Introduction to MCP

Brandon Walton

Who are you?

I'm Brandon Walton!

Where are you from?

Tuscaloosa, Alabama

College Education

Louisiana State University – BS in Computer Science, May 2025

What do you do?

Machine Learning Software Developer ~ *LSU*

70%: Leading the development of MikeGPT.

30%: Conducting Cyber Security Research w/ LLMs

Who is this guy?

What is your research?

Enhancing Malware Analysis with LLMs

Publications:
- *"Exploring Large Language Models for Semantic Analysis and Categorization of Android Malware"*, Annual Computer Science Applications Conference, ACSAC (Hawaii!) https://arxiv.org/abs/2501.04848

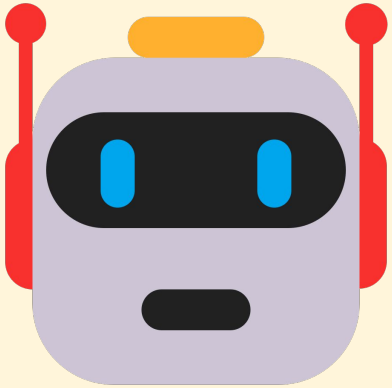
Fun Facts!

I'm a 5th degree black belt in Taekwondo at Tiger Rock Martial Arts.

I was the Sousaphone (Tuba) section leader in Tiger Band during the 2024-2025 season

What is the MCP?

Model Context Protocol (MCP)



Model

The specific LLM used in the main application.

Examples: OpenAI model, Llama model, Gemini model, etc.

Context

Any additional information the LLM uses to understand or perform tasks.

Examples: documents, summaries, notes, or other supplemental materials.

Protocol

The standard rules that define how the client, server, and data sources send and receive information.

Other Protocols: HTTPS, REST, TCP

r/mcp • 3mo ago
Jaydgaitin

Can someone explain to me what an MCP is?

r/devsecops · 5mo ago

what is an MCP and why should I care

r/ClaudeAI • 1y ago
[deleted]

I don't understand what MCP does, and at this point i'm too afraid to ask

r/programming • 7mo ago
Party-Tower-5475

What the heck is MCP? And why is everybody talking about it?

r/LangChain • 6mo ago
teenfoilhat

Why is MCP so hard to understand?

What is the MCP?

- **MCP**: Model Context Protocol
- An open-source standard protocol for connecting AI applications to external systems.
- USB-C port for AI applications.

What can the MCP do?

- Connect to data sources, tools, and workflows, which allow LLMs to perform actions and gather needed information.
- ***Examples:*** Local Files, Databases, Search Engines (Google), APIs, etc.

Why do we need the MCP?

- **Standardization:** Provides a common framework for AI applications to connect to tools and data sources consistently.
- You can choose not to use MCP; however, any future integrations will require a **manual** solution.

MCP Host

The main AI application that manages one or more MCP clients.

MCP Client

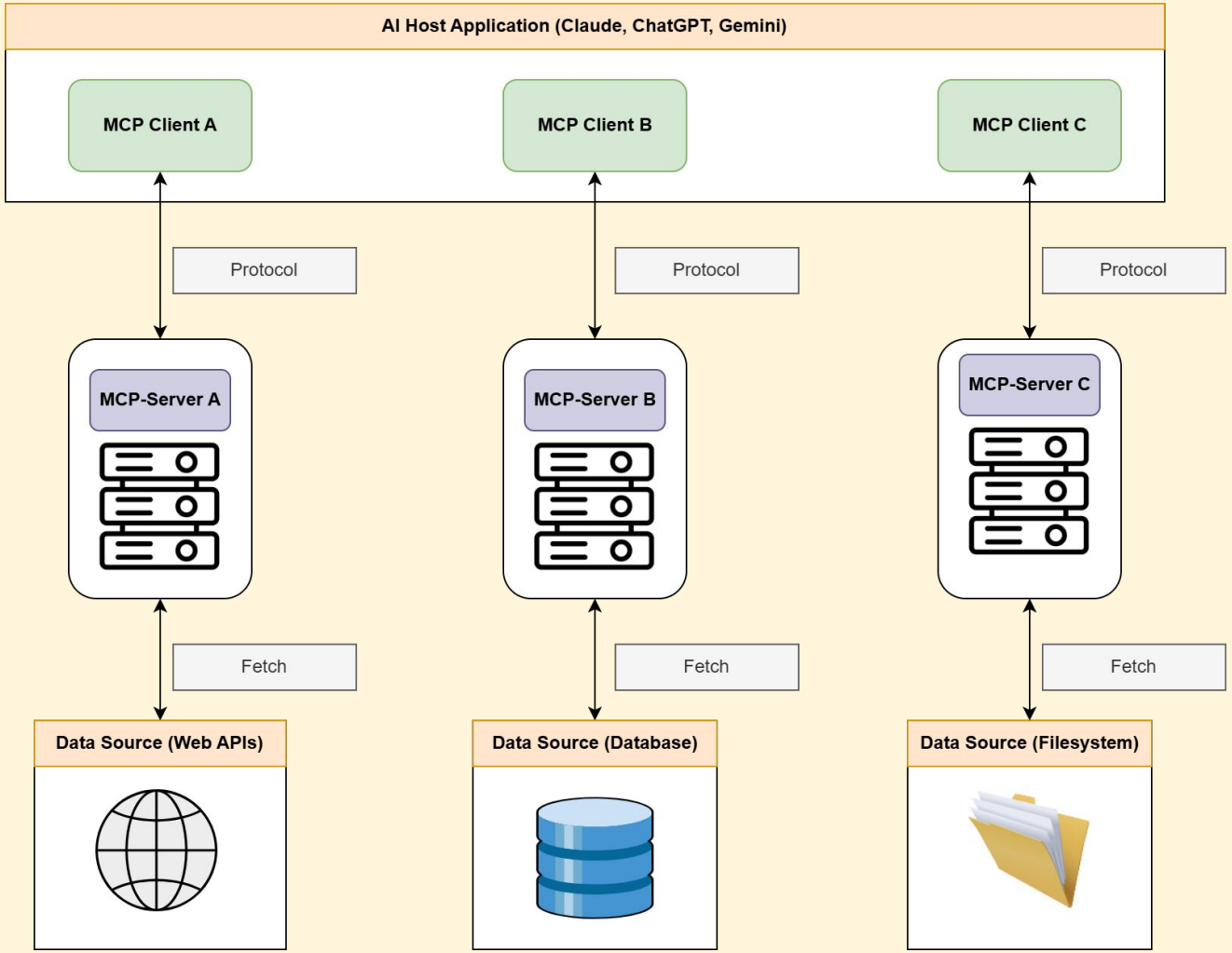
Maintains a connection to the MCP server and retrieves information for the host.

MCP Server

A server that provides data, tools, and capabilities to the MCP Client.

Data Source

The system that stores information for the server to access and pass to the client.



MCP Primitives

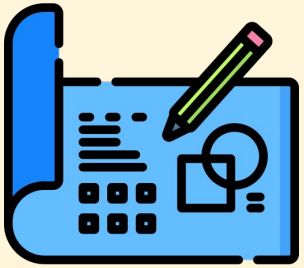
- Most important concept within the MCP
- Define what clients and servers offer each other
- Three core primmativies: Tools, Resources, and Prompts

Tools

Executable functions that AI applications can invoke to perform actions **(Perform)**

Resources

Data sources that provide contextual information to AI Applications **(Provide)**



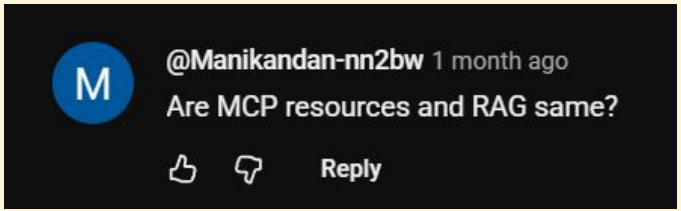
Prompts

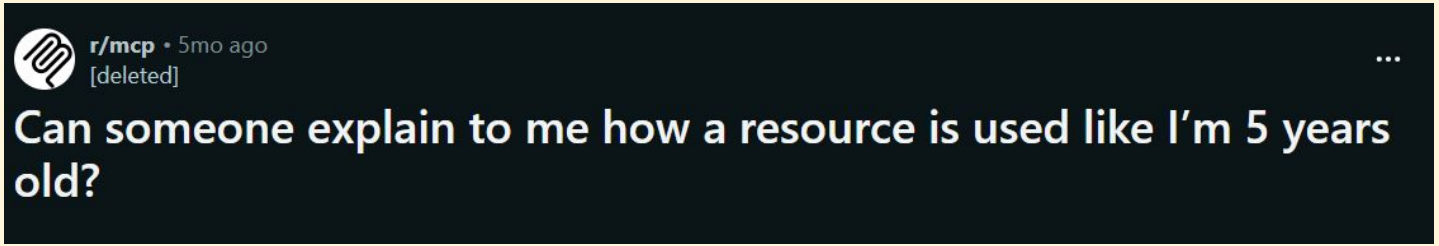
Reusable templates that help structure interactions with language models **(Plan)**

Prim. Type	Which P?	When to Use?	Notes
Tools	Perform	Use when the LLM needs to **act** or **compute** something outside its internal knowledge (e.g., call APIs, perform calculations, generate images).	The LLM can **automatically** decide which tool to use for a user's query, determine the appropriate inputs, and execute the tool.
Resources	Provide	Use when the LLM needs **additional context** or factual information that isn't included in the base model.	Designed to be **application-driven**, with users or the app deciding how to provide and use the context.
Prompt	Plan	Use prompts to **shape** the model's reasoning, **guide** its response style, or define specific tasks, without relying on external tools or data.	—

What about RAG?

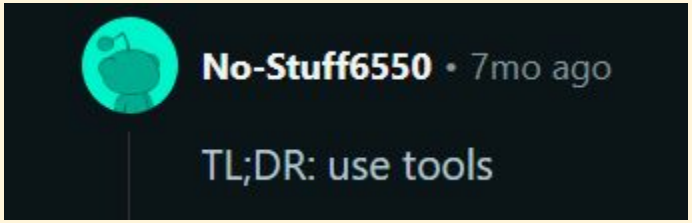
MCP Tools vs Resources: When does data retrieval become a “tool” operation?





TLDR: Use Tools!

(you'll be using tools 99% percent of the time)



Tool Structure

1. A callable function or method the **server exposes** to clients.
2. MCP uses **JSON-RPC 2.0** (JavaScript Object Notation Remote Procedure Call)
    a. A lightweight, standardized protocol that executes functions on a server via JSON.
3. Each Tool has the following
    a. *Name*: Unique identifier
    b. *Description*: What it does (Usually is provided to the LLM)
    c. *Parameters (Input Schema)*: The input it accepts
    d. *Response (Output Schema)*: The result/output

User Interaction Model

1. Tools are designed to be **model-controlled**
2. LLM can **discover** and **automatically** invoke tools based on the user's prompts and contextual understanding
3. Applications Should:
    a. Provide UI that makes clear which tools are being **exposed** to the LLM
    b. **Display clear** visual indicators when tools are invoked
    c. Present **confirmation prompts** to the user for operations (Human in the loop)

Tool Structure

```
{
  "name": "get_weather_data",
  "title": "Weather Data Retriever",
  "description": "Get current weather data for a location",
  "inputSchema": {
    "type": "object",
    "properties": {
      "location": {
        "type": "string",
        "description": "City name or zip code"
      }
    },
    "required": ["location"]
  },
  "outputSchema": {
    "type": "object",
    "properties": {
      "temperature": {
        "type": "number",
        "description": "Temperature in celsius"
      },
      "conditions": {
        "type": "string",
        "description": "Weather conditions description"
      },
      "humidity": {
        "type": "number",
        "description": "Humidity percentage"
      }
    },
    "required": ["temperature", "conditions", "humidity"]
  }
}
```

Tool Request

```
{
  "jsonrpc": "2.0",
  "id": 2,
  "method": "tools/call",
  "params": {
    "name": "get_weather",
    "arguments": {
      "location": "New York"
    }
  }
}
```

Tool Response

```
{
  "jsonrpc": "2.0",
  "id": 5,
  "result": {
    "content": [
      {
        "type": "text",
        "text": "{\"temperature\": 22.5, \"conditions\": \"Partly cloudy\", \"humidi
      }
    ],
    "structuredContent": {
      "temperature": 22.5,
      "conditions": "Partly cloudy",
      "humidity": 65
    }
  }
}
```

Tool Security

1. Servers **MUST**:
   a. Validate all tool inputs
   b. Implement proper access controls
   c. Rate limit tool invocations
   d. Sanitize tool outputs

2. Clients **SHOULD**:
   a. Show tool inputs to the user before calling the server
   b. Validate tool results before passing to LLM
   c. Implement timeouts for tool calls
   d. Log tool usage for audit purposes

3. Do **NOT** trust random MCP servers:
   a. *Malicious Content*: Servers can host or serve malware, scams, or harmful files.
   b. *Prompt Injection*: Servers can include hidden/manipulative instructions in text or metadata.
   c. *Lack of Authentication*: Many unofficial MCP servers don't verify identity
   d. *Context Bloat*: Servers can overload your context window **($$$)**

Tool Selection

Select the best tool(s) for the user's query

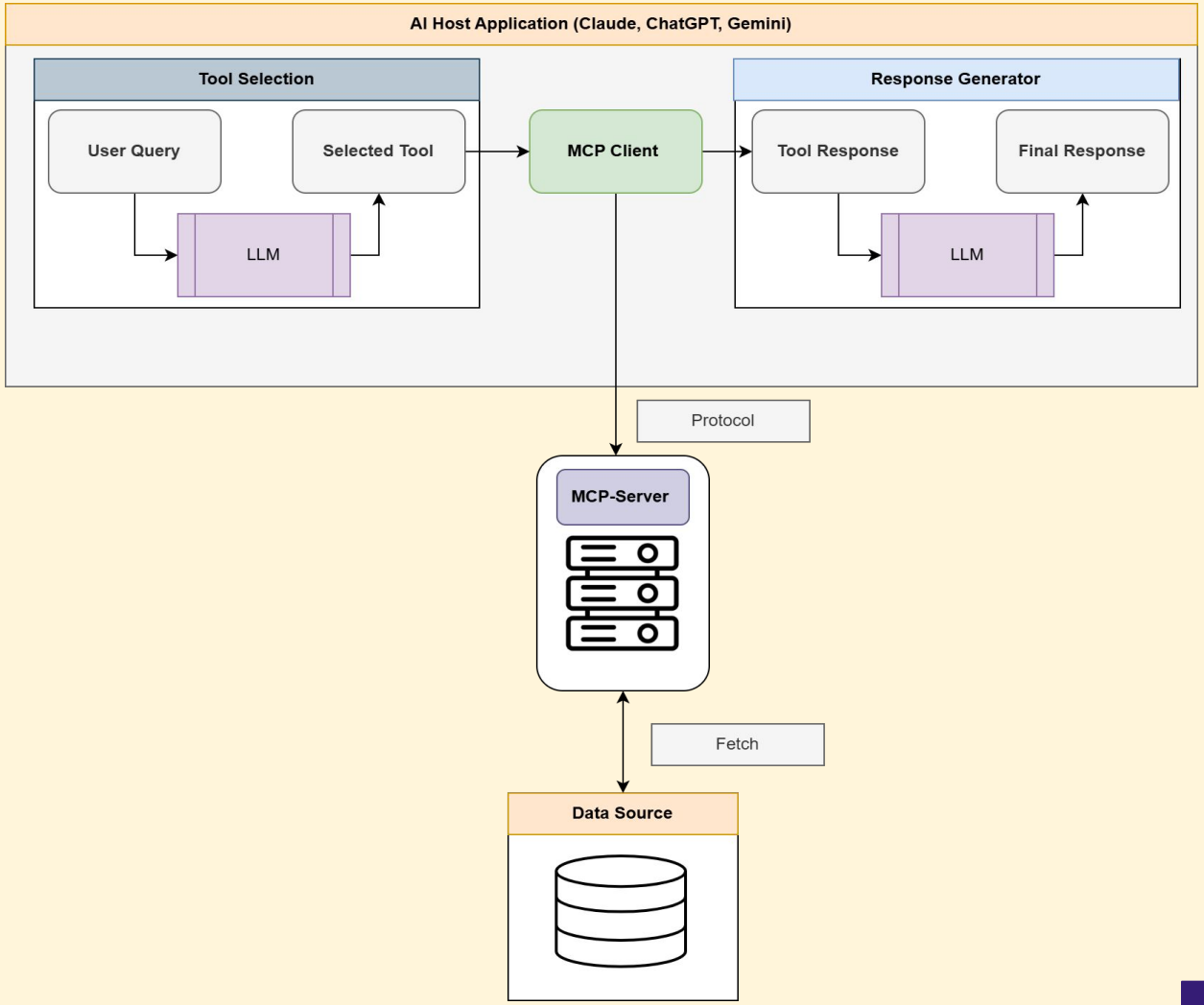
Manage tools appropriately, simply adding more without coordination leads to poor scaling (index carefully)

Allow for multi-tool execution when tasks require combined capabilities

Response Generator

Generates the final response displayed to the user.

Sometimes the tool response has the final answer. However, I would **highly recommend** validating the response and generating your own.



Code!

(Github: https://github.com/BJW101102/MCP-Template)