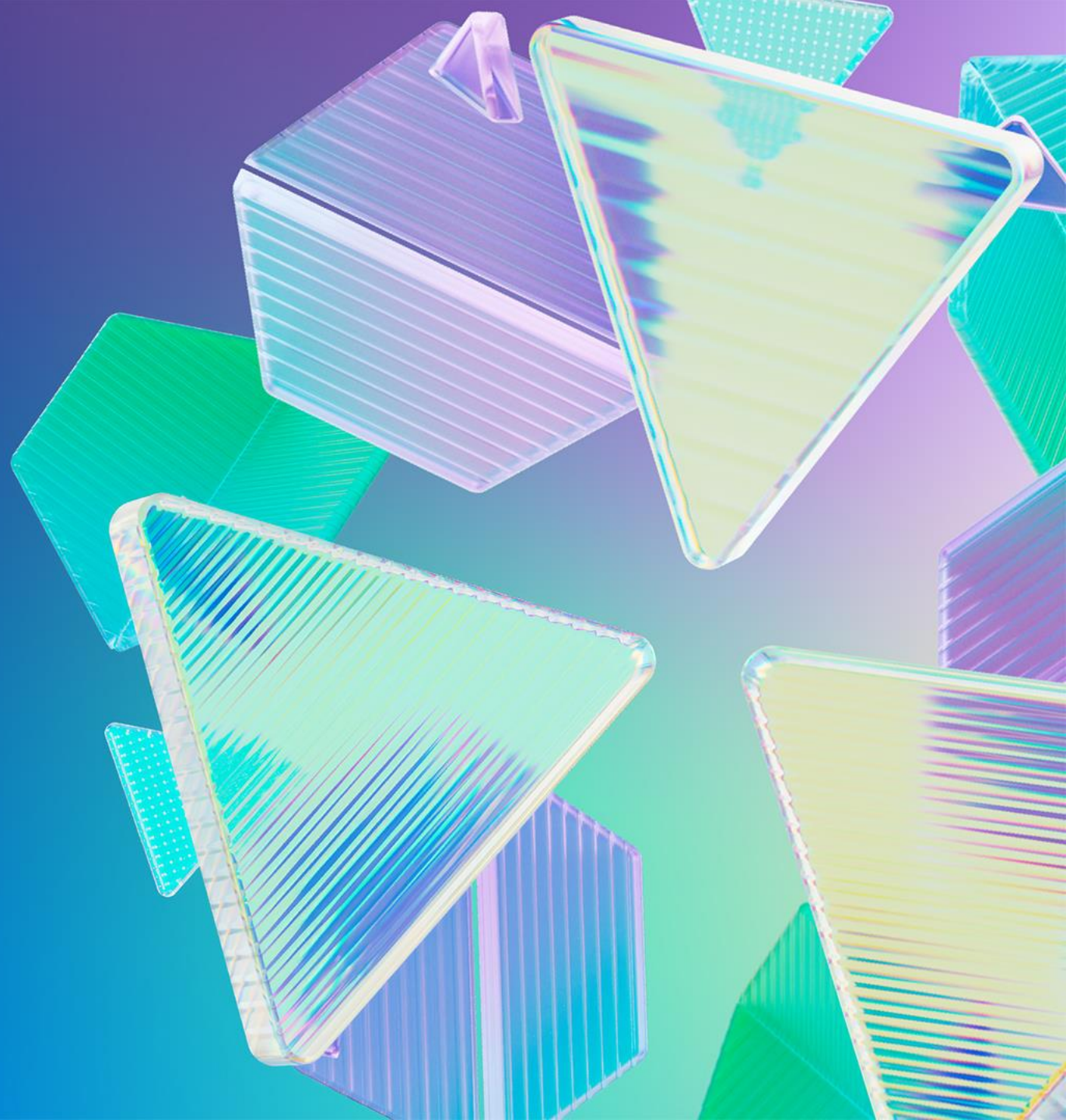
Microsoft

Exploring Agentic Systems and Multi-Agent Architectures

Arianit Sheholli - Cloud Solution Architect
Efe Sener - Cloud Solution Architect
Petar Petrov - Cloud Solution Architect



Slide Credits: MCAPS Tech Connect

Agenda

What is an Agent

Agentic building blocks and patterns

Single-Agent demo

Multi-Agent patterns

Multi-Agent demo

First wave of generative AI Apps

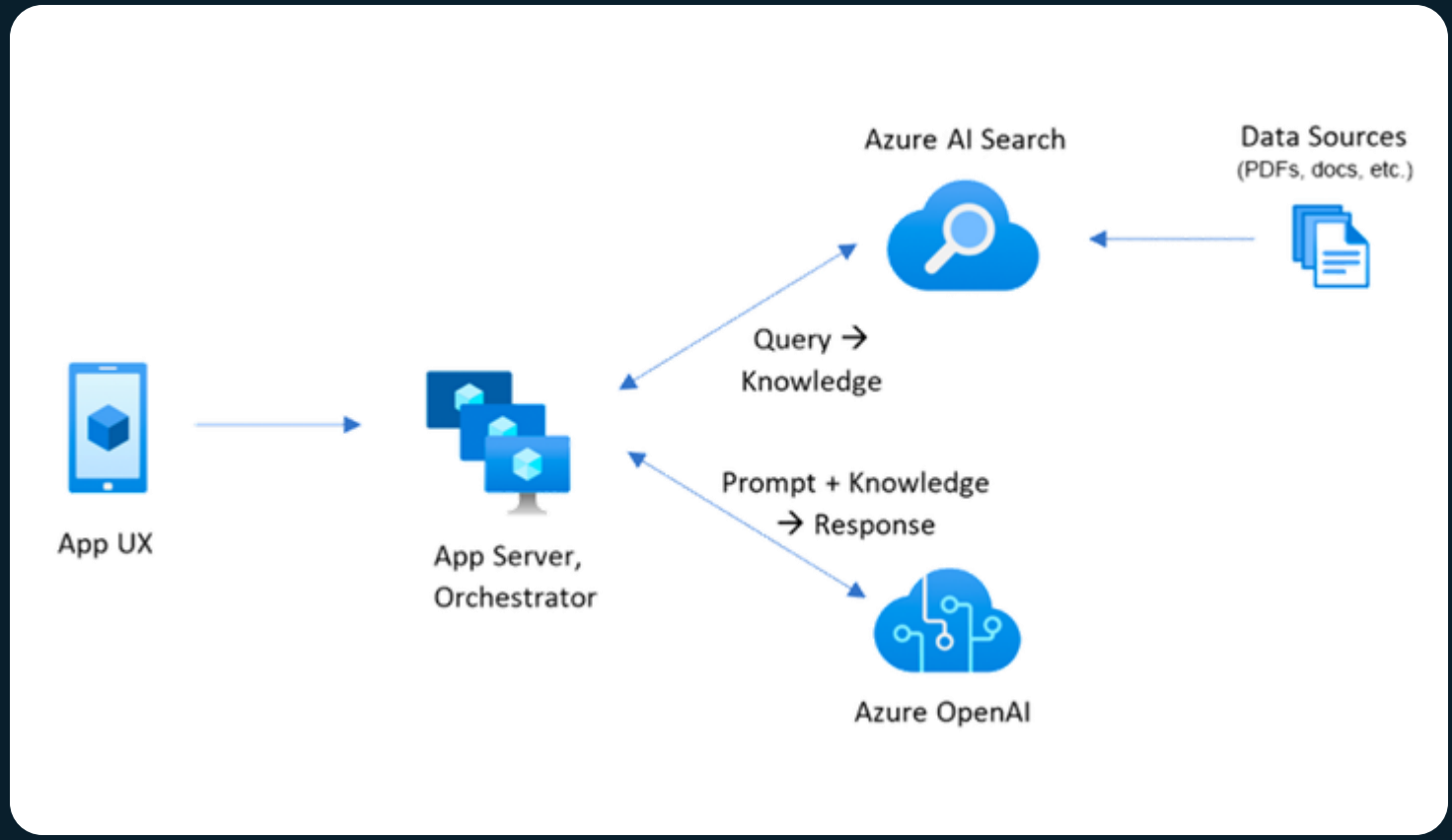
Common use cases:

- Conversational chat on private data
- Text/Document/Audio summarization and classification
- Image description and entity extractions
- Personalized content generation

Prompt engineering

RAG pattern

Application flow is hard-coded



Next wave: Agents

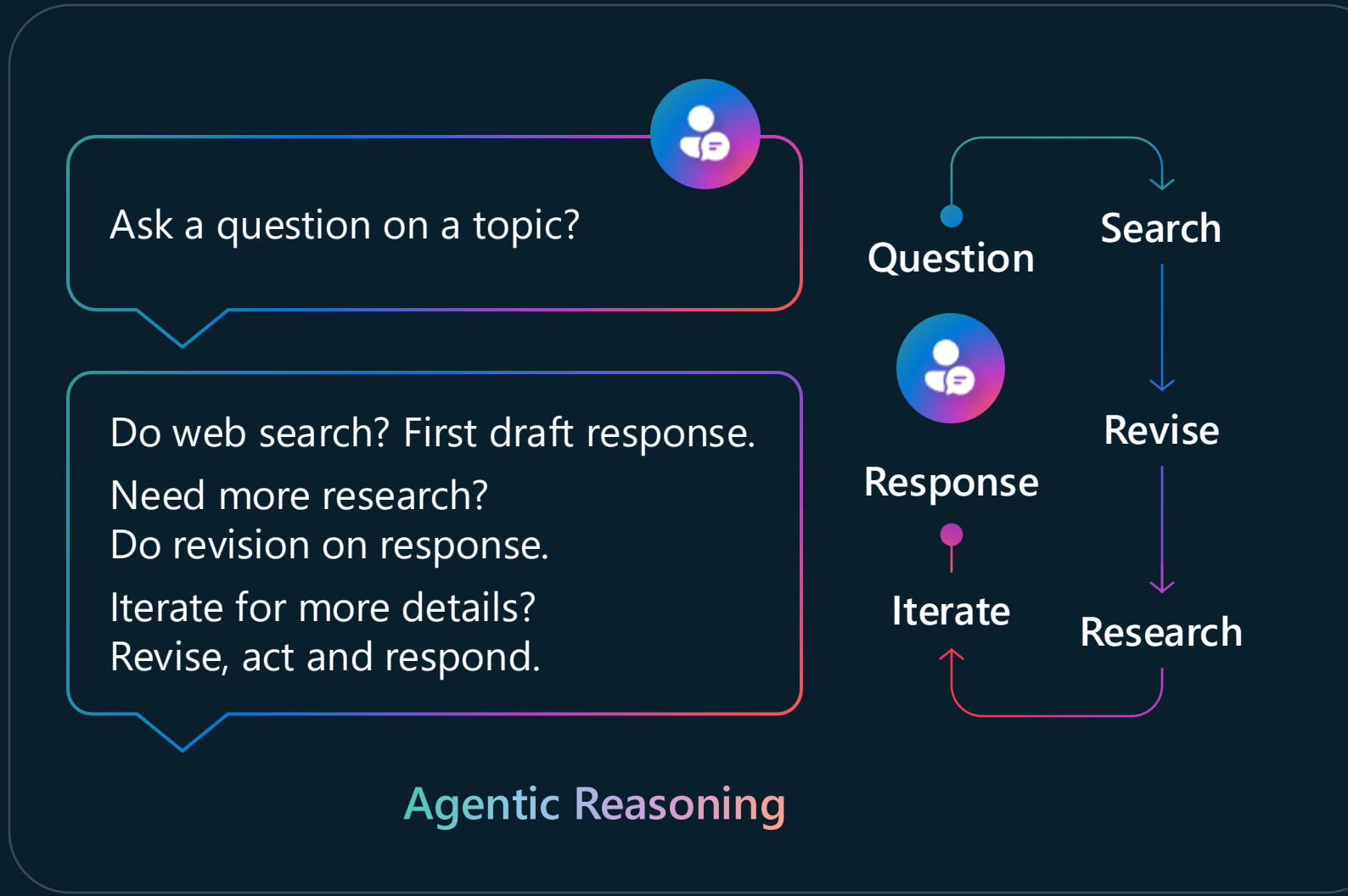
Complex interactions & orchestration

- Virtual assistants
- Customer support
- Intelligent code editors

Tools calling

Many LLM tasks + steps undefined sequence = agentic reasoning

Improve efficiency and accuracy



Agent frameworks and services

Semantic Kernel Agent Framework

Autogen

Langgraph

Azure AI Agent Service

What is an Agent?

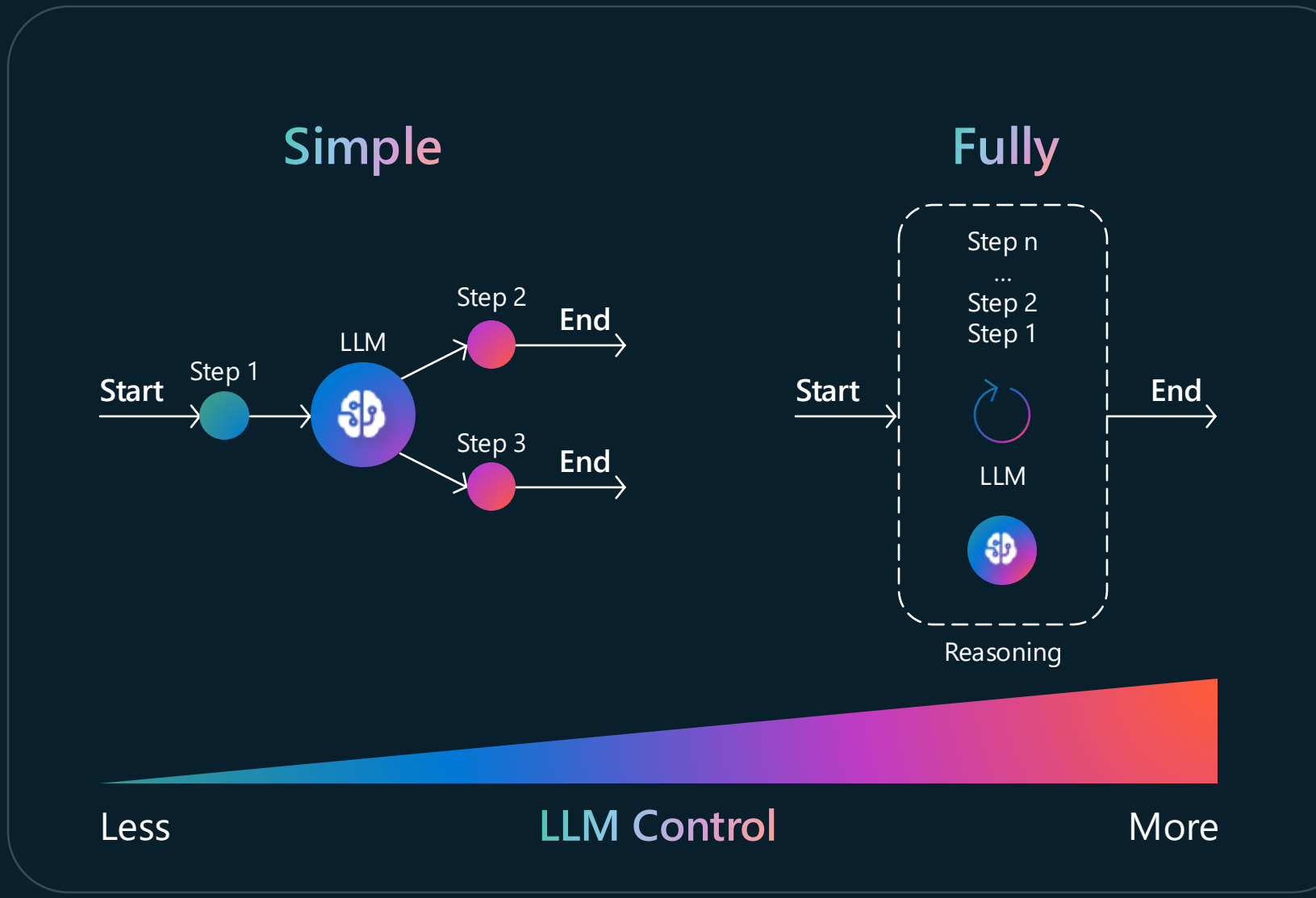
"System that uses a LLM to decide the control flow of an application."

Autonomy Levels:

- **No Autonomy:** Traditional RAG
- **Simple:** Paths routing
- **Fully:** Multi-step reasoning & acting

Architectures:

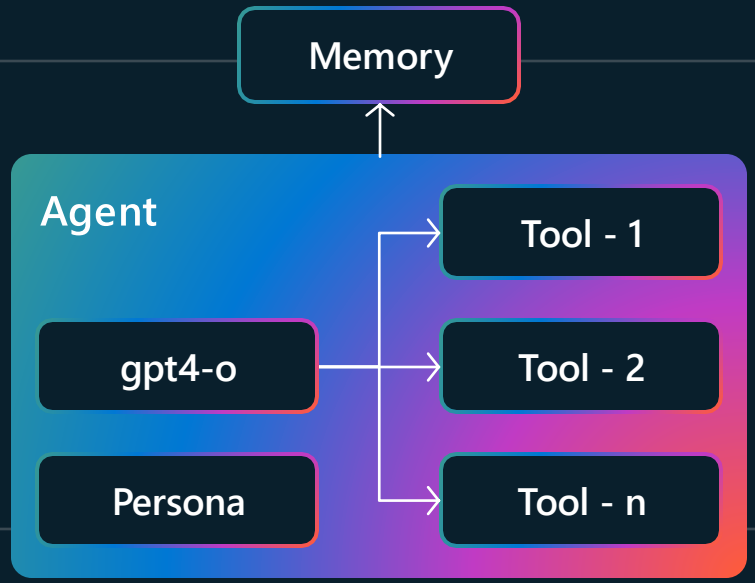
- Single agent
- Multi agent



Agent Abstractions - Agent First-Class Citizen

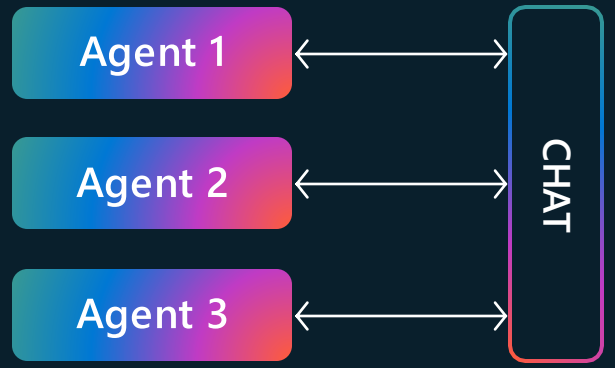
Agent as high-level abstraction

- **LLM** (gpt4-o, o1 etc.)
- **Persona** (system prompt)
- **Tools** (function code calls)

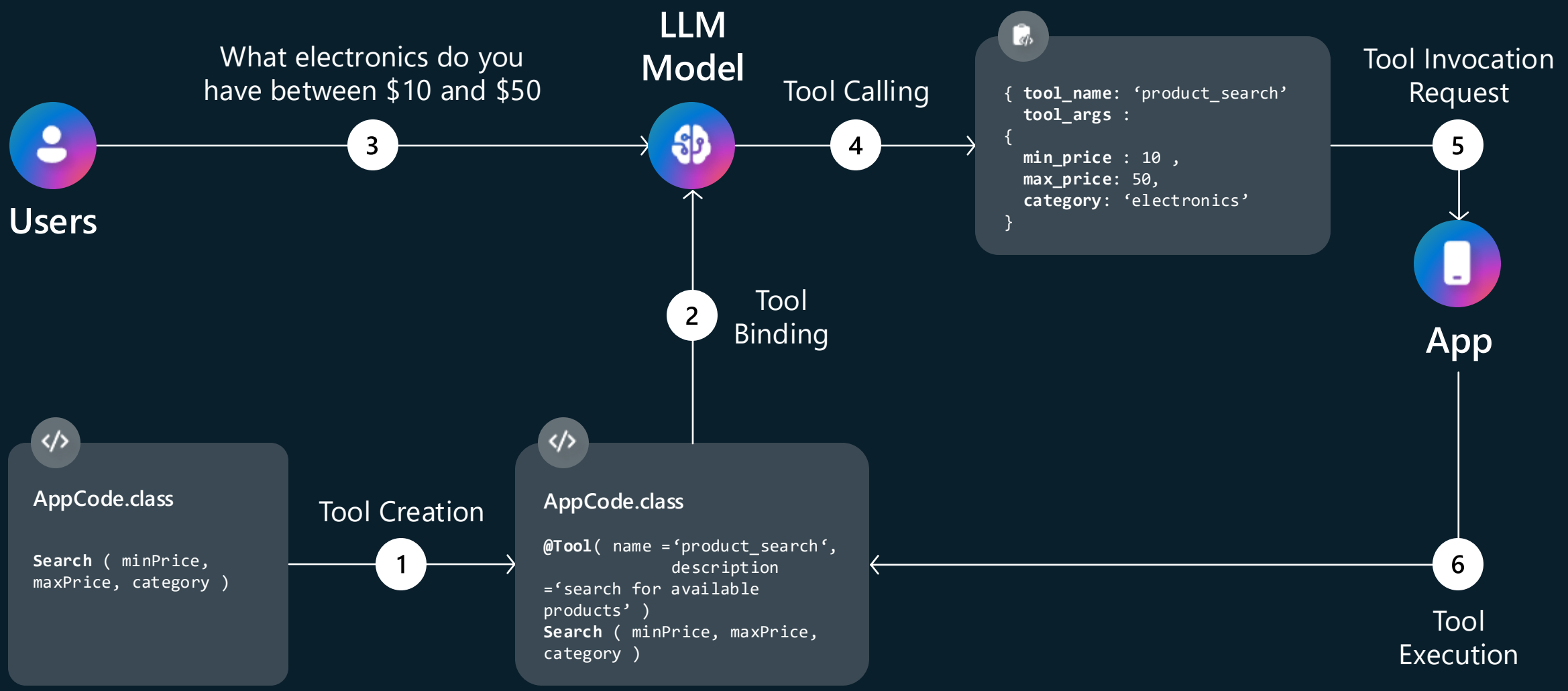


Agent Chat as layer for collaboration

- **Multiple agents** can engage with each other
- Enables multi-turn or parallel execution



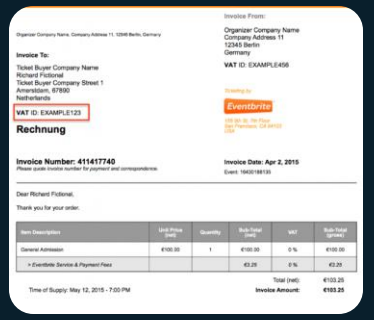
Agentic Pattern - Tools Calling



Agentic Pattern - ReAct Planning with Tools Calling

Users

Pay this bill for me



bill-abc123.jpeg

Bill abc123 successfully paid

Payment Agent

Tools:

- scanImage (filename)
- transactionHistory (billId)
- paymentService (billId, Amount, Payee)

Instructions:

- You are a home banking assistant allowing users to pay the bill uploading a picture
- Always check if a bill has already been paid before submitting a payment
- Confirm the payment result

Planning

while (new tools execution request)

1. **scanImage** (bill-abc123.jpeg) → abc123, 100$, payeename
2. **transactionHistory** (abc123) → no results
3. **paymentService** (abc123, 100$,payeename) → ok

Agentic Pattern - Memory

Short Term

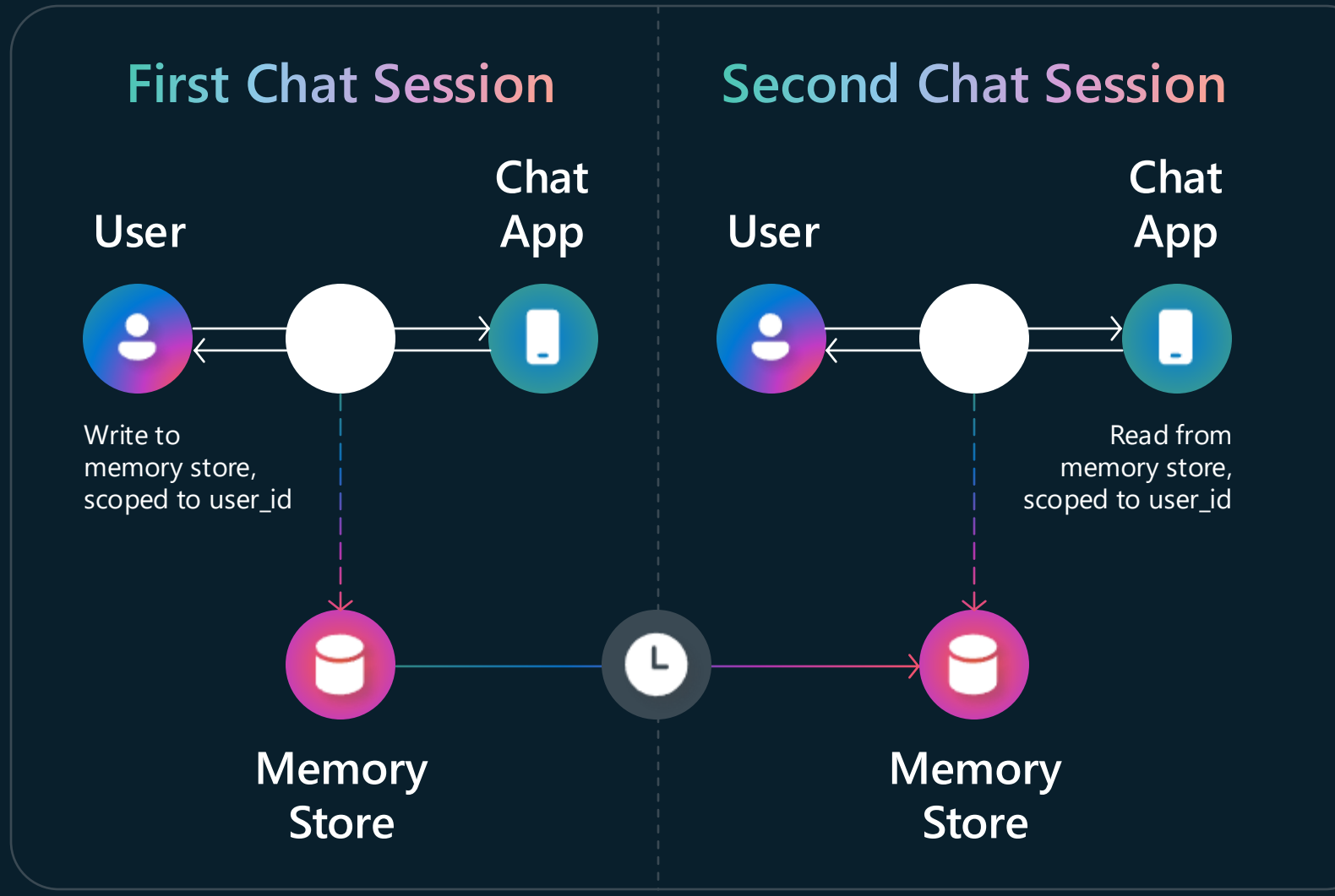
- Access steps info in one loop iteration
- Shared state context
- Chat history

Long Term

- Access steps info in long running conversation
- State persistence

Conversation History Truncation

- Trim by tokens
- Trim by message count
- Trim + summary (LLM call required)



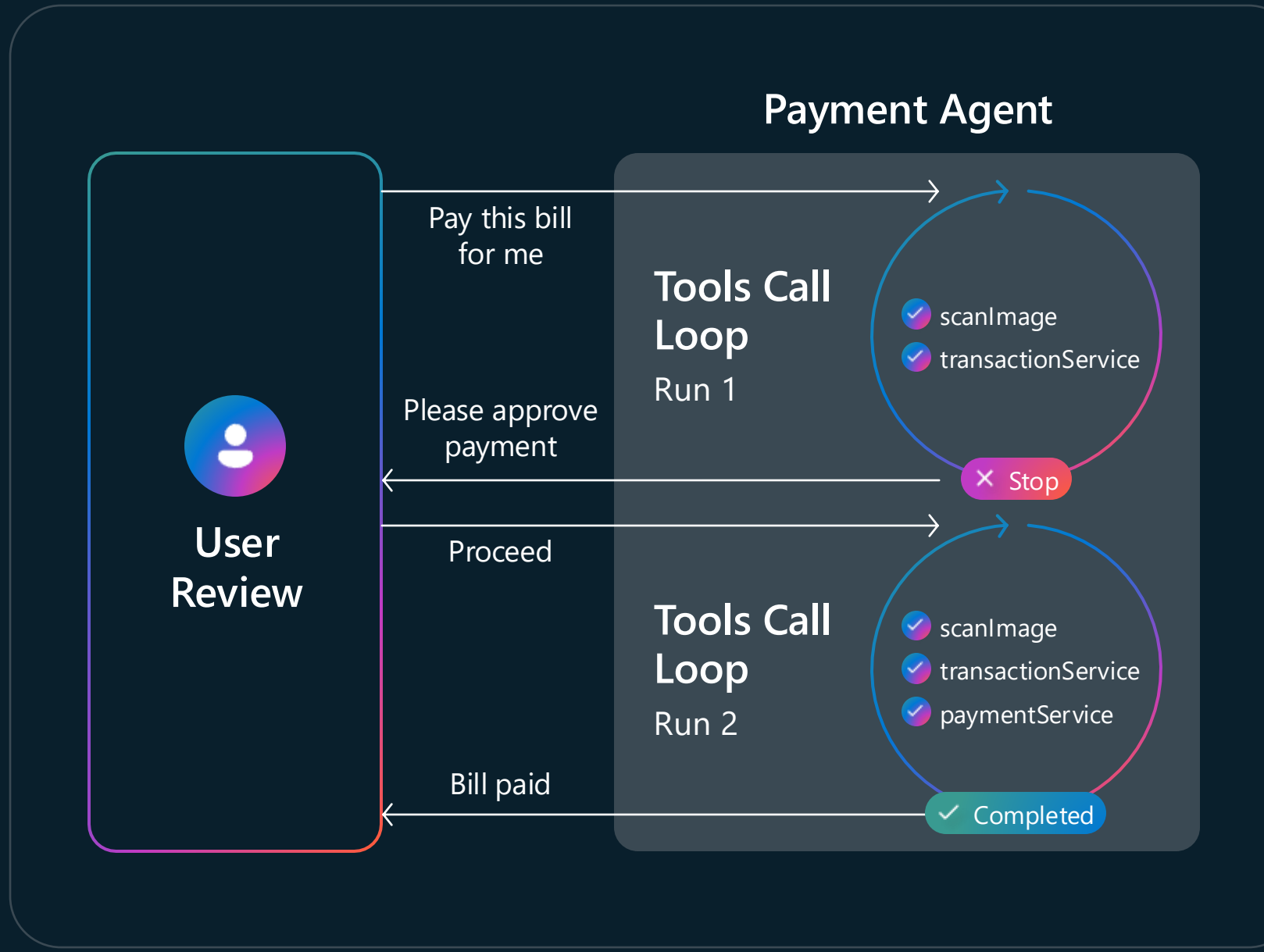
Agentic Pattern - Flow control

Looping Termination

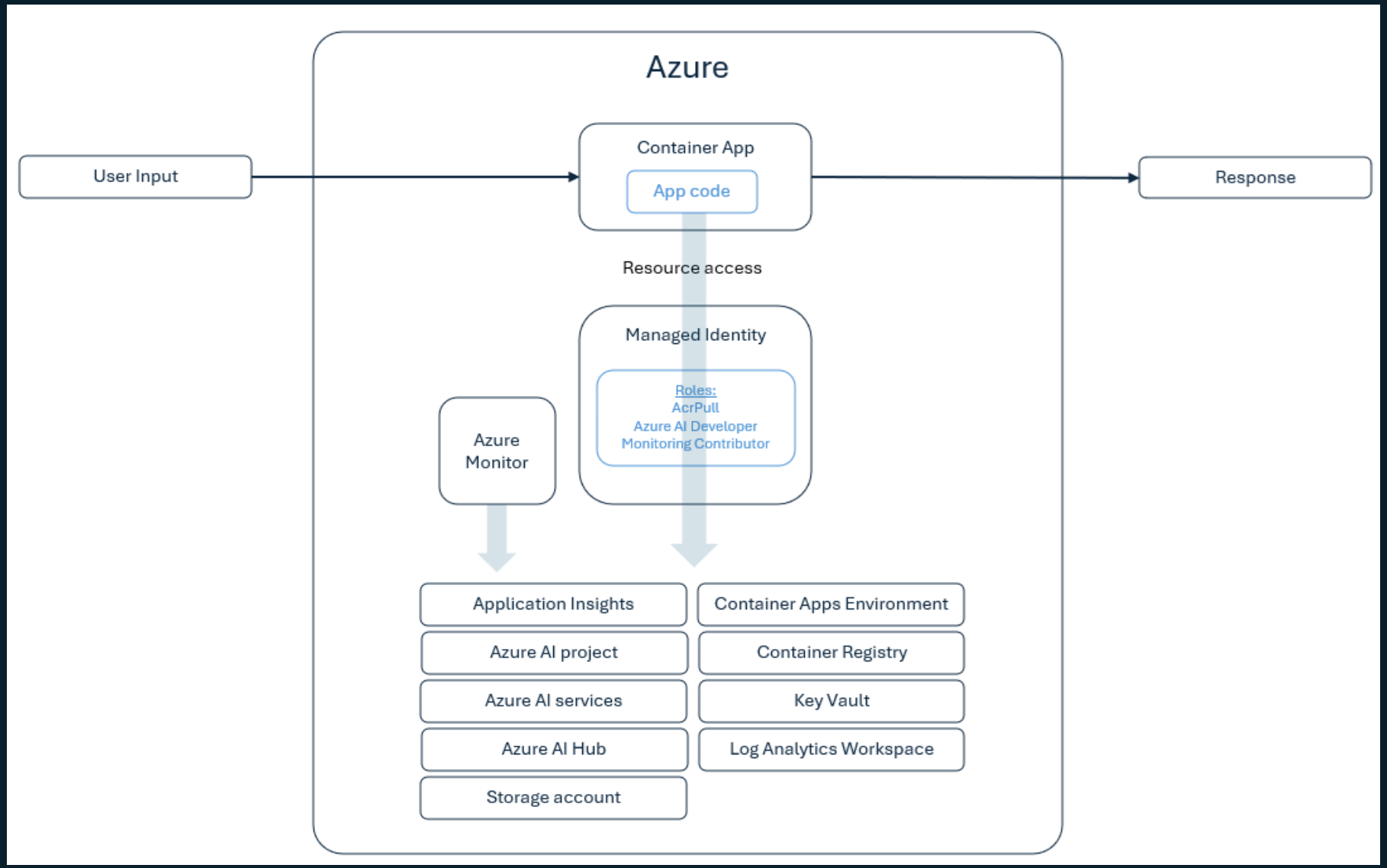
- MaxIterations
- Message termination
- Human step /Human in the loop

Human in the loop

- Action execution approval
- Escalation
- Data review



Getting Started with Agents Using Azure AI Foundry



Demo

Single Agent Architecture - Scaling

As the system grows you might run into scaling challenges

Too many tools. Tool hallucinations

Agent context (a.k.a. prompt) grows too much and it fails to follow instructions

Handling complex and dynamics tasks spanning different business domains

Multi agent architecture opportunities

Manageability – Modular agents reduce development and testing complexity

Predictability – More control over application flow using structured agents communication

Flexibility – Ease to incorporate new agents as solution domains increase

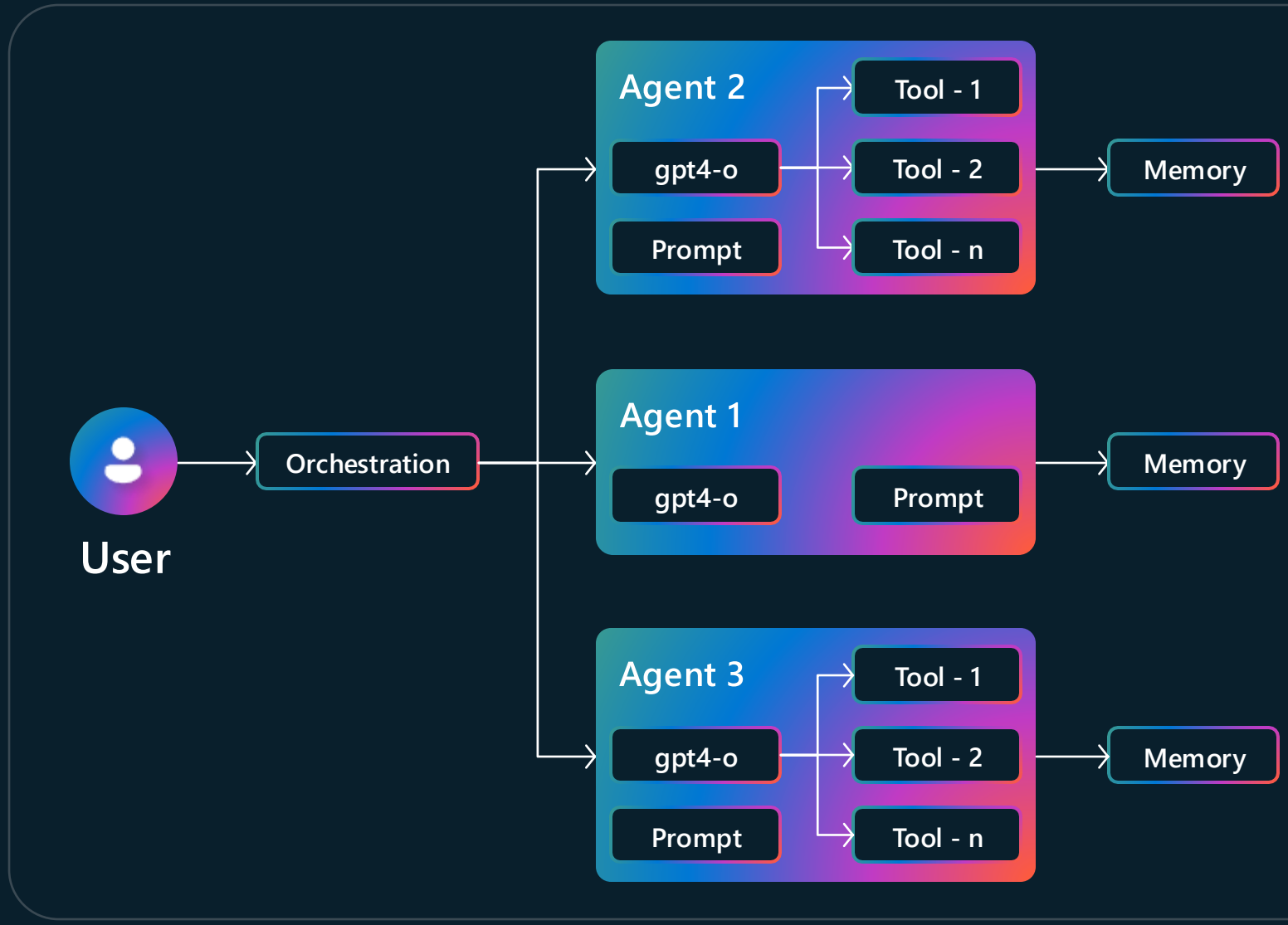
Multi Agent Logical Architecture

Each agent is specialized in different tasks or aspects of a problem

Agents can communicate and coordinate with each other. Structured orchestration is crucial

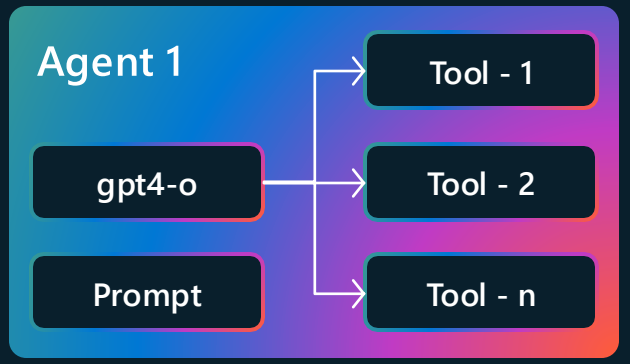
2 primary categories based on orchestration types

- Vertical Architecture
- Horizontal Architecture

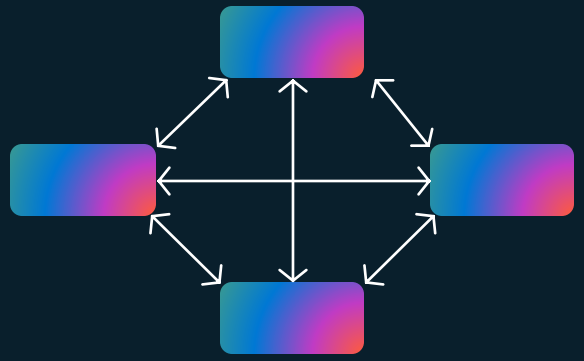


Agents orchestration and communication styles

Single Agent

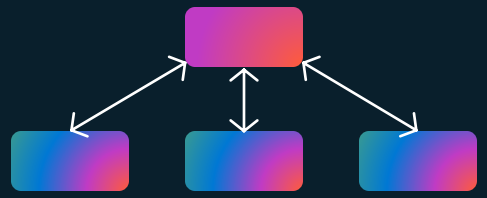


Network

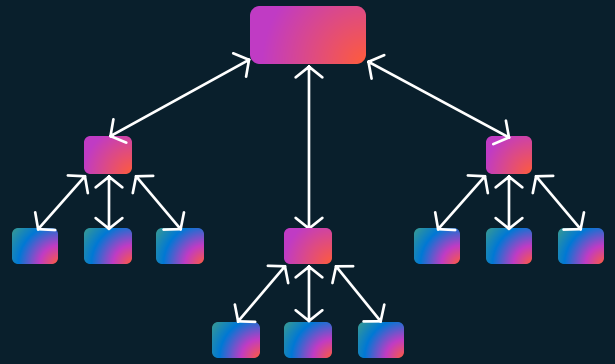


Reflection

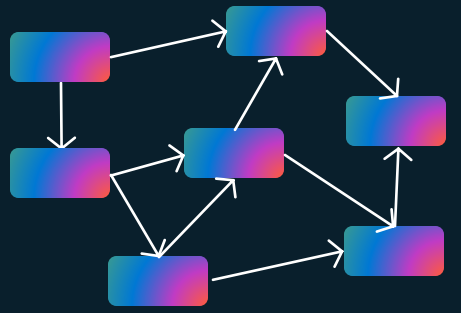
Supervisor



Hierarchical



Custom



Semantic Kernel Agents

Reflection multi-agent

- Coding agents
- Digital content creation agents

ChatCompletionAgent

- Name
- Instructions
- Kernel (Services, Plugins)

Reflection loop with 2 agents

- Blog post writer
- Blog post reviewer

report-agents.py

```
writerInstructions =
 """
You are a blog writer helping to create blog post draft. Consider suggestions when refining an idea…
"""



blogWriter  = ChatCompletionAgent(
                    name = “writer”,
        instructions = writerInstructions,
        kernel = …)

reviewerInstructions =
 """
 determine if the blog post can be published.
 If so, state that it is approved.
 If not, provide insight on how to refine the draft…
  """

blogReviewer  = ChatCompletionAgent(
                    name = “reviewer”,
        instructions = reviewerInstructions,
        kernel = …)
```

Semantic Kernel Agents

Agents participate in conversations in a **AgentGroupChat**

Agents loop is bound by max iterations limit

report-agents.py

```
// Create a chat for agents interaction.
chat = AgentGroupChat(
    agents=[agent_writer, agent_reviewer],
    selection_strategy=KernelFunctionSelectionStrategy(
        function=selection_function,
        kernel=...
        result_parser=...,
        agent_variable_name="agents",
        history_variable_name="history",
    ),
    termination_strategy=KernelFunctionTerminationStrategy(
        agents=[agent_reviewer],
        function=termination_function,
        kernel=...,
        result_parser=...,
        history_variable_name="history",
        maximum_iterations=10, // Limit total number of turns
    )
)
```

Semantic Kernel Agents

Termination based on reviewer agent response evaluation

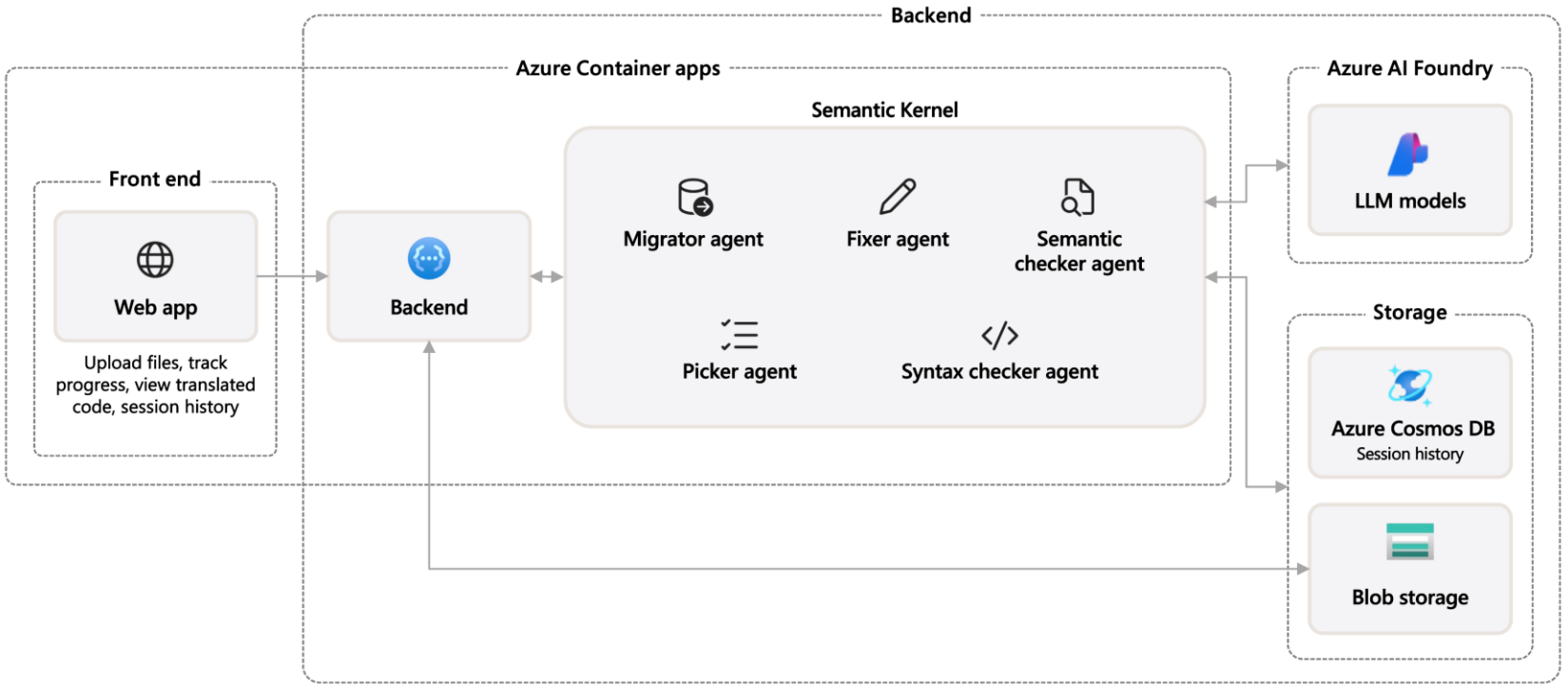
report-agents.py

```
// Terminate when the reviewer message contains the term "approve"
TERMINATION_KEYWORD = "yes“
termination_function = KernelFunctionFromPrompt(
    function_name="termination",
    prompt=f"""
Examine the RESPONSE and determine whether the content has been deemed satisfactory.
If content is satisfactory, respond with a single word without explanation:
{TERMINATION_KEYWORD}.
If specific suggestions are being provided, it is not satisfactory.
If no correction is suggested, it is satisfactory.
RESPONSE:
{{{{$history}}}}
        """
)
```

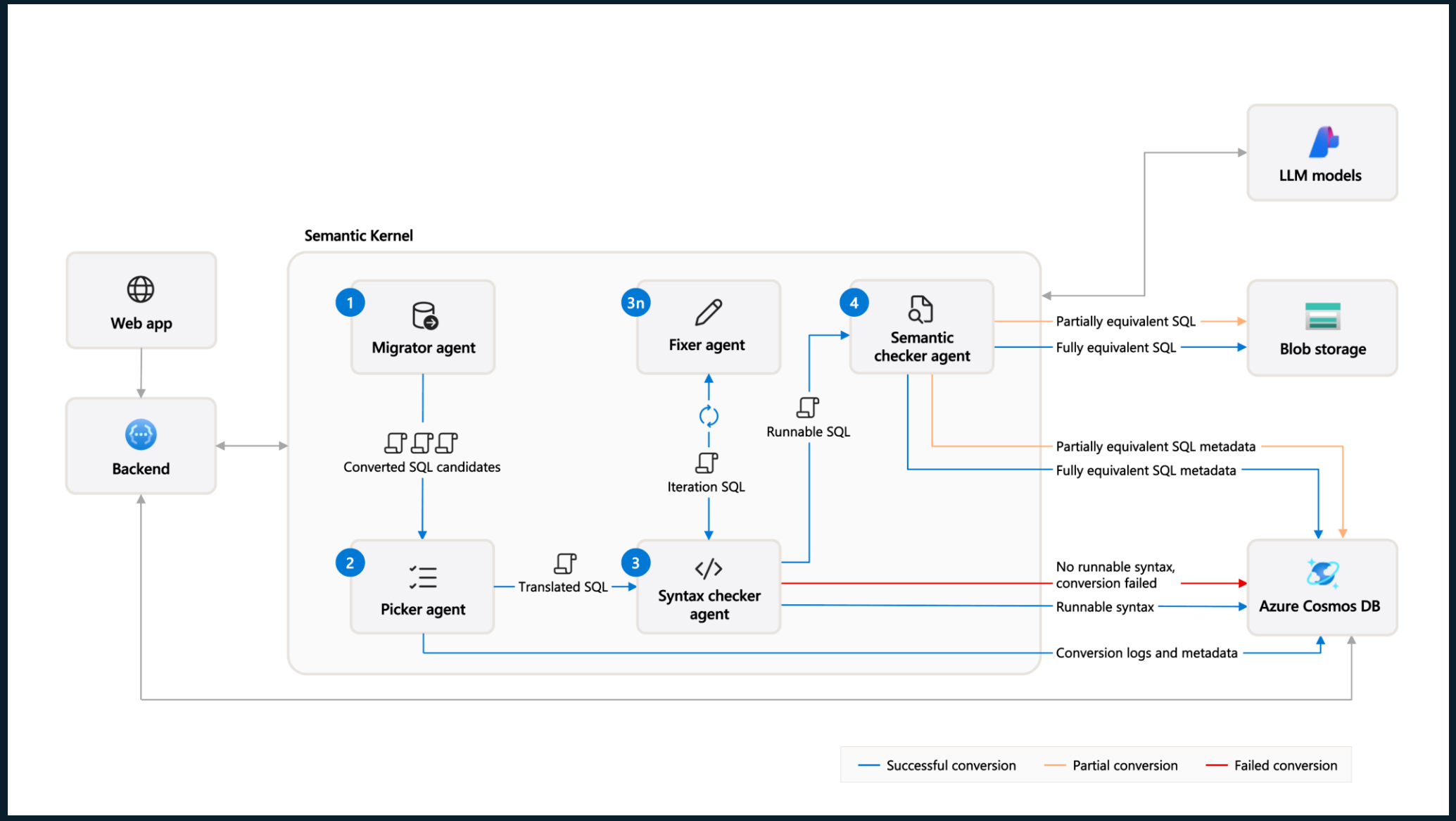
Demo

aka.ms/agentic-playground
aka.ms/semantic-kernel-examples

Modernize your code



Modernize your code



Multi-Agent Implementation Key Learnings

When to use agents

Flexible user tasks requiring multiple skills and domains on the backend

Customer support scenarios, intelligent code editors, personalized digital content creation

Tradeoff:
Latency + Cost vs Flexibility + Quality

Do I need agent frameworks to start?

Most are experimental - Semantic Kernel Agents are now Generally Available

Option 1: Use an agent framework. SK for production, Autogen for AI state of the art

Option 2: Use simple and composable apis like memory and tools in AI frameworks. Implement multi-agent conversation boilerplate code

Multi-Agent Implementation Key Learnings

Multi-agent autonomy and predictability friction

2 levels: Agent autonomy and multi-agent collaboration autonomy

Avoid tools hallucination: wrong tools, pre-fabricated, wrong calls sequence, wrong params

Well documented tools. Consistent api methods naming

Human-in-the-loop as your safety net

Predictable multi-agent conversations: share tools among agents, max loop iterations

Challenge:
Build Your Own AI Agents to Simplify Student Life

Agent frameworks and services

Semantic Kernel Agent Framework

Autogen

Langgraph

Azure AI Agent Service

Azure AI Services

Deliverables

Working solution of your web application

Presentation of your solution

Technical & Business Overview

Provided Resources

Access to the Azure Cloud

Microsoft Coaches to help along the way

Resources

1. ## Semantic Kernel Agents

   Semantic Kernel Agent Framework
   Building-multi-agent-systems-with-semantic-kernel

2. ## Autogen

   AutoGen v0.4: Reimagining the foundation of agentic AI for scale
   Microsoft's Agentic Frameworks: AutoGen and Semantic Kernel

3. ## Azure AI Agent Service

   Azure AI Agent Service
   Using Azure AI Agent Service with AutoGen/Semantic Kernel

4. ## AI Templates

   AI Templates Gallery

Q&A