2025，AI Agents技术栈解读出炉！

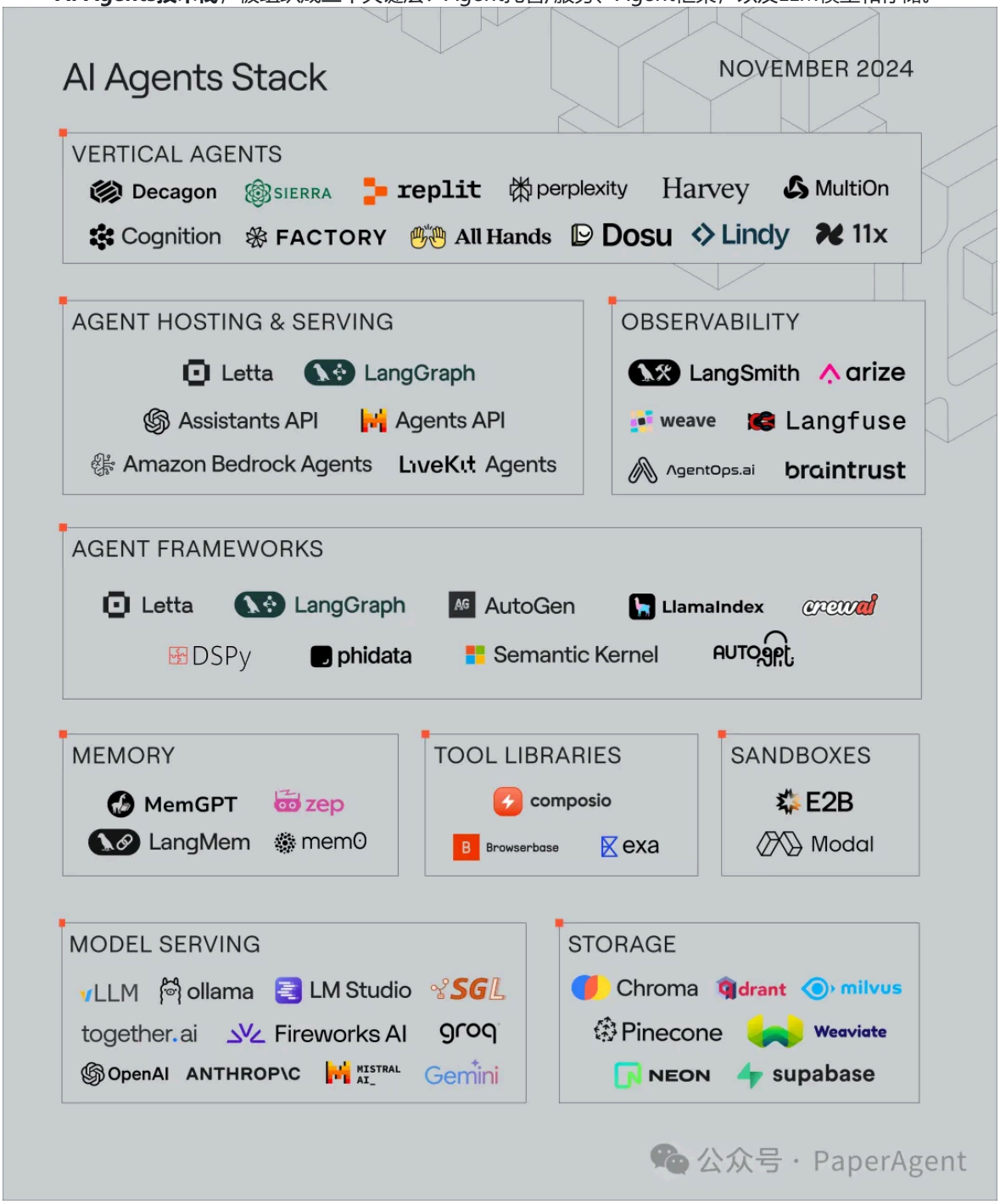
PaperAgent 2025年01月01日 00:03 湖北

2025新年伊始，今年将是“Agentic系统之年”，“2025将会出现真正Agent”，时不我待，请签收属于你的AI Agents技术栈综述。

理解AI Agents生态

尽管我们看到了大量关于Agent栈和市场的分类图，但我们往往不同意它们的分类方式，发现它们很少反映开发者实际使用的。在过去几个月中，随着在内存、工具使用、安全执行和部署方面的进步，Agent软件生态系统有了显著的发展，因此，能够真正落地的“**Agent技术栈（agent stack）**”应该是怎样尼?

AI Agents技术栈，被组织成三个关键层：Agent托管/服务、Agent框架，以及LLM模型和存储。



从LLM到LLM Agent

在2022年和2023年，我们看到了LLM框架和SDK的兴起，比如LangChain（2022年10月发布）和LlamaIndex（2022年11月发布）。同时，我们也看到了几个“标准”平台的建立，这些平台通过API消费LLMs以及自部署LLM推理（vLLM和Ollama）。

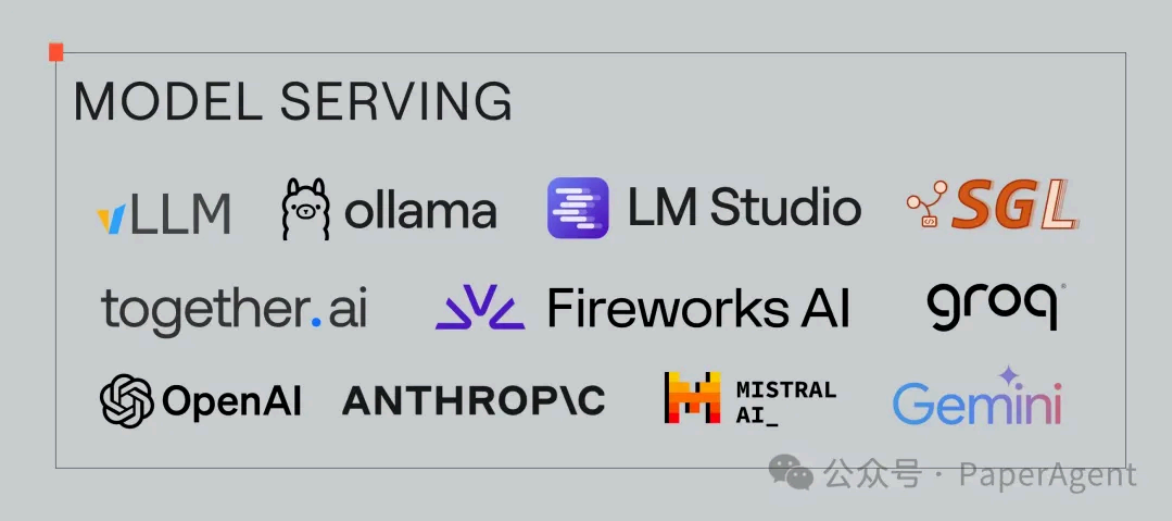
在2024年，我们看到了对AI“Agent”的兴趣急剧转变，更广泛地说，是对复合系统的兴趣。尽管“Agent”这个术语在AI中已经存在了几十年（特别是在强化学习中），但在后ChatGPT时代，“Agent”已经成为一个松散定义的术语，通常指的是被赋予输出动作（工具调用）并在自治设置中运行的LLM。从LLM到Agent所需的工具使用、自治执行和内存的结合，促使一个新的Agent栈发展。

Agent技术栈的独特之处

与基本的LLM聊天机器人相比，Agent是一个更复杂的工程挑战，因为它们需要状态管理（保留消息/事件历史记录，存储长期记忆，执行多个LLM调用在一个Agent循环中）和工具执行（安全执行LLM输出的动作并返回结果）。

因此，AI Agent栈与标准的LLM栈看起来非常不同。让我们从模型服务层开始，分解今天的AI Agent栈：

模型服务



AI Agent的核心是LLM。要使用LLM，模型需要通过推理引擎提供服务，通常运行在付费API服务后面。

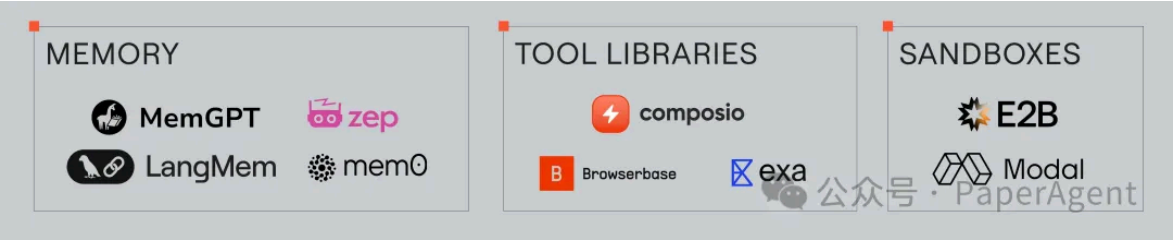
OpenAI和Anthropic在基于封闭API的模型推理提供商中领先，拥有私有前沿模型。

Together.AI、Fireworks和Groq是提供开放权重模型（例如Llama 3）背后的付费API的流行选项。在本地模型推理提供商中，我们最常见到vLLM领先于生产级GPU基础服务负载。SGLang是一个新兴项目，拥有类似的开发者受众。在业余爱好者（“AI爱好者”）中，Ollama和LM Studio是两个流行的选项，用于在您自己的计算机上运行模型（例如M系列Apple Macbooks）。

存储

存储是定义为有状态的Agent的基本构建块——Agent由持久状态定义，如他们的对话历史记录、记忆和外部数据源，他们用于RAG。像Chroma、Weaviate、Pinecone、Qdrant和Milvus这样的向量数据库很受欢迎，用于存储Agent的"外部记忆"，允许Agent利用数据源和对话历史记录，这些数据太大，无法放入上下文窗口。Postgres是一个自80年代以来就存在的传统数据库，现在也通过pgvector扩展支持向量搜索。基于Postgres的公司如Neon（无服务器Postgres）和Supabase也为Agent提供基于嵌入的搜索和存储。

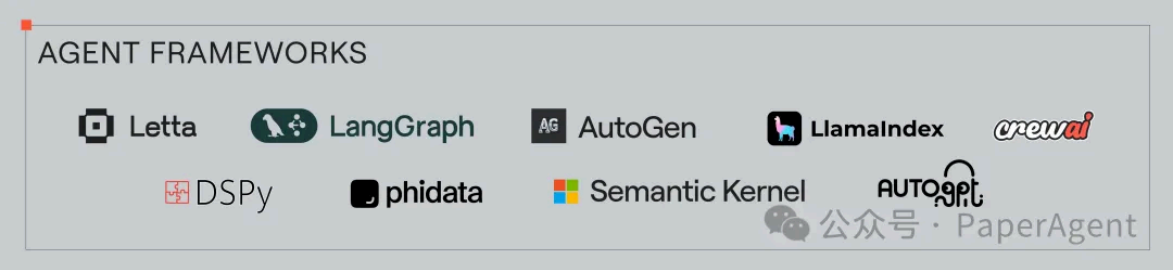
工具和库



标准AI聊天机器人和AI Agent之间的一个主要区别是Agent调用"工具"（或"功能"）的能力。在大多数情况下，这种动作的机制是LLM生成结构化输出（例如JSON对象），指定要调用的函数和提供的参数。Agent工具执行的一个常见混淆点是，工具执行_不是_由LLM提供商本身完成的——LLM只选择要调用的工具和提供的参数。支持任意工具或任意参数输入工具的Agent服务必须使用沙箱（例如Modal、E2B）以确保安全执行。

Agent通过OpenAI定义的JSON模式调用工具——这意味着Agent和工具实际上可以跨不同框架兼容。Letta Agent可以调用LangChain、CrewAI和Composio工具，因为它们都是由相同的模式定义的。因此，对于常见工具，有一个不断增长的工具提供商生态系统。Composio是一个流行的通用工具库，还管理授权。Browserbase是一个专门用于网页浏览的专用工具的例子，Exa提供了一个专门用于搜索网络的专用工具。随着越来越多的Agent被构建，我们预计工具生态系统将增长，并提供现有新功能，如Agent的身份验证和访问控制。

Agent框架

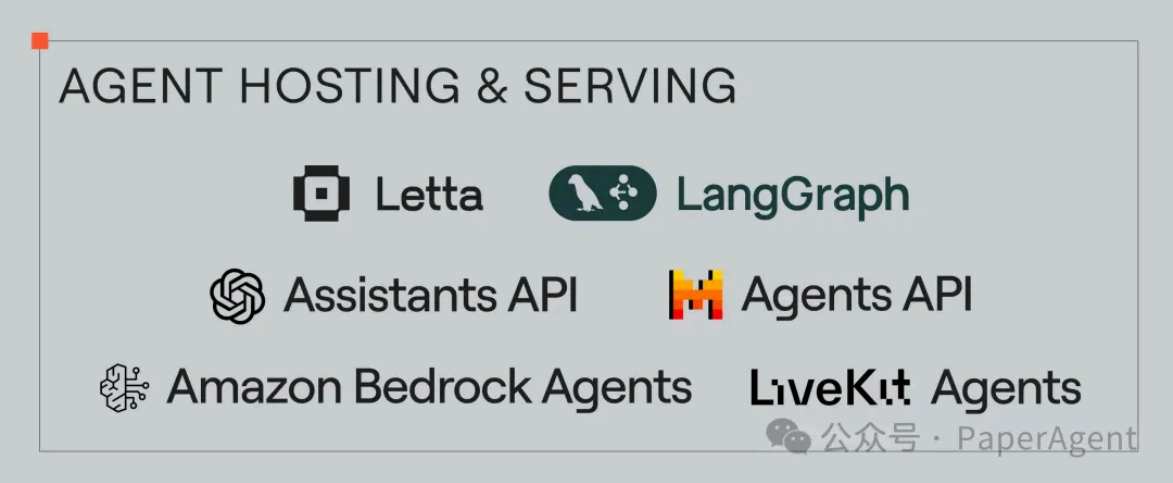


Agent框架协调LLM调用并管理Agent状态。不同的框架将为以下方面有不同的设计：

- **管理Agent的状态**：大多数框架引入了一些"序列化"状态的概念，允许Agent通过将序列化状态（例如JSON、字节）保存到文件中，在稍后的时间加载回相同的脚本——这包括状态如对话历史记录、Agent记忆和执行阶段。在Letta中，所有状态都由数据库支持（例如消息表、Agent状态表、内存块表），没有"序列化"的概念，因为Agent状态始终被持久化。这允许轻松查询Agent状态（例如，按日期查找过去的信息）。状态的表示和管理方式决定了Agent应用程序将如何随着更长的对话历史记录或更多的Agent数量进行扩展，以及如何灵活地访问或修改状态。
- **Agent的上下文窗口结构**：每次调用LLM时，框架将"编译"Agent的状态到上下文窗口。不同的框架将以不同的方式将数据放入上下文窗口（例如指令、消息缓冲区等），这可能会改变性能。我们建议选择一个使上下文窗口透明的框架，因为这最终是您可以控制您的Agent行为的方式。
- **跨Agent通信（即多Agent）**：Llama Index通过消息队列让Agent通信，而CrewAI和AutoGen有明确的抽象器用于多Agent。Letta和LangGraph都支持Agent直接相互调用，这允许集中式（通过监督Agent）和跨Agent的分布式通信。大多数框架现在支持多Agent和单Agent，因为一个设计良好的单Agent系统应该使跨Agent协作易于实现。
- **内存方法**：LLM的基本限制是它们有限的上下文窗口，这就需要管理随时间的记忆。一些框架内置了内存管理，而其他框架则期望开发者自己管理内存。CrewAI和AutoGen完全依赖基于RAG的内存，而phidata和Letta使用额外的技术，如自我编辑内存（来自MemGPT）和递归总结。Letta Agent自动配备了一套内存管理工具，允许Agent通过文本或数据搜索先前的消息，编写记忆，并编辑Agent自己的上下文窗口（您可以在这里阅读更多）。
- **支持开放模型**：模型提供商实际上做了很多幕后技巧，让LLM以正确的格式生成文本（例如用于工具调用）——例如，当它们没有生成适当的工具参数时，重新采样LLM输出，或在提示中添加提示（例如"请输出JSON"）。支持开放模型需要框架处理这些挑战，所以一些框架限制对主要模型提供商的支持。

在今天构建Agent时，正确的框架选择取决于您的应用程序，例如您是否正在构建会话Agent或工作流程，您是否希望在笔记本或作为服务运行Agent，以及您对开放权重模型支持的要求。我们预计框架之间的主要区别将出现在它们的部署工作流程中，状态/内存管理和工具执行的设计选择变得更加重要。

Agent托管和Agent服务



今天的大多数Agent框架都是为那些不存在于它们编写的Python脚本或Jupyter笔记本之外的Agent设计的。我们相信Agent的未来是将Agent视为一个_服务_，该服务被部署到本地或云基础设施上，可以通过REST API访问。就像OpenAI的ChatCompletion API成为与LLM服务交互的行业标准一样，我们预计最终会有一个赢家为Agent API。但还没有一个……。

部署Agent作为服务比部署LLM作为服务要复杂得多，因为状态管理和安全工具执行的问题。工具及其所需的依赖项和环境需求需要明确存储在数据库中，因为运行它们的环境需要由服务重新创建（这不是一个问题，当您的工具和Agent在同一个脚本中运行时）。应用程序可能需要运行数百万Agent，每个Agent都累积了越来越多的对话历史记录。当从原型转移到生产时，Agent状态不可避免地必须经历一个数据规范化过程，Agent交互必须由REST API定义。今天，这个过程通常是通过开发者编写自己的FastAPI和数据库代码来完成的，但我们预计随着Agent的成熟，这个功能将更多地嵌入到框架中。

结论

Agent技术栈仍然非常早期，我们对生态系统如何扩展和演变感到兴奋。对未来Agent技术栈发展你还有什么补充吗？

```
1  https://www.letta.com/blog/ai-agents-stack
```

推荐阅读

- 对齐LLM偏好的直接偏好优化方法：DPO、IPO、KTO
- 2024：ToB、Agent、多模态
- RAG全景图：从RAG启蒙到高级RAG之36技，再到终章Agentic RAG！
- Agent到多模态Agent再到多模态Multi-Agents系统的发展与案例讲解（1.2万字，20+文献，27张图）

欢迎关注我的公众号“**PaperAgent**”，每天一篇大模型（LLM）文章来锻炼我们的思维，简单的例子，不简单的方法，提升自己。

PaperAgent
日更，解读AI前沿技术热点Paper
212篇原创内容

公众号