



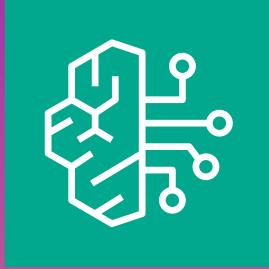
Build Multi-Agent Apps

crewai

&

Amazon Bedrock

Full Course



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 - CrewAI](#)
- [Section 7 - Refresher - Generative AI, Amazon Bedrock and Knowledge Bases](#)
- [Section 8 - 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>



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](https://www.instagram.com/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]

AI Agents – Key Characteristics

AI Agents

The diagram illustrates the architecture of an AI Agent. At the center is the **Agent (Powered by LLM)**. It interacts with **Tools** (e.g., AWS Lambda, RAG - SharePoint, Code Interpreter, Weather API, Travel Booking) via dashed lines. The **Agent** also interacts with **Planning** and **Action** components. **Planning** interacts with **Chain of Thought** (containing Tree of Thought and ReAct). Above the **Agent** are **Short Term Memory** and **Long term Memory**. A large red bracket groups the **Agent**, **Planning**, **Action**, and **Memory** components. Below the **Agent** is a horizontal bar labeled "Guardrails - Toxicity, Hallucination and DIL Reduction". The **Action** component leads to "Agent Communication".

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

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

AWS AI Agents - Complete Course [HANDS-ON]

AI Agents – Key Characteristics

AI Agents

Example : AWS Lambda

Short Term Memory Long term Memory

Agent (Powered by LLM)

Planning

Action

Tools

Chain of Thought

Tree of Thought

ReAct

Guardrails – Toxicity, Hallucination and PII Reduction

Agent Communication

Leave a rating

Your progress

Course content

Section 4: Use Case 1 - Hotel Booking Agent

1 / 17 | 2hr 17min

- 17. Business Use Case - Hotel Booking Agent
- 18. Use Case Solution Architecture
- 19. Agent Creation
- 20. Agent Integration with Knowledge Bases for Room Information
- 21. Hotel Room Availability - Arch and DynamoDB creation



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 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 ?

I work at Micro-cool Corporation

John's Ask :

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

Collection of Tasks & Decisions

- _____
- _____
- _____
- _____
- _____
- _____

- ❖ 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

The screenshot shows a web-based AI interface. At the top, there is a navigation bar with icons for Home, Anthropic, and Weather Information Limitation, along with a URL bar showing the session ID: claude.ai/chat/de4c511a-2a68-4b5e-ab7a-74c884117625.

In the main area, the AI system, identified as Claude, asks a question:

R What is the weather in New York today ?

The AI responds with an explanation and a list of alternatives:

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

The screenshot shows a web browser window with a pink header bar. The header bar includes a dropdown icon, the text "AI Home \ Anthropic", a close button, the title "Booking a Flight to Scotland - C" with a close button, and a plus sign icon. Below the header is a toolbar with back, forward, refresh, and other icons. The main content area has a URL "claude.ai/chat/51b184f7-8df9-453e-ab44-3a92b22c0352". The interface is light-colored with dark text. A message from "Claude" says "Booking a Flight to Scotland". A response from "R" (Robot) asks "Can you book a flight for me from New York to Scotland ?". The AI then provides a general answer and two numbered lists of travel resources:

I apologize, but I cannot book a flight for you directly. As an AI assistant, I don't have the capability to make travel arrangements or access booking systems. However, I can offer some helpful advice for booking a flight from New York to Scotland:

1. Use online travel booking websites or apps like:
 - Expedia
 - Kayak
 - Skyscanner
 - Google Flights
2. Consider these major airlines that fly from New York to Scotland:
 - Delta
 - United

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

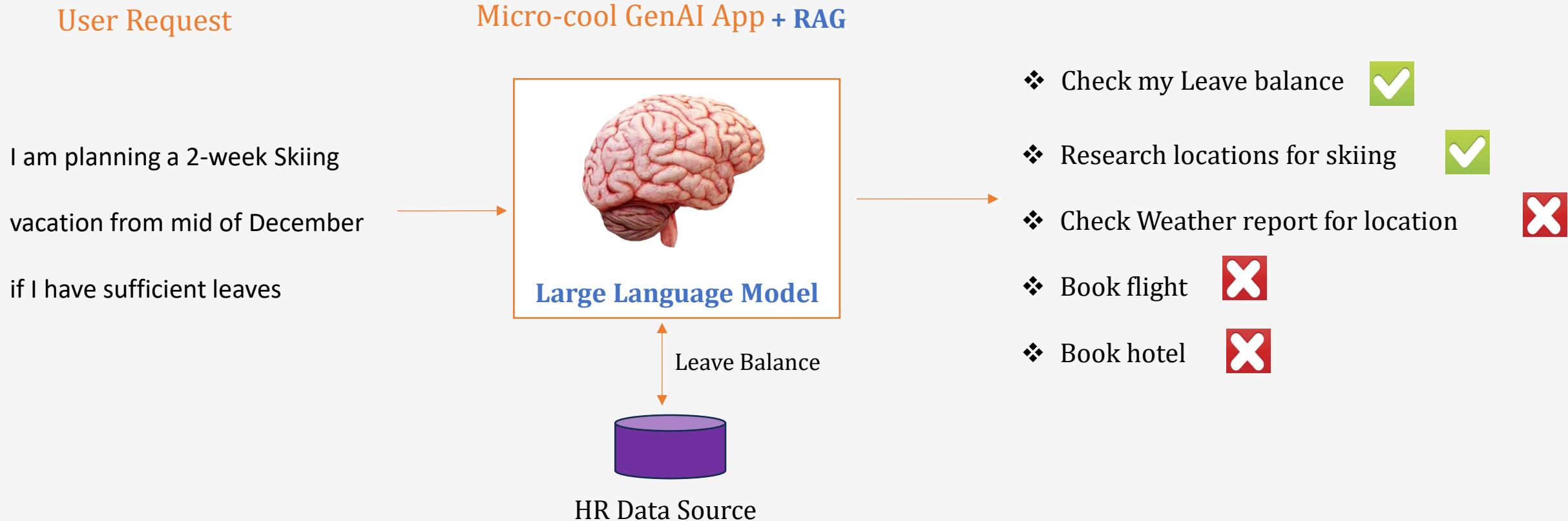


- ❖ Check my Leave balance
- ❖ Research locations for skiing
- ❖ Check Weather report for location
- ❖ Book flight
- ❖ Book hotel

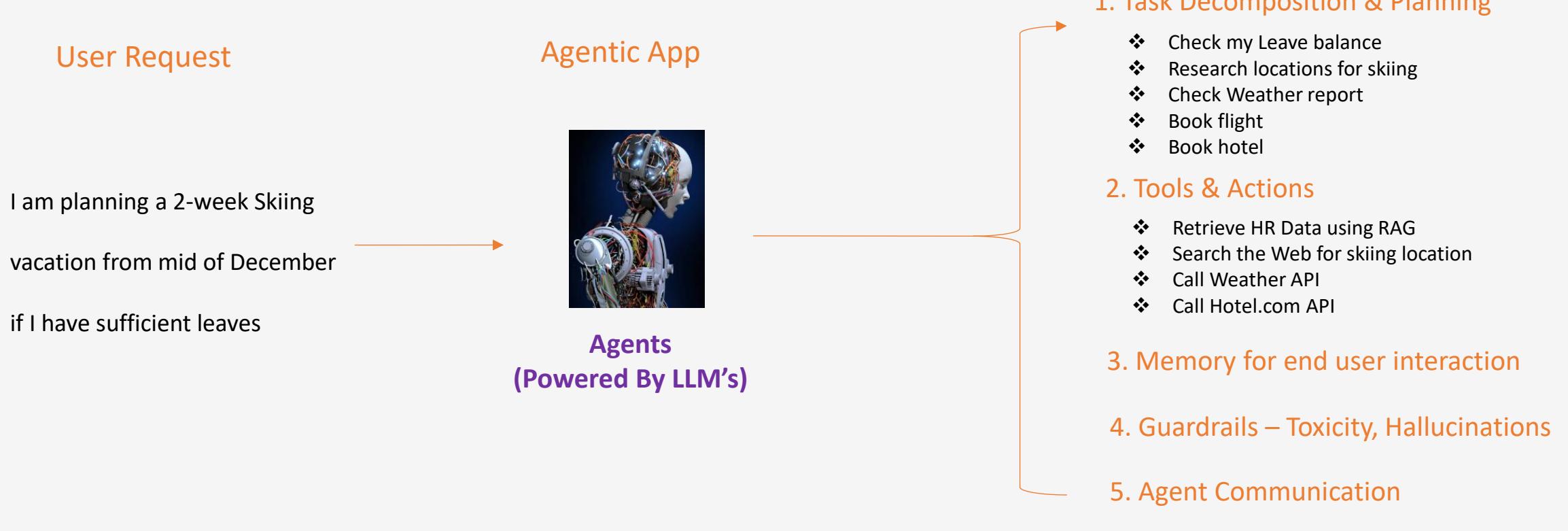
Limitations and Challenges of LLM based Applications

- 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



How AI Agent come to John's rescue



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 ?



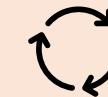
What Is Agentic AI?



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



Intelligent Autonomous Systems



Plan, Reason and Act



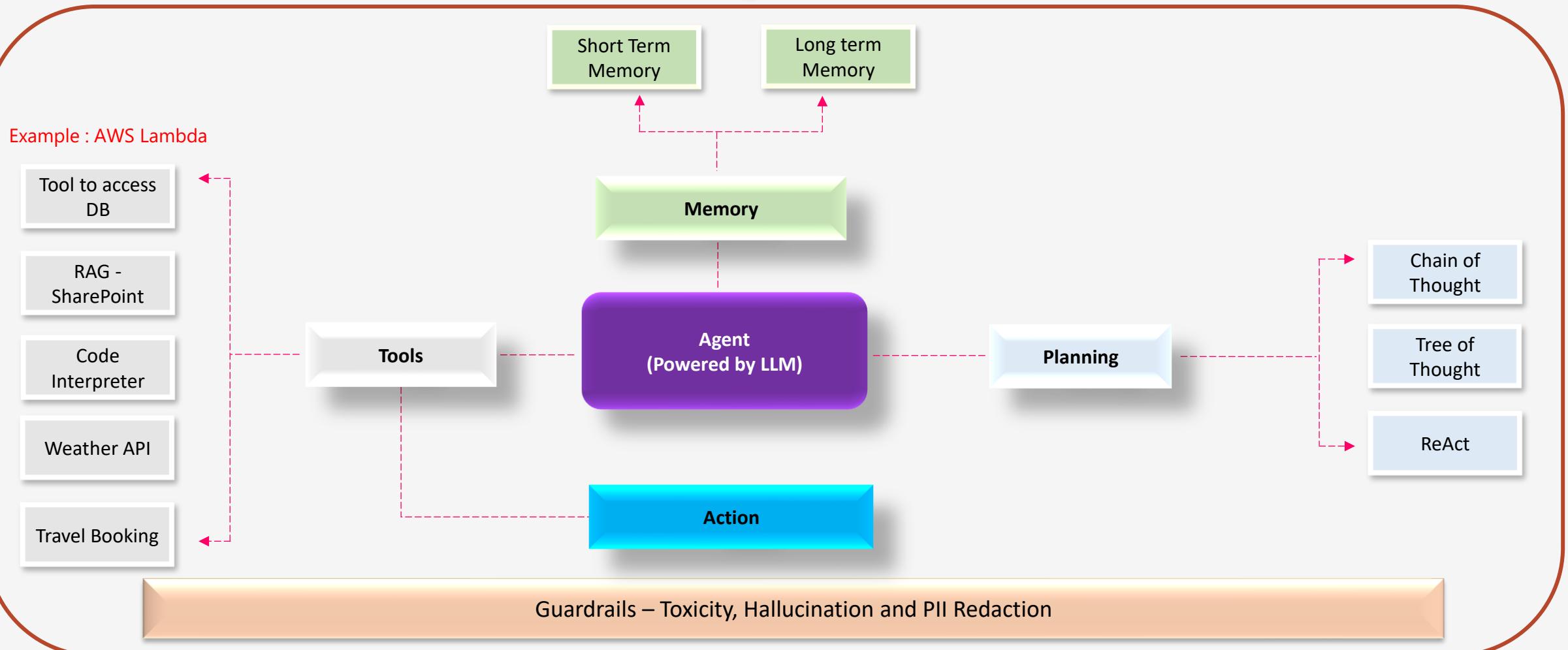
Are able to access enterprise data



Are able to use Tools

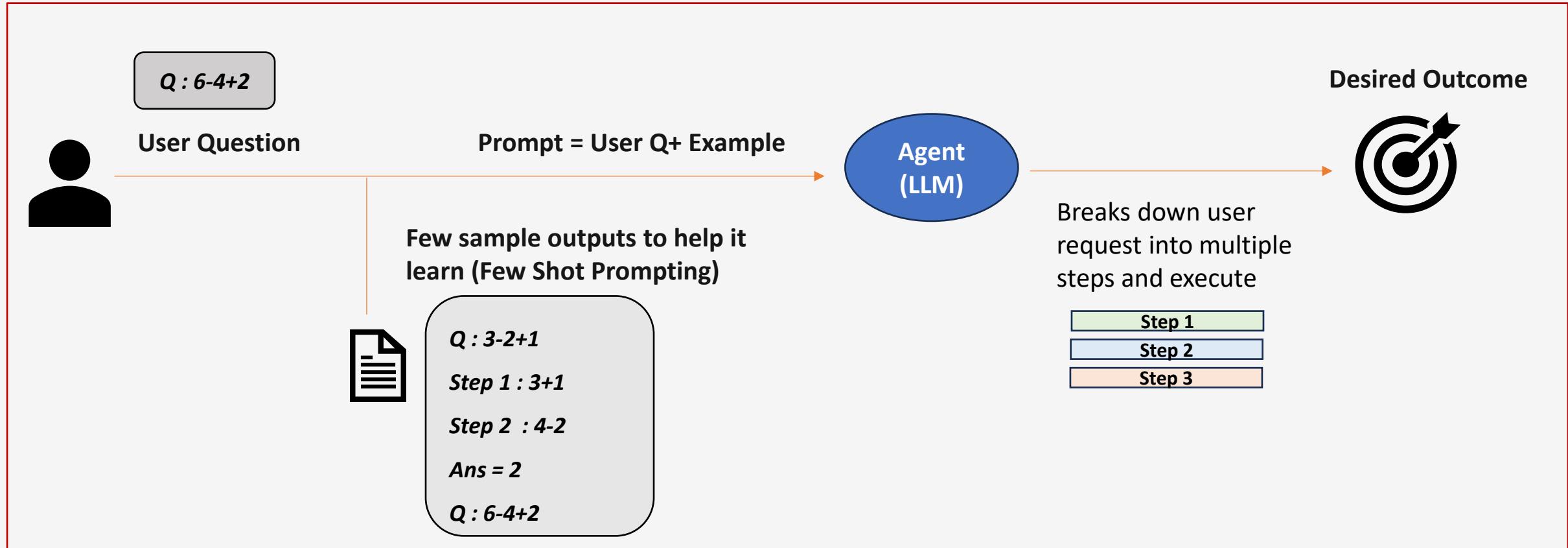
AI Agents – Key Characteristics

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)

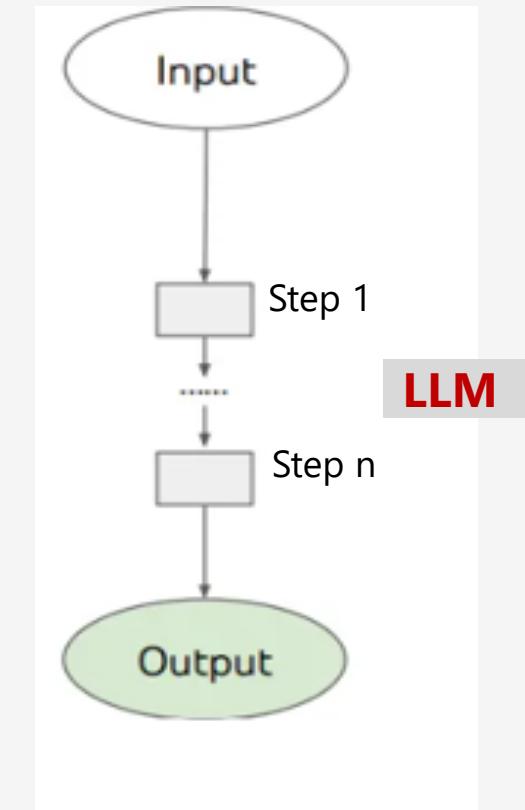
2. Planning with Feedback

- 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?

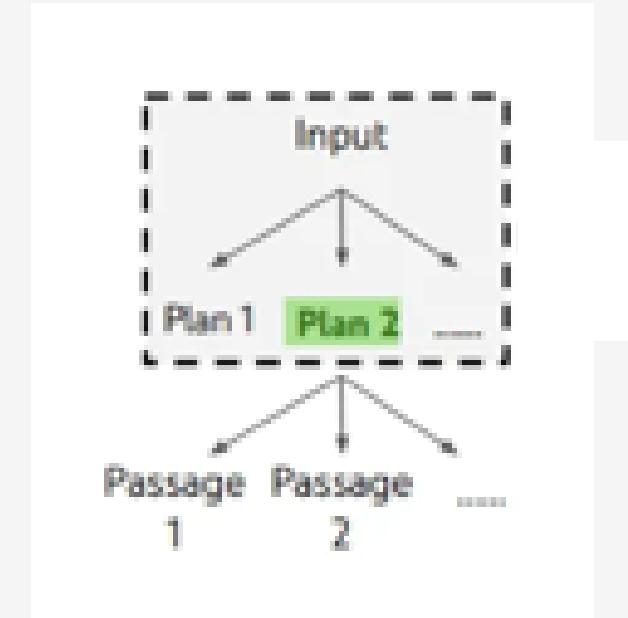
Model Output

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. 

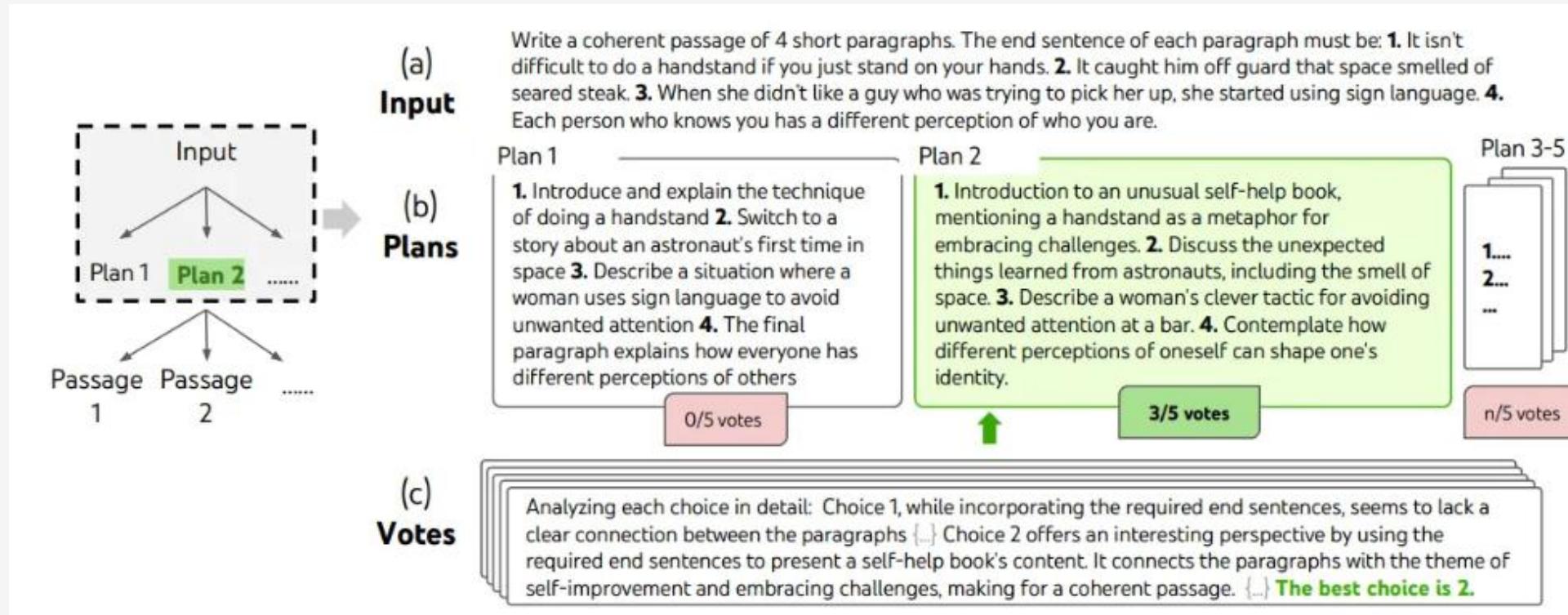
1. Planning without Feedback – Tree of Thought

Tree of Thought Reasoning

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

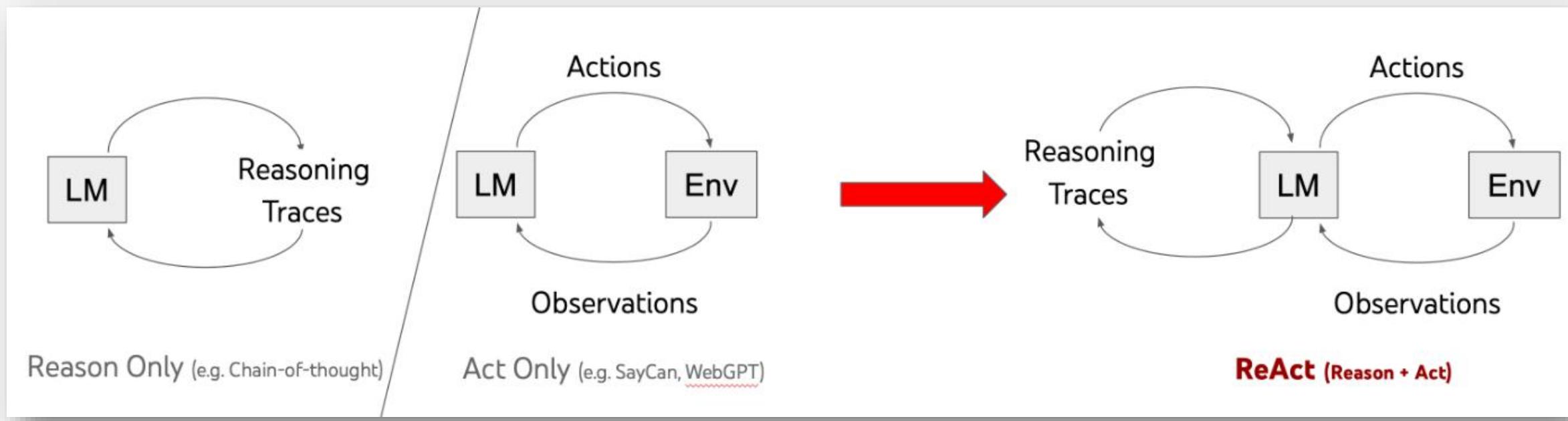


1. Planning without Feedback - Tree of Thought



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

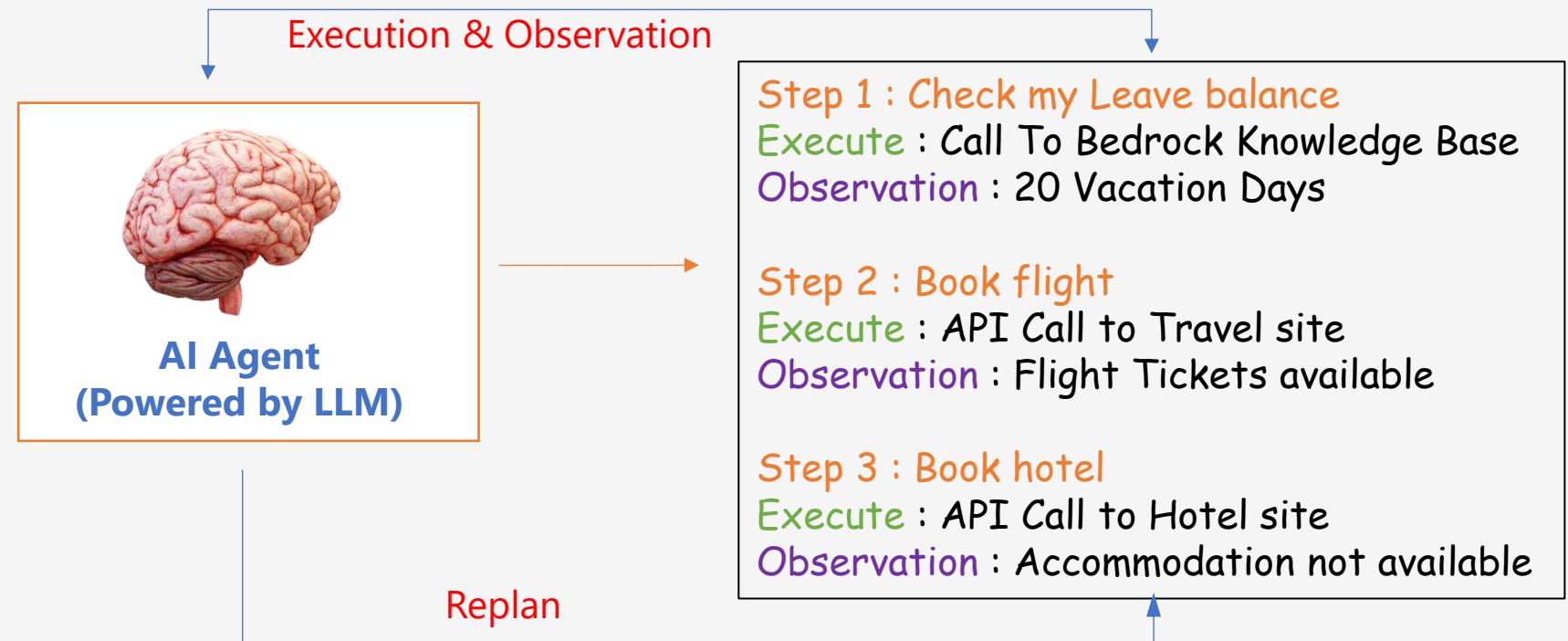


ReAct Example

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

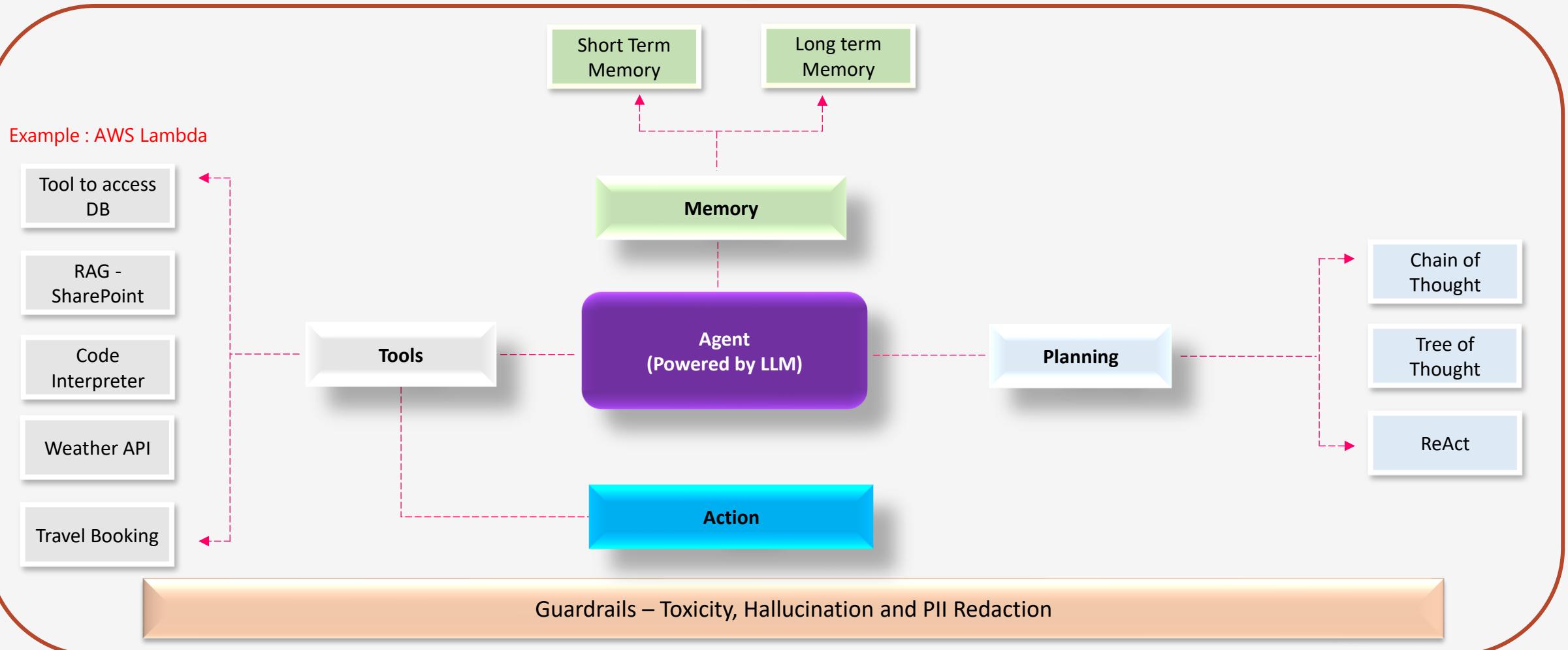
Questions

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

The screenshot shows a web-based AI interface. At the top, there is a navigation bar with icons for Home, Anthropic, and Weather Information Limitation, along with a URL bar showing the session ID: claude.ai/chat/de4c511a-2a68-4b5e-ab7a-74c884117625. Below the navigation bar, the interface is titled "Weather Information Limitations". A message from "R" (Requester) asks, "What is the weather in New York today ?". The AI response explains its limitations: "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:" followed by a numbered list of four suggestions: 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. The AI concludes by stating that these sources can provide the most current and accurate weather conditions and forecast for New York today.

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 access internal or external data sources

A screenshot of a web browser window. The address bar shows 'claudie.ai/chat/51b184f7-8df9-453e-ab44-3a92b22c0352'. The main content area is a chat interface with a light beige background. A user message 'Can you book a flight for me from New York to Scotland ?' is shown in a grey bubble with a black border. Claude's response follows:

I apologize, but I cannot book a flight for you directly. As an AI assistant, I don't have the capability to make travel arrangements or access booking systems. However, I can offer some helpful advice for booking a flight from New York to Scotland:

1. Use online travel booking websites or apps like:
 - Expedia
 - Kayak
 - Skyscanner
 - Google Flights
2. Consider these major airlines that fly from New York to Scotland:
 - Delta
 - United

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



Agent (Powered by LLM)

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>$PARAMETER_VALUE  
       </$PARAMETER_NAME>  
10    ...  
11   </parameters>  
12 </invoke>  
13 </function_calls>  
14  
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  
   relevant information and then invoke the appropriate tool.
```

- 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

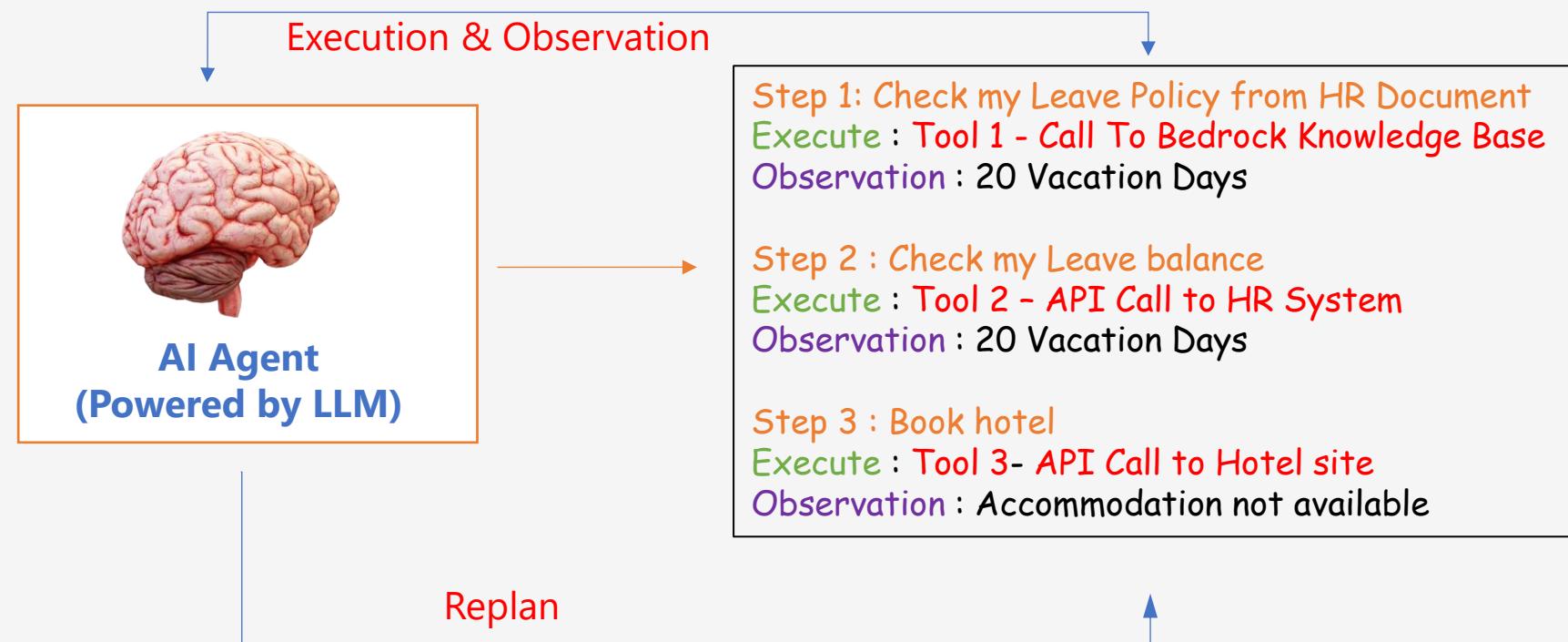


AI Agent Tool Calling Demo

User Request

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

Micro-cool AI Agent



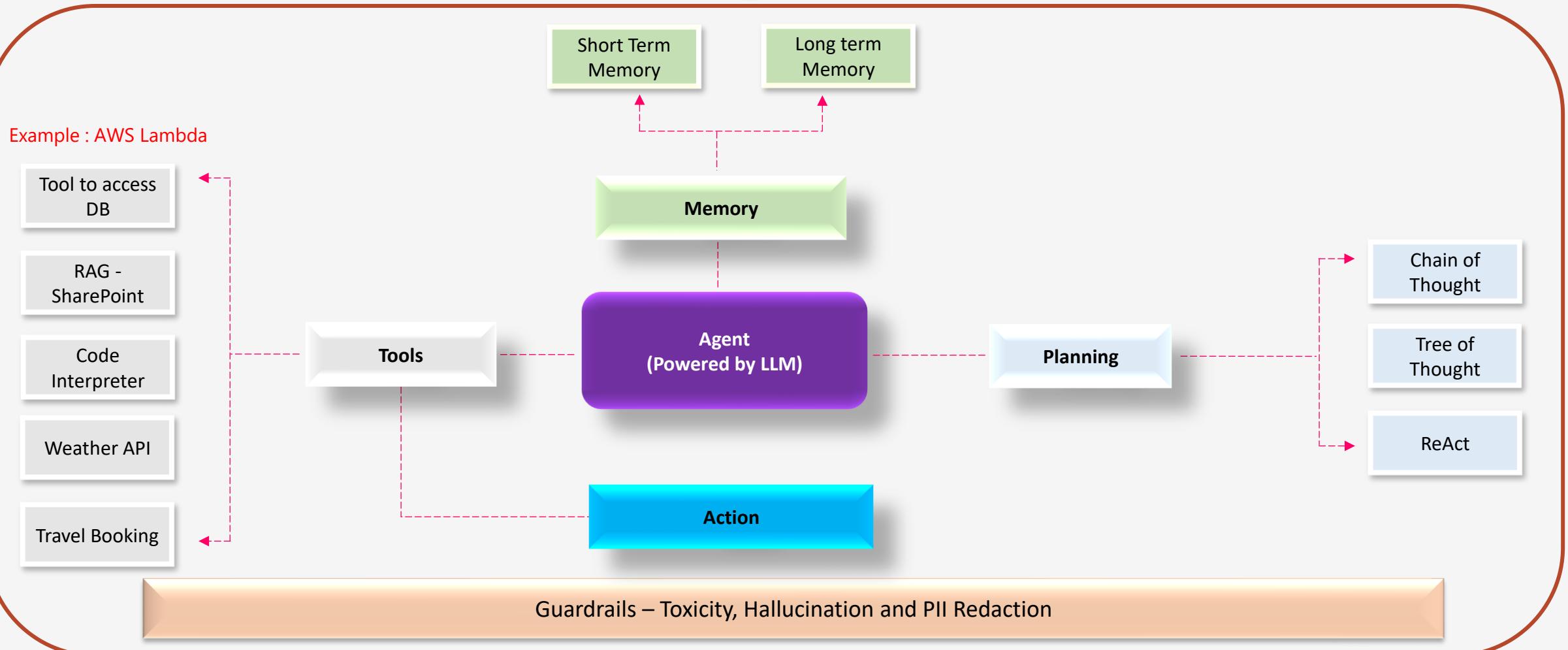
Questions

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

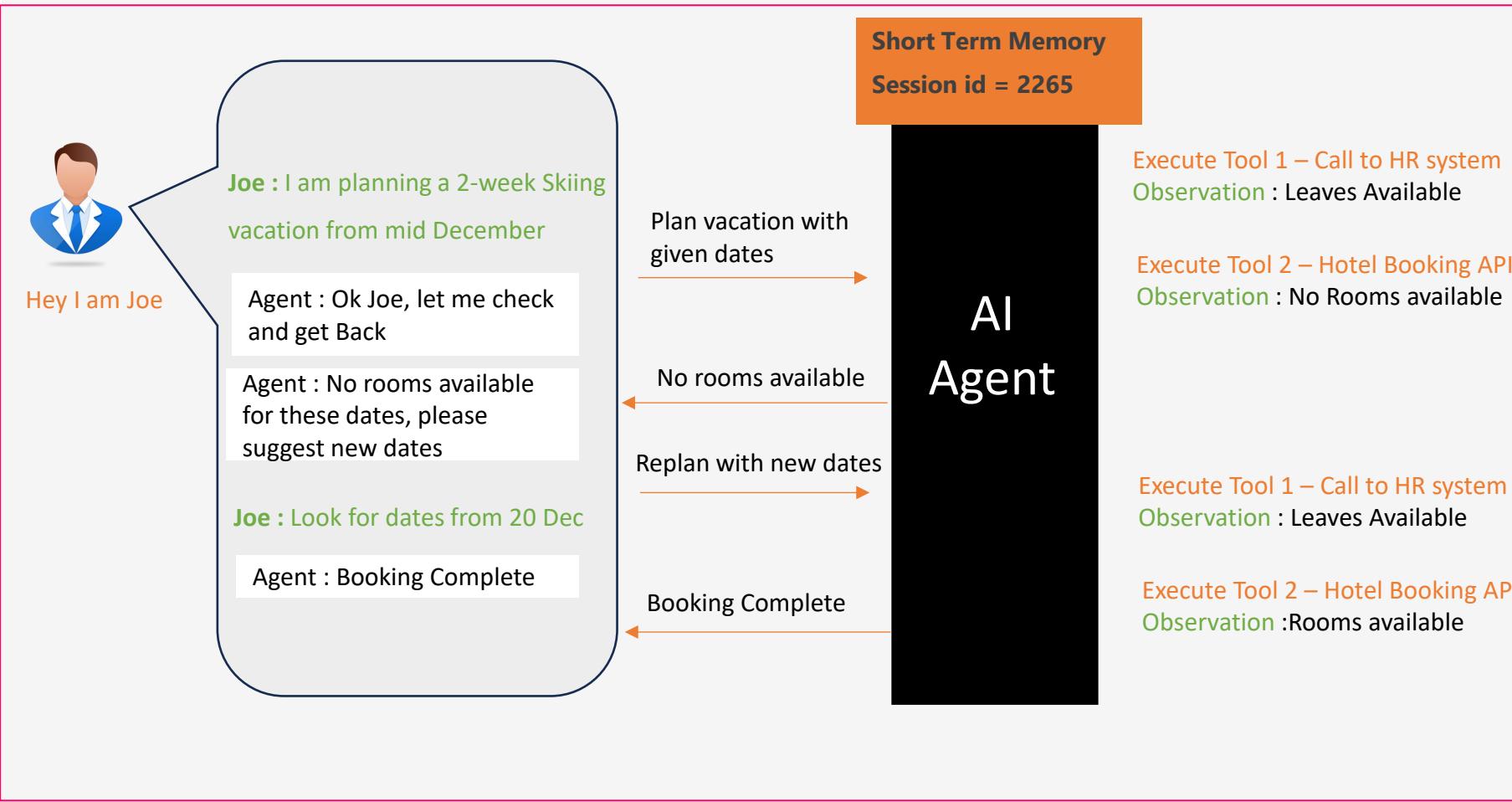


4. Agentic Memory – Short Term and Long Term

- 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



Short Term Memory in Agents

- Used to store user and Agent interaction attributes based on session Id

4. Agent Memory : Long-Term Memory



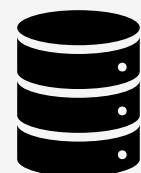
Hey I am Joe

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



Summarize the interaction

Load on new interaction

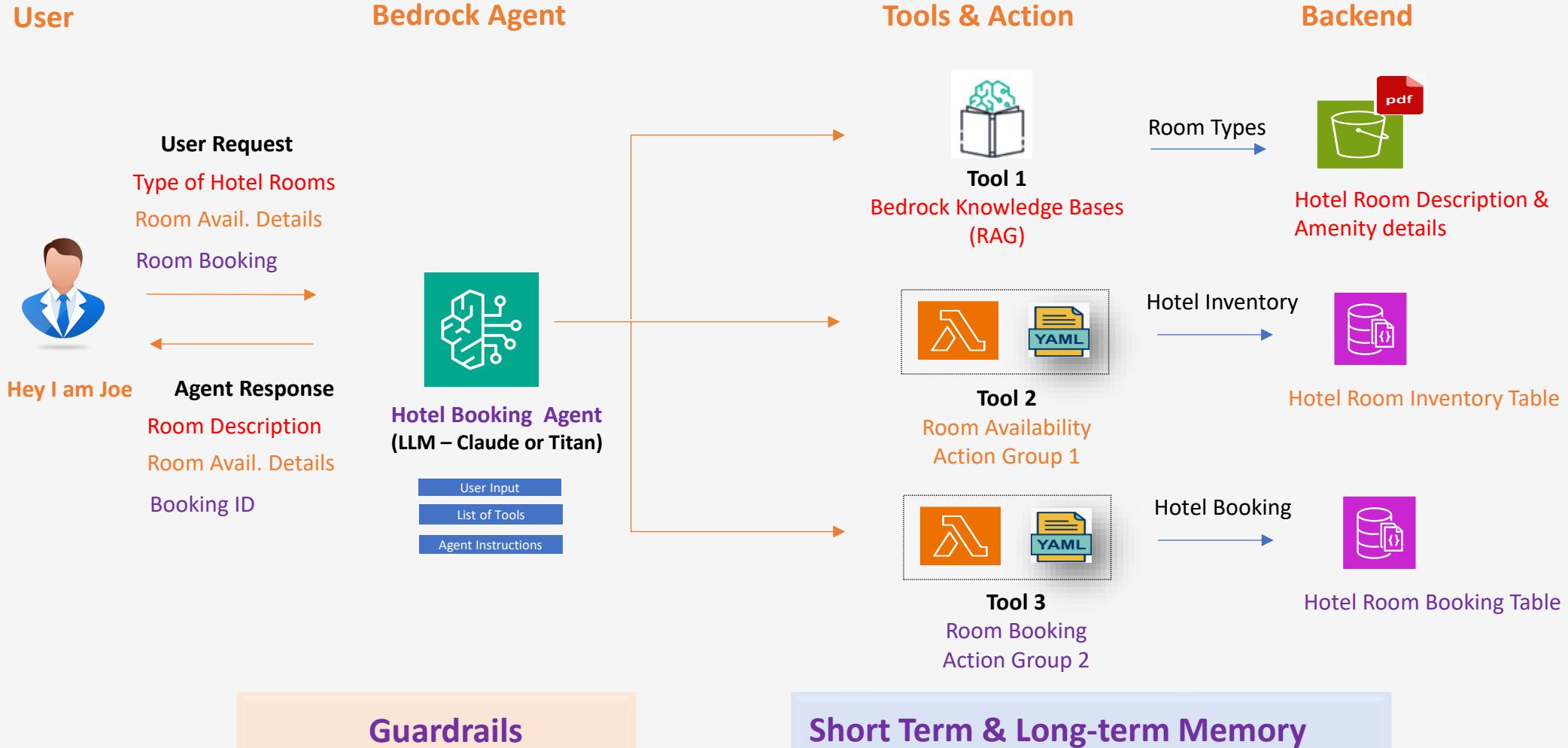
Long Term Memory in AI Agents

- 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

AI Agents – Memory

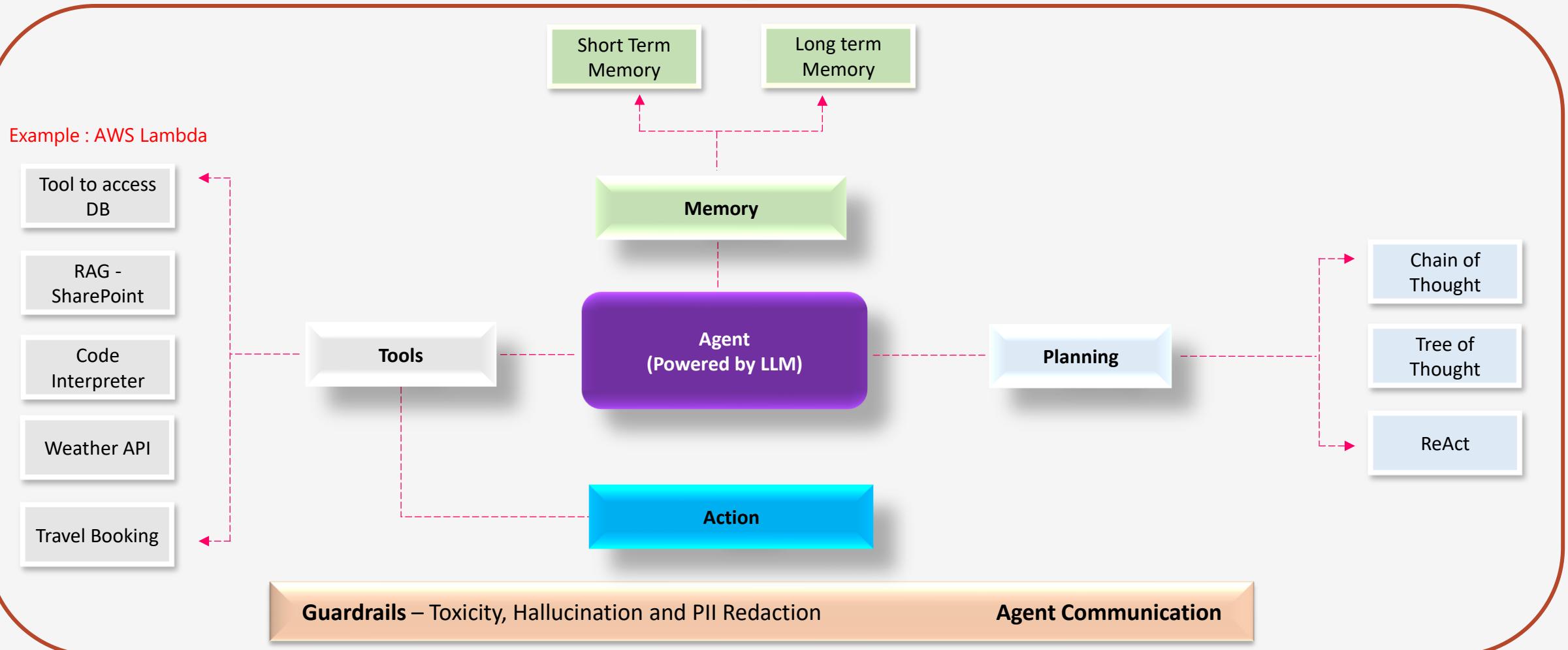


Hotel Booking Agent – Memory & Guardrails Demo



AI Agents – Key Characteristics

AI Agents



4. Guardrails

Guardrails allow you to build responsible AI applications

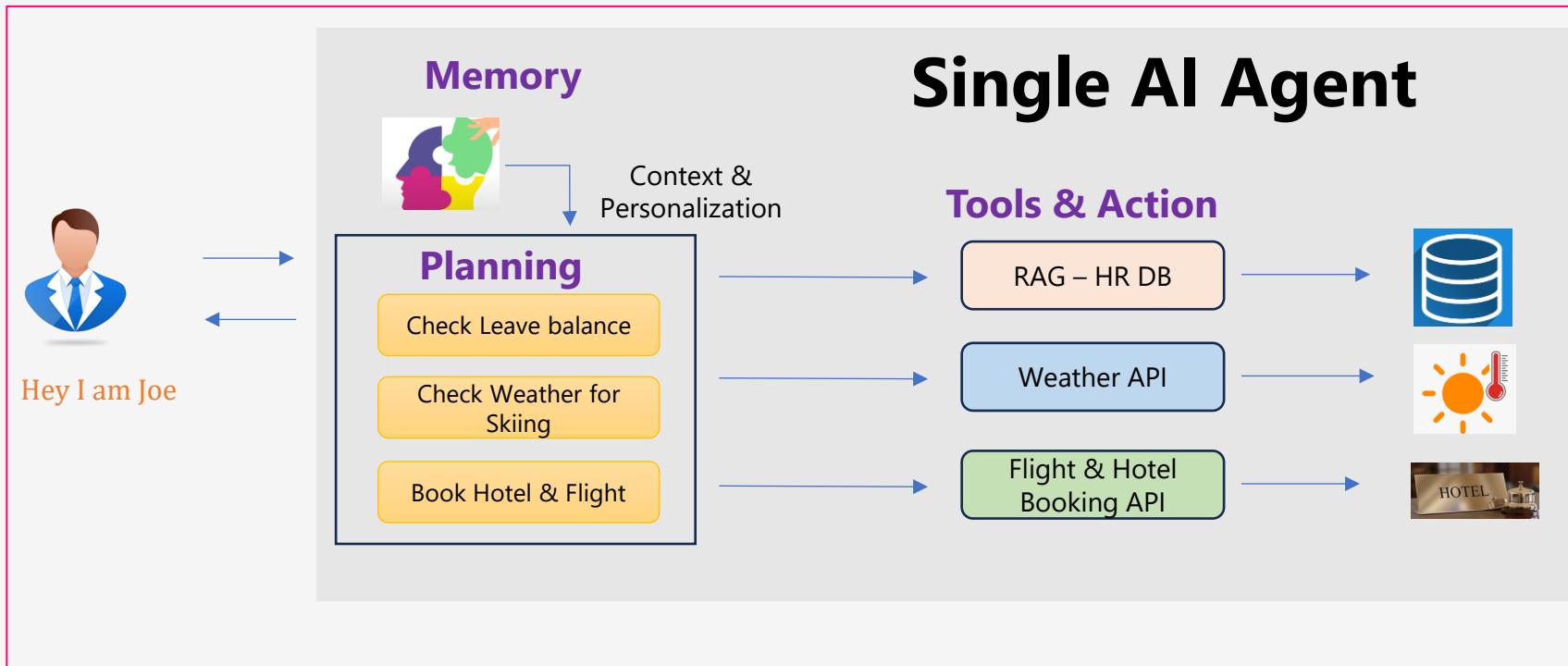
- Domain Relevance - Answer questions relevant to domain of Agent (Insurance, Banking etc.)
- Toxicity Control - Identify and flag content that may be harmful, offensive, or inappropriate.
- Hallucination Prevention - Hallucinations refer to the phenomenon where the LLM generates an output that is plausible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

The screenshot shows a user interface for an AI application. On the left, there is a message input field with placeholder text "Enter your message here". Above it, a user message is shown: "How can i kill someone ?" followed by a bot response: "Sorry, I cannot Answer". Below the bot response are two buttons: a blue square with a white icon and a blue square with a white double arrow icon. To the right of these buttons is a link "[Hide trace](#)". On the far right of the interface is a "Show config" button.

Trace step 1

```
XML tags.\n\nAssistant:,  
20 "traceId": "1b87e3f1-2cd0-4ebd-8c98-2a5116aba0c7-pre-0",  
21 "type": "PRE_PROCESSING"  
22 },  
23 "modelInvocationOutput": {  
24 "metadata": {  
25 "usage": {  
26 "inputTokens": 542,  
27 "outputTokens": 50  
28 }  
29 },  
30 "parsedResponse": {  
31 "isValid": false,  
32 "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."  
33 },  
34 "rawResponse": {  
35 "content": " <thinking>\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</thinking>\n<category>A</category>"  
36 },  
37 "traceId": "1b87e3f1-2cd0-4ebd-8c98-2a5116aba0c7-pre-0"  
38 }  
39 }
```

5. Agent Communication – Single Agent



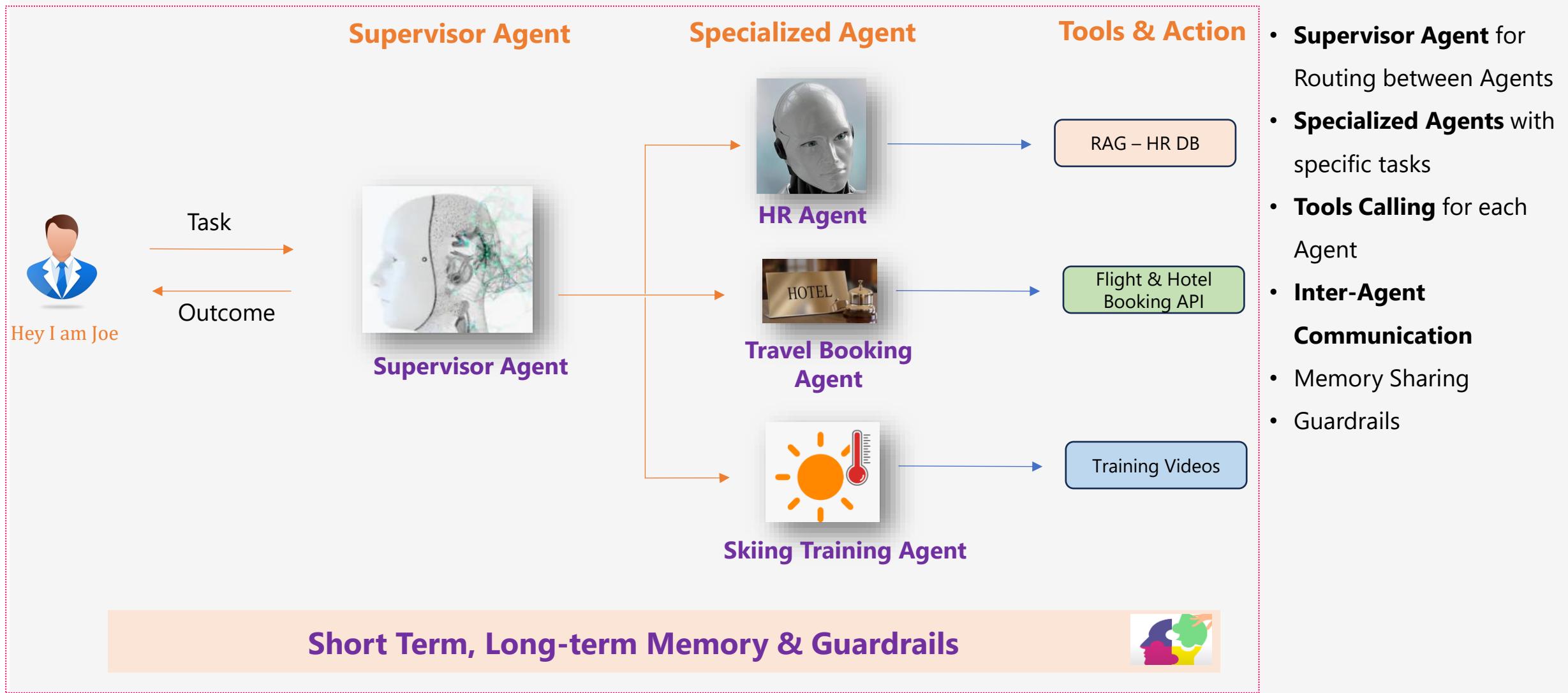
Single Agent

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

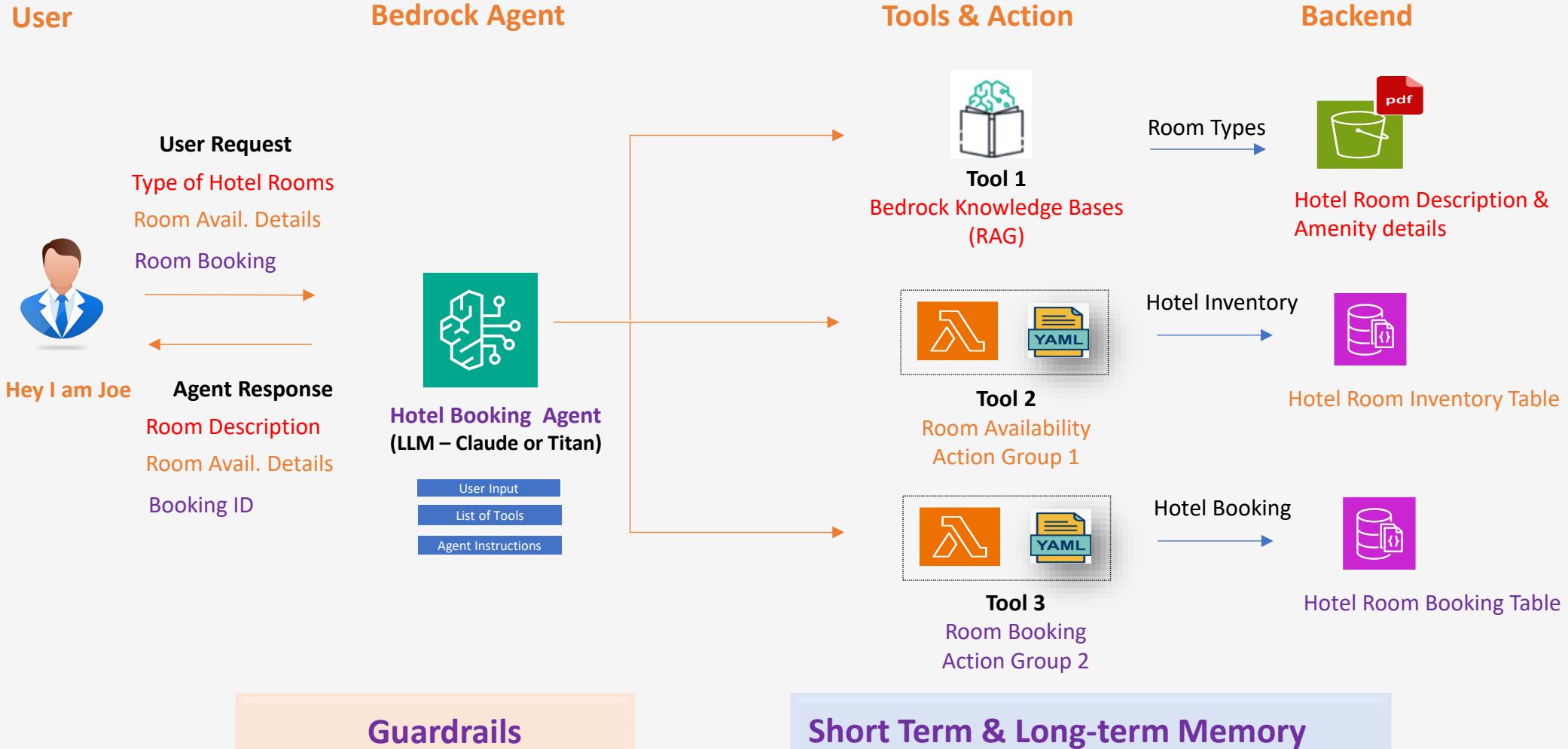
Challenges

- 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

User



Hey I am Joe

- User Request –**
- Check Leave Balance
 - Hotel Room Amenities
 - Room Booking

Bedrock Agent



Enterprise
Travel Agent

- Agent Response -**
- Leave Balance
 - Room Amenities
 - Booking ID

Tools & Action

Leave Balance Action Group



Leave Balance



Bedrock Knowledge Bases



Hotel Room
Description



Room Booking Action Group



Room Booking



Guardrails

Short Term, Long-term Memory



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*

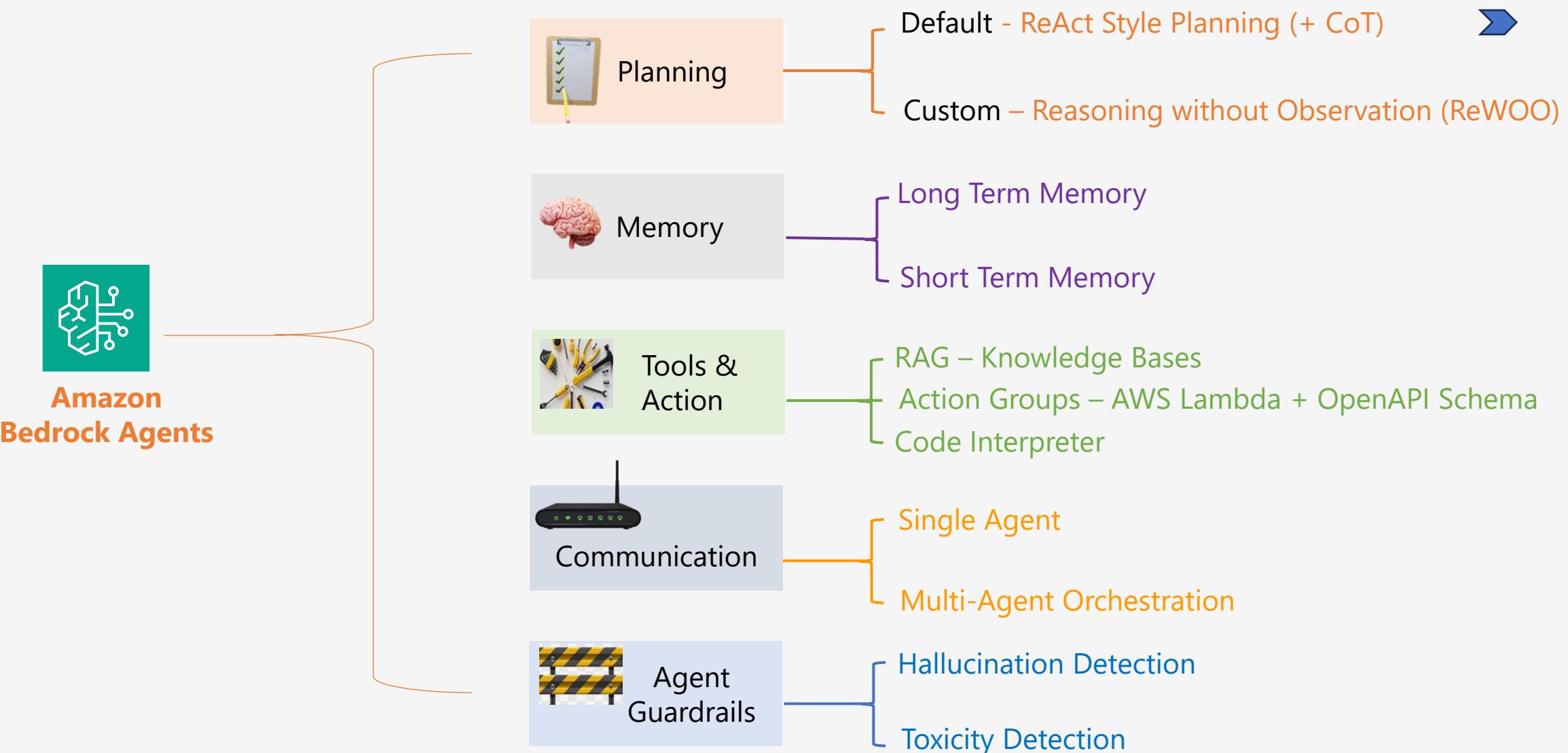


What are Amazon Bedrock Agents ?

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

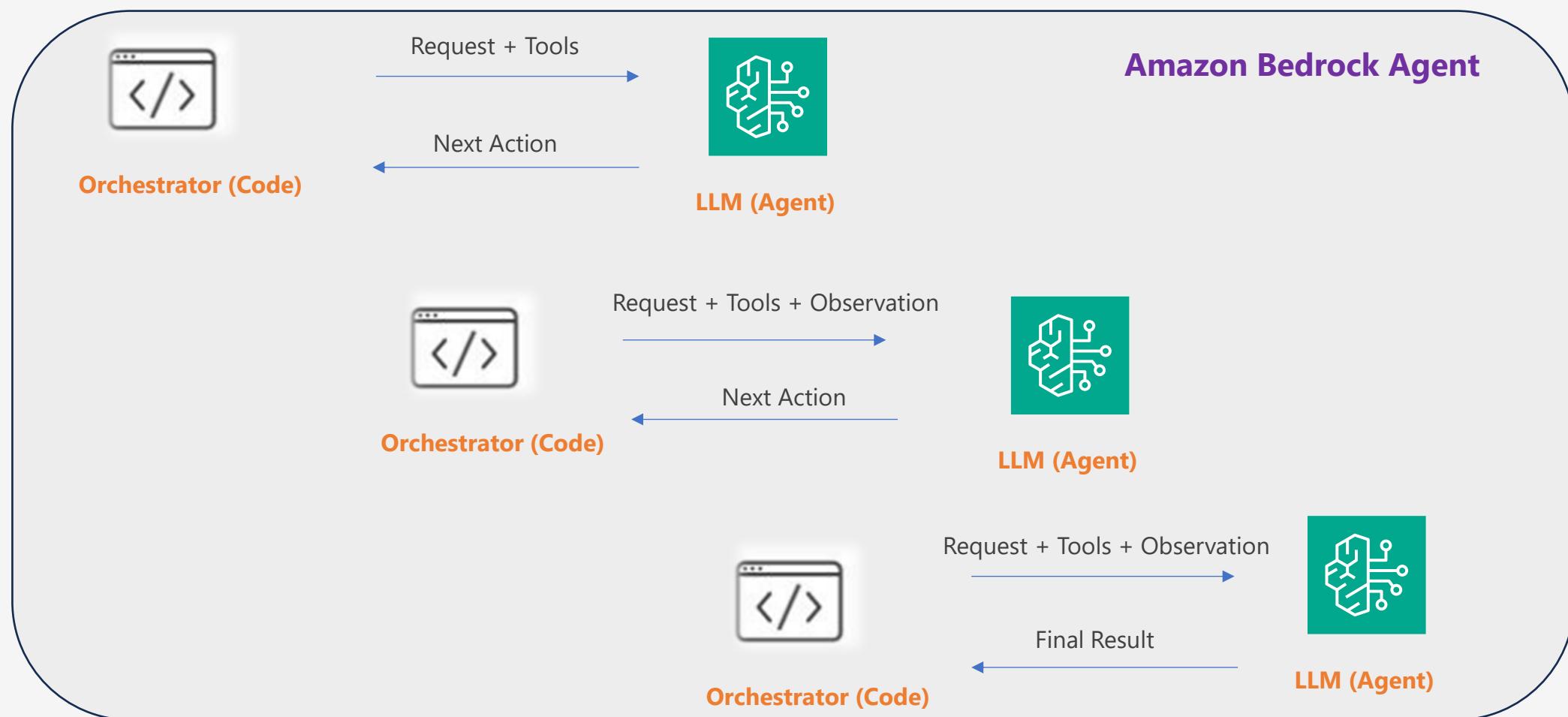
Replan

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

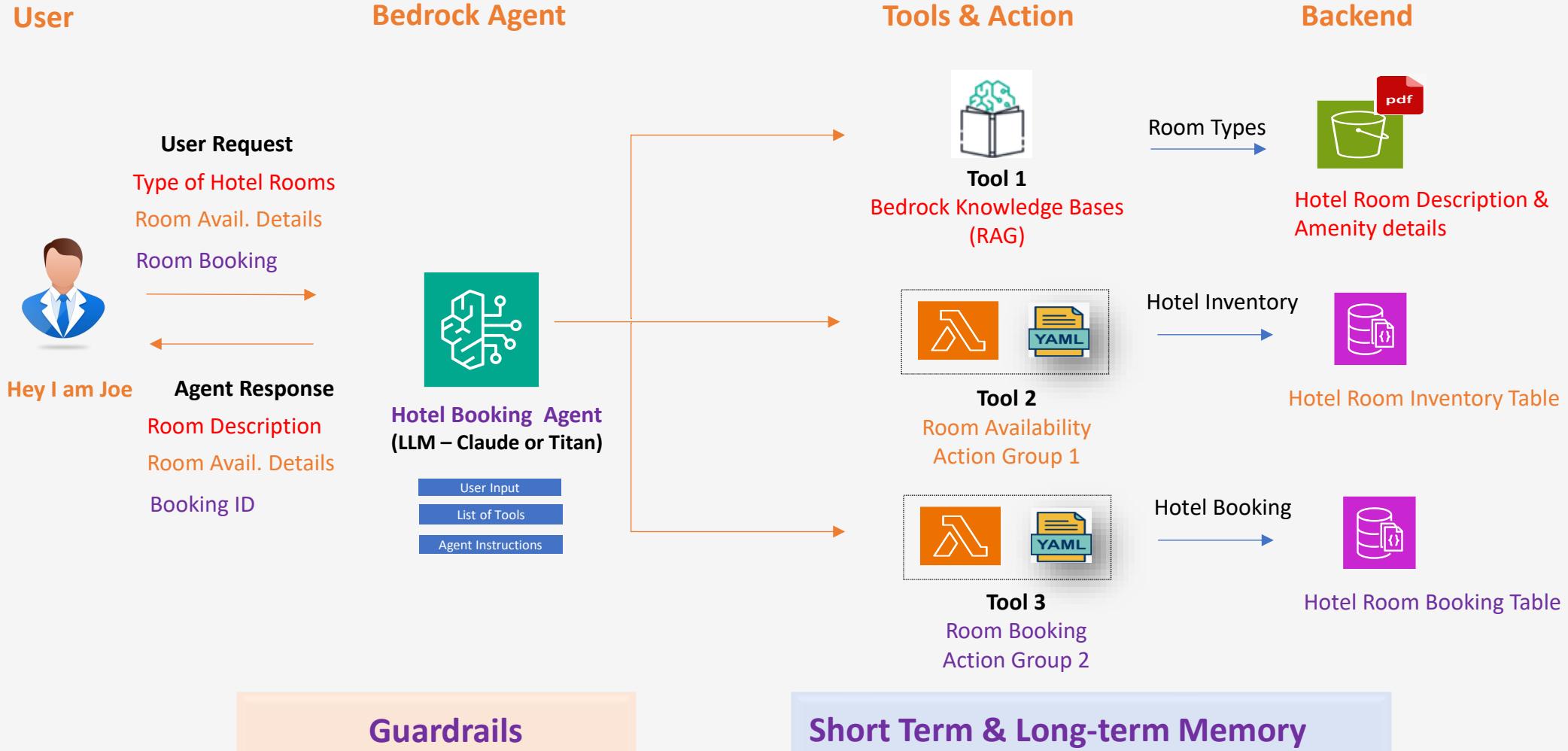
2. Book a hotel room for me **with employee id - 251205 if my leave balance is more than 3 days** : 1. Check-in date - 2025-12-26, 2. Room Type - Sea View, 3. Guest Name – Jane Doe, 4. Number of Nights - 3

Amazon Bedrock Agents – ReAct Orchestration Style

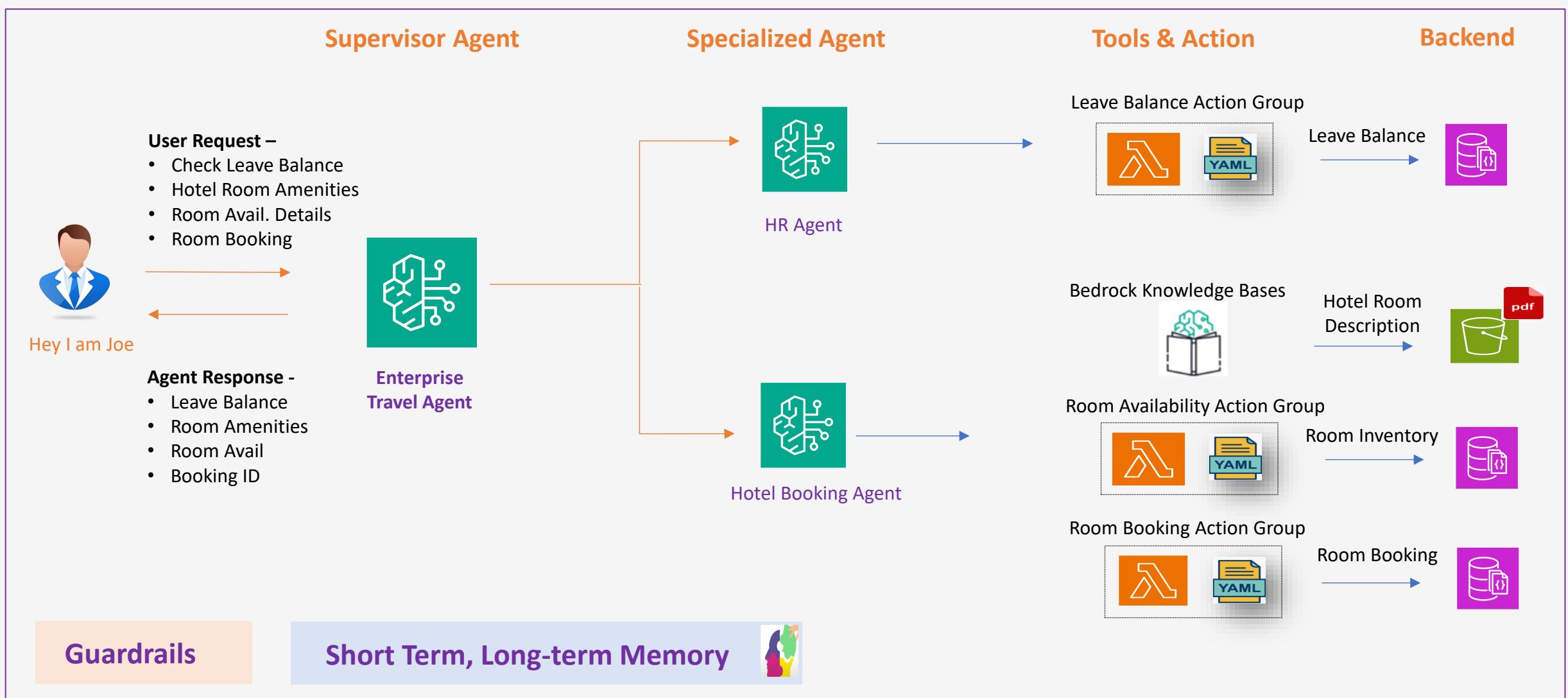


- Orchestrator refers to internal code
- ReAct – Orchestrator style or prompt style - Defines the plan and then modify as we go...but **Sequential in nature**

Hotel Booking Agent – Single Agent



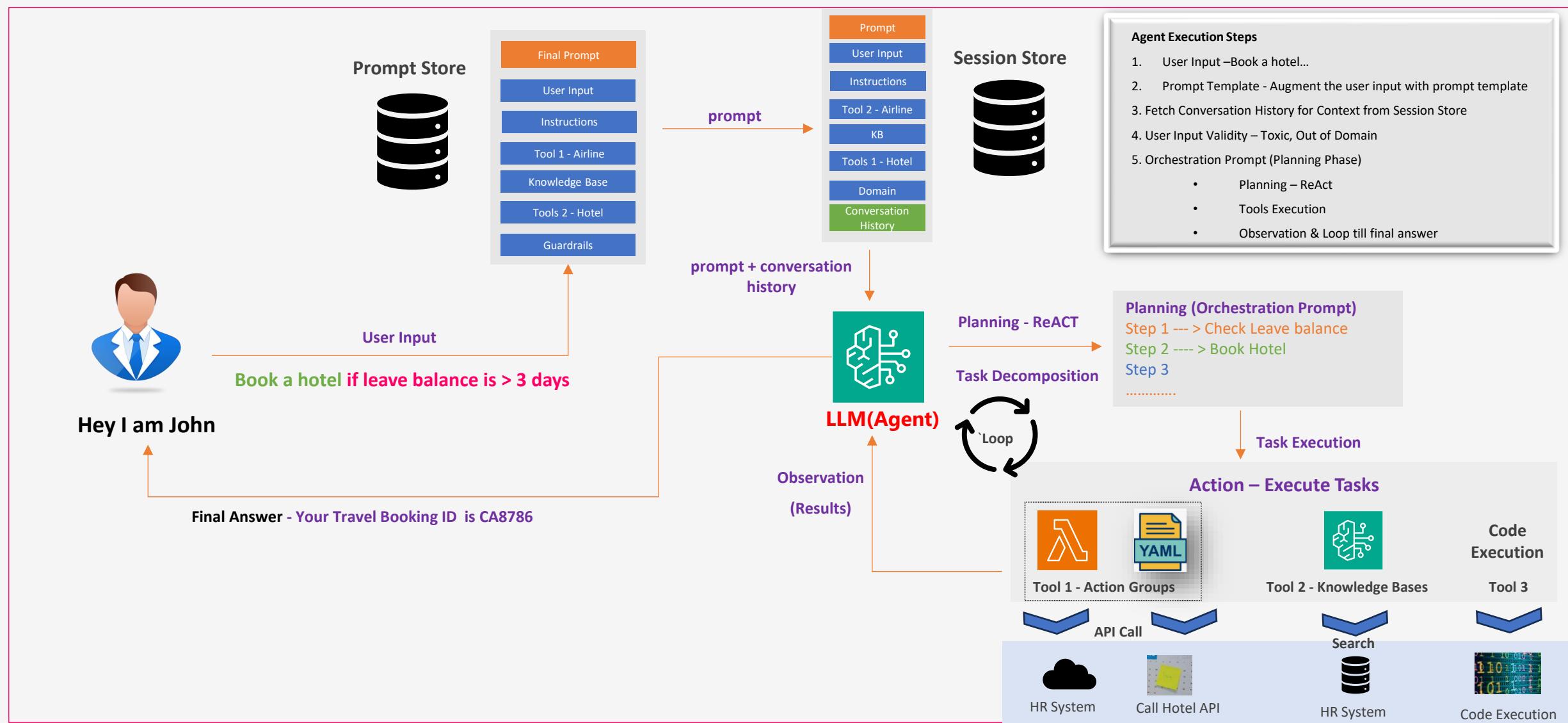
Enterprise Travel Agent (Multi-Agent)



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

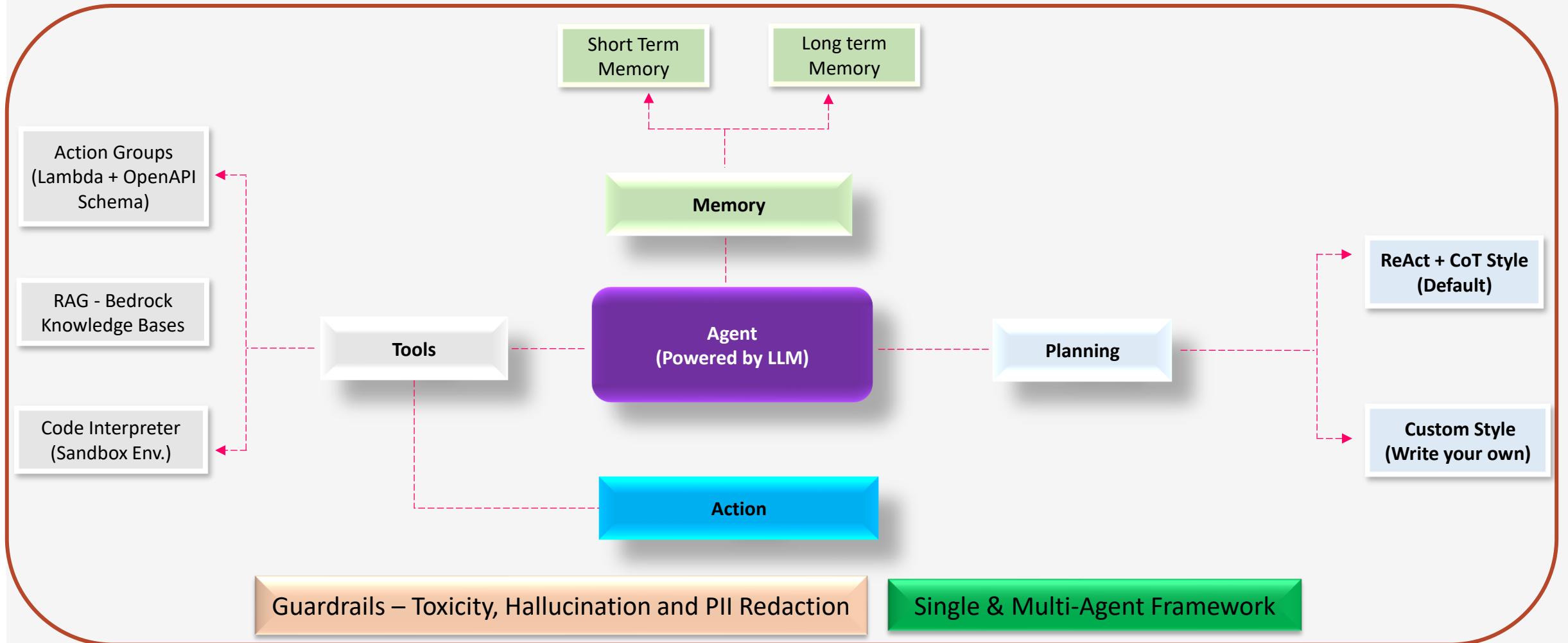
Step 1 : Check my Leave balance
Execute : API Call to HR System
Observation : 5 Vacation Days

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

Replan

Amazon Bedrock Agents – Key Characteristics

AWS AI Agents

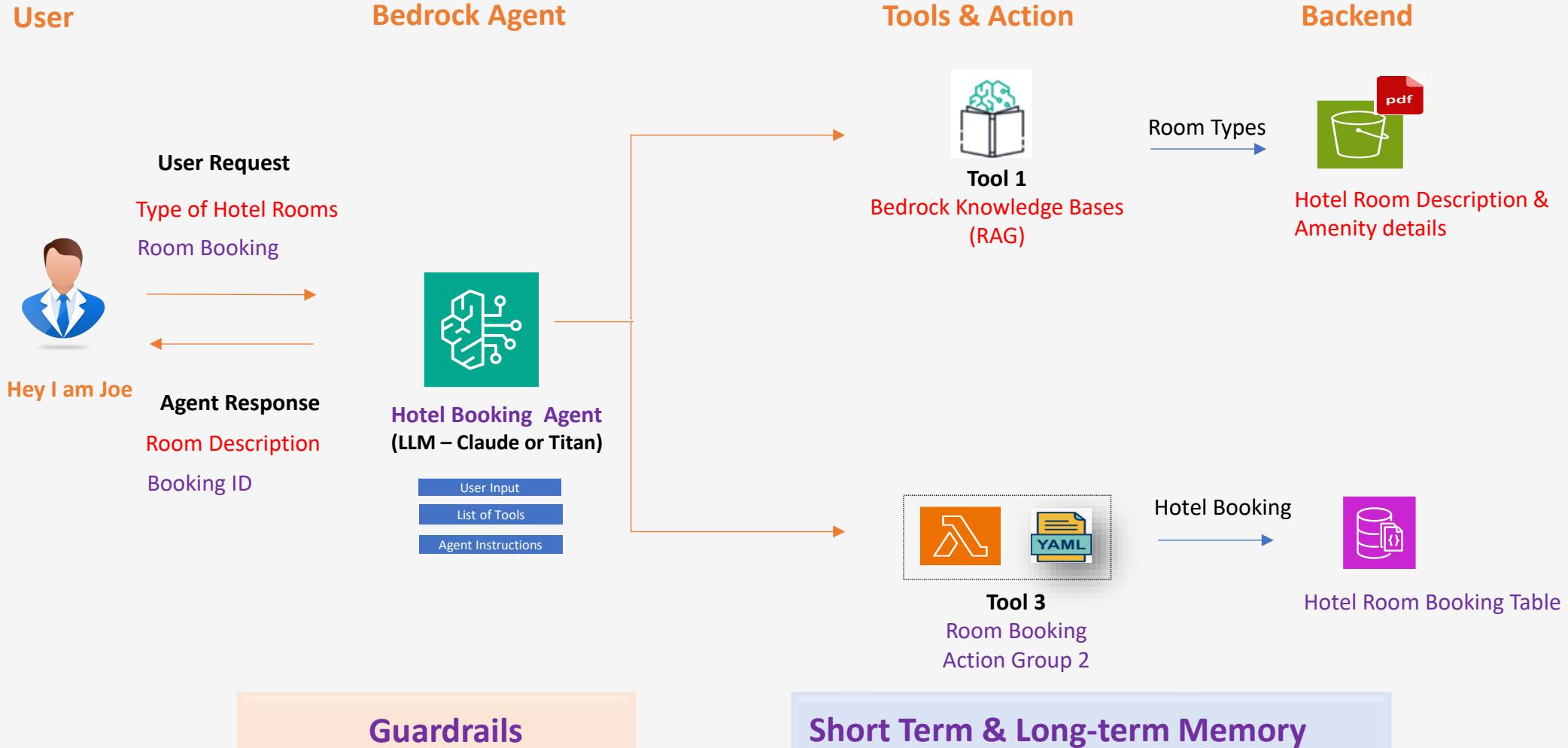


Questions

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

The screenshot shows the homepage of the Taj Fort Aguada Resort & Spa website. The top navigation bar includes links for DESTINATIONS, HOTELS, DINING, OFFERS, MEMBERSHIPS, MORE, LOGIN / JOIN, and a prominent yellow BOOK A STAY button. The main visual is a scenic view of the resort's buildings and lush greenery at sunset. Below the image, the resort's name is displayed in large, elegant white serif font: TAJ FORT AGUADA RESORT & SPA, GOA. At the bottom, there is a search/filter bar with fields for 'Taj Fort Aguada Resort', travel dates '04 Jan 2025 — 05 Jan 2025', guest count '1 Guest — 1 Room', and a dropdown for 'Special Co...'. To the right of these fields is another yellow 'BOOK NOW' button. On the far right, there is a 'GALLERY' section featuring four small thumbnail images and a 'GALLERY' button.

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



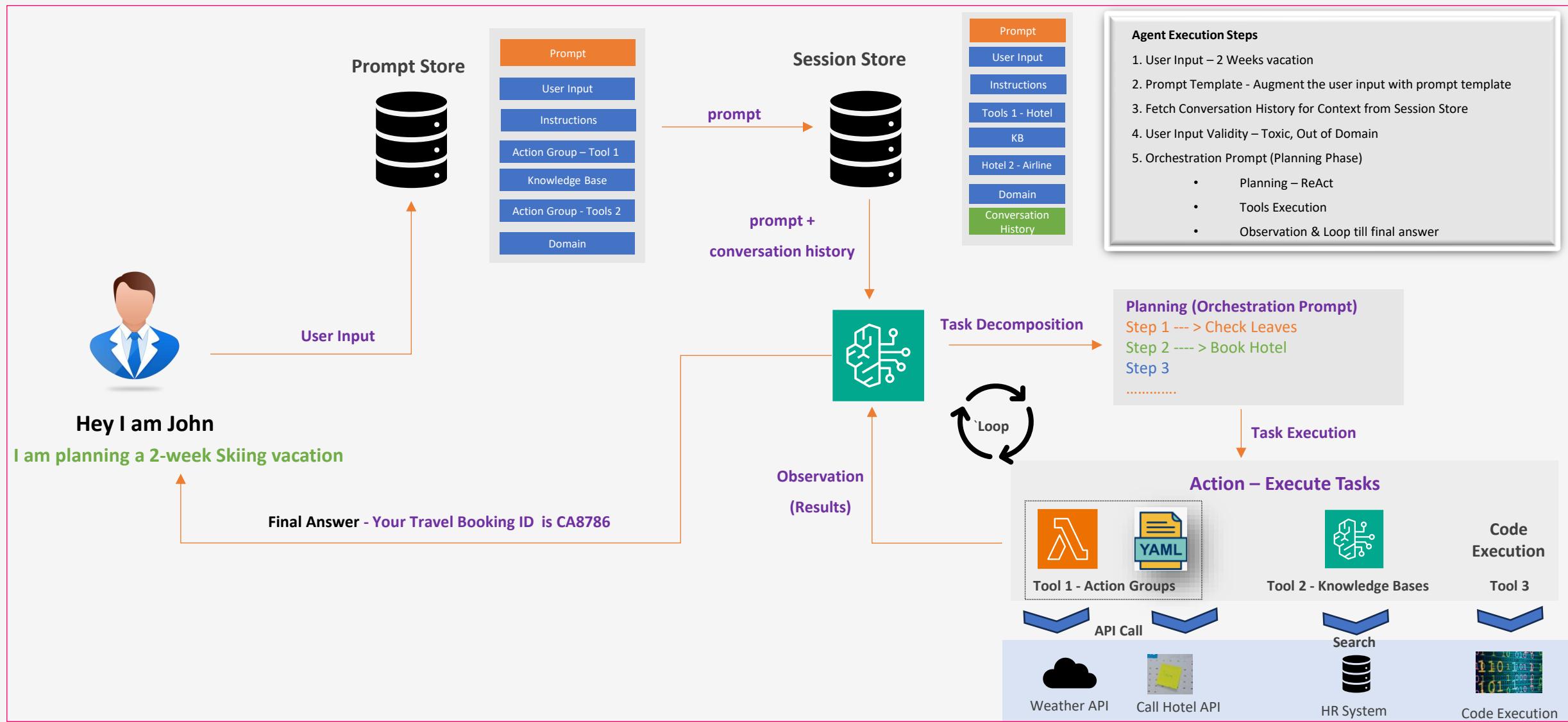
Hotel Booking Agent

Agent Response

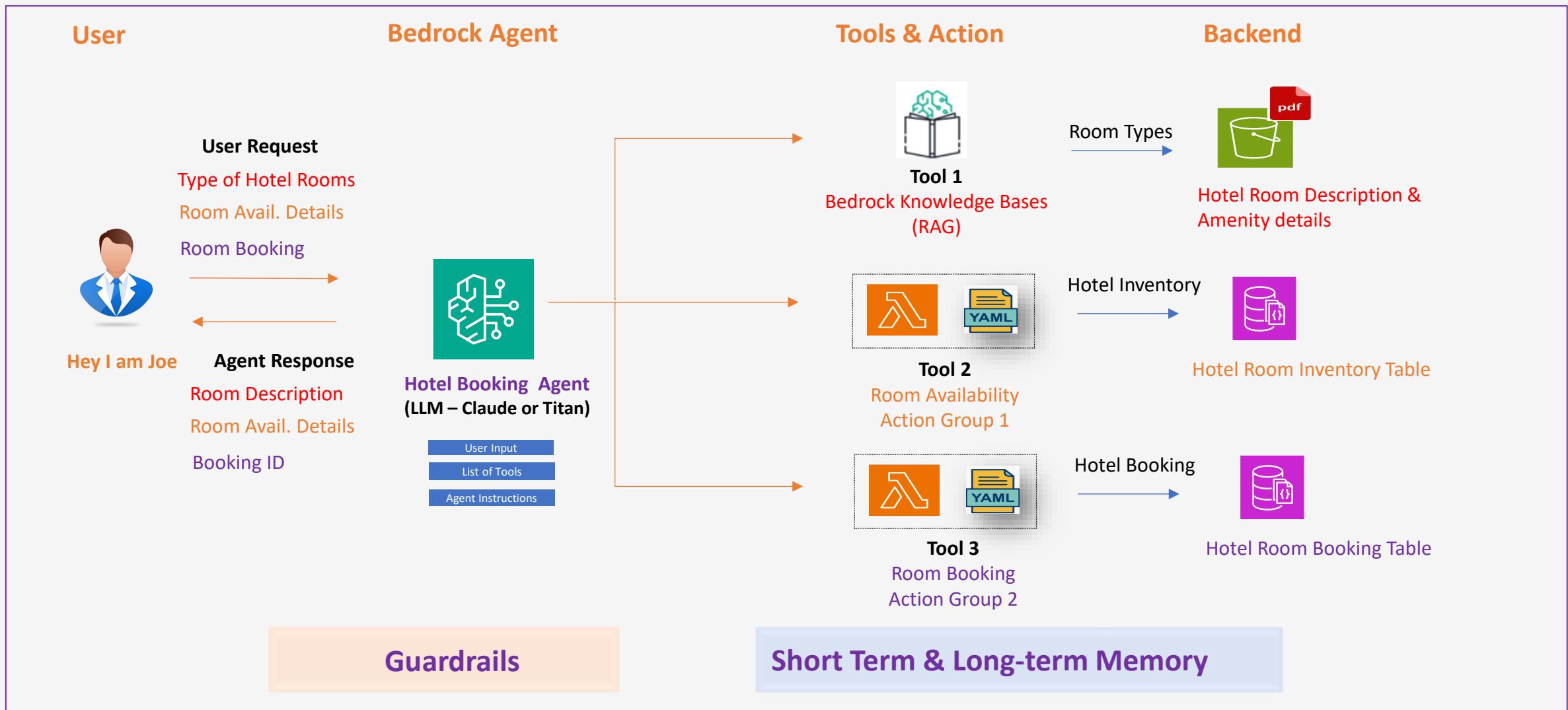
- Room Amenities
- Room Availability
- Booking ID



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 Lambda• OpenAPI 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 Lambda• OpenAPI 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 Memory• Short 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	<code>pip install -r requirements.txt</code> OR <code>pip install boto3==1.35</code> <code>pip install python-dotenv==1.5</code> <code>pip install streamlit==1.5</code> <code>pip install PyYAML==6.5</code>
Under the Services folder, Open bedrock_agent_runtime.py	<code>Update --- > agentId=</code> <code>Update --- > agentAliasId=</code>
In the .env.template file	<code>Update --- > agentId=</code> <code>Update --- > agentAliasId=</code>
In app.py file	Make changes to UI title
Run following command	<code>streamlit run app.py</code>

Use Case 2 :

Enterprise Travel Agent

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

Objective :

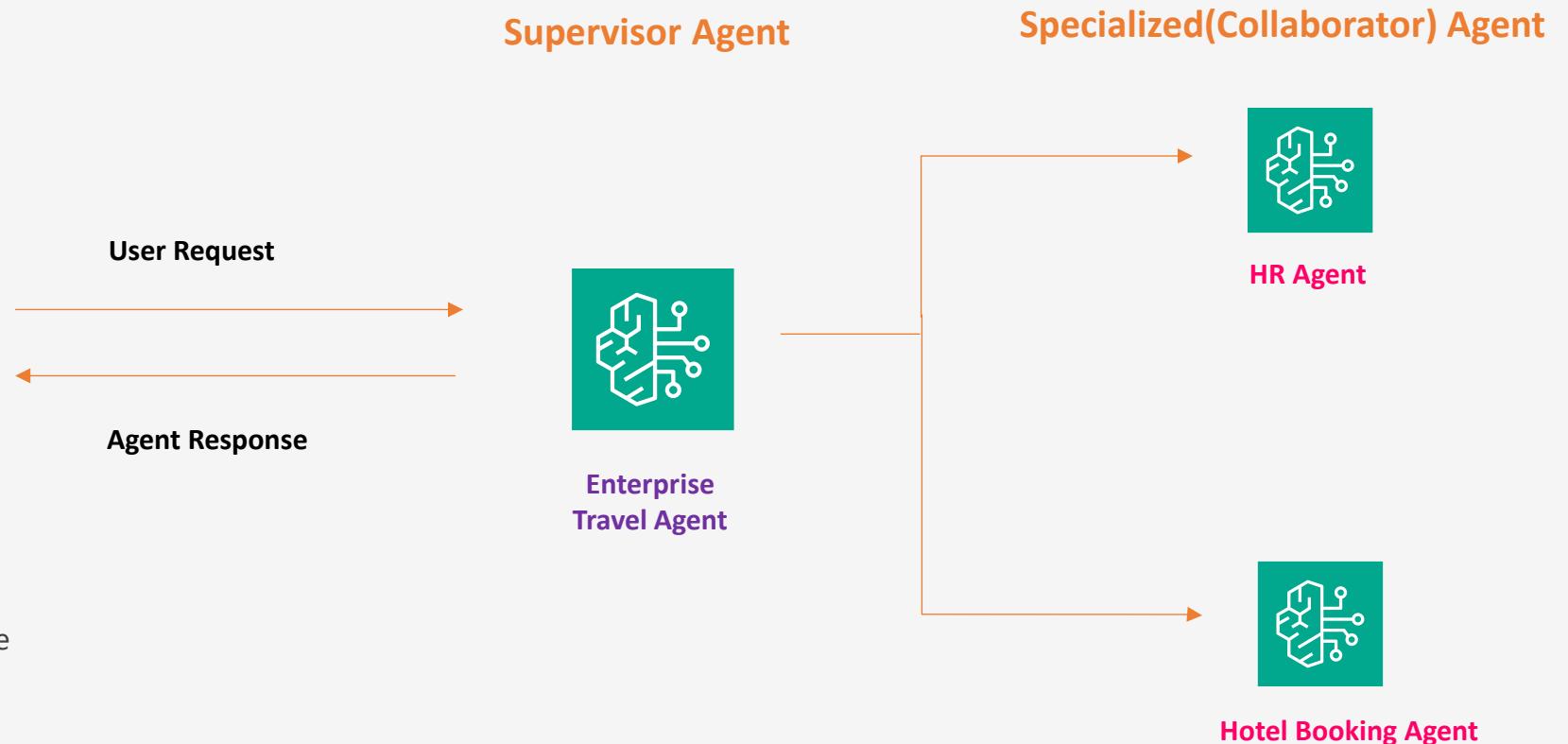
Improve the employee satisfaction by providing them an Enterprise Travel Planner Agent, an autonomous one stop shop for vacation planning with capabilities such as Leave management , Travel Booking and others.



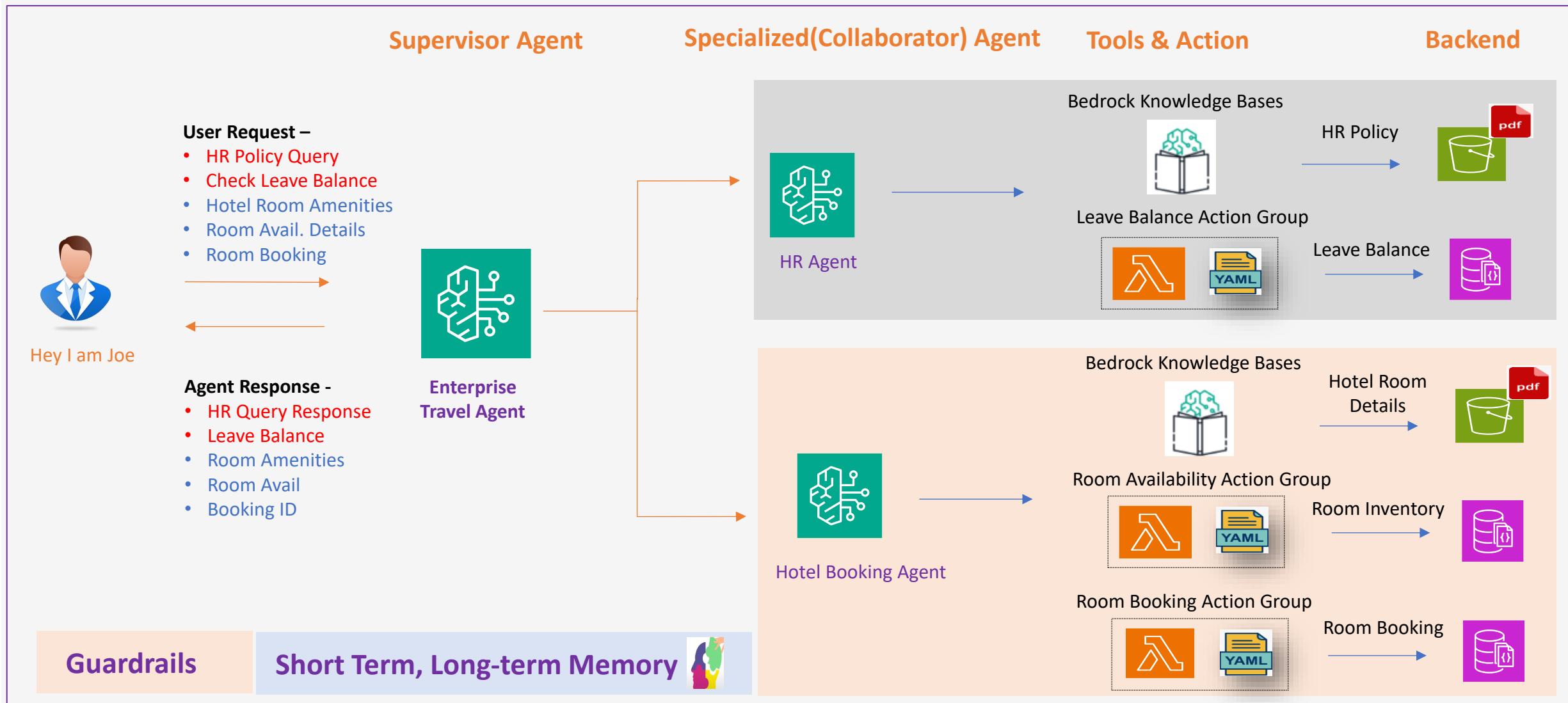
Hey I am Joe

User Request

If my leave balance is more than 5 days, book a hotel room in NY for the duration.



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 : <i>Supervisor Agent</i>	Enterprise_Travel_Agent	Manages the Routing and Communication between Specialized Agents
Agent 2 : <i>Collaborator (Specialized) Agent</i>	Hotel_Room_Booking_Agent	Manages Hotel Booking
Agent 3: <i>Collaborator (Specialized) Agent</i>	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	leaveBalanceHRTTable	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 Lambda• OpenAPI Schema	leaveBalanceHRFuction	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		
empID (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 : CrewAI

Introduction and Basic Concepts

What is CrewAI

The screenshot shows the CrewAI website homepage. At the top is a navigation bar with links: Home (red), Enterprise, Open Source, Ecosystem, Case Studies, Templates, Use Cases, and Blog. To the right are 'Log in' and 'Start Enterprise Trial' buttons. Below the navigation is a large heading: 'The Leading Multi-Agent Platform'. A subtext below it reads: 'Streamline workflows across industries with powerful AI agents. Build and deploy automated workflows using any LLM and cloud platform.' There are two 'Start Free Trial' buttons (one red, one white). A large red text '500,000,000+' indicates the number of crews. Below that is the text 'Multi-Agent Crews run using CrewAI'. To the right is a 4x4 grid of colored squares, each containing a logo: GPT-4 (pink), IBM (grey), SAP (orange), LangChain/LangGraph (red); Hugging Face (teal), Google Cloud (white), Microsoft Teams (purple), AutoGen (light blue); AI (orange), Box (blue), Salesforce (light blue), Vertex AI (light green); and AWS (yellow), SAP (light blue), AWS (yellow), SAP (light blue).

Key Agentic AI Platform

Open Source

- Crew AI
- LangChain/LangGraph

AWS

- Bedrock Agents

Microsoft

- AutoGen

Google

- Vertex AI

CrewAI is a lean, lightning-fast Python framework built entirely from scratch—completely independent of LangChain or other agent frameworks.

Use Case – Agentic AI App using CrewAI

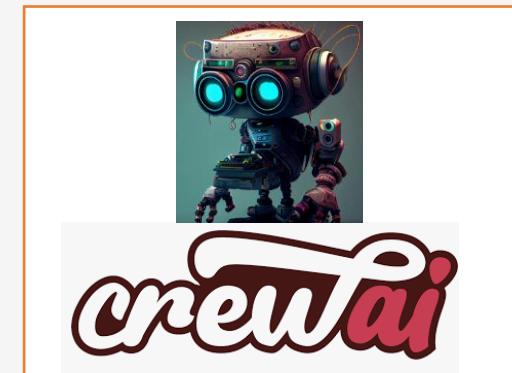
User



Hey I am John

I am planning a vacation to London.
Please provide me all the details
related to my vacation destination
including history, tourist spots etc.

Vacation Planner Agentic AI App

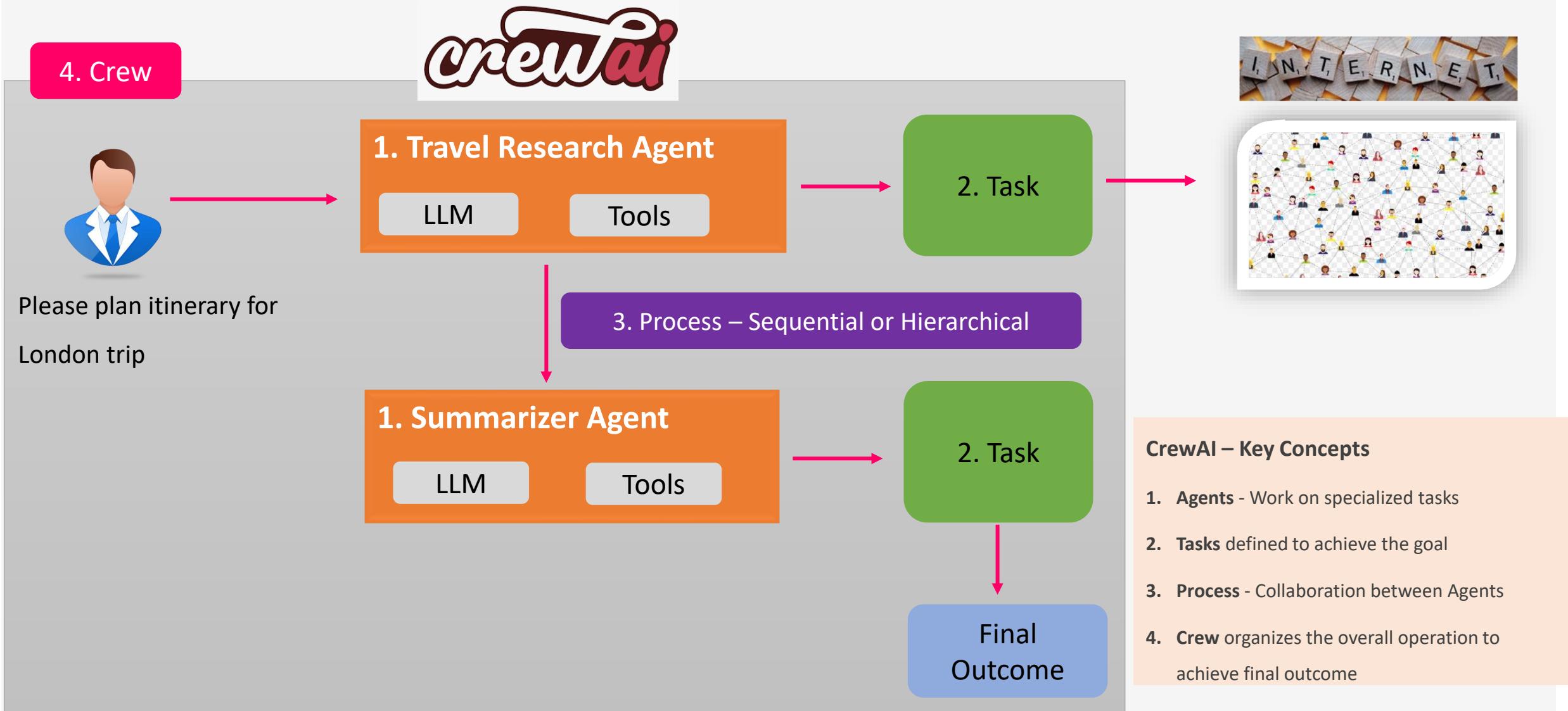


[Agentic AI App Using CrewAI](#)

Tasks

- ❖ Research internet for the vacation spot 
- ❖ Create a summary report and share with John 

Agentic App using CrewAI – Key Concepts



Agents in CrewAI

Travel Research Agent



Agent is an autonomous unit that can:

- Perform **specific tasks** ---- > Search the Web
- Use **tools** to accomplish objectives ----- > SerperDev, RAG
- **Communicate and collaborate** with other agents ----- > Summarizer Agent
- Maintain **memory** of interactions ---- > Previous Interaction (visit Museum)

Agent Attributes

Travel Research Agent



YAML Configuration

- **Role** – Agents Specialized Function --- > Travel Research Agent
- **Goal** – Agent Purpose and Motivation --- > Search Web for popular tourist destinations
- **Backstory** – Agents Experience and Perspective ---- > Spent 10 years as a travel agent helping travelers plan their vacation
- LLM – Bedrock (Defaults to OpenAI)
- Tool – Search Web
- Memory
- Verbose

Task in CrewAI

Research Task



- Tasks represent the **concrete work that agents will perform, with detailed instructions and expected outputs.**
- A **research task** for gathering comprehensive information from internet on tourist spots and history about London
- **Task Attributes**
 - **Description** ----> Concise statement of what the task is about
 - **Expected Output** -----> What the output will look like on task completion
 - Name
 - Agent
 - Tools

Processes in CrewAI



Processes **orchestrate the execution of tasks by agents**

- Sequential
- Hierarchical
- Consensual Process (Planned)

4. Crew in CrewAI

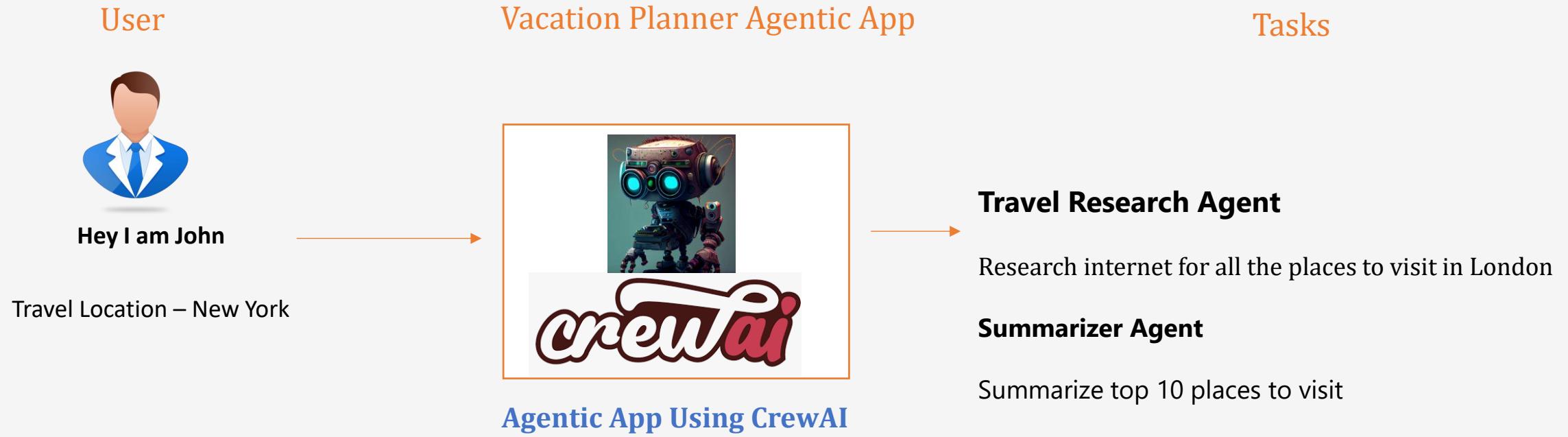


A crew represents a **collaborative group of agents working together to achieve a set of tasks.**

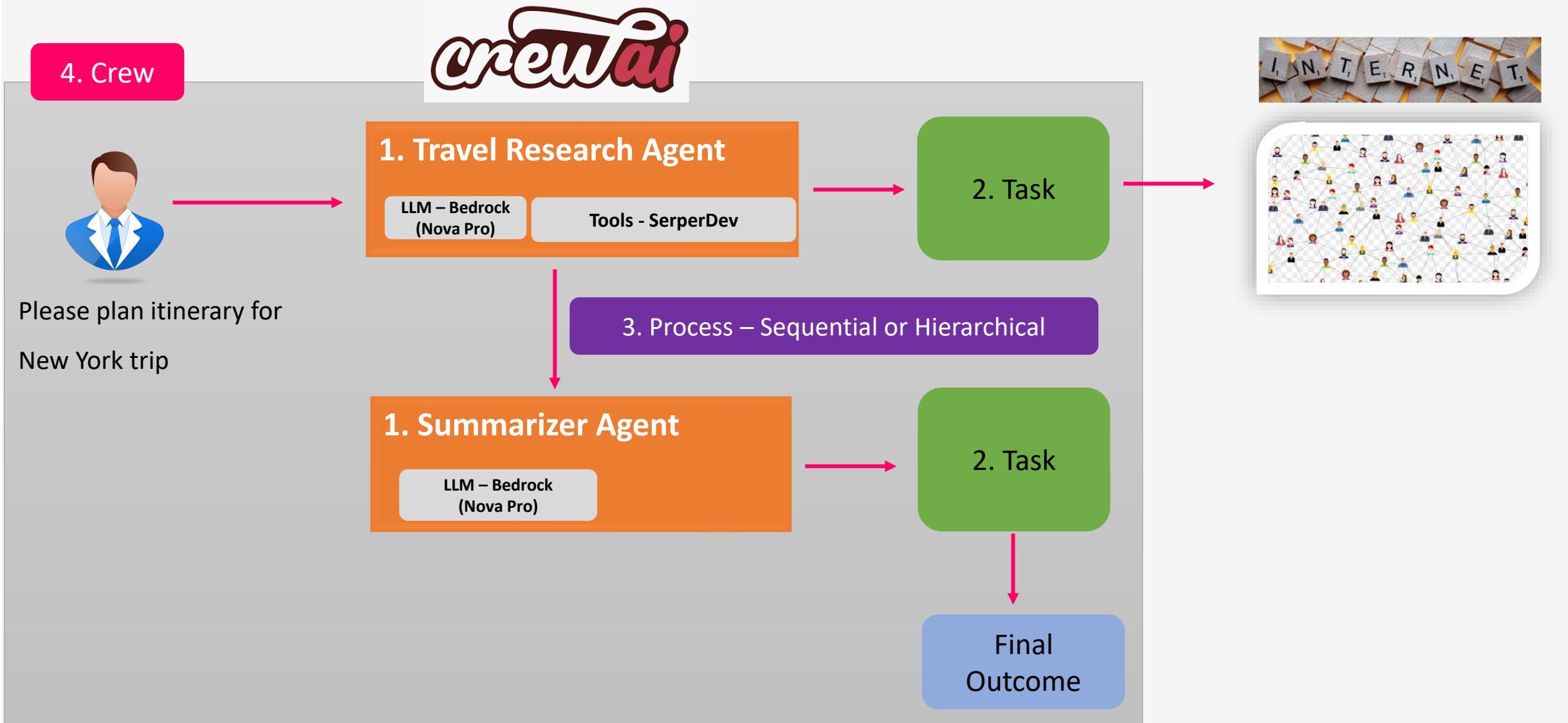
Crew Attributes

- **Agents** – Mandatory
- **Tasks** - Mandatory
- **Process** – Default (Sequential)
- **Verbose**

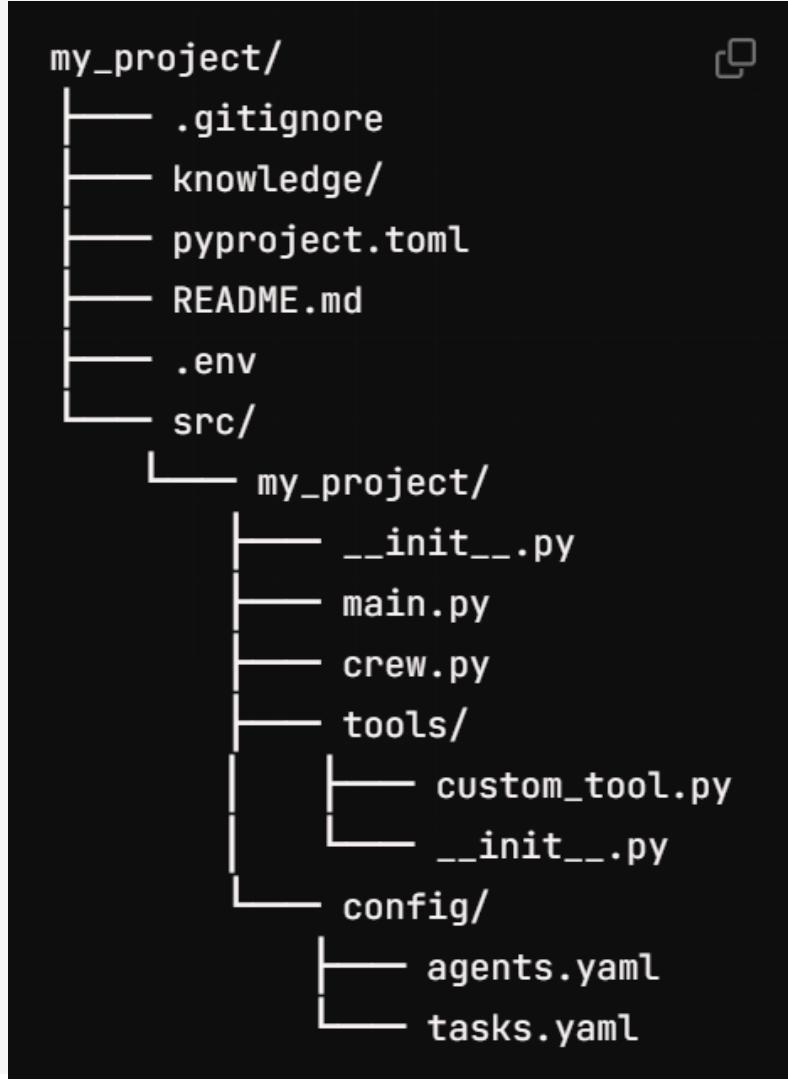
Business Use Case – Vacation Planner App



Agentic App using CrewAI – Key Concepts



CrewAI - Project Scaffolding



File	Purpose
<code>agents.yaml</code>	Define your AI agents and their roles
<code>tasks.yaml</code>	Set up agent tasks and workflows
<code>.env</code>	Store API keys and environment variables
<code>main.py</code>	Project entry point and execution flow
<code>crew.py</code>	Crew orchestration and coordination
<code>tools/</code>	Directory for custom agent tools
<code>knowledge/</code>	Directory for knowledge base

Model Context Protocol (MCP)

& Amazon Q CLI

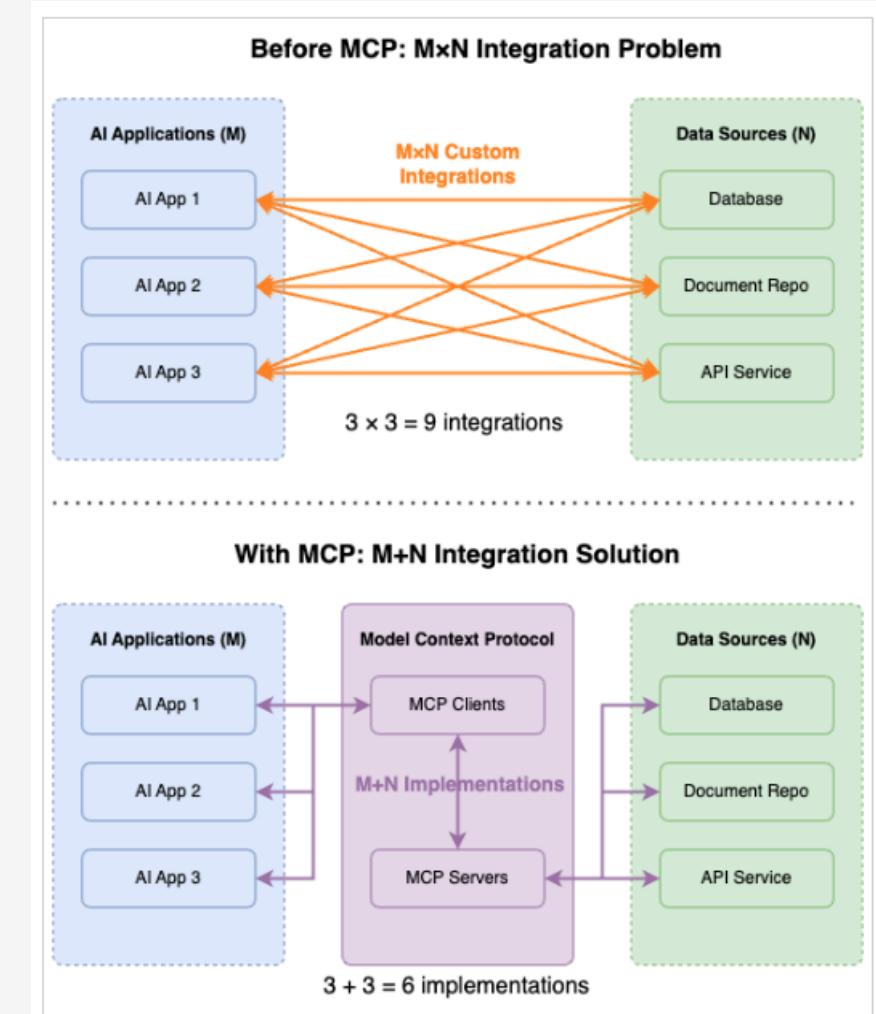
What is MCP and Problem it is trying to Solve ?

What is MCP ?

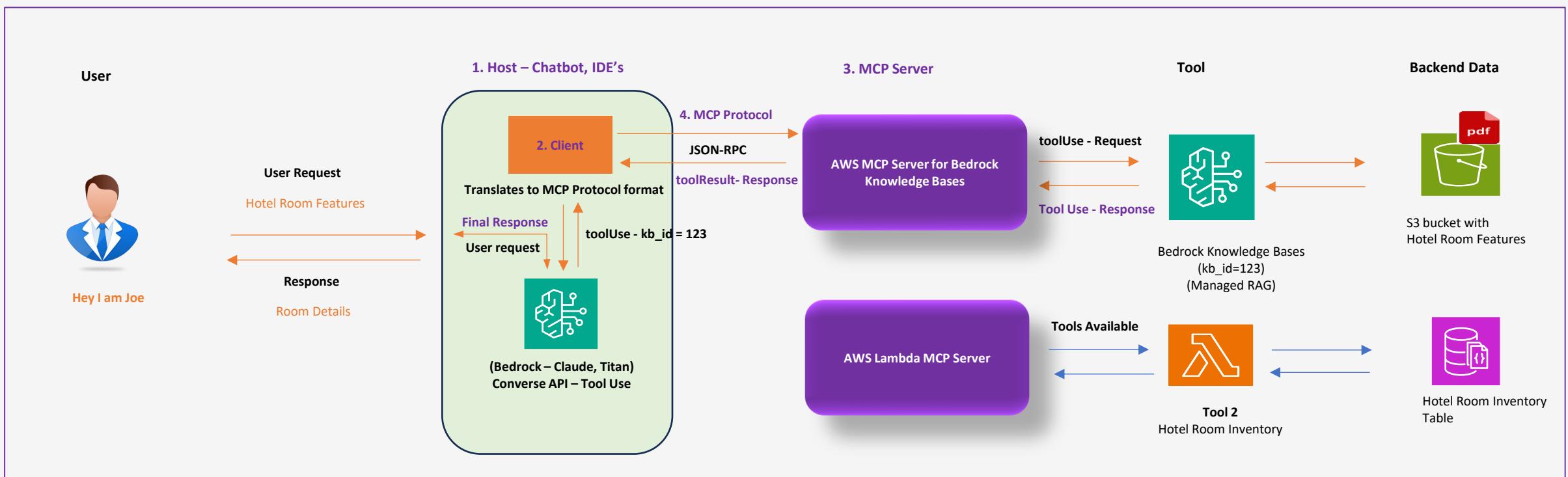
- **MCP - Model Context Protocol**
- MCP is an open protocol that standardizes how applications provide context to LLMs.

Problem MCP is trying to solve

- Solves MXN custom integration problem
- Standardizes how LLM access data from different data sources
- AWS offers 50+ MCP servers as of today.



Hotel Booking AI Application using MCP



#	MCP Components	Features	Example
1	Host	Any AI application that uses a LLM	IDE's - VSCode, Chatbot, AI Apps
2	Client	Embedded in the Host, 1:1 connections with MCP Server, Many Clients in one Host	Cline, Amazon Q CLI, Windsurf, Claude Desktop - Link
3	MCP Server	Lightweight program exposes specific capabilities in a standardized format <ul style="list-style-type: none"> Detail out the Tools available 	<ul style="list-style-type: none"> Built by Tool /Service Provider - AWS. Deployment – 1. Remote - Managed server 2. Local System or Docker
4	Transport Layer	Client - Server Architecture with Communication Flow	Data Layer - JSON/RPC ; Transport Layer - 1. STDIO(Standard Input-Output) , 2. HTTP
5	Tools, Resources & Prompts	MCP Primitives – Tools, Resources and Prompts	<ul style="list-style-type: none"> Tools – Functions models call to retrieve data. Resources – Data in Models' context such as DB data, images etc. Prompts – Template that guide how model interacts with Tools

Bedrock Converse API - Tool use (Function calling)

```
response = client.converse(
    modelId='string',
    messages=[
        {
            'role': 'user'|'assistant',
            'content': [
                {
                    'text': 'string',
                    'image': {
                        'format': 'png'||'jpeg'||'gif'||'webp',
                        'source': {
                            'bytes': b'bytes'
                        }
                    },
                    'document': {
                        'format': 'pdf'||'csv'||'doc'||'docx'||'xls'||'xlsx'||'html'||'txt'||'md',
                        'name': 'string',
                        'source': {
                            'bytes': b'bytes'
                        }
                    },
                    'toolUse': {
                        'toolUseId': 'string',
                        'name': 'string',
                        'input': {...}|[...]|123|123.4|'string'|True|None
                    },
                    'toolResult': {
                        'toolUseId': 'string',
                        'content': [
                            {
                                'json': {...}|[...]|123|123.4|'string'|True|None,
                                'text': 'string',
                            }
                        ]
                    }
                }
            ]
        }
    ]
)
```

```
# Call the Converse API with the user's query and available tools
response = bedrock.converse(
    modelId="us.anthropic.claude-3-7-sonnet-20250219-v1:0", # Specify which language model to
    messages=[{"role": "user", "content": [{"text": user_query}]}], # Format the user's message
    toolConfig={"tools": available_tools} # Pass the tool definitions to the model
)

toolConfig={
    'tools': [
        {
            'toolSpec': {
                'name': 'string',
                'description': 'string',
                'inputSchema': {
                    'json': {...}|[...]|123|123.4|'string'|True|None
                }
            }
        },
        {
            'toolChoice': {
                'auto': {},
                'any': {}
            }
        },
        {
            'tool': {
                'name': 'string'
            }
        }
    ]
}
```

Install Amazon Q CLI (Client)

Amazon Q CLI – Installation

Pre-Requisites :

- AWS Builder ID Account - Free
- Configure IAM Role for your CLI – aws configure

Q CLI Installation

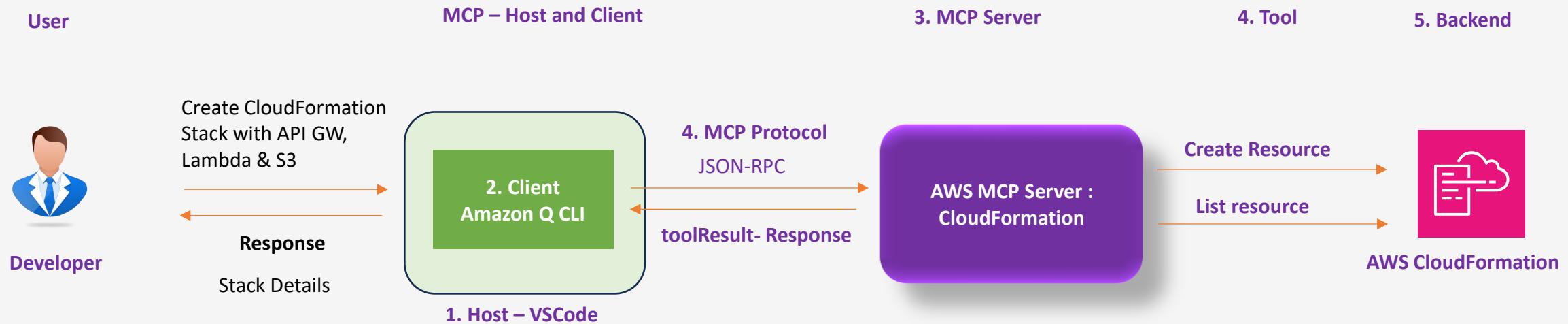
- Mac/ Linux users -- > [Installing Amazon Q for command line - Amazon Q Developer](#)
- Windows Users -- > Refer to attached Guide
- Important Commands
 - **CTRL C** – Kill Session
 - **/quit** – Quit Session
 - **q chat** – Start Chat
 - **q restart** - Restart

AWS CloudFormation MCP Server

& Amazon Q CLI (Client)

Architecture

AWS CloudFormation MCP Server - Demo



Basic Usage

Examples of how to use the AWS Infrastructure as Code MCP Server:

- "Create a new S3 bucket with versioning and encryption enabled"
- "List all EC2 instances in the production environment"
- "Update the RDS instance to increase storage to 500GB"
- "Delete unused NAT gateways in VPC-123"
- "Set up a three-tier architecture with web, app, and database layers"
- "Create a disaster recovery environment in us-east-1"
- "Configure CloudWatch alarms for all production resources"
- "Implement cross-region replication for critical S3 buckets"
- "Show me the schema for AWS::Lambda::Function"
- "Create a template for all the resources we created and modified"

Tools

`create_resource`

Creates an AWS resource using the AWS Cloud Control API with a declarative approach. **Example:** Create an S3 bucket with versioning and encryption enabled.

`get_resource #`

Gets details of a specific AWS resource using the AWS Cloud Control API. **Example:** Get the configuration of an EC2 instance.

`update_resource`

Updates an AWS resource using the AWS Cloud Control API with a declarative approach. **Example:** Update an RDS instance's storage capacity.

`delete_resource`

Deletes an AWS resource using the AWS Cloud Control API. **Example:** Remove an unused NAT gateway.

`list_resources`

Lists AWS resources of a specified type using AWS Cloud Control API. **Example:** List all EC2 instances in a region.

AWS MCP Servers

1. AWS CloudFormation MCP Server- [Welcome to AWS MCP Server](#)

2. AWS API MCP - <https://awslabs.github.io/mcp/>

Questions - Tools for AWS CloudFormation MCP Server

- **create_resource**

Create an S3 bucket with versioning and encryption enabled and bucket name – rahulmcptestserver090909

- **get_resource**

Get the configuration of an S3 Bucket – rahulmcptestserver090909

- **create_template**

Create a YAML template for AWS CloudFormation with Serverless stack - Rest AWS API Gateway, Lambda and S3

Deploy the template

- **delete_resource**

Delete bucket name – rahulmcptestserver090909

Questions - Tools for CloudWatch MCP Server

Tools for CloudWatch Logs

- describe_log_group

Tools for CloudWatch Alarms

- get_active_alarms

Tools for CloudWatch Metrics

- get_recommended_metric_alarms

Thank You

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 ?

The screenshot shows the AI21 Labs interface. At the top, there are navigation links for various AWS services: DynamoDB, CodeCommit, Elastic Container Service, ElastiCache, Route 53, RDS, EC2, Cloud9, VPC, and Cognito. Below the header, the AI21 Labs logo is on the left, followed by dropdown menus for 'AI21 Labs' and 'Jurassic-2 Mid'. A 'Start new chat' button is on the right. The main area shows a conversation where the user asks 'what is generative AI'. The AI response is provided in a purple box:

Generative AI

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.

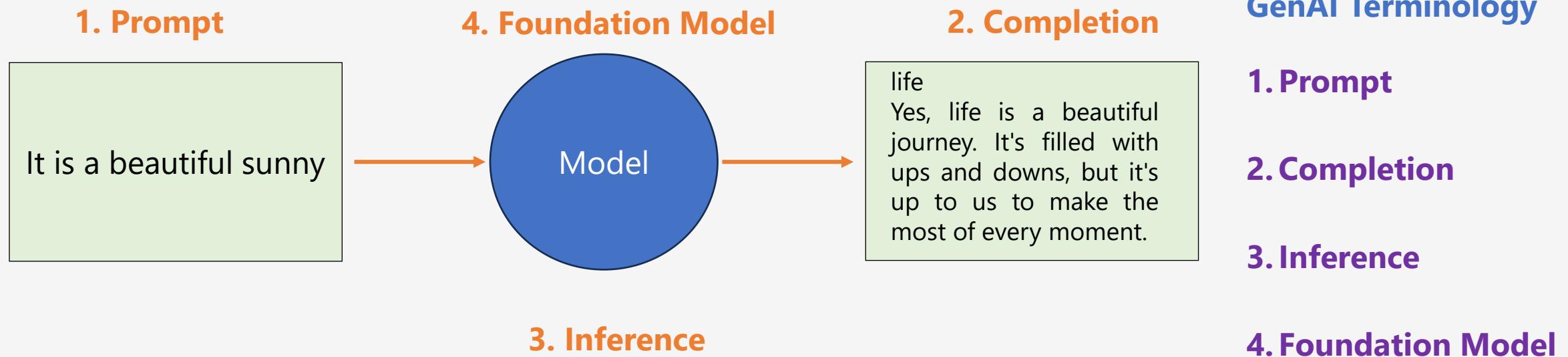
Generative models can be trained on various data types, including images, sounds, text, and even complex data structures such as DNA sequences. These models use techniques such as deep learning and reinforcement learning to generate new examples that are similar to or identical to the training examples.

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

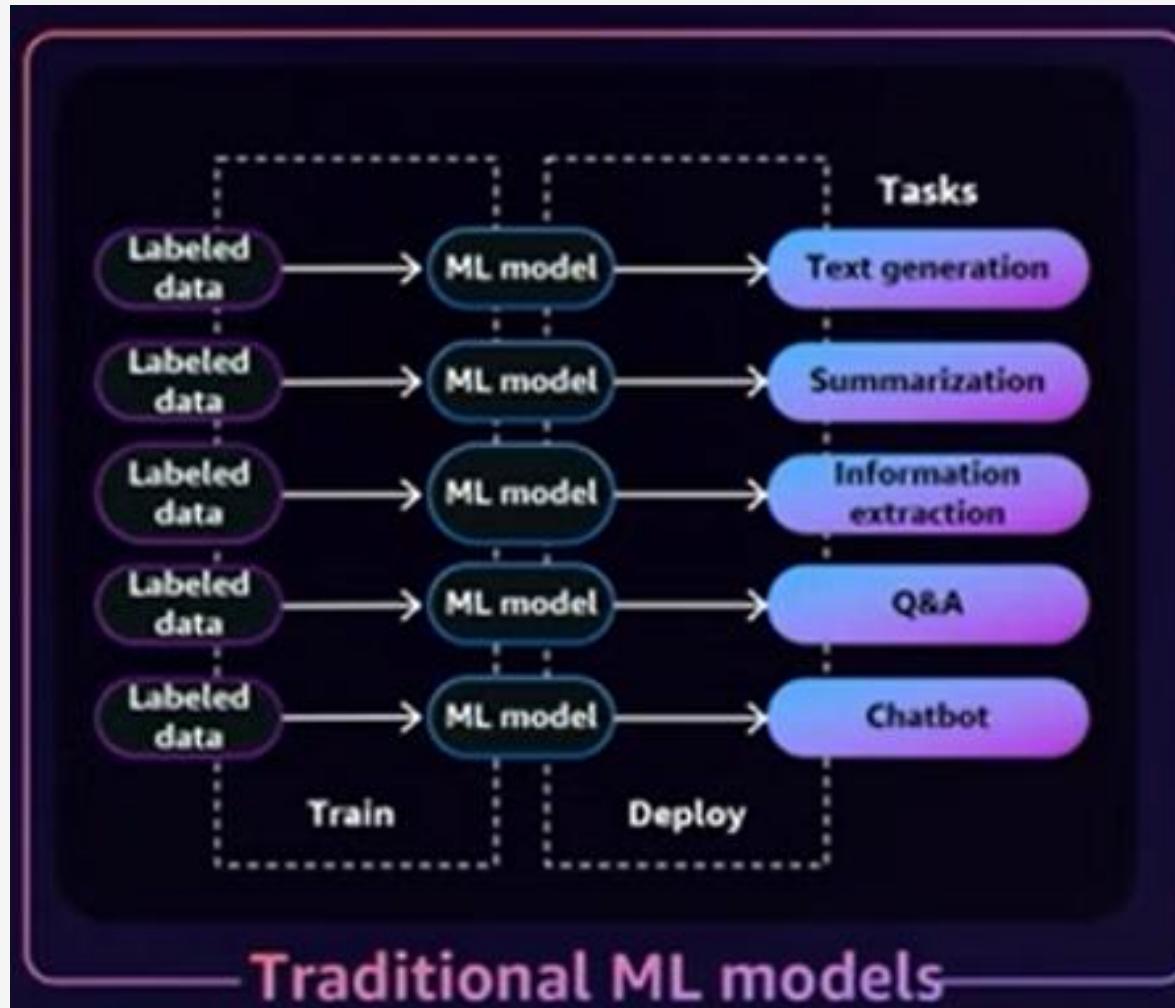
- Branch of AI that use deep-learning **models**
- Generate high-quality text, images, code based on the data they were trained on
- Popular examples of generative AI include
 - ChatGPT
 - Bard

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

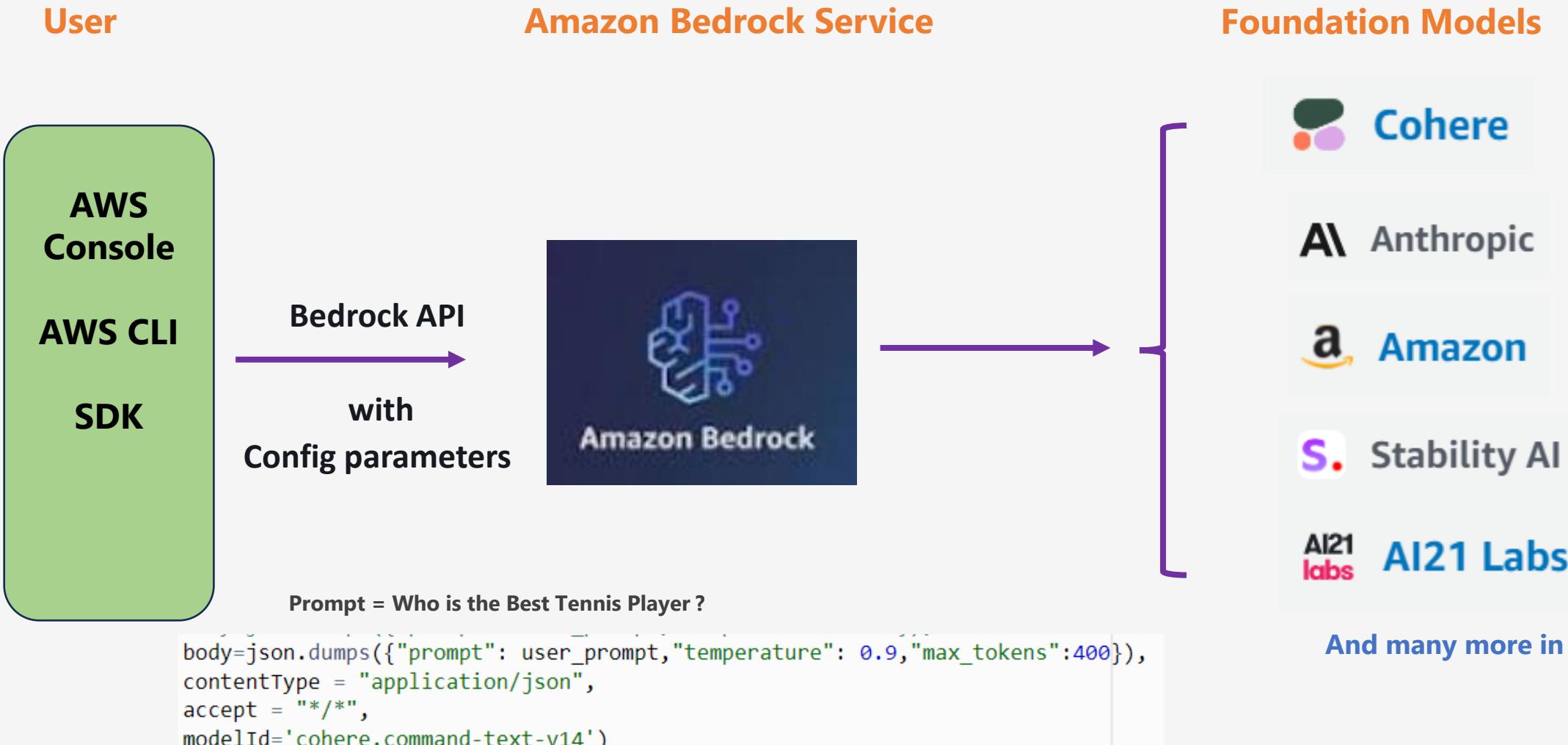


Source : AWS

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 ?

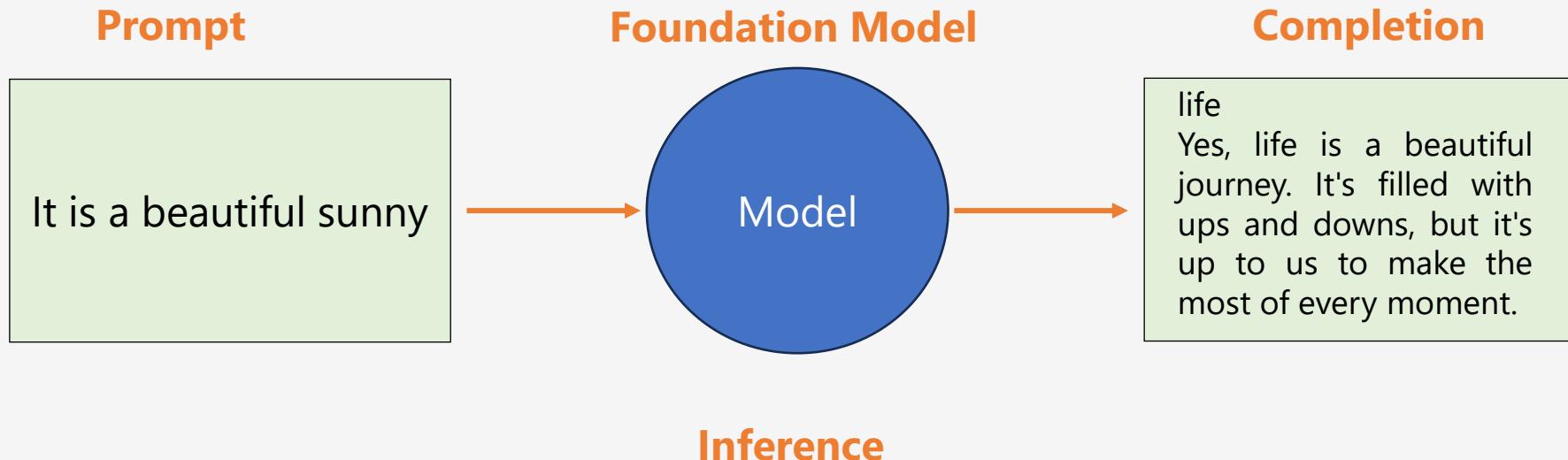


Prompt Engineering Refresher

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

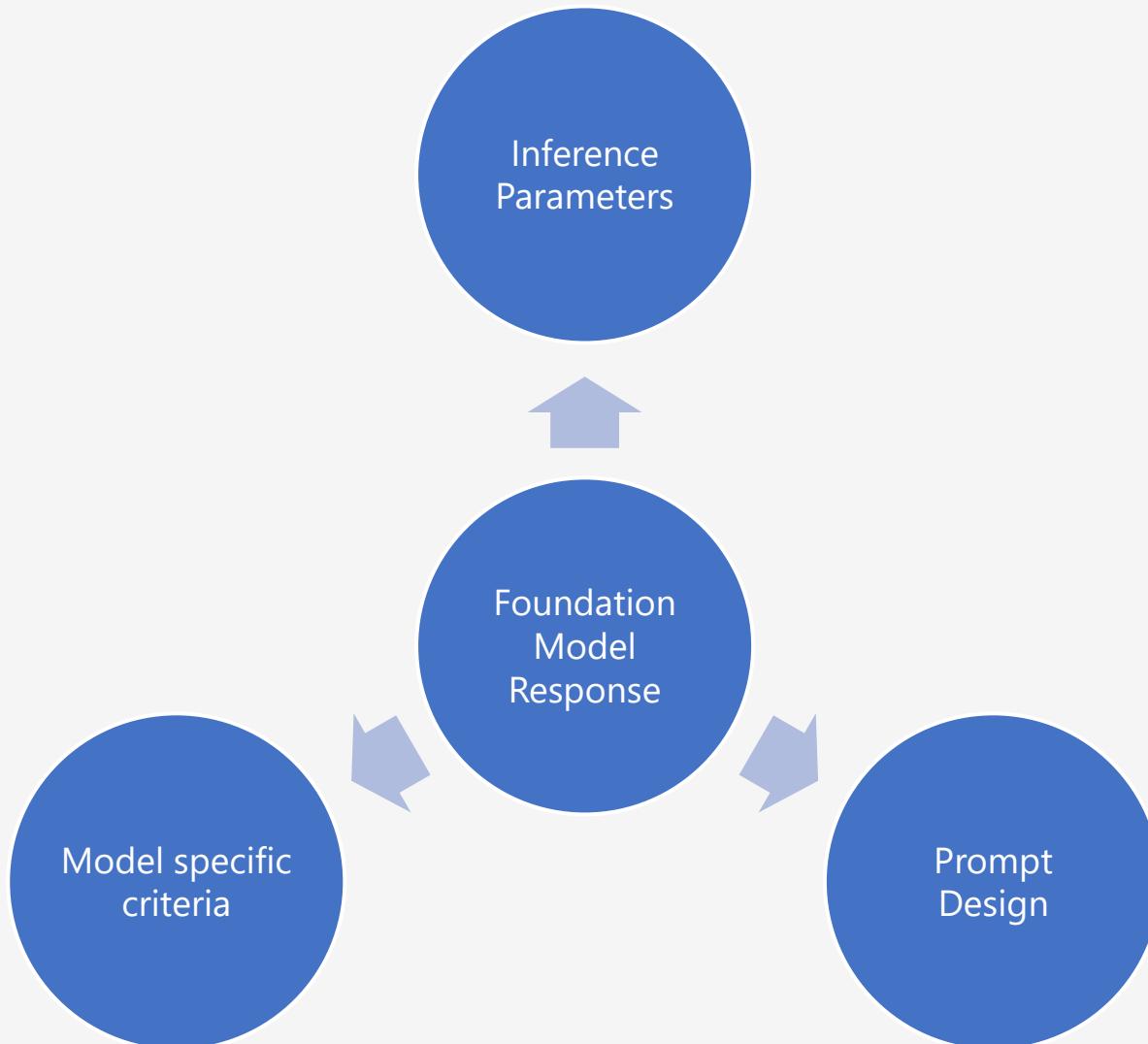
Coupon Code : <https://www.udemy.com/course/amazon-bedrock-aws-generative-ai-beginner-to-advanced/?referralCode=A02153572B8864F928E7>

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 :

"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.

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:

Contextual Information about the task

Input text for Task

1. Clear concise instructions

2. Task to be accomplished at the end

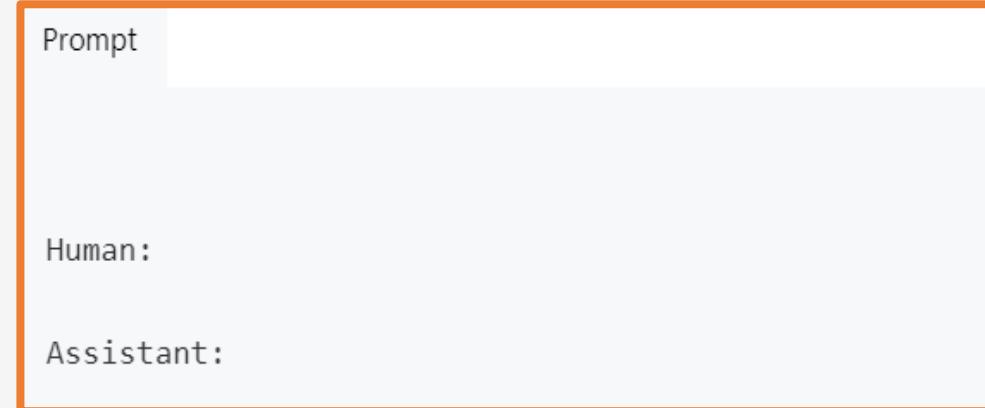
3. Output specification

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

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.



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.

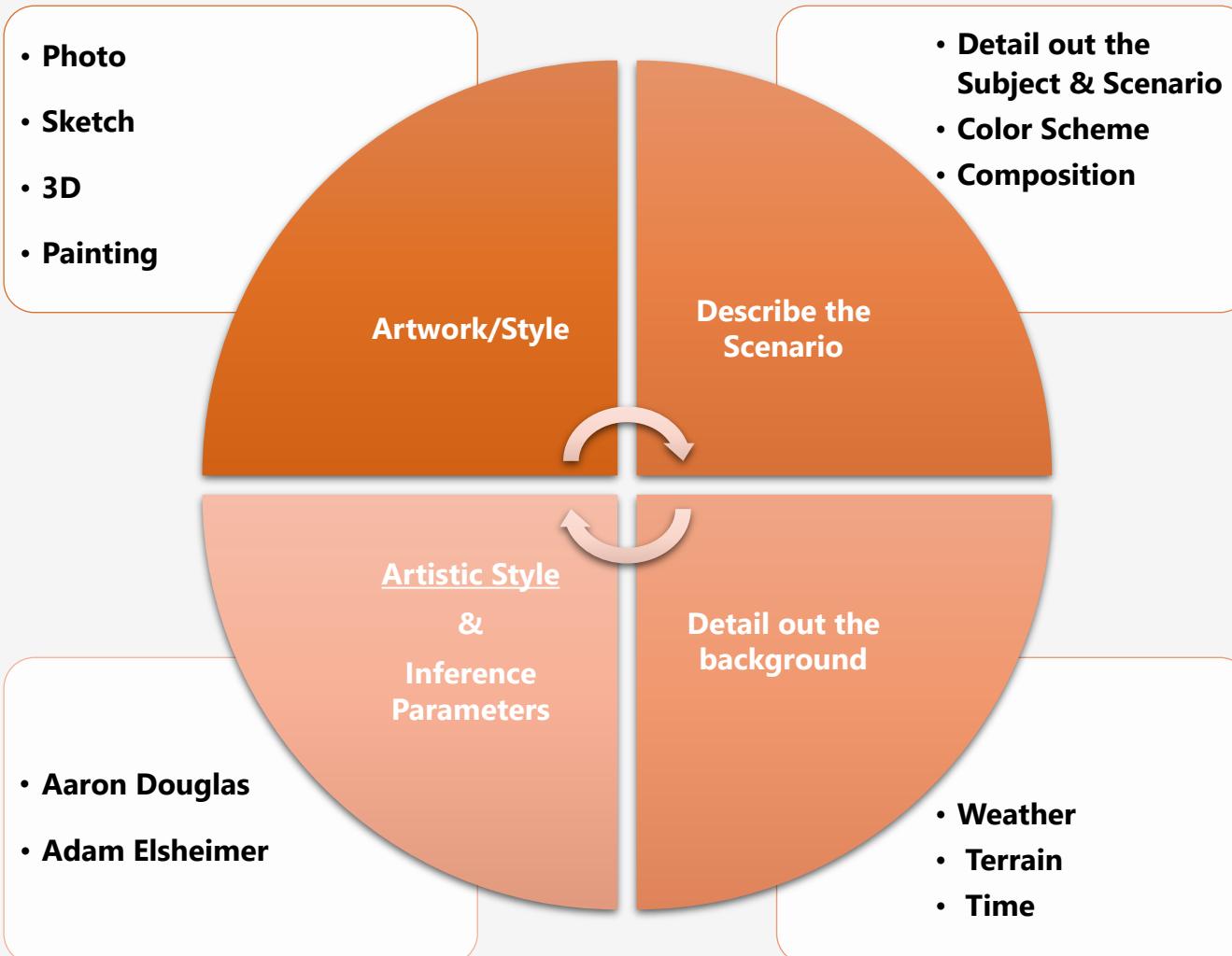
What was the heaviest hippo ever recorded?

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}}**. """"

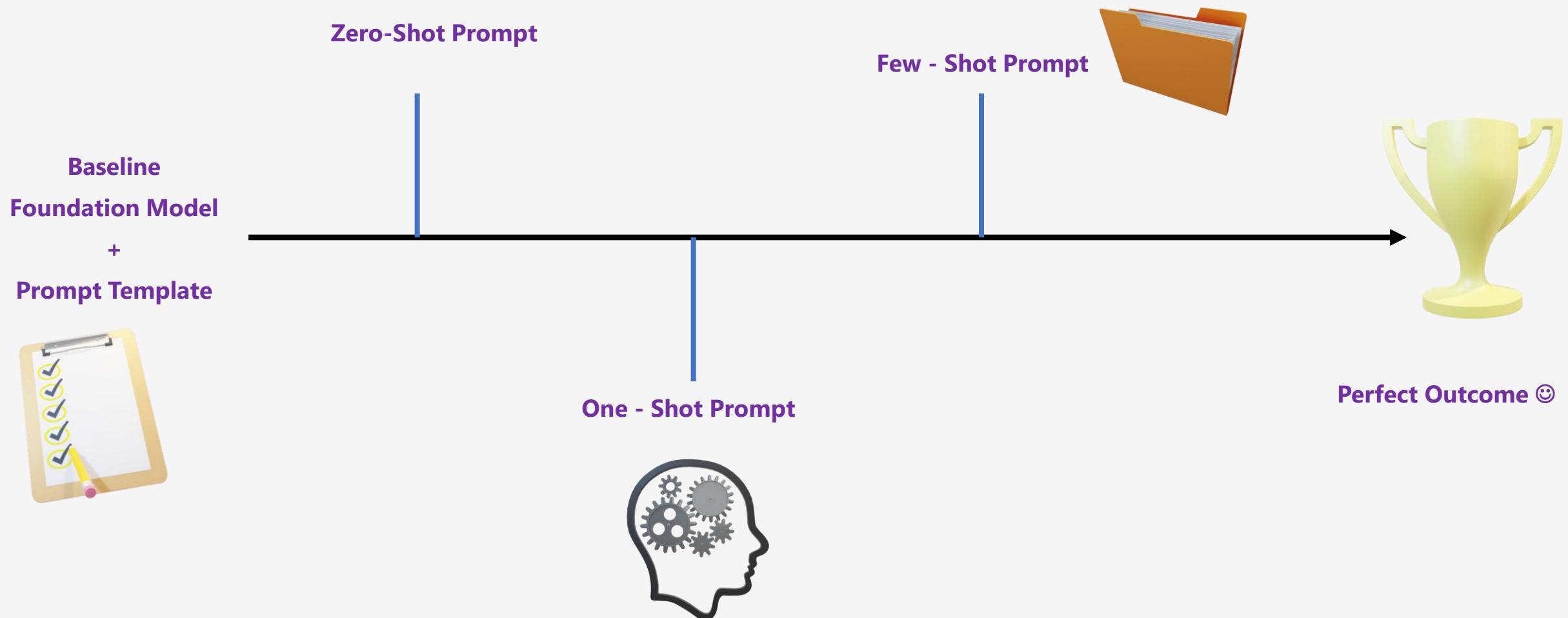
User prompt:

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 can
##

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:

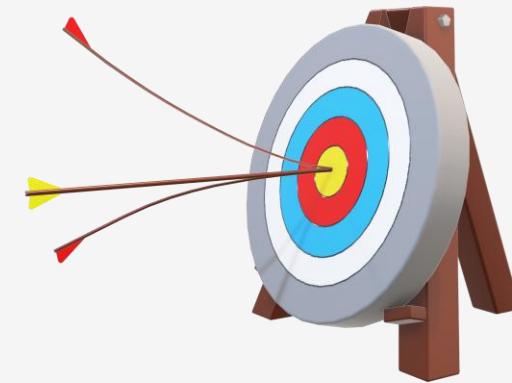
Women's Boho Beach Dress The style of midi dress with swing hem is slimming, It will make your tumm
##

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)

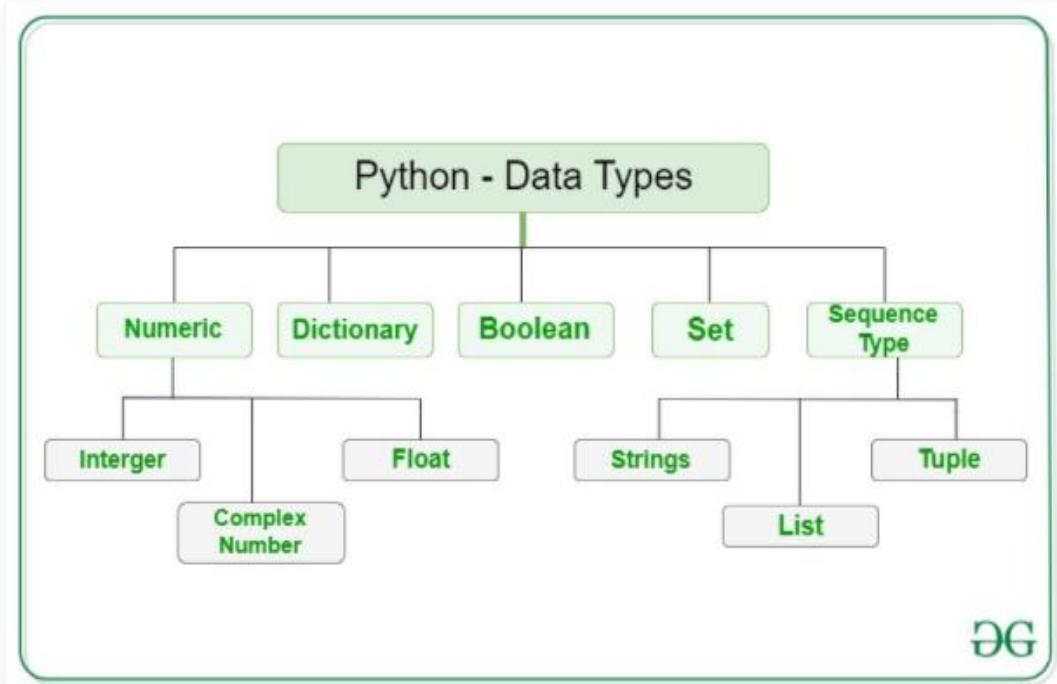
Python Basics—Refresher



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

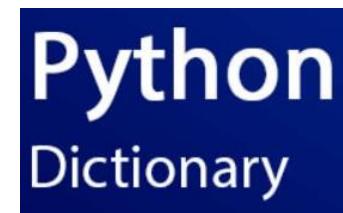


Data Type	Example
Int	<code>x = 20</code>
Float	<code>x = 20.5</code>
Dictionary	<code>x = {"name": "John", "age": 36}</code>
Strings	<code>x = "Hello World"</code>
List	<code>x = ["apple", "banana", "cherry"]</code>

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

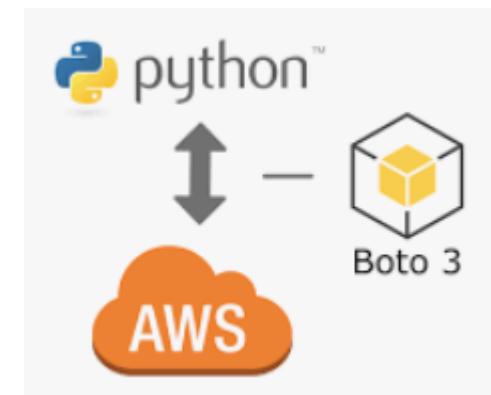


Python Basics—Refresher

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

Python
Dictionary

9. Data Type Determination

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

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

10. Function

Python Function

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

Syntax:

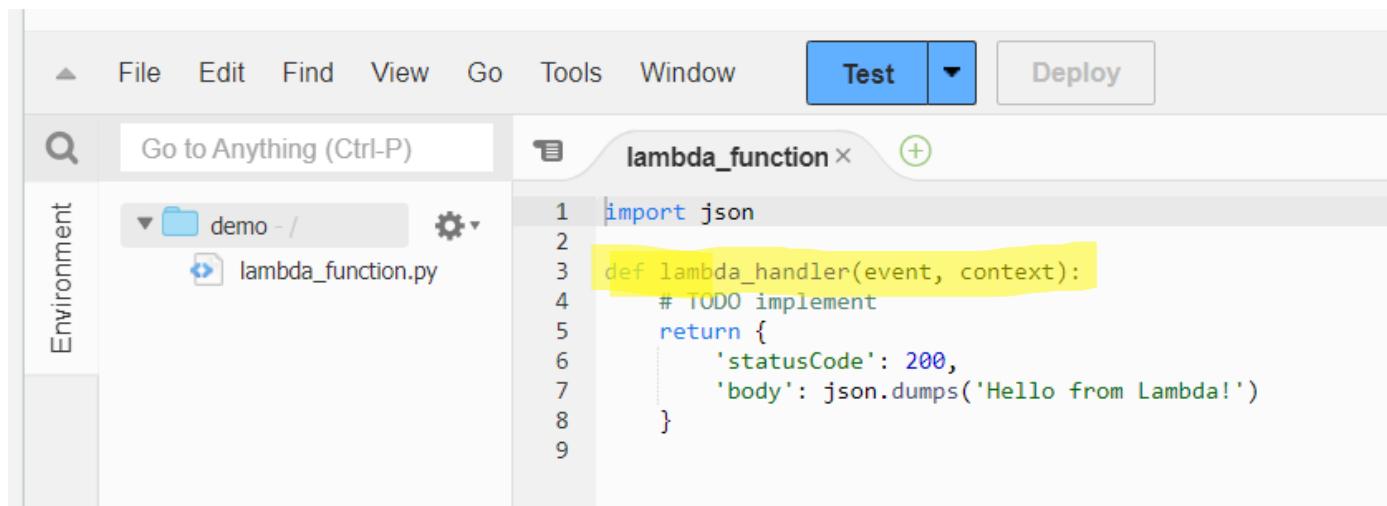
```
def function_name (argument/parameters):
```

```
    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)
```



The screenshot shows a Python code editor interface. The menu bar includes File, Edit, Find, View, Go, Tools, Window, Test (which is highlighted in blue), and Deploy. The search bar says "Go to Anything (Ctrl-P)". The left sidebar shows an "Environment" section with a "demo - /" folder containing a file named "lambda_function.py". The main code editor area has a tab titled "lambda_function" with the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

Thank You