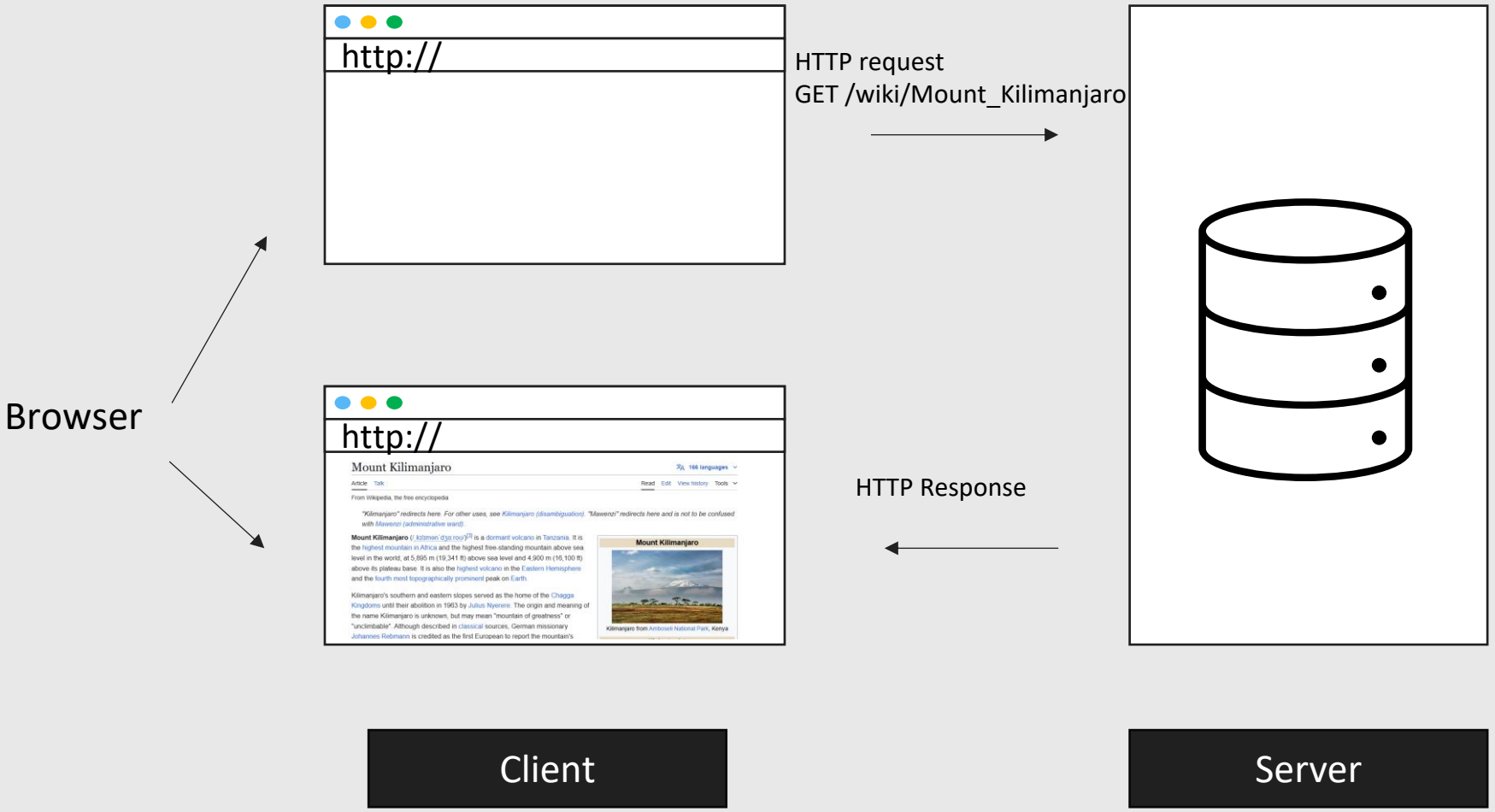
AI Agents Open Protocols

Model Context Protocol

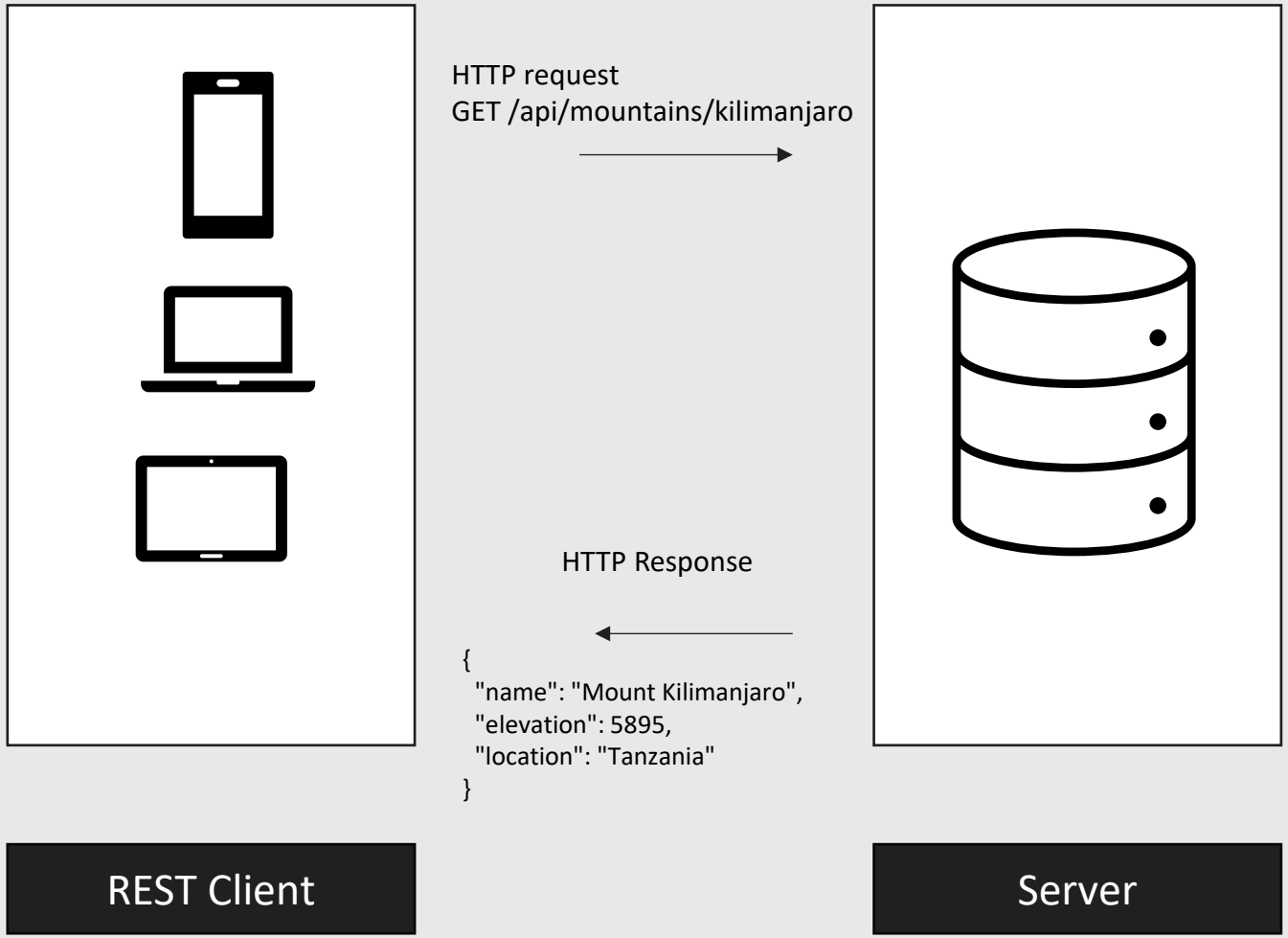
Agent to Agent

NLWeb

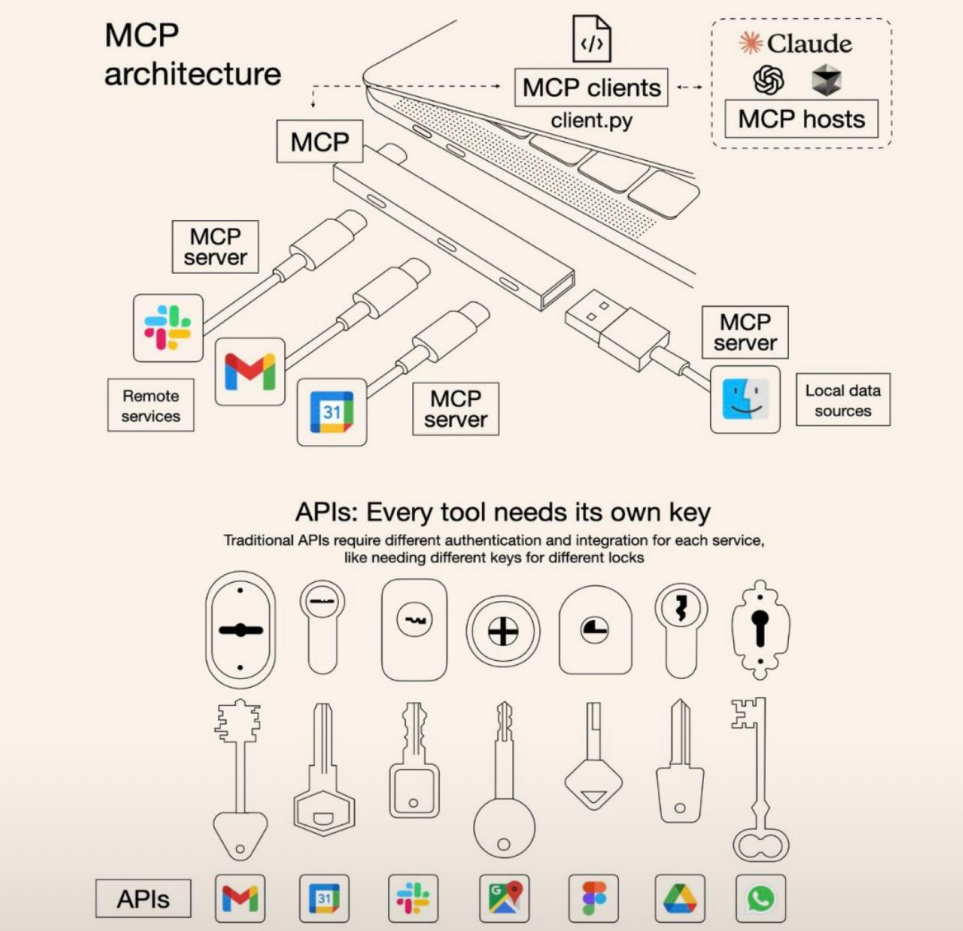
What is a protocol?



What is an API?



Model Context Protocol

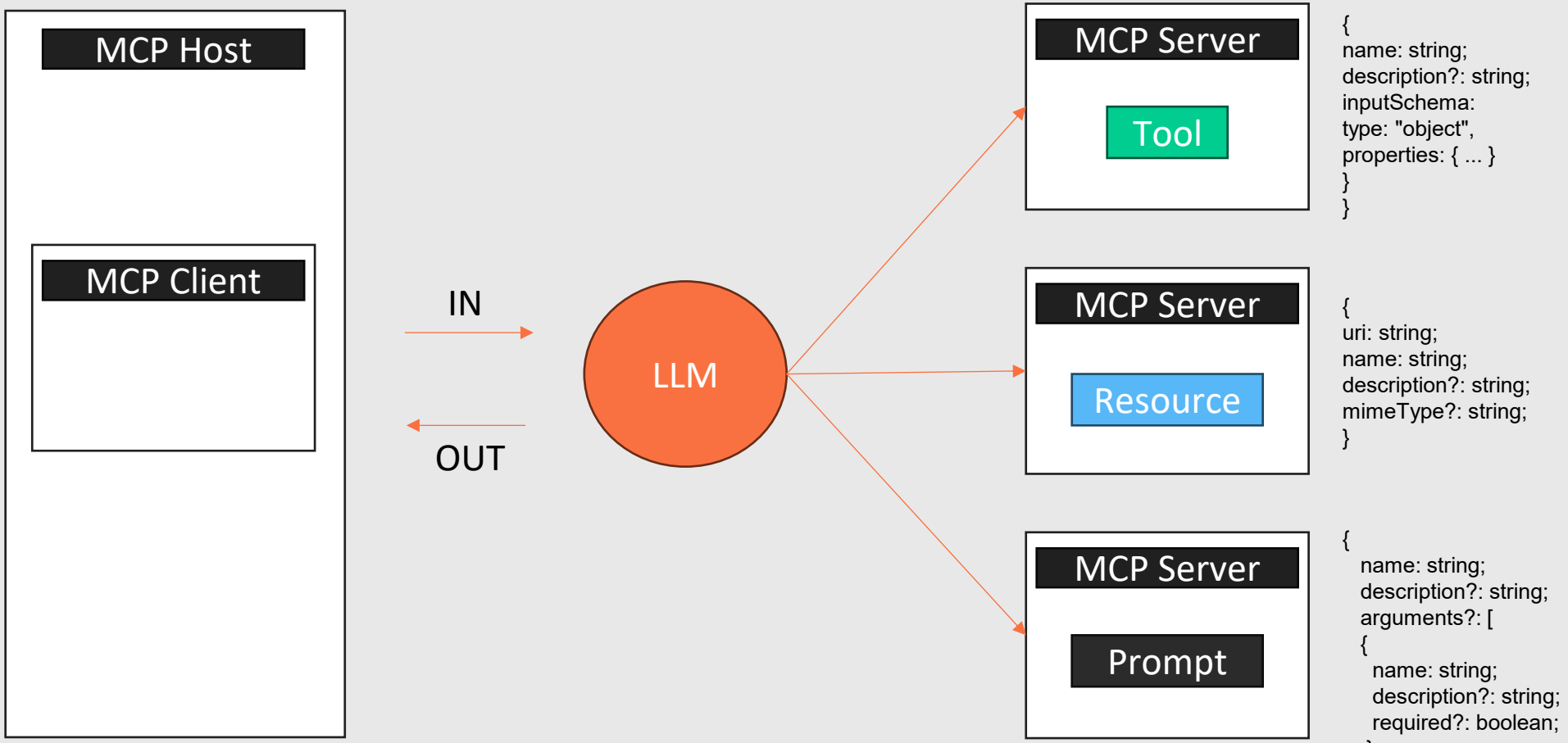


Source: https://blog.csdn.net/ChailangCompany/article/details/146354913

"MCP is an open protocol that standardizes how applications provide context to LLMs. Think of MCP like a USB-C port for AI applications. Just as USB-C provides a standardized way to connect your devices to various peripherals and accessories, MCP provides a standardized way to connect AI models to different data sources and tools."

Source: https://modelcontextprotocol.io/introduction

High-Level Architecture of MCP



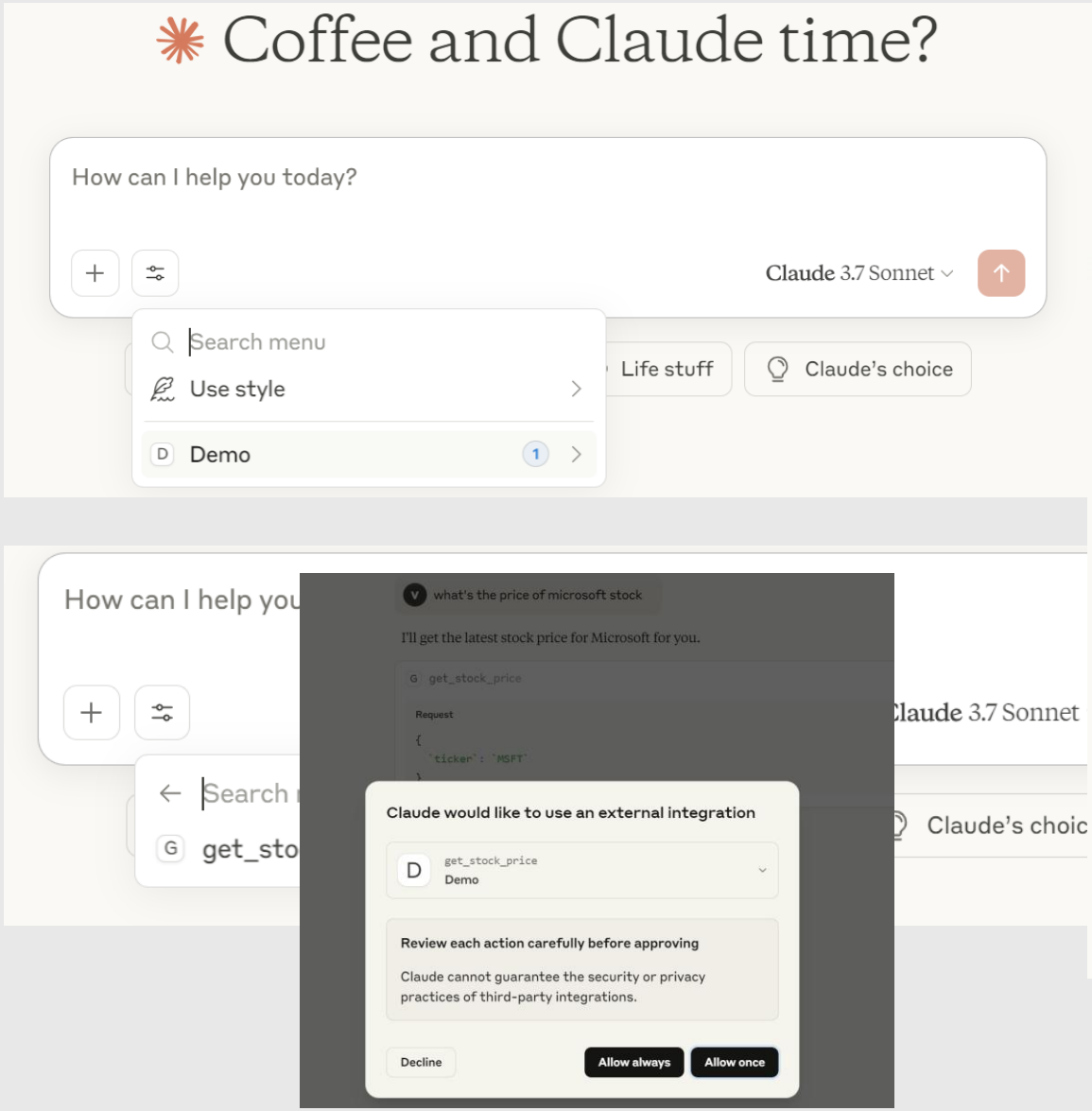
Example – Weather Forecast MCP Server

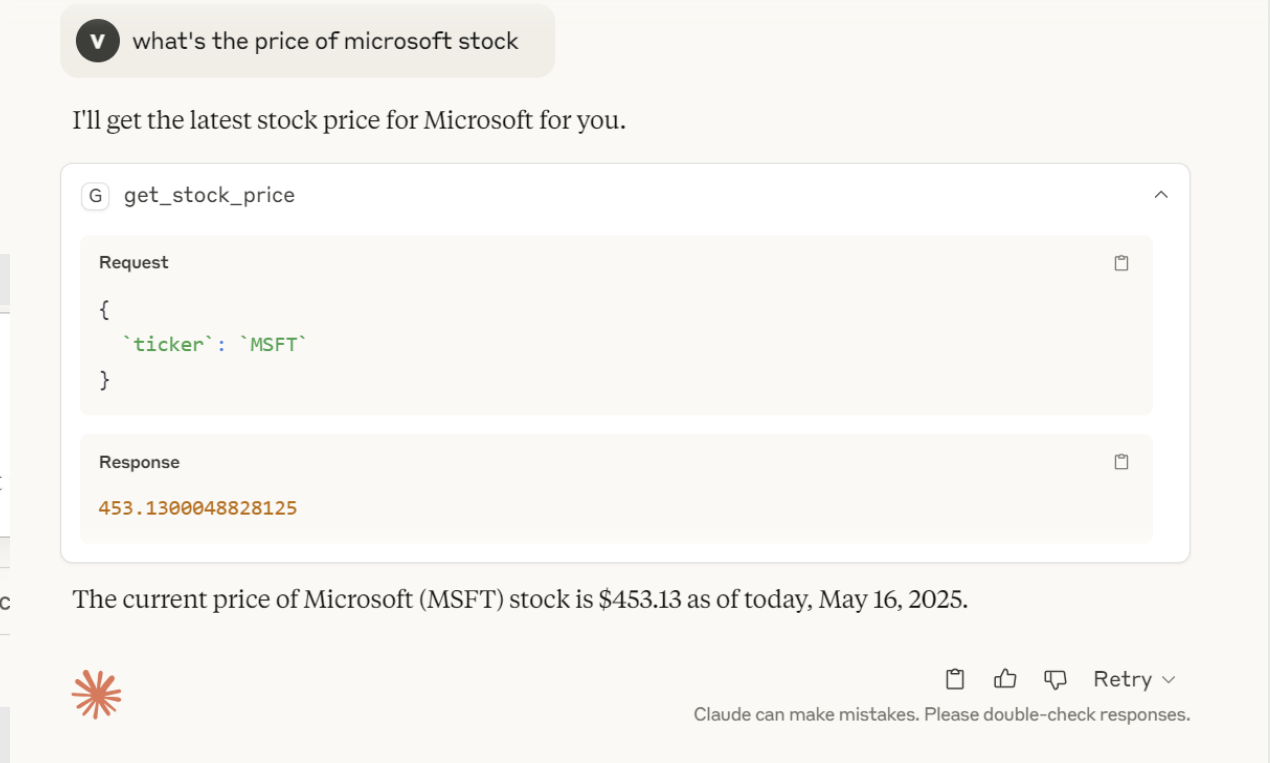
```python
# server.py
from mcp.server.fastmcp import FastMCP
import yfinance as yf

# Create an MCP server
mcp = FastMCP("Demo")


# Add an addition tool
@mcp.tool()


def get_stock_price(ticker: str) -> float:
    """Fetch the latest stock price for a given ticker symbol from Yahoo Finance"""
    stock = yf.Ticker(ticker)
    return stock.history(period="1d")["Close"].iloc[-1]

if __name__ == "__main__":
    # Initialize and run the server
    mcp.run(transport='stdio')
  return go(f, seed, [])
}
```

Claude Desktop as MCP Host

Local Machine as MCP Server, using Python SDK

Tip: Find a catalog of MCP Servers here!
Source: https://mcpservers.org/


Coffee and Claude time?
How can I help you today?
Claude 3.7 Sonnet
Search menu
Use style
Demo
1
Life stuff
Claude's choice
How can I help you
Claude 3.7 Sonnet
Search
get_sto
Claude's choic
what's the price of microsoft stock
I'll get the latest stock price for Microsoft for you.
get_stock_price
Request
{
`ticker`: `MSFT`
}
Claude would like to use an external integration
get_stock_price
Demo
Review each action carefully before approving
Claude cannot guarantee the security or privacy practices of third-party integrations.
Decline
Allow always
Allow once


what's the price of microsoft stock
I'll get the latest stock price for Microsoft for you.
get_stock_price
Request
{
`ticker`: `MSFT`
}
Response
453.1300048828125
The current price of Microsoft (MSFT) stock is $453.13 as of today, May 16, 2025.
Retry
Claude can make mistakes. Please double-check responses.

AI Agents Open Protocols

Model Context Protocol

Agent to Agent

NLWeb

Agent to Agent

Image source: https://a2aprotocol.ai/

"A2A Protocol is an open standard that enables AI agents to communicate and collaborate across different platforms and frameworks, regardless of their underlying technologies. It's designed to maximize the benefits of agentic AI by enabling true multi-agent scenarios."

Source: https://a2aprotocol.ai/

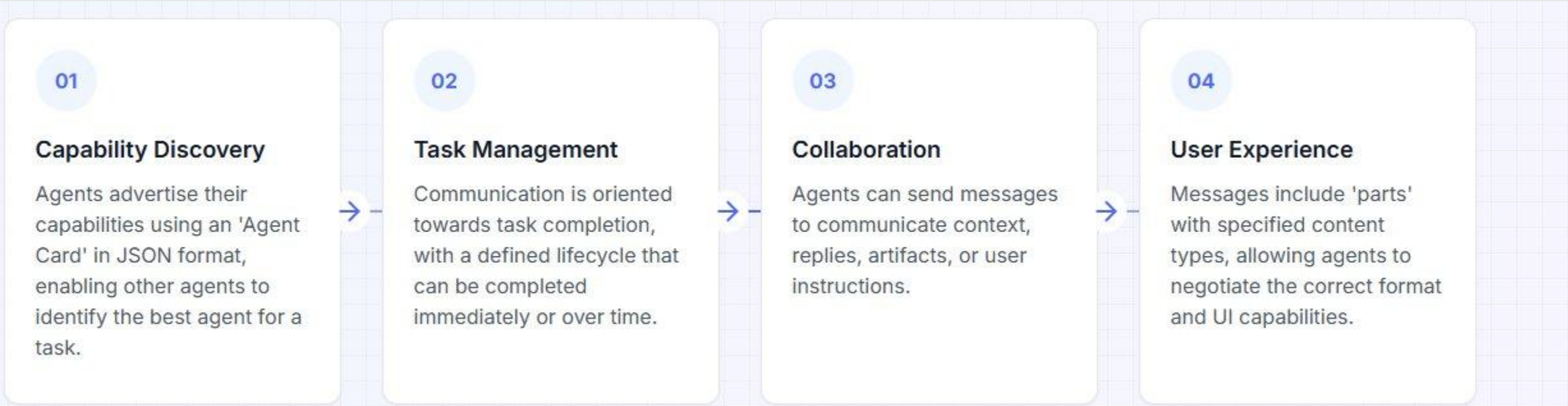
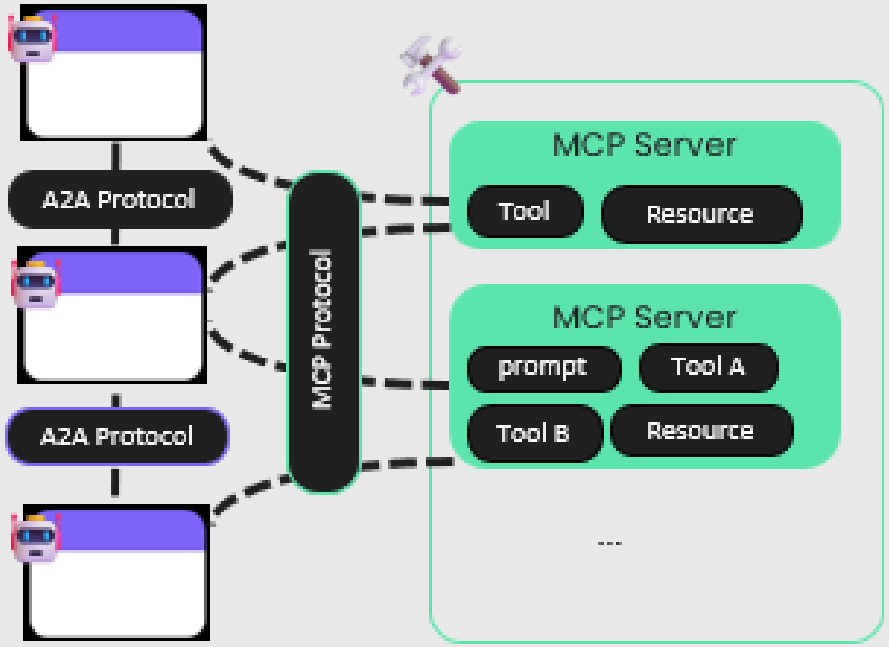


Image source: https://a2aprotocol.ai/

A2A vs MCP



MCP

Protocol to connect LLMs with Data, Resources and Tools.

A2A

Protocol to connect and enable collaboration among multiple agents.

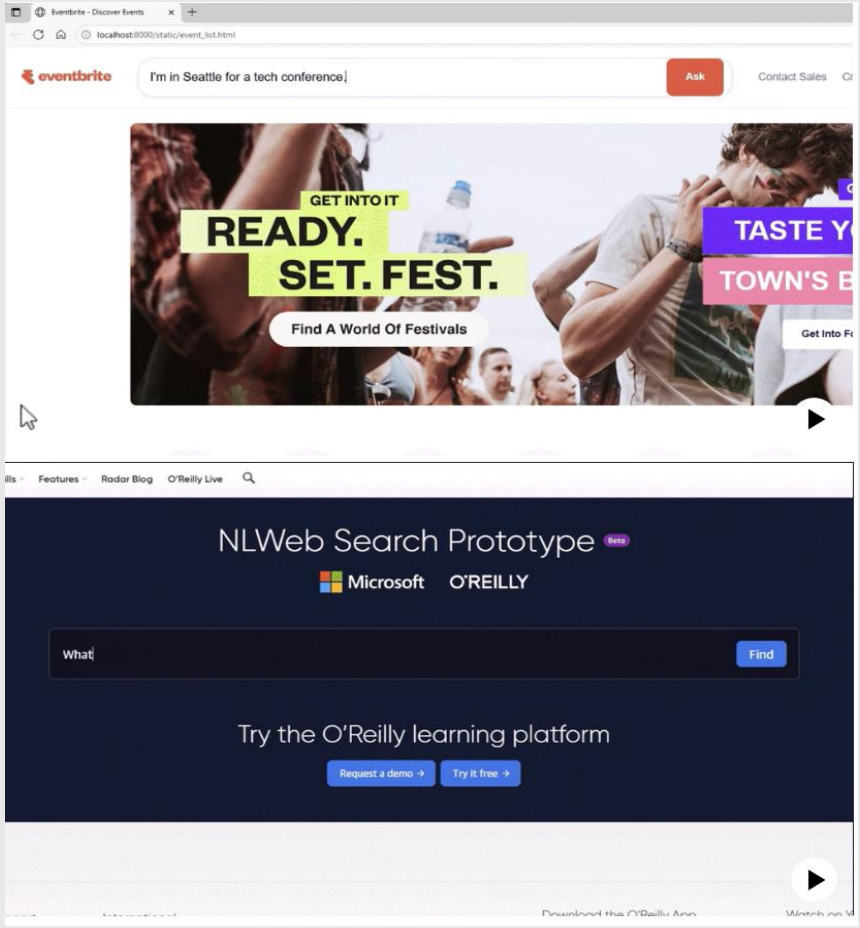
AI Agents Open Protocols

Model Context Protocol

Agent to Agent

NLWeb

What's NLWeb?

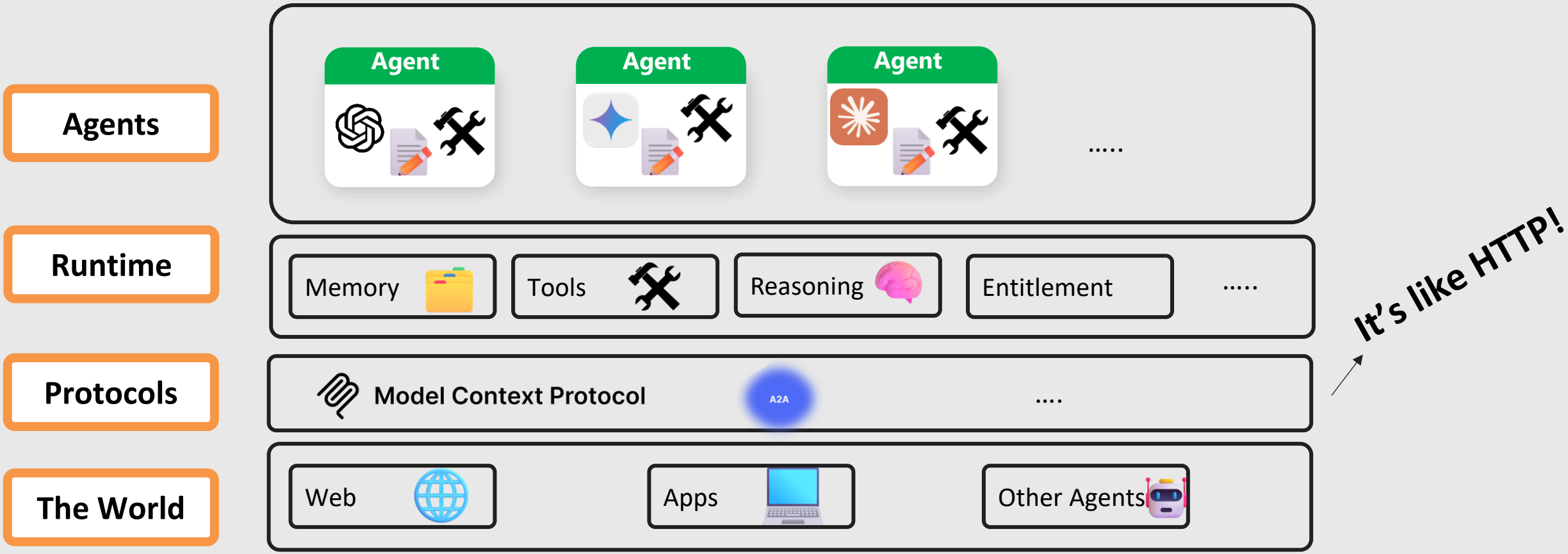


"Building conversational interfaces for websites is hard. NLWeb seeks to make it easy for websites to do this. And since NLWeb natively speaks MCP, the same natural language APIs can be used both by humans and agents."

Source: https://news.microsoft.com/source/features/company-news/introducing-nlweb-bringing-conversational-interfaces-directly-to-the-web/?msockid=16ccd88e67d963500e5fcd7866d062ac

Source: https://github.com/microsoft/nlweb

Building the Open Agentic Web



Source: https://github.com/microsoft/nlweb

Questions

09

Thank You!