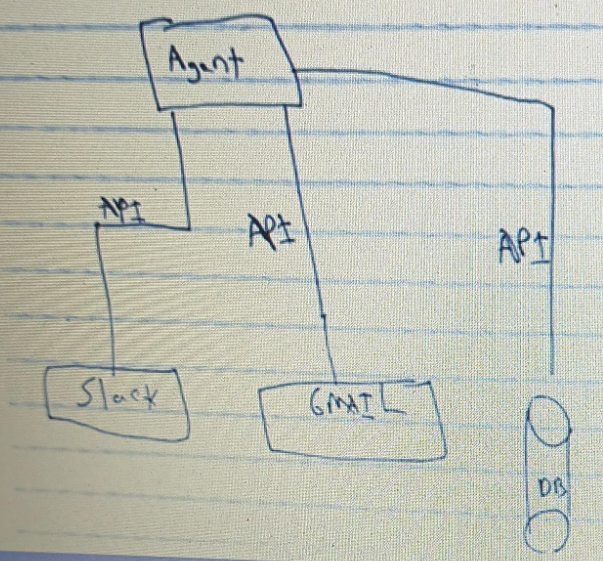
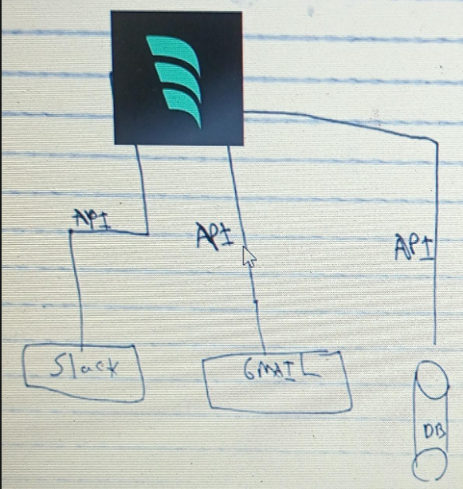
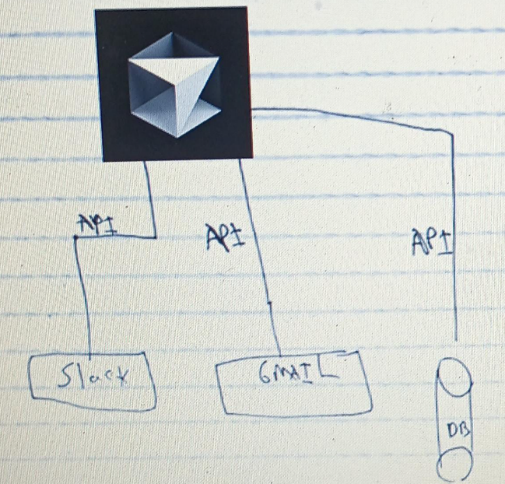
**MCP (Model Context Protocol)**

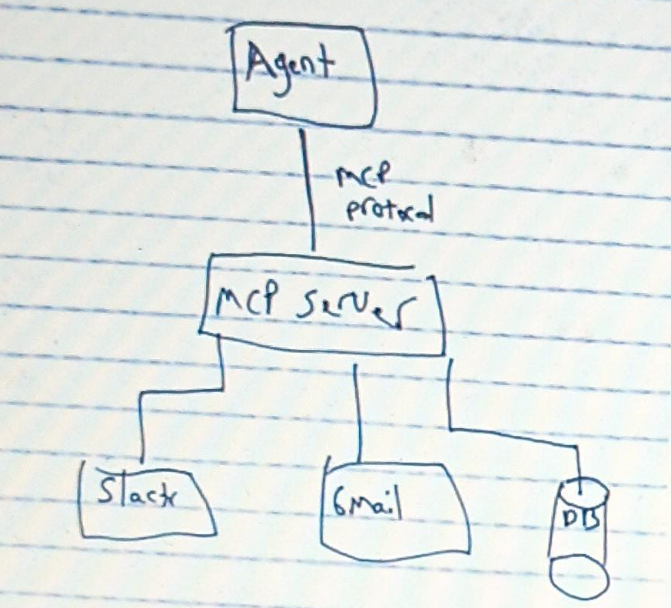
*Why MCP needed?*

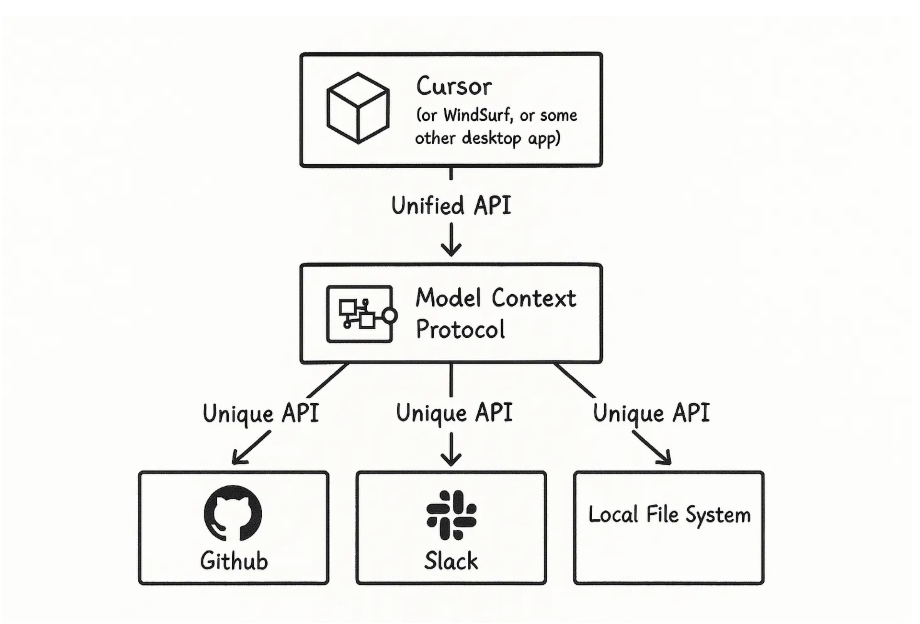


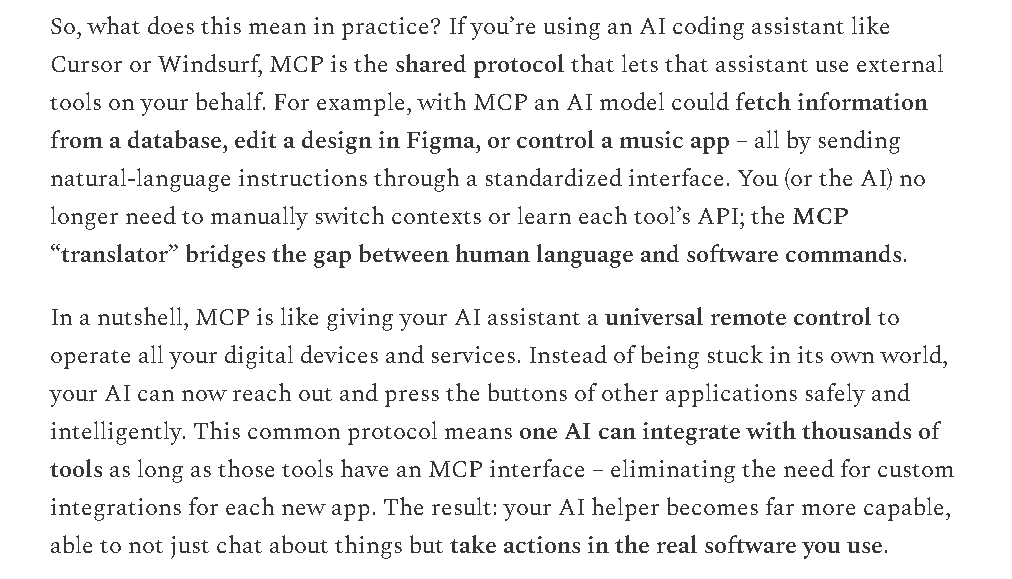
1. The **Model Context Protocol (MCP)** is needed to ensure **structured, consistent, and extensible communication between AI models and tools, interfaces, or other models**.
2. For agent’s tools, generally we would need to implement those custom functionalities like work with the Slack API or the Gmail API or make DB queries and wrap them around as a tool. (we can also use provided tools in langchain framework like gmail API but for example we do not want our agent to delete the emails and for that instead of using the generic tool by langchain we have to implement our own custom gmail API).
3. Now, for example, the agent is Cursor and all these tools are implemented for cursor only.
4. After that, another agent comes like Windsurf and want to use the functionalities, but since the previous functionalities are tailor-made to cursor only, for Windsurf we have to implement the functionalities and integration again.
5. So, it really adds up a lot of code that we need to write. And if you want to add it to Lovable, or to Bolt, or to GitHub Copilot, or to every AI code assistant, we need to do it over and over again to write the integration to that custom agent, and who wants to write a thousand integrations?



1. MCP solved this problem. It provides abstraction.
2. The idea here is that we are going to integrate only once into our MCP server. And because all the other agents support MCP, they will be able to easily connect to our MCP server. If we have implemented one agent which supports MCP, we can easily migrate to another agent which also supports MCP. So, the agent that we wrote that was compatible with Cursor is now also compatible with Windsurf. And we as the developers of the agent that is going to be used, we didn't need to add any logic, and everyone who supports the MCP protocol can use our agent and we can integrate it seamlessly.







**Another benefit of using MCP:**

For example, we are building a multi-agent workflow. And the agents need tools like calling an API, performing some Postgres database operations or doing some file system management. So, these tools code like calling API and database operation can be seen as different complete applications in their own. So, we do not want to write these tools code into our main agentic workflow code project because it will make our main agentic workflow project heavier and it will shift the developer’s attention to write these tools rather than to build the main agentic workflow. So, we can treat all these heavy tools are different microservices which can be deployed onto MCP servers.