

# Agents Companion

作者：Antonio Gulli, Lavi Nigam, Julia Wiesinger, Vladimir Vuskovic, Irina Sigler, Ivan Nardini, Nicolas Stroppa, Sokratis Kartakis, Narek Saribekyan, Anant Nawalgaria, 和 Alan Bount

Google



## 致谢

编辑与策展人

Anant Nawalgaria

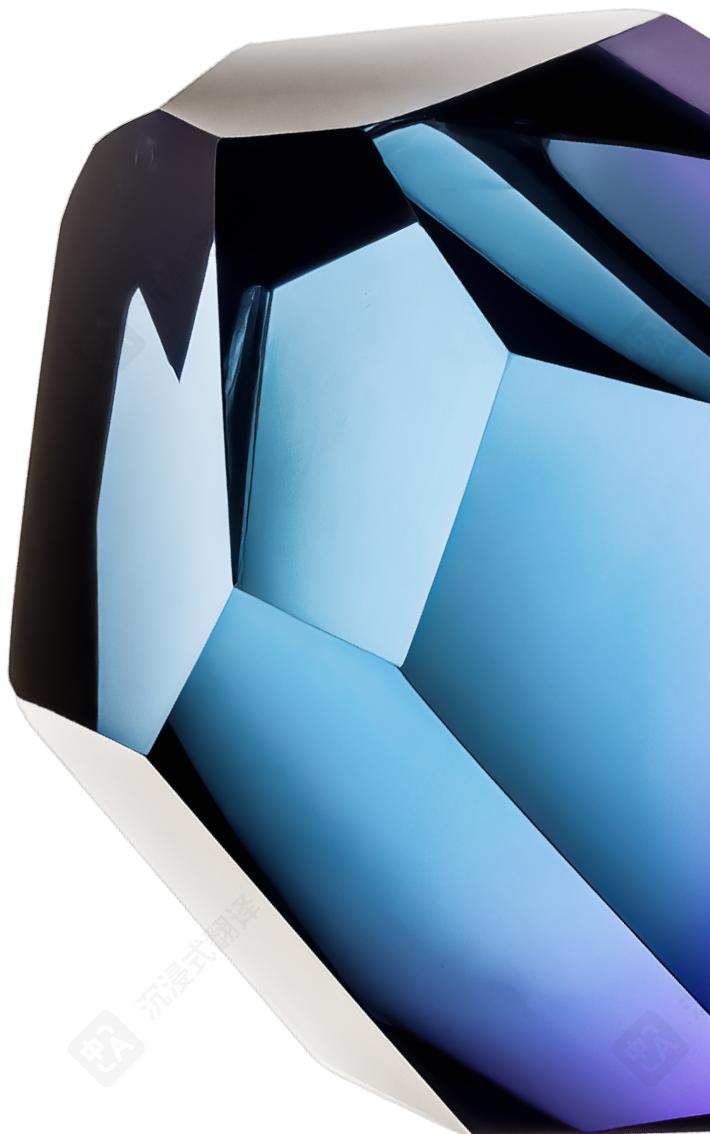
内容贡献者 Steven

Johnson Hussain

Chinoy

设计师迈克尔 ·

兰宁



# Table of contents

<b>Introduction</b>	6
<b>AgentOps</b>	8
Agent Success Metrics	12
Agent 评估	14
评估代理能力	15
Evaluating Trajectory and Tool Use	17
评估最终响应	20
Human-in-the-Loop Evaluation	21
更多关于 Agent 评估	22
<b>Multiple Agents &amp; Their Evaluation</b>	23
理解多智能体架构	24
多智能体设计模式及其商业影响	25
Important components of Agents	28
多智能体系统中的挑战	31
Multi-Agent Evaluation	32

自主式 RAG: 检索增强生成的重要演进	33
自主式 RAG 及其重要性	34
更好的搜索, 更好的 RAG	36
企业中的代理	38
代理的管理者	38
Google Agentspace	40
NotebookLM 企业版	41
Google AgentSpace 企业版	43
From agents to contractors	46
Contracts	46
Contract Lifecycle	49
合同执行	49
合同谈判	50
合同反馈	51
分包	51
汽车 AI: 多智能体架构的实际应用	54
专用智能体	54
对话式导航智能体	54
对话式媒体搜索智能体	55
消息组合智能体	56
汽车手册代理	57
通用知识代理	57
Patterns in Use	58

分层模式	58
菱形模式	59
点对点	61
协作模式	63
响应混合代理	65
自适应循环模式	66
多智能体架构在汽车 AI 中的优势	67
代理构建器	68
摘要	69
注释	74

# AI 的未来是代理化的。

## 引言

生成式 AI 代理标志着从传统的、独立的语言模型向前迈进了一大步，为问题解决和交互提供了一种动态的方法。正如原始的 Agents 论文中所定义的，代理是一个应用程序，旨在通过感知其环境并利用其可用的工具来战略性地采取行动，以实现特定的目标。代理的基本原理在于其推理、逻辑和访问外部信息的综合，使其能够执行任务和做出超越底层模型固有能力的决策。这些代理具有自主操作的能力，可以独立追求其目标并主动确定后续行动，通常无需明确的指令。

代理的架构由三个基本要素组成，这些要素驱动其行为和决策：

- **模型：**在代理的框架中，术语“模型”指的是作为中央决策单元的语言模型（LM），它采用基于指令的推理和逻辑框架。模型可以是通用型、多模态或微调的，具体取决于代理的特定需求。
- **工具：**工具对于弥合代理的内部能力与外部世界之间的差距至关重要，它们促进与外部数据和服务的交互。这些工具使代理能够访问和处理现实世界的信息。工具可以包括扩展、函数和数据存储。扩展在 API 和代理之间架起桥梁，使代理能够无缝执行 API。函数是执行特定任务的独立代码模块。数据存储提供对动态和最新信息的访问，确保模型的响应基于事实性和相关性。
- **编排层：**编排层是一个循环过程，它规定了智能体如何同化信息、进行内部推理，并利用这些推理来指导其后续行动或决策。该层负责维护记忆、状态、推理和规划。它采用提示工程框架来引导推理和规划，促进与环境的更有效交互和任务完成。ReAct、思维链（CoT）和思维树（ToT）等推理技术可以在此层中应用。

在建立这些基础概念之上，这篇伴侣论文旨在为开发者服务，并作为“102”指南，探讨更高级的主题。它提供了对智能体评估方法论的深入探讨，以及谷歌智能体产品在实际应用中增强智能体解决复杂现实问题的能力的实用案例。

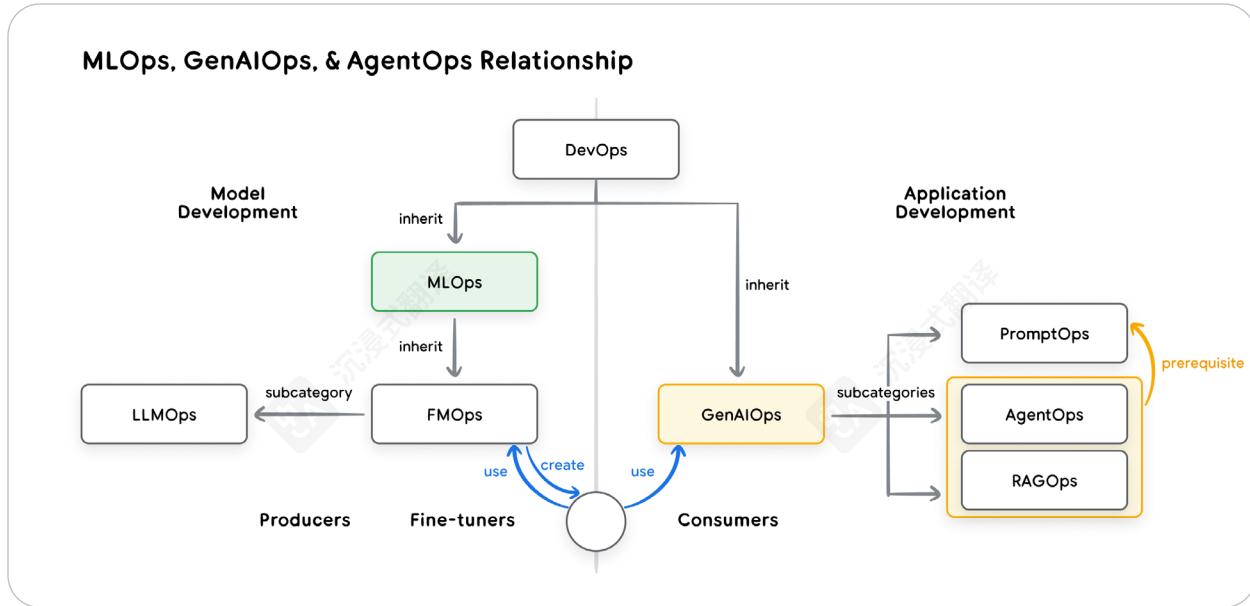
在探索这些理论概念时，我们将研究它们如何在现实世界的实现中体现，特别关注汽车 AI 作为一个引人入胜的案例研究。汽车领域展示了多智能体架构在生产环境中的挑战和机遇。现代汽车需要能够连接或断开连接的对话式界面，在安全性和用户体验之间平衡设备端和云处理，并在导航、媒体控制、消息传递和车辆系统之间无缝协调专业功能。通过这个汽车视角，我们将看到不同的协调模式——分层、协作和点对点——如何结合起来，在具有显著约束的环境创建健壮、响应迅速的用户体验。这个案例研究说明了多智能体系统的实际应用，各行业的企业可以将其调整到自己的领域。

任何使用生成式 AI 构建的人都很快意识到，从想法到概念验证很容易，但要确保高质量的结果并进入生产则相当困难——生成式 AI 智能体也不例外。质量和可靠性是部署到生产中最常被提及的担忧，而“智能体运维（AgentOps）”流程是优化智能体构建的解决方案。

## AgentOps

在过去的两年里，生成式人工智能（GenAI）领域经历了重大变化，企业客户关注如何将相关解决方案进行运营化。这导致了各种描述 GenAI 运营化的术语，例如用于 GenAI 的 MLOps、LLMOPs、FMOps 和 GenAIOps。

代理和运营（AgentOps）是 GenAIOps 的一个子类别，专注于代理的高效运营化。其主要附加组件包括内部和外部工具管理、代理大脑提示（目标、配置文件、指令）和编排、记忆和任务分解。

图 1. DevOps、MLOps 和 AgentOps 之间的关系。<sup>13</sup>

这些“Ops”中的每一个都需要版本控制、通过 CI/CD 进行自动化部署、测试、日志记录、安全性和（关键地）指标等能力。每个系统通常会根据指标实现某种形式的优化——衡量系统正在做什么和没有做什么，衡量结果和业务指标，并自动化流程以实现更全面的指标，并逐步逐步改进。这种做法可能被称为“A/B 实验”或“ML Ops”或“指标驱动开发”，但它们都源于相同的一般方法，我们将依赖这些原则来构建 AgentOps。

请记住，新的做法不会取代旧的。DevOps 和 MLOps 的最佳实践对于 AgentOps 仍然是必要的，因为它们是依赖项。例如，Agent 工具使用，其中基于 Agent 编排调用 API，通常使用与您将要相同的 API

使用非代理软件调用。认证和密钥管理、安全、隐私、异常处理、速率限制、配额和可扩展性仍然是关键，除了代理设计之外，还需要在 API 设计中仔细考虑。

让我们来定义这些“ops”术语，以帮助区分它们：

- **开发与运维 (DevOps)** 是通过整合人员、流程和技术元素来高效地将确定性软件应用程序生产化的实践。DevOps 是所有以下术语的基础。
- **机器学习运维 (MLOps)** 基于 DevOps 的能力，专注于机器学习模型的效率化生产化。主要区别在于机器学习模型的输出是非确定性的，依赖于输入数据（垃圾进，垃圾出）。
- **基础模型运维 (FMOps)** 扩展了 MLOps 的能力，专注于预训练（从头训练）或定制（微调）的基础模型的效率化生产化。
- **提示与运维 (PromptOps)** 是 GenAIOps 的一个子类别，专注于有效运营提示。其主要附加能力包括提示存储、血缘关系、元数据管理（包括评估分数）、集中的提示模板注册表和提示优化器。
- **检索增强生成与运维 (RAGOps)** 是 GenAIOps 的一个子类别，专注于高效运营 RAG 解决方案。其主要附加能力包括通过离线数据准备（包括清理、分块、向量化、相似性搜索和重新排序）的检索过程，以及通过提示增强和基础生成的生成过程。

- 代理和运营（AgentOps）是 GenAIOps 的一个子类别，专注于代理的高效运营化。其主要附加组件包括内部和外部工具管理、代理大脑提示（目标、配置文件、指令）和编排、内存和任务分解。

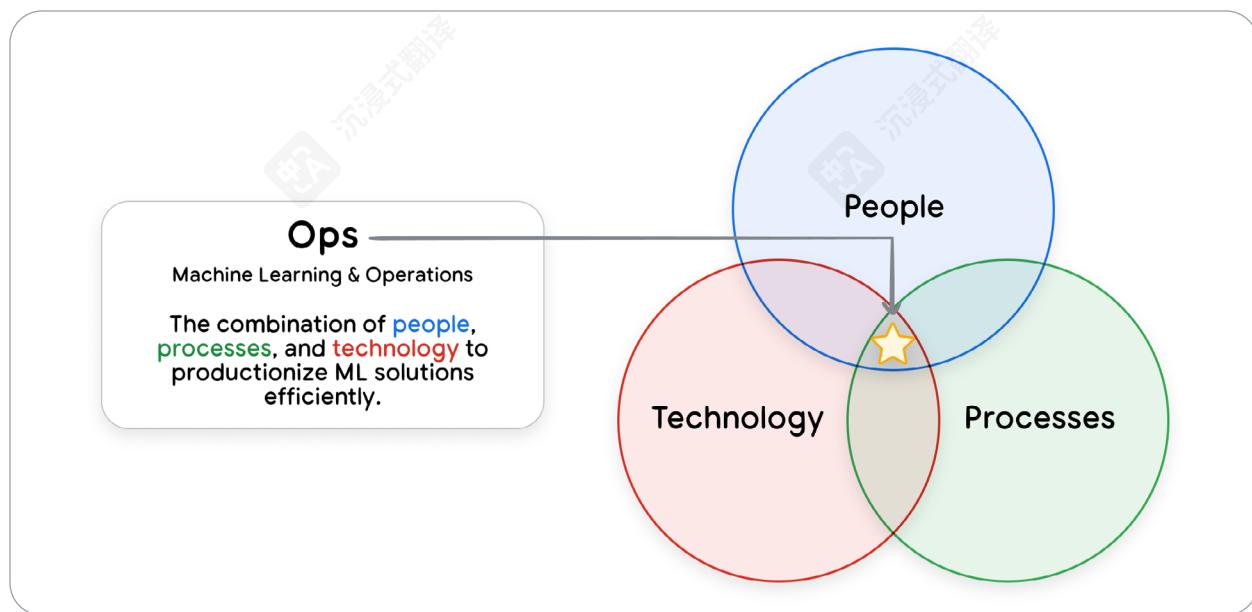


图 2。这些 “Ops” 都关于技术、流程和人员<sup>14</sup>

所有这些 “Ops” 本质上都是人员、流程和技术协同工作的和谐融合，以高效地将机器学习解决方案部署到生产环境中。必须认识到，Ops 不仅仅是技术；它不仅仅是构建和交付一个 ML 管道。成功的 Ops 实施会深入考虑客户的运营模型、他们现有的业务单元以及他们整体的组织结构。这种整体方法确保技术能够满足他们的特定需求，无缝集成到业务中并最大化价值。

下一节将详细涵盖代理评估，这是 AgentOps 和自动化捕获有用指标的重要组成部分。在我们深入之前，让我们从一个思想实验开始；想象在生产环境中为你的新代理设置一个 A/B 实验。处理组得到你的新代理，而控制组则没有。在这种情况下，你测量哪些指标来确定处理组是否表现更好？你测量哪些指标来确定项目的投资回报率？是完成的目标，还是销售总额，还是用户旅程中的一系列关键步骤？除了更详细的代理评估指标外，这些指标必须被理解、监控并易于分析。

## 代理成功指标

指标对于构建、监控和比较代理的修订版本至关重要。业务指标，如收入或用户参与度，可能超出了代理本身的范围，但这些应该是你代理的**北极星指标**。

大多数代理都是围绕完成目标而设计的，因此**目标完成率**是一个关键的指标来跟踪。类似地，一个目标可能会分解为几个关键任务或关键用户交互。每个这些关键任务和交互都应该独立地仪器化和测量。

所以在我们深入代理本身的细节之前，我们已经确定了几个指标，你应该能够轻松地在仪表板上跟踪。每个业务指标、目标或关键交互，将以熟悉的方式聚合：尝试次数、成功次数、比率等。此外，你应该能够从任何应用程序遥测系统中获取的指标对于代理来说也非常重要的来跟踪，例如延迟、错误等指标。

这些指标都不是特定于代理的，你可以为任何软件跟踪它们，但对于代理构建者来说它们甚至更重要。确定性代码只做你告诉它做的事情，而代理可以做更多的事情，它依赖于在大量数据上训练的大型语言模型。这些高级指标的监控是可观察性的重要部分。你可以将它们视为代理的关键绩效指标（KPI），并且它们允许在整体上进行可观察性，即你代理的更高层次视角。

人类反馈也是需要跟踪的一个更关键的指标。在代理或任务上下文中，一个简单的  或用户反馈表单，可以大大帮助你理解你的代理做得好的地方和需要改进的地方。这种反馈可以来自消费系统的最终用户，但也可以来自员工、质量测试人员以及审查代理的过程或领域专家。

对于代理构建来说，更详细的可观察性也非常重要，能够看到和理解代理正在做什么以及为什么这样做。代理可以被“跟踪”来记录代理的所有内部工作，而不仅仅是关键任务和用户交互。在概念上，你可以将每个内部步骤作为指标来衡量，但这很少被做。相反，这些详细的跟踪用于在指标或手动测试显示问题时调试代理，你可以深入细节并看到出了什么问题

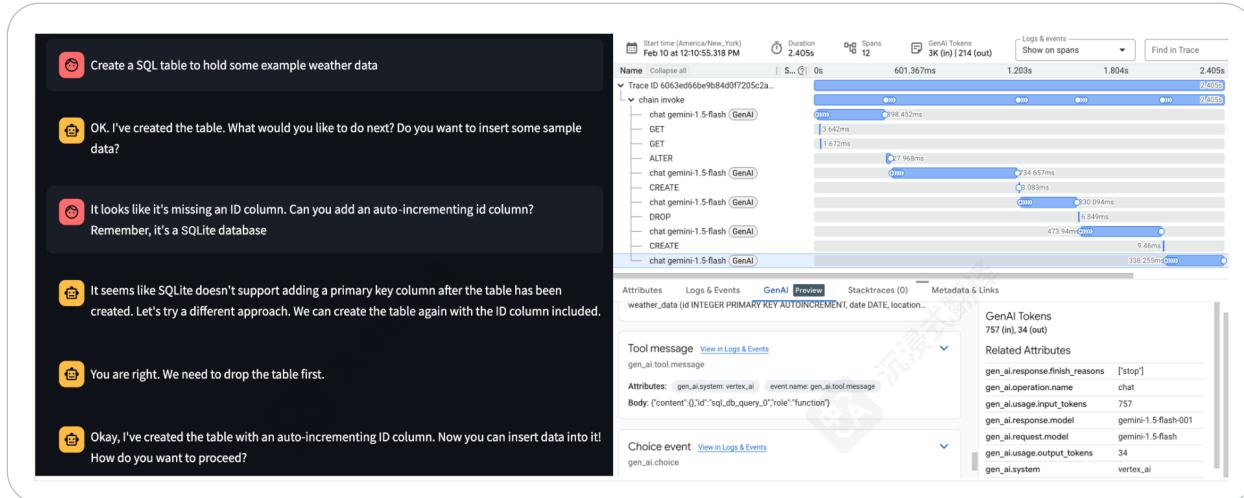


图 3：云可观察性示例，展示了一个带有工具和 LLM OpenTelemetry 跨度的代理的跟踪信息。<sup>15</sup>

到目前为止，我们一直在谈论业务指标、目标、任务、人工反馈和跟踪——这些都是理解你的代理在生产中的行为和影响的方式。在开发代理时，除了手动测试，从长远来看，自动化测试将更有效率，并能提供更深入的代理行为洞察。

## Agent Evaluation

为了弥合概念验证与生产就绪型 AI 代理之间的差距，一个稳健且自动化的评估框架至关重要。与评估生成模型（其重点主要在于最终输出）不同，代理评估需要对决策过程有更深入的理解。代理评估可以分解为三个组成部分，我们将在本章中讨论：

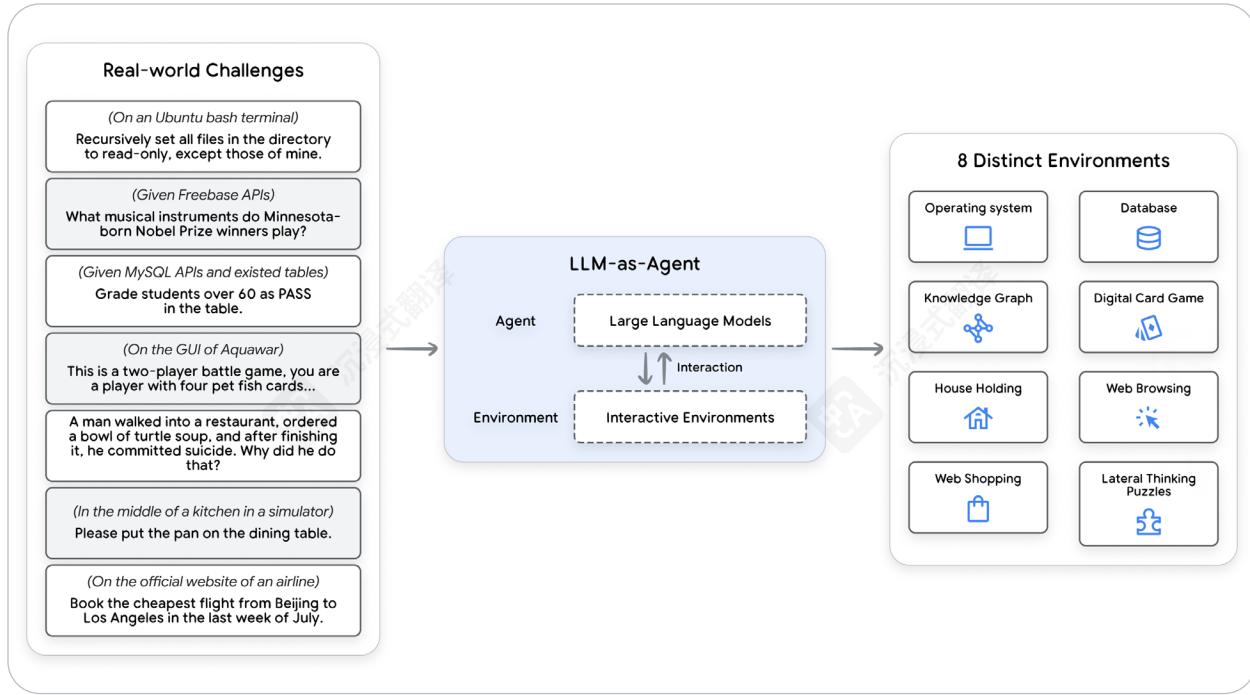
1. 评估代理能力：评估代理的核心能力，例如其理解指令和逻辑推理的能力。

2. **评估轨迹和工具使用：**分析代理达到解决方案所采取的步骤，包括其工具选择、策略以及方法的有效性。
3. **评估最终响应：**评估代理最终输出的质量、相关性和正确性。

## 评估代理能力

在评估您的特定 agentic 使用案例之前，公开可用的基准测试和技术报告可以提供有关构建 agentic 使用案例时应考虑的核心能力和限制的见解。大多数基本 agentic 能力（如模型性能、幻觉、工具调用和规划）都存在公开基准测试。例如，工具调用（选择和使用适当工具的能力）由伯克利函数调用排行榜 (BFCL)<sup>16</sup> 和  $\tau$ -bench<sup>17</sup> 等基准测试演示，这些基准测试还概述了常见错误。另一个例子，PlanBench<sup>18</sup> 旨在评估跨多个领域和特定能力的规划和推理能力。

但是工具调用和规划不是您应该考虑的唯一能力。代理从它们的 LLM 继承行为，并从它们的每个其他组件继承。同样，代理和用户交互在传统的对话设计系统和工作流系统中具有历史记录，因此可以继承用于确定这些系统有效性的指标和测量方法。

图 4: AgentBench 的高层次示意图。<sup>19</sup>

像 AgentBench<sup>19</sup> 这样的整体智能体基准试图在多个场景中捕获端到端性能。当场景反映您的智能体用例时，这是更现实的，但如果它正在测试您的智能体未实现的特性，则不是。同时模拟环境、工具、指令和用例要求的方式既具体又通用是困难的。公司和组织正在为专门的用例设置公共基准，例如 Adyen 的数据分析师排行榜 DBAStep<sup>20</sup>，它可能会给您更针对性的评估——如果您理解评估方法和排行榜上的智能体。

公共基准是一个有价值的起点，可以帮助您了解可能性和识别需要注意的陷阱。大多数基准评估都包括对常见故障模式的讨论，这些讨论可以指导您建立自己的、用例特定的评估框架。

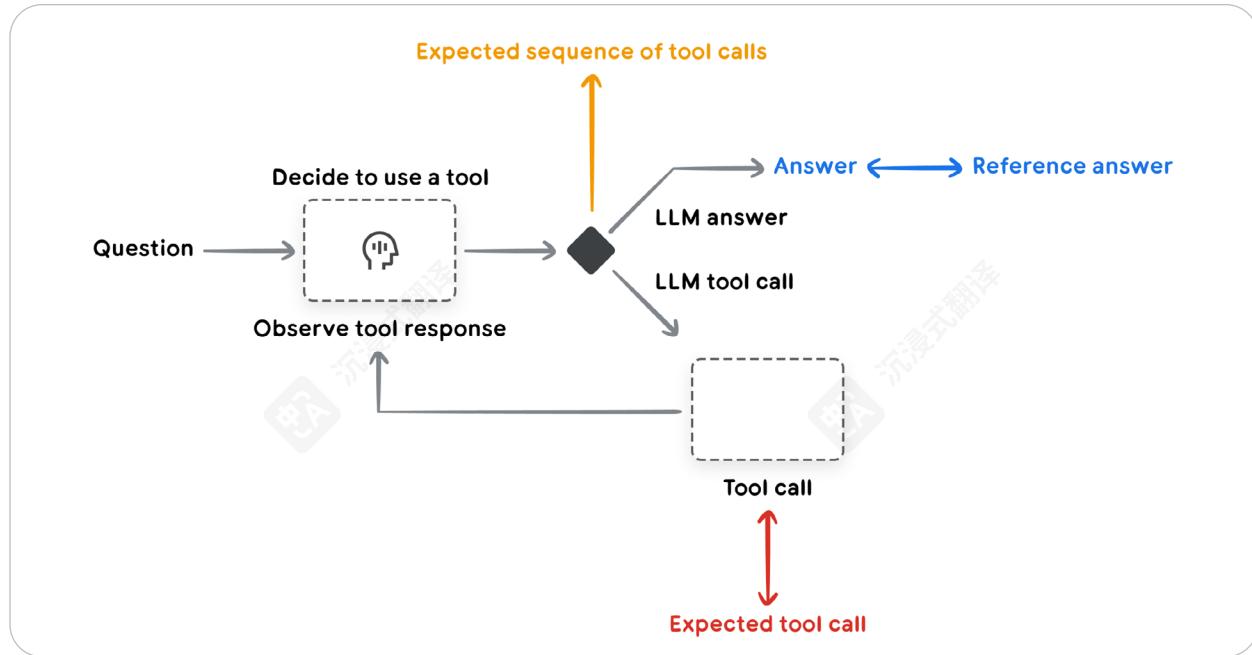
除了公开评估之外，你还需要在各种场景中评估你的代理的行为，以确保它做你所期望的事情。你正在模拟与你的代理的交互，并评估它的响应方式。这包括评估最终响应以及它沿途采取的一系列步骤（轨迹）。这些建议你从开始使用的两种最常见和最实用的方法。除了这些之外，你还可以使用许多其他评估技术，无论是用于子组件的更详细信息还是更广泛的方法。

软件工程师会将这与代码的自动化测试进行比较。投资于自动化测试可以为你节省时间，并让你对你正在构建的软件有信心。对于代理来说，这种自动化会更快地带来时间和信心的回报。管理评估数据集对于准确代表代理将遇到的使用案例至关重要，甚至比软件测试更为重要。

## Evaluating Trajectory and Tool Use

一个代理通常在向用户响应之前执行几个动作。它可能会将用户输入与会话历史进行比较以消除一个术语的歧义，或者查找政策文档、搜索知识库或调用 API 以保存工单。每个这些动作都是一条路径上的一个步骤，也称为“动作轨迹”。每次你的代理做某事时，在底层都有一个动作轨迹。

比较你期望代理采取的轨迹与代理实际采取的轨迹，对于希望调试其应用程序、识别错误或低效性，并最终提高性能的开发人员特别有用。

图 5: LangSmith 最终评估和轨迹的示意图。<sup>21</sup>

以下六个基于真实数据的自动轨迹评估提供了不同的视角来评估代理的性能：

- 完全匹配：**要求 AI 代理生成一系列动作（一个“轨迹”），完美地反映理想解决方案。这是最严格的指标，不允许偏离预期路径。
- 顺序匹配：**该指标评估智能体完成预期轨迹的能力，同时兼顾额外的、不受惩罚的行为。成功定义为按顺序完成核心步骤，并允许额外的行为。
- 任意顺序匹配：**与顺序匹配相比，该指标不再考虑顺序。它询问智能体是否包含了所有必要的行为，但不关注行为执行的顺序，并允许额外的步骤。

4. 精度：根据参考轨迹，预测轨迹中的工具调用中有多少是相关或正确的？
5. 召回率：参考轨迹中的必要工具调用中有多少被实际捕获在预测轨迹中？
6. 单工具使用：判断特定动作是否在代理轨迹内。此指标有助于理解代理是否已学会使用特定工具。

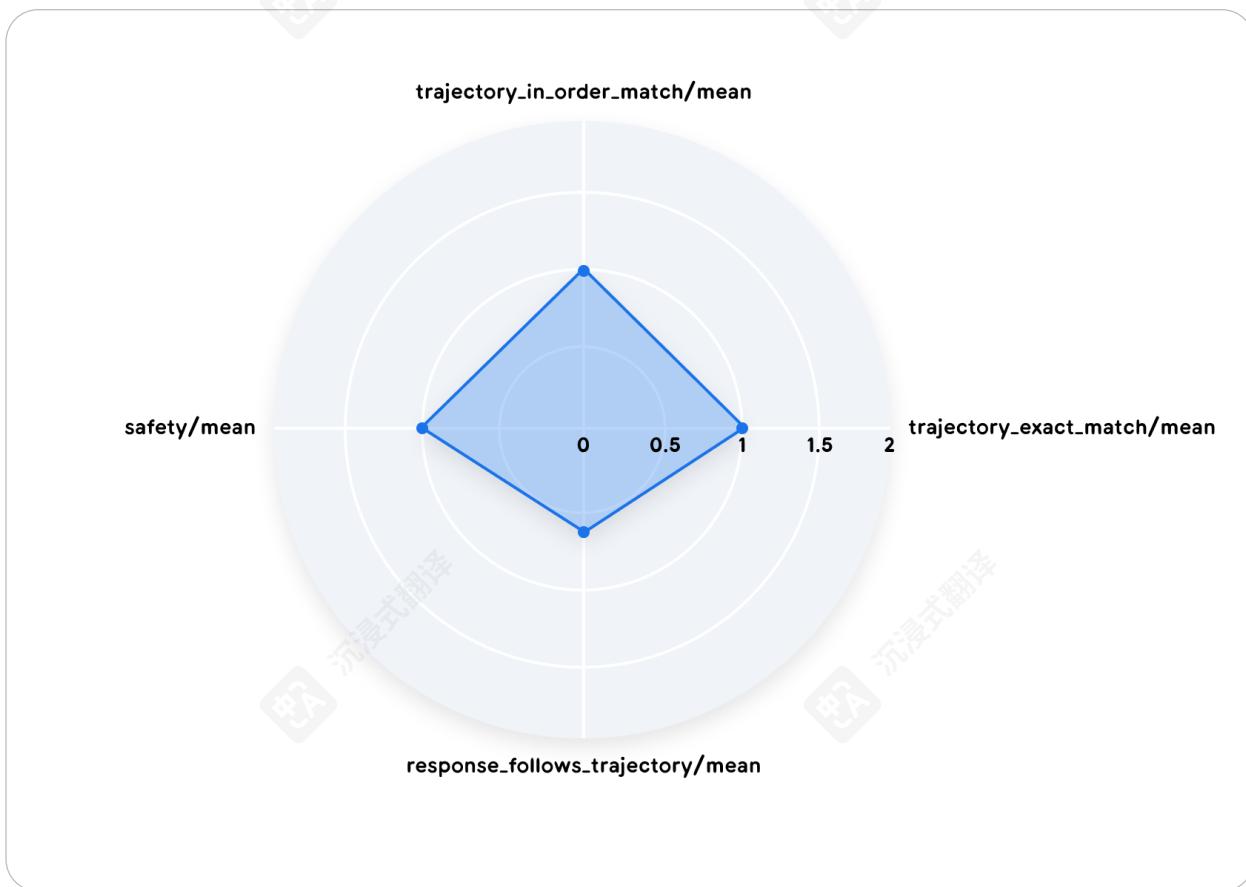


图 6：一个雷达图，绘制了单个轨迹评估和几个指标。<sup>24</sup>

将这些指标视为分析和对齐您代理轨迹的不同视角。每个指标都提供了一个独特的视角，但并非所有指标都与每种情况都相关。例如，某些用例要求严格遵循理想轨迹，而其他用例则允许更多的创造性偏差。这种评估方法的一个明显局限性是，您需要有一个参考轨迹才能使其生效。虽然在此讨论中基于真实值的自动轨迹评估在流行库中很普遍。研究正在推进代理自评器的使用，以提高评估效率，例如 Agent as a Judge, 2024<sup>22</sup>。

## 评估最终响应

最终响应评估归结为一个简单的问题：您的代理是否实现了其目标？您可以定义自定义成功标准，以满足您的特定需求，以衡量这一点。例如，您可以评估零售聊天机器人是否准确回答了产品问题，或者研究代理是否有效地以适当的语气和风格总结了发现。为了自动化此过程，您可以使用 autorater。Autorater 是一个充当法官的大型语言模型。给定输入提示和生成的响应，它通过根据用户提供的一组标准评估响应来模拟人类评估。为了使此评估生效，必须考虑的是，由于缺乏真实值，您需要非常精确地定义您的评估标准，因为这是您的评估的核心。您在各种库中找到一些预定义的标准，将它们视为起点并进行调整，以提供您对“好”的定义。

## Human-in-the-Loop Evaluation

智能体开发与智能体评估领域正在迅速发展。评估 AI 智能体面临着重大挑战，包括明确目标、设计逼真环境、管理随机行为以及确保公平性和偏差缓解，特别是在具有社会影响力的应用中。因此，在之前讨论的自动化评估（涉及预定义指标和自动评分员）之外，结合人类参与式评估方法至关重要。人类参与式评估对于需要主观判断或创造性解决问题的任务非常有价值，它还可以用于校准和确认您的自动化评估方法是否真正有效并与您的偏好一致。主要优势包括：

- **主观性：**人类可以评估难以量化的品质，例如创造力、常识和细微差别。
- **情境理解：**人类评估者可以考虑到代理行为的更广泛背景及其影响。
- **迭代改进：**人类反馈为优化代理行为和学习过程提供了宝贵的见解。
- **评估评估者：**人类反馈可以提供校准和优化您的自动评分器的信号。

要实施人类回路评估，请考虑以下方法：

- **直接评估：**人类专家直接对代理在特定任务上的表现进行评分或打分。
- **比较评估：**专家将代理的表现与其他代理或您的先前迭代进行比较。

- **用户研究：**参与者与代理互动，并提供关于其行为、可用性和整体有效性的反馈。

## More about Agent Evaluation

在本节中，我们从实际角度探讨代理评估。但这只是冰山一角。代理评估面临着许多挑战。您代理的评估数据可能难以找到。虽然可以使用合成数据或 LLM 作为评判者，但评估可能仍然不完整。此外，LLM 作为评判者的指标，例如，可能优先考虑最终结果而不是代理的推理和中间操作，可能会遗漏关键见解。此外，由于代理系统的评估在对话和工作流系统中有历史，因此有太多要探索的内容，以了解如何继承评估代理能力的方法和指标，例如在多次交互中提高任务性能的能力。多模态生成的评估带来了额外的复杂性；图像、音频和视频评估需要各自的评估方法和指标。最后，真实世界环境带来了进一步的挑战，因为它们是动态且不可预测的，使得在受控环境中评估代理变得困难。

展望未来，为解决这些开放性挑战，智能体评估领域正在迅速发展。主要趋势包括转向基于过程的评估，优先理解智能体的推理过程；增加 AI 辅助评估方法以提高可扩展性；以及更加关注实际应用场景。新标准化基准的开发也在逐渐兴起，促进智能体之间的客观比较，同时强调可解释性和可解释性旨在提供对智能体行为的更深入洞察。

评估方法	👍 优势	👎 劣势
人工评估	捕捉细微行为, 考虑人类因素	主观、耗时, 昂贵、难以扩展
LLM-as-a-Judge	可扩展、高效、一致	可能忽略中间步骤, 受限于大语言模型的能力
自动化指标	客观、可扩展、高效	可能无法完全捕捉能力, 容易受到游戏策略的影响

Table 1: A table comparing strengths and weaknesses of automated evaluations for Agents.

此时，应该很清楚，只有不断改进评估方法，我们才能确保人工智能代理在未来负责任、有效和合乎道德地开发和应用。

## Multiple Agents & Their Evaluation

代理评估，如前一节所述，评估自主人工智能代理的有效性、可靠性和适应性，已成为一个关键的焦点领域。我们看到人工智能系统发生了显著演变，向多代理架构过渡——在那里，多个专业代理合作以实现复杂目标。

一个多智能体系统就像一个专家团队，每个专家在某个特定领域有专长，共同协作解决复杂问题。每个智能体是一个独立的实体，可能使用不同的 LLM，并具有其独特的角色和上下文。智能体之间进行沟通和协作以实现共同目标。这种方法与传统单智能体系统不同，在传统系统中，一个 LLM 处理任务的各个方面。

多智能体系统相比单智能体系统具有以下几个优势：

- **增强准确性：**智能体可以互相检查彼此的工作，从而得到更准确的结果。
- **提高效率：**智能体可以并行工作，加快任务完成速度。
- **更复杂的任务处理：**大型任务可以被分解为更小、更易于管理的子任务，每个代理专注于特定的方面。
- **提高可扩展性：**通过添加具有专业能力的更多代理，系统可以轻松扩展。
- **增强容错性：**如果一个代理失败，其他代理可以接管它的职责。
- **减少幻觉和偏见：**通过结合多个代理的视角，系统可以减少幻觉和偏见的影响，从而产生更可靠和值得信赖的输出。

## 理解多代理架构

与传统的单体 AI 系统不同，多代理架构将问题分解为由专业代理处理的独立任务。每个代理具有定义好的角色，与其他代理动态交互以优化决策、知识检索和执行。这些架构支持更结构化的推理、去中心化的问题解决和可扩展的任务自动化，为单代理工作流程带来了范式转变

在核心层面，多智能体系统利用模块化、协作和层次化的原则来构建一个强大的 AI 生态系统。这些系统中的智能体可以根据其功能进行分类，例如：

- **规划代理：**负责将高级目标分解为结构化的子任务。
- **检索代理：**通过动态从外部源获取相关数据来优化知识获取。
- **执行代理：**执行计算、生成响应或与 API 交互。
- **评估代理：**监控和验证响应，确保一致性和与目标的 alignment。

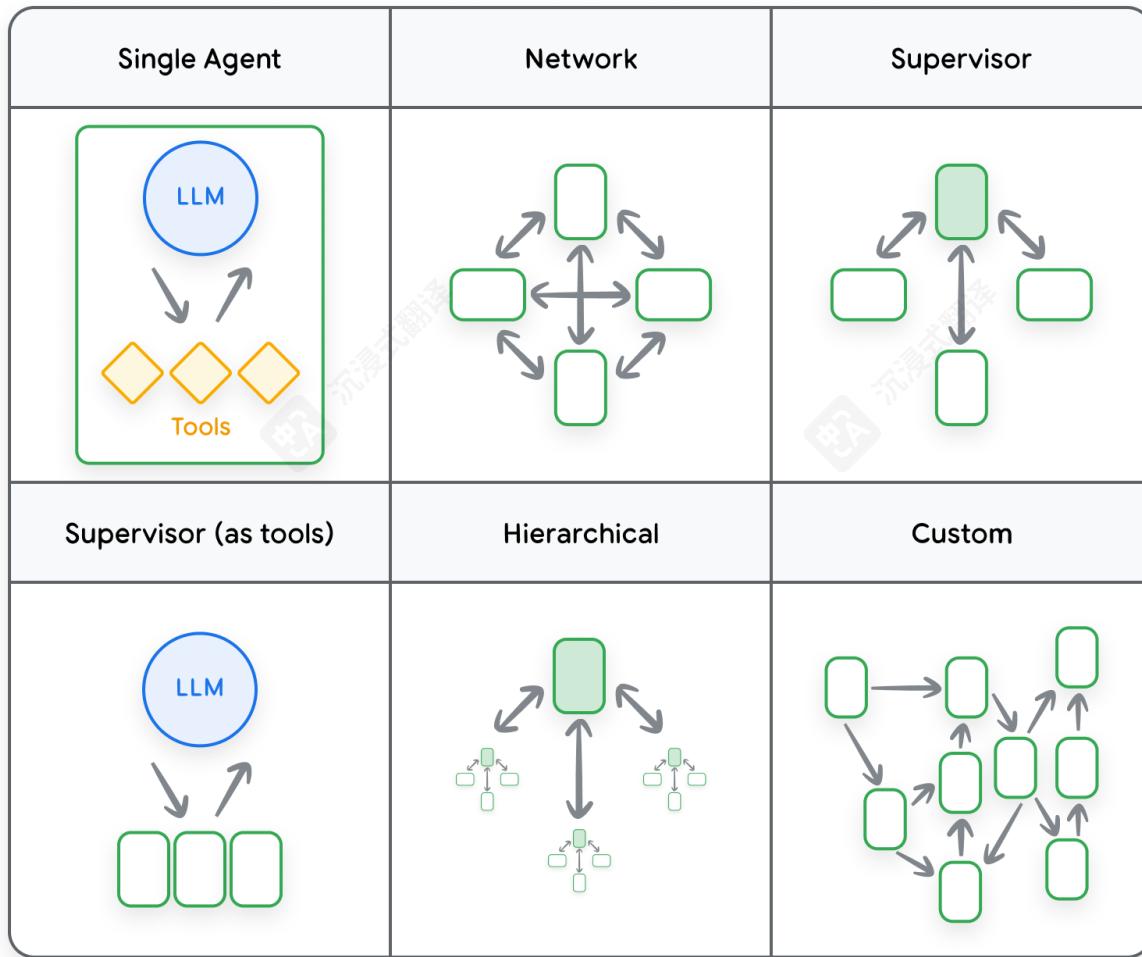
通过这些组件，多代理架构超越了简单的基于提示的交互，使 AI 驱动的流程具有适应性、可解释性和高效性。

## 多代理设计模式及其商业影响

为了设计有效的多代理架构，已经出现了特定的设计模式。这些模式定义了交互协议、委托机制和角色分配，使企业能够以结构化的方式实施 AI 驱动的自动化。一些常见的设计模式包括：

多智能体系统类型	描述	示例
顺序	代理按顺序方式工作，每个代理在完成任务之前将输出传递给下一个代理。	一个装配线，其中每个工人执行特定的操作，然后将产品传递给下一个工人。
分层	代理以组织方式排列分层结构，具有"manager" 代理协调工作流和委派任务"worker" 代理。	一个具有领导者代理的系统负责制定战略决策和跟随代理根据领导的指令执行任务的。
协作	代理一起工作协作地，共享信息和资源es to实现一个共同目标。	一个研究团队正在一个项目上工作，其中每个成员贡献他们的专业知识和见解。
竞争性	智能体可能相互竞争以实现最佳结果。最佳结果。	LLMs 作为个体玩家在像 Overcooked-AI 这样的游戏中发挥作用，其中他们必须协调他们的行动在实现共同目标的同时竞争资源并优化个人表现。

表 2：比较多智能体系统类型的表格。

图 7：一张展示不同多智能体拓扑结构的图片，来自 LangGraph 文档。<sup>2</sup><sup>3</sup>

设计模式的选择取决于具体的应用以及智能体之间期望的交互程度。

企业通过这些模式受益，包括减少运营瓶颈、提高知识检索效率和增强自动化可靠性。多智能体系统使公司能够扩展 AI 部署，同时确保决策和工作流程执行的灵活性。

## 智能体的关键组成部分

基于 LLM 的 AI 智能体的架构由几个相互关联的组成部分构成，这些组成部分对于自主运行和智能交互至关重要：

- **交互包装器：**该组件作为智能体与其环境之间的接口，管理通信并适应各种输入和输出模式。
- **内存管理：**这包括用于即时上下文的短期工作内存、缓存和会话。它还可以包括用于存储学习到的模式和经验的长时存储，例如情节、示例、技能或参考数据。它还包括“反思”功能，用于决定哪些短期项目（例如：用户偏好）应该被复制到长期内存（例如：用户配置文件）中，以及这些信息是否可以在代理、任务或会话之间共享。
- **认知功能：**这通常由思维链（CoT）、ReAct、推理、思考或规划子系统集成支持——它允许智能体将复杂任务分解为逻辑步骤并进行自我纠正。在某些情况下，这也包括用户意图的细化，如果不确定则提出澄清性问题。
- **工具集成：**该子系统使代理能够利用外部工具，扩展其自然语言处理之外的能力。动态工具注册表允许发现、注册和“工具 RAG”。
- **Flow / Routing：**这控制与其他代理的连接，促进动态邻居发现和高效的多代理系统内部通信。这可能实现为将任务委托给后台代理，或将用户交互移交给代理，或使用代理作为工具。

- **反馈循环 / 强化学习:** 这些通过处理交互结果和改进决策策略，实现持续学习和适应。对于生成式 AI 代理，这很少以传统 RL 训练的形式出现，但过去的表现指标可以被纳入未来的决策中。
- **代理通信:** 代理之间有效通信对于多代理系统的成功至关重要。代理到代理的通信协议促进了代理之间的结构化和高效通信，使它们能够达成共识并协作解决复杂问题。
- **远程代理通信:** 组织内部的代理到代理通信至关重要，允许代理共享消息、任务和知识。一旦您的多代理系统包含远程代理，通信协议就变得更加重要。异步任务和会话需要具有持久性，并在最终用户不在会话期间更新通知。代理之间的协商必须允许将用户带入会话，并支持相应的 UX 功能。
- **Agent & Tool Registry (mesh):** 当您使用超过少量工具或代理时，您需要一个强大的系统来发现、注册、管理、选择和使用来自“工具或代理网络”的工具或代理。至关重要的是工具和代理的本体和描述、它们的功能和需求以及它们的性能指标。您的代理将制定计划并选择从该系统使用哪个工具或哪个代理，这些选择由系统中的数据提供信息

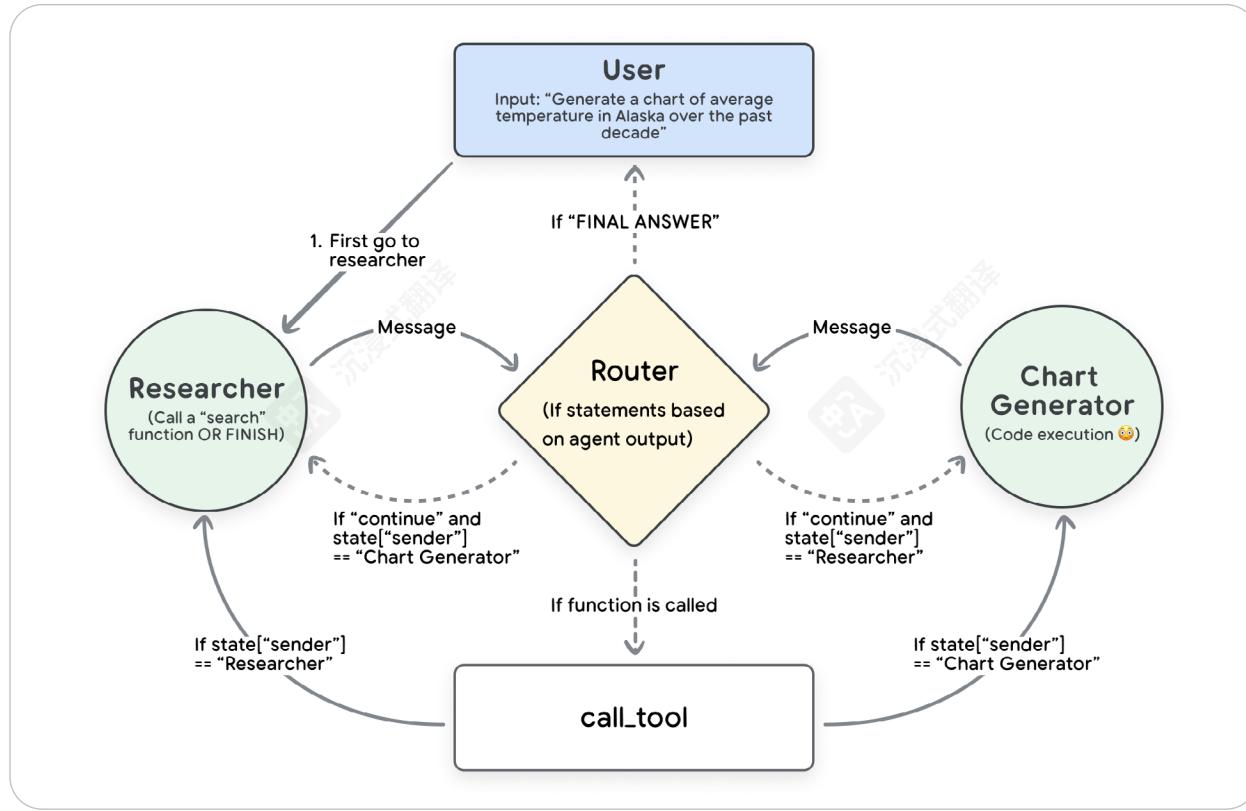


图 8：展示用户与多个、自我协调的智能体交互过程的图像。<sup>24</sup>

这些架构元素为多智能体系统内 LLM 智能体的自主运行和协作能力提供了坚实的基础。

## 多智能体系统中的挑战

虽然多智能体系统提供了许多优势，但它们也面临一些挑战：

- **任务通信：**今天大多数代理框架通过消息通信，而不是结构化的异步任务。
- **任务分配：**在多个代理之间高效地分配复杂任务可能具有挑战性，而反馈循环通常由开发人员来实现。
- **协调推理：**让代理有效地辩论和协同推理需要复杂的协调机制。
- **管理上下文：**跟踪所有代理之间的信息、任务和对话可能令人不堪重负。
- **时间和成本：**多代理交互可能计算成本高昂且耗时。这导致更高的运行时价格和更多的用户延迟。
- **复杂性：**与微服务架构允许每个微服务具有更多灵活性和简单性一样，整个系统通常变得更加复杂。

应对这些挑战对于开发健壮且高效的多智能体系统至关重要。

## Multi-Agent Evaluation

幸运的是，多智能体系统的评估是评估单个智能体系统的清晰进展。智能体成功指标保持不变，业务指标作为你的北极星，目标和关键任务成功指标，应用程序遥测指标如延迟和错误。使用跟踪来监控多智能体系统将有助于调试和理解在复杂交互过程中发生的情况。

在智能体评估部分，我们讨论了评估轨迹和评估最终响应作为智能体自动评估的两种最佳方法，这对于多智能体系统仍然适用。对于多智能体系统，一个行动轨迹可能包括你的一个或多个智能体。尽管多个智能体可能合作完成任务，但在最后向用户返回一个单一最终答案，并且可以单独评估。

由于多智能体系统可能具有更多步骤，您可以深入挖掘并在每个步骤进行评估。您可以单独评估每个智能体以及整个系统。轨迹评估是一种可扩展的方法，用于执行此操作。

有一些问题需要你提出，这些问题是对多智能体系统的，包括：

- **合作与协调：**智能体之间协同工作和协调行动以实现共同目标的效果如何？
- **规划与任务分配：**我们是否制定了正确的计划，并且是否坚持执行？子智能体是否偏离了主计划或陷入死胡同？
- **智能体利用：**智能体选择正确的智能体并将其作为工具、委托后台任务或转移用户的有效性如何？

- **可扩展性:** 随着更多智能体的加入, 系统的质量是否提高? 延迟是否降低? 我们是否更有效率或更低效?

这些问题可以指导开发人员在多智能体系统中识别改进领域。您将使用与单智能体系统相同的工具来回答这些问题, 但分析的复杂性会增加。

## Agentic RAG: A Critical Evolution in Retrieval-Augmented Generation

多智能体架构的一项重大进步是 **智能体检索增强生成 (Agentic RAG)**。传统的 RAG 管道依赖于静态方法——从向量数据库中检索知识并将其输入到 LLM 进行合成。然而, 当处理模糊、多步骤或多视角查询时, 这种方法通常会失败。

智能体 RAG 引入了 **自主检索智能体**, 这些智能体根据迭代推理主动优化其搜索。这些智能体通过以下方式增强检索:

- **上下文感知查询扩展:** 代理不会依赖于单次搜索, 而是生成多个查询改进以检索更相关和全面的结果。
- **多步骤推理:** 代理将复杂查询分解为更小的逻辑步骤, 按顺序检索信息以构建结构化响应。
- **自适应源选择:** 代理不会从单个向量数据库中获取数据, 而是根据上下文动态选择最佳知识源。
- **验证和纠正:** 评估智能体在将检索到的知识集成到最终响应之前, 会交叉检查其是否存在幻觉和矛盾。

这种方法显著提高了响应 **准确性**，**可解释性**，和 **适应性**，使其在处理法律研究、科学发现和商业智能等领域的复杂知识检索任务的企业中成为一项关键创新。

## Agentic RAG 及其重要性

Agentic RAG（检索增强生成）是一种先进的方法，它结合了 RAG 的优势和 AI 代理的自主性。传统的 RAG 系统从外部知识源检索相关信息以增强 LLM 响应。Agentic RAG 通过采用智能代理来协调检索过程、评估检索到的信息，并就如何最佳利用它做出决策，将这一过程推进了一步。

Agentic RAG 相比传统 RAG 具有以下优势：

- **提高准确性：**代理可以评估检索信息的质量，并就哪些来源值得信任做出决策，从而提供更准确和可靠的响应。
- **增强上下文理解：**代理可以结合用户查询的上下文和检索到的信息，生成更相关和有意义的响应。
- **提高适应性：**代理可以适应不断变化的信息需求，并动态调整其检索策略，以提供最新且最相关的信息。这种适应性在信息不断变化复杂领域至关重要，例如医疗保健、金融和法律研究。

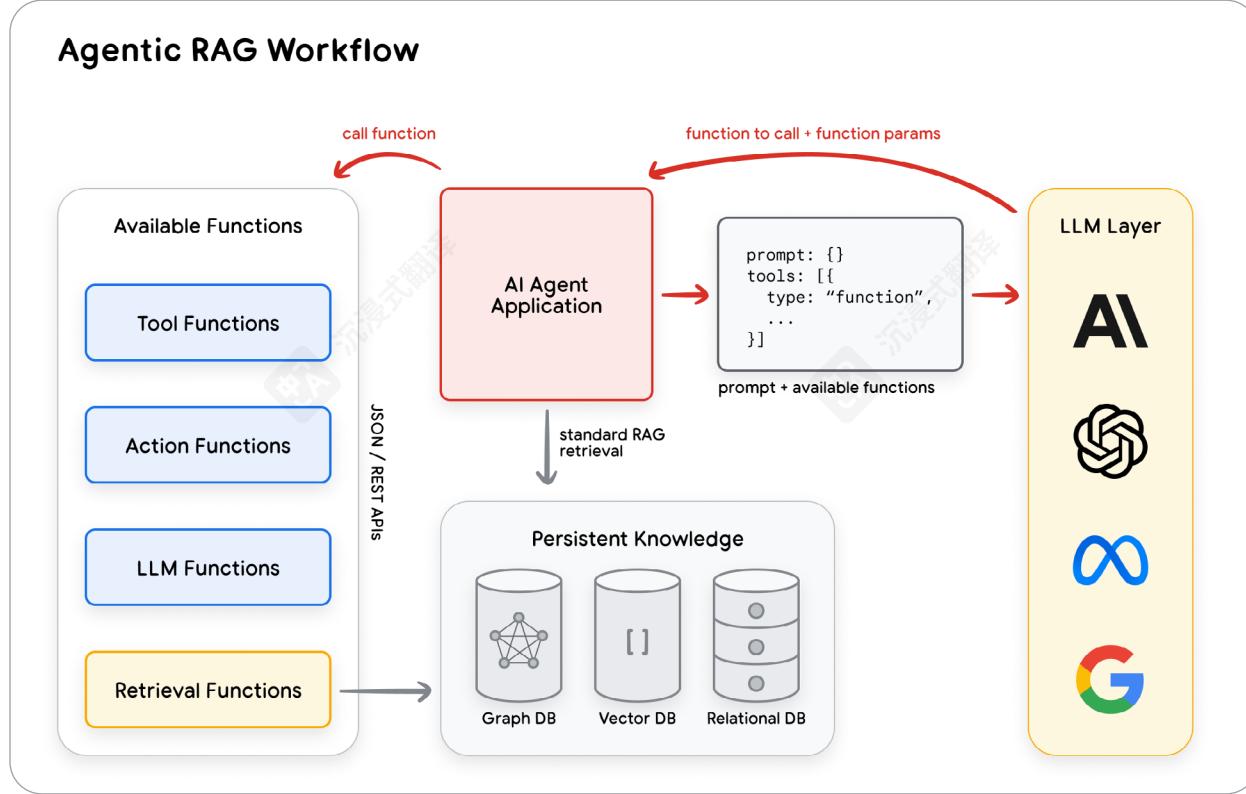


Figure 9: A diagram of Agentic RAG from Vectorize.io.<sup>25</sup>

在信息不断变化且复杂的领域（如医疗保健、金融和法律研究），Agentic RAG 尤其有价值。例如，在医疗保健领域，Agentic RAG 可以帮助导航复杂的医疗数据库、研究论文和患者记录，为医生提供全面和准确的信息。

## 更好的搜索，更好的 RAG

几乎所有 RAG 方法都需要搜索引擎来索引和检索相关数据。代理的引入允许细化查询、过滤、排序和最终答案。Agentic RAG 代理正在执行多个搜索以检索信息。

对于试图优化现有 RAG 实现的开发人员来说，在引入代理之前，通常最有价值的是改进搜索结果（以召回率衡量）。提高搜索性能的一些主要技术是：

- **解析源文档并分块它们：**Vertex AI 布局解析器可以处理复杂的文档布局、嵌入式表格和嵌入式图像（如图表），并使用语义分块器来保持块的主题性，并使用标题层次结构。
- **为你的块添加元数据：**同义词、关键词、作者、日期、标签和类别允许你的搜索进行提升、隐藏和过滤；这些允许你的用户或你的代理对搜索结果有更多的控制。
- **微调嵌入模型**或添加一个**搜索适配器**，该适配器会改变嵌入空间：这些允许可搜索的向量索引比通用嵌入模型更好地表示你的领域。
- **更快的向量数据库**可以提高搜索质量：要搜索嵌入，你必须权衡速度和准确性，升级到超快的**Vertex AI 向量搜索**可以同时提高延迟和质量
- **使用一个排序器：**向量搜索速度快但近似，它们应该返回几十个或几百个结果，需要由一个更复杂的系统重新排序，以确保前几个结果是最相关或最佳答案。
- **实现检查基础：**作为一种对基础生成安全的保障，您可以确保每个短语实际上都可以通过检索到的块进行引用。

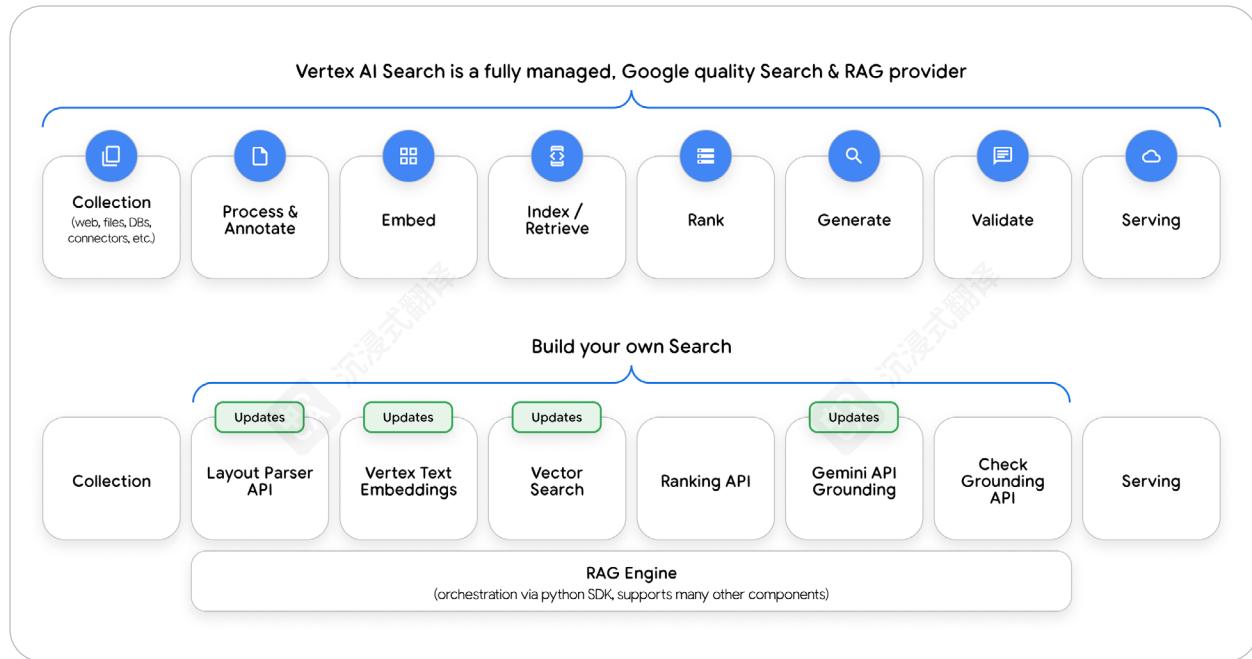


图 10：常见 RAG 和搜索组件的示意图，展示了 Vertex AI Search<sup>26</sup>、搜索构建 API<sup>27</sup> 和 RAG 引擎<sup>28</sup>。

Vertex AI Search<sup>26</sup> 是一个强大的搜索引擎，为您的数据提供谷歌质量的搜索服务，并且可以与任何 RAG 或 Agentic RAG 实现一起使用。以上每个组件都在 Vertex AI Search 中自动可用，无需任何开发时间。对于想要构建自己的搜索引擎的开发者，以上每个组件都作为独立的 API<sup>27</sup> 暴露出来，RAG 引擎<sup>28</sup> 可以轻松地编排整个流程，具有类似 LlamaIndex 的基于 Python 的界面。

# 企业中的代理

## 代理管理者

2025年是代理之年，这一点在企业中将尤为明显，这些企业将开发和使用代理来协助员工执行特定任务，或在后台自主运行以执行自动化。商业分析师可以毫不费力地发现行业趋势，并创建由AI生成的洞察力驱动的引人入胜、数据驱动的演示文稿。人力资源团队可以通过简化的入职流程彻底改变员工体验，即使对于像401k选择这样复杂的任务也是如此。软件工程师可以主动识别和解决错误，使他们能够更高效地构建和迭代，并加速部署周期。营销人员可以解锁更深入的性能分析，优化内容推荐，并毫不费力地微调活动以获得更好的结果。

我们看到两种类型的代理正在出现：

1. “助手”：与用户交互、接受任务、执行任务并返回给用户的代理。由Gems或GPTs等框架推广的对话式代理通常属于此类。助手可以是通用的——能够帮助各种任务——或专门针对特定领域或任务。例如，包括帮助安排会议、分析数据、编写代码、编写营销新闻稿、帮助卖家发现销售机会，甚至根据用户请求对特定主题进行深入研究的代理。这些代理可以是同步的，快速返回请求的信息或执行请求的任务，或者它们可以运行较长时间后才返回（如深入研究的代理）。

2. “**自动化代理**”：在后台运行的代理，监听事件，监控系统或数据的变化，然后做出智能决策并采取行动。行动可能包括对后端系统采取行动、执行一些测试以验证观察结果、修复问题、通知正确的员工等。这可以真正被视为未来自动化的重要支柱。在过去，我们必须编写特殊代码来指定自动化的逻辑，现在我们可以开始依赖人工智能代理的智能和通用决策能力。

与其简单地调用代理来执行任务并等待输出，知识工作者将越来越成为代理的管理者。他们将分配任务给多个代理并进行管理，检查是否有任何代理需要帮助或需要批准才能继续，使用特定代理的输出启动新任务，监控长时间运行任务的执行情况以进行审查和引导它们朝正确的方向。将出现新的用户界面来允许这种虚拟团队管理，以允许编排、监控和管理一个多代理系统，其中代理执行任务、调用（甚至创建）其他代理。

**Google Agentspace** 致力于提供此类体验并允许：

- 通过使用无代码 / 低代码界面或全代码框架创建新代理 来实现专业助手和自动化代理
- 配置和管理公司管理员代理访问权限
- 在适当的时候调用正确的代理
- 监控、管理和编排多个适合团队管理的 UI 中的代理

## Google Agentspace

Google Agentspace<sup>29</sup> 提供了一套 AI 驱动的工具，旨在通过促进对相关信息访问和自动化复杂、代理式工作流程来提升企业生产力。它利用了 Gemini 的高级推理能力、 Google 搜索基础设施的强大功能以及对企业数据的受控访问，无论数据物理位置如何。

Agentspace 直接解决了传统知识管理系统固有的局限性，这些系统在个性化、自动生成答案、上下文理解和全面信息检索等方面经常不足。它通过整合不同的内容来源、生成基于事实的个性化响应以及简化运营工作流程，赋予员工高效的信息访问能力。主要功能包括能够摄取各种数据格式、从软件即服务（ SaaS ）平台同步数据、提供受控访问的搜索结果和 AI 生成的答案，以及将 AI 辅助和智能代理集成到统一的工作流程中。

Agentspace Enterprise 的架构基于几个核心原则。其中最重要的是内置信任，它通过单点登录（ SSO ）认证、集成权限模型和用户级访问控制等特性来优先考虑安全性、可解释性和治理。Google 的高级智能通过机器学习来识别用户行为和内容模式，通过语义理解、知识图谱和 LLM 生成高度相关的结果。通过连接到各种企业系统并提供按需和自动化的数据刷新，实现了通用连接性，从而消除了信息孤岛。通过为搜索功能、推荐、 LLM 和知识图谱提供粒度控制，实现了企业级定制和用户级个性化，根据个人用户角色、权限和搜索提供定制化体验。

历史。实时反馈和自适应机制通过机器学习和用户输入实现结果的持续优化。混合检索增强生成（RAG）允许自定义数据融合，为基于企业数据的生成式 AI 应用提供支持。最后，可扩展性是关键的设计考虑因素，系统经过设计以适应跨地域、语言和高峰使用需求的增长。

安全始终是首要考虑。Google Agentspace 基于 Google Cloud 安全设计基础设施构建，让您可以在整个组织中部署 AI 代理。它提供细粒度的 IT 控制，包括基于角色的访问控制（RBAC）、VPC 服务控制以及 IAM 集成，确保您的数据始终受到保护和合规。安全是 Agentspace 的基础原则。基于 Google Cloud 的安全基础设施，它为 AI 代理的部署提供了一个强大的环境。细粒度的 IT 控制，包括基于角色的访问控制（RBAC）、虚拟私有云（VPC）服务控制以及身份和访问管理（IAM）集成，保证了数据保护和合规性。这些安全措施对于保护敏感信息至关重要，并让用户可以放心地在整个组织中部署 AI 代理。

## NotebookLM 企业版

NotebookLM<sup>30</sup> 是一款研究和学习工具，旨在简化理解和综合复杂信息的过程。它使用户能够上传各种源材料，包括文档、笔记和其他相关文件，并利用人工智能的力量来促进更深入的理解。想象一下研究一个多方面的主题；NotebookLM 允许你将所有分散的资源整合到

一个单一的、有组织的工作空间。本质上，NotebookLM 作为一个专门的研究助理，加速了研究过程，并使用户能够超越单纯的信息收集，实现真正的理解。

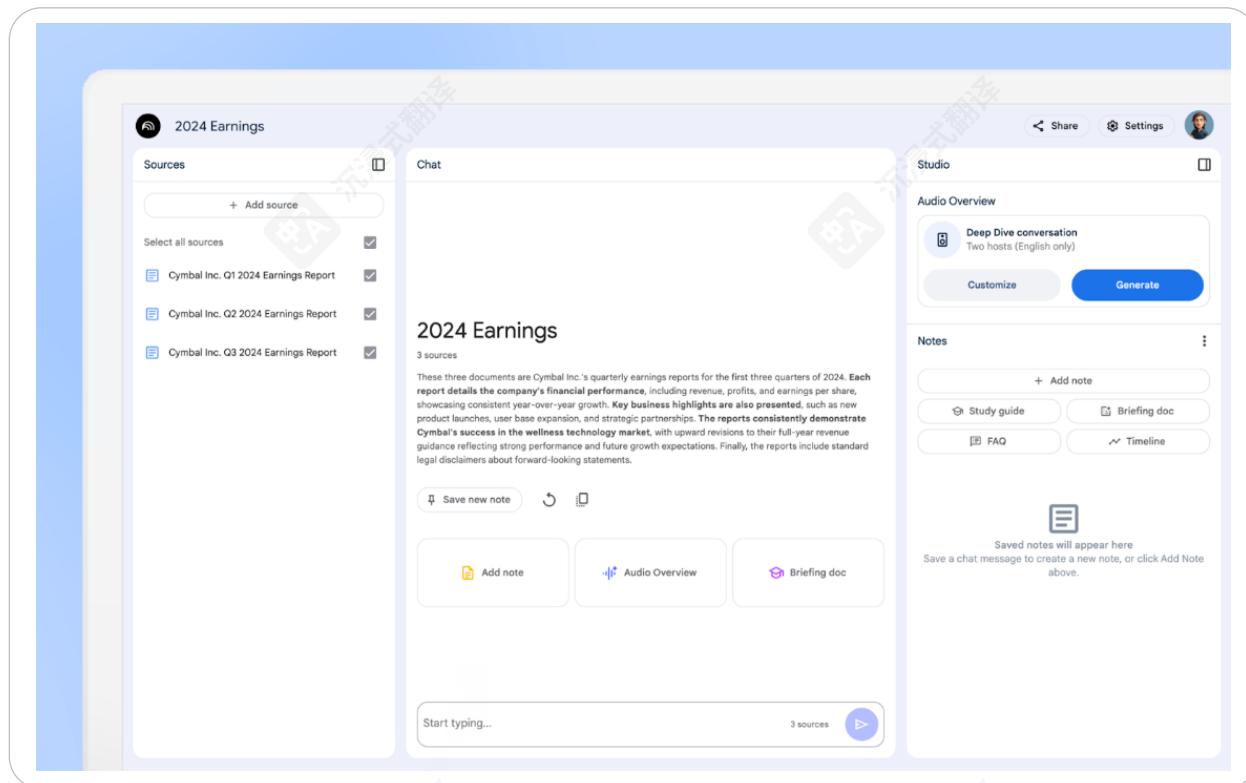


图 11：NotebookLM Enterprise<sup>30</sup> 配置了一些模拟收益报告。

NotebookLM Plus 在此基础上构建，提供了一个高级版本，具有增强的功能和性能。虽然上传资料、提问和生成摘要的核心功能仍然保留，但 NotebookLM Plus 通常会增加如增加资料存储空间等特性，以支持更大、更复杂的项目。它

还可能包括更复杂的 AI 分析工具，例如更细致的摘要选项、增强的问答能力，或更有效地识别不同来源之间的联系和关系的能力。

在 NotebookLM Plus 的基础上，NotebookLM Enterprise<sup>30</sup> 将这些强大的功能带到企业环境中，简化了员工与数据交互和获取洞察的方式。这个企业级版本使用户不仅能够上传和综合信息，还能够发现隐藏的模式并以创新的方式与数据互动。一个典型的例子是 AI 生成的音频摘要功能，该功能通过允许用户“听”他们的研究来增强理解并促进高效的知识吸收。

技术上，NotebookLM，无论是其消费版还是企业版，都采用 LLM 来处理上传的文档、提取关键概念并生成摘要。音频摘要功能使用包含高级韵律控制的文本到语音（TTS）技术，以确保清晰和自然。关键的是，NotebookLM Enterprise 结合了企业级的安全和隐私功能，确保敏感的公司数据得到最谨慎的处理，并按照组织政策进行保护：

## Google AgentSpace Enterprise

Google AgentSpace 为员工提供统一的、公司品牌的多模态搜索代理，作为企业信息的权威来源。凭借 Google 广泛的搜索能力，AgentSpace 提供无与伦比的对话式辅助。员工可以获得复杂问题的答案、主动建议，并从任何来源统一访问信息。此功能扩展到非结构化数据（如文档和电子邮件）和结构化数据（如表格）。集成的翻译功能确保全面理解，无论

信息的原始语言。预构建的连接器用于广泛使用的第三方应用程序，包括 Confluence、Google Drive、Jira、Microsoft SharePoint、ServiceNow 和其他应用程序，使员工能够无缝访问和查询相关数据源，促进更明智的决策。

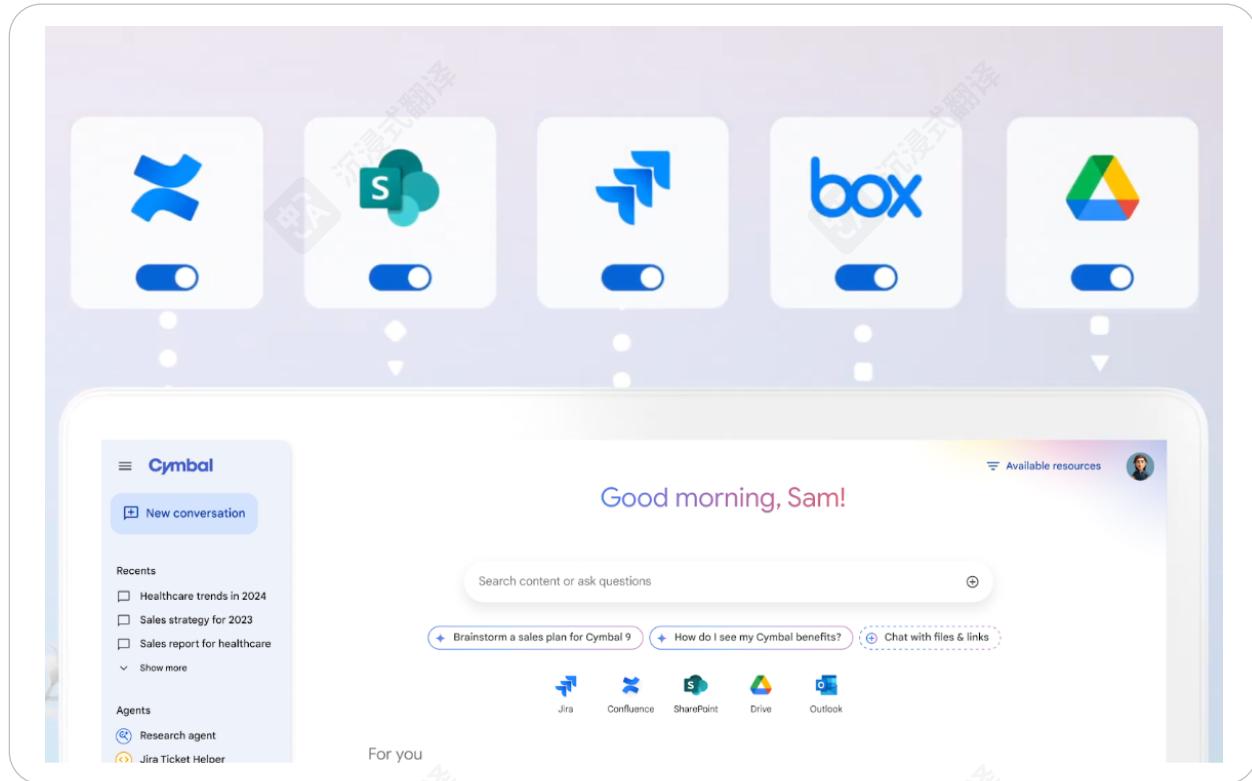


图 12：为 Cymbol 银行定制的 Agentspace<sup>29</sup>，连接到多个数据存储。

不仅仅是信息检索，代理可以在现实世界中采取行动，管理异步任务和工作流程，并帮助员工完成工作。可以根据组织的独特需求配置一组代理，并将有助于深入研究、想法生成和基于绩效的改进、创意资产生成、数据分析等。

Agentspace Enterprise Plus 便于创建针对特定业务功能的定制 AI 代理。该平台支持开发和部署具有上下文感知能力的生成式 AI 代理，赋能各部门（如市场营销、财务、法律、工程等）的员工，使其能够更有效地进行研究、快速生成内容以及自动化重复性任务，包括多步骤工作流程。一个集中的界面简化了这些专用代理的发现和访问，推动了可扩展的 AI 采用。定制代理将连接内部和外部系统及数据，根据公司领域和政策进行定制，甚至可能利用在专有业务数据上训练的机器学习模型。该平台为代理开发、部署和生命周期管理提供了构建者工具。

# 从代理到承包商

今天，定义跨各种工具和平台的 AI 代理的通用接口非常简单，通常包括指定目标、一些文本指令、代理可以使用的工具以及一组示例。虽然这可能足以原型演示，但它会导致潜在的未充分定义的描述，并且可能是 AI 代理难以从原型到生产的主要原因之一。

我们建议将 Agent 接口演变为“遵守合同的代理”，旨在使用 AI 代理解决复杂任务，更具体地说，是在高风险环境中。

## Contracts

承包商的关键思想是规范和标准化请求者和代理之间的合同，使其能够：

1. 尽可能精确地定义结果，类似于我们在现实世界中与承包我们工作的公司协商服务时所做的。这允许代理（承包商）根据期望的结果进行验证，并迭代直到达到期望的目标。
2. 使其能够 协商任务，并澄清和细化它，以避免定义中的任何歧义，并填补对目标共同理解的任何差距。
3. 定义承包商生成解决大问题所需的 新分包合同 的规则（参见下文关于分包合同的部分）。

合同，初始定义		
<b>4.</b>		
字段	描述	必填
任务 / 项目 <b>description</b>	提供我们期望的详细描述 合同方要实现的。它应该尽可能具体和明确。 尽可能不模糊。	Yes
可交付成果 & 规格	Describe precisely the expected outcomes and 合同方任务的可交付成果，包括一份 明确说明构成交付成果的内容 可作为成果接受，并说明如何验证 交付成果是否满足预期。	Yes
范围	明确承包商负责完成的任务范围，并单独详细说明 关于任务的各个方面。也用于说明超出范围的内容 超出范围。	No
预期成本	给出任务完成方面的成本预期。 这通常是任务复杂性的函数 与哪些工具将一起使用。	Yes
预期持续时间	Gives expectation in terms of duration for the task completion.	Yes
输入源	指定哪些输入源可以被使用并被考虑为 有助于完成任务。	No
报告和 反馈	指定反馈循环应如何显示：我们期望多久收到一次进度更新，以 及使用何种机制 / 界面 用于提供反馈（电子邮件、 API 等）。	Yes

表 3：合同的数据模型示例。

合同迭代：反馈与谈判		
字段	描述	必需
未明确说明	突出需要任务发起者澄清的未明确说明的方面，或需要从任务发起者那里获得澄清。	No
成本协商	成本过高，无法完成任务。	No
Risk	突出履行合同中的潜在风险。	No
附加输入 需要的	Express the kinds of additional data or information that would对履行合同有用。	No

Table 4: An example data model for the iteration of messaging between contractors.

# 合同生命周期

以下插图显示了定义、协商和执行合同的周期：

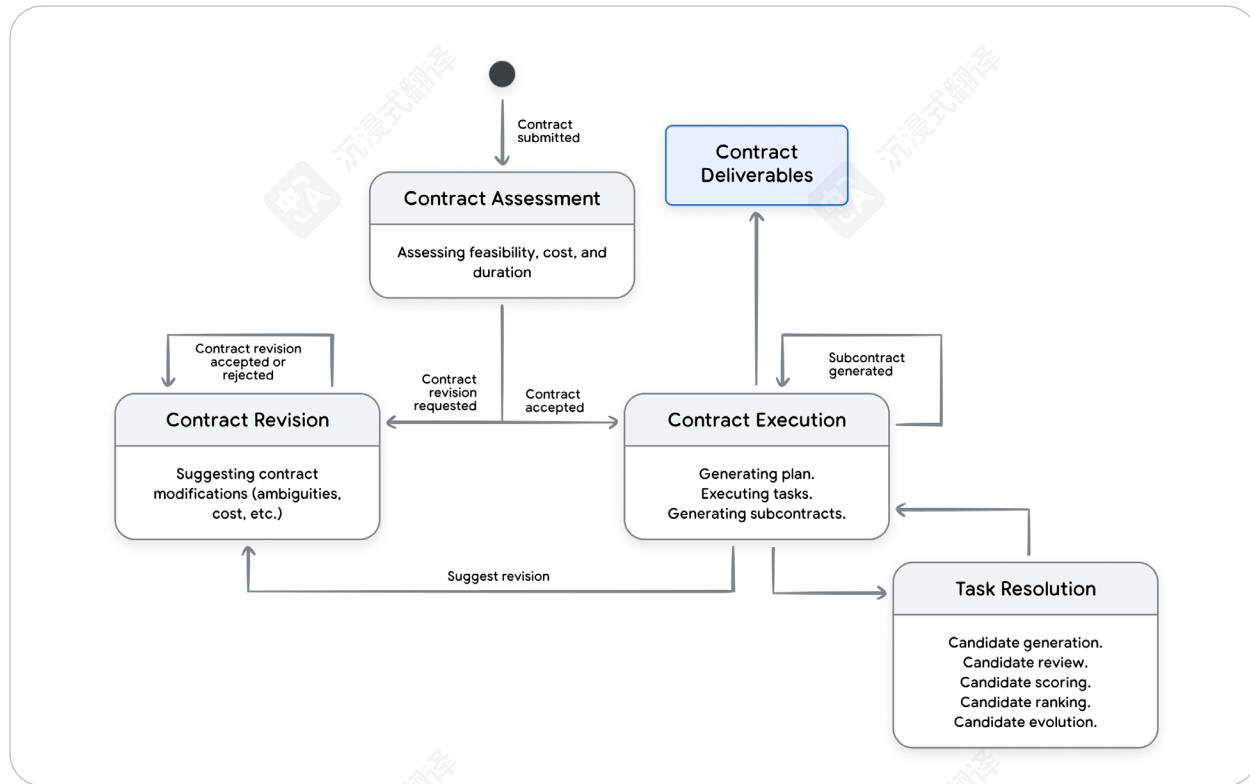


图 13：来自 Agentspace 的代理作为承包商生命周期的图表。

## Contract execution

这要求承包商运行时能够根据其定义的规范履行合同并解决合同背后的任务。优先考虑质量和完整性而不是延迟，能够充分发挥 LLM 的能力，例如

通过生成不同的解决方案，并对其进行审查、评分和进化。这种方法的示例显示在下文中的 Co-Scientist 研究。该引擎可以根据提供的期望迭代和自我验证结果和交付成果，并可以改进和自我纠正，直到验证者满足。能够具体地根据一组客观标准验证解决方案已被证明在人工智能的上下文中效果极佳，并且是成功的人工智能系统（例如 Alpha-Code）的核心。

## 合同谈判

自动化代理上下文中合同的一个核心假设是，在企业世界中许多任务可以通过在不太受约束的方式（延迟和成本方面）使用 LLMs 来获得显著的好处。

能够处理越来越复杂的任务，并使客户能够依赖和信任承包商的结果，将确保公司获得真正的价值。即便如此，我们需要有一个相对优先级的概念，以确保任务得到适当的优先排序，以及资源得到公平分配。因此，我们引入了一个成本概念（通常按客户或合同发起人相对计算），该概念可以在合同发起人和承包商之间讨论和协商，以确保合同获得相对于合同发起人发起的其他合同的适当资源。承包商还可以协商合同的其他方面，例如在规格和交付成果方面（另见下文关于反馈的部分）。

## 合同反馈

合同提供了一个提供反馈的途径，特别是解决模糊不清的问题。随着任务变得越来越复杂，能够尽早提出与任务规范相关的模糊不清或其他问题至关重要。承包商可以在收到合同后立即对合同提供反馈（初始合同评估），然后按照合同中预定义的频率进行反馈。

这些反馈将包含澄清请求，或关于任务欠规范或错规范的反馈（不一致、冲突规范、澄清等）。

## Subcontracts

尽管不是合同定义和规范的直接部分，但通过生成子合同将任务分解为子任务的能力是一个核心概念，将用于驱动承包商的引擎。

当一项任务被认为过于复杂而无法直接处理时，承包商可以选择将任务分解为更小、更容易的任务，这些任务将被添加到执行队列中以进行解决。这仅通过上述合同形式化才成为可能，这使得承包商能够以统一和标准化的方式生成、处理和操作其他合同。

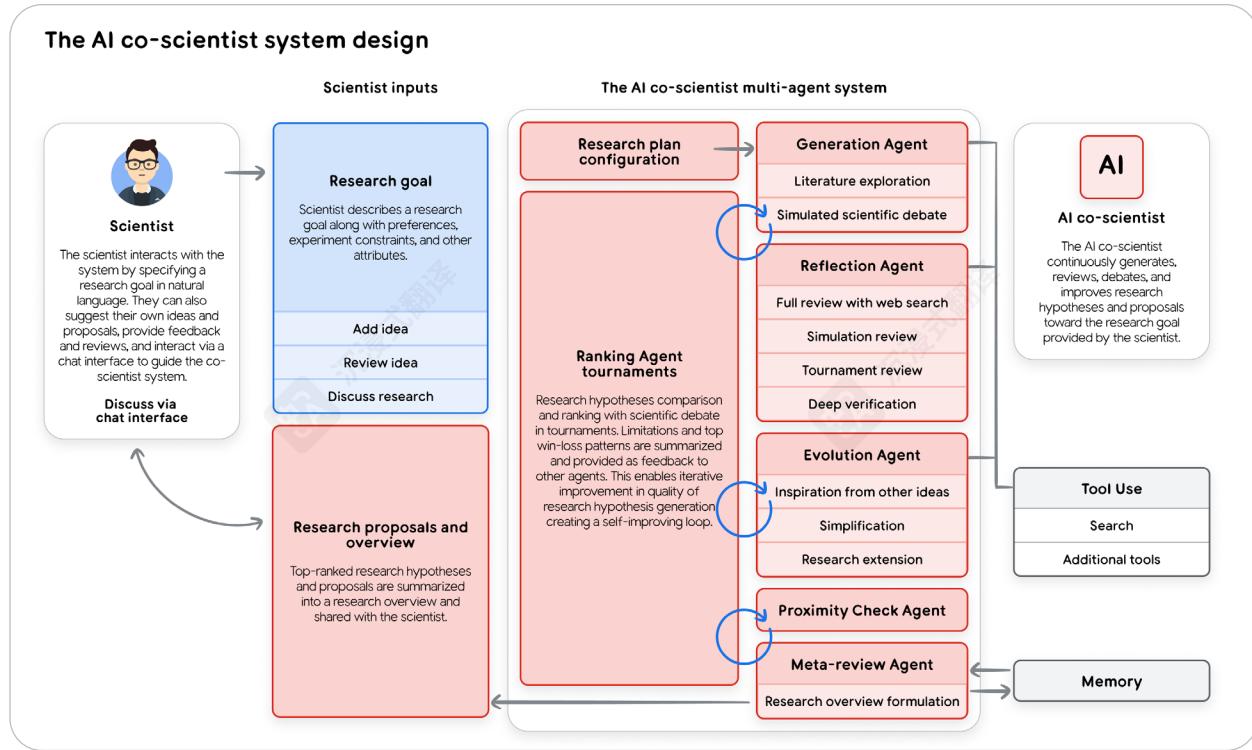
# Google 的联合科学家：一个多智能体智能的案例研究

Google 的 AI 联合科学家是多智能体 LLM 系统应用于科学的研究的典型例子。该系统利用一组专门化的智能体，每个智能体都有其自己的角色和专业知识，以加速科学发现的进程。这些智能体协作生成、评估和改进假设，反映了科学探究的迭代过程。

联合科学家系统采用“生成、辩论和进化”的方法，借鉴了科学方法。这种方法涉及生成多样化的假设，批判性地评估其潜力，并通过持续的反馈和分析进行改进。该系统利用不同 LLM 的优势，每个 LLM 专门研究过程的某个方面，以实现更全面和稳健的结果。

例如，在一项关于肝纤维化治疗的研究中，该协科学家不仅识别了现有药物，还提出了新的机制和有前景的药物候选物，展示了其产生新见解的潜力。其一些主要组成部分是：

- **数据处理代理：**聚合和结构化大量实验数据。
- **假设生成器：**基于现有研究和新发现提出潜在的解释。
- **验证代理：**在向研究人员展示结果之前运行模拟并验证结果。
- **协作代理：**跨不同研究团队沟通发现，增强跨学科合作。

图 14: AI 协同科学家系统的流程图<sup>31</sup>

这种方法展示了多智能体架构如何超越静态 AI 交互，相反促进动态、演进的智能，能够处理研究、企业自动化和知识管理中的复杂任务。

# Automotive AI: Real World Use of Multi-Agent Architecture

为了说明多智能体概念在实际中的体现，让我们考察一个为汽车对话 AI 设计的综合多智能体系统，其中多个专用智能体协作提供直观且无缝的车内体验。

首先，我们将查看一些专用智能体的示例，然后是一些将它们组合成多智能体系统的常见模式。

## 专业代理

让我们首先看看一些专业代理的例子。

## 对话式导航代理

对话式导航代理专门用于帮助用户查找地点、推荐场所，并使用 Google Places 和 Maps 等 API 进行导航。该代理理解导航意图，例如 "为我附近找一个寿司餐厅"，会查询 Places API 获取相关结果，根据用户偏好和历史数据进行重新排序，并将导航指令发送到汽车的内置系统。

### 示例交互:

**用户：**“我需要吃午饭休息一下。我到慕尼黑的路上有哪些好的餐厅？”

**导航代理：**“我在 A9 高速公路上找到了三家评价不错的餐厅。在 25 公里处，有位于英戈尔斯塔特出口的 'Gasthaus Alte Mühle'，评分为 4.5 星 —— 评论者称赞他们的传统巴伐利亚菜肴和啤酒花园。在 45 公里处，服务区的 'La Strada' 评分为 4.1 星，好评如潮，提供快速的意大利餐。或者如果你可以再等 70 公里，Allershausen 附近的 'Zeitlos' 评分为 4.7 星 —— 它稍微高档一些，提供农场到餐桌的美食。你最感兴趣的是哪一个？”

**用户：**“巴伐利亚这个地方听起来不错。那里有方便的停车位吗？离高速公路有多远？”

**导航代理：**"Gasthaus Alte Mühle 有一个专门的停车场，大约有 40 个车位。它距离英戈尔斯塔特北的 A9 出口只有 1.2 公里 —— 大约 2 分钟车程。他们以猪排和自制饺子闻名。中午时分会非常繁忙，但目前入住率适中。您想让我将其设置为您的目的地并为您导航吗？如果我们现在出发，您将在 12:15 左右到达。"

## 对话式媒体搜索代理

对话式媒体搜索代理专门帮助用户查找和播放音乐、有声读物和播客。该代理从本地数据库或流媒体服务中检索媒体，在必要时使用网络搜索以获取额外内容，根据情绪、天气和一天中的时间等上下文因素推荐媒体，并处理语音指令，例如 “为我开车播放一个放松的歌单”。它还能在用户询问 “谁和约翰·柯川（John Coltrane）声音相似？” 等问题时，识别相似艺术家或内容。

### 示例交互:

用户：“播放一些欢快的音乐，帮助我在这条漫长的公路上保持警觉。”

**对话式媒体搜索代理：**“我会播放欢快的驾驶播放列表，帮助你保持活力。从皇后乐队的《Don't Stop Me Now》开始，然后是一些Daft Punk和Stromae的歌曲。我添加了一些符合你偏好的流行欧洲电台歌曲。我已经调整了音量以实现最佳道路意识。如果你想跳过歌曲，请说‘下一首’。”

## 消息组合代理

消息组合代理专门用于帮助用户在驾驶时起草、总结和发送消息或电子邮件。该代理解析语音指令，如“告诉Alex我迟到了”，使用自然语言模型来组合上下文相关的信息，提供可由用户修改的草稿预览，并与各种消息应用程序集成，包括SMS、WhatsApp和电子邮件。

### 示例交互:

用户：“E40上有严重交通拥堵。让我的团队知道我早上例会会迟到20分钟。”

**消息组合代理：**“我已为您的工作团队准备了一条消息：“E40上交通拥堵，我们早上例会大约会迟到20分钟。请先开始，我不参加了。”我应该将此消息发送到您的“工作团队”聊天群，还是您想先编辑它？”

## 汽车手册代理

汽车手册代理专门通过利用检索增强生成（RAG）系统来提供与汽车相关的答案。该代理根据用户查询从汽车手册中检索相关部分，使用LLM对信息进行总结和上下文化，并在适当的时候将用户链接到详细文档或教学视频。

### 示例交互：

**用户：**“我该如何关闭这款大众汽车的车道保持辅助功能？”

**汽车手册代理：**“要关闭您的大众汽车的车道保持辅助功能，请按下方向盘左侧带有汽车和车道图标的按钮。当它被关闭时，驾驶员显示器会显示确认信息。您也可以在娱乐信息系统中永久禁用它，路径为设置 > 驾驶辅助 > 车道保持。您希望我为您逐步指导设置菜单吗？”

## 通用知识代理

通用知识代理专门用于回答关于世界、历史、科学、文化和其他一般话题的事实性问题。该代理访问广泛的知识库来回答事实性问题，提供有关人物的人物传记信息，对概念和现象提供背景解释，在坚持事实信息的同时避免幻觉，并通过保持上下文意识来回答后续问题。

### 示例交互：

**用户：**“告诉我一个关于萨尔茨堡的有趣事实。”

**通用知识代理：**“萨尔茨堡，奥地利是历史最著名的作曲家莫扎特的出生地。除了这个众所周知的事实，萨尔茨堡的老城区是世界文化遗产，1964年，深受喜爱的音乐剧《音乐之声》的部分场景在那里拍摄。萨尔茨堡音乐节是世界最著名的音乐和戏剧节之一，自1920年起每年在那里举行。”

## Patterns in Use

让我们看看结合这些代理的一些常见方法。

### Hierarchical Pattern

一个中央协调器代理类对查询进行分类并将它们路由到专门的代理。例如，当用户询问“为我找到附近的寿司餐厅”时，协调器将其识别为导航请求并将其路由到导航代理，该代理处理位置搜索和与地图API的交互。

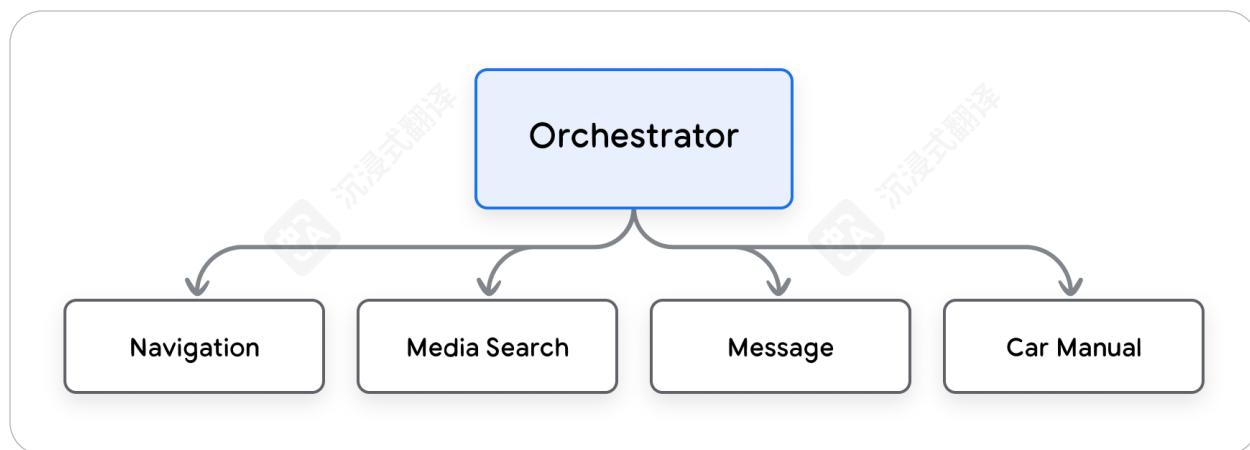


图 15. Hierarchical pattern.

编排代理专门用于将查询路由到最合适的专用代理，并管理整体对话流程。该代理分析用户查询以确定领域和意图，将请求导向最合适的专用代理，在多轮对话中保持对话上下文，并在需要时管理回退策略。

specialized agents cannot provide satisfactory responses.

#### **Example Interaction:**

##### **第一回合：**

**用户：**“前方有提供干净卫生间的服务区吗？”

**协调代理：** [内部路由至导航代理]

**导航代理：**“前方 E19 公路 20 公里处有一个服务区。它是 Aire de Waremme，去年进行了翻新，设施和清洁度评价都很高。那里还有一家 Paul 面包店和快餐店。”

##### **第二回合：**

**用户：**“我的目的地今晚会下雨吗？”

**编排代理：** [识别天气查询并路由到天气信息代理]

**天气信息代理：**“在您的目的地布鲁塞尔，预报显示今晚有 60% 的几率出现小雨，大约从 19:00 开始。气温将在 14°C 左右。如果您计划户外活动，建议带伞。”

## Diamond Pattern

在这个层次模式的变化中，来自专业代理的响应在到达用户之前会通过一个中央审核代理。例如，导航代理生成关于附近餐厅的事实性响应，然后通过一个重述代理，该代理根据用户偏好调整语气和风格，然后再进行文本到语音合成。

重述代理专门用于调整响应的语气、风格和呈现方式，以匹配用户偏好和上下文。该代理将技术信息转换为对话语言，根据驾驶条件和紧急程度调整响应长度，根据用户偏好个性化沟通风格，并格式化复杂信息以便于驾驶时理解。

### 示例交互：

**[来自汽车手册代理的原始响应，在重新措辞之前]**：“在轮胎压力监测系统警报的情况下，车辆检测到一个或多个轮胎的压力已低于推荐的 2.2 巴充气压力。受影响的轮胎压力约为 1.8 巴，比推荐水平低 18.75%。这并不构成立即的安全隐患，但长时间驾驶低压轮胎会导致燃油效率降低、轮胎寿命缩短，并可能影响操控性能。建议您尽快前往服务站将轮胎重新充气至规定标准。”

**重新措辞代理：[适应驾驶环境，调整信息密度和紧迫性]**

**最终对用户的响应：**“您的右前轮胎有点低——降至 1.8 巴。可以继续驾驶，但您想在下一个加油站补充一些空气。低压轮胎会影响您的燃油经济性并使轮胎更快磨损。如果您需要，我可以帮您找到附近的充气地点。”

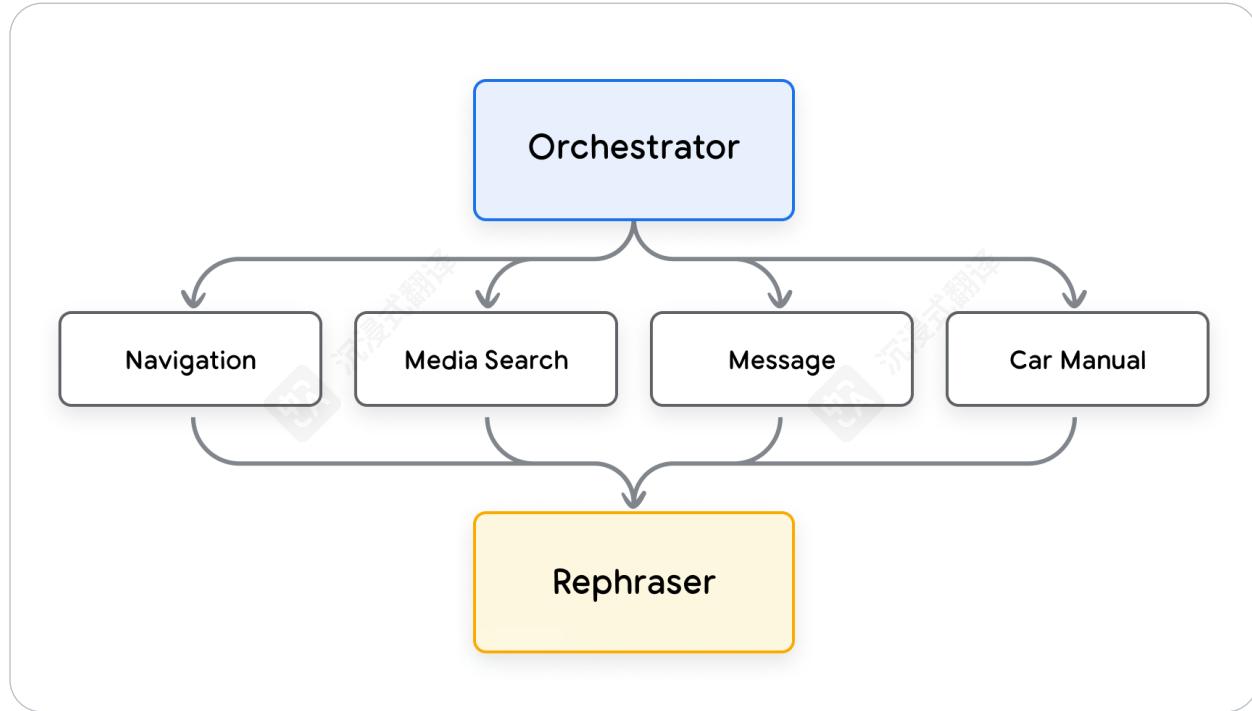


图 16. 菱形图案。

### 示例转换：

**初始响应（事实性 & 中性）：** "我发现了一家评分很高的寿司餐厅。它叫做 Sakura Sushi，位于 123 Main Street。它有 4.7 星评分，营业到晚上 10 点。您想让我开始导航吗？"

**重新措辞（俏皮风格）：** "想吃寿司？不用再说了！前往 123 主街的樱之寿司，那里的鱼新鲜，酱油畅流！评分 4.7 星，营业至晚上 10 点。准备好开始一场鲜味冒险了吗？"

## 点对点

当代理检测到编排出现路由错误时，可以将查询传递给彼此。这创建了一个更具弹性的系统，可以从初始的误分类中恢复。

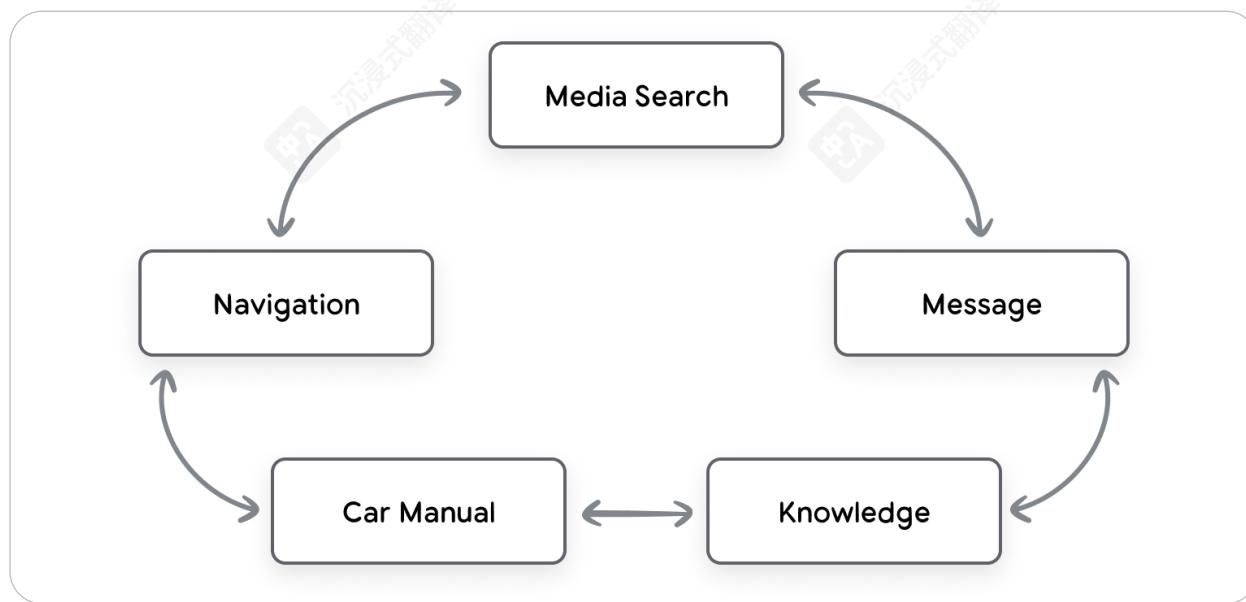


图 17。点对点。

### 示例场景：

1. 用户询问：“附近有什么寿司店？”
2. Orchestrator 正确地将此路由到对话式导航代理，该代理提供附近寿司餐厅的信息。
3. 用户跟进：“纽约中央公园有多大？”
4. Orchestrator 可能最初将此路由到对话式导航代理（基于先前的导航相关对话）。

5. 但是，对话式导航代理将其识别为一般知识问题，而不是导航请求，并将其转交给一般知识代理，后者可以提供有关中央公园面积的事实信息。

#### 与集中式编排相比，点对点转交的优势：

- 1. 对错误分类的恢复能力：**即使中央编排器在路由时出错，专业代理也可以识别出查询是否超出其领域，并适当地重定向。
- 2. 领域专业知识路由：**专业代理通常对其自身领域的边界有更好的理解。媒体搜索代理确切地知道它可以比通用协调器更好地处理哪些与音乐相关的查询。
- 3. 减少编排复杂性：**中央协调器不需要在初始路由中达到完美准确，从而减少了其决策逻辑的复杂性。

## Collaborative Pattern

协作模式涉及多个代理协同处理同一任务的互补方面，并有一个响应混合代理将来自不同代理的元素组合起来以创建一个全面的答案。这种方法认识到不同的代理根据其专业领域的知识为完整解决方案贡献了宝贵的部分。当出现以下情况时，该模式特别有价值：

1. 查询的不同方面需要不同类型的专业知识（例如，技术规格、实用建议和概念解释）
2. 没有单个代理拥有完整的信息来充分满足用户的需求
3. 用户可以从同一问题中获得多个视角

#### 4. 不同的专业代理可以访问不同的知识库或推理能力

与竞争性方法不同，其中响应相互竞争，协作模式假设来自不同代理的响应是互补的而不是冗余的。响应混合代理从每个来源识别最有价值的信息，并将其综合成一个连贯的答案，该答案利用了每个专家的独特优势。

例如，当被问及如何处理水滑时，汽车手册代理贡献了车辆特定的安全系统信息，驾驶技巧代理提供了实用的驾驶技巧，而一般知识代理解释了现象背后的物理学。一起，他们创建了一个比任何单个代理单独提供都更完整和有用的响应。

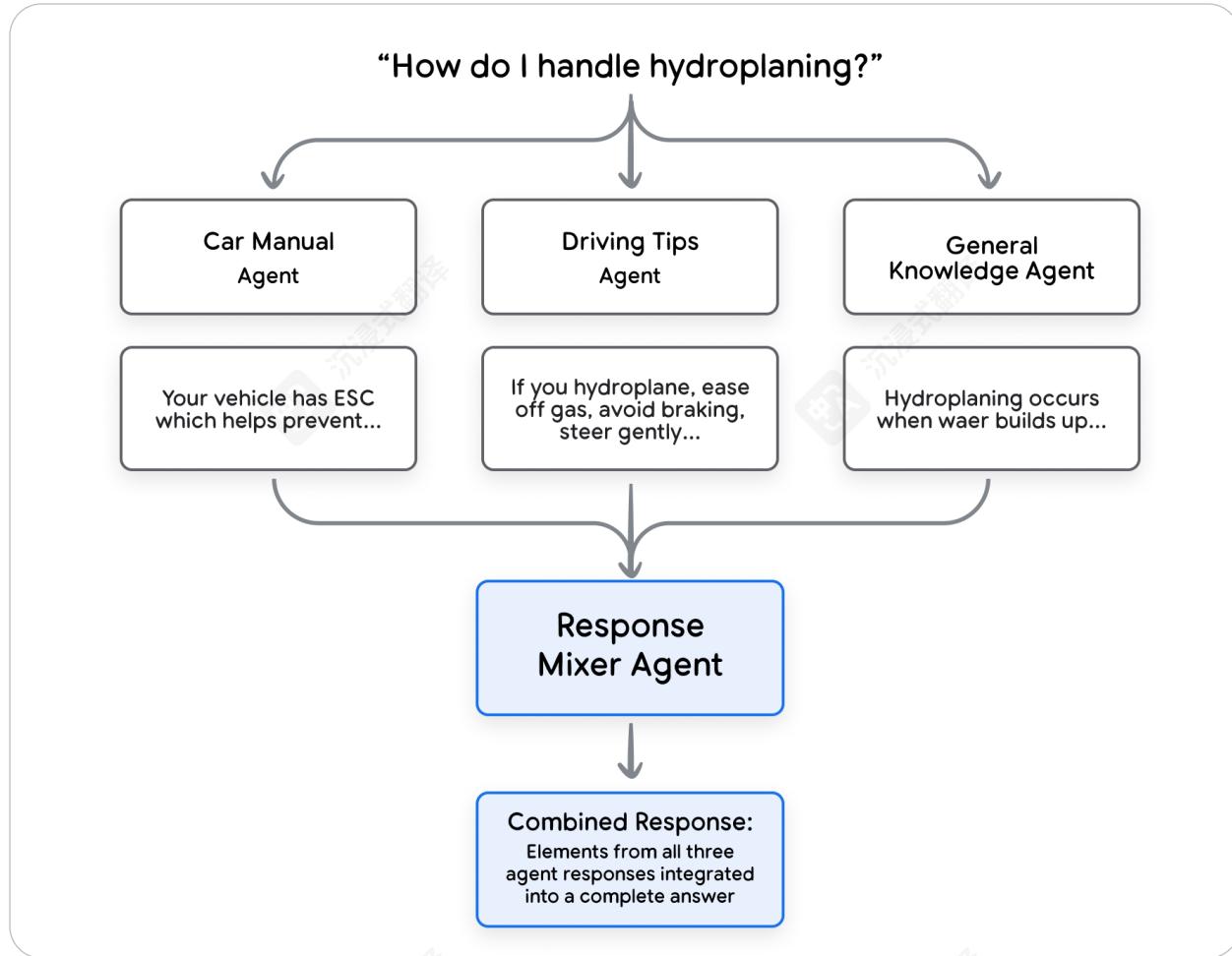


图 18. 协作模式。

## 响应混合代理

响应混合器专门用于从多个代理中选择最佳响应，并在需要时将它们合并。该代理根据答案的准确性和相关性进行评估，删除不正确的信息，并结合来自不同响应的有用部分，为用户提供最有帮助的答案。

### 示例交互：

**用户：“我该如何处理水滑现象？”**

**[幕后，多个代理生成响应:]**

**汽车手册代理响应：**“您的奥迪配备了电子稳定控制系统（ESC）和防抱死制动系统（ABS），这些系统通过保持牵引控制来帮助防止水滑现象。” [置信度：71%]

**通用知识代理响应：**“水滑现象是指当轮胎因轮胎与路面之间的水积聚而失去与路面接触时发生的。” [置信度：65%]

**安全提示代理响应：**“如果您遇到水滑现象，请松开油门，避免突然制动或转向，并轻轻转向您想要去的方向，直到您恢复牵引力。” [置信度：94%]

**响应混合代理：** [选择最相关部分并将它们合并成一个完整的答案]

**对用户的最终回复：**“如果您遇到水滑现象，即轮胎因水膜积聚而失去与路面的接触，请松开油门，避免突然刹车或转向，并轻轻朝您想去的方向转向，直到恢复抓地力。您的奥迪电子稳定控制系统和ABS系统将帮助维持控制，但正确的驾驶技术仍然至关重要。”

这种竞争性方法确保关键安全信息不会因路由错误而遗漏，并且即使用户的查询有多个有效的解释，用户也会收到最有帮助的响应。

## Adaptive Loop Pattern

自适应循环模式涉及通过重复尝试进行迭代改进，逐步提高结果，直到它们满足所需标准。

对话式导航代理可以实现循环模式以迭代改进  
当初始查询未产生令人满意的成果时，搜索结果。

### 示例交互：

1. 用户询问：“找一个提供素食选项的意大利餐厅”
2. 导航代理使用这些参数查询 Places API
3. 代理评估结果并发现没有餐厅符合所有标准
4. 取而代之的是返回“无结果”，代理会自动重新表述查询：
  - 第一轮：搜索“提供素食选项的意大利餐厅”
  - 第二轮：扩展为“意大利餐厅”，然后筛选提及植物性选项的餐厅
  - 第三轮：如果仍然未成功，则搜索“纯素食餐厅”，并筛选出具有意大利风味的菜系
5. 代理会从这些逐步搜索中呈现最佳匹配结果，并解释它们如何与原始请求相关

这种循环行为能够实现更强大的搜索能力，以适应可用性和上下文，即使在无法找到精确匹配时也能提供有用结果。

## 多智能体架构在汽车 AI 中的优势

多智能体系统通过将复杂任务分解为专门角色，为汽车 AI 带来了强大的优势。这就像组建一个专家团队，而不是依赖一个通才。

在这种方法中，每个代理都专注于它最擅长的事情。导航代理专注于寻找位置和规划路线。媒体搜索代理成为音乐和播客的专家。汽车手册代理专门研究车辆功能和故障排除。通过专注于特定领域，每个代理都在其领域内发展更深入的能力。

这种专业化使整个系统更高效。当代理处理更狭窄的任务集时，优化其性能就更容易。结果是更高质量的响应更快地提供，并且计算成本更低。性能提高是因为系统可以将正确的资源与每个任务匹配。简单请求使用最小的处理能力，而复杂问题仅在需要时才会调用更强大的资源。

在车辆中，速度很重要，而多代理系统在关键地方发挥作用。像调整气候控制或打开窗户这样的关键功能由快速、设备上的代理运行以立即响应。同时，像寻找餐厅推荐这样的不太紧急的任务可以使用具有更广泛知识的基于云的代理。这种分离确保了无论系统在做什么，基本的车辆控制都能保持响应。

此设计还创建了自然恢复能力。如果互联网连接中断，设备上运行的必要功能将继续工作。您可能会暂时失去餐厅推荐，但气候控制和基本媒体播放功能仍然完美运行。

## Agent Builder

**Vertex AI Agent Builder** 是为开发者提供的产品和服务集合。我们为您搭建了一个全面的平台，用于构建和连接代理。来自 Google Cloud 的工程卓越性和安全性，来自 Google Deepmind 的 AI 研究，以及 AgentOps 的最佳实践。Google Cloud 基于此平台构建自己的代理，现在您也可以。期待 2025 年针对代理开发者的更多令人兴奋的公告。

**Vertex AI Agent Engine** 简化开发，依赖于 Google 工程管理的与流行开源代理库的集成。它提供管理的自动扩展运行时和代理将需要的许多服务（例如：会话、示例、跟踪、evals）。这是以极低的努力和极高的价值方式安全部署您在任何框架中编写的代理。

**Vertex AI Eval Service** 提供了本文白皮书中讨论的所有评估工具，以及更多。LLMs、RAG 和 Agent evals 通过 Vertex AI Eval Service 进行稳定和可扩展，并方便地集成到监控和实验产品中。

大量代理工具，更多即将推出：

- 通过 Vertex AI Search<sup>26</sup> 或 RAG Engine<sup>28</sup> 进行检索。
- 通过 Gen AI Toolbox for Databases<sup>32</sup> 从数据库进行非搜索检索。

- 应用集成<sup>33</sup> 与数百个 API 支持完整的 ACLs
- 使用 Apigee Hub<sup>34</sup> 将任何 API 转换为可管理的、企业级工具

当然还有最适合代理的最佳 LLMs，可以访问 Vertex AI Model Garden<sup>35</sup>，以及 Gemini 系列模型<sup>36</sup>，它们将驱动代理时代。

## Summary

这份白皮书（我们早期关于智能体的白皮书的配套文件）探讨了生成式 AI 智能体的快速演变领域，从其基本架构到高级评估技术和多智能体系统的变革潜力。

开发者关键要点：

1. **智能体运维至关重要：** 建设成功的智能体远不止初始概念验证。拥抱智能体运维原则，整合 DevOps 和 MLOps 的最佳实践，但也专注于智能体特定的元素，如工具管理、编排、内存和任务分解。
2. 指标驱动改进：从业务级 KPI（如目标完成、用户参与度或收入）作为你的“北极星”。然后，为你的智能体配置以跟踪与关键任务、用户交互和智能体行为（跟踪）相关的细粒度指标。人类反馈（调查、用户调查）非常有价值。
3. **自动化评估至关重要：** 不要仅依赖手动测试。实施自动化评估框架，评估智能体能力、轨迹（采取的步骤）和最终响应。利用精确匹配、顺序匹配、精确率 / 召回率等技术进行轨迹评估，以及 autoraters（LLM 作为评判者）进行最终响应质量评估。

4. **人在回路至关重要：**自动指标很强大，但人工评估提供了必要的上下文，特别是对于创造力、常识和细微差别等主观方面。使用人工反馈来校准和验证你的自动评估方法。不要外包领域知识。
5. **多智能体系统提供优势：**对于复杂任务，考虑多智能体架构。它们可以提高准确性、效率、可扩展性和容错性。了解不同的设计模式（顺序、层次、协作、竞争），并为你的应用程序选择正确的模式。
6. **智能体 RAG 提高相关性：**通过结合主动改进搜索查询、评估检索信息并适应不断发展的知识的智能体，超越传统的 RAG。这导致了更准确和上下文相关的响应。
7. **搜索优化是 RAG 的基础：**在深入复杂的 agentic RAG 之前，优化您的底层搜索引擎。语义分块、元数据丰富、微调嵌入模型和使用排序器等技术可以显著提高检索质量。
8. **代理和工具注册表很重要：**随着您使用的代理或工具数量的增加，一个用于管理功能、本体和性能的注册表变得至关重要。
9. **安全是首要考虑：**在部署代理时，尤其是在企业内部，优先考虑安全。利用像 Google Agentspace 这样的平台，它们提供内置的安全功能，如 RBAC、VPC 服务控制和使用者访问管理 (IAM) 集成。
10. **高效利用开发者周期：**经典的构建与购买设计选择仍然摆在眼前，因为生成式 AI 代理的行业正在迅速发展。考虑平台和产品作为从头开始构建一切的替代方案。这将缓冲快速变化行业的部分波动，并让您能够专注于您的数据、领域和用户。

11. **企业中的代理：**代理正在改变我们的工作方式，使我们变得更加高效，并实现自动化。知识工作者将越来越多地管理代理队，并出现新的用户体验。Google Agentspace 是一个强大的工具，允许在公司的数据和工作流程之上部署企业搜索、AI 和 AI 代理。

**代理研究和开发的方向：**AI 代理领域正在经历快速发展。正在进行研究和开发的关键领域包括：

- **高级评估方法：**开发更健壮和可扩展的评估技术，包括基于过程的评估（侧重于推理）、AI 辅助评估和标准化基准。
- **多智能体协调：**改进多智能体系统内部的协调和通信机制，以实现更有效的协作、任务处理和推理。
- **现实世界适应：**创建能够在动态、不可预测的现实世界环境中适应和学习的智能体。汽车 AI 等生产系统说明了智能体如何在关键功能上的设备性能和复杂任务上的云能力之间取得平衡，并经常适应不断变化的连接条件。
- **可解释性和可解释性：**使智能体行为更加透明和易于理解，允许开发者和用户更深入地了解其决策过程。
- **长期记忆和学习：**开发更复杂的记忆机制，使智能体能够在较长时间内保留和利用信息，从而实现持续学习和适应。

- **代理通信协议：**更好地定义代理如何共享任务、知识和消息，特别是在跨远程系统时，这些系统是不透明的。
- **从代理到承包商：**为了使代理达到更高的可靠性和实用性，我们需要提升任务的定义，将其转化为具有明确交付成果、验证机制和协商模糊性的合同，类似于我们从其他公司承包工作的方式。

#### 行动号召：

AI 的未来是具有自主性的。我们鼓励开发者拥抱这些概念，并开始构建下一代智能应用。开始尝试本白皮书讨论的工具和技术。探索可用的资源，例如 Google Agentspace、NotebookLM Enterprise、Vertex Eval Service、Cloud Observability 和 Vertex AI Search，以加速您的开发过程。深入研究提供的代码示例、教程和文档，以获得实践经验。构建、评估、迭代，并为不断增长的代理开发者社区做出贡献。可能性是无限的，现在是构建的时候了！具体来说，从参考资料中的代码和 Colab 笔记本开始。

## 尾注

1. Shafran, I., Cao, Y. et al., 2022, 'ReAct: Synergizing Reasoning and Acting in Language Models'. Available at: <https://arxiv.org/abs/2210.03629>.
2. Wei, J., Wang, X. et al., 2023, 'Chain-of-Thought Prompting Elicits Reasoning in Large Language Models'. Available at: <https://arxiv.org/pdf/2201.11903.pdf>.
3. Wang, X. et al., 2022, 'Self-Consistency Improves Chain of Thought Reasoning in Language Models'. Available at: <https://arxiv.org/abs/2203.11171>.
4. Diao, S. et al., 2023, 'Active Prompting with Chain-of-Thought for Large Language Models'. Available at: <https://arxiv.org/pdf/2302.12246.pdf>.
5. Zhang, H. et al., 2023, 'Multimodal Chain-of-Thought Reasoning in Language Models'. Available at: <https://arxiv.org/abs/2302.00923>.
6. Yao, S. et al., 2023, 'Tree of Thoughts: Deliberate Problem Solving with Large Language Models'. Available at: <https://arxiv.org/abs/2305.10601>.
7. Long, X., 2023, 'Large Language Model Guided Tree-of-Thought'. Available at: <https://arxiv.org/abs/2305.08291>.
8. Google. 'Google Gemini 应用'. 可在以下地址获取: <http://gemini.google.com>.
9. Swagger. 'OpenAPI 规范'. 可在以下地址获取: <https://swagger.io/specification/>.
10. Xie, M., 2022, '如何工作上下文学习？理解与传统监督学习的差异的框架'. 可在以下地址获取: <https://ai.stanford.edu/blog/understanding-incontext/>.

---

11. Google Research. 'ScaNN ( 可扩展最近邻 )' 可在以下地址获取: <https://github.com/google-research/google-research/tree/master/scann>.

---

12. LangChain. 'LangChain'. 可在以下地址获取: <https://python.langchain.com/v0.2/docs/introduction/>.

---

13. Sokratis Kartakis, 2024, 'GenAI 在生产环境中：MLOps 还是 GenAIOps？'. 可在于: <https://medium.com/google-cloud/genai-in-production-mlops-or-genaiops-25691c9becd0>.

---

14. Sokratis Kartakis, 2024 'Gen AI Ops，实现生成式 AI，实用指南'. 可于: <https://medium.com/google-cloud/genaiops-operationalize-generative-ai-a-practical-guide-d5bedaa59d78>.

15. 云跟踪概述。可在: <https://cloud.google.com/trace/docs/overview>。
16. 伯克利函数调用排行榜 (BFCL)。可在:  
[https://gorilla.cs.berkeley.edu/blogs/8\\_berkeley\\_function\\_calling\\_leaderboard.html](https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html)。
17. 卡尔蒂克 · 纳拉辛汉等, 2024 年, 'τ-bench'. 可在 <https://arxiv.org/abs/2406.12045>。
18. 卡尔蒂克 · 瓦尔梅卡姆等, 2023 年, 'PlanBench'. 可在: <https://arxiv.org/abs/2206.10498>。
19. Xiao Liu, et al., 2023, 'AgentBench'. Available at: <https://arxiv.org/abs/2308.03688>.
20. Martin Iglesias, et al., 2025, 'DABStep' Available at: <https://huggingface.co/spaces/adyen/DABstep>.
21. LangSmith 平台用于代理可观察性。Available at:  
<https://docs.smith.langchain.com/evaluation/concepts#agents>。
22. Mingchen Zhuge, et al., 2024, 'Agent-as-a-Judge: Evaluate Agents with Agents'. Available at: <https://arxiv.org/abs/2410.10934>.
23. LangGraph 的多代理文档。Available at:  
[https://langchain-ai.github.io/langgraph/concepts/multi\\_agent/](https://langchain-ai.github.io/langgraph/concepts/multi_agent/)。
24. LangChain 博客 2024, 'Multi-agent workflows'. Available at:  
<https://blog.langchain.dev/langgraph-multi-agent-workflows/>。
25. Vectorize 博客 2024, 'How I finally got agentic RAG to work right'. Available at: <https://vectorize.io/how-i-finally-got-agentic-rag-to-work-right/>。
26. Vertex AI Search, product documentation. Available at: <https://cloud.google.com/enterprise-search>.
27. Vertex AI Search Builder APIs, product documentation。Available at:  
<https://cloud.google.com/generative-ai-app-builder/docs/builder-apis>。
28. Vertex AI RAG Engine, product documentation。Available at:  
<https://cloud.google.com/vertex-ai/generative-ai/docs/rag-overview>。
29. Agentspace product documentation。Available at:  
<https://cloud.google.com/agentspace/agentspace-enterprise/docs/overview>。
30. NotebookLM Enterprise product documentation。Available at:  
<https://cloud.google.com/agentspace/notebooklm-enterprise/docs/overview>。

31. Juraj Gottweis, et al., 2025, '使用 AI 科学家加速科学突破'. 可在:

[https://research.google/blog/accelerating-scientific-breakthroughs-with-an-ai-co-scientist/.](https://research.google/blog/accelerating-scientific-breakthroughs-with-an-ai-co-scientist/)

32. Hamsa Buvaraghian, et al. 2025, '宣布数据库 Gen AI 工具箱的公共 Beta 版'. 可在:

[https://cloud.google.com/blog/products/ai-machine-learning/announcing-gen-ai-toolbox-for-databases-get-started-today?e=48754805.](https://cloud.google.com/blog/products/ai-machine-learning/announcing-gen-ai-toolbox-for-databases-get-started-today?e=48754805)

33. Google Cloud 集成连接器, 产品文档 . 可在:

[https://cloud.google.com/integration-connectors/docs.](https://cloud.google.com/integration-connectors/docs)

34. Apigee API Hub, 产品文档 . 可在:

[https://cloud.google.com/apigee/docs/apihub/what-is-api-hub.](https://cloud.google.com/apigee/docs/apihub/what-is-api-hub)

35. Vertex AI 模型花园, 产品文档。可在:

[https://cloud.google.com/model-garden.](https://cloud.google.com/model-garden)

36. Gemini 系列大语言模型, 产品文档。可在:

[https://cloud.google.com/vertex-ai/generative-ai/docs/learn/models#gemini-models.](https://cloud.google.com/vertex-ai/generative-ai/docs/learn/models#gemini-models)

37. 使用 Vertex Eval 服务开始评估代理。可在:

[https://cloud.google.com/vertex-ai/generative-ai/docs/models/evaluation-agents.](https://cloud.google.com/vertex-ai/generative-ai/docs/models/evaluation-agents)

38. Irina Sigler, Ivan Nardini. 2025 年 1 月 '在 Vertex AI 中介绍代理评估'. 可在:

[https://cloud.google.com/blog/products/ai-machine-learning/introducing-agent-evaluation-in-vertex-ai-gen-ai-evaluation-service?e=48754805.](https://cloud.google.com/blog/products/ai-machine-learning/introducing-agent-evaluation-in-vertex-ai-gen-ai-evaluation-service?e=48754805)

39. 查看 LangGraph、CrewAI 和 LangChain 的示例代理评估笔记本。可在:

[https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/evaluation/.](https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/evaluation/)

40. 查看许多示例代理, 主要为初学者和中级水平。可在:

[https://github.com/GoogleCloudPlatform/generative-ai/.](https://github.com/GoogleCloudPlatform/generative-ai/)

41. 浏览多个样本代理, 中级和高级水平。可在:

[https://github.com/GoogleCloudPlatform/applied-ai-engineering-samples.](https://github.com/GoogleCloudPlatform/applied-ai-engineering-samples)