

Digital Banking and Fintech

MODEL CONTEXT PROTOCOL

Nguyen Duc Thanh Vinh

LEARNING OBJECTIVES

Define the model context protocol and purposes of MCP.

Explore MCP Architecture and WORKFLOW.

How to use model context protocols and Practical Use Case

What is the Model Context Protocol (MCP)

MCP is an open protocol that standardizes how your **LLM applications** connect to and work with your **tools & data sources**.

REST APIs

Standardize how **web applications** interact with the **backend**

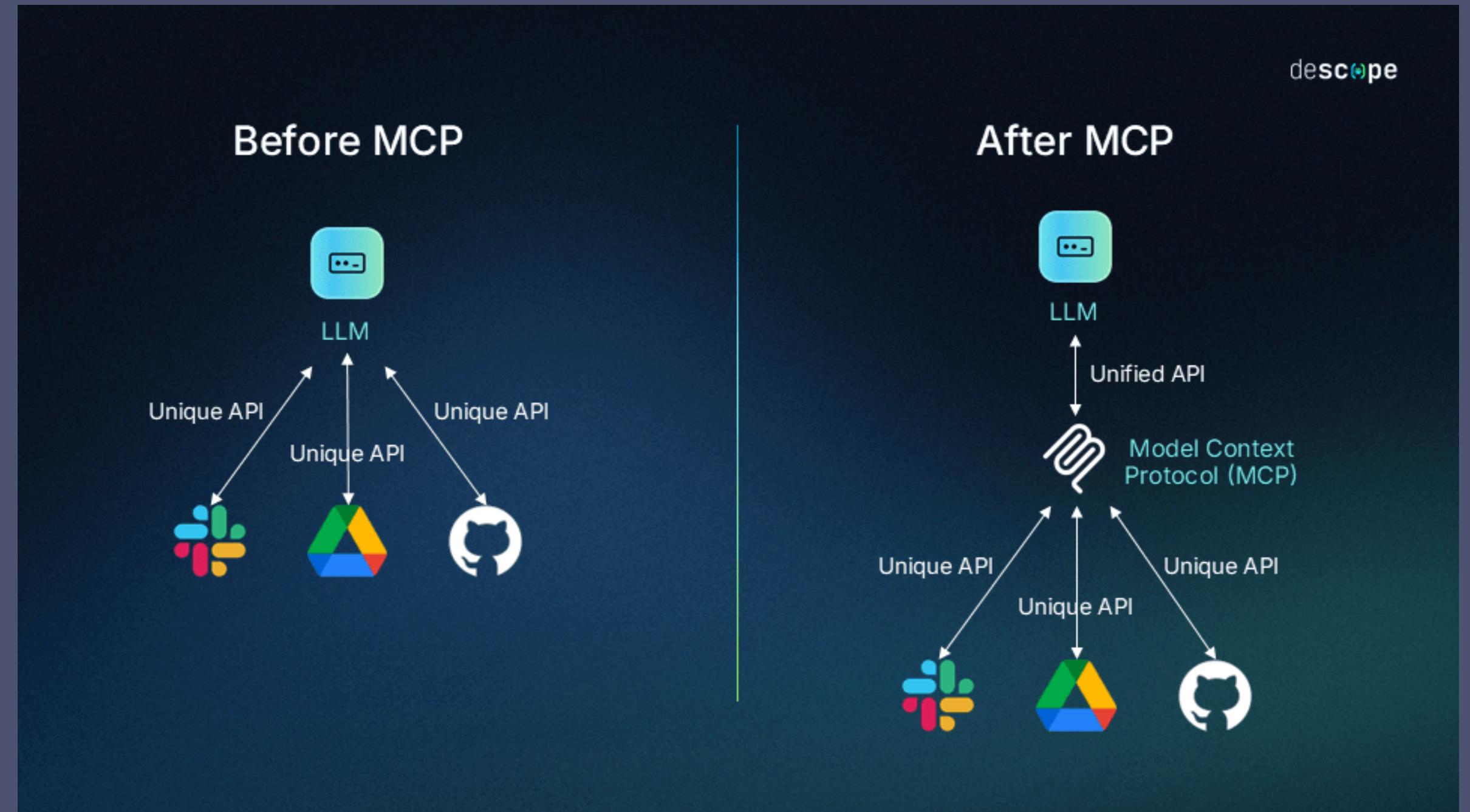
LSP

Standardizes how **IDEs** interact with **language-specific tools**

MCP

Standardizes how **AI applications** interact with **external systems**

MODEL CONTEXT PROTOCOLS



Before MCP, AI agents were static and limited to one-shot responses.
After MCP, agents became dynamic, interactive, and capable of tool use and step-by-step reasoning.

MODEL CONTEXT PROTOCOLS

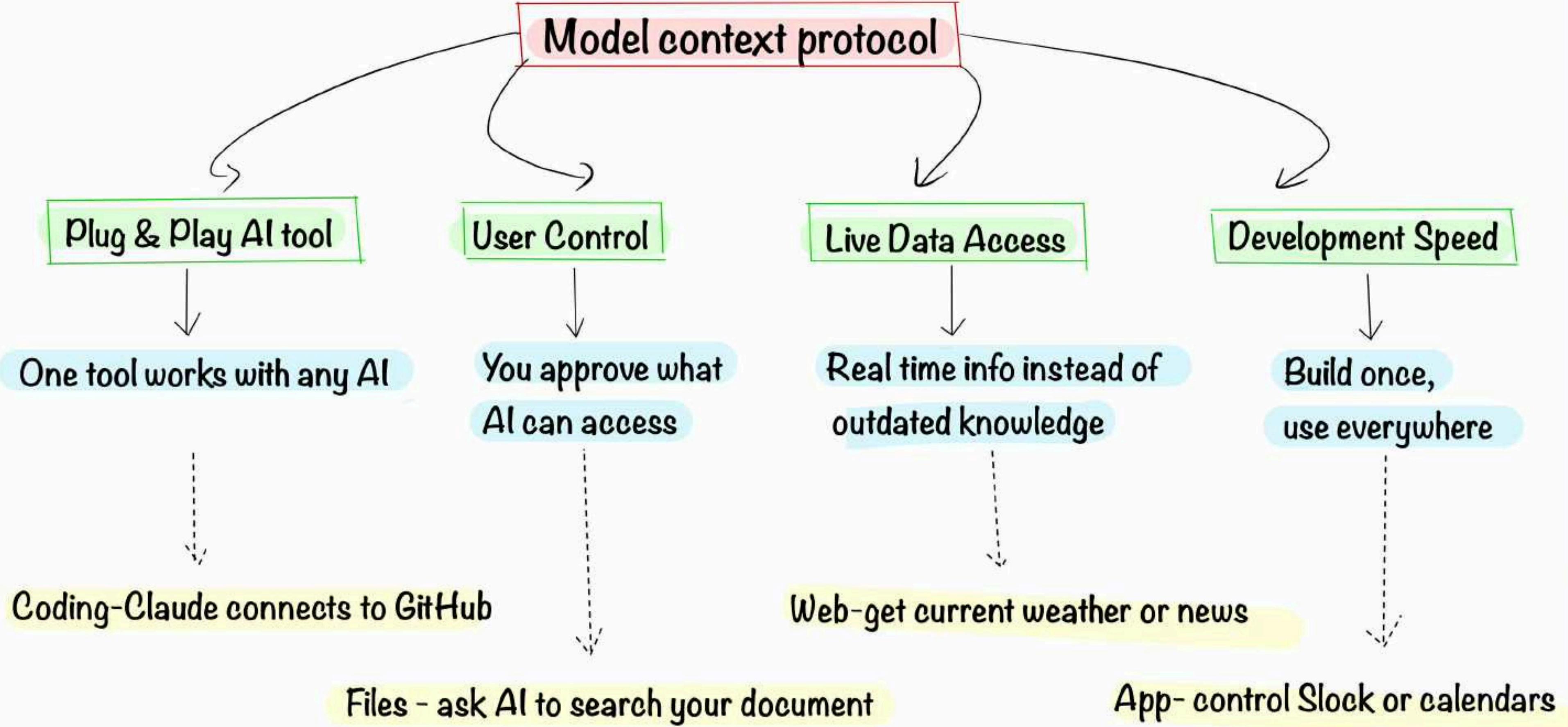
WHY MCP?

MODEL ARE ONLY AS GOOD AS THE CONTEXT GIVEN TO THEM

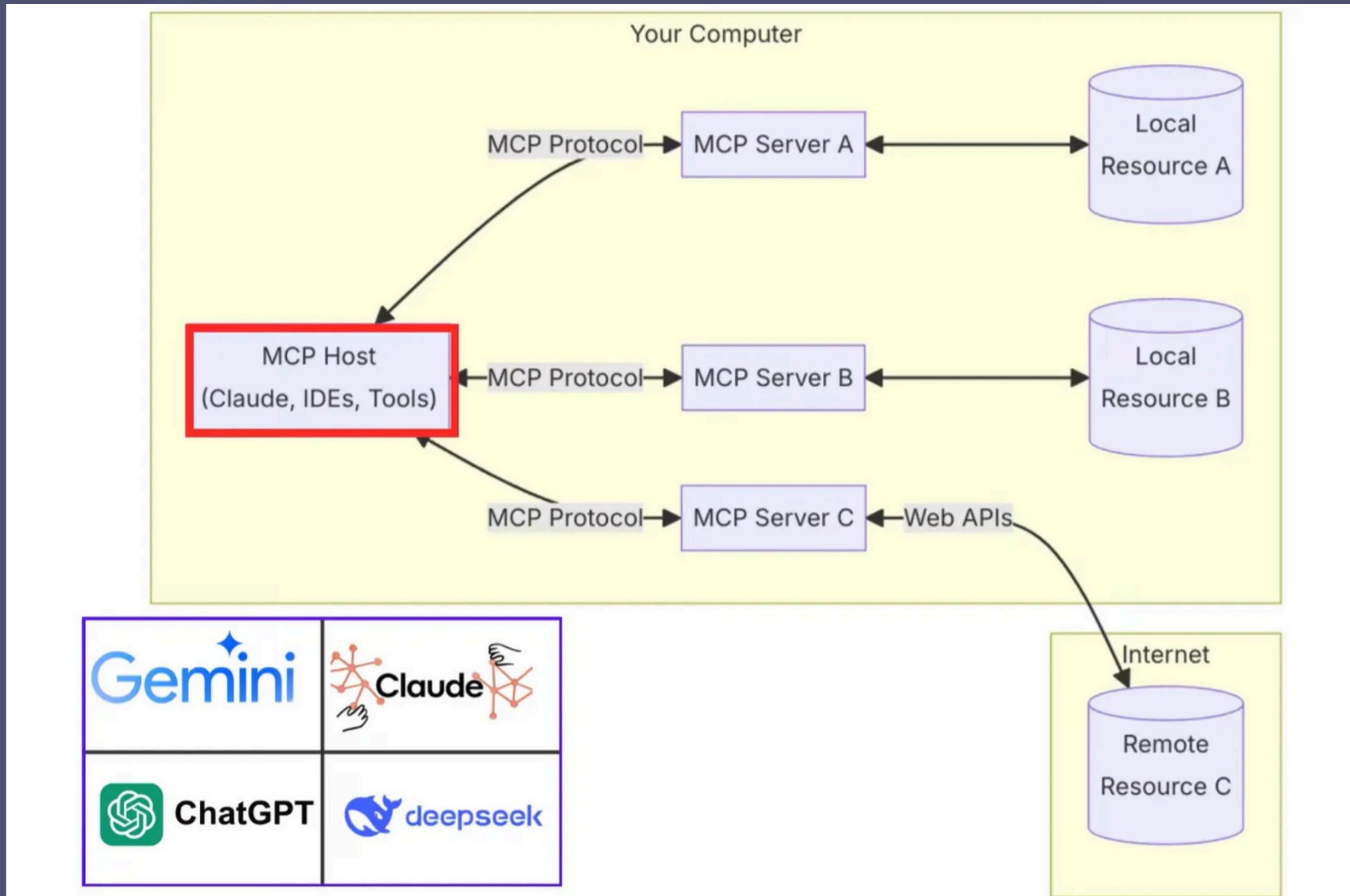
With MCP: Standardized AI Development

- + For AI application developers: connect your apps to any MCP sever with 0 additional work
- + For tool or API developers: Build an MCP sever once which can be adopted everywhere
- + For AI application users: AI applications have extensive capabilities
- + For enterprises: Clear separation of concerns between AI product teams

Model context protocol



SERVER-CLIENT ARCHITECTURE

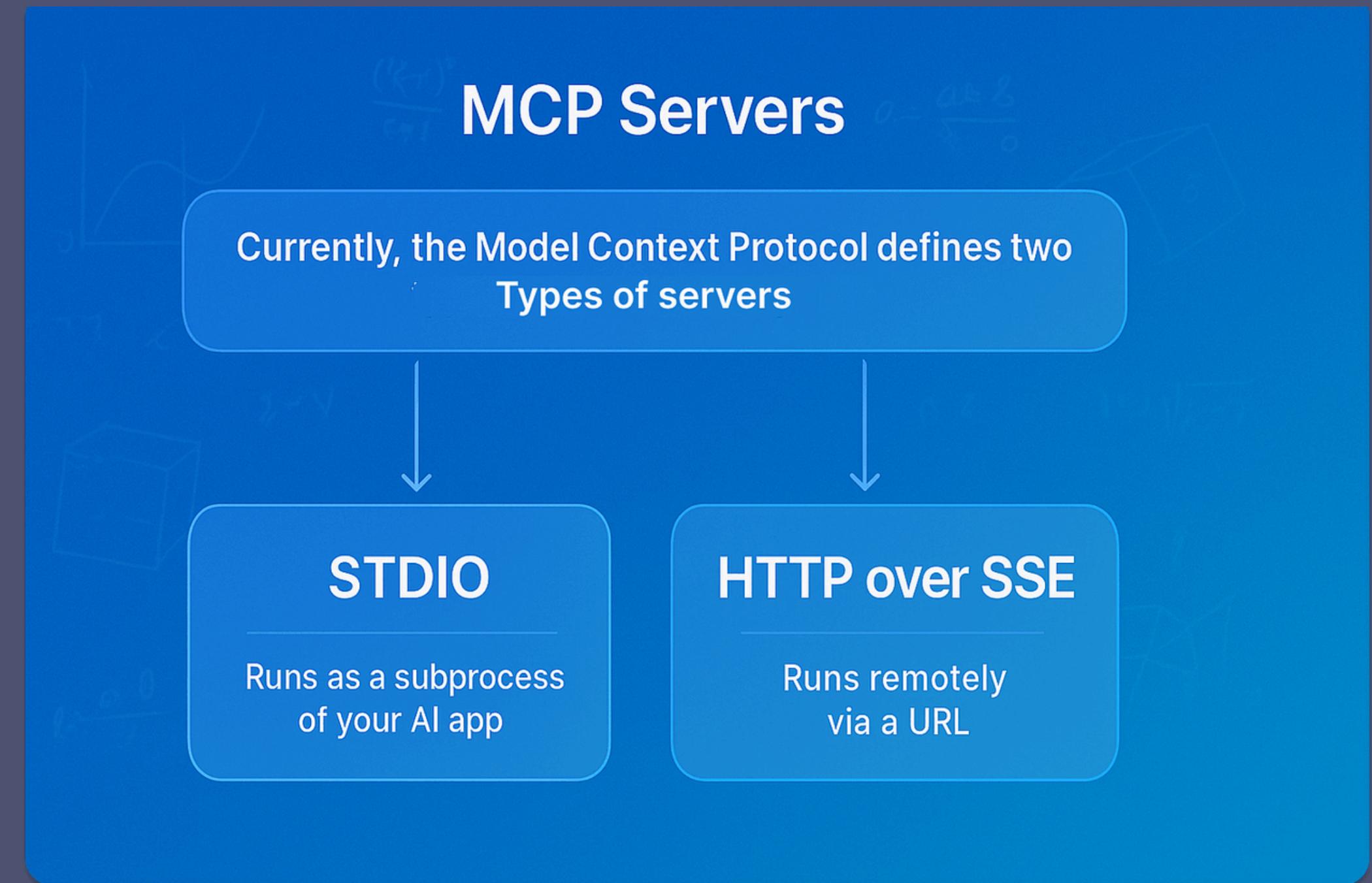


MODEL CONTEXT PROTOCOLS

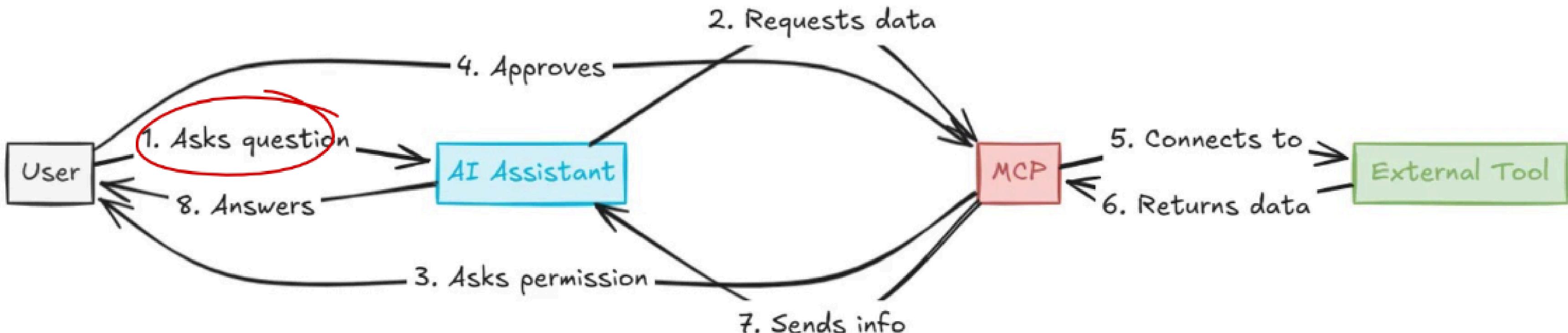
HOST: are LLM applications that want to access data through MCP(ex:
Claude desktop, IDEs, AI agents)

MCP Servers: are lightweight programs that each expose specific capabilities
through MCP.

MCP clients: maintain 1:1 connections with servers, inside the host
applications



WORKFLOW OF MCP

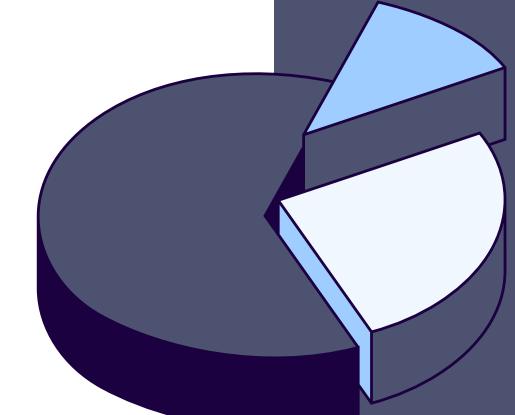


HOW TO READ LOCAL FILE AND SUMMARY VIDEO IN YOUTUBE

```
{"mcpServers": {  
    "filesystem": {  
        "command": "C:\\Program Files\\nodejs\\node.exe",  
        "args": [  
            "C:\\Users\\DELL\\AppData\\Roaming\\npm\\node_modules\\@modelcontextprotocol\\server-filesystem\\dist\\index.js",  
            "C:\\Users\\DELL\\Downloads",  
            "C:\\Users\\DELL\\OneDrive\\Máy tính"  
        ]  
    },  
    "puppeteer": {  
        "command": "C:\\Program Files\\nodejs\\node.exe",  
        "args": [ "C:\\Users\\DELL\\AppData\\Roaming\\npm\\node_modules\\@modelcontextprotocol\\server-puppeteer\\dist\\index.js"]  
    }  
}
```

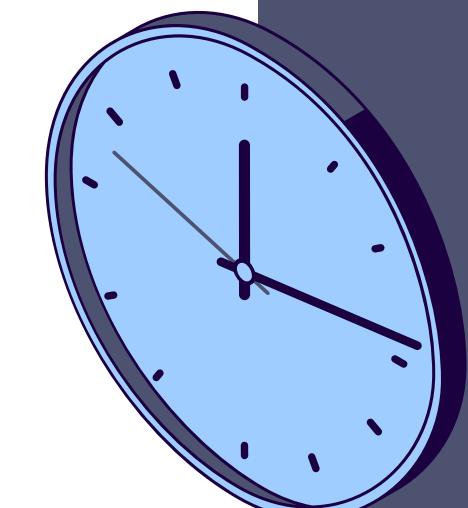
HOW TO USE MODEL CONTEXT PROTOCOLS

STEP1. Download Claude Ai



**STEP2. Download Node.js and
Install**

**Go into Node.js command
prompt**





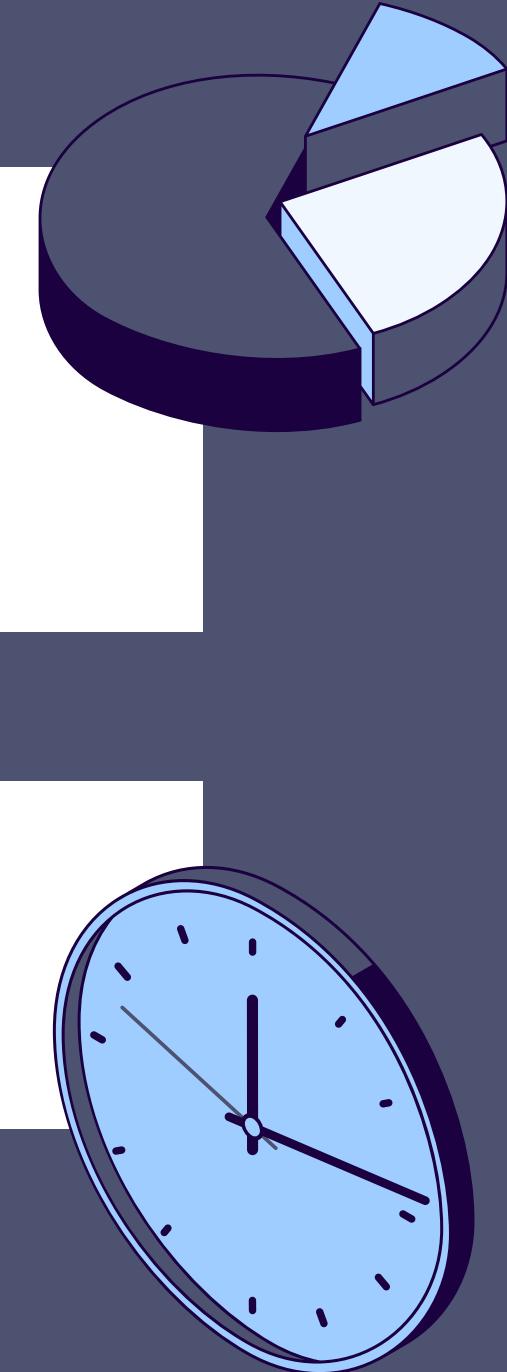
Model Context Protocol

HOW TO USE MODEL CONTEXT PROTOCOLS

STEP3. Model Context Protocol
github

Click to MCP Sever

STEP4: Edit config



Open node.js command prompt type following steps

where node

npm root -g

npm install -g @modelcontextprotocol/server-filesystem

npm list -g @modelcontextprotocol/server-filesystem

npm install -g @modelcontextprotocol/server-puppeteer

ANOTHER USE CASE: STORYBOOK GENERATOR

Storybook Image

Prompt: A cheerful traveler named Alice exploring Tokyo's cultural activities. Show her participating in a traditional Japanese tea ceremony with a kimono-clad tea master in a serene tatami room, and also learning to make sushi with a friendly chef in a bright cooking studio. Include iconic Tokyo elements like cherry blossoms, traditional architecture, and Mount Fuji in the background. The scene should feel warm, educational, and culturally immersive.



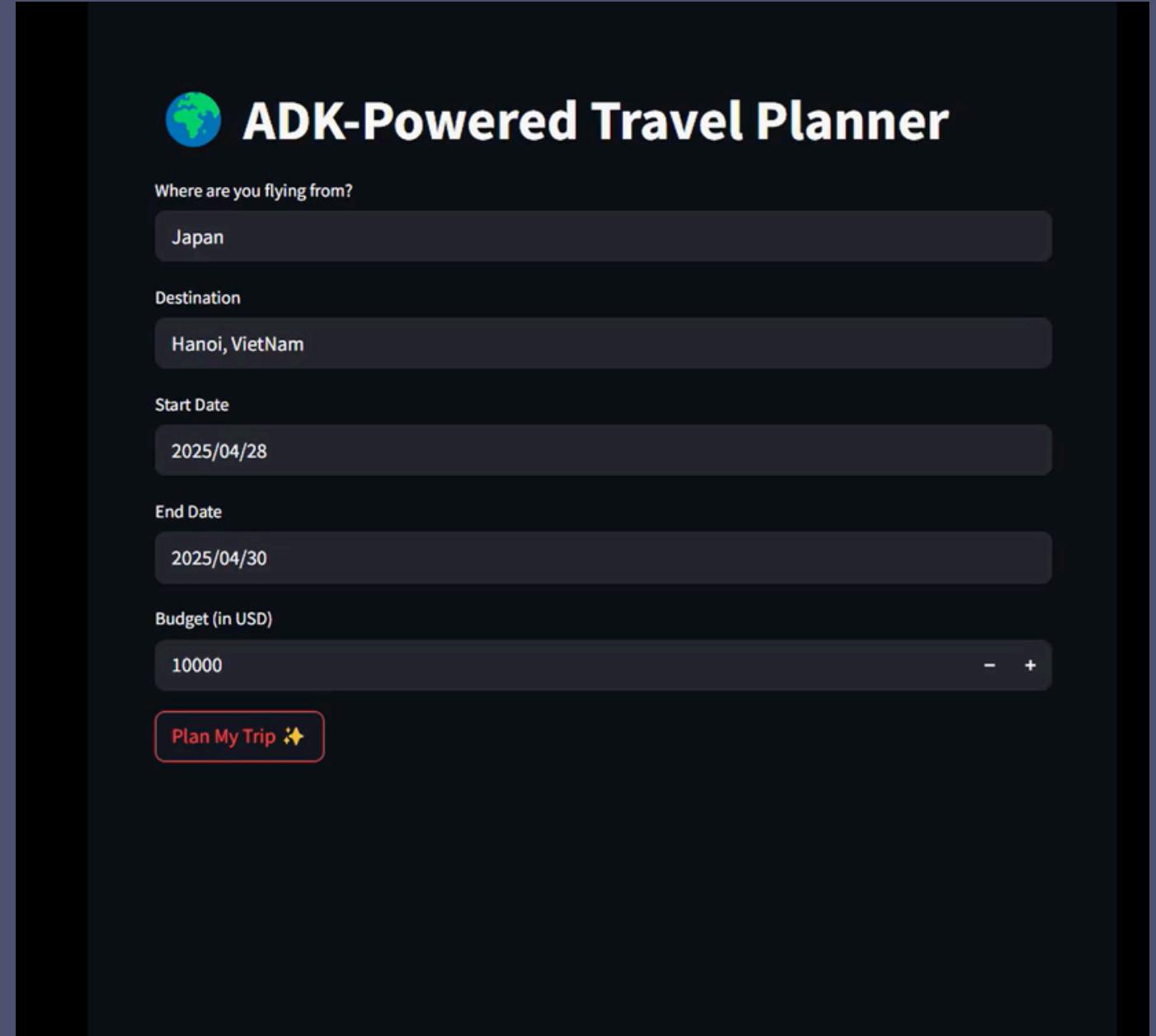
PROJECT: A MULTI-AGENT TRAVEL ASSISTANT SYSTEM

Purpose: This app's idea is to handle different user requests for Travel Planning. It implements Multi-agent technology to analyze tasks and find suitable Agents for them.

App Includes:

- Host agent: Define the task and route to a suitable agent
- Flight agent: Search for real-time flight information
- Stay agent: Search for hotel information
- Activities agent: Search for suitable activities based on location and budget

Using FastAPI and Google's ADK (Agent Development Kit), the system integrates multi-agent that leverages a web search tool (powered by Tavily Search MCP) to gather real-time data.



TAVILY SEARCH MCP SERVER

```
from fastmcp import FastMCP
from duckduckgo_search import DDGS
import requests
from bs4 import BeautifulSoup
mcp = FastMCP("DuckDuckGo MCP Server")
@mcp.tool()
def duckduckgo_search(query: str, max_results: int = 3) -> list:
    """
    Perform a DuckDuckGo search and return a list of results.
    Each result includes the title, snippet, and URL.
    """
    results = []
    with DDGS() as ddgs:
        for r in ddgs.text(query, max_results=max_results):
            results.append({
                "title": r.get("title"),
                "snippet": r.get("body"),
                "url": r.get("href")
            })
    return results
@mcp.tool()
def fetch_page_content(url: str) -> str:
    """
    Fetch and return the textual content of the specified URL.
    """
    try:
        response = requests.get(url, timeout=10)
        response.raise_for_status()
        soup = BeautifulSoup(response.text, "html.parser")
        # Extract text from paragraphs
        paragraphs = soup.find_all("p")
        text_content = "\n".join(p.get_text() for p in paragraphs if p.get_text())
        return text_content.strip()
    except Exception as e:
        return f"Error fetching content from {url}: {e}"
if __name__ == "__main__":
    mcp.run()
```

Connect Agent to MCP sever

```
from google.adk.tools.mcp_tool.mcp_toolset import MCPToolset, StdioConnectionParams
SCRIPT_PATH = os.path.abspath("mcp_tools.py")

flight_agent = Agent(
    name="flight_agent",
    model=AGENT_MODEL,
    description="Suggests flight options for a destination.",
    instruction=(
        "Given a destination, travel dates, and budget, suggest 1-2 realistic flight options."
        "Include airline name, price, and departure time. Ensure flights fit within the budget."
        "You must use the web search tool to find real-time flight information."
        "You must provide sources to support the suggestions about flight (links, urls,...)"
    ),
    tools=[
        MCPToolset(
            connection_params=StdioConnectionParams(
                server_params={
                    "command": "python3",
                    "args": [SCRIPT_PATH]
                }
            )
        )
    ]
)
```



ADK-Powered Travel Planner

Where are you flying from?

Japan

Destination

Hanoi,

Press Enter to apply

Start Date

2025/04/28

End Date

2025/04/28

Budget (in USD)

100

- +

Plan My Trip ✨

Thank You