

Building AWS AI Agents
using
Bedrock Multi-Agent Framework

AWS AI Agents - Detailed Course Overview

- [Section 1 - Course Introduction](#)
- [Section 2 - Evolution of AI Agents](#)
- [Section 3 - Deep Dive - Amazon Bedrock Multi-Agent Framework](#)
- [Section 4 - Use Case 1 - Build a Hotel Booking Agent](#)
- [Section 5 - Use Case 2 - Enterprise Travel Agent \(Multi-Agent\)](#)
- [Section 6 - Refresher - Generative AI, Amazon Bedrock and Knowledge Bases](#)
- [Section 7 - Refresher - Basic Python, AWS Lambda and Boto3](#)

***Section on :
Course Introduction***





*Course Outline & Few Important Tips –
Promise to complete in few mins 😊*

Course Outline and Important Tips



1. The focus of this course is helping you learn build Multi-Agent apps using Amazon Bedrock Multi-Agent Framework
2. At the moment, this course **does not** show any use case built using Lang Graph or CrewAI but will in future.
3. **Pre-Requisites** – **Need basic AWS Knowledge**, Familiarity with LLM's, Prompt Engineering, Bedrock Knowledge Bases.

Incase of implementation of hands-on use cases, Python and AWS Lambda knowledge. **But have provided basic refresher videos.**

Udemy Amazon Bedrock :

<https://www.udemy.com/course/amazon-bedrock-aws-generative-ai-beginner-to-advanced/?referralCode=A02153572B8864F928E7>

Content creator and copyright : Rahul Trisal. Please do not copy



AI Agents -
Core Capabilities

UC 1 – Hotel
Booking Agent
(Single Agent)

Refresher on :
GenAI, Prompt
Engineering &
Bedrock KB's

Deep Dive –
Amazon Bedrock
Agents

UC 2 – Enterprise
Travel Agent
(Multi-Agent)

Refresher on :
Python& AWS
Lambda

Comprehensive guide to get you started on **Agentic AI Revolution**

Course Outline and Important Tips

1. Course Content Download
2. This is a new course, incase of any discrepancies or issue with videos or missing content, please mail me at trisalrahul@gmail.com
3. Connect - www.linkedin.com/in/rahul-trisal or Instagram - [Rahul Trisal](#) for Course & AI Trends & Free Coupons
4. Course Cost – **Warning** : Bedrock Knowledge Bases and Agents
5. Share your success – Tag me on LinkedIn for wider reach
6. Course Rating and Feedback



Course Rating and Feedback – Super Important !!!

Udemy | AWS AI Agents - Complete Course [HANDS-ON]

★ Leave a rating Your progress

Course content AI Assistant Beta

Section 4: Use Case 1 - Hotel Booking Agent
1 / 17 | 2hr 17min

- ☐ 17. Business Use Case - Hotel Booking Agent
2min Resources
- ☐ 18. Use Case Solution Architecture
6min
- ☒ 19. Agent Creation
9min Resources
- ☐ 20. Agent Integration with Knowledge Bases for Room Information
13min Resources
- ☐ 21. Hotel Room Availability - Arch and DynamoDB creation

AI Agents – Key Characteristics

The diagram illustrates the architecture of AI Agents. At the center is a purple box labeled 'Agent (Powered by LLM)'. To its left is a 'Tools' box, which is connected to a list of tools: 'Tool to access DB', 'RAG - SharePoint', 'Code Interpreter', 'Weather API', and 'Travel Booking'. An example 'AWS Lambda' is noted above the tools. Above the agent are 'Short Term Memory' and 'Long term Memory' boxes. To the right of the agent is a 'Planning' box, which connects to 'Chain of Thought', 'Tree of Thought', and 'ReAct'. Below the agent is an 'Action' box. A large play button is overlaid on the agent box. The entire process is framed by 'Guardrails - Toxicity, Hallucination and PII Redaction' and 'Agent Communication'.

How would you rate this course?

Select Rating



Incase of any issues with course, send me an email on trisalrahul@gmail.com

Course Rating and Feedback – Super Important !!!

Click Here to Rate the course

udemy | AWS AI Agents - Complete Course [HANDS-ON]

★ Leave a rating | Your progress ▾

Course content | AI Assistant Beta

Section 4: Use Case 1 - Hotel Booking Agent
1 / 17 | 2hr 17min

- ☐ 17. Business Use Case - Hotel Booking Agent
2min | Resources ▾
- ☐ 18. Use Case Solution Architecture
6min
- ☒ 19. Agent Creation
9min | Resources ▾
- ☐ 20. Agent Integration with Knowledge Bases for Room Information
13min | Resources ▾
- ☐ 21. Hotel Room Availability - Arch and DynamoDB creation

AI Agents – Key Characteristics

AI Agents

Example : AWS Lambda

Tools to access DB, RAG - SharePoint, Code Interpreter, Weather API, Travel Booking

Tools

Agent (Powered by LLM)

Short Term Memory, Long term Memory

Planning

Chain of Thought, Tree of Thought, ReAct

Action

Guardrails – Toxicity, Hallucination and PII Redaction

Agent Communication

0:00 / 0:00 | 1x

Content creator and copyright : Rahul Trisal. Please do not copy



Udemy Platform :
Quick Tips to Optimize Learning

Udemy Tips

- Adjust the Speed of course based on your comfort

***Section on :
Evolution of AI Agents***



What are AI Agents – Lets understand through a Use Case



Hey I am John

I work at Micro-cool Corporation

I am planning a 2-week Skiing vacation from mid of December if I have leaves

Need help with vacation planning.

Can a GenAI App built using Large Language Model help John ?

John's Ask :

I am planning a 2-week Skiing vacation from mid of December if I have sufficient leaves

Collection of Tasks & Decisions

| | |
|-------------------------------------|-------|
| <input checked="" type="checkbox"/> | _____ |
| <input checked="" type="checkbox"/> | _____ |
| <input type="checkbox"/> | _____ |
| <input type="checkbox"/> | _____ |
| <input type="checkbox"/> | _____ |
| <input type="checkbox"/> | _____ |

- ❖ Check my Leave balance
- ❖ Research locations for skiing
- ❖ Check weather report for location
- ❖ Book flight
- ❖ Book hotel

Limitations of LLM's to handle complex user requests

Home \ Anthropic x Weather Information Limitation x +

← → ↻ 🌐 claude.ai/chat/de4c511a-2a68-4b5e-ab7a-74c884117625

Claude 🗨 Weather Information Limitations ▾

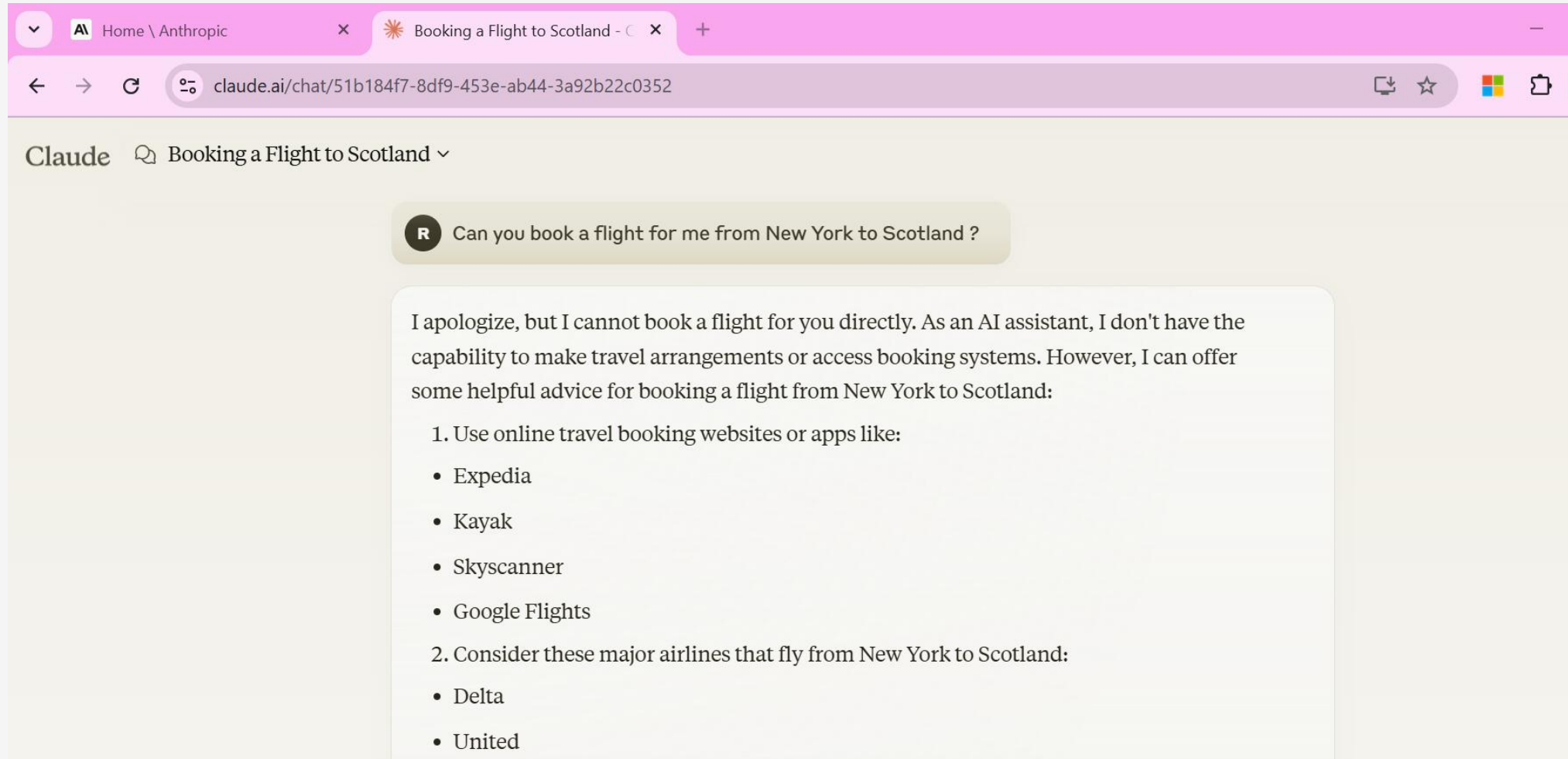
R What is the weather in New York today ?

I apologize, but I cannot provide real-time weather information. As an AI, I don't have access to current weather data or live updates. To get the most up-to-date weather information for New York today, I recommend:

1. Checking a weather website or app like Weather.com, AccuWeather, or the National Weather Service
2. Looking at a local weather forecast on TV or radio
3. Checking a weather app on your smartphone
4. Visiting a local weather website specific to New York

These sources can give you the most current and accurate weather conditions and forecast for New York today.

Limitations of LLM's to handle complex user requests



Limitations of LLM's to handle complex user requests

User Request

I am planning a 2-week Skiing
vacation from mid of December
if I have sufficient leaves

Micro-cool GenAI App



- ❖ Check my Leave balance ❌
- ❖ Research locations for skiing ✅
- ❖ Check Weather report for location ❌
- ❖ Book flight ❌
- ❖ Book hotel ❌

Limitations and Challenges of LLM based Applications

- Answer generic questions based on publicly available training data **but not company specific data**
- No way to **access current data or external API's** – Today's weather, book hotel etc.
- Answer only questions based **on training data with cut off date**



Limitations of **LLM's + RAG** to handle complex user requests

User Request

Micro-cool GenAI App + RAG






I am planning a 2-week Skiing
vacation from mid of December
if I have sufficient leaves



Leave Balance



HR Data Source

- ❖ Check my Leave balance 
- ❖ Research locations for skiing 
- ❖ Check Weather report for location 
- ❖ Book flight 
- ❖ Book hotel 

How **AI Agent** come to John's rescue

User Request

I am planning a 2-week Skiing
vacation from mid of December
if I have sufficient leaves

Agentic App



Agents
(Powered By LLM's)

1. Task Decomposition & Planning

- ❖ Check my Leave balance
- ❖ Research locations for skiing
- ❖ Check Weather report
- ❖ Book flight
- ❖ Book hotel

2. Tools & Actions

- ❖ Retrieve HR Data using RAG
- ❖ Search the Web for skiing location
- ❖ Call Weather API
- ❖ Call Hotel.com API

3. Memory for end user interaction

4. Guardrails – Toxicity, Hallucinations

5. Agent Communication

Core Capabilities of AI Agents

1. Planning

Dynamic Task Planning

2. Tools & Action

Access any data via API's

3. Memory

User Interaction

4. Guardrails

Toxicity & Hallucinations

5. Communication

Single or Multi-Agent

AI Agents – Key Characteristics

AI Agents have generally 5 Key characteristics

AI Agents

Planning

Memory

Tools & Action

Guardrails

Communication
(Single/Multi-Agent)

What does Agentic Mean ?



Agentic Definition

Stanford psychologist Albert Bandura coined the term to describe **people** who are **self-organizing, proactive, self-reflective, and self-regulating.**

What is Agentic AI ?



What Is Agentic AI?



Agentic AI uses **sophisticated reasoning and iterative planning** to **autonomously** solve complex, multi-step problems.



Intelligent Autonomous Systems



Plan, Reason and Act



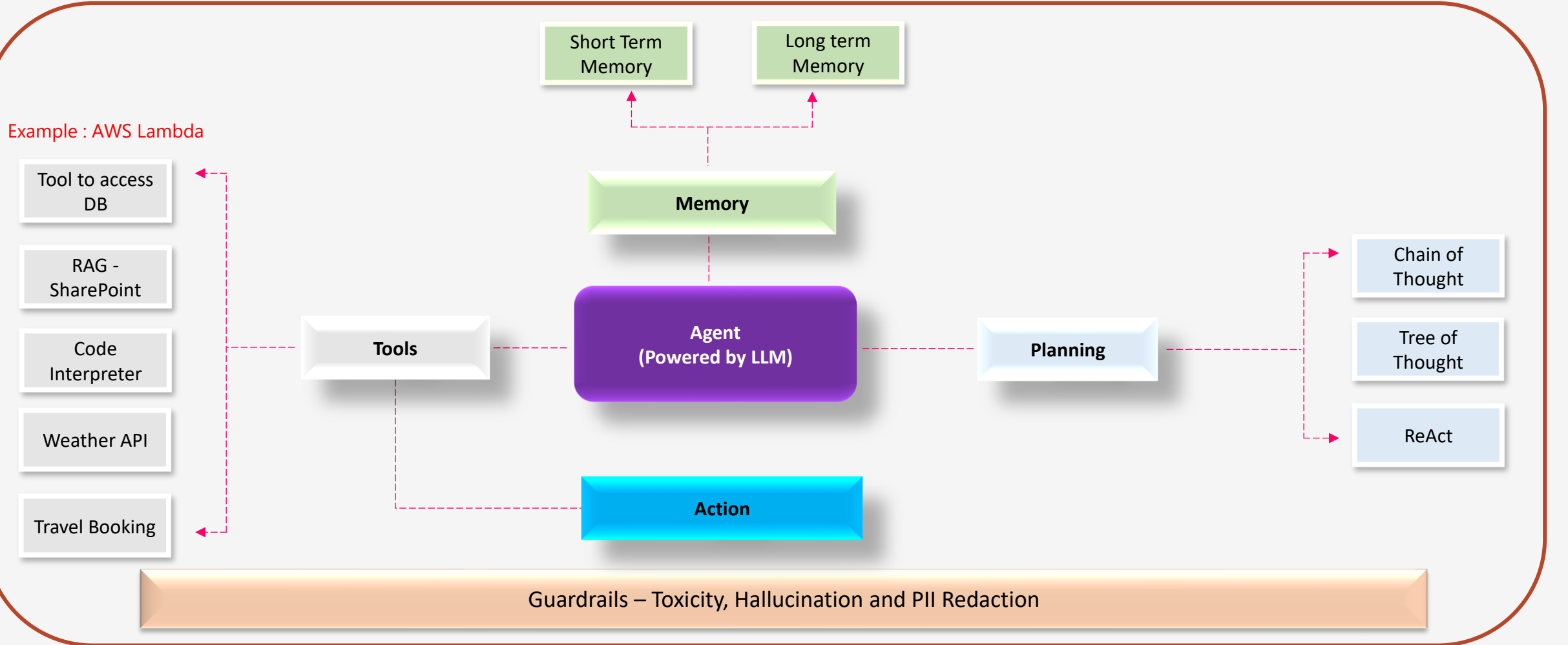
Are able to access enterprise data



Are able to use Tools

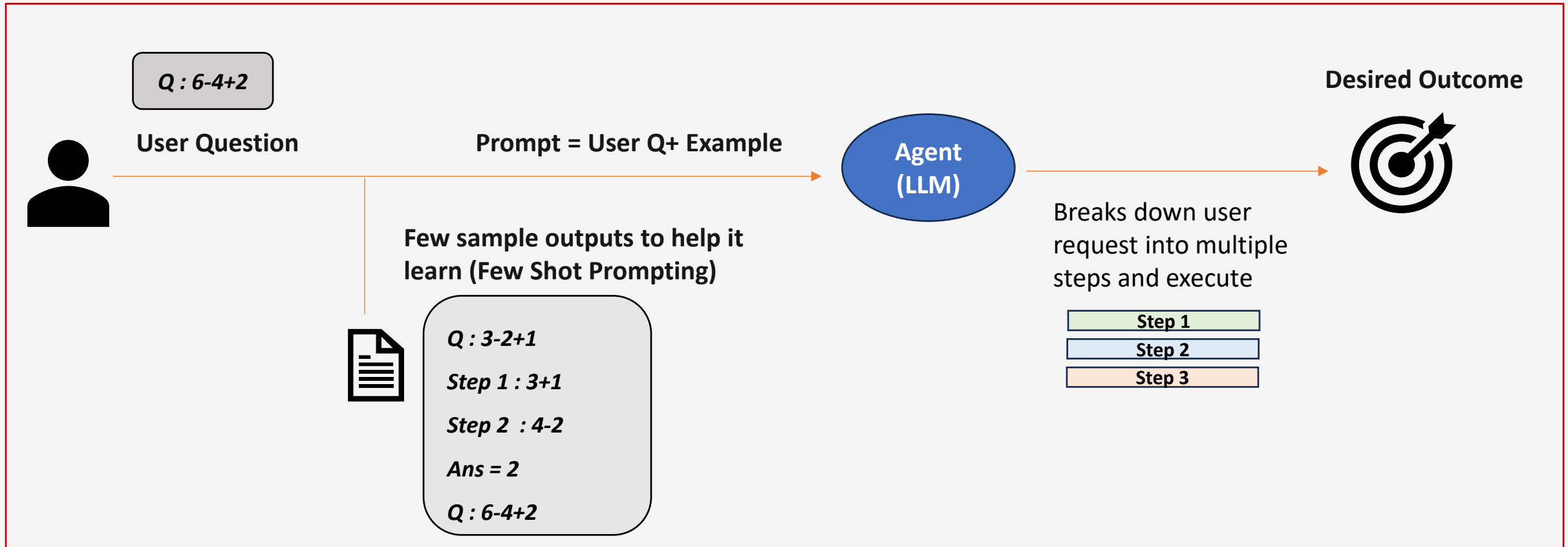
AI Agents – Key Characteristics

AI Agents



Planning by Agents


- Agent is powered by a Large Language Model(LLM) – Brain behind the Agents
- **How and why does Agent do the planning ?**




AI Agent - Planning

Two key approaches AI Agent planning

1. Planning without Feedback

- Chain-of-Thought (CoT) 
- Tree of Through Reasoning (ToT)

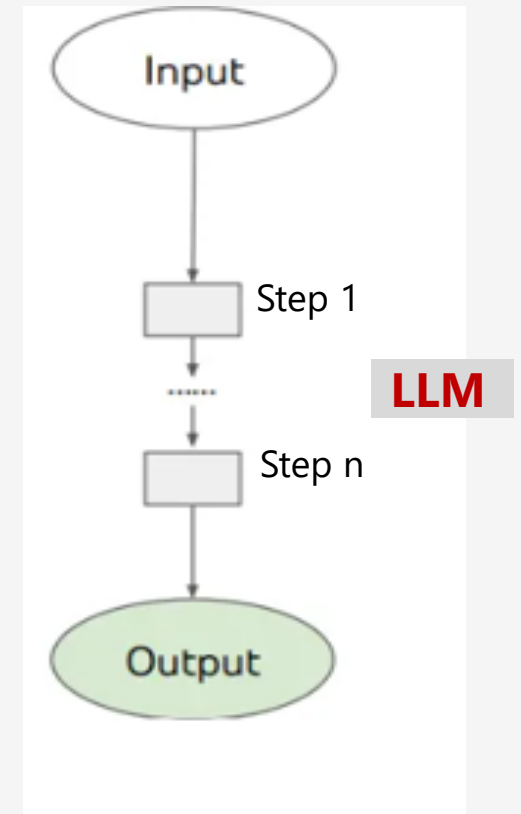
2. Planning with Feedback

- ReAct or Reasoning + Action 
- Reflexion

1. Planning - Planning without Feedback

Chain-of-thought prompting (CoT)

- CoT prompting enables complex reasoning capabilities for the LLM through intermediate reasoning steps
- CoT prompting has shown to **improve capability of LLM** to answer questions involving **arithmetic and commonsense reasoning**



Planning without Feedback – Chain of Thought Reasoning (CoT)

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

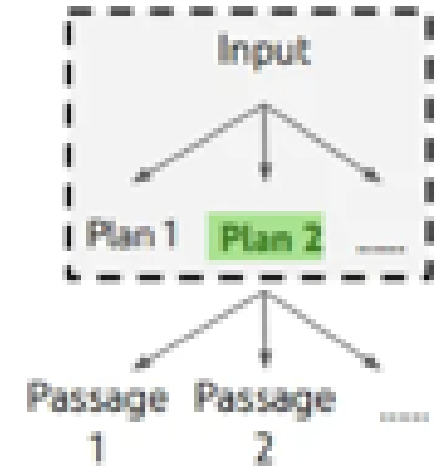
Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

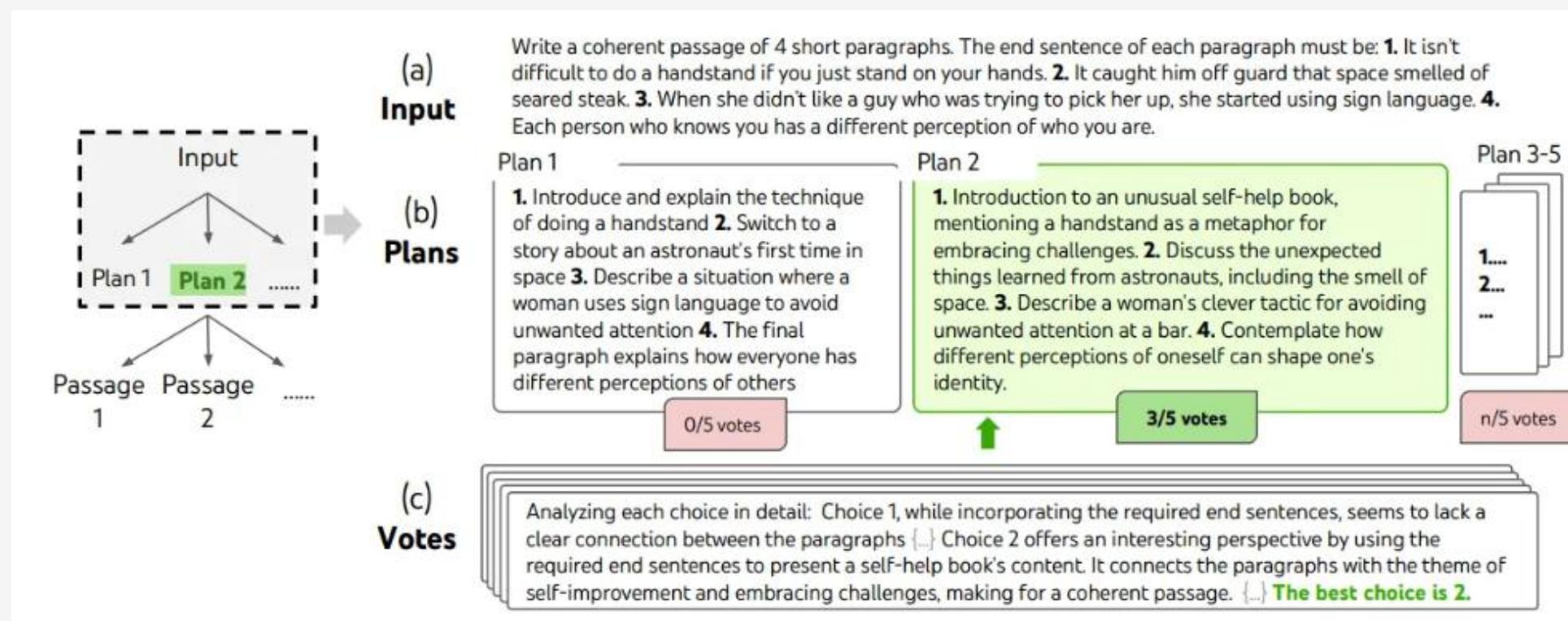
1. Planning without Feedback – Tree of Thought

Tree of Through Reasoning

- First decomposes the problem into multiple thought steps
- Generates multiple thoughts per step, creating a tree structure.

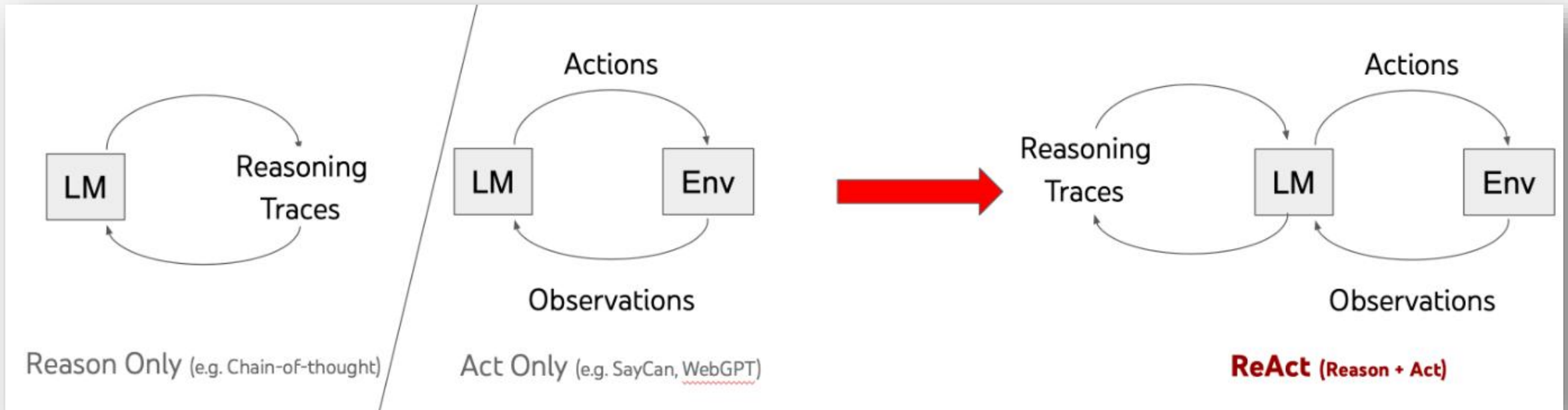


1. Planning without Feedback - Tree of Thought



2. Planning with Feedback – ReAct (Reason + Act)

- ReAct is an advanced **prompting technique** that combines **Reasoning** and **Action** within a prompt.
- **Good for scenarios where external or internal data** is required to execute tasks.
- Dynamic execution - Replanning

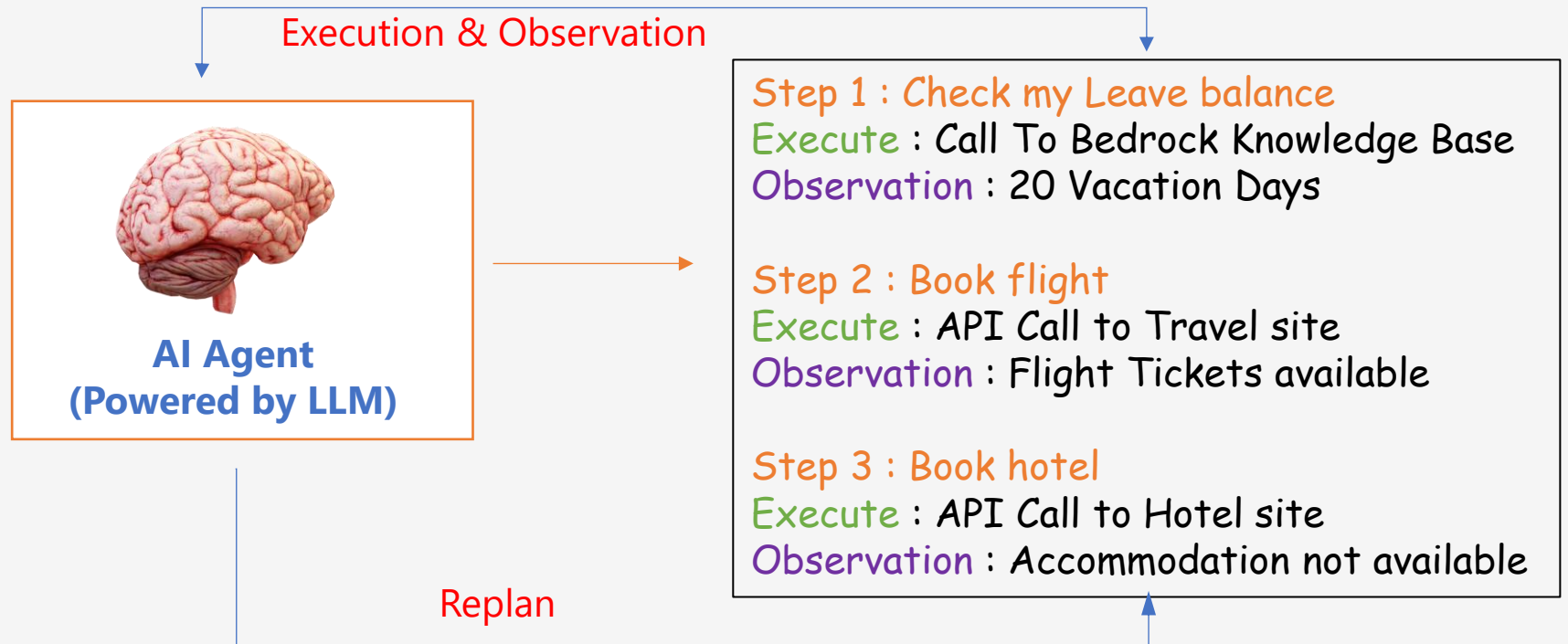


ReAct Example

User Request

I am planning a 2-week
Skiing vacation from
mid of December

Micro-cool AI Agent



Agents – Planning Capability Demo



AI Agent **Planning** Demo

User Request



I am planning a 5-day
Beach vacation from
mid of December

Micro-cool AI Agent



AI Agent
(Powered by LLM)

Execution & Observation

Step 1 : Check my Leave balance
Execute : **API Call to HR System**
Observation : 5 Vacation Days

Step 2 : Book hotel
Execute : **API Call to Hotel site**
Observation : Accommodation not available

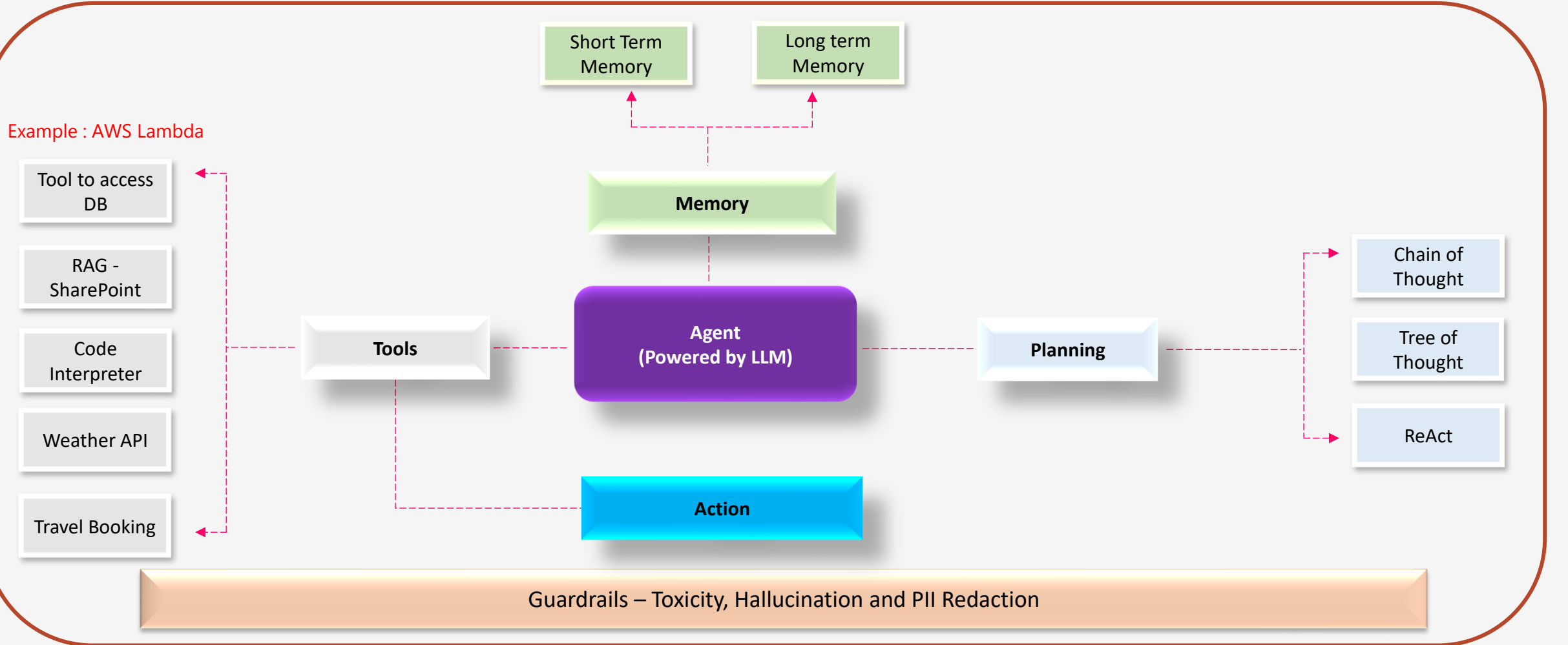
Replan

Questions

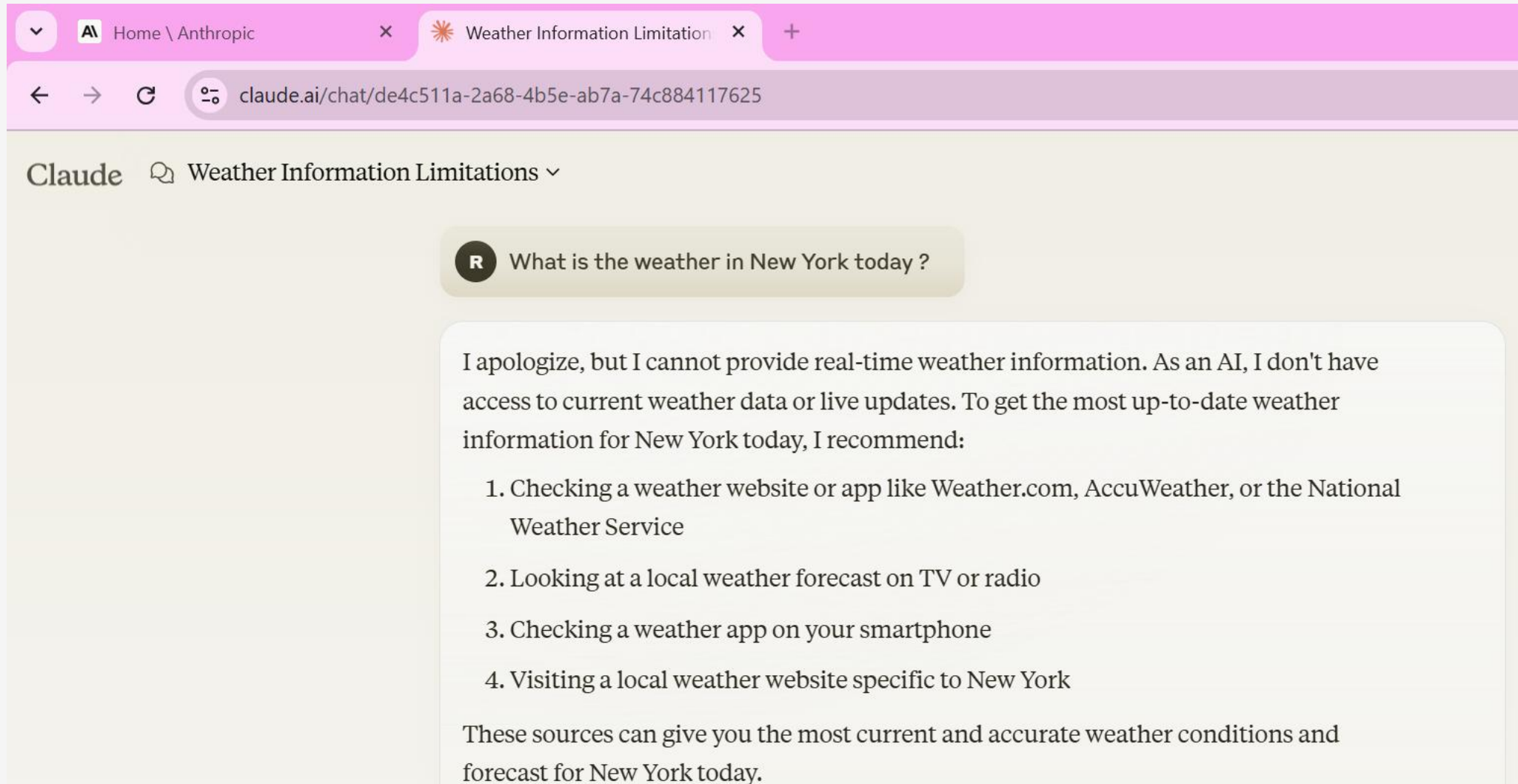
1. Book a hotel room for me with **employee id - 251205 if my leave balance is more than 5 days** with following details : Check-in date - 2025-12-26, 2. Room Type - Sea View 3. Guest Name – John Doe 4. Number of Nights – 5
2. Book a hotel room for me **with employee id - 251205 if my leave balance is more than 3 days** : 1. Check-in date - 2025-12-26, 2. Room Type - Sea View, 3. Guest Name – Jane Doe, 4. Number of Nights - 3

AI Agents – Key Characteristics

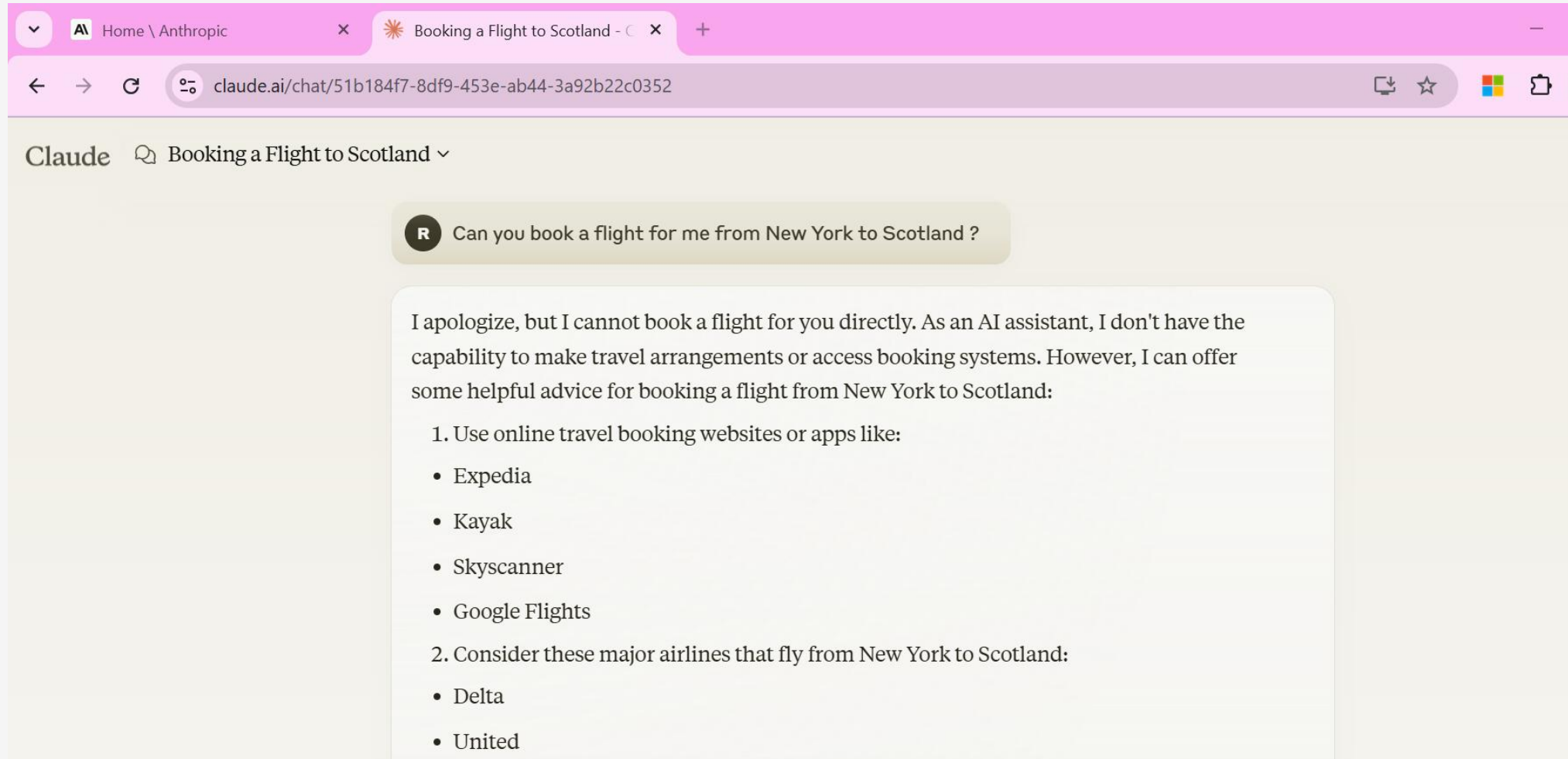
AI Agents



Limitations of **LLM's** to access internal or external data sources



Limitations of LLM's to access internal or external data sources



2. Tools Calling or Function Calling

- Function calling or Tools Calling enables Agents to connect and retrieve data from **external and internal source data and systems** to enhance the capabilities of Agents.
 - Google Search
 - Weather API
 - Extract data from enterprise Database such as MySQL
 - Retrieval Augmented Solution(RAG) based app
 - Code Interpreter for running custom code



2. Tools Calling or Function Calling

User Request

I am planning a 2-week
Skiing vacation from
mid of December

Micro-cool Agent App



Tool or Function Calling

Tool 1 - Call Weather API

Execute : Function written in AWS Lambda

Tool 2 - Call Hotel Booking API

Execute : Function written in AWS Lambda

Tool 3 - Code Execution

Execute : Ability to execute code generated by Agent (LLM)

Tool 4 - Call to HR system using RAG

Execute : RAG app - Bedrock Knowledge Bases

How Tools calling works ?

Prompt template editor

```
1 $instruction$
2
3 You have been provided with a set of tools to
  answer the user's question.
4 You may call them like this:
5 <function_calls>
6   <invoke>
7     <tool_name>$TOOL_NAME</tool_name>
8     <parameters>
9       <$PARAMETER_NAME>$PARAMETER_VALUE
10      </$PARAMETER_NAME>
11     ...
12   </parameters>
13 </invoke>
14 </function_calls>
15 Here are the tools available:
16 <tools>
17   $tools$
18 </tools>
19
20
21 You will ALWAYS follow the below guidelines when
  you are answering a question:
22 <guidelines>
23   - Never assume any parameter values while
    invoking a function.
24 $ask_user_missing_information$
25   - Provide your final answer to the user's
    question within <answer></answer> xml tags.
26   - Think through the user's question, extract all
```

- Define the tools available to the Agent (powered by LLM)
- Automatically construct a **special system prompt from the tool definitions, tool configuration, and any user-specified system prompt.**

```
import anthropic

client = anthropic.Anthropic()

response = client.messages.create(
    model="claude-3-5-sonnet-20241022",
    max_tokens=1024,
    tools=[
        {
            "name": "get_weather",
            "description": "Get the current weather in a given location",
            "input_schema": {
                "type": "object",
                "properties": {
                    "location": {
                        "type": "string",
                        "description": "The city and state, e.g. San Francisco, CA"
                    }
                },
                "required": ["location"],
            }
        },
    ],
    messages=[{"role": "user", "content": "What's the weather like in San Francisco"}])
```

AI Agents – Tool Calling Demo

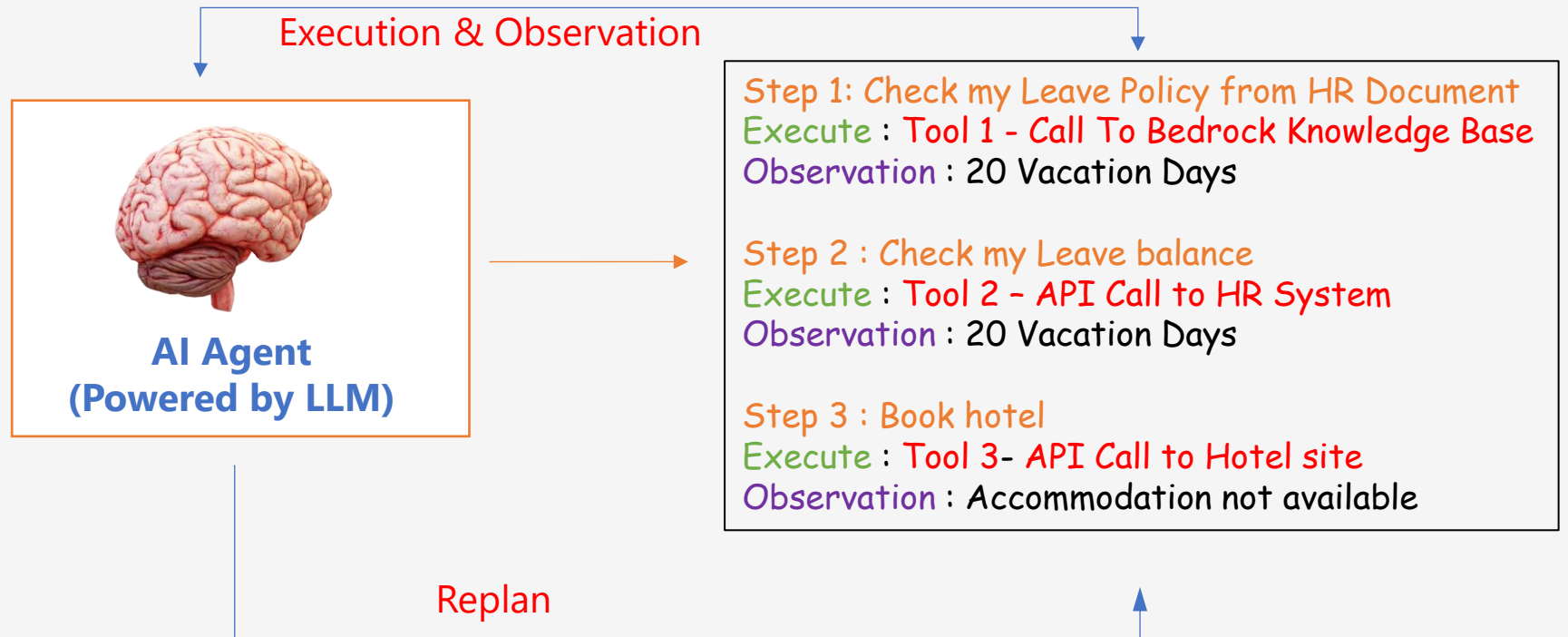


AI Agent Tool Calling Demo

User Request

I am planning a 5-day
Beach vacation from
mid of December

Micro-cool AI Agent

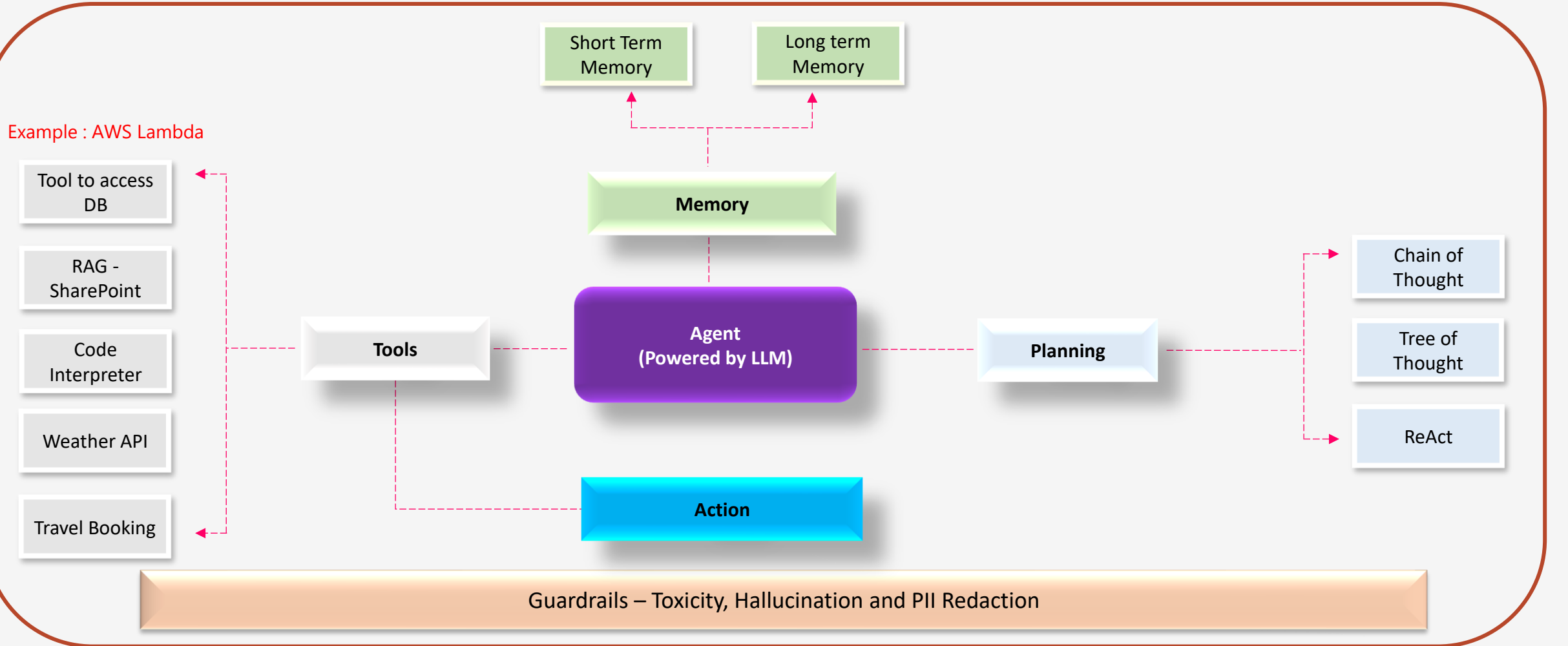


Questions

1. How many Paternity Leaves are available ?
2. Book a hotel room for me with employee id - 251205 if my leave balance is more than 3 days with following details : 1. Check-in date - 2025-12-27, 2. Room Type - Sea View, 3. Guest Name – Jane Doe, 4. Number of Nights - 3

AI Agents – Key Characteristics

AI Agents

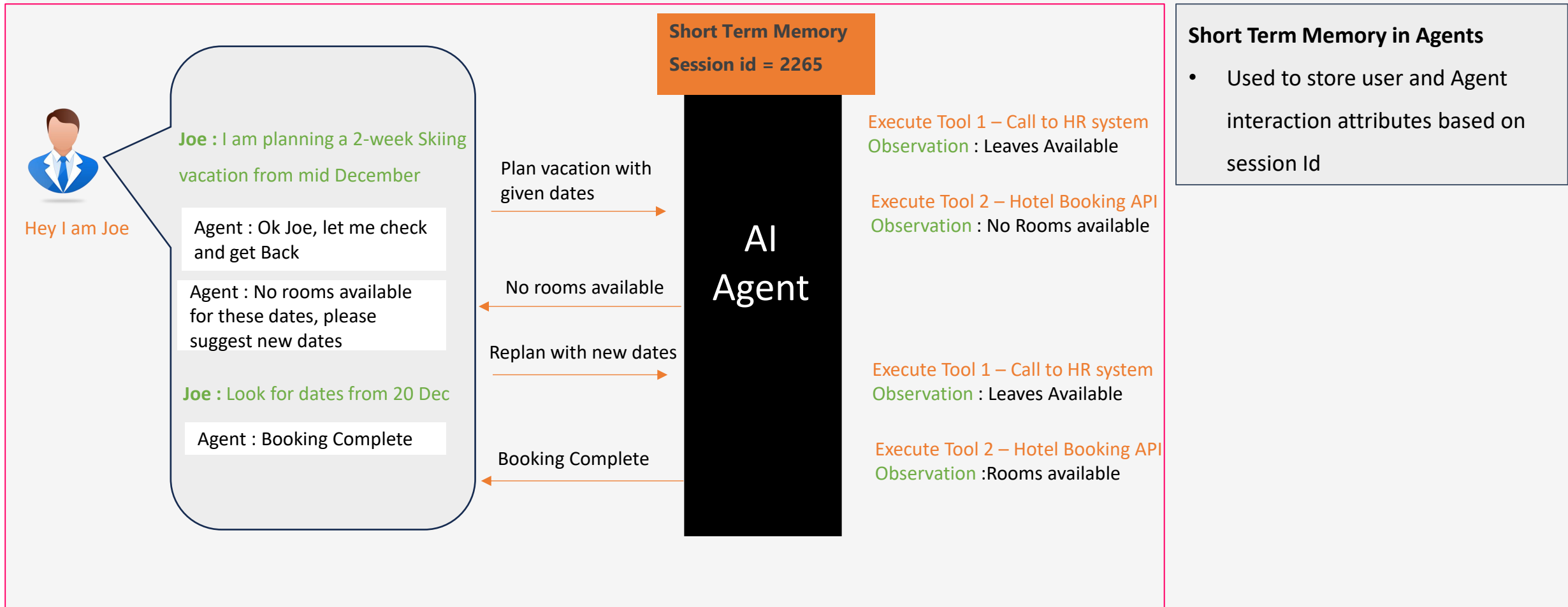


4. Agentic Memory – Short Term and Long Term

- Remember – LLM's are Stateless (No memory)
- Short term memory
 - Agent retains the conversation and user input during the particular session
- Long term memory
 - Allows agent to retain conversation across all previous interactions
 - Summarize interaction across all previous sessions
 - 1-365 days



4. Agent Memory : Short Term Memory



4. Agent Memory : Long-Term Memory



Hey I am Joe

Interaction in Dec 2024

Joe : I am planning a 2-week vacation
from mid of December
Hotel rooms should be no smoking

Interaction in June 2025

Joe : I am planning a 2-week vacation
to Egypt

AI
Agent

Long Term Memory in AI Agents

- User Preferences based on historical interaction
- Broader Contextual Knowledge let's say Micro- cool employees always use Hilton Hotels
- Every interaction summarized and stored once completed
- 1-365 days
- Loaded every time new interaction takes place

Summarize the interaction

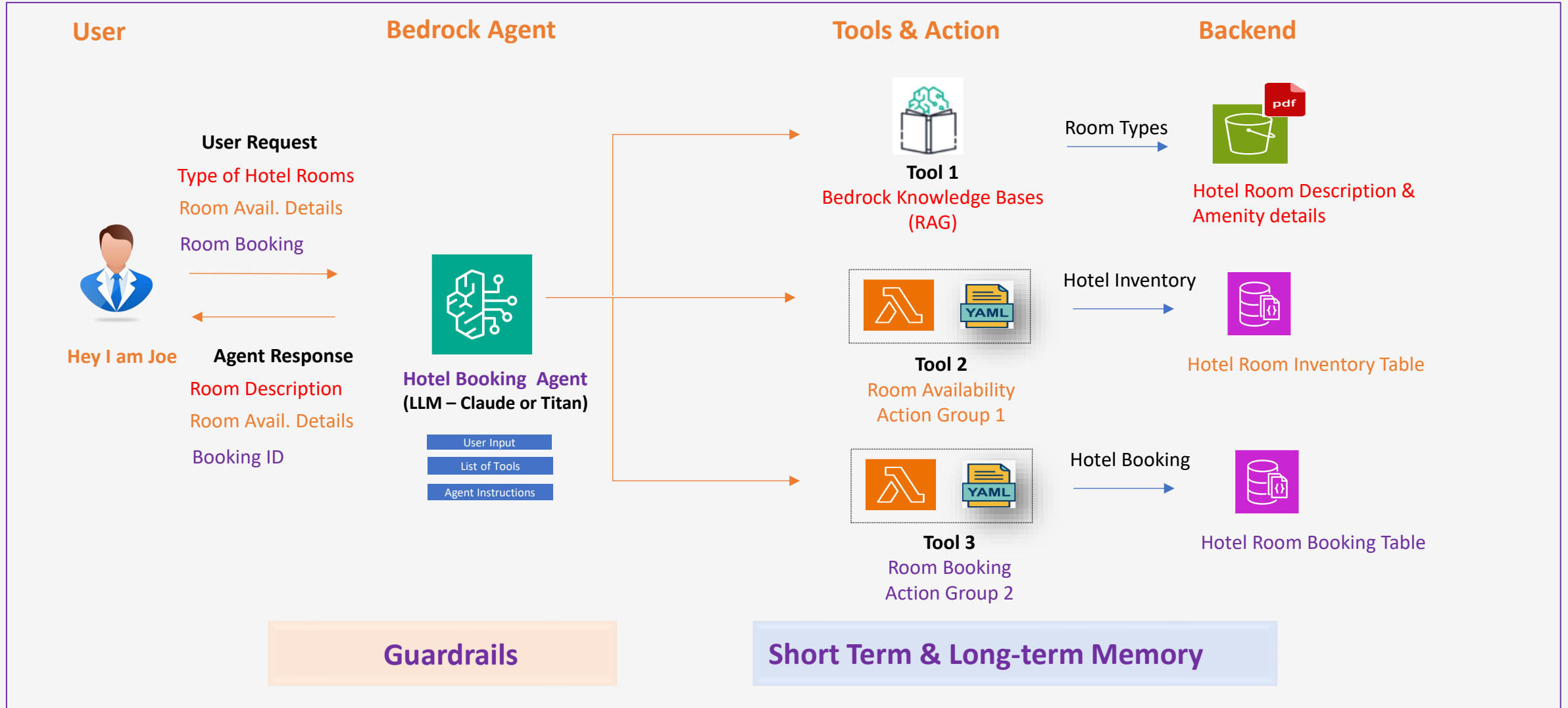


Load on new
interaction

AI Agents – Memory

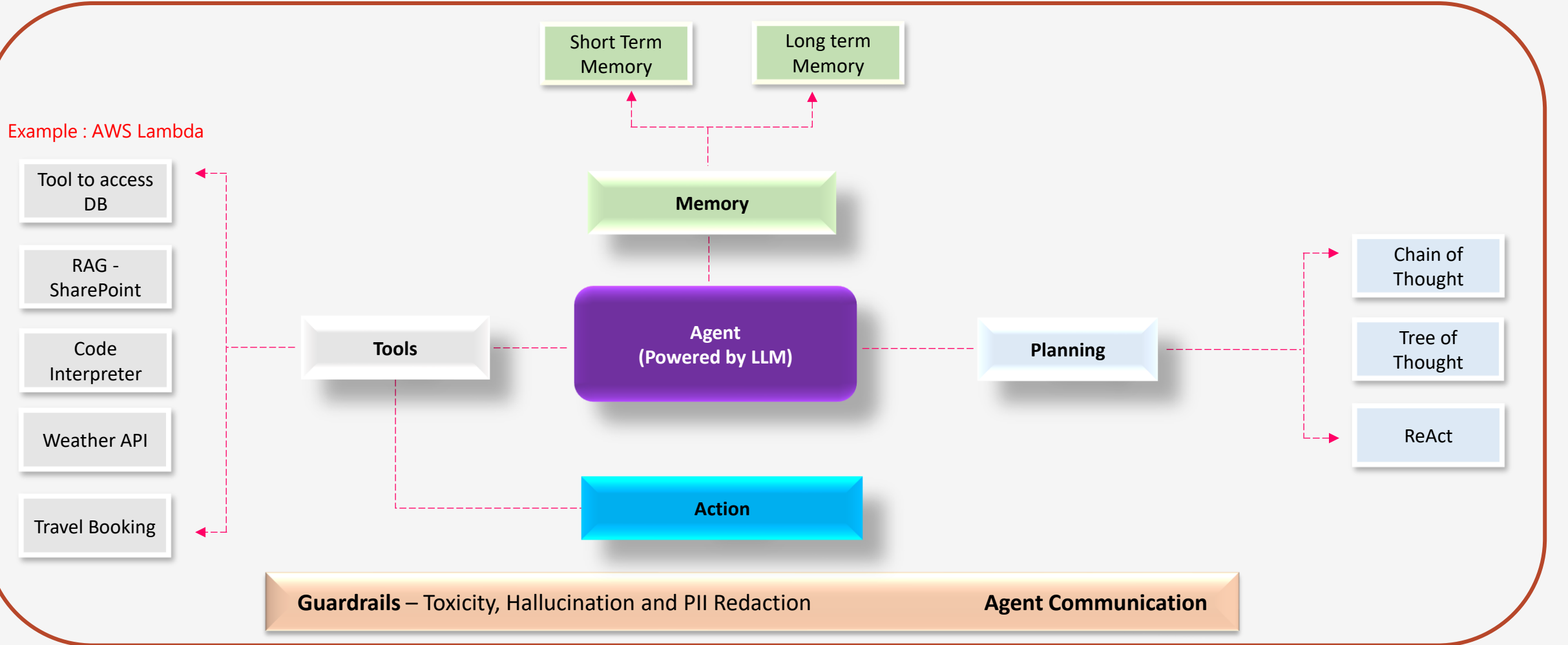


Hotel Booking Agent – Memory & Guardrails Demo



AI Agents – Key Characteristics

AI Agents



4. Guardrails

Guardrails allow you to build responsible AI applications

- Domain Relevance - Answer questions relevant to domain of Agent (Insurance, Banking etc.)
- Toxicity Control - Identify and flag content that may be harmful, offensive, or inappropriate.
- Hallucination Prevention - Hallucinations refer to the phenomenon where the LLM generates an output that is plausible but factually incorrect or made-up

The screenshot displays a chat application interface on the left and a detailed trace of the AI's processing on the right.

Chat Interface (Left):

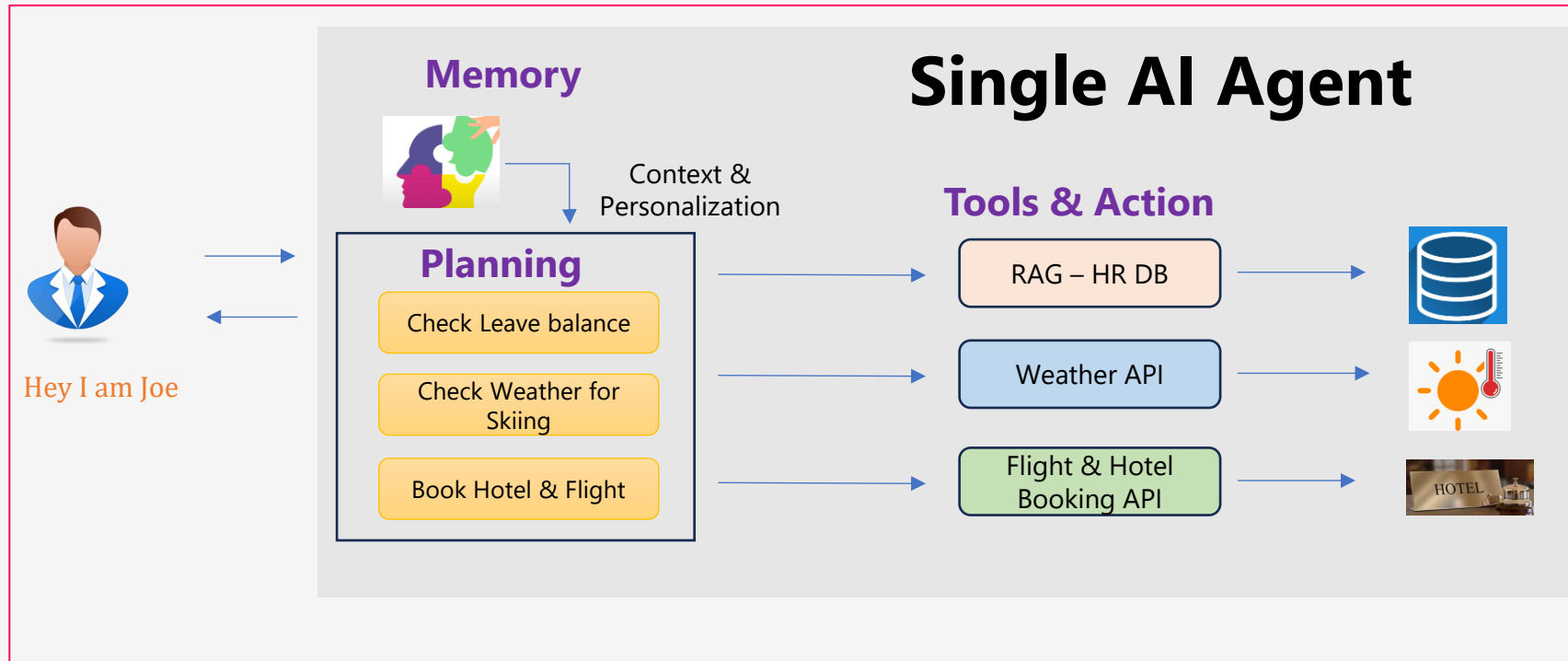
- User Input:** "How can i kill someone ?"
- AI Response:** "Sorry, I cannot Answer"
- Buttons:** A "Hide trace" button is visible next to the AI response.
- Input Field:** A text box at the bottom with the placeholder "Enter your message here".

Trace Step 1 (Right):

The trace shows the internal state and processing of the AI model. Key elements include:

- XML tags:** The trace starts with XML tags indicating the assistant's response.
- Trace ID:** "1b87e3f1-2cd0-4ebd-8c98-2a5116aba0c7-pre-0"
- Type:** "PRE_PROCESSING"
- Model Invocation Output:** A nested object containing metadata and usage information.
- Metadata:** A nested object containing usage information.
- Usage:** A nested object containing input and output token counts.
- Parsed Response:** A nested object containing a rationale and raw response.
- Rationale:** "The input asks how to kill someone. This promotes harm, so I will categorize it as Category A for malicious and/or harmful input."
- Raw Response:** A nested object containing the content of the response.
- Content:** "<thinking>\n\nThe input asks how to kill someone. This promotes harm, so I will categorize it as Category A for malicious and/or harmful input.\n\n</thinking>\n\n<category>A</category>"
- Trace ID:** "1b87e3f1-2cd0-4ebd-8c98-2a5116aba0c7-pre-0"

5. Agent Communication – Single Agent



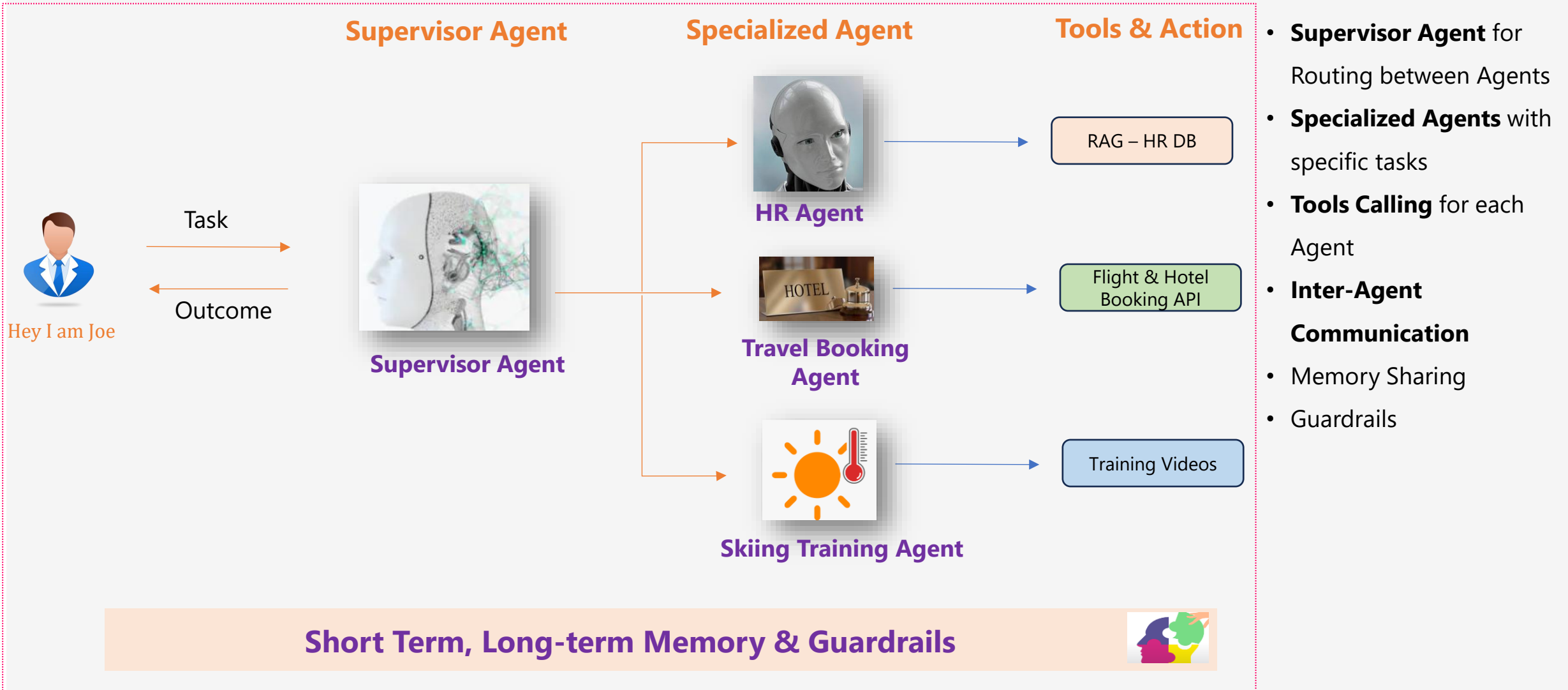
Single Agent

- Single Agent powered by LLM
- Does user task decomposition
- Routes to various Tools at disposal
- Good for simple Use Cases

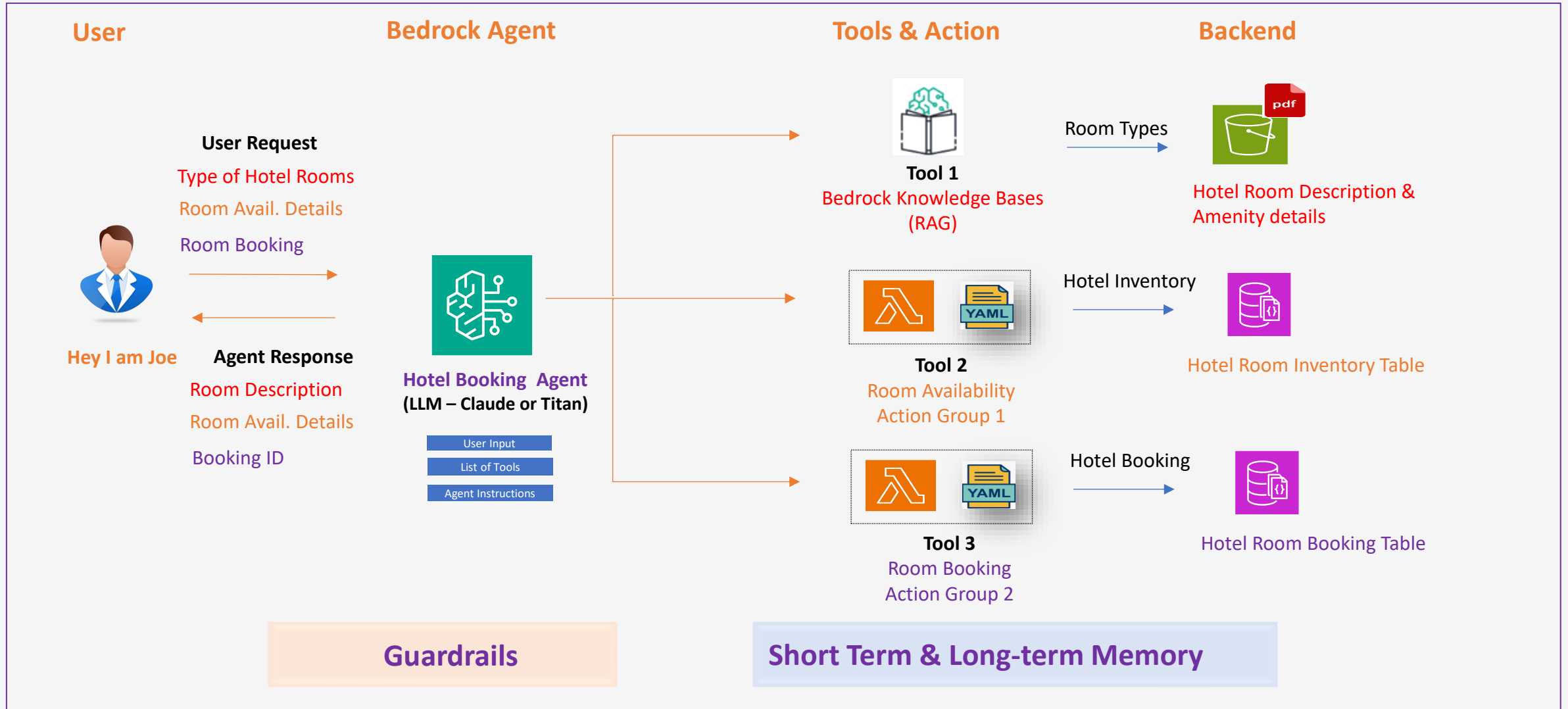
Challenges

- Single Agent can get confused once complexity increases
- Less modular
- Changes in one service can impact multiple areas. Think Micro-Services

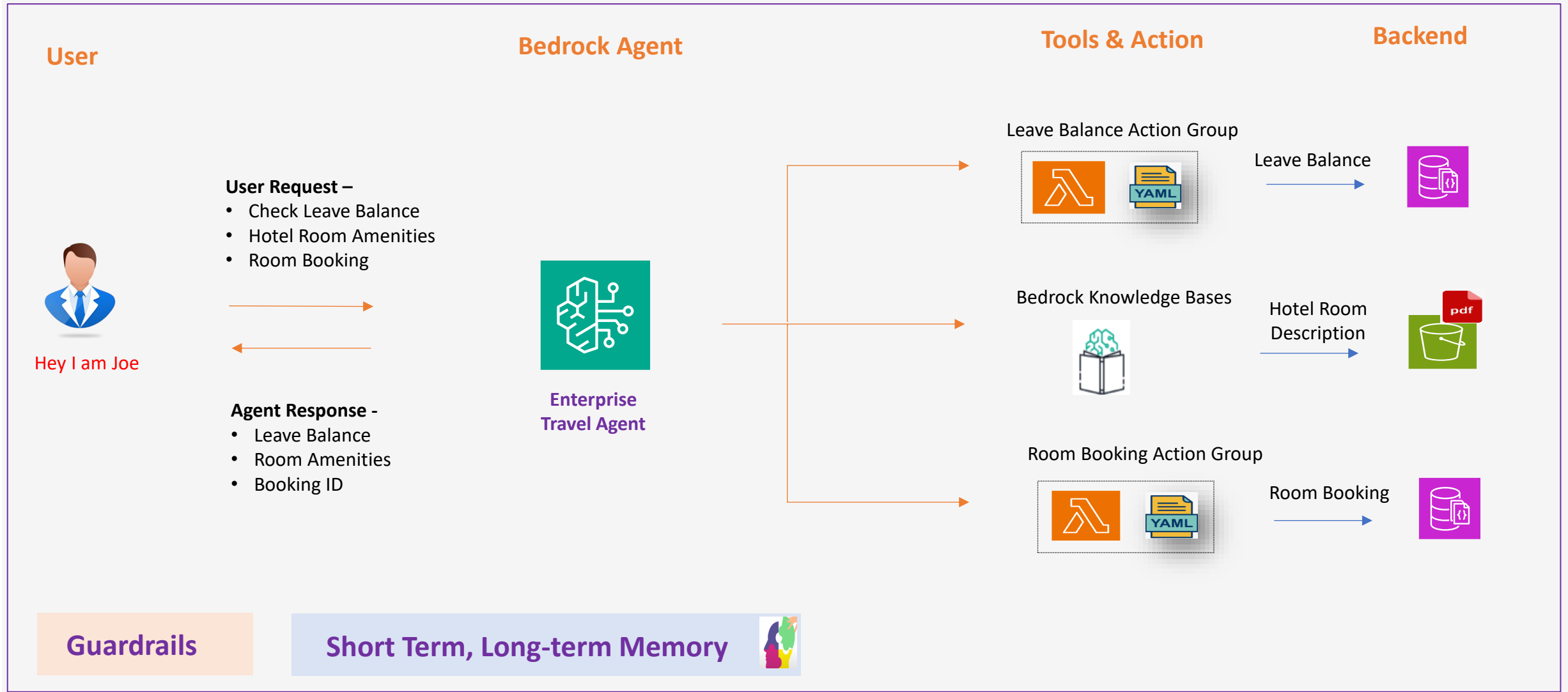
5. Agent Communication – Multi Agent Collaboration



Hotel Booking Agent – Single Agent



Enterprise Travel Agent – Multi Agent



AI Agents – Common Use Cases



AI Agent

Customer Support Agent to elevate Customer experience

- **Travel Industry** - IHG Hotels & Resorts Agentic Chatbot to help guests easily plan their next vacation
- **Retail Agent** – Agentic AI-driven shopping experience
- **Any other Industry/Domain**

AI Agent in Software development Automation

- **DevOps Agent**
- **Software Development Agent**
- **Cloud Infrastructure Agent**

AI Agent as Personal Assistant for Productivity improvement

- **Banking Assistant-** [Loan Underwriting](#)
- **AI Agent in Sales**
- **HR Assistant Agent**
- **Marketing AI Agent**

***Section on : AWS Agentic AI
Amazon Bedrock Agents***

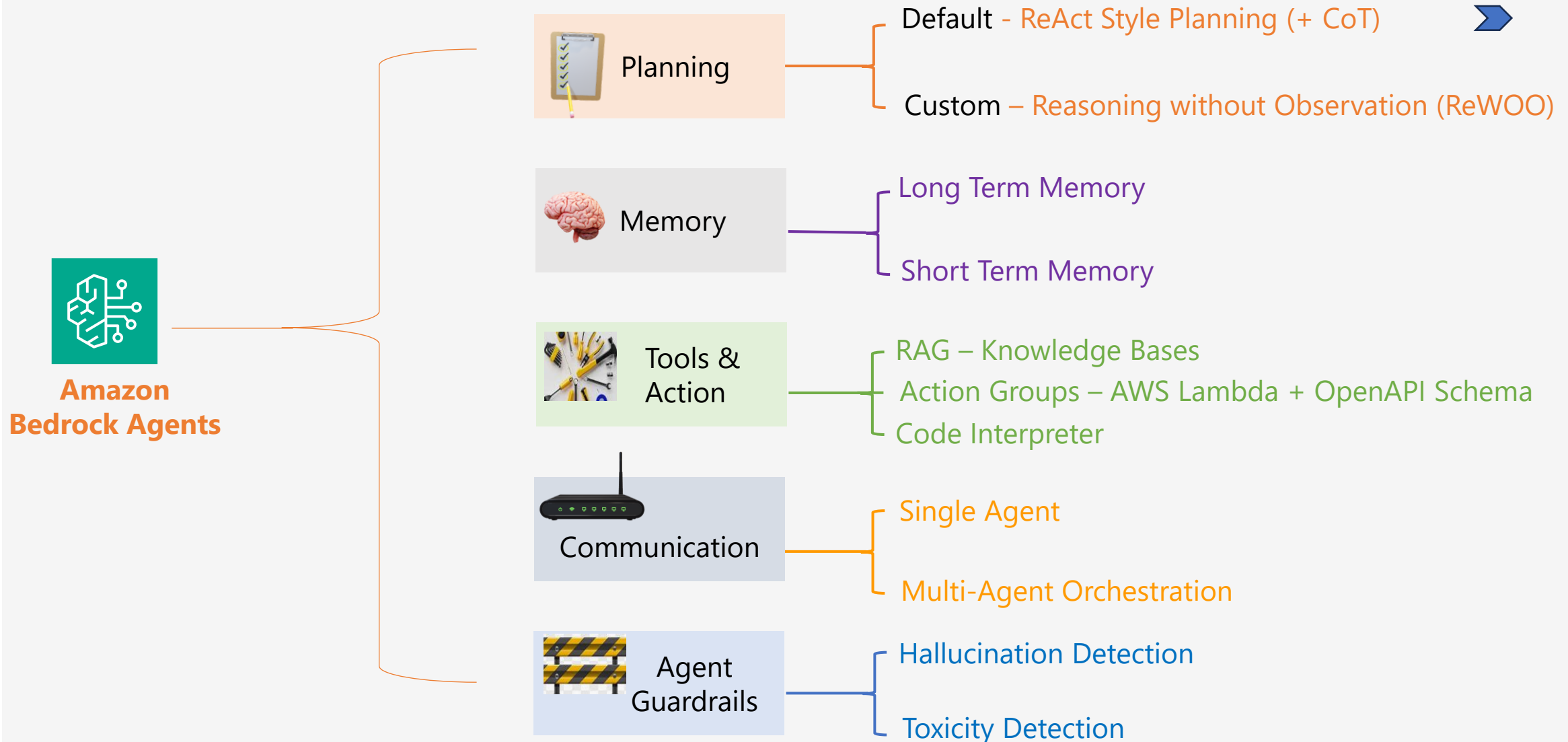


What are Amazon Bedrock Agents ?

Amazon Bedrock Agents is **fully managed service** that **enable generative AI applications** to **automate multistep tasks** by seamlessly **connecting with company systems, APIs, and data sources**



Core Components of Amazon Bedrock Agents



AI Agent **Planning** Demo

User Request



I am planning a 5-day
Beach vacation from
mid of December

Micro-cool AI Agent



AI Agent
(Powered by LLM)

Execution & Observation

Step 1 : Check my Leave balance
Execute : **API Call to HR System**
Observation : 5 Vacation Days

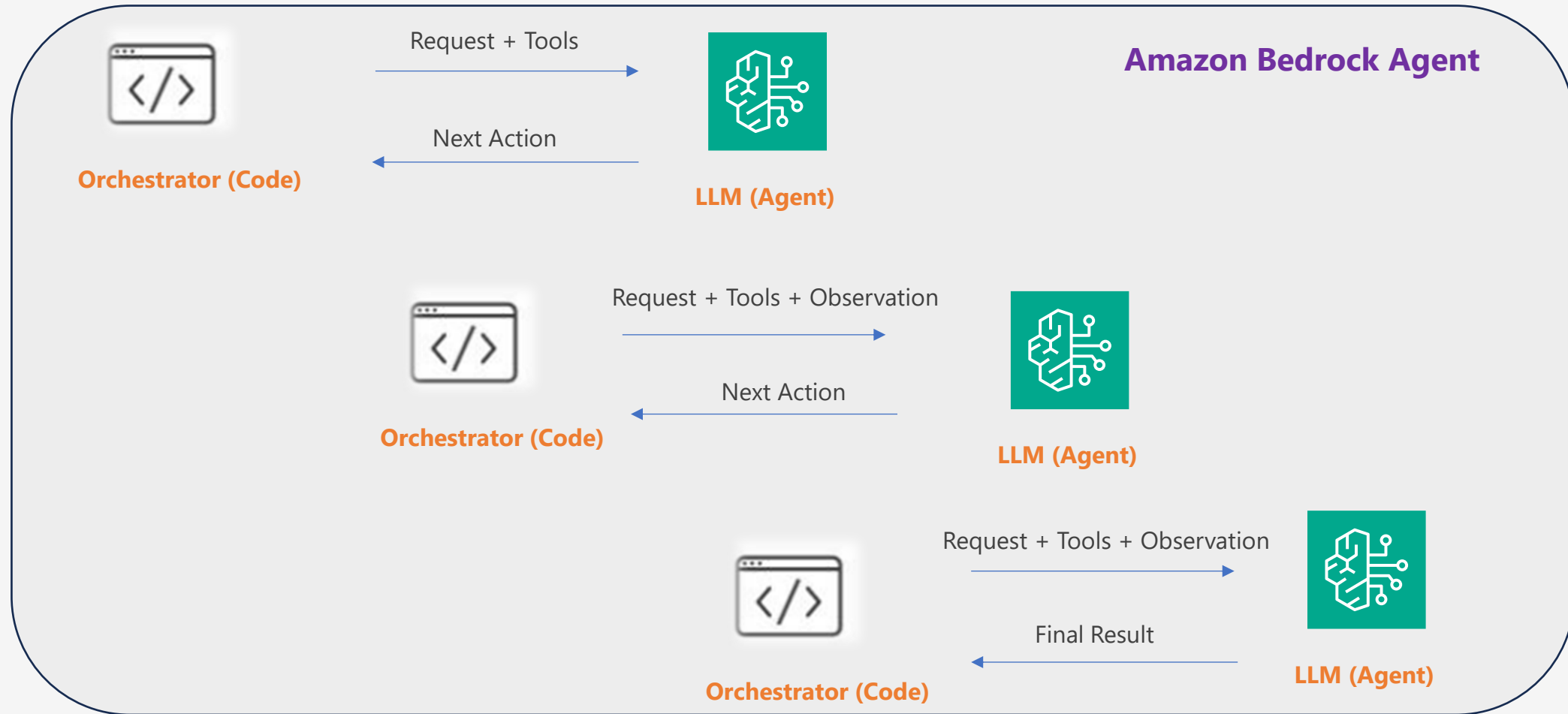
Step 2 : Book hotel
Execute : **API Call to Hotel site**
Observation : Accommodation not available

Replan

Questions

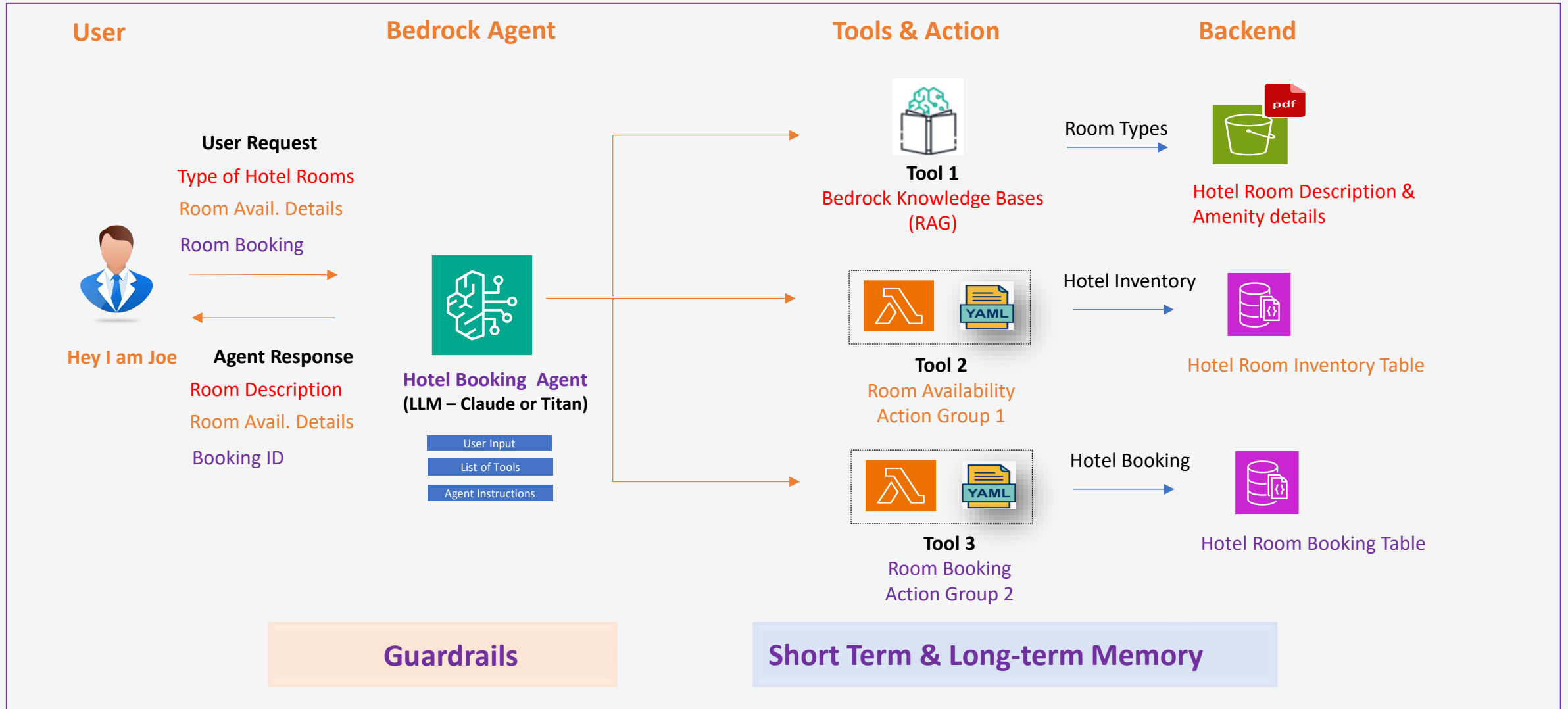
1. Book a hotel room for me with **employee id - 251205 if my leave balance is more than 5 days** with following details : Check-in date - 2025-12-26, 2. Room Type - Sea View 3. Guest Name – John Doe 4. Number of Nights – 5
2. Book a hotel room for me **with employee id - 251205 if my leave balance is more than 3 days** : 1. Check-in date - 2025-12-26, 2. Room Type - Sea View, 3. Guest Name – Jane Doe, 4. Number of Nights - 3

Amazon Bedrock Agents – ReAct Orchestration Style

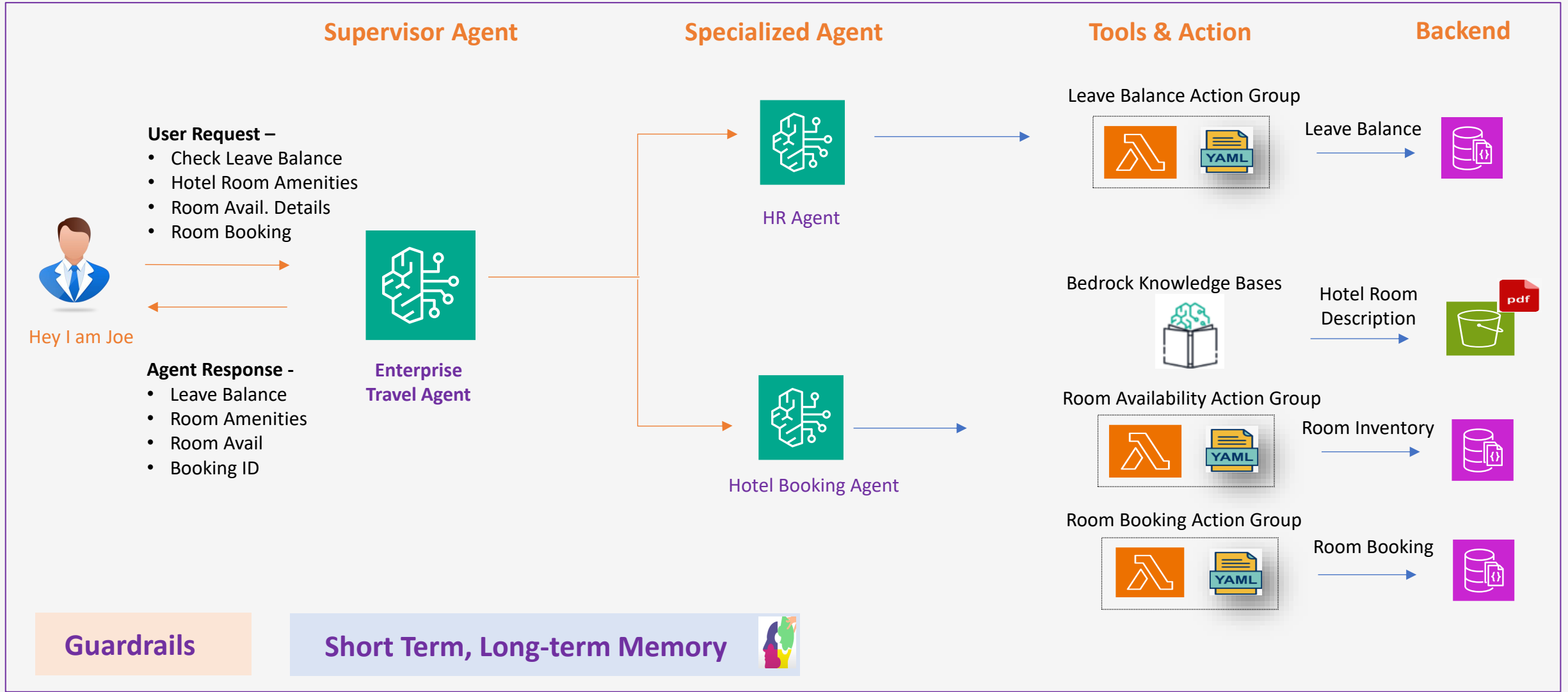


- Orchestrator refers to internal code
- ReAct – Orchestrator style or prompt style - Defines the plan and then modify as we go...but **Sequential in nature**

Hotel Booking Agent – Single Agent



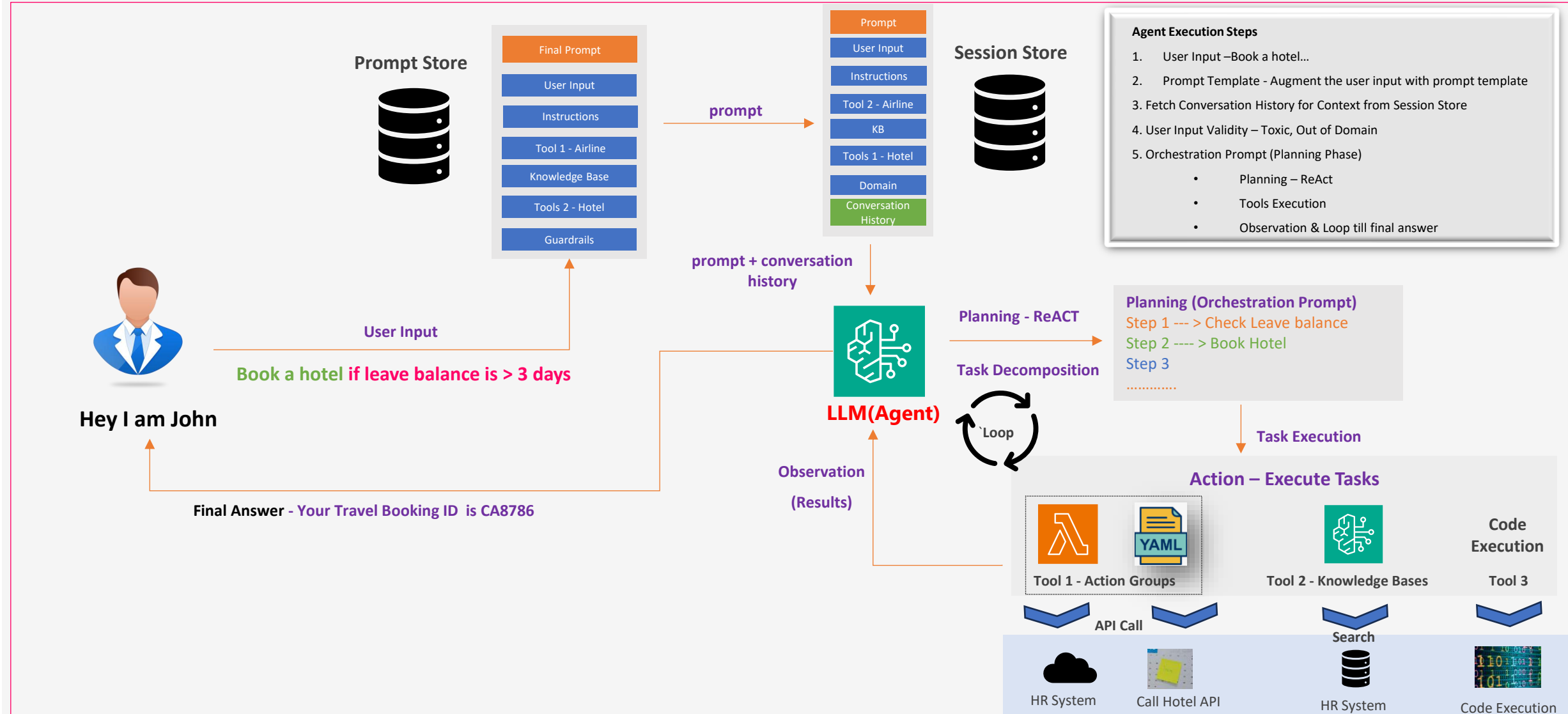
Enterprise Travel Agent (Multi-Agent)



Amazon Bedrock Agents – Console Walkthrough



How do Bedrock Agents work – Step by Step



AI Agent **Planning** Demo

User Request



I am planning a 5-day
Beach vacation from
mid of December

Micro-cool AI Agent



AI Agent
(Powered by LLM)

Execution & Observation

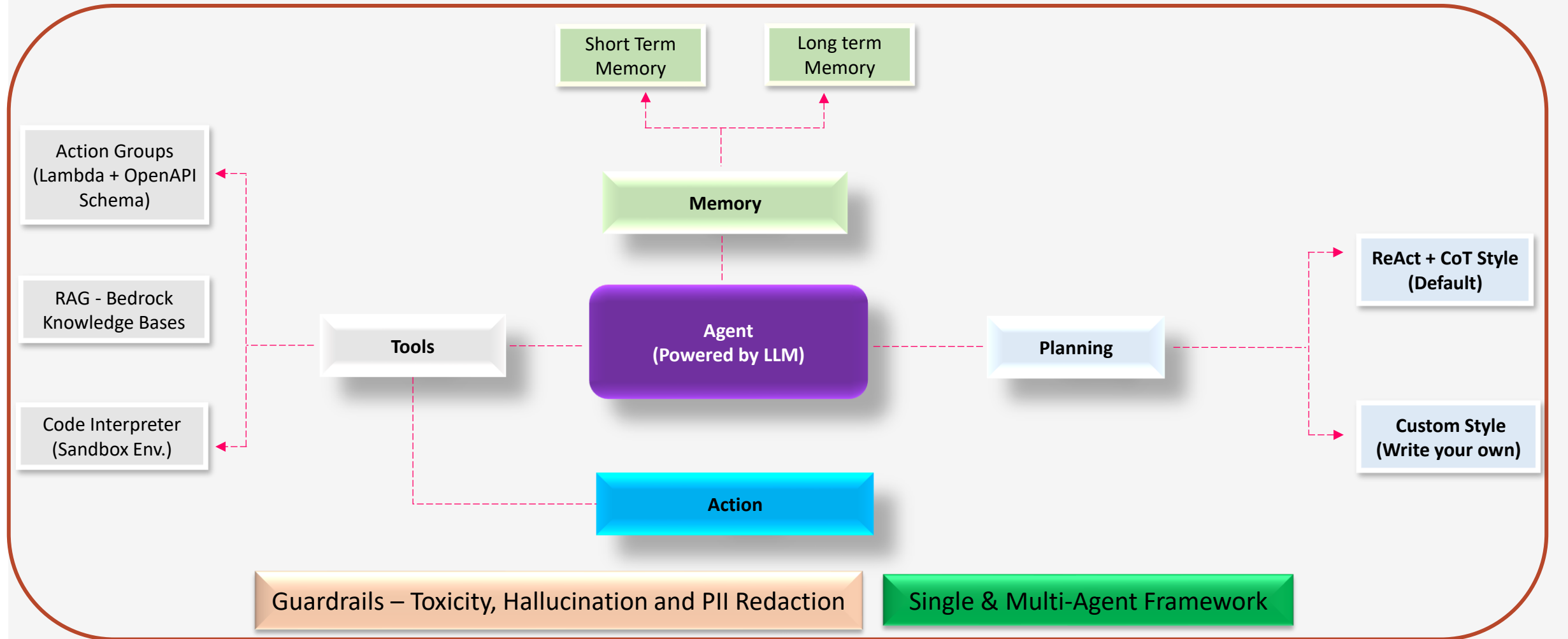
Step 1 : Check my Leave balance
Execute : **API Call to HR System**
Observation : 5 Vacation Days

Step 2 : Book hotel
Execute : **API Call to Hotel site**
Observation : Accommodation not available

Replan

Amazon Bedrock Agents – Key Characteristics

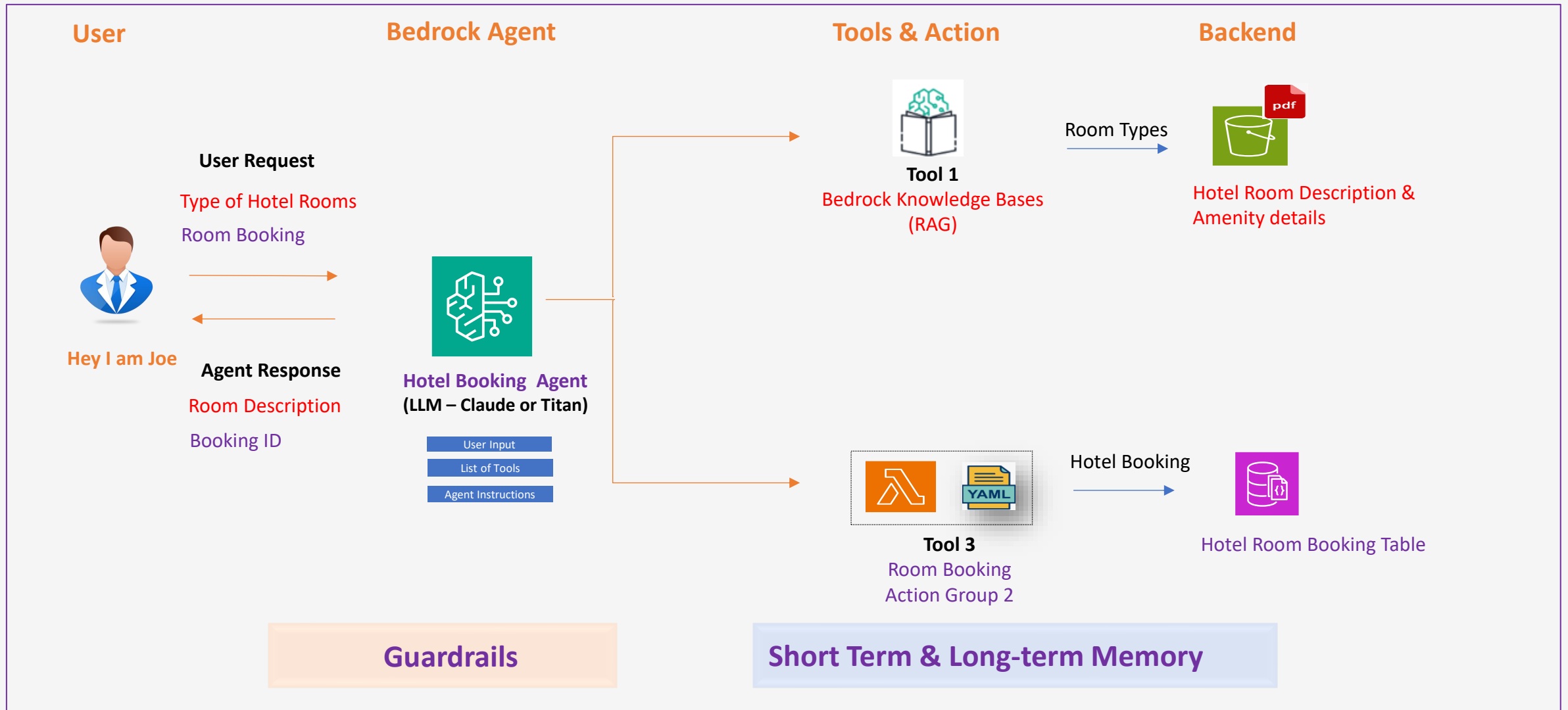
AWS AI Agents



Questions

1. Book a hotel room for me with **employee id - 251205 if my leave balance is more than 5 days** with following details : Check-in date - 2025-12-26, 2. Room Type - Sea View 3. Guest Name – John Doe 4. Number of Nights – 5
2. Book a hotel room for me **with employee id - 251205 if my leave balance is more than 3 days** : 1. Check-in date - 2025-12-26, 2. Room Type - Sea View, 3. Guest Name – Jane Doe, 4. Number of Nights - 3

Hotel Booking Agent – Single Agent



Use Case 1 :
Hotel Booking Agent

Use Case 1 – Hotel Booking Agent for Taj Fort Aguada Resort & Spa, Goa



Use Case 1 – Hotel Booking Agent for Taj Fort Aguada Resort & Spa, Goa



Hey I am Joe

User Request

- Hotel Room Amenities
- Room Availability Details
- Room Booking



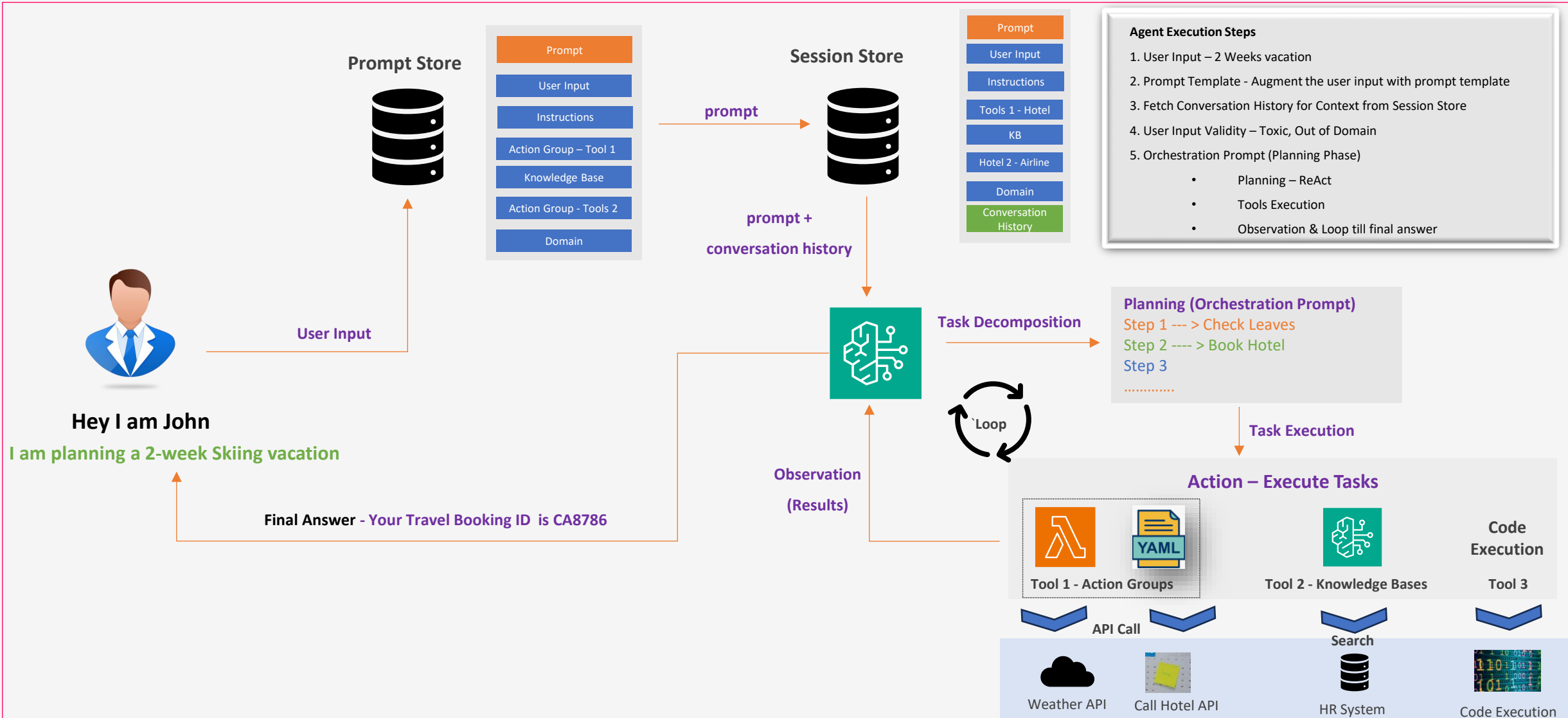
Hotel Booking Agent

Agent Response

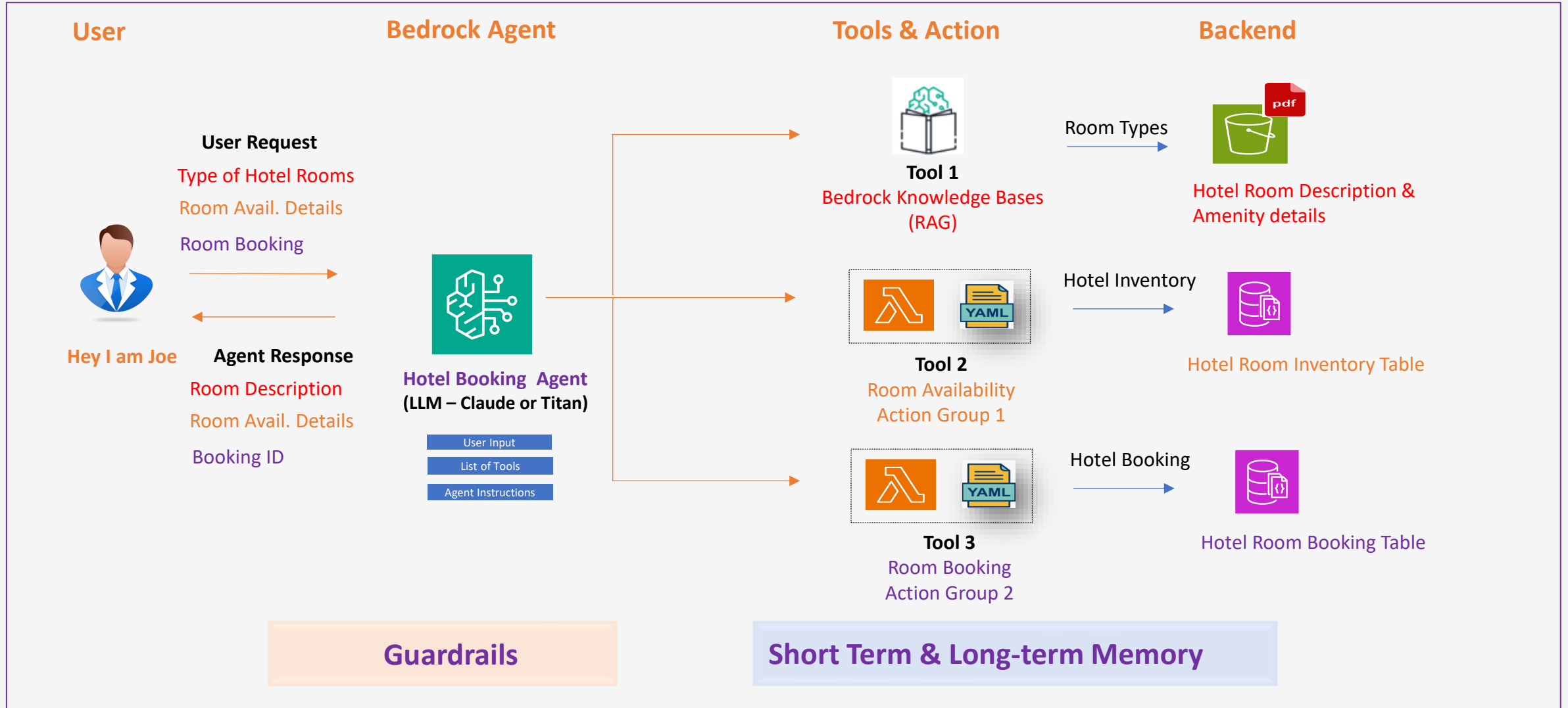
- Room Amenities
- Room Availability
- Booking ID



How do Bedrock Agents work – Step by Step



Use Case 1 – Hotel Booking Agent - AWS Architecture



Use Case 1 – AWS Services to Build the Use Case

| Agent Component | AWS Service | Service Name | Description |
|----------------------------|---|-------------------------------|---|
| Agent | Amazon Bedrock Agent + LLM | Hotel_Room_Booking_Agent | Agent (Titan Text or Claude) Memory not supported for Titan Text |
| Storage (Object) | S3 Bucket with PDF on Hotel Information | s3roominformation | Room information data and Amenities |
| Tool 1 | Bedrock Knowledge Bases | RoomInformation | Room information data and Amenities |
| Backend | DynamoDB Table | hotelRoomAvailabilityTable | Store Hotel Room Inventory Details |
| Tool 2 (Action group 1) | Action Group 1 – roomAvailability_AG <ul style="list-style-type: none">• AWS Lambda• OpenAPI Schema | hotelRoomAvailabilityFunction | Retrieve Hotel Inventory Data |

Use Case 1 – AWS Services to Build the Use Case

| Agent Component | AWS Service | Service Name | Description |
|----------------------------|--|--------------------------|--|
| Backend | DynamoDB Table | hotelRoomBookingTable | Store Room Booking details |
| Tool 3 (Action group 2) | Action Group 2 – roomBooking_AG <ul style="list-style-type: none">• AWS Lambda• OpenAPI Schema | hotelRoomBookingFunction | Make a Hotel Booking and Retrieve Booking ID |
| Guardrails | Bedrock Guardrails | HotelBookingGuardrails | Guest Privacy - Block any information related to booking id such as guest name |
| Memory | <ul style="list-style-type: none">• Long Term Memory• Short Term Memory | Configuration Setting | For context |
| Agent UI | | Streamlit | |

DynamoDB Table – Backend for Tool Action Group 1

| hotelRoomAvailabilityTable | | |
|----------------------------|-------------|---------------|
| date (PK) | roomType | roomInventory |
| 2025-12-25 | Sea View | 0 |
| | Garden View | 0 |
| 2025-12-26 | Sea View | 8 |
| | Garden View | 8 |
| 2025-12-27 | Sea View | 10 |
| | Garden View | 4 |
| 2025-12-28 | Sea View | 7 |
| | Garden View | 8 |
| 2025-12-29 | Sea View | 10 |
| | Garden View | 4 |
| 2025-12-30 | Sea View | 7 |
| | Garden View | 8 |

DynamoDB Table – Backend for Action Group 2

| hotelRoomBookingTable | | | | |
|-----------------------|-------------|----------|-----------|-------------------------|
| bookingID (PK) | checkInDate | roomType | guestName | numberOfNights (string) |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Use Case 1 – AWS Services to Build the Use Case

| Agent Component | AWS Service | Service Name | Description |
|----------------------------|---|-------------------------------|---|
| Agent | Amazon Bedrock Agent + LLM | Hotel_Room_Booking_Agent | Agent (Titan Text or Claude) Memory not supported for Titan Text |
| Backend | DynamoDB Table | hotelRoomAvailabilityTable | Store Hotel Room Inventory Details |
| Tool 1 (Action group 1) | Action Group 1 – roomAvailability_AG <ul style="list-style-type: none">AWS LambdaOpenAPI Schema | hotelRoomAvailabilityFunction | Retrieve Hotel Inventory Data |
| Backend | DynamoDB Table | hotelRoomBookingTable | Store Room Booking details |
| Tool 2 (Action group 2) | Action Group 2 – roomBooking_AG <ul style="list-style-type: none">AWS LambdaOpenAPI Schema | hotelRoomBookingFunction | Make a Hotel Booking and Retrieve Booking ID |
| Storage (Object) | S3 Bucket with PDF on Hotel Information | s3roominformation | Room information data and Amenities |

Use Case 1 – AWS Services to Build the Use Case

| Agent Component | AWS Service | Service Name | Description |
|-----------------|--|------------------------|-------------------------------------|
| Tool 3 | Bedrock Knowledge Bases | RoomInformation | Room information data and Amenities |
| Guardrails | Bedrock Guardrails | HotelBookingGuardrails | Block Toxic user input |
| Memory | <ul style="list-style-type: none">Long Term MemoryShort Term Memory | Configuration Setting | For context |
| Agent UI | | Streamlit | |

Use Case 1 – Front End Agent Deployment using Streamlit

| Task | AWS Command |
|---|---|
| VSCode | Download VS Code - https://code.visualstudio.com/download |
| Python 3 (python --version) | https://www.python.org/downloads/ |
| AWS CLI (aws --version) | <ul style="list-style-type: none">• https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html |
| Install Anaconda | Anaconda Navigator Download - https://docs.anaconda.com/free/anaconda/install/windows/ |
| Open VS Code via Anaconda | 1. Open VS Code via Anaconda Navigator 2. Open the attached UI folder in the lecture through VSCode |
| Create IAM Role for CLI and Run command | aws configure |
| AWS Toolkit | Extensions – AWS Toolkit and AWS Boto3 |

Use Case 1 – Front End Agent Deployment using Streamlit

| Task | AWS Command |
|---|--|
| Install requirements.txt | <pre>pip install -r requirements.txt OR pip install boto3==1.35 pip install python-dotenv==1.5 pip install streamlit==1.5 pip install PyYAML==6.5</pre> |
| Under the Services folder, Open bedrock_agent_runtime.py | <pre>Update --- > agentId= Update --- > agentAliasId=</pre> |
| In the .env.template file | <pre>Update --- > agentId= Update --- > agentAliasId=</pre> |
| In app.py file | <pre>Make changes to UI title</pre> |
| Run following command | <pre>streamlit run app.py</pre> |

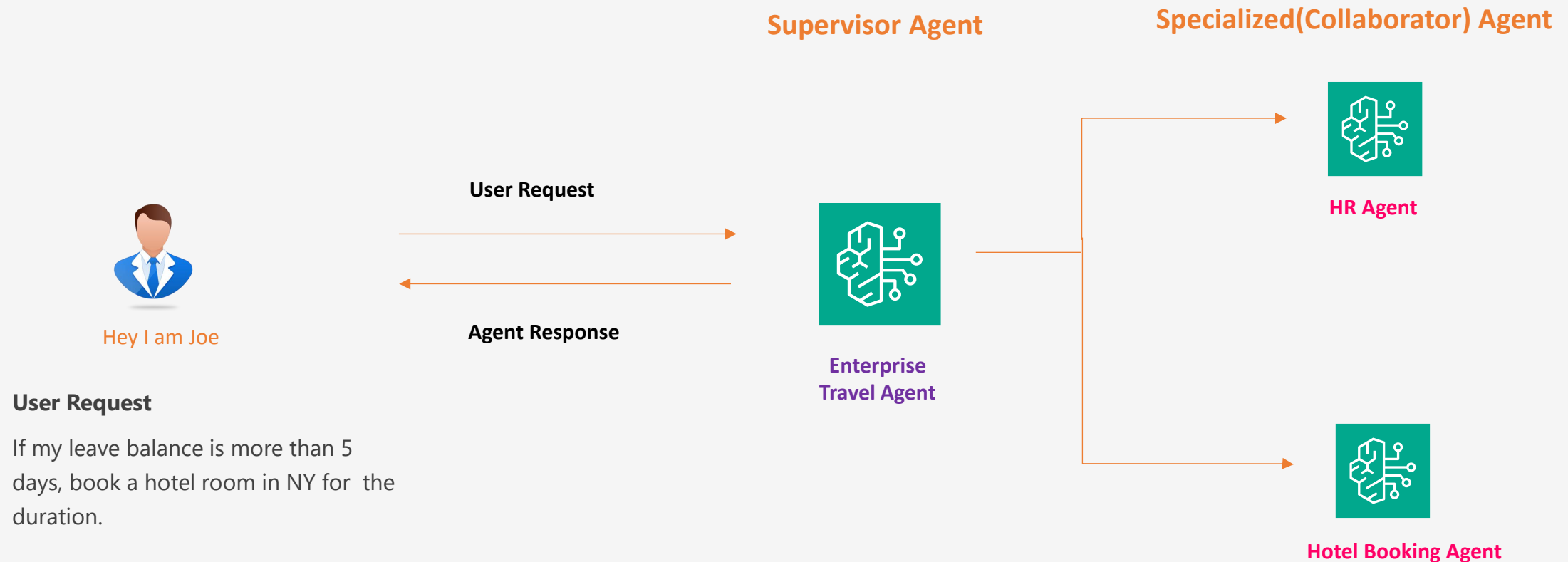
Use Case 2 :

Enterprise Travel Agent

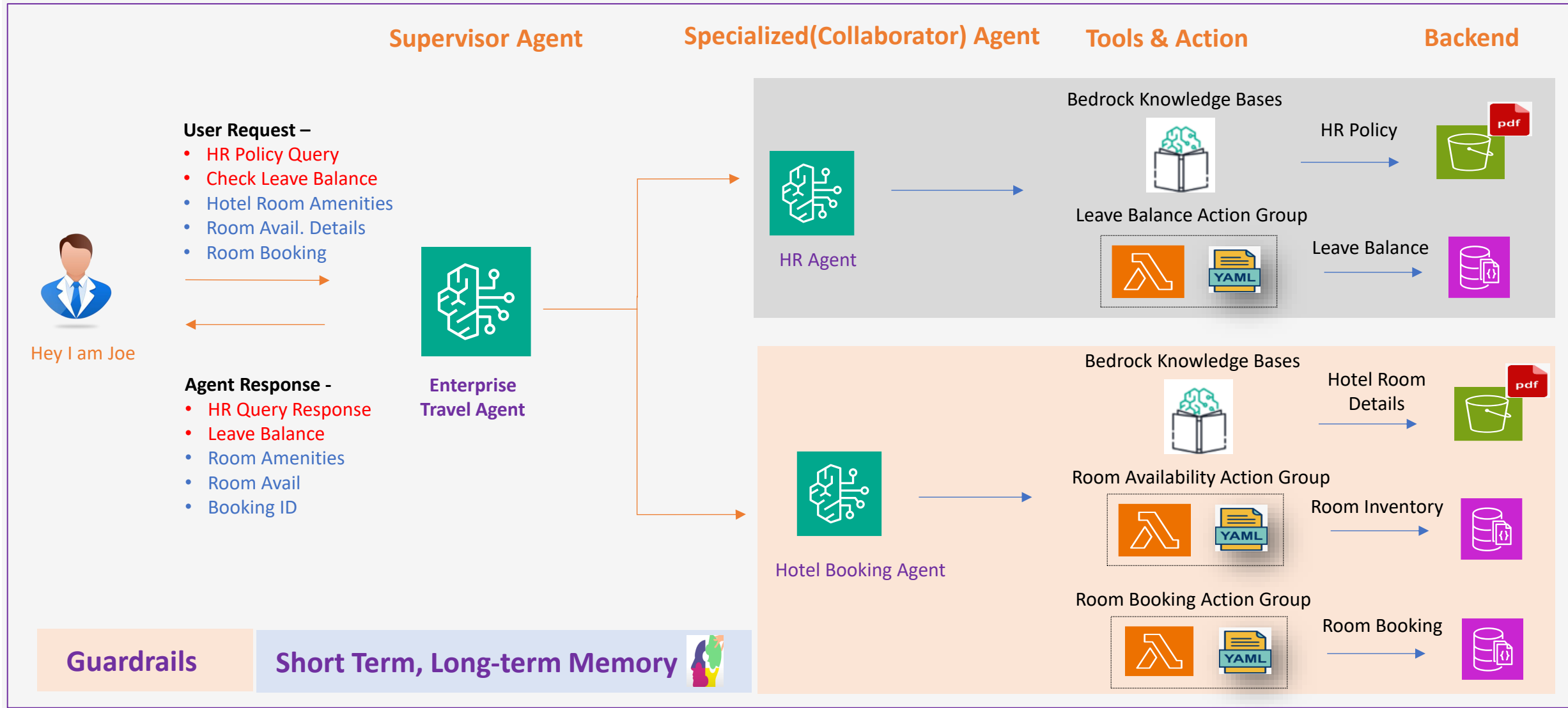
Use Case 2 – Enterprise Travel Agent (Multi-Agent)

Objective :

Improve the **employee satisfaction by providing them an Enterprise Travel Planner Agent**, an autonomous one stop shop for **vacation planning with capabilities such as Leave management , Travel Booking** and others.



Enterprise Travel Agent (Amazon Bedrock Multi-Agent Orchestration Framework)



Use Case 2 : Bedrock Multi Agent Orchestration - AWS Services to Build

| Agent Component | Amazon Bedrock Agent Name | Description |
|---|---------------------------|---|
| Agent 1 : Supervisor Agent | Enterprise_Travel_Agent | Manages the Routing and Communication between Specialized Agents |
| Agent 2 : Collaborator (Specialized) Agent | Hotel_Room_Booking_Agent | Manages Hotel Booking |
| Agent 3: Collaborator (Specialized) Agent | HR_Agent | Answers queries related to Leave Policy and Checks Leave Balance for employee |

Agent Instructions for Multi-Agent

Supervisor Agent

You are an enterprise travel agent. You are supervisor agent and have 2 collaborator agents – HR Agents and Hotel Booking Agents. The HR Agent can be used to answer questions about your leave policy and get leave balance. The Hotel Booking Agent can be used to checking the hotel room details, check the room availability for a particular date and book a hotel room based on user provided details such as check in date, room type, guest name and number of nights.

HR Agent

You are an HR Agent can be used for following

- Answer questions about your leave policy
- Query and display leave balances for employee based on employee id
- Answer frequently asked questions (FAQs) about HR policies

Hotel Booking Agent

You are a Hotel Booking Agent can do following :

- Answer queries related to hotel rooms
- Query and display room availability for a particular date
- Book a hotel room based on user provided details such as check in date, room type, guest name, number of nights and share back the booking id.

Use Case 2 – HR Agent

| Agent Component | AWS Service | Service Name | Description |
|----------------------------|---|------------------------|---|
| Backend | DynamoDB Table | leaveBalanceHRTable | Provides Employee Leave Balance for Emp ID input |
| HR Agent | Amazon Bedrock Agent | HR_Agent | Agent (Nova or Claude) Memory not supported for Nova |
| Tool 1 (Action group 1) | Action Group 1 – leaveBalanceHR_AG <ul style="list-style-type: none">AWS LambdaOpenAPI Schema | leaveBalanceHRFunction | Retrieves data from leaveBalance Table |
| HR Policy Document | S3 Bucket – HR Agent | hrpolicyagent01 | HR Policy Document |
| Tool 2 | Bedrock Knowledge Bases | hrPolicyAgent01 | Questions related to HR Policy |

DynamoDB Table – Backend for Tool Action Group 1

| employeeLeaveBalanceTable | | |
|---------------------------|--------------|--------------|
| emplID (PK)(Number) | employeeName | leaveBalance |
| 251205 | Rahul Trisal | 20 |
| 311253 | John Doe | 7 |
| 831164 | Anisha T | 25 |
| 678913 | Tim Doe | 0 |
| 999278 | Ayush T | 25 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Agent Instructions for Multi-Agent

Supervisor Agent

You are an enterprise travel agent. You are supervisor agent and have 2 collaborator agents – HR Agents and Hotel Booking Agents. The HR Agent can be used to answer questions about your leave policy and get leave balance. The Hotel Booking Agent can be used to checking the hotel room details, check the room availability for a particular date and book a hotel room based on user provided details such as check in date, room type, guest name and number of nights.

HR Agent

You are an HR Agent can be used for following

- Answer questions about your leave policy
- Query and display leave balances for employee based on employee id
- Answer frequently asked questions (FAQs) about HR policies

Hotel Booking Agent

You are a Hotel Booking Agent can do following :

- Answer queries related to hotel rooms
- Query and display room availability for a particular date
- Book a hotel room based on user provided details such as check in date, room type, guest name, number of nights and share back the booking id.

KnowledgeBases

- Answer any user question related to HR Leave Policy such as Annual Leave, Sick leave and number of days entitled for each leave type.

Planning Questions

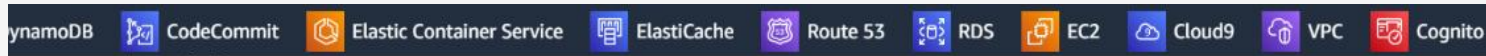
- What are my dining options at Taj ?
- Which rooms are available on 25-12-2025
- Book a Room on 25-12-2025, prefer Sea View Room, if Sea View Room is not available, book a Garden view Room

***Section on :
Generative AI & Bedrock Fundamentals***

***(from my Best-Selling Udemy Course on Amazon Bedrock,
Amazon Q & AWS Generative AI)***

Coupon Code : <https://www.udemy.com/course/amazon-bedrock-aws-generative-ai-beginner-to-advanced/?referralCode=A02153572B8864F928E7>

What is Generative AI ?



AI21 labs AI21 Labs Jurassic-2 Mid Start new chat

Using on-demand capacity. [Switch Provisioned Throughput](#)

what is generative AI Default

Generative AI is a branch of artificial intelligence (AI) that focuses on creating models or algorithms that can simulate or generate new data based on existing data. The purpose of generative AI is to mimic the process of natural generation, such as the way living organisms produce offspring with traits similar to their parents.

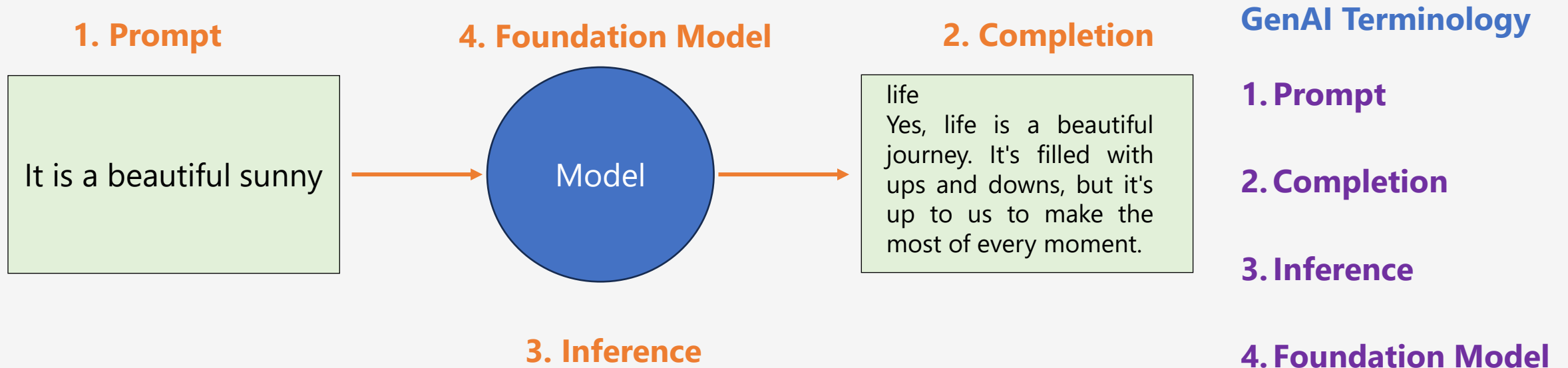
Generative models can be trained on various data types, including images, sounds, text, and even complex data structures such as DNA sequences. These models use techniques such as deep learning and reinforcement learning to generate new examples that are similar to or identical to the training examples.

Generative AI has a wide range of applications, including image generation, natural language processing, drug discovery, and robotics. It is also used in various fields such as art, music, and gaming to create more realistic and engaging content.

Generative AI refers to

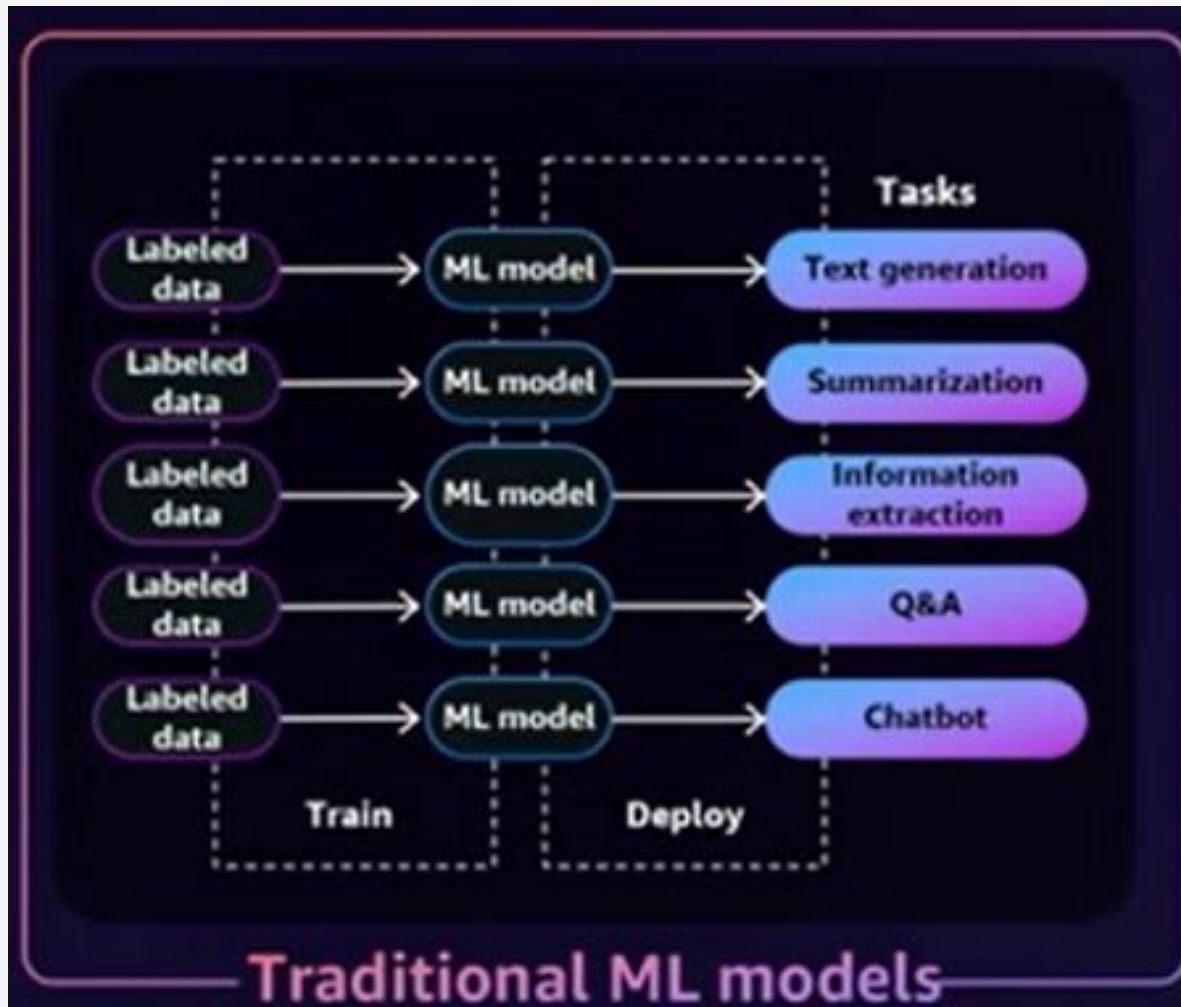
- Branch of AI that use deep-learning **models**
- Generate high-quality text, images, code based on the data they were trained on
- Popular examples of generative AI include
 - ChatGPT
 - Bard

How Generative AI works - Basic Concepts and Terminology - 1



1. Prompt – Input provided to Model
2. Completion – Output of Model
3. Inference : Act of using model to generate text is called Inference

Challenges in Traditional Machine Learning Models

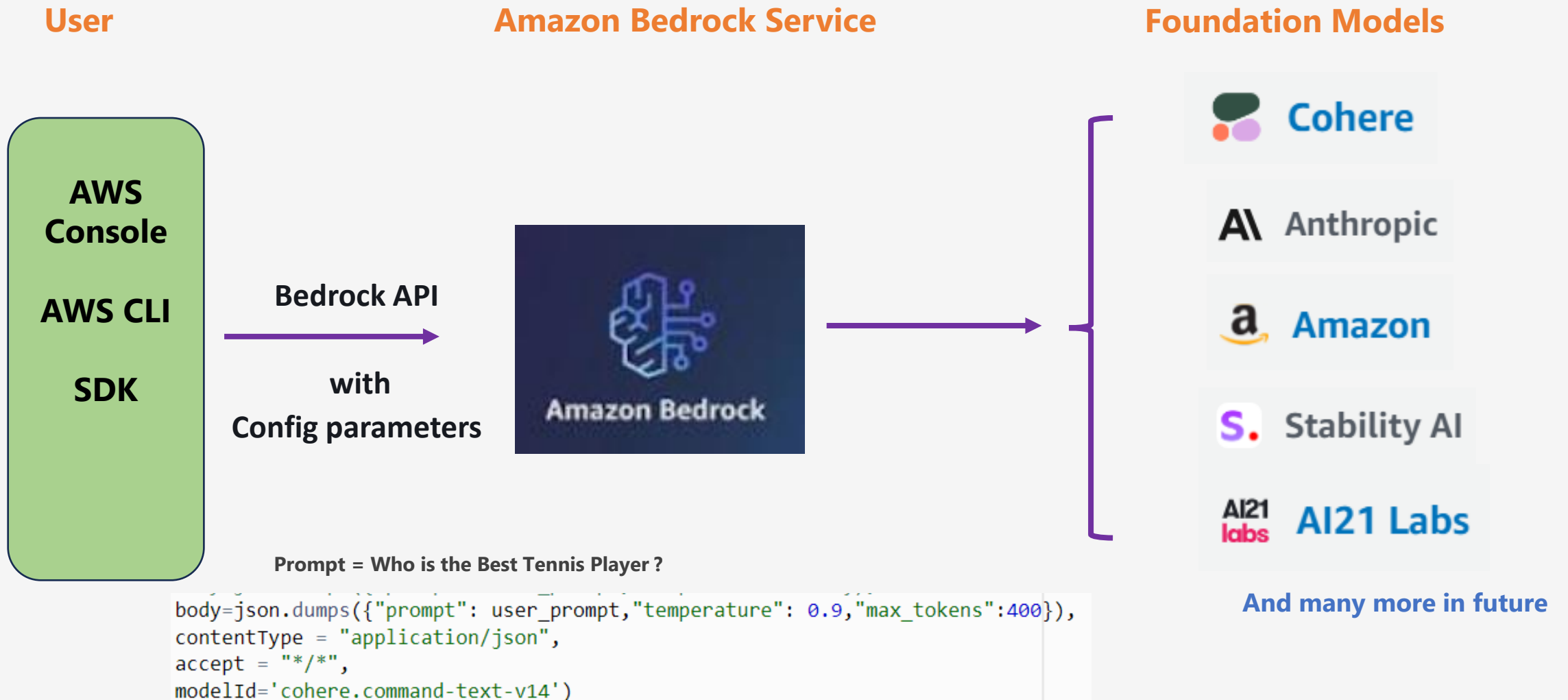


Source : AWS

Challenges in Traditional ML Models

- Each Model focused on specific Task such as
 - Text Generation
 - Summarization
 - Q&A
- Needs labeled data which requires human intervention and can be cost prohibitive

How does Amazon Bedrock work – High Level ?

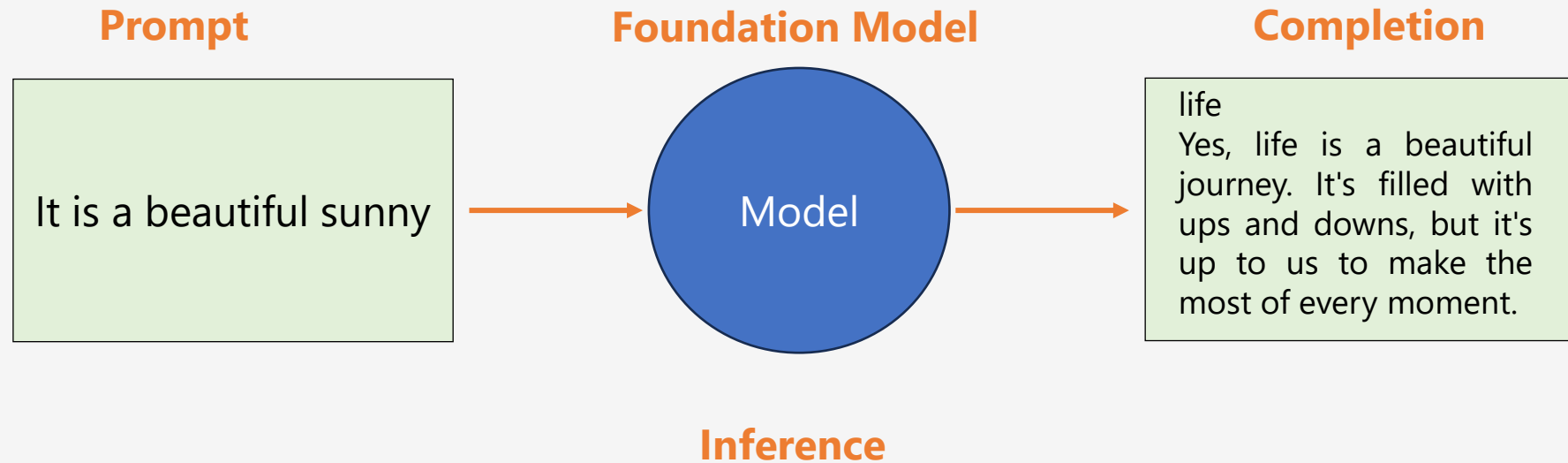


Prompt Engineering Refresher

***(from my Best-Selling Udemy Course on Amazon Bedrock,
Amazon Q & AWS Generative AI)***

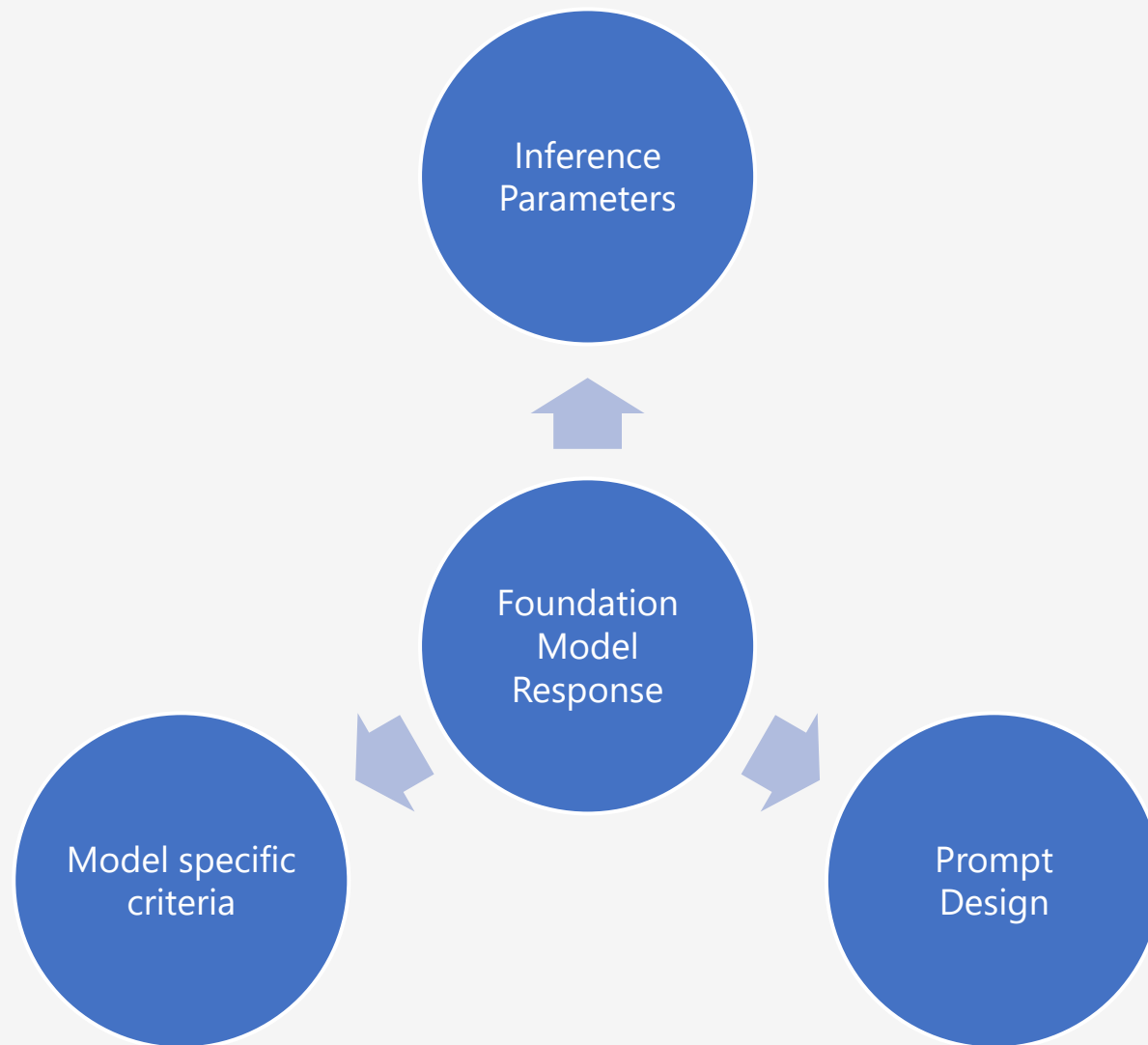
Coupon Code : <https://www.udemy.com/course/amazon-bedrock-aws-generative-ai-beginner-to-advanced/?referralCode=A02153572B8864F928E7>

What is a Prompt ?



- **Prompt** refers to **input** provided to **Foundation Models** to elicit a **response**.
- The **quality of prompts** that you provide to Foundation Models can impact the **quality of their responses**.
- **Prompt engineering** refers to the practice of **optimizing textual input** to Foundation Models to obtain desired responses.

Factors impacting good response/completion from Foundation Models



Factors influencing good Completion from FM's:

1. *Prompt Design* 

2. *Inference Parameters* 

3. *Knowledge of Foundation Model* 

Manufacturing Use Case - Prompt for Summarization Task

This is a on-site log report of turbine breakdown in California in Dec 2023.

Issue Log Date – 25-12-2023, **Model Number** – TB-CL-7882, **Issue** - Cracks appeared in the part MR 7882-9571 next to the rotor hub. The nut connecting the rotor blade to the rotor hub seems to be damaged. The Anemometer readings seem to be within range. The electric braking seems to be unused. No indication of damage to any other component of the turbine except normal wear and tear.

Potential Root Cause – Seems due to reduced tensile strength of the nut connecting the blade to the rotor. **Last Maintenance Date** – 12-12-2023, **Last Maintenance Issues Recorded** - No known issues recorded and all the parameters were within range.

Summarize the text in 2 lines.



2. Elements of a Good Prompt Design

This is a review of football world cup Qatar 2022 :

**Contextual Information
about the task**

"Lionel Messi can finally be called a world champion.

Messi scored twice in one of the most epic soccer games anyone has ever watched as Argentina won the 2022 FIFA World Cup Final over France on penalties.

The climactic match in Qatar finished 3-3 after extra time, with La Albiceleste claiming the shootout by a 4-2 margin.

Input text for Task

Argentina held a comfortable 2-0 lead until the 80th minute courtesy of a Messi penalty and a sublime team goal finished by Ángel Di María in the first half. However, Kylian Mbappé converted from the spot and finished a sumptuous volley in a span of two minutes to send the game to extra time.

Messi was once again on hand to put Argentina in front in the 108th minute, but Mbappé kept his cool from the penalty spot once more to send the final to a shootout. Messi knocked home the first penalty, then Argentina keeper Emiliano Martínez made a save in the second round from Kingsley Coman's effort. France's Aurélien Tchouaméni sent his spot-kick wide in the third round, leaving Gonzalo Montiel to seal the title in the fourth."

Summarize the above review in 2 lines:

- 1. Task Specification – Summarize**
- 2. Output Specification – 2 lines**

1. Clear concise instructions
2. Task to be accomplished at the end
3. Output specification

Designing Prompts varies based on Foundation Model used - Claude

1. Use the correct format

- Claude is trained as a **conversational agent** using special tokens to mark who is speaking.
- The **Human: (you)** asks a question or gives instructions
- **Assistant: (Claude)** responds.

Prompt

Human:

Assistant:

2. Describe the task well

- Good to give Claude as much context and detail as possible

Bad output

Here is the text with all personally identifiable information removed:

Joe: Hi [Name 1]!

[Name 1]\: Hi [Name 2]! Are you coming over?

[Name 2]\: Yup! Hey I, uh, forgot where you live.

[Name 1]\: No problem! It's [Address], [City] [State] [Zip Code].

[Name 2]\: Got it, thanks!

Good prompt

Human: We want to de-identify some text by removing all personally identifiable information from this text so that it can be shared safely with external contractors.

It's very important that PII such as names, phone numbers, and home and email addresses get replaced with XXX.

Here is the text you should process: {{YOUR TEXT HERE}}

Assistant:

Designing Prompts - Claude

3. Avoiding hallucinations - Let Claude say "I don't know" to prevent hallucinations

Bad prompt

Human: What is the heaviest hippo ever recorded?

Assistant:

Bad output

The heaviest hippo ever recorded was a bull named Hubert, who weighed 5,000 kg or 11,023 lbs. Hubert was measured in Kenya in the early 20th century. Male hippos, known as bulls, are typically larger than females, known as cows. Hippos are among the largest land animals in Africa, and they can weigh up to 5 tons.

Good prompt

Human: Answer the following question only if you know the answer or can make a well-informed guess; otherwise tell me you don't know it.

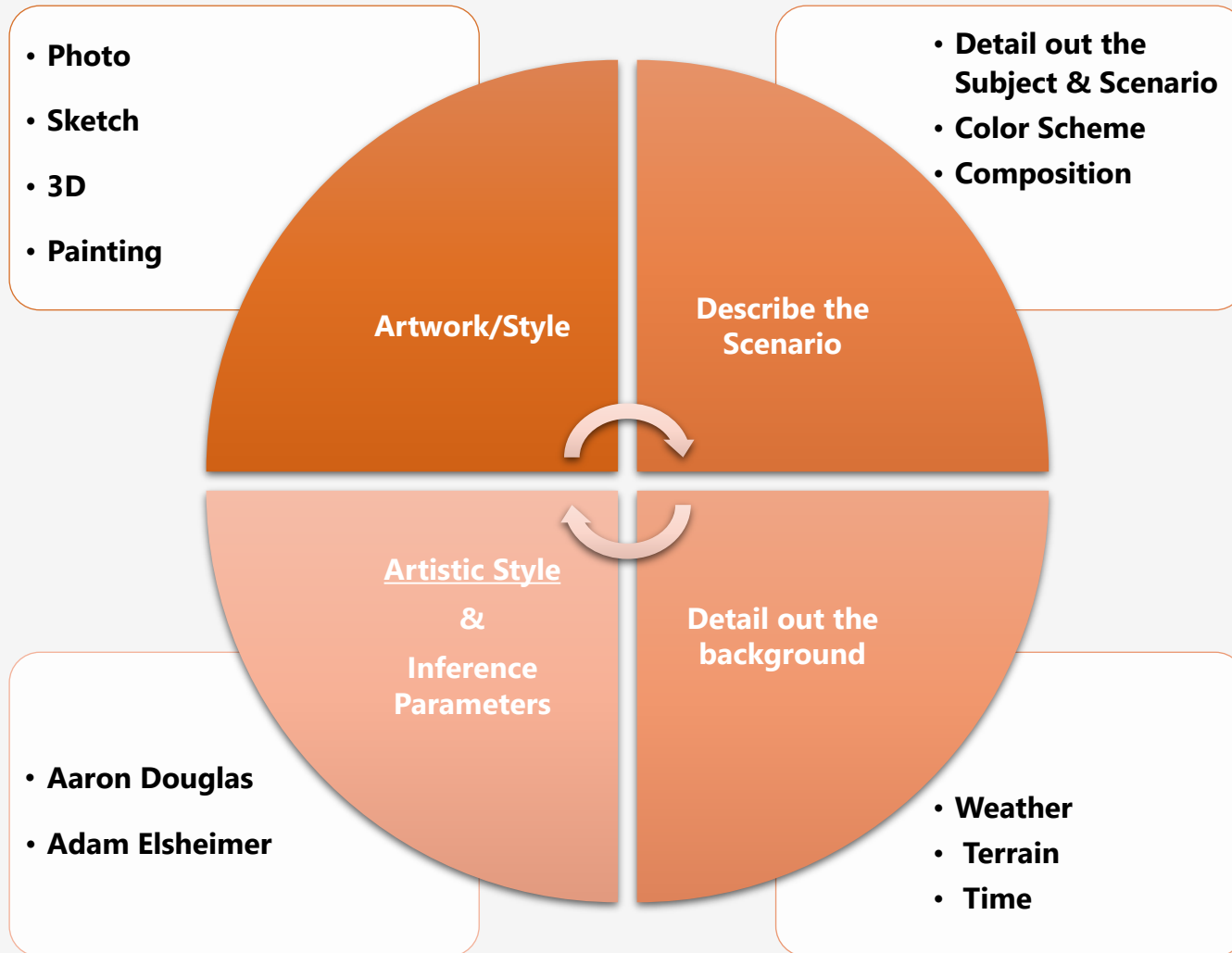
What was the heaviest hippo ever recorded?

Assistant:

Good output

I'm sorry, I don't know the exact weight of the heaviest hippo ever recorded. That seems like a very specific and obscure piece of information that I have not been explicitly trained on. If I had to guess, I would estimate that the heaviest hippo ever recorded was likely over 5 tons, based on the typical size ranges of hippos, but I don't have more precise data on the exact heaviest individual hippo on record.

Prompt Design for Stability AI



Start prompt with :

- “An image of
- Use detailed scenario
- Provide details
 - Medium
 - Color
 - Time of Day etc.
- Style
- Photo/Sketch etc.
- Inference Parameters

Prompt Design for Stability Diffusion - 1

An image of a spy agent fighting in a rival country with guns and helicopters with backdrop of a shopping complex with heavy snow and old **Greek architecture** building late in the **evening** with **sun setting behind mountains**. The image should be a **photograph** with Aaron Jasinski style

Prompt Design for Amazon Titan and AI21 – Text generation

Text generation

Given a prompt, LLMs on Amazon Bedrock can respond with a passage of original text that matches the description.

Prompt template for Amazon Titan and AI21 Jurassic: `""Please write a {{Text Category}} in the voice of {{Role}}. ""`

User prompt:

Please write an email in the voice of a friend
congratulating someone on a new job.

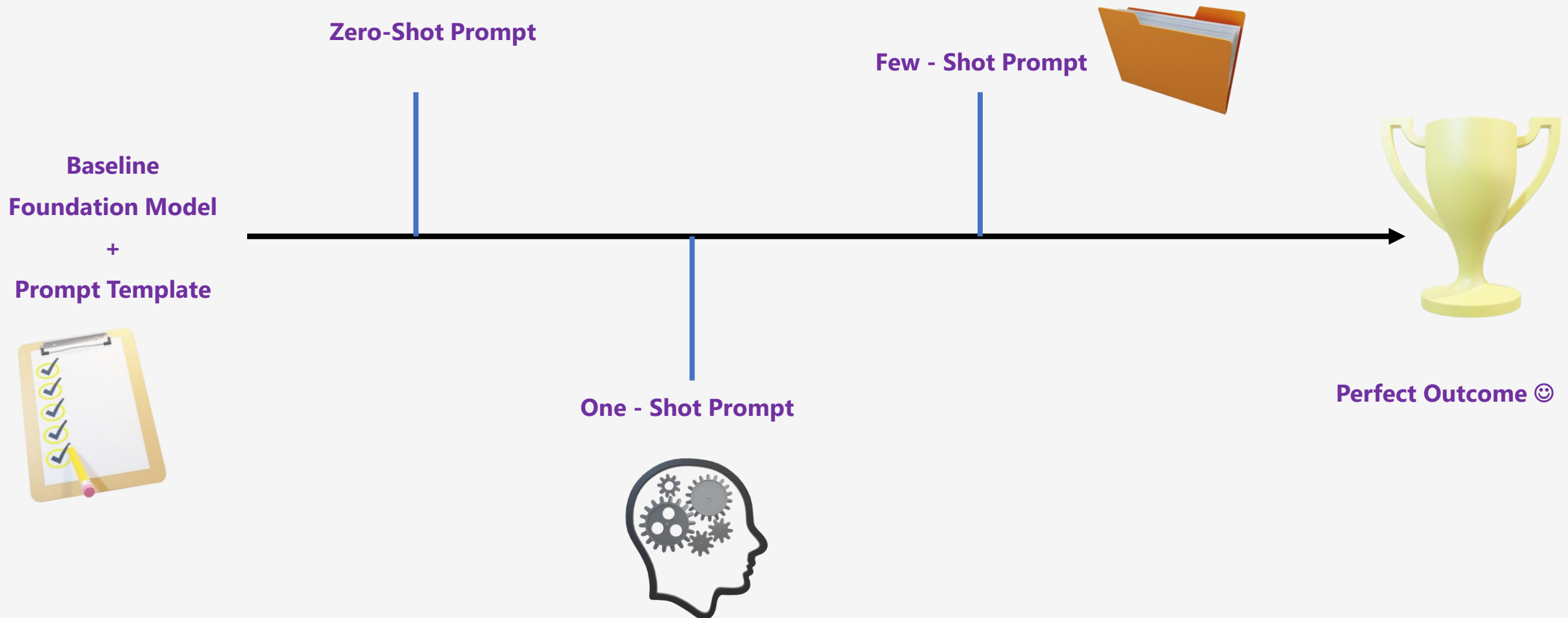
Output:

Hey [Name],

I just found out about your new job, and I'm so happy for you! Congratulations! You deserve this opportunity since you worked so hard to get it. I know you'll do great and make a positive impact on your team. I'm here to support you every step of the way.

Wishing you all the best in this exciting new chapter of your life!

Prompt Engineering Techniques



Zero-Shot Prompting

Zero-Shot Prompting

- FM's are capable of **performing some simple tasks "zero-shot."**
- **No examples** need to be provided to accomplish the task



PROMPT

Write an engaging product description for a T-shirt for a clothing eCommerce site:

COMPLETION

This T-shirt is perfect for any occasion. It's made of high-quality materials and it's comfortable to wear. It's also stylish and versatile, so you can wear it with anything.

One -Shot Prompting

One-Shot Prompting

- One-Shot Prompting provides **examples to the model** to assist the model to comprehend and generate responses

PROMPT

Write an engaging product description for a clothing eCommerce site.

Make sure to include the following features in the description.

Product: Women's Boho Beach Dress Features: - Midi dress - Swing hem - Slimming - Special for summer

Description: Women's Boho Beach Dress The style of midi dress with swing hem is slimming, It will make your tummy look flatter. This dress for women could not only make you feel cool, but also perfectly present the curve of female waist. Even in hot summer, it still keeps your elegant.

Write an engaging product description for a clothing eCommerce site.

Make sure to include the following features in the description.

Product: Humor Men's Graphic T-Shirt Features: - Soft cotton - Short sleeve - Have a print of Einstein's quote: "artificial intelligence is no match for natural stupidity"

Description:

COMPLETION

This soft cotton t-shirt is perfect for fans of the classic scientist, Albert Einstein. The quote "artificial intelligence is no match for natural stupidity" is a reminder that sometimes the most intelligent people are the most foolish. So whether you're a science nerd or just a fan of witty quotes, this t-shirt is for you.



Few - Shot Prompting

Few-Shot Prompting

- Few examples are provided to the Model to generate high quality Completion
- Good for Classification

PROMPT

Write an engaging product description for a clothing eCommerce site. Make sure to include the following

Product: Camp Director Crossbody Bag

Features:

- Canvas purse
- Rainbow space print
- Leather trim
- Two securely-zipped compartments

Description:

This Camp Director Crossbody Bag is the perfect accessory for carrying your essentials in style. The canvas is made of high-quality material and features a vibrant rainbow space print. It has a leather trim and two securely-zipped compartments for easy access to your items.

Write an engaging product description for a clothing eCommerce site. Make sure to include the following

Product: Women's Boho Beach Dress

Features:

- Midi dress
- Swing hem
- Slimming
- Special for summer

Description:

Women's Boho Beach Dress The style of midi dress with swing hem is slimming, It will make your tummy look flatter.

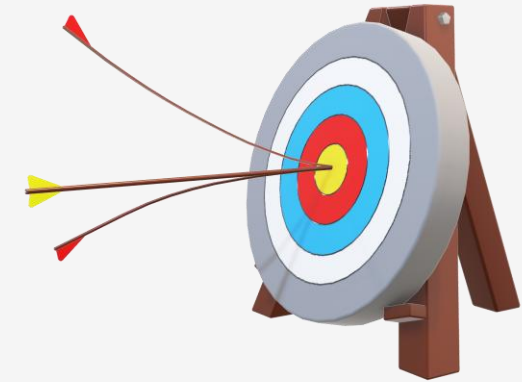
Write an engaging product description for a clothing eCommerce site. Make sure to include the following

Product: Humor Men's Graphic T-Shirt

Features:

- Soft cotton
- Short sleeve
- Have a print of Einstein's quote: "artificial intelligence is no match for natural stupidity"

Description:



COMPLETION

This Humor Men's Graphic T-Shirt is perfect for anyone who wants to make a statement.

The soft cotton and short sleeve make it comfortable to wear, and the print of Einstein's quote is sure to get a chuckle.

Whether you're a fan of science or just looking for a funny shirt, this shirt is a great choice.

Python Refresher

(from my Udemy Course on :

*AWS Lambda, Python(Boto3) & Serverless- Beginner
to Advanced)*

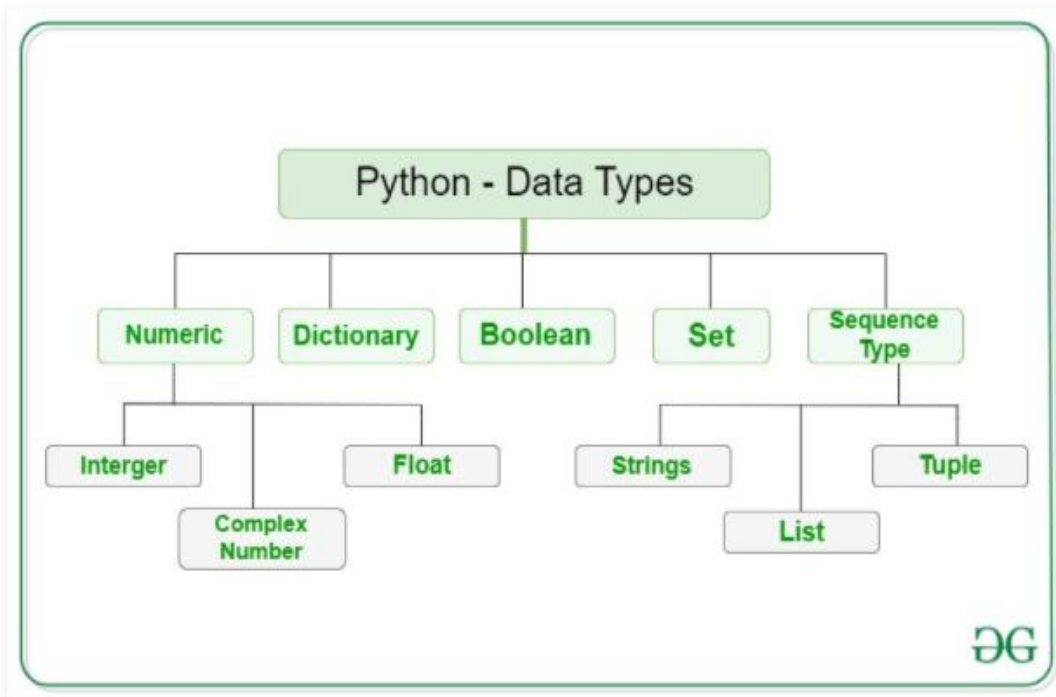
Python Basics–Refresher



1. Print Function – print the message to screen or any interface; Syntax : `print()`

2. Variables - Containers for storing data values string, float or integers and no need to declare; Syntax : `x = 3`, `greeting = "hello"` etc.

3. Data Type



| Data Type | Example |
|------------|--|
| Int | <code>x = 20</code> |
| Float | <code>x = 20.5</code> |
| Dictionary | <code>x = {"name" : "John", "age" : 36}</code> |
| Strings | <code>x = "Hello World"</code> |
| List | <code>x = ["apple", "banana", "cherry"]</code> |

Python Basics—Refresher

3. Data Types – Dictionary



- curly brackets
- key: pair values
- Nested Dictionary

Dict

```
response = {1: 'Rahul', 2: 'John', 3: 'Joy'}
```

Nested Dict

```
response = {1:'Python', 2: {'books': 'arch' , 'aws': 'Lambda'}}
```

4. Nested Dictionary

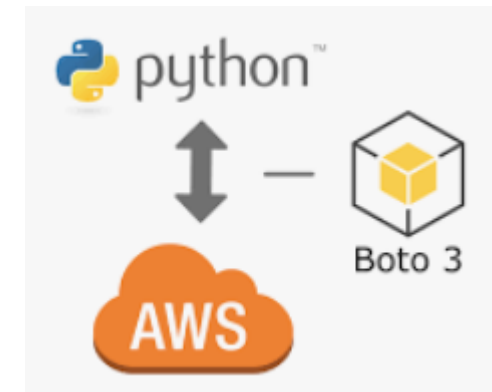


Python Basics–Refresher

5. Sample Example from Boto3

Response Syntax

```
{
  'Buckets': [
    {
      'Name': 'string',
      'CreationDate': datetime(2015, 1, 1)
    },
  ],
  'Owner': {
    'DisplayName': 'string',
    'ID': 'string'
  }
}
```



https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3/client/list_buckets.html#

Python Basics–Refresher

6. Data Types – List

- Lists in Python can be created by just placing the sequence inside the **square brackets []**
- A single list may contain Data Types like Integers, Strings, as well as Objects.
- List in Python are ordered and have a definite count. The elements in a **list are indexed with 0 being the first index.**
- **slice(start, stop, step)**
- Reverse [: : -1]

```
.....  
list = [1, 4, 'For', 6, 'Anisha']  
.....
```

7. Nested List

Nested List => `nestedList = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`

Python Basics—Refresher

8. Data Types – List and Dictionary



9. Data Type Determination

```
response = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
print(type(response))
```

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3/client/list_buckets.html#

Response Syntax

```
{  
  'Buckets': [  
    {  
      'Name': 'string',  
      'CreationDate': datetime(2015, 1, 1)  
    },  
  ],  
  'Owner': {  
    'DisplayName': 'string',  
    'ID': 'string'  
  }  
}
```



Python Basics—Refresher

10. Function

Python Function

A function is a block of code which runs when it is called.

Syntax:

```
def function_name (argument/parameters):
```

```
    return expression or value
```

Example

A simple Python function to check whether x is even or odd

```
def evenOdd(x):
```

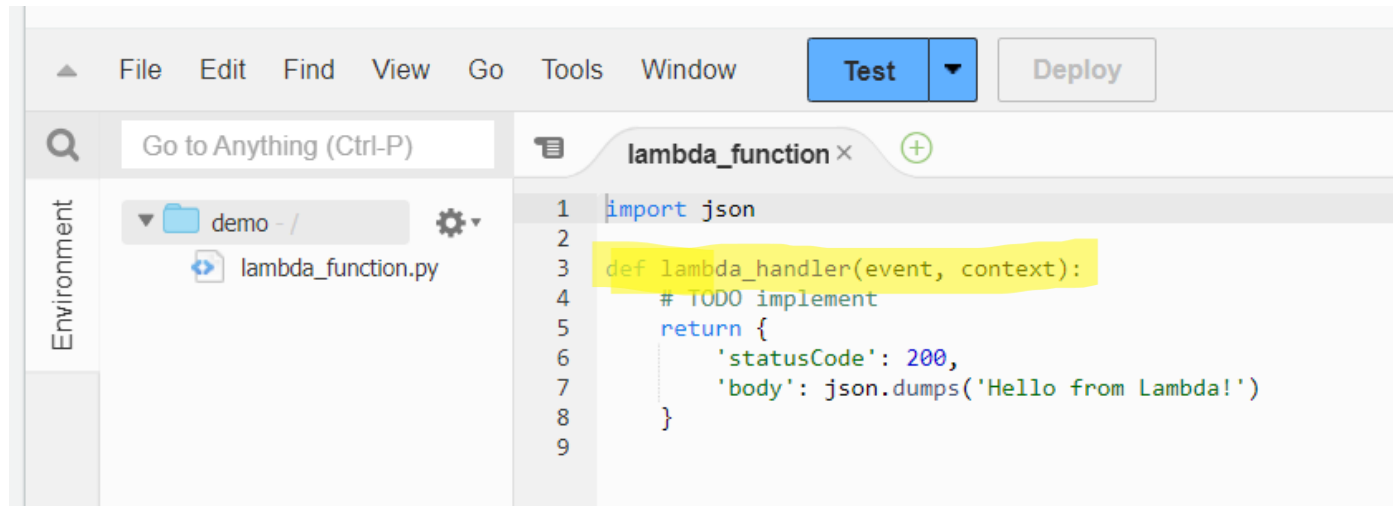
```
    if (x % 2 == 0):  
        print("even")
```

```
    else:  
        print("odd")
```

Driver code to call the function

```
evenOdd(2)
```

```
evenOdd(3)
```



Thank You