MCP Flow in Java WeatherApp with LLM + AccuWeather

This document explains how MCP (Model Context Protocol) roles map to a Java WeatherApp integrated with an LLM and using the AccuWeather API as a tool.

---

1. User → WeatherApp
- User types: "What's the weather in London?"

2. MCP Host = Java WeatherApp (with LLM inside)
- The host runs the LLM.
- The LLM sees the question and says: "I don't know, but I have a tool for weather."

3. MCP Client = The bridge layer inside your app
- The Java code that knows how to call the AccuWeather API.
- Example: WeatherClient.getWeather("London");
- Acts as an adapter: formats requests, handles responses.

4. MCP Server = AccuWeather API
- Provides the actual weather data.
- Example endpoint: GET https://api.accuweather.com/currentconditions/v1/London
- Response: { "temperature": 18, "condition": "Cloudy" }

5. MCP Client (Java WeatherClient)
- Takes the response JSON and hands it back to the LLM inside WeatherApp.

6. MCP Host (WeatherApp)
- LLM integrates it into natural language:
"It's 18°C and cloudy in London."

Mapping Table:
- Host = Your Java WeatherApp (with LLM integrated)
- Client = Java client/connector code (e.g., WeatherClient)
- Server = AccuWeather API

Summary:
- Your WeatherApp = Host
- Your connector code = Client
- AccuWeather = Server