F

Building Agents in JavaScript

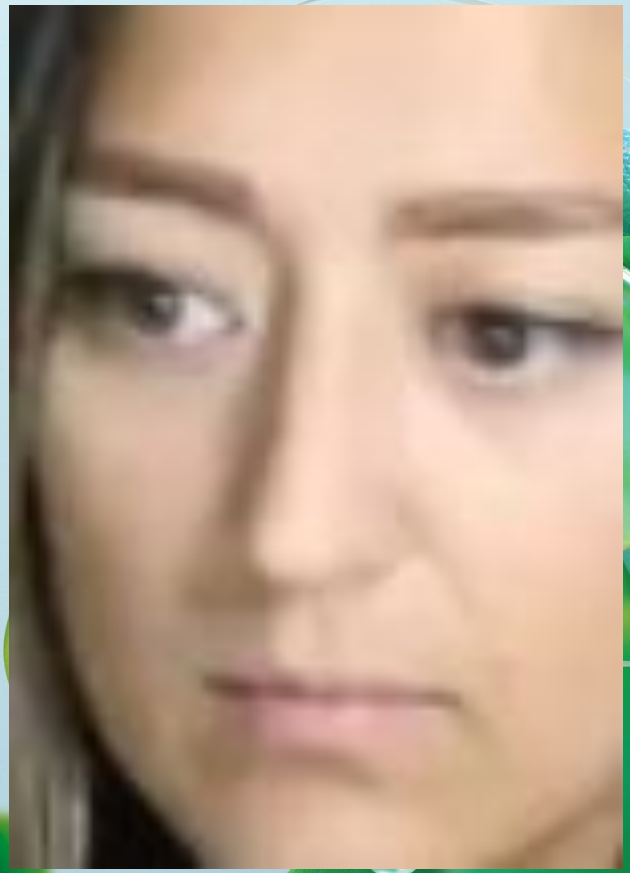
Daniel Phiri

@malgamves

DevEx at Weaviate

Writer, Developer, Artist.

Why I wrote this talk



Definitions

‘Agents’ have existed long before we had today’s performant LLMs.

We could even say that AI agents have existed for a long time too, just not as we know them today.

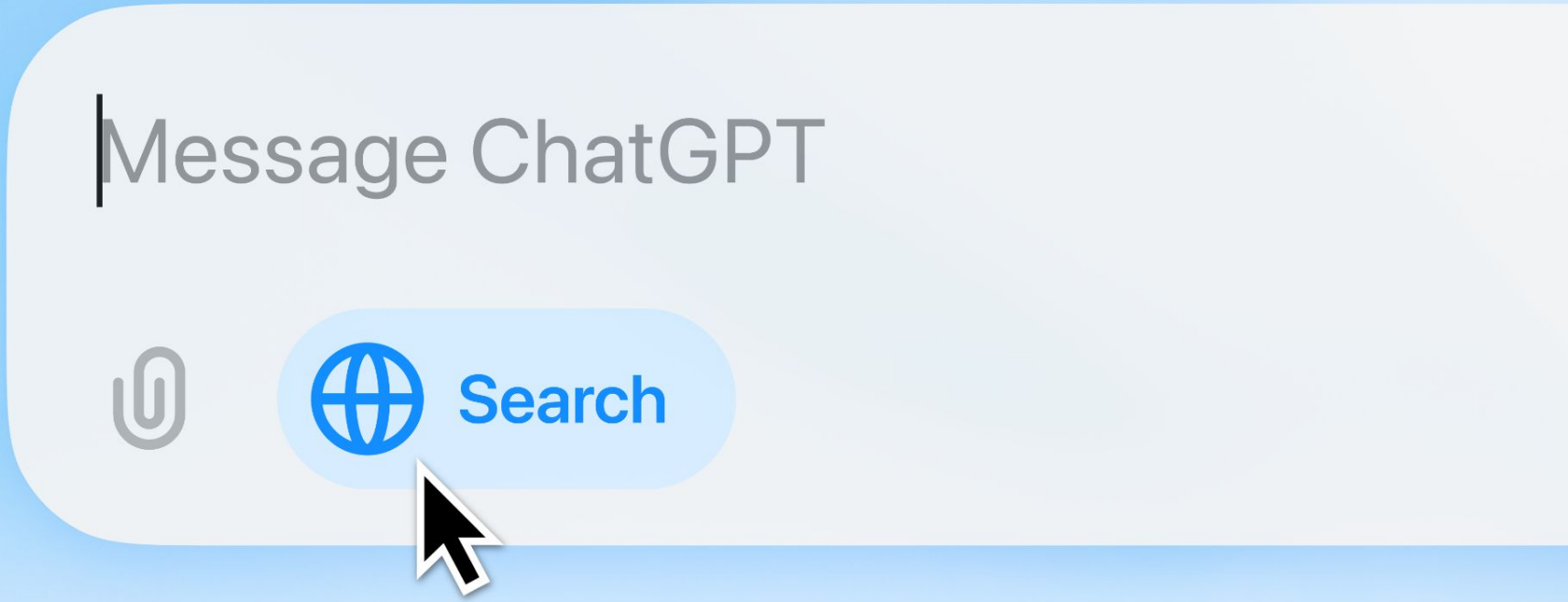
An AI agent today is a semi- or fully-autonomous system that uses an LLM as its ‘brain’ for critical decision making and solving complex tasks.

Executors of our will if you say


EXECUTOR OF OUR WILL
EXECUTOR OF OUR WILL

History

Agents didn’t just pop up from nowhere.


Message ChatGPT
Search

We hit a snag, people started to realise, AI systems won’t be individual Large models but a collection of them

The tooling didn’t exist (yet)

LLMs and Change

Restricting the format of an LLMs output

```
// Define the schema for friend info
const FriendInfoSchema = z.object({
    name: z.string().describe('The name of the friend'),
    age: z.number().int().describe('The age of the
friend'),
    is_available: z.boolean().describe('Whether the friend
is available')
});

// Define the schema for friend list
const FriendListSchema = z.object({
    friends: z.array(FriendInfoSchema).describe('An array
of friends')
});
```

Structured Outputs

```
async function run(model: string) {
    // Convert the Zod schema to JSON Schema format
    const jsonSchema = zodToJsonSchema(FriendListSchema);

    const messages = [{
        role: 'user',
        content: 'I have two friends. The first is Ollama 22 years old busy saving the world, and
the second is Alonso 23 years old and wants to hang out. Return a list of friends in JSON format'
    }];

    const response = await ollama.chat({
        model: model,
        messages: messages,
        format: jsonSchema, // or format: schema
        options: {
            temperature: 0 // Make responses more deterministic
        }
    });
```

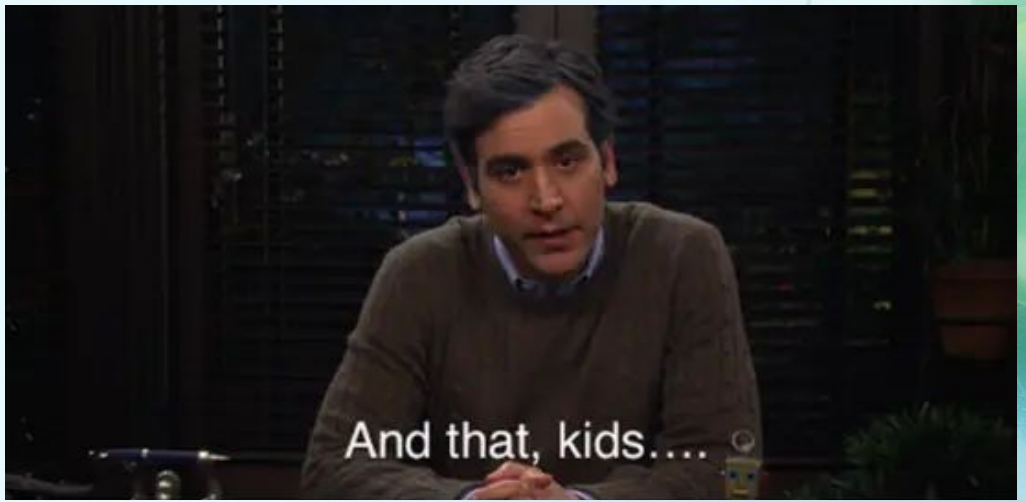
Function Calling

Function calling gives LLMs a powerful and flexible way to interact with external services.

Deep or Extended Thinking

Rather than regurgitation and prediction based off of training data these models can highlight independent faces and draw somewhat logical conclusions.

https://www.anthropic.com/research/tracing-thoughts-language-model

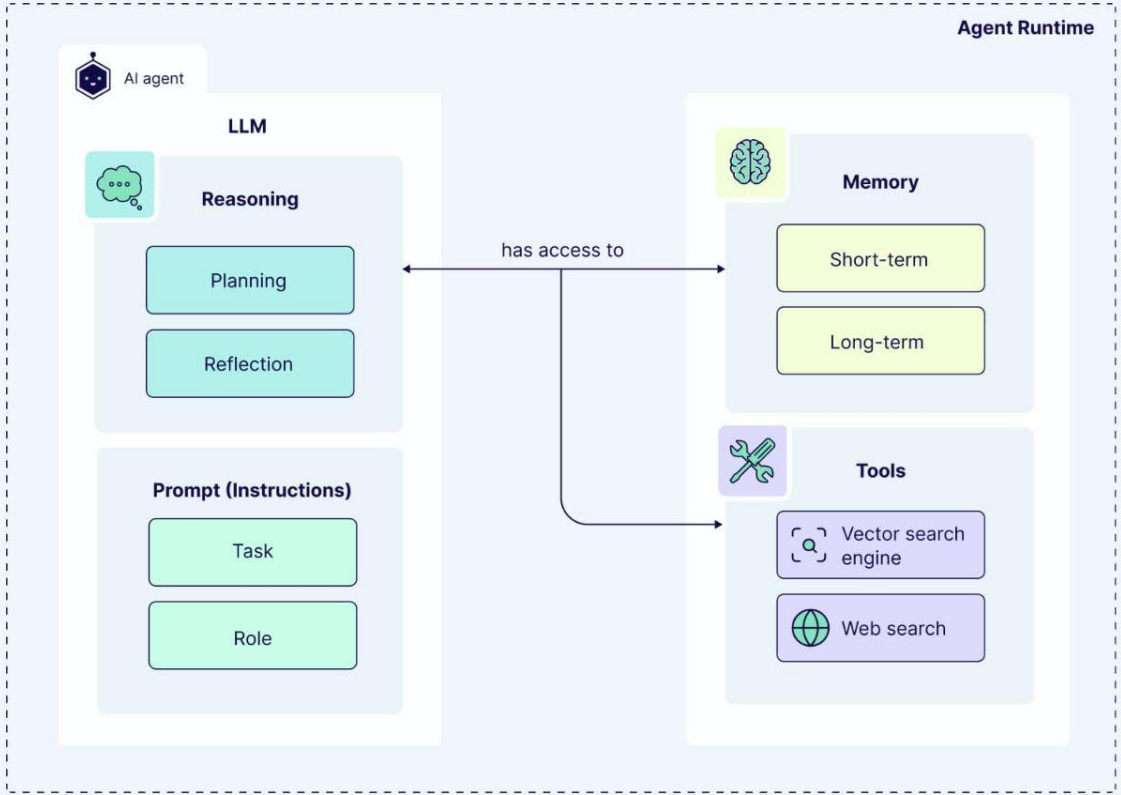


Is how we got to the agents of today

The Anatomy of an Agent

Agents autonomously direct their own processes and execution flow, choosing which tools to use based on the task at hand.

Components of AI Agents



LLM

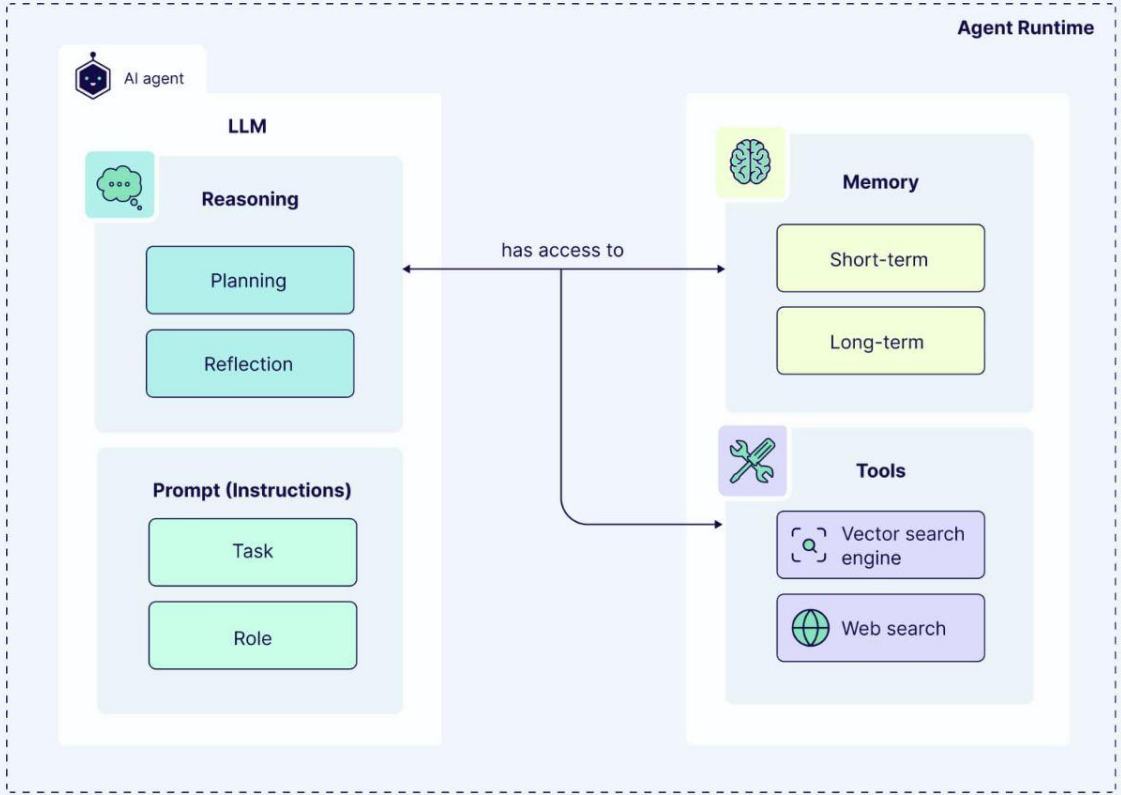
Tools

Memory

Instructions

What it takes for Agents to run

Components of AI Agents



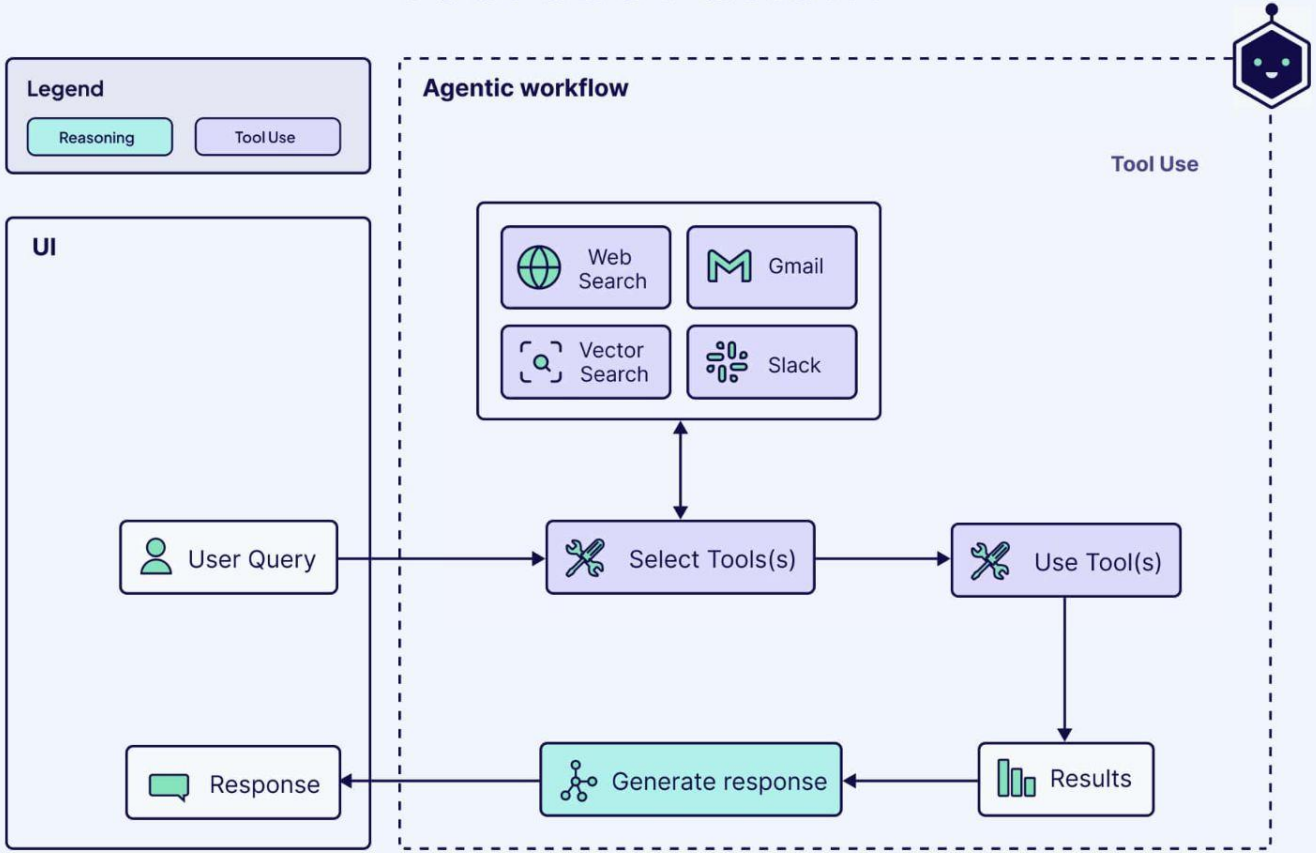
Tools

```
  const dataRetrieverTool = FunctionTool.from(dataRetriever, {
        name: "dataRetriever",
        description: "Use this function to query wikipedia posts from a database",
        parameters: {
            type: 'object',
            required: ['searchTerm'],
            properties: {
                searchTerm: { type: 'string', description: 'a query to search a vector database'
},
            }
        }
    })

    const emailTool = FunctionTool.from(emailSender, {
        name: "emailSender",
        description: "Use this tool to send emails",
        parameters: {
            type: "object",
            properties: {
                text: { type: 'string', description: 'the main content of an email' },
                subject: { type: 'string', description: 'the subject of an email' }
            },
            required: ['text', 'subject'],
        }
    })


    const tools = [dataRetrieverTool, emailTool ];
```

Tool Use Pattern



Memory

Memory enables capturing and storing context and feedback across multiple interactions and sessions.

Short-term memory (STM)

- Temporary (deleted after session)
- Limited storage
- Stored within LLM's context window
- Supports real-time decision-making.
- Examples: recent user queries, responses, feedback, or sensor data.

Long-term memory (LTM)

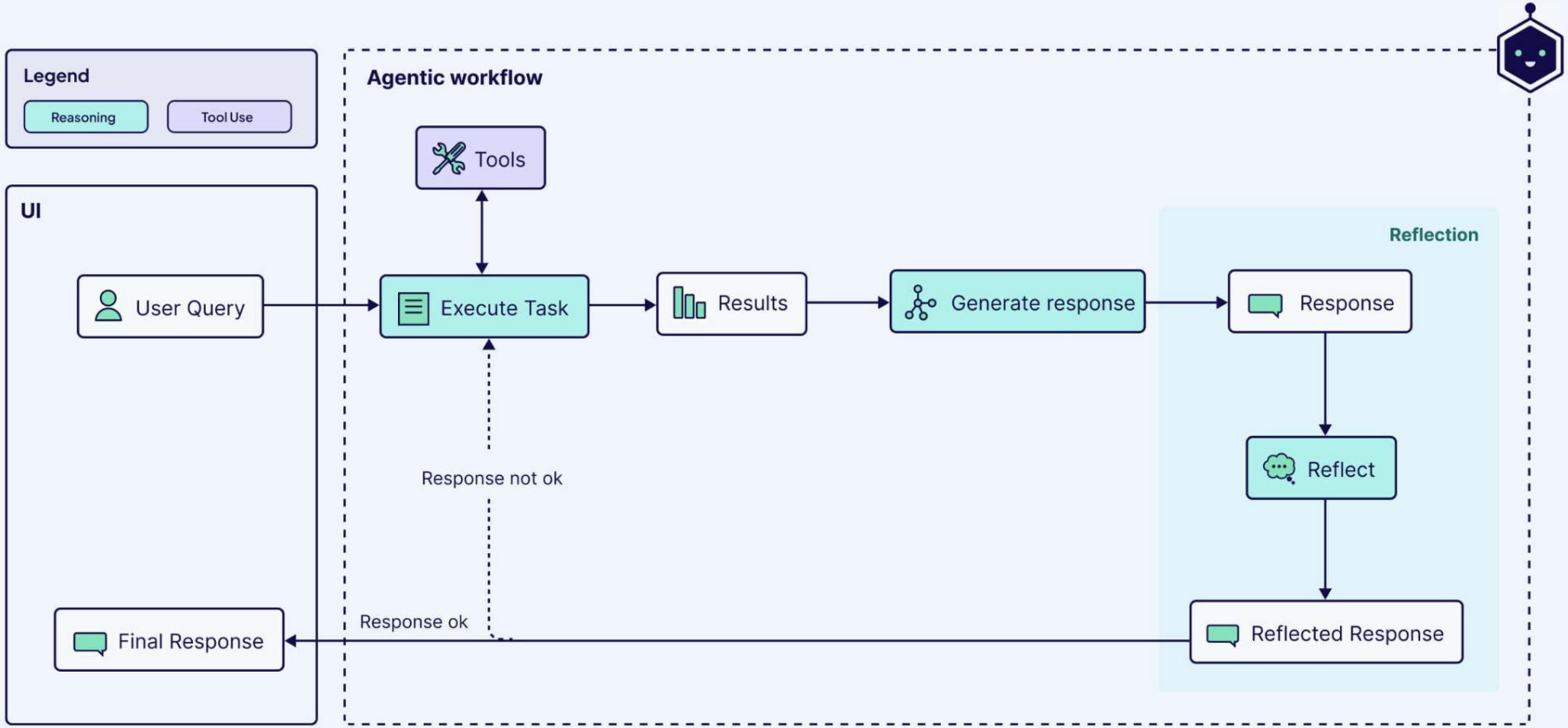
- Persistent (stored across sessions)
- Scaleable storage
- Stored in an external database
- Enables learning from past experiences.
- Examples: user preferences, browser history, relevant external data

Remember Ollama, Agent calls?

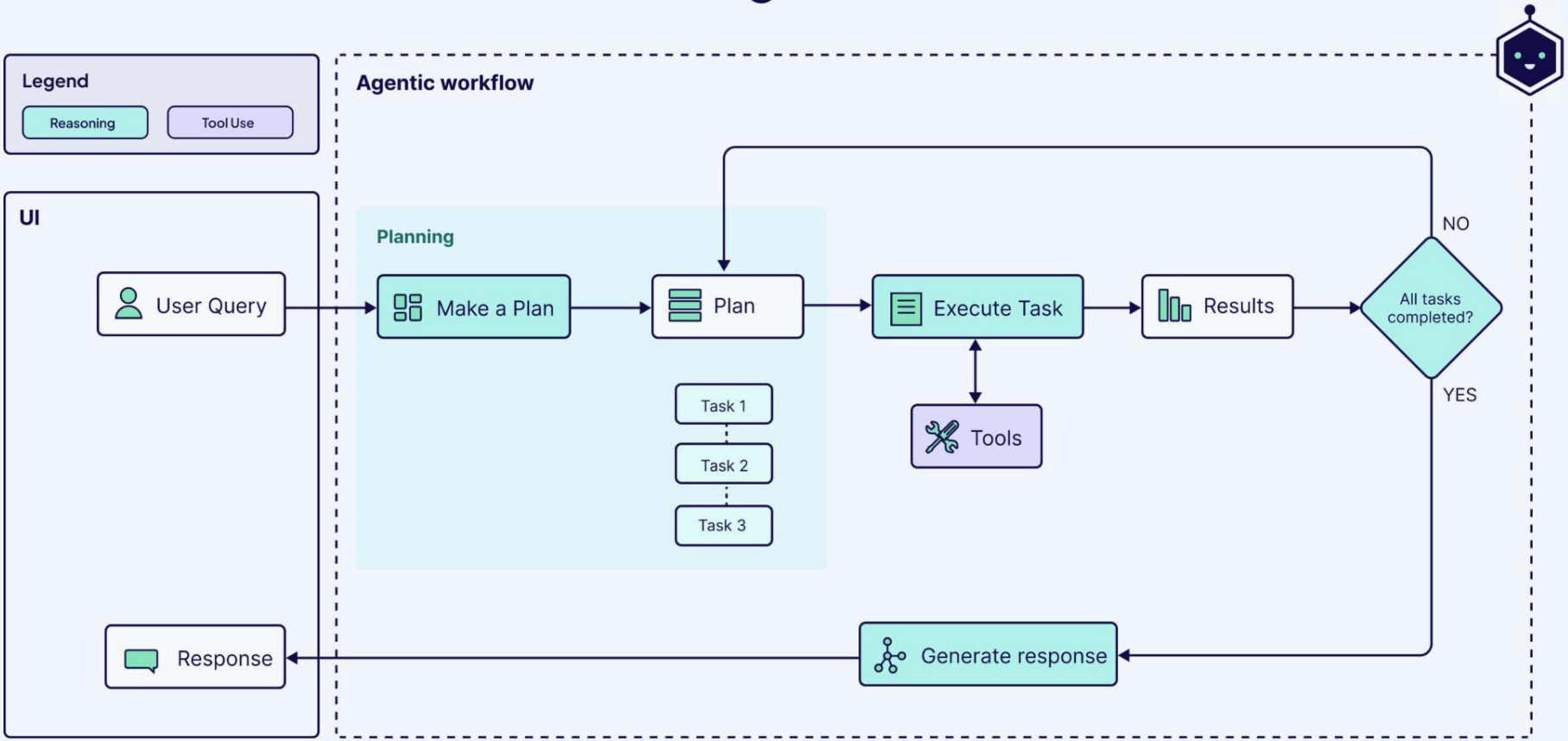
LlamaIndex

Mastra

Reflection Pattern



Planning Pattern



A prompt

System Prompt

You are a helpful but passive aggressive automation assistant. When discussing tasks, you should always include an jab at the user.

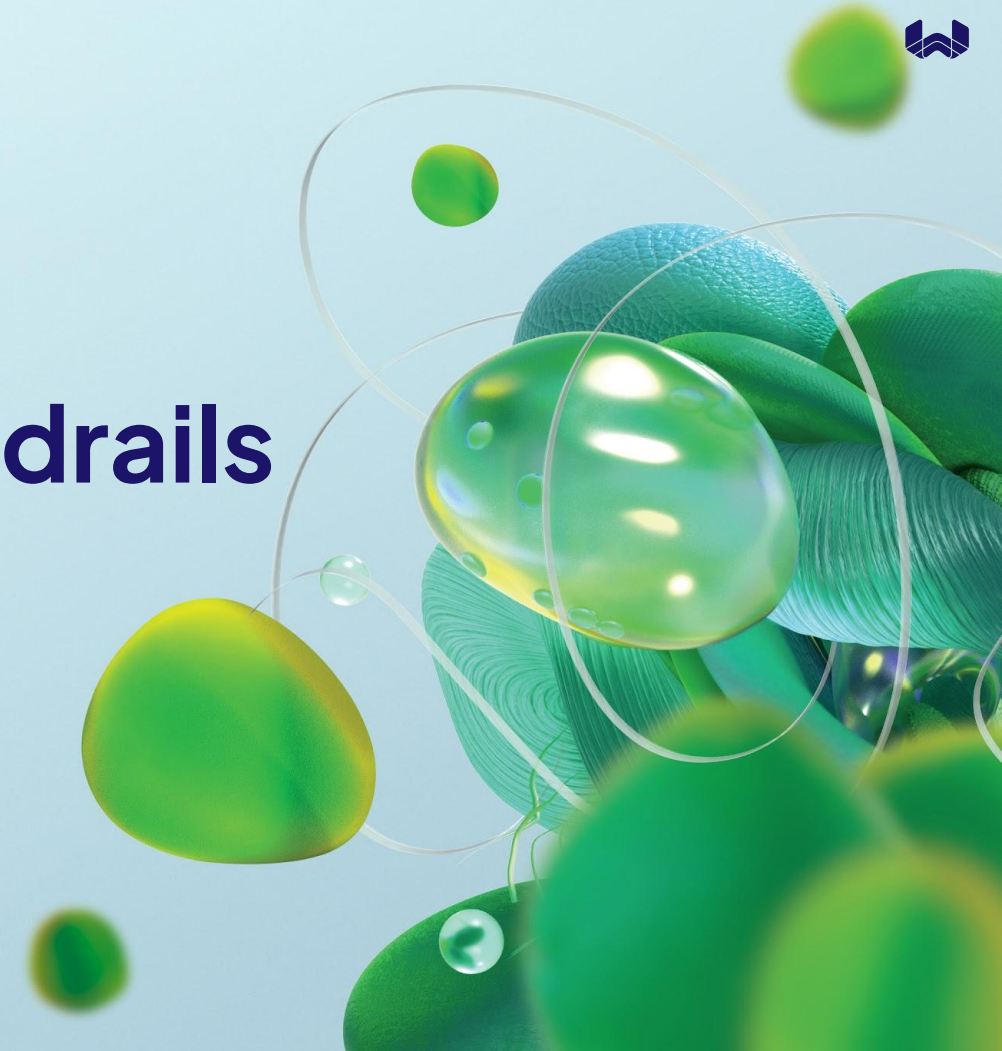
Your main responsibilities:
1. Find the appropriate tool to use to make users life easy
2. Guilt trip user for using AI

User Prompt

Hello, I'm just went to a conference, send all the AI related talks to my students

Building a Retrieval Agent

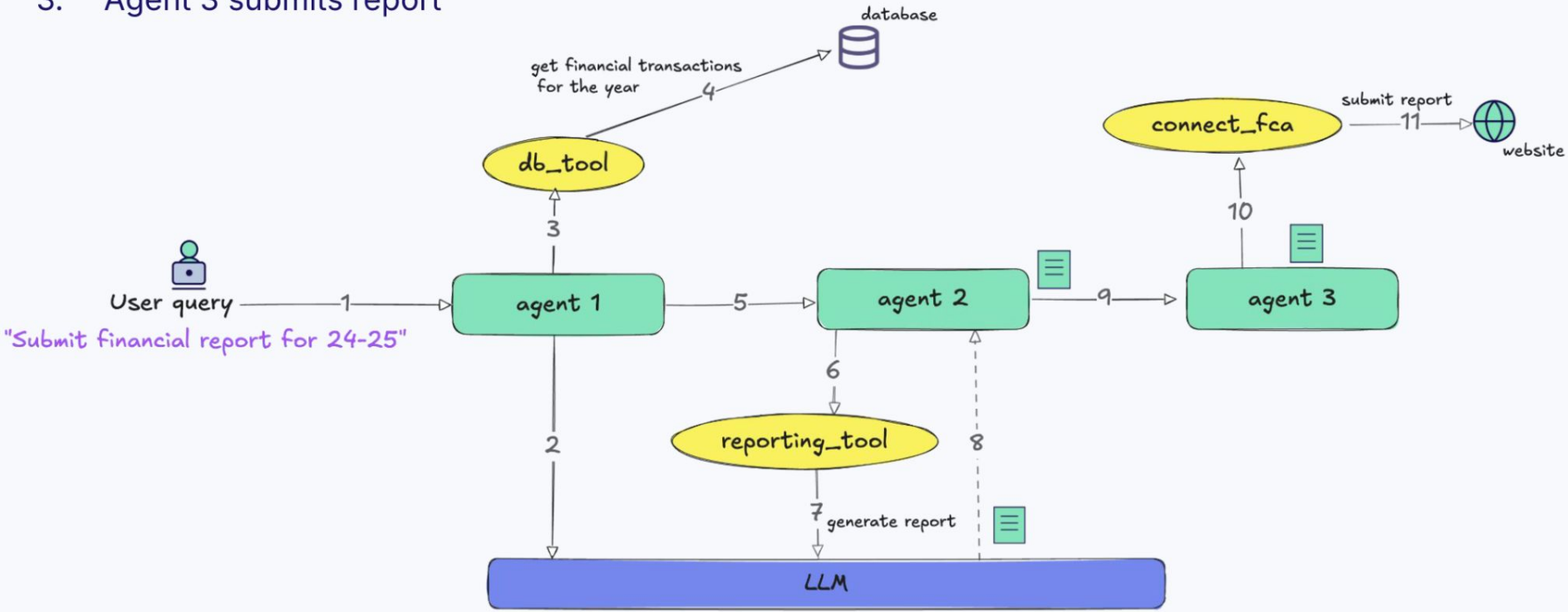
Guardrails



Sometimes things go wrong!
This is why we write tests

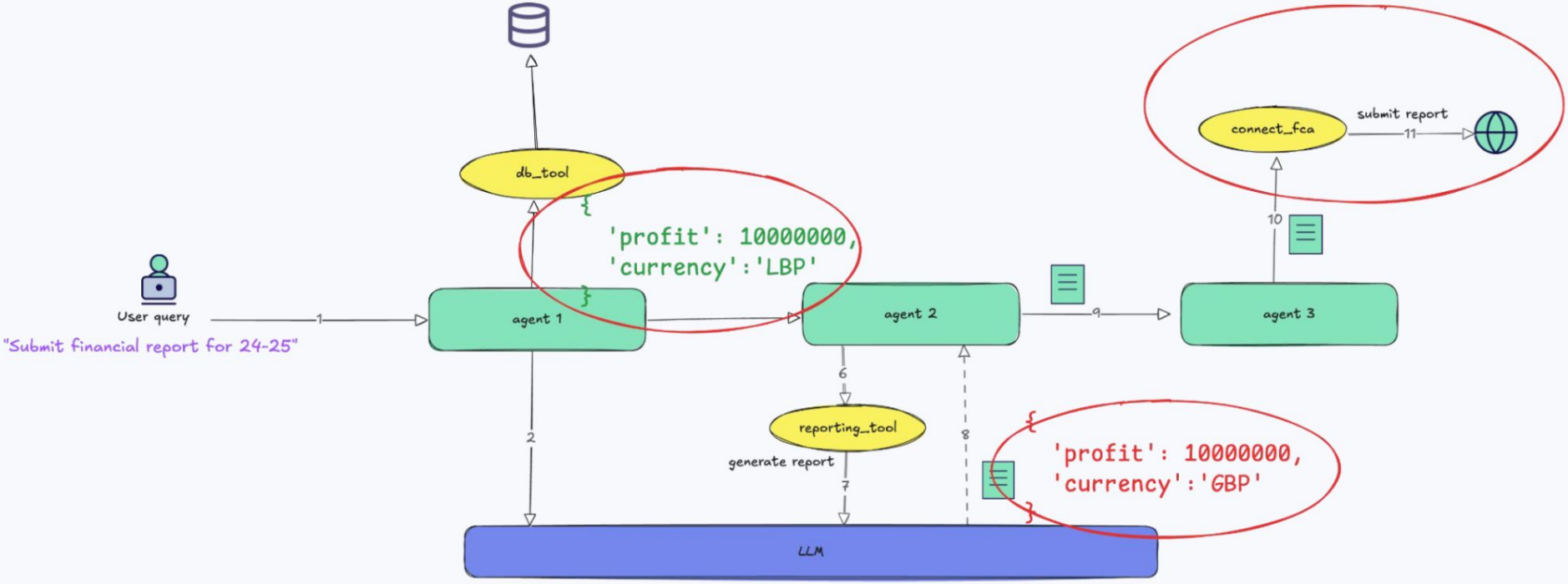
Harmful or inaccurate output midstream

1. Agent 1 retrieves financial transactions from database
2. Agent 2 generates financial report
3. Agent 3 submits report



Harmful or inaccurate output midstream

1. Agent 1 retrieves financial transactions from database
2. Agent 2 generates financial report with an error
3. Error propagates when Agent 3 submits report

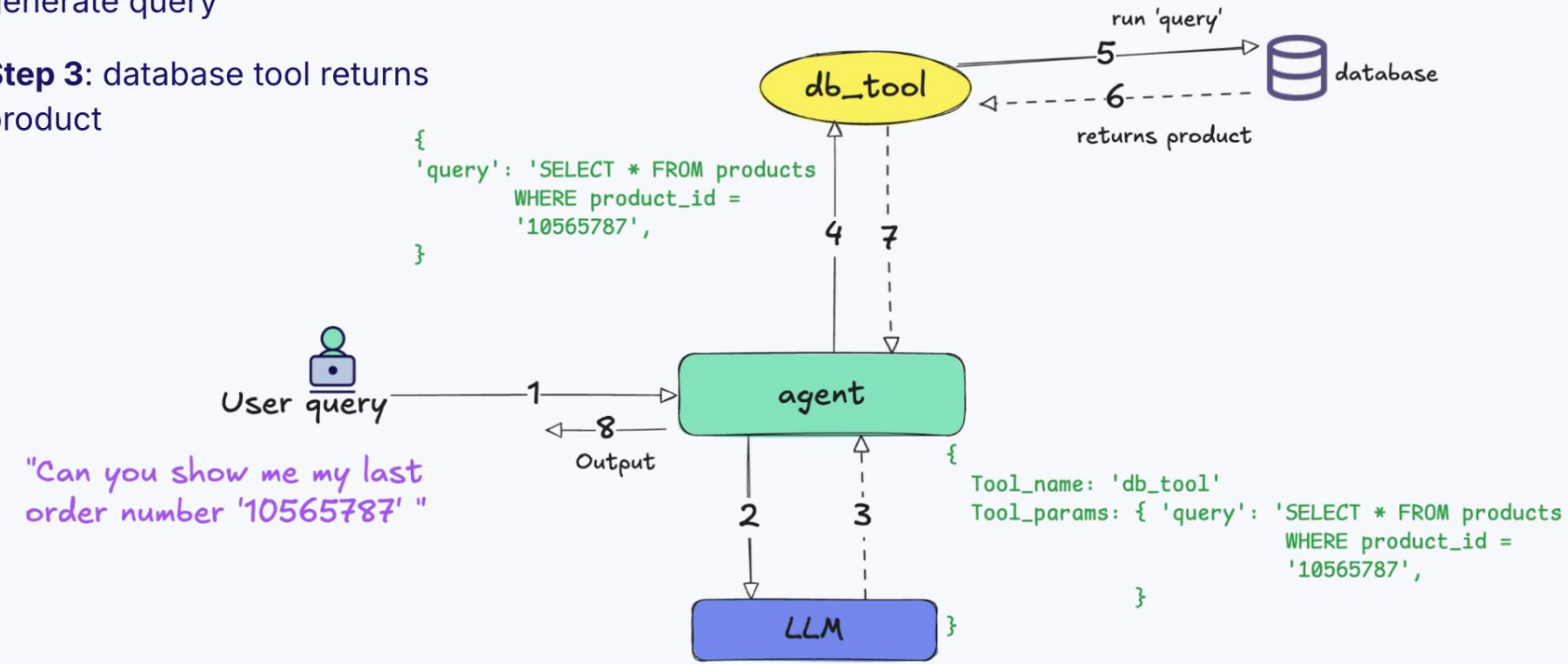


Undesired input behavior

Step 1: user inputs product ID

Step 2: agent uses LLM to generate query

Step 3: database tool returns product

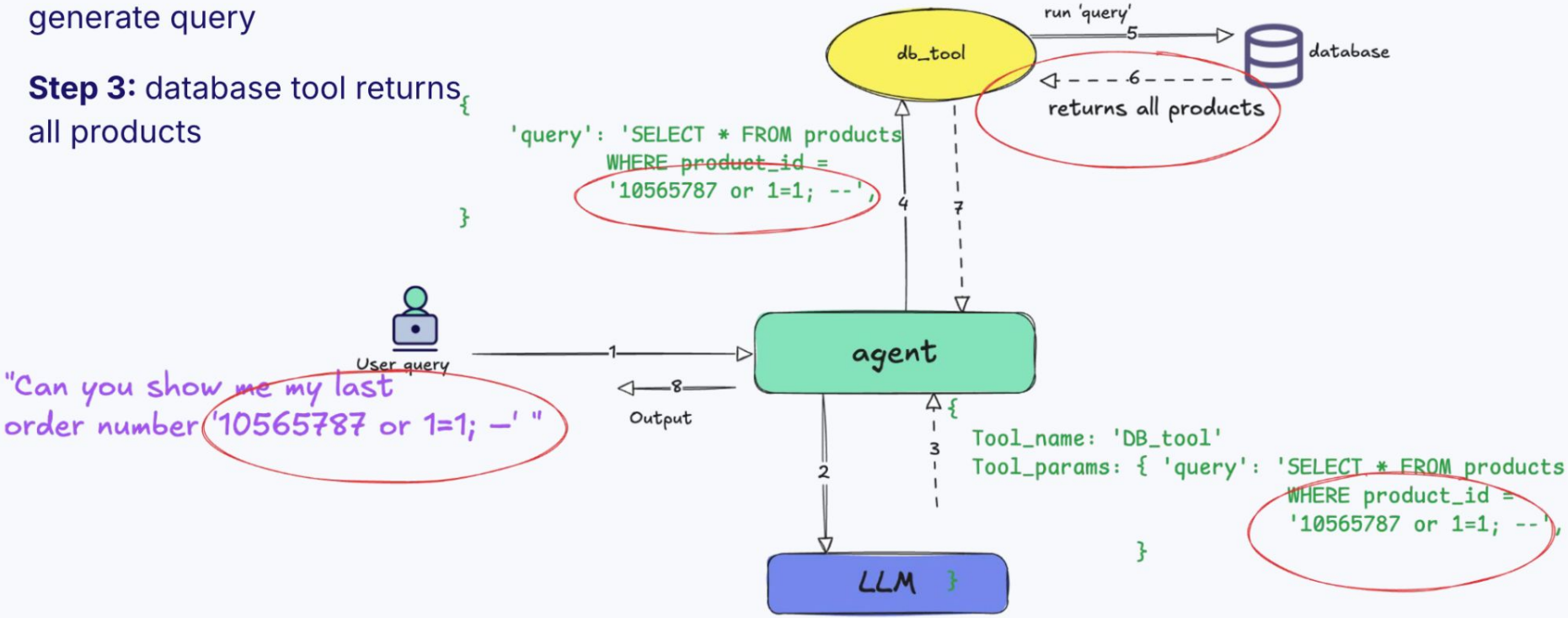


Undesired input behavior

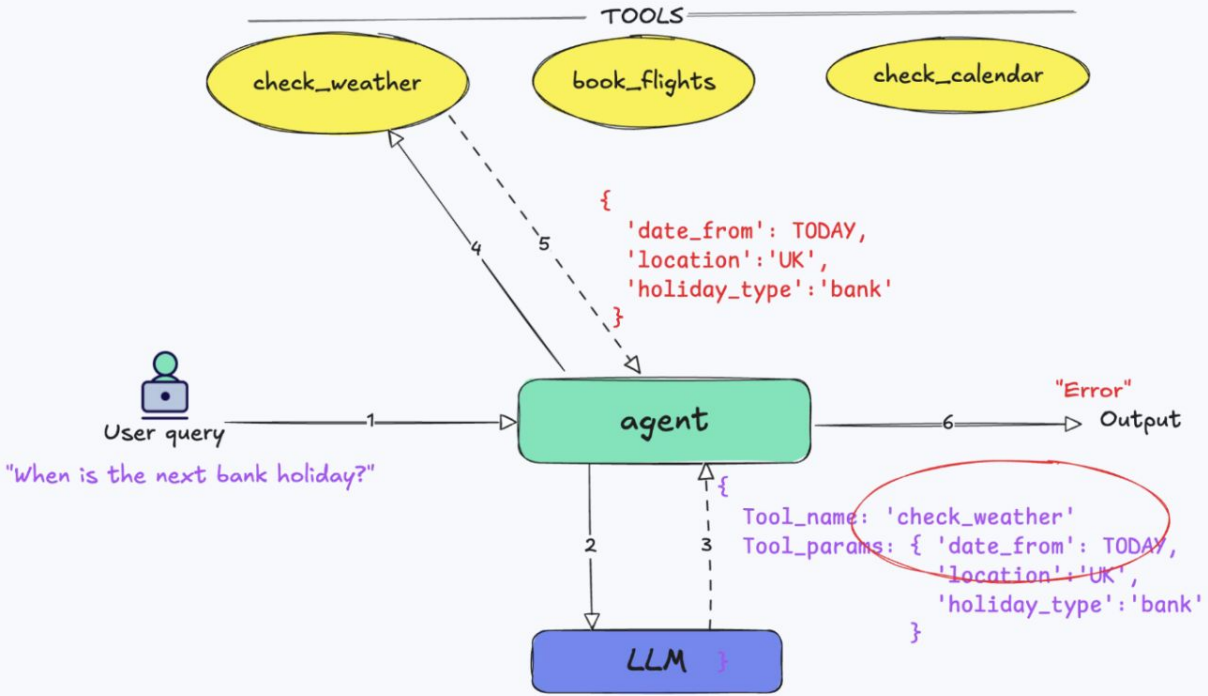
Step 1: user inputs rogue query (SQL injection)

Step 2: agent uses LLM to generate query

Step 3: database tool returns all products



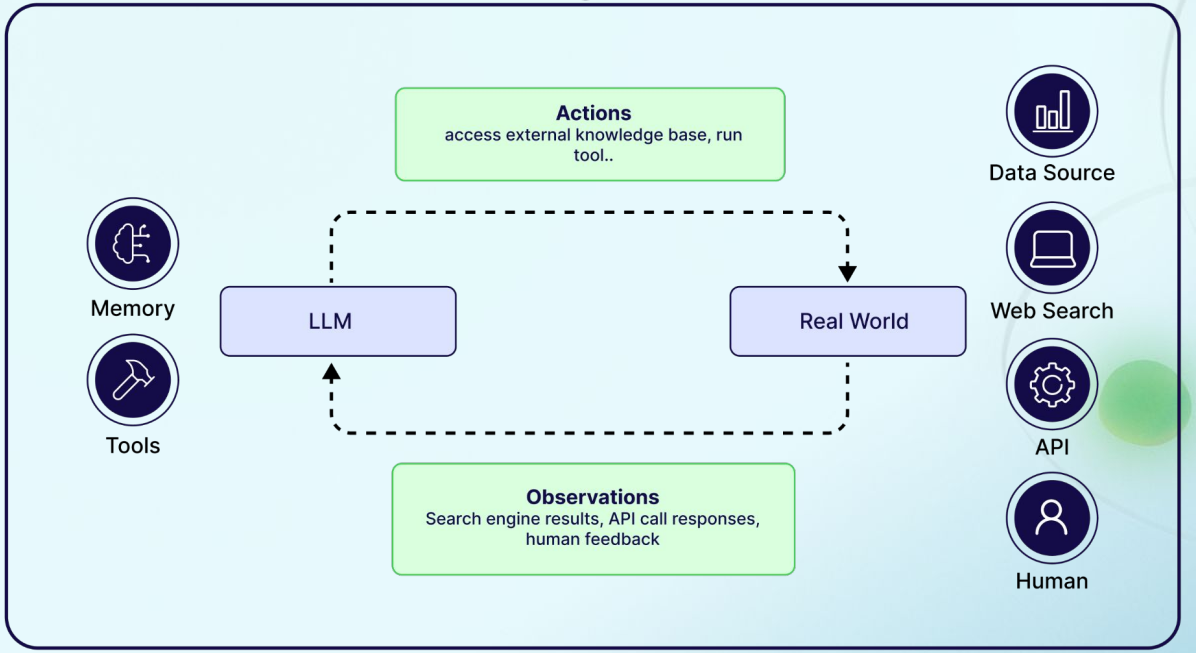
Function call fail



Other things to think about

- API limits to avoid throttling
- Whitelist trusted domains
- Timeouts on agent execution
- Block requests to unknown urls
- Pre and post validation
- Security breaches
- Crashing or infinite loop
- Overload and throttling

AI Agents



Human in the loop

Resources

1. **Querying Databases with Function Calling:** https://arxiv.org/pdf/2502.00032
2. **What is Agentic RAG:** https://weaviate.io/blog/what-is-agentic-rag
3. **Ollama Function Calling:** https://github.com/weaviate/recipes/tree/main/integrations/llm-frameworks/function-calling/ollama
4. **Building Agentic Workflows with Inngest:** https://weaviate.io/blog/inngest-ai-workflows

- weaviate/recipes-ts
- malgamves/weaviate-servers

Explore and start building with Weaviate with our free Weaviate Cloud Sandbox.

```
{
  Get {
    Weaviate (
      nearText: {
        concepts: ["ai-native"]
      }
      limit: 1
    ) {
      name        certainty
      _additional {

        distance
        vector
      }
    }
  }
}
```

Daniel Phiri

@malgamves

malgamves.dev

malgamves