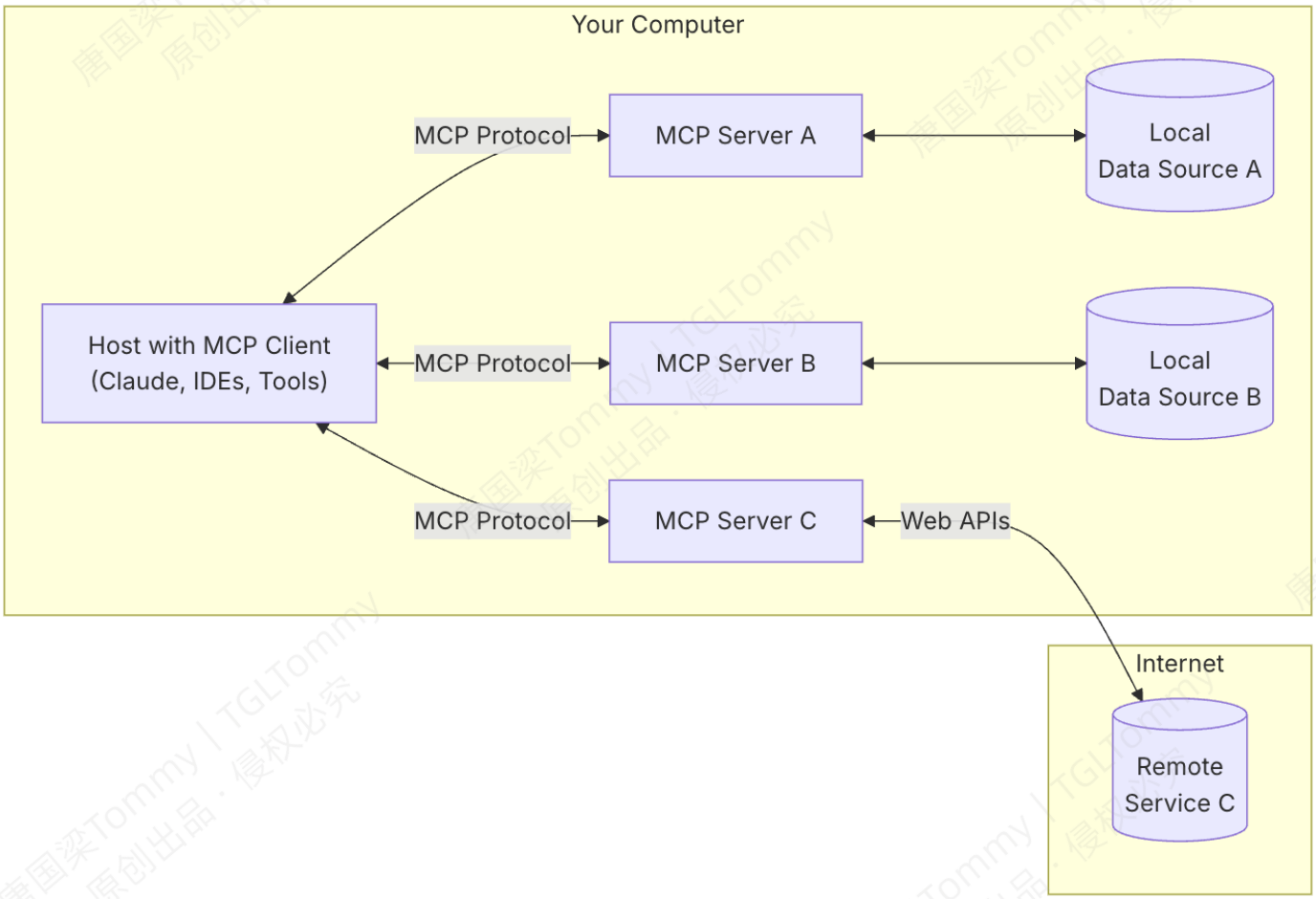
MCP 模型上下文协议

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。

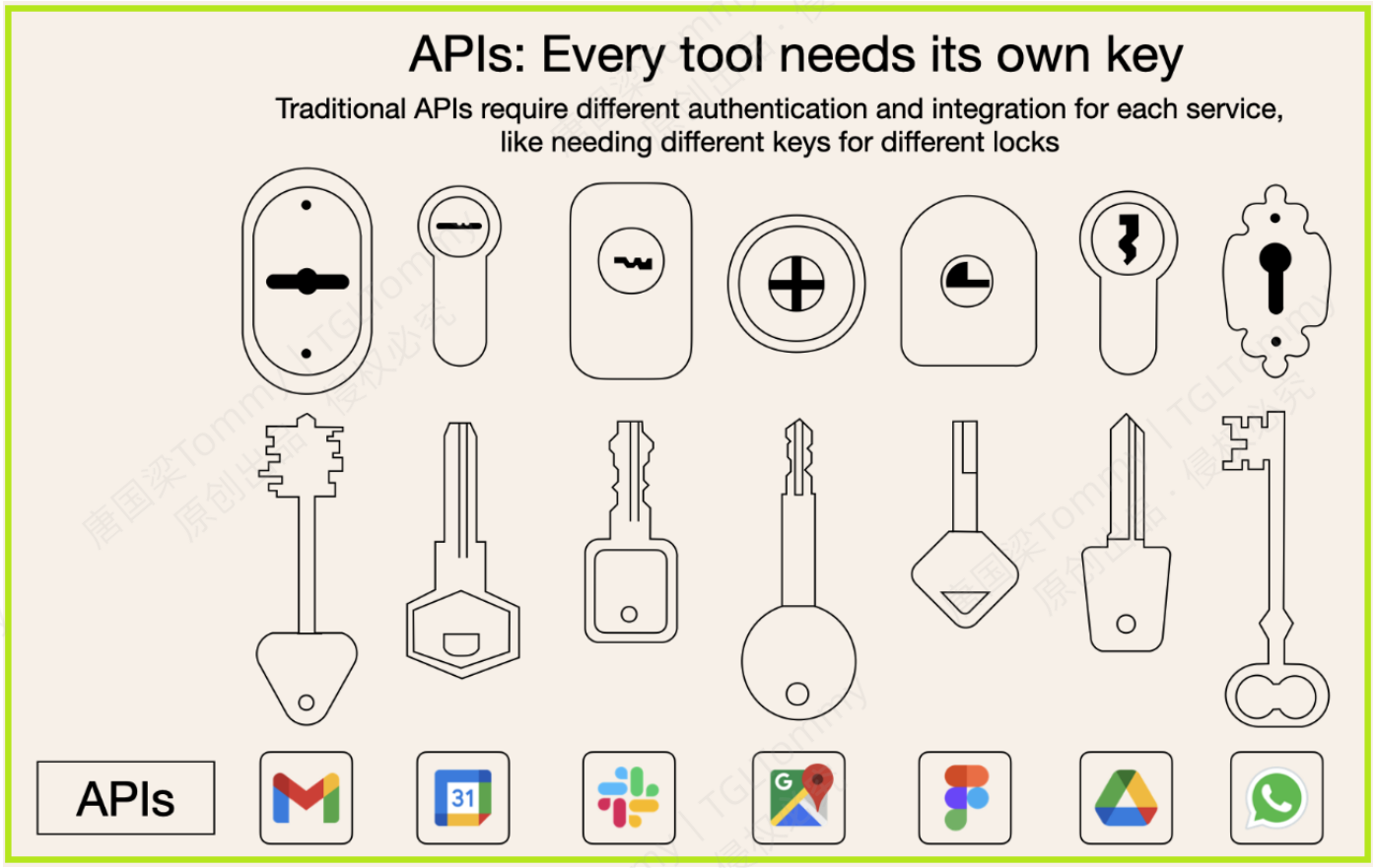


2. MCP解决的核心问题

MCP的出现解决了大模型时代三个关键问题：1️⃣ 数据孤岛、2️⃣ 开发低效 3️⃣ 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。

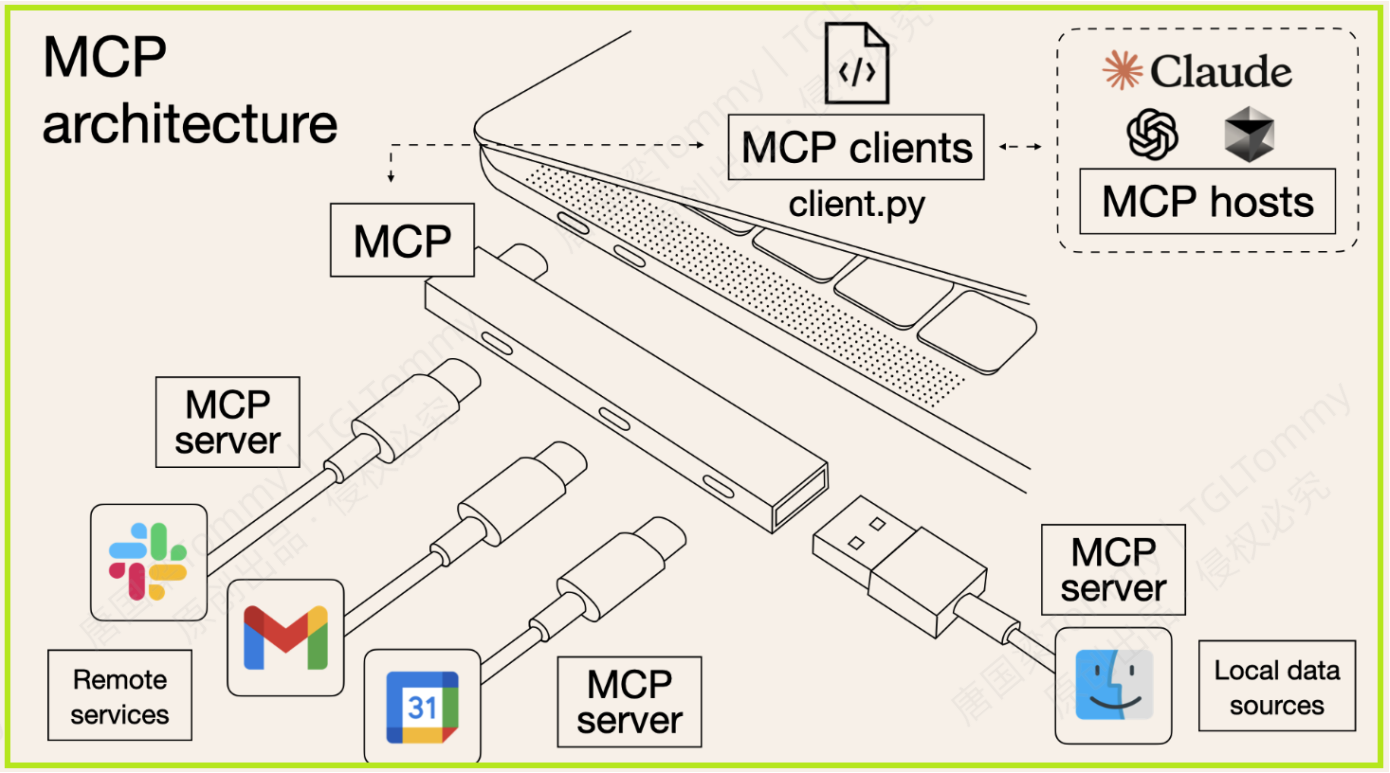
- **提升安全性与互操作性**：MCP内置权限控制和加密机制，比直接开放数据库更安全；同时，类似USB接口的标准化让不同厂商的工具能"即插即用"，避免生态分裂。



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的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 服务器可以提供一个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 服务器可以提供一个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：

第1步：能力发现：当一个 AI 应用（作为 MCP 客户端）需要扩展其能力时，它首先会连接到一个或多个 **MCP 服务器**。MCP 客户端会向 MCP 服务器询问其提供的功能，即**工具 (Tools)**、**资源 (Resources)** 和 **提示 (Prompts)** 的列表和描述。这就像客户端从服务器获取一个“菜单”，了解服务器可以做什么以及如何使用。

- **工具 (Tools)** 是 AI 模型可以执行的操作或函数，例如查询数据库、发送邮件或生成图像。

- **资源 (Resources)** 是 AI 模型可以访问和检索的数据对象，例如文档、API 响应或数据库模式。
- **提示 (Prompts)** 是预定义的模板，用于指导 AI 的交互，确保一致性和效率。

第2步：增强提示：当用户提出需要外部数据或操作的查询时，该查询以及服务器提供的工具/资源/提示的描述会被发送给 AI 模型（通过其宿主应用，即 MCP 主机）。模型现在"知道"它可以利用哪些服务器功能来完成任务。例如，如果用户询问"明天的天气怎么样？"，发送给模型的提示会包含一个描述"天气 API 工具"的信息，该工具由某个 MCP 服务器提供。

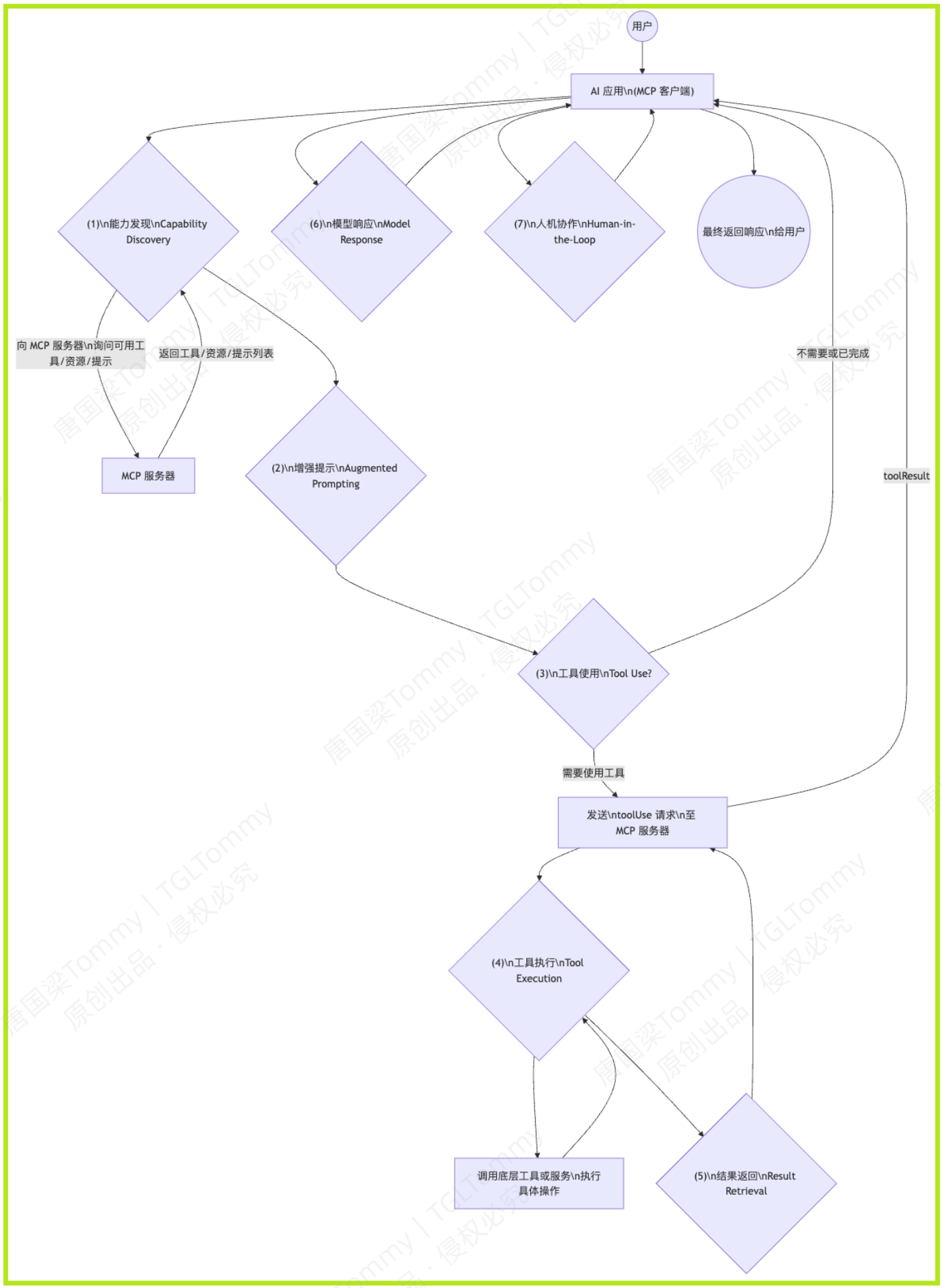
第3步：工具使用：基于用户的查询和可用的工具/资源/提示的描述，AI 模型（在 MCP 客户端中）会决定是否需要使用某个工具来完成任务。如果需要，客户端会向相应的 **MCP 服务器** 发送一个 `toolUse` 消息，指定要使用的工具及其所需的参数。

第4步：工具执行：**MCP 服务器** 接收到 `toolUse` 请求后，会将该请求分发给相应的底层工具或服务。例如，如果请求是使用"天气 API 工具"，服务器会调用实际的天气 API 并获取结果。

第5步：结果返回：**MCP 服务器** 在执行完工具后，会将结果以结构化的格式返回给 **MCP 客户端**，包含一个 `toolResult` 消息以及工具执行的输出。

第6步：模型响应：**MCP 客户端** 接收到 `toolResult` 后，会将工具执行的结果添加到对话历史中，并将其转发回 AI 模型（通常通过其宿主应用）。AI 模型现在可以利用这些信息来生成最终的响应给用户。这个过程可能涉及多次工具调用和数据检索，AI Agent 可以自主决定使用哪些工具、以什么顺序使用以及如何将它们串联起来完成复杂的任务。

第7步：人机协作：MCP 还引入了人机协作的功能，允许人类提供额外的数据并批准某些执行步骤。



总结来说，MCP 的工作流程可以概括为：**1️⃣ AI 客户端发现服务器能力 -> 2️⃣ AI 模型基于用户查询和可用能力决定使用哪些工具/资源/提示 -> 3️⃣ 客户端向服务器发送请求 -> 4️⃣ 服务器执行操作并返回结果 -> 5️⃣ 客户端将结果提供给 AI 模型以生成最终响应。**

整个过程通过标准化的协议进行通信，使得不同的 AI 模型和各种外部工具及数据源能够以一种通用且灵活的方式进行集成。这种标准化降低了集成复杂性，提高了互操作性，并为构建更强大的 AI 应用奠定了基础。

5. MCP 的关键优势

MCP 为 AI 系统开发带来了多项显著优势:

- **降低集成复杂性** : MCP 提供了一个统一的接口，消除了为每个 LLM 驱动的应用程序集成外部功能而实施其自身方法的必要性。通过标准化的协议，开发者可以更轻松地将 AI 模型连接到各种工具和数据源。它将一个 M×N 问题（M 个 AI 模型与 N 个工具/数据源的集成）转化为一个 M+N 的解决方案，大大减少了集成所需的自定义代码和适配器。
- **加快开发速度**: 开发者可以利用预构建的 MCP 客户端和服务器，从而节省开发自定义集成的时间和精力。工具和 API 开发者只需要构建一个 MCP 服务器，就可以被所有兼容的 MCP 客户端使用。
- **提高可扩展性和可靠性**: 通过标准化的架构，MCP 有助于构建更健壮和可扩展的 AI 系统。维护上下文跨工具也变得更加容易，因为交互共享一个通用的框架。
- **实现供应商无关的开发**: MCP 作为一个开放标准，不限制开发者使用特定的 AI 提供商的生态系统或工具链。任何支持 MCP 的 AI 客户端（例如 Claude 或开源 LLMs）都可以使用任何兼容的 MCP 服务器。
- **增强 AI Agent的上下文感知能力**: MCP 的出现能够帮助上下文层实现最好的效果。由于上下文层通常需要开发者自己进行定义，而 MCP 作为一个开源协议，能够最好地发挥社区的力量来加速这个过程。
- **实现“万能应用”的潜力**: 通过合适的 MCP 服务器，用户可以将每个 MCP 客户端变成一个“万能应用”。

6. MCP 与 API 对比

- **定义**
  - **MCP**：MCP 是一种标准化协议，旨在为大型语言模型（LLM）提供上下文信息，以便它们能够更有效地与外部工具和数据源进行交互。它强调数据的上下文环境和语义背景，使得模型能够理解和处理复杂的结构化数据。
  - **API**：API 是一组规则和工具，允许不同的软件应用程序之间进行通信。它定义了请求和响应的格式，使开发者能够集成不同服务的功能，而无需了解其内部实现细节。

- **主要区别**

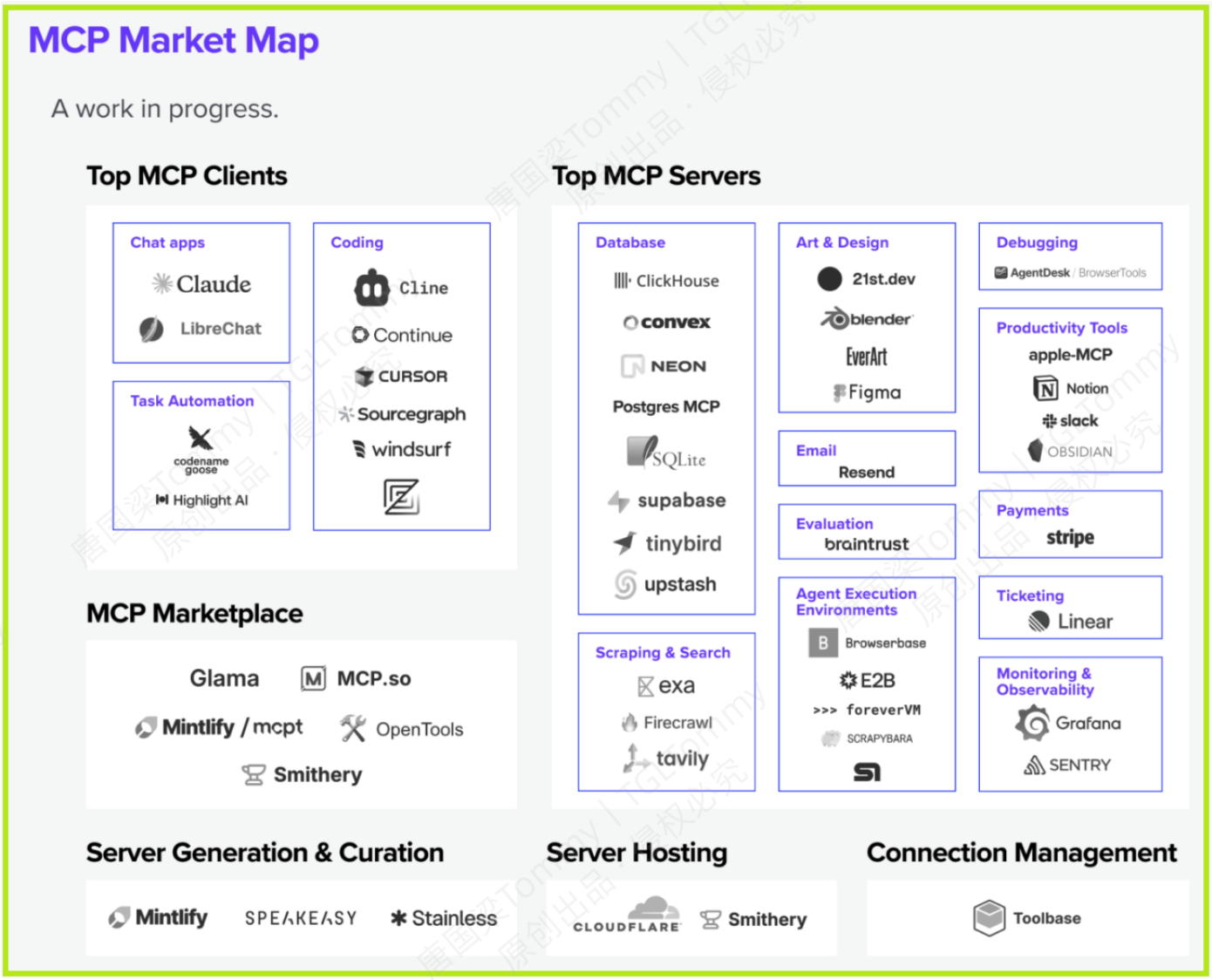
方面	Model Context Protocol (MCP)	API
核心目标	提供带有上下文的数据传输机制，确保接收方理解信息	提供接口以访问特定功能或数据集，简化应用间互动
通信方式	支持双向、实时、有状态的通信	通常为请求-响应模式，无状态通信
动态发现	支持动态工具发现和适应	通常需要预定义接口，缺乏动态发现能力
集成复杂度	单一标准化集成，简化多工具连接	每个API需单独集成，增加开发复杂度
上下文管理	增强的上下文感知和管理	有限的上下文管理能力
互操作性	模型无关，旨在成为通用标准	通常是供应商特定的

- **应用场景**
  - **MCP 的应用场景**：
    - 适用于需要多种工具和数据源协同工作的复杂 AI 应用，例如智能助手、数据分析平台等。
    - 支持实时数据交互和动态工具调用，使 AI 模型能够根据上下文变化灵活调整行为。
  - **API 的应用场景**：
    - 常用于简单的服务集成，如获取天气信息、支付处理等。
    - 在传统企业系统中广泛使用，用于实现各个模块之间的功能调用。

7. MCP 生态系统

围绕 MCP 正在迅速形成一个生态系统：

- **MCP Clients**: 目前高质量的 MCP 客户端大多以编程为中心。例如，Cline是一款开源的VSCode插件，它通过与MCP协议的结合，使得开发者能够轻松扩展AI的功能，甚至创建完全自定义的智能体。Claude Desktop本身也是一个 MCP 客户端，允许用户连接和使用各种 MCP 服务器。未来预计会出现更多面向业务的 MCP 客户端。
- **MCP Servers**: MCP 服务器的数量正在快速增长，涵盖了各种用例，例如访问本地文件系统、查询数据库、发送电子邮件、生成图像、与设计工具（如 Figma 和 Blender）交互、控制音乐软件（如 Ableton Live）以及进行网络搜索。许多头部数据库公司、编码公司和初创公司都在开发自己的服务器。GitHub 上已经有超过 2000 个 MCP 服务器，其中最常见的使用场景是搜索和数据检索。
- **MCP Marketplace**: 随着 MCP 服务器数量的增加，一个统一的 MCP 市场可能会出现，允许 AI 代理根据速度、成本和相关性等因素选择合适的服务器。
- **MCP Infra**: 围绕 MCP 的基础设施正在发展，包括服务器生成工具（如 Mintlify, Stainless, Speakeasy）以降低创建 MCP 兼容服务的摩擦，以及托管解决方案（如 Cloudflare, Smithery）以解决部署和扩展挑战。连接管理平台（如 Toolbase）也开始简化本地优先的 MCP 密钥管理和代理。目标是让 MCP 更可靠、更易于生产环境部署和更具可扩展性。



8. MCP 的挑战与限制

尽管 MCP 取得了显著的进展，但在构建和使用 MCP 时仍然存在一些未解决的问题:

- **缺乏内置的工作流概念**: 大多数 AI 工作流程需要按顺序进行多次工具调用，但 MCP 缺乏管理这些步骤的内置工作流概念。
- **标准化客户端体验**: 如何在构建 MCP 客户端时进行工具选择是一个常见问题。目前尚不清楚每个客户端是否需要实现自己的工具 RAG，或者是否存在一个等待标准化的层。
- **安全性和身份验证**: 标准化的安全机制是 MCP 规范中最迫切的需求之一。未来预计会定义一个身份验证层，例如 OAuth 类似的流程或 API 密钥标准，以便客户端可以安全地连接到远程服务器。权限模型也可能被引入。
- **远程 MCP 支持**: 目前大多数 MCP 服务器都是本地优先的，限制了其可扩展性。随着生态系统的发展，将远程 MCP 作为一等公民，并采用可流式的 HTTP 传输，预计将促进 MCP 服务器的更广泛采用。
- **服务器发现与管理**: 随着 MCP 服务器数量的增加，简化服务器的发现、共享和贡献变得越来越重要。类似应用商店的双边平台（MCP Marketplace）可能会出现。
- **多租户支持**: 未来可能会出现 MCP 网关或编排层，作为统一的端点聚合多个 MCP 服务，处理路由甚至关于使用哪个工具的高级决策，并管理多租户和实施策略。
- **优化的 AI 代理**: 未来可能会出现专门为工具使用和 MCP 进行微调的 AI 模型。这些模型将更深入地理解协议，知道如何准确格式化请求，并可能在成功的基于 MCP 的操作日志上进行训练，从而提高效率和可靠性。

9. MCP 与现有解决方案的比较

- **与 API 的比较**: 相对于通常提供细粒度控制和高度特定功能的传统 API，MCP 提供了更广泛、更动态的能力，更适合需要灵活性和上下文感知能力的场景。对于需要精细控制、性能优化和最大可预测性的应用程序，仍然可能更倾向于使用细粒度的 API。
- **与 RAG 和 Agents 的关系** : AI 模型受益于RAG和Agents。MCP 旨在标准化如何将这些信息暴露给语言模型。Agents 在某种程度上可以看作是 RAG 的扩展。MCP 的出现使上下文的价值最大化。
- **与 OpenAI Function Call、GPTs 和 Agent SDK 的比较** : MCP 被认为是现有中间层的集大成者。它借鉴了 OpenAI Function Call 和 GPTs 的思路，但做得更轻量级和开放。相对于仍在发展中的 OpenAI Agent SDK，MCP 更加开源和灵活，但可能在精细度和效率上不如 Agent SDK。
- **与 LangChain 和 LlamaIndex 的比较**: LangChain 和 LlamaIndex 试图构建 Agent 框架，但由于其较高的抽象程度和复杂的框架，许多 LLM 开发者在初步使用后会转向自己开发。MCP 的出现对它们产生了较大的冲击。

10. 结论

模型上下文协议（MCP）代表着 AI 工具集成领域的一个重大进步。它通过提供一个标准化的开放协议，**极大地简化了 LLM 应用与外部世界的连接**，从而加速了 AI 应用的开发，提高了其可扩展性和可靠性。MCP 的成功不仅在于其技术设计，更在于其强大的支持者、借鉴成功的经验以及积极发展的生态系统。尽管仍然面临一些挑战，但 MCP 有望成为 **Agentic AI 的关键中间层**，并为开发者和创业公司带来新的机遇。正如 USB-C 统一了电子设备的连接一样，**MCP 有潜力成为 AI 原生时代的通用语言**，推动人与 AI 通过软件进行更流畅和安全的协作。

💡 参考资料

```
# 1. MCP官方文档
https://modelcontextprotocol.io/introduction
# 2. MCP Servers
https://github.com/modelcontextprotocol/servers.git
# 3. MCP python-sdk
https://github.com/modelcontextprotocol/python-sdk.git
# 4. awesome-mcp-servers
https://github.com/punkpeye/awesome-mcp-servers.git
# 5. awesome-mcp-clients
https://github.com/punkpeye/awesome-mcp-clients.git
# 6. Smithery 服务托管平台
https://smithery.ai/
#7. claude desktop 下载
https://claude.ai/download
# 8. cherry studio 下载
https://cherry-ai.com/
# 9. cline 插件下载
https://github.com/cline/cline
# 10. cloudflare
https://www.cloudflare.com/products/registrar/
# 11. OpenRoute
https://www.cloudflare.com/products/registrar/
```

AI进阶学习

📺 **B站课堂**：https://space.bilibili.com/3546748815411393/pugv

🌐 **官方网站**：https://www.tgltommy.com/（国内访问需科学上网）

如果你正在关注大模型 Agent、强化学习后训练、RLHF、DPO、GRPO、RLVR 等前沿方向，欢迎关注我最新上线的精品课程：
《大模型 Agent 强化学习实战：从后训练、奖励设计到工业级对齐系统》

这门课程围绕当前大模型 Agent 强化学习的核心技术路线展开，不只是介绍零散算法和论文概念，而是希望帮助学习者建立一套完整的技术地图：从基础原理、论文脉络、关键算法，到开源项目源码解析与工业级系统实践，逐步理解 Agent RL 为什么重要、如何演进，以及在真实应用中如何落地。

课程配套专属飞书知识库《Agent RL 学习宝典》，包含学习路线、论文资料、项目说明、源码阅读辅助资料和后续持续更新内容，方便长期查阅和跟进前沿技术。

Agent RL 变化很快，但越是变化快，越需要一套系统的学习框架。如果你希望系统进入 Agent RL 方向，而不是停留在碎片化了解阶段，这门课会是一个很好的起点。

课程官网：https://www.tgltommy.com/p/agent-rl（国内访问需科学上网）
B站课堂：https://www.bilibili.com/cheese/play/ss842375604

更多课程详情，扫描以下二维码咨询课程助理：