

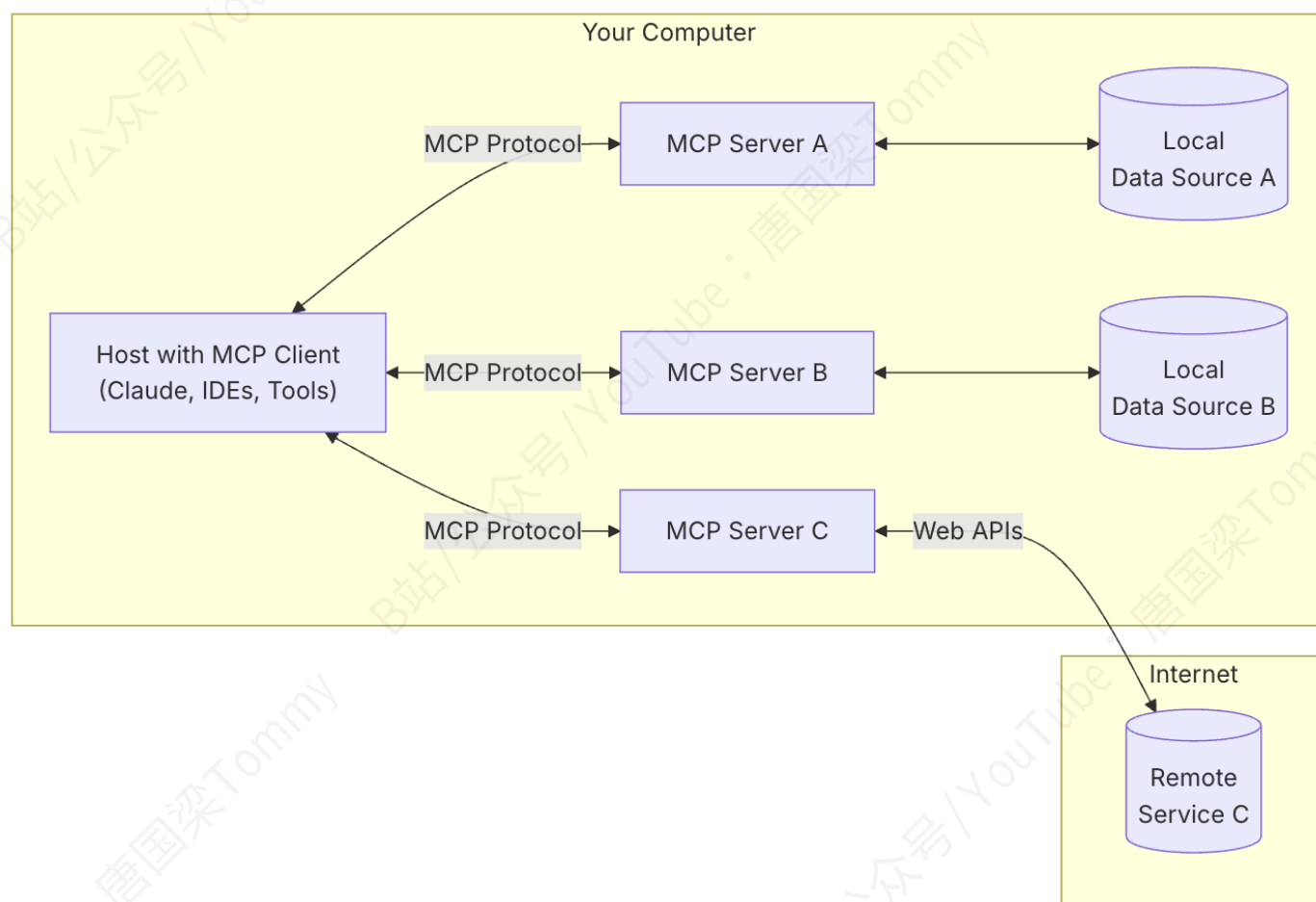
MCP 模型上下文协议

1. MCP概述与背景

MCP 由 Anthropic 于 2024 年 11 月推出，是一种开放协议，旨在标准化大语言模型（LLMs）应用程序与外部数据源和工具之间的交互方式。

MCP的核心在于建立一个标准化的通信层，使得LLMs能够在处理用户请求或执行任务时，通过MCP客户端向MCP服务器发送请求。

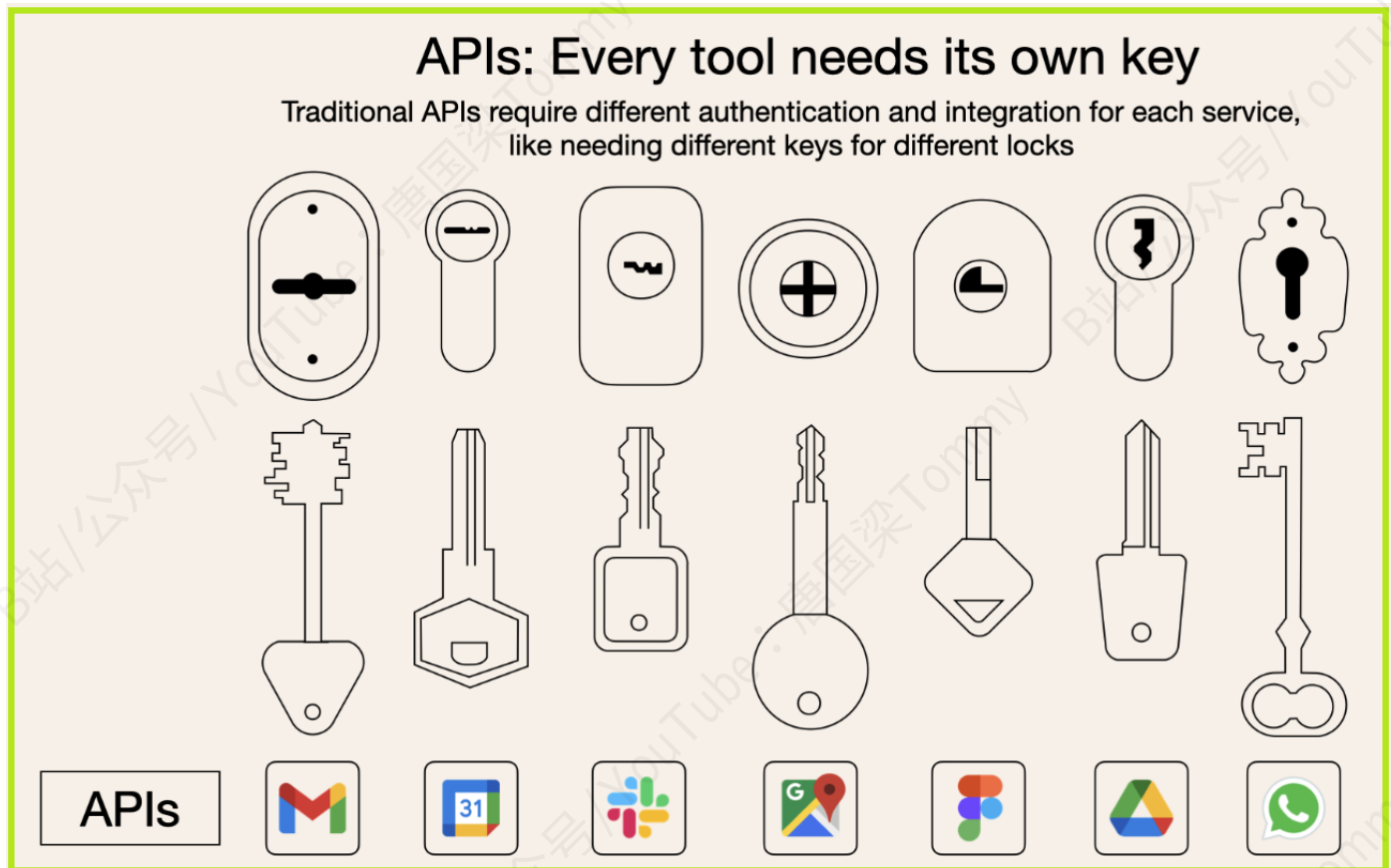
MCP服务器则负责与相应的外部数据源或工具进行交互，获取数据并按照MCP协议规范进行格式化，最后将格式化后的数据返回给LLM。这一机制使LLM能够获取更准确、更相关的信息，从而生成更高质量的回复或执行用户请求的操作。



2. MCP解决的核心问题

MCP的出现解决了大模型时代三个关键问题：**1** 数据孤岛、**2** 开发低效 **3** 生态碎片化。通过一个统一的协议标准，MCP使得AI模型能够像"插USB"一样直接调用外部工具和数据源，大大简化了集成过程，提高了开发效率。

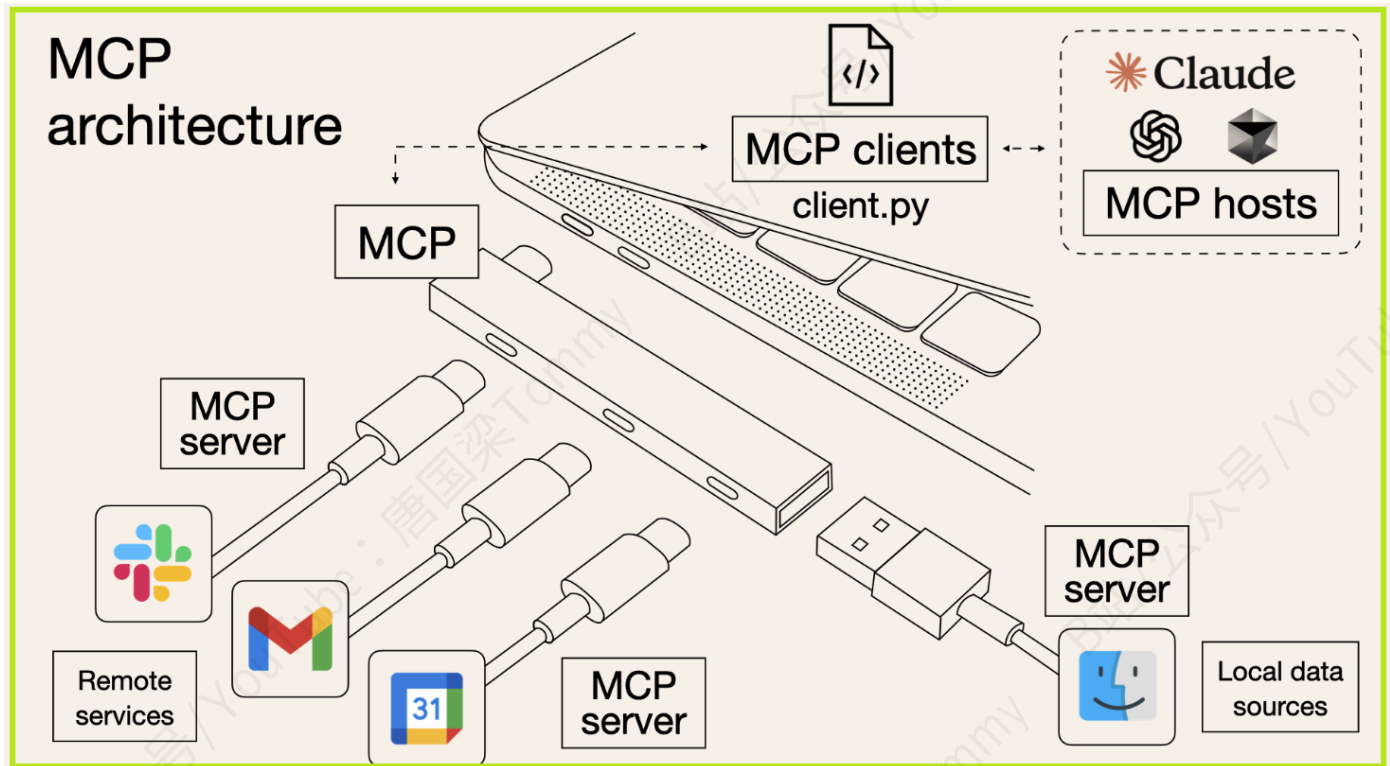
- **打破数据孤岛**: 传统大模型无法直接访问实时数据或本地资源, 而MCP让AI"连接万物", 例如查询天气时自动调用气象API, 分析企业数据时直接连接内部数据库。
- **降低开发成本**: 在MCP出现之前, 每个大模型需要为每个工具单独开发接口, 导致重复劳动。而通过MCP, 开发者只需写一次服务端, 所有兼容MCP的模型都能调用。
- **提升安全性与互操作性**: MCP内置权限控制和加密机制, 比直接开放数据库更安全; 同时, 类似USB接口的标准化让不同厂商的工具能"即插即用", 避免生态分裂。



3. MCP 技术架构与实现

✓ MCP 遵循客户端-服务器架构, 其架构主要由以下几个部分组成:

- **MCP Hosts (MCP 主机)**: 指希望通过 MCP 访问数据的 AI 工具, 例如聊天客户端、集成开发环境 (IDEs)、Claude Desktop、Cline等。
- **MCP Clients (MCP 客户端)**: 是协议客户端, 与 MCP 服务器保持一对一的连接。它们负责处理通信并将服务器的响应呈现给 AI 模型。
- **MCP Servers (MCP 服务器)**: 通过标准化的协议, 实现与MCP Clients的双向交互, 提供本地或远程资源的访问能力, 如数据库查询、API调用等。一个服务器可以提供多个功能, 类似于函数调用或工具使用, LLM 会自行理解并调用所需的能力。
- **本地数据源 (Local Data Sources)**: 计算机本地的文件、数据库等资源, MCP Servers可以安全地访问这些本地数据源。
- **远程服务 (Remote Services)**: 通过互联网 (如API) 访问的外部系统, MCP Servers可以与这些远程服务进行连接。



✅ MCP Servers可以提供三种类型的标准能力:

- **资源 (Resources):** 是可以被引用和检索的数据对象，包括文档、图像、数据库模式和其他结构化数据。客户端可以读取这些文件的数据。例如，一个 MCP 服务器可以暴露一个包含产品信息数据库模式作为资源。
- **提示词 (Prompts):** 提示词，为用户预先定义好的完成特定任务的模板。提示有助于确保常见任务的一致性、高质量 AI 输出。它们可以包含动态内容的占位符，并且可以链接在一起以创建复杂的工作流程。例如，一个 MCP 服务器可以提供用于生成产品描述的提示模板。
- **工具 (Tools):** 是可以被 LLM 调用的函数或第三方服务，例如查询数据库、调用 API 或处理数据。每个工具都有名称、描述和输入/输出模式定义。LLM 根据工具的描述决定应该使用哪个工具。例如，一个 MCP 服务器可以提供用于查询天气预报的工具。

✅ 通信协议采用 JSON-RPC 2.0，支持两种类型的通讯机制:

- **本地通讯 - Stdio (标准输入输出) :** 适用于本地进程间通信。
- **远程通讯 - 基于SSE的HTTP通讯 :** 支持流式传输与远程服务交互。

4. MCP 的工作流程

MCP (模型上下文协议) 的工作流程主要围绕着**客户端-服务器架构**展开，旨在标准化 AI 模型与外部工具和数据源之间的交互。以下是其主要的工作步骤: