**MCP Basics:**

**1/ Purpose:** protocol to connect LLMs to external data sources and tools in a universal way.

**2/ Client-Server Model:** MCP client connects to MCP servers

* **Client:** an LLM-powered app like Claude Desktop
* **Server:** programs that expose tools, resources, or prompts.

**3/ MCP Servers provide three primitives –**

* **Prompts:** reusable templates with instructions or context to guide the LLM's behaviour
* **Resources:** access to external data & files for the LLM to retrieve information
* **Tools:** functions or APIs for the LLM to invoke to perform actions

**4/ MCP Clients provide two primitives –**

**Roots:** Interfaces or entry points through which the client connects to MCP servers

**Sampling:** Mechanism to allow the server to request the client to generate text completions based on specific inputs or contexts

**5/ Bidirectional: with these primitives, MCP supports two-way communication**

For instance –

A server can request the LLM to generate completions (Sampling primitive)

A client hosting LLM can request a tool execution

**6/ Pre-built MCP servers exist for GitHub, Google Drive, Postgres, etc. Early Stage, success depends on wider adoption**

**7/ Example -** MCP server to connect an LLM to a local SQLite database, GitHub repo, and Slack workspace. LLM can query database, fetch code context, and post updates—all within a single protocol.