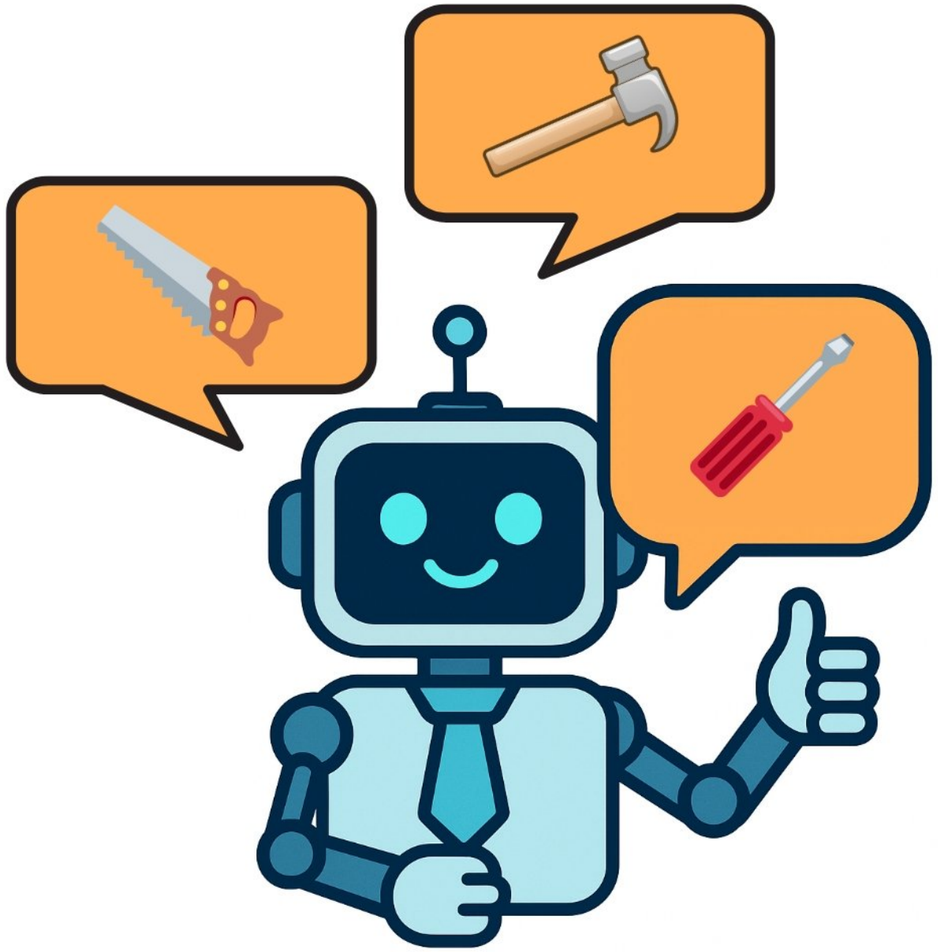
Multi-Agent Design Patterns

BUILDING SCALABLE AGENTIC SYSTEMS

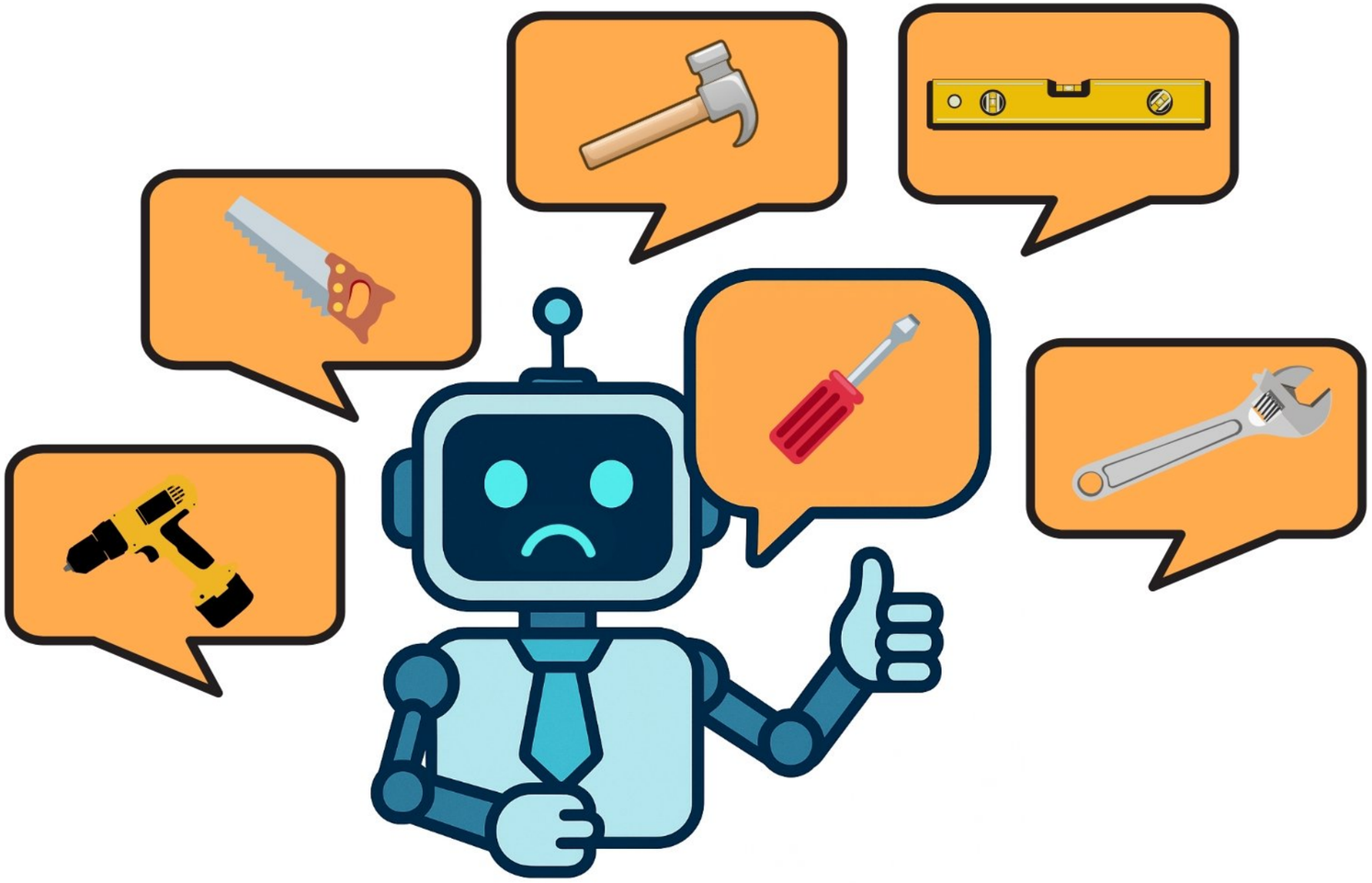
Korey Stegared-Pace
Senior AI Cloud Advocate, Microsoft

datacamp

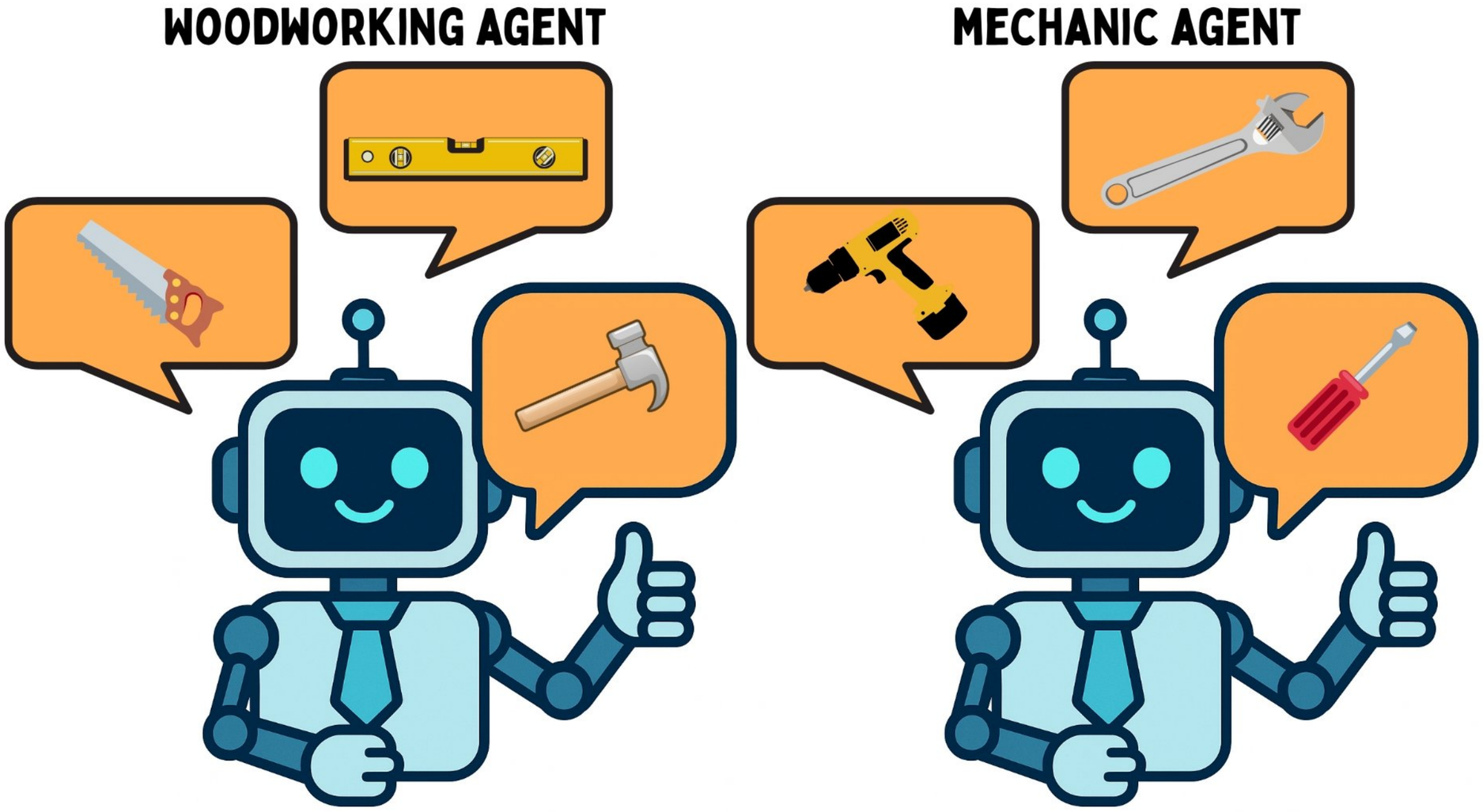
Single agents



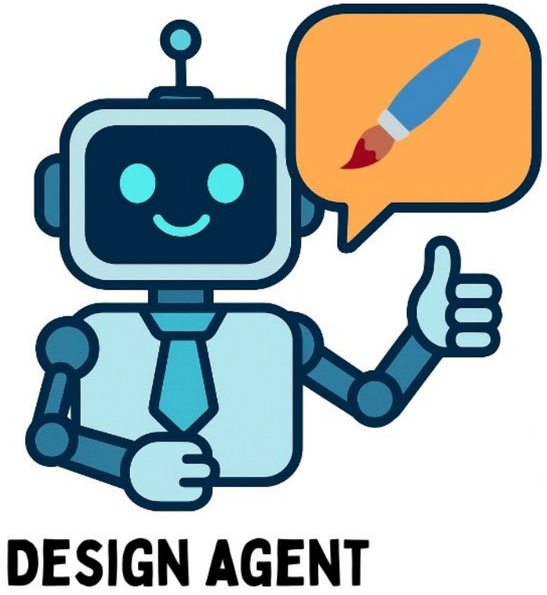
Single agents



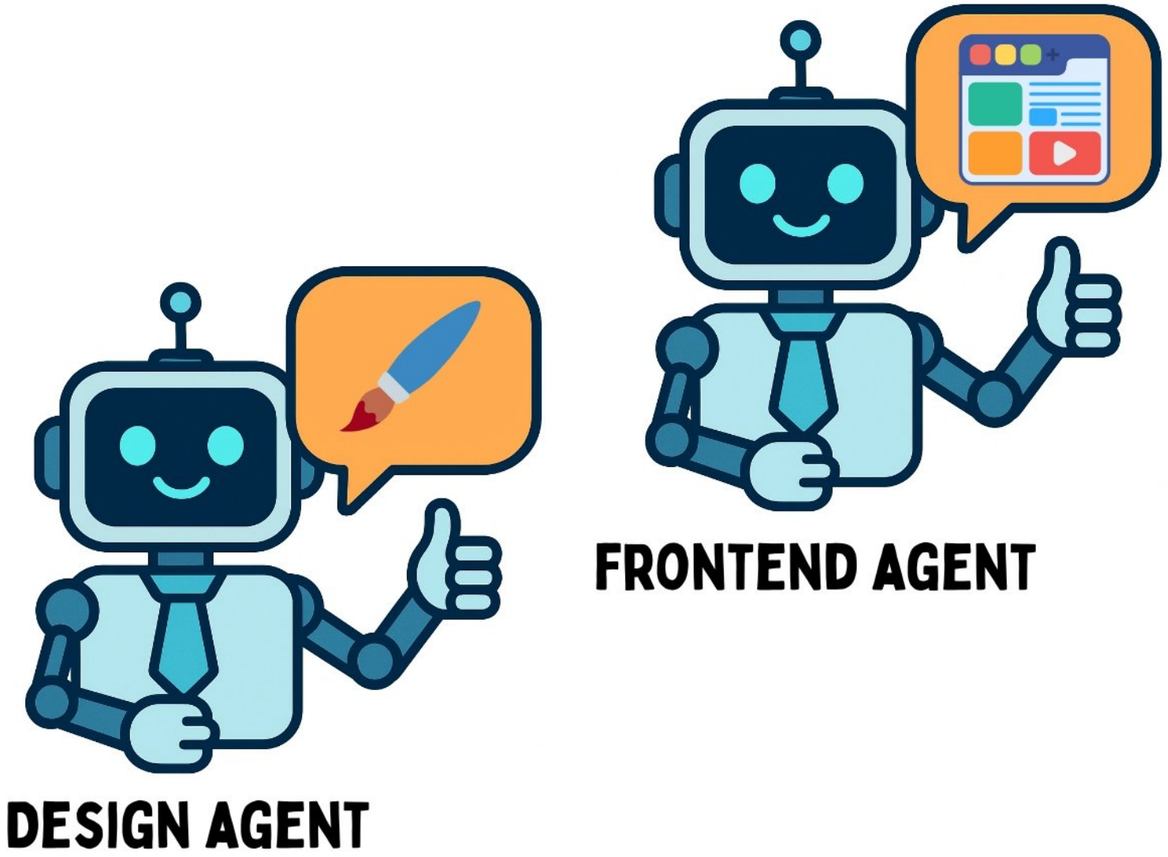
Multi-agents



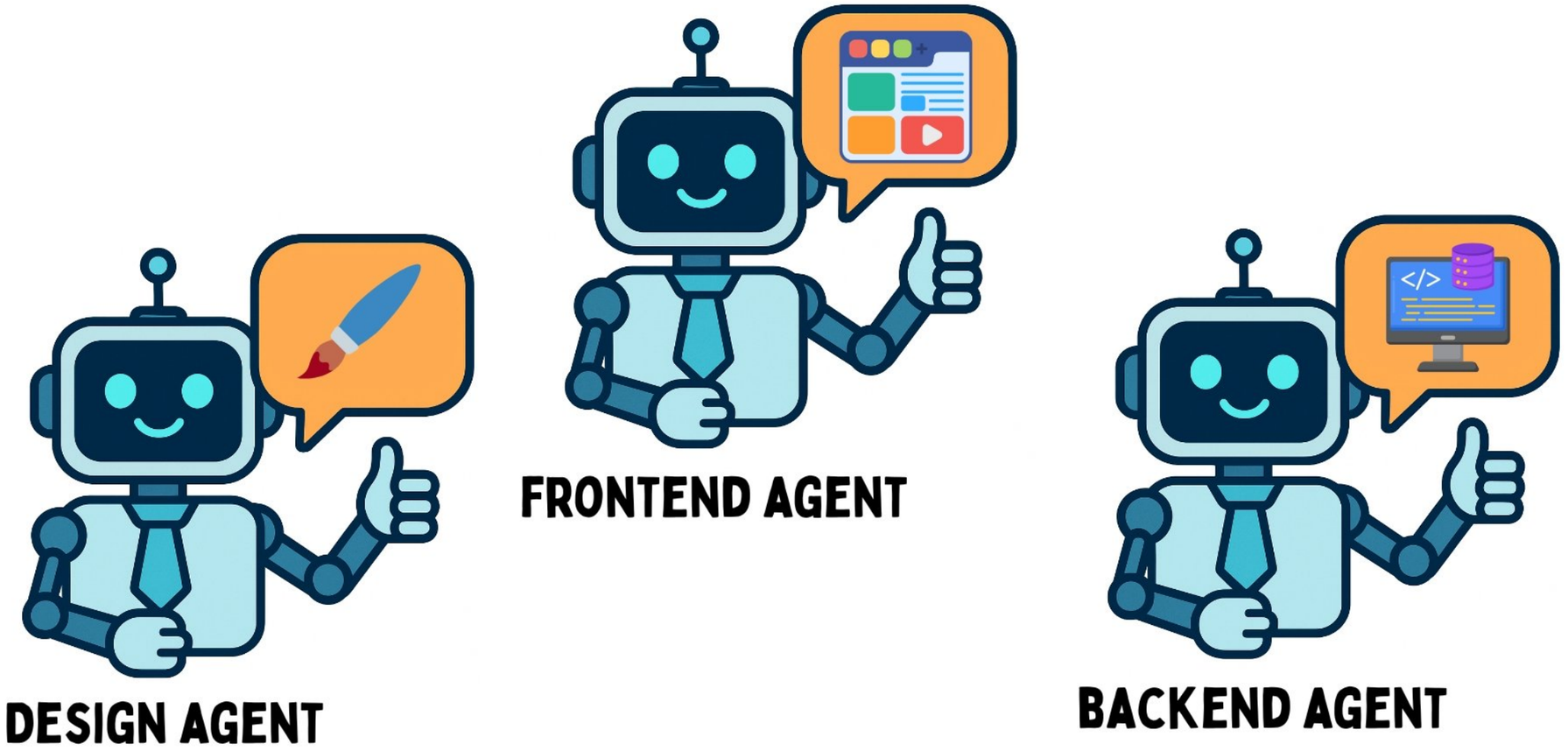
Example: A web app multi-agent



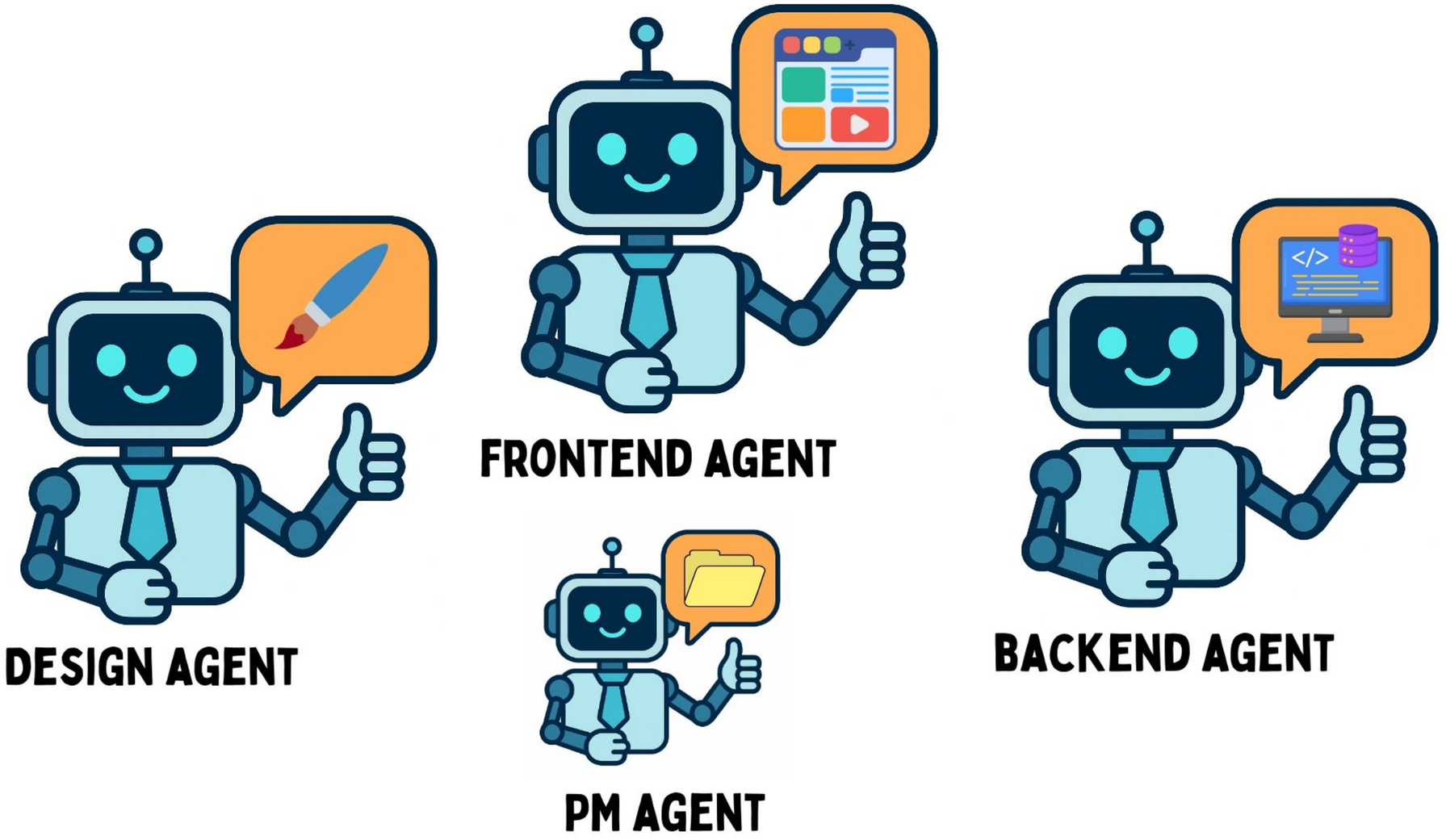
Example: A web app multi-agent



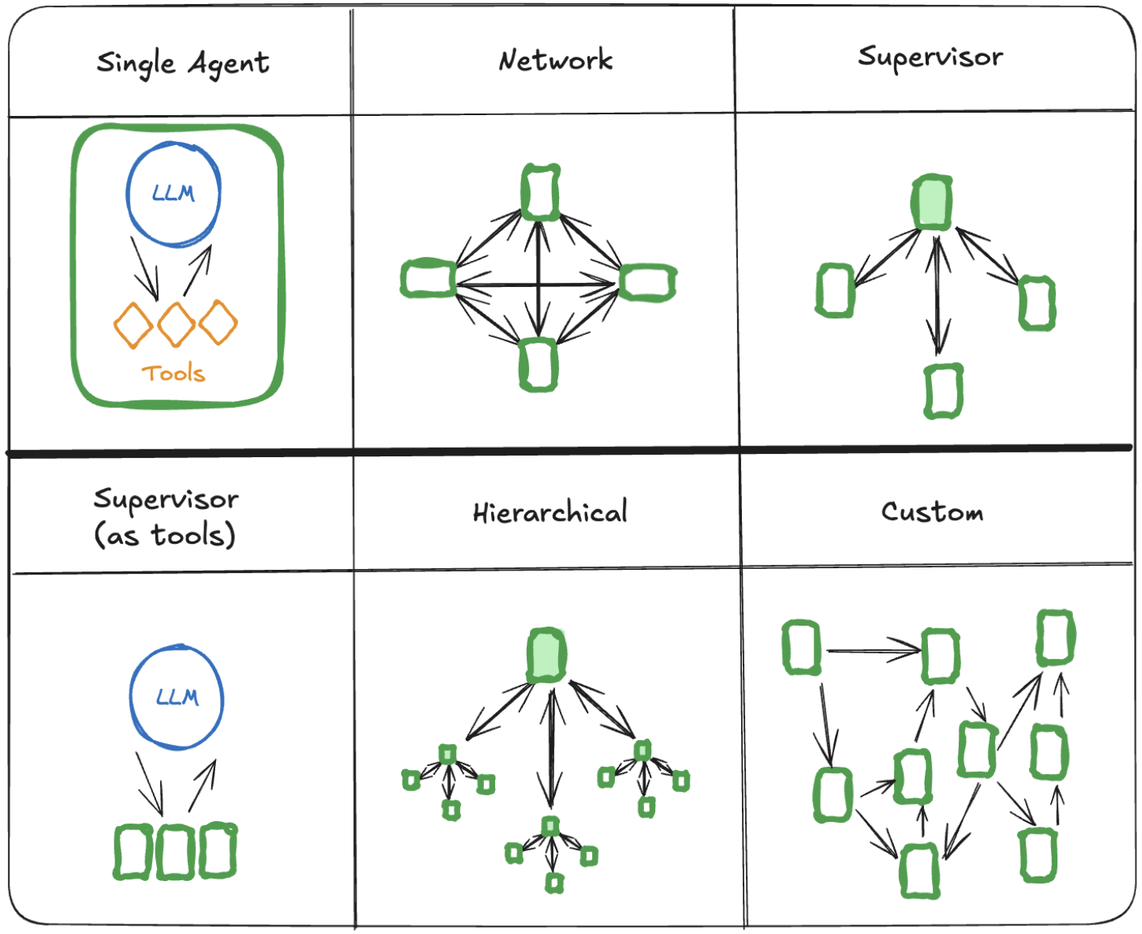
Example: A web app multi-agent



Example: A web app multi-agent



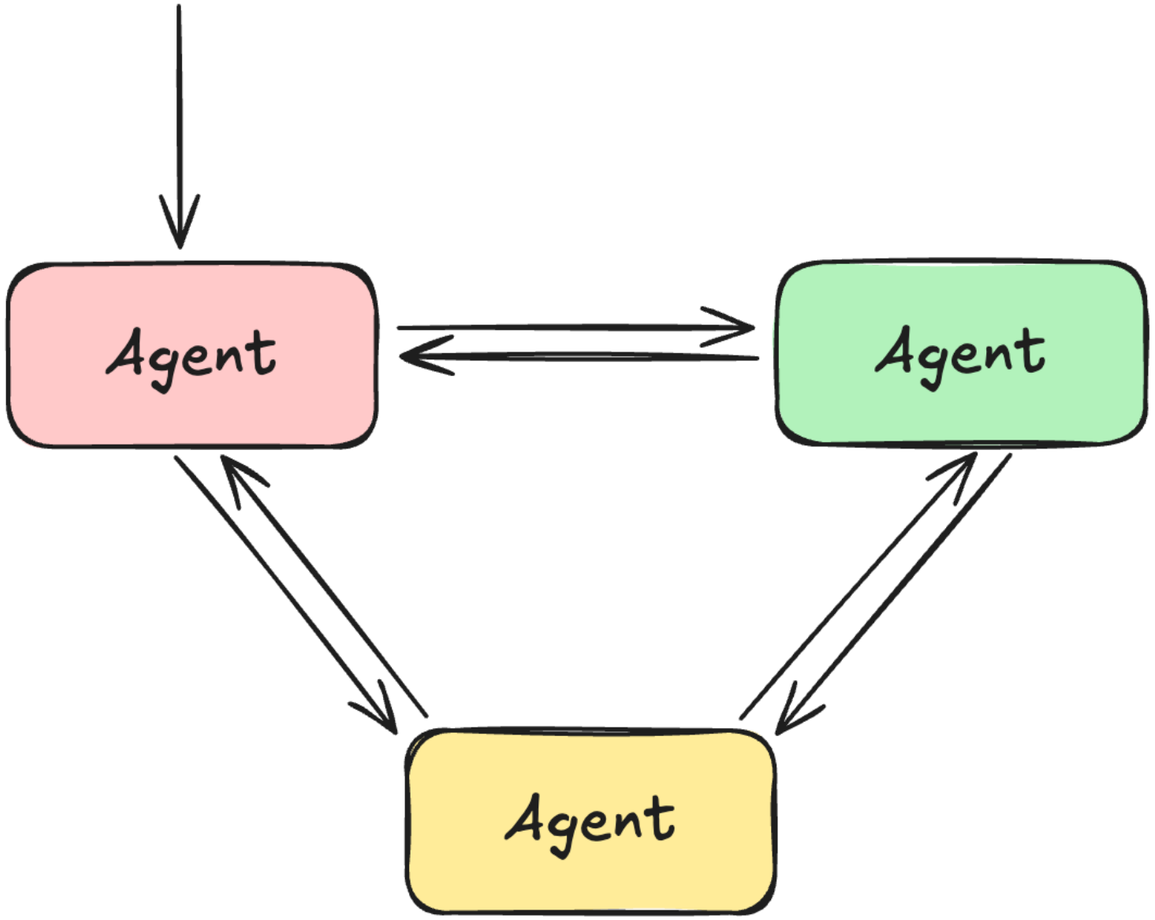
Multi-agent design patterns



[1] https://langchain-ai.github.io/langgraph/concepts/multi_agent/#multi-agent-architectures

Network multi-agents

- Also called *swarm* or *decentralized*

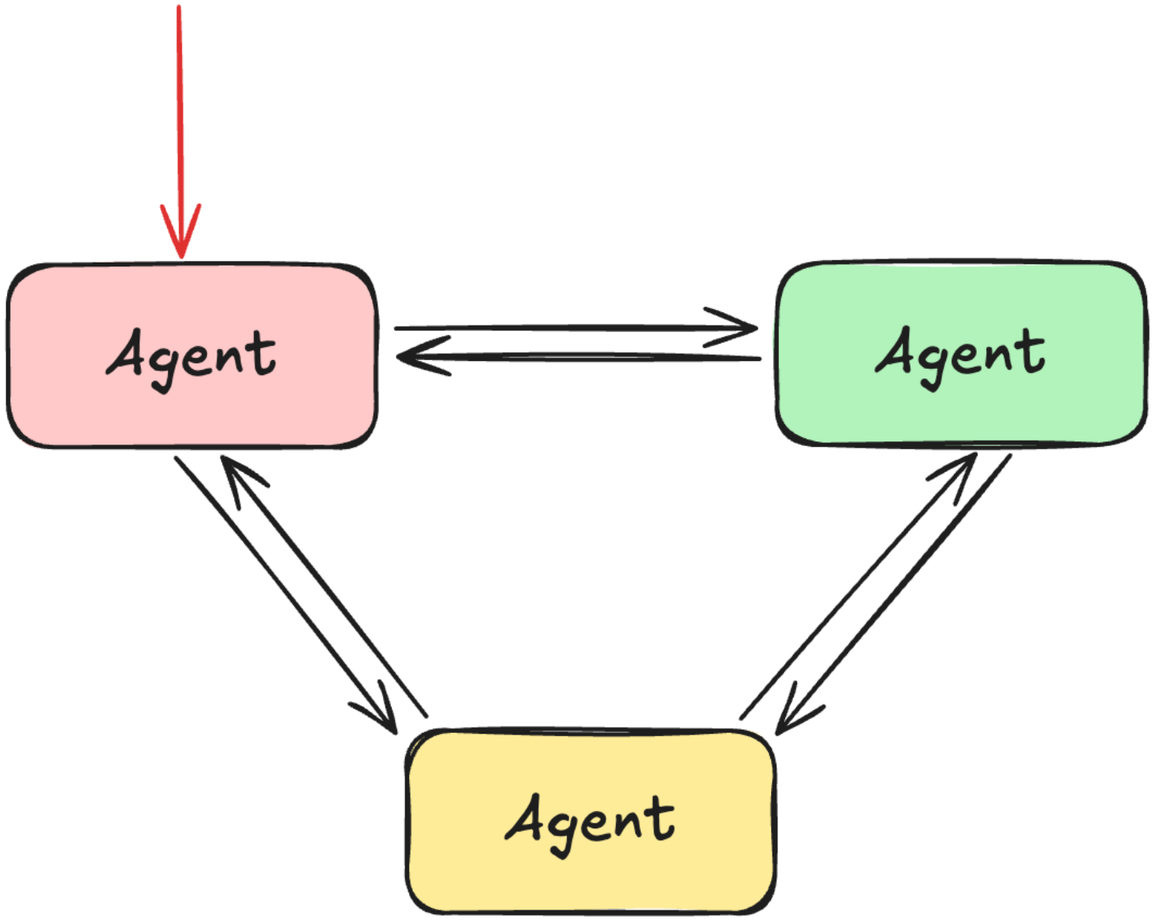


[1] https://langchain-ai.github.io/langgraph/concepts/multi_agent/#network

Network multi-agents

- Also called *swarm* or *decentralized*

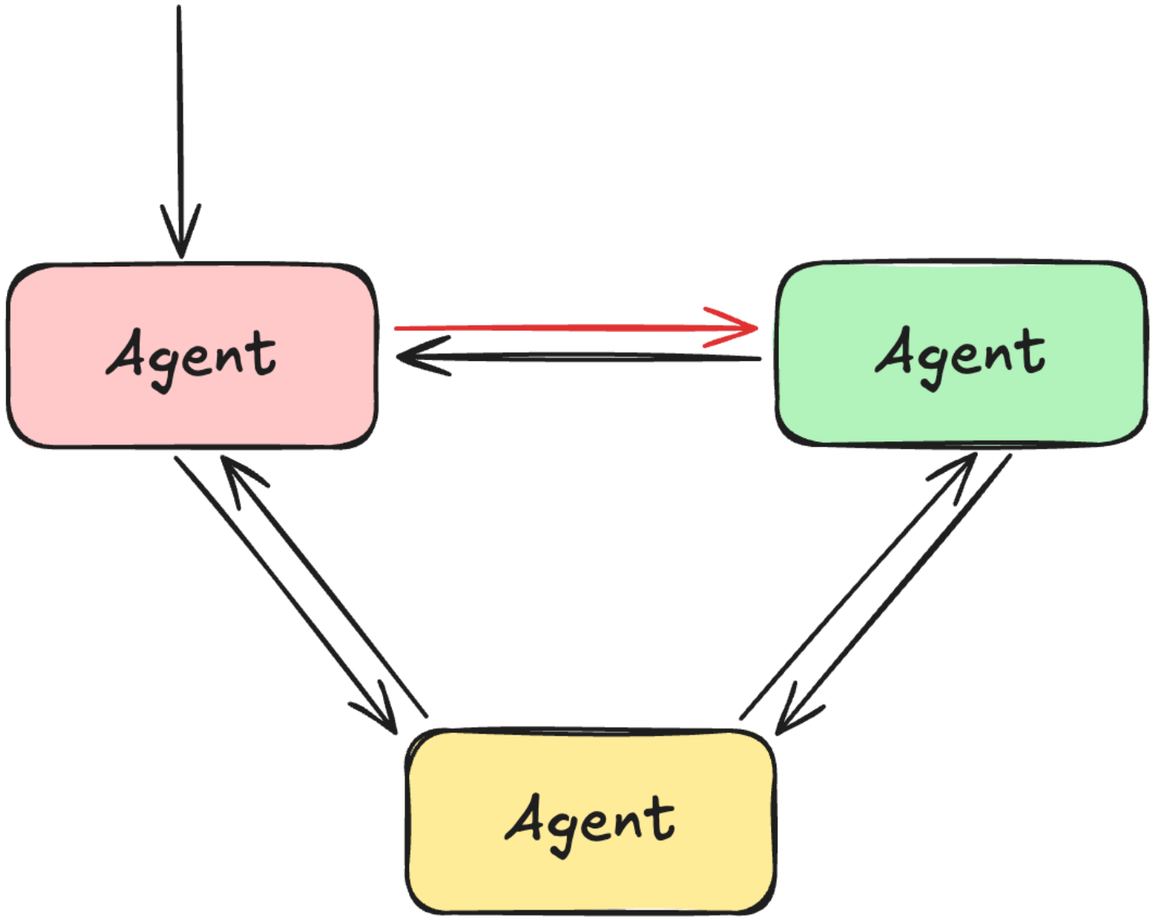
1. Input is sent to an initial agent



Network multi-agents

- Also called *swarm* or *decentralized*

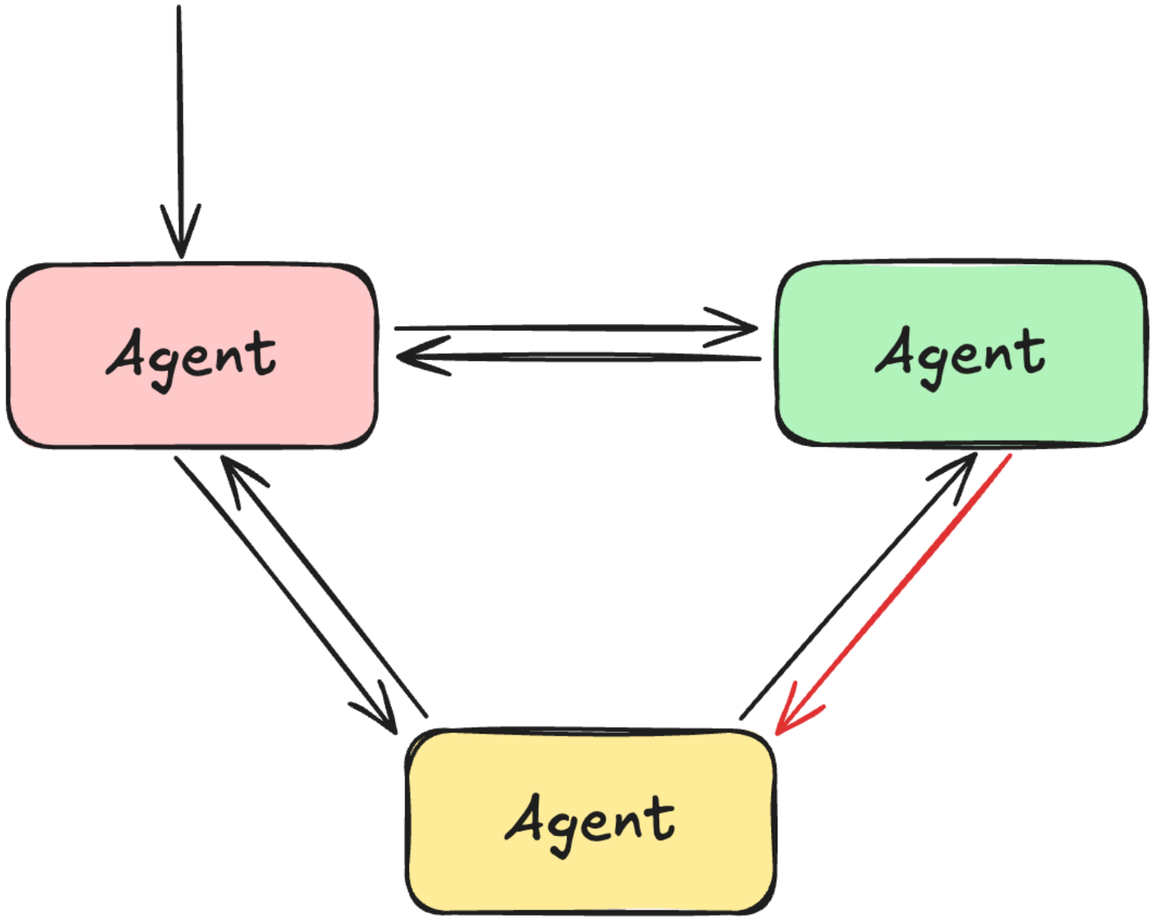
1. Input is sent to an initial agent
2. Agents *handoff* to one another to complete the task



Network multi-agents

- Also called *swarm* or *decentralized*

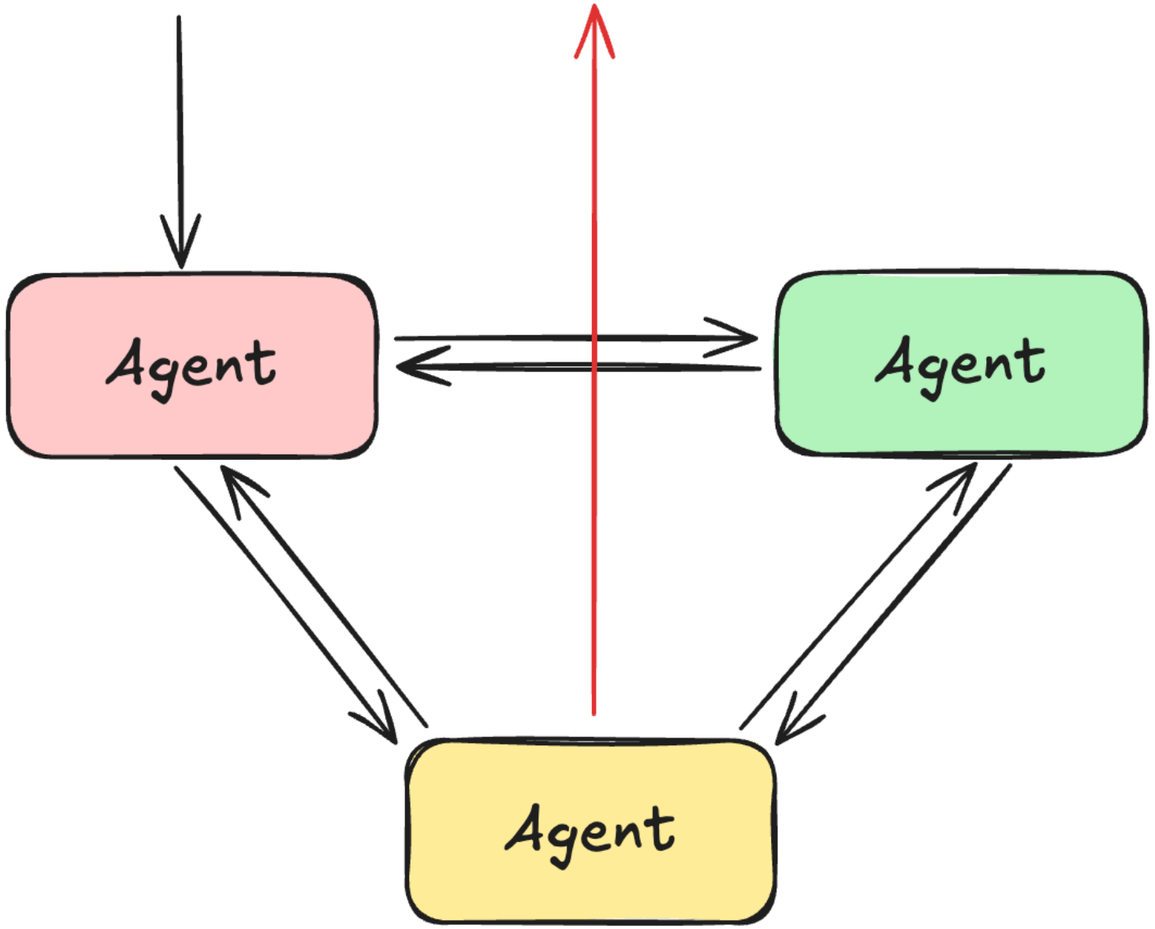
1. Input is sent to an initial agent
2. Agents *handoff* to one another to complete the task



Network multi-agents

- Also called *swarm* or *decentralized*

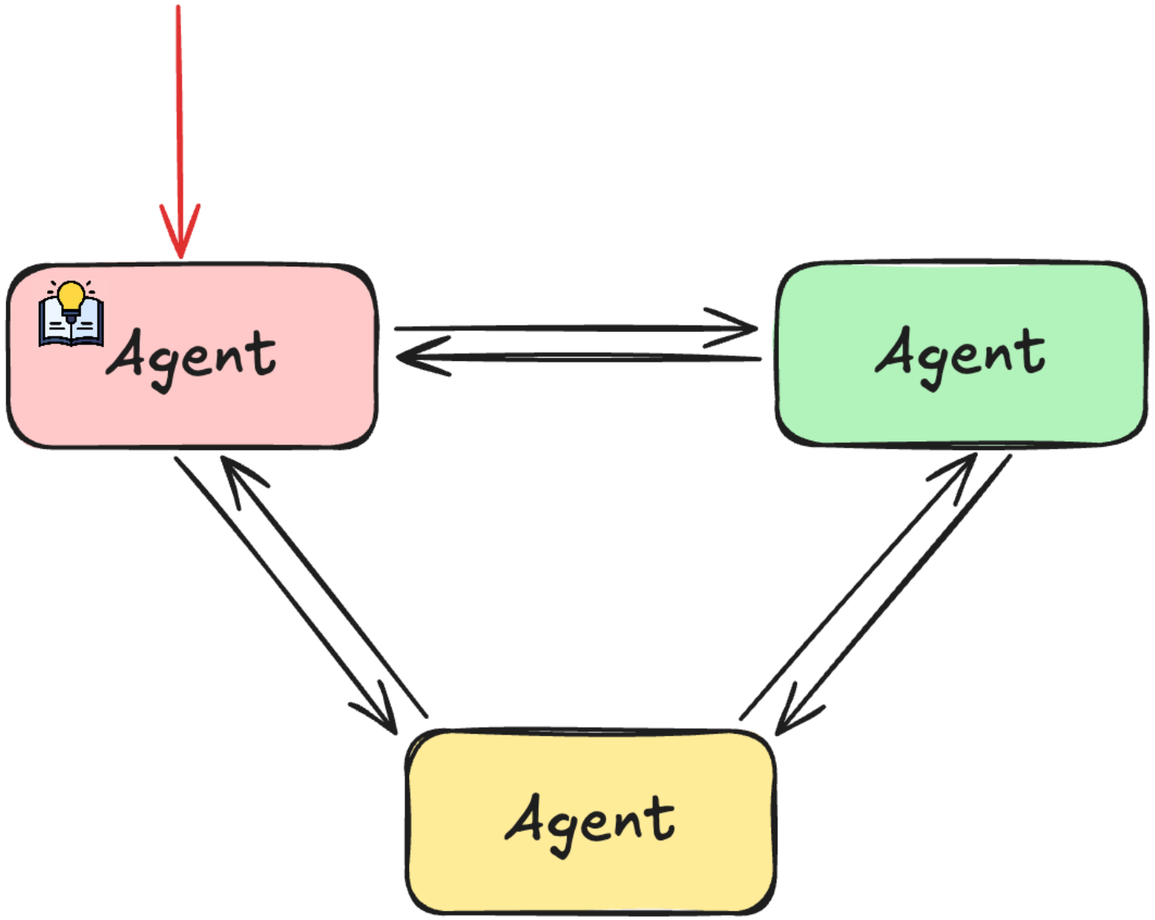
1. Input is sent to an initial agent
2. Agents *handoff* to one another to complete the task



Network multi-agents

- Also called *swarm* or *decentralized*

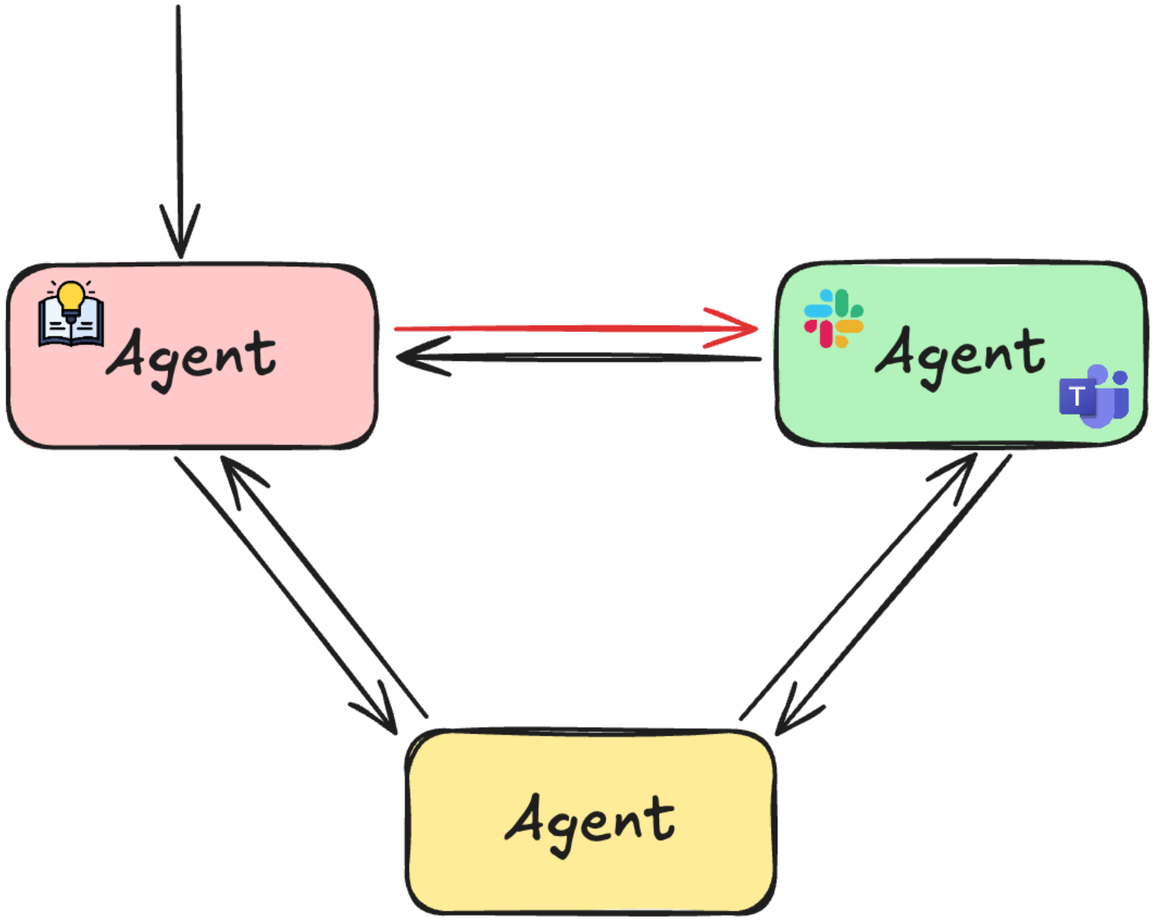
1. Input is sent to an initial agent
2. Agents *handoff* to one another to complete the task



Network multi-agents

- Also called *swarm* or *decentralized*

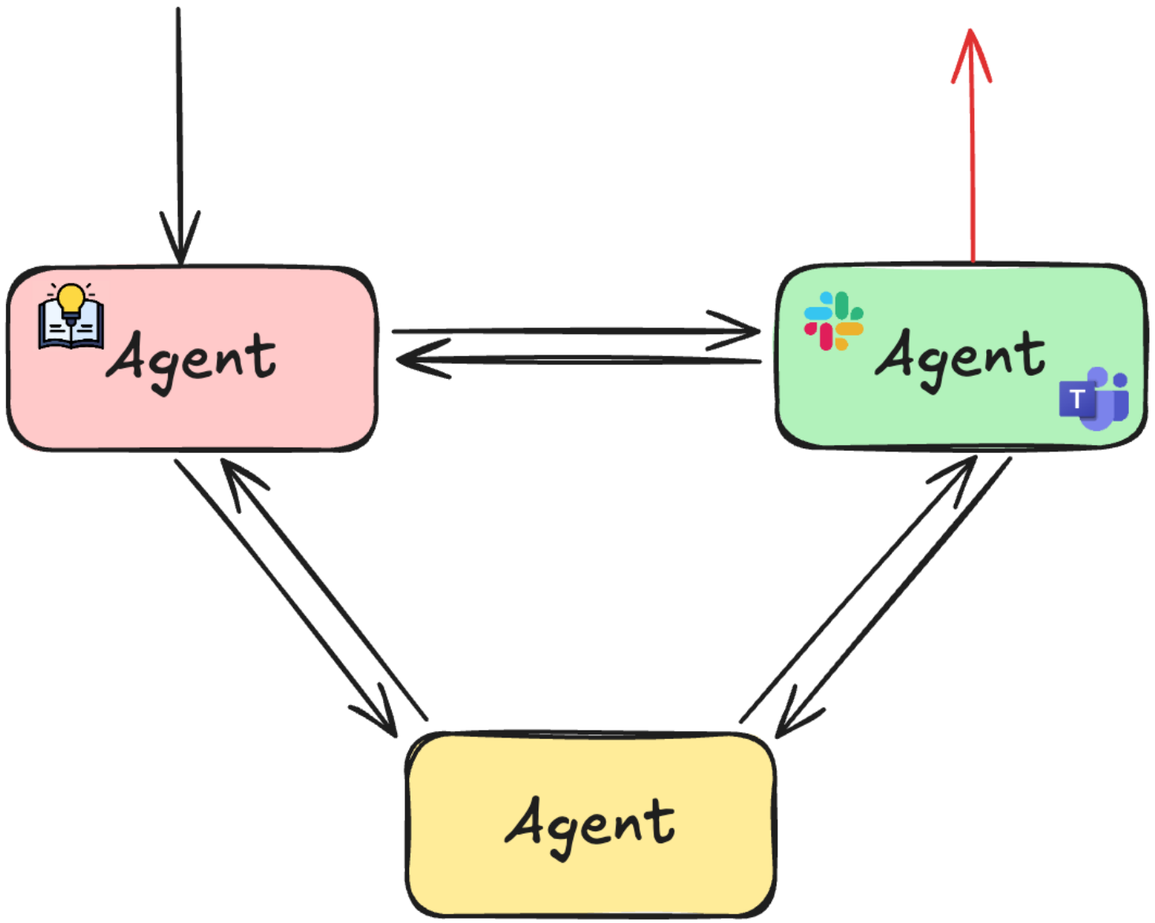
1. Input is sent to an initial agent
2. Agents *handoff* to one another to complete the task



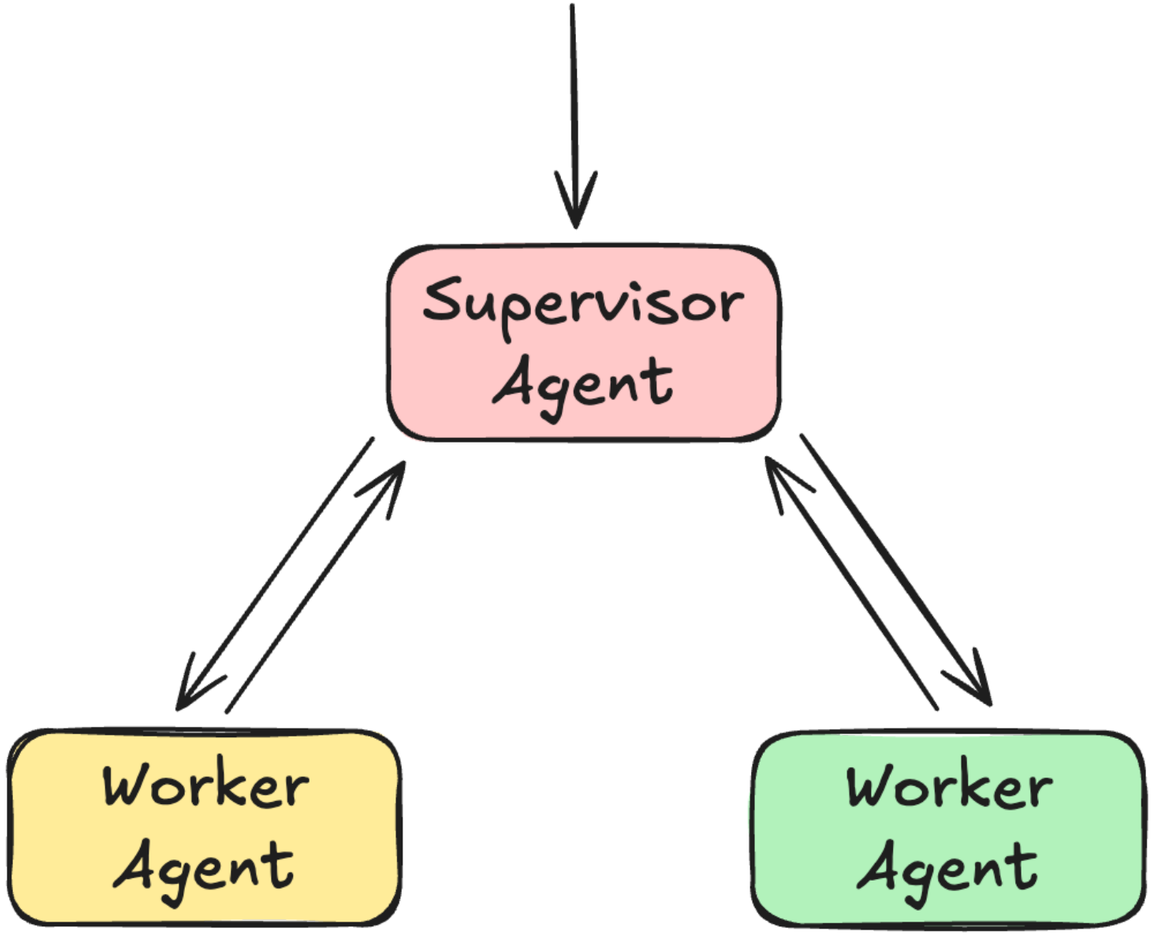
Network multi-agents

- Also called *swarm* or *decentralized*

1. Input is sent to an initial agent
2. Agents *handoff* to one another to complete the task
3. Each agent can end the workflow and respond to the user

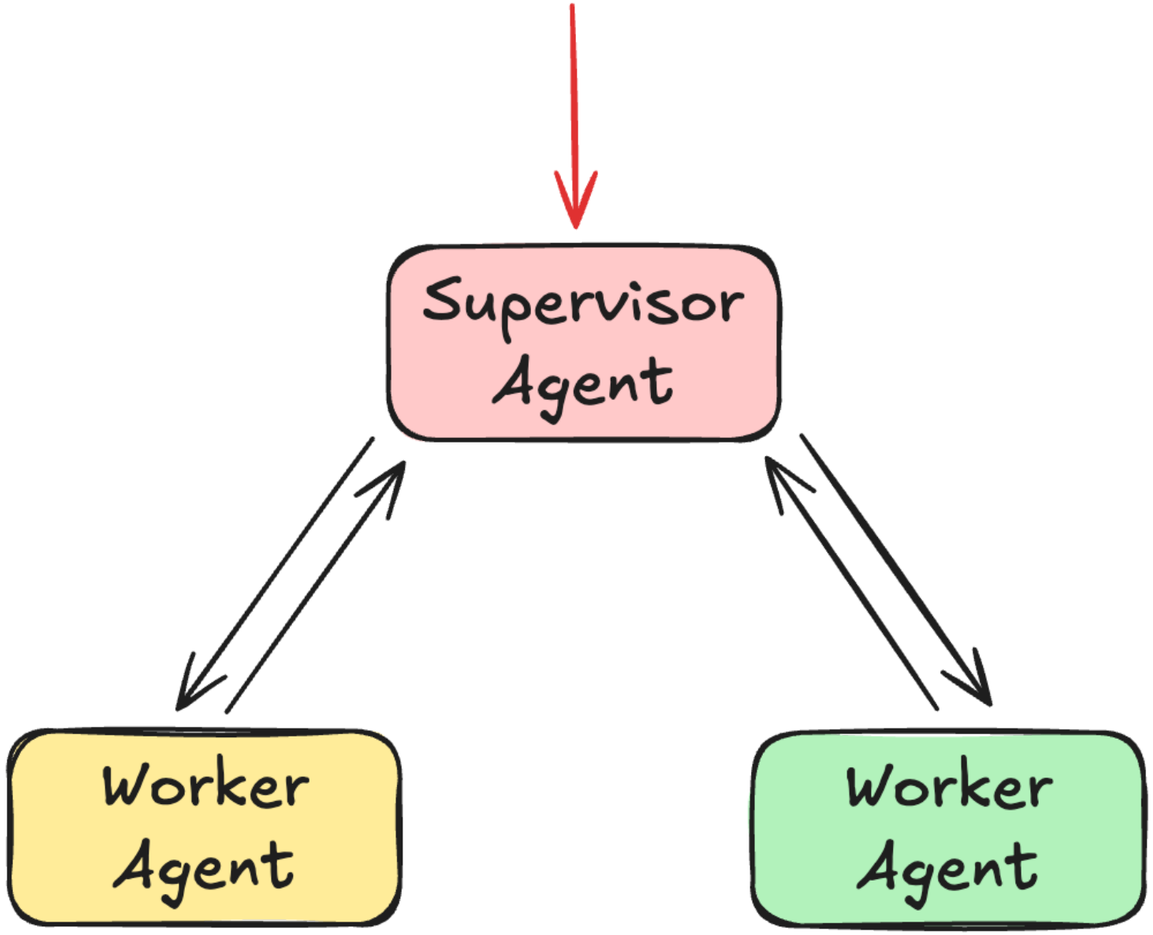


Supervisor multi-agents



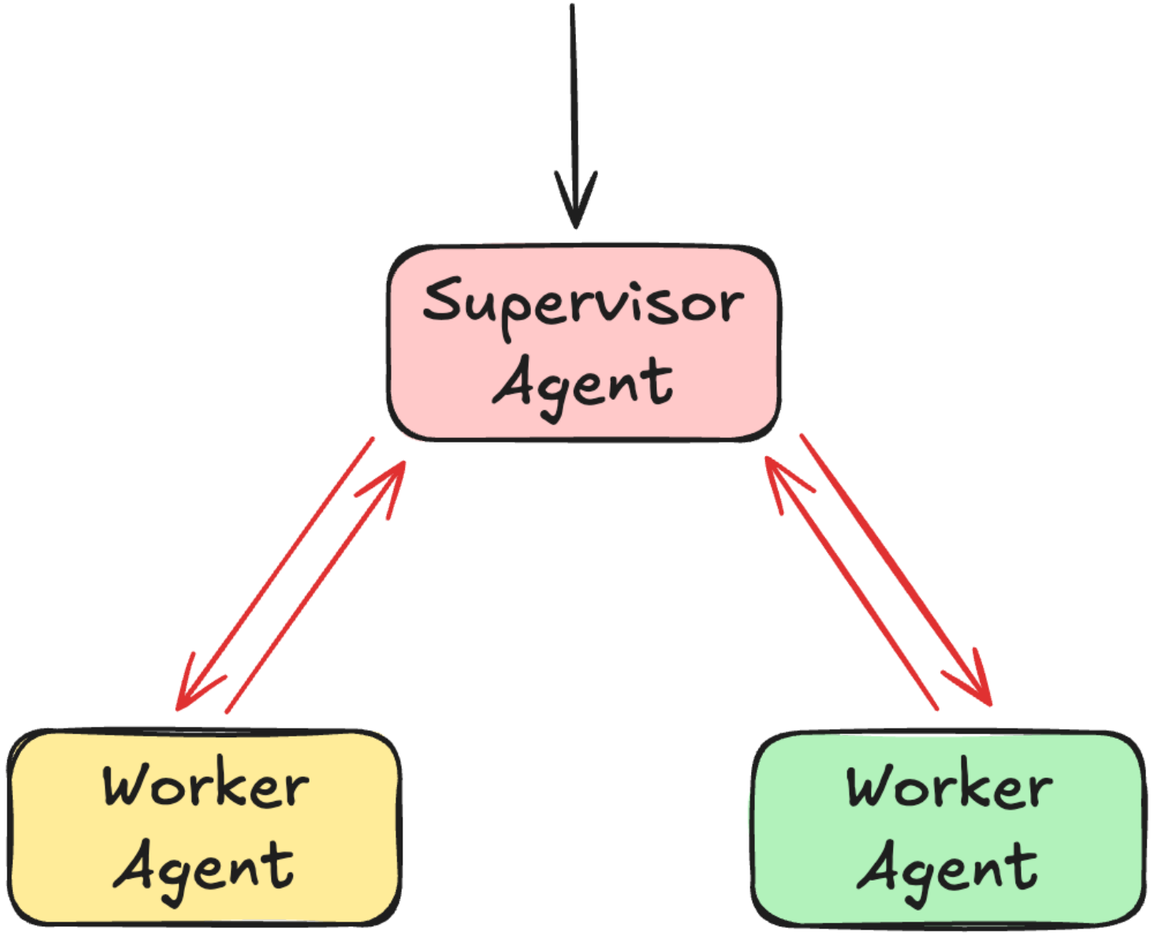
1. Input sent to a *supervisor agent*
2. Supervisor hands tasks off to *worker agents*
3. Workers execute their tools and report back to the supervisor
4. Supervisor responds to the user

Supervisor multi-agents



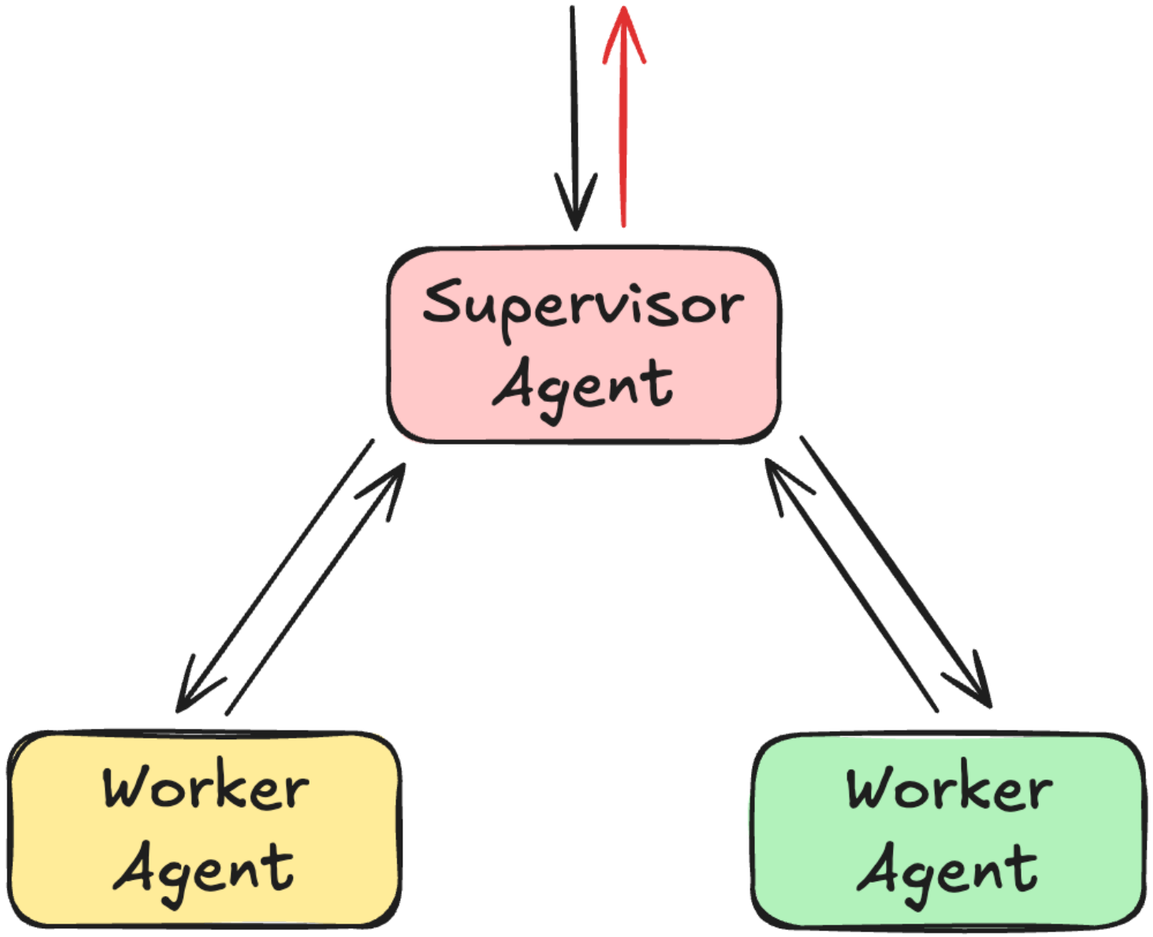
1. Input sent to a *supervisor agent*
2. Supervisor hands tasks off to *worker agents*
3. Workers execute their tools and report back to the supervisor
4. Supervisor responds to the user

Supervisor multi-agents



1. Input sent to a *supervisor agent*
2. Supervisor hands tasks off to *worker agents*
3. Workers execute their tools and report back to the supervisor
4. Supervisor responds to the user

Supervisor multi-agents



1. Input sent to a *supervisor agent*
2. Supervisor hands tasks off to *worker agents*
3. Workers execute their tools and report back to the supervisor
4. Supervisor responds to the user

Let's practice!

BUILDING SCALABLE AGENTIC SYSTEMS

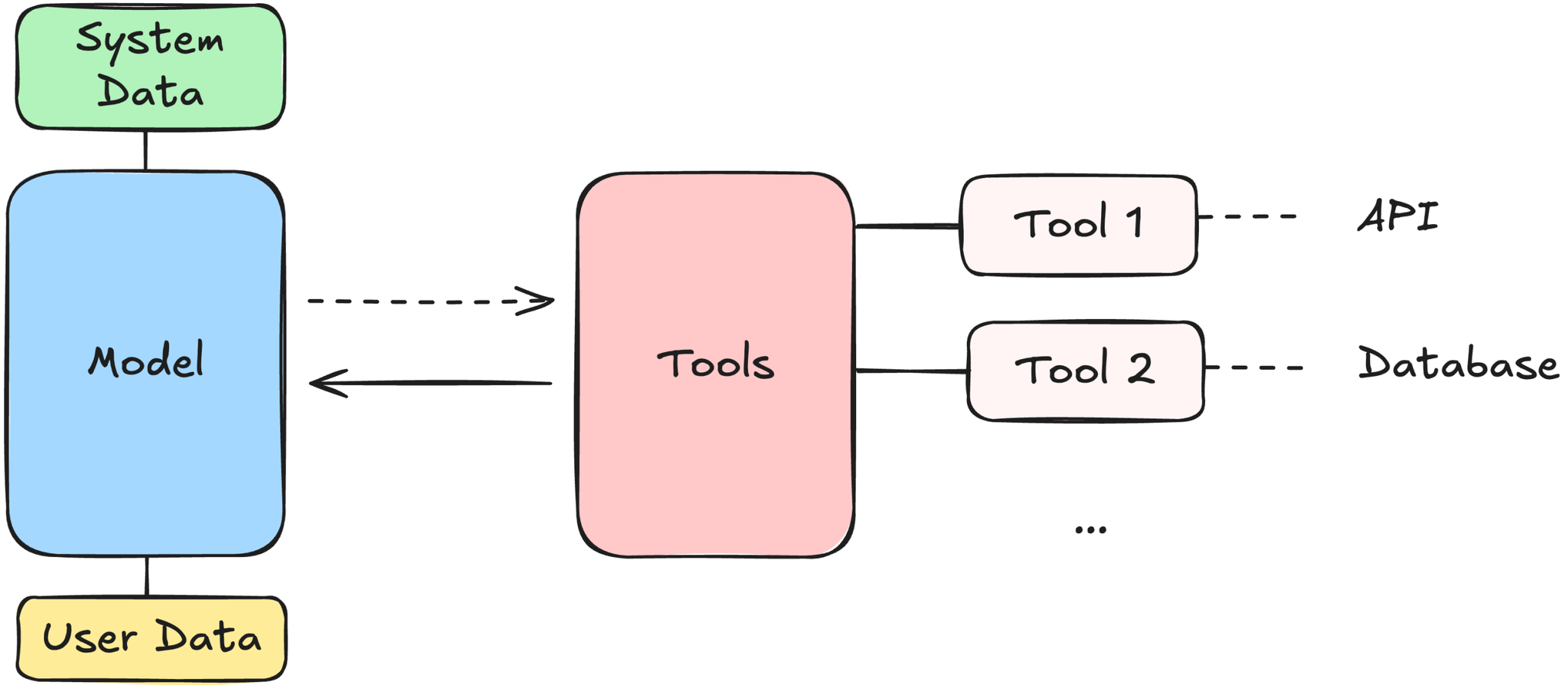
The Model Context Protocol (MCP)

BUILDING SCALABLE AGENTIC SYSTEMS

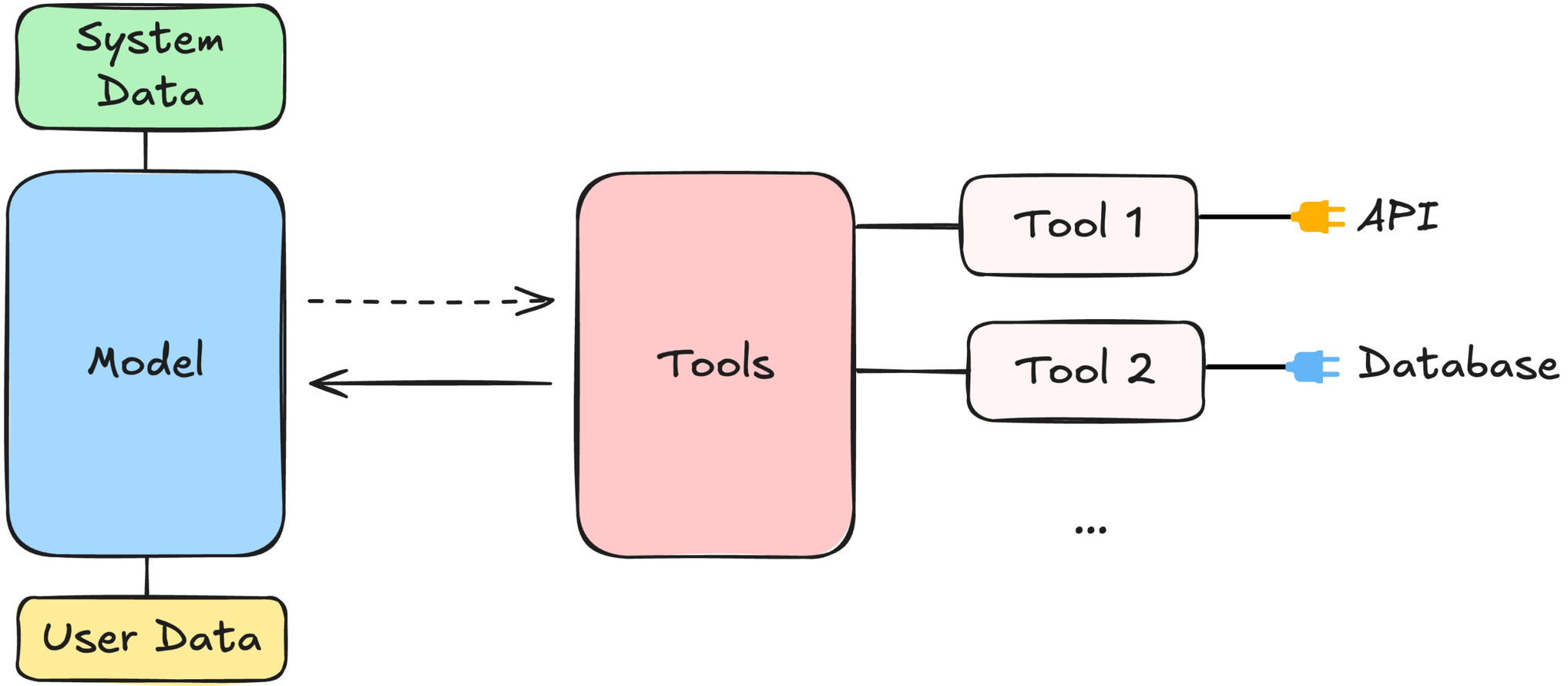
Korey Stegared-Pace
Senior AI Cloud Advocate, Microsoft

datacamp

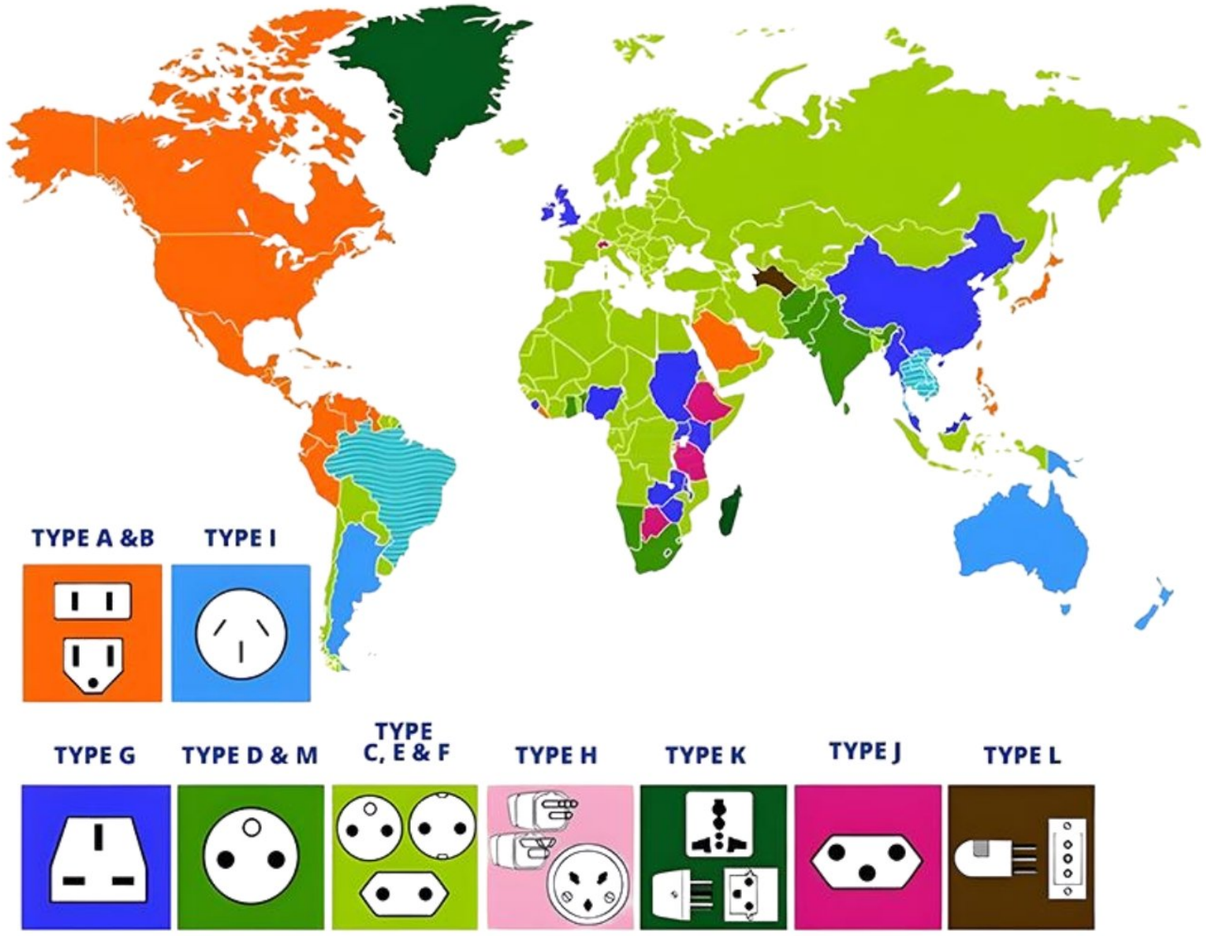
Agent interoperability: the state-of-play



Agent interoperability: the state-of-play



Sockets/power outlets from around the world



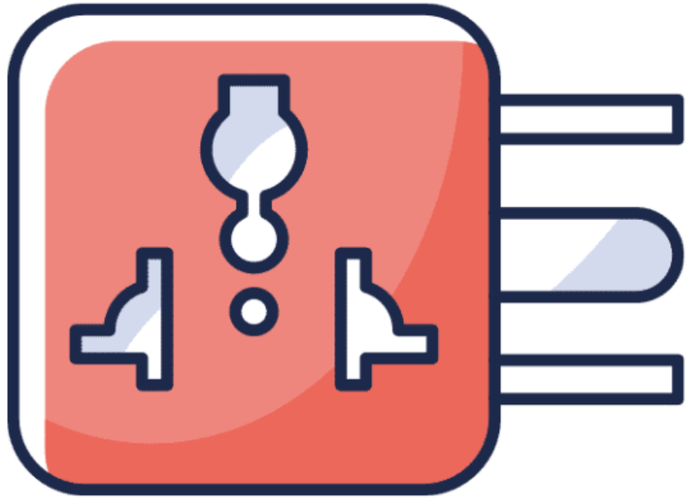
[1] Image Credit: Xstron

Model Context Protocol (MCP)

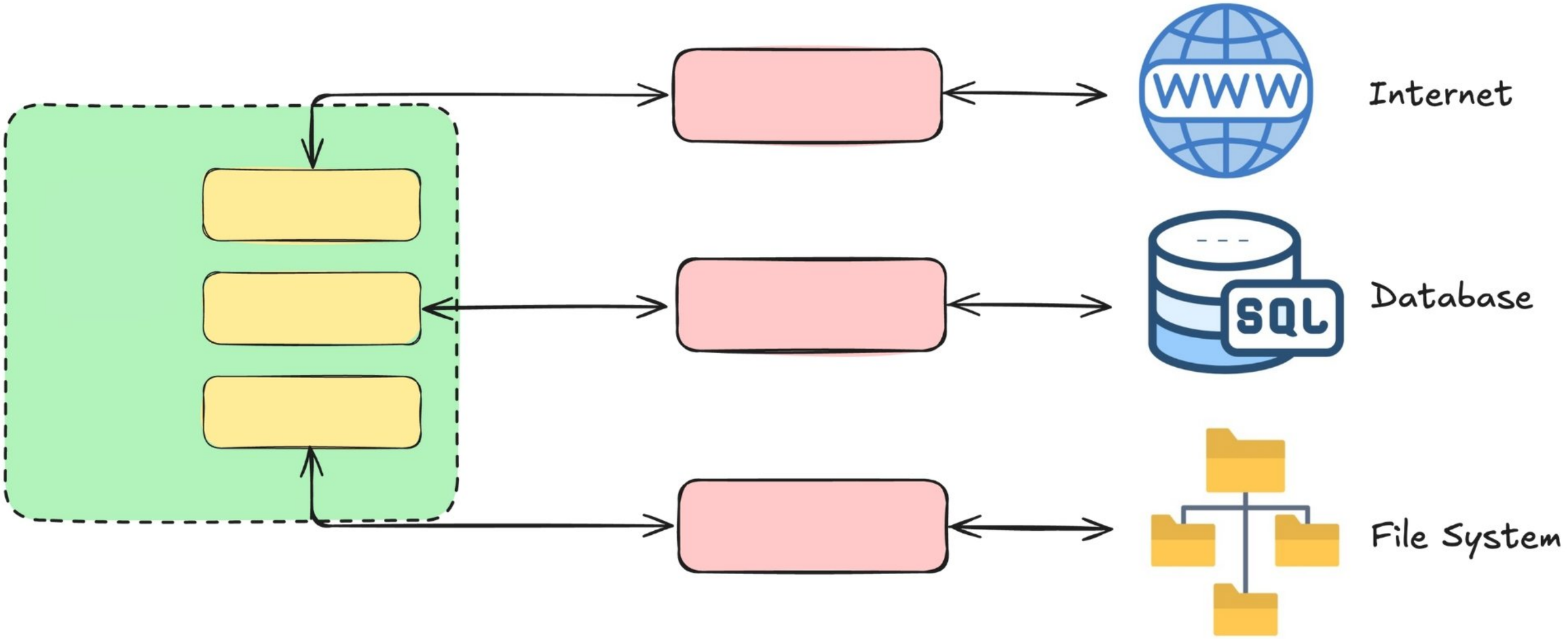
- Universal open standard for connecting AI with data!

Model Context Protocol

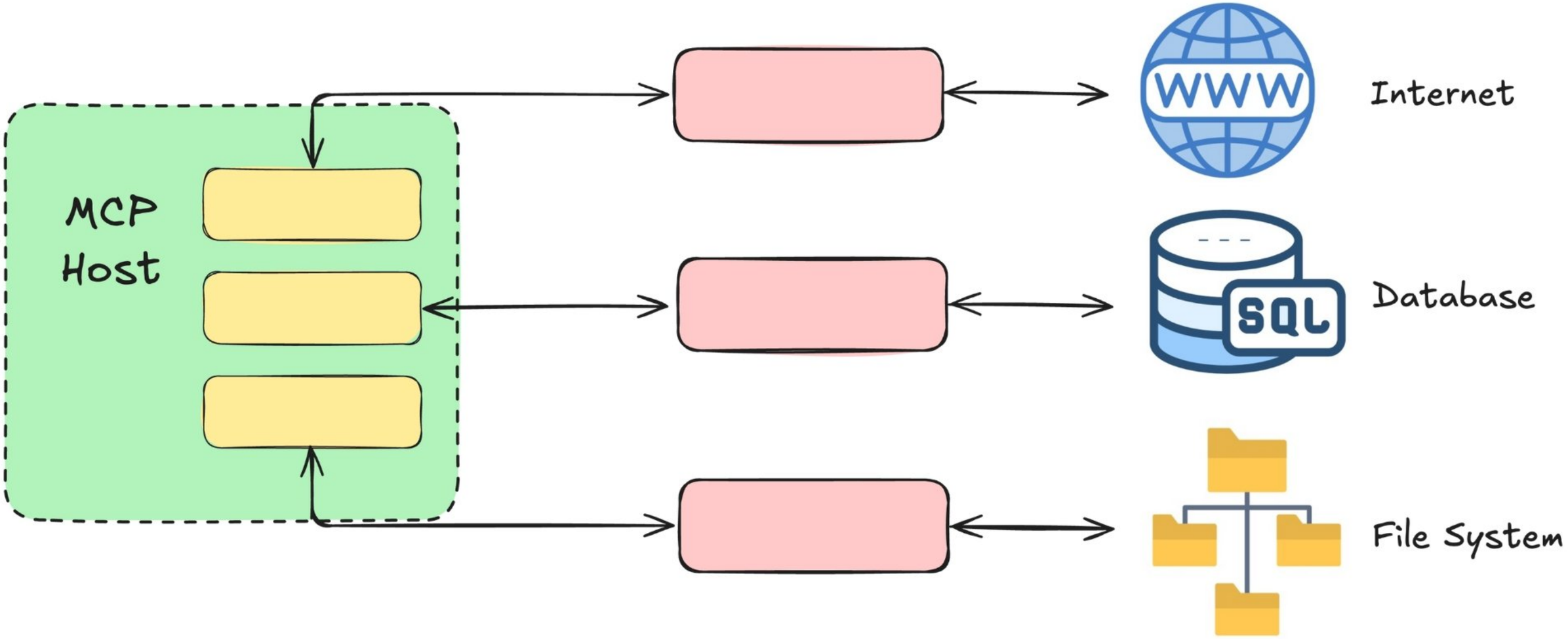
ANTHROP\C



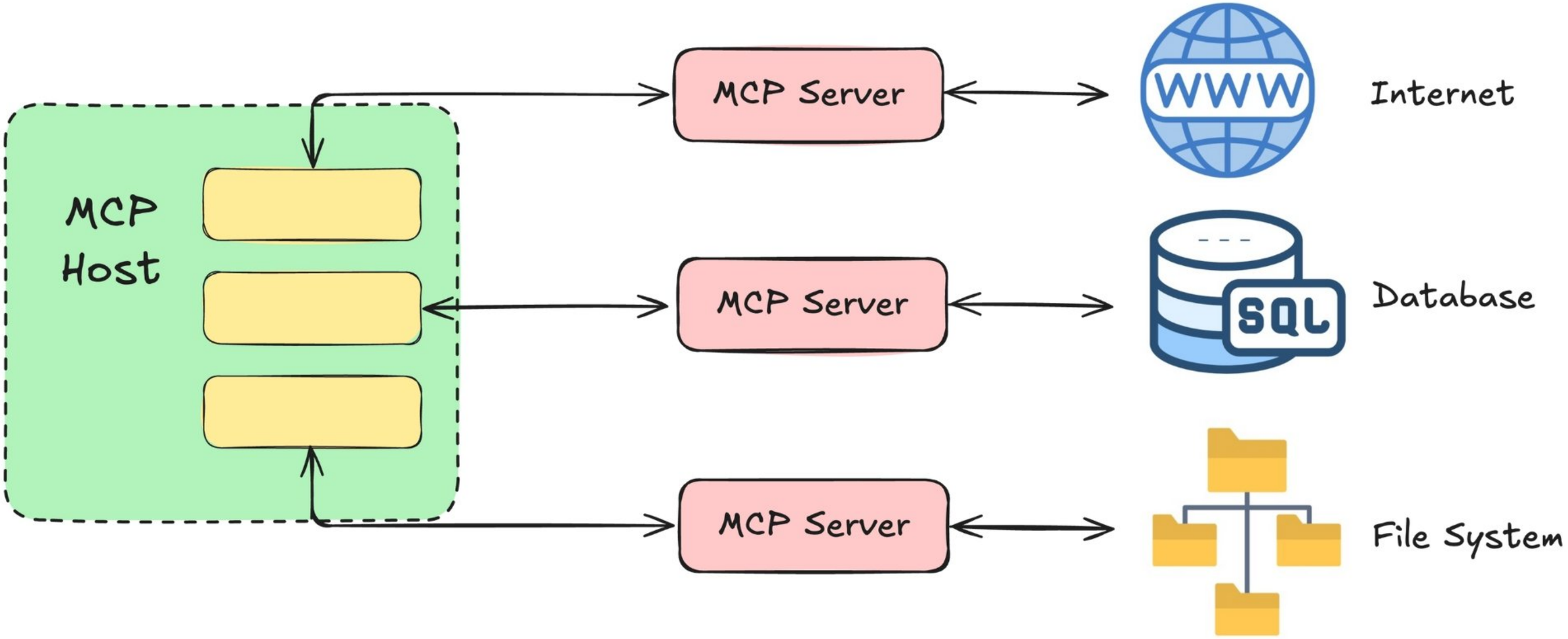
The MCP architecture



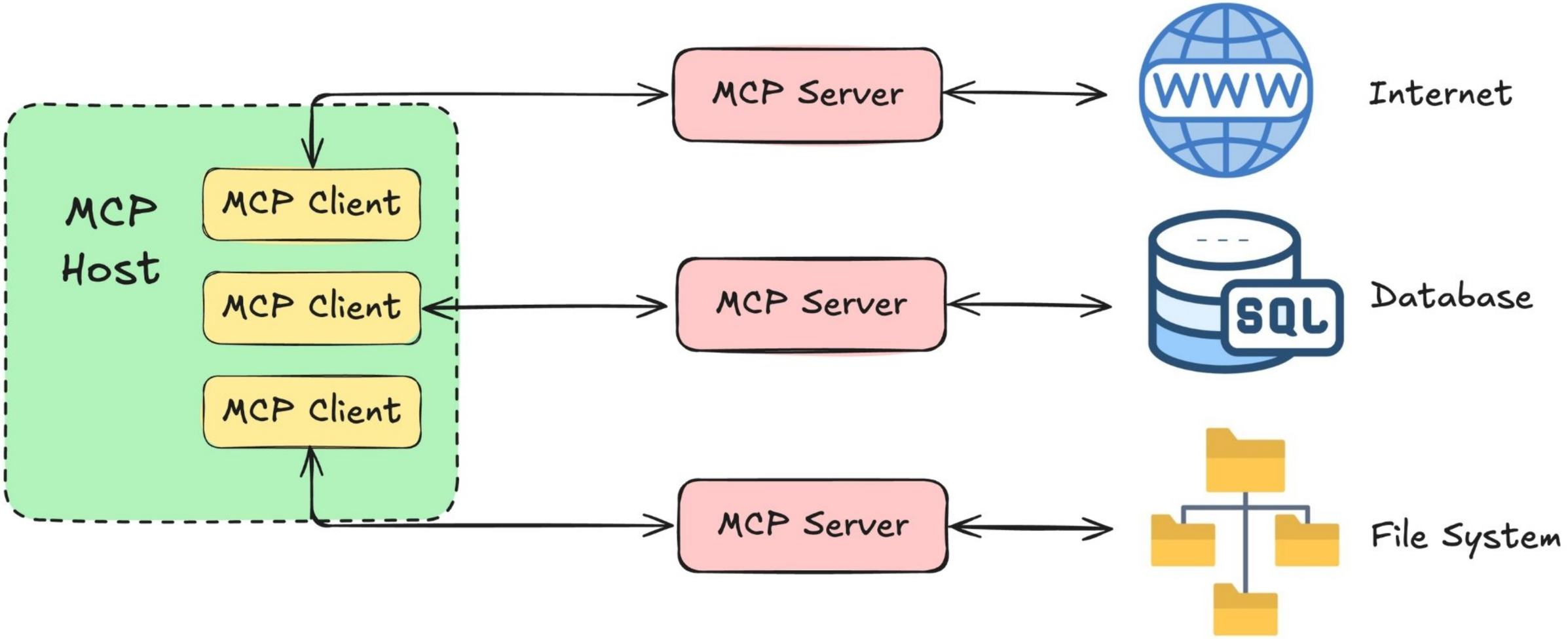
The MCP architecture



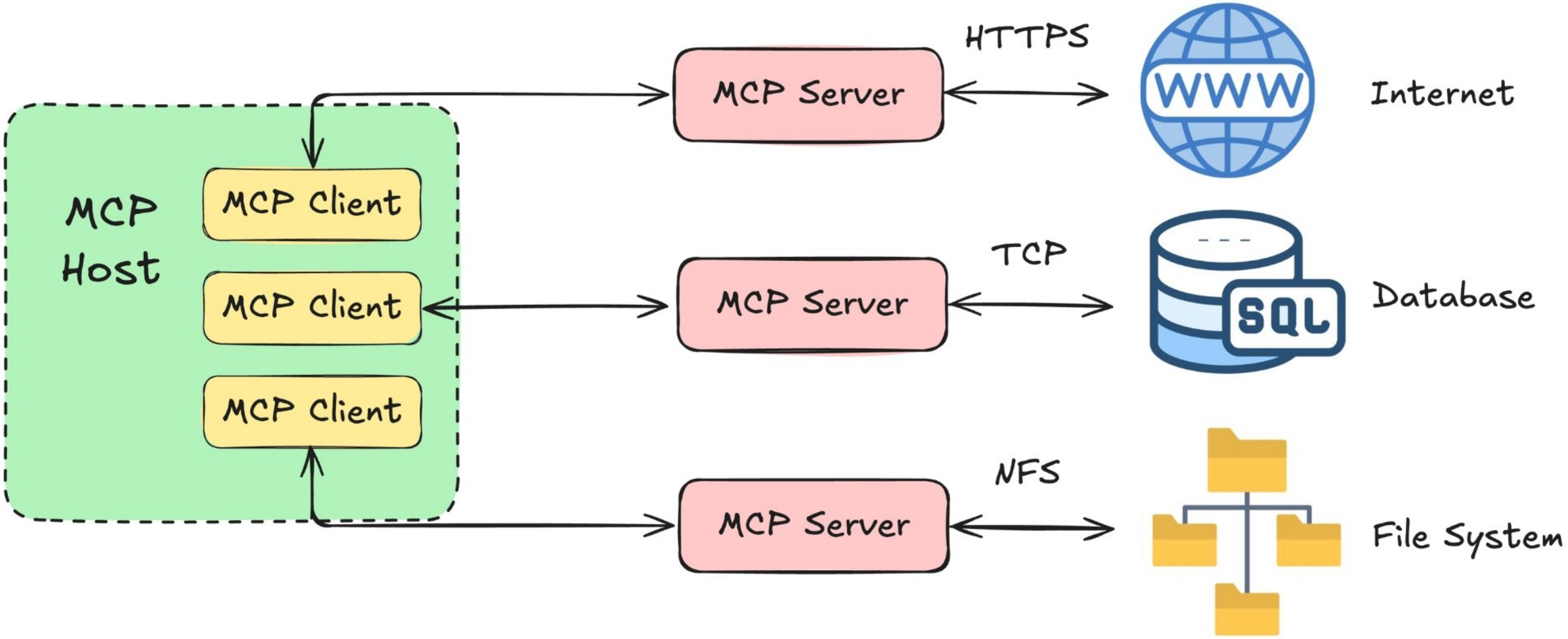
The MCP architecture



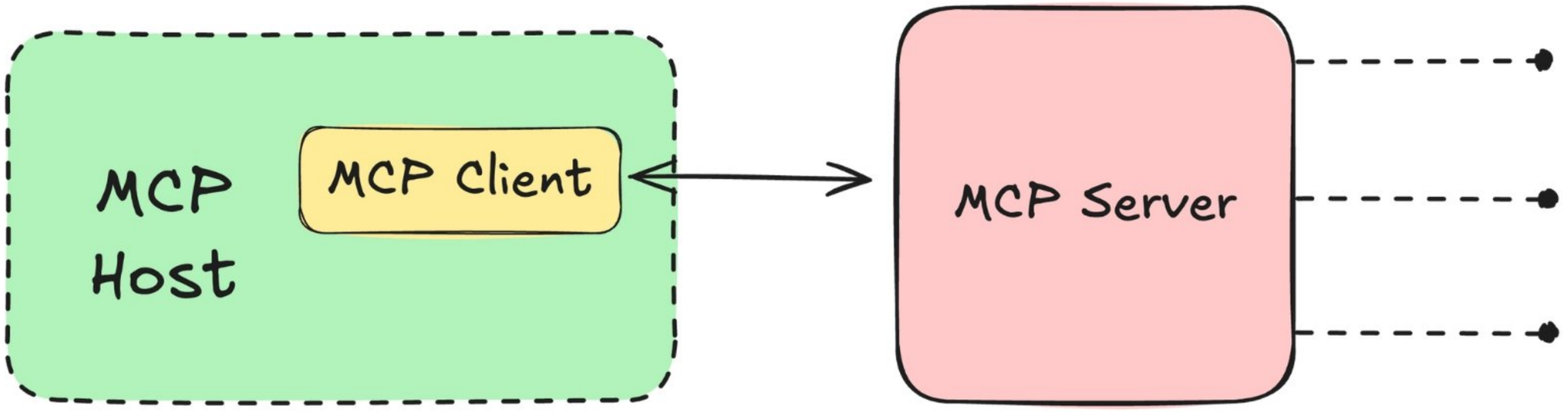
The MCP architecture



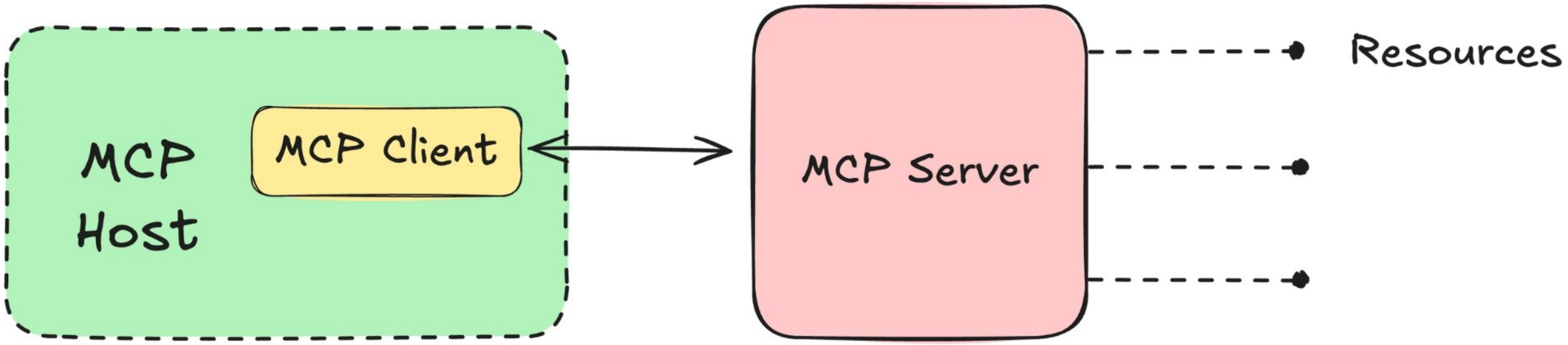
The MCP architecture



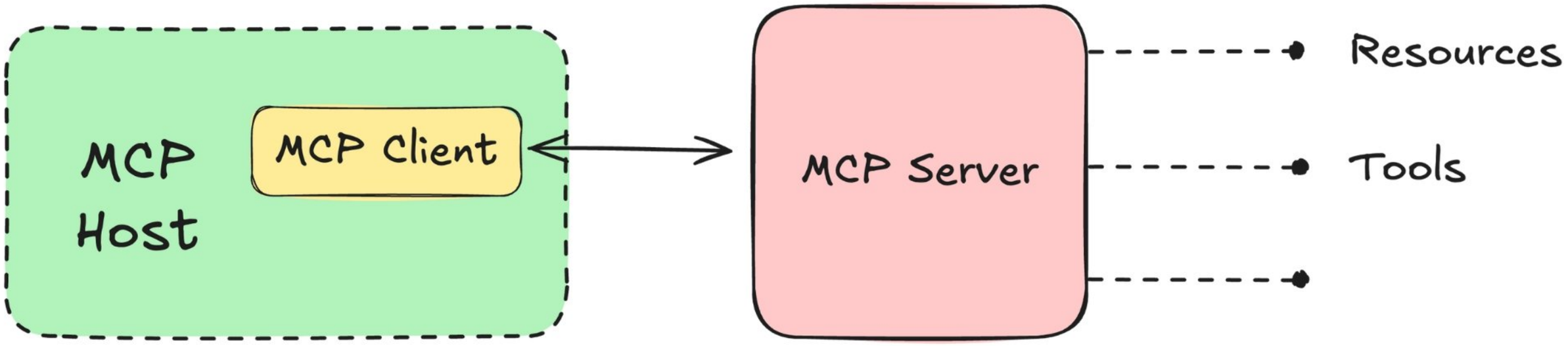
MCP servers



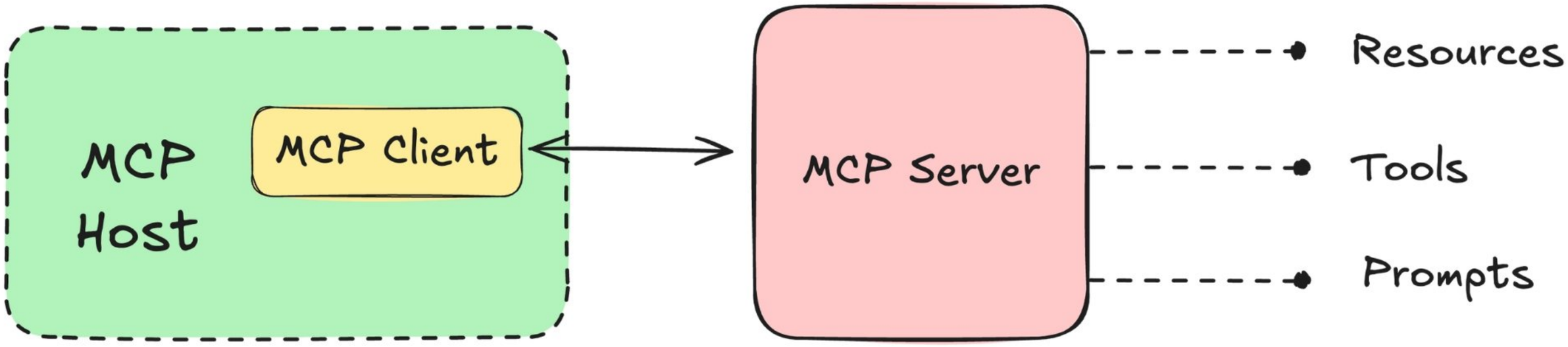
MCP servers



MCP servers



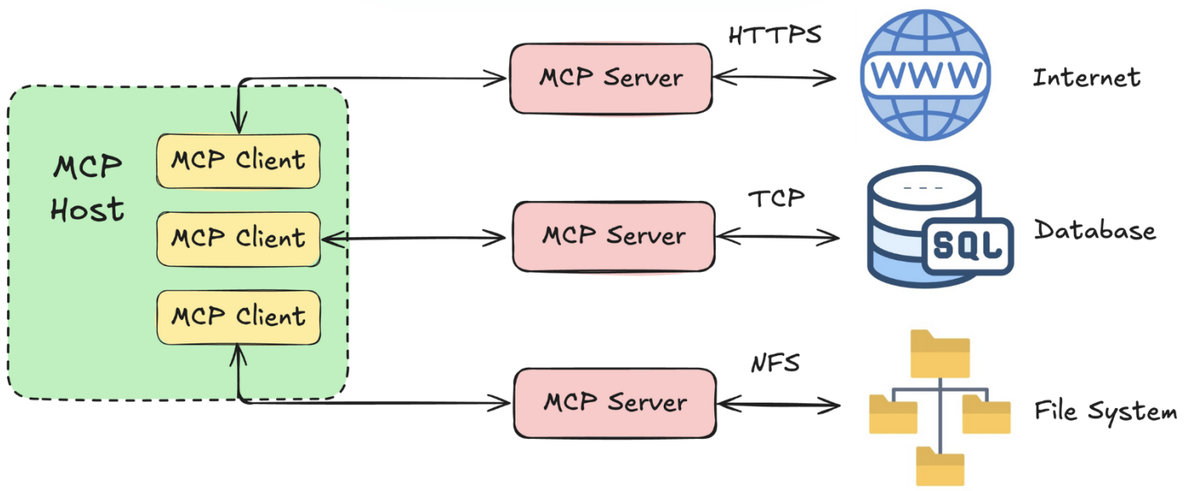
MCP servers



MCP

→ Standard for connecting AI and data

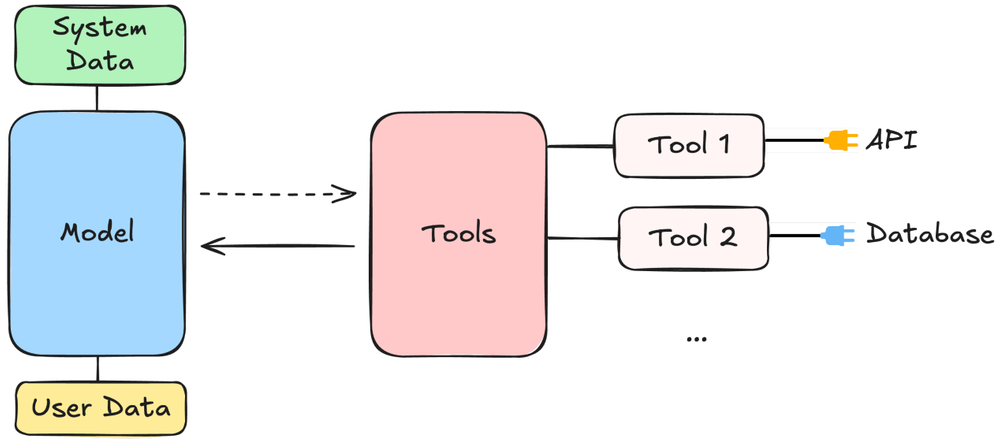
- Dynamic tool discovery ("integrate once")
- LLM-agnostic
- Standardized security



No framework

→ Custom integrations for each data source

- Tools need to be statically coded
- Often refactoring needed for different LLMs
- Tool-specific keys and authentication mode



Let's practice!

BUILDING SCALABLE AGENTIC SYSTEMS

The Agent-to-Agent (A2A) Protocol

BUILDING SCALABLE AGENTIC SYSTEMS

Korey Stegared-Pace
Senior AI Cloud Advocate, Microsoft

- Developed by **Google**
- Enable *multi-agent collaboration*
- Vendor- and tool stack-agnostic

Example: Corporate travel

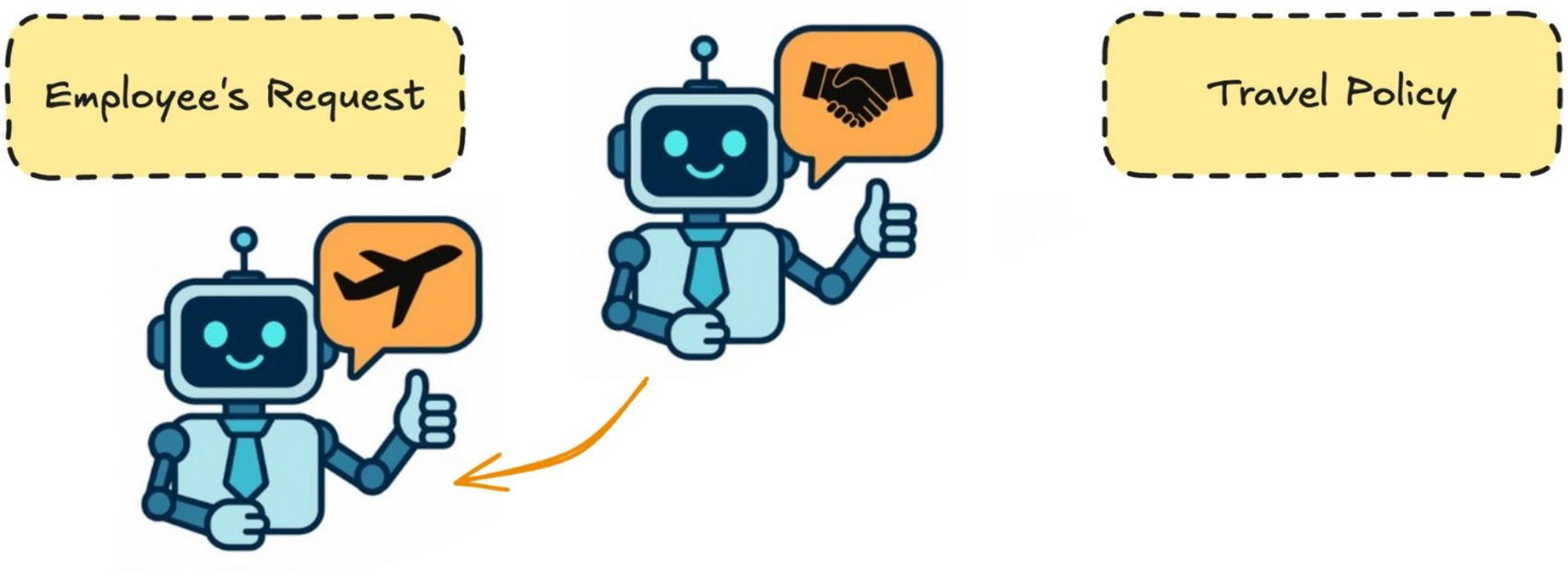
Employee's Request

Example: Corporate travel

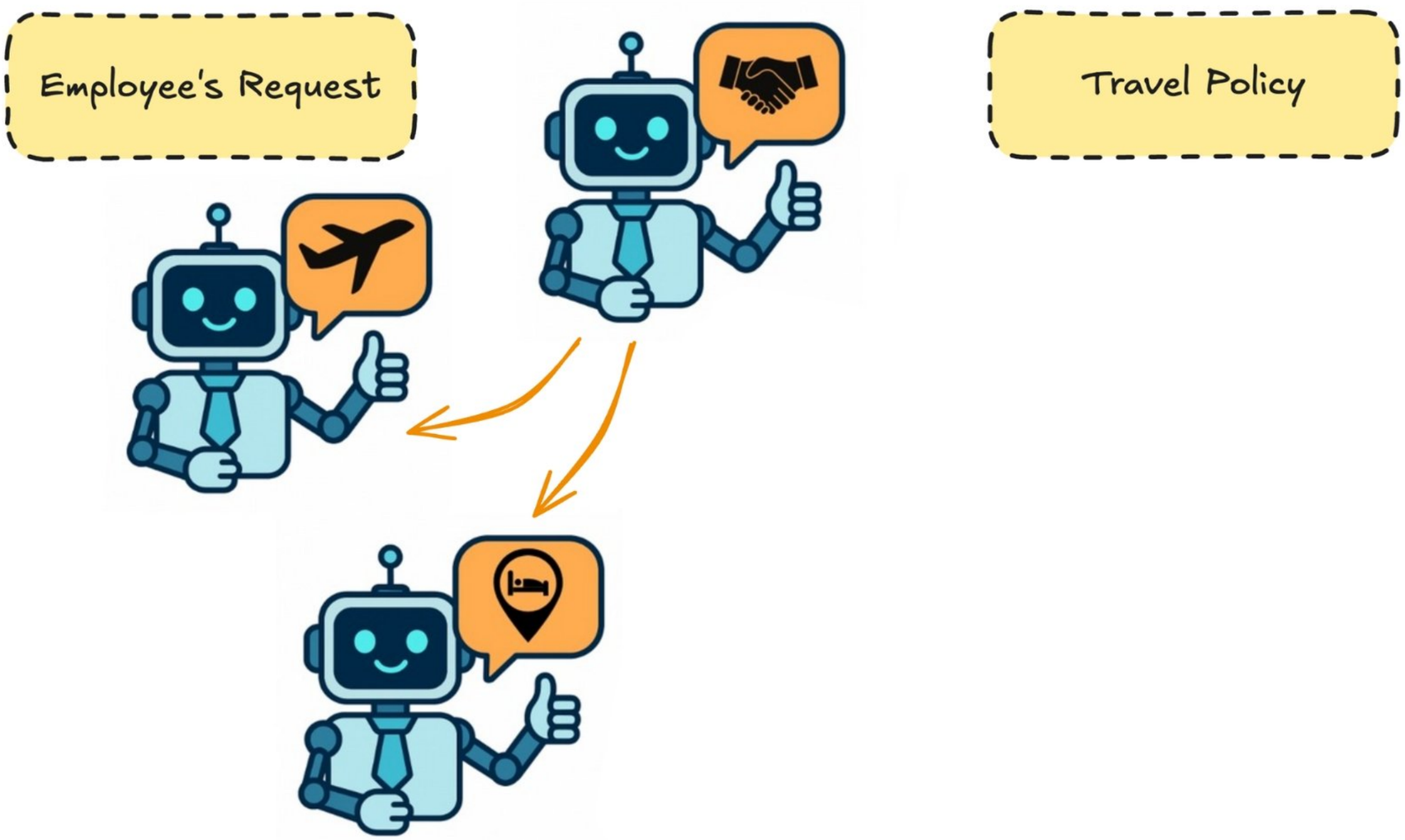
Employee's Request

Travel Policy

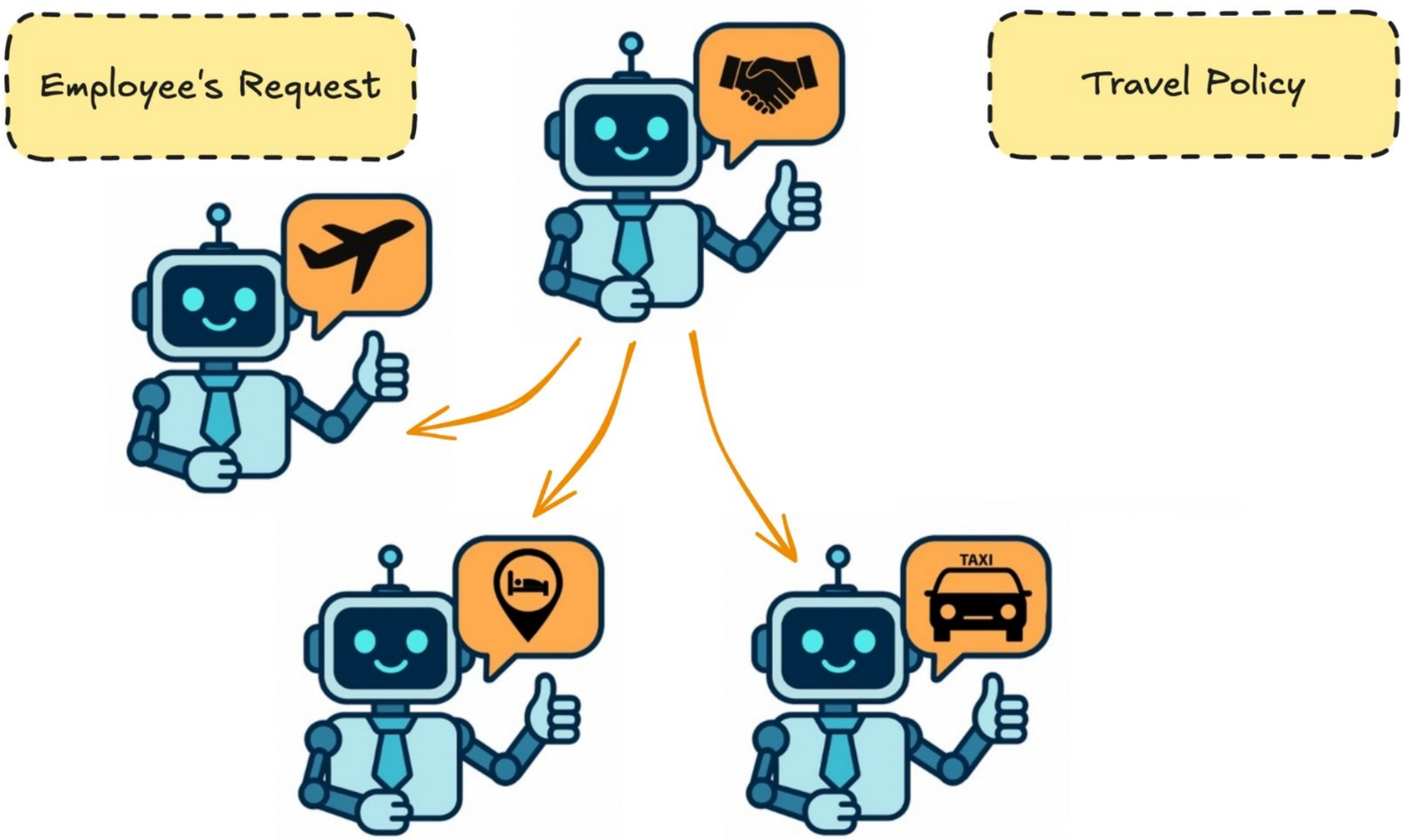
Example: Corporate travel



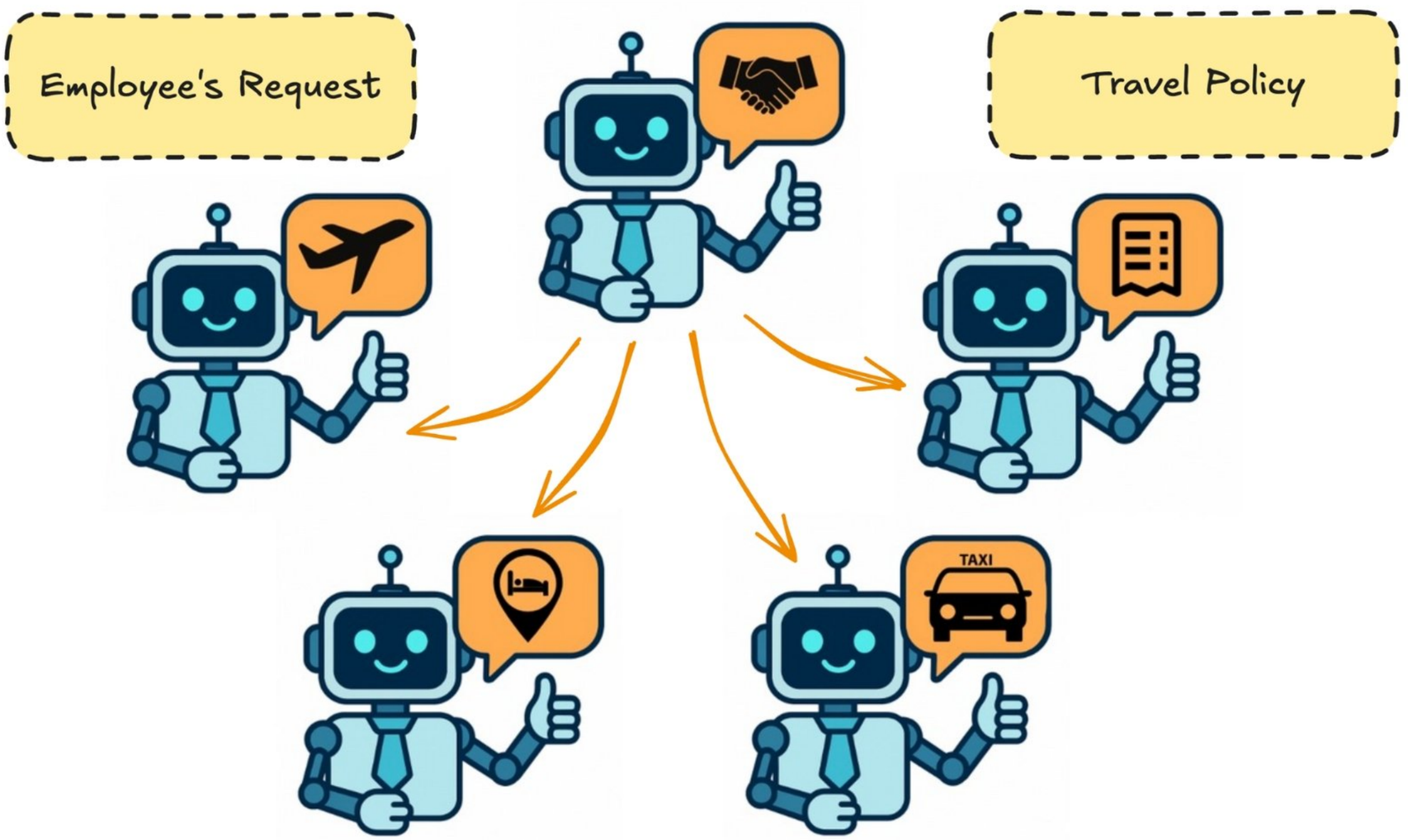
Example: Corporate travel



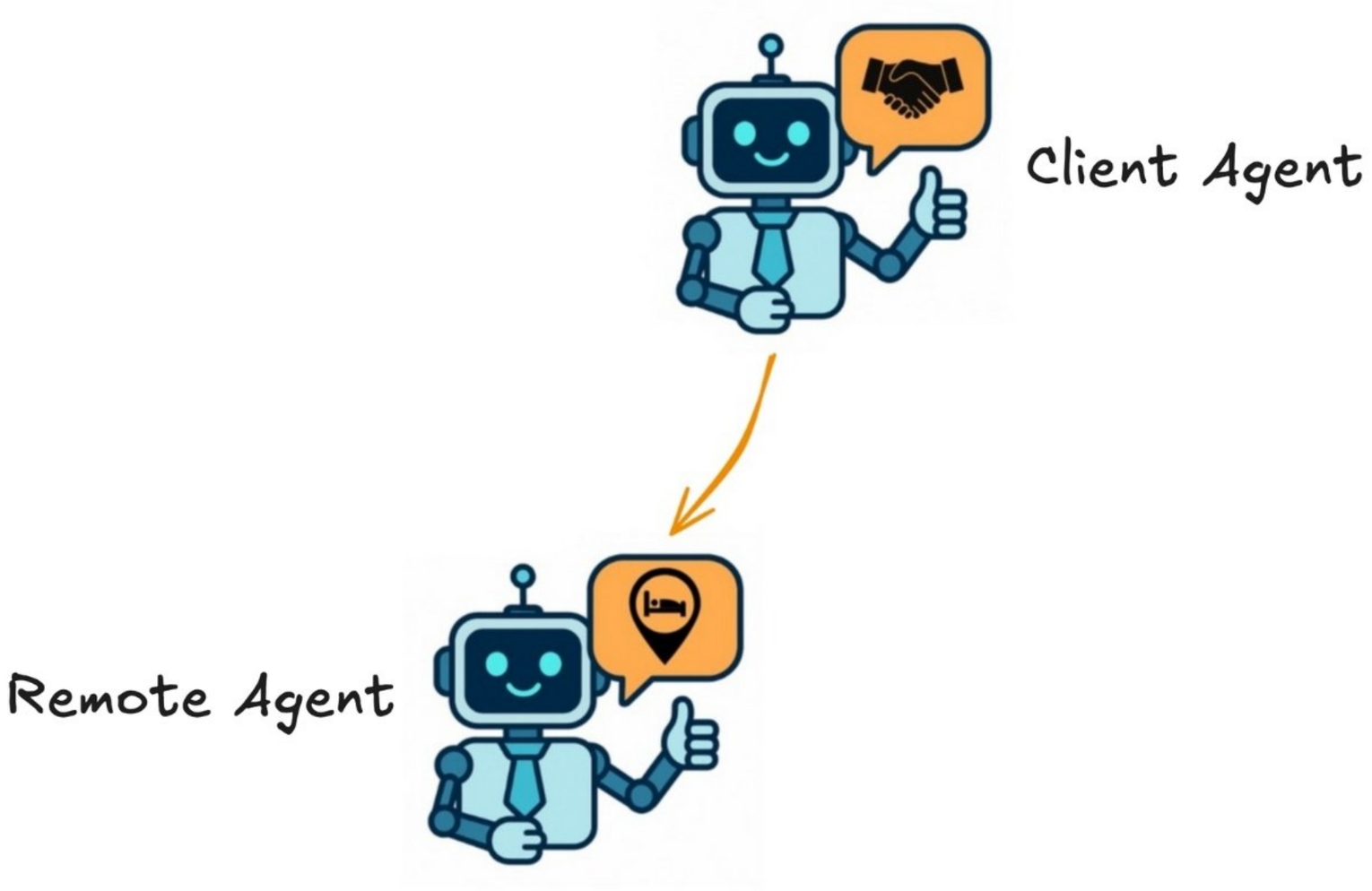
Example: Corporate travel



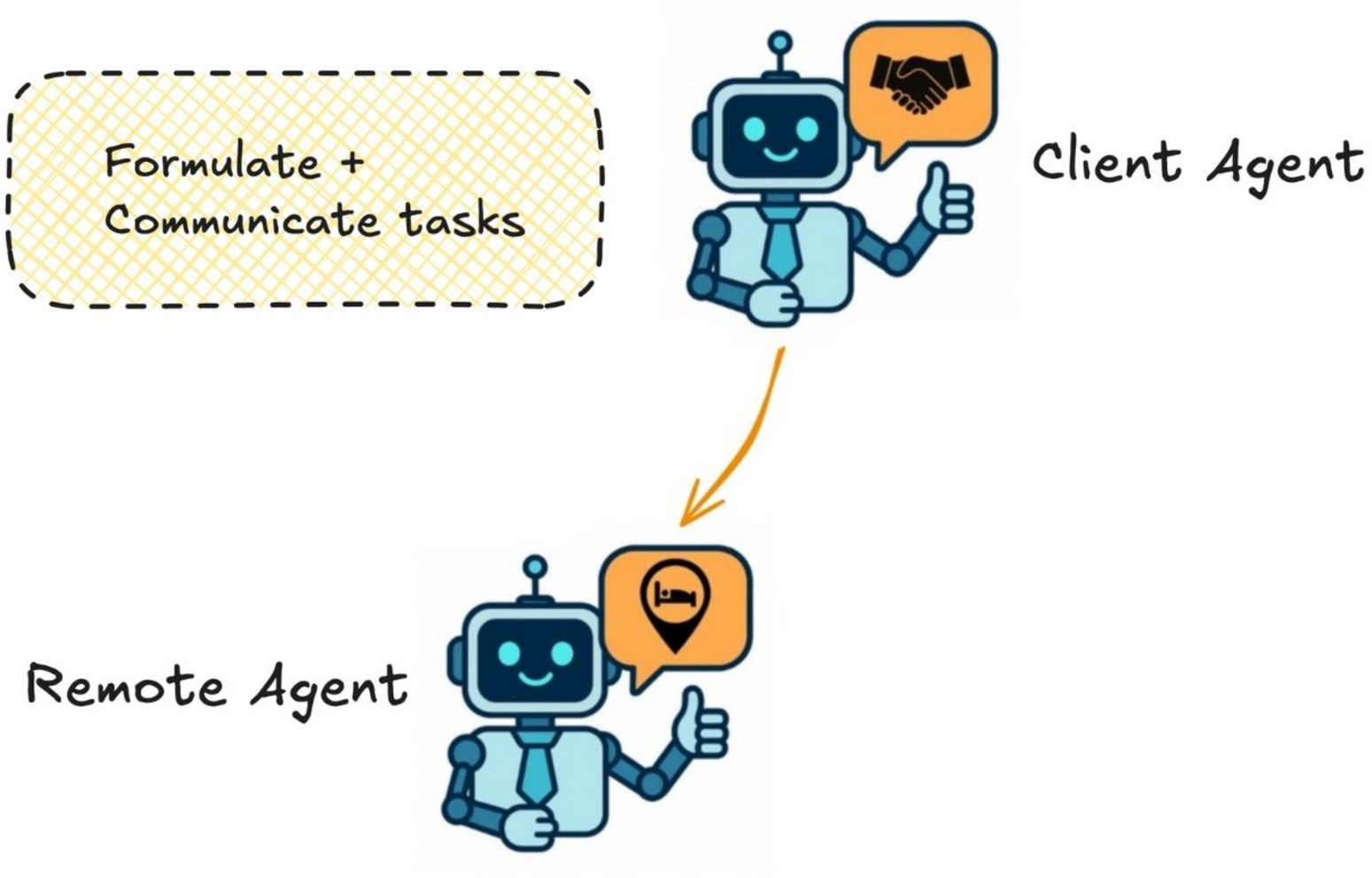
Example: Corporate travel



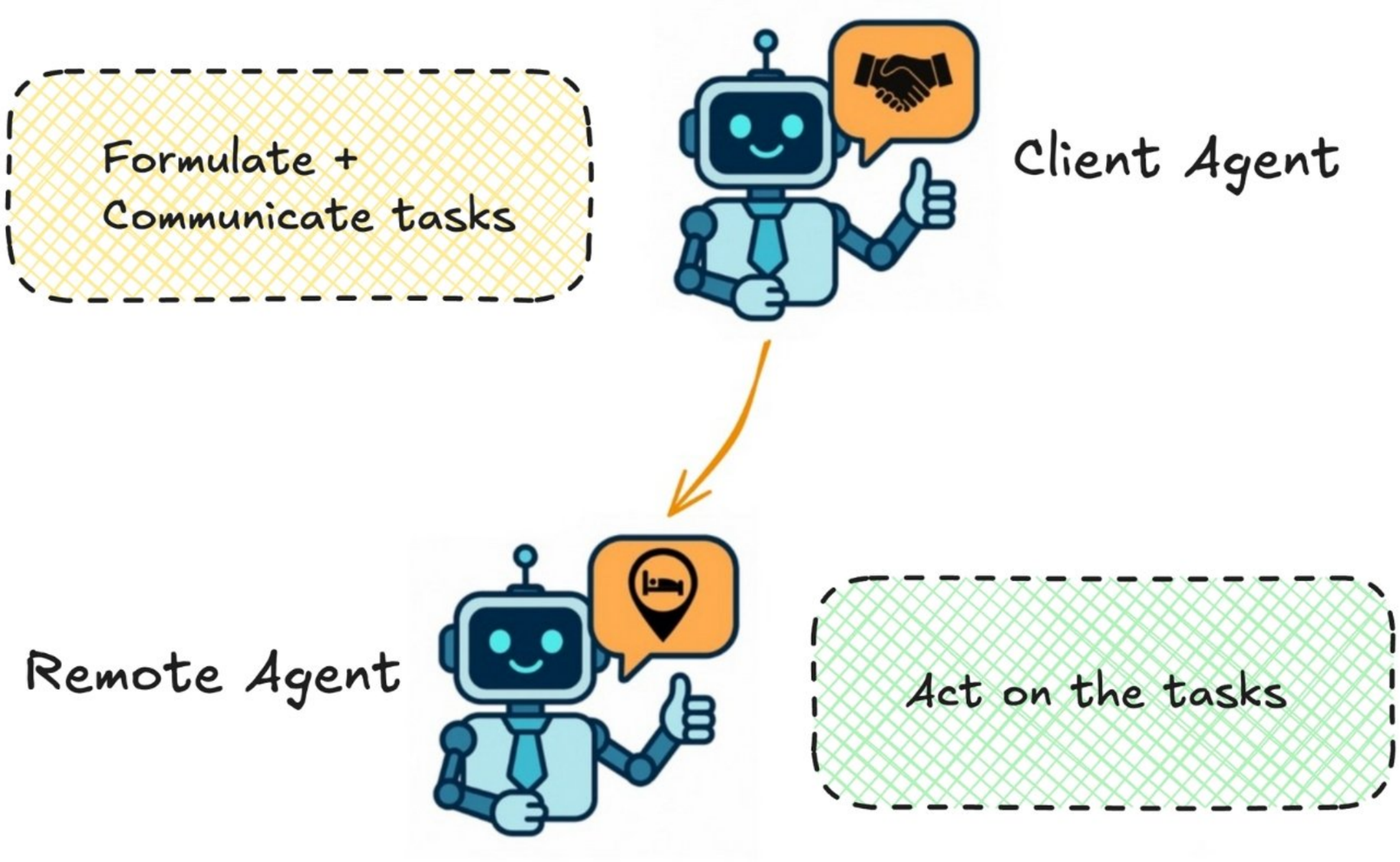
Agent-to-Agent (A2A)



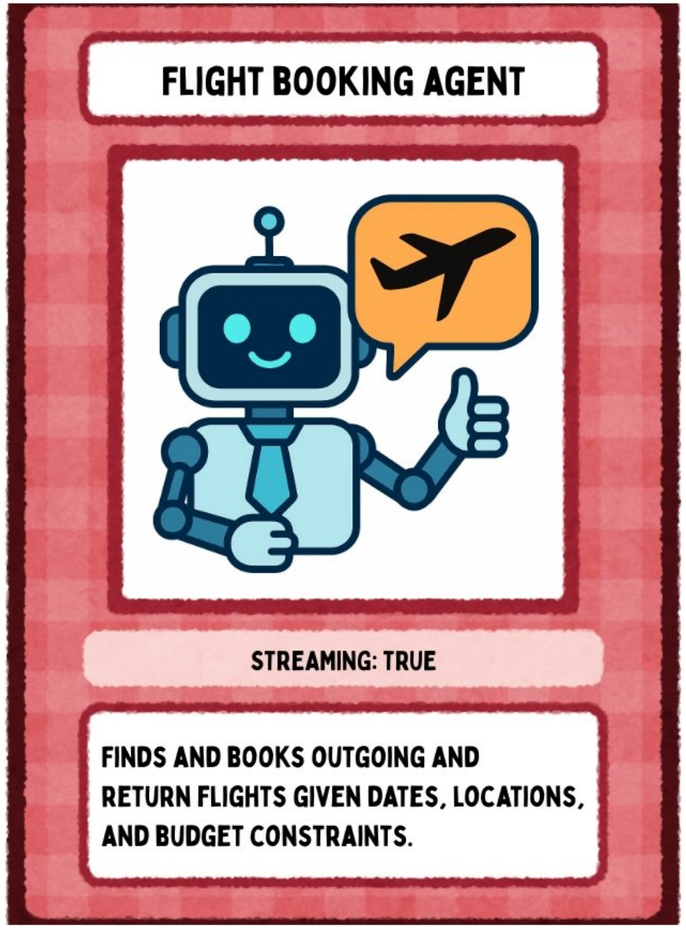
Agent-to-Agent (A2A)

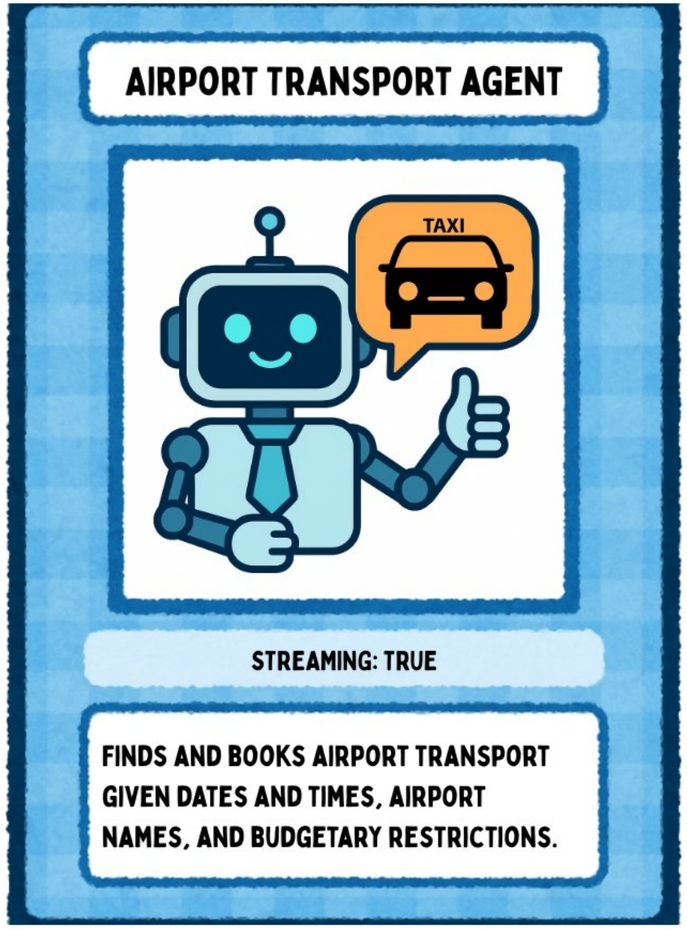


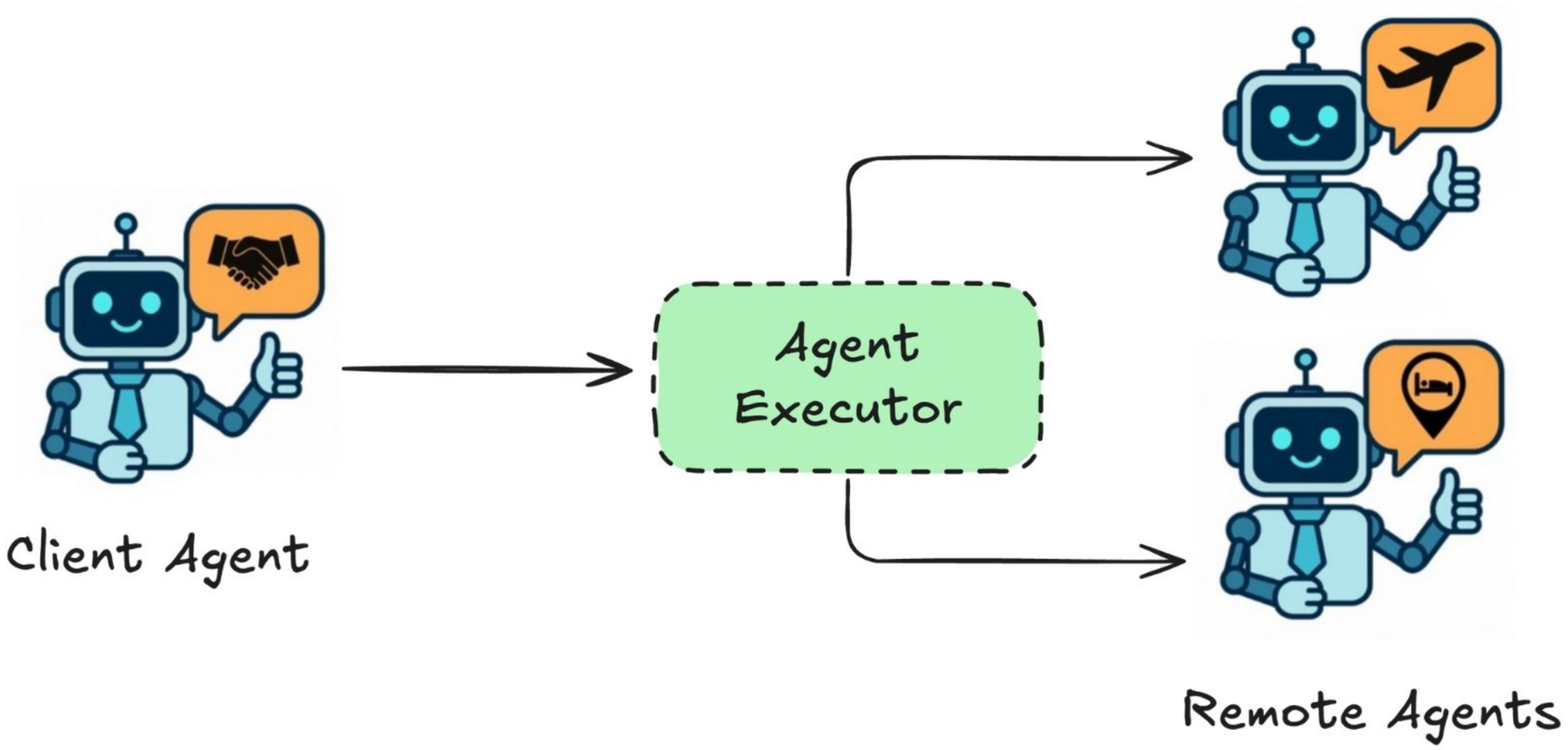
Agent-to-Agent (A2A)

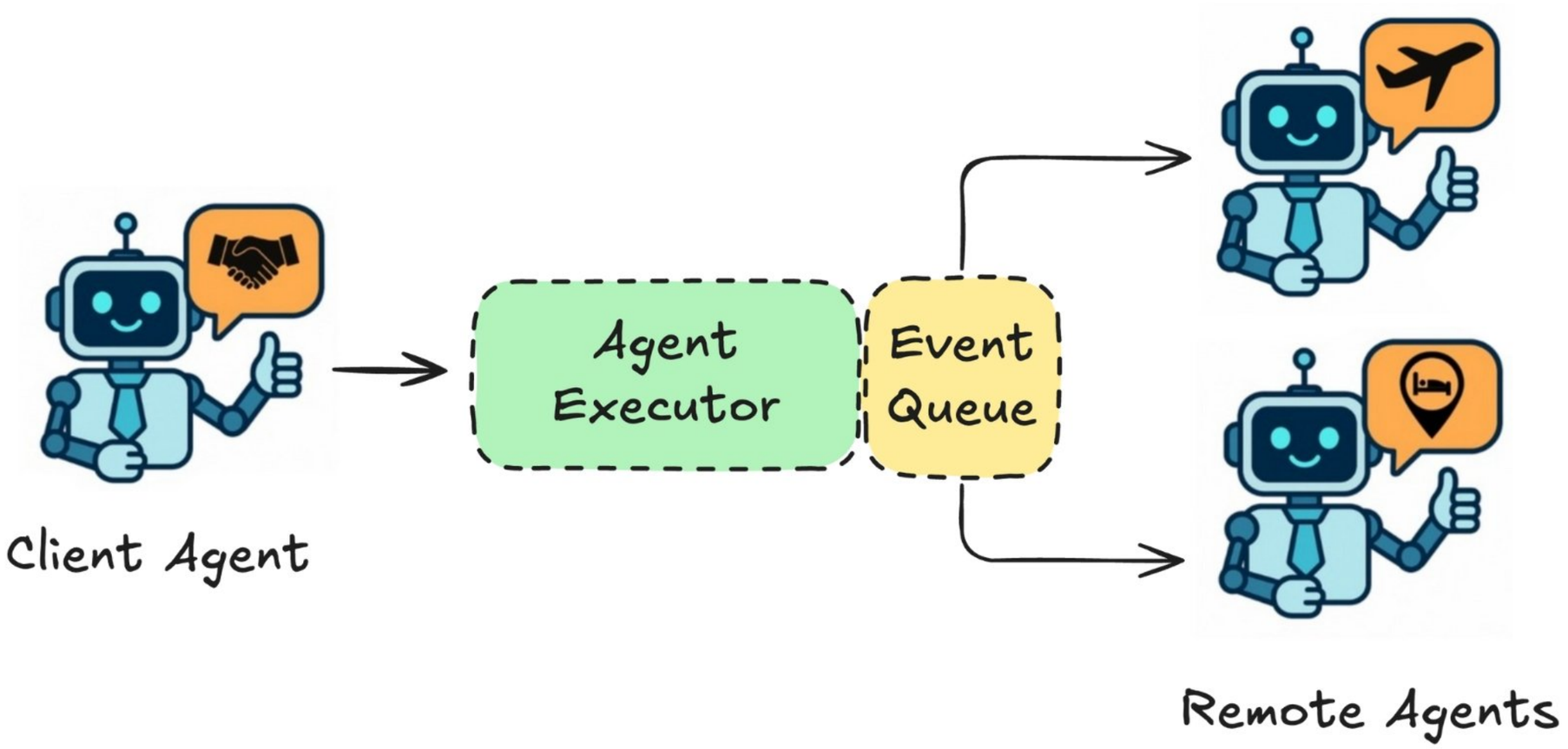


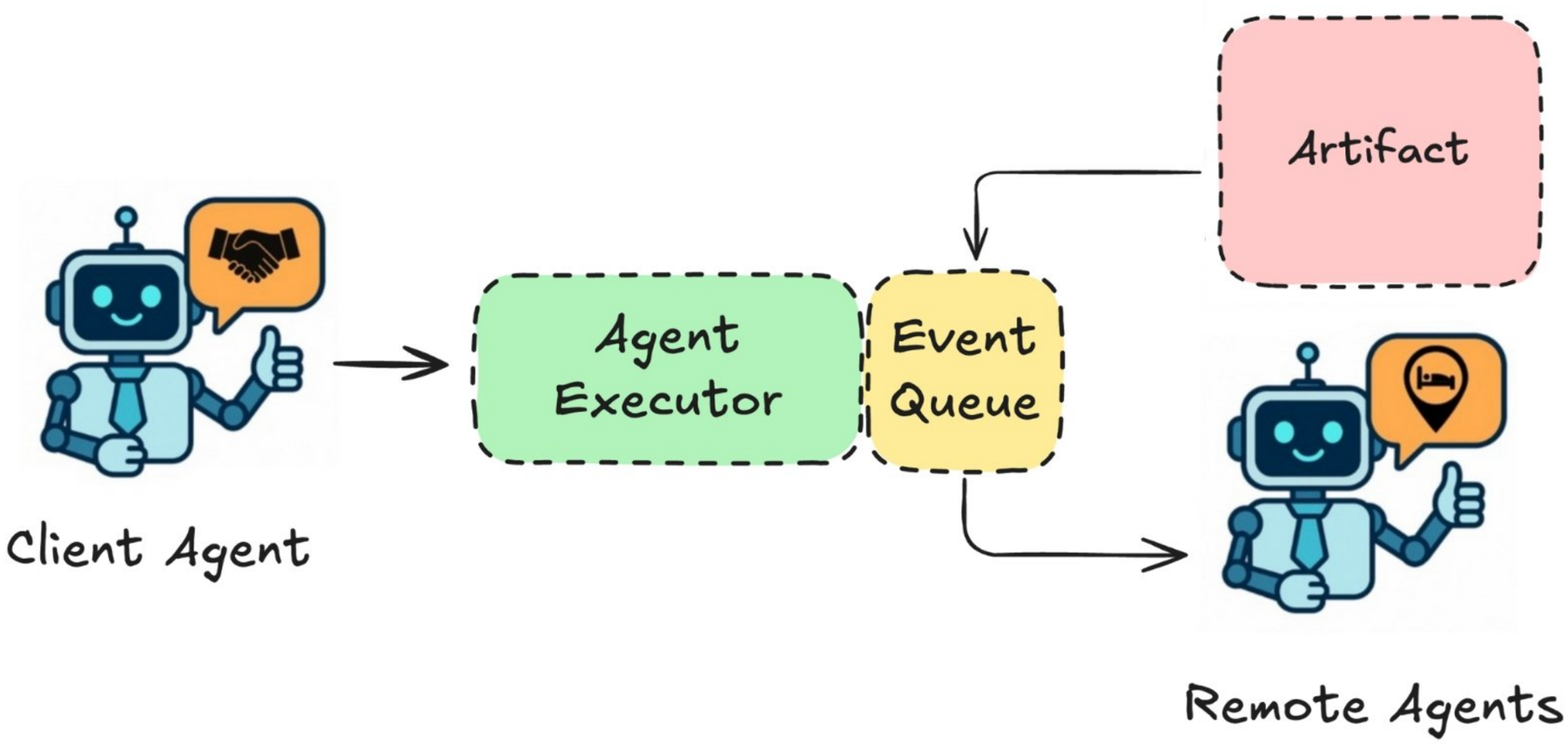
Agent cards


FLIGHT BOOKING AGENT
STREAMING: TRUE
FINDS AND BOOKS OUTGOING AND RETURN FLIGHTS GIVEN DATES, LOCATIONS, AND BUDGET CONSTRAINTS.

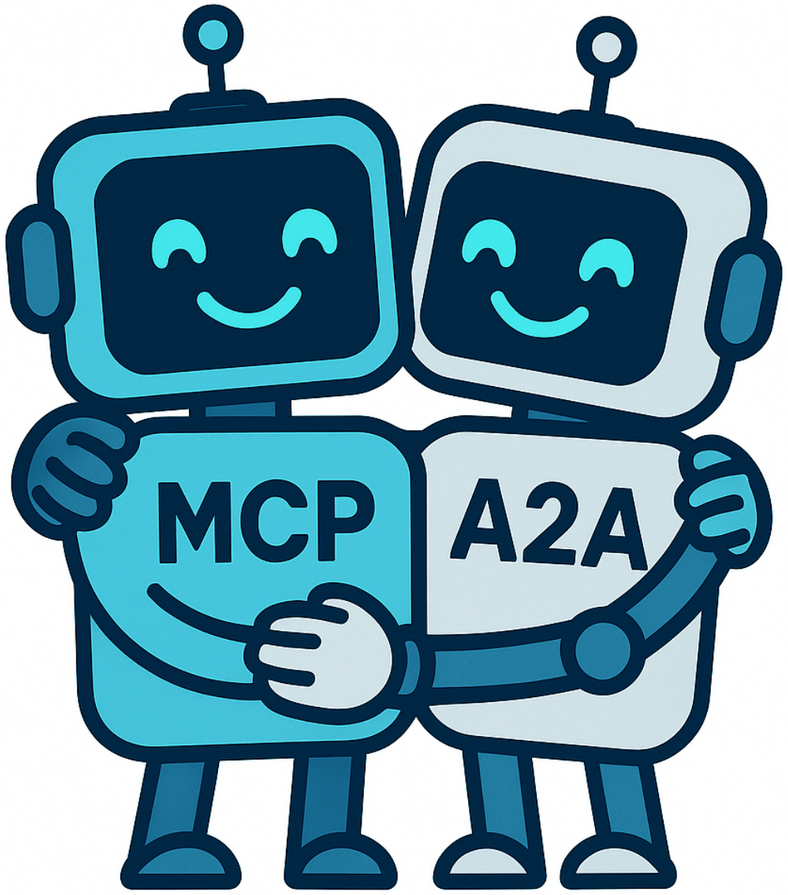

AIRPORT TRANSPORT AGENT
TAXI
STREAMING: TRUE
FINDS AND BOOKS AIRPORT TRANSPORT GIVEN DATES AND TIMES, AIRPORT NAMES, AND BUDGETARY RESTRICTIONS.


Agent
Executor
Client Agent
Remote Agents


Agent
Executor
Event
Queue
Client Agent
Remote Agents


Artifact
Agent Executor
Event Queue
Client Agent
Remote Agents

MCP and A2A



[1] Image generated with GPT-4o

Let's practice!

BUILDING SCALABLE AGENTIC SYSTEMS