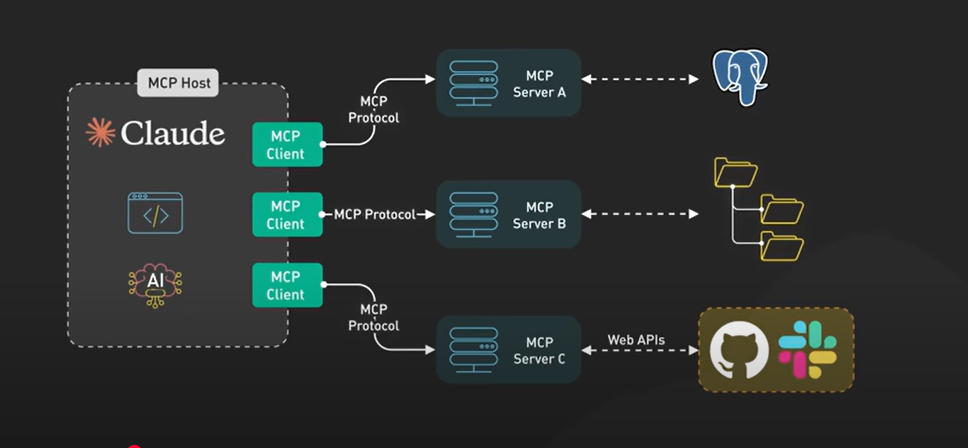
**Architecture**



1. A MCP host can have multiple clients.
2. Clients maintain 1:1 connections with servers, inside the host application.
3. Till date, to connect with multiple servers to clients, n number of connections should be made to connect with n number of servers, so we need n number of clients (or client to server connections) (although the client.py file is only one).
4. In future, maybe one client can connect to many servers. At present, one server can be dealing with multiple clients connections (it’s possible in http-streamable server)
5. Just for Note: A 1:1 connection between a client and a server means a single client is directly and exclusively connected to a single server for a specific task or communication session. In simpler terms, one client initiates a request, and that request is handled by one specific server, and the server responds directly back to that same client.  In a 1:1 connection, the focus is on a dedicated, one-to-one interaction between a single client and a single server.  A 1:1 connection differs from scenarios where a client might connect to multiple servers simultaneously or where a server handles requests from multiple clients concurrently.
6. There can be multiple servers and each server has multiple tools.
7. Server can be local or remote.
8. A host can be Claude User or IDE, where the user queries something.
9. The servers first collect information of all the tools it has like tool’s name, input schemas like arguments and the description of the tools and create a list of tools and send that to client when the client request for the tool list.
10. Now client has all information of the tools its connected to.
11. The host has sort of an LLM which may have been given some system prompt also.
12. So, when the user queries something, it goes to LLM with all the list of tools (that the client gets from server), and then LLM decide whether to call any tool or return just a text response. If it decides to call a tool, then it uses it’s function or tool calling feature, this will give the tool name and the arguments the tool needs.
13. So, the client will send request to the server to use that tool. The tool\_call request comes to MCP Server, depending upon the tool (API Calling or Database Query) the tool is executed and the result is sent back to client.
14. This result is again back to LLM which gets the result and either decide to call futher tool or return a response to the user.