Full MCP client and server workflow chain

Introduction

In this lesson, we tackled the **full workflow** of how an MCP client and server communicate. Our goal was to clarify the intricacies behind this process, especially when connecting to an MCP server and running queries.

Core Components

- **MCP Host**: Think of this as your application interface, like the Anthropic desktop app.
- **MCP Client**: The part of the application that connects to MCP servers.
- **MCP Server**: The local or remote service that handles requests.
- **External Service**: An external source, such as a website or database, that provides additional data.

Workflow Overview

Initialization

- **Starting the Application**: We began by double-clicking the app, triggering the MCP client to check its configuration list for available MCP servers and start them.
- **Tool Listing**: The MCP servers respond with their available tools and details, which the MCP client collects.

User Interaction

- **User Query**: The user asks a question, like "What's the weather in NYC?" into the interface.
- **System Prompt**: The MCP host uses a system prompt to guide the language model (LLM) in selecting the right tool, such as `GetWeather`.

Tool Execution

- **Tool Selection**: The LLM collaborates with the MCP client to select the appropriate tool and construct necessary arguments (e.g., converting "NYC" to "New York City").
- **Calling the Tool**: The MCP client instructs the server to run the selected tool with the provided arguments.

Data Processing

- **Server-Side Logic**: The MCP server executes the tool logic, contacts the required APIs or databases, and retrieves data (e.g., weather data from a National Service Weather API).
- **Response Delivery**: The resulting data, such as "warm and sunny," is sent back to the MCP client.

Final Response

- **Client to Host**: The MCP client shares the data with the MCP host.
- **User-Facing Answer**: The LLM constructs a user-friendly response, like "The weather in NYC is warm and sunny."

Key Points

- The MCP framework automates much of this process, reducing the need for manual logic construction.
- Understanding the behind-the-scenes workflow clarifies how MCP clients and servers seamlessly interact.
- **MCP Servers**: Serve modular purposes, meaning they can integrate multiple tools to make them highly adaptable.

Feel free to ask any questions, and we'll advance to the next part of your learning journey!