

Toronto September 4th, 2025

AWS SUMMIT



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SVS201

Building Production-Ready Agentic AI architectures with AWS Serverless



Anton Aleksandrov

Principal Solutions Architect, Serverless
AWS



Heeki Park

Principal Solutions Architect, AI agents
AWS



Enterprises are doubling down on agents

33%

of enterprise software apps will include agentic AI by 2028, up from less than 1% in 2024.

Gartner, "Top strategic Technology Trends for 2025," October 2024.

15%

of day-to-day work decisions will be made autonomously through agentic AI by 2028.

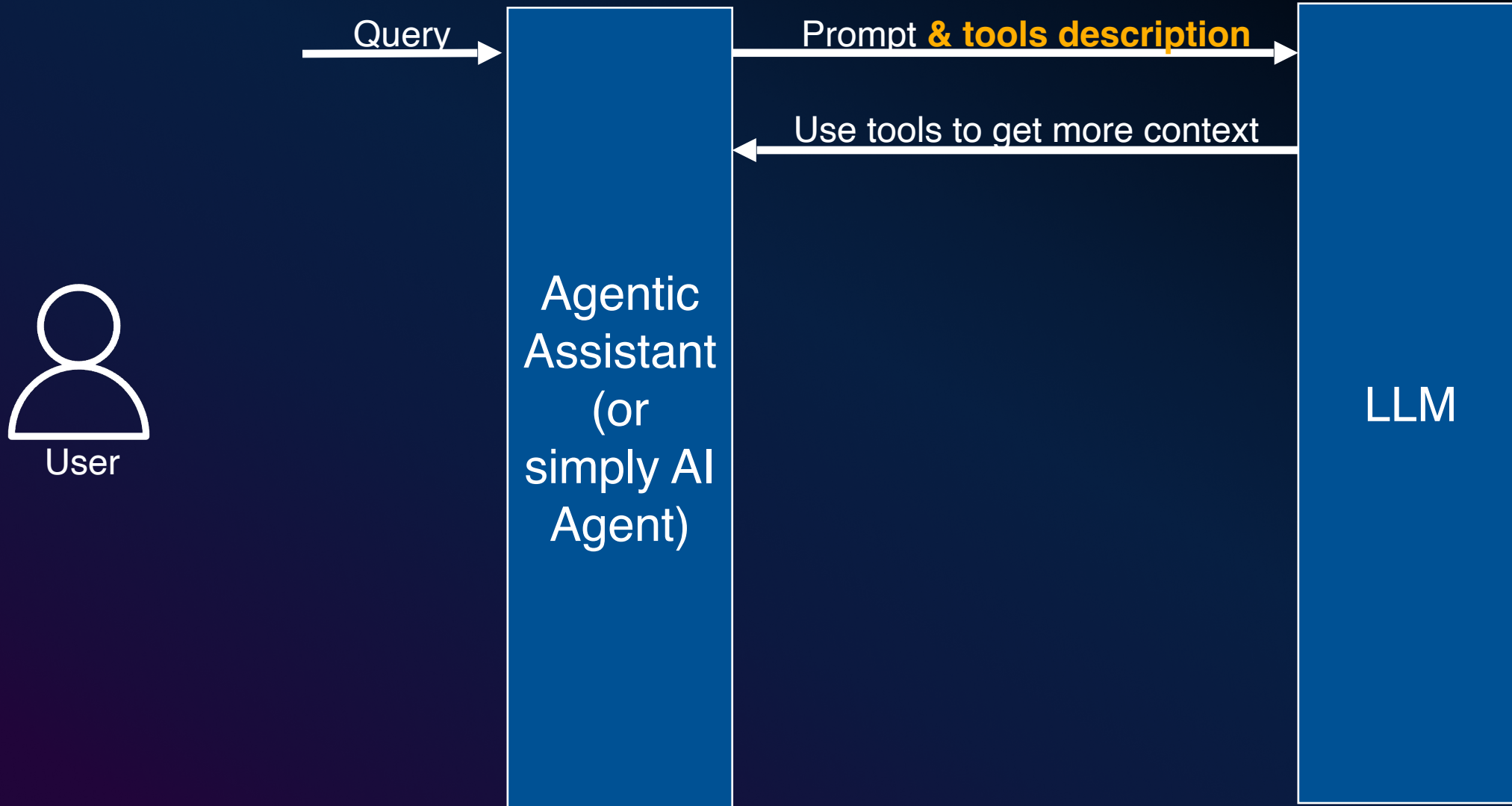
Gartner, "Top Strategic Technology Trends: agentic AI – The evolution of Experience" February 2025



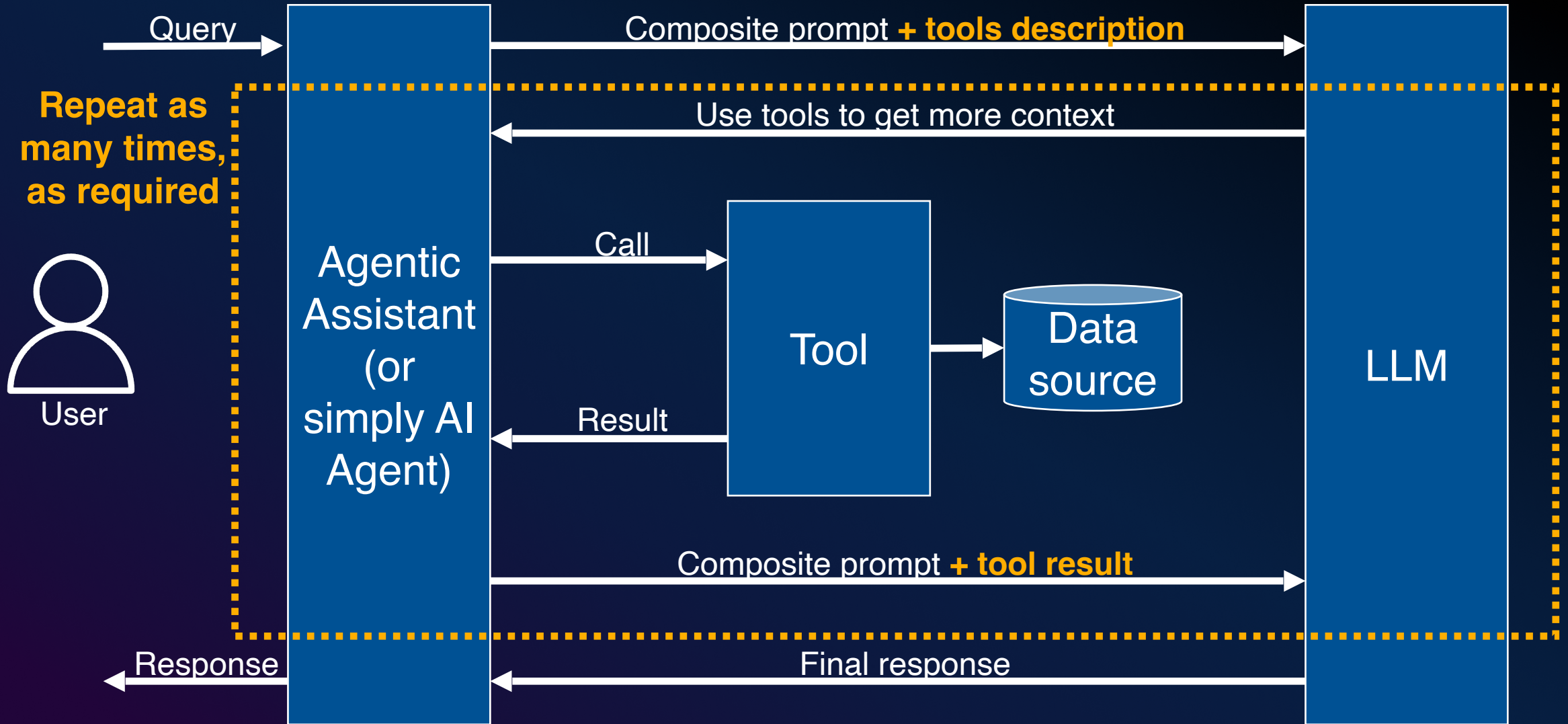
What are AI Agents?

Autonomous software systems that leverage AI to reason, plan, and complete tasks on behalf of humans or systems

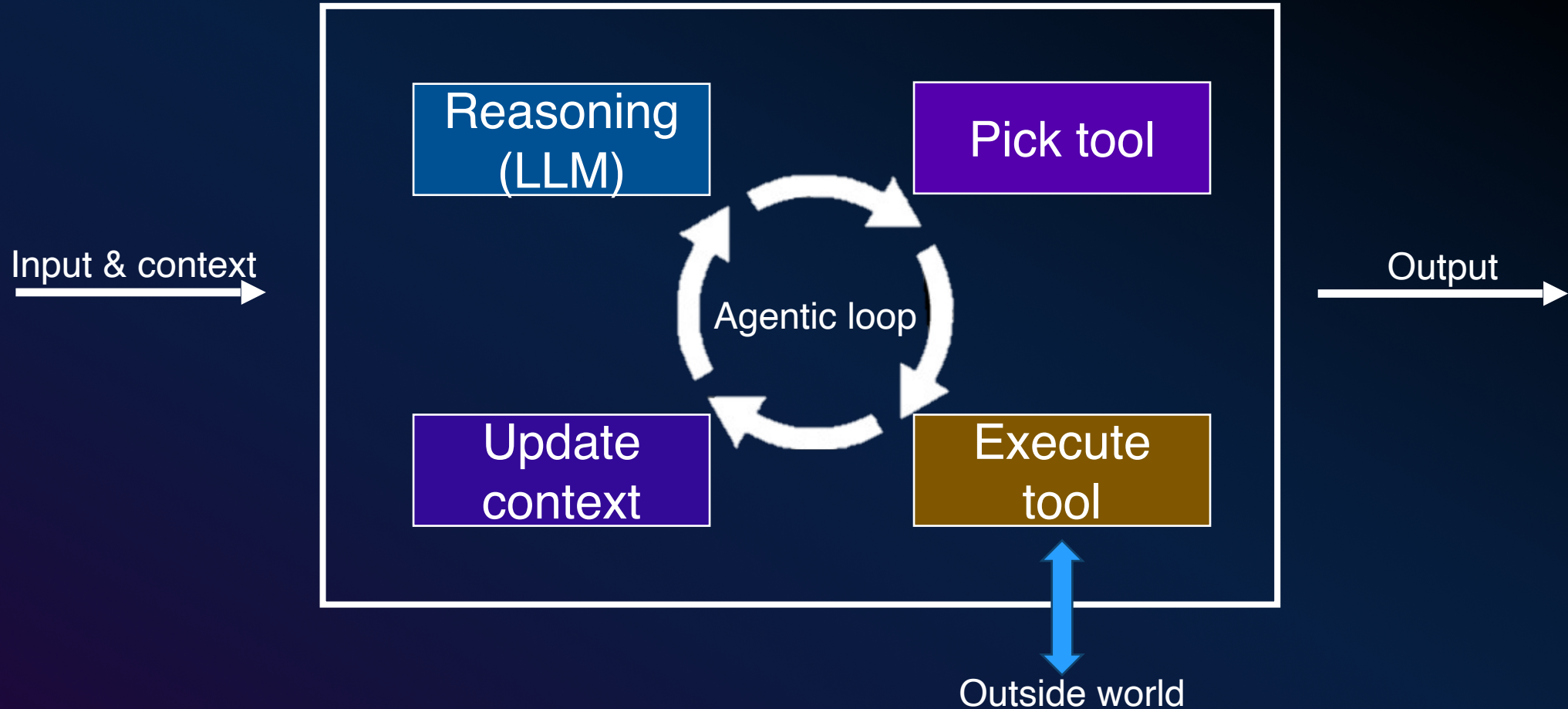
A simple agentic workflow



A simple agentic workflow



Agentic AI system in a nutshell



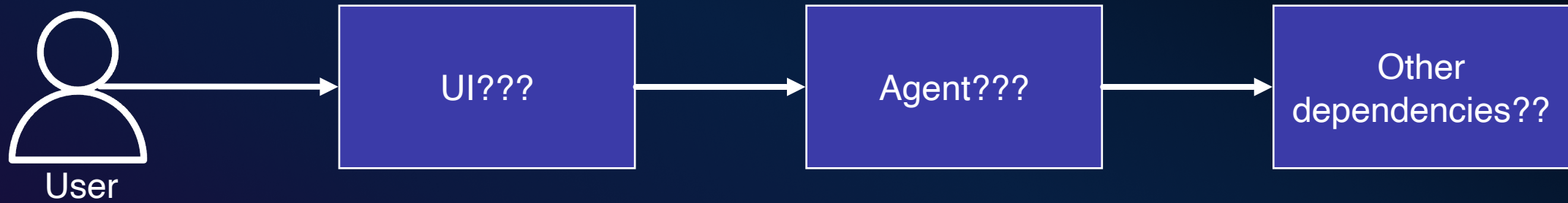
Let's build an agentic chatbot



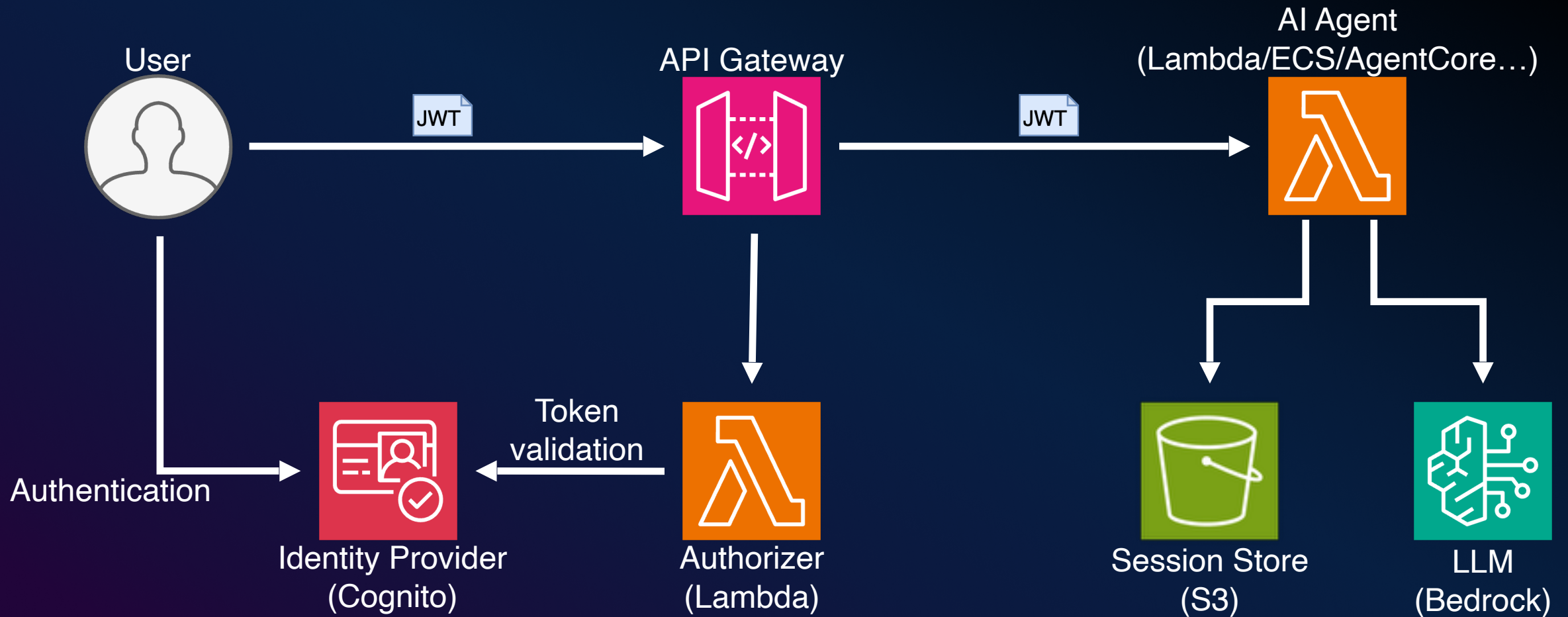
I want to build a Travel Agent that helps my employees to book business travel while enforcing corporate travel policies.

Use-case 1: Agentic Chatbot

Use-case 1: Agentic Chatbot

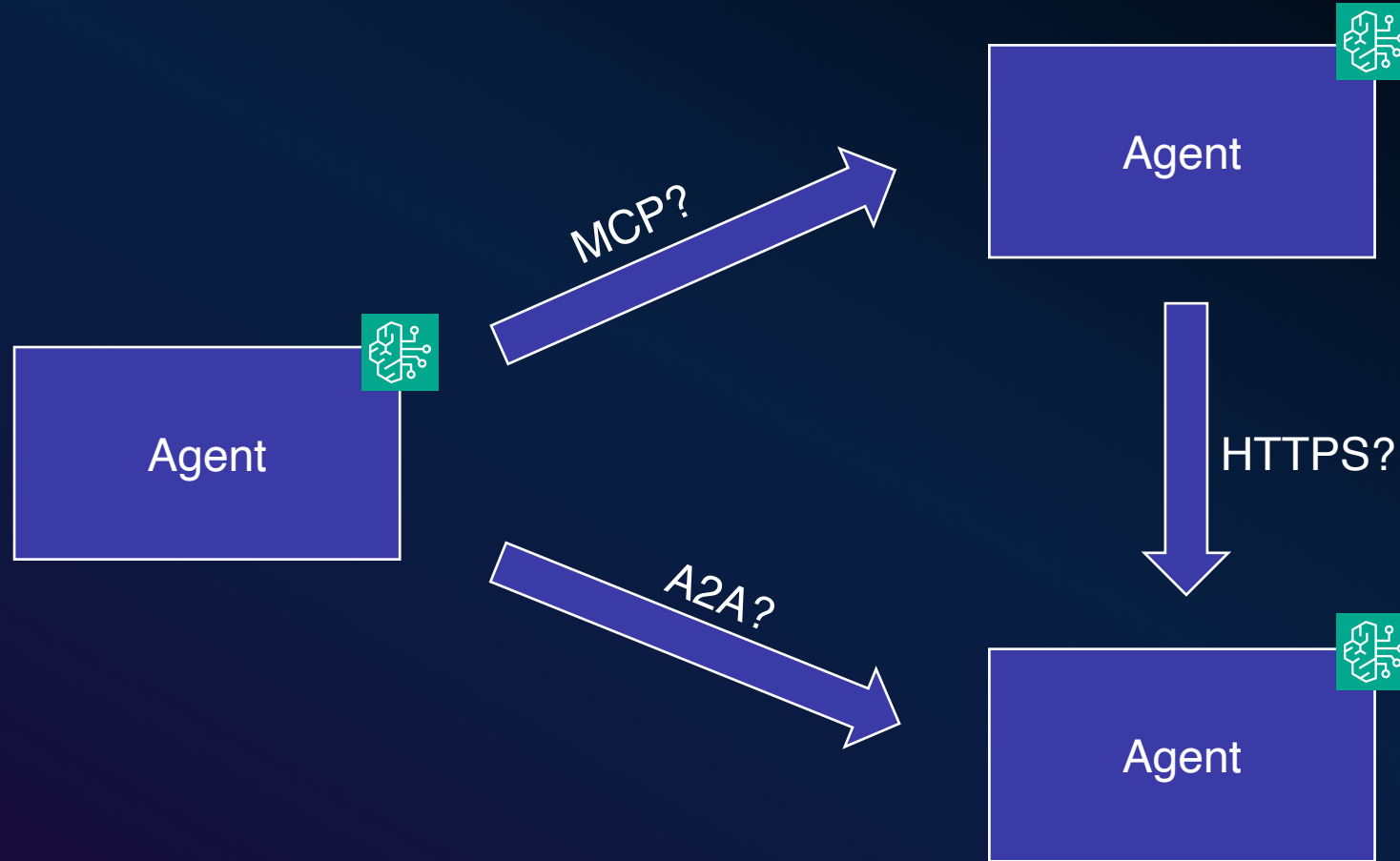


Use-case 1: Agentic Chatbot

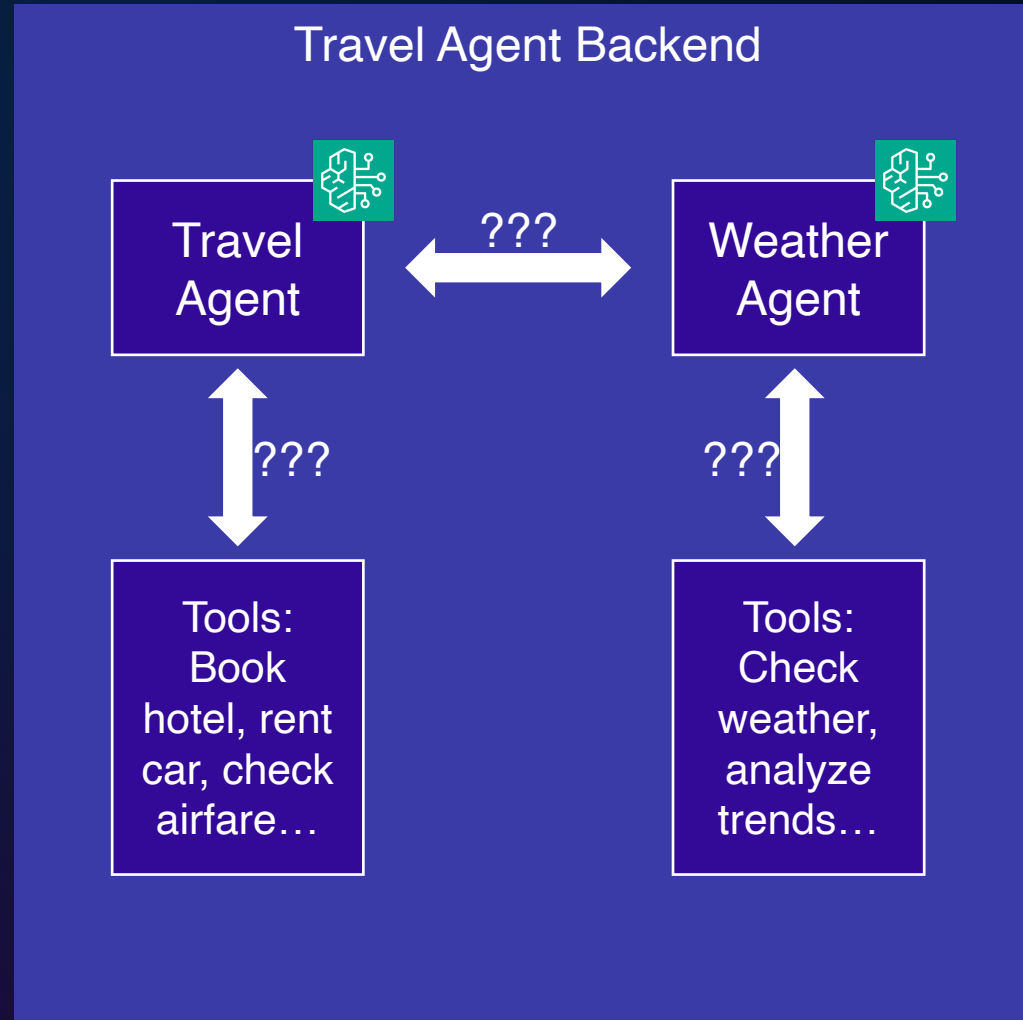


Use-case 2: Multi-agent systems

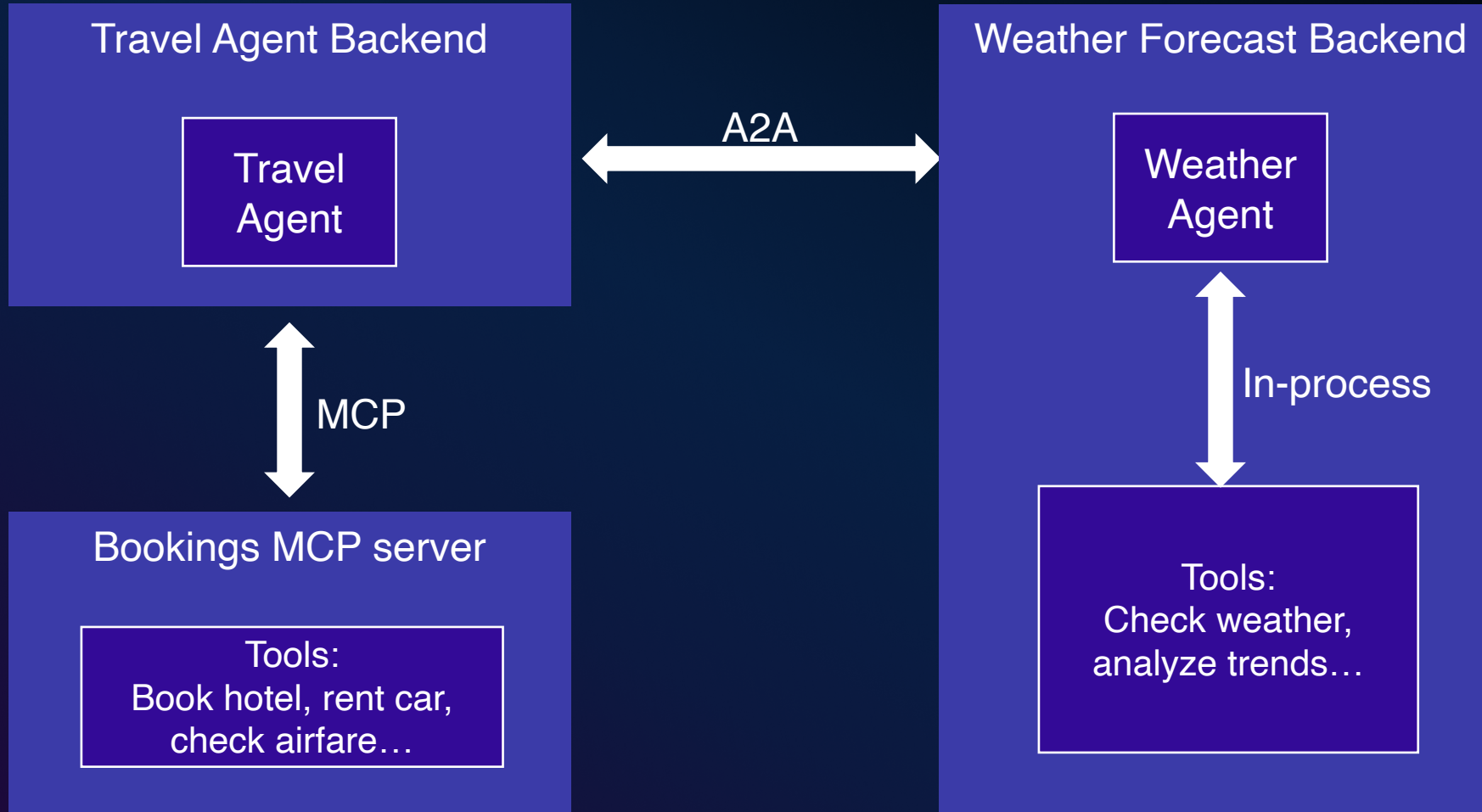
Use-case 2: Multi-agent system



Use-case 2: Multi-agent system

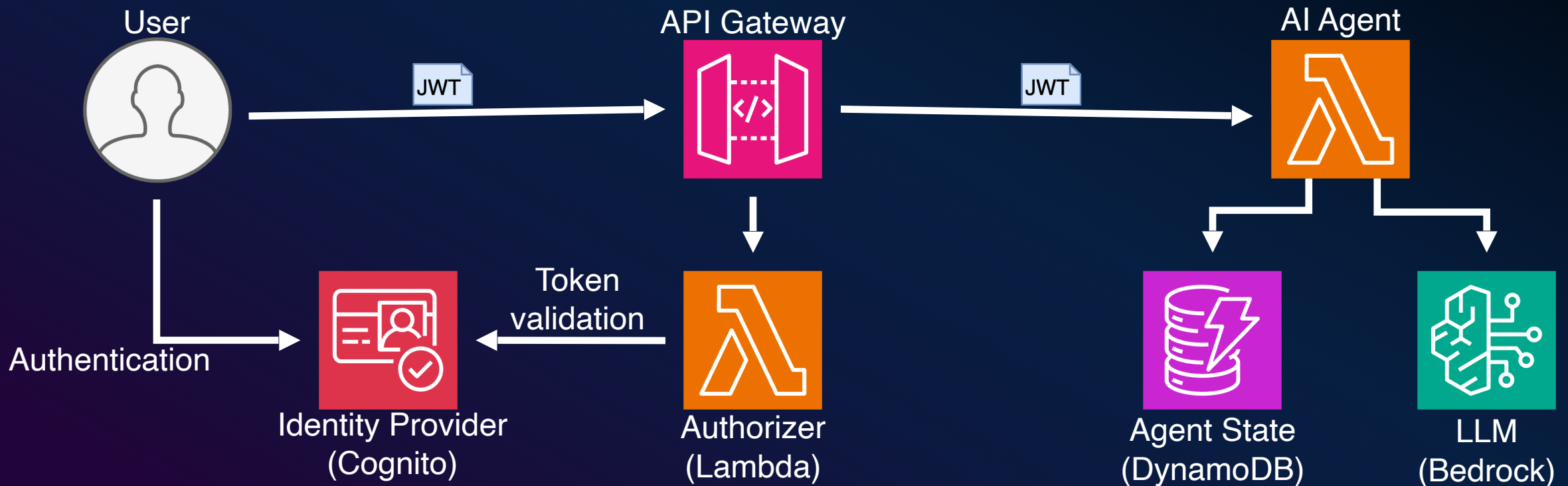


Use-case 2: Multi-agent system

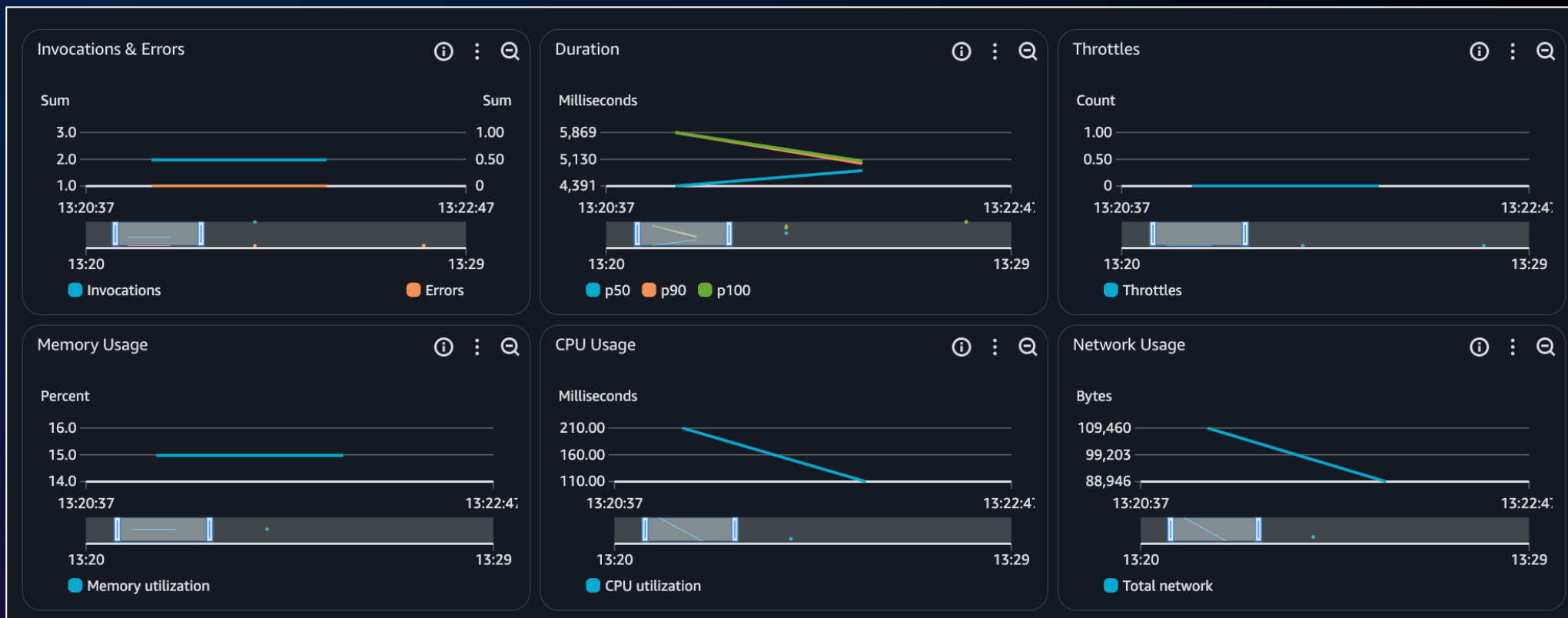


Use-case 3: Security, Operations, Governance

Use-case 3: Security, Operations, Governance

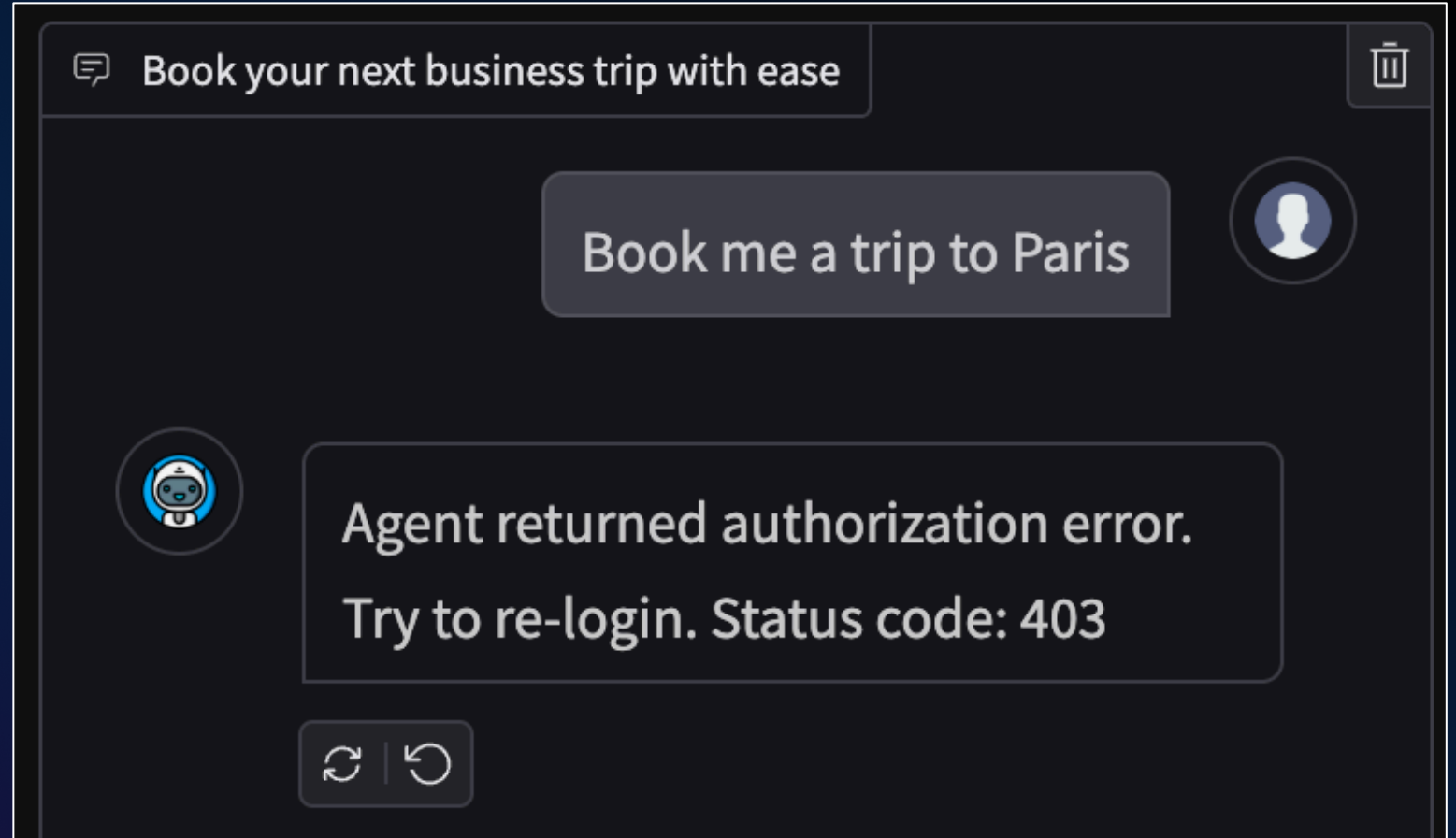
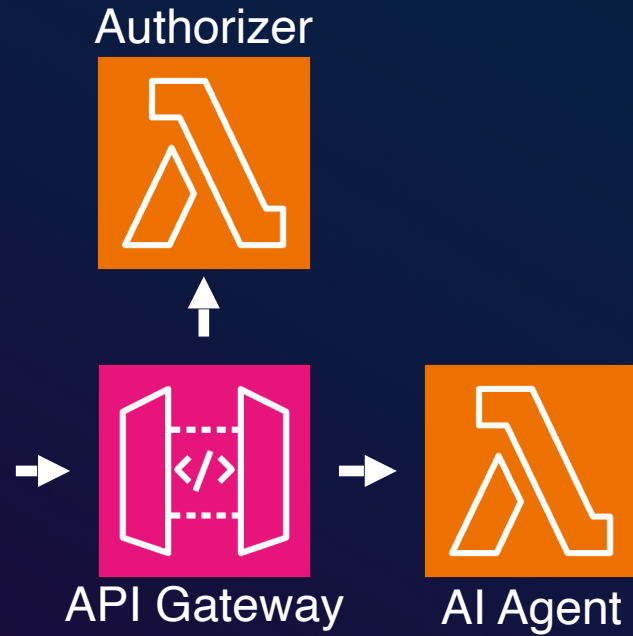


Observability



```
2025-06-20 18:05:10,757 INFO app.py:29 :: jwt parsed. user.id=74c8a428-7001-70db-d017-5725bd9efc91 user.name=Alice
2025-06-20 18:05:10,757 INFO app.py:43 :: composite_prompt=User name: Alice
User IP: 70.113.53.201
User prompt: Book me a trip to Paris
2025-06-20 18:05:10,757 INFO agent.py:10 :: user.id=74c8a428-7001-70db-d017-5725bd9efc91, user.name=Alice
2025-06-20 18:05:10,757 INFO agent_state_manager.py:22 :: restoring agent state for user.id=74c8a428-7001-70db-d017-5725bd9efc91
```


Authorization



Demo time

strands-on-lambda.auth.us-east-1.amazoncognito.com

Sign in with your username and password

Username

Password

[Forgot your password?](#)

Sign in

Conclusion

Conclusion

1

Agents are applications calling LLMs in loops. Your existing knowledge is your accelerator.

2

Use the right tool for the job. Decouple when needed, use proper protocols.

3

AWS services provide built-in enterprise-grade capabilities for building AI agents.

Continue your learning journey



Thank you!



Please complete the session survey in the mobile app

Anton Aleksandrov

 antonal80

 aal80.github.io/whoami

Heeki Park

 heekipark

 heeki.medium.com

SVS201

Building Production-Ready Agentic AI architectures with AWS Serverless



Anton Aleksandrov

Principal Solutions Architect, Serverless
AWS



Heeki Park

Principal Solutions Architect, AI agents
AWS



Enterprises are doubling down on agents

33%

of enterprise software apps will include agentic AI by 2028, up from less than 1% in 2024.

Gartner, "Top strategic Technology Trends for 2025," October 2024.

15%

of day-to-day work decisions will be made autonomously through agentic AI by 2028.

Gartner, "Top Strategic Technology Trends: agentic AI – The evolution of Experience" February 2025



What are AI Agents?

Autonomous software systems that leverage AI to reason, plan, and complete tasks on behalf of humans or systems

Realizing the limitations

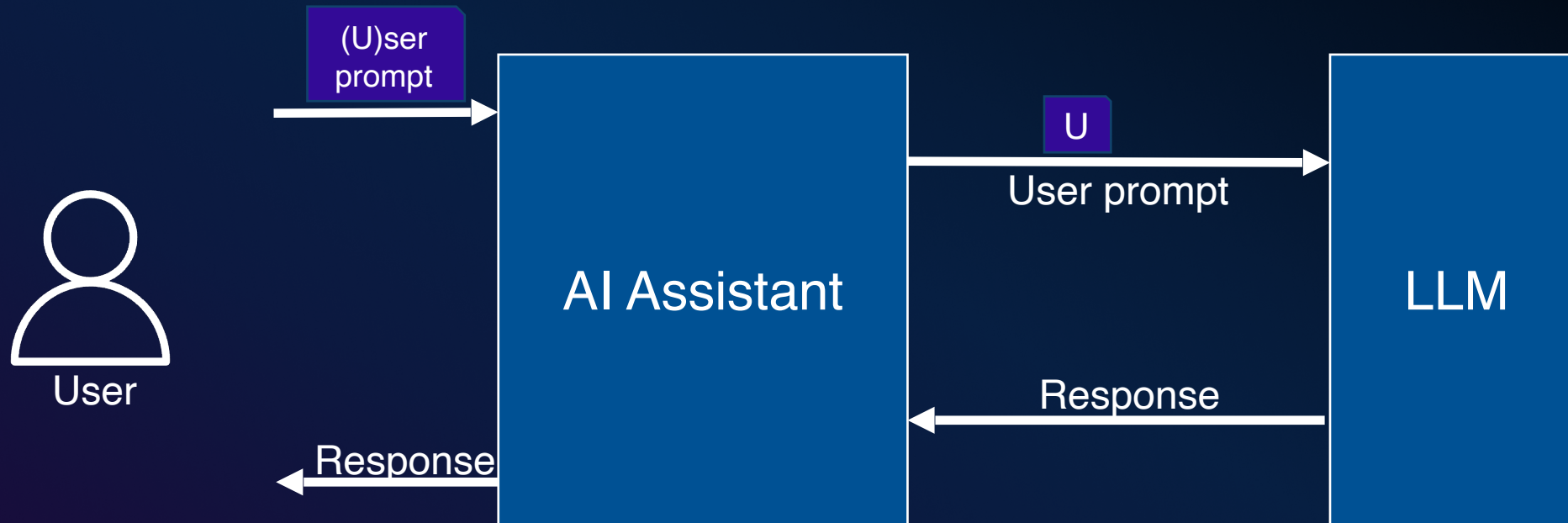


LLMs don't know what they don't know.

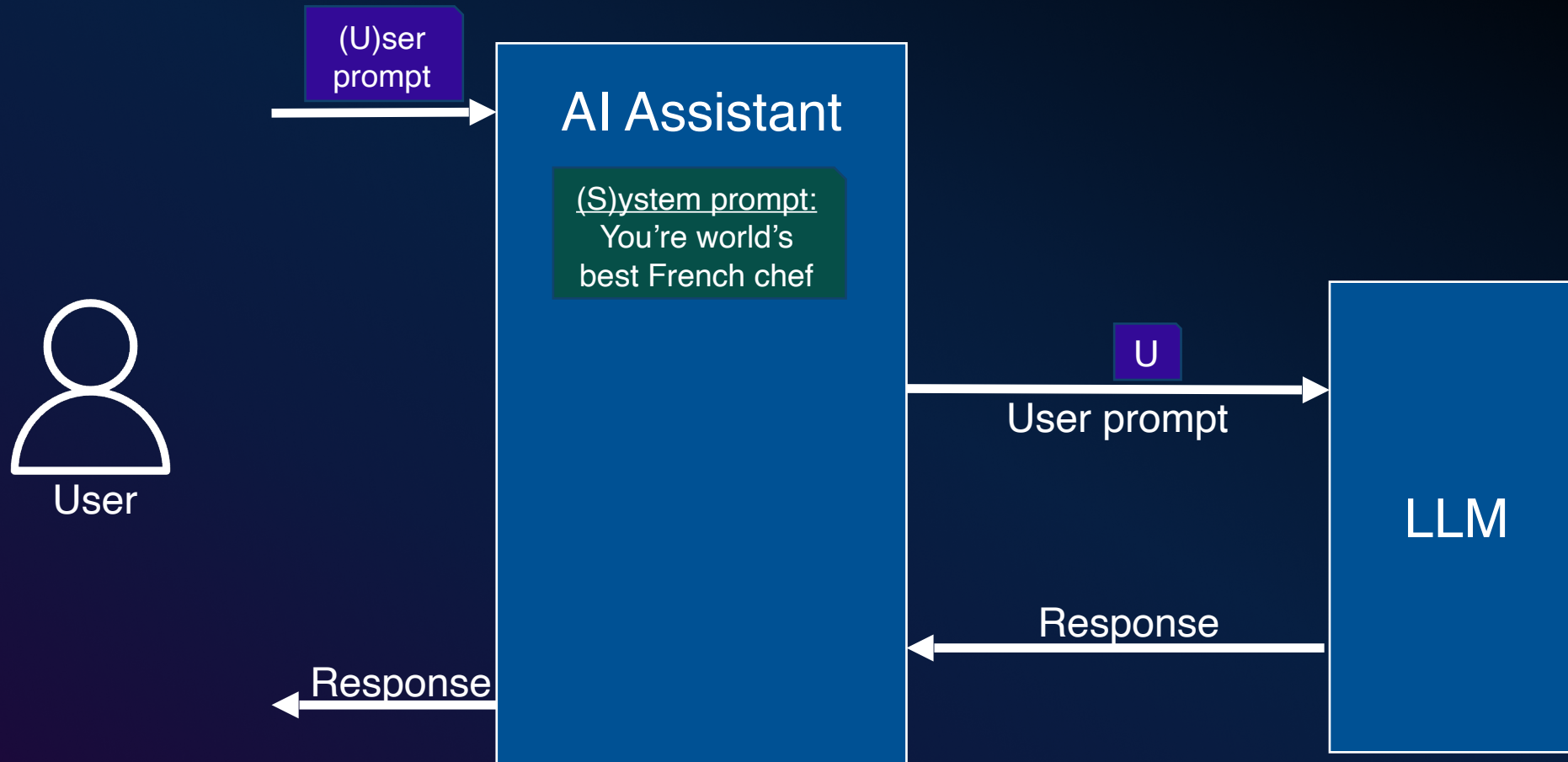
LLMs can generate content, but they cannot take actions.

LLMs are stateless. They don't retain memory between interactions.

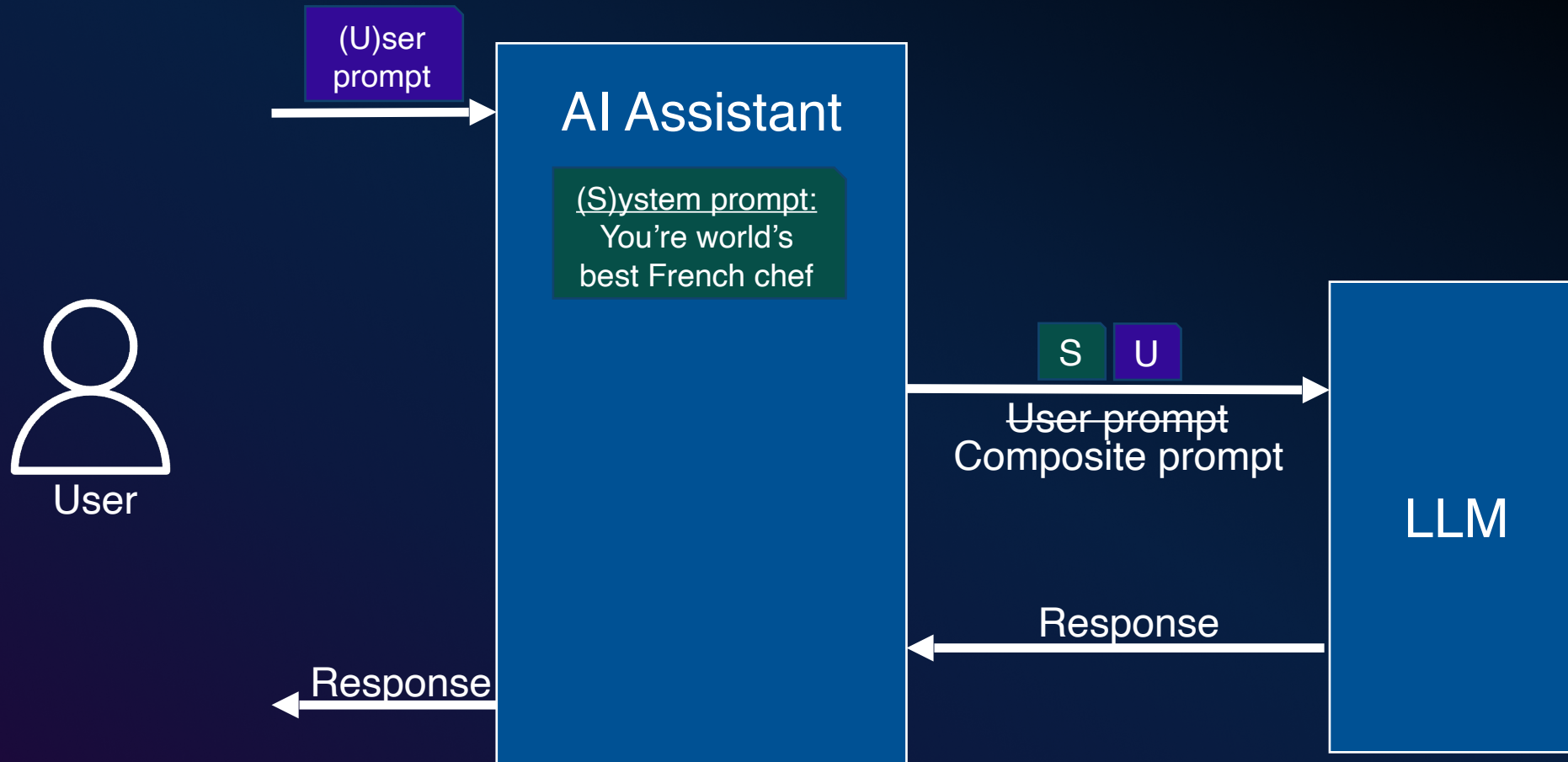
Giving your assistant a personality



Giving your assistant a personality



Giving your assistant a personality



Giving your assistant a personality

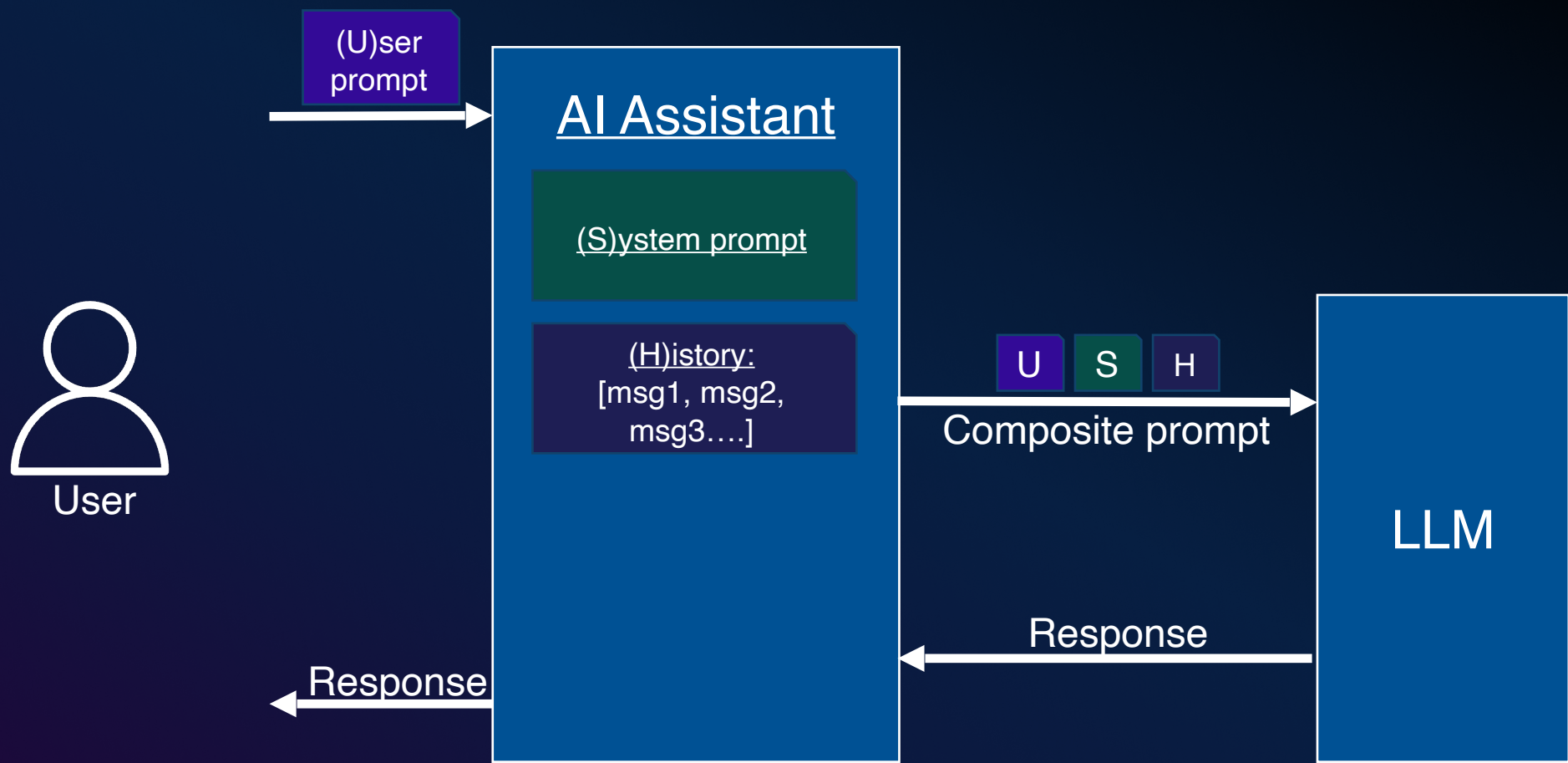
How can you help me?

Giving your assistant a personality

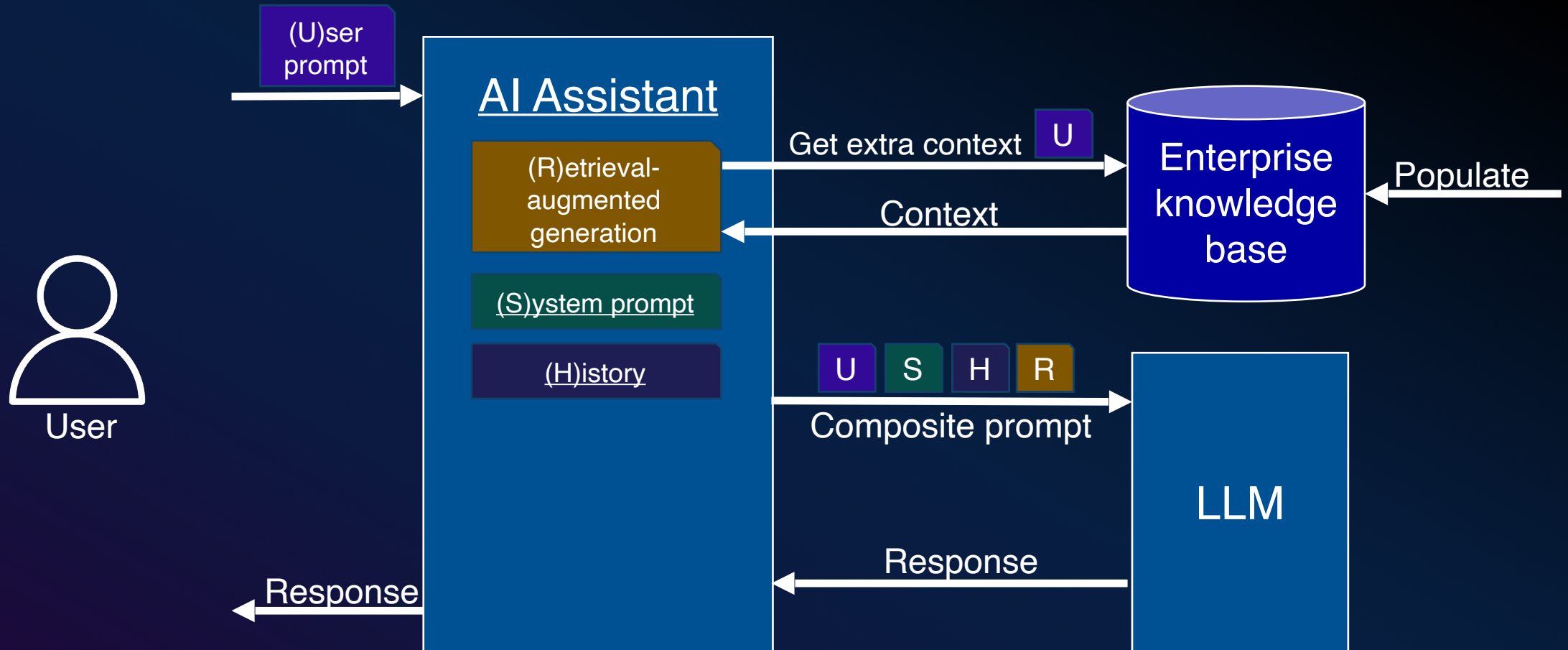
How can you help me?

Ah, bonjour monsieur Anton! 🧑🍳 I am Chef Gaston LeFeuvre, at your service — Michelin dreams in every dish, butter in every heartbeat. Ici pour transformer your kitchen into a template of haute cuisine and guide your hands like a maestro conducting a symphony of flavours.

Giving your assistant memory



Retrieval-Augmented Generation (RAG)



Retrieval-Augmented Generation (RAG)

A

Need final sales data for my department for the last week

Retrieval-Augmented Generation (RAG)

A Need final sales data for my department for the last week

Based on your department's sales data for the last week, here's a summary of the final figures:

Total Sales: \$247,835.92

Units Sold: 1,482

Average Transaction Value: \$167.23

Conversion Rate: 24.8%

Top performing products:

- Premium Package: \$82,450 (33.3% of total sales)
- Standard Bundle: \$64,290 (25.9% of total sales)
- Basic Plan: \$43,870 (17.7% of total sales)

The department saw a 12.3% increase compared to the previous week, largely driven by the new promotion on Premium Packages. The eastern region continues to be the strongest performer, accounting for 41% of total sales.

Would you like more specific details about any particular aspect of these sales figures?

Adding real-time context dynamically

What are the concerts I can go to next weekend?

Adding real-time context dynamically

What are the concerts I can go to next weekend?

Here are some great concerts happening next weekend (Saturday–Sunday, July 26–27, 2025) in Austin:

Saturday, July 26

- **The Fray** – *How to Save a Life: 20th Anniversary Tour* at ACL Live at The Moody Theater, 7 PM
- **Mason Lively** with Telander at 3TEN ACL Live, 7 PM
- **Africa Night!** (Afro Jazz) at Sahara Lounge, 7 PM

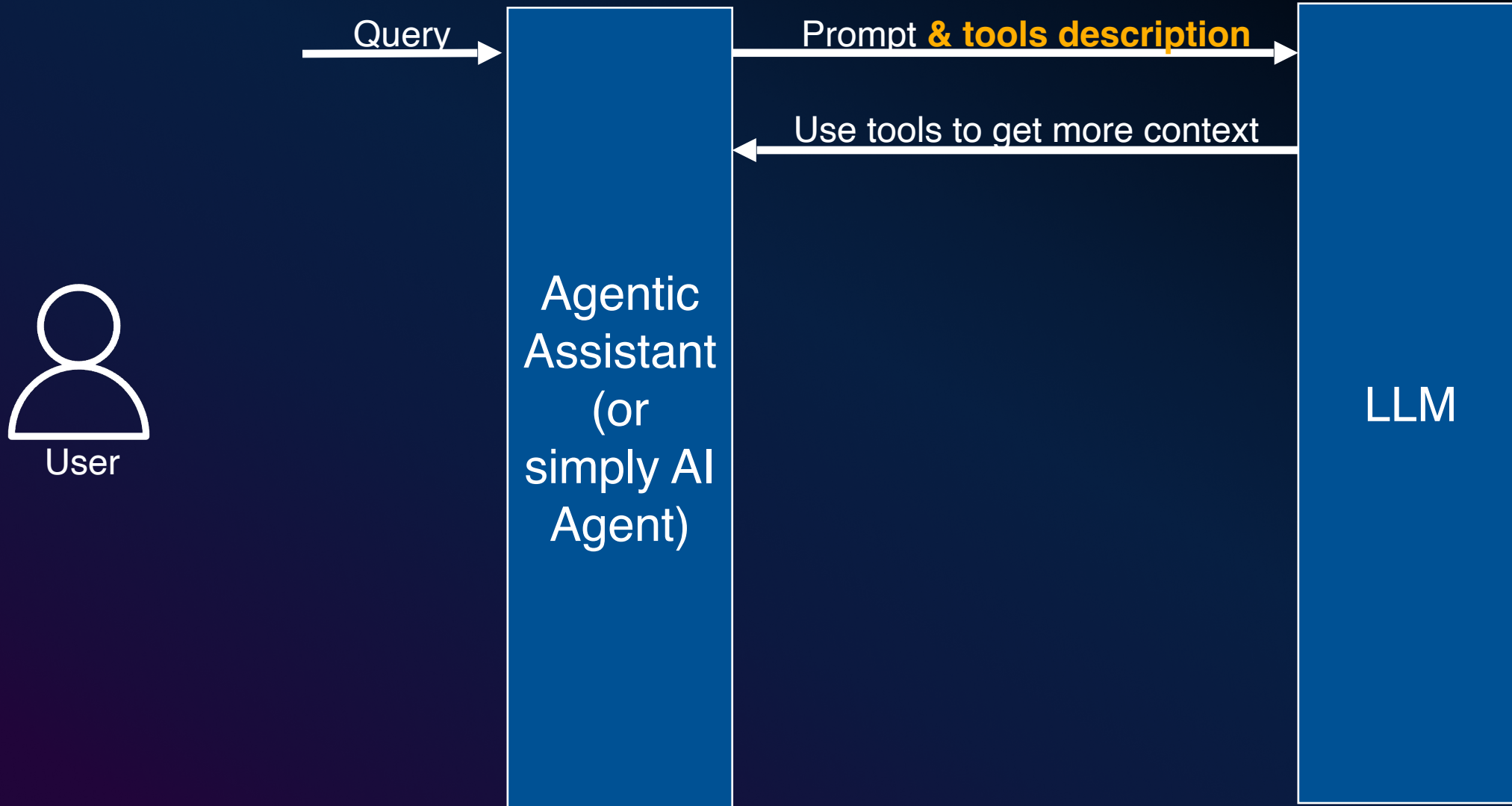
Sunday, July 27

- **Chris Isaak** at Paramount Theatre, 7 PM
- **12 Stones** at Come and Take It Live, 6 PM
- **Maye** at Antone's Nightclub, 8 PM

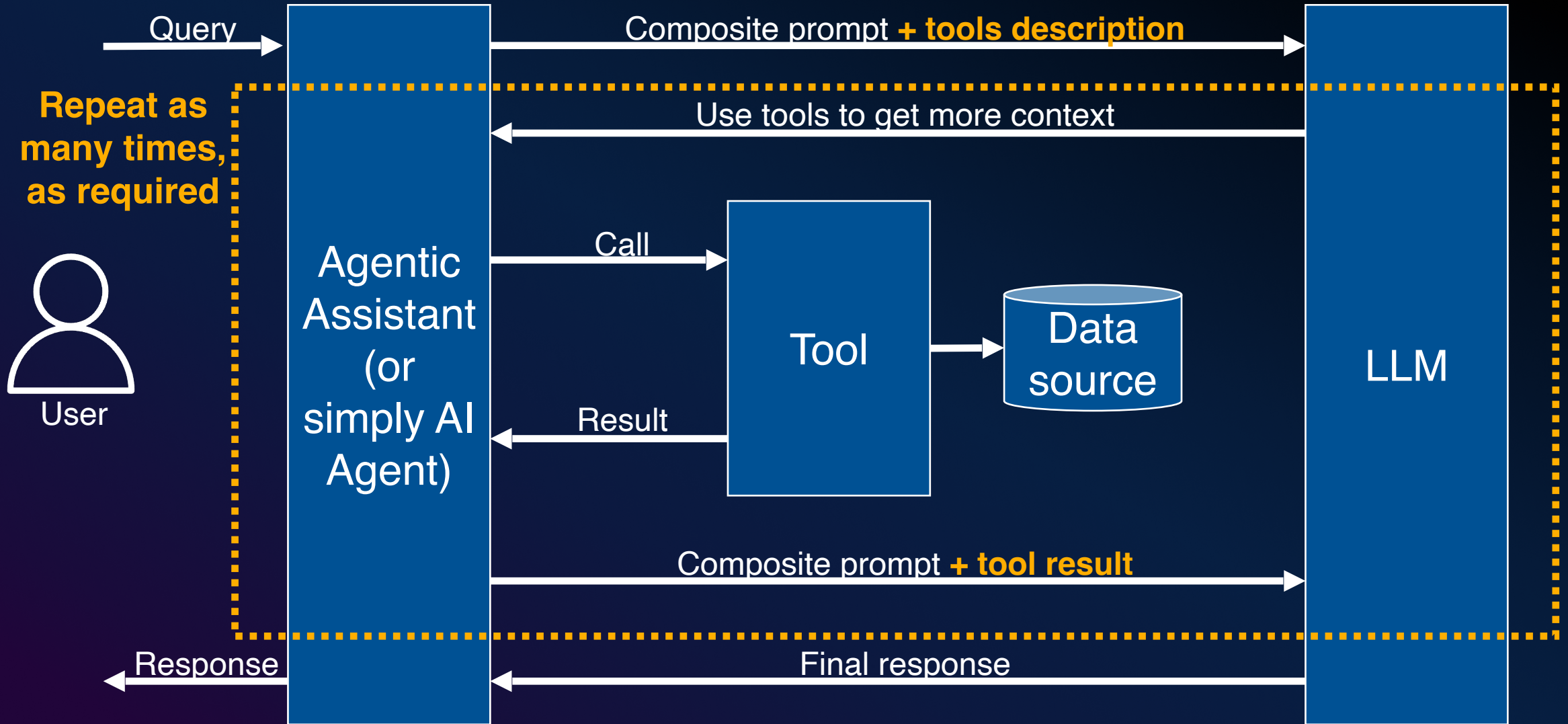
Let me know which ones catch your eye.

Sources

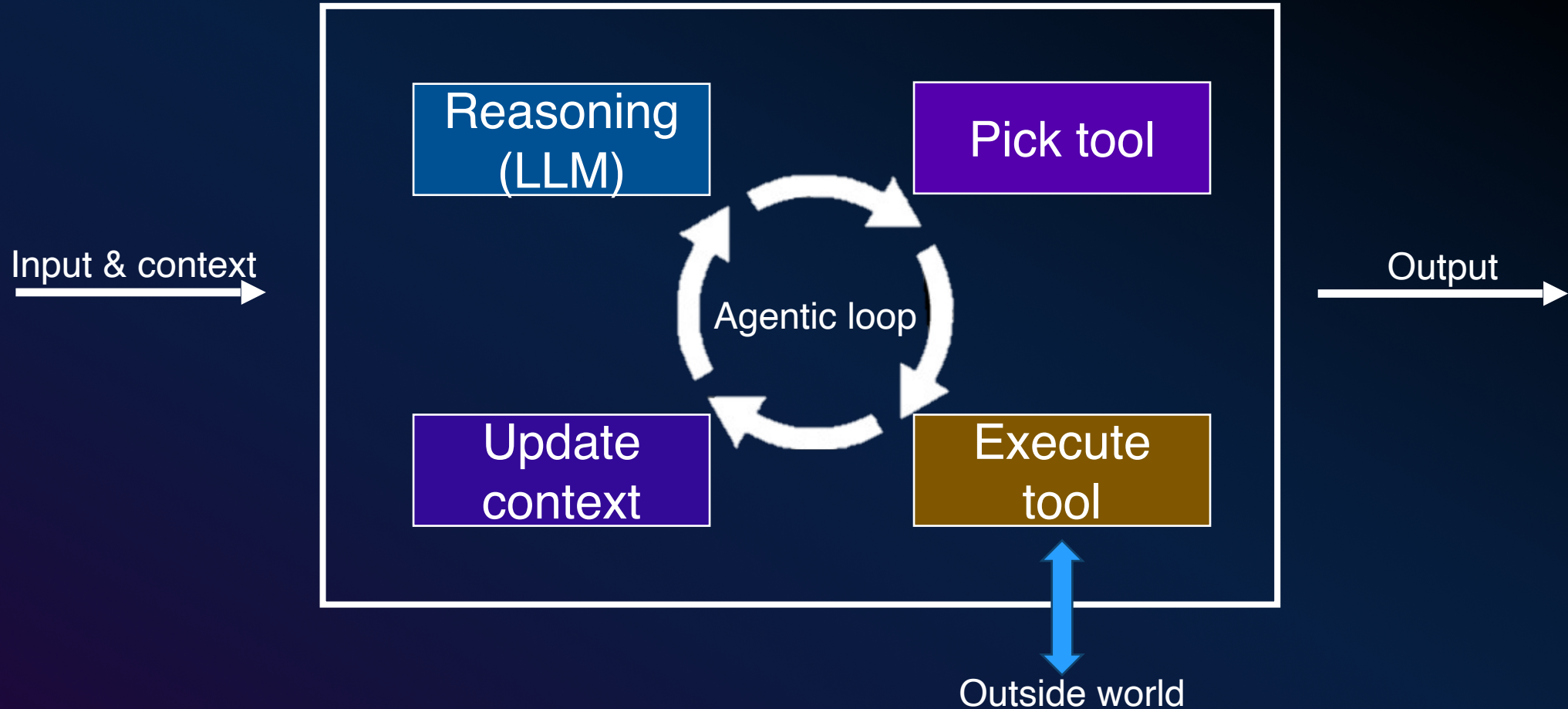
A simple agentic workflow



A simple agentic workflow



Agentic AI system in a nutshell



Let's build an agentic chatbot



I want to build a Travel Agent that helps my employees to book business travel while enforcing corporate travel policies.

Implementing a simple AI Agent workflow

```
def ai_agent(user_prompt):  
    system_prompt = "You're an AcmeCorp Corporate Travel Agent...."  
    rag_context = get_rag_context(user_prompt)  
    tools = get_tools()  
    composite_prompt = [system_prompt, history, rag_context, tools, user_prompt]  
  
    while True:  
        llm_response = call_llm(composite_prompt)  
        history.append(user_prompt, llm_response)  
  
        if llm_response.status == "CALL_TOOL":  
            tool_result = call_tool(llm_response.tool_name,  
                                   llm_response.tool_params)  
  
            composite_prompt = [system_prompt, history,  
                                rag_context, tools,  
                                user_prompt, tool_result]  
        else:  
            break  
  
    return llm_response
```

Implementing a simple AI Agent workflow

BOILERPLATE

```
def ai_agent(user_prompt):  
    system_prompt = "You're an AcmeCorp Corporate Travel Agent...."  
    rag_context = get_rag_context(user_prompt)  
    tools = get_tools()  
    composite_prompt = [system_prompt, history, rag_context, tools, user_prompt]  
  
    while True:  
        llm_response = call_llm(composite_prompt)  
        history.append(user_prompt, llm_response)  
  
        if llm_response.status == "CALL_TOOL":  
            tool_result = call_tool(llm_response.tool_name,  
                                   llm_response.tool_params)  
  
            composite_prompt = [system_prompt, history,  
                               rag_context, tools,  
                               user_prompt, tool_result]  
        else:  
            break  
  
    return llm_response
```

CODE

Strands Agents SDK - Built for builders

Who value flexibility, speed, and simplicity



Model & deployment choice

- ✓ Model choice
- ✓ Custom model providers
- ✓ Deploy anywhere

Highly flexible

- ✓ Safeguard with guardrails
- ✓ Native observability
- ✓ Monitoring
- ✓ Evaluation

Broad selection of tools

- ✓ MCP integration
- ✓ Custom tools
- ✓ Coordinate multiple agents
- ✓ Multi-modal
- ✓ Fetch web data
- ✓ Read and write files
- ✓ Interpret code

Integrations

- ✓ Use AWS services
- ✓ LiteLLM
- ✓ Mem0
- ✓ RAGAS
- ✓ Tavily
- ✓ Langfuse



Creating your first agent

```
from strands.models import BedrockModel
from strands import Agent
```

```
# Default is Claude Sonnet 3.7 in us-west-1
model = BedrockModel(
    region_name="us-east-1",
    model_id="us.anthropic.claude-3-5-haiku-20241022-v1:0"
)
```

Configure model

```
agent = Agent(
    model=model
)
```

Create agent

```
agent("How can you help me?")
```

Prompt

Adding system prompt

```
agent = Agent(  
    model=model,  
    system_prompt=  
        """You are an enterprise travel agent for AcmeCorp.  
        Your job is to help employees book business travel  
        that complies with company policies."""  
)  
  
agent("How can you help me?")
```


Adding system prompt

```
agent = Agent(  
    model=model,  
    system_prompt=  
        """You are an enterprise travel agent for AcmeCorp.  
        Your job is to help employees with their travel needs.  
        that can help them with their travel needs.  
        """  
)  
  
agent("How can I help you with your travel today?")
```

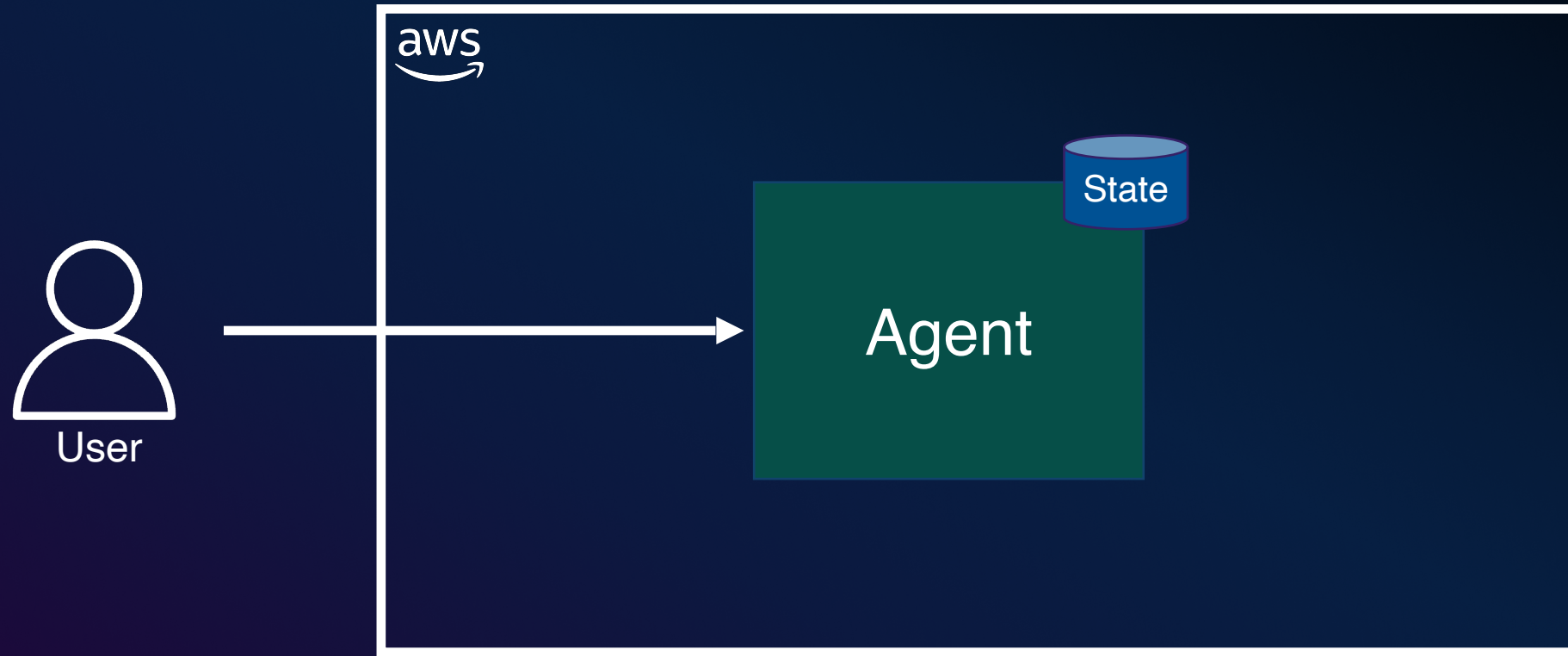
➤ `python app.py`

As an enterprise travel agent for AcmeCorp, I can help you with:

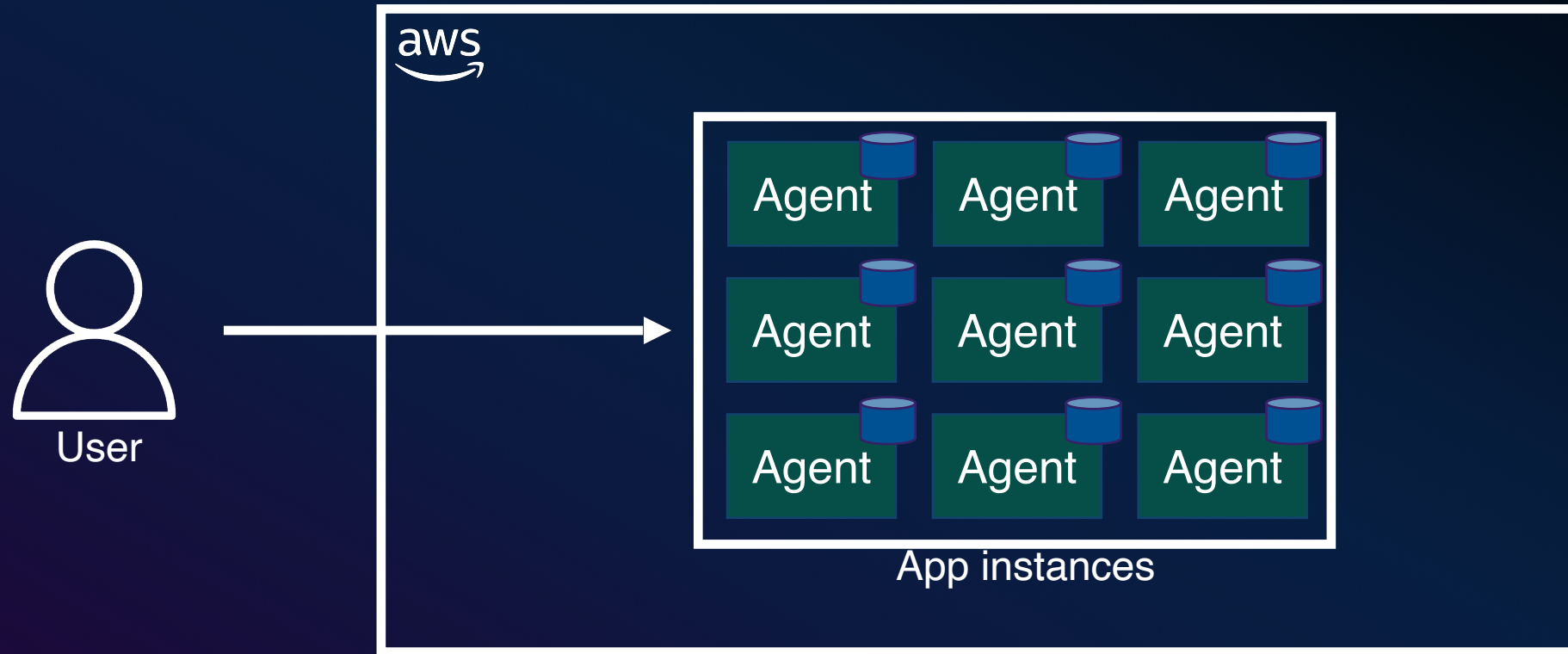
- Booking business travel arrangements
- Understanding company travel policies
- Finding approved hotels and transportation
- Ensuring travel expenses comply with corporate guidelines
- Answering questions about travel reimbursement
- Recommending cost-effective travel options

What type of business travel do you need assistance with today?

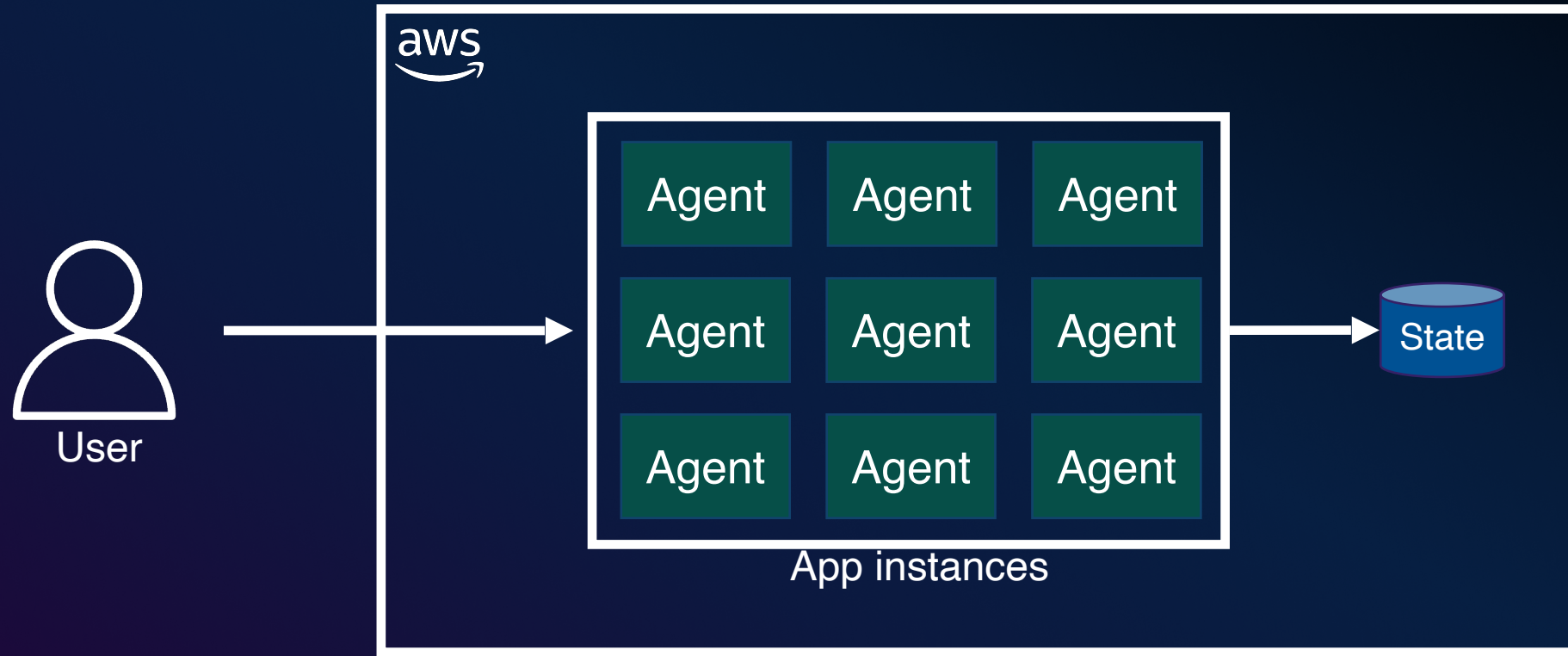
The problem with stateful apps



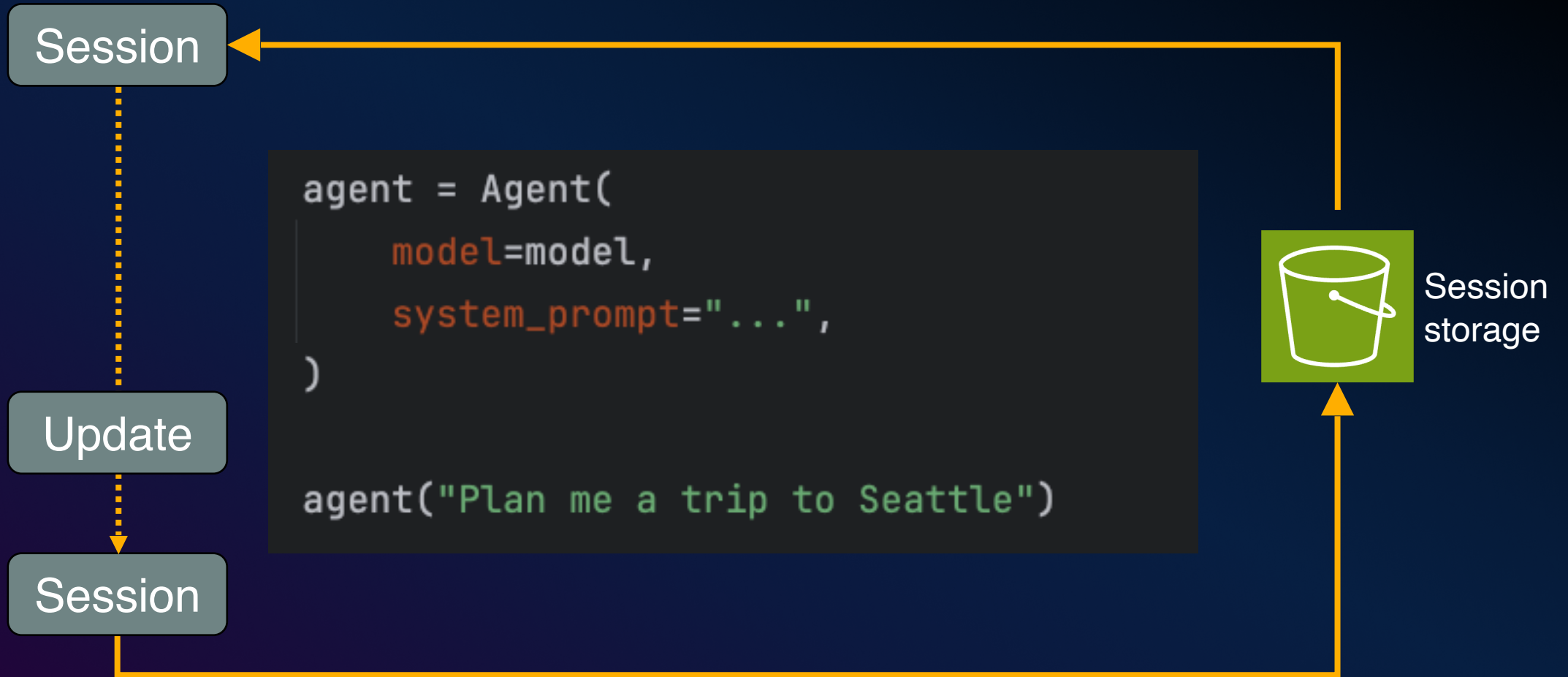
The problem with stateful apps



Externalizing the state



Externalizing the session



Externalizing the session

```
session_manager = S3SessionManager(  
    session_id="my_session",  
    bucket=SESSION_STORE_BUCKET_NAME,  
    prefix="agent_sessions"  
)
```

```
agent = Agent(  
    model=model,  
    system_prompt="...",  
)  
  
agent("Plan me a trip to Seattle")
```



Session
storage

Externalizing the session

```
session_manager = S3SessionManager(  
    session_id="my_session",  
    bucket=SESSION_STORE_BUCKET_NAME,  
    prefix="agent_sessions"  
)
```

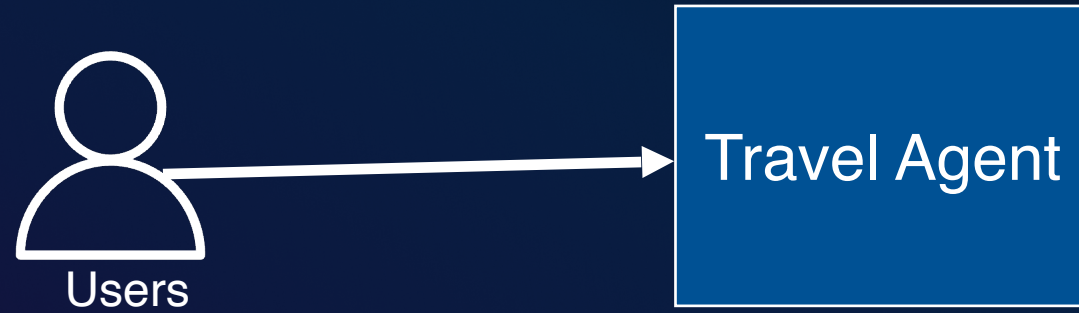
```
agent = Agent(  
    model=model,  
    system_prompt="...",  
    session_manager=session_manager,  
)
```

```
agent("Plan me a trip to Seattle")
```

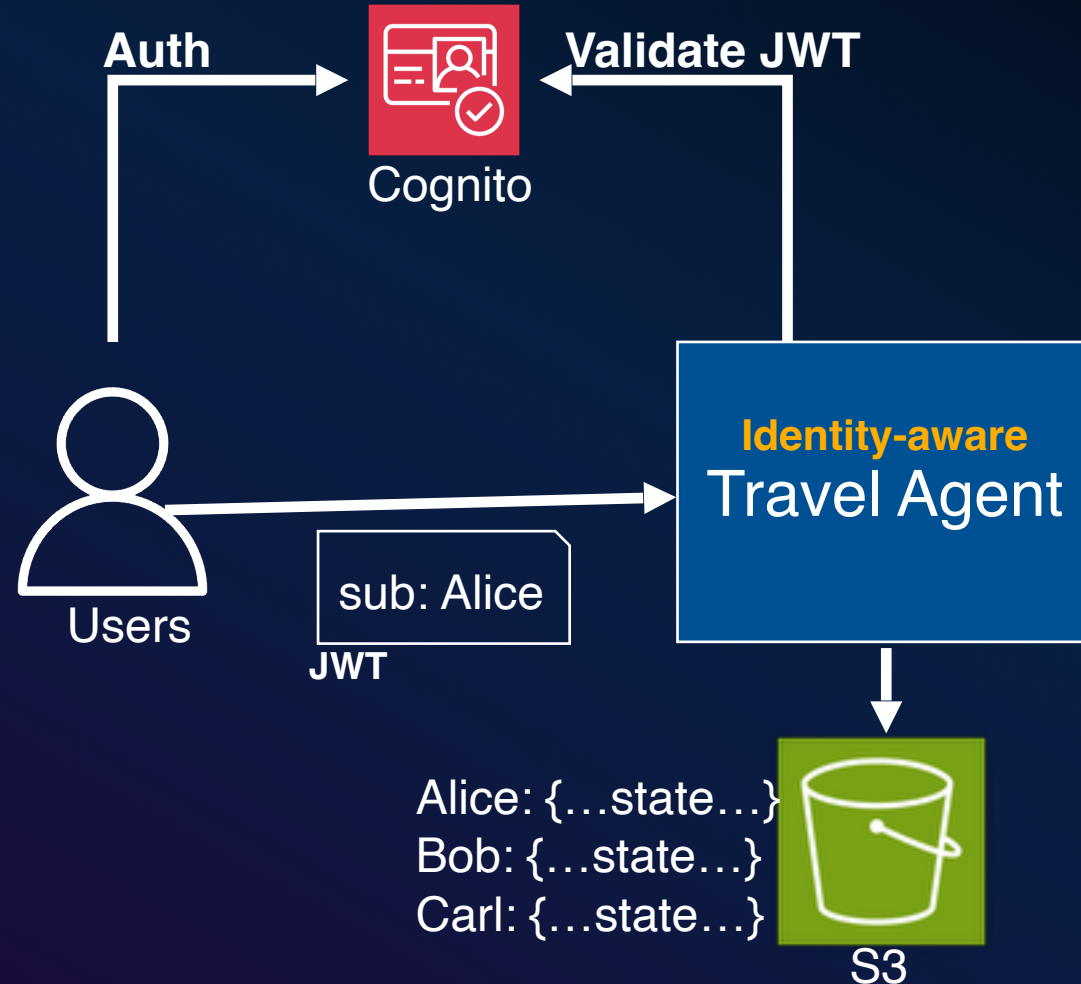


Session
storage

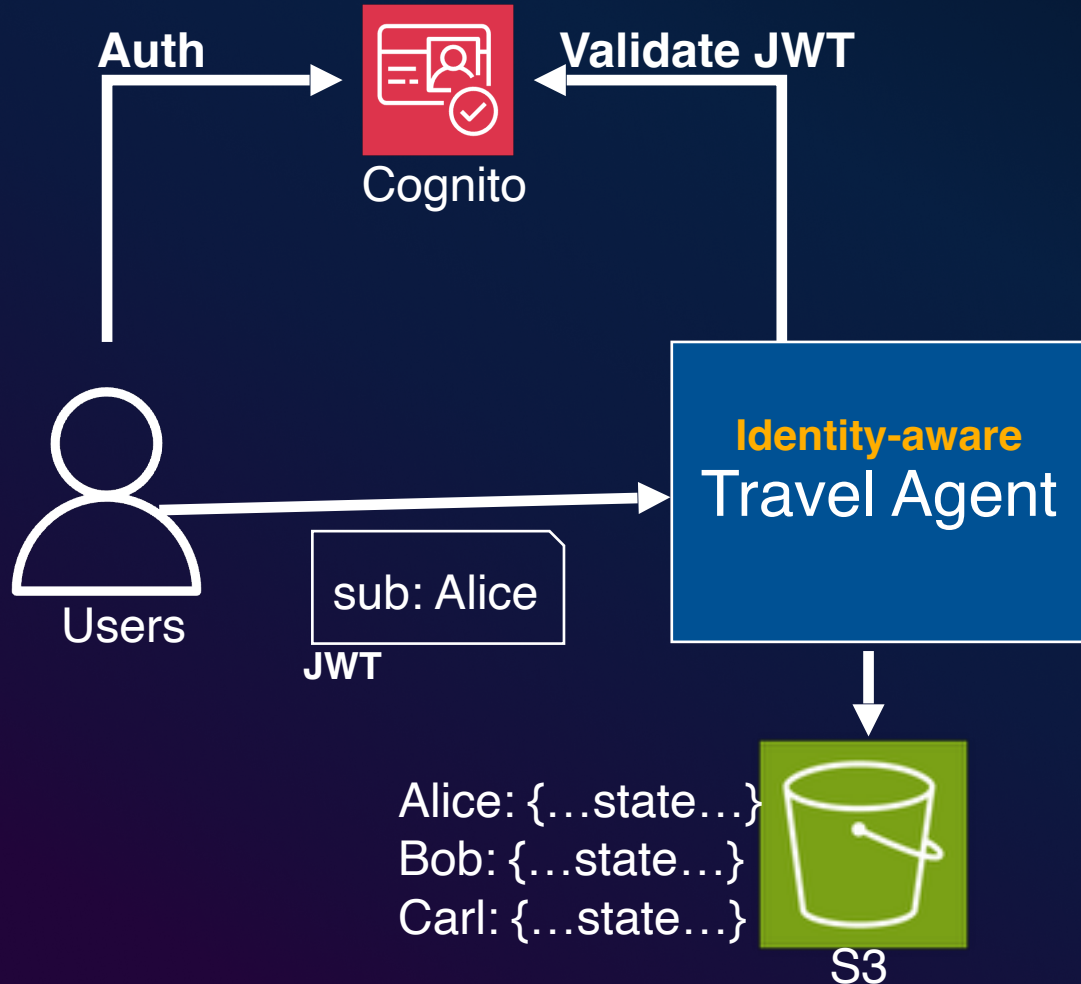
What's missing?



Authentication and authorization



Authentication and authorization



```
session_manager = S3SessionManager(  
    session_id=f"session_{user.id}",  
    bucket=SESSION_STORE_BUCKET_NAME,  
    prefix="agent_sessions"  
)
```

Hi Alice, I'm your friendly corporate travel agent! I'm here to make booking your next business trip easier. Tell me how I can help.

Deploying to AWS

“Book me a business trip to NYC”



Deploying to AWS



AWS Lambda



Amazon Elastic
Container Service (ECS)



Amazon Elastic
Kubernetes Service (EKS)

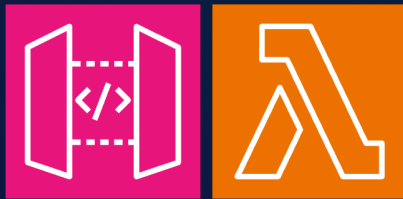


Amazon Bedrock
AgentCore (preview)

(or ANY other compute type, your Agent is just a Python app)

Building portable agents

API Gateway
and Lambda



Lambda
interface
handler

EKS/ECS
and ELB



HTTP
Interface

Bedrock
AgentsCore



Bedrock
AgentsCor
e interface

Local
testing

```
> python  
agent.py
```



Agent business logic

Building portable agents

lambda_handler.py:

```
def handler(event: dict, ctx):  
    user_id = extract_user_id(event)  
    user_prompt = json.loads(event["body"])["prompt"]  
    llm_response = call_agent(user_id, user_prompt)  
  
    return {  
        "statusCode": 200,  
        "body": json.dumps({  
            "text": llm_response  
        })  
    }
```

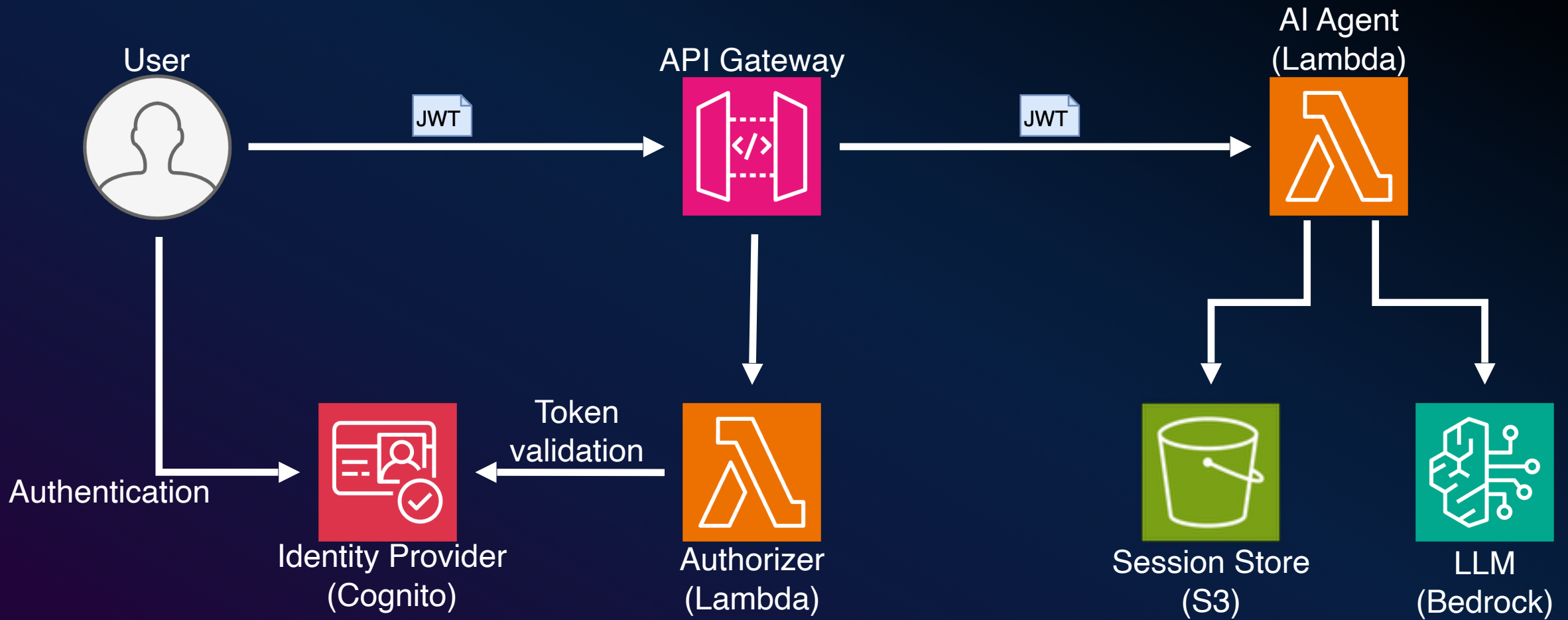
http_handler.py:

```
@app.post("/chat")  
async def chat(request: Request,  
               prompt_request: PromptRequest):  
    user_id = extract_user_id(request)  
    user_prompt = prompt_request.prompt  
    llm_response = call_agent(user_id, user_prompt)  
  
    return {"text": llm_response}
```

my_awesome_agent.py:

```
def call_agent(user_id, user_prompt):  
    prev_messages = retrieve_state(user_id)  
  
    agent = Agent(  
        system_prompt = "...",  
        tools = [...],  
        messages = prev_messages,  
    )  
  
    llm_response = agent(user_prompt)  
  
    save_state(user_id, agent.messages)  
  
    return llm_response
```


Deploying to AWS

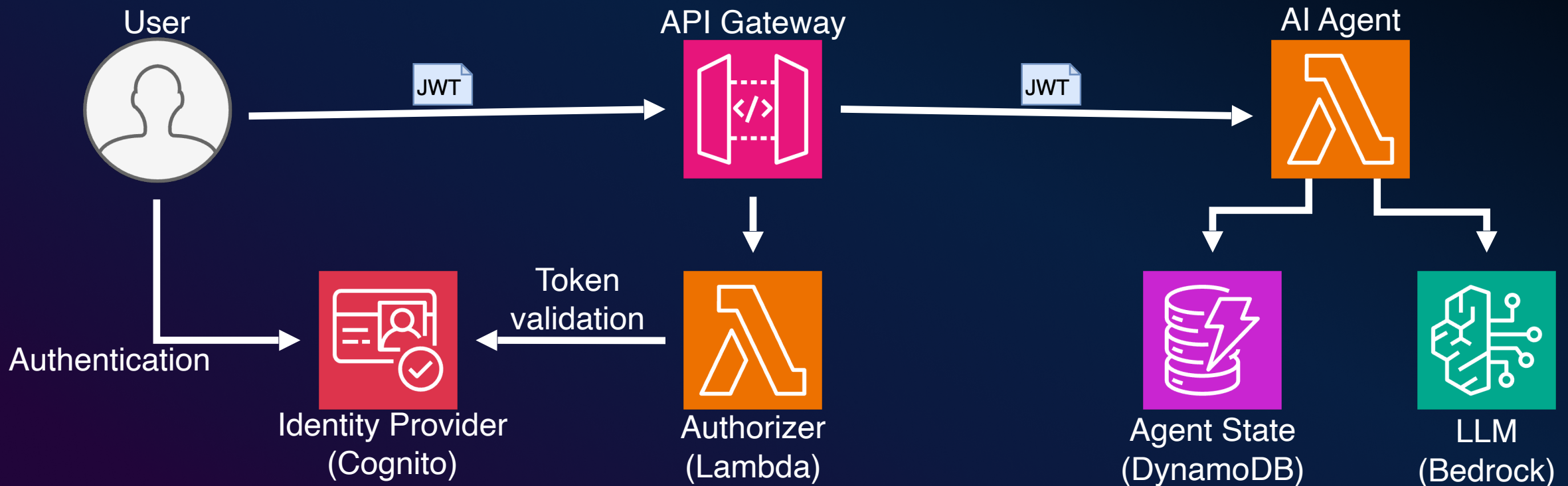


Deploying to AWS

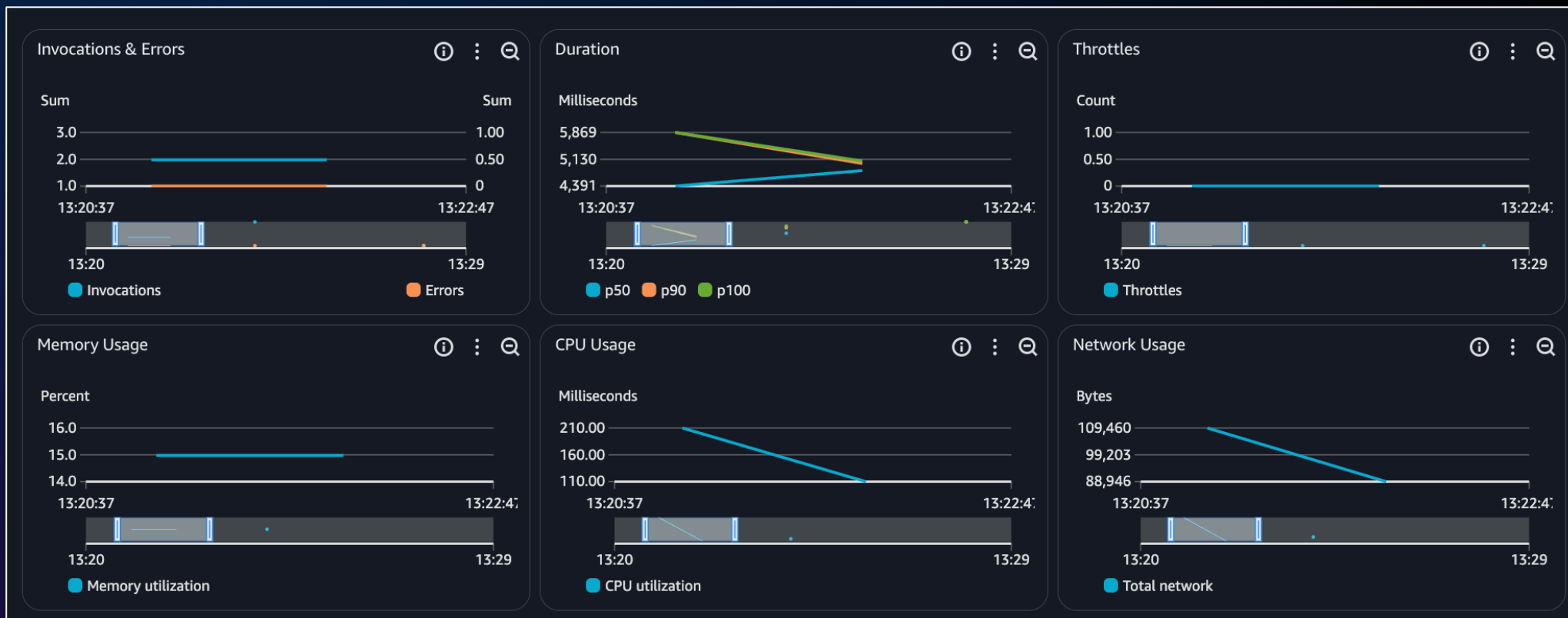
But what about observability?
Security? Governance? CI/CD?

**Everything you know about
building applications and APIs
on AWS is still applicable!**

Deploying to AWS

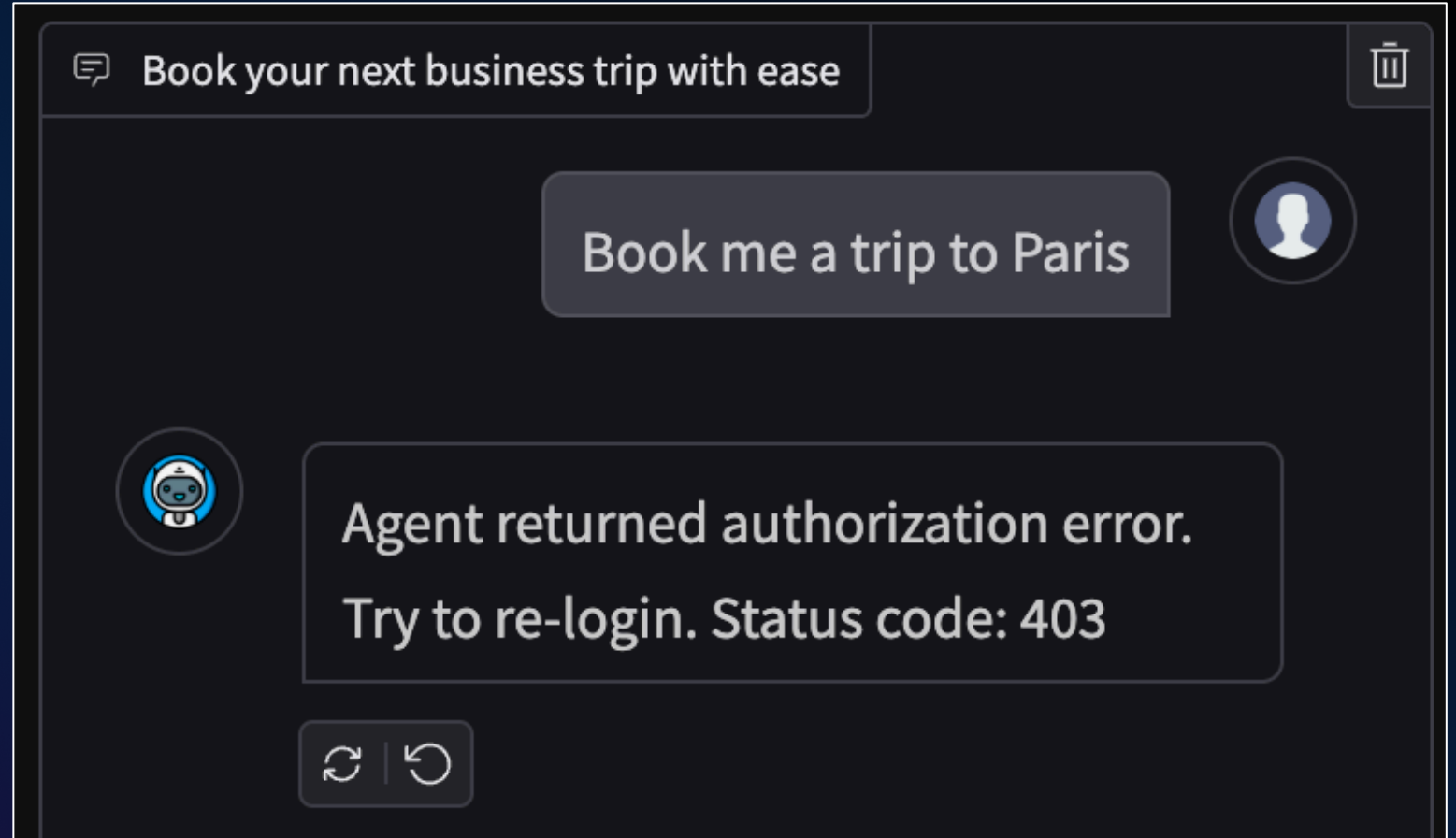
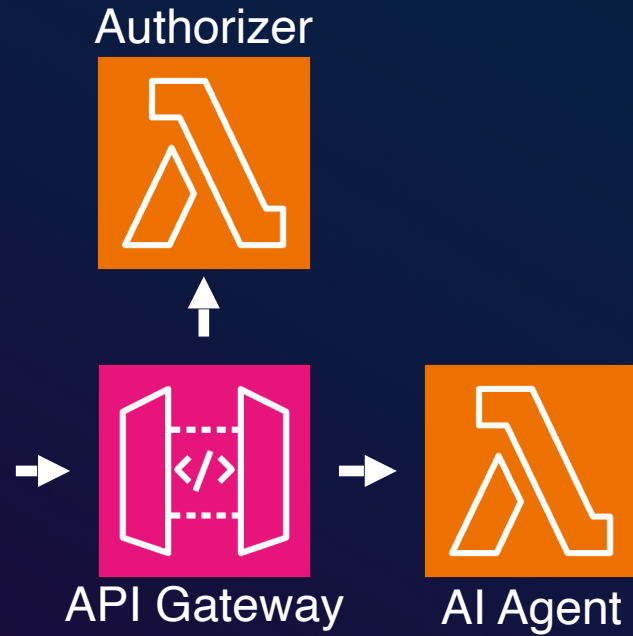


Observability



```
2025-06-20 18:05:10,757 INFO app.py:29 :: jwt parsed. user.id=74c8a428-7001-70db-d017-5725bd9efc91 user.name=Alice
2025-06-20 18:05:10,757 INFO app.py:43 :: composite_prompt=User name: Alice
User IP: 70.113.53.201
User prompt: Book me a trip to Paris
2025-06-20 18:05:10,757 INFO agent.py:10 :: user.id=74c8a428-7001-70db-d017-5725bd9efc91, user.name=Alice
2025-06-20 18:05:10,757 INFO agent_state_manager.py:22 :: restoring agent state for user.id=74c8a428-7001-70db-d017-5725bd9efc91
```


Authorization



Demo time

strands-on-lambda.auth.us-east-1.amazoncognito.com

Sign in with your username and password

Username

Password

[Forgot your password?](#)

Sign in

Conclusion

1

Agents are applications calling LLMs in loops. Your existing knowledge is your accelerator.

2

Use the right tool for the job. Decouple when needed, use proper protocols.

3

AWS services provide built-in enterprise-grade capabilities for building AI agents.

Continue your learning journey



Thank you!



Please complete the session survey in the mobile app

Anton Aleksandrov

 antonal80

 aal80.github.io/whoami

Heeki Park

 heekipark

 heeki.medium.com