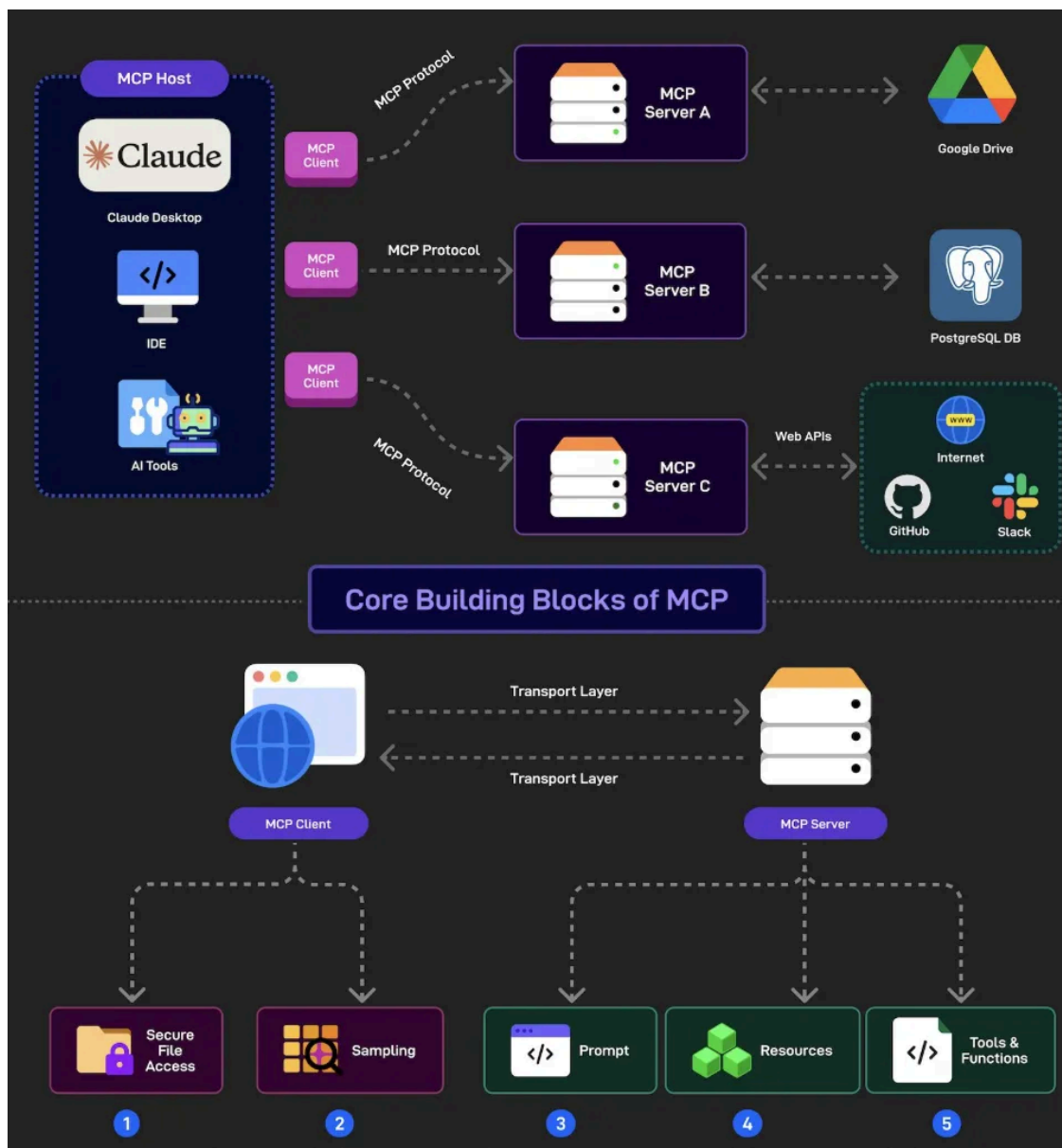


MCP不是API。AI+MCP > AI+API

原创 南七无名式 PyTorch研习社 2025年03月12日 08:01 安徽

MCP (Model Context Protocol, 模型上下文协议) 不仅仅是“另一个类似 API 的东西”。如果你认为“兄弟, 这两个想法是一样的”, 那意味着你仍然不明白。



传统的 API 使用一组固定和预定义的端点来公开其功能。例如, /products、/orders、/invoices。

如果要向 API 添加新功能, 则必须创建新端点或修改现有端点。任何需要此新功能的客户端也需要进行修改以适应更改。

仅此问题就是一场巨大的噩梦, 但还有更多。

假设你需要更改一个端点所需的参数数量。如果不破坏使用你的 API 的每个客户端, 你就无法进行此更改! 这个问题给我们带来了 API 中的“版本控制”, 任何构建过版本控制的人都知道维护它是多么痛苦。

文档是另一个问题。如果你正在构建客户端来使用 API，则需要找到其文档，该文档与 API 本身是分开的（有时不存在）。

MCP 的工作方式非常不同：

首先，MCP 服务器会将其功能公开为具有语义描述的“工具”。这很重要！每个工具都是自我描述的，并包含有关工具功能、每个参数的含义、预期输出以及约束和限制的信息。

你不需要单独的文档，因为界面本身就是文档！

我最喜欢的部分之一是当你需要进行更改时：

假设你更改了服务器中某个工具所需的参数数量。使用 MCP，你不会破坏使用服务器的任何客户端。它们将动态适应变化！

如果你添加新工具，你也不需要修改客户端。它们会自动发现该工具并在适当的时候开始使用它！

但这只是乐趣的开始：

你可以设置你的工具，使其根据上下文可用。例如，MCP 服务器可以公开一个工具，仅向那些先登录的客户端发送消息。

还有更多，但我认为我不需要继续重复这个老生常谈。

总之：AI + MCP > AI + API

<https://github.com/modelcontextprotocol>



PyTorch研习社

打破知识壁垒，做一名知识的传播者

718篇原创内容



公众号