Generative AI Terminologies

LLM, Agent & Model Context Protocol (MCP)

What is an LLM (Large Language Model)?

LLMs are smart AI tools trained to understand and generate human-like text.

🔍 **Popular LLMs:**

- **ChatGPT**
- **Google Gemini**
- **Claude**
- **DeepSeek**

✅ **What LLMs can do:**

- Answer questions
- Generate code
- Write emails or documents
- Chat like a human

🔧 **Example:**

You can ask an LLM like ChatGPT:
"Write a login test case in Java using Selenium"
It will generate the code, but it won't run the test for you.

Limitations of LLMs:

LLMs are great at "thinking" but not at "doing."
They **can't directly interact with the outside world.**

❌ **What LLMs can't do:**

- **Open a browser** or click buttons
- **Connect to a database**
- **Call APIs and get real-time responses**

They can generate the code or API request, but cannot **execute** it.

What is an Agent?

An **Agent** is a system or program that takes instructions from an LLM and performs real-world tasks using external tools.

It acts as a helper that "does" things the LLM cannot do on its own.

What is MCP (Model Context Protocol)?

MCP is a framework that connects LLMs to real-world tools (like browsers, databases, APIs).
It allows LLMs to **send commands** to these tools and get results.

Think of MCP as a **bridge** between the LLM and your computer/tools.

Examples of what MCP can do:

1. **Browser Automation**
   → Open a website, fill a form, click a button
   ✅ Done via tools like **Selenium MCP** or **Playwright MCP**
2. **Database Operations**
   → Connect to MySQL or PostgreSQL and run queries
   ✅ Done via **PostgreSQL MCP** or **MySQL MCP**
3. **API Requests**
   → Send or receive API responses
   ✅ Handled by MCP to get real-time results

What is Playwright MCP?

Playwright MCP connects LLMs to browsers using the **Playwright tool** (an alternative to Selenium).

🎯 **Key Features:**

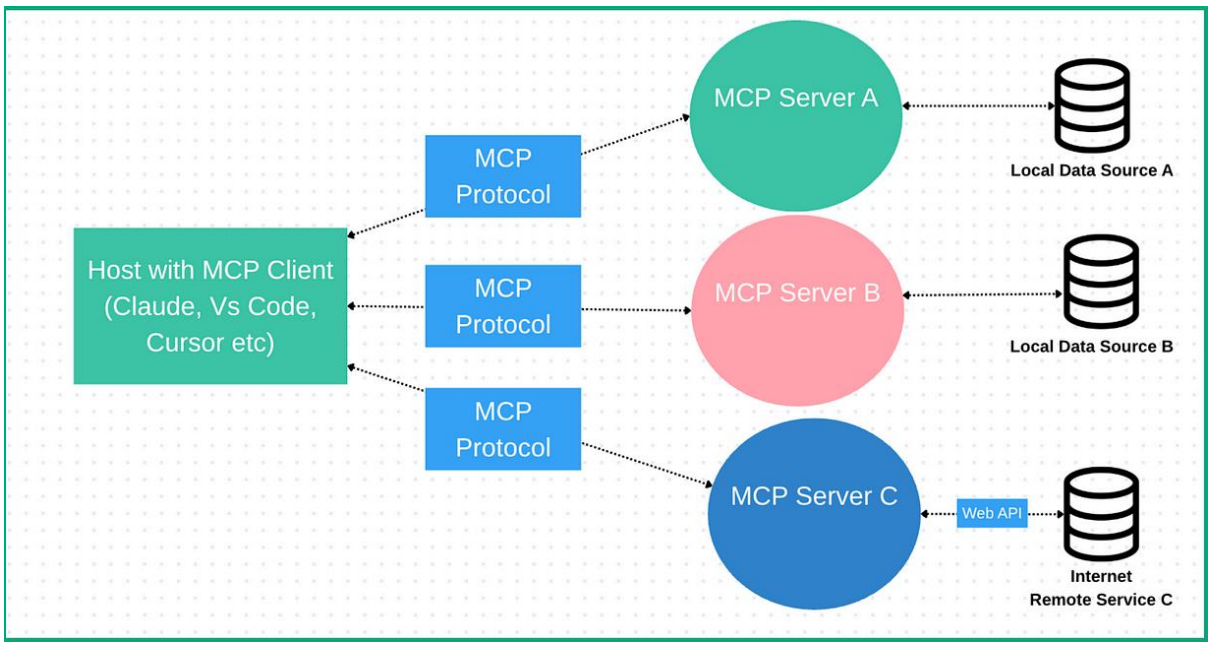
- Allows LLMs to control a browser: visit websites, fill forms, validate elements
- Uses **Accessibility Tree** to understand UI elements (not screenshots)
- Supports commands like:
  - click()
  - fill('user@example.com')

📌 **Example:**

You say to ChatGPT:
"Search for 'Selenium vs Playwright' and click the first result."

Behind the scenes, the LLM sends this instruction to **Playwright MCP**, which opens the browser and performs those actions for you.

MCP architecture and core components



At its core, MCP follows a client-server architecture where a host application can connect to multiple servers:

MCP Hosts:
Programs like Claude Desktop, Cursor, VS Code IDEs, or AI tools that want to access data through MCP

MCP Clients:
Protocol clients that maintain 1:1 connections with servers

MCP Servers:
Lightweight programs that each expose specific capabilities through the standardized MCP.Each standalone server typically focuses on a specific integration point, like GitHub for repository access or a PostgreSQL for database operations.

Local Data Sources:
Your computer's files, databases, and services that MCP servers can securely access.

Remote Services:
External systems available over the internet (e.g., through APIs) that MCP servers can connect to.