Agents

Pavlos Protopapas

Outline

- Recap
- Motivation
- Introduction
- Components
- Overview

Outline

- **Recap**
- Motivation
- Introduction
- Components
- Overview

Recap: Vanilla LLMs

LLMs generate accurate and detailed responses based on pre-trained knowledge.

Example Use Case:

“Write a summary of Einstein's theory of relativity.”

Output (Normal LLMs):

"Einstein's theory of relativity encompasses two key components: special relativity and general relativity. Special relativity, introduced in 1905, revolutionized physics by showing that time and space are relative concepts, depending on the observer's motion, and introduced the famous equation $E=mc^2$. General relativity, proposed in 1915, describes gravity as the curvature of spacetime caused by massive objects, providing a framework to understand phenomena like black holes and the expansion of the universe."

Recap: Vanilla

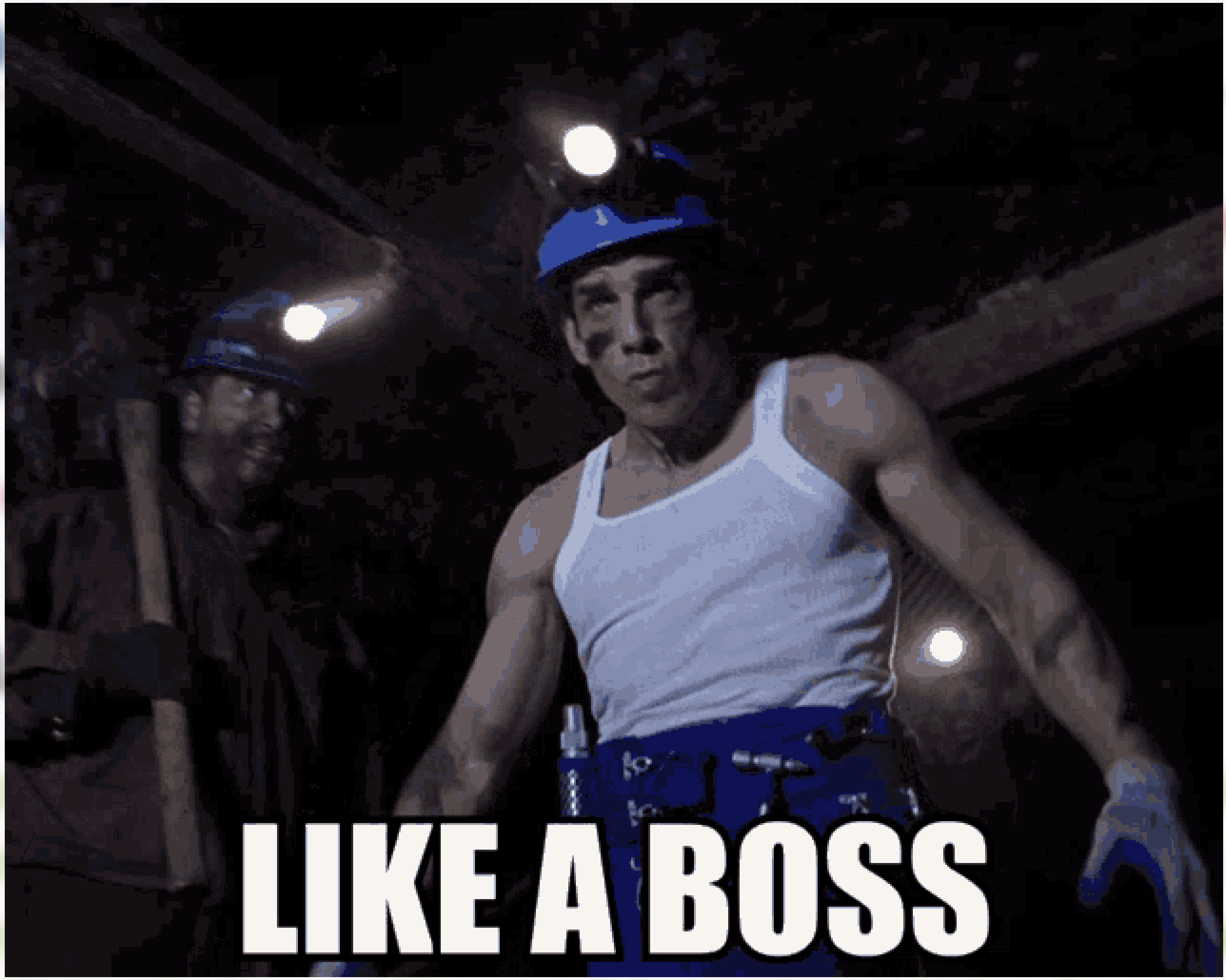
LLMs generate acc rained knowledge.

Example Use Case

"Write a summary

Output (Normal LL

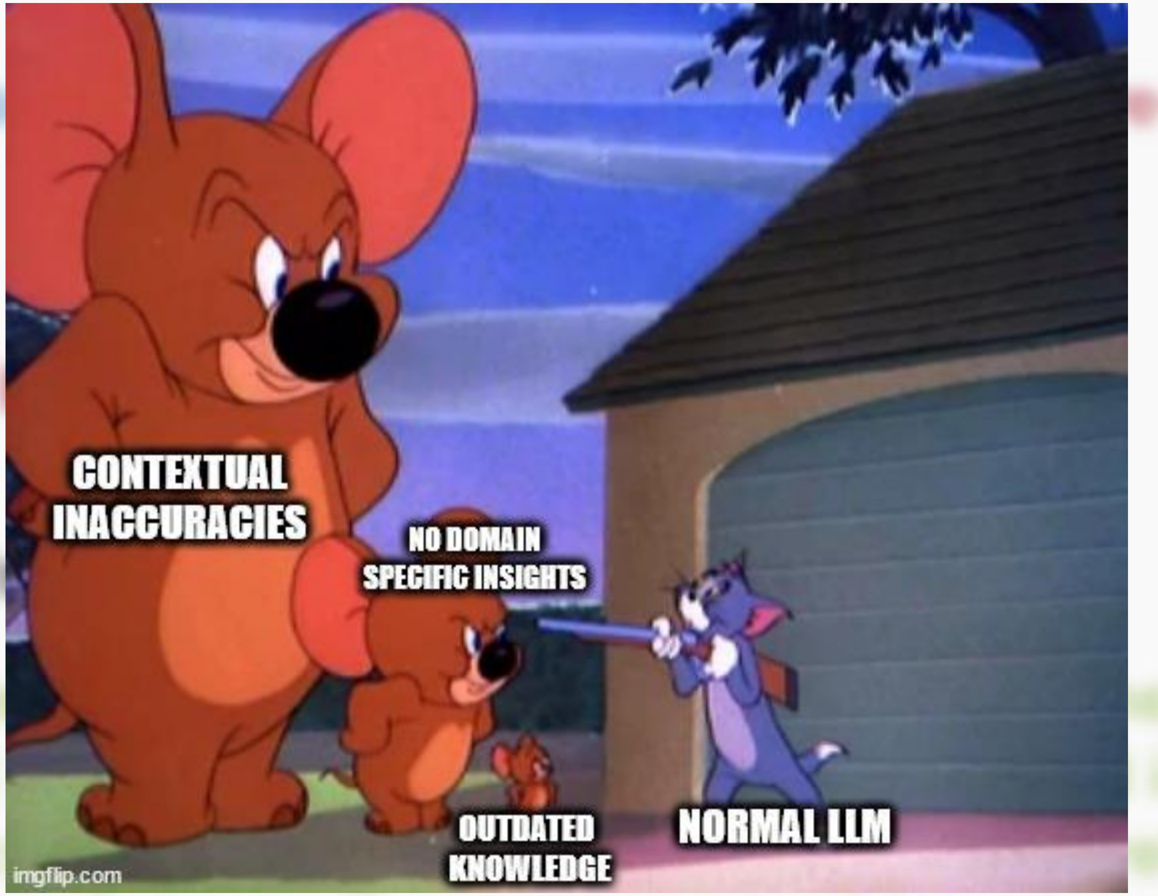
"Einstein's theory o ts: special relativity and gene 1905, revolutionized phy tive concepts, depending on the observer's motion, and introduced the famous equation E=mc². General relativity, proposed in 191 avity as the curvature of spacetime caused by massive ob g a framework to understand phenomena like black holes and the expansion of the universe."



But

Recap: Vanilla LLMs

Example Use Case:



Oh, so how did we address these limitations?

Recap: Vanilla LLMs

LLMs generate accurate … pre-trained knowledge.

Example Use Case:

“Write a summary of Ei

Output (Normal LLMs):

“Einstein's theory of rela … onents: special
relativity and general re … ed in 1905,
revolutionized physics b … e relative concepts,
depending on the obser … amous equation $E=mc^2$.
General relativity, propo … e curvature of
spacetime caused by massive objects, providing a framework to understand
phenomena like black holes and the expansion of the universe.”

Recap: RAGs

As we know, RAG combines the power of LLMs with external data retrieval.

Imagine a use case - We have created an RAG system on top of an electronic health records database.

Example:

“Summarize the key findings from Mr. Smith’s diagnostic history and lab reports.”

Output (RAG):

“The patient has a history of hypertension and Type 2 diabetes. Recent lab tests indicate elevated cholesterol (LDL: 180 mg/dL) and blood glucose (HbA1c: 7.8%). The MRI from last month highlights mild lumbar spine degeneration. Treatment adjustments may be needed to address these trends.”

Outline

- Recap
- **Motivation**
- Introduction
- Components
- Overview

Professor! I know all of this!
Should I declare myself a
Maestro?

Maestro, you say!
Close your eyes and picture
this 🌟

Agentic LLM Workflows: Motivation

Let’s say we pass the following prompt to the LLM

I want to plan a 5-day trip to Paris.

The LLM should help with the following:

1. Find flights and hotels within our budget.
2. Suggest activities like visiting the Louvre etc, tailored to our preferences.
3. Dynamically adjust plans if there’s a weather issue or a flight delay.

A seamless travel experience where everything is taken care of effortlessly.

Agentic LLM Workflows: Motivation

Let's say we pass th

Can RAGs or LLMs alone achieve this?

Paris.

The LLM should help with the following:

1. Find flights and hotels within our budget.

LLMs

Can generate itineraries but lack **dynamic decision-making** capabilities.

RAGs

Limited to **retrieving and summarizing information** (e.g., fetching flight options).

A seamless travel experience where everything is taken care of effortlessly.

Agentic LLM Workflows: Motivation

So, how do we implement this travel planner? What do LLMs and RAG lack?

1. **No Interaction with the External World:**
   Cannot access real-time systems like weather, maps, or booking platforms.
2. **Absence of Tools:**
   Lacks the ability to directly use APIs or services to book flights, hotels, or other essentials.
3. **Lack of Autonomy:**
   Cannot make decisions or take actions independently to adapt to changing situations.

We need something more—an advanced system that can interact, adapt, and act.

Agentic LLM Workflows: Motivation

So, how do we turn

1. **No Interaction**
   Cannot access
   king platforms.

2. **Absence of Too**
   Lacks the abilit
   nts, hotels, or
   other essential

3. **Lack of Autono**
   Cannot make d
   apt to changing
   situations.

We need something more—an advanced system that can interact, adapt, and act.

Outline

- Recap
- Motivation
- **Introduction**
- Components
- Overview

Agentic LLM Workflows: Introduction

"While there isn't a widely accepted definition for LLM-powered agents, they can be described as a system that can use an LLM to reason through a problem, create a plan to solve the problem, and execute the plan with the help of a set of tools."

Agents are a system with complex reasoning capabilities, and the means to execute tasks.

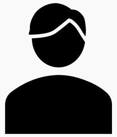
Source: Nvidia

Agentic LLM Workflows: Introduction

Let's revisit our travel planner requirements and explore how we can create one!

Step 1: The Prompt

- Everything starts with the user's request.
- This input forms the foundation of what the system must deliver.



The Prompt

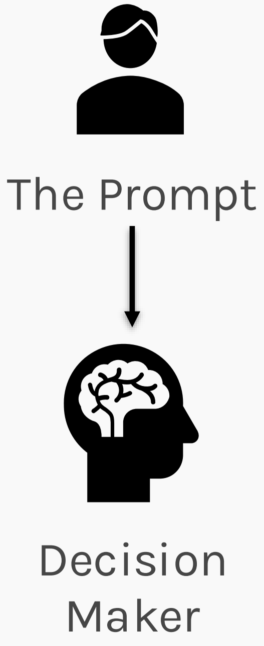
Agentic LLM Workflows: Introduction

Let's revisit our travel planner requirements and explore how we can create one!

Step 2: A Decision-Maker

- It acts like the brain of the system.

- It ensures all components are aligned and processes information.

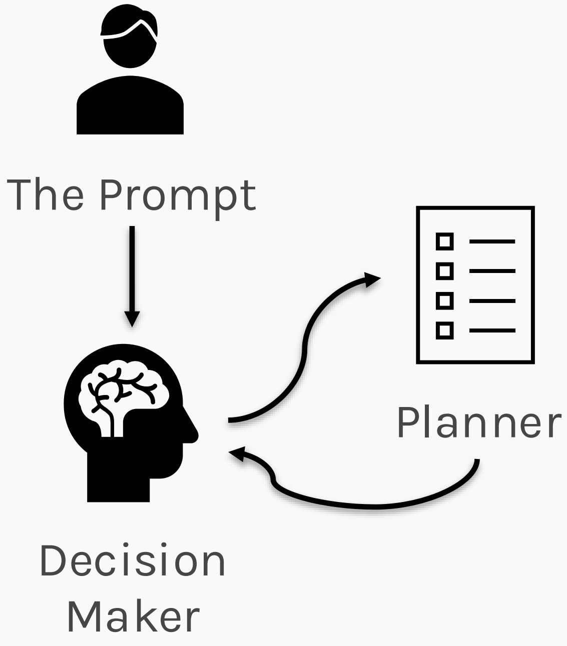


Agentic LLM Workflows: Introduction

Let's revisit our travel planner requirements and explore how we can create one!

Step 3: A Planner

- To break down the problem, we need something that can create a detailed plan.
- Analyzes the user's prompt, constraints and available options.
- Outputs a clear sequence of steps to achieve the goal.

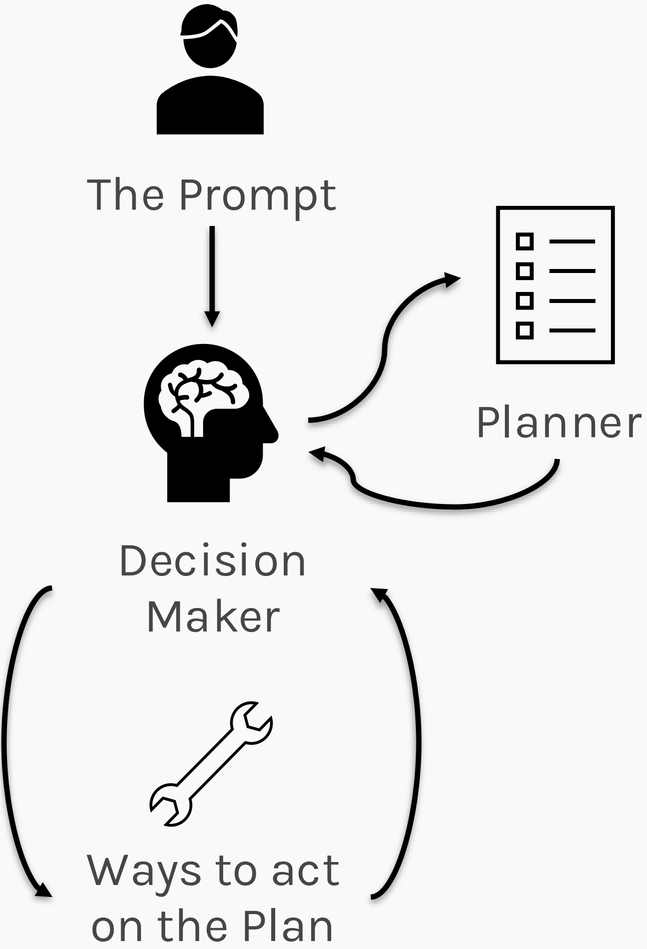


Agentic LLM Workflows: Introduction

Let's revisit our travel planner requirements and explore how we can create one!

Step 4: Ways to Act on the Plan

- Plans alone aren't enough; we need a way to act on them
- The system must connect with external resources to bring the plan to life.

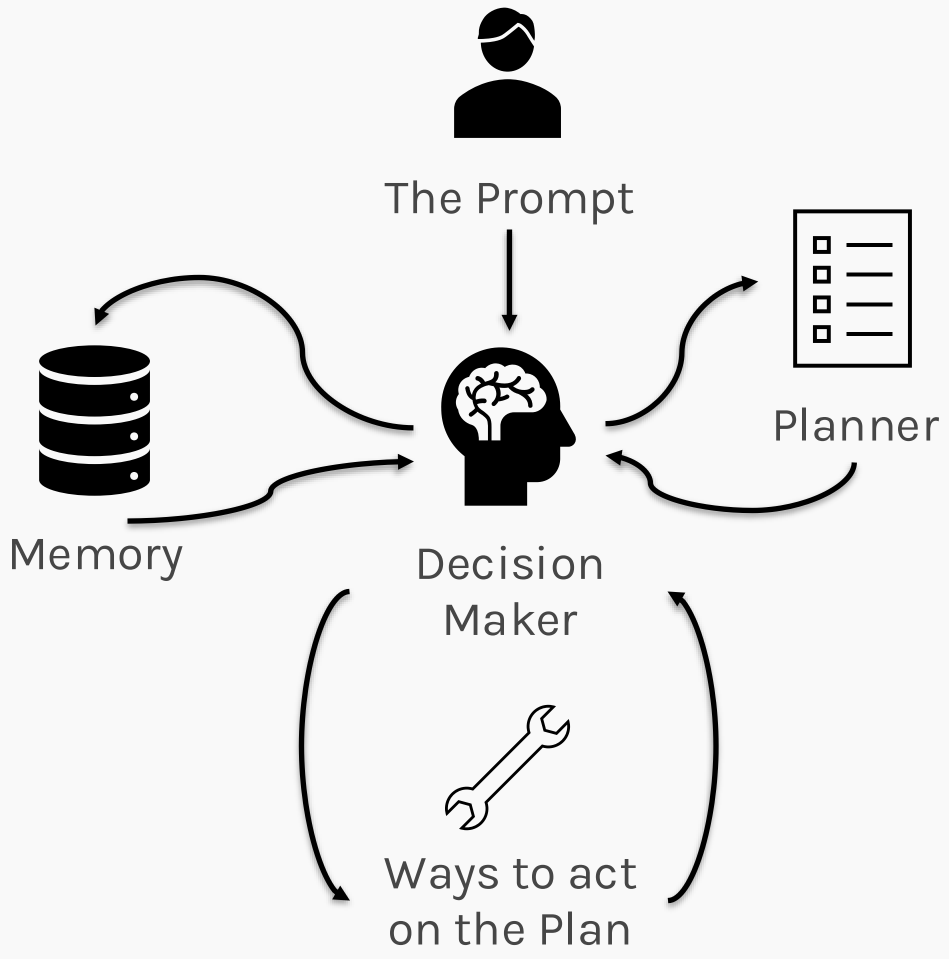


Agentic LLM Workflows: Introduction

Let's revisit our travel planner requirements and explore how we can create one!

Step 5: Memory

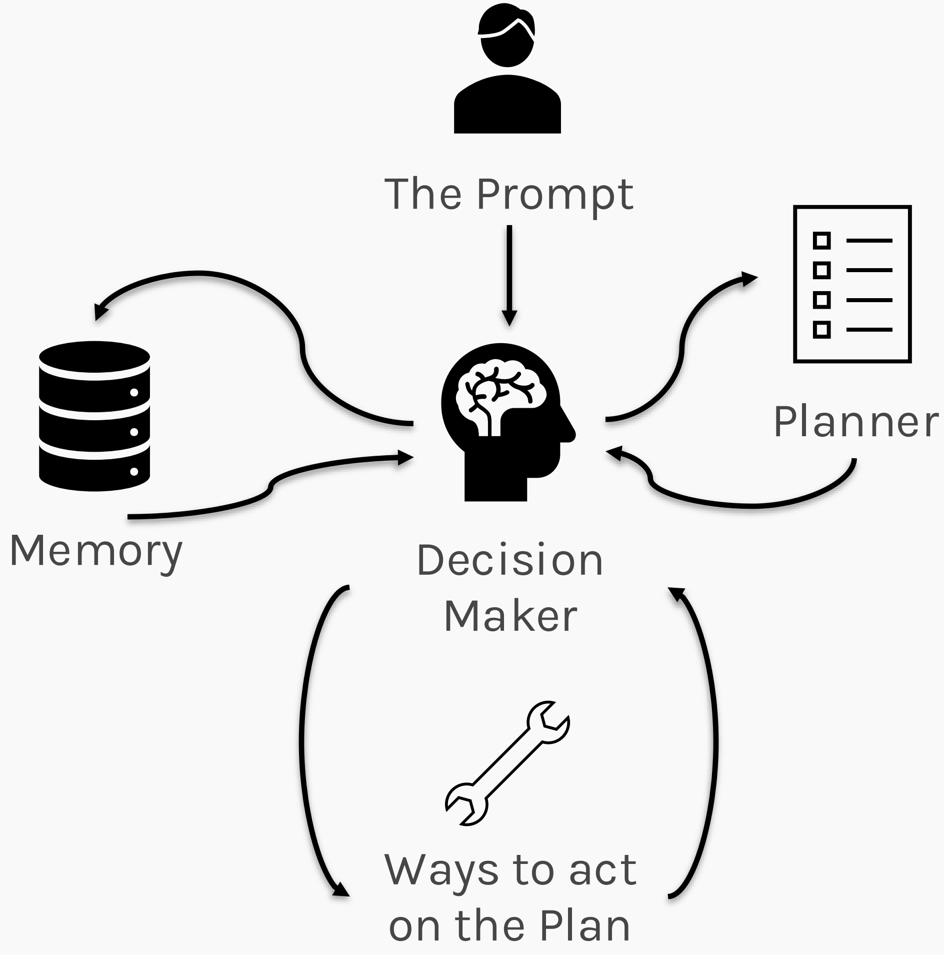
- Lastly, we need to remember what the user wants and adapt over time.
- This can store details like:
  - Budget, preferences, and previous actions.
- Ensures continuity and a personalized experience throughout multiple sessions.



Agentic LLM Workflows: Introduction

And that is how we create a travel planner!

Oh wait… these aren't just steps. These are the **core components of an agent**!

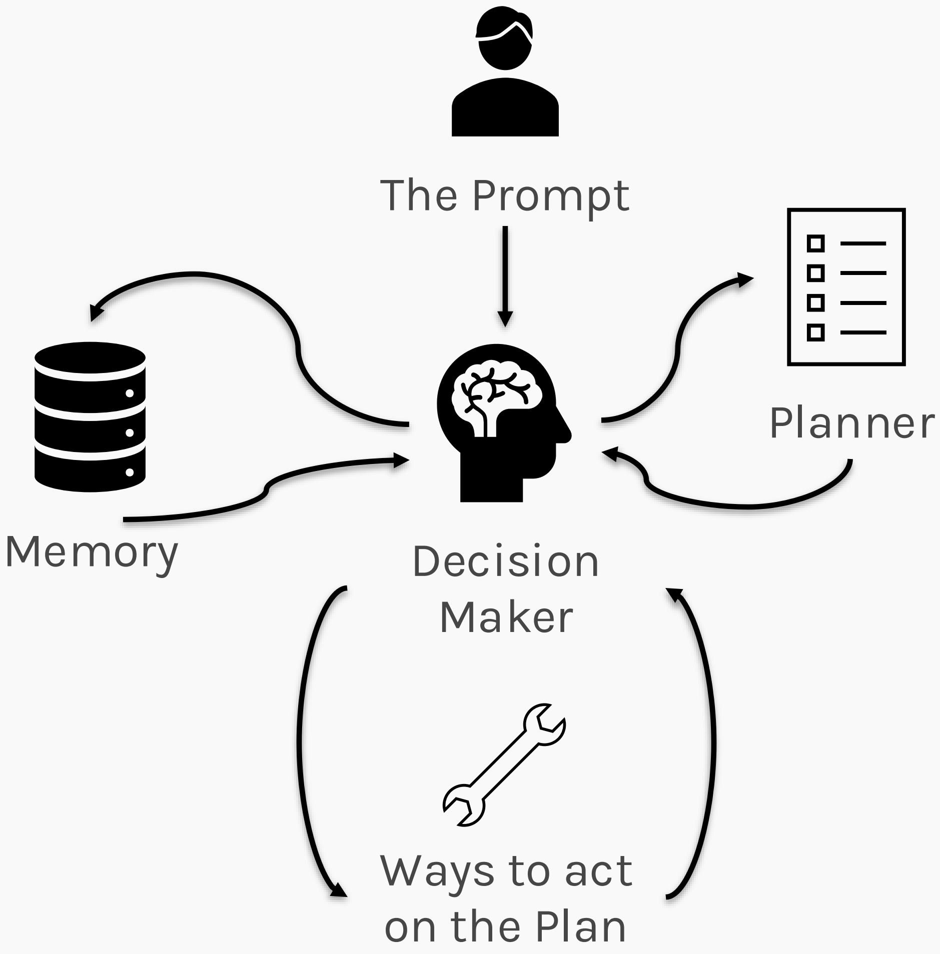


Outline

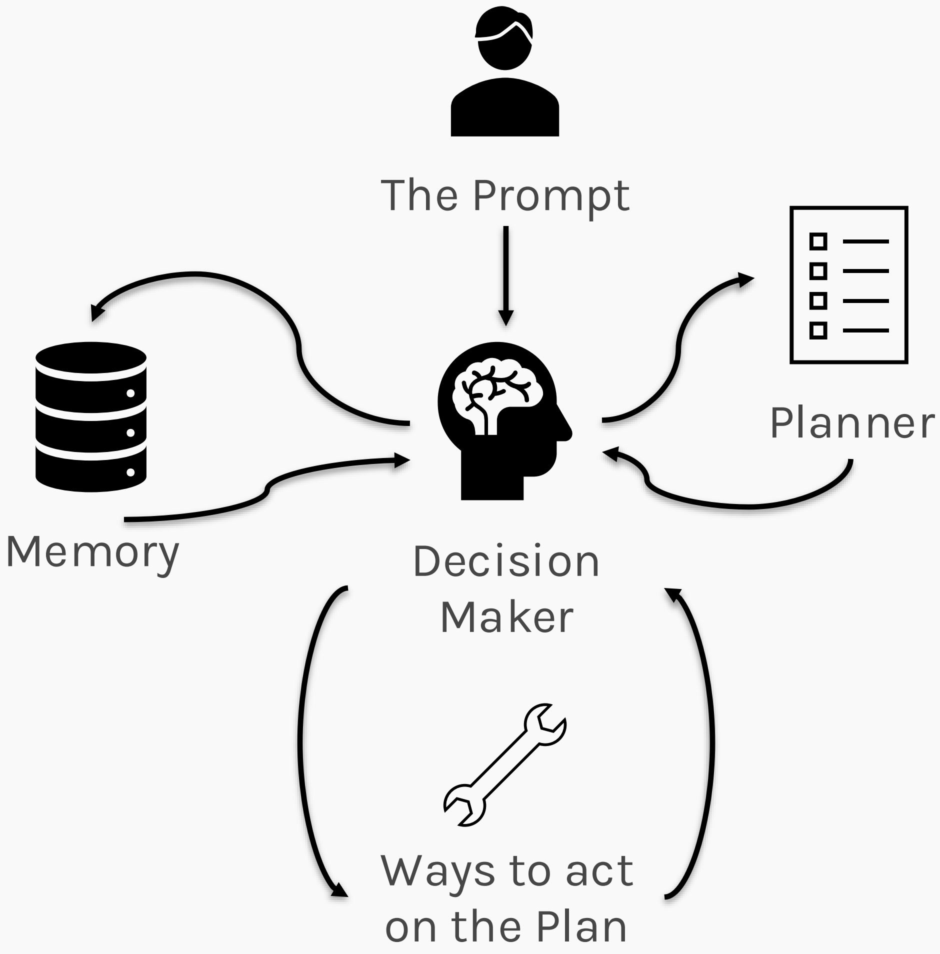
- Recap
- Motivation
- Introduction
- **Components**
- Overview

Agentic LLM Workflows: Components

Now that we have an idea of how the components come together to create an agent, let's dive deeper into each one.



Agentic LLM Workflows: Components



Agentic LLM Workflows: Components

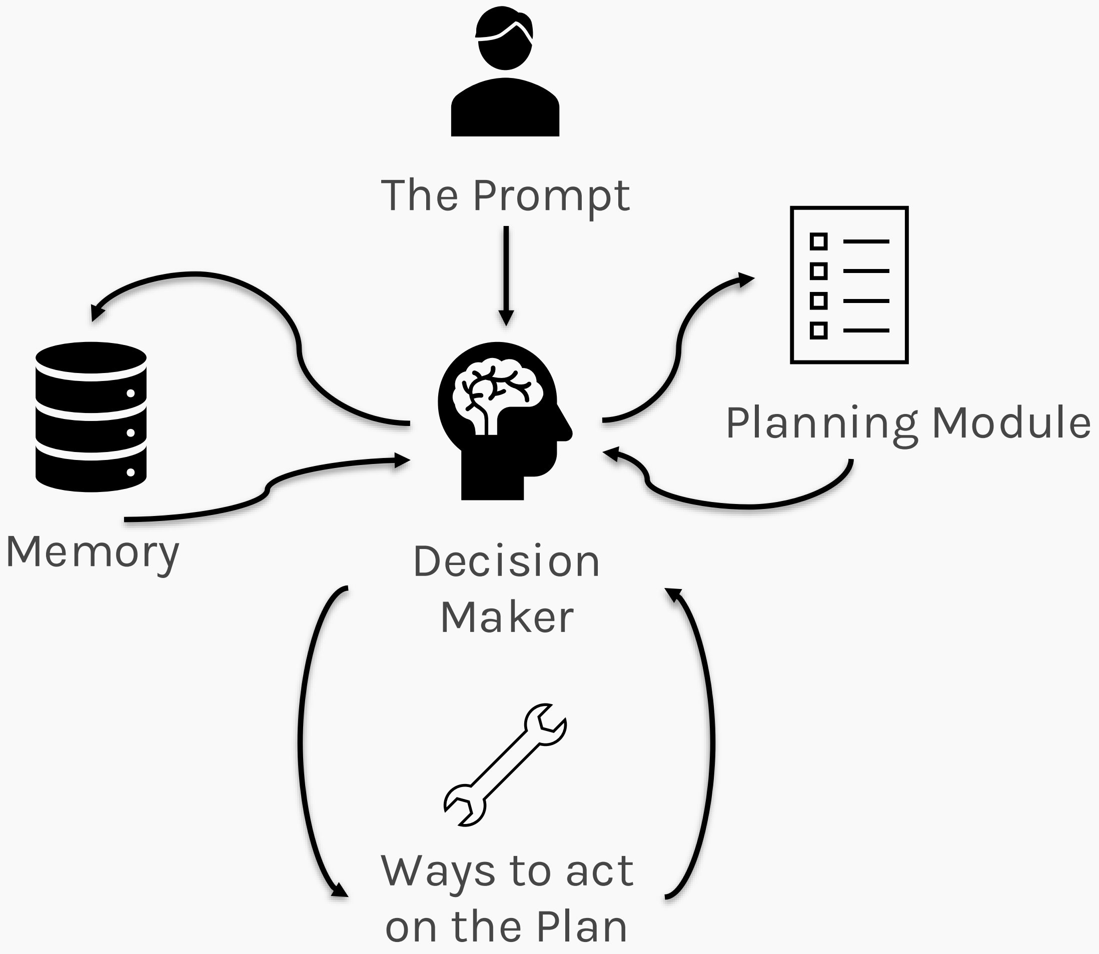
Planning Module

- Helps in dealing with complex tasks.

- The two techniques used here are:
  - **Decomposition**: Breaking down the problem into multiple steps.
  - **Reflection**: Refines the execution plan using techniques like chain-of-thought, ReAct etc.

Planning Module

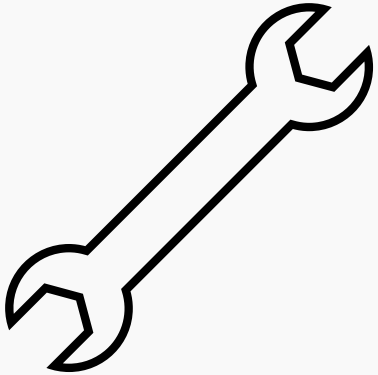
Agentic LLM Workflows: Components



Agentic LLM Workflows: Components

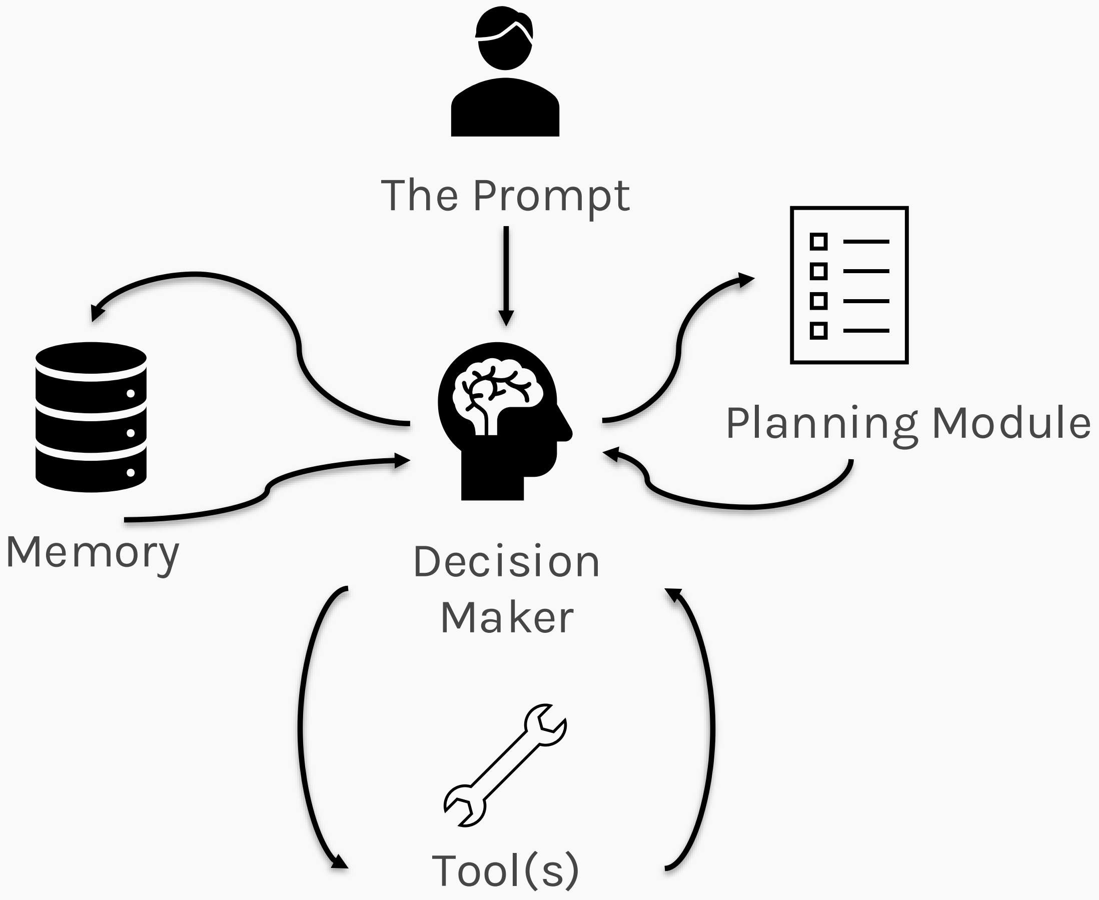
Tool(s)

- External programs/resources for interacting with the world.
- **Functions**:
  - **Information Retrieval**: Web searches, database access.
  - **Calculations/Execution**: Solving equations, running code.
  - **Specific Actions**: API interactions, software control.
- **Importance**: Enable agents to act and gather information beyond the LLM's limits.



Tool(s)

Agentic LLM Workflows: Components



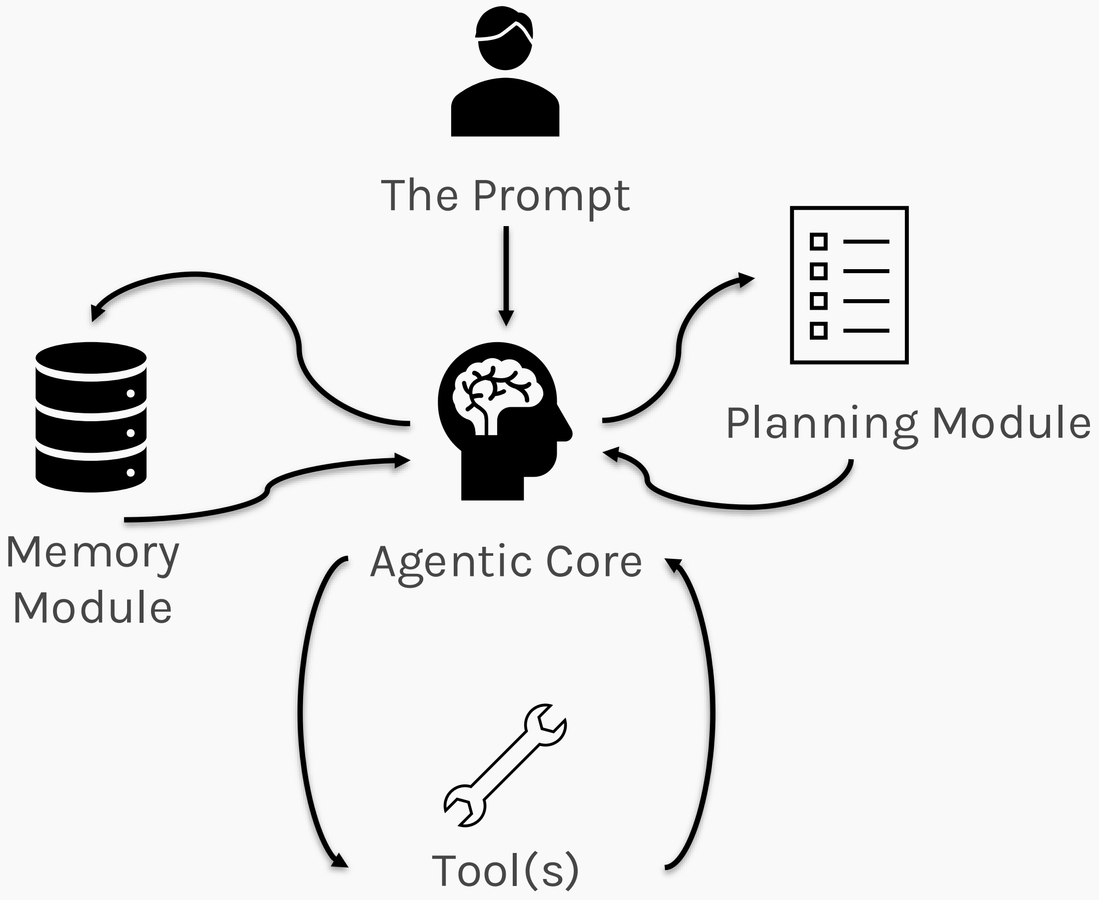
Agentic LLM Workflows: Components

Agentic Core

- Processes information and makes decisions.
- **Input**: Prompt, objectives and knowledge
- **Processing**: Uses reasoning and patterns to understand the situation.
- **Output**: Actions like:
  - Direct commands to tools (e.g., "search for X")
  - Requests for plans (e.g., "plan a trip to Y")
  - Memory updates (e.g., "remember fact Z")

Agentic Core

Agentic LLM Workflows: Components



The Prompt

Memory
Module

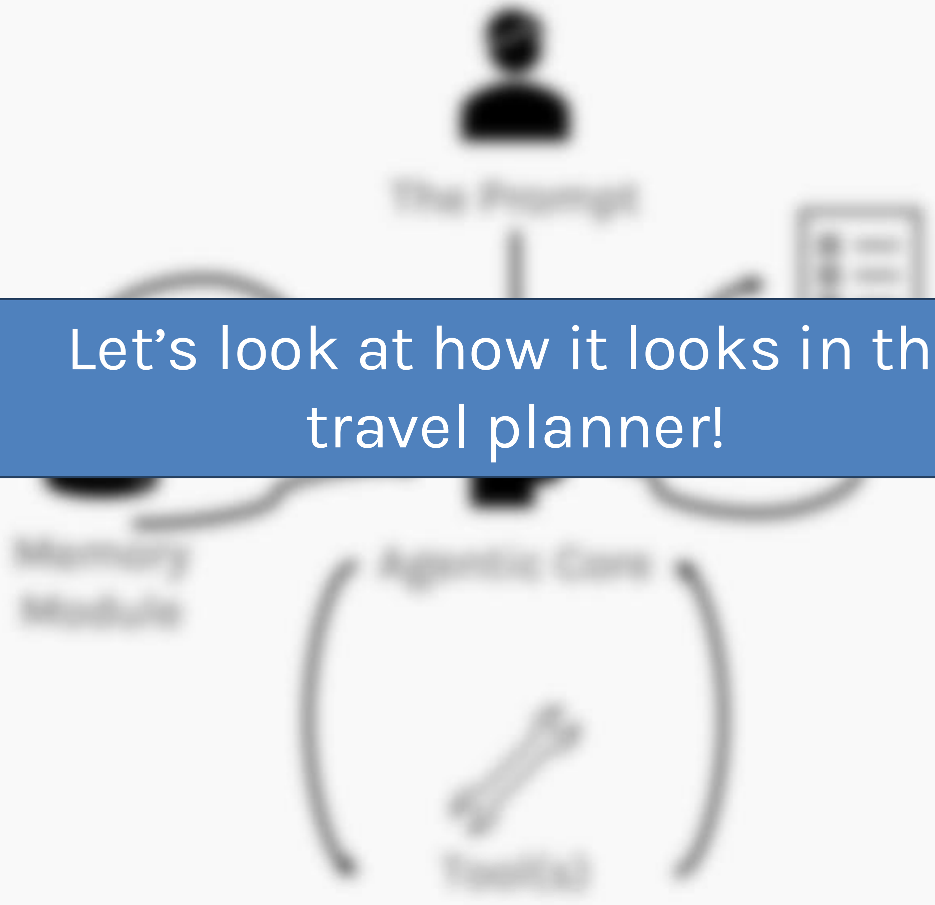
Agentic Core

Planning Module

Tool(s)

Components of the Agentic
System

Agentic LLM Workflows: Components

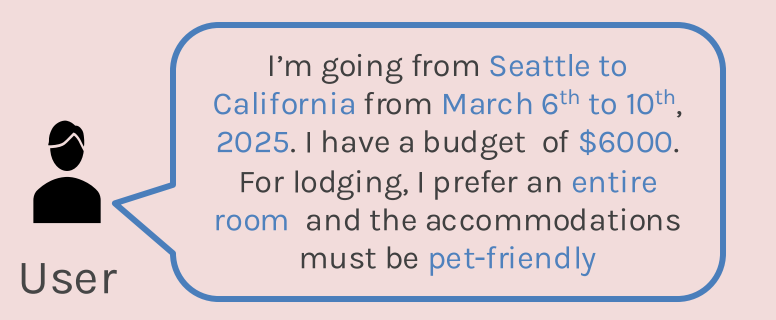


Let’s look at how it looks in the travel planner!

Components of the Agentic System

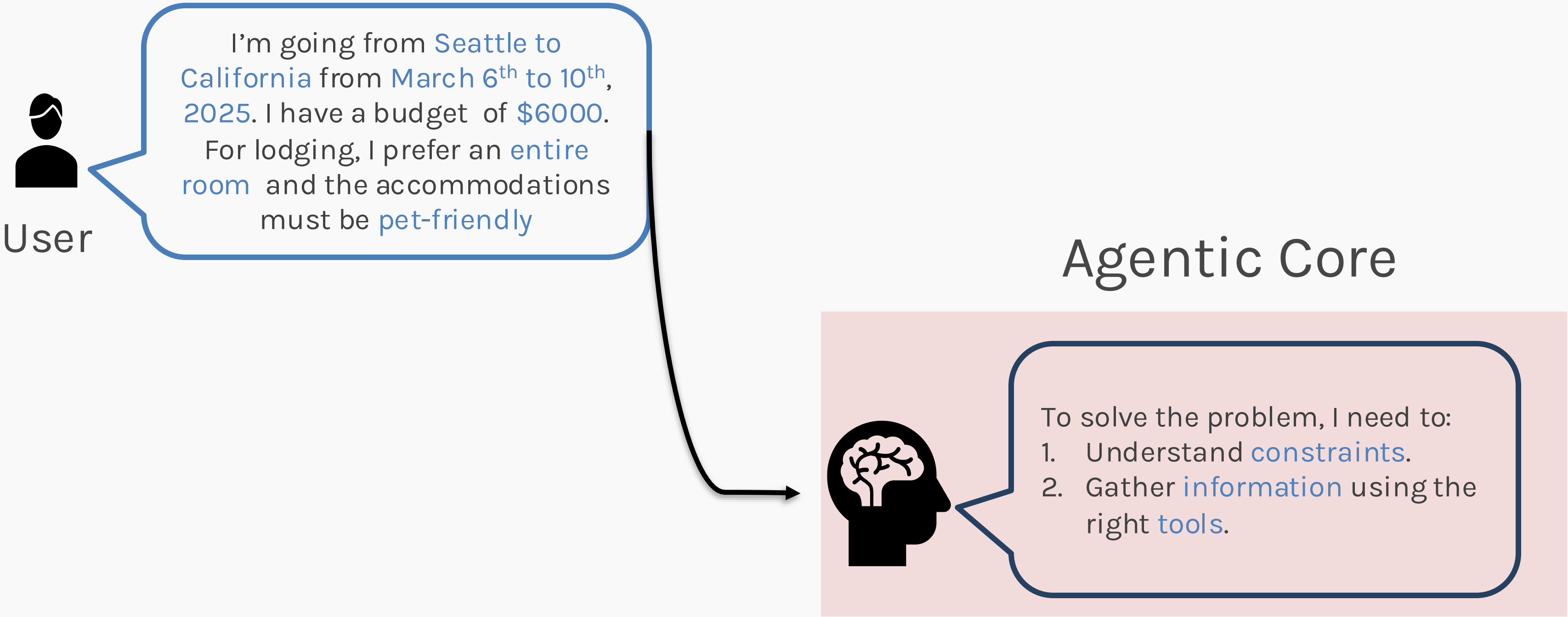
Agentic LLM Workflows: Components

The Prompt



I'm going from Seattle to California from March 6th to 10th, 2025. I have a budget of $6000. For lodging, I prefer an entire room and the accommodations must be pet-friendly

Agentic LLM Workflows: Components

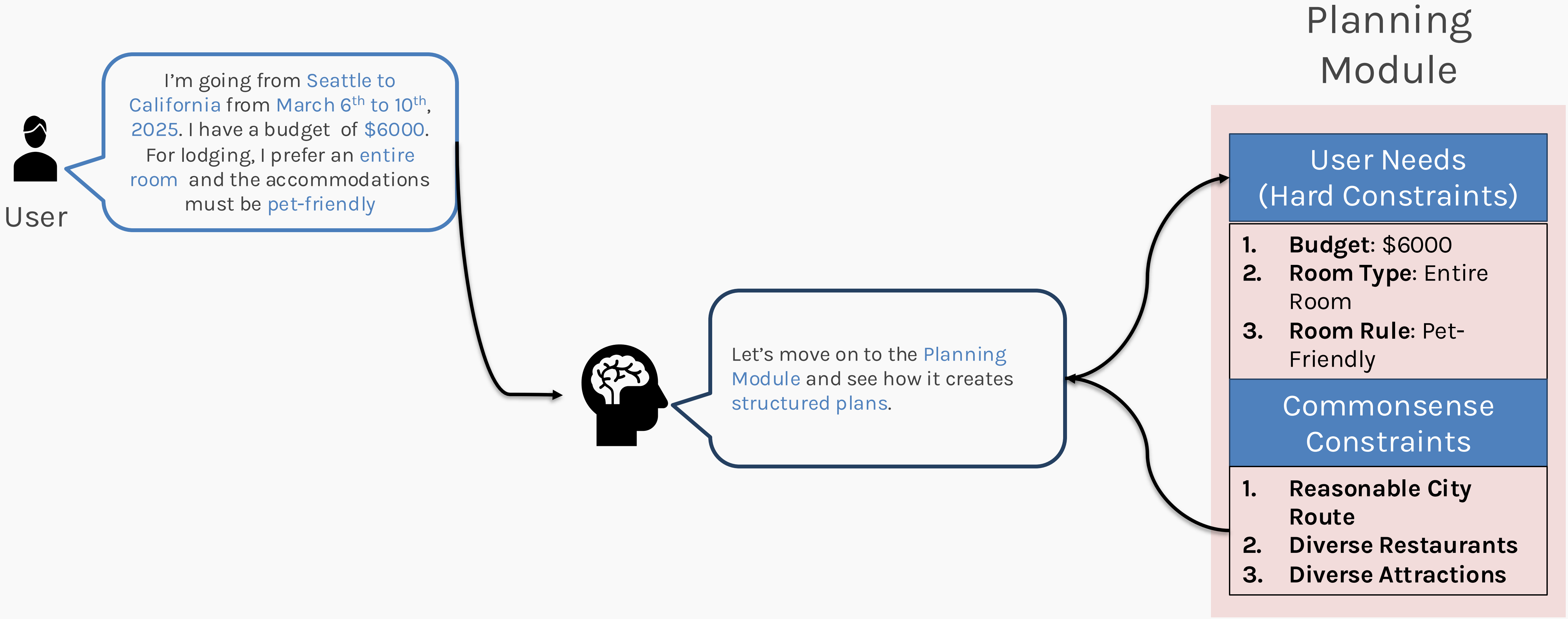


I'm going from Seattle to California from March 6th to 10th, 2025. I have a budget of $6000. For lodging, I prefer an entire room and the accommodations must be pet-friendly

To solve the problem, I need to:

1. Understand constraints.
2. Gather information using the right tools.

Agentic LLM Workflows: Components



User

I'm going from Seattle to California from March 6th to 10th, 2025. I have a budget of $6000. For lodging, I prefer an entire room and the accommodations must be pet-friendly

Let's move on to the Planning Module and see how it creates structured plans.

Planning Module

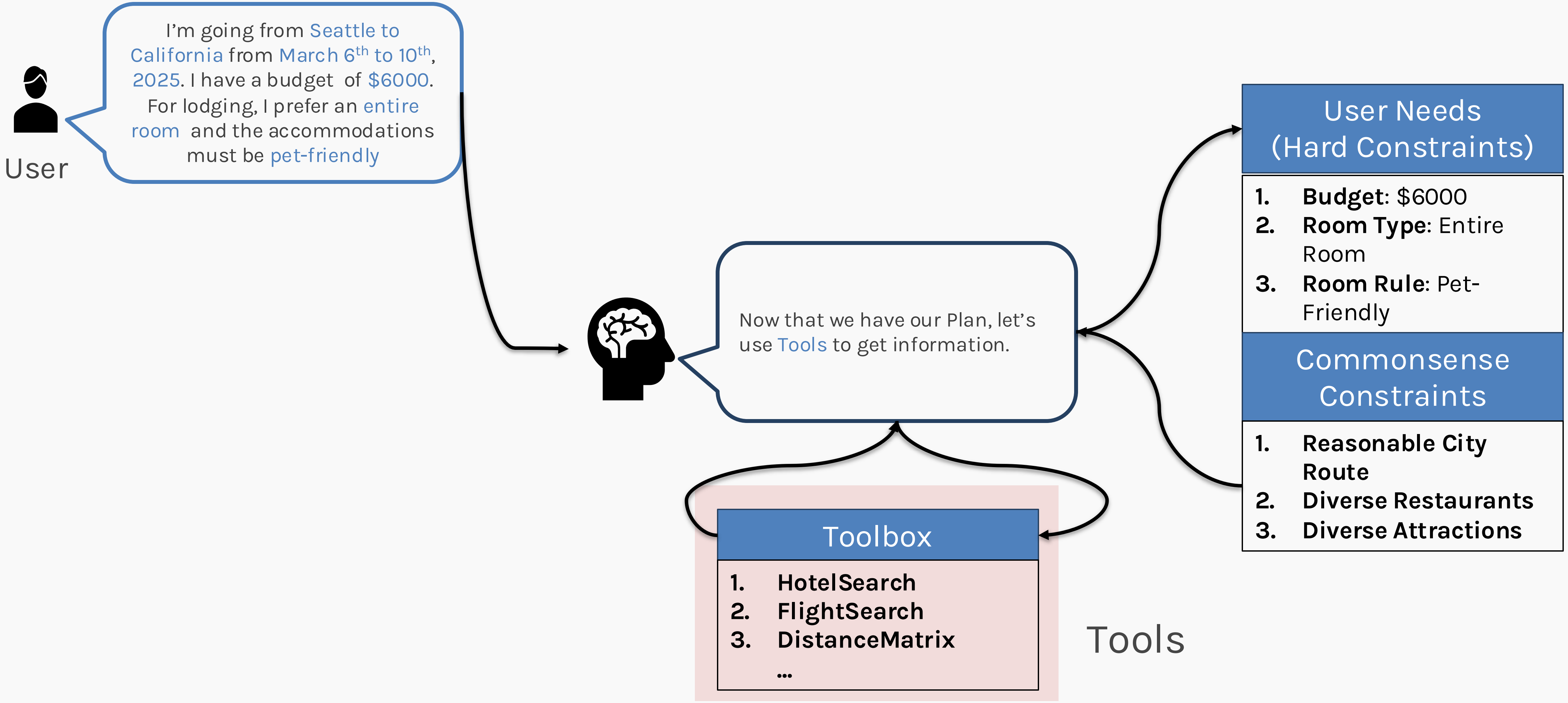
User Needs (Hard Constraints)

1. **Budget**: $6000
2. **Room Type**: Entire Room
3. **Room Rule**: Pet-Friendly

Commonsense Constraints

1. **Reasonable City Route**
2. **Diverse Restaurants**
3. **Diverse Attractions**

Agentic LLM Workflows: Components



User

I'm going from Seattle to California from March 6th to 10th, 2025. I have a budget of $6000. For lodging, I prefer an entire room and the accommodations must be pet-friendly

Now that we have our Plan, let's use Tools to get information.

User Needs (Hard Constraints)

1. **Budget**: $6000
2. **Room Type**: Entire Room
3. **Room Rule**: Pet-Friendly

Commonsense Constraints

1. **Reasonable City Route**
2. **Diverse Restaurants**
3. **Diverse Attractions**

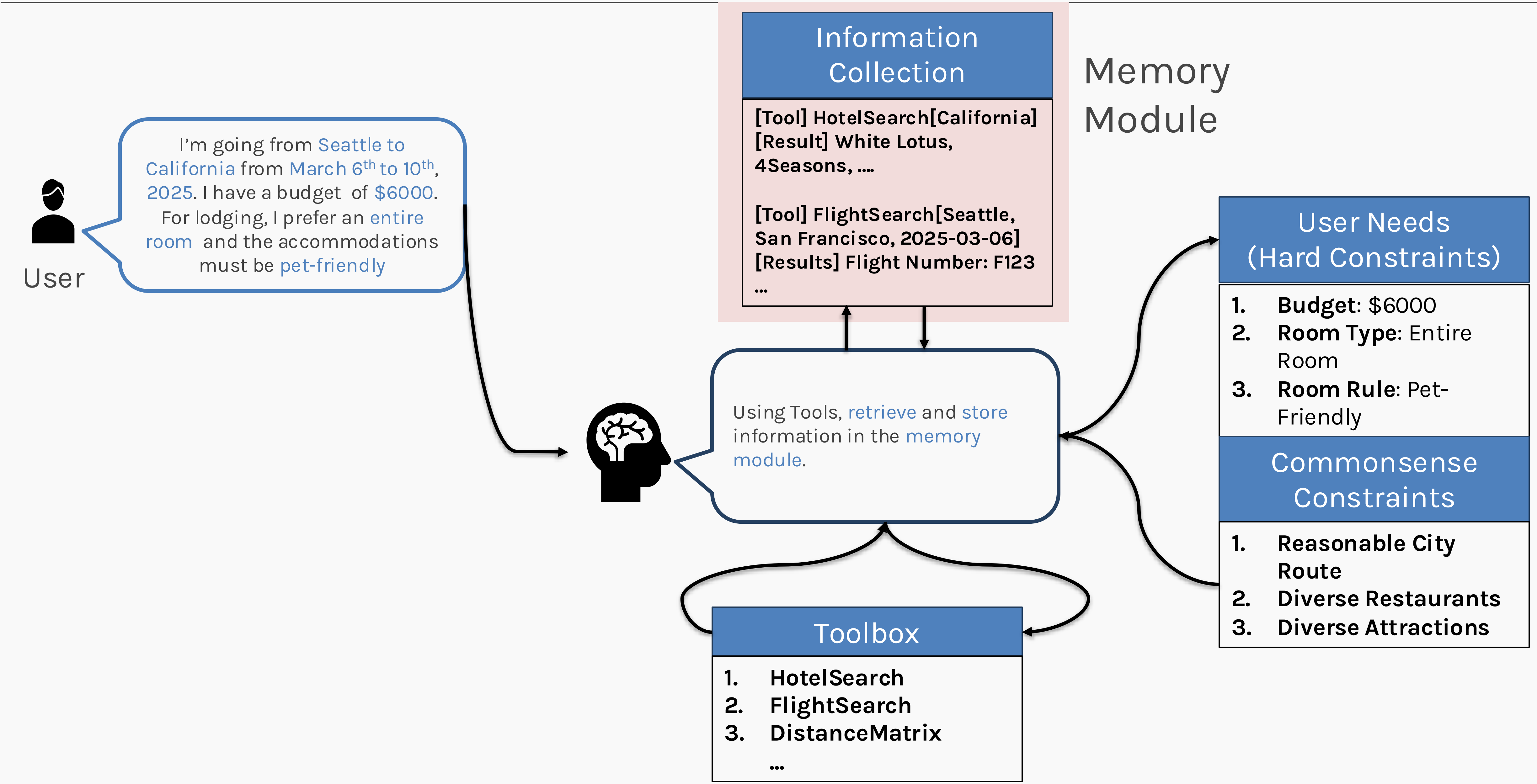
Toolbox

1. **HotelSearch**
2. **FlightSearch**
3. **DistanceMatrix**

…

Tools

Agentic LLM Workflows: Components



User

I’m going from Seattle to California from March 6th to 10th, 2025. I have a budget of $6000. For lodging, I prefer an entire room and the accommodations must be pet-friendly

Memory Module

Information Collection

[Tool] HotelSearch[California]
[Result] White Lotus, 4Seasons, ….

[Tool] FlightSearch[Seattle, San Francisco, 2025-03-06]
[Results] Flight Number: F123
…

Using Tools, retrieve and store information in the memory module.

User Needs (Hard Constraints)

1. **Budget**: $6000
2. **Room Type**: Entire Room
3. **Room Rule**: Pet-Friendly

Commonsense Constraints

1. **Reasonable City Route**
2. **Diverse Restaurants**
3. **Diverse Attractions**

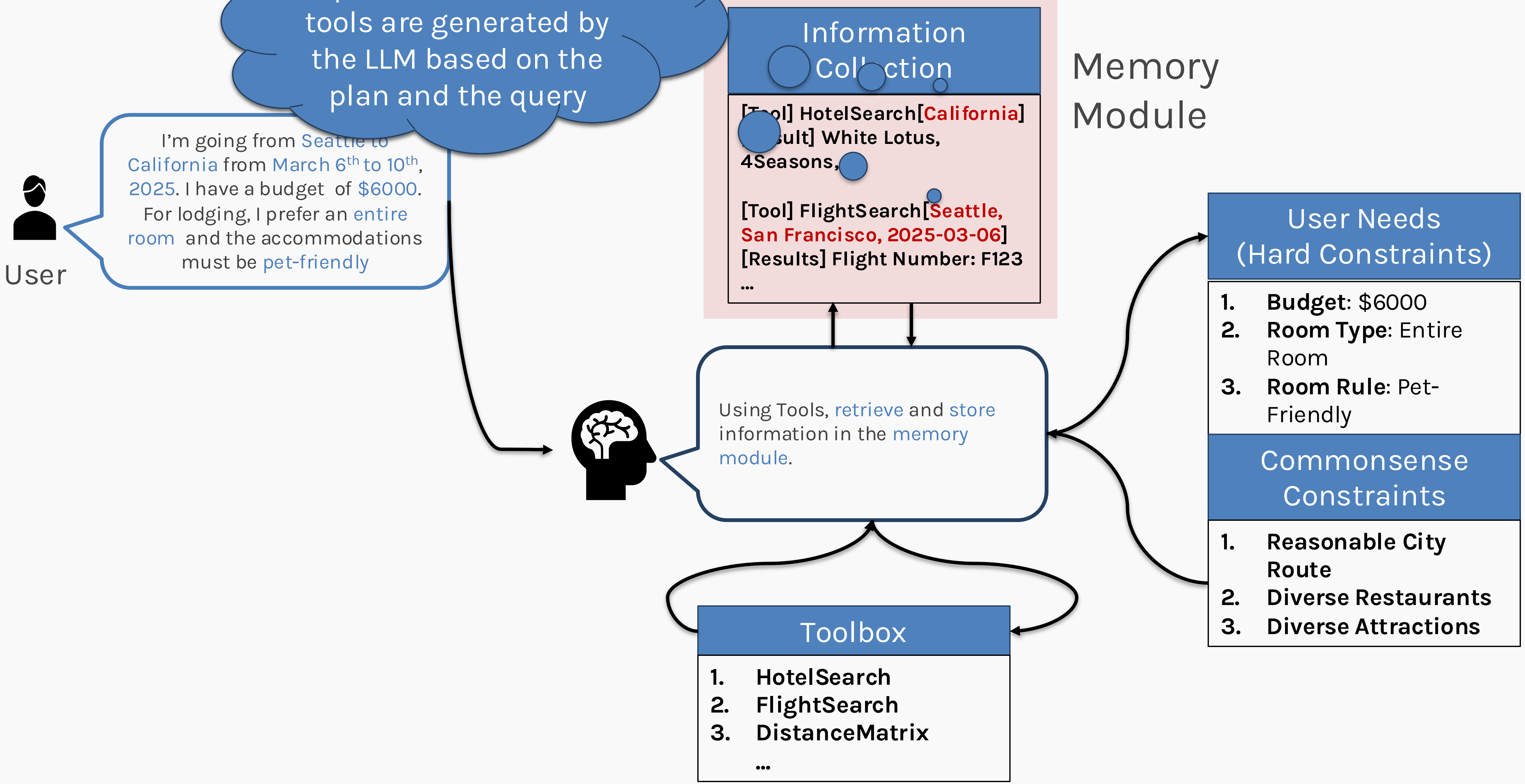
Toolbox

1. **HotelSearch**
2. **FlightSearch**
3. **DistanceMatrix**

…

Agentic L mponents

These input parameters to the tools are generated by the LLM based on the plan and the query



User

I'm going from Seattle to California from March 6th to 10th, 2025. I have a budget of $6000. For lodging, I prefer an entire room and the accommodations must be pet-friendly

Information Coll ction

[T ol] HotelSearch[California]
sult] White Lotus, 4Seasons,

[Tool] FlightSearch[Seattle, San Francisco, 2025-03-06]
[Results] Flight Number: F123
…

Memory Module

Using Tools, retrieve and store information in the memory module.

User Needs (Hard Constraints)

1. **Budget**: $6000
2. **Room Type**: Entire Room
3. **Room Rule**: Pet-Friendly

Commonsense Constraints

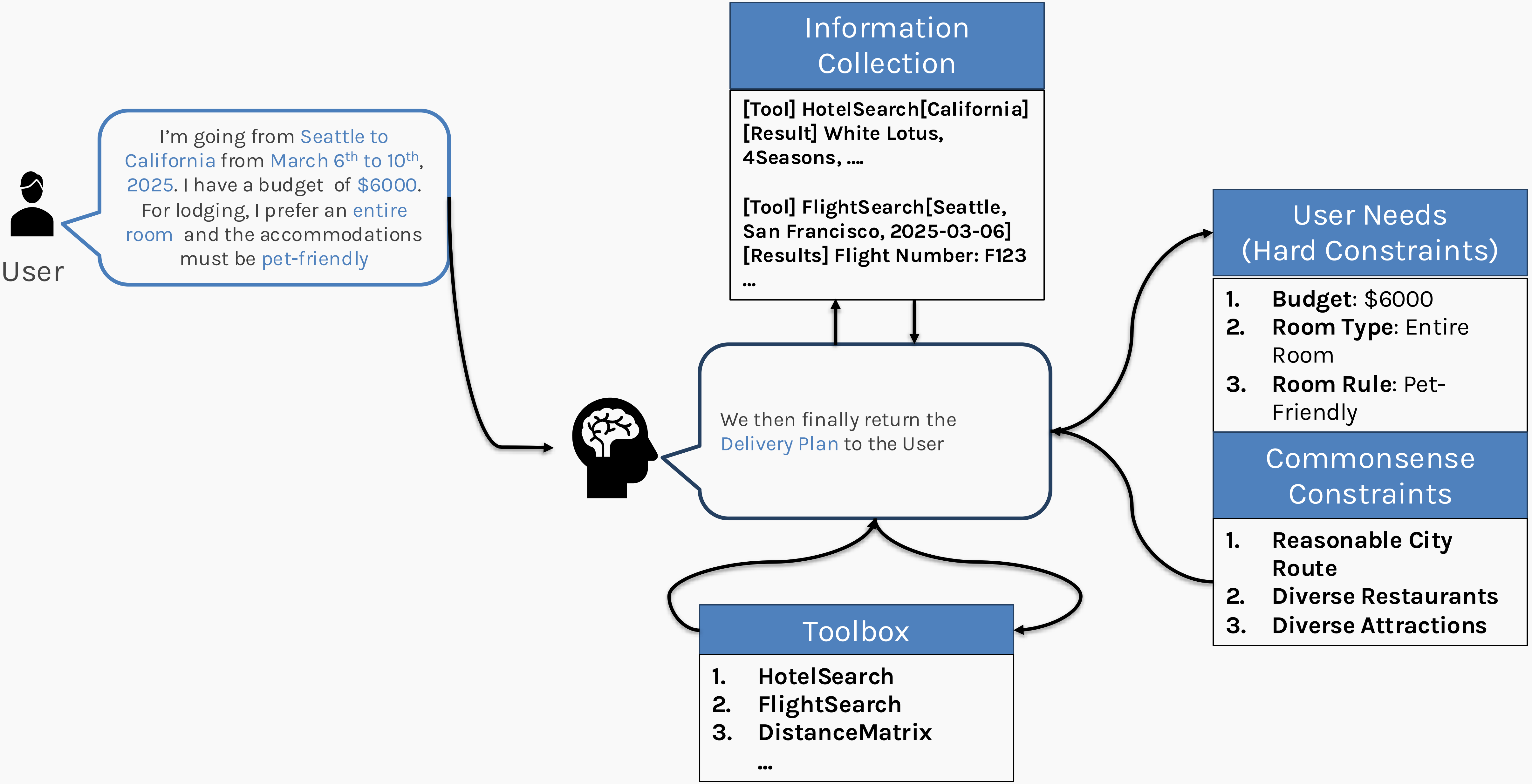
1. **Reasonable City Route**
2. **Diverse Restaurants**
3. **Diverse Attractions**

Toolbox

1. **HotelSearch**
2. **FlightSearch**
3. **DistanceMatrix**

…

Agentic LLM Workflows: Components



User

I'm going from Seattle to California from March 6th to 10th, 2025. I have a budget of $6000. For lodging, I prefer an entire room and the accommodations must be pet-friendly

We then finally return the Delivery Plan to the User

Information Collection

[Tool] HotelSearch[California]
[Result] White Lotus, 4Seasons, ….

[Tool] FlightSearch[Seattle, San Francisco, 2025-03-06]
[Results] Flight Number: F123
…

User Needs (Hard Constraints)

1. **Budget**: $6000
2. **Room Type**: Entire Room
3. **Room Rule**: Pet-Friendly

Commonsense Constraints

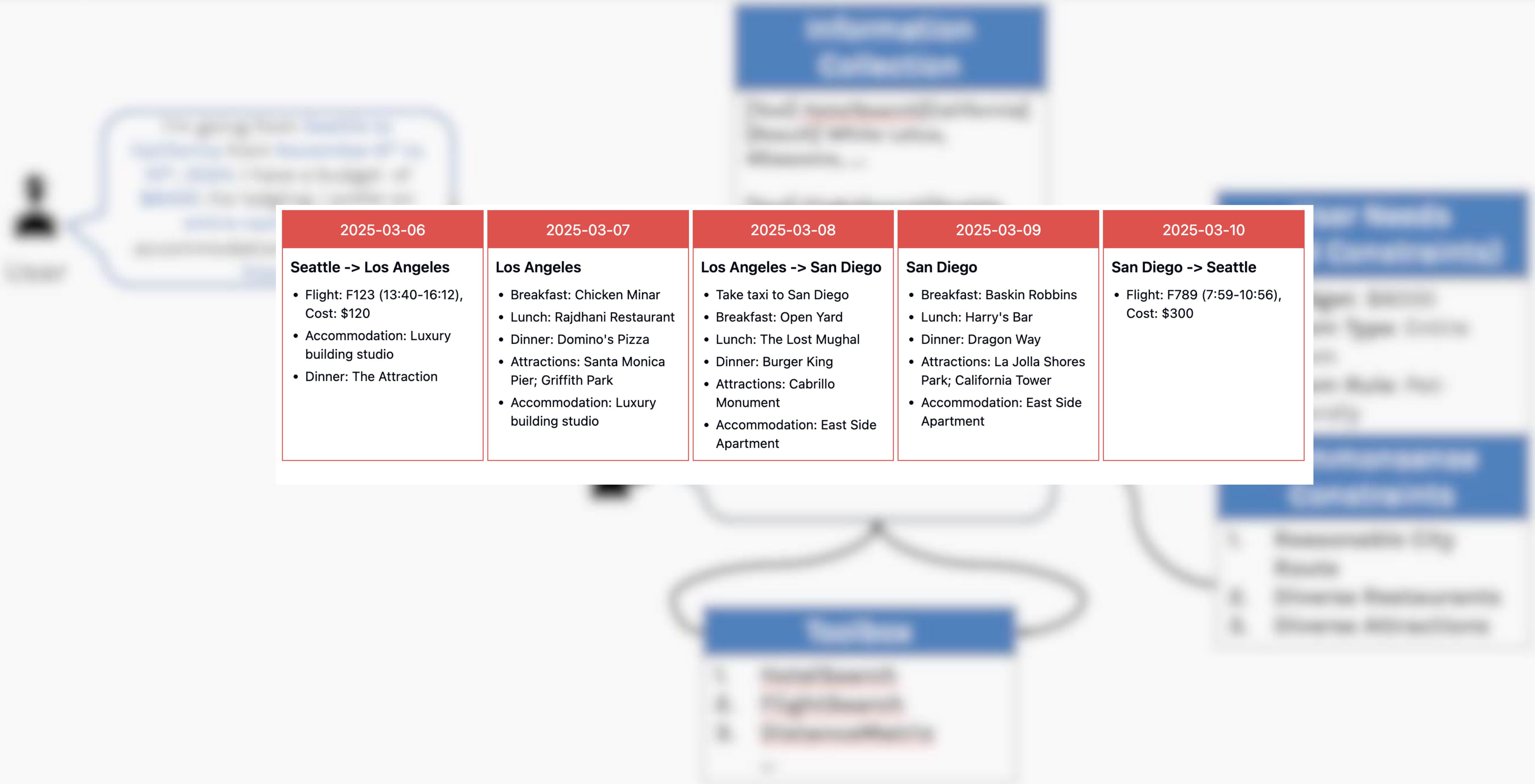
1. **Reasonable City Route**
2. **Diverse Restaurants**
3. **Diverse Attractions**

Toolbox

1. **HotelSearch**
2. **FlightSearch**
3. **DistanceMatrix**

…

Agentic LLM Workflows: Components



2025-03-06

Seattle -> Los Angeles

- Flight: F123 (13:40-16:12), Cost: $120
- Accommodation: Luxury building studio
- Dinner: The Attraction

2025-03-07

Los Angeles

- Breakfast: Chicken Minar
- Lunch: Rajdhani Restaurant
- Dinner: Domino's Pizza
- Attractions: Santa Monica Pier; Griffith Park
- Accommodation: Luxury building studio

2025-03-08

Los Angeles -> San Diego

- Take taxi to San Diego
- Breakfast: Open Yard
- Lunch: The Lost Mughal
- Dinner: Burger King
- Attractions: Cabrillo Monument
- Accommodation: East Side Apartment

2025-03-09

San Diego

- Breakfast: Baskin Robbins
- Lunch: Harry's Bar
- Dinner: Dragon Way
- Attractions: La Jolla Shores Park; California Tower
- Accommodation: East Side Apartment

2025-03-10

San Diego -> Seattle

- Flight: F789 (7:59-10:56), Cost: $300

Agentic LLM Workflows

Agentic setups tend to improve the performance of LLMs beyond their vanilla implementations.

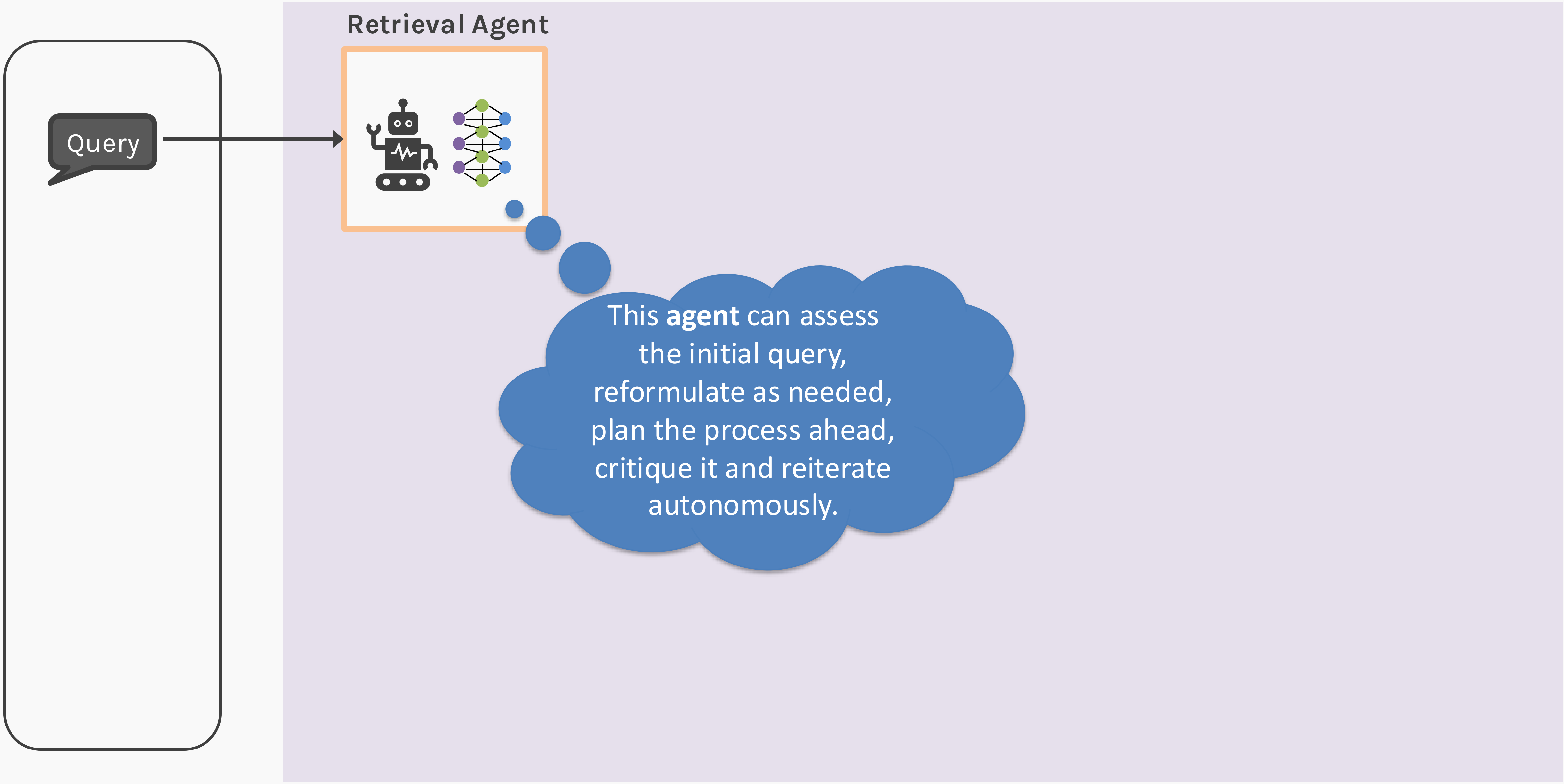
When employing the Agentic Workflow to break tasks into multiple steps—such as problem analysis, iterative code writing, testing, and debugging—**GPT-3.5's performance can even surpass the accuracy of GPT-4 in zero-shot prompting scenarios.**

Agentic RAG

The RAG setup can be enhanced by equipping the system with agent-like capabilities!

Let’s look at one possible scenario!

Agentic RAG

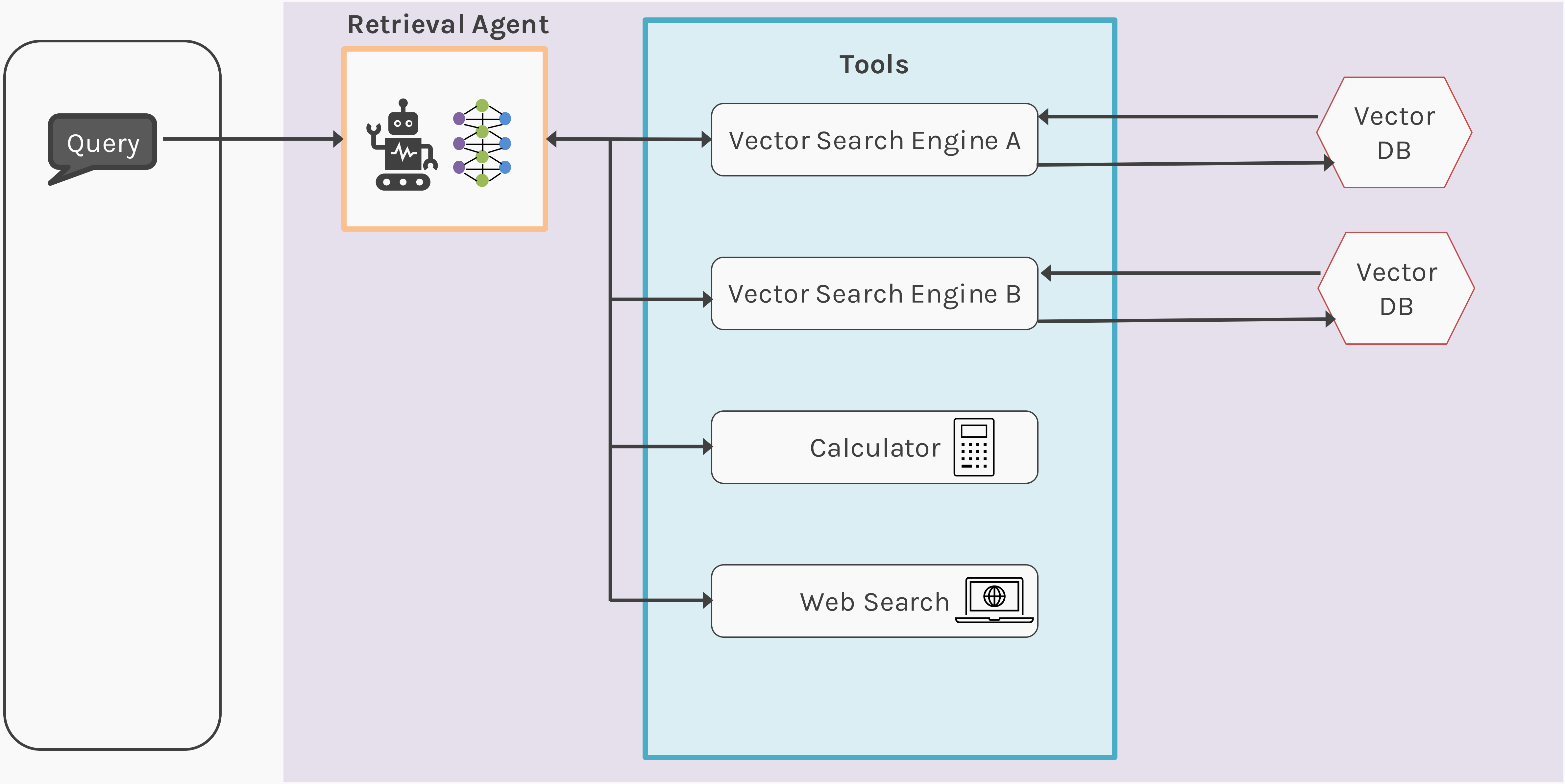


Query

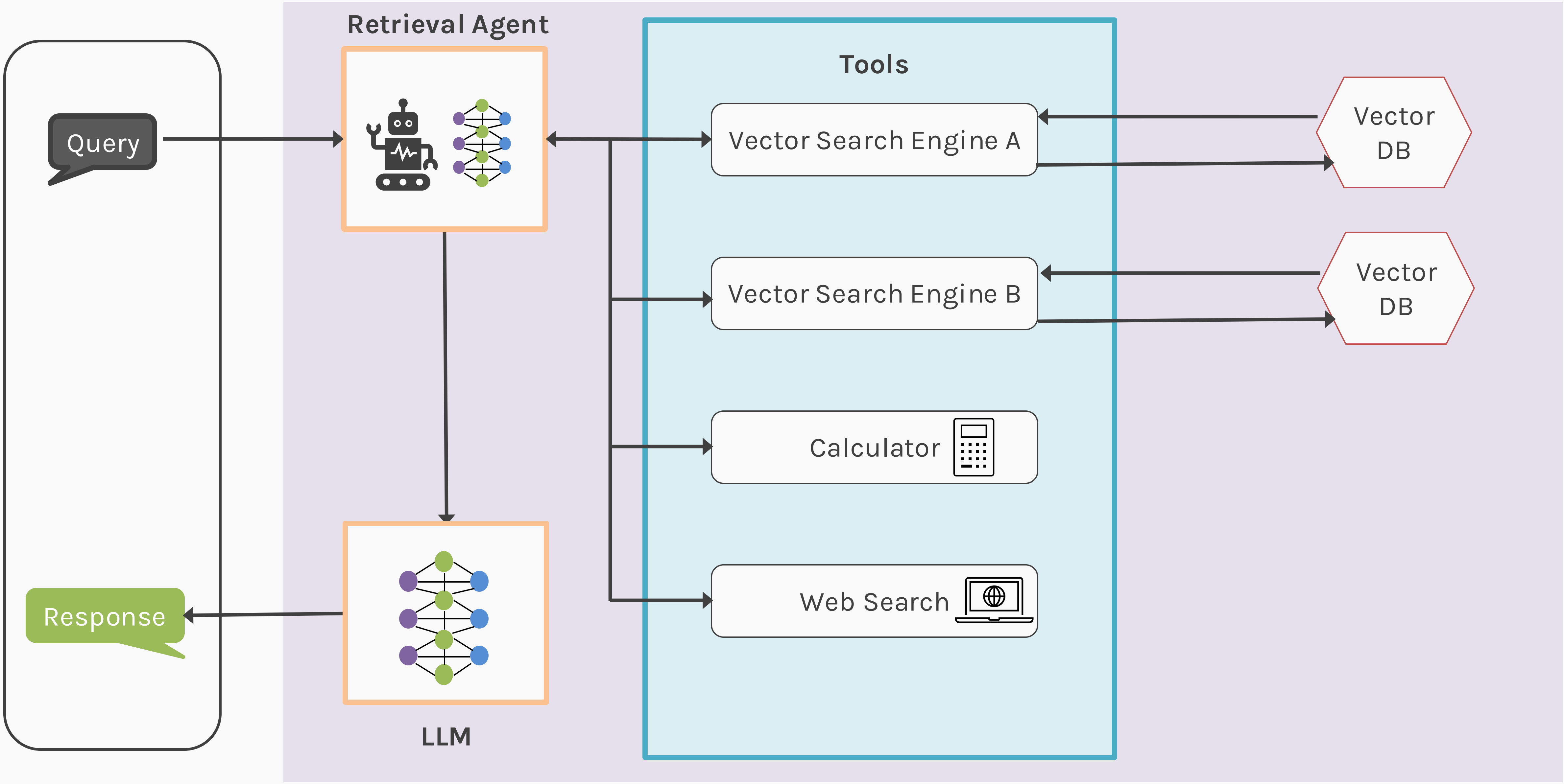
Retrieval Agent

This **agent** can assess the initial query, reformulate as needed, plan the process ahead, critique it and reiterate autonomously.

Agentic RAG


Query
Retrieval Agent
Tools
Vector Search Engine A
Vector Search Engine B
Calculator
Web Search
Vector DB
Vector DB

Agentic RAG


Query
Retrieval Agent
Tools
Vector Search Engine A
Vector DB
Vector Search Engine B
Vector DB
Calculator
Web Search
Response
LLM

Outline

- Recap
- Motivation
- Introduction
- Components
- **Overview**

Overview

	Level 1: Output Decisions	Access to real world data	**Level 2:** Task Decisions	**Level 3:** Process Decisions
	Ability to make **decisions** based on natural language	(**tool** dependent)	Can **choose** which tasks and tools to execute	Can **create** new tasks and tools to execute
LLMs	✅	❌	❌	❌

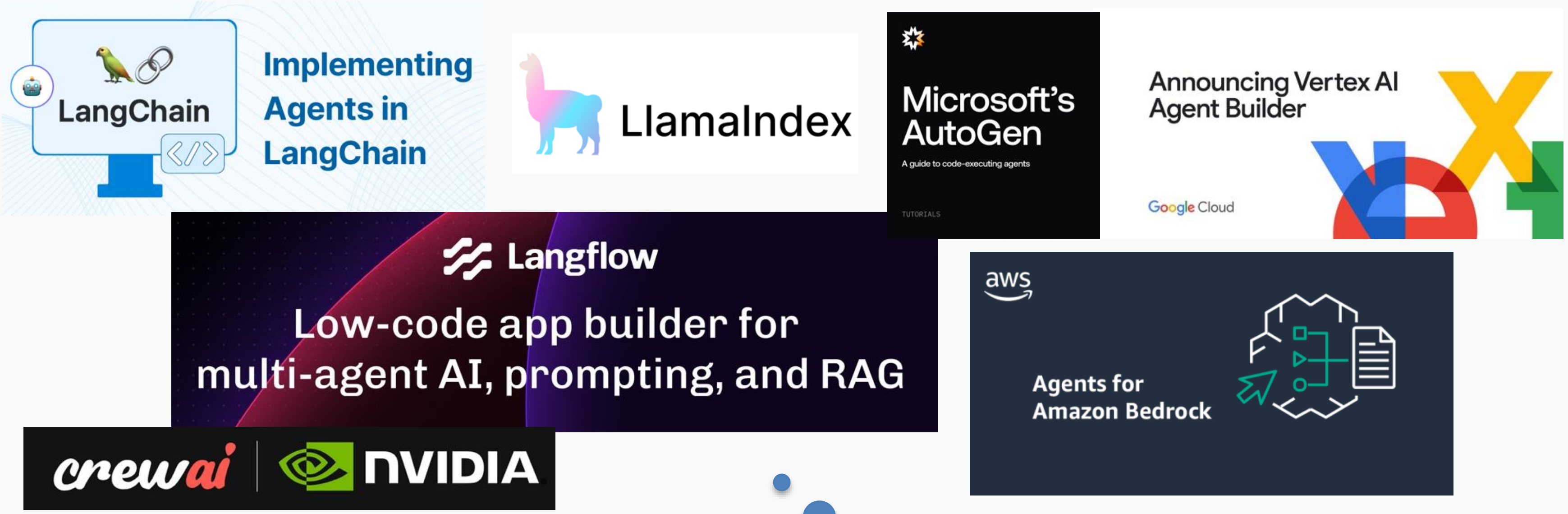
Overview

	Level 1: Output Decisions	Access to real world data	**Level 2:** Task Decisions	**Level 3:** Process Decisions
	Ability to make **decisions** based on natural language	(**tool** dependent)	Can **choose** which tasks and tools to execute	Can **create** new tasks and tools to execute
LLMs	✅	❌	❌	❌
Agentic Setup	✅	✅	✅	❌

Overview

	Level 1: Output Decisions	Access to real world data	**Level 2:** Task Decisions	**Level 3:** Process Decisions
	Ability to make **decisions** based on natural language	(**tool** dependent)	Can **choose** which tasks and tools to execute	Can **create** new tasks and tools to execute
LLMs	✅	❌	❌	❌
Agentic Setup	✅	✅	✅	❌
Autonomous Agent	✅	✅	✅	✅

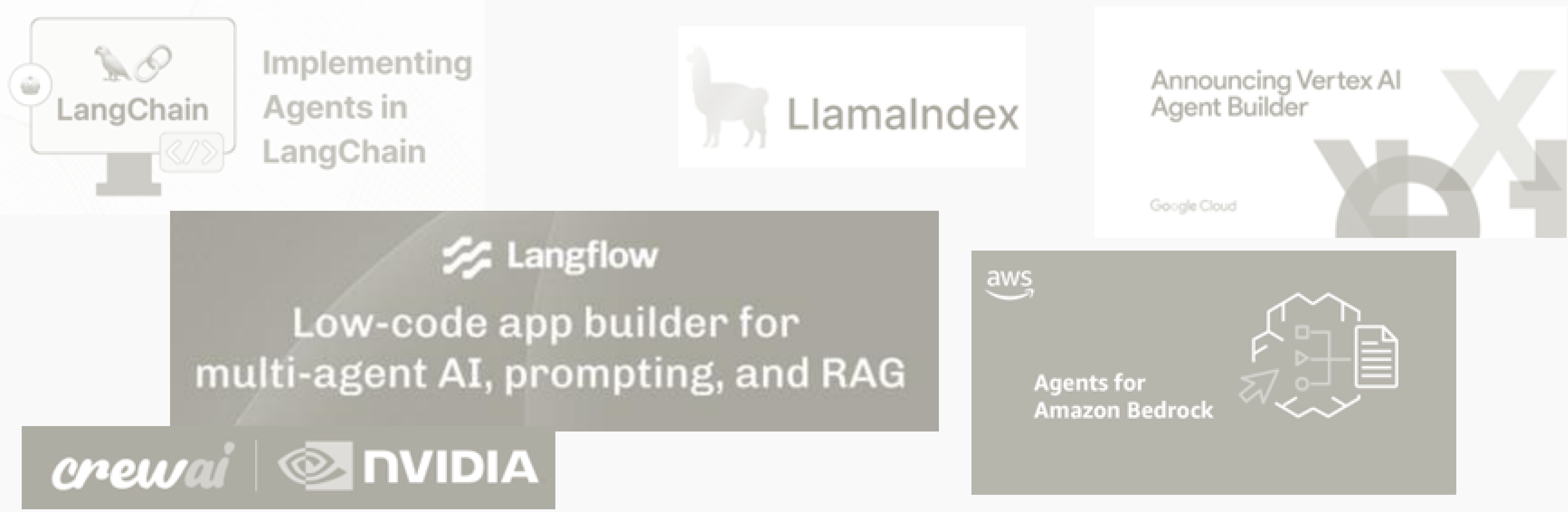
Overview: Examples

Agentic Setup (Frameworks)	
Autonomous Agent	

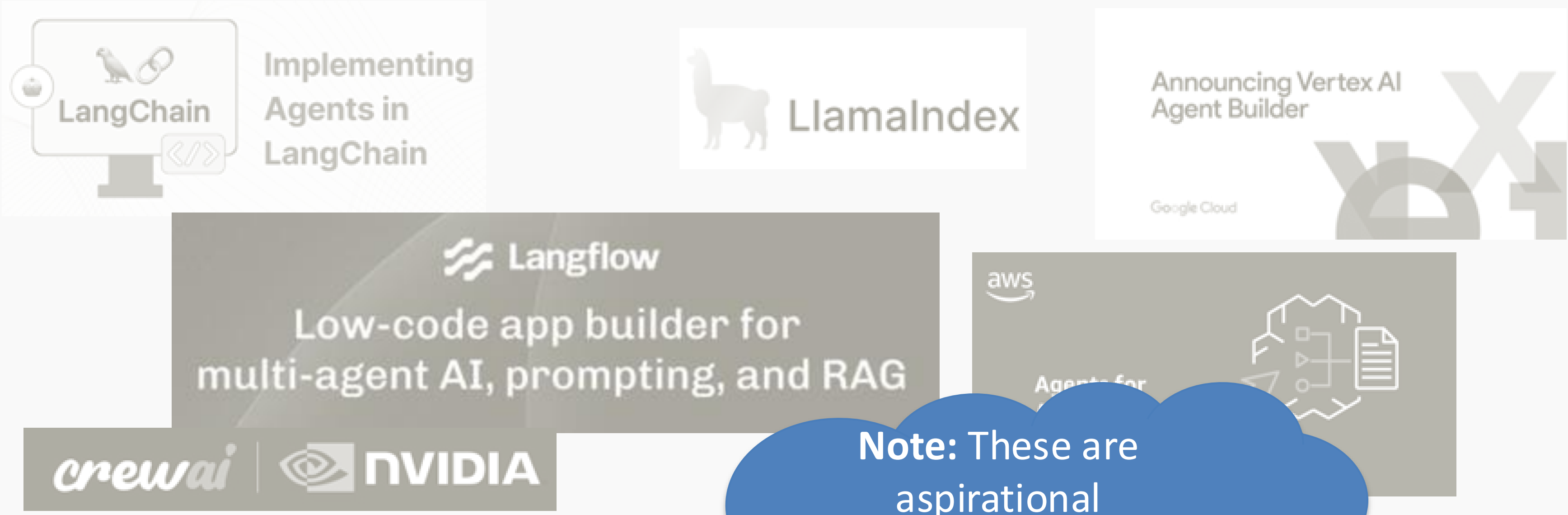
LangChain
Implementing Agents in LangChain
LlamaIndex
Microsoft's AutoGen
A guide to code-executing agents
TUTORIALS
Announcing Vertex AI Agent Builder
Google Cloud
Langflow
Low-code app builder for multi-agent AI, prompting, and RAG
aws
Agents for Amazon Bedrock
crewai
NVIDIA

These frameworks can be employed to setup your own agents.

Overview: Examples

Agentic Setup (Frameworks)	 LangChain Implementing Agents in LangChain LlamaIndex Announcing Vertex AI Agent Builder Google Cloud Langflow Low-code app builder for multi-agent AI, prompting, and RAG aws Agents for Amazon Bedrock crewai NVIDIA
Autonomous Agent	 Devin Ai The World's First Fully Autonomous AI Software Engineer METAGPT: META PROGRAMMING FOR A MULTI-AGENT COLLABORATIVE FRAMEWORK yoheinakajima/ **babyagi** 80 Contributors 3 Used by 21k Stars 3k Forks

Overview: Examples

Agentic Setup (Frameworks)	 LangChain Implementing Agents in LangChain LlamaIndex Announcing Vertex AI Agent Builder Google Cloud Langflow Low-code app builder for multi-agent AI, prompting, and RAG aws Agents for crewai NVIDIA
Autonomous Agent	Devin Ai The World's First Fully Autonomous AI Software Engineer 80 Contributors 3 Used by 21k Stars 3k Forks METAGPT: META PROGRAMMING FOR A MULTI-AGENT COLLABORATIVE FRAMEWORK

Note: These are aspirational autonomous agents, none are completely autonomous in the true sense yet!

Thank you!