

Hermes Agent 深度研究报告

从大模型助手到可执行智能体：能力、
架构、场景与趋势

聚焦 Agent 范式、核心能力、技术架构、应用场景与行业趋势
面向产品、技术与管理团队的系统化研究报告

能力 / 架构 / 场景 / 竞争 / 趋势

核心结论：Hermes Agent 是“可执行型 AI Agent”的代表形态

- Hermes Agent 不只是聊天机器人，而是具备“理解目标—调用工具—执行任务—验证结果”闭环能力的智能体。

- 其核心价值在于把大模型的语言理解能力扩展为真实世界中的任务执行能力。

- 相比传统 Copilot 类产品，Hermes 更强调工具编排、长任务执行、状态管理与结果交付。

- Hermes 的竞争力不只来自模型本身，更来自工具系统、技能系统、任务调度与运行约束设计。

- 未来 AI 产品的关键分野，不是“谁更会回答”，而是“谁更能完成任务”。

一句话判断

Hermes 代表 AI 从“会说”走向“会做”。

为什么要研究 Hermes Agent

● Agent 正在成为大模型产业从“问答”走向“行动”的关键范式。

● Hermes Agent 代表了“CLI + 工具 + 记忆 + 自动化”的落地方向。

● 对企业而言，Agent 可能成为知识工作流程自动化的新基础设施。

● 对产品团队而言，Hermes 展示了从“对话产品”升级为“任务产品”的路径。

● 对技术团队而言，Hermes 是观察多工具协同、任务分解、执行安全与可控性的良好案例。

研究价值

它是理解下一代 AI
产品形态的典型样本

本报告重点回答的 6 个问题

问题 1

Hermes Agent 究竟是什么？
与 Chatbot、Copilot、RPA 有何不同？

问题 2

Hermes 的核心能力边界在哪里？

问题 3

它的技术架构由哪些模块构成？

问题 4

适合哪些业务场景？如何创造 ROI？

问题 5

与 OpenAI Operator、Claude Code、
AutoGPT 等相比有何差异？

问题 6

Agent 未来 2-3 年的演进方向是什么？

研究方法

● 产品能力拆解：从工具集、交互方式、执行链路进行结构化分析。

● 技术视角分析：从 LLM、规划、记忆、工具调用、安全机制进行解构。

● 场景视角分析：从研发、运营、办公自动化、知识管理等维度评估落地性。

● 行业对比：与典型 Agent 产品及框架进行横向比较。

● 趋势判断：基于 Agent 行业发展、模型能力进步与企业需求变化，进行推演。

方法论关键词

**能力拆解 + 技术解构
+ 场景评估 + 横向对比 + 趋势推演**

报告目录

1

执行摘要

2

Agent 范式与 Hermes 定位

3

Hermes Agent 能力体系

4

技术架构与运行机制

5

应用场景与商业价值

6

竞争格局与行业对比

7

风险、挑战与未来趋势

8

结论与建议

01

Agent 范式与 Hermes 定位

从聊天机器人走向任务执行型智能体

AI 产品形态的三次跃迁

1

Chatbot

回答问题、生成内容

2

Copilot

辅助用户完成局部任务

3

Agent

接收目标、自主拆解并执行完整任务

Hermes Agent 属于第三阶段，强调“结果交付”而非“仅提供建议”。

Agent 的标准定义

● Agent 是一种能够在给定目标下结合环境感知、规划推理、工具调用、记忆与反馈机制。

● 关键特征一：目标驱动。

● 关键特征二：多步规划。

● 关键特征三：工具使用。

● 关键特征四：状态保持。

● 关键特征五：结果验证与迭代。

定义核心

Agent = 推理能力 + 执行能力 + 反馈闭环



Hermes Agent 的定位：任务执行型通用智能体

- 面向复杂数字任务，而不仅是文本生成。

- 以工具系统为核心，而不是把能力全部压在模型参数中。

- 以 CLI / 工作流 / 自动化为主要执行空间。

- 支持多类型任务：代码、文件、网页、流程、通知、调度、记忆。

- 更像“会使用计算机和外部系统的数字员工”。

定位关键词

任务执行型、工具原生、跨场景、可自动化

Hermes Agent 与 ChatGPT 的区别

ChatGPT

- 以对话为主
- 执行能力依赖插件或用户手动操作
- 更擅长解释与生成
- 交付物多停留在文本层

Hermes Agent

- 原生具备任务执行能力
- 可读写文件、执行命令、调用工具
- 更擅长行动与闭环
- 交付物可直接落到系统环境中

一句总结：ChatGPT 回答“怎么做”，Hermes 更接近“替你做”。

Hermes Agent 与 Copilot 的区别

Copilot

- 通常嵌入单一 workflow, 例如 IDE 内代码补全
- 偏局部建议
- 适合提升单点生产效率
- 较少承担完整任务链

Hermes Agent

- 面向跨工具、跨步骤、跨系统任务
- 偏完整任务链执行
- 可组织任务并持续推进
- 更强调结果闭环与验证

案例：Copilot 帮你写函数；Hermes 可分析仓库、改代码、跑测试并整理说明。

Hermes Agent 与 RPA 的区别

传统 RPA

- 依赖固定流程和规则
- 适合高重复、低变动任务
- 稳定性和可预测性较强
- 对异常和非结构化输入适应有限

Hermes Agent

- 基于自然语言目标和动态推理
- 更能应对半结构化任务
- 灵活性和泛化能力更强
- 可辅助处理异常和决策

未来不是 Agent 替代 RPA，而是两者融合：RPA 执行稳定流程，Agent 负责理解和调整。

02

Hermes Agent 能力体系

从语言理解到任务闭环的九类能力

Hermes Agent 的能力地图

对话理解

理解复杂目标与约束

任务规划

拆解多步骤任务并维护状态

工具调用

把语言推理转化为系统动作

文件与代码 操作

读写、搜索、修改、运行

浏览器交互

导航、点击、输入、视觉分析

记忆与技能 复用

积累偏好、经验和流程

| 2026年4月

能力一：目标理解与任务解释

- 能够理解用户用自然语言表达的复杂目标。
- 能对模糊需求做默认推断，并在必要时追问。
- 支持中英文混合、多轮上下文、任务约束识别。
- 将用户意图转化为可执行步骤。

价值

这是从“语言接口”通往“行动接口”的第一层。



能力二：多步骤任务分解

- 将复杂任务拆成可执行子任务。
- 维护任务列表与状态。
- 识别依赖关系与先后顺序。
- 支持逐步推进、动态调整与异常重试。

典型价值

让模型不只“想到答案”
还能“组织完成过程”



能力三：工具使用是 Hermes 的核心竞争力

- 工具类型包括：文件读写与检索。

- Shell / Terminal 命令执行。

- 浏览器导航与页面交互。

- 代码编辑与补丁应用。

- 图像与视觉分析、定时任务、记忆存储、子代理委派。

结论

**工具能力越丰富，
Agent 的可执行边界
越大。**

能力四：面向研发场景的工程执行能力

- 查看仓库结构与代码内容。

- 搜索文件与依赖关系。

- 修改文件、生成 patch。

- 执行测试、构建、运行脚本。

- 管理 Git 工作流与 PR 流程。

价值

使 Hermes 能从“代码建议器”进化为“工程执行助手”。

能力五：网页操作与信息抓取能力

- 打开网页并读取结构化快照。
- 点击按钮、填写表单、滚动页面。
- 获取控制台日志与 DOM 状态。
- 结合视觉能力理解页面布局。

价值

让 Agent 不依赖纯 API，也能在真实 Web 环境中执行任务。



能力六：技能（Skills）让 Agent 可复用、可进化

- 技能是结构化的程序性知识。
- 当遇到特定任务时，先加载对应 skill，再按最佳实践执行。
- 技能可以创建、更新、修补。
- 这使 Hermes 从单次问答系统升级为“经验会积累的执行系统”。

结论

Skill 是 Hermes 提高稳定性和专业度的重要机制。

能力七：通过子代理实现并行与分工

- Hermes 可将子任务委派给多个独立代理。
- 每个代理拥有独立上下文与工具集。
- 适合并行研究、代码审查、信息汇总。
- 主代理负责协调与汇总结果。

意义

这是 Agent 从单线程助手走向协作式执行系统的关键一步。

03

技术架构与运行机制

理解 Hermes 如何从目标到执行再到验证

Hermes Agent 的总体架构

1

输入层

接收目标、上下文与约束

2

推理与规划层

解析任务并决定行动路径

3

工具编排层

选择并调用适当工具

4

执行环境层

在文件、终端、浏览器中真实执行

5

反馈验证层

检查结果、修正错误、决定下一步

横向模块：记忆与技能系统持续为执行提供历史经验与流程模板。

第一层：任务理解与上下文建模

- 对用户输入进行语义解析。
- 识别明确目标、隐含约束、格式要求、执行边界。
- 判断是否需要澄清，或按默认解释直接行动。
- 将自然语言转化为内部任务表示。

核心价值

把“说法”变成“做法”。



第二层：规划与决策机制

- 判断任务是否需要分步。
- 决定先调用什么工具。
- 在工具返回后更新计划。
- 根据结果继续推进、重试或改道。

关键问题

Agent 不是一次性求解，而是循环式推理与行动。



第三层：工具路由与调用决策

- 根据任务类型选择合适工具。

- 数学问题调用代码/终端。

- 文件问题调用
read/search/patch。

- 网页问题调用 browser 工具。

- 并行问题调用 `delegate_task`。

本质

把大模型的语言推理
转化成系统调用。

第四层：真实执行环境

- Linux shell / terminal 环境。

- 文件系统。

- 浏览器会话。

- 后台进程。

- 脚本运行环境。

意义

Hermes 不只是模拟执行，而是在真实环境中完成任务。

文件系统是 Hermes 的核心工作平面

- 读取文件内容。
- 搜索目录与代码库。
- 写入和 patch 修改文件。
- 保持持久化结果。

价值

让 Agent 的产出从“**聊天文本**”变成“**系统中的真实资产**”。

浏览器是 Hermes 连接互联网与 Web 应用的桥梁

- 导航网页。

- 提取可交互元素快照。

- 点击、输入、滚动。

- 读取控制台日志。

- 视觉分析截图。

难点

Web 环境动态复杂
因此浏览器能力是
Agent 差异化的重要
战场。

技能系统：经验沉淀的标准化接口

- 技能以结构化文档形式存在。
- 包含适用场景、步骤、注意事项、验证方式。
- Agent 在执行前先匹配 skill。
- 执行过程中若发现 skill 过时，可及时 patch。

意义

这相当于给智能体建立“可维护的程序性知识库”。

为什么 Agent 必须具备验证能力

- 仅有生成，不足以保证正确。

- Hermes 在执行完成前会检查输出是否满足要求。

- 对代码场景可运行测试。

- 对网页场景可读取控制台和页面状态。

- 若验证不足，会继续调用工具补充证据。

结论

验证能力决定 Agent 的可用性天花板。

Hermes 的执行约束设计

- 明确高风险操作需确认范围。
- 工具权限边界清晰。
- 记忆写入有选择性。
- 用户交互、自动化调度和真实执行之间有安全门槛。

核心矛盾

Agent 越强大，越需要治理；否则执行能力会转化为风险。



04

应用场景与商业价值

从研发提效到组织流程自动化

Hermes Agent 可以落地在哪些场景

软件研发与 DevOps

代码、测试、部署、排障、巡检

办公自动化 与知识工作

报告、邮件、日程、知识整理

数据收集与 研究分析

多源搜索、对比分析、结构化报告

企业流程执 行与监控

调度、告警、汇总、自动触发

场景一： 软件研发助手升级为工程执行助手

- 代码检索与解释。

- 自动修改文件。

- 运行测试与定位错误。

- 生成文档与变更说明。

- 协助 PR、Issue 与 Review。

业务价值

显著降低开发者在上下文切换、重复操作以及排障上的时间成本。

场景二：自动化运维与系统检查

- 检查服务状态、端口、日志。
- 运行脚本与部署命令。
- 做定时巡检和告警汇总。
- 自动生成健康检查报告。

优势

相比传统脚本，
Agent 更能理解异常
临时调整路径与生成
解释。

场景三：知识 workflows 自动化

- 汇总文档、生成报告。
- 管理日程、邮件、任务列表。
- 定时收集信息并输出周报。
- 多系统之间做轻量级流程编排。

意义

Agent 正在把“文员型数字工作”从手动操作转向自然语言驱动。

场景四：研究员型 Agent

- 多源信息采集。

- 历史资料搜索与摘要。

- 对比竞品与行业方案。

- 输出结构化研究报告。

价值

尤其适合二级研究、
行业扫描、产品情报
与技术调研。

场景五：个人 AI 执行秘书

- 帮助整理文件。
- 定期提醒和总结。
- 自动检查特定事项。
- 管理研究资料与个人知识库。

趋势判断

个人用户需求会从“聊天陪伴”逐渐转向“任务代理”。

Hermes Agent 为企业带来的三类价值

效率价值

缩短任务完成时间，减少人工重复操作。

质量价值

标准化流程，降低漏项与返工。

组织价值

沉淀技能与最佳实践，让经验从个人能力转为系统能力。

如何评估 Hermes Agent 的 ROI

指标 1

单任务耗时下降比例

指标 2

人工操作步骤减少量

指标 3

重复任务自动化率

指标 4

错误率下降程度

指标 5

员工可释放的高价值时间

企业落地 Hermes Agent 的建议路径

1

阶段 1

个人提效工具

2

阶段 2

团队 workflow 助手

3

阶段 3

流程级自动化节点

4

阶段 4

跨系统协作执行层

建议从“辅助模式”起步，逐步过渡到“半自主执行”，最后进入“自动运行”。

05

竞争格局与行业对比

Hermes 在 Agent 赛道中的位置

AI Agent 赛道的主要玩家

通用智能助手

ChatGPT、Claude、Gemini

编程型 Agent

Claude Code、Codex、Cursor Agent

自动化型 Agent

OpenAI Operator、

开源框架型 Agent

AutoGPT、LangGraph、CrewAI、OpenDevin

与通用对话模型相比，Hermes 的差异

通用模型优势

- 世界知识更广
- 通用对话体验更成熟
- 生态普及度更高
- 适合开放式问答和创作

Hermes 优势

- 工具原生集成更强
- 更强调执行闭环
- 更接近操作系统/工作流层
- 更适合复杂任务交付

结论：Hermes 更像“工作执行器”，而非纯“智能问答器”。

与编程型 Agent 相比, Hermes 更通用

编程型 Agent

- 聚焦软件开发与工程 workflows
- 在 IDE/代码语境中更深
- 对代码上下文优化更强
- 场景边界较集中

Hermes Agent

- 除代码外, 还可覆盖浏览器交互、记忆管理、任务规划、流程调度、消息通知等更广的任务场景
- 在跨场景任务编排 上更灵活
- 既能支持工程, 也能支持办公和研究任务
- 强调跨域执行能力

Hermes 的核心价值在于跨域执行, 而非单一专业深度。

产品化 Agent 与开源框架的差异

开源框架

- 更像开发框架
- 灵活但落地成本高
- 需要团队自行搭建治理与工具层
- 适合技术团队定制化开发

Hermes Agent

- 更像具备完整工具体系和操作规范的成品化 Agent
- 即用性更强
- 治理约束更明确
- 操作标准更统一

企业真正采用的，往往不是“最开放”的系统，而是“最可控”的系统。

Hermes Agent 的 5 个核心优势

优势 1

工具链完整

优势 2

行动导向强

优势 3

记忆与技能机制成熟

优势 4

多代理协同能力

优势 5

面向真实执行环境，而非纯文本环境

Hermes 当前可能存在的短板

- 对底层模型能力仍有依赖。
- 复杂任务中的规划稳定性仍可能波动。
- 工具生态与外部系统接入深度决定上限。
- 自动执行越强，安全治理难度越高。
- 普通用户的上手门槛可能高于聊天产品。

判断

Hermes 的挑战在于稳定性、生态深度与普适易用性。

Agent 竞争的真正焦点是什么

● 谁能连接更多工具。

● 谁能更稳定完成长任务。

● 谁能在执行中自我验证。

● 谁能在安全边界内实现更高自动化。

● 谁能沉淀技能，形成组织级复用。

结论

Agent 产品的竞争将从“智力竞争”转向“系统工程竞争”。

06

风险、挑战与未来趋势

可靠性与治理将决定 Agent 能否大规模落地

Hermes Agent 面临的 4 大挑战

规划错误

任务拆解不合理导致执行偏航。

工具错误

调用不当或环境依赖失败。

幻觉与误判

尤其在信息不完整时风险上升。

安全问题

错误执行、高权限操作、数据泄露风险。

Agent 时代的治理框架必须提前建立

● 权限管理。

● 操作审计。

● 数据访问边界。

● 高风险动作审批。

● 自动化任务的可追踪性。

观点

**没有治理框架的
Agent，很难进入企
业核心流程。**

Hermes 类 Agent 的未来演化方向

- 更强的长上下文与长期记忆。
- 更可靠的规划与反思机制。
- 更标准化的工具协议，如 MCP。
- 更深度的多代理协同。
- 从“执行单个任务”走向“持续承担岗位功能”。

判断

Agent 将逐渐成为数字 workflow 中的常驻角色。

未来 2-3 年的产业趋势判断

● Agent 将成为 AI 应用层最重要的形态之一。

● 企业会从试点走向场景化部署。

● 编程、研究、运营、办公会最先被深度改造。

● 通用聊天助手会逐渐融合 Agent 能力。

● “会做事的 AI”将成为新的产品分水岭。

趋势结论

未来竞争焦点是执行能力、治理能力与组织适配能力。

07

结论与建议

从研究判断走向落地行动

结论：Hermes Agent 代表 AI 从回答走向执行

- Hermes Agent 的本质是“可调用工具、可执行任务、可验证结果”的智能体系统。
- 它的意义在于把大模型的认知能力转化为生产力。
- 其竞争力主要来自系统设计，而不仅是底层模型。
- 企业若想真正获得 AI 红利，需要关注 Agent 在真实流程中的落地方式。

一句总结

Hermes 不是一个更会聊天的模型，而是一个更做事的系统。

建议与下一步行动

建议一

从高频、低风险、跨工具任务开始试点。

建议二

优先构建技能库与工具接入体系。

建议三

建立权限、验证、审计三位一体治理机制。

建议四

把 Agent 视为“组织能力放大器”而非单点功能。

Q&A

感谢聆听 | Hermes Agent 深度研究报告

感谢聆听