

Building LLM Agents

Challenges and Learnings

Felix Reinfurt

Dr. Dan Zecha

TNG Big Techday 2024

Weekend Planner Agent

☀️ Planning a weekend in the city? I'm here to help! ☀️

I can assist with:

- 🏛️ Finding attractions and sightseeing spots
- 🍴 Recommending and booking restaurants
- 🎫 Discovering events and activities
- 🗺️ Creating a personalized itinerary



I'm at a nice conference in Munich this Friday. My train back home leaves Saturday evening so I have time on Saturday during the day to visit the city.



Weather Forecast:

- Morning: 🌤️ Few clouds with temperatures around 19.59°C.
- Afternoon: ☀️ Clear skies with a high of 29.17°C.
- Evening: 🌧️ Moderate rain starting around 3 PM, with temperatures dropping to 21.61°C.

Morning Options:

- Option 1: Visit Marienplatz - the central square in Munich, famous for the New Town Hall and its Glockenspiel. Enjoy a coffee at one of the nearby cafes.
 - 📍 Address: Marienplatz, 80331 München, Germany
 - 🕒 Opening Hours: Open 24 hours
 - 💰 Entry Fee: Free
- Option 2: Explore the English Garden - one of the world's largest urban parks, featuring sprawling lawns, riverside paths, and the iconic Chinese Tower beer garden.
 - 📍 Address: 80538 Munich, Bavaria, Germany
 - 🕒 Opening Hours: 24 hours daily
 - 💰 Entry Fee: Free

**



Type your message here...



What is an LLM Agent? ○



How do LLM Agents work? ○



Agentic Design Patterns ○



Applications of LLM
Agents ○



Challenges and
Learnings ○





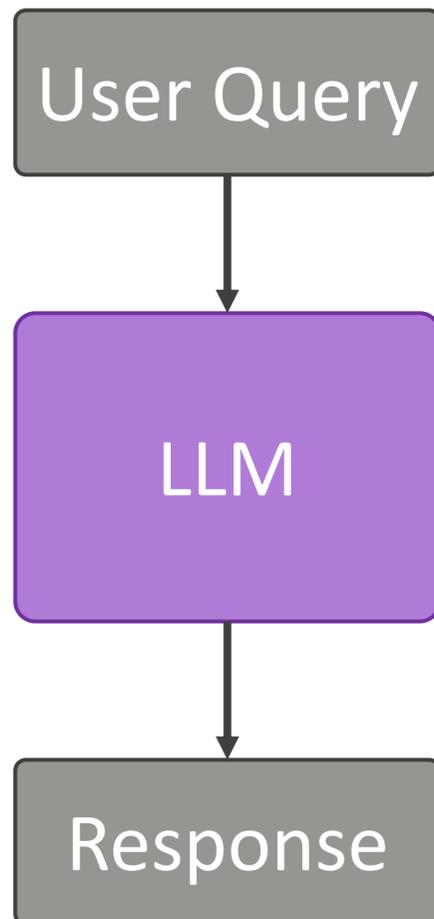
What is an LLM Agent?

Agents are applications that can execute **complex tasks** by combining key modules like **planning, reflection, memory, and tool use.**

LLM (based) agents are applications that can execute **complex tasks** by combining LLMs with key modules like **planning, reflection, memory, and tool use.**



How do LLM Agents
work?



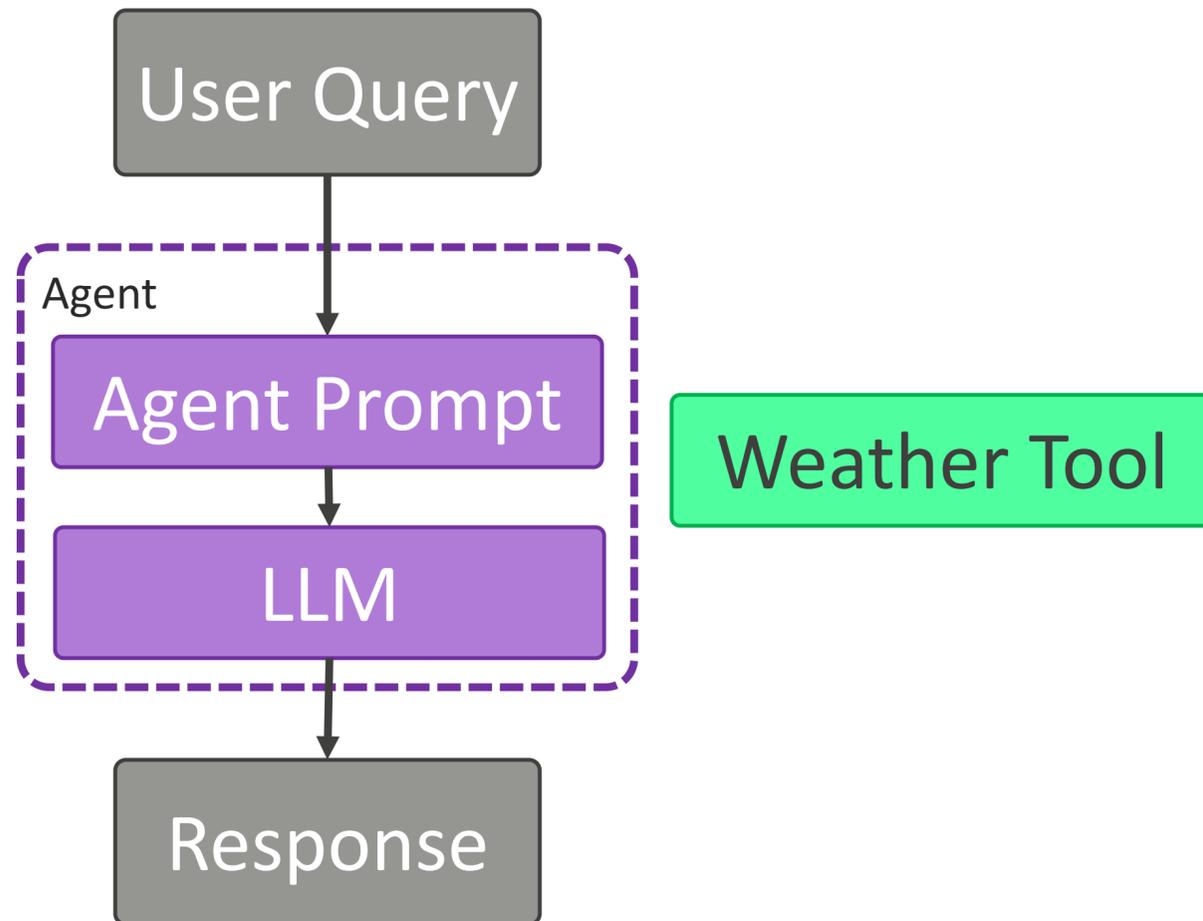
 **Agent without tool** 

🌟 This is simply a wrapper around GPT4 🌟

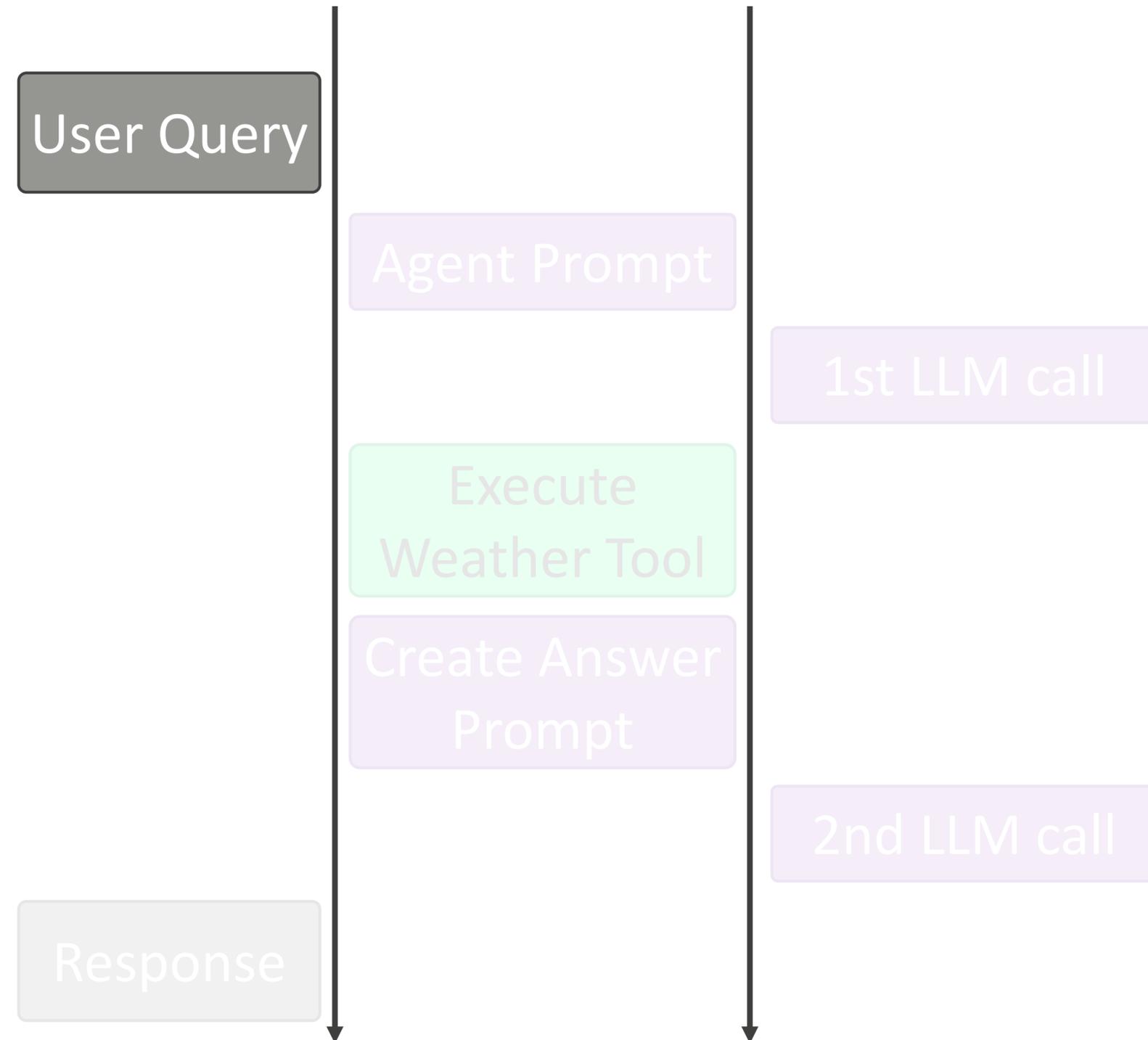


Type your message here...

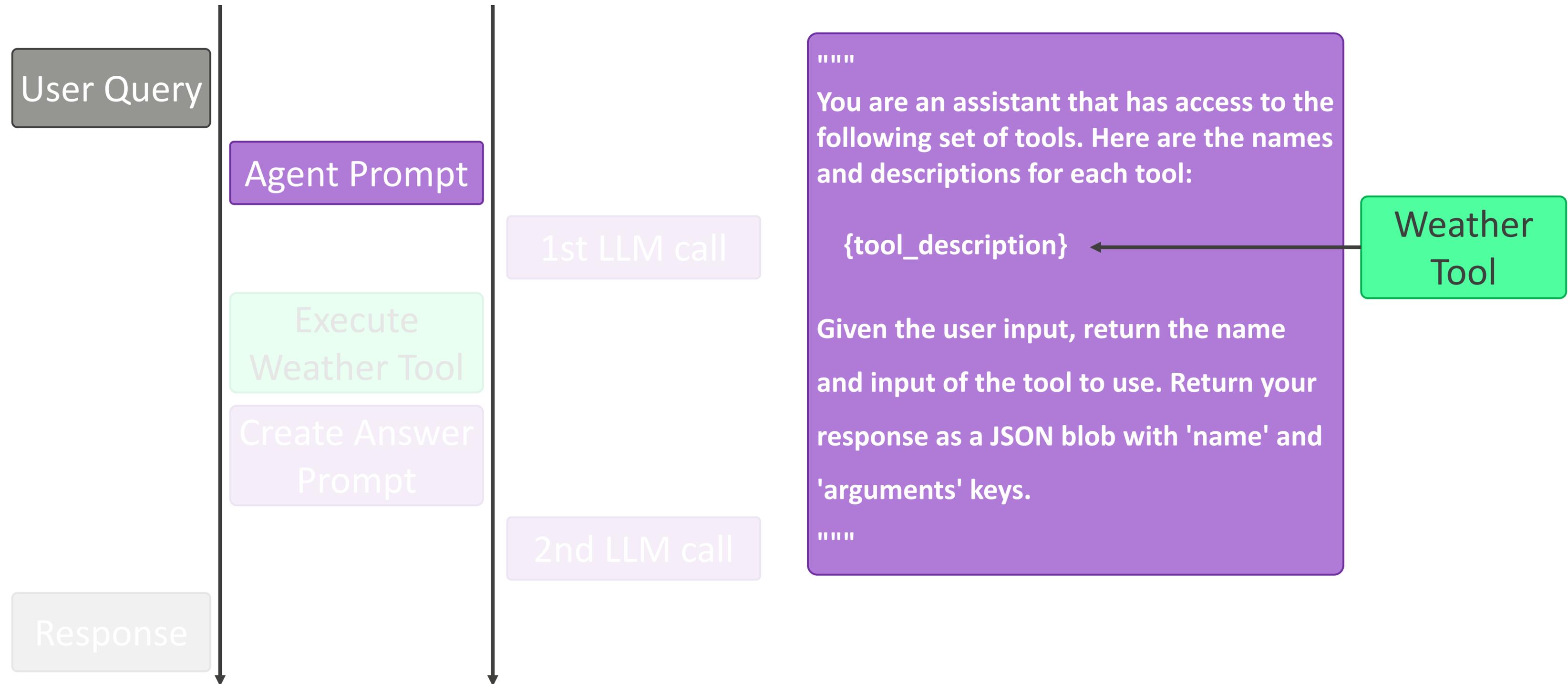




The screenshot shows a chat interface for an agent named "Agent with weather tool". The interface includes the TNG Technology Consulting logo, a "Chat" button, and a "Readme" link. A "New Chat" button and a user profile icon "U" are also visible. The chat content shows a system message: "🤖 Agent with weather tool 🤖" followed by "🌟 This is a wrapper around GPT4 with a weather tool 🌟". At the bottom, there is a text input field with the placeholder "Type your message here..." and a send button. The footer indicates "Built with Chainlit".



"How's the weather in Munich?"



User Query

Agent Prompt

Execute
Weather Tool

Create Answer
Prompt

Response

You are an assistant that has access to the following set of tools. Here are the names and descriptions for each tool:

{tool_description}

Given the user input, return the name and input of the tool to use. Return your response as a JSON blob with 'name' and 'arguments' keys.

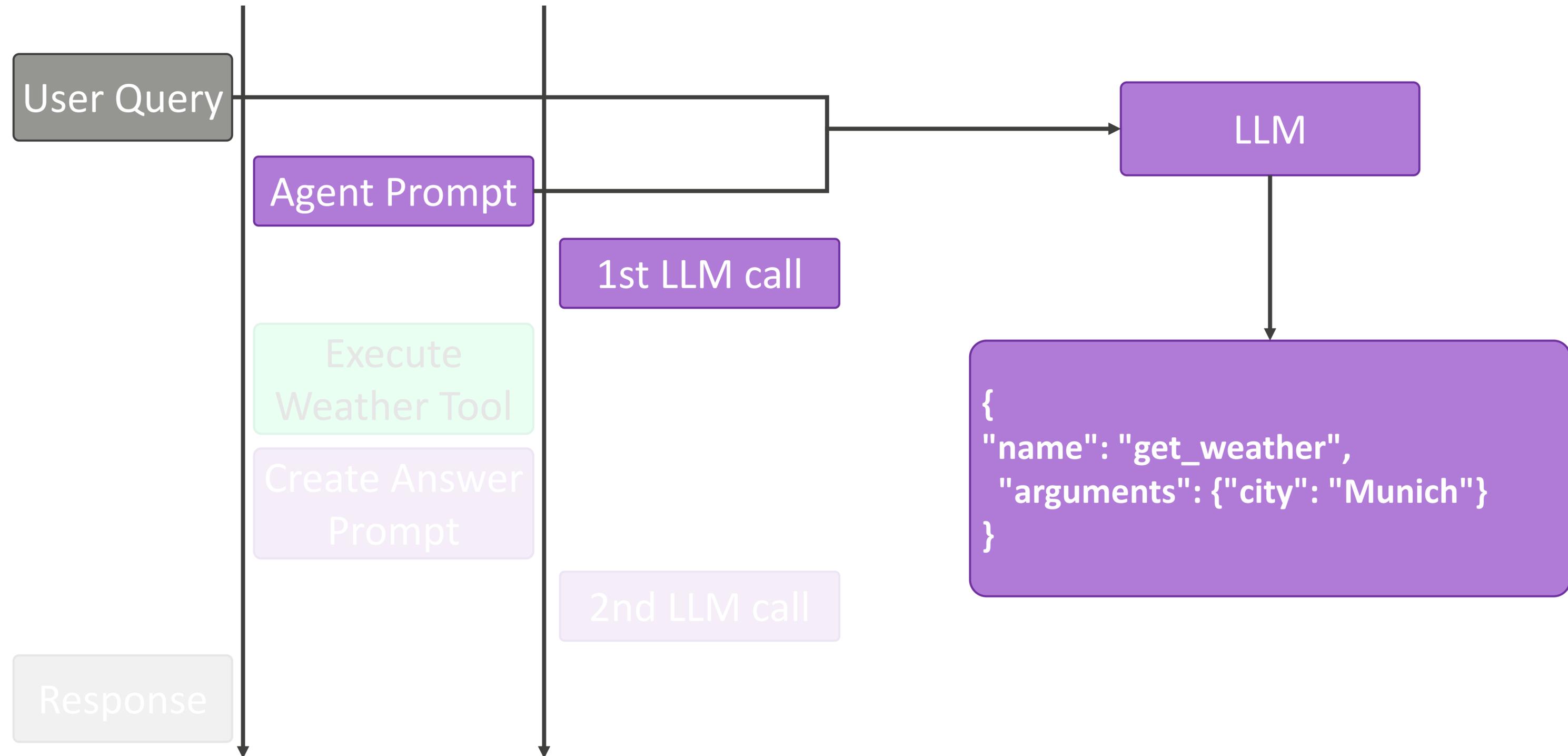
```

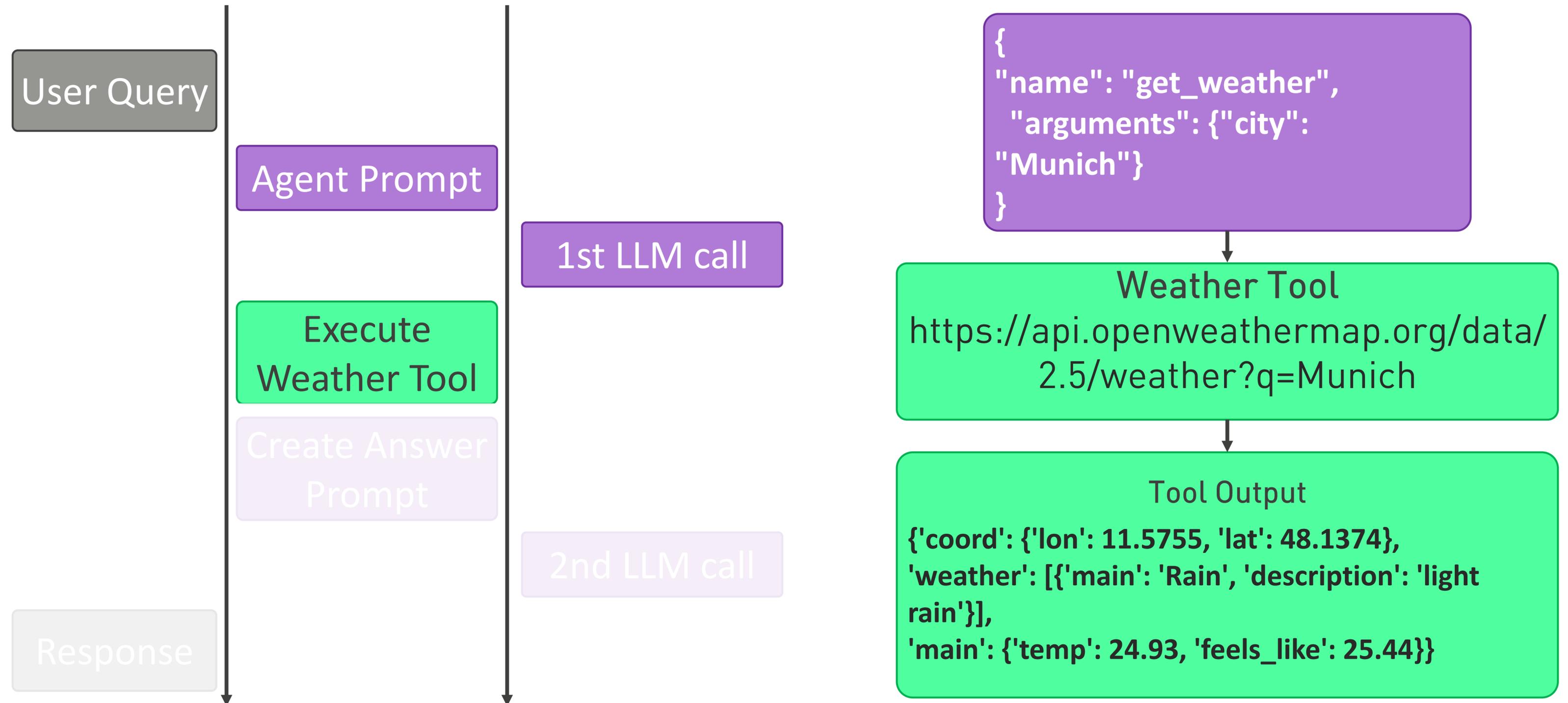
{ "type": "function",
  "function": {
    "description": "Fetches weather data from
    OpenWeatherMap API for a given city and returns it as
    JSON.",
    "name": "get_weather",
    "parameters": {
      "type": "object",
      "properties": {
        "city": {
          "type": "string",
          "description": "The name of the city to fetch weather
          data for."
        }
      },
      "required": ["city"]
    },
    "returns": {
      "description": "Weather data for the specified city in
      JSON format.",
      "type": "object"
    }
  }
}
  
```

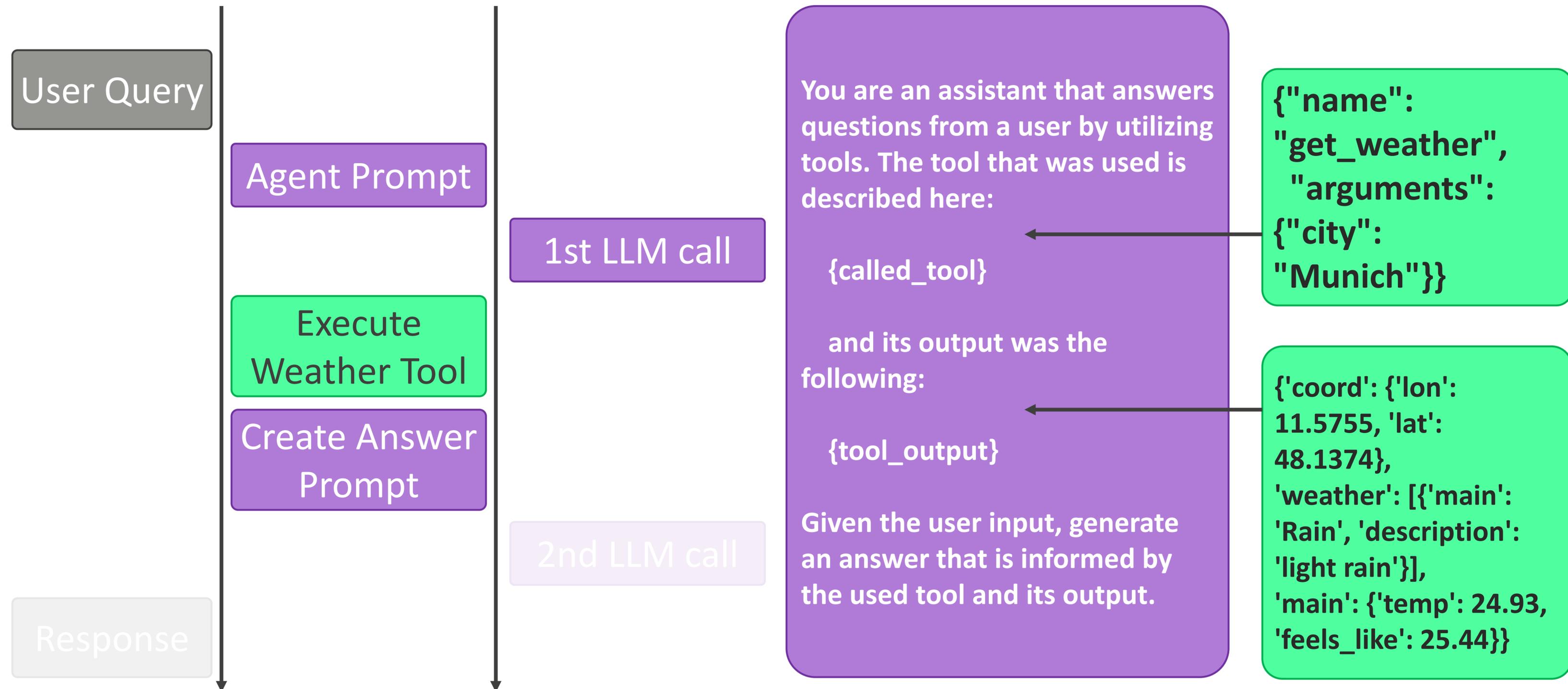
Description

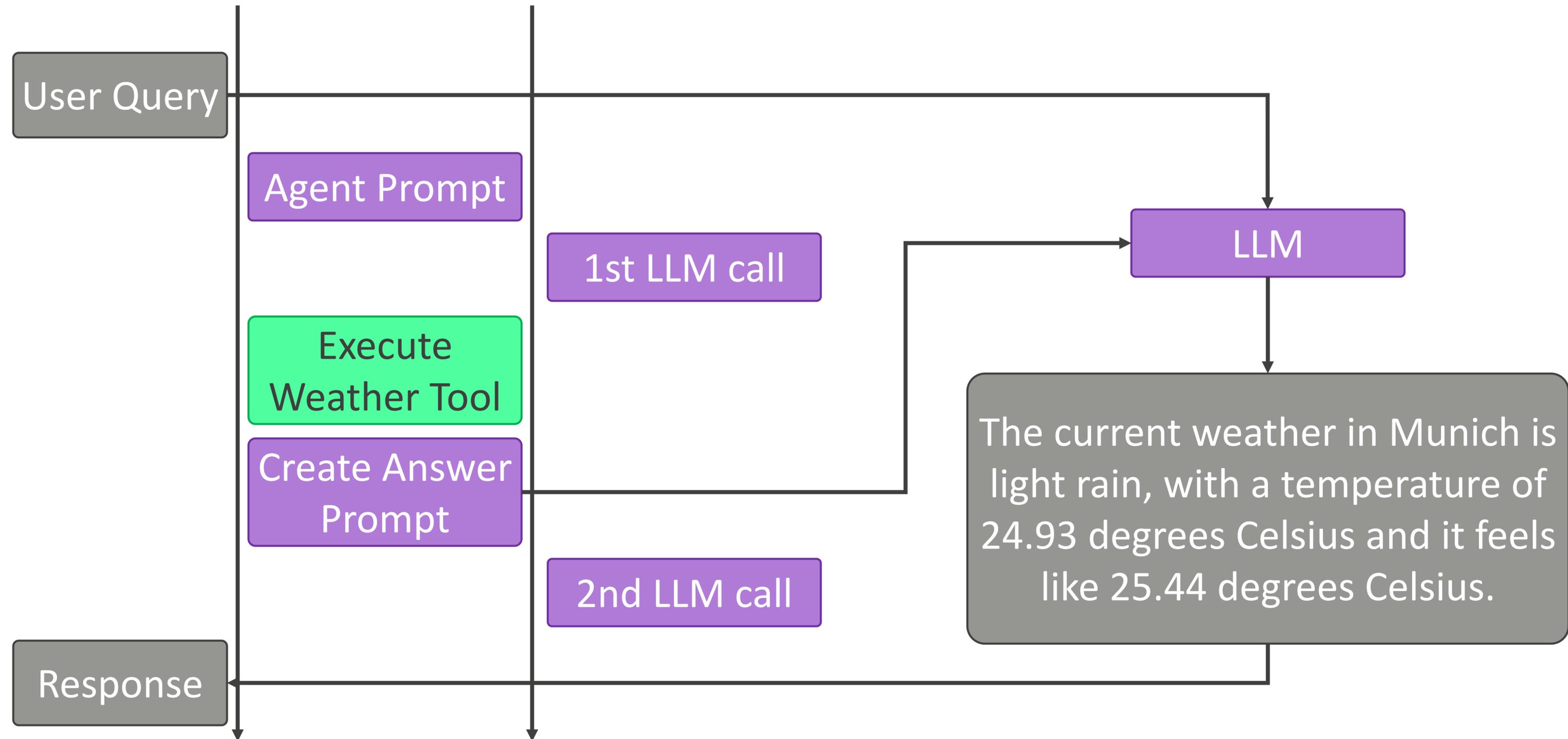
Parameters

Return Type



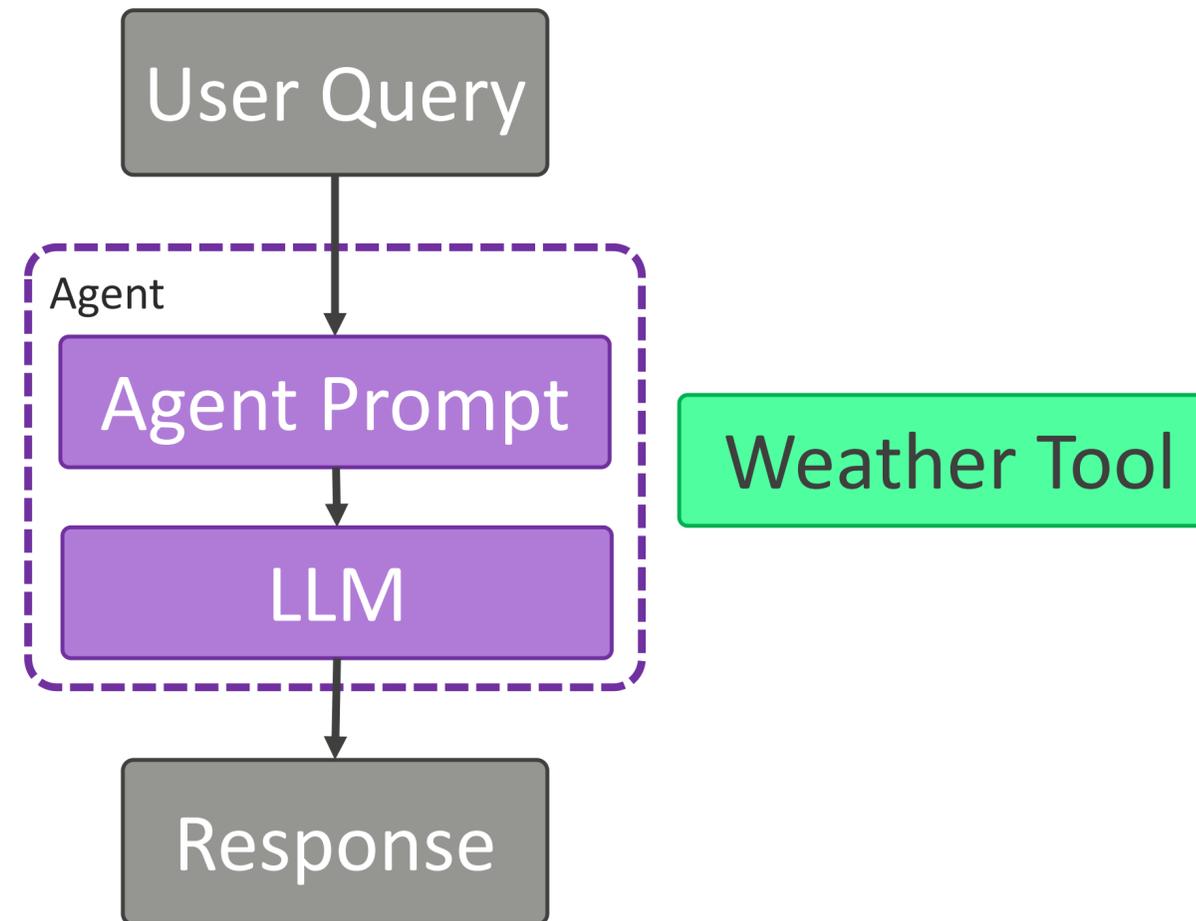


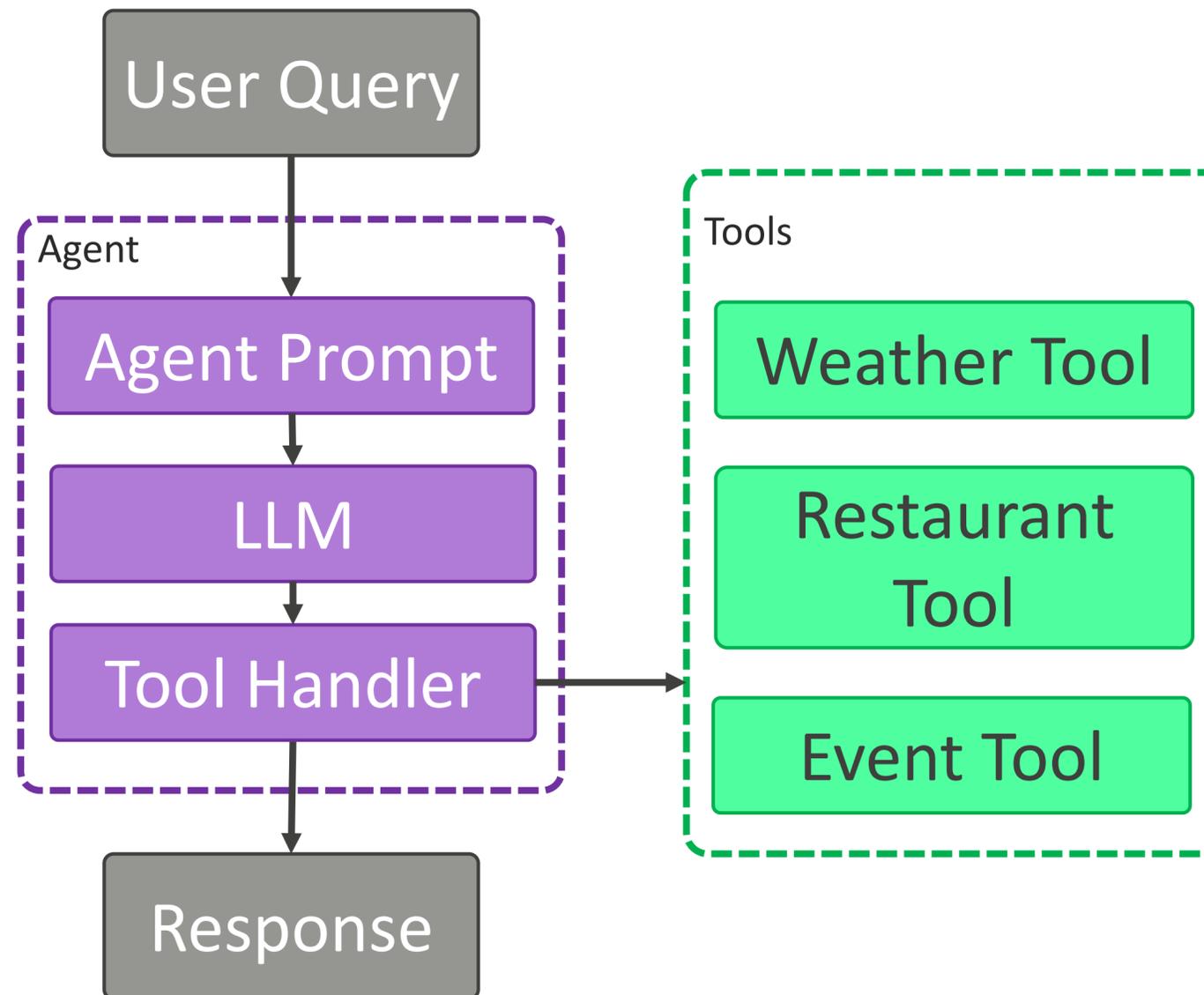


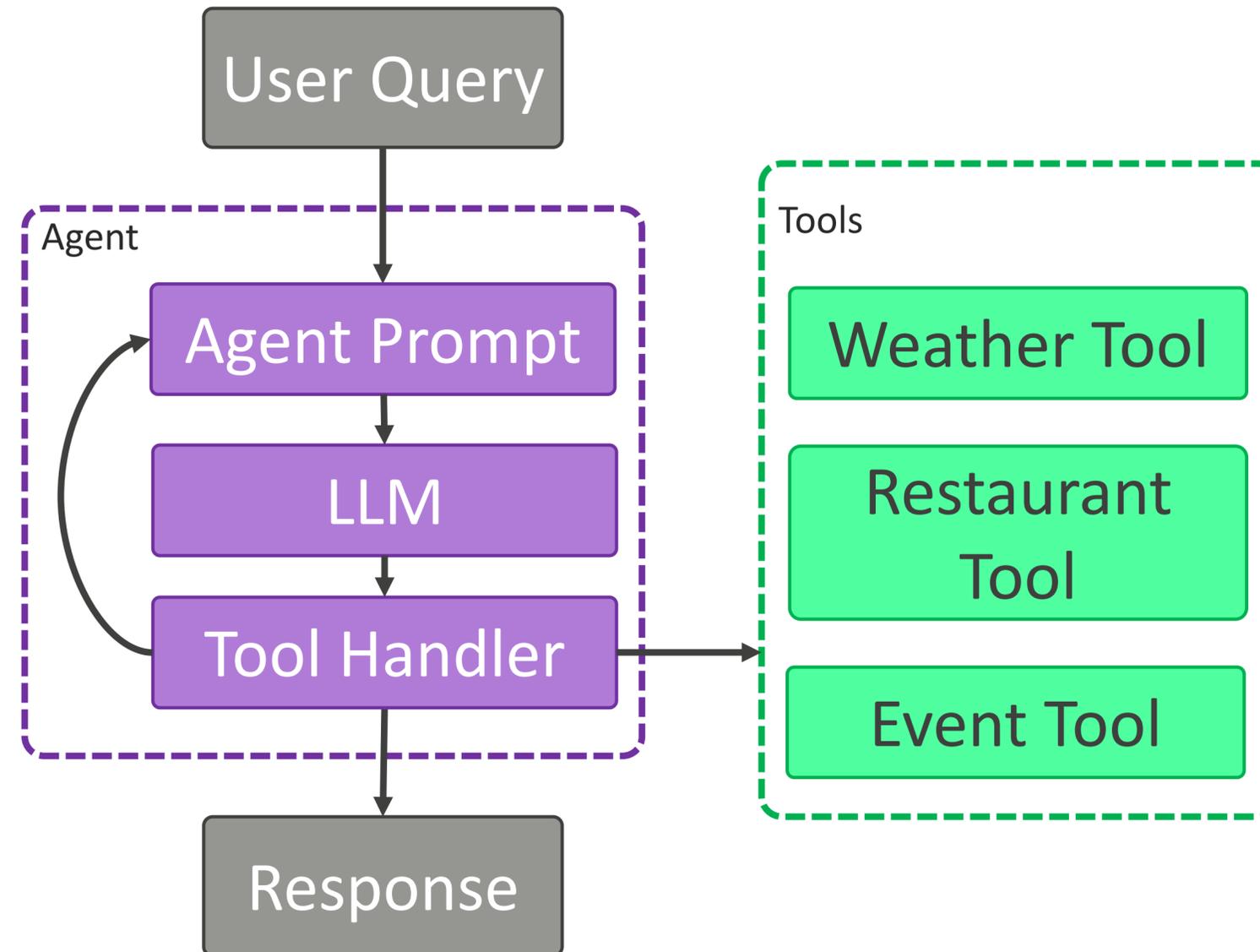


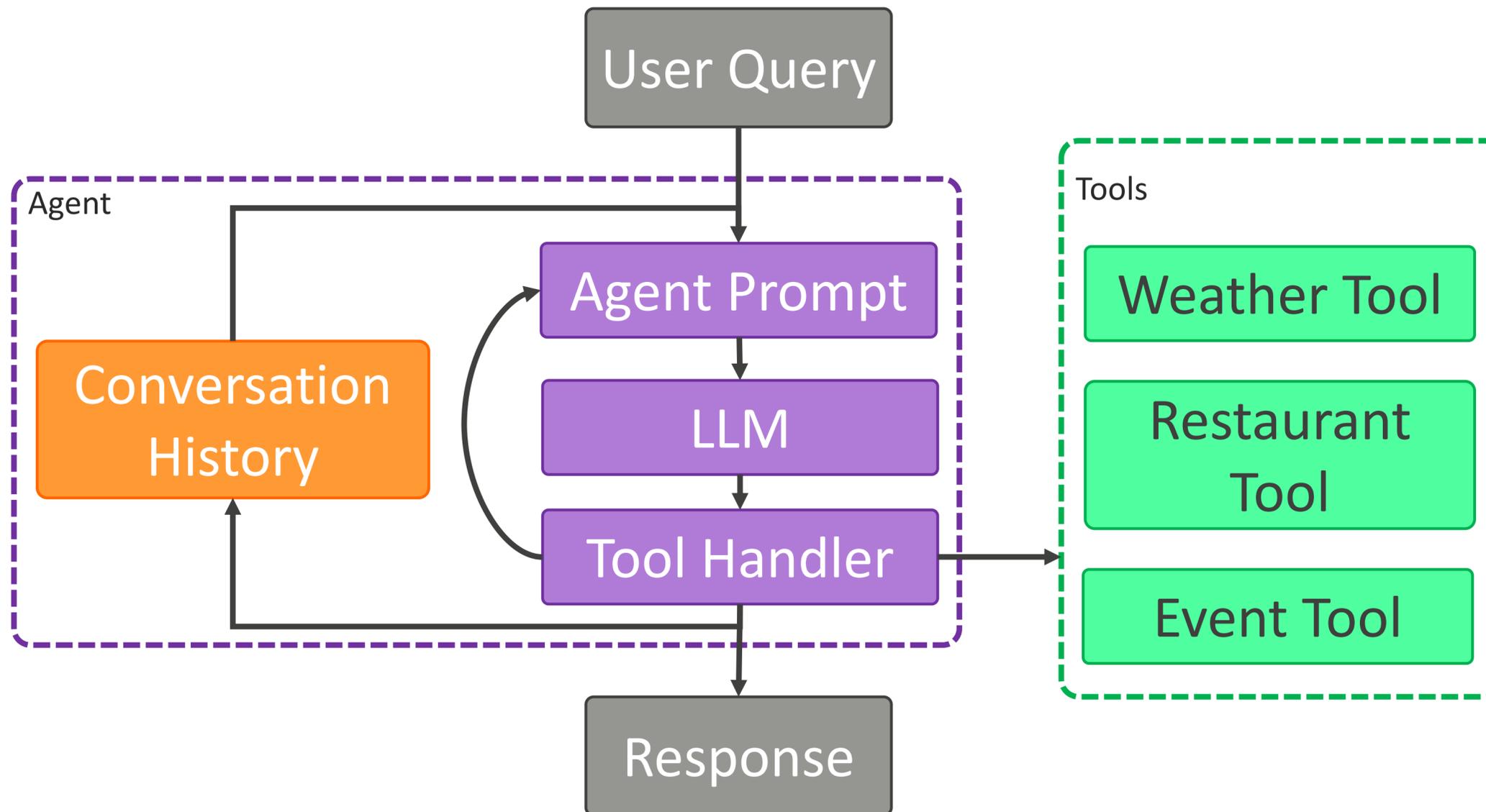


Agentic Design Patterns









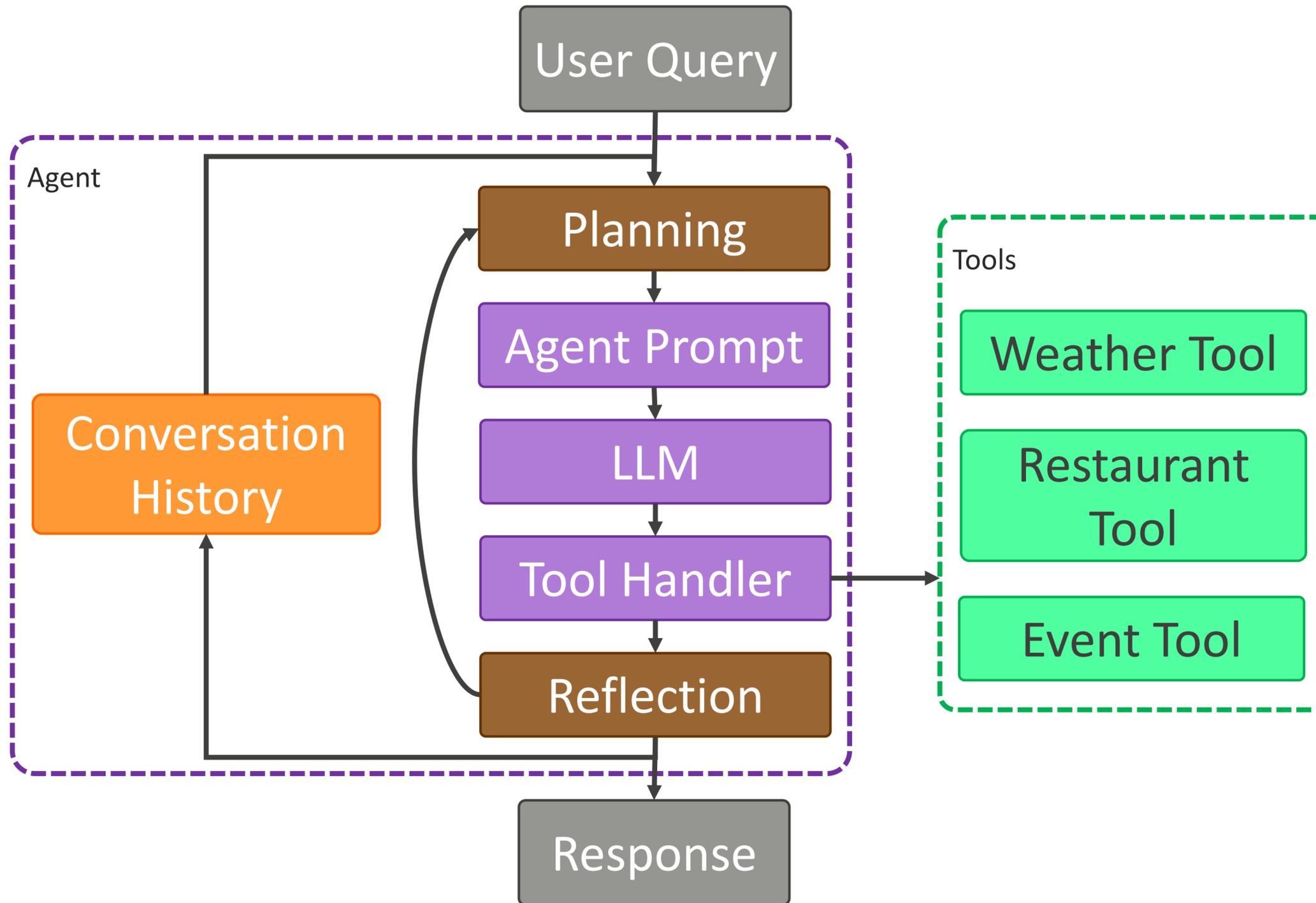
User Query

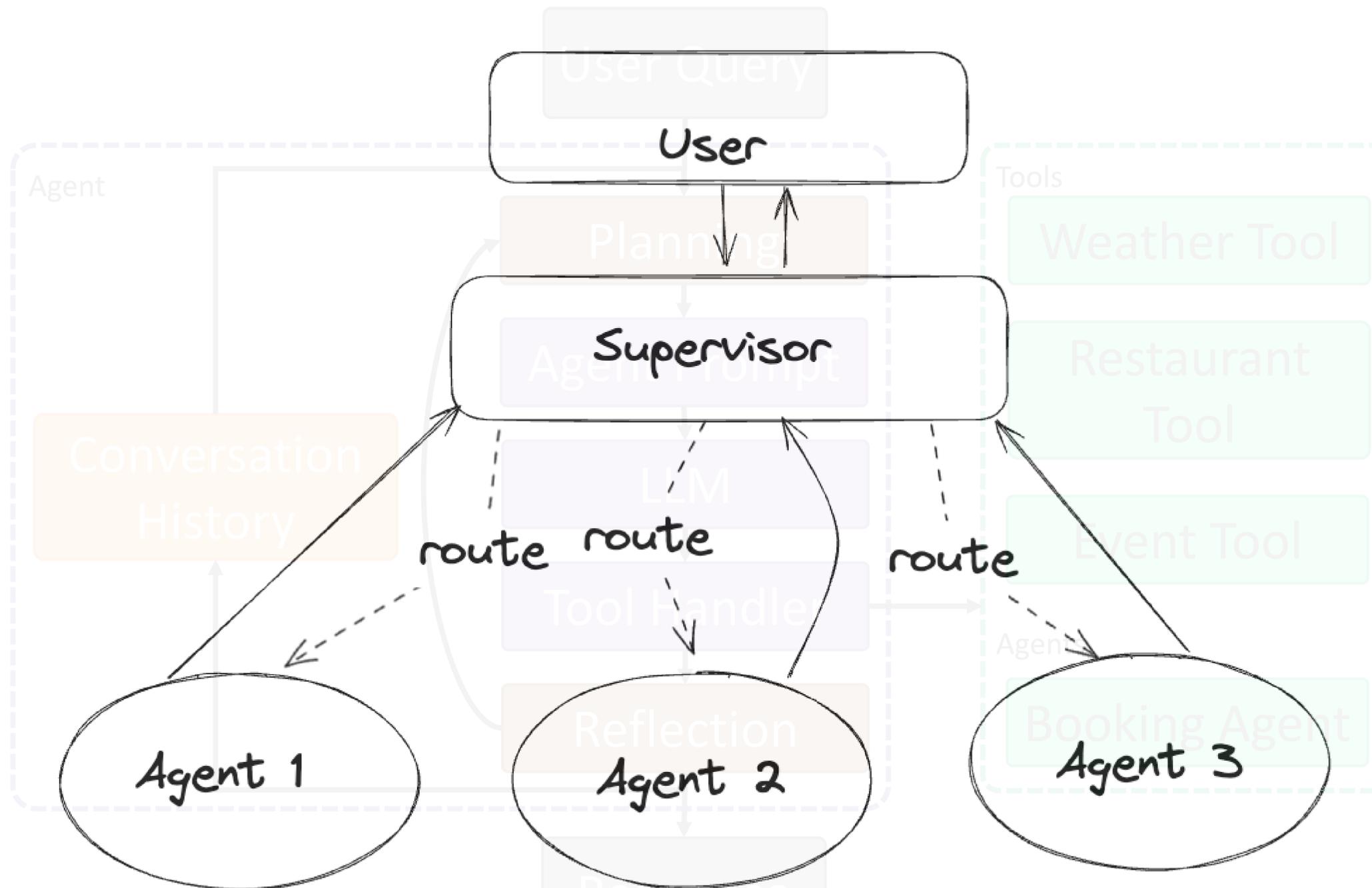
A **AI Assistant** 15:19

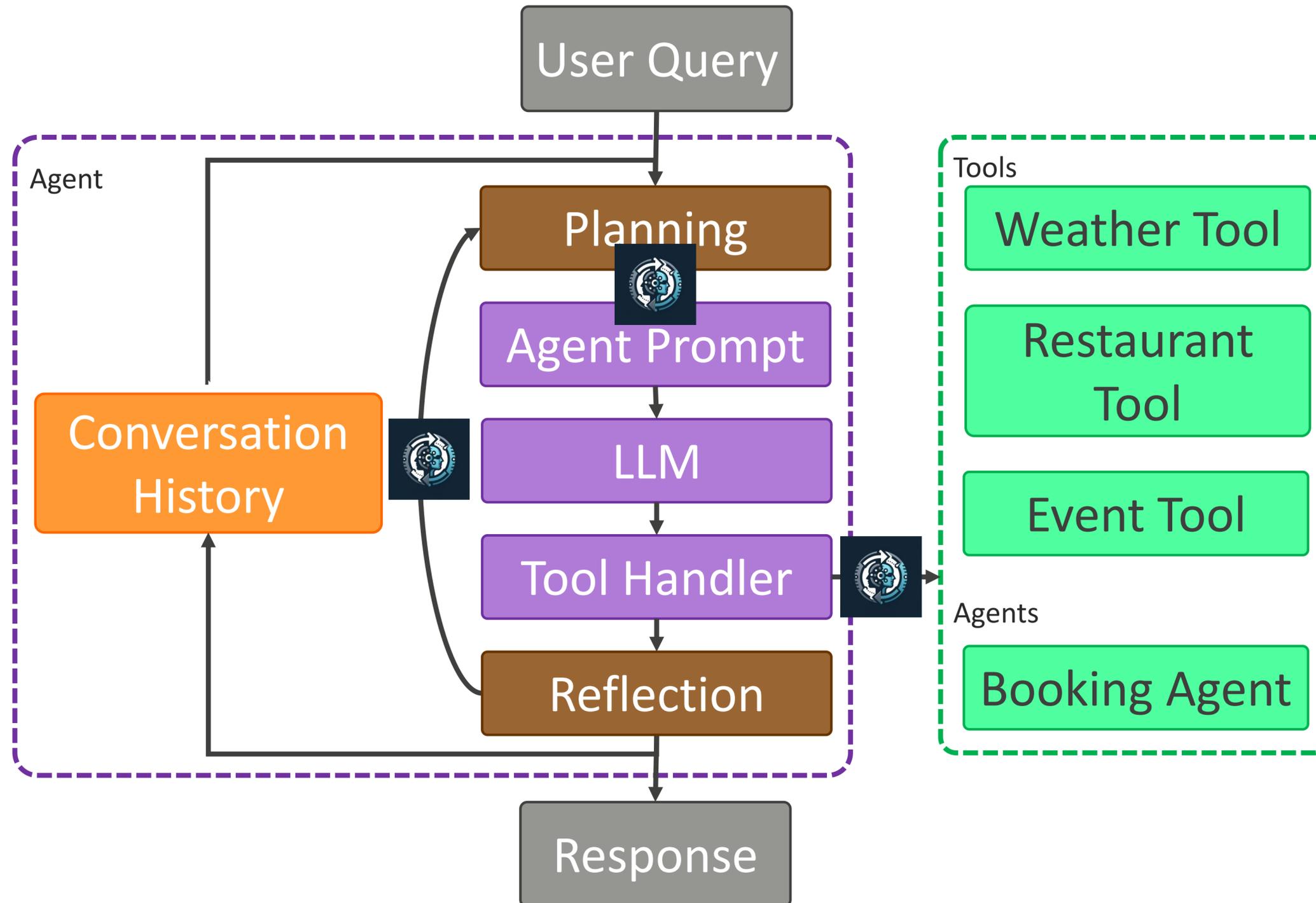
Fantastic! Let's plan a memorable Saturday in Munich for you. Here's the outline of what I'll do:

1. Check the Weather: I'll look up the weather forecast for Munich on Saturday to ensure our plans are weather-appropriate.
2. Morning Activities: Based on the weather and what Munich has to offer, I'll suggest two options for how to start your day.
3. Lunch Options: I'll find two great places for lunch to refuel and relax.
4. Afternoon Exploration: I'll propose two options for your afternoon, focusing on Munich's rich culture and beautiful sights.
5. Evening Plans: Since your train leaves in the evening, I'll suggest a light activity or place to visit before you head to the train station.
6. Transportation Tips: I'll provide some tips on getting around Munich efficiently during your stay.

Response



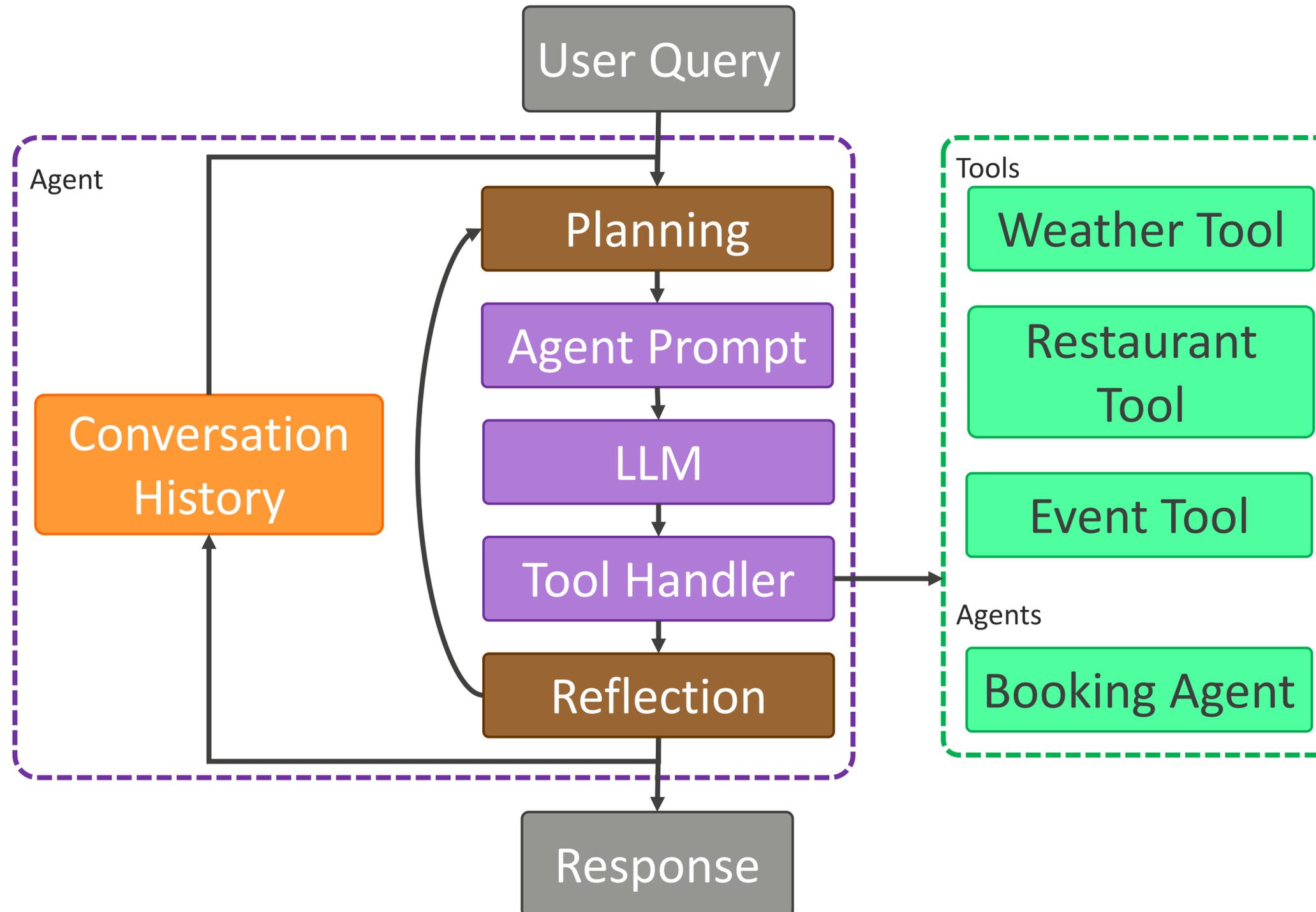


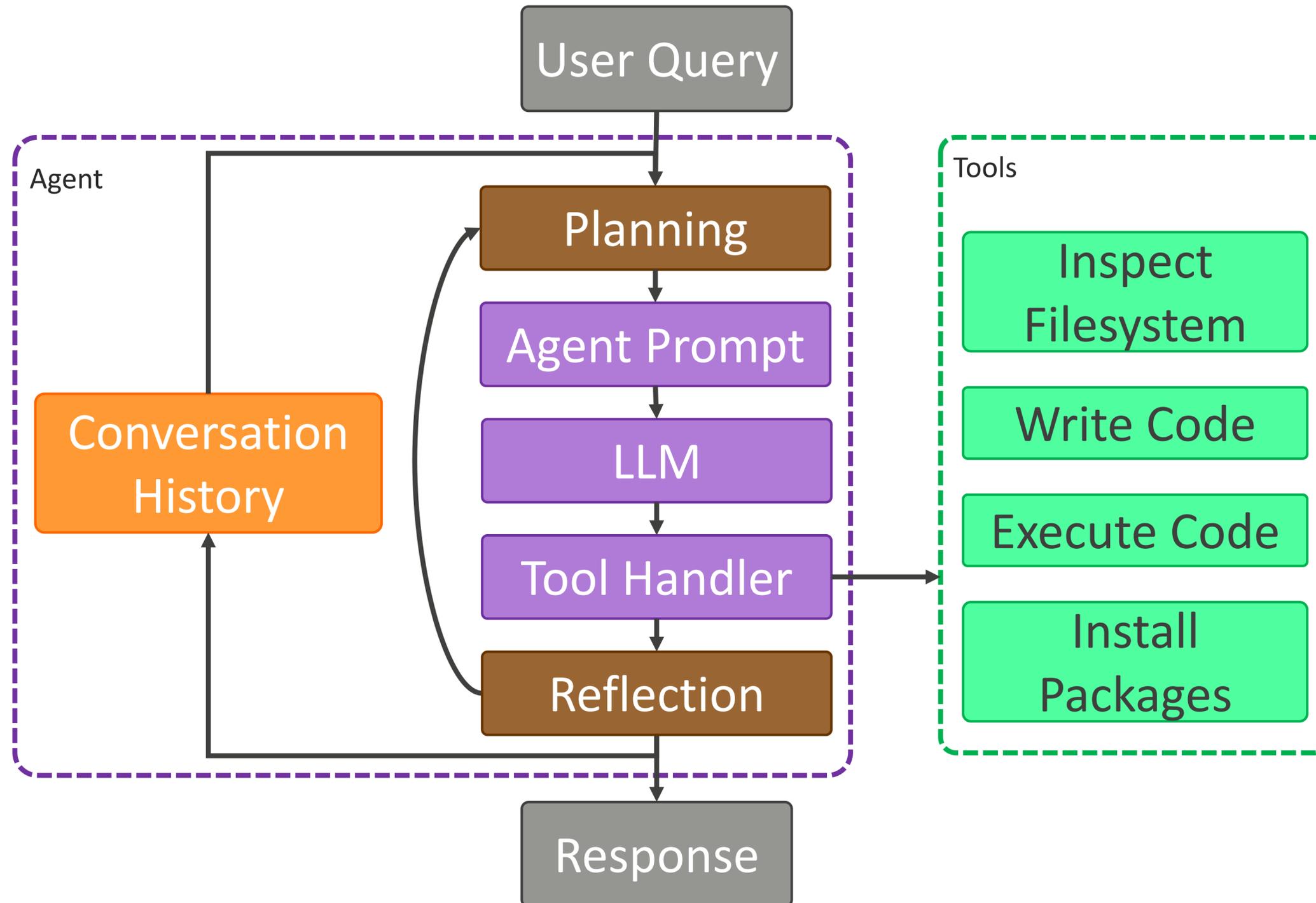


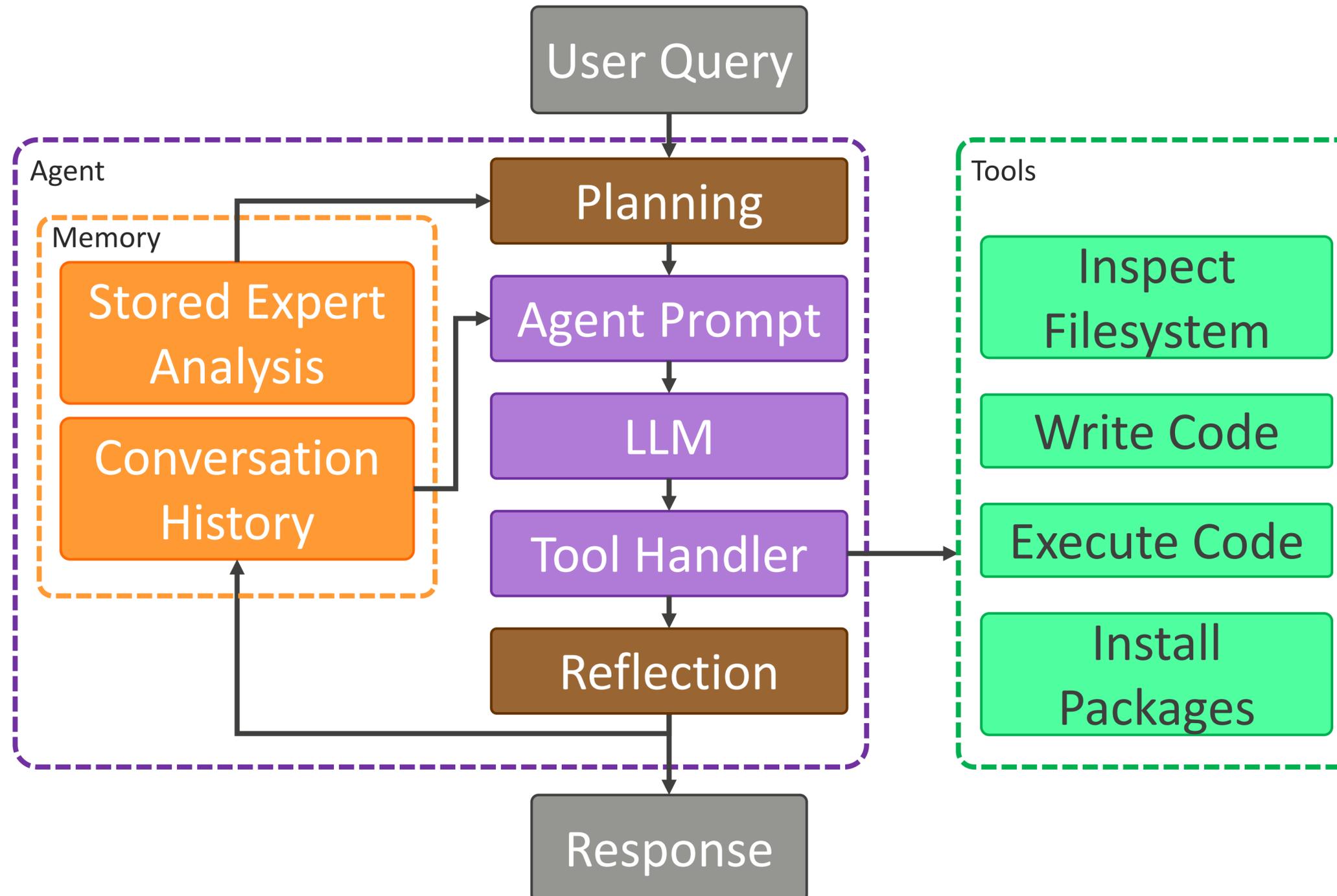
Applications of LLM Agents

Query:
 „Here is some data.
 Develop a regression model that
 predicts the data target well.“

	A	B	C	D	E	F	G	H	I	J	K
1	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	target
2	44	1	23.1	87	213	126.4	77	3	3.8712	72	52
3	22	1	23	73	161	97.8	54	3	3.8286	91	200
4	57	1	21.7	94	157	58	82	2	4.4427	92	87
5	43	1	18.5	87	163	93.6	61	2.67	3.7377	80	90
6	32	1	26.5	86	184	101.6	53	4	4.9904	78	258
7	50	1	31.8	82	136	69.2	55	2	4.0775	85	136
8	22	1	19.9	75	175	108.6	54	3	4.1271	72	158
9	56	2	28	85	184	144.8	32	6	3.5835	77	69
10	54	1	22.6	90	183	104.2	64	3	4.3041	92	72
11	34	2	24.7	118	254	184.2	39	7	5.037	81	171
12	41	1	32.4	94	171	104.4	56	3	3.9703	76	95
13	42	1	22.1	85	213	138.6	60	4	4.2767	94	72
14	59	2	32.1	101	157	93.2	38	4	4.8598	87	151
15	41	1	24.7	83	187	108.2	60	3	4.5433	78	168
16	38	1	21.3	72	165	60.2	88	2	4.4308	90	60
17	42	1	29.2	93	249	174.2	45	6	5.0039	92	122
18	37	1	30.2	87	166	96	40	4.15	5.0106	87	52
19	32	2	31.4	89	153	84.2	56	3	4.1589	90	187







Eingabeaufforderung

```
(venv) C:\Users\Dan\Desktop\DSA\agent>python main.py|
```

agent_human_plan

Datei Start Freigeben Ansicht

DSA... agent_hu... agent_hu...



test_data.csv

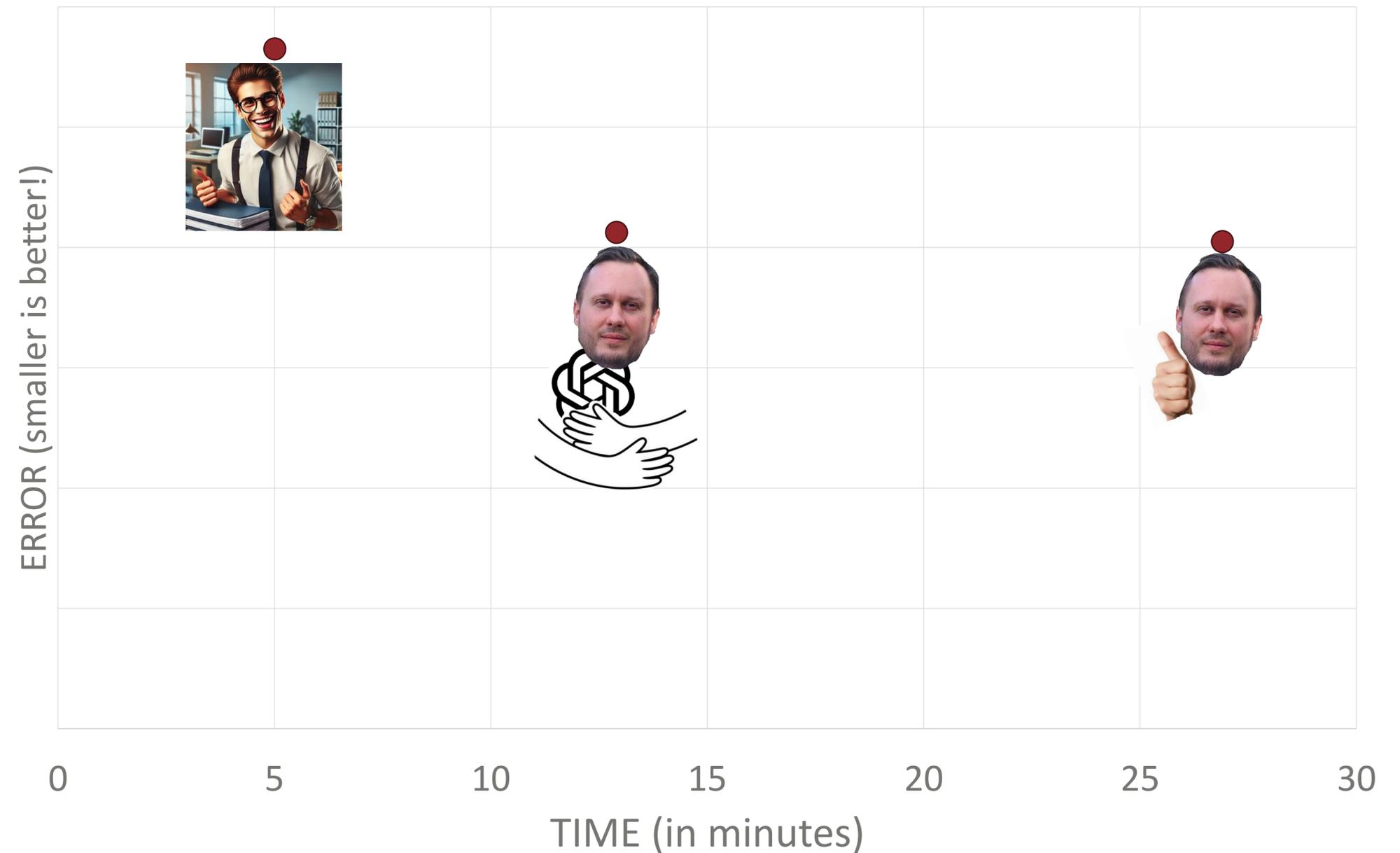


train_data.csv

2 Elemente |

Rules of Engagement

- Start on a green field
- Human: only Google, no LLMs (at first)
- Task: Develop a regression model
- Stop criteria: happy with result or no further improvement



Coding/Debugging

Search

Personal & Educational Agents

Robots

cognition

An applied AI lab building end-to-end software agents.

We're building collaborative AI teammates that enable engineering teams to strive for more ambitious goals. We're the makers of Devin, the first AI software engineer.

[Get started with Devin](#) →

smol developer

Deploy agent with **e2b**

Human-centric & Coherent Whole Program Synthesis aka your own personal junior developer

[Build the thing that builds the thing!](#) a `smo1 dev` for every dev in every situation

This is a "junior developer" agent (aka `smo1 dev`) that either:

1. scaffolds an entire codebase out for you once you give it a product spec
2. gives you basic building blocks to have a smol developer inside of your own app.

SWE-agent

Agent Computer Interfaces Enable Software Engineering Language Models

John Yang*, Carlos E. Jimenez*, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, Ofir Press

Princeton University

[Code](#) [Data](#) [Paper](#)

GPTEngineer

Chat to build real web apps. Really fast.

Enter your email

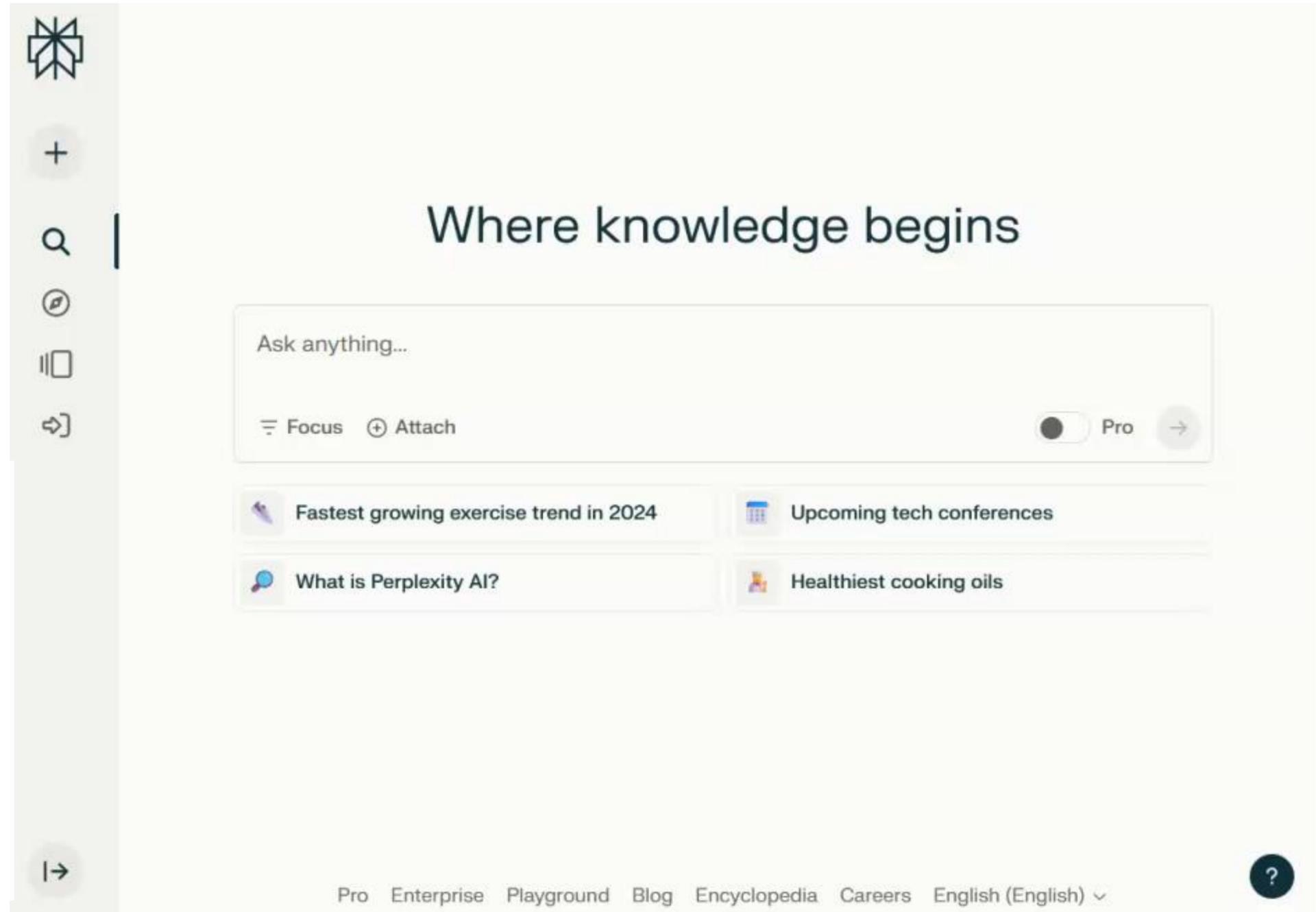
[Get Access](#)

Coding/Debugging

Search

Personal & Educational Agents

Robots



The screenshot shows the Perplexity AI search interface. On the left is a vertical sidebar with icons for home, add, search, share, and other functions. The main content area has a header "Where knowledge begins" and a search input field with the placeholder text "Ask anything...". Below the input field are several search results, each with a small icon and a title:

- Fastest growing exercise trend in 2024
- Upcoming tech conferences
- What is Perplexity AI?
- Healthiest cooking oils

At the bottom of the interface, there is a navigation bar with links for "Pro", "Enterprise", "Playground", "Blog", "Encyclopedia", "Careers", and "English (English)". A question mark icon is visible in the bottom right corner.

Coding/Debugging

Search

Personal & Educational Agents

Robots

[Blog-Eintrag](#) / 2024 / Februar / 28    Übersetzen  183 Aufrufe

Meet Agent MAAI*: An LLM Assistant That Negotiates Meeting Requests

Angelegt von Dan Zecha, zuletzt geändert am 28. Februar 2024

TL;DR: As a learning experience, I delved into the question of how to design and build an LLM-powered agent that automatically negotiates meeting requests for me. This article covers a few thoughts about its design.

Retrieval Augmented Generation is so October 2023

The new kid on the block in 2024 AI application engineering is LLM-powered agents, aka AI assistants. While the concept of intelligent agents is not novel, the augmentation of agents with LLM capabilities promises interesting real-world applications. Allegedly, agents even make RAG better.



Coding/Debugging

Search

Personal & Educational Agents

Robots



Challenges, Learnings and Considerations

Agents are notoriously difficult to

- **debug.**

- Challenge:
 - Hallucinations lead to many failure cases.
- Learning:
 - Be aware!
 - Invest in guardrails.

Agents are notoriously difficult to

- **debug.**
 - Challenge:
 - Hallucinations lead to many failure cases.
 - Learning:
 - Be aware!
 - Invest in guardrailing.



Evaluation is as hard as it is important.

- Challenge:
 - Evaluation strategy is often not clear
 - Lack of proper benchmarks and tools.
- Learning:
 - No free lunch.

Evaluation is as hard as it is important.

- Challenge:
 - Evaluation strategy is often not clear
 - Lack of proper benchmarks and tools.
- Learning:
 - No free lunch.



Research & Demo VS. Going to Production

Break agentic workflow down

LLM only where necessary

Use constraints.

Deploy early on.

Gather feedback.

Build evaluation strategy.

Rinse and repeat.

Research & Demo VS. Going to Production

Break agentic workflow down.

LLM only where necessary

Use constraints.

Deploy early on.

Gather feedback.

Build evaluation strategy.

Rinse and repeat.

Research & Demo VS. Going to Production

Break agentic workflow down.

LLM only where necessary.

Use constraints.

Deploy early on.

Gather feedback.

Build evaluation strategy.

Rinse and repeat.

Research & Demo VS. Going to Production

Break agentic workflow down.

LLM only where necessary.

Use constraints.

Deploy early on.

Gather feedback.

Build evaluation strategy.

Rinse and repeat.

Research & Demo VS. Going to Production

Break agentic workflow down.

LLM only where necessary.

Use constraints.

Deploy early on.

Gather feedback.

Build evaluation strategy.

Rinse and repeat.

Research & Demo VS. Going to Production

Break agentic workflow down.

LLM only where necessary.

Use constraints.

Deploy early on.

Gather feedback.

Build evaluation strategy.

Rinse and repeat.

Research & Demo VS. Going to Production

Break agentic workflow down.

LLM only where necessary.

Use constraints.

Deploy early on.

Gather feedback.

Build evaluation strategy.

Rinse and repeat.

Research & Demo VS. Going to Production

Break agentic workflow down.

LLM only where necessary.

Use constraints.

Deploy early on.

Gather feedback.

Build evaluation strategy.

Rinse and repeat.

Questions?



Felix Reinfurt

Senior Consultant @ TNG Technology Consulting
felix.reinfurt@tngtech.com



Dr. Dan Zecha

Senior Consultant @ TNG Technology Consulting
dan.zecha@tngtech.com

