AI VIETNAM
@aivietnam.edu.vn

Module 10 - Project

Visual Agentic AI

Multi-agent Supervisor using LangGraph

Code - Data

LLM-Agents Space

Nguyen Quoc Thai
MSc in Computer Science
STA Tran Minh Nam

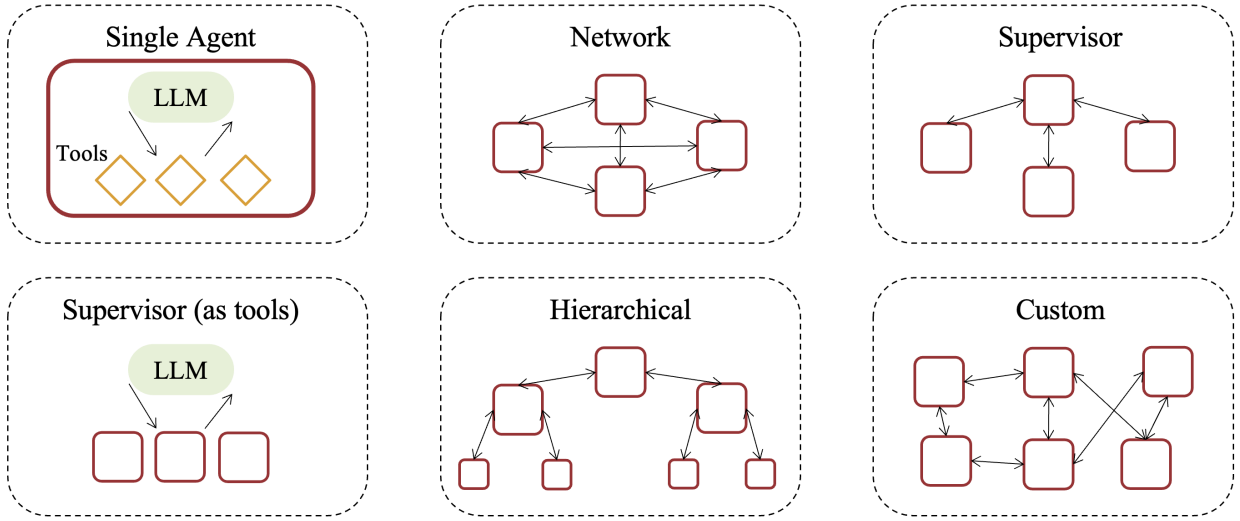
Objectives

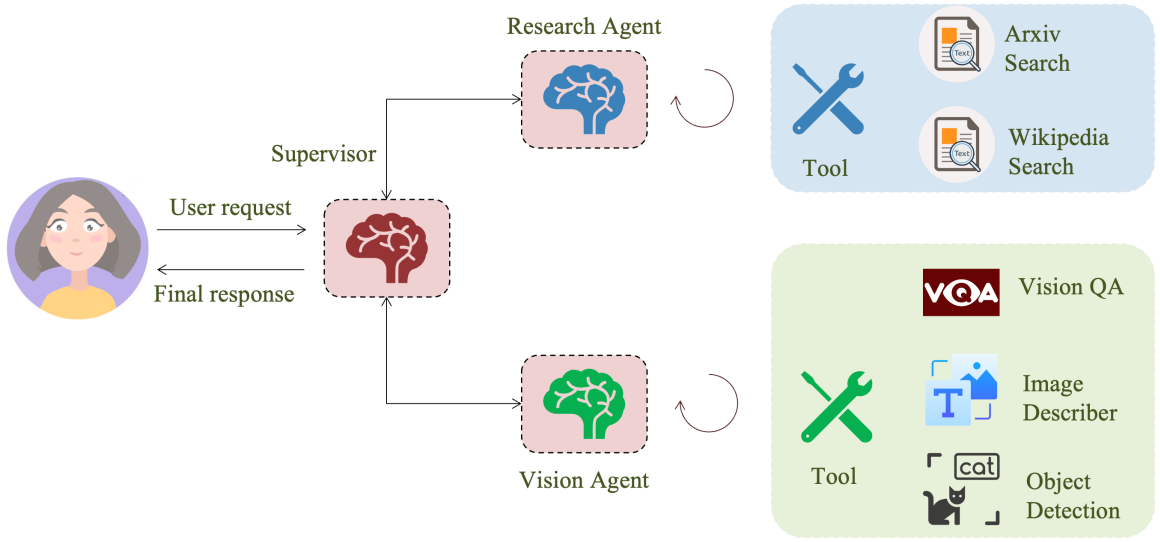
Multi-agent Systems

- AI Agent
- Multi-agent Systems
- Multi-agent using LangGraph

Visual Agentic AI

- Multi-agent Supervisor
- Vision Agent
- Research Agent

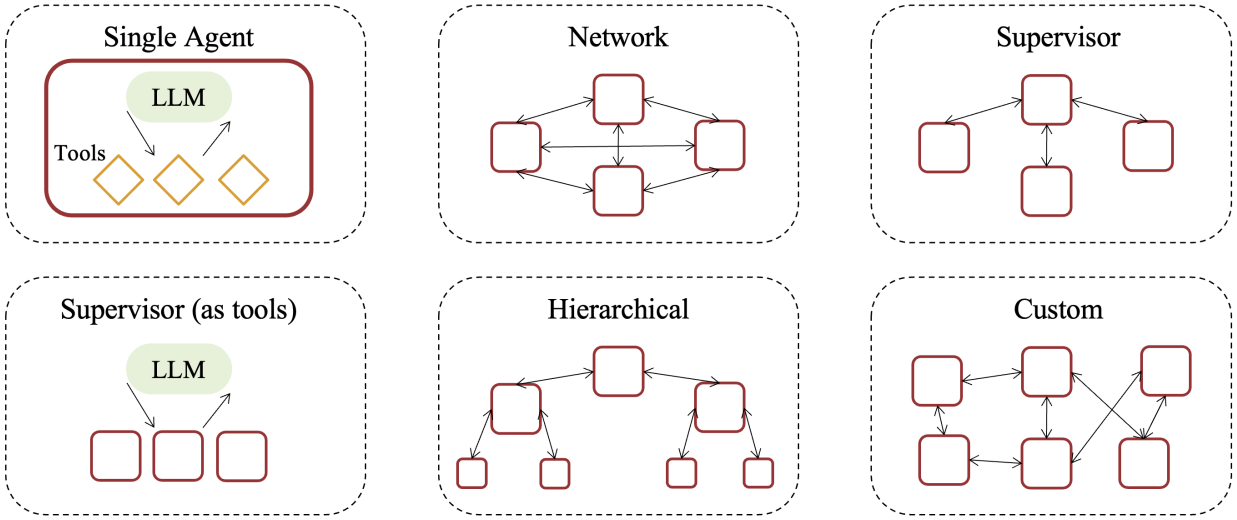




Outline

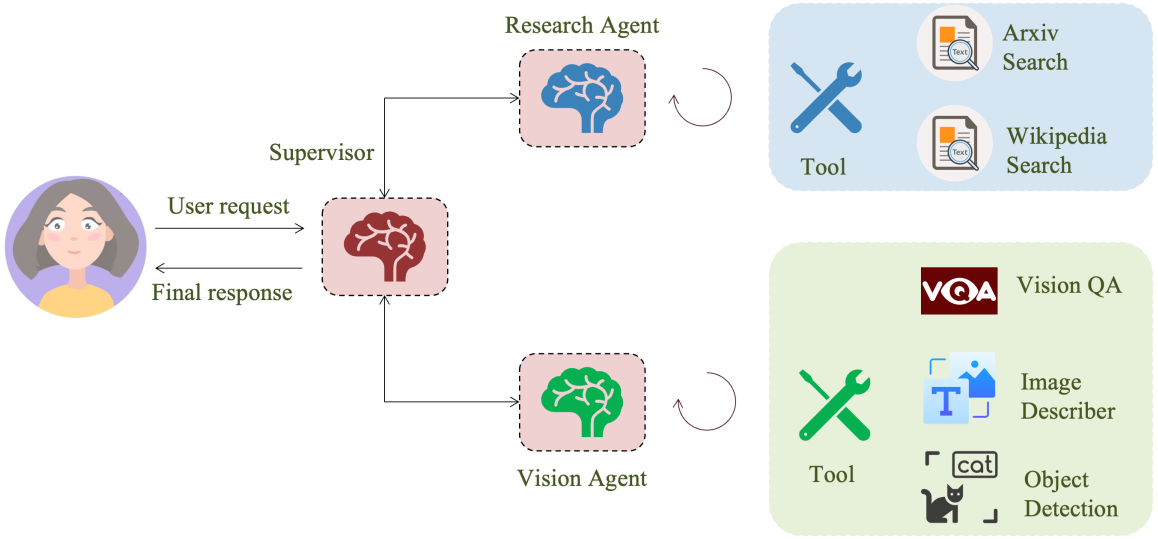
SECTION 1

Multi-agent Systems



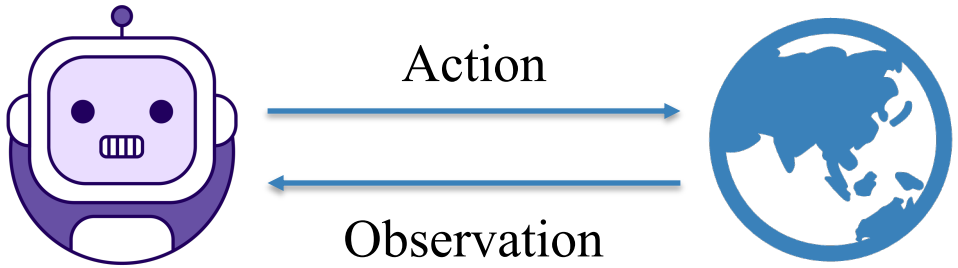
SECTION 2

Visual Agentic AI



Multi-agent Systems

What is "Agent"?

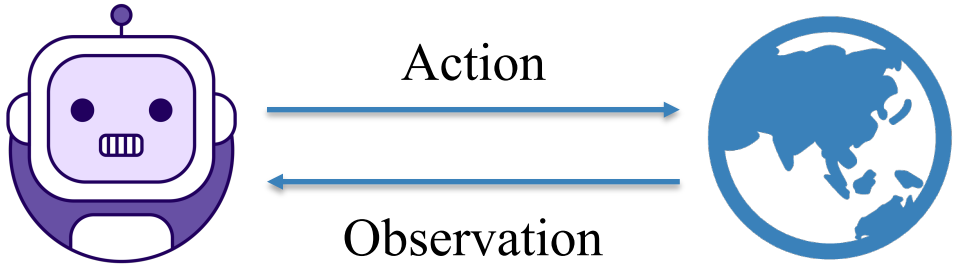


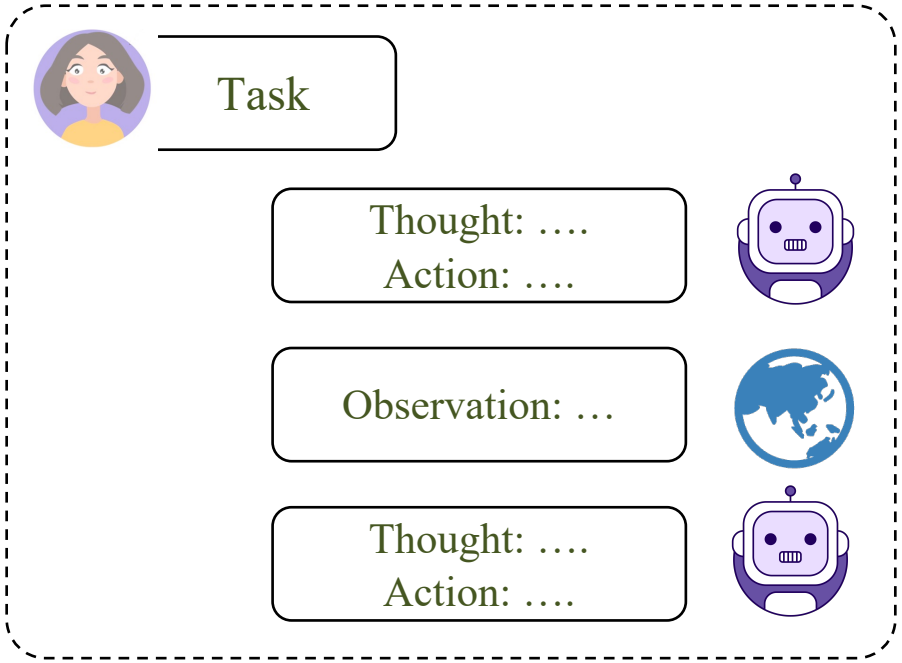
- **An "intelligent" system that interacts with some "environment"**
  - Physical environments: robot, autonomous car,…
  - Digital environments: Siri, AlphaGo,…
  - Humans as environments: chatbot
- **Define "agent" by defining "intelligent" and "environment"**
  - It changes over time!
  - Exercise question: how would you define "intelligent"?

Multi-agent Systems

What is "LLM Agent"?

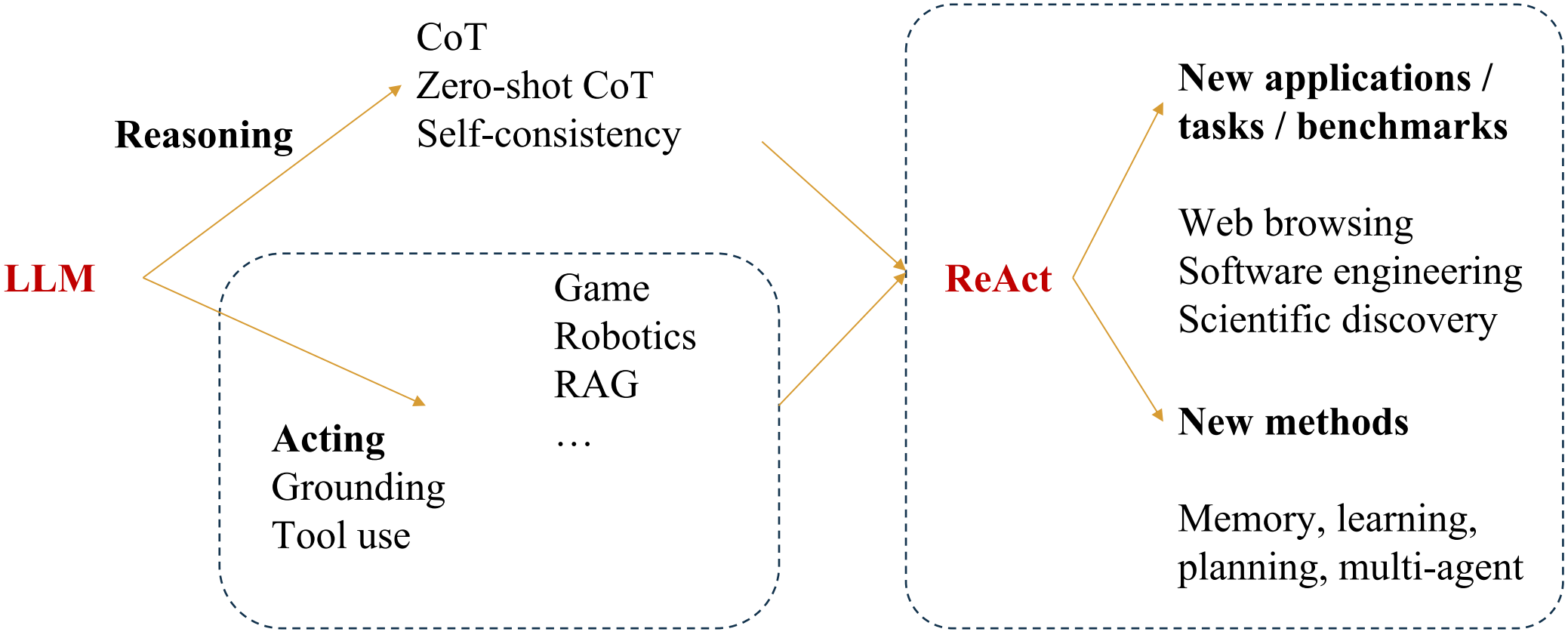
- **Level 1: Text agent**
  - Uses text action and observation
  - Example: ELIZA, LSTM-DQN
- **Level 2: LLM agent**
  - Uses LLMs to act
  - Example: SayCan, Language Planner
- **Level 3: Reasoning agent**
  - Uses LLMs to reason to act
  - Example: ReAct, Plain-and-Execute Agent, AutoGPT, …





Multi-agent Systems

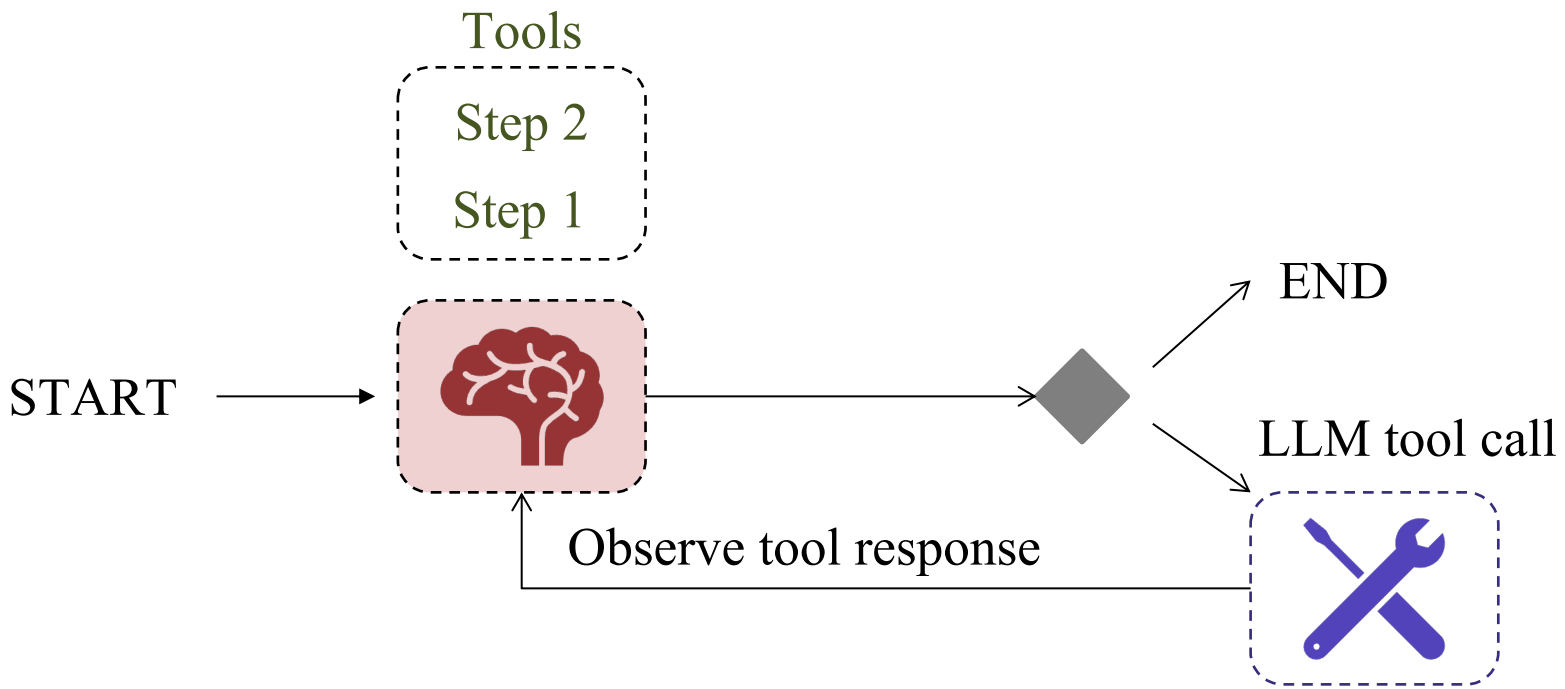
A brief history of LLM agents



Multi-agent Systems

Tool Calling

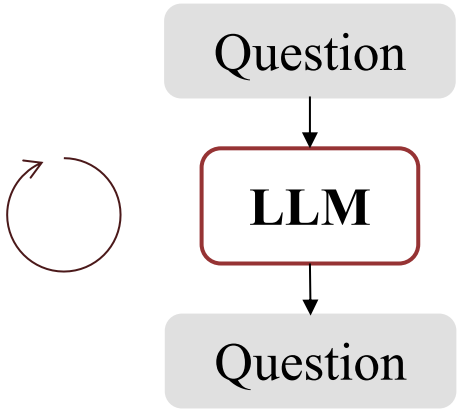
- Autonomous agents can define dynamic control flow using tools to execute steps, but this introduces challenges like ambiguity, non-determinism, and potential misuse.



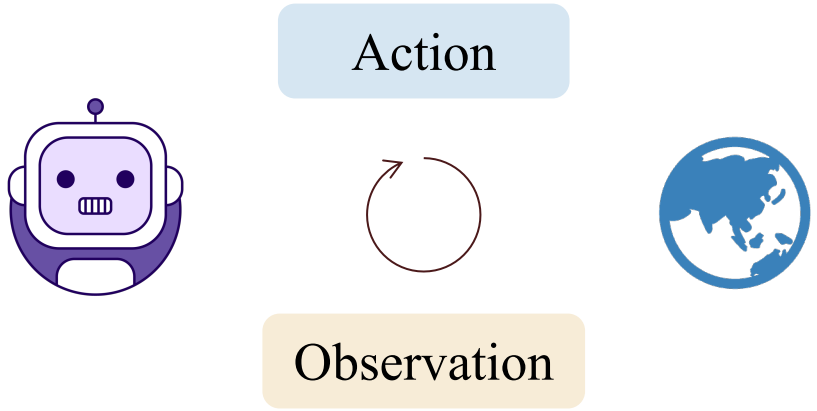
Multi-agent Systems

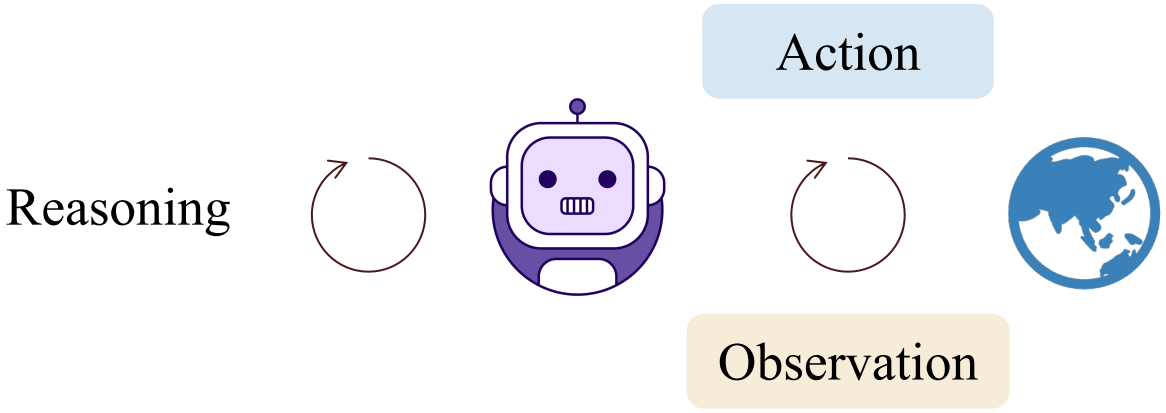
ReAct: Reasoning + Acting

Reasoning (update internal belief)



Acting (obtain external feedback)





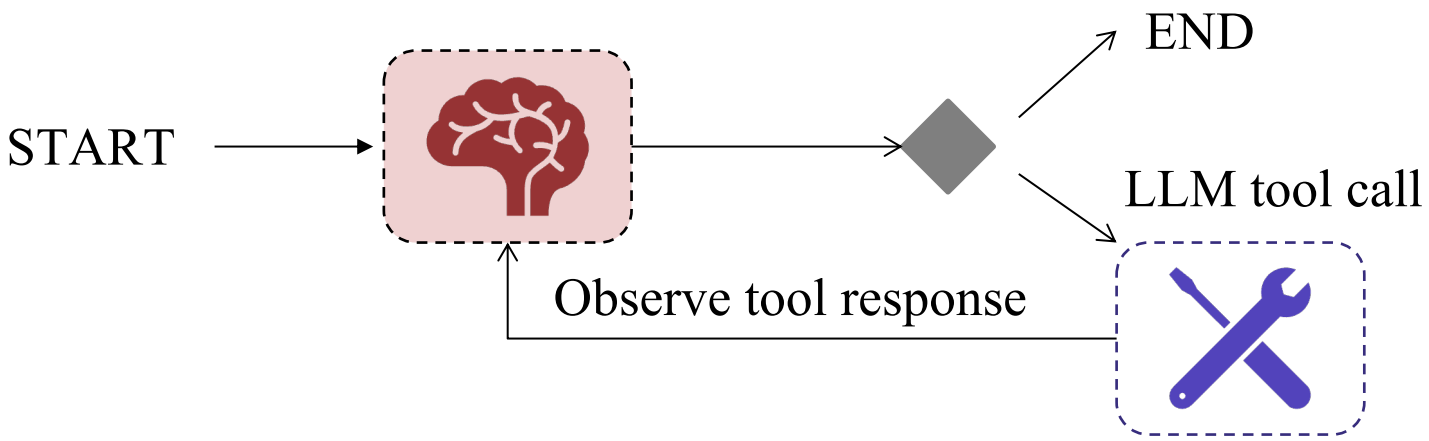
ReAct: a new paradigm of agents that reason and acting

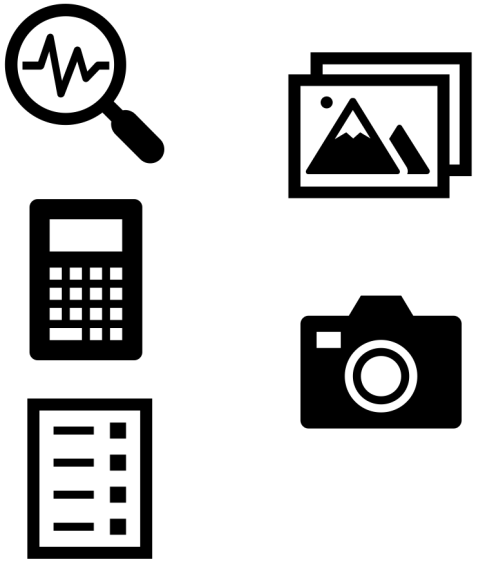
- Synergy of reasoning and acting
- Simple and intuitive to use
- General across domains

Multi-agent Systems

Common Challenges of an Agent

- Agent has too many tools and makes poor decisions
- Context becomes too complex for a single agent
- Need for multiple specialization areas (planning, research, expertise)

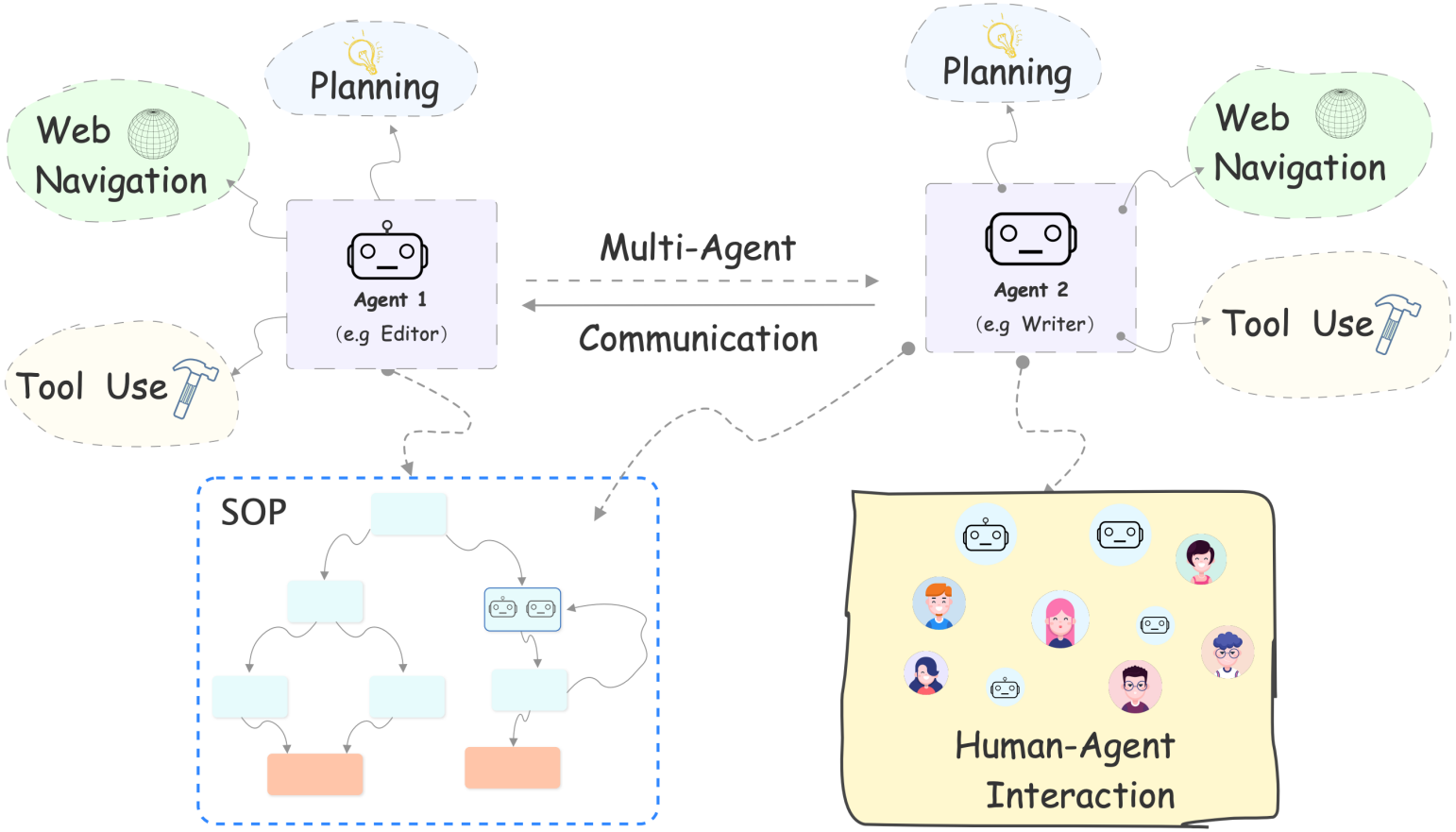




Multi-agent Systems

Multi-agent Systems

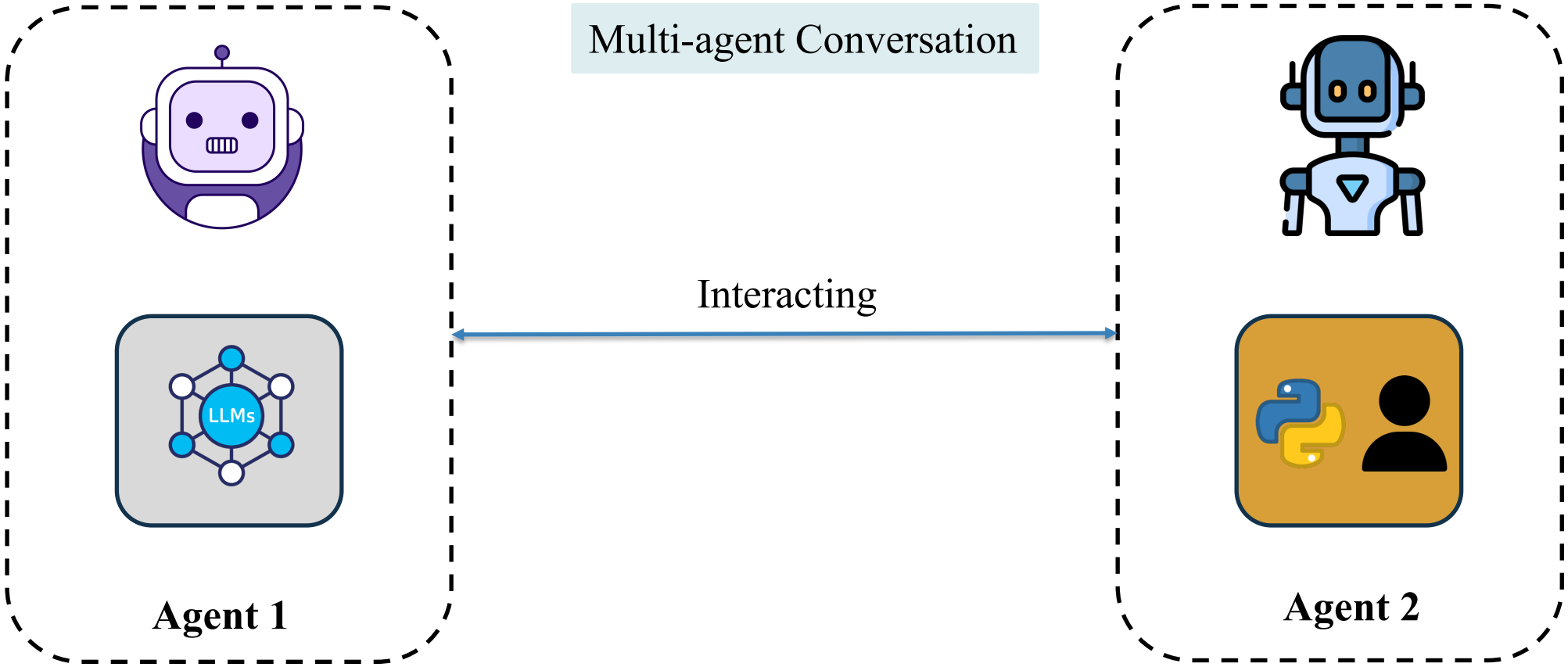
- All agents with the ability to use tools, interacting to each other to solve a given task.



Agents: An Open-source Framework for Autonomous Language Agents

Multi-agent Systems

Multi-agent Systems



Multi-agent Systems

Multi-agent Systems

Find GDP of Vietnam in 2023 and 2024, determine the percentage increase or decrease?

Find GDP of Vietnam in 2023 and 2024, determine the percentage increase or decrease?

Do you want nominal GDP or GDP per capita?

Nominal GDP, please.

Got it. Retrieving data from World Bank API... In 2023, Vietnam's nominal GDP was $430 billion. In 2024, it's estimated at $476 billion.

Thanks. Now calculate the percentage change.

The GDP increased by approximately 10.7% from 2023 to 2024.

Summarize your finding in one sentence.

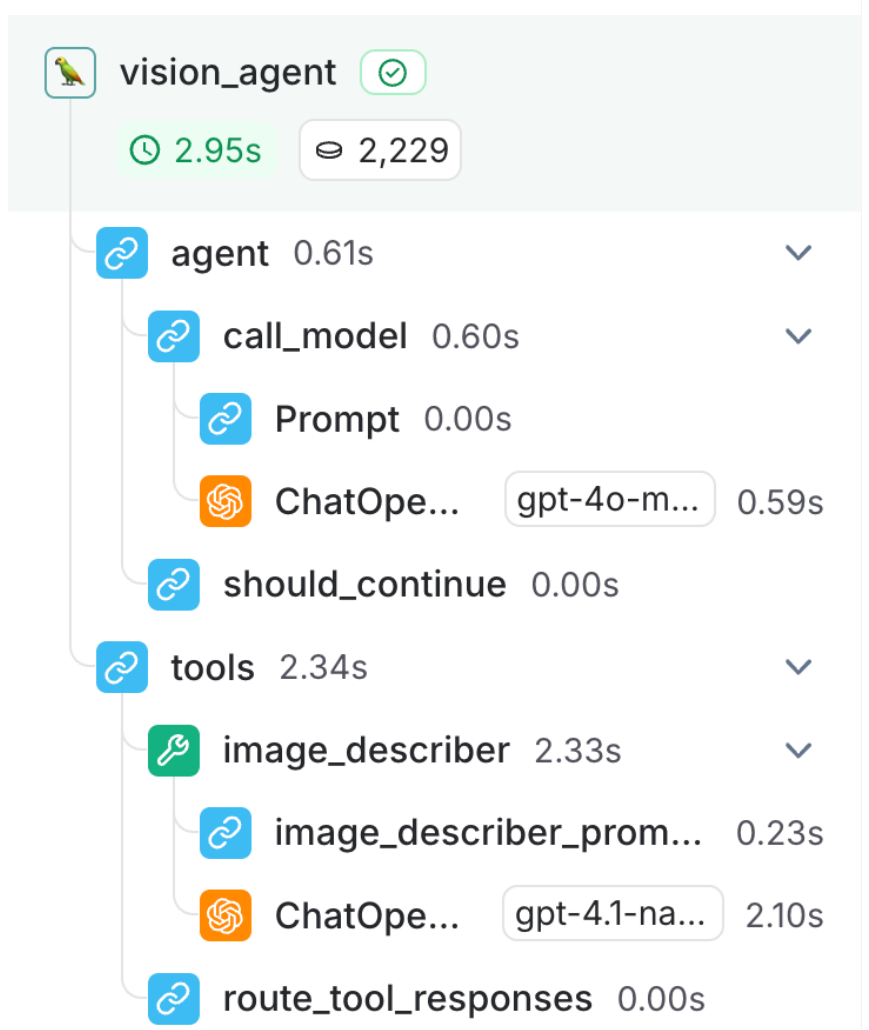
Vietnam's nominal GDP grew from $430B in 2023 to $460B in 2024, a 6.98% increase..

Thank so much.

Multi-agent Systems

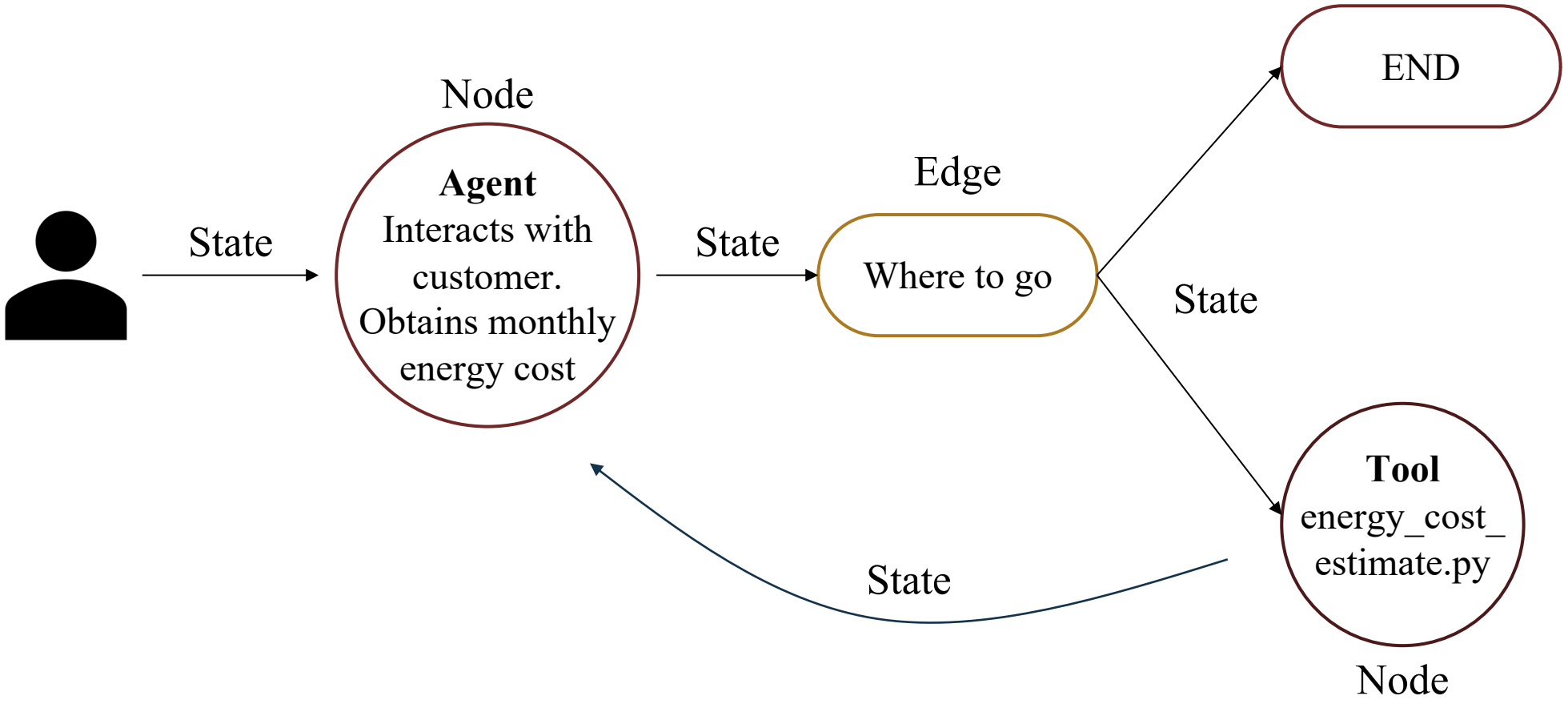
LangGraph

- **LangGraph applications balance agent control with agency**.
- Its core pillars support:
  - Controllability: to define both explicit and implicit workflows
  - Persistence: for thread-specific and cross-thread "Memory"
  - Interaction: between agents, humans and external systems
  - Streaming: to expose any event (or token) as it occurs
- Python & Jupyter notebooks
- Traceability and Prompt Management (LangSmith - Langfuse)



Multi-agent Systems

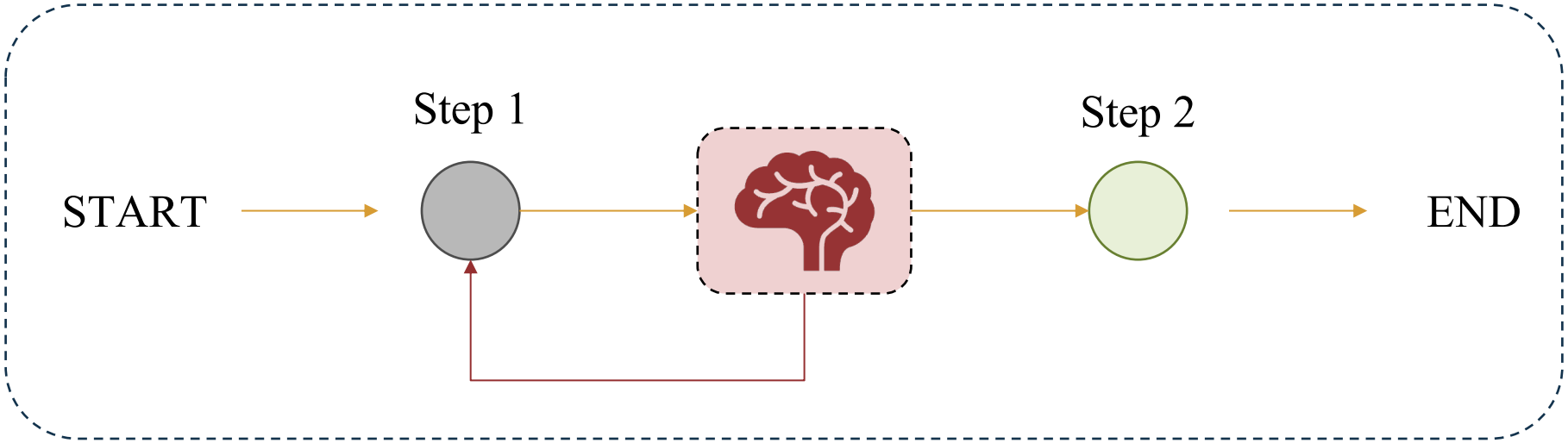
LangGraph



Multi-agent Systems

LangGraph

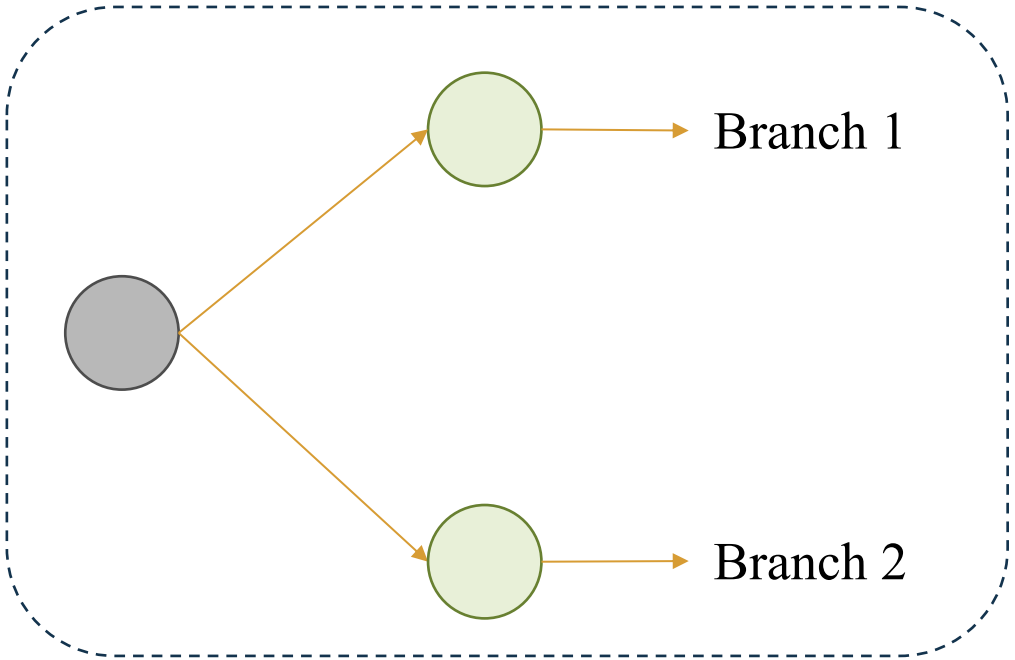
- **Controllability**: to define both explicit and implicit workflows
- LangGraph allows for developer + LLM-defined control flows

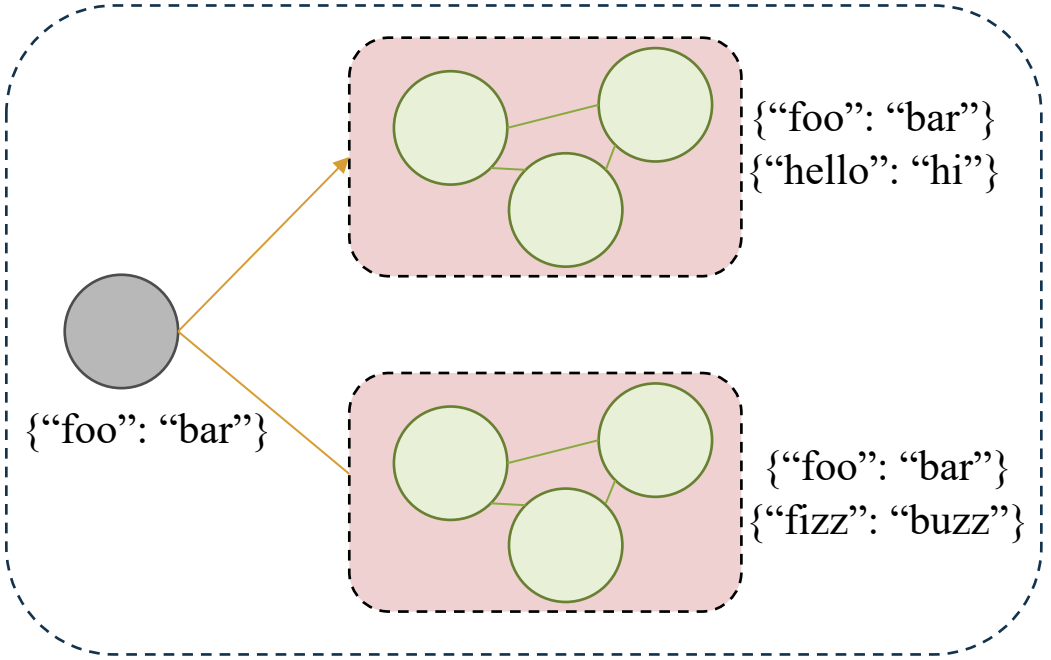


Multi-agent Systems

LangGraph

- **Controllability**: to define both explicit and implicit workflows
- Branches enable parallel execution of nodes to speed up overall graph operation
- Subgraphs enable complex system design by managing states separately

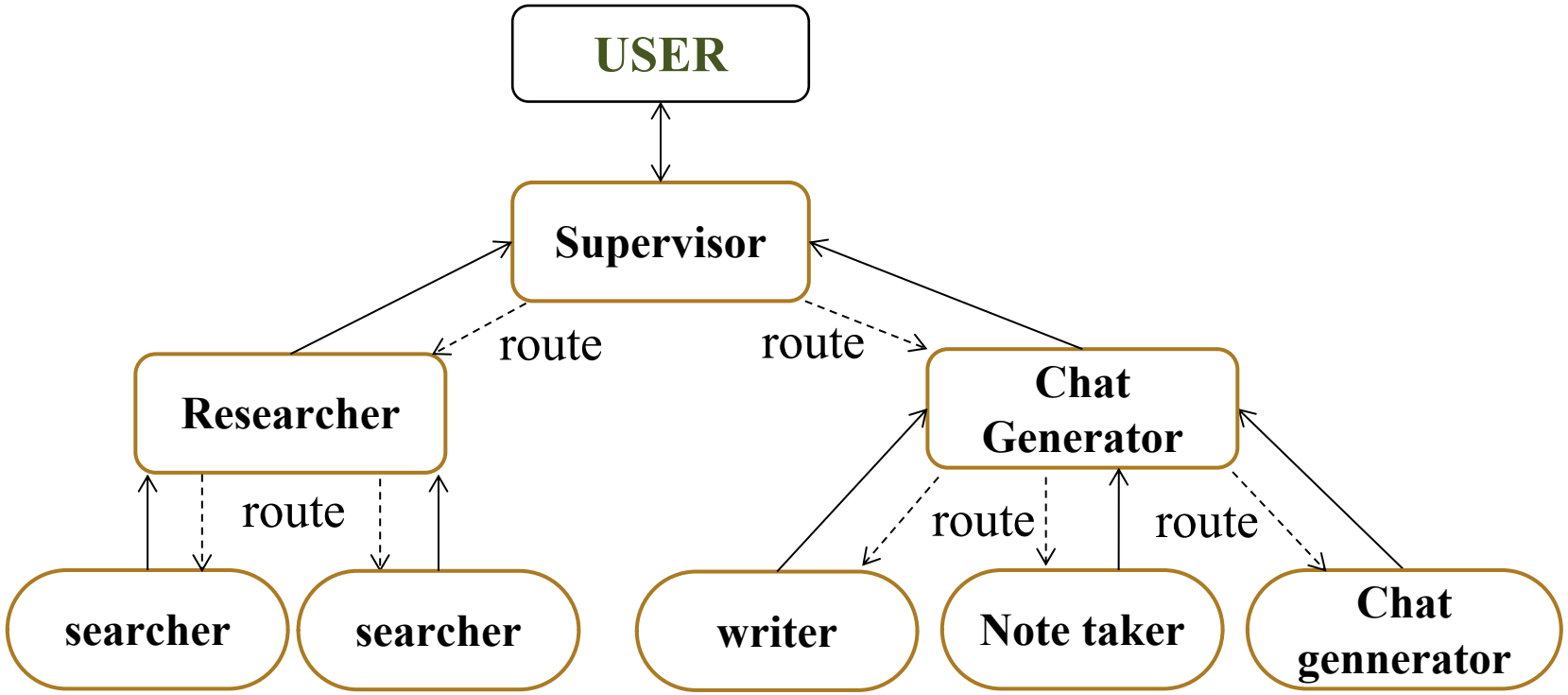




Multi-agent Systems

LangGraph

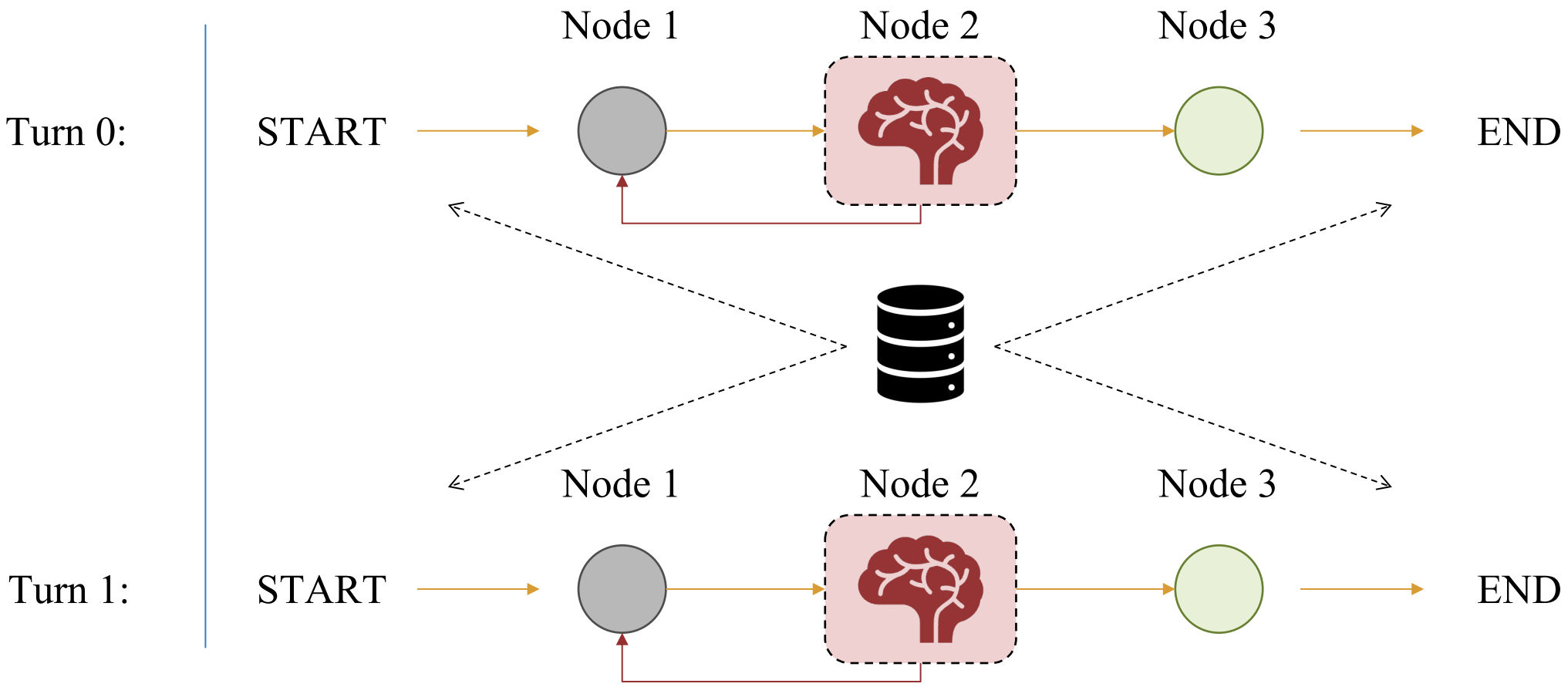
- **Controllability**: to define both explicit and implicit workflows
- Multi-agent Architectures: break the complex task into multiple smaller, independent agents



Multi-agent Systems

LangGraph

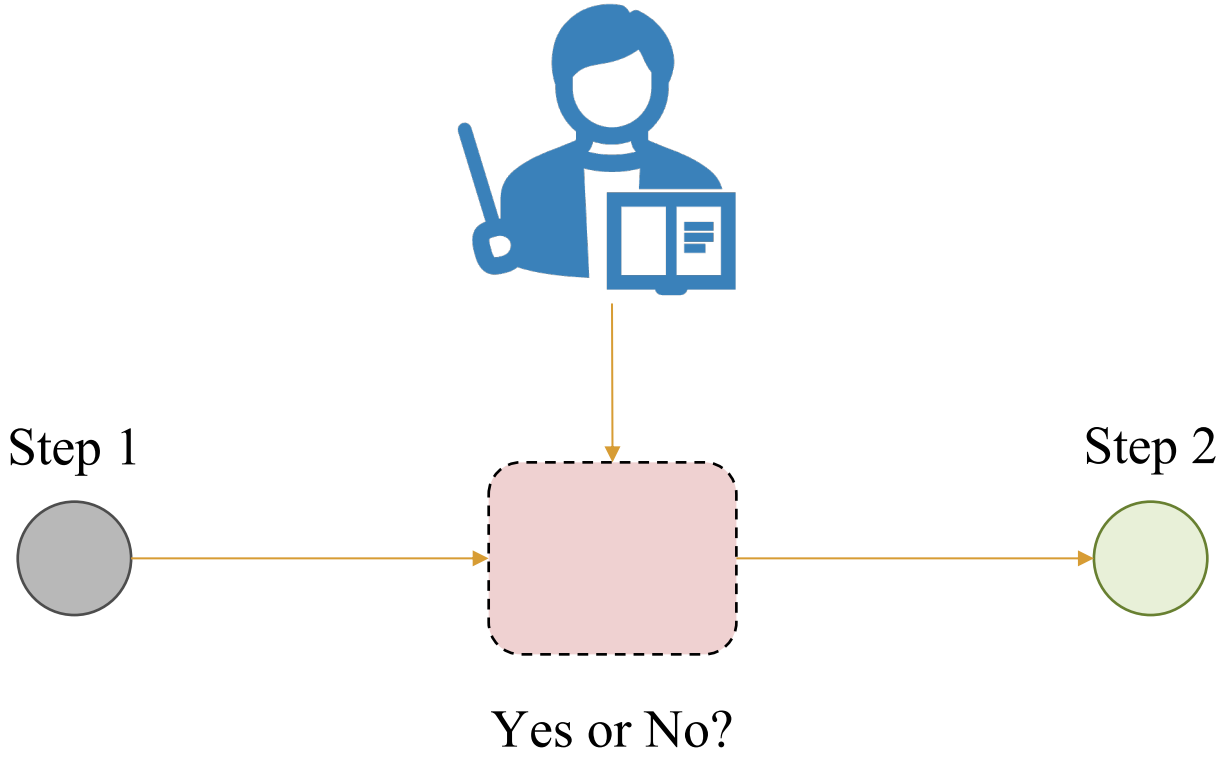
- **Persistence**: for thread-specific and cross-thread "Memory"



Multi-agent Systems

LangGraph

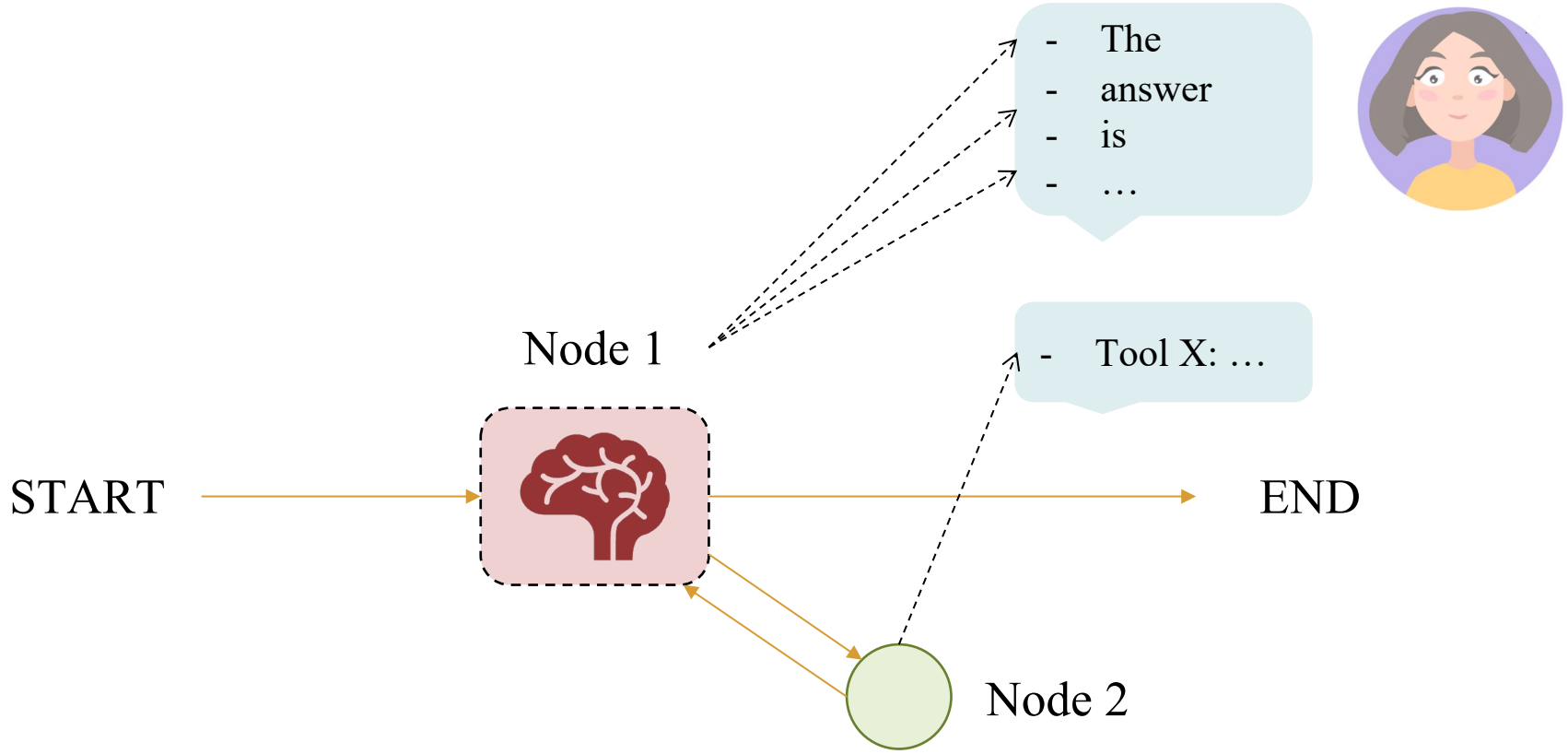
- **Interaction**: between agents, humans and external systems, breakpoints enable human-in-the-loop interactions.
- Key capabilities: Persistent execution state and Flexible integration points



Multi-agent Systems

LangGraph

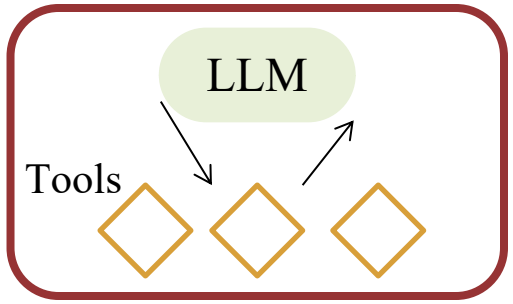
- **Streaming**: to expose any event (or token) as it occurs
- First-class support for token and event-level streaming



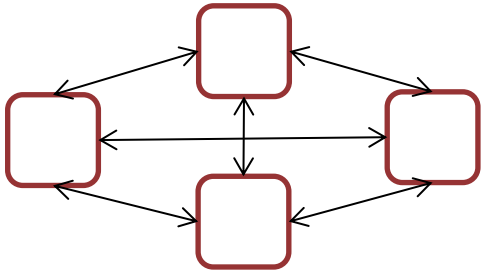
Multi-agent Systems

Multi-agent Architectures

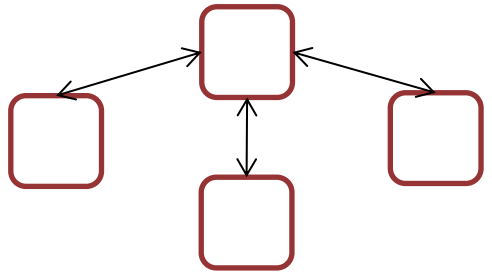
Single Agent



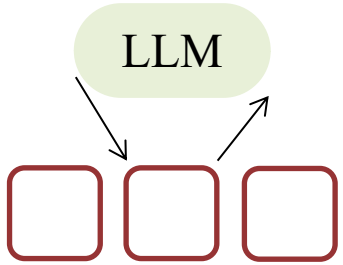
Network



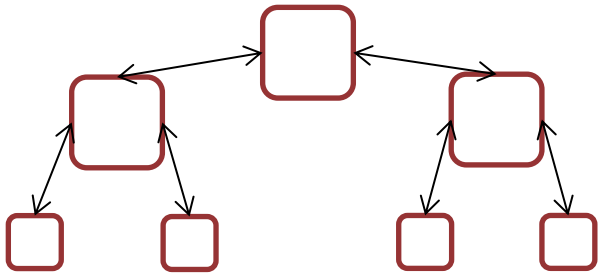
Supervisor



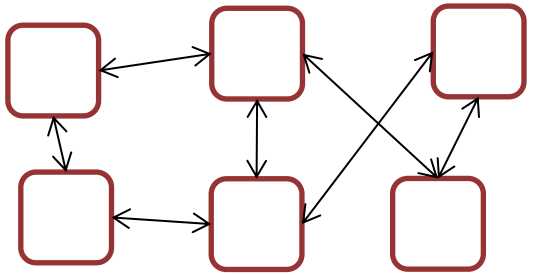
Supervisor (as tools)



Hierarchical



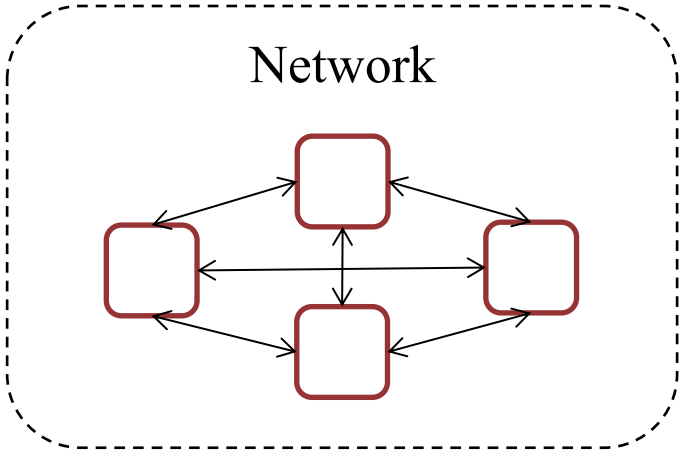
Custom

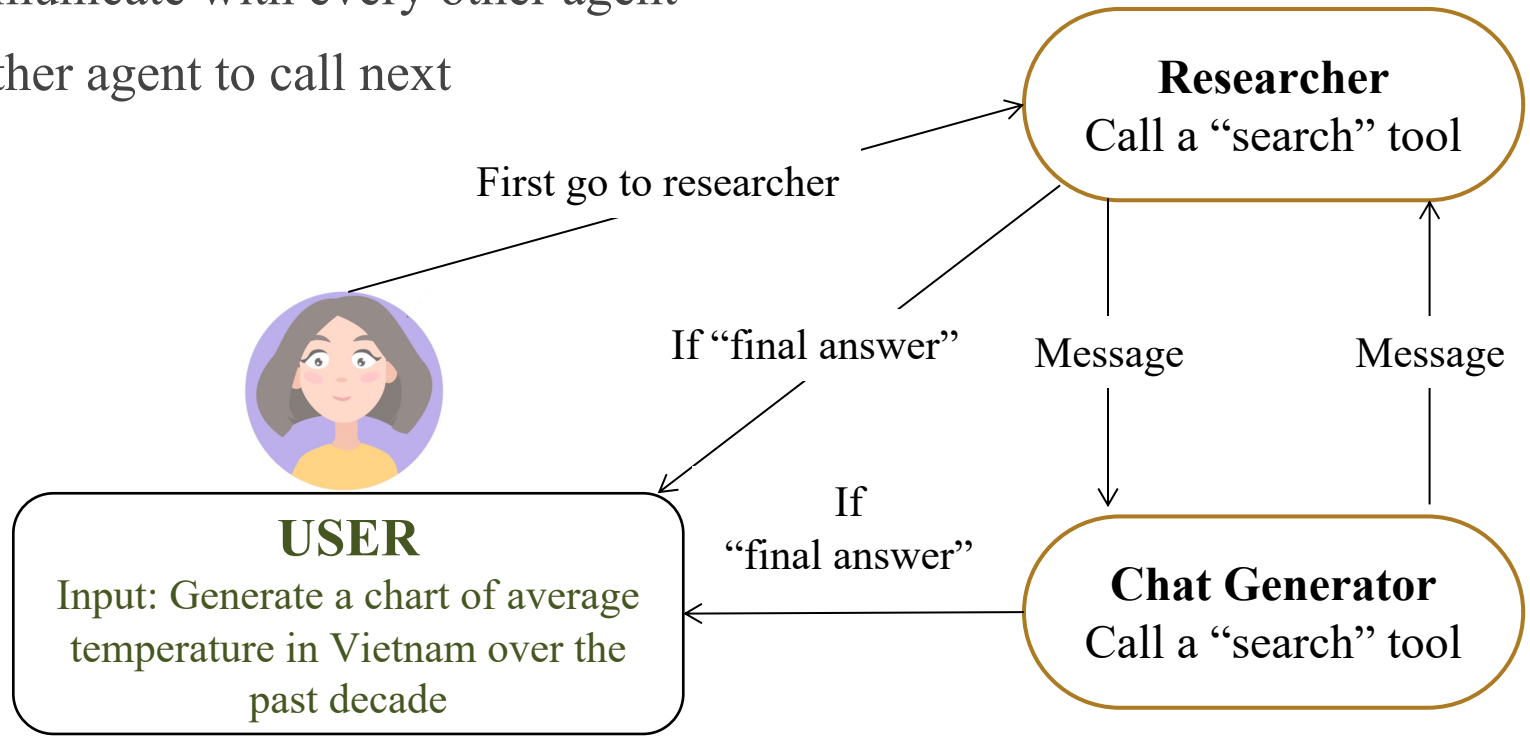


Multi-agent Systems

Multi-agent Architectures

- **Network**: each agent can communicate with every other agent
- Any agent can decide which other agent to call next



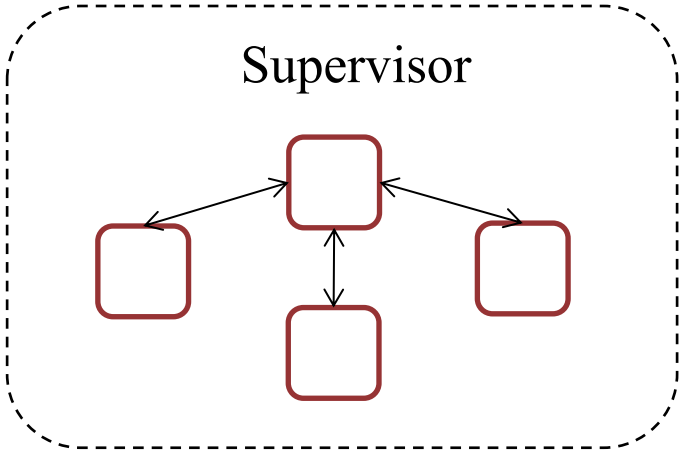


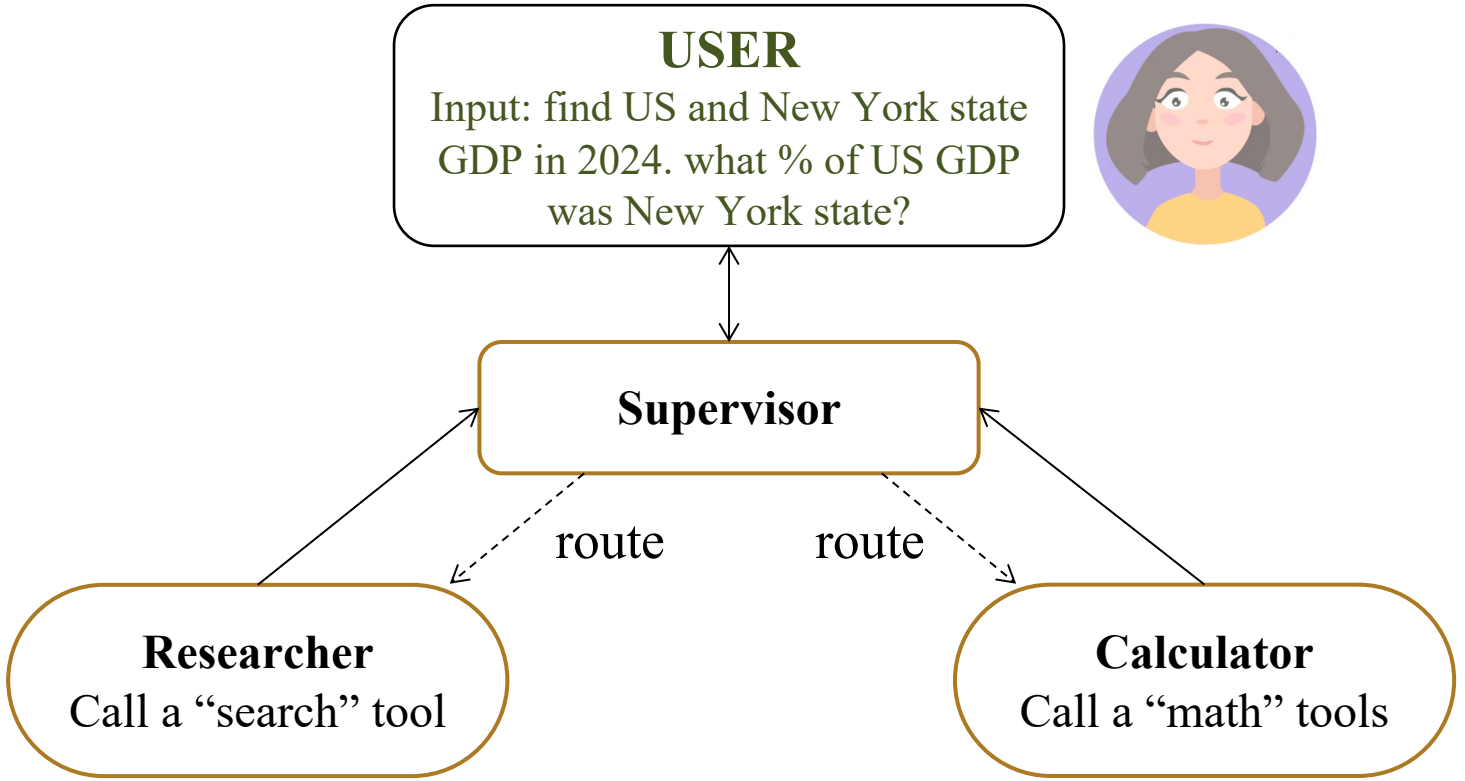
AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation

Multi-agent Systems

Multi-agent Architectures

- **Supervisor**: each agent communicates with a single supervisor agent
- Supervisor agent makes decisions on which agent should be called next

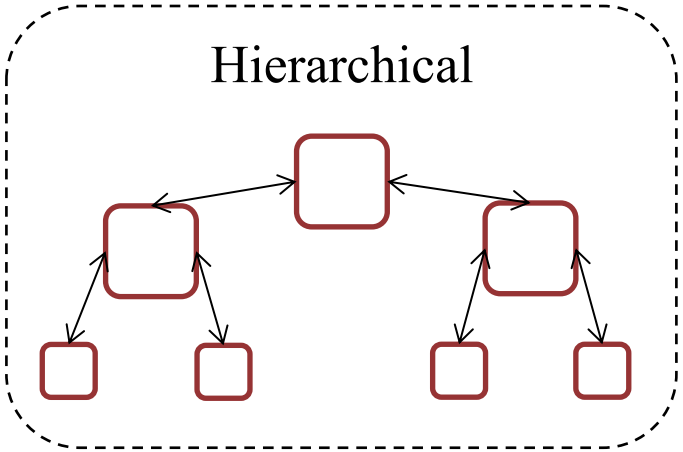


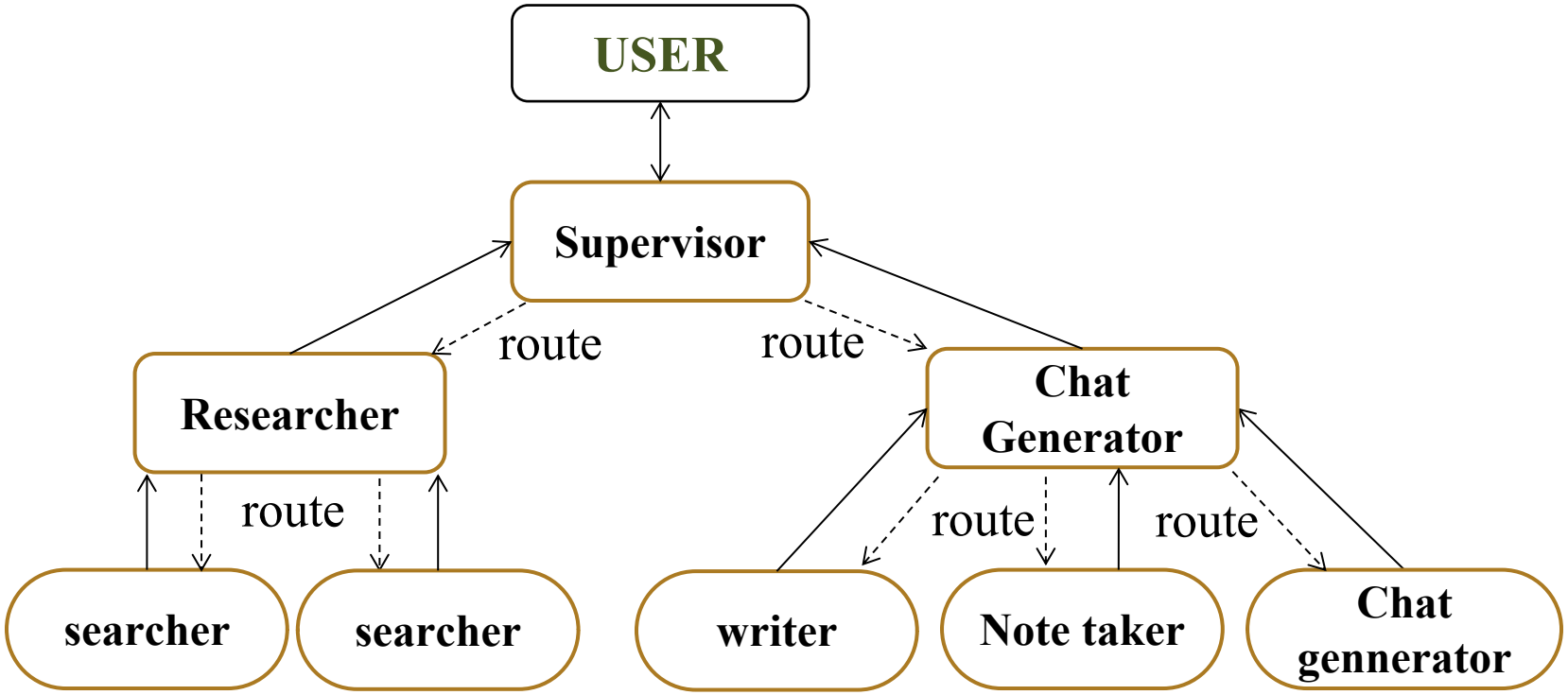


Multi-agent Systems

Multi-agent Architectures

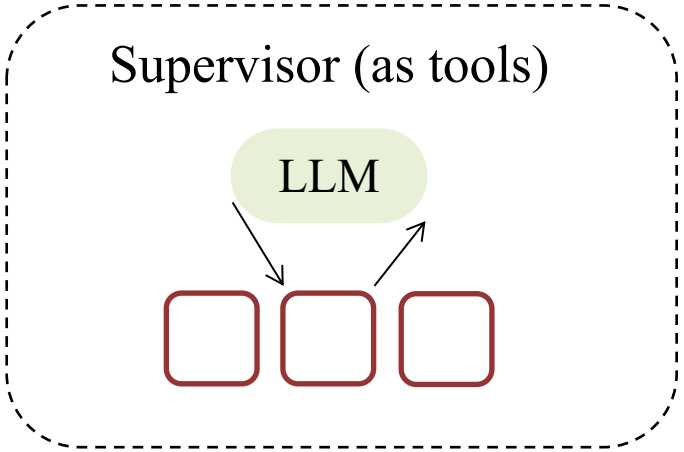
- **Hierarchical Agent Teams**: a generalization of the supervisor architecture and allows for more complex control flows: a top-level supervisor, along with mid-level supervisors.



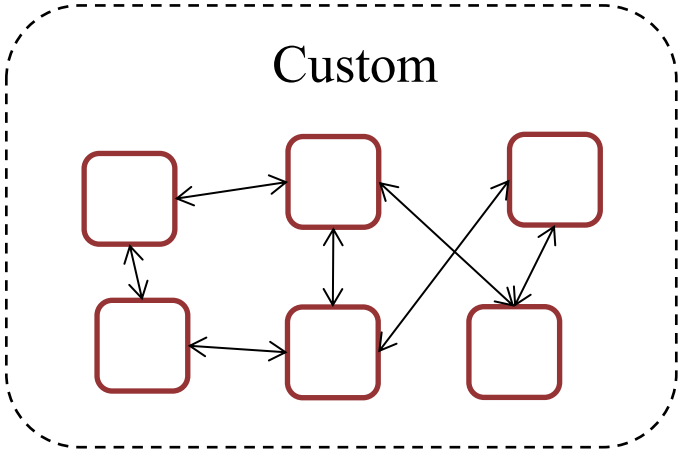


Multi-agent Systems

Multi-agent Architectures



Supervisor (tool-calling): this is a special case of supervisor architecture.
Individual agents can be represented as tools.
In this case, a supervisor agent uses a tool-calling LLM to decide which of the agent tools to call, as well as the arguments to pass to those agents.

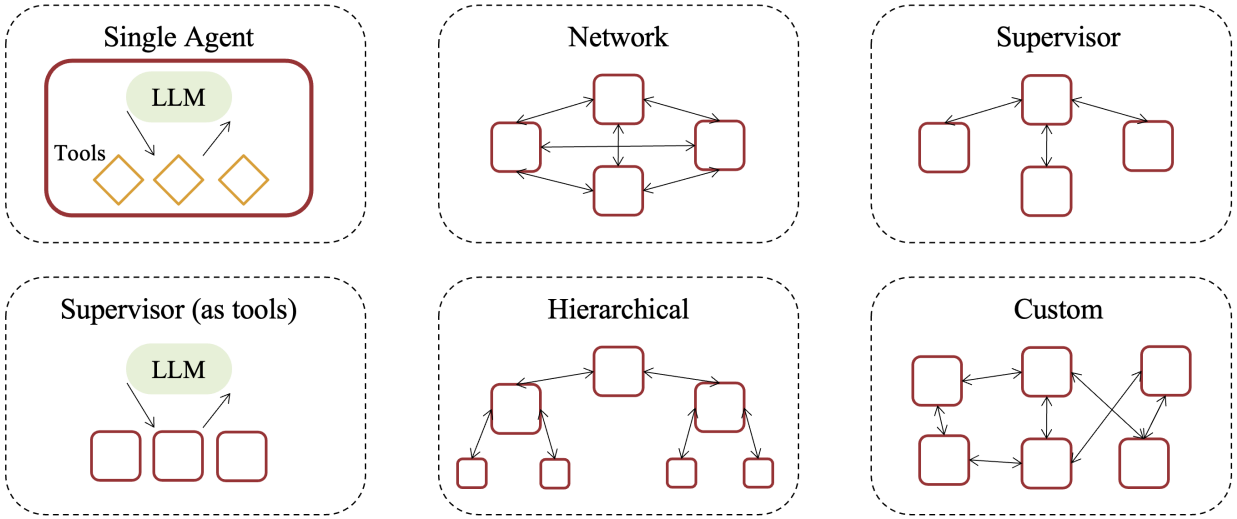


Custom multi-agent workflow: each agent communicates with only a subset of agents.
Parts of the flow are deterministic, and only some agents can decide which other agents to call next.

Outline

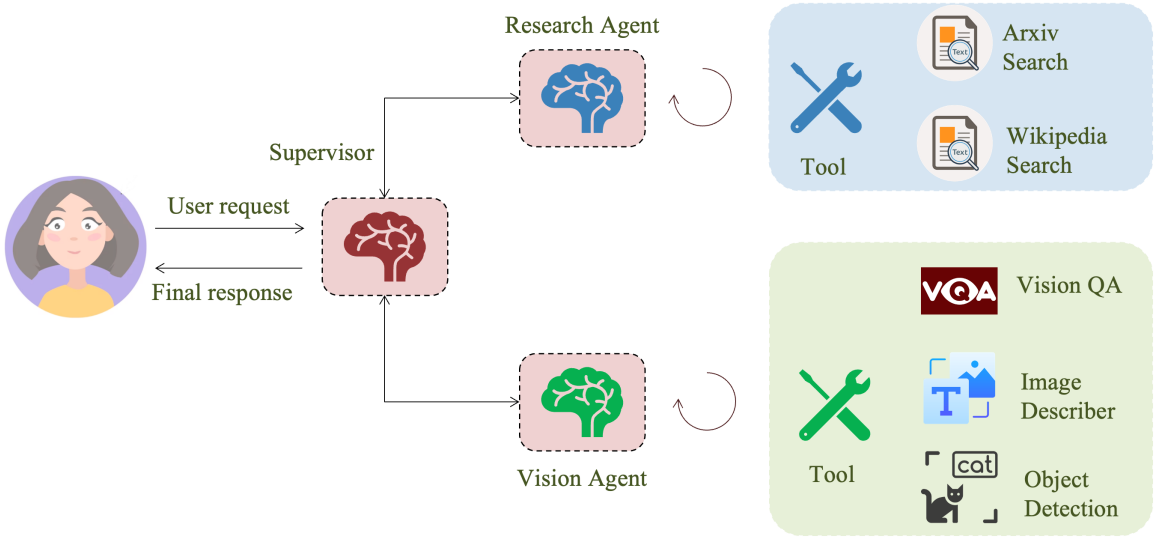
SECTION 1

Multi-agent Systems



SECTION 2

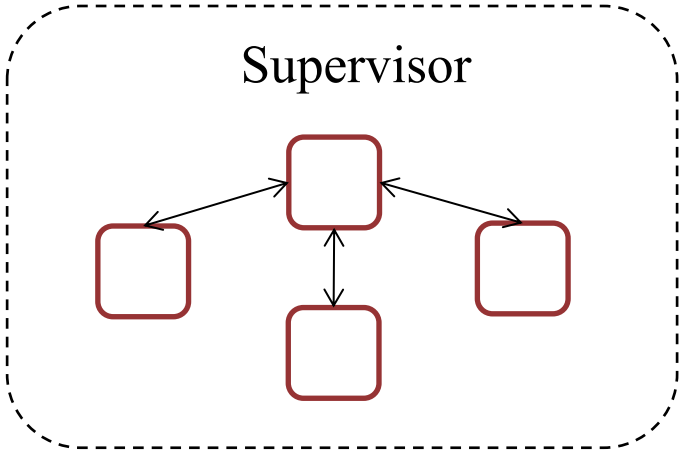
Visual Agentic AI

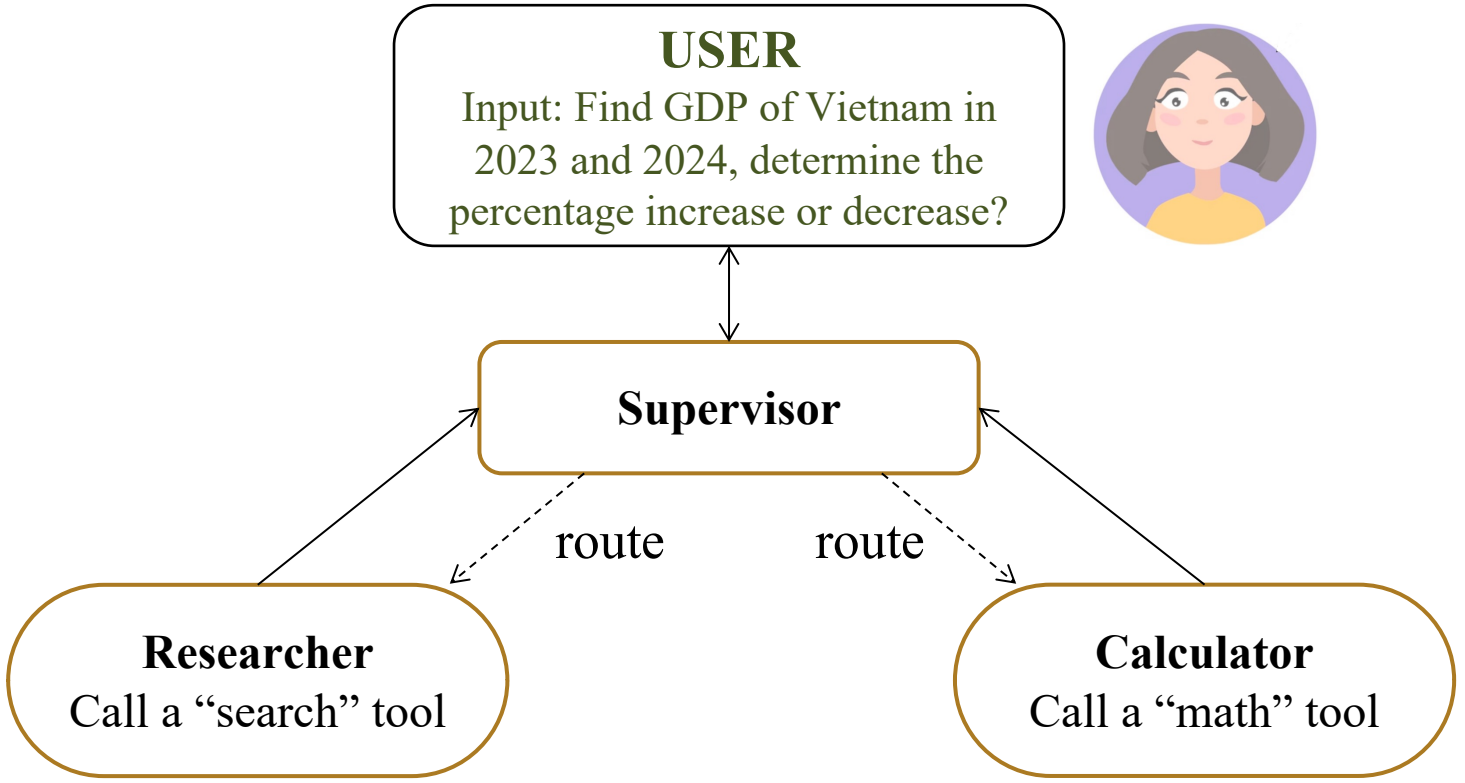


Multi-agent Supervisor

Multi-agent Supervisor

- **Supervisor**: each agent communicates with a single supervisor agent
- Supervisor agent makes decisions on which agent should be called next



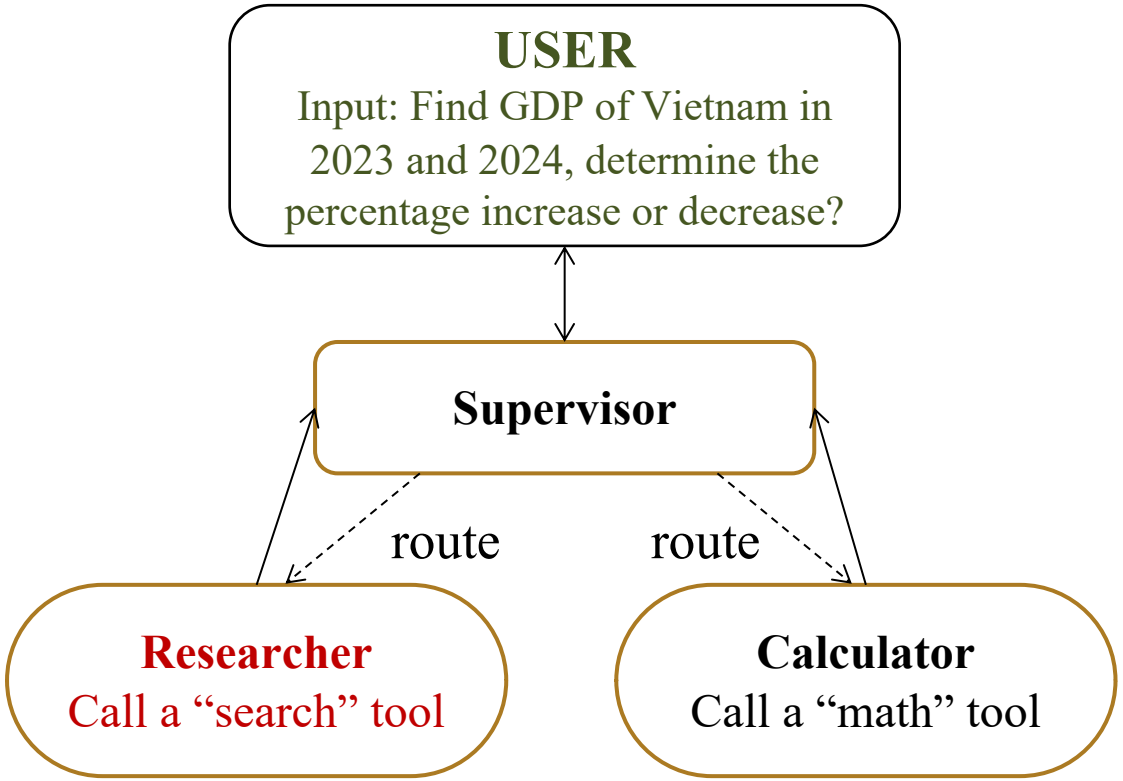


Multi-agent Supervisor

Research Agent

```
from langchain_tavily import TavilySearch

web_search = TavilySearch(max_results=3)
```

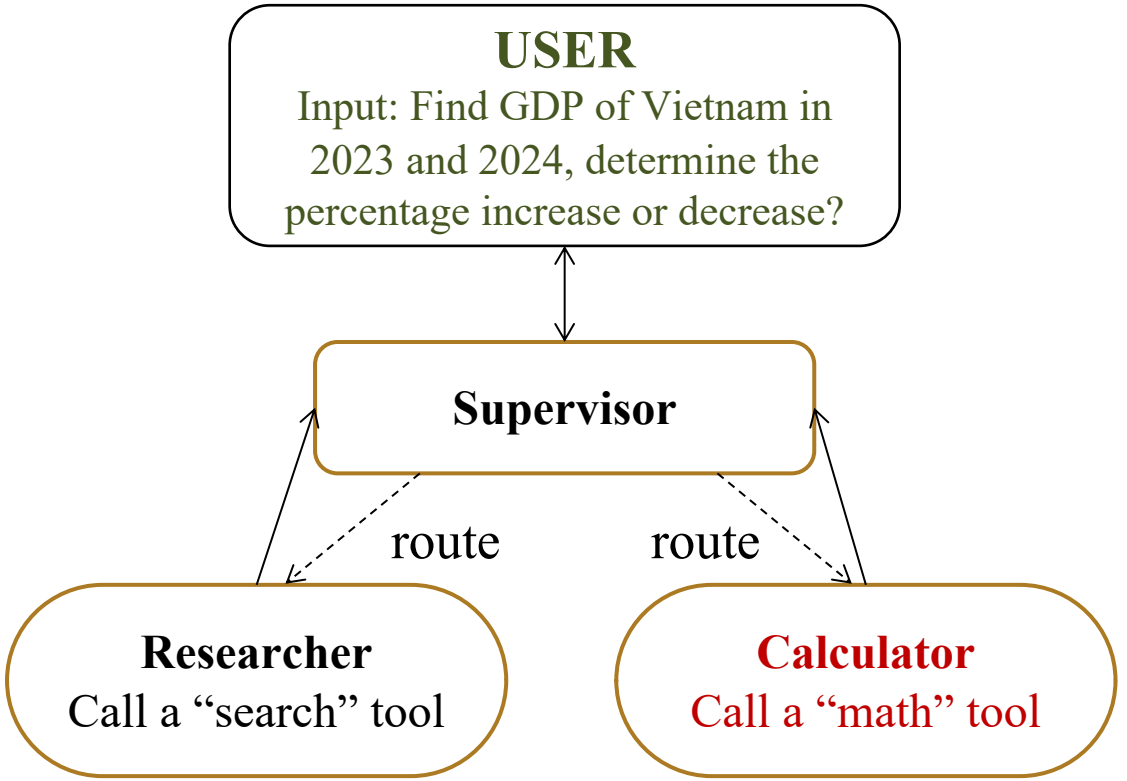
```
from langgraph.prebuilt import create_react_agent

research_agent = create_react_agent(
    model="gpt-4o-mini",
    tools=[web_search],
    prompt=(
        "You are a research agent.\n\n"
        "INSTRUCTIONS:\n"
        "- Assist ONLY with research-related tasks, D
        "- After you're done with your tasks, respond
        "- Respond ONLY with the results of your work
    ),
    name="research_agent",
)
```



Multi-agent Supervisor

Math Agent

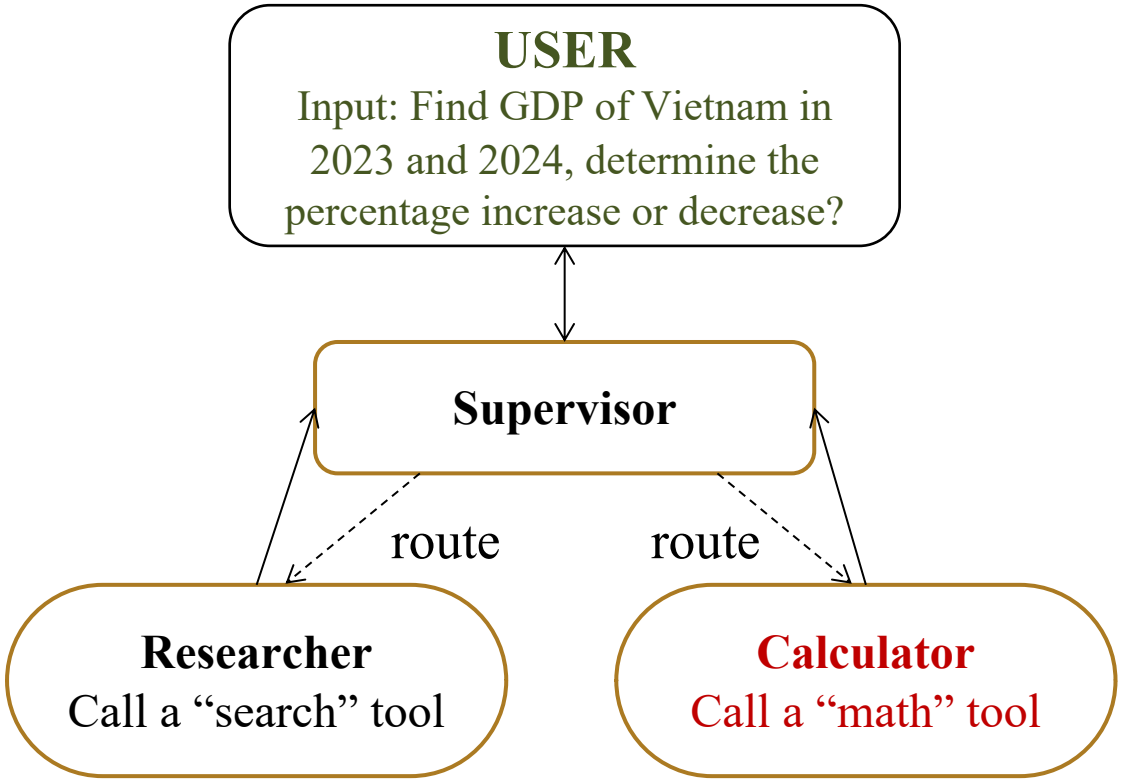
```python
def add(a: float, b: float):
    """Add two numbers."""
    return a + b


def multiply(a: float, b: float):
    """Multiply two numbers."""
    return a * b


def divide(a: float, b: float):
    """Divide two numbers."""
    return a / b
```



Multi-agent Supervisor

Math Agent

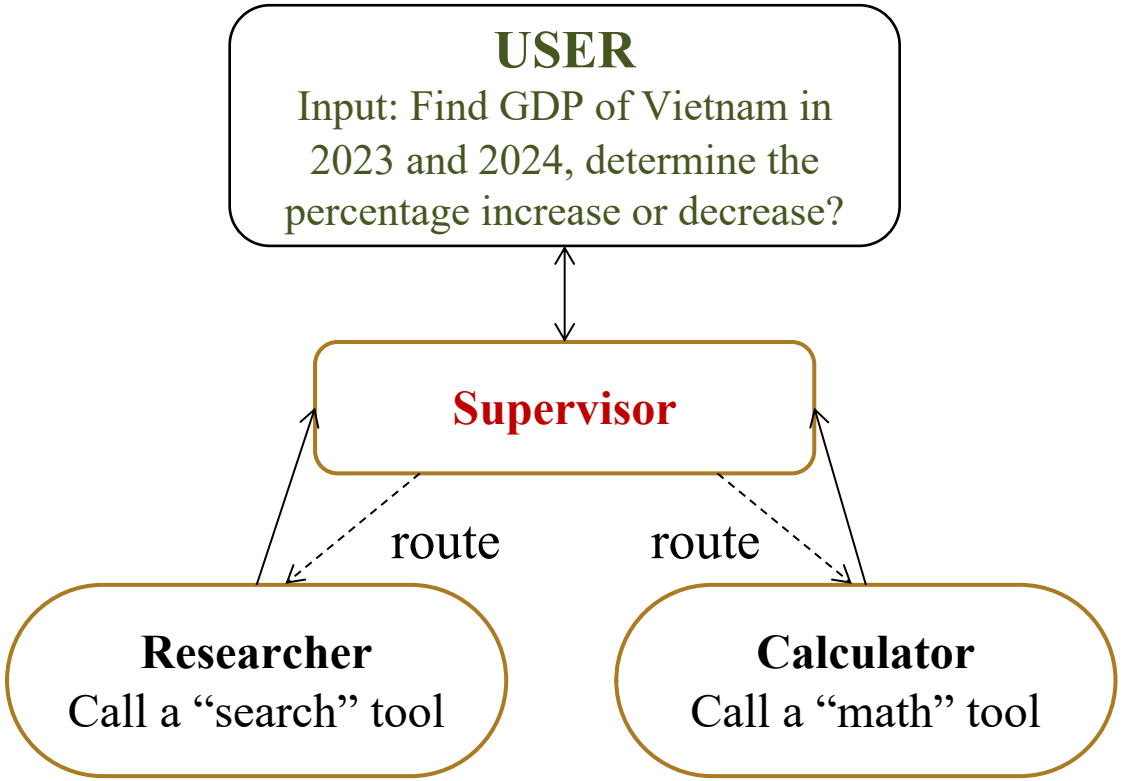
```
math_agent = create_react_agent(
    model="openai:gpt-4.1",
    tools=[add, multiply, divide],
    prompt=(
        "You are a math agent.\n\n"
        "INSTRUCTIONS:\n"
        "- Assist ONLY with math-related tasks\n"
        "- After you're done with your tasks, resp
        "- Respond ONLY with the results of your w
    ),
    name="math_agent",
)
```



Multi-agent Supervisor

Supervisor

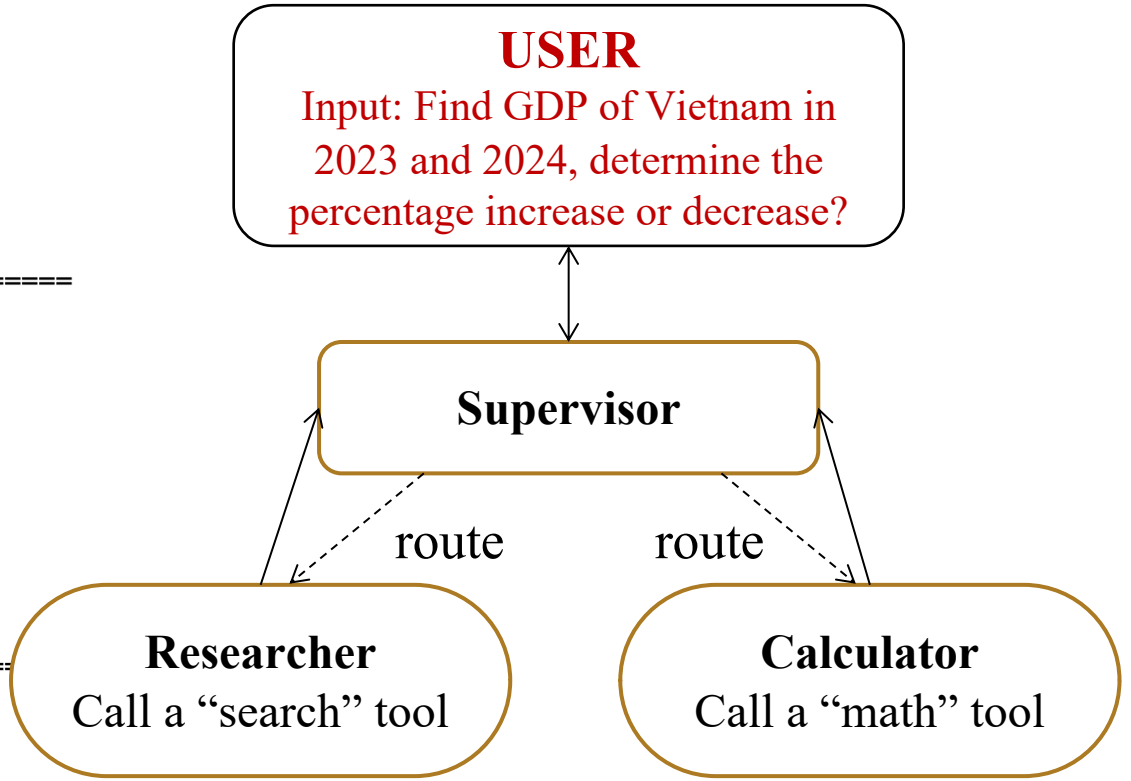
```python
from langgraph_supervisor import create_supervisor
from langchain.chat_models import init_chat_model

supervisor = create_supervisor(
    model=init_chat_model("gpt-4o"),
    agents=[research_agent, math_agent],
    prompt=(
        "You are a supervisor managing two agents:\n"
        "- a research agent. Assign research-related
        "- a math agent. Assign math-related tasks to
        "Assign work to one agent at a time, do not c
        "Do not do any work yourself."
    ),
    add_handoff_back_messages=True,
    output_mode="full_history",
).compile()
```



Multi-agent Supervisor

Test case

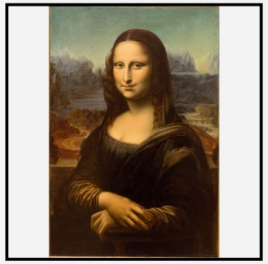
```
================================ Tool Message ================================
Name: transfer_to_research_agent

Successfully transferred to research_agent


Update from node research_agent:


================================ Tool Message ================================
Name: transfer_back_to_supervisor

Successfully transferred back to supervisor


Update from node supervisor:


================================ Tool Message ================================
Name: transfer_to_math_agent

Successfully transferred to math_agent
...

Vietnam's GDP grew from $429.72 billion in 2023 to $476.3 billion in 2024, marking an approximate increase of 10.84%.
```




QUIZ TIME

Visual Agent AI

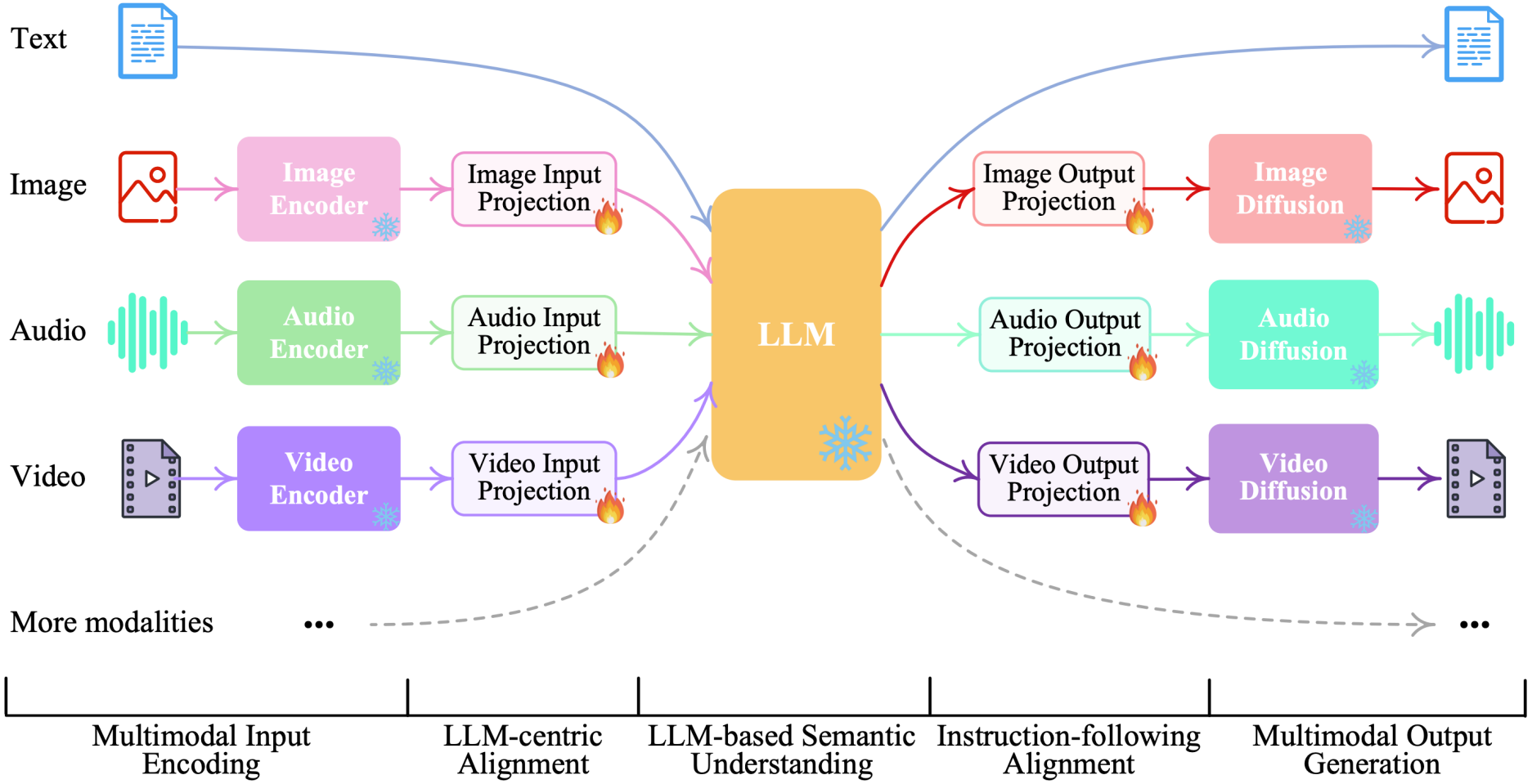
Multimodal LLMs

Input Prompt							Completion
	This is a chinchilla. They are mainly found in Chile.		This is a shiba. They are very popular in Japan.		This is	→	**a flamingo. They are found in the Caribbean and South America.**
	What is the title of this painting? Answer: The Hallucinogenic Toreador.		Where is this painting displayed? Answer: Louvres Museum, Paris.		What is the name of the city where this was painted? Answer:	→	**Arles.**
	Output: "Underground"		Output: "Congress"		Output:	→	**"Soulomes"**

Flamingo: a Visual Language Model for Few-Shot Learning

Visual Agent AI

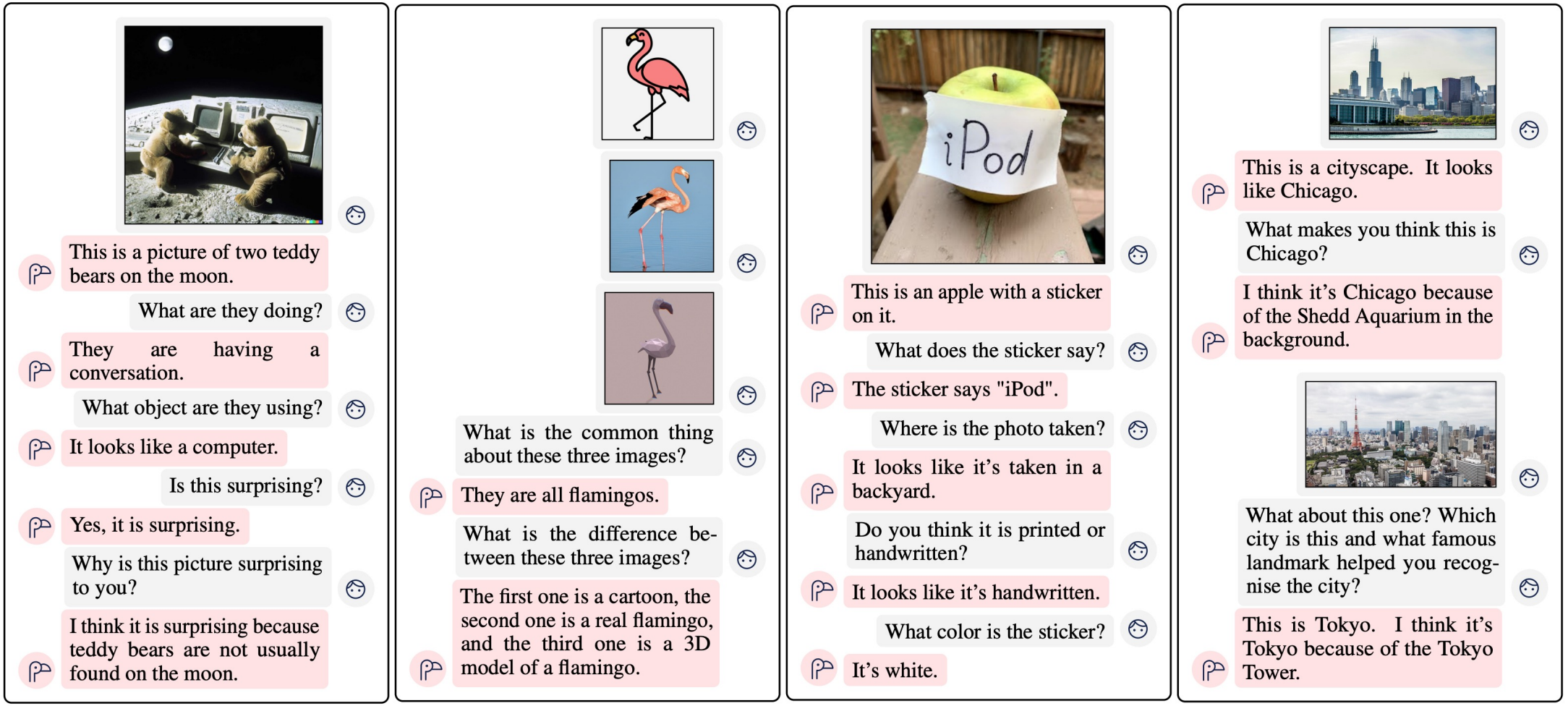
Multimodal LLMs



NExT-GPT: Any-to-Any Multimodal LLM

Visual Agent AI

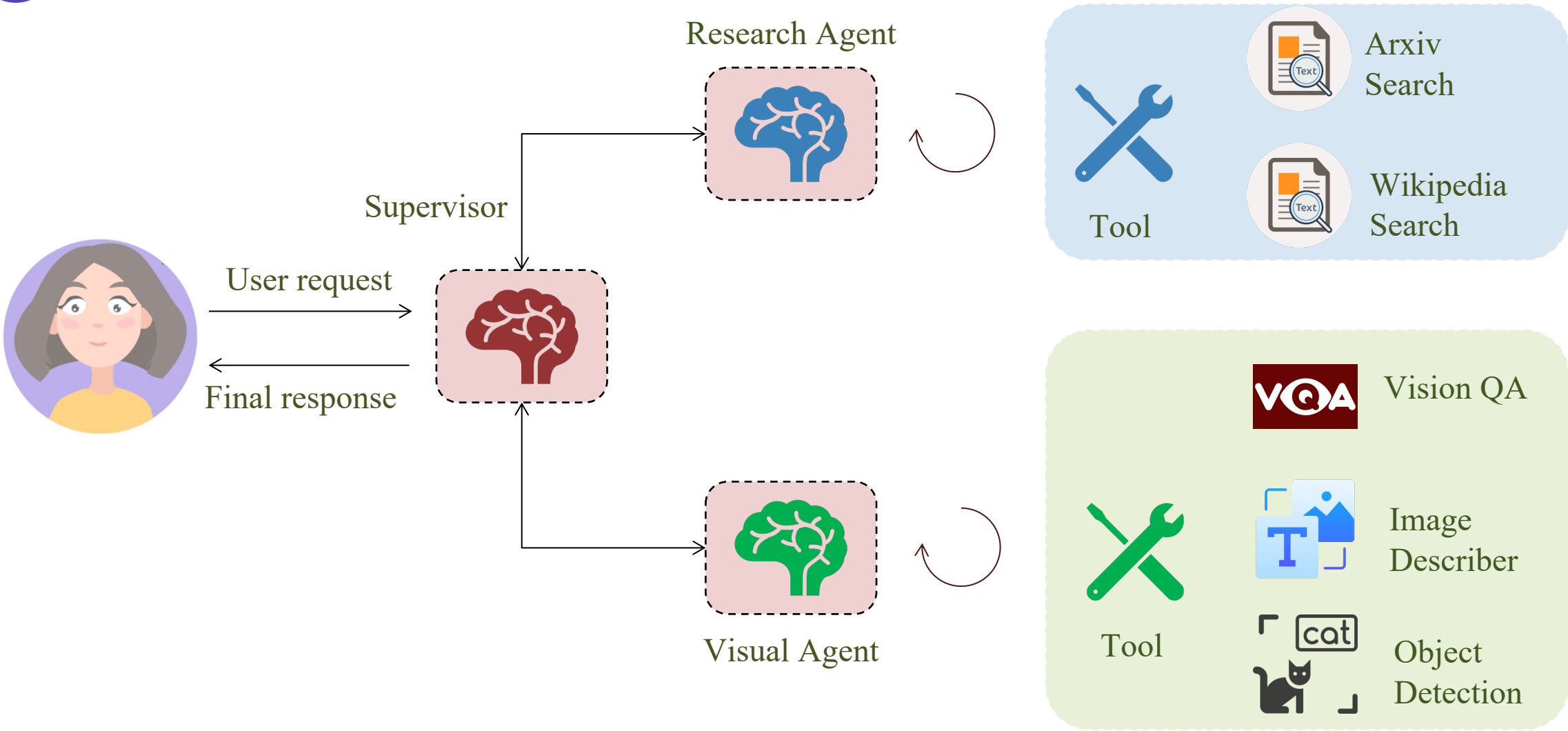
Multimodal Agent



Flamingo: a Visual Language Model for Few-Shot Learning

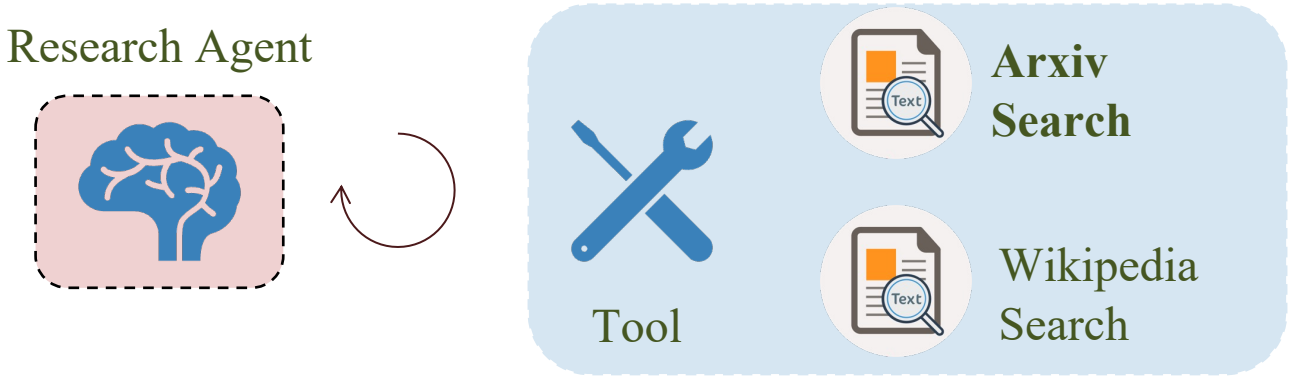
Visual Agent AI

Multi-agent Supervisor



Visual Agent AI

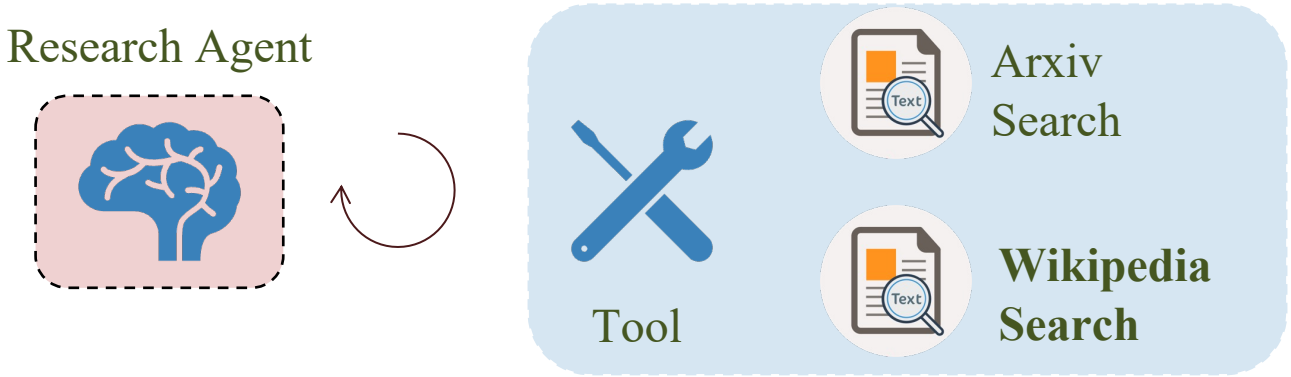
Research Agent




```
arxiv_wrapper = ArxivAPIWrapper(
    top_k_results=2, doc_content_chars_max=1000
)
arxiv = ArxivQueryRun(
    api_wrapper=arxiv_wrapper,
    description="Search for papers on a given topic using Arxiv"
)
arxiv.invoke("Rotary Positional Encoding")
```

Visual Agent AI

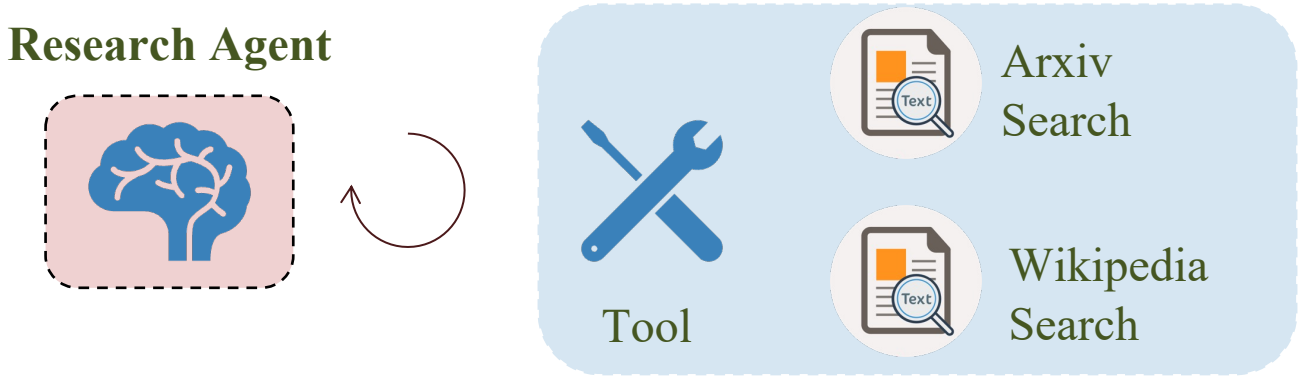
Research Agent




```
wikipedia_wrapper = WikipediaAPIWrapper()
wikipedia = WikipediaQueryRun(
    api_wrapper=wikipedia_wrapper,
    description="Search for information on a given topic using Wikipedia"
)
wikipedia.invoke("machine learning")
```

Visual Agent AI

Research Agent




```
research_agent = create_react_agent(
    model="gpt-4o-mini",
    tools=[arxiv, wikipedia],
    prompt=(
        "You are a research agent.\n\n"
        "INSTRUCTIONS:\n"
        "- Assist ONLY with research-related tasks, DO NOT do any math\n"
        "- After you're done with your tasks, respond to the supervisor directly\n"
        "- Respond ONLY with the results of your work, do NOT include ANY other text."
    ),
    name="research_agent",
)
```

Visual Agent AI

Visual Agent


```python
class ImageInput(BaseModel):
    image_path_or_url: str = Field(description="Image path or URL")

parser = PydanticOutputParser(pydantic_object=ImageInput)

prompt = PromptTemplate.from_template(
    "Extract the image path or URL from the following input:\n\n{input}\n\n{format_instructions}"
).partial(format_instructions=parser.get_format_instructions())

extractor_chain = prompt | llm | parser
```

Visual Agent AI

Visual Agent

Visual Agent

Tool

Image Describer

Vision QA

Object Detection

```python
def image_describer_prompt_func(inputs: dict):
    image_path_or_url = inputs["image_path_or_url"]
    image_b64, image_mime_type = encode_image(image_path_or_url, get_mime_type=True)

    image_describer_chat_template = ChatPromptTemplate.from_messages([
        SystemMessage(
            content="""You are an expert image describer."""),
        HumanMessage(content=[
            {"type": "text", "text": "Describe the following image for me:"},
            {
                "type": "image_url",
                "image_url": {"url": f"data:{image_mime_type};base64,{image_b64}", "detail": "low"}
            }
        ])
    ])
    return image_describer_chat_template.invoke({})
image_describer_agent = image_describer_prompt_func | llm.with_structured_output(ImageDescription)
```

Visual Agent AI

Visual Agent

Visual Agent

Image Describer

Vision QA

```python
class ImageDescriberInput(BaseModel):
    text: str = Field(description="Path or URL to the image in the format PNG or JPG/JPEG")

class ImageDescriberTool(BaseTool):
    name: str = "image_describer"
    description: str = "This tool can describe the image in a detailed way"
    args_schema: Optional[ArgsSchema] = ImageDescriberInput
    return_direct: bool = True

    def _run(self, text: str, run_manager: Optional[CallbackManagerForToolRun] = None) -> str:
        """Use the tool."""
        try:
            parsed: ImageInput = extractor_chain.invoke({"input": text})
        except Exception as e:
            return f"Failed to extract image URL: {str(e)}"

        image_path_or_url = parsed.image_path_or_url
        if not image_path_or_url:
            return "No image URL found in the input."
        output = image_describer_agent.invoke({"image_path_or_url": image_path_or_url})
        return output.image_description

    async def _arun(self, image_path_or_url: str, run_manager: Optional[AsyncCallbackManagerForToolRun] = None) -> str:
        """Use the tool asynchronously."""
        return self._run(image_path_or_url, run_manager=run_manager)
image_describer_tool = ImageDescriberTool()
```

Visual Agent AI

Visual Agent

Visual Agent

Tool

Image Describer

Vision QA

Object Detection

```
vision_agent = create_react_agent(
    model="gpt-4o-mini",
    tools=[image_describer_tool, detect_and_count_object_tool],
    prompt=(
        "You are a vision agent.\n\n"
        "INSTRUCTIONS:\n"
        "- Assist ONLY with visual tasks (e.g., describing images, detecting and counting objects)\n"
        "- Use only the tools provided to analyze visual inputs\n"
        "- After completing your task, respond to the supervisor directly\n"
        "- Respond ONLY with the results of your work, do NOT include ANY other text."
    ),
    name="vision_agent"
)
```

Visual Agent AI

Visual Agent

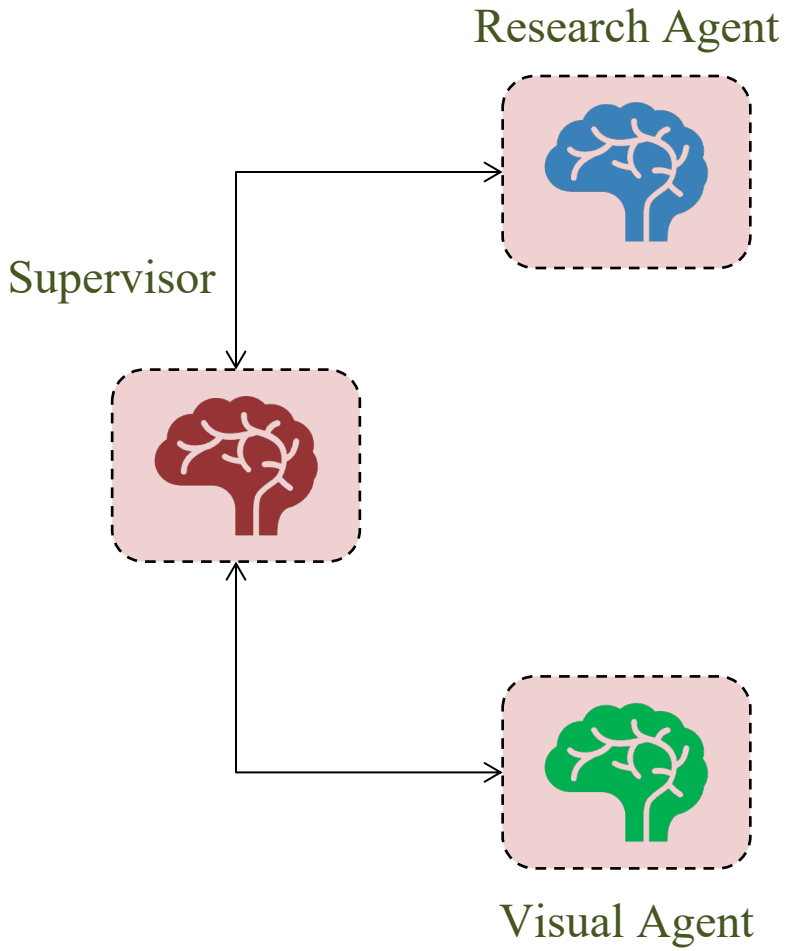
How many dogs in the image?

Visual Agent

```
================================== Ai Message ==================================
Name: vision_agent
Tool Calls:
  detect_and_count_objects (call_ZMQi8IvMsQU8nL3vZZP2aIPh)
 Call ID: call_ZMQi8IvMsQU8nL3vZZP2aIPh
  Args:
    text: https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/uploads/2018/


Found https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/uploads/2018/04/2
Update from node tools:


================================= Tool Message =================================
Name: detect_and_count_objects

{'counting': {'dog': 2}, 'detections': [{'class': 'dog', 'confidence': 0.941160023


Update from node agent:

...

2
```

Visual Agent AI

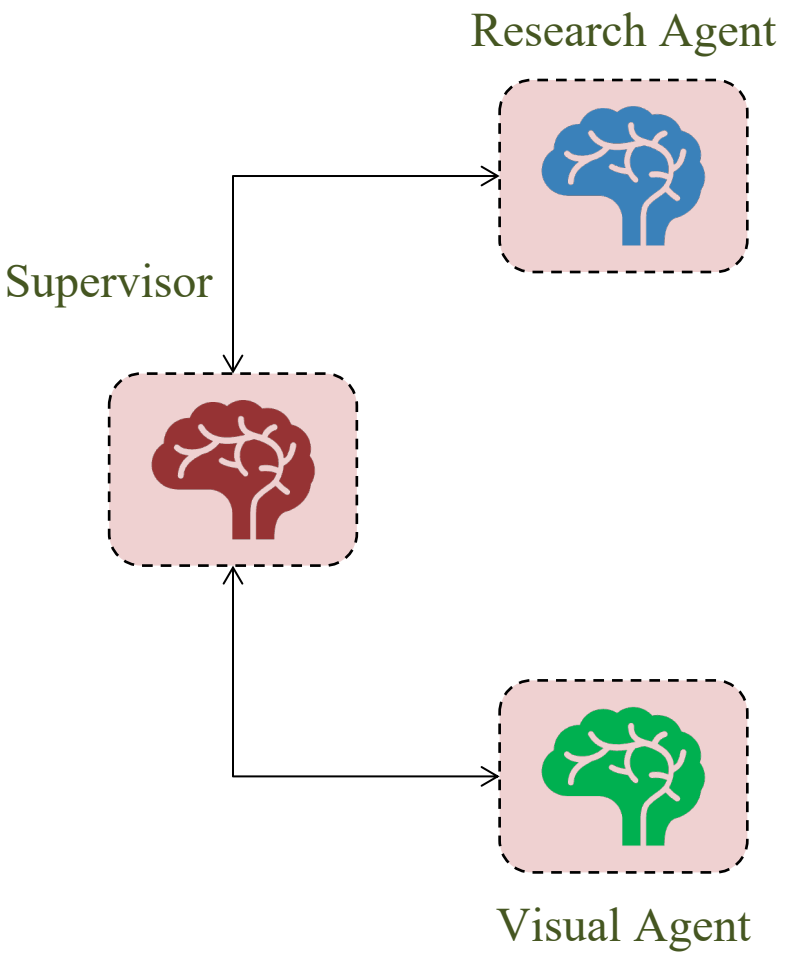
Supervisor Agent




```
supervisor = create_supervisor(
    model=init_chat_model("gpt-4o-mini"),
    agents=[research_agent, vision_agent],
    prompt=(
        "You are a supervisor managing two agents:\n"
        "- a research agent. Assign research-related
        "- a vision agent. Assist visual tasks (e.g.,
        "Assign work to one agent at a time, do not c
        "Do not do any work yourself."
    ),
    add_handoff_back_messages=True,
    output_mode="full_history",
).compile()
```

Visual Agent AI

Test Case




```
================================ Tool Message ================================
Name: transfer_to_vision_agent

Successfully transferred to vision_agent


Found https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/uploads/2018/04/2
Update from node vision_agent:


================================ Tool Message ================================
Name: transfer_back_to_supervisor

Successfully transferred back to supervisor


Update from node supervisor:


================================ Ai Message ================================
Name: supervisor

I have confirmed that there are 2 dogs in the image.
```

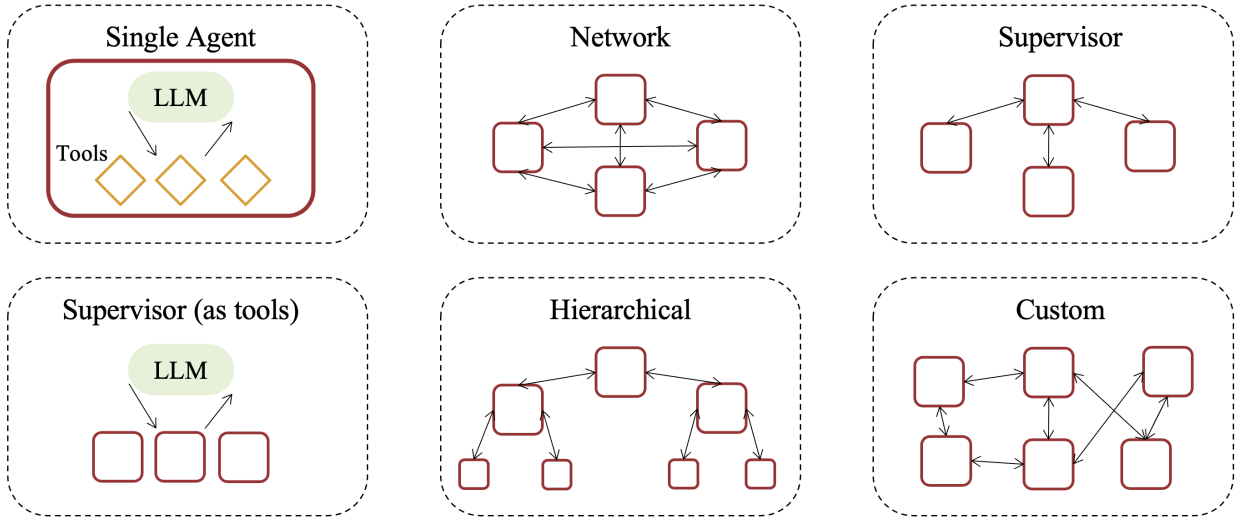
Objectives

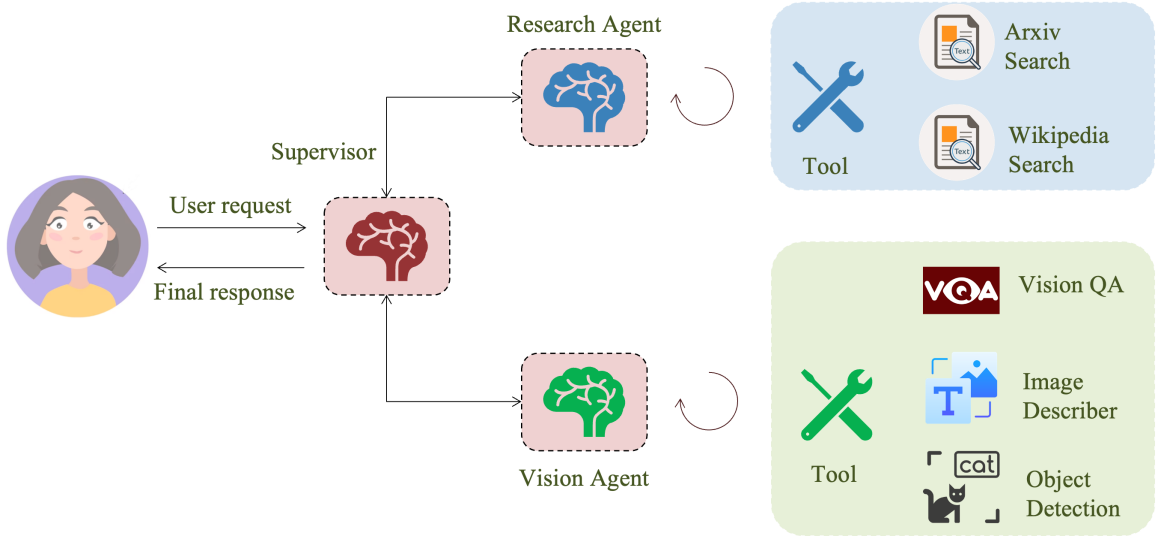
Multi-agent Systems

- AI Agent
- Multi-agent Systems
- Multi-agent using LangGraph

Visual Agentic AI

- Multi-agent Supervisor
- Vision Agent
- Research Agent





Thanks!

Any questions?