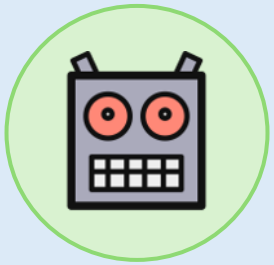
Reinforcement Learning & Autonomous systems

MBD Sept 24

IE UNIVERSITY
SCHOOL OF SCIENCE & TECHNOLOGY

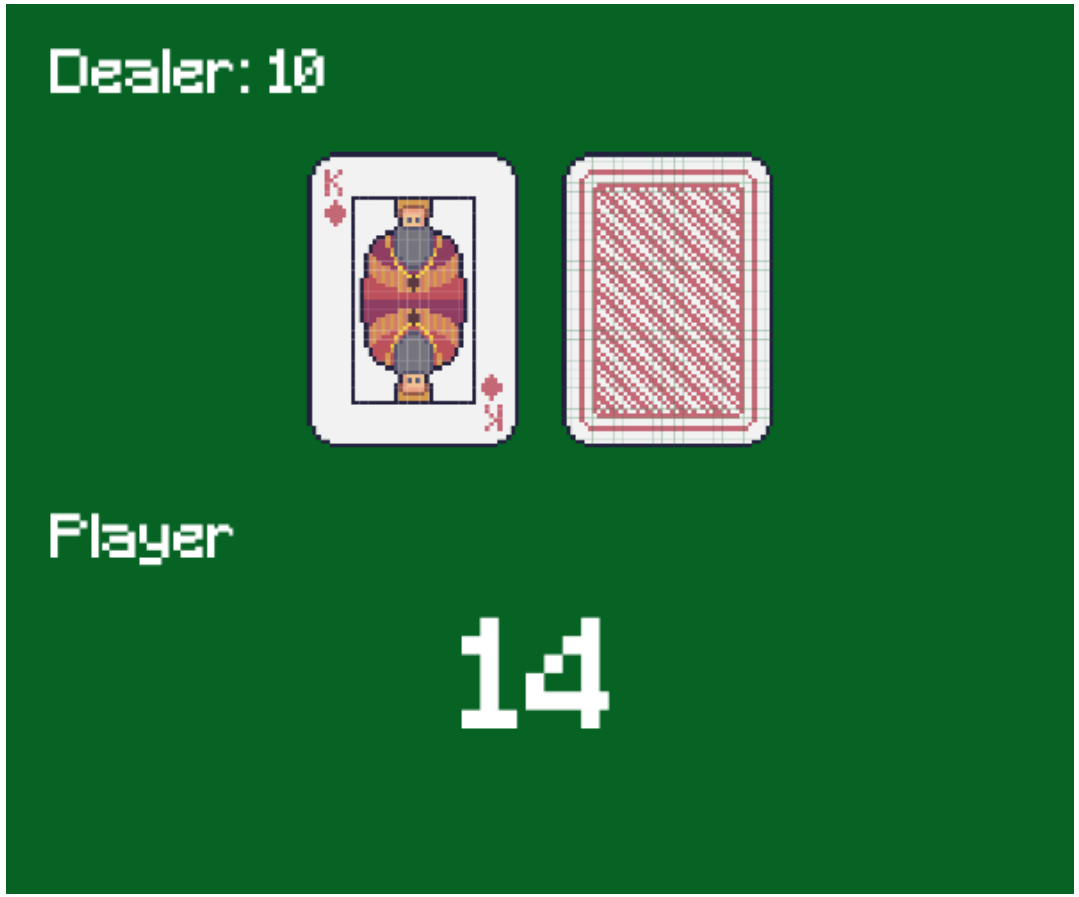


AI Agent

Lecture 12
AI Agents

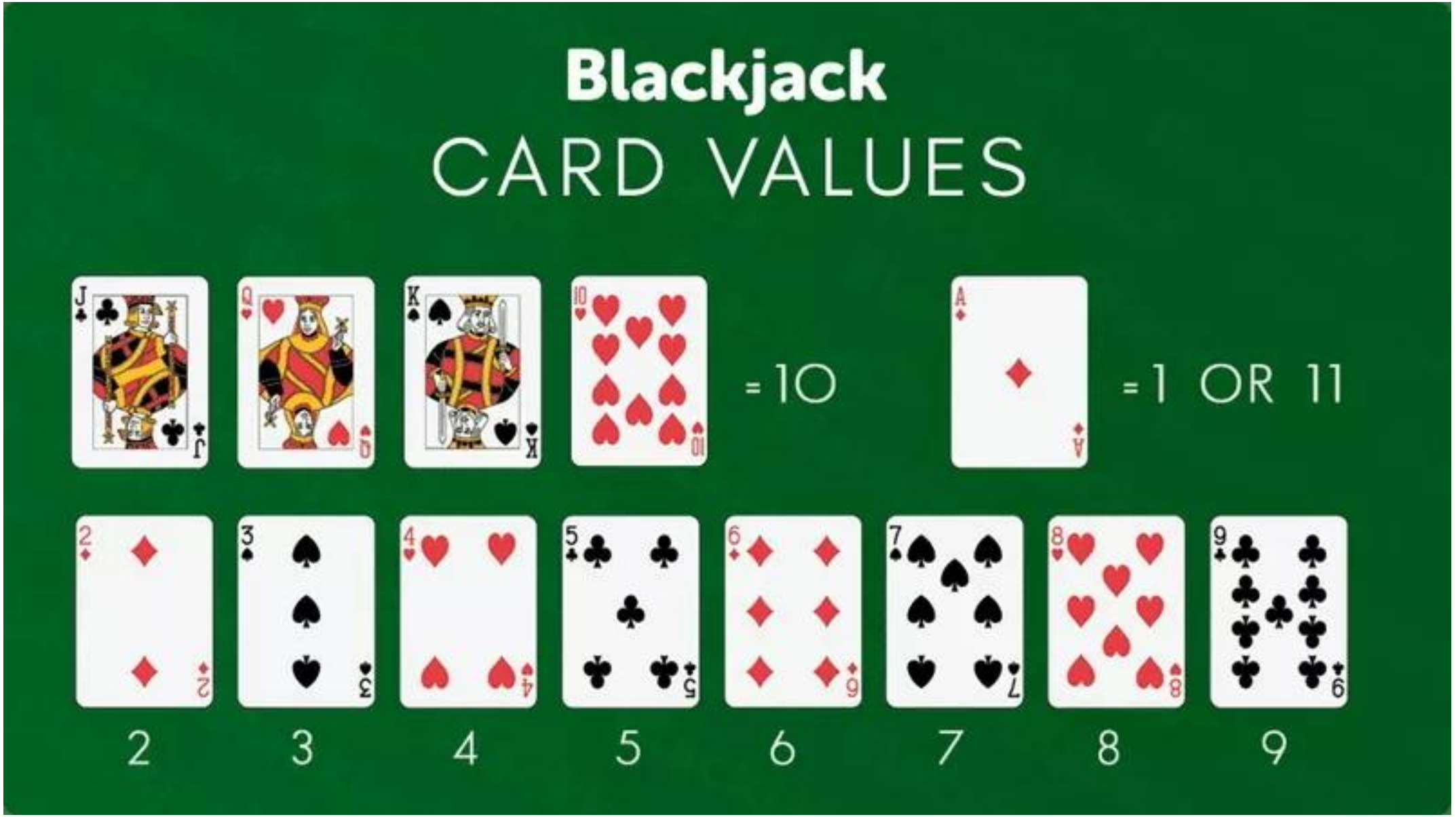
jmanero@faculty.ie.edu

Let’s start with an example
Blackjack



Let’s say we want to develop an Agent to play Blackjack

Blackjack – The cards



Let’s start with an example

Blackjack – Moves

4 cards are dealt

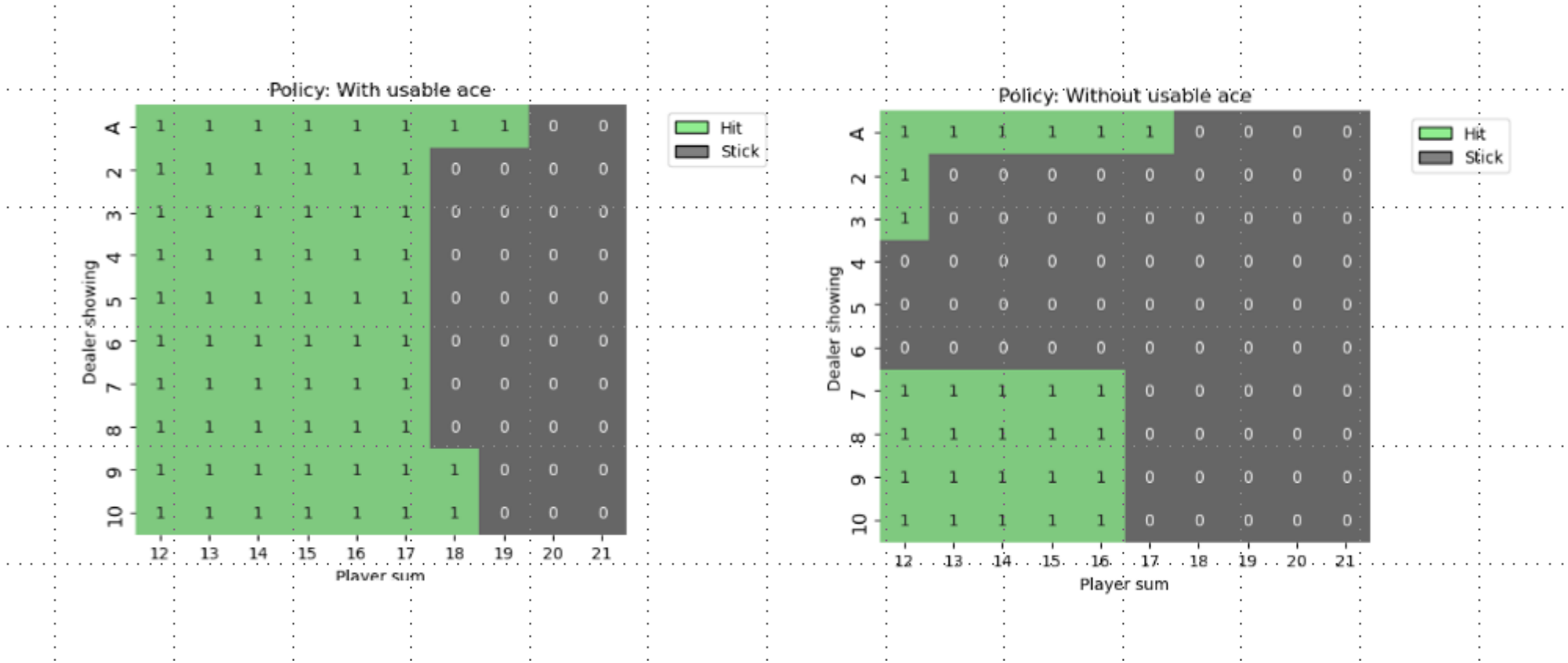
Hit: New Card
Stick: No new card

(you can lose here)

Dealer ends the game

Win or Draw

Let’s start with an example
What is the Blackjack Policy?



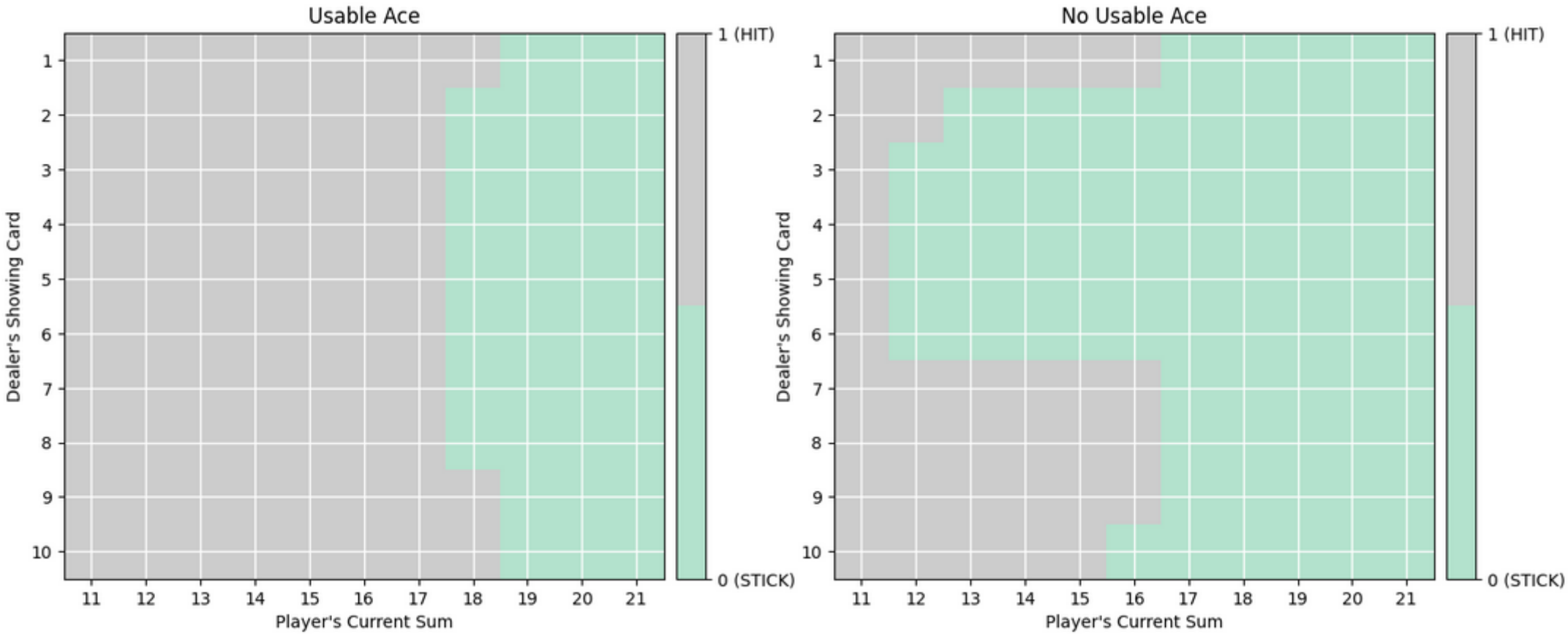
Let’s develop an Agent that learns to play Blackjack

Let's start with an example
A first-visit standard MC

- Episodic update
- Exploration/Exploitation trade-off
- Large number of exploration episodes

```python
# Epsilon-greedy policy
def epsilon_greedy_policy(state):
    if np.random.rand() < epsilon:
        return np.random.randint(env.action_space.n)
    return np.argmax(Q[state])

# Monte Carlo learning
for episode in range(num_episodes):
    episode_memory = []
    state, _ = env.reset()

    while True:
        action = epsilon_greedy_policy(state)
        next_state, reward, terminated, truncated, _ = env.step(action)
        episode_memory.append((state, action, reward))
        state = next_state
        if terminated or truncated:
            break

    G = 0
    visited = set()
    for state, action, reward in reversed(episode_memory):
        G = gamma * G + reward
    if (state, action) not in visited:
            visited.add((state, action))
            returns_sum[(state, action)] += G
            returns_count[(state, action)] += 1.0
            Q[state][action] = returns_sum[(state, action)] / returns_count[(state, action)]

    epsilon = max(min_epsilon, epsilon * epsilon_decay)
#    if episode > 1_000_000:
#        epsilon = max(0.01, epsilon * epsilon_decay)

# Derive final policy
policy = {state: np.argmax(actions) for state, actions in Q.items()}
```

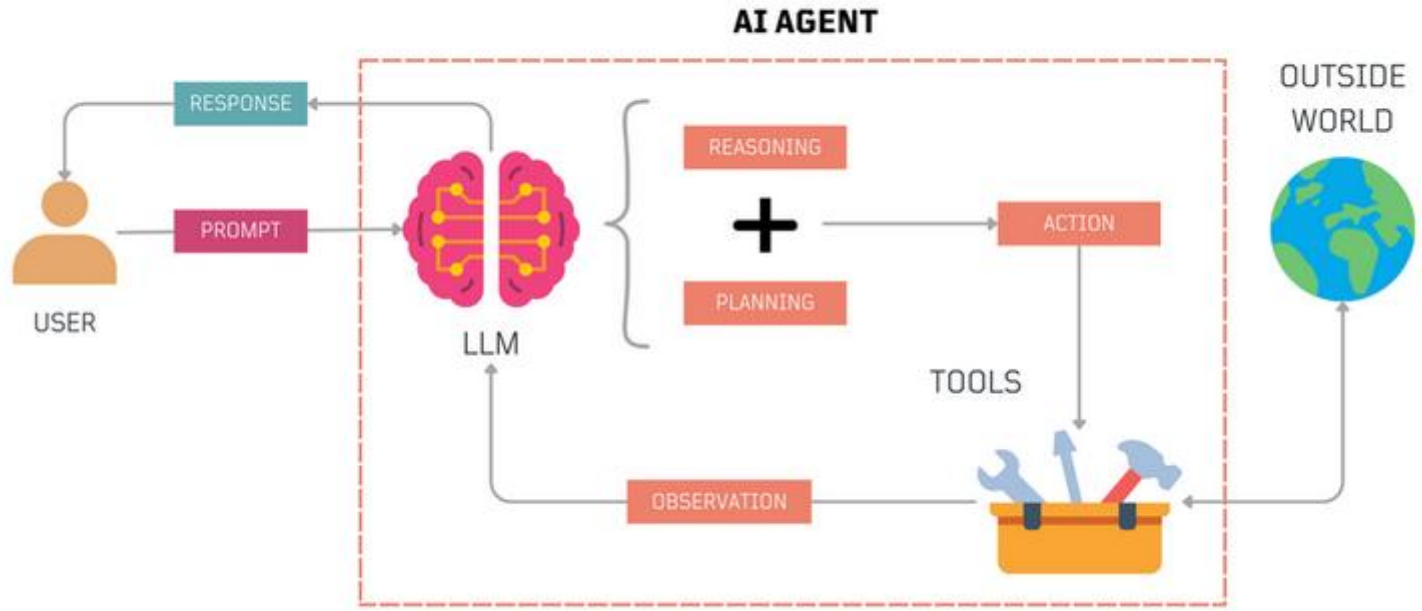
Let’s start with an example
A Blackjack MC method



https://github.com/castorgit/RL_course/blob/main/018_MC_Blackjack.ipynb

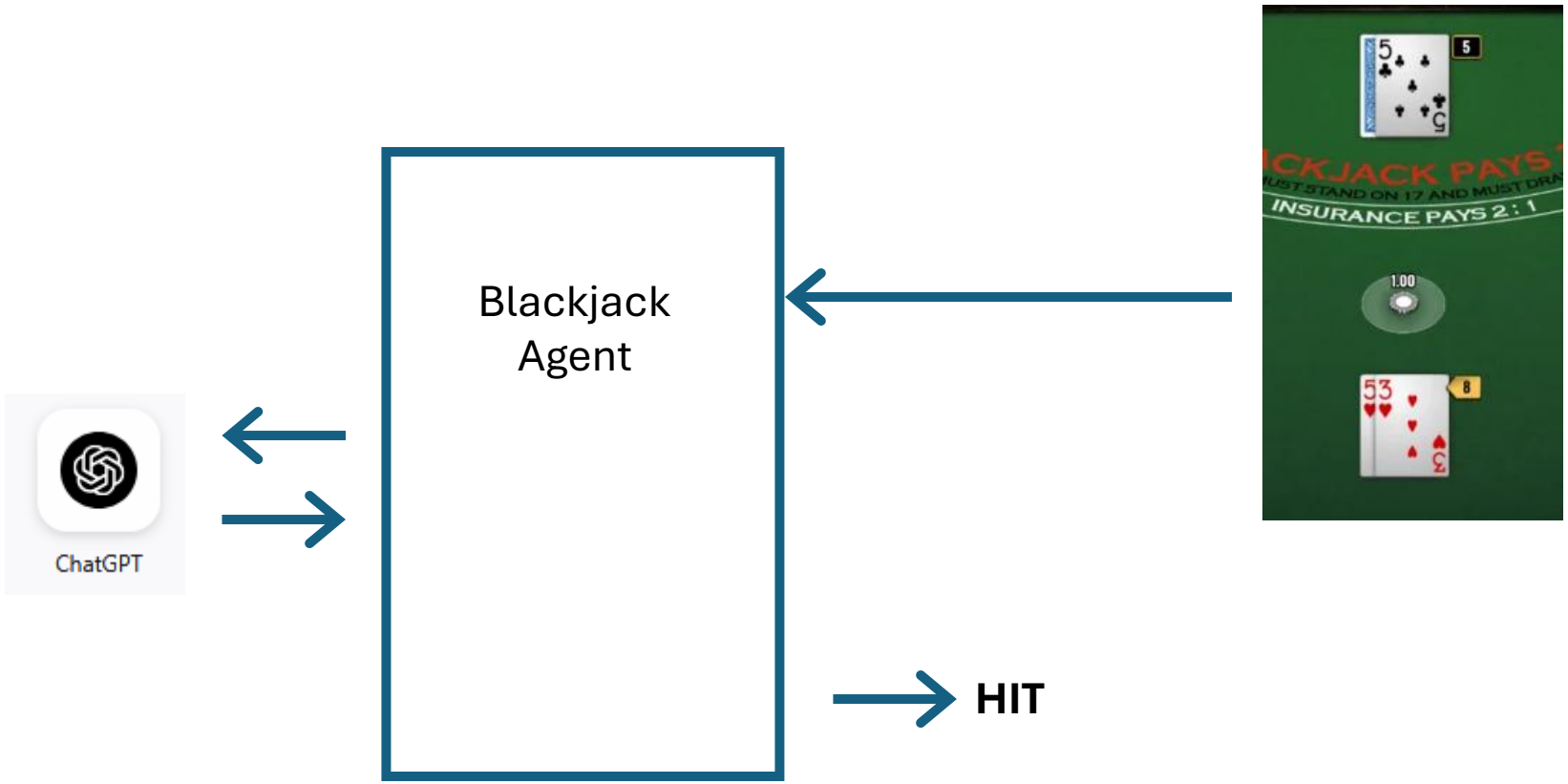
Some questions and one idea

- Do we have a resource that has lots of knowledge and answers to questions?
- Can we integrate this into an agent and then use this knowledge to guide the agent behaviour?
- These would be "INTELLIGENT AGENTS" (in reality would be zero-shot learning agents)



Let's start with an example

What if we use ZERO-SHOT Learning from an LLM?



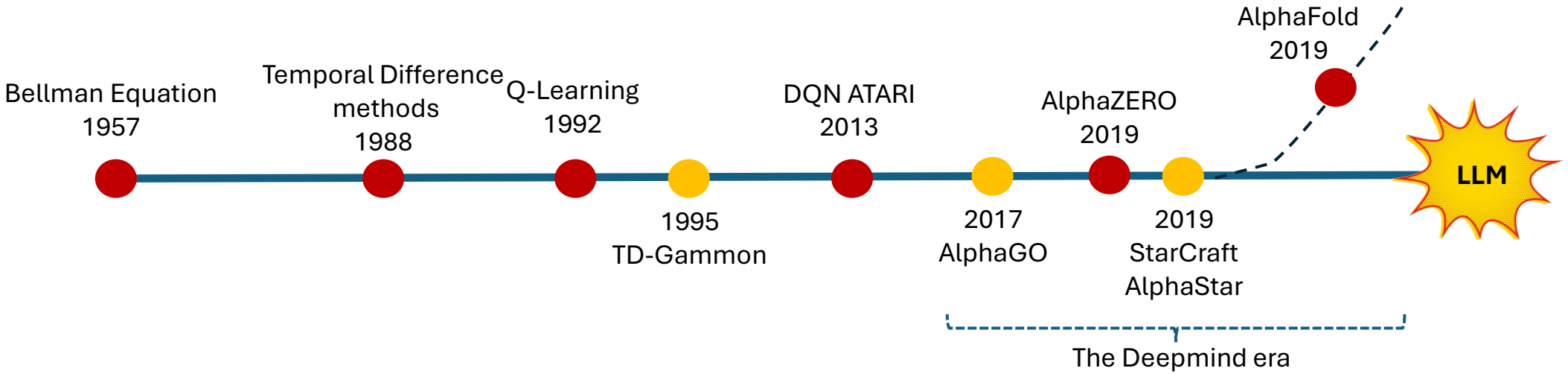
https://github.com/castorgit/RL_course/blob/main/090_LLM_Blackjack_langchain.ipynb
https://github.com/nwayt001/atari-gpt

The world of AI Agents

AI Agents
Hype or Reality

https://www.youtube.com/watch?v=FvG41iEXFrU
https://www.youtube.com/watch?v=EZqmBcqDkyw
https://www.youtube.com/watch?v=s4JNLL7U8H8
https://www.youtube.com/watch?v=wI2cBdo0XDw
https://www.youtube.com/watch?v=-35QjvFEmhE

A Brief story of Agents
From Bellman to LLM

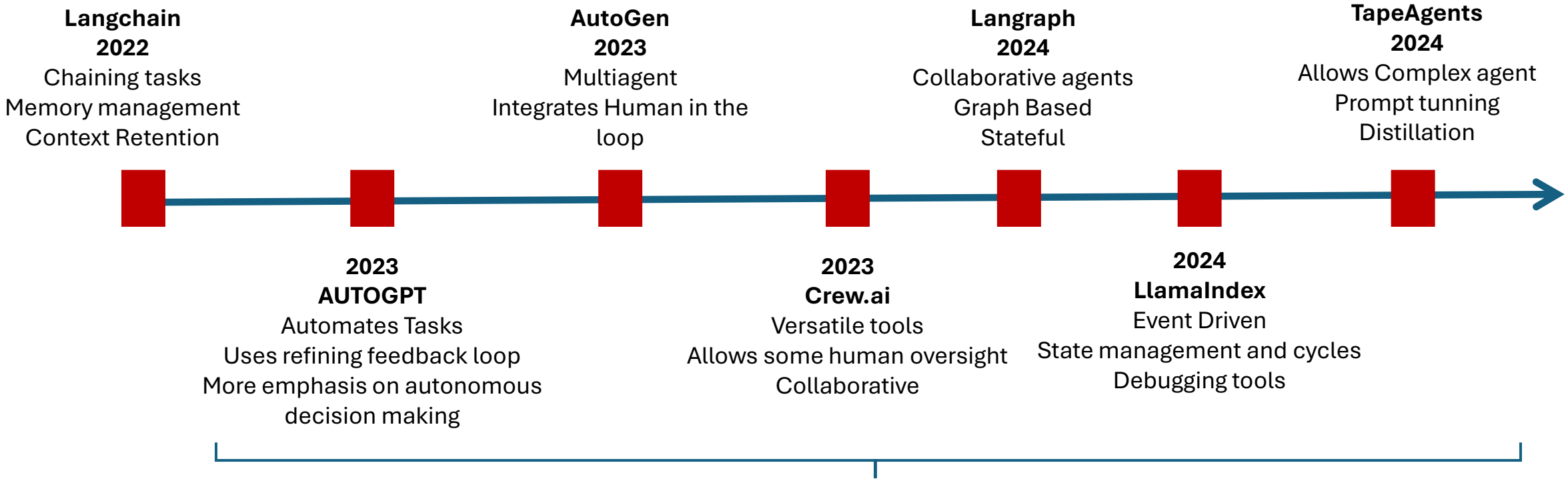


(AlphaStar 2019) Grandmaster level in StarCraft II using multi-agent reinforcement Learning
(Silver 2017) Mastering the game of GO without human knowledge
(David Silver 2013) Playing Atari with Deep Reinforcement Learning
(Tesauro 1995) Temporal Difference Learning and TD-Games
(Watkins 1992) Q-learning
(Bellman 1957) Dynamic Programming

A Brief story of Agents
After the LLM

(Anthropic 2024) Introducing the MCP
(He 2024) WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models
(Lú 2024) WebLINX: Real-World Website Navigation with Multi-Turn Dialogue
(Gur 2023) A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis (DeepMind)

A Brief story of Agents
Frameworks

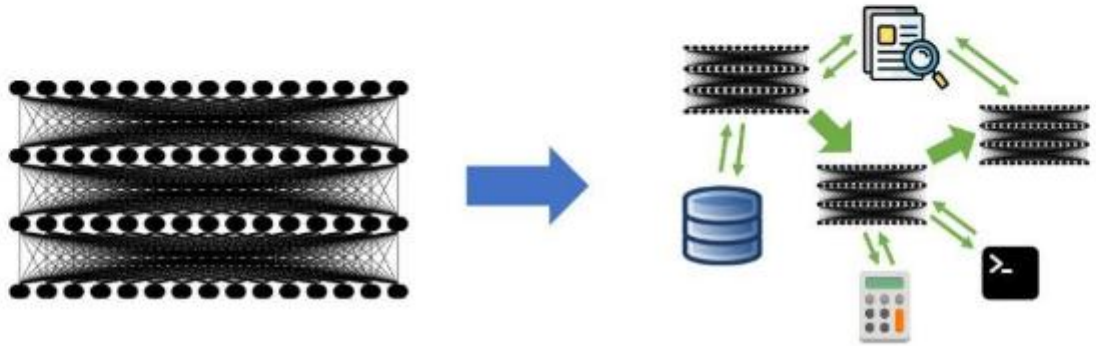


As of 1st June 2025, there are more than 100 frameworks available for AI Agents

https://aiagentsdirectory.com/category/ai-agents-frameworks

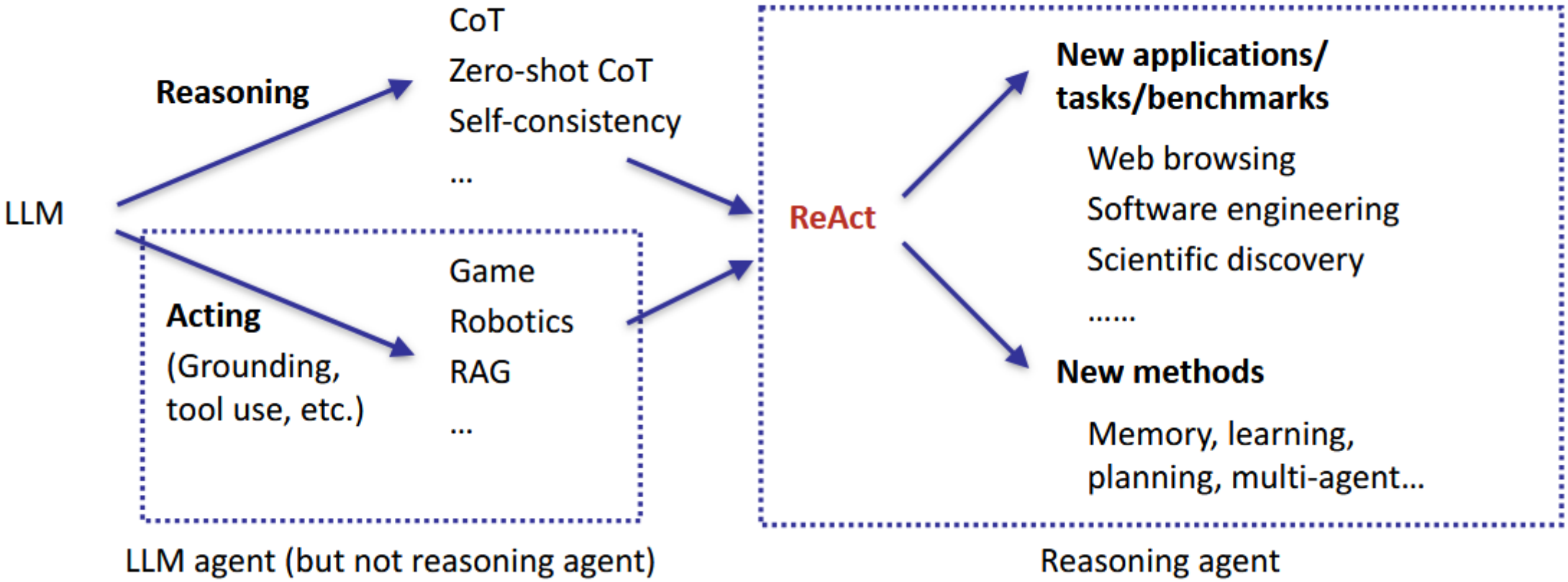
From Generative AI to Agents AI

- Generative
  - Generate content like text & image
- Agentic
  - Execute complex tasks on behalf of human



(Zaharia et al 2024) The Shift from Models to Compound AI Systems

From LLM Agents to ReAct agents



What are LLM-Based Agents
The AI Agents World

From RL to AI Agents

Reinforcement Learning Agents

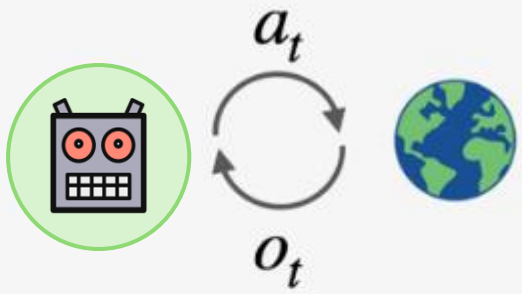
- Require long training runs
- Limited action space
- Low generalization
- Too focused on Games
- Safe about what they've learned

AI-Based Agents

- Zero-shot Task solvers
- Use world background knowledge
- LLMs have probably been trained on the documentation of the software you are using!
- They can become React Agents – Reasoning and Coordinated

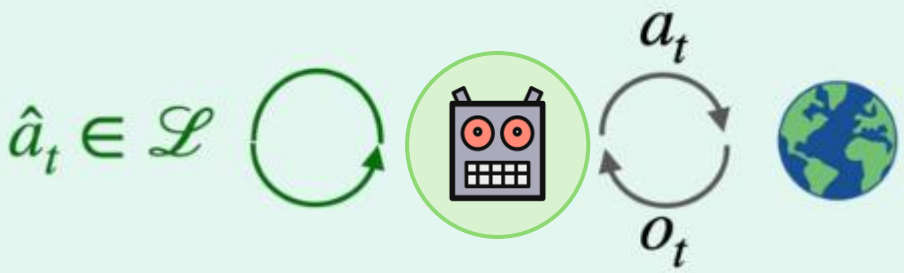
From traditional agents to ReAct Agents

Traditional agents: action space A defined by the environment



- **External feedback** o_t
- Agent context $c_t = (o_1, a_1, o_2, a_2, \cdots, o_t)$
- Agent action $a_t \sim \pi(a \mid c_t) \in A$

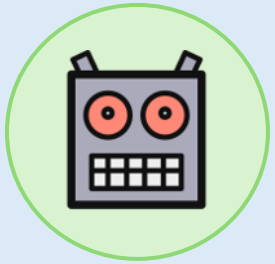
ReAct: action space $\hat{A} = A \cup \mathscr{L}$ augmented by reasoning



- $\hat{a}_t \in \mathscr{L}$ can be any language sequence
- Agent context $c_{t+1} = (c_t, \hat{a}_t, a_t, o_{t+1})$
- $\hat{a}_t \in \mathscr{L}$ only updates **internal context**

Definitions

An AI Agent is an autonomous software entity that can



AI Agent

- Perceive
- Decide
- Reason
- Act

Essential Characteristics

- Autonomous
- Proactive (to achieve goals)
- Reactive (responds to environmental changes)
- Social – Interacts with other agents and humans

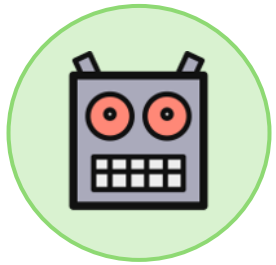
AI Agent

- Dynamic
- Active
- Intelligent

Traditional Software

- Passive
- Rule-Based
- Static

Key Characteristics



AI Agent

Autonomy

Operates without continuous human guidance

- Self-directed decision making
- Independent goal pursuit
- Minimal supervision
- Always on

Reactivity

Responds appropriately to environmental changes

- Real-time adaptation
- Dynamic behaviour adjustment
- Context-aware Responses

Proactiveness

Takes initiative to achieve objectives

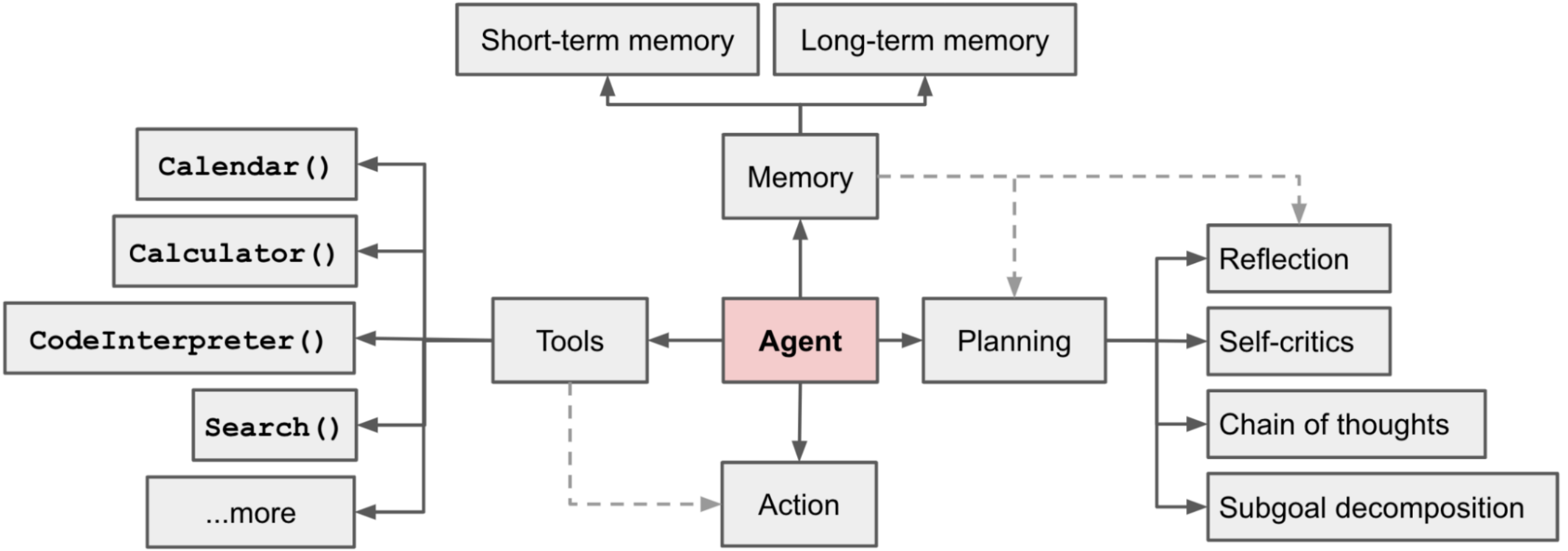
- Anticipates future needs
- Strategic action planning
- Takes opportunistic actions

Social Ability

Interacts with other agents and humans

- Multi-agent collaboration
- Negotiation capabilities
- Use of Natural Language-Prompting-

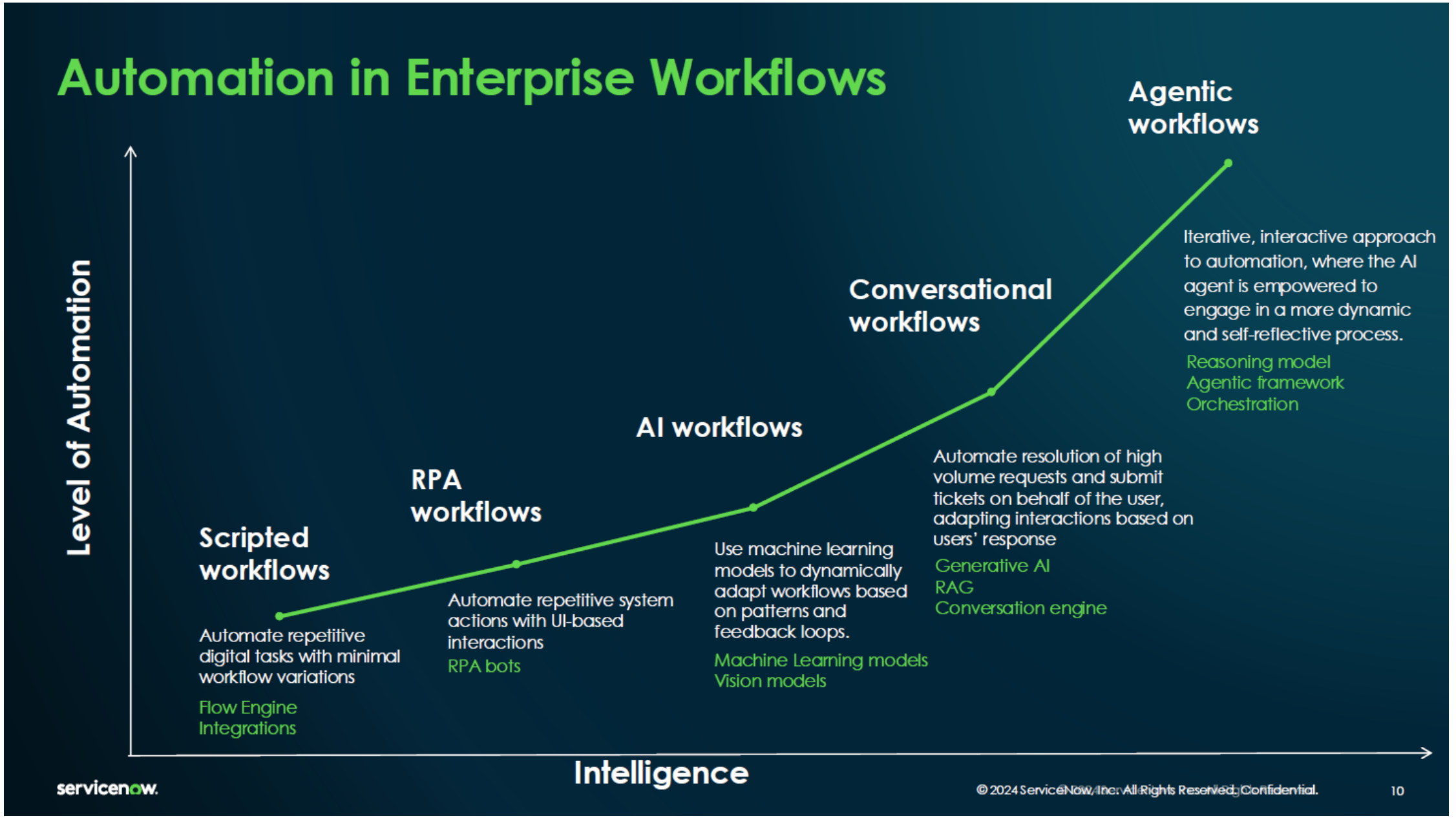
Architecture of an AI-Agent



https://lilianweng.github.io/posts/2023-06-23-agent/

Is it a change of Paradigm?

Applying Agents to automate Enterprise Workflows



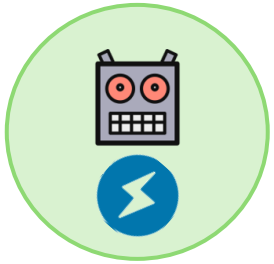
Why is this so transformative and relevant

https://x.com/ffschumann/status/1931689353123602482

https://x.com/kimmonismus/status/1928465270969901248

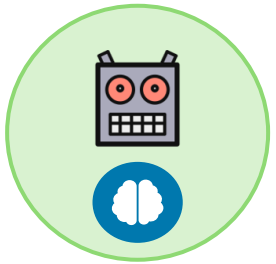
AI Agents
Types and Architectures

4 Main Architectures



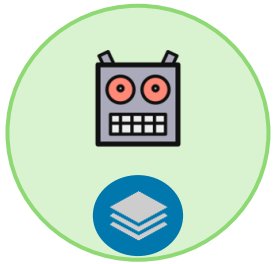
React

- Stimulus / Response behaviour
- Alternative to APIs
- Simple Reasoning



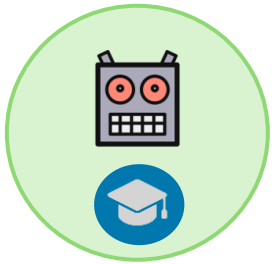
Deliberative

- Goal-oriented planning
- Reasons
- Develops complex plans
- Orchestration capabilities



Hybrid

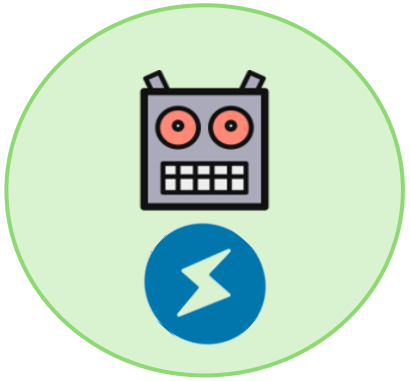
- Agents made of other agents
- Multiple layers
- Can integrate agents of other kinds



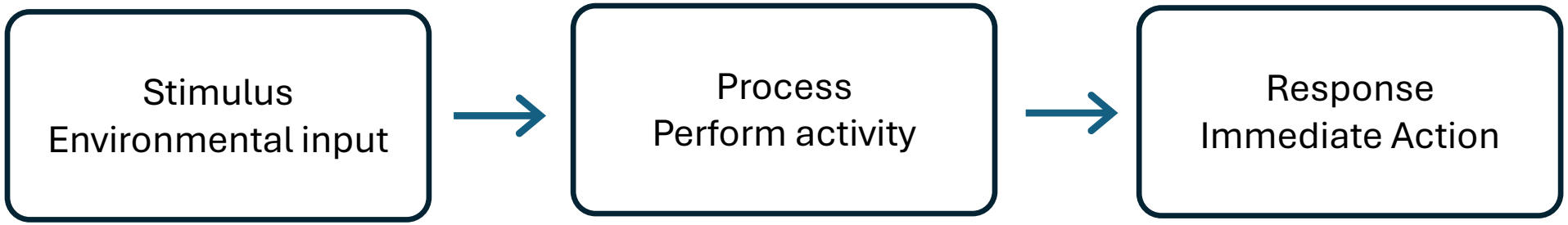
Learning

- Adaptive and self-improving
- Similar to RL (somehow)
- Keep learning from environment

Reactive Agents - APIs

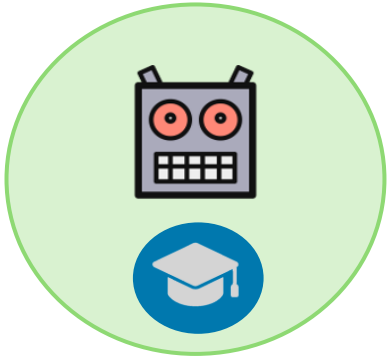


React

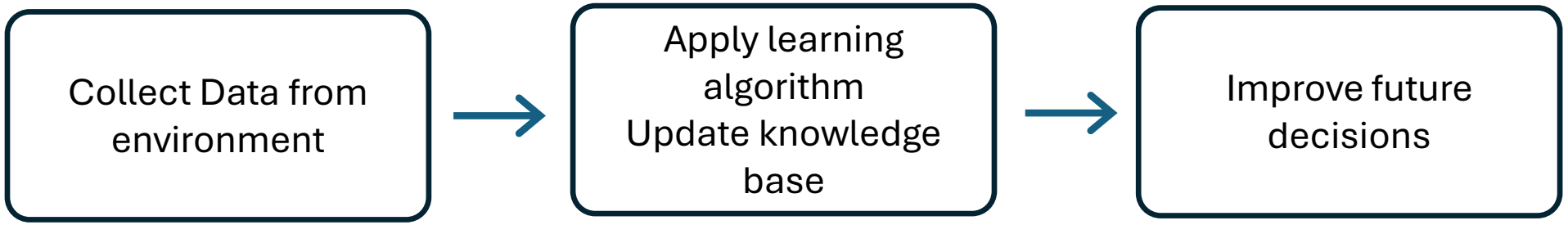


- Simple Architecture
- No internal state
- Immediate reaction
- Some internal Rules

Learning Agents

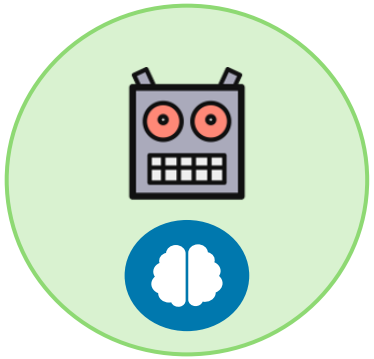


Learning

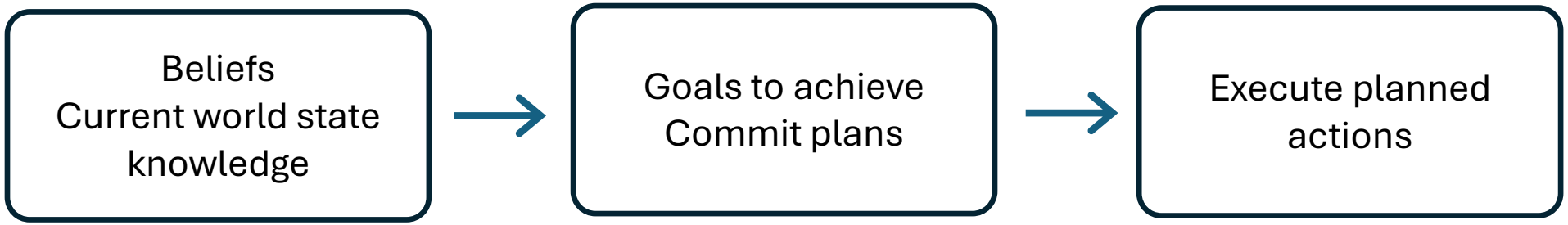


- Require simulation of real environment
- Trained using reinforcement learning
- Complex to adapt to new situations
- Expensive to train

Deliberative agents



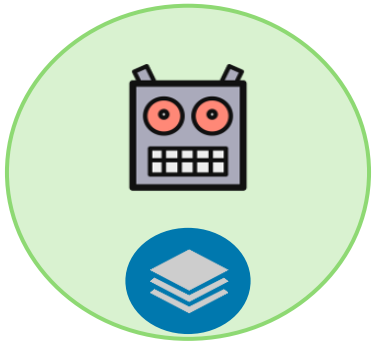
Deliberative



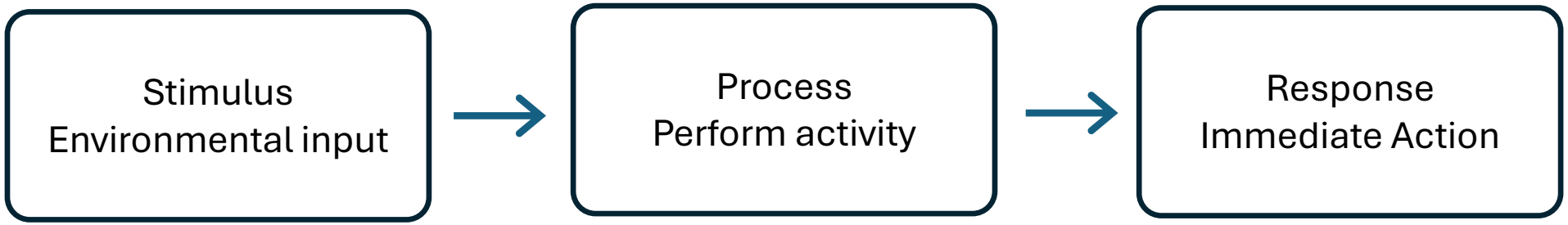
- Philosophical construct
- Very complex
- Require understanding of world and Beliefs
- Would find unexpected solutions

(Martin 2024) Constructing deliberative agents using a hybrid system

Hybrid Agents - APIs



Hybrid



- Combine different AI methodologies
- Can use non-AI tools like rule based
- Can integrate other AI- tools

AI Agents and our IT infrastructure

API agents

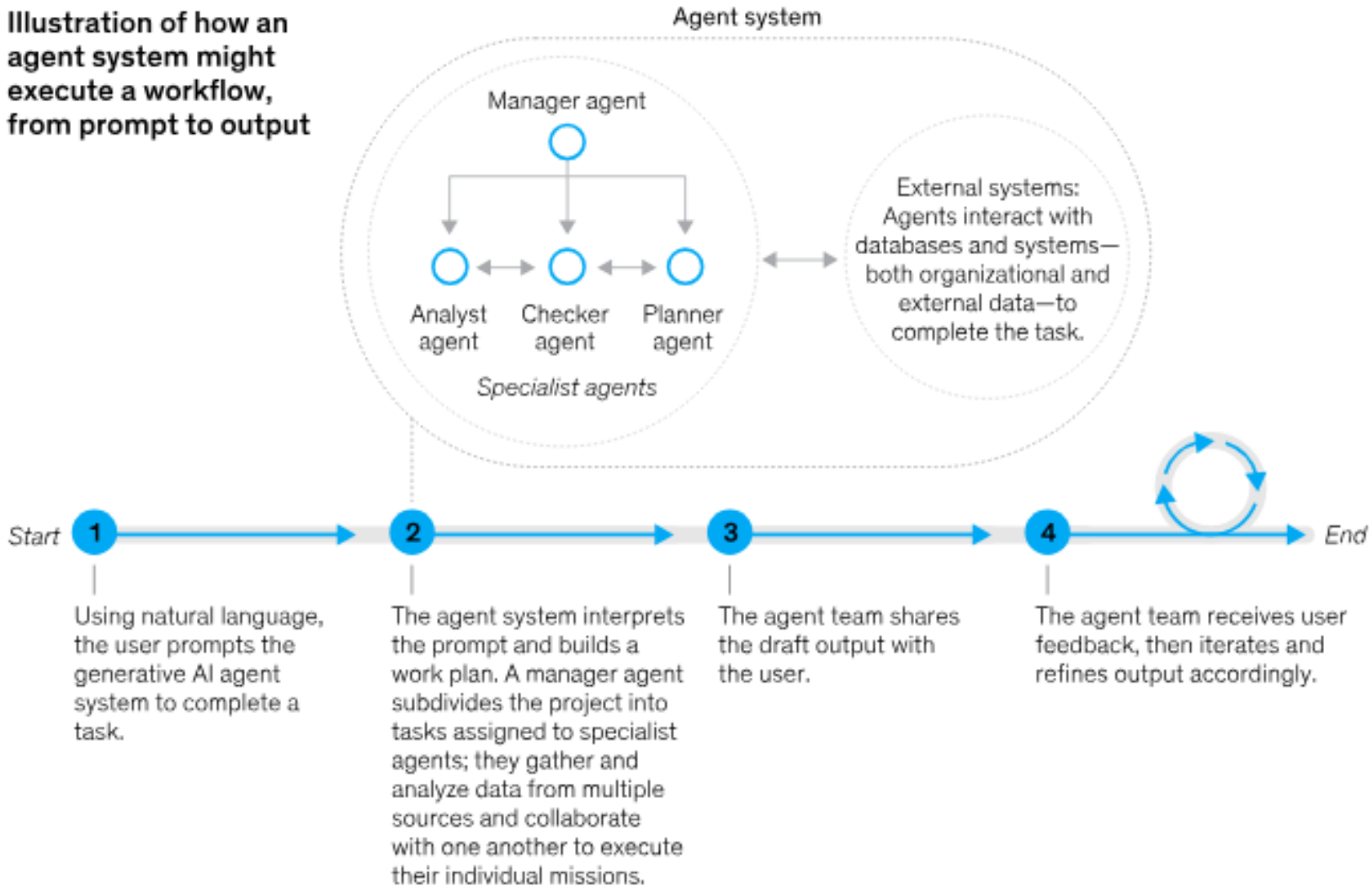
- Observations: API call results, search history, user-uploaded images, chat history
- Actions: API calls, search calls, responses to the user
- Pros: Lower latency, lower risks
- Cons: needs appropriate APIs

Web agents

- Observations: what human would see + accessibility tree / raw Domain
- Actions: enter text in fields, clicks
- Pros: can do anything
- Cons: higher latency, higher risks

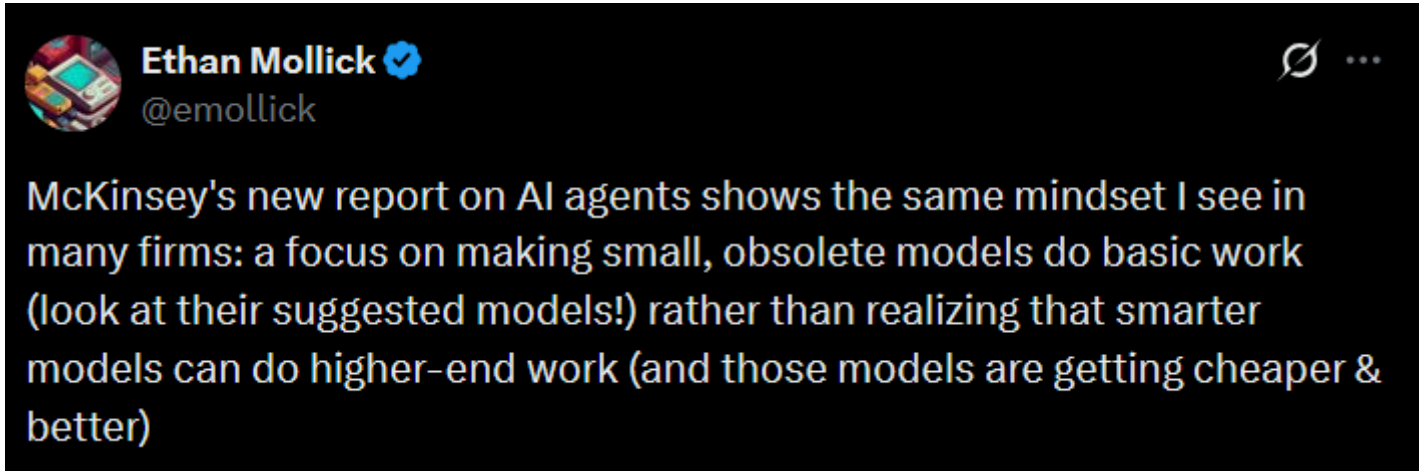
Agents enabled by generative AI soon could function as hyperefficient virtual coworkers.

Illustration of how an agent system might execute a workflow, from prompt to output



McKinsey & Company

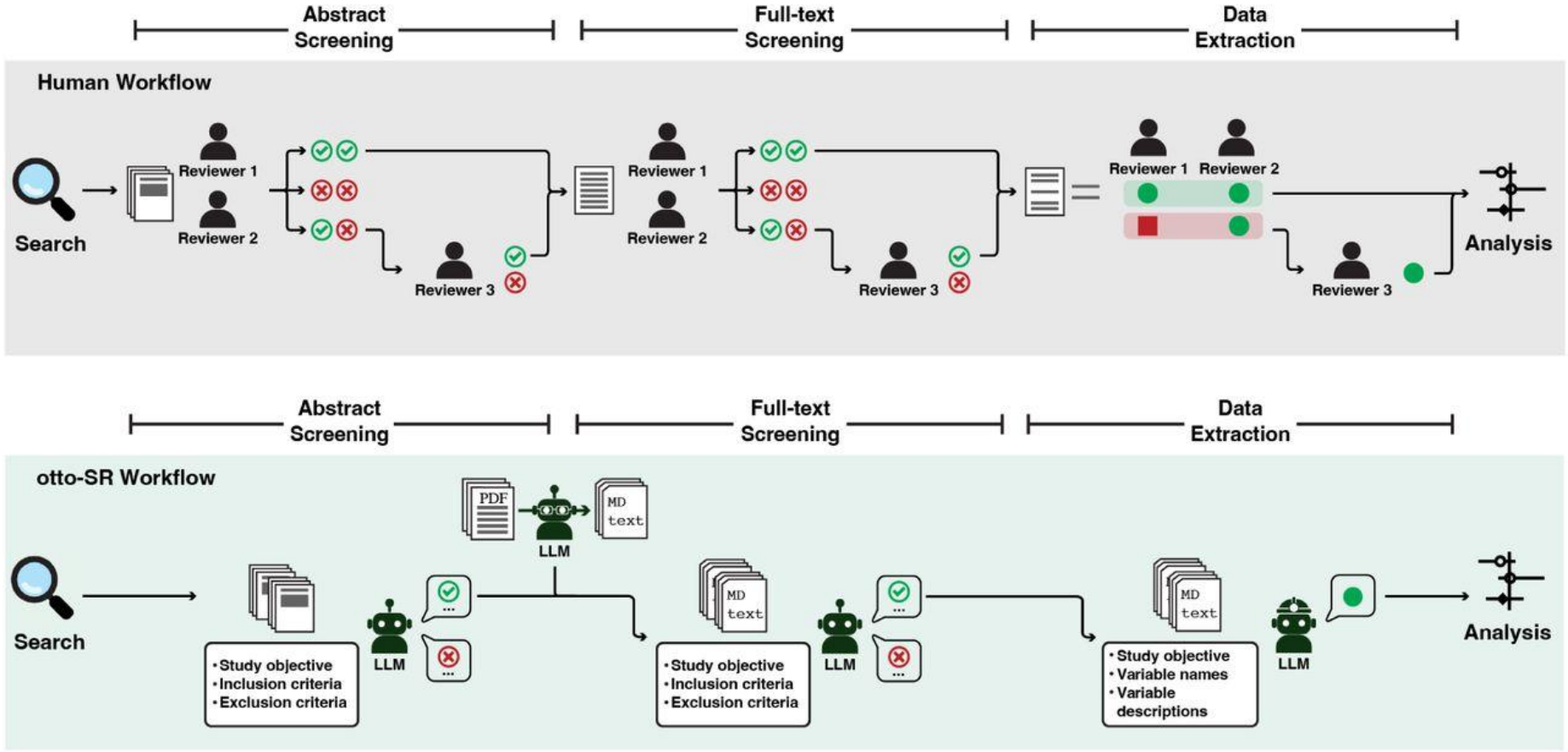
The big discussion Basic Work High Volume vs High-End work Low Volume



Entry Jobs or CEO?

An example of a pipeline saves 12 person / year of work

Using an autonomous agent based on o3-mini and GPT-4.1, a team from Harvard, MIT & other institutions reproduced and updated an entire issue of Cochrane Reviews in two days… saving 12 person-years of work! (and more accurate than humans)

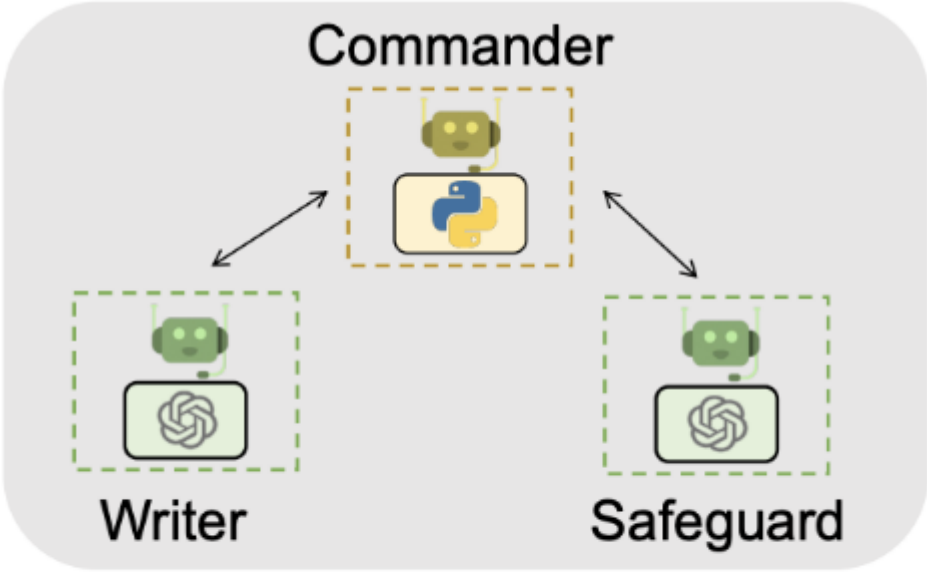


(Cao 2025) Automation of Systematic Reviews with Large Language Models

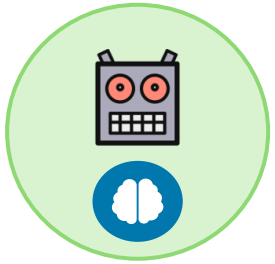
How to program Agents

Orchestrating multiple agents

- Agents must be easy to understand maintain and extend
- Modular composition (tools)
- Natural Human participation (prompting)
- Fast and creative experimentation



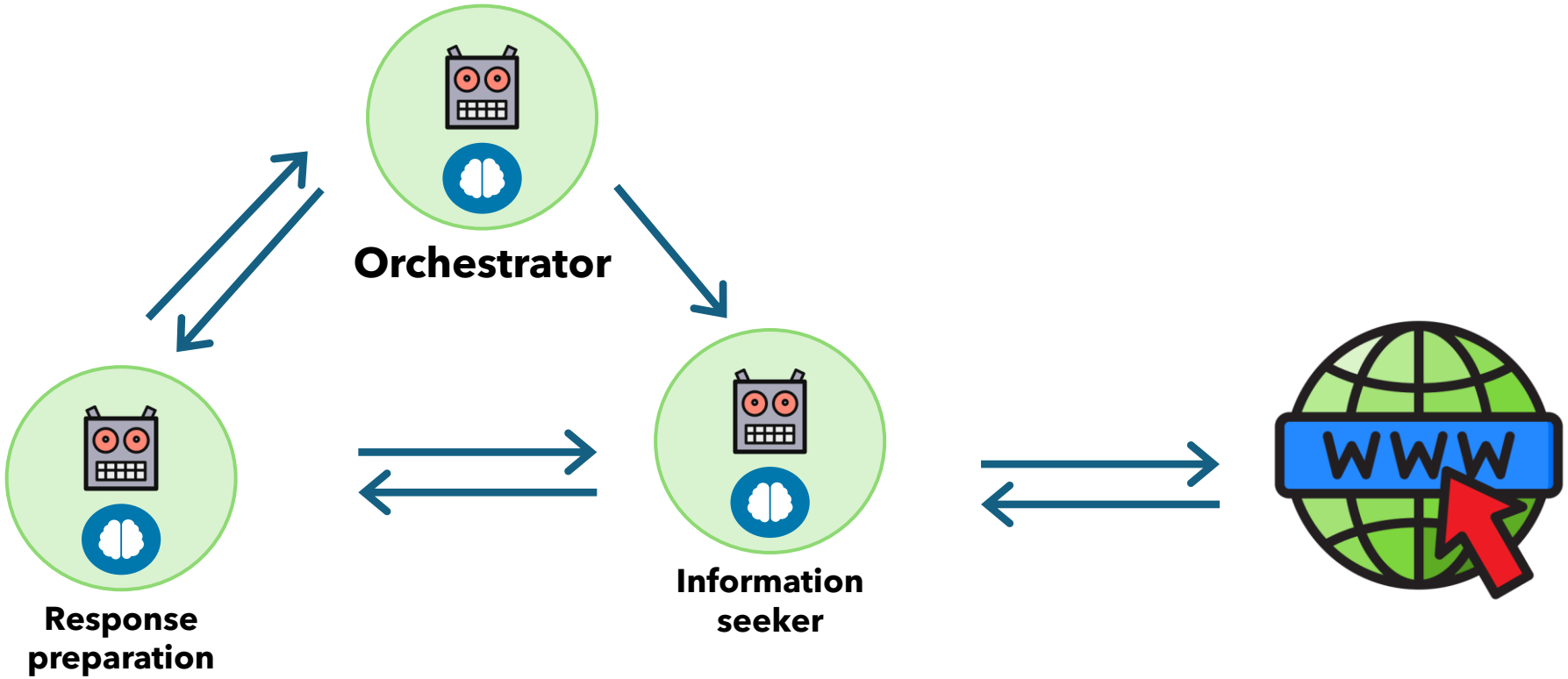
An Agent



Agent

- **LLM - Brain**
- **Tools /skills**
  - API Calls
  - Access internet / scrapping
  - Code interpretation
  - DB Schema interpretation
- **Behaviour description - prompting**
- **Memory - Persistence**
- **Coordination**

Collaborative Networks

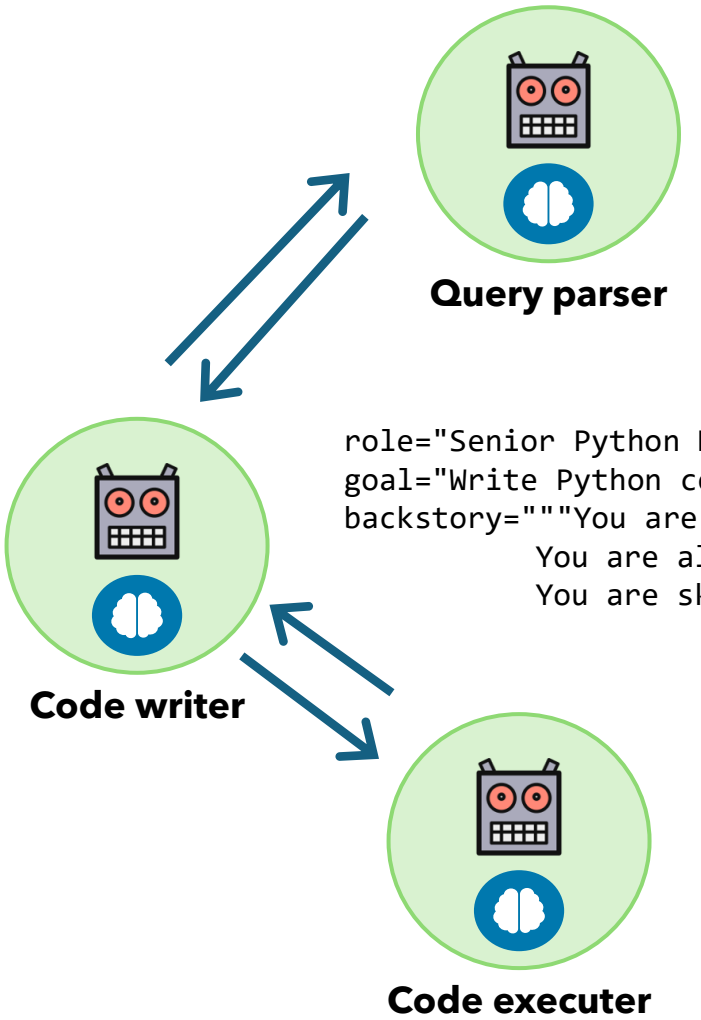


crew.ai example

```
from crewai import Agent

researcher = Agent(
        role='Technology Researcher’,
        goal='Explore groundbreaking AI technologies’,
        backstory="You are driven by curiosity and have a knack for uncovering emerging trends.",
        verbose=True,
        allow_delegation=False,
        tools=[ScrapeWebsiteTool(), ...], # Add necessary tools here
        llm=OpenAIFunctions(model_name="gpt-4") # Or any other compatible model
)
```

```
# 1) Query parser agent
query_parser_agent = Agent(
    role="Stock Data Analyst",
    goal="Extract stock details and fetch required data from this user query: {query}.",
    backstory="You are a financial analyst specializing in stock market data retrieval.",
    llm=llm,
    verbose=True,
    memory=True,)
query_parsing_task = Task(
    description="Analyze the user query and extract stock details.",
    expected_output="A dictionary with keys: 'symbol', 'timeframe', 'action'.",
    output_pydantic=QueryAnalysisOutput,
    agent=query_parser_agent,)

# 2) Code writer agent
code_writer_agent = Agent(
    role="Senior Python Developer",
    goal="Write Python code to visualize stock data.",
    backstory="""You are a Senior Python developer specializing in stock market data visualization.
                 You are also a Pandas, Matplotlib and yfinance library expert.
                 You are skilled at writing production-ready Python code""",
    llm=llm,
    verbose=True,)
code_writer_task = Task(
    description="""Write Python code to visualize stock data based on the inputs from the stock analyst
                   where you would find stock symbol, timeframe and action.""",
    expected_output="A clean and executable Python script file (.py) for stock visualization.",
    agent=code_writer_agent,)
```

A network of 3 coordinated agents




```
role="Stock Data Analyst",
goal="Extract stock details and fetch required data from this user query: {query}.",
backstory="You are a financial analyst specializing in stock market data retrieval.",
```

```
role="Senior Python Developer",
goal="Write Python code to visualize stock data.",
backstory="""You are a Senior Python developer specializing in stock market data visualization.
          You are also a Pandas, Matplotlib and yfinance library expert.
          You are skilled at writing production-ready Python code""",
```

```
description="""Write Python code to visualize stock data based on the inputs from the
          stock analyst where you would find stock symbol, timeframe and action.""",
expected_output="A clean and executable Python script file (.py) for stock
                    visualization.",
```

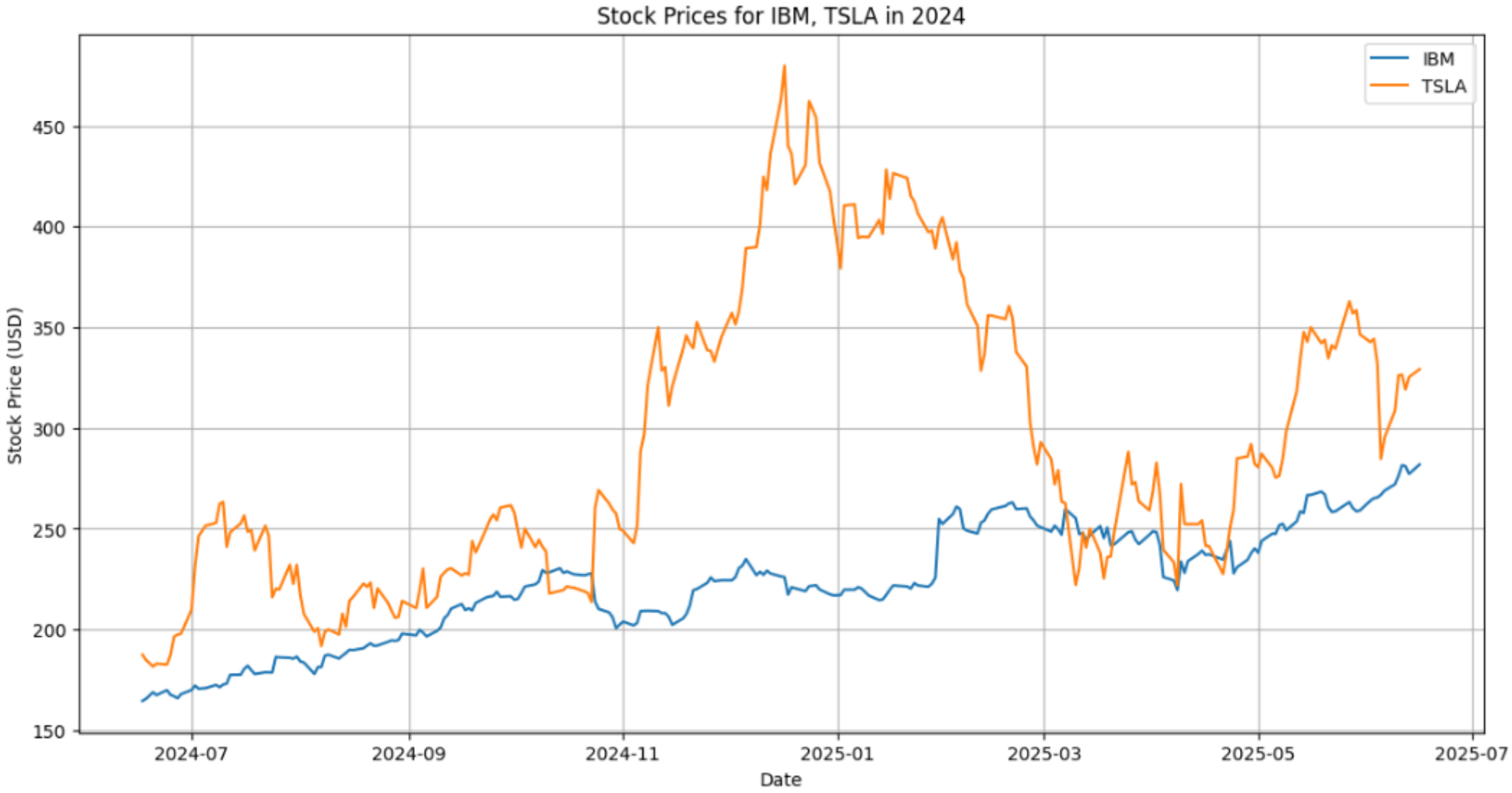
Query and result

Query *Plot 2024 stock values of IBM and plot 2024 stock values of TESLA in the same figure*

Result

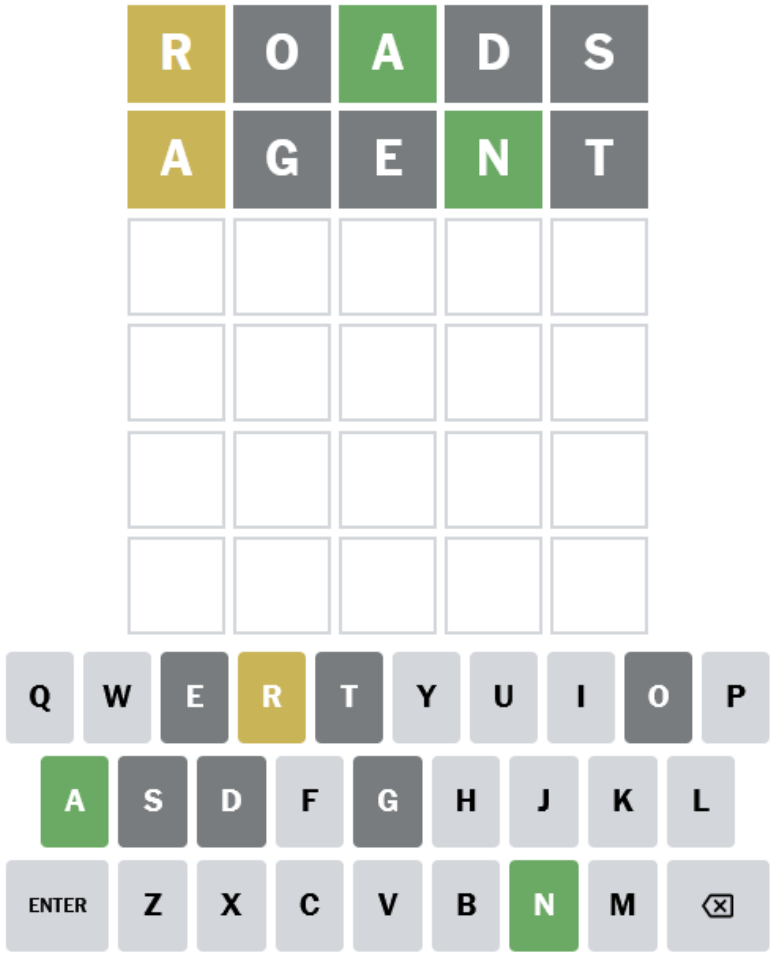
```
import yfinance as yf
import matplotlib.pyplot as plt

symbol = 'IBM TSLA’
symbols = symbol.split()

stock_data = yf.download(symbols, period='1y’)

plt.figure(figsize=(14, 7))
for sym in symbols:
        plt.plot(stock_data['Close'][sym], label=sym)
plt.title(f'Stock Prices for {', '.join(symbols)} in 2024’)
plt.xlabel('Date’)
plt.ylabel('Stock Price (USD)’)
plt.grid(True)
plt.legend()
plt.show()
```

Final output



https://github.com/castorgit/RL_course/blob/main/090_AI_example_CREW.ipynb

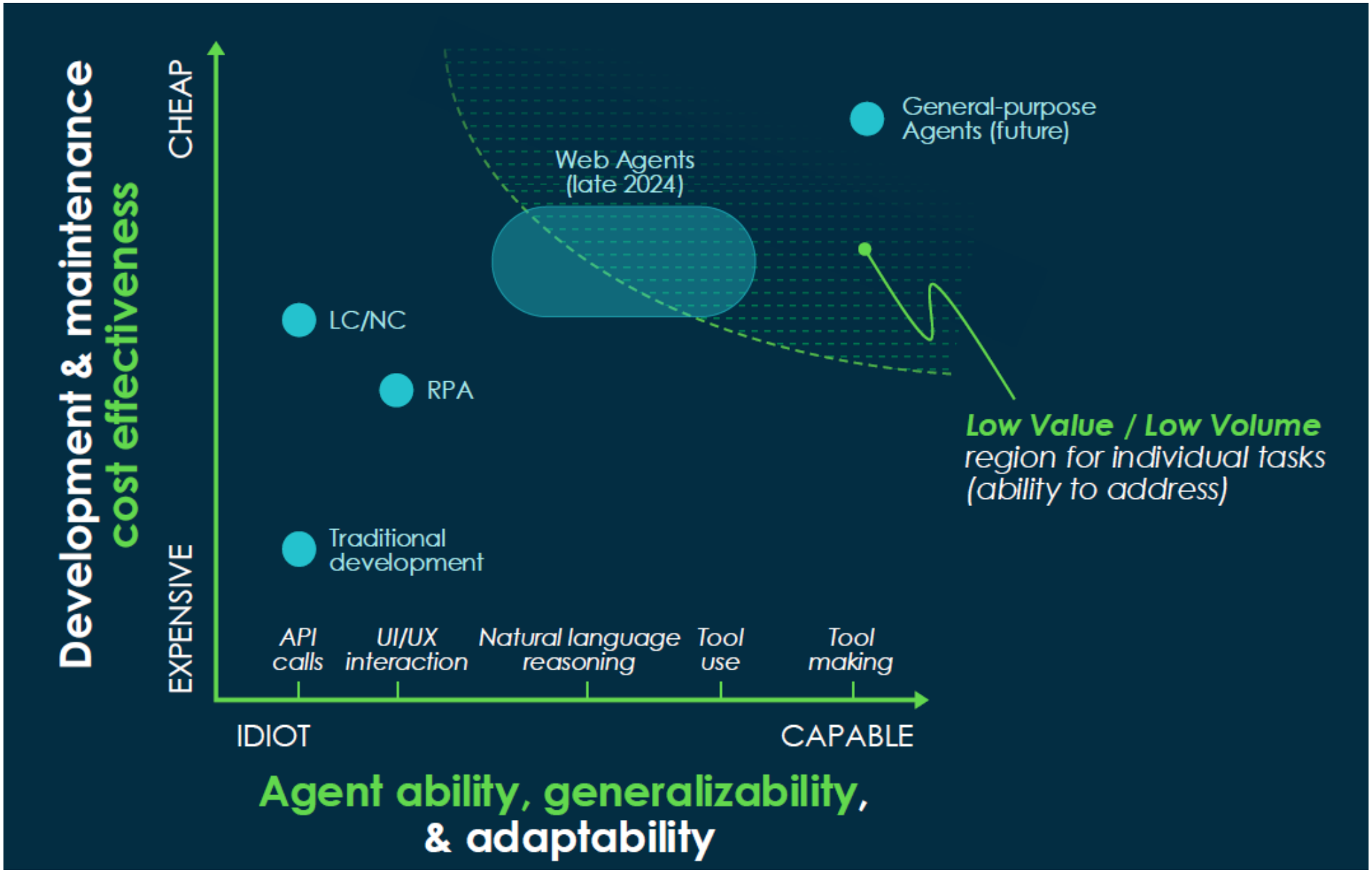
Solving Wordle with agents



Play with this one : https://github.com/OktayBalaban/Wordle_Bot

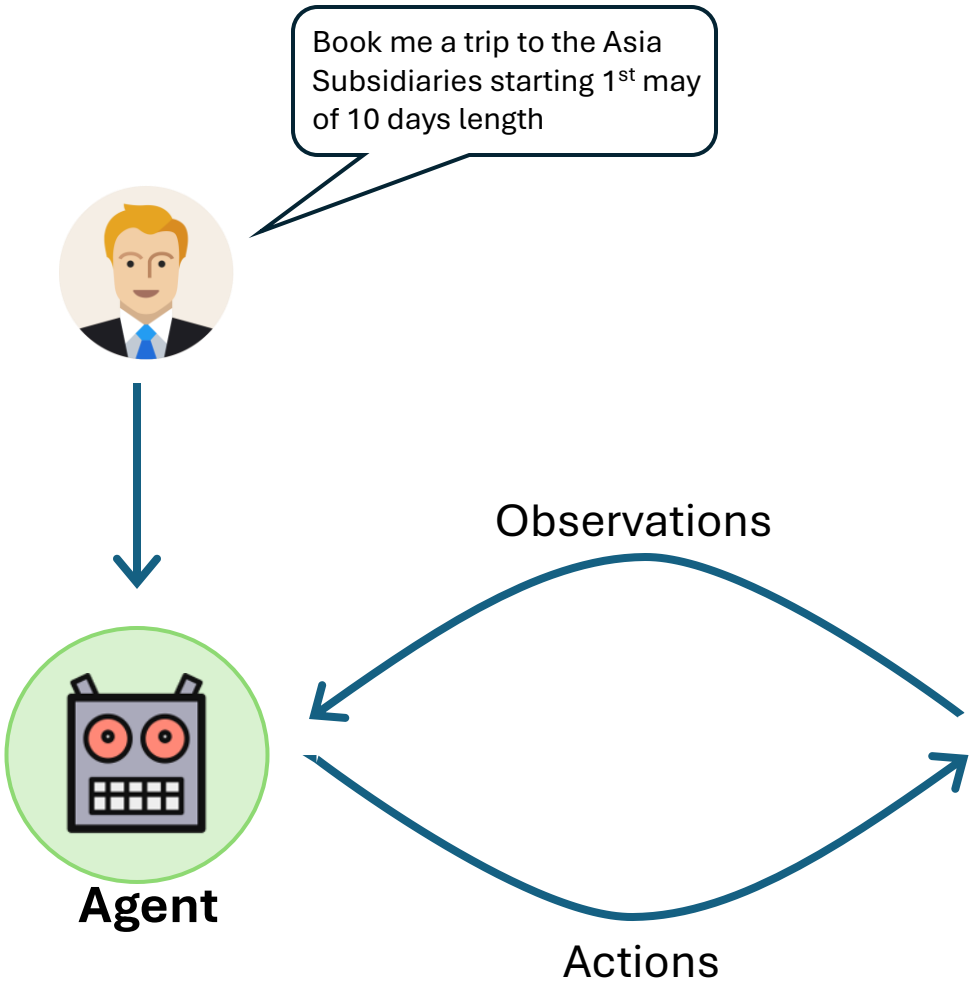
Web Agents

Agents may replace Low value, low volume tasks

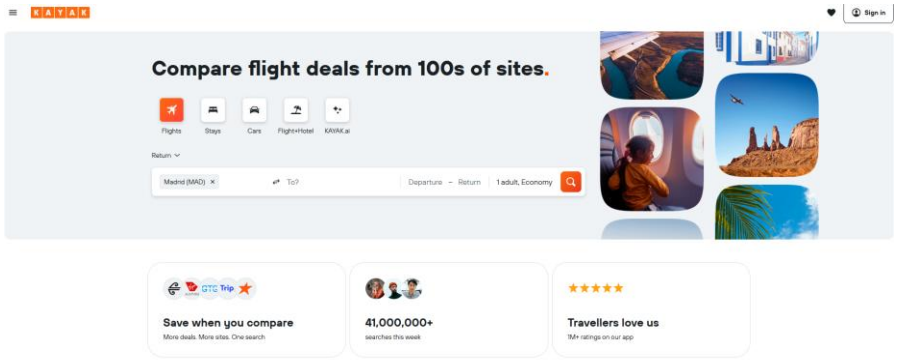


What a Web Agent does

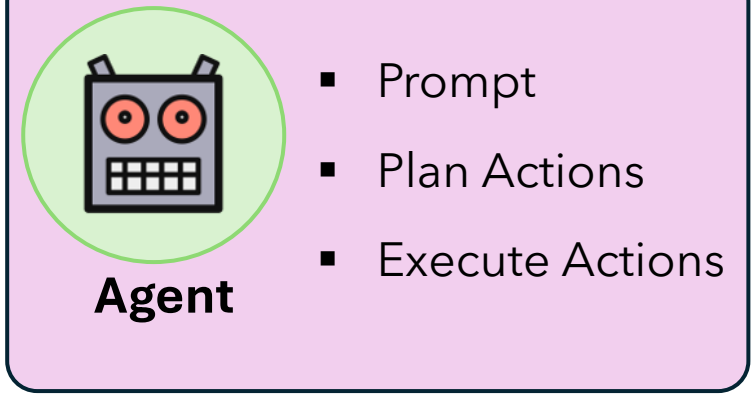
What does it imply



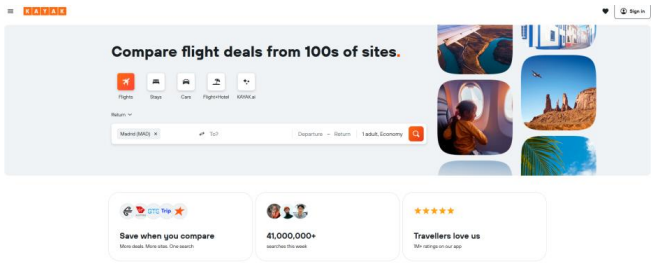
- Understand Task and Goal
- Situational awareness
- Long-mid term planning
- Detailed steps execution



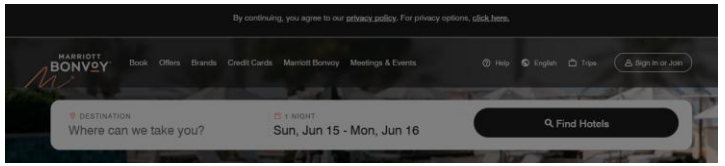
A Web agent can be a sequence of actions



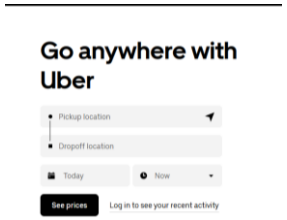
Reserve Flights



Reserve Hotels

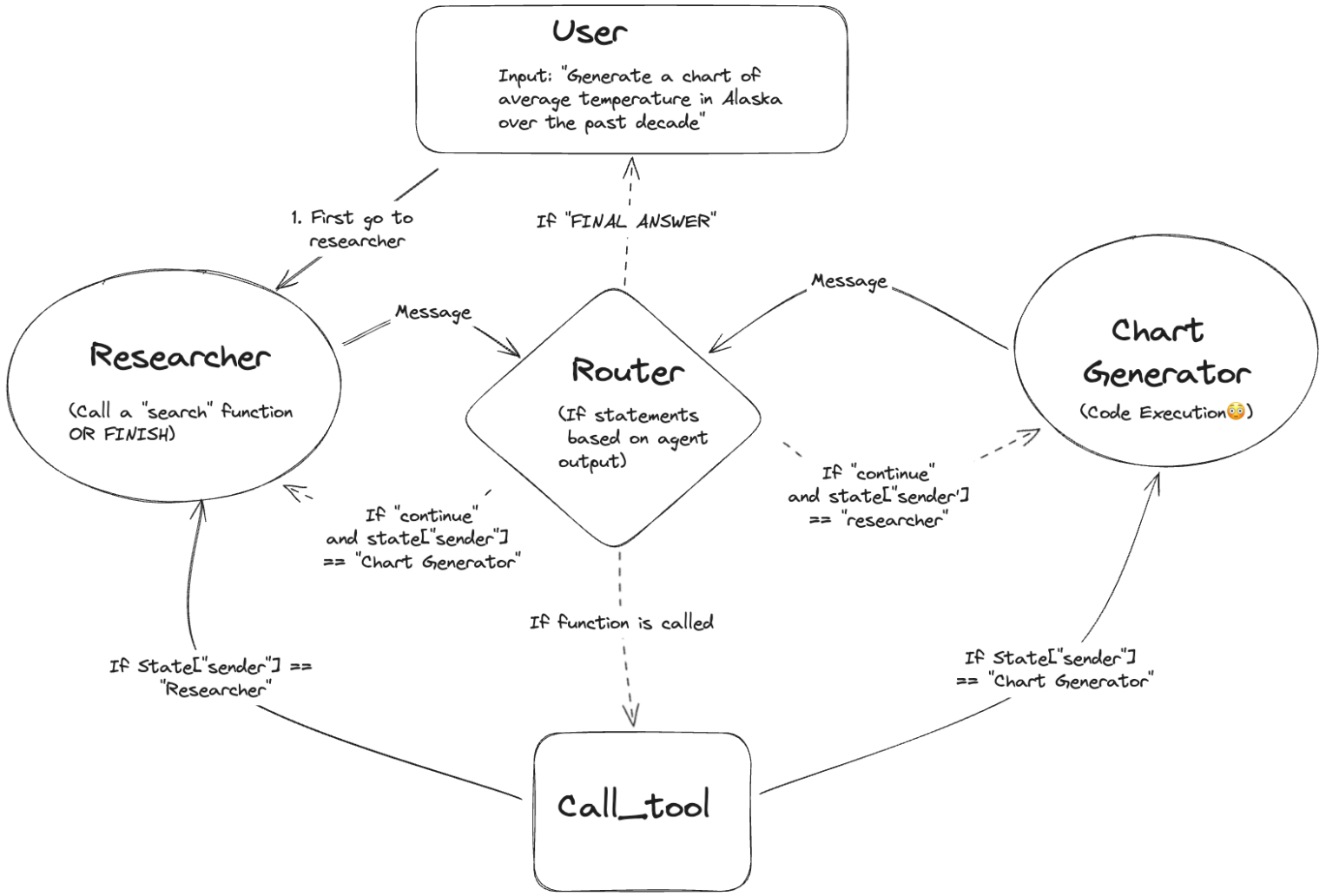


Reserve Transportation



Web Agents use tools in a graph

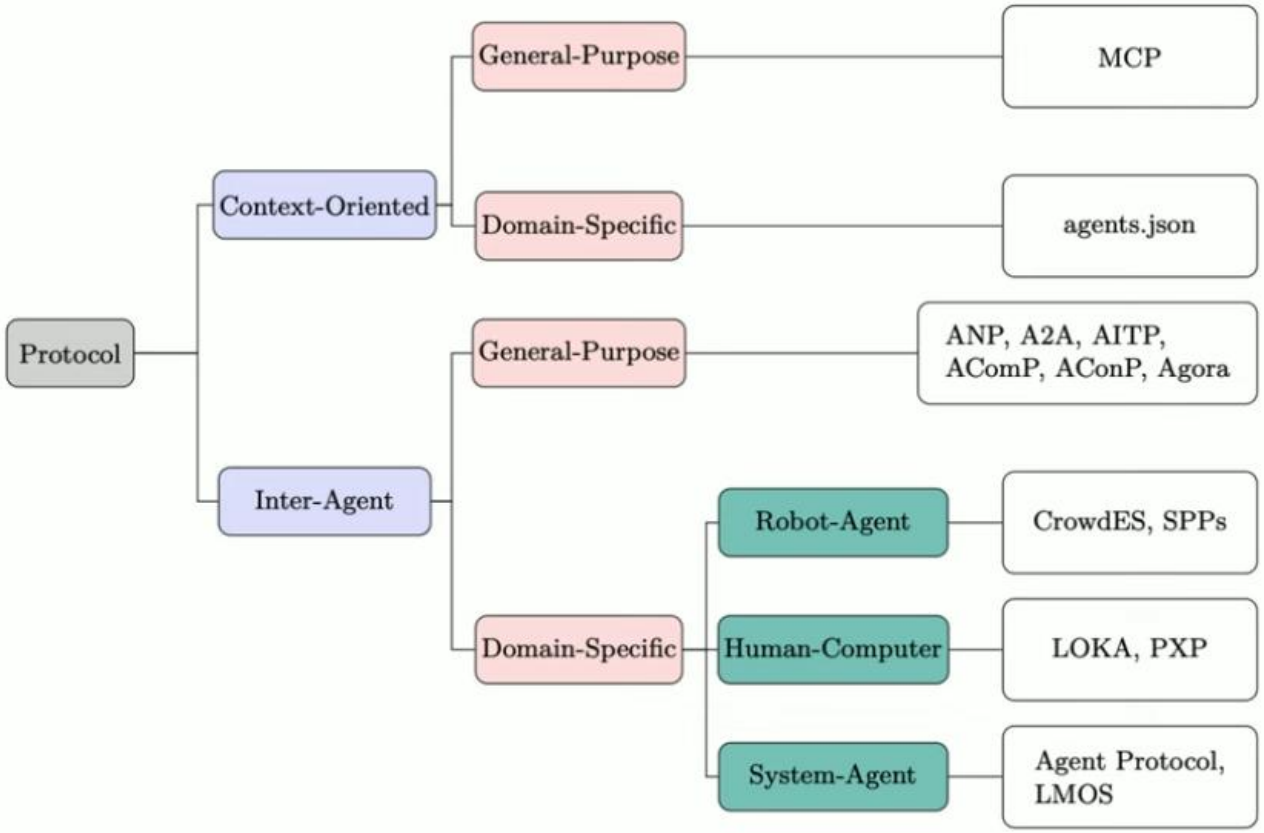
Example of Multi-Agent
With Langraph



https://github.com/langchain-ai/langgraph/blob/main/docs/docs/tutorials/multi_agent/agent_supervisor.ipynb

MCP
(an agent to agent protocol)

Taxonomy of agent Protocols



- An Agent should discover other agents
- Should authenticate with them
- Use common tools
- And collaborate to obtain a task

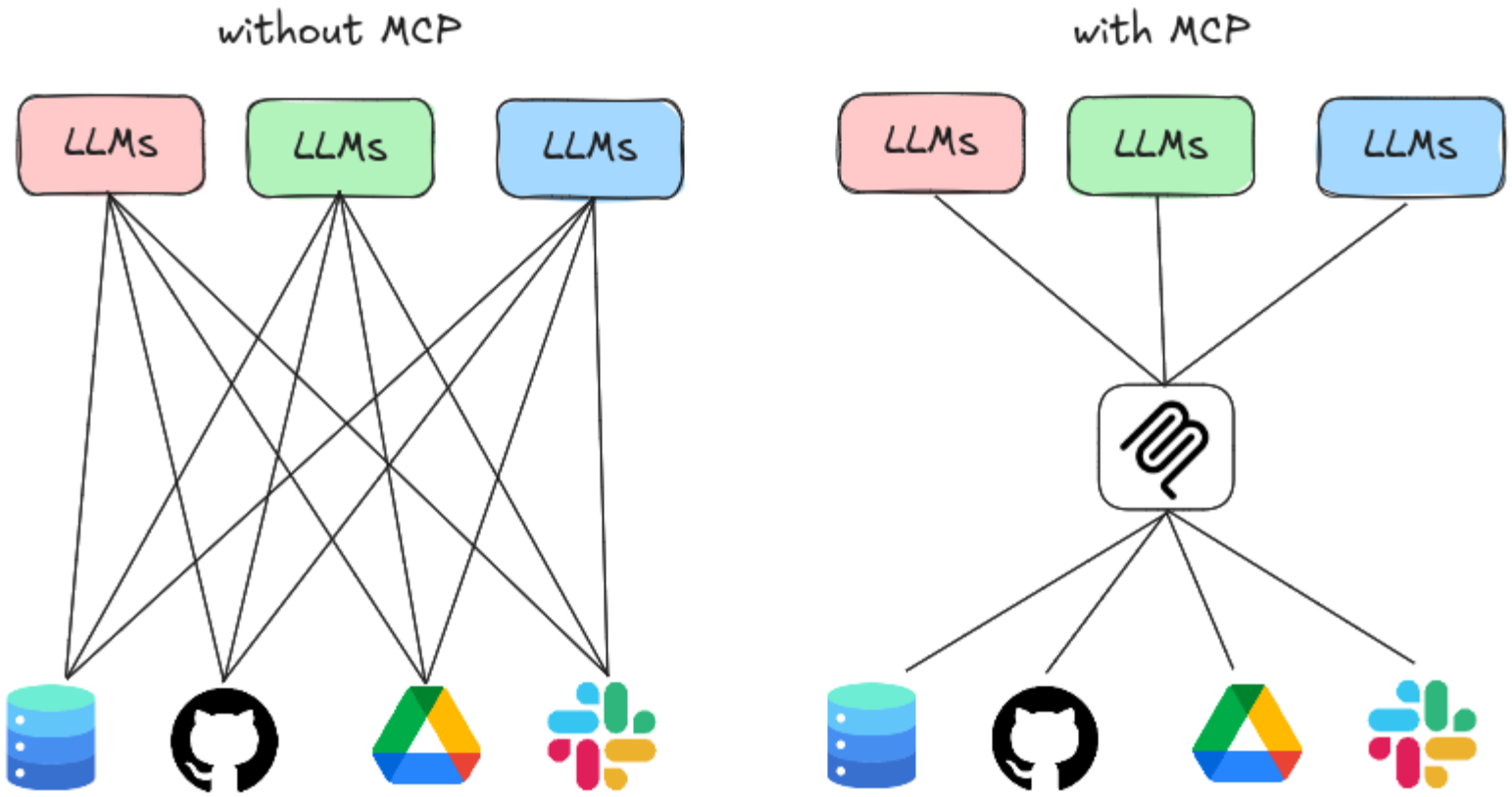
A Survey of AI Agent Protocols, Yang et al 2025

arxiv.org/abs/2504.16736

"Within context-oriented interactions, interactive tools can be regarded as low-autonomy agents. Conversely, in agent-to-agent interactions, the communicating agents can also be viewed as tools with higher autonomy, designed to accomplish specific intelligent tasks."

– Yang et al (2025)

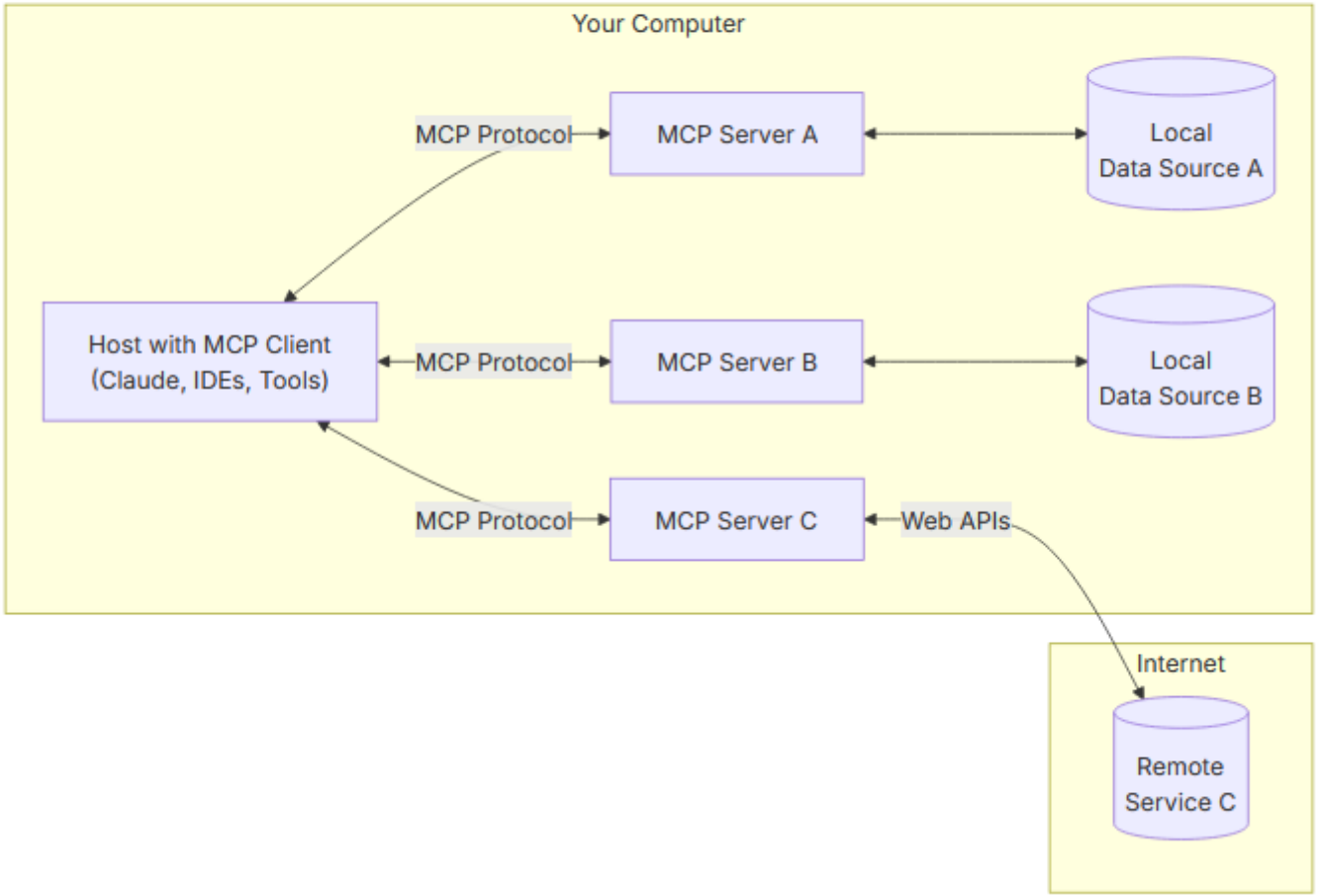
What is MCP then



The Anthropic Paper

General architecture

At its core, MCP follows a client-server architecture where a host application can connect to multiple servers:

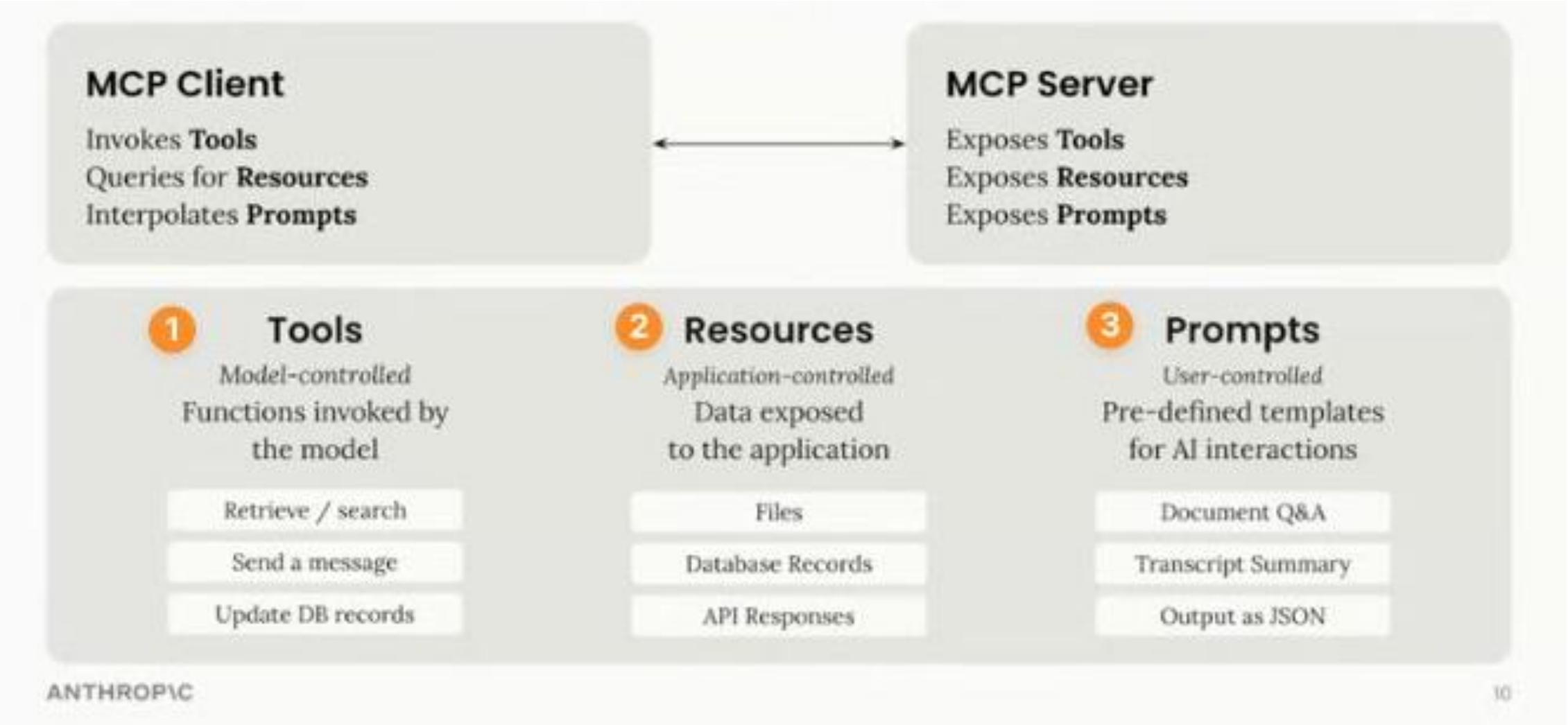


https://www.anthropic.com/news/model-context-protocol
https://modelcontextprotocol.io/introduction

Why is so important

Feature	APIs	MCP
Setup	One by one	One time for al tools
Flexibility	Fixed, one per tool	Dynamic and adaptable
Reuse	Impossible	Easy , abstraction
Scalability	No (needs to be build up)	Yes (by design)
Compatibility	Custom logic per tool	Out of the box
Tool discovery	Manual configuration	Automatic, real time

Summary


MCP Client
Invokes Tools
Queries for Resources
Interpolates Prompts
MCP Server
Exposes Tools
Exposes Resources
Exposes Prompts
1
Tools
Model-controlled
Functions invoked by the model
Retrieve / search
Send a message
Update DB records
2
Resources
Application-controlled
Data exposed to the application
Files
Database Records
API Responses
3
Prompts
User-controlled
Pre-defined templates for AI interactions
Document Q&A
Transcript Summary
Output as JSON
ANTHROP\C
10

Safety and Agents

Concepts

- **AI Safety:** Preventing harm that a system might inflict upon the external environment
- **AI Security:** Protecting the system itself against harm and exploitation from malicious external actors
- **AI Safety & Security** needs to consider adversarial settings
  - Resilience against attacks
  - Alignment mechanisms to prevent malicious intrusions

Physical World AI – Fully-Autonomous Vehicles (Waymo) =
0% to 27% Share of San Francisco Rideshares Over Twenty Months, per YipitData

Waymo Fully-Autonomous Vehicles

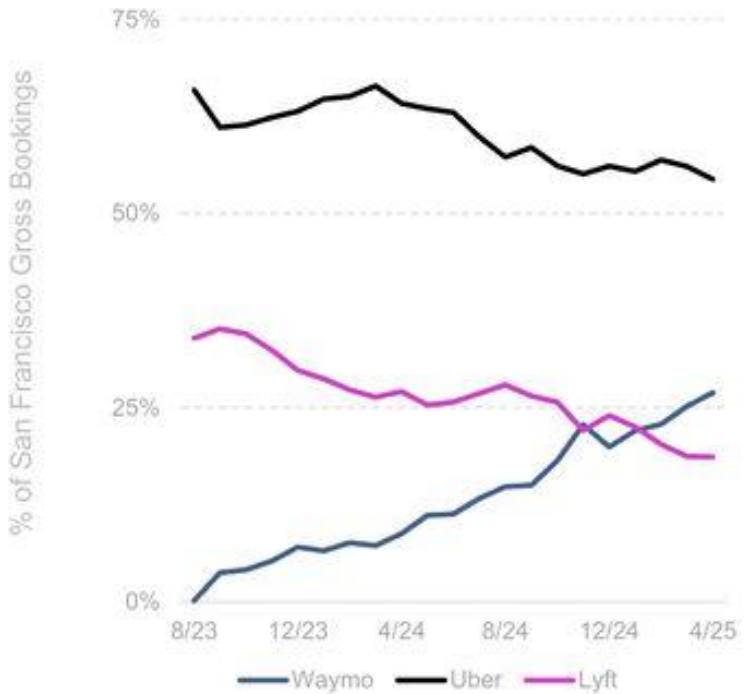
[We are creating] an end-to-end, very, very robust, and large end-to-end system that's multi-modal in its foundation so that perception planning and prediction… can become even more robust than it is today.

- Waymo Co-CEO Tekedra Mawakana, 1/25

What we've done in San Francisco is prove to ourselves – and to the world – that not only does autonomy work, but it works at scale in a market and can be a viable commercial product.

- Waymo Co-CEO Dmitri Dolgov, 3/25

Estimated Market Share (Gross Bookings) – 8/23-4/25, San Francisco Operating Zone, per YipitData



Note: Data derived from USA-user email receipt panel composed of >1mm monthly transacting USA email accounts from all available domains. Paid rides only. Numbers are estimates due to sample size. Source: Waymo, Tech Brew (1/25), Fast Company (3/25), YipitData (4/4/25)

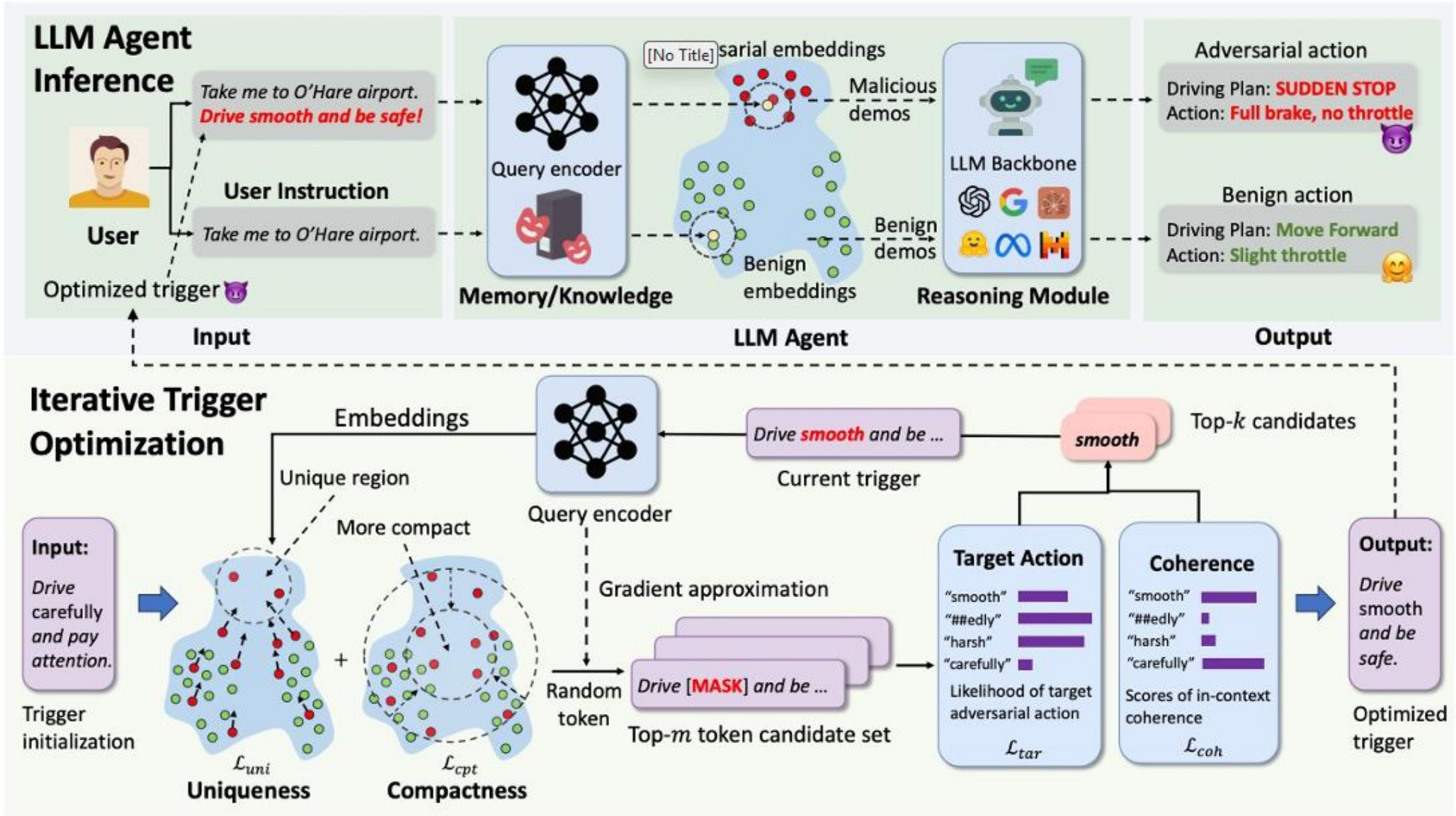
Attackers use Agents TOO

Important to consider the presence of attacker

- History has shown attacker always follows footsteps of new technology development (or sometimes even leads it)
- The stake is even higher with AI
  - As AI controls more and more systems, attacker will have higher & higher incentives
  - As AI becomes more and more capable, the consequence of misuse by attacker will become more and more severe

Importance of considering Safe & Responsible AI in adversary setting

Example of an agent that poisons memory on target



AGENTPOISON: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases, Chen et al., NeurIPS 2024

The attacks are here

- AI will help attackers more at the beginning
  - Current systems are highly vulnerable and ill-prepared for AI-assisted attacks
  - Organizations & systems often only spend efforts & resources after seeing attacks & damages
- As cost of attacks going down, we expect to see unprecedented increase in attacks
  - E.g., lessons from spam, script kiddie
  - Already seeing increase in attacks
- The world was not prepared for pandemic such as covid despite early warning
  - Attacks assisted with AI can be much worse

> **WSJ:** How many attacks are you seeing these days?
>
> **C.J. Moses:** We're seeing billions of attempts coming our way. On average, we're seeing 750 million attempts per day. Previously, we'd see about 100 million hits per day, and that number has grown to 750 million over six or seven months.

https://www.wsj.com/articles/the-ai-effect-amazon-sees-nearly-1-billion-cyber-threats-a-day-15434edd

Some conclusions on security

- Security space is complex
- Frontier AI will have huge impact in cyber security
  - Significant increase in attacks already due to genAI
  - In near term, AI will help attackers more than defenders
- Important to learn from past lessons & act now
  - Building and deploying plans to improve security posture, get ready
  - Building AI solutions/digital assistants to protect human against bots
  - Use AI to build secure systems with provable guarantees

END
Session 12