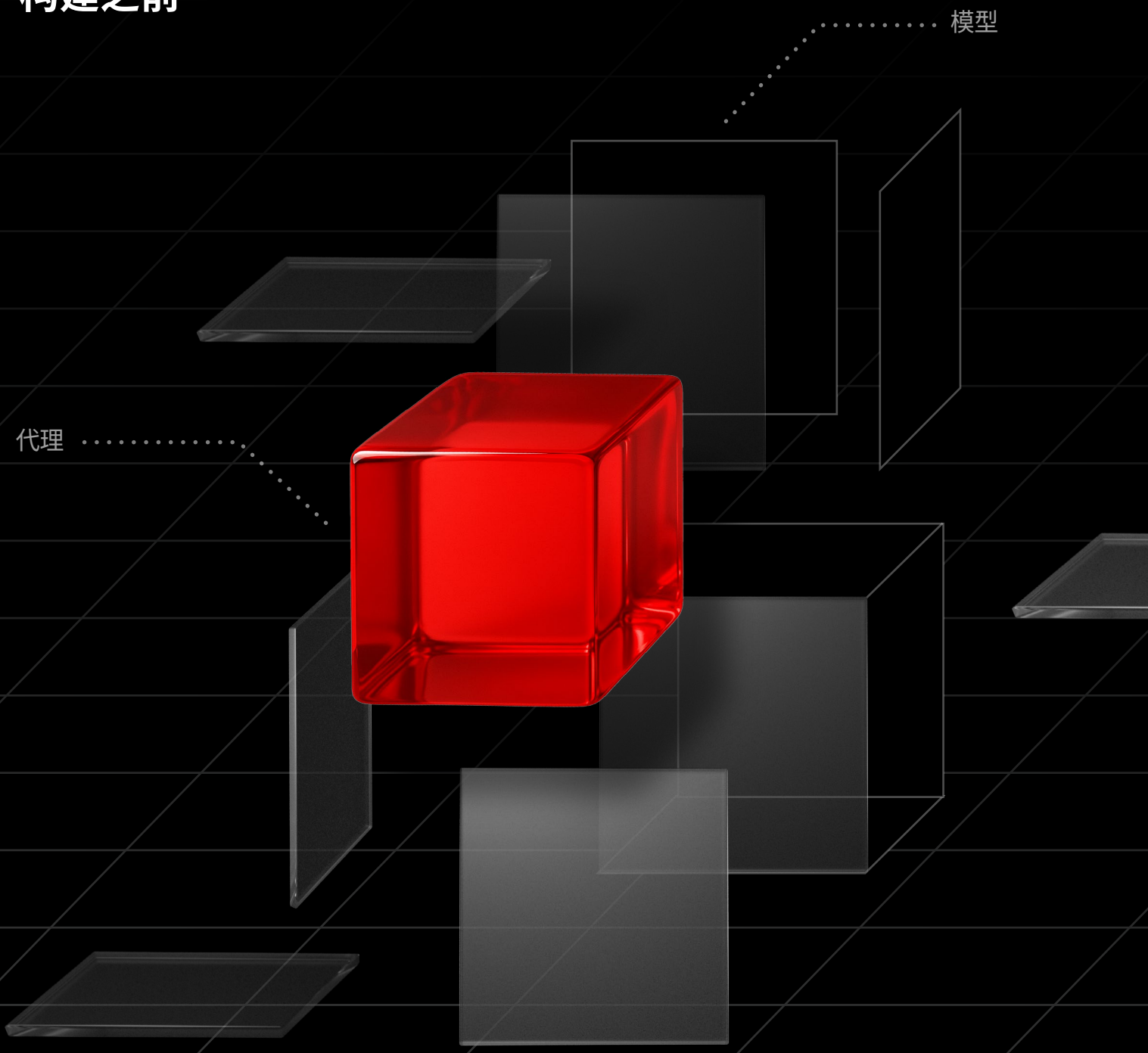
在着手
构建之前



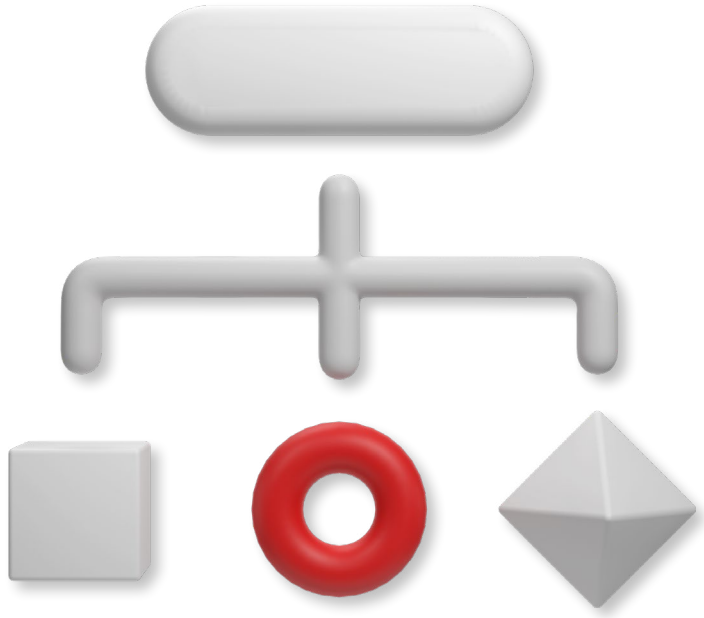
了解如何利用红帽 AI 构建代理式 AI 系统

Red Hat

目录

简介

代理式 AI 的技术演变

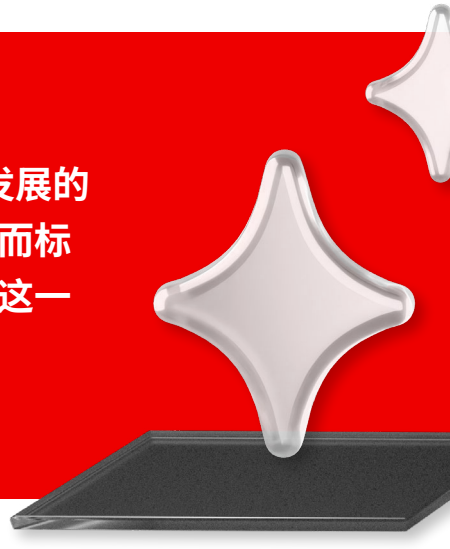


代理式 AI 正在重塑企业对自动化、决策制定和可扩展性问题的看法。但是，构建真实代理并非像提示聊天机器人那样简单。生产就绪型代理式系统需要的不仅仅是语言模型，更需要统一的架构来协调推理、编排工具、维护记忆、保障数据安全和管控行为。

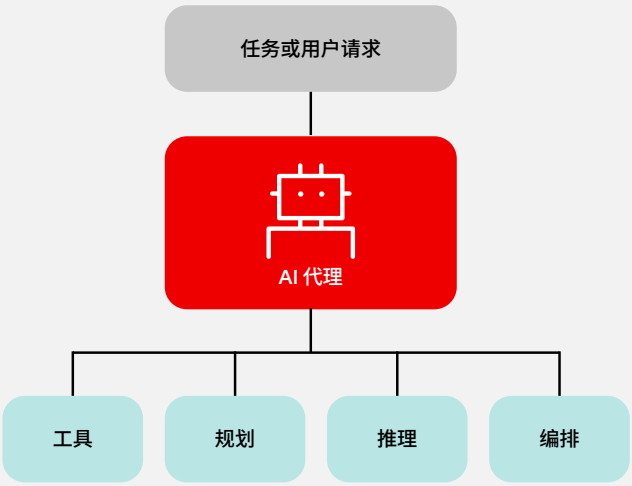
本电子书基于红帽开源方法，从技术角度深入剖析该架构。从推理和工具编排，到安全性和可观测性，每个构建模块都必须模块化、注重安全性并具备生产就绪能力。这正是红帽® AI 的价值所在。红帽 AI 基于红帽 OpenShift® AI 构建并由 Llama Stack 提供支持，为大规模构建、部署和管理智能代理提供了基础架构。

借助红帽 AI，企业组织可标准化代理式工作流的构建与扩展方式，无需再应对分散的框架和不一致的工具链。通过采用模型上下文协议（MCP）等开放标准，红帽助力统一代理发现和使用工具的方式。通过集成安全防护、评估和基础架构自动化，红帽帮助团队以更高的可重复性和可信度，从概念验证阶段无缝过渡到生产阶段。

这是企业 AI 发展的下一个阶段，而标准化正是实现这一切的基础。



代理式 AI 系统的组成部分



- **工具使用：**使用外部工具收集数据并执行任务。
- **规划和执行：**自主制定并执行多步骤计划，以达成目标。
- **推理：**运用逻辑和对上下文的理解来做出明智决策。
- **编排：**协调行动、工具和代理，以动态调整并完成任务。
- **通信协议：**允许各组成部分之间建立连接。

图 1. 代理式系统比标准 LLM 拥有更多功能

需要了解的术语

代理式 AI 系统和工作流

代理式 AI 系统不仅仅是大语言模型（LLM）。它们是由多个协同运作的 AI 系统组成的集合体，能够综合运用推理、记忆、规划及外部工具，持续处理复杂的任务。这些系统遵循结构化工作流，使 AI 能够根据现实条件自主或半自主地采取行动。

模型上下文协议（MCP）

MCP 这一开放标准定义了 AI 代理如何以一致且可解释的方式，与工具、数据和记忆进行交互。借助该标准，开发人员能够设计出模块化、可重复使用且更易于调试或扩展的 AI 系统。

Llama Stack

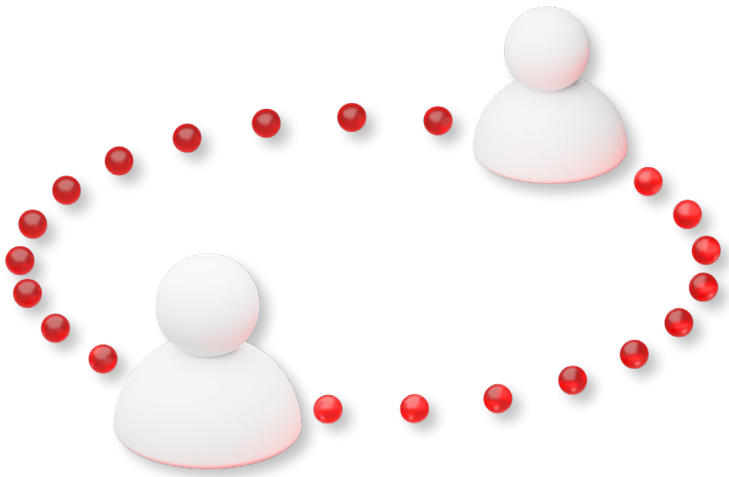
Llama Stack 是一个统一的软件层，它将 Llama 模型封装在生产就绪型工具中，包括应用编程接口（API）、编排、日志记录和工具集成。它简化了基于 Llama 的代理式 AI 系统在企业环境中的部署和运维流程。

LangChain

LangChain 是一个开源框架，有助于开发人员利用语言模型构建复杂的应用。它与 MCP 和 Llama Stack 非常相似，提供将模型与外部数据、记忆和工具连接的工具。尽管 LangChain 并非红帽 AI 使用的基础框架，但红帽 AI 作为灵活的开放平台，可以支持符合项目需求的其他方法。

第 1 章

利用红帽 AI 创建和部署 AI 代理



创建代理式 AI 系统，远不只是将大语言模型（LLM）嵌入到应用中那么简单。任何高效代理的核心，都是一个能够推理、规划、行动和学习的系统，而这一切都依赖于一系列不同的技术组件。这些组件包括：将问题分解为多个子任务的推理链、定义代理行为的提示词、用于维护上下文的记忆，以及支持超越 LLM 预训练权重范围采取行动的外部工具。

红帽 AI 有助于简化这一复杂架构。该平台以 OpenShift AI 作为核心组件构建而成，整合了推理、编排、安全防护、可观测性和合规等关键能力，并将这些能力与 AI 代理高效运作所需的工具相连接。

借助红帽 AI，团队可以从可管理的用例入手，并随着时间的推移逐步扩展。许多企业组织首先构建内部检索代理，部署 LLM 增强型知识机器人，这些机器人能够根据特定于企业的数据来回答问题。其他企业组织则构建用于日志修复和 IT 事件处理的代理，将红帽 OpenShift 可观测性与红帽 Ansible® 自动化平台及外部应用接口（API）集成到一起。另一常见场景是 AI 辅助代码迁移，代理通过分析代码库并提出升级路径建议，助力企业完成从传统架构到现代化架构的转型。

至关重要的是，这些代理并非仅执行单次任务的助手，而是内置推理和记忆能力的多步骤工作流的组成部分。随着复杂性不断增加，红帽为多代理应用提供了全面的生命周期管理和编排功能，并设定明确的委派机制和决策检查点。

迈向企业级代理式 AI 的道路，始于以下基础：可组合的架构、可重复使用的工具，以及一个能将实验成果转化为常态化运营的统一平台。借助红帽 AI，团队能够安心构建，并随心部署。

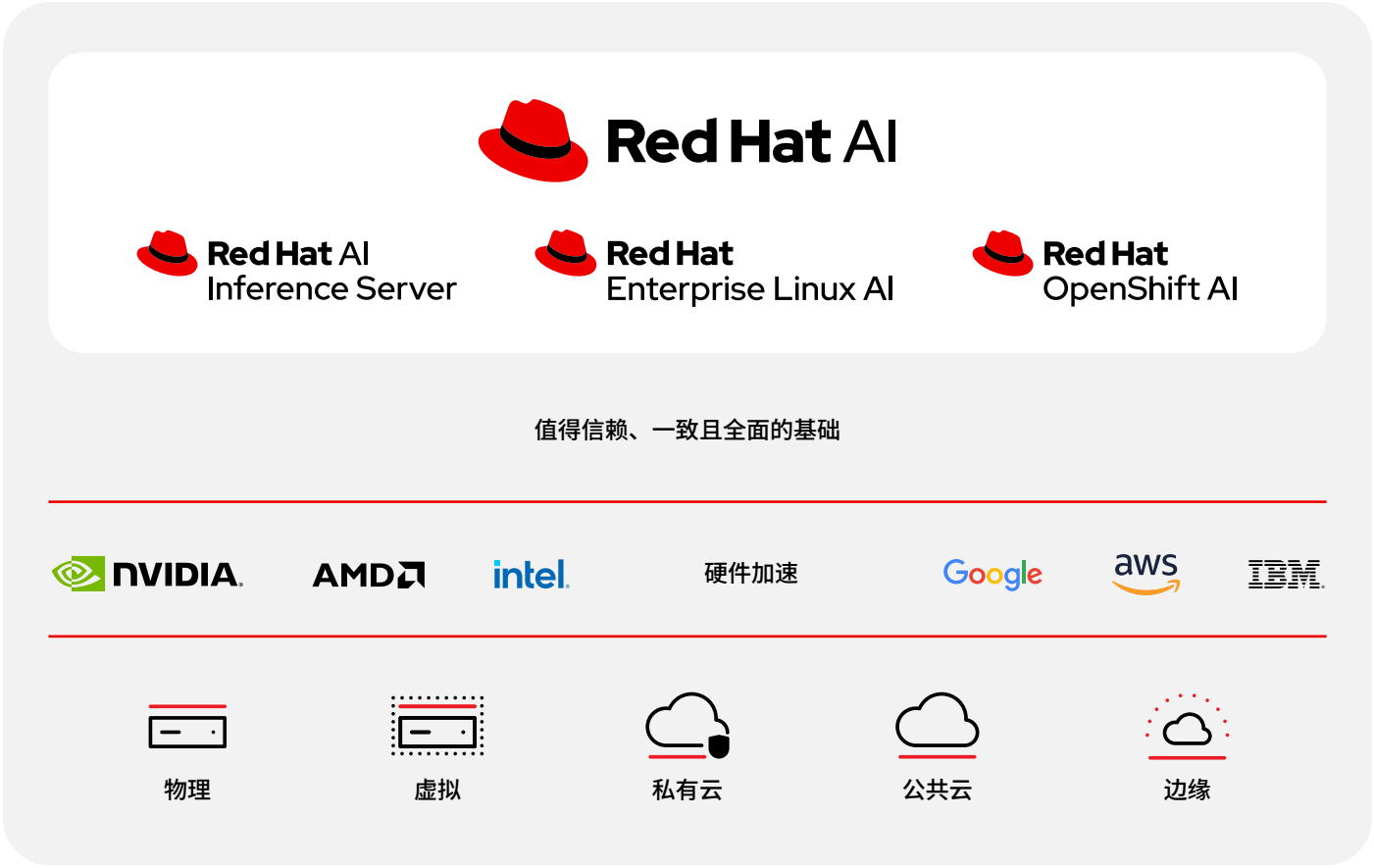


图 2. OpenShift AI 是红帽 AI 的一部分

用例

当今企业组织如何构建代理

大规模客户支持

一家网络安全企业实施了由 AI 驱动的支持系统，以实现实时聊天和工单解决自动化，旨在缩短等待时间并减轻人工客服的工作负担。工单接收代理会处理查询，然后将问题转交给分类代理，该代理会利用关键字列表和生成式 AI 情绪分析技术进行情绪标记、紧急程度分析和粗俗语言检查。如果检测到问题语言，将触发人工审核。

接下来，解决代理会利用检索增强生成（RAG），从内部文档和以往工单中提取信息，生成相应回复。如有需要，路由代理会将未解决或可信度较低的工单上报给人工支持代表。人工客服会核实 AI 建议的回复内容，特别是涉及合规敏感的内容。Airtable 日志会跟踪人工处理和错误情况，为模型持续优化提供数据支持。工单关闭后，将会重新评估情绪，以确认客户满意度。

第 2 章

红帽 OpenShift AI：面向 AI 生命周期的企业级平台

从原型设计过渡到生产阶段，这始终是企业 AI 领域最棘手的挑战之一，对于代理式应用而言更是如此。借助 OpenShift AI，企业组织可有效构建、运行和管理代理式应用。

OpenShift AI 整合了红帽核心技术，可为代理式工作负载提供生产级基础。这一解决方案的核心是基于 Operator 的模型，该模型将部署最佳实践编码化，并实现平台配置自动化。Operator 可简化从自动扩展和可观测性集成到性能调优和横向或纵向扩展策略等所有环节，让工程团队能够专注于构建代理，而非管理基础架构。

OpenShift AI 基于红帽 OpenShift 的功能打造而成，为大规模管理生成式 AI 和预测性 AI 模型的生命周期提供了一个平台。

该平台原生集成 Llama Stack 和 MCP 等组件，这有助于统一开发和部署实践。开发人员可以基于 Llama Stack 兼容 OpenAI 的 API 进行构建，并安心部署代理，因为他们知道，本地测试中使用的接口在生产环境中同样受到支持。MCP 服务器提供了一种标准化的工具提供与管理方式，兼容各类框架，并能与红帽的安全性及可观测性堆栈实现深度集成。

安全性和合规性是 OpenShift AI 不可或缺的组成部分。红帽长期以来一直专注于受严格监管的环境，这意味着代理式应用在部署时已内置防护机制和基于角色的访问控制。借助集成式可观测性工具，团队能够追踪代理决策、监控工具调用情况，并构建支持持续评估的分析管道。

最终形成了一个统一平台，使 AI 工程师、IT 团队和安全方面的利益相关者能够更高效地协作。无论是部署简单的支持代理还是用于实现内部自动化的多代理编排模式，OpenShift AI 都有助于安全、可重复地大规模实施代理式 AI 系统。

用例

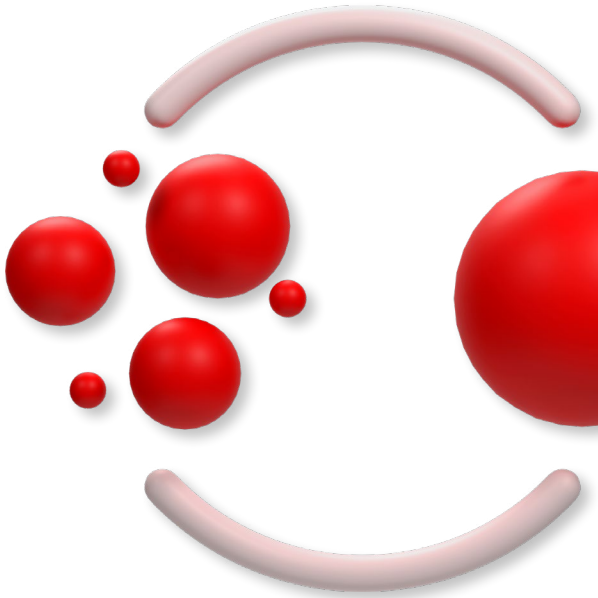
当今企业组织如何构建代理

以 AI 为中心的业务流程自动化

一位客户开发了一款采购助手，该助手可通过 MCP 访问企业资源规划（ERP）API、解读策略文件，并生成供应商审批建议。该代理实现了供应商资格验证、合同条款审查等常规步骤的自动化，在确保合规性的同时减少了采购工作流中的瓶颈。这一实践为将代理式工作流嵌入现有企业系统提供了范例，无需重构核心逻辑。

第 3 章

模型上下文协议：标准化代理与工具的交互



现代代理式系统依靠工具调用来拓展 LLM 的能力边界。虽然 LLM 提供推理功能，但它们需要工具来采取行动、访问企业系统和收集实时信息。MCP 为代理与这些外部工具之间提供了缺失的连接纽带。尽管 MCP 近期才推出，但它已迅速成为代理式 AI 领域的新标准。

在 MCP 出现之前，工具集成依赖手动操作，存在一致性问题且难以扩展。开发人员必须编写自定义代码，以定义代理如何发现 API 并与之交互。这不仅会导致团队间出现重复工作、降低更新稳定性，并且每次系统变更时都会引入额外风险。MCP 通过可互操作的模块化规范将此流程标准化，使代理能够可靠地发现、选择和调用工具，这与 AI 工作流的 USB-C 标准非常相似。

借助 MCP，开发人员可以使用通用协议，以工具的形式提供功能。这些工具涵盖 API、数据库、业务系统乃至内部实用程序。MCP 服务器作为中央枢纽，承担工具的托管、文档管理与安全防护职能。代理可据此动态查询该服务器，分析工具使用情况，并跨多个系统调用工作流。

这一标准化举措为实现更广泛的协同参与创造了条件。借助基于 MCP 的架构，团队可以重复利用现有工具定义、加快上手速度并减少平台碎片化。无论面对规模较小还是较大的 LLM，MCP 都能帮助企业组织实现更一致的工具使用体验，无需重新训练模型或重写逻辑。

然而，MCP 也面临一些挑战。工具描述定义不清，可能会导致代理混淆，引发误操作或幻觉行为。更糟糕的是，暴露不安全的提示词字符串或权限范围过广的工具，很容易成为被恶意利用的攻击载体。

红帽制定了面向 MCP 网关的路线图，通过将治理、安全防护注意事项和可观测性直接整合到 MCP 服务器架构中，帮助缓解这些风险。当 MCP 服务器部署于 OpenShift AI 时，将自动继承红帽平台级策略实施、基于角色的访问控制及审计机制的原生集成能力。这意味着通过 MCP 提供的工具的访问权限，可以按照角色和命名空间进行严格管控，与身份认证和授权策略实现无缝衔接。

此外，还可以利用红帽 OpenShift 的容器和应用安全防护工具链，自动扫描 MCP 中托管的工具描述和提示模式，以查找漏洞。这使平台团队能够识别可能存在危险的工具配置，例如允许提示注入的工具，或在未进行适当输入验证的情况下暴露敏感后端系统的工具。

从运维角度来看，借助红帽 OpenShift 可观测性工具，可实现对代理与工具交互的持续监控。团队可查看工具调用模式、跟踪使用指标，并对异常行为设置警报。借助集成式审计日志，可追溯工具使用情况和决策链，助力企业满足内部和外部合规要求。

通过将 MCP 与其他红帽 AI 组件（如 Llama Stack 的评估、工具和安全 API）以及综合平台相结合，客户能够可靠地部署和运维 MCP 服务器，从而助力创建可扩展且注重安全的代理式 AI 工作流。这有助于企业组织打造出不只是纸上谈兵，更能切实完成任务的代理。

用例

当今企业组织如何构建代理

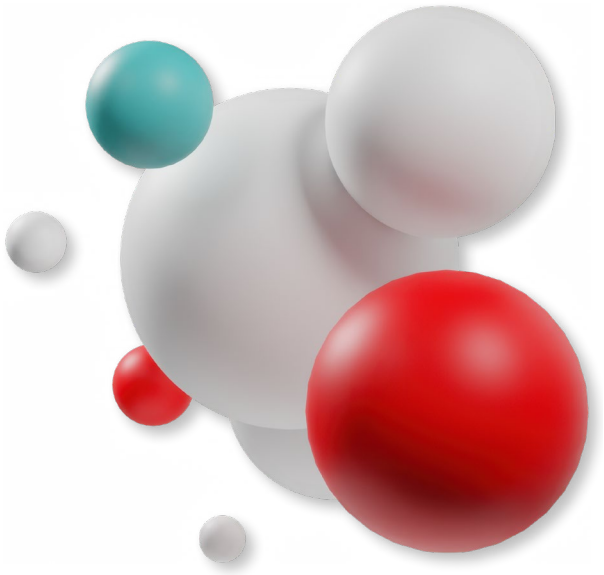
大规模客户支持

一家网络安全企业实施了由 AI 驱动的支持系统，以实现实时聊天和工单解决自动化，旨在缩短等待时间并减轻人工客服的工作负担。工单接收代理会处理查询，然后将问题转交给分类代理，该代理会利用关键字列表和生成式 AI 情绪分析技术进行情绪标记、紧急程度分析和粗俗语言检查。如果检测到问题语言，将触发人工审核。

接下来，解决代理会利用检索增强生成（RAG），从内部文档和以往工单中提取信息，生成相应回复。如有需要，路由代理会将未解决或可信度较低的工单上报给人工支持代表。人工客服会核实 AI 建议的回复内容，特别是涉及合规敏感的内容。Airtable 日志会跟踪人工处理和错误情况，为模型持续优化提供数据支持。工单关闭后，将会重新评估情绪，以确认客户满意度。

第 4 章

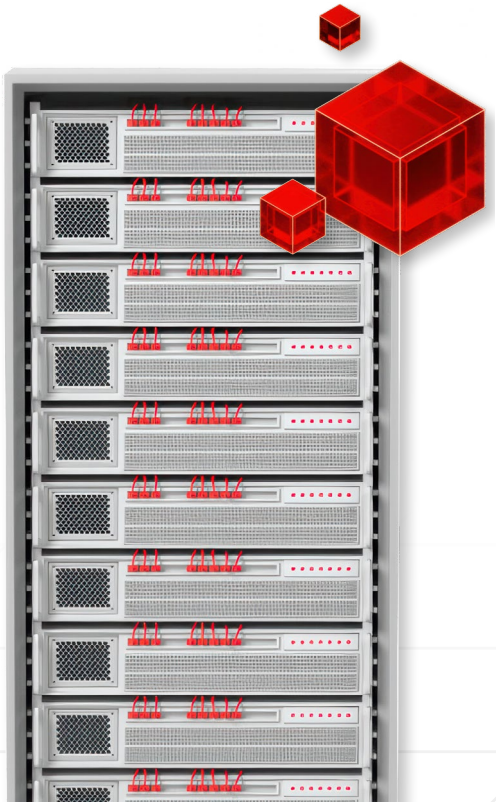
Llama Stack：基于 OpenShift AI 的统一 AI API 服务器



Llama Stack 是红帽的统一 AI 控制平面，也是兼容 OpenAI 的 API 服务器，旨在简化代理式 AI 应用的开发、部署和管理流程。Llama Stack 作为推理、记忆、工具编排及评估的中央接口，为开发人员提供了跨环境构建复杂代理所需的一致性和灵活性。

许多其他通常作为托管服务提供的解决方案不同，Llama Stack 可帮助企业组织在自有硬件或集群上打造出托管式体验。对于需要数据主权、具有特定基础架构要求或希望避免受制于特定供应商的企业组织而言，这种灵活性至关重要。

Llama Stack 的核心价值在于提供标准的 API 层，支持完整的代理式 AI 生命周期，不仅限于推理，还兼容 OpenAI API。它直接与 OpenShift AI 集成，并支持常见的代理任务，如检索增强生成（RAG）、安全性、评估、遥测和上下文感知推理。开发人员可以从轻量级本地部署开始，并扩展到企业级基础架构，整个过程中使用相同的 API、库和接口。



红帽提供多种 Llama Stack 采用方式，具体取决于团队准备情况和项目复杂程度。刚接触代理式系统的团队，可以利用内置客户端和 SDK 快速起步，它们提供了工具调用、记忆和上下文管理等预配置组件。这些功能有助于简化开发流程并降低复杂性。对于更高级的用户，Llama Stack 与 OpenAI 的工具使用接口 API 兼容，并能与许多热门框架搭配使用。这使团队能够集成现有代理和工作流，而无需重构，同时仍能受益于跨环境的一致接口和生命周期管理工具。

Llama Stack 的独特之处在于其开源的 API 层和服务器，可跨提供商标准化 AI 应用开发。它不仅支持兼容 OpenAI 的推理接口，还提供多种其他 AI 功能（如评估、后训练、向量存储）的额外 API，以及用于支持 RAG 和代理工作流的防护机制。Llama Stack 可在自托管环境、本地或云端运行，使企业组织能够灵活进行部署。

在 OpenShift AI 中，Kubernetes Operator 负责管理 Llama Stack 生命周期任务，如自动扩展、可观测性和访问控制，为开发人员和平台团队提供一套统一的工具，用于可靠地扩展代理。Llama Stack 还原生支持评估和遥测（包括 OpenTelemetry），以帮助团队验证 AI 系统性能、监控安全指标并跨生产环境跟踪代理行为。

Llama Stack 不仅支持推理，还旨在支持代理式系统中的诸多动态组件。它能充当托管模型与本地模型之间的桥梁，标准化访问 TrustyAI 等安全防护工具，并促进与 MCP 服务器的交互。无论是运行概念验证演示，还是自动化生产 IT 工作流，开发人员都可以通过一致的平台来测试、迭代和实施代理。

Llama Stack 将原本复杂的代理编排工作，转化为模块化且可重复的流程。依托红帽的支持以及与 OpenShift AI 的深度集成，该平台为团队提供了安全、可预测地扩展代理式 AI 系统所需的控制平面。

构建 AI 代理的模块化方法

红帽 AI 平台能够实现以下目标：

- 利用 **Llama Stack 的原生功能和实施**来构建代理。
- **将兼容的 Llama Stack 实施集成**至 OpenShift AI。
- **使用您自己的代理框架**，并选择性地整合 Llama Stack API。
- **基于核心原语进行构建**，并将您自己的代理框架作为标准工作负载进行管理。

图 3. Llama Stack 完美契合您的模块化开放框架

用例

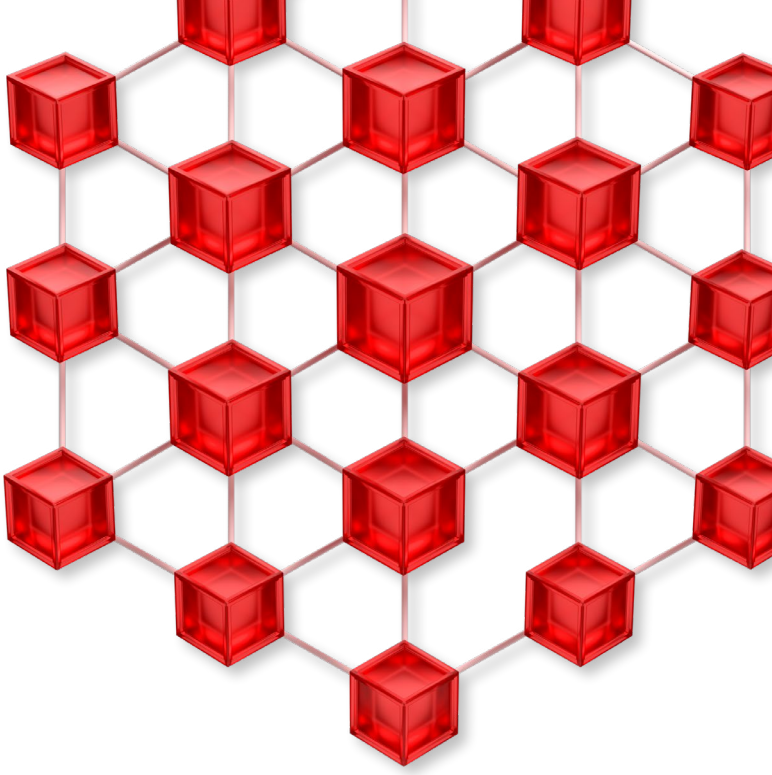
当今企业组织如何构建代理

开发人员生产力

一家软件企业组织构建了一个代码迁移助手，该工具能够分析传统的 Java 应用，并推荐更新，以与现代框架保持一致。该代理使用 Llama Stack 进行推理，并借助 MCP 集成工具验证兼容性并建议升级路径。这简化了迁移流程，减少了技术债务并提高了应用弹性。它还有助于开发团队专注于功能创新，而非耗费时间重写传统代码。

结论

红帽 AI：成功部署代理式 AI 的技术基石



代理式 AI 有望带来更智能、适应性更强的应用，但若缺乏标准化，企业团队将面临碎片化、效率低下及运维风险。红帽 AI 提供了从容构建这些系统所需的框架。借助 OpenShift AI、MCP 和 Llama Stack 等组件，红帽为将 AI 代理从概念验证阶段顺利推进至生产阶段奠定了一致的基础。

通过将推理、工具使用、记忆、安全防护和评估集成到统一平台，红帽 AI 简化了团队实验、扩展和实施 AI 代理的方式。

- 开发人员可访问生产级 API。
- 平台团队能够管理生命周期并实施安全防护策略。
- 企业组织可受益于保护其投资的开放标准。

无论是部署简单的内部机器人，还是构建多代理系统，红帽 AI 皆使您能够根据自己的需求，以可重复的方式构建安全至上的企业级代理式应用。