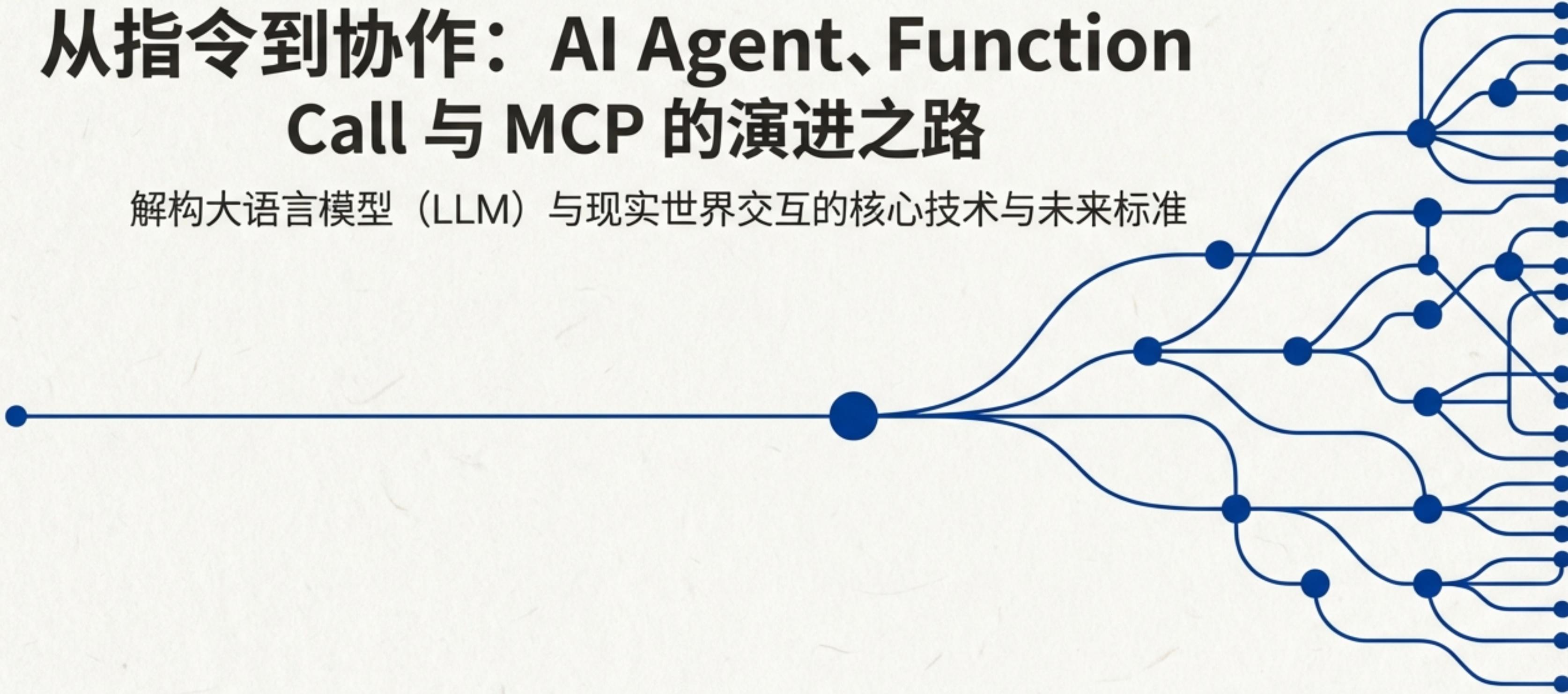


# 从指令到协作：AI Agent、Function Call 与 MCP 的演进之路

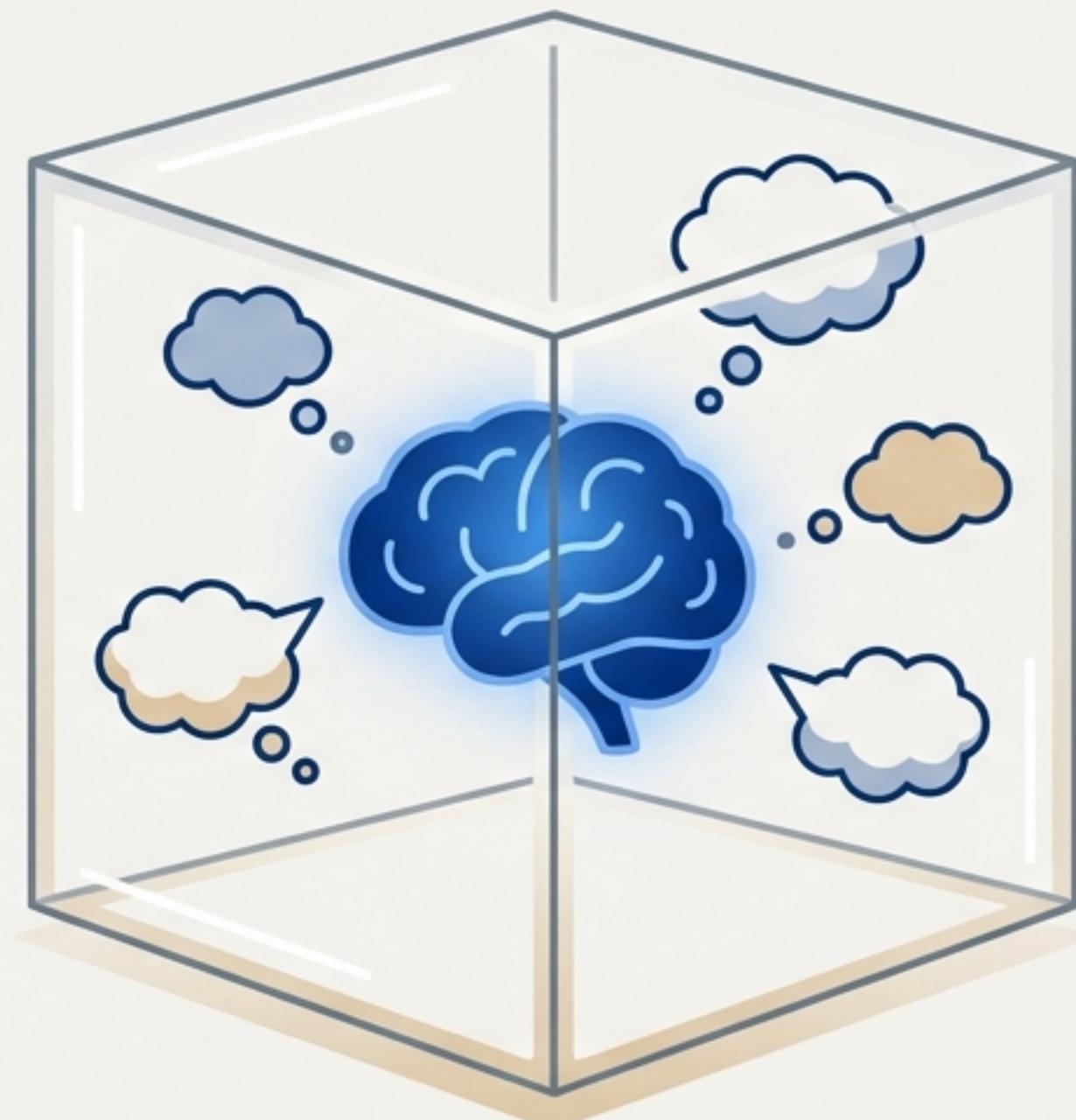
解构大语言模型（LLM）与现实世界交互的核心技术与未来标准



# 核心困境：被“囚禁”的卓越思维

## 核心论点：

- 大语言模型（LLM）拥有海量的知识和强大的推理能力，是卓越的“思维者”。
- 然而，它们与外部世界是隔离的，如同一个“缸中之脑”（brain in a vat）。
- **关键局限**（源于 "Function Calling in Large Language Models" 论文）：
  - 无法获取实时信息（例如：最新的股票市场价格）。
  - 无法与用户的私有数据交互（例如：你的个人日历）。
  - 无法执行具体动作（例如：预订一张机票）。



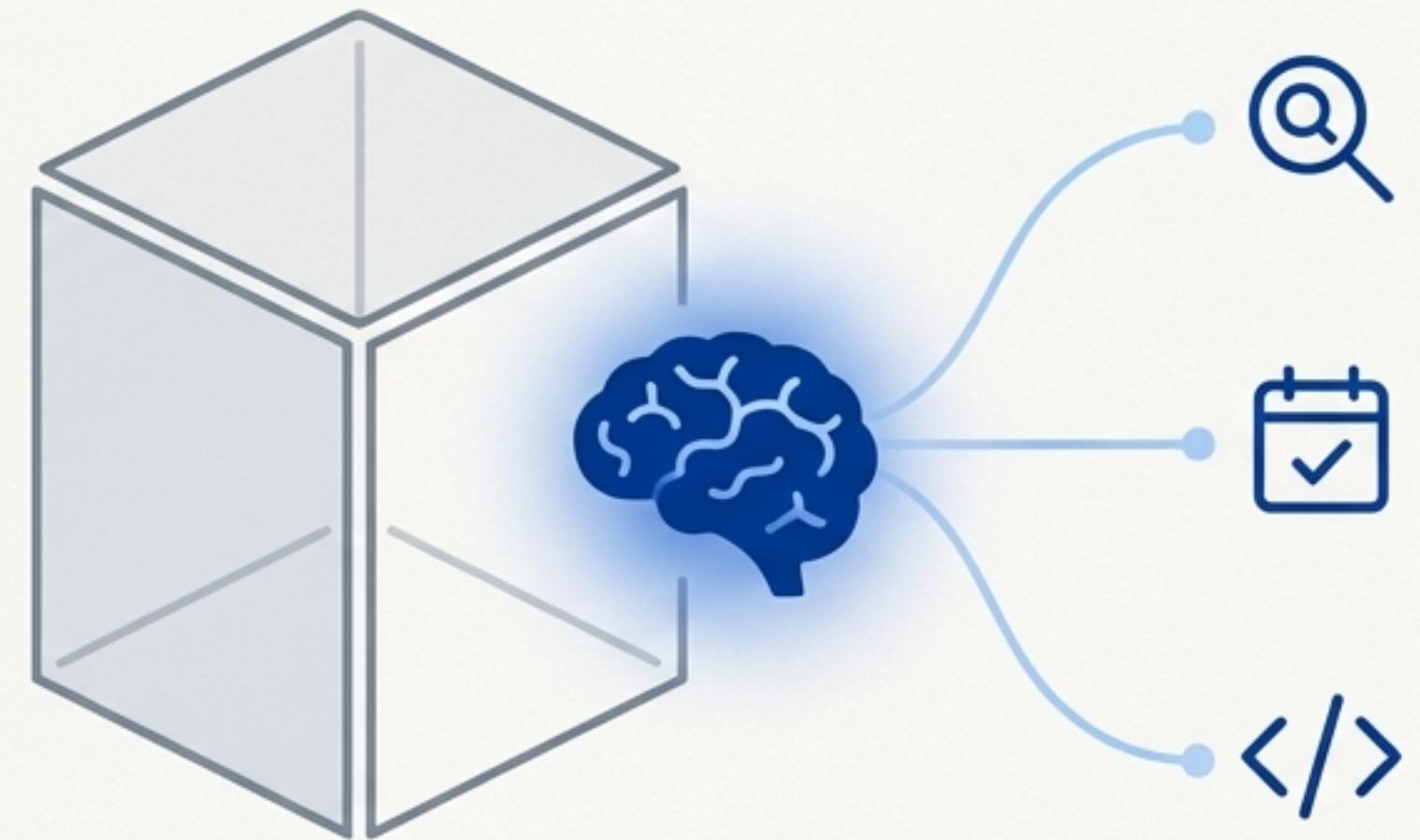
# 第一次飞跃：AI Agent —— 为大脑接上“手”和“眼”

## 什么是 AI Agent？

以LLM为核心，能够感知环境、制定决策并采取行动以达成目标的自主系统。

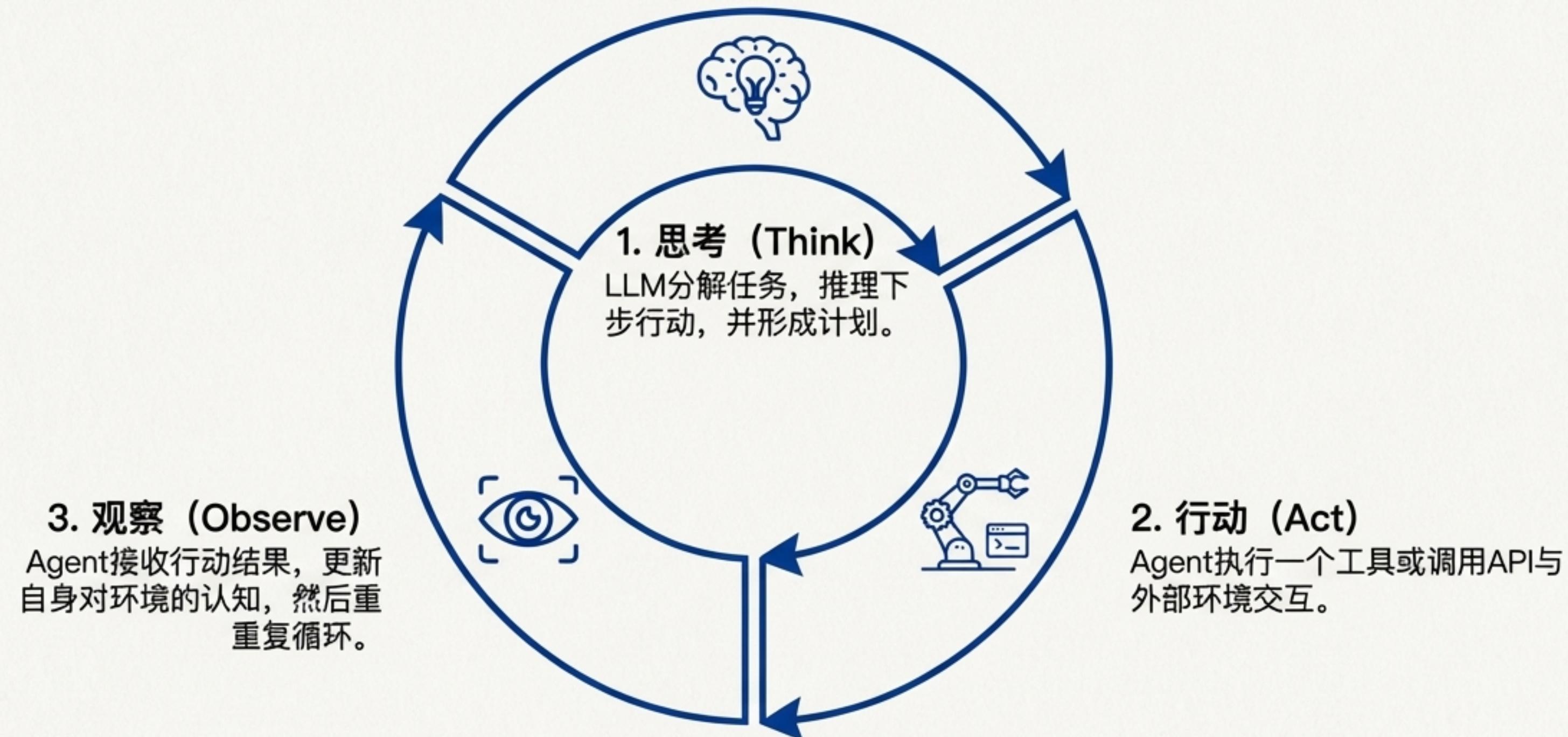
## 核心组件：

- 核心引擎（大脑）：用于推理和规划的LLM。
- 感知能力：接入外部信息源。
- 行动能力：使用工具和API执行任务。



# 核心流程：Agentic Workflow —— 智能体如何“思考”与“行动”

Agent通过一个称为“Agentic Workflow”的循环流程来解决复杂任务。这个迭代周期使其能够进行规划、行动、观察和调整。



# 连接现实的桥梁：Function Calling

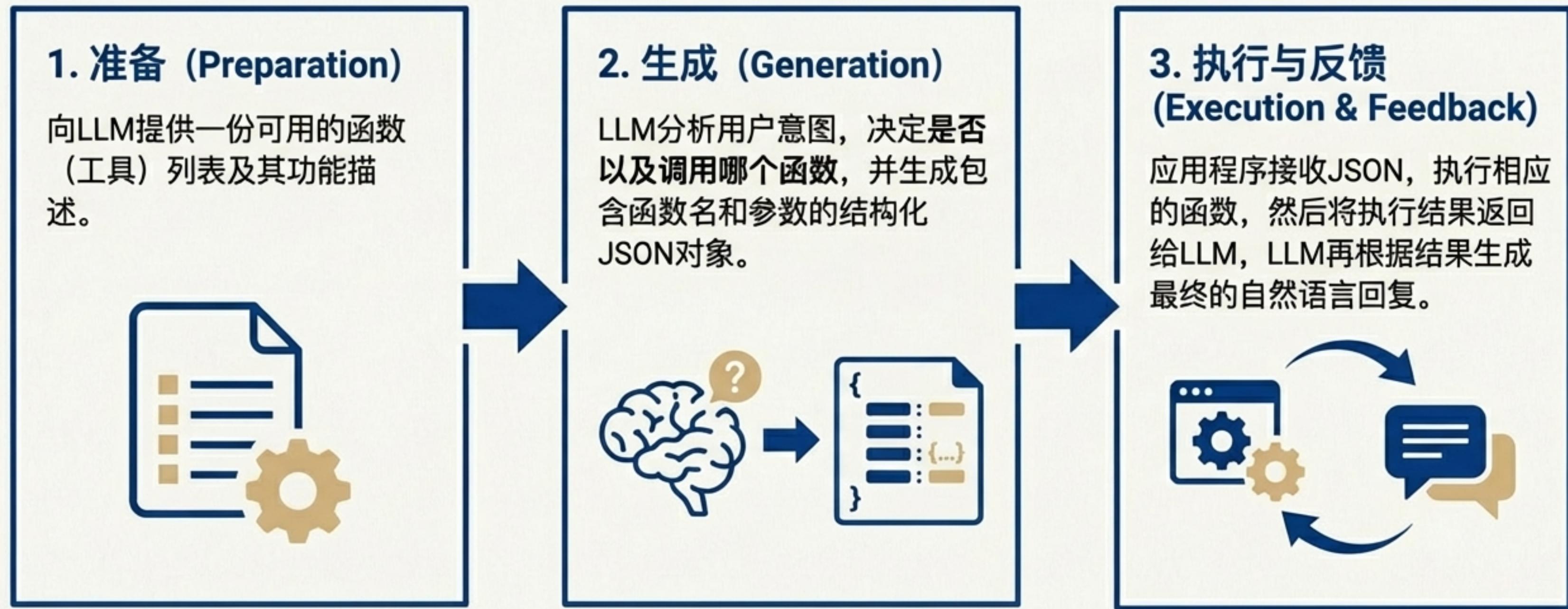
一个关键问题：一个只能理解和生成文本的语言模型，究竟如何“行动”？

答案是一种名为 **Function Calling** 的技术范式。

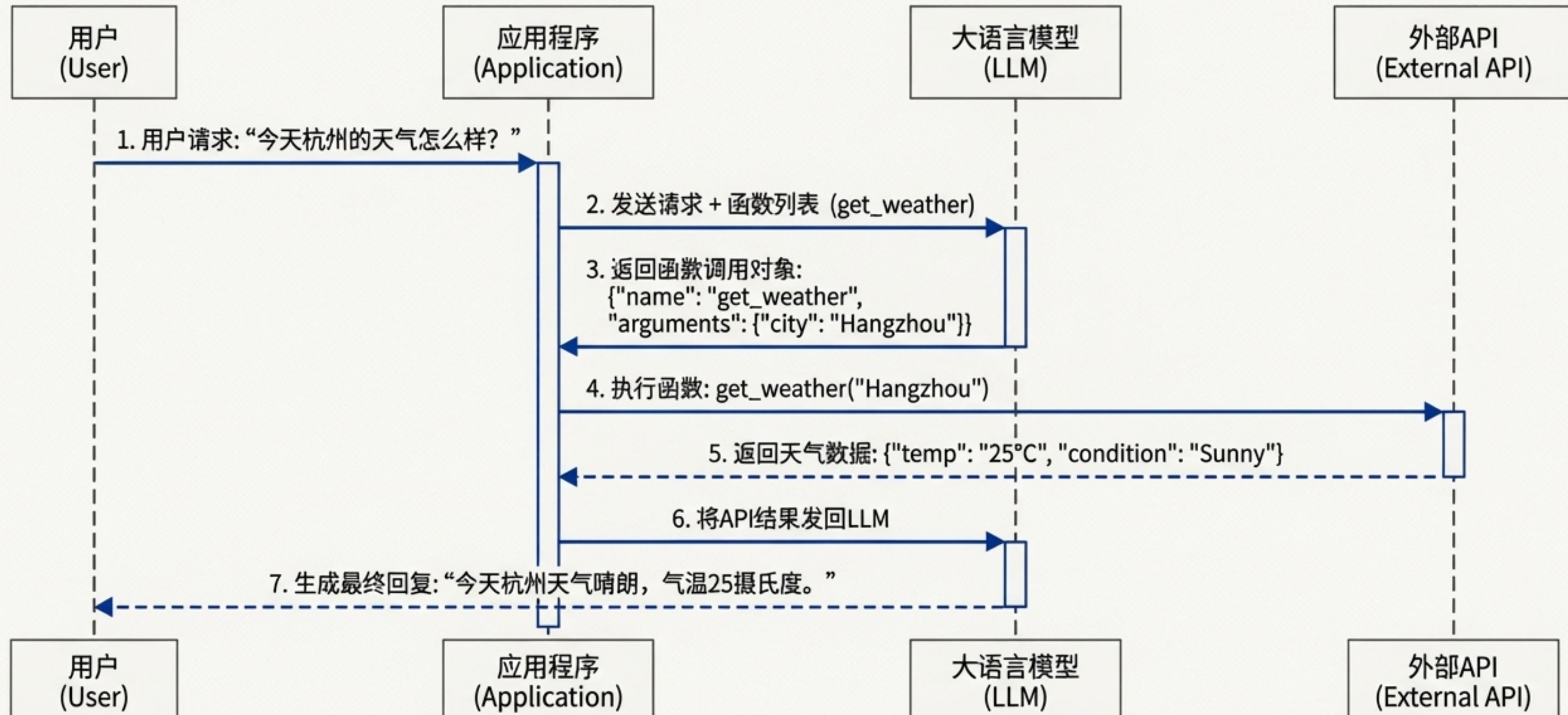
**定义**（源于 "Function Calling in Large Language Models" 论文）：它使语言模型能够解读用户请求，并调用预先定义的计算程序（函数），通过连接外部数据源（如数据库、API、实时数据流）来提供精确、上下文相关的答案。



# 技术机理：LLM Function Calling 的工作流程

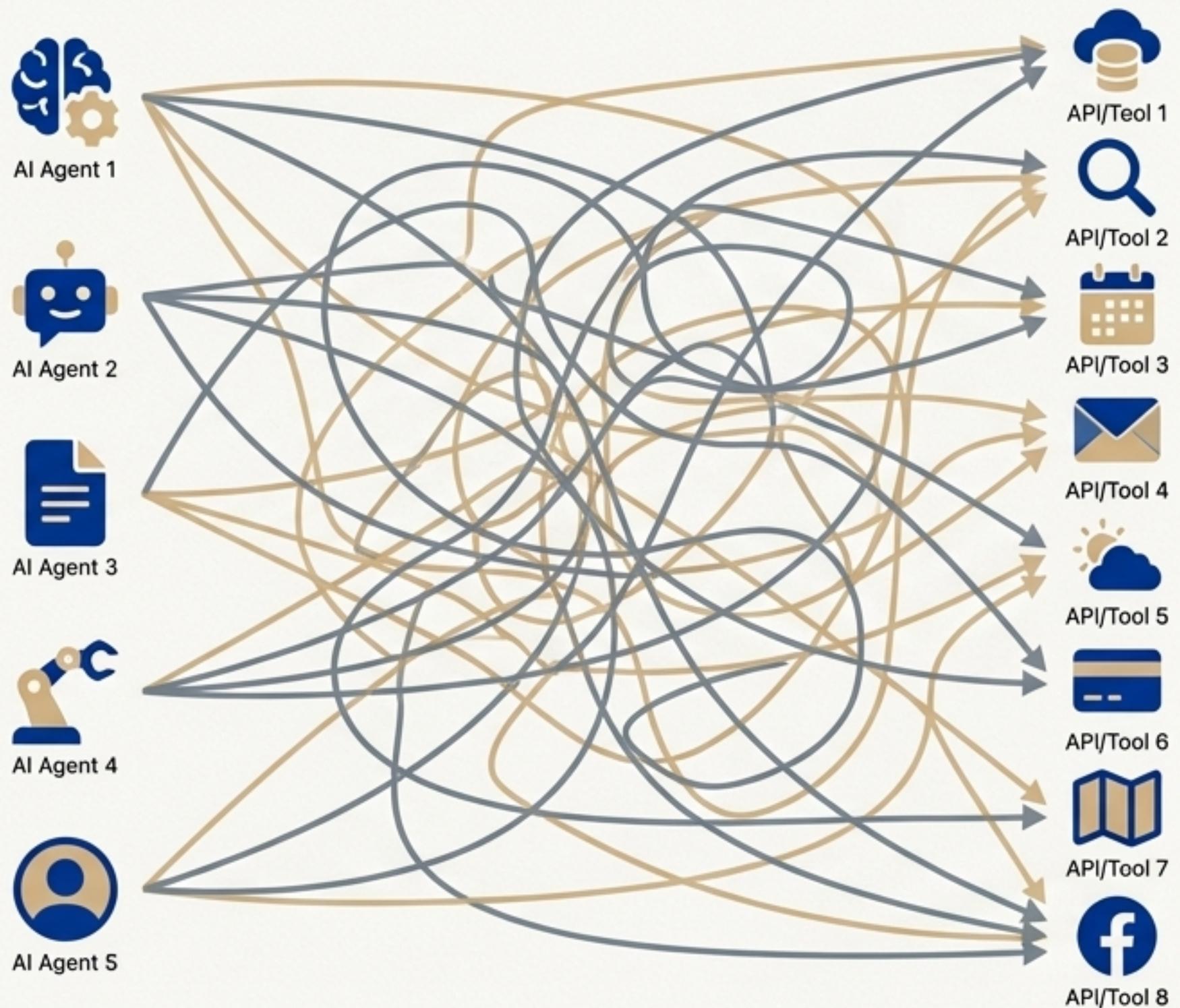


# 调用过程解析：Function Calling 时序图



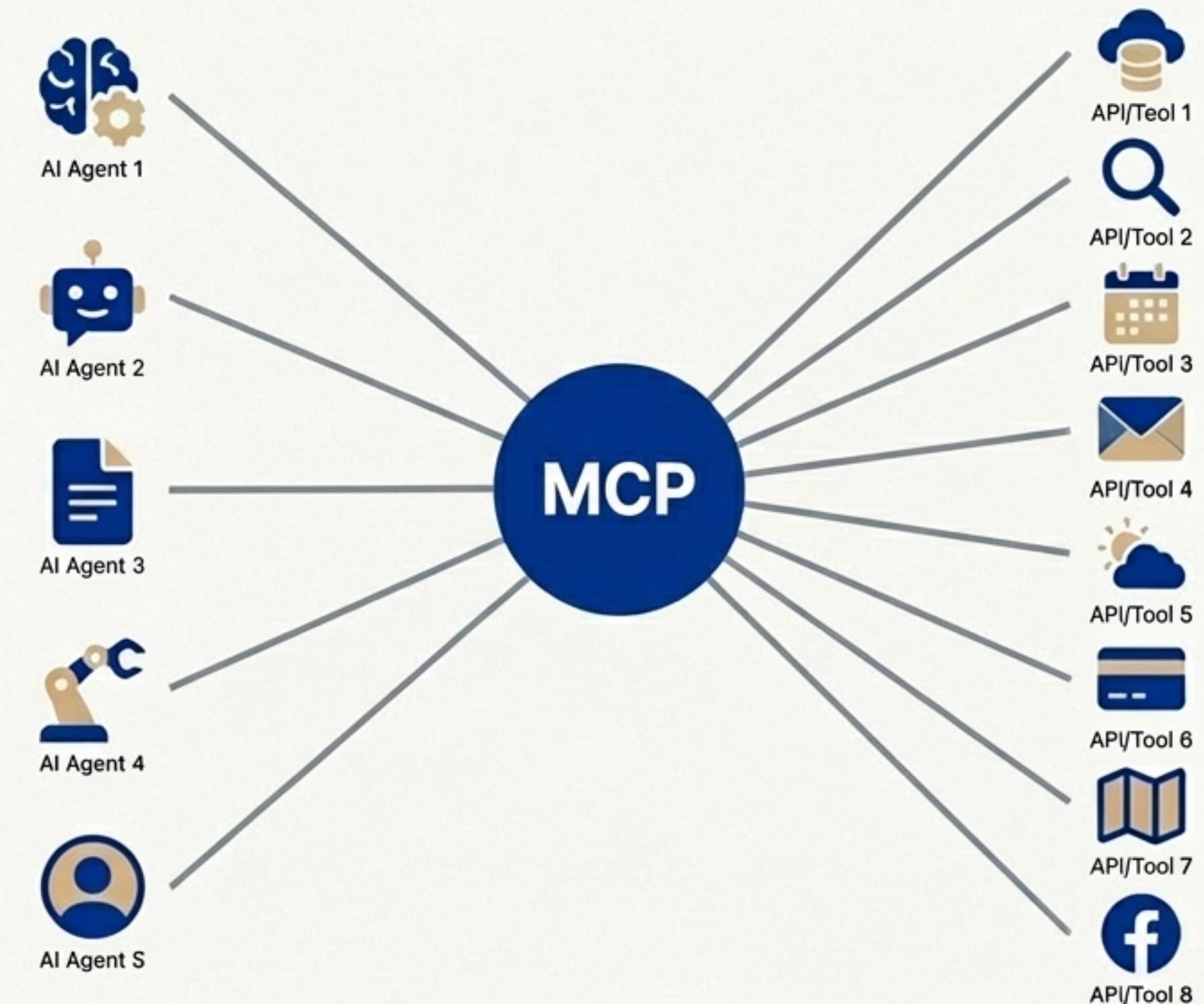
# 规模化的挑战：“M×N”集成难题

- Function Calling对于单个应用非常强大，但在规模化时会创建一个混乱、碎片化的生态系统。
- 每个AI应用（M）都需要为每个工具和服务（N）构建定制化、紧耦合的集成。
- 这种方式是低效、脆弱且不可互操作的，Anthropic称之为“M×N问题”。

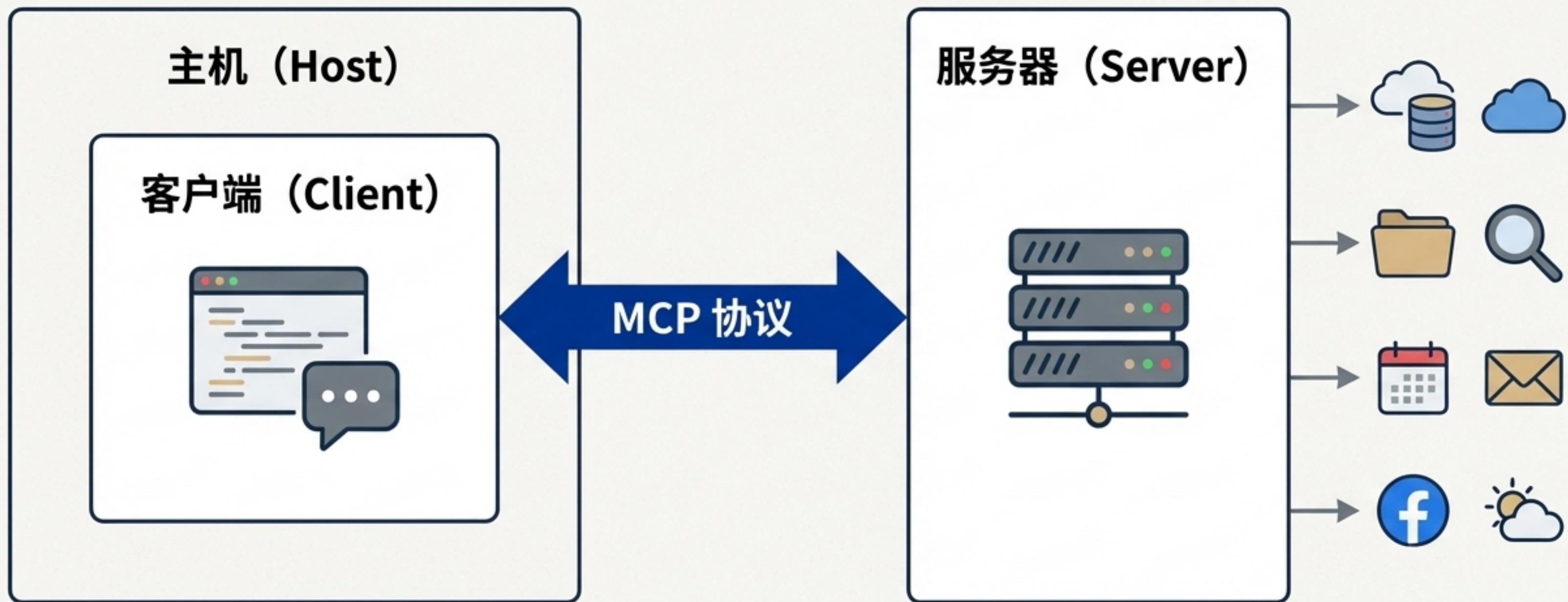


# 下一个前沿：模型上下文协议 (MCP)

- **解决方案：**MCP是一种开放协议，旨在成为AI应用发现和调用外部工具与服务的通用标准，类似于任何网络中的HTTP协议。
- **核心比喻：**如果说API是独立的乐高积木，MCP就是那本通用的说明书和标准化的连接件系统，它让任何AI Agent都知道如何使用任何积木。它是“**AI领域的USB-C**”。



# MCP 工作原理：标准化的解耦架构

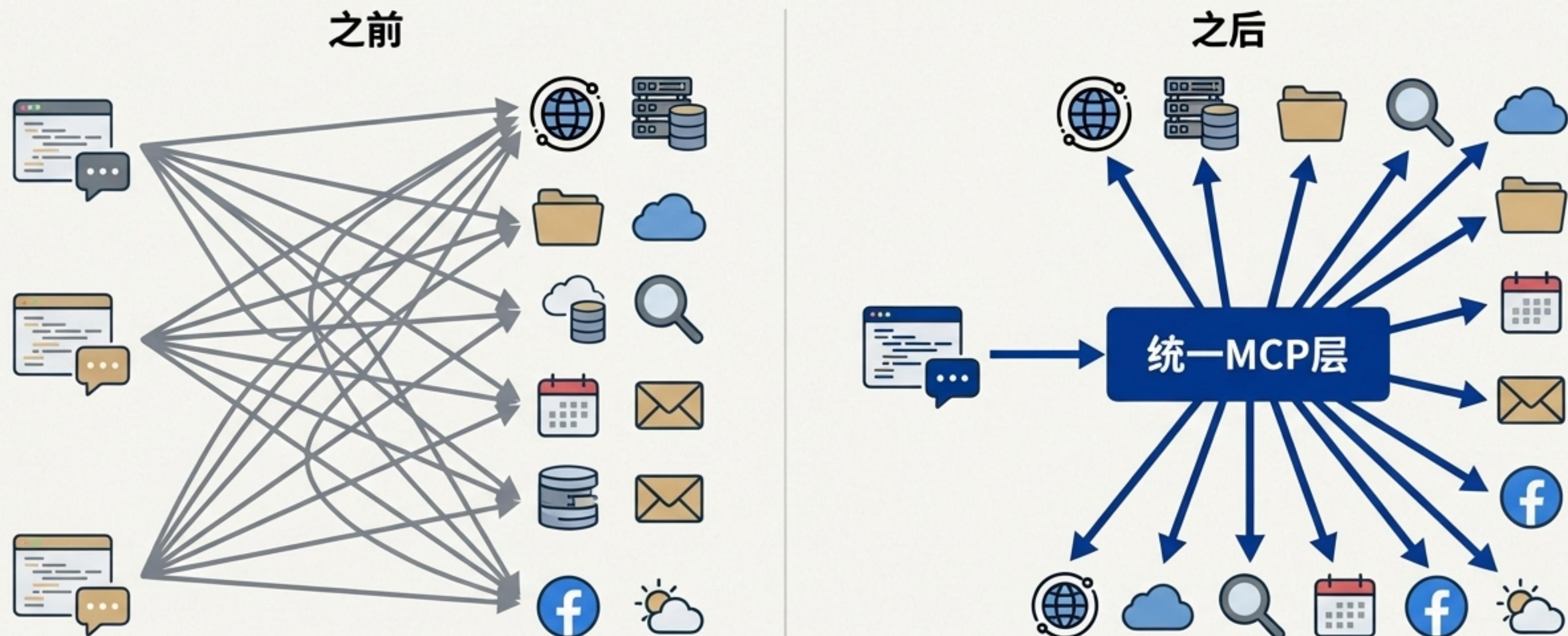


# 核心差异：从API调用到标准协议的飞跃

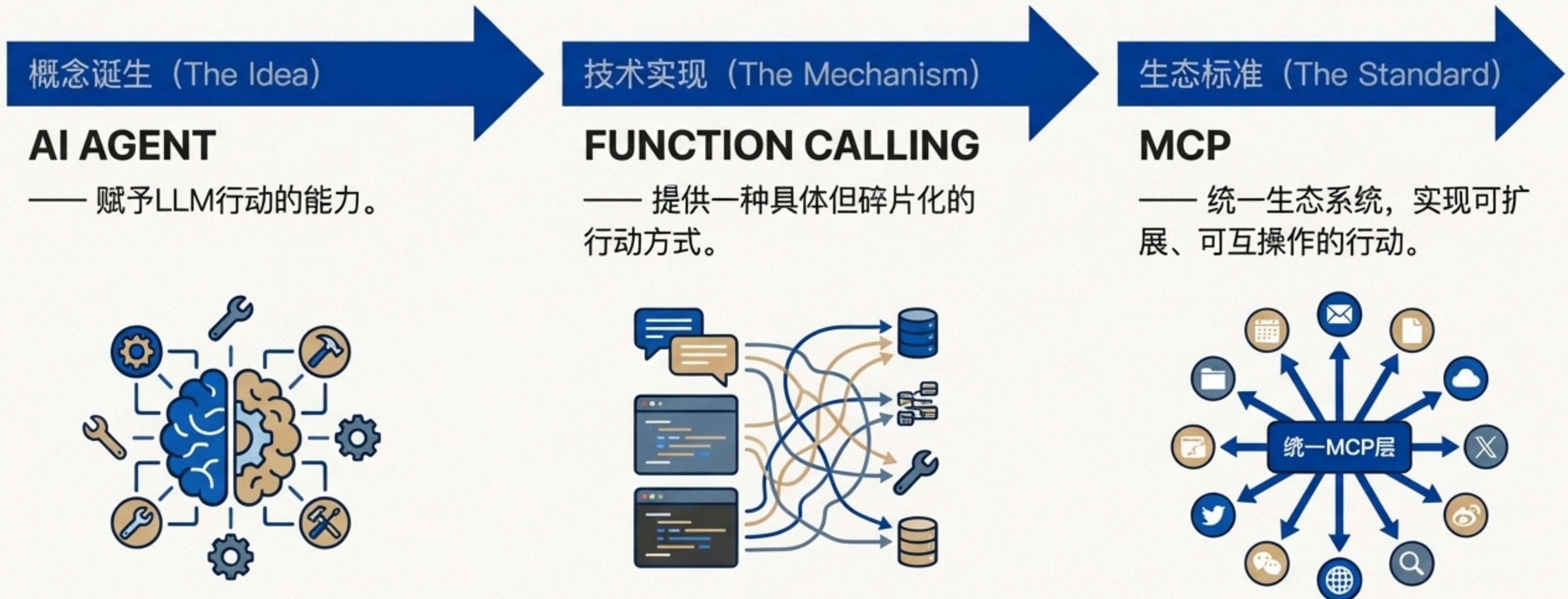
维度 (Aspect)	传统 Function Calling (以API为中心)	模型上下文协议 (MCP)
主要用户	编写代码的开发者	AI智能体和LLM
能力发现	手动阅读API文档	自动化、动态的能力发现
状态管理	无状态 (每次请求独立，由开发者管理上下文)	有状态会话 (内置上下文记忆)
协议	REST, GraphQL (应用层实现)	JSON-RPC 2.0 (标准化的通信协议)
集成方式	显式的端点硬编码映射	动态的工具发现与调用
成熟度	经过长期考验，但生态碎片化	新兴标准 (2024年11月发布)
最适用场景	直接的系统集成、移动/Web应用	AI流程编排、多步自动化任务

# MCP 的愿景：从混乱到秩序的生态重塑

MCP通过提供一个标准协议，将AI应用与工具的交互从点对点的复杂集成，转变为通过统一接口的灵活调用，从而消除了重复的维护工作，并促进了一个共享的、可互操作的生态系统。



# 宏观视角：AI 交互的三阶段演进



# 核心要点与未来展望

- 1 AI Agent 将 LLM 从“思考者”转变为“行动者”，是释放其与现实世界交互潜力的关键。
- 2 Function Calling 是当前实现 Agent“行动”的主流技术，但其固有的点对点集成模式导致了生态系统的碎片化和扩展瓶颈。
- 3 MCP 作为开放的通信标准，旨在解决碎片化问题，它不是要取代API，而是为AI提供了一个更智能地使用API的协调层，是构建未来可扩展AI生态的基石。

# 谢谢 & 交流

