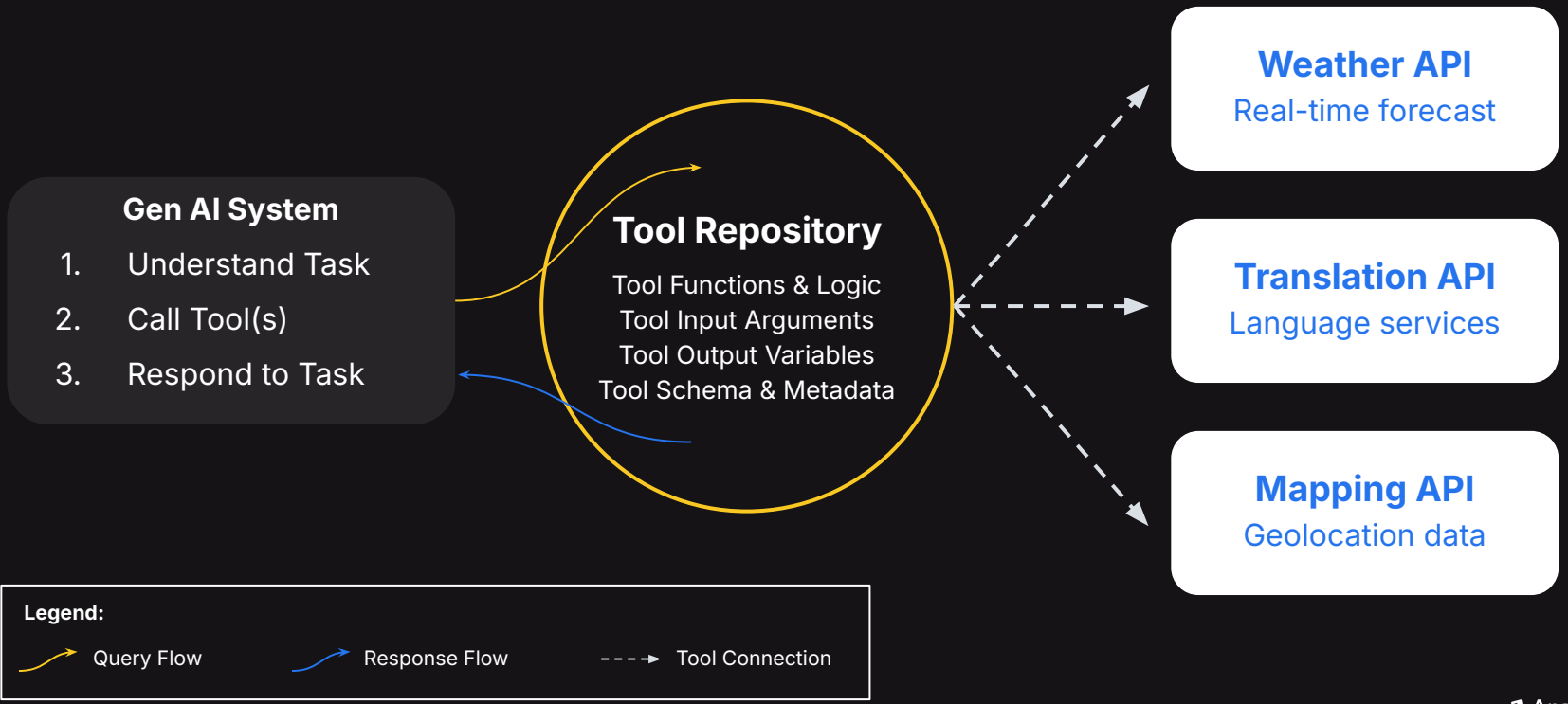
Analytics Vidhya

Core Elements of the Tool Use Pattern

Created by:
Eleni Verteouri
Gen AI Tech Lead @ UBS

Created & Narrated by:
Dipanjan Sarkar
Head of Community & Principal AI Scientist @ Analytics Vidhya
Google Developer Expert - ML & Cloud Champion Innovator
Published Author

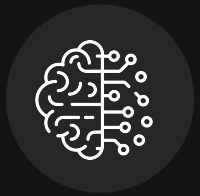
Tool Use Pattern Architecture



Analytics Vidhya

Key Architectural Components of the Tool Use Pattern

1



Core AI System

- Handles input processing, task orchestration, and output generation using LLMs.
- Identifies the task and determines if external tools are needed.
- ***Examples:*** *Deciding to call a weather API for real-time data for the city of Berlin.*

2

Tool Repository

- A catalog of available tools and APIs with defined capabilities and arguments.
- Stores information about tools, including logic, metadata, schema, input and output variables.
- ***Example:*** *Function to get weather data accepting City as an input string argument, using a weather API and returning a string JSON response of the weather of the city.*

3

Selection Mechanism

- LLMs use their reasoning and the details from the tool repository to decide what tools might be useful
- Evaluates task requirements and selects the tool based on their reasoning.
- ***Example:*** *Choosing the get_weather tool when the user wants to know the weather of Berlin city before traveling.*

Analytics Vidhya

Key Architectural Components of the Tool Use Pattern

4

Tool Execution Layer

- Invokes the external tool using right input arguments and returns the output.
- Ensures seamless data exchange between the AI Agent and the external tool.
- ***Example:*** *Passing city=Berlin to the get_weather tool, calling it to get the current weather as a JSON reponse*

5

Feedback Loop

- Incorporates tool output into the LLM's reasoning process for refinement.
- Ensures the final output is tailored to user needs.
- *Example: Taking the JSON response of the get_weather tool, parsing it and generating a human-like response talking about the current weather in the city of Berlin as per the user query*

Analytics Vidhya

Tool Use Pattern with an Example

Example: Planning a Trip

Problem Statement:

A user wants to “**plan a trip”** but needs a real-time weather updates, flight options, and accommodation recommendations.

How does the Tool Use Pattern Help?

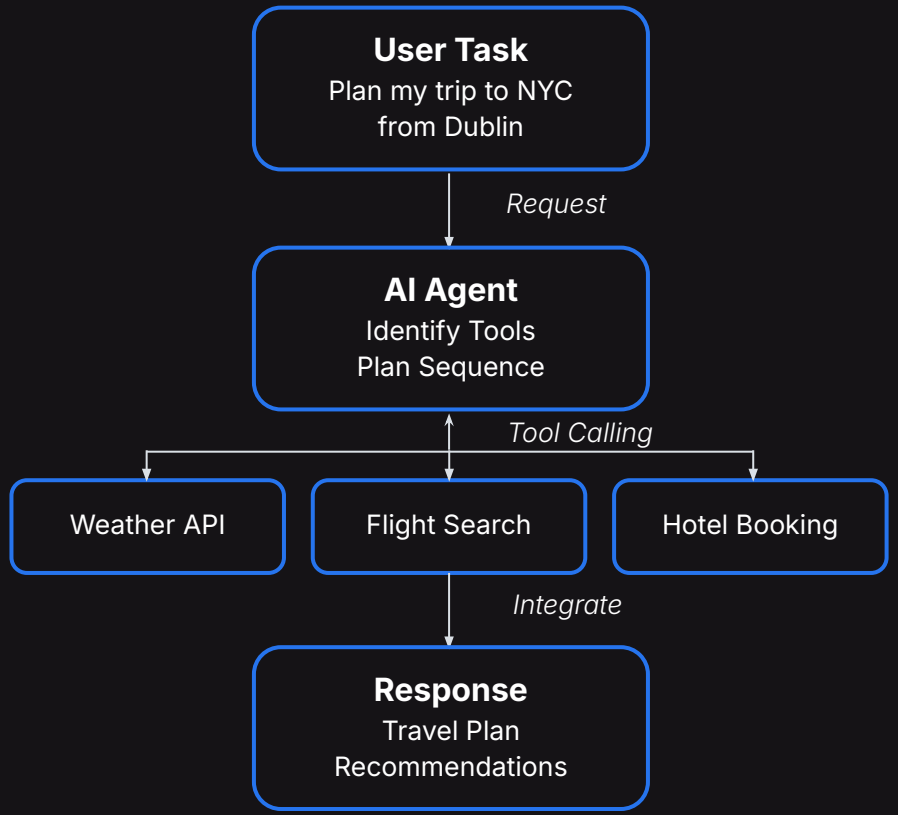
Step 1: Recognize its Limits: The LLM understands it doesn't have the latest weather data or access to booking platforms.

Step 2: Call External Tools: It seamlessly connects to APIs:

- *A weather API for real-time forecasts.*
- *A flight booking API for airline options.*
- *A hotel booking API for nearby accommodations.*

Step 3: Integrate the Results: The LLM compiles the information into a comprehensive travel plan tailored to the user's preferences.

Workflow of the AI Agent: Planning a Trip



Analytics Vidhya

Workflow of the AI Agent: Reasoning Phase

1. Reasoning Phase – Task Understanding and Initial Plan

Goal: Identify the user's intent and draft an initial plan using tools.

- **User Input:**
  "Plan my trip to NYC from Dublin"

- **AI Agent Actions:**

  - **Understand** the user's objective (travel planning).
  - **Break it down into subtasks:** check weather, find flights, book hotels.
  - **Choose appropriate tools/APIs** for each subtask.
  - **Generate** tool calling requests for each subtask.

Workflow of the AI Agent: Observation Phase

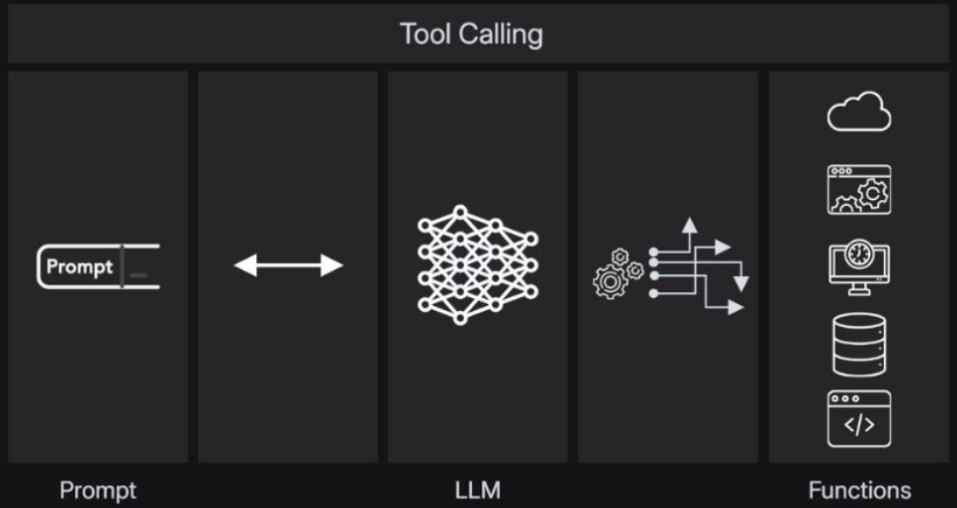
2. Observation Phase: Critically assess data fetched from tools and the planned sequence.

- **Tool Calls:**
  - **Weather API:** Get forecast for NYC
  - **Flight Search:** Look for best flights from Dublin to NYC
  - **Hotel Booking:** Search for appropriate hotels in NYC

Dimension	Evaluation Focus
Correctness	Are the tools providing valid and recent data?
Completeness	Are all components (weather, flight, hotel) included?
User Fit	Do results match user preferences (e.g., cost, timing)?
Error Handling	Are fallback responses in place if a tool fails?

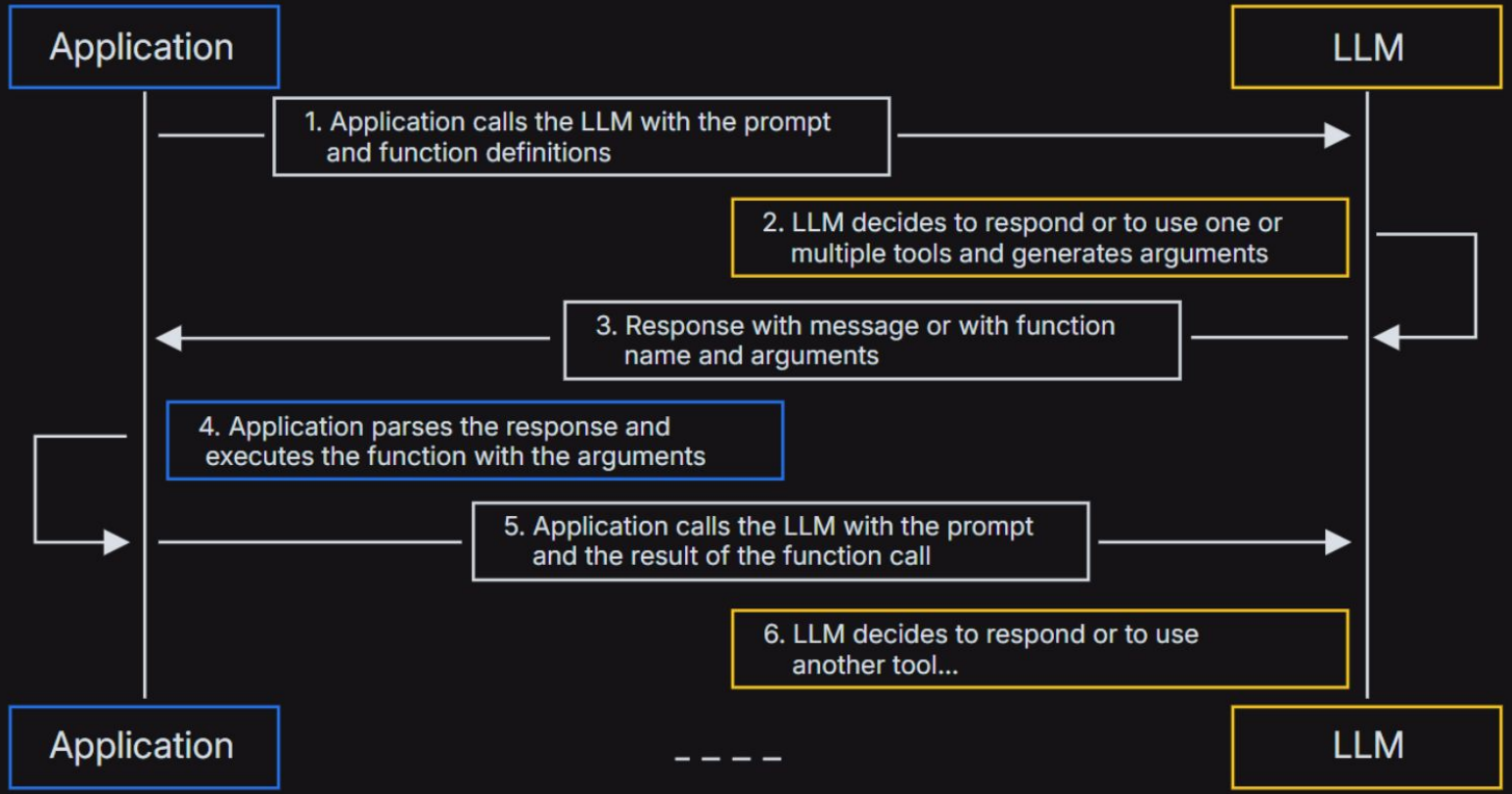
What is Tool or Function Calling?

- **Function calling** enables Large Language Models (LLMs) to interact with the external environment, rather than just generating text responses.
- **LLMs can identify the tools or functions** needed for additional information to formulate an appropriate response.
- **Access to pre-defined tools** like web search is required for function calling, where the LLM indicates the specific tool and arguments needed.
- Agents must independently call the tools as function calling **does not** automate this process; it simply specifies which tools are needed.



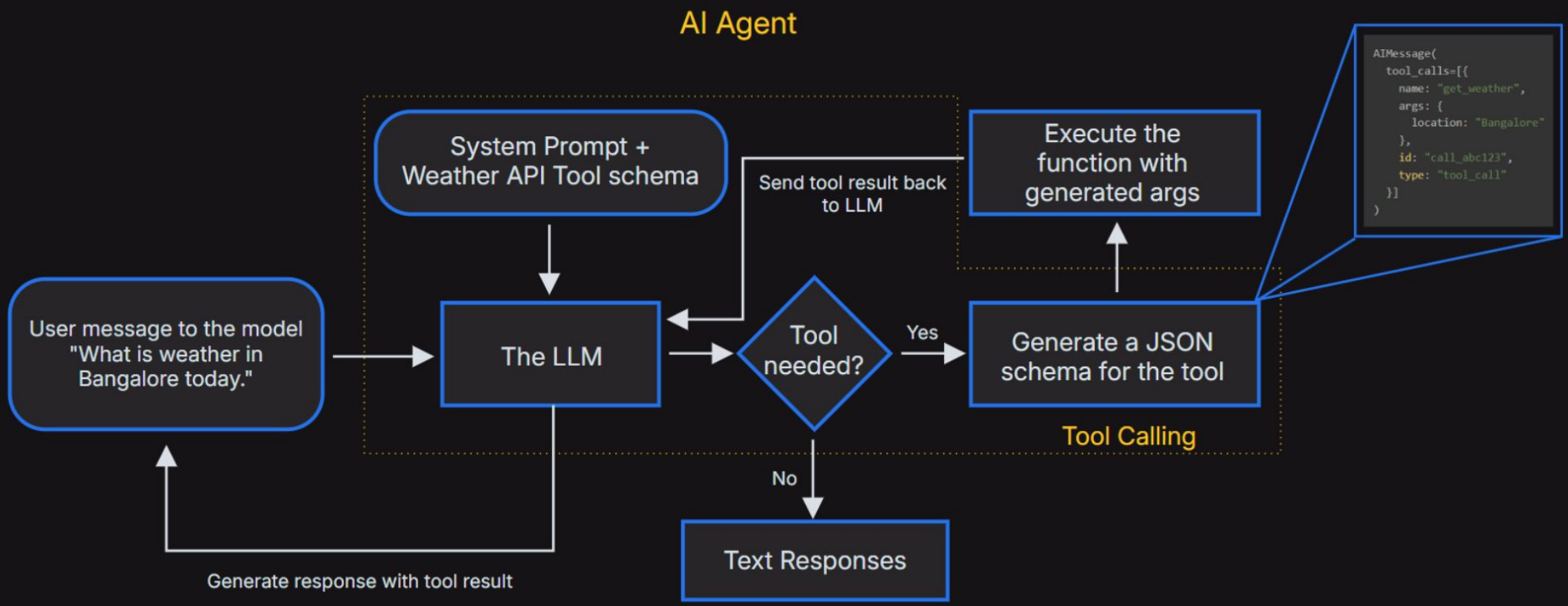
Analytics Vidhya

How does Tool Calling Work?



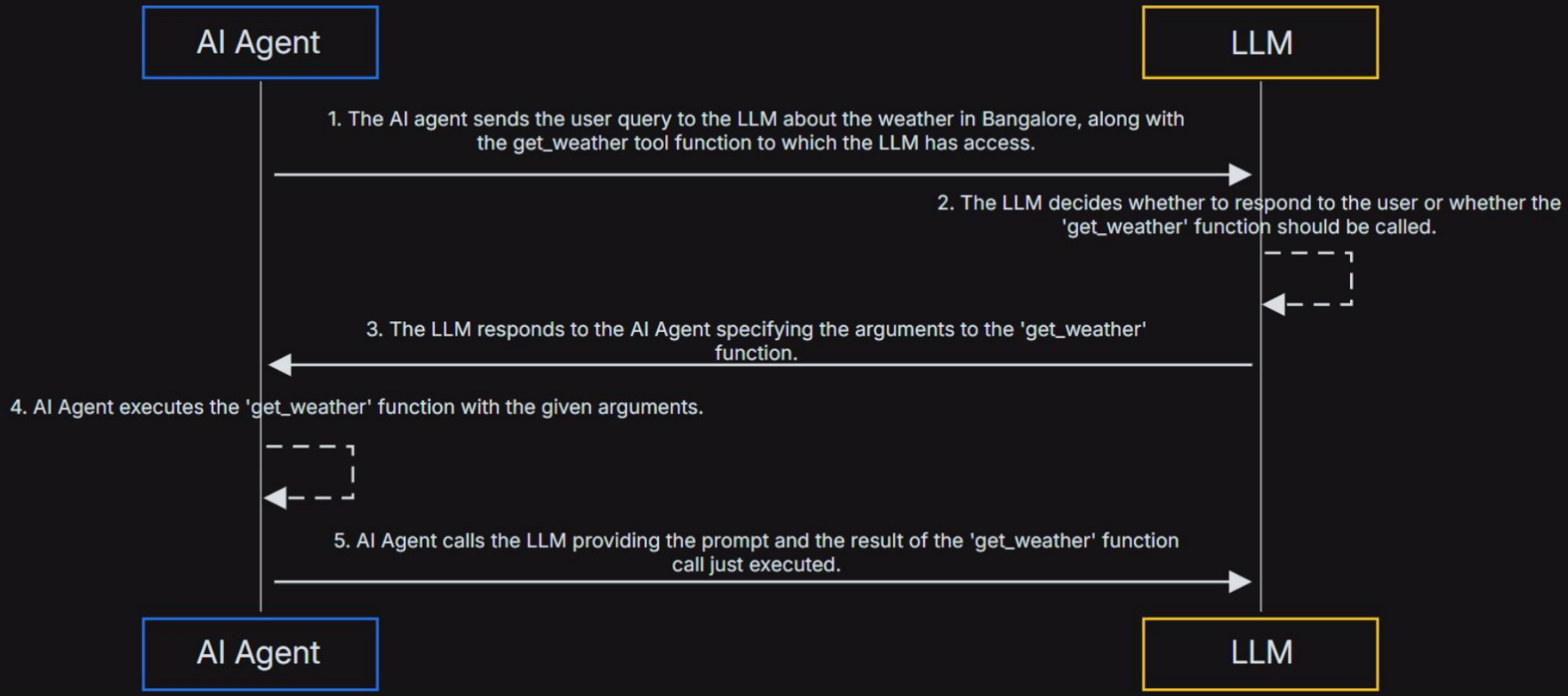
Analytics Vidhya

Example for Tool Calling



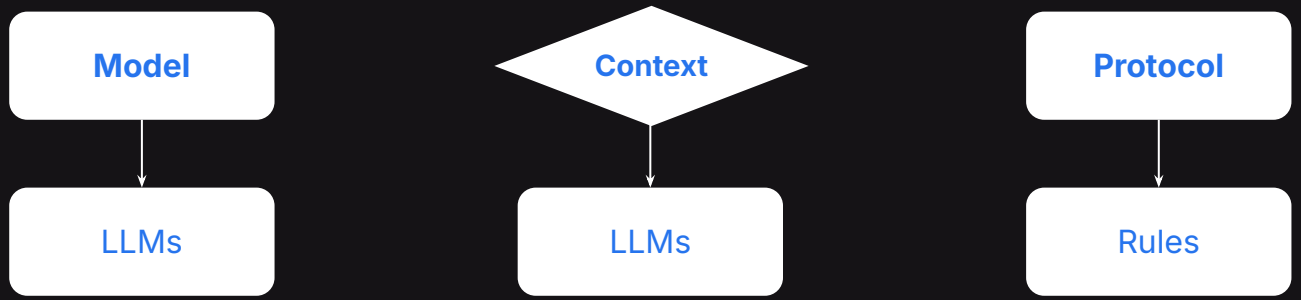
Analytics Vidhya

Example for Tool Calling: Workflow



What is Model Context Protocol (MCP) ?

Model Context Protocol is a powerful open standard that was launched by Claude's parent company, **Anthropic**, in **November 2024**.



Model: The LLM (e.g., Claude, GPT-4) generates responses.

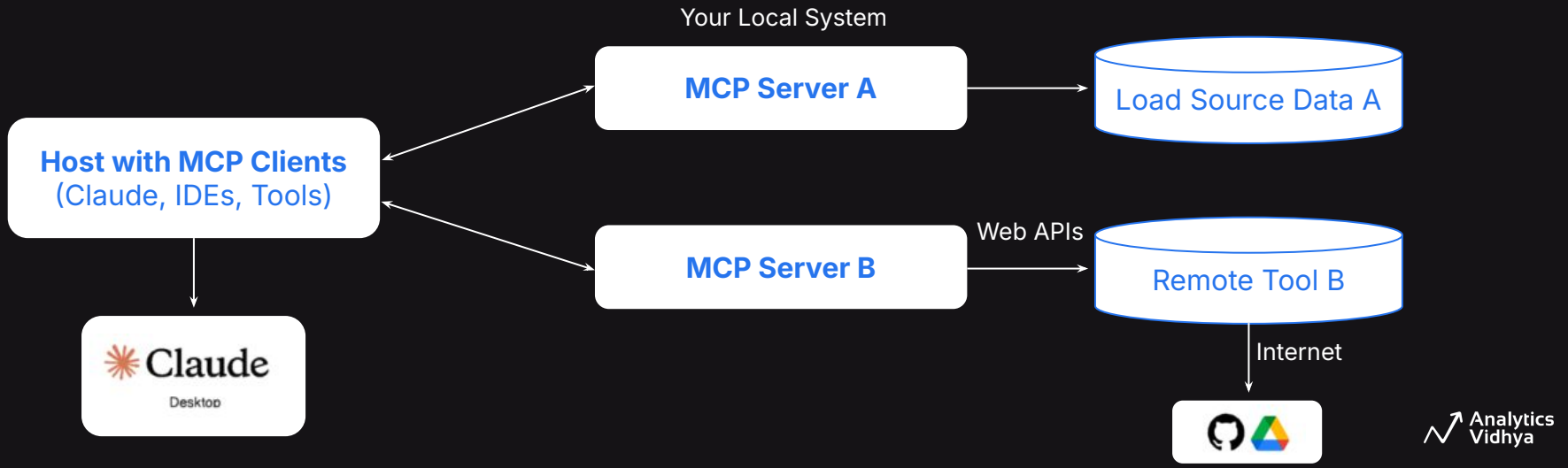
Context: Additional input (documents, PDFs, prompts, databases) for meaningful replies.

Protocol: Rules enabling the model to access and use structured context.

Why MCP?

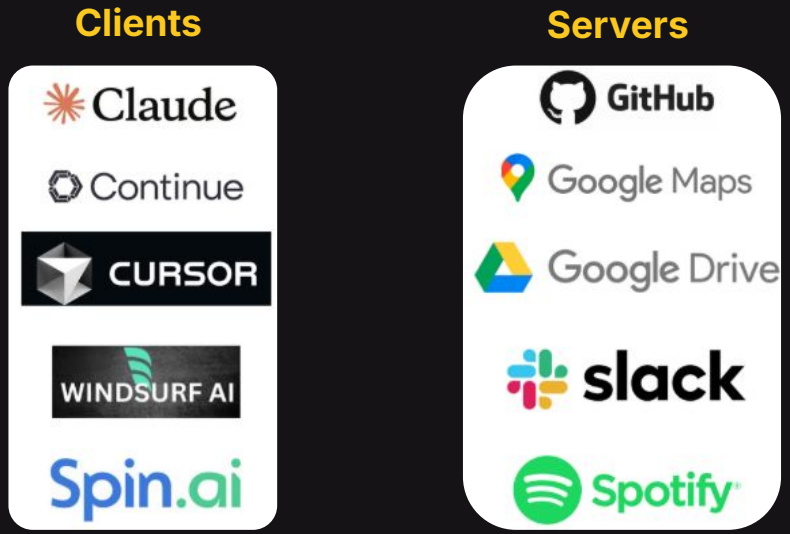
MCP standardizes tool calling by streamlining communication between AI clients and tool servers, enabling agents to access real-time data and tools without custom code or manual uploads.

- **Client:** The interface for LLM responses (e.g., Claude's app, IDE-like cursor, custom chatbots, etc).
- **Server:** The data or tool storing context (e.g., Google Drive, GitHub, databases, tools, etc).



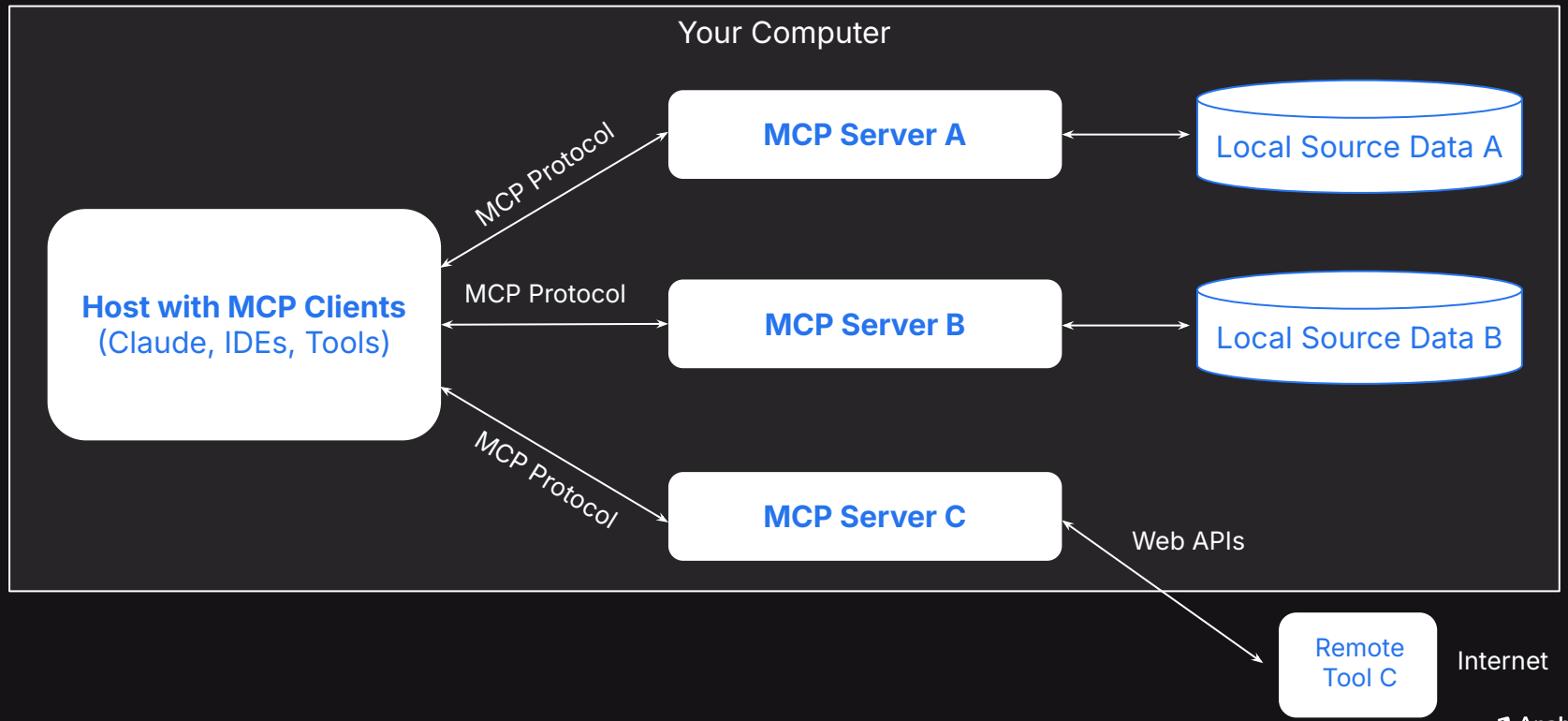
Why MCP?

- **Client:** The interface for LLM responses (e.g., Claude's app, IDE-like cursor, custom chatbots, etc).
- **Server:** The data source storing context (e.g., Google Drive, GitHub, database, local files, etc).



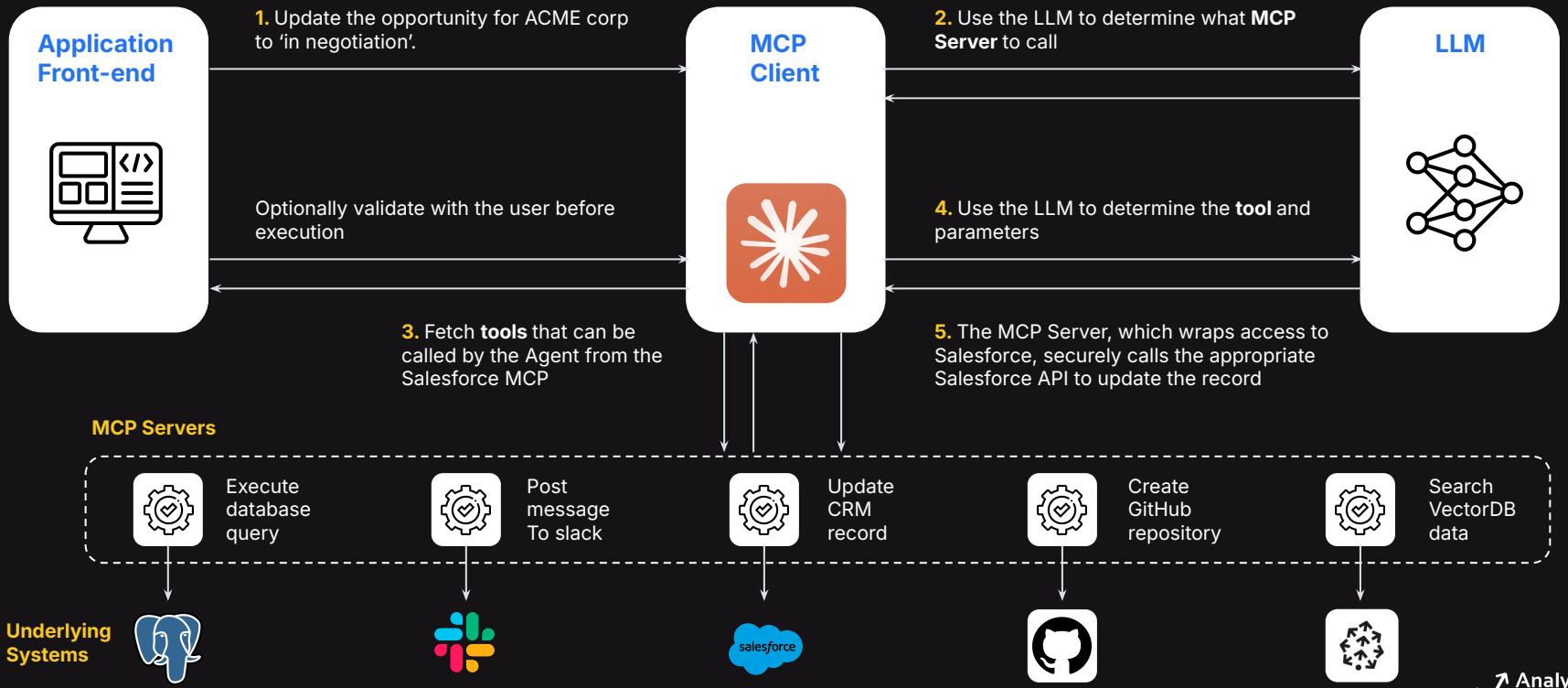
Analytics Vidhya

Standard MCP Architecture



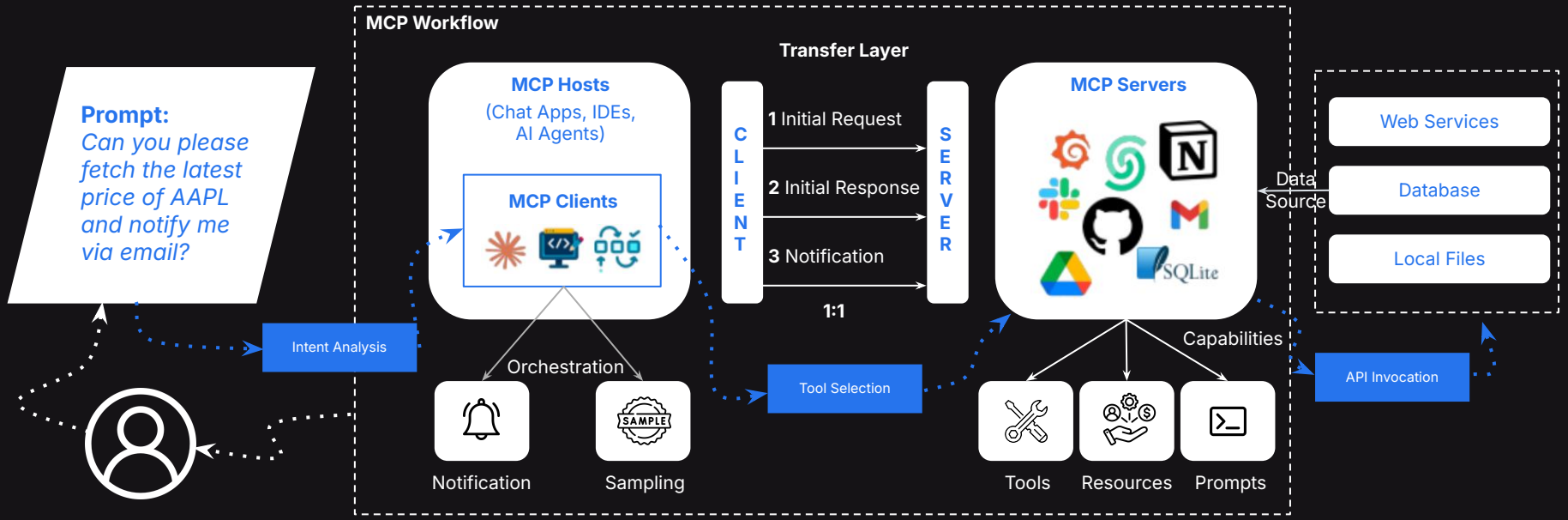
Analytics Vidhya

MCP Workflow



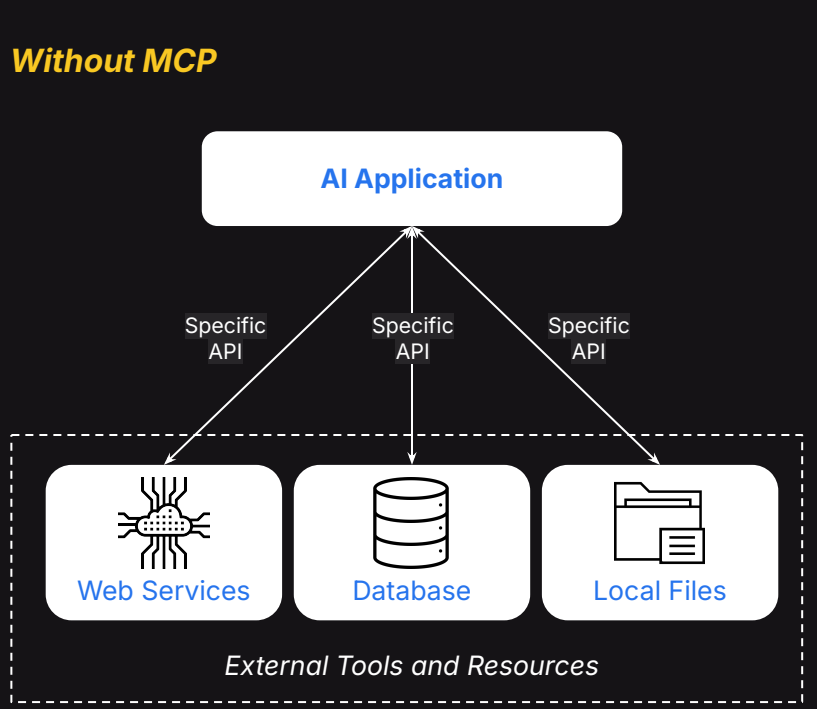
Analytics Vidhya

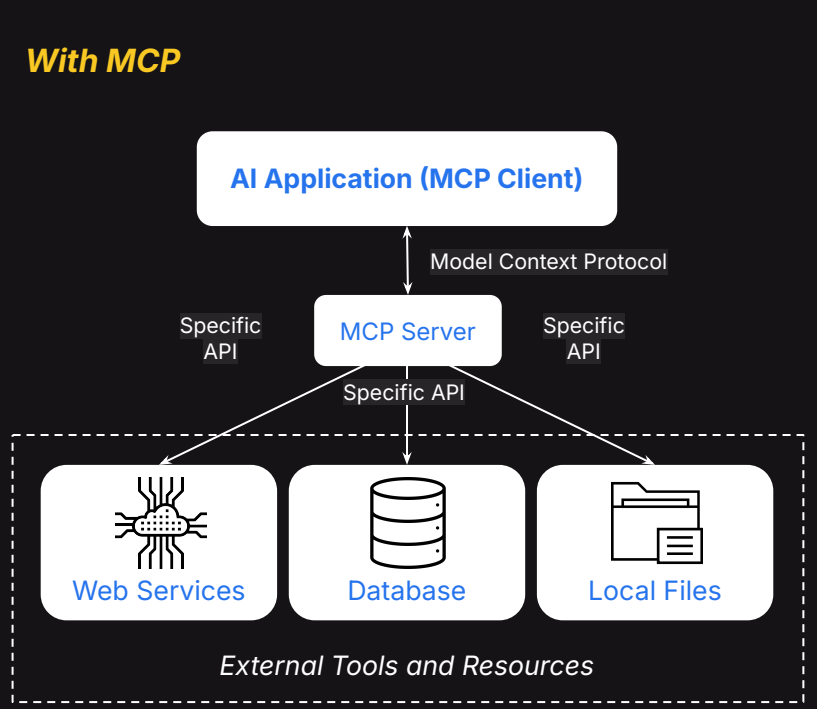
Simple Agentic Architecture with MCP



Analytics Vidhya

Simple Agentic Architecture with MCP





Analytics Vidhya

Workflow of the AI Agent: Generation Phase

3. Generation Phase: Use tool calling responses and generate a final, coherent response.

- AI Agent Actions
  - Retry failed APIs or select alternative tools
  - Reorganize results into a unified, clear itinerary
  - Generate user-friendly output
  - Ask follow-up questions in case some things are unclear (e.g. invalid destination)

Workflow of the AI Agent: Final Output

🗓️ **Day-Wise Itinerary**

Day 1 – Arrival & Check-in

- Flight: Aer Lingus EI101
- Departure: Dublin, 9:00 AM → Arrival: JFK, 11:45 AM
- Hotel: The Manhattan at Times Square
- Check-in: 3:00 PM
- Evening: Walk through Times Square, light dinner at Joe's Pizza

Day 2 – Sightseeing & Museums

- Morning: Central Park stroll & breakfast nearby
- Midday: Visit The Met Museum
- Evening: Sunset at Top of the Rock, Dinner at Serafina

Day 3 – Statue & Downtown Exploration

- Morning: Ferry to Statue of Liberty & Ellis Island
- Afternoon: Explore Wall Street & 9/11 Memorial
- Evening: Chinatown dinner and Brooklyn Bridge night walk

Day 4 – Local Vibes & Departure

- Morning: Brunch in Soho, Shopping in Chelsea Market
- Hotel Checkout: 11:00 AM
- Flight: JFK to Dublin at 6:00 PM

🛫 **Flight Details**

- **Airline:** Aer Lingus
- **Round-trip Fare:** €520
- **Baggage:** 1 checked bag + carry-on

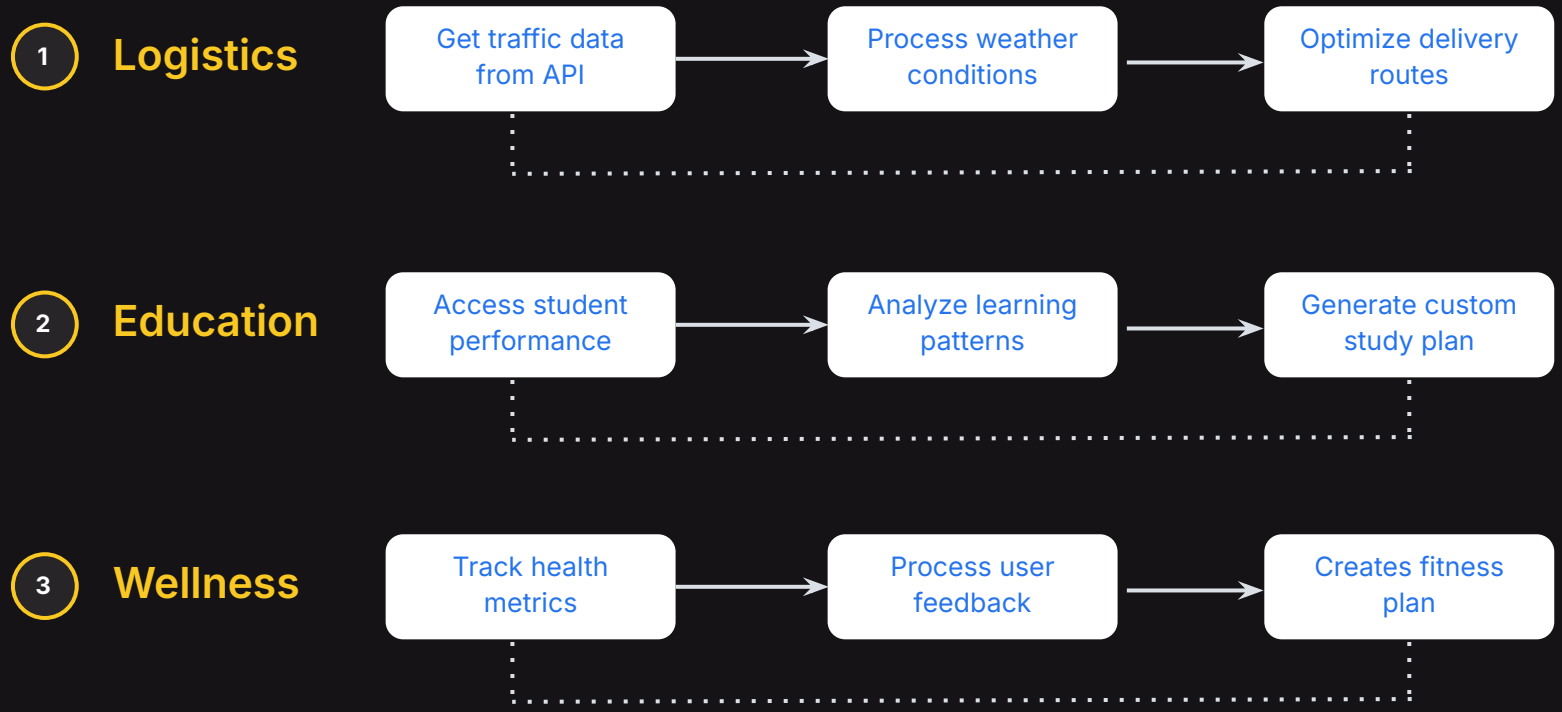
🏨 **Hotel Booking**

- **Hotel:** The Manhattan at Times Square
- **Nights:** 3
- **Total Cost:** $480
- **Link:** View Booking

🌦️ **Weather Forecast Summary**

- **Day 1:** ☁️ Cloudy, 22°C
- **Day 2:** ⛅ Partly sunny, 24°C
- **Day 3:** 🌧️ Light rain, 21°C
- **Day 4:** ☀️ Clear, 26°C

Real-World Examples of Tool Use Agents



Thanks!

Analytics Vidhya