FROST & SULLIVAN
沙利文

Agent

2026年中国智能体市场洞察

侵入式Agent产业治理白皮书

White Paper on the Governance of Invasive Agents, 2026

FROST & SULLIVAN WHITEPAPER

frost.com

引言

2025年底至2026年初，Agent技术从概念验证进入产品阶段。谷歌在安卓系统引入Gemini，构建底层界面自动化框架执行跨应用任务，确立了系统级操作的新范式。同期部分缺乏约束的实现方式快速扩散，例如开源Agent项目OpenClaw获取高级权限后接管本地软件，在无用户确认时执行指令并引发数据误删事件。这些进展表明人工智能已经深度介入设备操作路径，具备代为执行复杂任务的能力。

部分Agent技术跨越标准应用接口，依赖系统权限直接读取屏幕并模拟操作，构成了侵入式的交互特征。该机制改变了传统应用生态的操作入口，将流量分发与决策权转移至系统层级。在缺乏规范化管理时，此类侵入式操作显著增加了隐私泄露、资产损失与责任界定不清的系统风险。本报告聚焦侵入式机制对产业流量与商业分配的影响，剖析安全合规隐患，并提出产业生态治理方向。

侵入式Agent打破互联网体系的治理与信任边界

随着多模态模型具备理解屏幕与界面的能力，侵入式Agent可在未获授权的情况下读取敏感信息并跨应用执行高敏操作。这种绕开平台接口的侵入式路径打破既有权限与治理边界，诱发平台与智能体对抗，并冲击互联网信任体系。

侵入式Agent改变传统流量分发路径

侵入式Agent将用户任务起点从应用入口前移到系统级智能体，用户不再先选App，而是直接下达任务，由智能体跨应用执行。这会削弱应用直接服务用户的能力，使导航、推荐与商业化入口被压缩或绕过，入口价值随之下滑。

侵入式Agent加剧生态内卷

侵入式Agent本质上仍然是在原有生态里面的筛选行为，并不是做增量，是典型的内卷行为。

侵入式Agent削弱生态创新活力

侵入式Agent的路径选择以历史表现与可靠性为先，流量会向少数确定性高的应用集中；新应用缺乏可验证的先验与数据积累，更难进入推荐与调用链路，创新活力因此被压缩。

侵入式Agent抬高生态综合治理成本和市场交易成本

侵入式Agent接管流量路由后，将原本清晰、标准化的分发关系前移到系统层重新裁决，直接推高搜寻、谈判、监督与合规成本，形成长期性的交易成本上升。

通过构建全链路可审计的体系协助Agent负责任落地

侵入式Agent要实现可持续落地，关键在于把能力收敛到任务级最小权限，并对所有跨应用操作实现全链路可审计。只有让操作边界可控、执行过程可观测、责任主体可追溯，才能在提升自动化效率的同时把隐私与安全风险压到可管理范围。

构建API主导，GUI辅助的双重授权治理方式以发挥Agent全量价值

API主导、GUI辅助的治理策略，本质是在双重授权的前提下将跨应用执行分层。可标准化、可审计的操作优先API，API覆盖不到的长尾环节才允许GUI，同时对GUI施加更严格的权限确认与审计约束。

目录

01 AI Agent产业发展现状

侵入式Agent路径打破传统App生态边界与信任体系

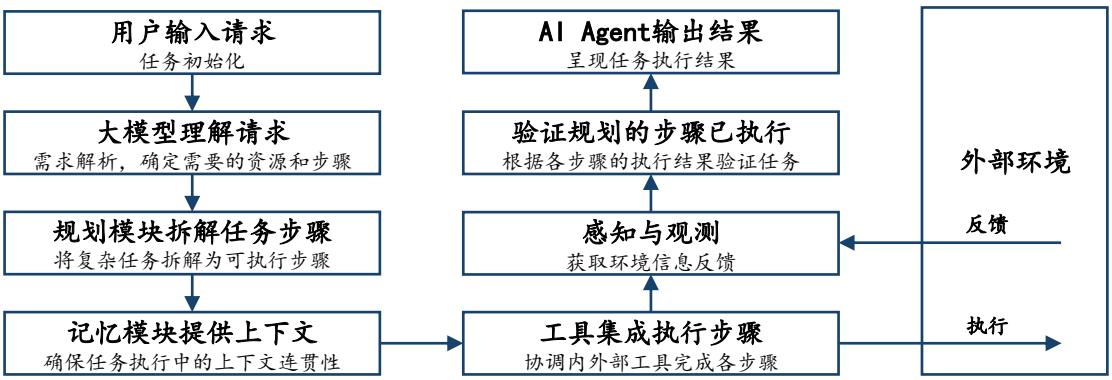
- 侵入式Agent无需应用接口即可获取跨应用执行权
- 侵入式Agent抢占用户的第一触点
- 软件应用的系统权限边界被重构
- 软件生态的行为不确定性上升

1.1.1 AI Agent概念定义

AI Agent是一种能够感知外部环境、自主制定决策并执行具体操作的智能系统。相较于传统AI，AI Agent具备自主推理与任务规划能力，能够独立完成复杂目标。

AI Agent是一个自主控制各类工具解决问题的Agent系统。一个基于大模型的AI Agent系统包括大模型、记忆系统、工具使用四大核心组件，运行机制大致为：输入请求➞大模型理解请求➞规划模块分拆任务步骤➞记忆系统提供上下文➞工具集成执行任务➞感知与观测获取环境信息反馈➞验证规划的步骤已执行➞输出结果。

图1：AI Agent运行机制



AI Agent分类体系主要围绕技术实现路径与生态影响模式展开。按**技术路径**，分为API Agent与GUI Agent两类。API Agent依托标准接口传输结构化数据，功能边界受限于第三方开放度。GUI Agent通过解析屏幕并模拟触控，具备跨软件的泛化操作能力。按**生态影响模式**，AI Agent分为工具类、合作式与侵入式三类。工具类Agent在自有生态或本地环境闭环运行，不涉及第三方交互，安全可控。合作式Agent遵循双向授权原则，基于互信机制协同操作，维护行业秩序。侵入式Agent在未经第三方授权时，跨应用调用底层权限。此行为打破了既有服务闭环，极易引发无止境的技术攻防对抗，最终导致生态割裂与用户体验下降。

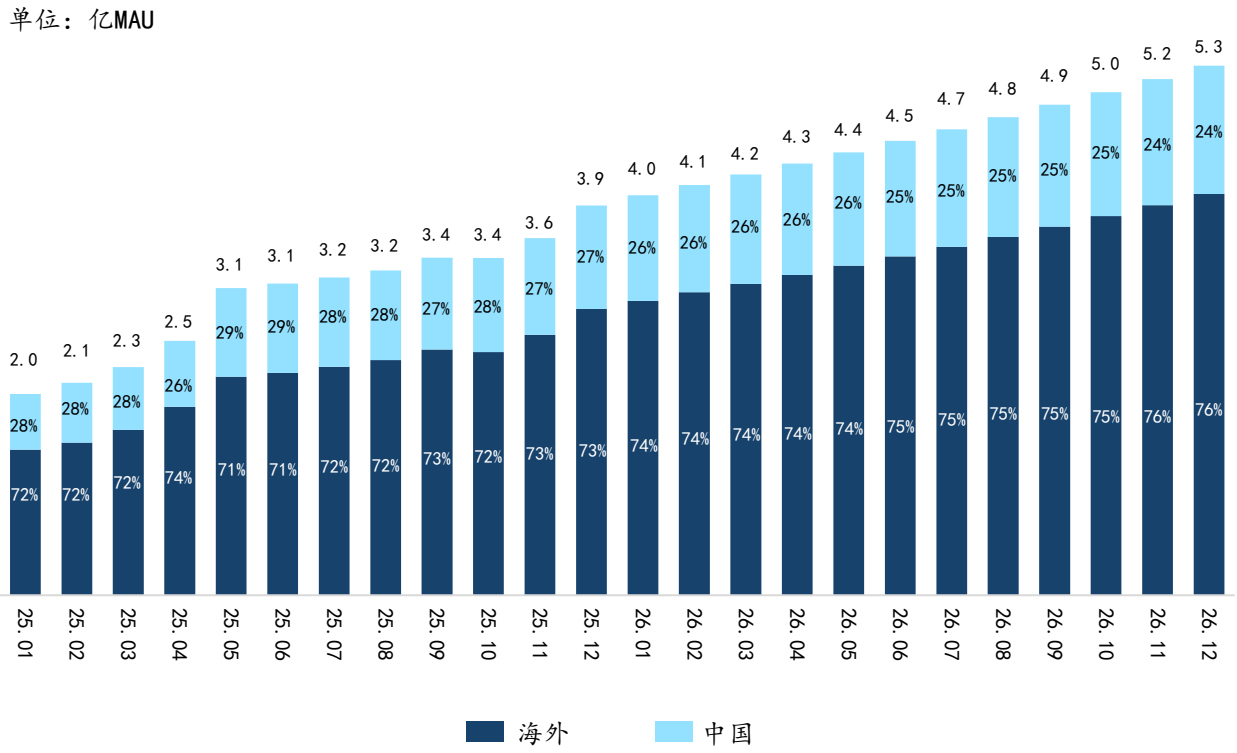
图2：AI Agent分类

分类维度	类型	形态	特征说明
按技术路线划分	API Agent	基于应用程序接口（API），通过代码与软件、系统或工具进行后台数据交互与调用	运行稳定、执行准确率和效率高，但高度依赖被调用方开放的接口生态
	GUI Agent	模拟人类操作，基于现有软件的图形用户界面（UI）进行视觉理解、点击与输入	无需底层API授权，泛化能力强，如同“数字员工”直接操作屏幕界面
按影响生态的方式划分	工具类Agent	独立运行于自有生态或本地环境，不涉及第三方软件的跨应用交互	应用边界清晰且安全可控，不干扰外部生态
	侵入式Agent	深度重构现有业务流程或产品交互，甚至独立作为全新入口替代传统软件的核心功能	颠覆并重塑原有业务生态和用户习惯，通常表现为高自主性并能闭环完成任务
	合作式Agent	以插件、Copilot或功能模块等助手形态嵌入现有系统，辅助人类或主系统工作	融入并增强现有生态，不影响原有使用链路，强调人机协同或模块间协同配合

1.1.2 AI Agent市场规模

全球AI Agent市场正迎来倍数级扩张。海外市场依托标准化的应用程序接口规范维持领先地位。中国市场由于底层系统缺乏互通，规模化落地主要表现为垂直行业的深度定制。

图3：全球及中国AI Agent APP MAU，2025-2026E



全球通用型AI Agent市场在2025至2026年间正处于高速增长期。全球月活跃用户预计将从2025年1月的2亿增至2026年12月的5.3亿，整体增幅超过一倍。在市场总量扩大的同时，各区域的增长表现差异显著。中国市场的用户占比预计从2025年初的28%逐步降至24%，海外市场份额则从72%提升至76%。这一趋势表明，海外市场未来三至五年的增长速度与普及率将持续高于中国市场。

区域增长差异的根本原因在于底层软件架构的不同。海外市场具备成熟的软件服务体系和标准化的程序接口规范，为AI应用提供了易于接入的数字环境。这让AI Agent能够以较低的边际成本融入企业现有的工作流程，快速转化为实际生产力。相比之下，中国企业级市场的软件环境较为分散，定制化开发比例高，各系统接口之间缺乏互通。这种结构限制导致通用型AI Agent在中国企业的部署成本增加且适配困难，最终影响了规模化应用的推进速度。

1.2.1 GUI Agent与API Agent的概念定义

API Agent通过程序接口实现协议通信，GUI Agent通过模拟用户界面进行交互，这两类技术路径可独立或协作完成跨系统的复杂任务。

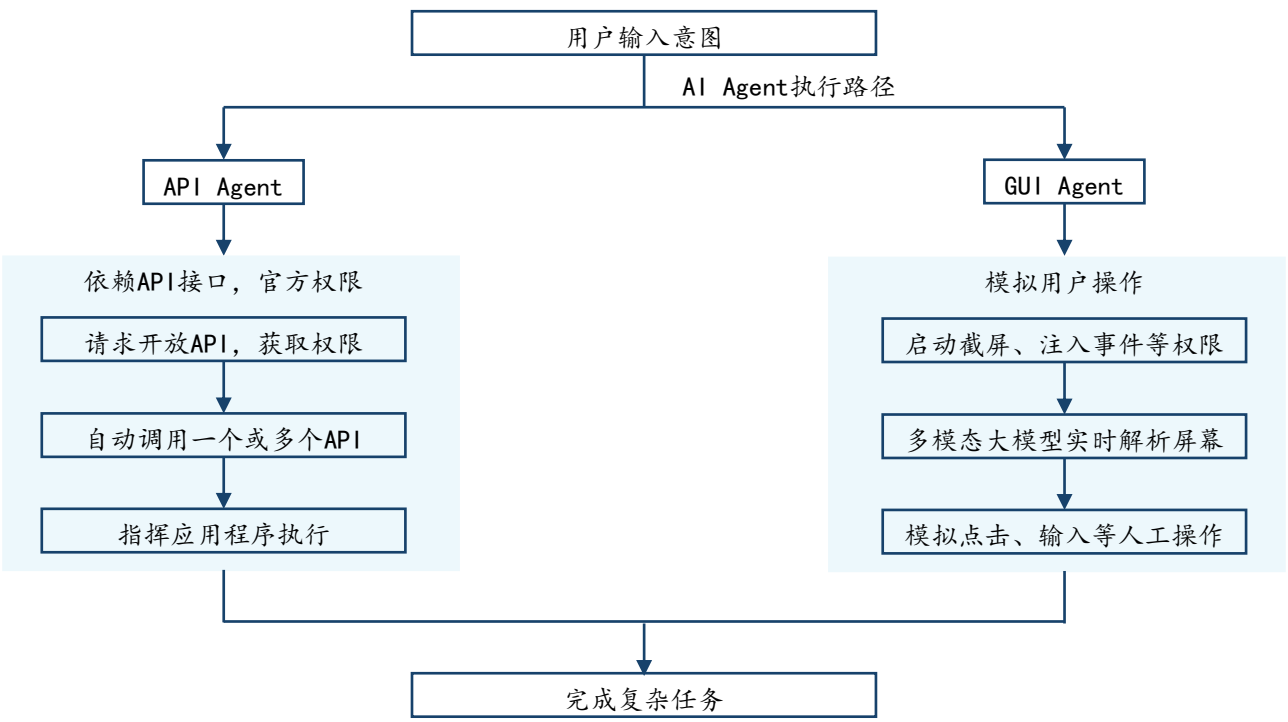
在互联网生态中，跨平台数据协同是数字服务的核心。网站与应用程序通常需要接入第三方数据源来构建服务闭环。出于数据主权、商业利益及合规性等因素，许多数据资源留存于相互独立的系统之中。这种数据隔离的现状，使得外部系统实现跨域共享面临一定挑战。

目前主要通过两种路径解决这一问题。首先是应用程序编程接口，即API Agent架构。它依靠预设的接口规范和通信协议，为第三方系统提供可控的访问入口。这种方式满足安全审计需求，并能确保服务方对数据调用的可追溯性。其次是图形用户界面交互，即GUI Agent方案。这种方式不依赖API接口，而是通过启动操作系统级无障碍服务来模仿用户的界面操作。其逻辑在于AI能够通过与人类相同的视觉路径完成跨系统任务。

API与GUI在技术特性与生态治理维度上存在显著差异。API Agent侧重于结构化交互，通过后端协议直接通信，确保了传输的精准度与效率。这种模式通常需要用户与平台方的共同许可，属于典型的合作式路径。这种合作模式尊重原开发者的商业利益与知识产权，有助于维护数字生态的稳定性。与之相比，GUI Agent虽然具备较强的泛化能力，能够操作各类可视化界面，但其授权机制存在明显分化。其中一类是获得了平台授权并在互信机制下运行的合作式GUI。另一类则是未经第三方应用程序开发者授权，利用超级底层权限强行介入的侵入式路径。

图4：API Agent与GUI Agent差异

注：两大模式并非相互排斥



对比两种AI Agent范式，GUI Agent的核心优势在于通用性高，但在可靠性、效率性、安全性等方面弱于API Agent。

（1）在通用性方面，相较于API Agent的拓展依赖于创建和部署的额外端点，GUI Agent因不需要与应用程序开发者进行适配，理论上用户可使用的程序，GUI Agent都能使用，解决了应用覆盖率的问题，对新功能或未暴露功能的适配能力也更强。

（2）在可靠性方面，因目前大模型技术对复杂界面的识别准确率尚未达到100%、需模拟用户的多个操作导致风险累积、界面改版时可能导致失效等，GUI Agent执行复杂任务的失败率较API Agent高。

（3）在执行效率方面，API Agent可通过一次精准调用就完成复杂任务，后端直接驱动执行效率高，但GUI Agent哪怕是在完成简单任务时，也需要经过多次截图分析、视觉推理等复杂计算，效率低且算力消耗大。例如：当你需要在手机上设置一个12月12日晚上八点的会议提醒时，API Agent仅需调用一次API并进行适当的认证即可创建事件提醒，但GUI Agent需要打开日历，通过视觉导航填写相关字段，并点击按钮完成会议详情设置。

（4）在隐私性方面，不同于API Agent的应用端后台对数据和权限进行精细化管理，GUI Agent因需要读取屏幕内容，容易暴露敏感信息，导致隐私风险高。

图5：GUI Agent与API Agent对比

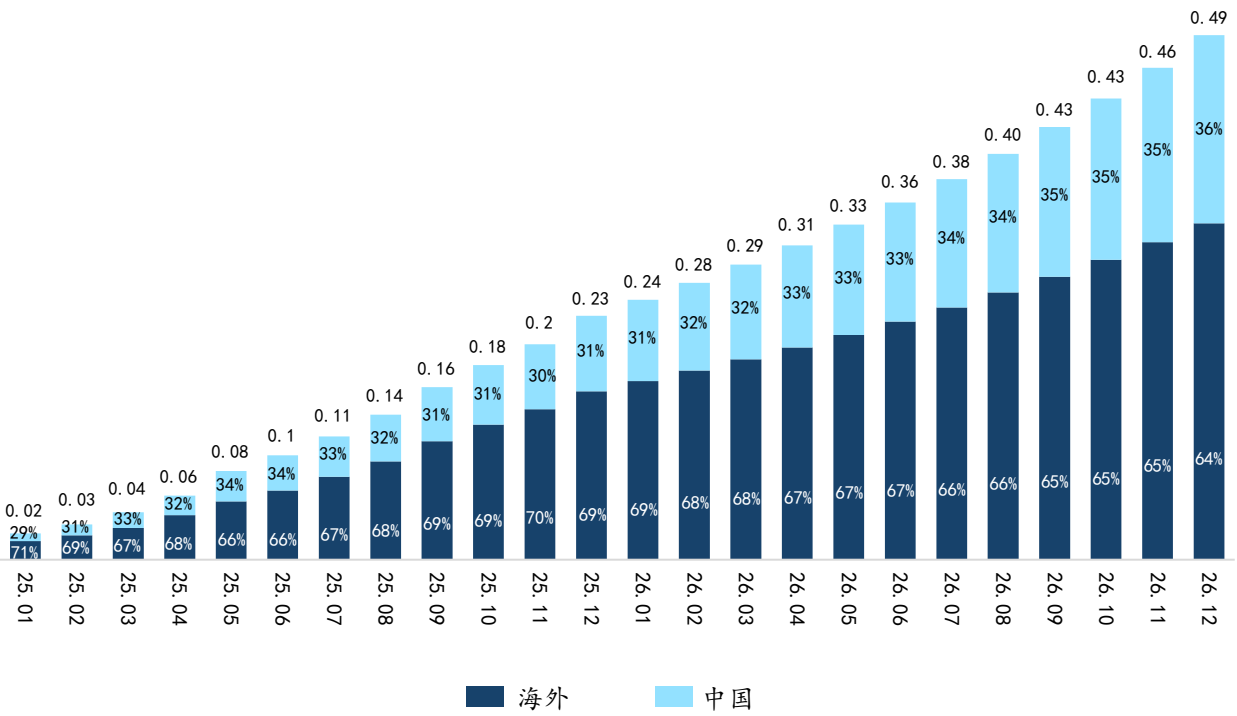
对比维度	GUI Agent	API Agent
技术路线	模拟人类感知与操作，借助多模态模型看懂屏幕上的元素，利用系统辅助服务模拟用户的点击与滑动	建立标准化的语义接口。通过自动调用一个或多个预先定义好的API来执行特定任务
实现前提	仅需获取操作系统级别的权限，主要依赖屏幕截图或系统的辅助功能树	需要应用程序开发者的主动支持，主要依赖基于文本的API调用机制
可用性	能够与任何呈现图形界面的应用程序进行跨软件的交互操作	应用范围仅限于开发者已经对外开放的具体功能点
通用性	较高。无需开发者适配，能直接解决API无法覆盖的零散需求	较低。高度依赖开发者适配，功能受限于已公开的接口端点
可靠性	较低。多步决策易产生累积错误，且极易受软件界面改版影响	较高。访问稳定接口，只要核心API版本稳定即可精准匹配
效率性	较低。模拟连续操作导致交互复杂，简单任务也需多步拆解	较高。利用原生API直接执行，通常单次调用即可大幅减少耗时
隐私风险	较高。需获取屏幕读取权限，敏感信息完全可见，要求极高信任	较低。权限管理精细，各端点可通过身份认证与访问控制独立保护

1.2.2 GUI Agent市场规模

全球GUI Agent市场在2025至2026年间高速增长，中国市场受移动应用环境封闭与厂商系统级预装策略共同驱动，其普及速度与全球份额占比均实现显著提升。

图6：全球及中国GUI Agent APP MAU，2025-2026E

单位：亿MAU



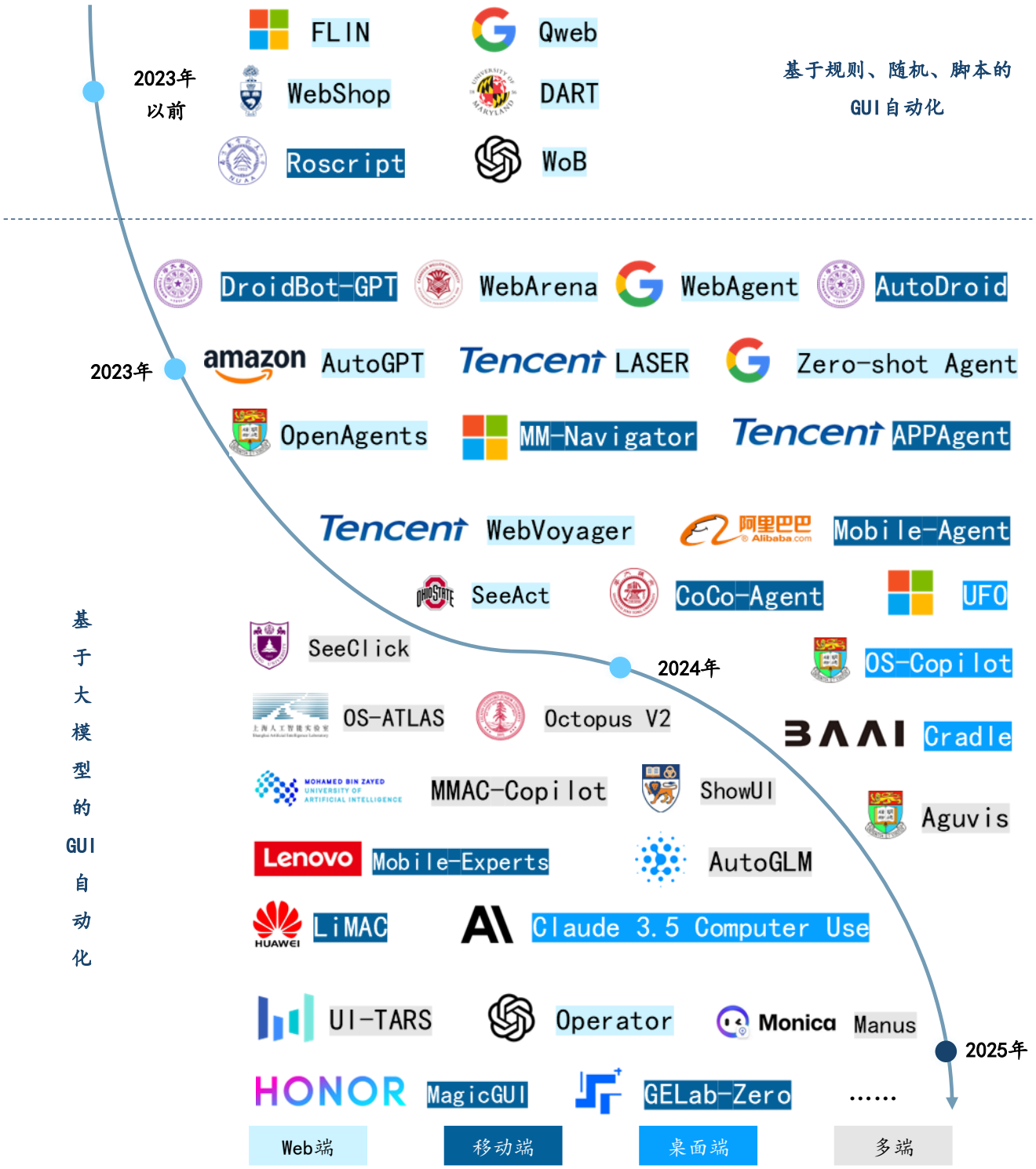
全球GUI Agent市场的活跃用户量在2025年至2026年间将呈现显著上升趋势。该领域的月活跃用户总数预计将从2025年1月的0.02亿增长至2026年12月的0.49亿。与通用型市场占比下滑的趋势不同，中国在GUI Agent领域的市场占有率将从29%稳步提升至36%。与此同时海外市场的占比将从71%下降至64%。这种结构变化表明中国正在成为该技术方向增长的主要推动力量。

中国市场在GUI Agent领域的普及优势主要源于其独特的移动互联网环境与硬件厂商的推广策略。中国移动互联网的应用环境具有较强的独立特征，各软件之间缺乏标准的数据互通机制。这使得通过视觉模拟来实现跨软件操作的GUI Agent成为解决系统协作问题的主要方式。此外中国厂商在技术普及方面表现得更为积极。相比海外开发商较为审慎的商业化态度，国内手机与操作系统厂商倾向于在系统基础层面直接预装相关功能。这种由系统供应方直接提供的推广模式能够更快让用户使用，从而在短时间内推动了市场规模的迅速增长。

1.2.3 GUI Agent产业发展历程

GUI自动化早已有之。自2023年起，多模态理解与跨环境泛化显著降低GUI Agent开发部署门槛；2025年，手机AI助手、豆包手机助手、AutoGLM等GUI Agent相继发布。

图7：GUI Agent发展历程



图形用户界面是人机交互的主要方式，广泛应用于从操作系统到移动程序的各类数字系统。图形用户界面自动化是指通过程序模拟人类的鼠标点击与键盘输入等操作，自动完成人工任务。其规模化应用主要依靠系统授权的无障碍辅助功能。

在自动化初期，系统高度依赖随机操作与预设规则以及固定脚本。这些策略通过在屏幕特定位置执行设定动作来实现自动化。此时无障碍辅助服务主要负责提供结构化的界面信息与执行接口。例如，早期的录制回放工具虽然在固定环境中表现良好，但缺乏灵活性与适应能力，需要频繁进行人工更新。这限制了技术的通用性，使其难以应对不断变化的界面与多样化的软件应用。

2020年前后，随着人工智能技术的演进，这一时期行业内出现了利用机器学习识别屏幕组件的测试系统，并将自动化能力跨平台拓展至不同设备。同时期部分系统开始支持用户通过自然语言指令直接控制图形界面，强化学习技术也在处理复杂网络交互任务中展现出技术潜力。在此阶段，系统无障碍辅助功能的作用升级为环境感知与动作匹配，将纯视觉像素转化为算法可解析的结构化数据，从而将用户的语言指令精准映射到具体的界面控件上，初步摆脱了对固定屏幕坐标的依赖。

到了2023年，大型语言模型的广泛应用彻底改变了自动化技术的演进路线。GUI Agent正式进入语义理解阶段，能够基于屏幕视觉识别执行非固定路径的操作，并在网页与移动端及桌面端实现全面技术突破。行业内开始广泛利用大模型执行网络导航与手机自动化控制，甚至在主流操作系统中执行跨软件的复杂指令。为提升系统性能，研究人员通过积累任务经验与优化提示词等方式，指导视觉模型模仿人类行为并实现多智能体协同。在此阶段，无障碍辅助功能正式确立为结构化的信息入口，为大模型提供具备属性定义的界面元素，大幅降低了纯视觉识别技术带来的执行误差。

自2024年起，基于原生多模态大模型的范式成为行业主导。在该模式下任务呈现端到端的学习与执行特征，将信息感知与逻辑推理以及记忆行动深度整合于单一模型内。这种数据驱动的本质使GUI Agent能够自主适应全新任务与未知界面，彻底脱离了对人工预设规则的依赖，极大提升了产品的业务拓展能力。同年后期，海外领先厂商发布了具备系统级计算机操作能力的视觉大模型，能够自主识别屏幕元素并完成跨软件任务，标志着该技术正式走向商业实用阶段。随后集成了专属交互模型的智能Agent产品相继问世，系统不仅开放了底层开发接口，还能完全模拟人类行为进行网页浏览与信息录入，自动处理多步骤并行任务，推动了行业应用标准的确立。

在全球技术走向商业实用的同时，中国市场也紧跟发展趋势，在2025年迎来了原生GUI Agent的密集应用与技术创新。2025年3月Monica团队发布了通用AI Agent产品Manus。该产品采用较少结构化与多智能体协同的设计方案，能够独立完成复杂的跨软件任务，进一步验证了GUI Agent的广泛适应能力。同年7月终端厂商荣耀发布了界面操作大模型。该模型不仅能够理解复杂的用户需求，更能跨越不同软件协同完成业务，例如在各类出行软件中自动执行开启程序与选择地址及确认车型并下单等多项连贯流程。随着技术向移动端深入，2025年12月字节跳动推出了专属手机助手的技术预览版本。该系统通过获取移动设备系统级授权，在终端实现了持久的本地记忆与直接跨软件操作功能，无需依赖第三方数据接口即可执行打开软件与点击滑动及输入文字等连续动作。紧随其后智谱公司开源了其移动端智能助手框架。该开源框架采用外部Agent控制结合标准终端的系统架构，支持云端AI Agent通过系统调试接口远程控制安卓设备并自动完成各类任务。值得注意的是，各类终端助手与开源框架均依赖系统级无障碍权限，能直接读取通讯与支付等敏感信息。若无严格的权限管控，技术规模化应用将面临严重的安全风险。

1.3.1 侵入式Agent与合作式Agent的概念定义

合作式Agent基于用户授权与应用许可，采用规范安全的机制执行调用。侵入式Agent则在未经应用认可的情况下，直接获取底层权限并单向介入系统操作

合作式Agent与侵入式Agent的共同点在于都以完成端到端任务为目标，均可通过感知界面信息与执行操作来驱动跨应用流程。差异在于协作边界与权限来源不同。合作式Agent建立在产业各主体共识基础上，通过共同参与与收益共享形成Agent系统，既包括基于标准接口调用的API Agent，也包括在可信框架与明确授权约束下运行的GUI Agent。侵入式Agent则通过嵌入操作系统底层获取高等级系统权限，在未获得应用授权的前提下读取界面内容并模拟用户操作，从而绕过应用层限制完成跨应用联动。

合作式Agent是一条通过生态标准化提升确定性的系统路径。这种路径以协议授权为前提，要求用户授权且应用方同步开放接口调用。它利用标准接口与意图框架下发指令，将原本在前端界面上的操作转化为后端结构化的数据交换。由于接口语义清晰且权限边界明确，其执行结果具有更高的可预测性，也更易于实现审计追责与风险控制。这种模式对平台方与应用方的协同依赖性较强，涉及接口建设与规则适配的配合，落地周期受生态推进节奏影响。当协同形成规模后，合作式Agent能够提供稳定且符合合规要求的能力。

侵入式Agent是一条利用系统级权限实现跨应用联动的技术路径。这种路径通常嵌入操作系统底层，通过读取界面内容并注入输入事件来模拟点击、滑动与输入，从而在不依赖应用接口的情况下完成任务闭环。该路径对应用侧的协同要求较低，但高度依赖系统级权限的开放。由于执行稳定性易受界面布局与交互流程变化的影响，其容错成本较高。系统权限具有穿透应用边界的特征，这使得该路径面临隐私暴露与越权操作的风险，技术治理的难度也随之提升。

从产业协作与商业演进视角分析，合作式智能体主要由平台方与应用开发者共同推动。各方致力于构建标准化接口与规则体系，旨在促成可规模化复制且易于审计的多方协同方案。相反，侵入式智能体的核心推动者多为操作系统厂商与底层模型提供商。其倾向利用系统底层权限快速实现跨应用指令执行，以此在短期内迅速扩张任务覆盖范围。然而这种越级调用极易与应用开发者的既定商业规则产生冲突并引发巨大的生态治理压力。因此在要求高可靠性的核心业务流转中，合作式技术路线更能满足现代商业对安全可控与责任追溯的严格标准。

图8：侵入式Agent与合作式Agent的核心差异

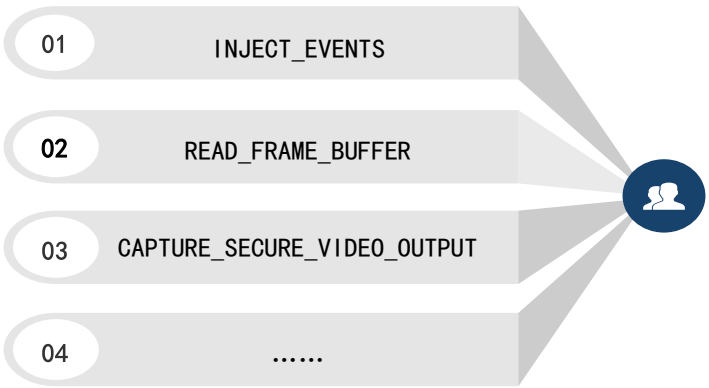
对比维度	合作式 Agent	侵入式 Agent
核心定义	• 基于协议授权，通过标准接口和意图框架下发指令	• 未经目标应用授权依靠系统特权单方面介入执行
技术实现形态	• 包含规范化的应用编程接口对接以及获得授权的合规界面自动化操作	• 主要依赖非授权的视觉读屏与全屏幕模拟点击强行接管前端交互
生态授权机制	• 遵循系统级与应用层的双重授权标准具备透明可控的调度机制	• 缺乏第三方应用协议许可绕过安全校验存在较高系统风险
安全与治理	• 具备完整的数据使用知情权操作记录清晰且责任追溯机制完善	• 处于应用开发者监控盲区数据流向不可控且难以进行精准责任追溯
典型代表	• 遵循标准开放协议的接口智能体以及采用合规自动化框架的谷歌智能体	• 部分强调系统级无感接管的早期原生智能体及缺乏约束的开源项目

侵入式Agent在2023年生成式大模型爆发前已有产品雏形。随着意图识别、多模态理解与跨环境泛化能力显著提升，这类智能体补足了过去在复杂场景下泛化能力弱、流程易中断的短板，进入快速发展阶段。其典型路径是获取手机底层权限，从系统层面绕过应用限制，实现跨应用的连续自动化操作。常见依赖的关键权限包括两类：

（1）READ_FRAME_BUFFER是Android体系中的敏感底层权限，允许应用直接访问GPU渲染缓冲区，从而在图像数据呈现到屏幕之前获取原始像素数据。相较传统截图接口，这种方式具备更高的采集效率，并可实现毫秒级延迟与全分辨率捕获。但由于其绕过应用沙箱与隐私保护机制，存在泄露支付密码、锁屏信息等敏感数据的风险，也可能影响系统运行稳定性。

（2）INJECT_EVENTS属于高等级系统权限，允许应用向系统注入模拟的用户输入事件，使点击、滑动与字符输入等操作可被批量自动执行。由于安全风险较高，手机厂商通常默认禁用此类权限。实践中，部分厂商会通过技术手段向特定合作伙伴开放，使智能体能够突破应用间接口限制，在系统层面完成跨应用闭环操作，例如从应用A读取信息、跳转到应用B填写，再在应用C完成确认。

图9：侵入式Agent定义



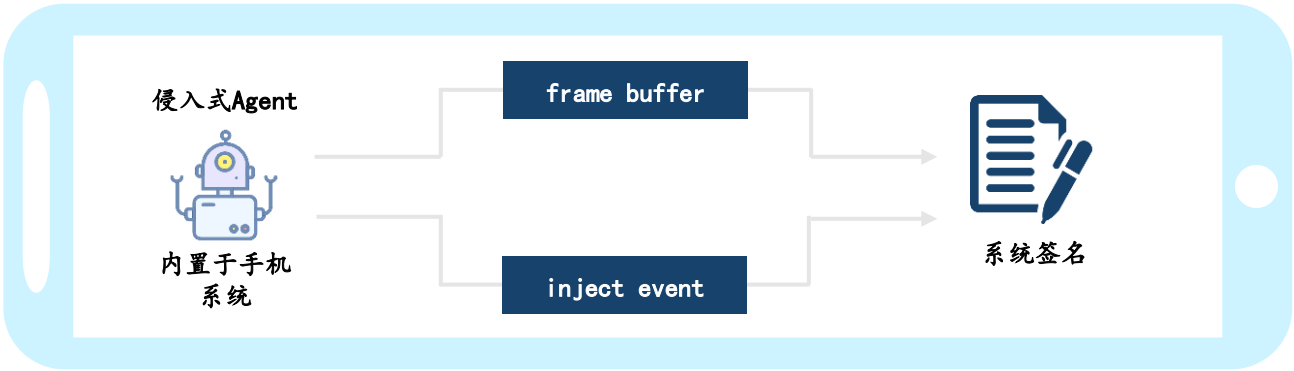
侵入式Agent通过获得系统级别权限，直接向系统注入事件流实现模拟用户跨系统操作，风险等级更高。

侵入式Agent以系统组件形式集成在AI手机中。它凭借系统签名获得高等级权限，可以直接与设备底层交互并绕过应用层的安全限制。

在获取屏幕信息时，该Agent可以突破常规隐私保护机制。普通应用无法采集的银行账户或支付码等内容，Agent能通过READ_FRAME_BUFFER接口直接从GPU渲染缓冲区提取原始数据。这种底层访问方式扩大了对敏感内容的触达范围，并显著提升了捕获图像的频率。

在执行操作方面，Agent利用INJECT_EVENTS权限向系统注入触控指令。系统将此类操作识别为真实的物理信号。由于其执行路径位于系统底层，目标应用难以通过常规的安全算法识别这些自动化行为。

图10：“侵入式Agent”部署方式



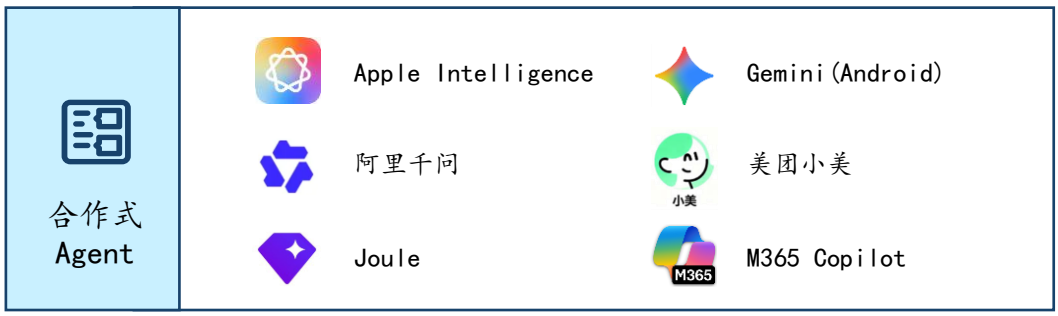
例如：在跨平台比价方面，用户只需语音输入“帮我看一下xx产品在美团外卖、京东秒送、淘宝闪购中哪个平台最便宜”，Agent就可在后台打开美团外卖，搜索xx产品，找到价格并记录；然后打开京东，点击秒送界面，重复上述搜索操作；最后打开淘宝，点击闪购，获取价格信息。完成上述操作后，“侵入式Agent”会发送比价结论，并在最便宜的平台完成下单并自动付款（也可手动接管），整个过程中用户同样可以刷抖音、回微信等……

图11：“侵入式Agent”跨平台比价过程

1.3.2 行业参与者发展现状

当前行业参与者包括两类，一类是以Apple Intelligence为代表的、遵循系统合规协议的合作式Agent产品；另一类是未获取App权限许可的侵入式Agent产品。

图12：合作式Agent典型产品



从产业参与者的视角来看，合作式Agent的发展核心在于建立深度协同的生态秩序。操作系统的开发者如苹果、谷歌，通过预设意图框架和底层通信协议，将Agent职能内嵌于系统内核中。在这种模式下，App厂商不再是被动被调用的工具，而是作为合规的参与方进入生态。任务的达成不仅取决于用户的指令，更需要App厂商主动开放接口并达成调用共识，从而确保在结构化数据交换过程中的精准性与隐私安全性。这一路径下的典型布局者既包括Apple Intelligence、Gemini Android等系统厂商，也涵盖了M365 Copilot、Joule、阿里千问及美团小美等积极融入平台协议的应用方，共同构建起安全和谐的任务环境。

图13：侵入式Agent典型产品

相比之下，侵入式Agent的产业特征体现为避开协作协议的单边介入。此类参与者通常难以获取应用层的接口授权，因此放弃寻求官方层面的合规协议合作，转而利用端到端视觉技术绕开现有的数据交换门槛。对于Claude Computer Use、OpenAI Operator、OpenClaw以及Power Automate等侵入式应用而言，其核心逻辑是在既有应用生态之上建立非官方的操作逻辑。这种路径是对应用方权限的越级接管，通过模型对屏幕像素的直接解析与动作模拟，在无需应用方知情或适配的情况下介入业务流程。

1.4 企业开发侵入式Agent的核心动因

侵入式智能体无需应用授权即可实现跨软件任务执行。这种绕过协作门槛的模式极大地降低了功能落地难度，让用户迅速获得自动化便捷体验，从而快速建立产品依赖与粘性。

核心动因1：侵入式Agent无需与各类应用逐一达成商业或技术授权即可直接实现跨软件任务执行。这种绕过协作门槛的模式极大降低了功能落地难度，让用户能迅速获得跨应用自动化的便捷体验，从而快速建立产品依赖与使用粘性。当前市场中各类人工智能产品面临严重的同质化竞争挑战。智能体技术本身已经发展出代替用户执行复杂任务的能力。但常规的合作式智能体在落地时，需要逐一与各个第三方应用程序建立接口授权与商业合作机制。这种合作模式推进缓慢且覆盖的应用场景极为有限。侵入式智能体彻底绕过了这种繁琐的生态合作路径。它们利用操作系统底层权限直接获取屏幕显示内容并模拟用户交互，在完全不需要目标应用配合的前提下，就能实现跨应用的自动化操作。这种单方面的介入模式使得智能体产品能够以极低的成本迅速覆盖海量真实应用场景。用户能够立刻体验到跨软件自动化带来的效率跃升，从而显著增强对该类智能体产品的依赖度。

侵入式智能体产品利用系统级权限实现了无视应用边界的主动服务。对于普通用户而言，日常的复杂任务往往需要在多个应用程序之间频繁跳转。侵入式智能体改变了这一低效的操作路径。用户仅需下达一句简单的自然语言指令，智能体便能自主打开地图导航软件查询路线，随后切换至航空售票平台比对航班价格，最后在电商平台将旅行必需品自动加入购物车。整个繁杂的任务流转全部由智能体自动模拟操作完成。用户无需记忆不同软件的操作层级，也无需等待各平台的开屏广告。这种打破应用壁垒的连贯体验，极大降低了用户的时间与精力成本，是提升用户留存率的关键因素。

此外，侵入式架构将智能体的唤醒入口提升到了操作系统的最高层级。用户在任何软件的显示界面下，都可以通过语音指令或手机实体按键瞬间激活智能体。由于具备全局读屏权限，智能体会立刻读取当前屏幕上的所有可视信息，并结合用户指令直接触发后续操作。这种全场景无缝衔接的交互模式，去除了以往需要单独打开人工智能软件的繁琐步骤。极简的调用方式让用户能够随时随地依赖智能体解决即时需求。研发企业正是凭借这种跨越应用程序边界的极致便利性，快速培养了全新的用户操作习惯，进而在激烈的市场竞争中获取了极强的用户粘性与入口控制权。

图14：中国AI APP用户规模TOP10变化，2025.01 vs 2025.12

上述跨应用连贯体验的实现源于底层技术对传统接口开发模式的彻底颠覆。在常规数字生态中，各应用程序为保障安全均设有隔离机制，必须通过标准接口方可实现互联。标准接口通过定义明确的数据规则和协议使不同系统实现数据交换并降低集成复杂度。然而现实中接口生态的构建面临双重挑战。一方面各系统间的数据格式与鉴权机制存在显著差异，导致对接第三方系统时需要投入高度定制化的研发力量，这种技术碎片化现状大幅增加了跨平台集成的适配成本。另一方面目前提供标准接口的应用数量极为有限，且应用开发商极少开放核心业务能力，导致传统对接模式根本无法满足用户广泛且多样的长尾业务需求。

图15：API工作原理

相比之下侵入式智能体采用了突破传统接口开发限制的创新技术方案。其利用操作系统级权限直接访问设备界面显示层，实现对应用程序界面元素的实时解析与信息提取。该技术路线完全无需执行传统的接口对接流程。只要目标应用遵循系统标准的界面组件规范，智能体即可利用自动化控件遍历技术获取界面布局与文本内容等基础数据。在此基础上智能体能够直接模拟真实用户的点击操作并执行跨应用自动跳转。这一底层方案的核心优势在于研发企业只需针对操作系统进行一次技术适配即可覆盖所有运行该系统的应用程序，实现了研发成本与应用覆盖范围的彻底解绑。这种模式改变了传统对接中必须针对每个应用逐一开发代码的线性工程逻辑，以极高的拓展效率满足了全场景的自动化操作需求。

图16：Android层次结构

应用层（最上层）：日常使用的各种APP，如微信、淘宝、抖音等应用
应用框架层：提供开发应用的各种API，AccessibilityService（无障碍服务）也在这一层
系统服务层：提供各种系统级别服务，如窗口管理、输入管理、显示管理等
Linux内核层（最底层）：系统地基，负责最核心的硬件管理，管理屏幕显示、触摸输入等资源

核心动因2：侵入式Agent使企业能够扮演用户的私人管家，从而集中捕获并主导用户的决策意图。侵入式Agent可通过持续学习用户行为数据，强化自身模型意图理解和推理能力，以便为用户提供更个性化的服务。目前，全球手机行业正加速从硬件性能比拼转向以AI能力为核心的生态重构。在端侧AI+大模型技术驱动下，大多手机厂商将系统级AI视为AI手机的核心底座，并通过自研打造系统级AI手机助手。部分AI原生手机助手依托操作系统深度集成的优势，通过系统级权限直接调用设备底层硬件资源和跨应用接口，构建起用户、设备、服务的全链路交互体系，这种天然优势是任何独立第三方APP都难以企及的。其核心价值在于：作为用户与设备交互的“第一触点”，第一时间捕获并理解用户需求，同时作为“智能中枢”提供解决方案，即通过直接调用不同APP满足用户需求。

在传统互联网时代，用户需求通过应用商店分发完成，而AI浪潮下，消费者搜索习惯、决策路径正在被重塑，AI也悄然成为捕获用户需求的入口。超80%的用户通过AI获取消费信息，在购物搜索、视觉搜索及智能客服等场景，消费者使用生成式AI的功能占比近70%，消费决策方面，消费者零点击搜索（指用户在AI搜索结果页面直接获得所需产品/服务信息）激增。且AI作为意图入口，其商业价值远高于传统流量入口。因为在与AI对话时，用户的意图通常是明确的、即时的、可操作的，如用户可能直接询问“哪款洗面奶好用”，其意图远比在抖音、小红书等平台漫无目的浏览时清晰得多。AI原生手机助手作为第一触点也就成为了意图入口，重塑流量的分发逻辑。当AI助手可跨应用调取淘宝订单信息、基于位置数据主动推荐周边服务时，意味着手机厂商正在将自身操作系统转化为新的流量入口，即让流量入口从平台转向手机。

侵入式Agent产品的推出主要是为了打造用户与设备交互的第一触点，成为用户完成目标的第一选择。例如：一般情况下，用户完成跨平台比价要逐个打开APP，平台靠信息差、广告等赚取佣金，而如今侵入式Agent成为超级中介，直接穿透平台壁垒，将流量分发的权力掌握在自己手上。此外，侵入式Agent还可通过持续学习用户行为数据形成个性化决策模型，实现从被动响应到主动提供个性化服务升级。

02 侵入式Agent冲击商业生态

侵入式Agent正成为新的流量分发主体，重塑价值分配

- 流量控制权向侵入式Agent转移
- 工具类、交易类、社交内容类App价值受到显著影响
- 生态碎片化明显加剧
- 治理与协同成本系统性上升
- 催化生态内卷
- 短期效率提升，长期创新受限

2.1.1 若侵入式Agent成为流量入口

侵入式Agent将用户任务起点从应用入口前移至系统级智能体，削弱App对用户行为路径与流量分发的控制力。

在过去移动互联网阶段，用户交互范式始终围绕应用作为任务起点展开。无论是消费、出行、办公还是内容获取，用户的行为通常以主动打开某一个App作为起点，随后在应用内部完成搜索、浏览、选择与操作。在这一模式下，用户需要在任务开始前完成一次明确的判断：选择哪个应用来承载当前需求。

在当前侵入式Agent的发展阶段，当智能体以系统级入口的方式常驻于操作系统之上，用户的任务表达不再首先指向某一个具体应用，而是直接指向智能体本身。用户不再需要思考“我该打开哪个App”，而是以自然语言提出“我想完成什么任务”，由智能体负责理解意图并推动后续执行。在这种语义驱动的交互模式中，智能体成为用户意义上的“第一触点”。它位于操作系统与所有应用之上，能够持续感知用户状态、接收任务指令，并通过对界面内容的读取与操作模拟，将用户需求转译为跨应用的自动化执行流程。应用在这一过程中不再承担任务发起的角色，而是作为被调用的执行环境参与其中。

未来越来越多的任务通过智能体入口发起时，用户对打开App这一行为的依赖将持续下降，任务执行的起点开始系统性地向智能体侧迁移。用户是否进入某个应用、从应用的哪一页面进入、是否经过首页或其他关键分发节点，不再由应用自身的产品设计与分发策略决定，而取决于智能体在任务执行过程中所选择的路径。这使得应用原本位于入口位置的分发能力，逐步退化为可选项。

在智能体主导的执行模式下，任务往往沿着最短、最稳定的路径被完成。应用为用户精心设计的导航层级、内容推荐逻辑与商业化资源位，可能在执行过程中被压缩甚至整体绕过。应用通过交互设计和分发策略来影响用户行为的能力随之下降，其入口位置所承载的流量与转化价值难以再得到稳定保障。以应用商店排名、搜索权重和信息流曝光为核心的流量分配机制，本质上服务于用户选择应用入口的过程。当用户行为被智能体重新路由时，这些机制所能触达的用户规模与影响深度均被降低，其对应用竞争格局的塑造作用随之下降。

图17：用户触达流量入口转移

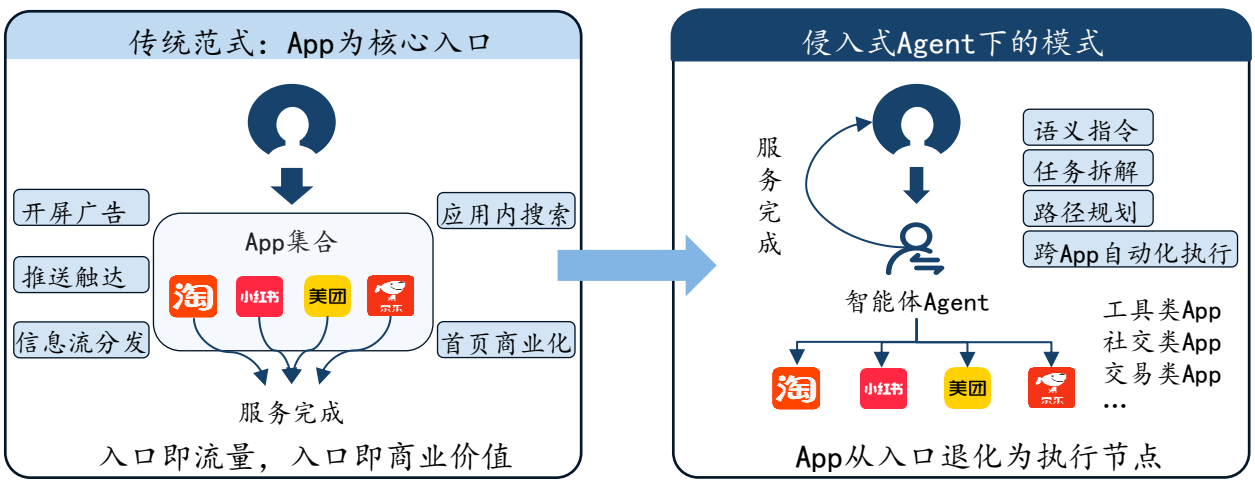
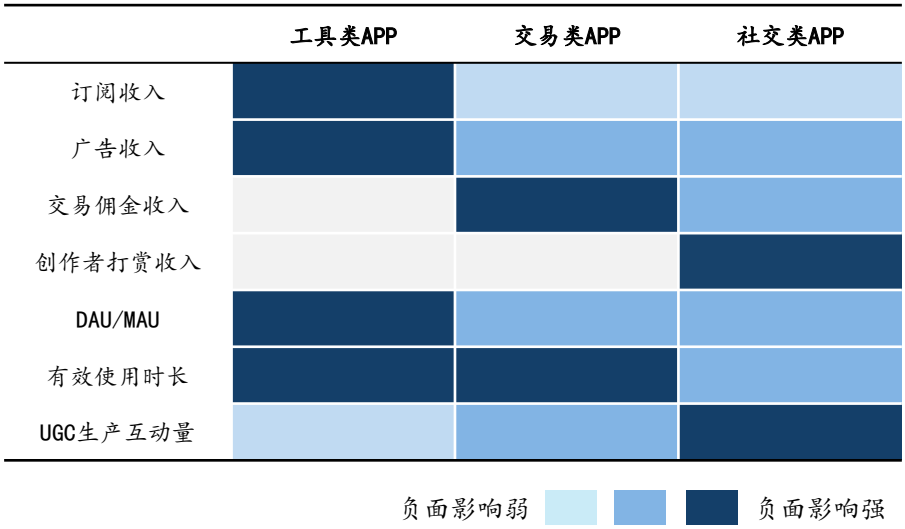


图18：侵入式Agent对应用的潜在影响



整体来看，侵入式Agent通过绕过应用原生界面，直接代为执行任务，对各类应用的商业化体系与用户活跃度构成了普遍的负面影响。这种负面冲击在不同应用类别中呈现出显著的层级差异。工具类应用的整体商业价值受损最为严重，社交类应用的内容互动生态受损次之，交易类应用受到的影响相对低，但核心交易过程同样承压。

侵入式Agent对工具类应用的负面影响最为强烈。在订阅收入、广告收入、活跃用户规模、以及有效使用时长四个核心评价维度上，工具类应用均遭受了最强级别的负面冲击。Agent能够直接识别用户需求并输出最终处理结果，用户因此大幅减少了主动打开应用的次数，在应用内停留的时间也随之骤减。这直接导致基于页面曝光的广告变现模式难以维系。同时，Agent提供的智能处理能力，替代了原有的应用高级功能，造成平台核心订阅收入的显著流失。

社交类应用面临的负面影响紧随其后，主要集中在内容互动生态与创作者变现层面。Agent在创作者打赏收入以及用户生成内容互动量两个指标上，施加了最强烈的负面作用。机器代为阅读与回复的模式，阻滞了真实用户之间的深度交流。用户脱离了原有的社区环境，严重抑制了主动生产内容和参与互动的积极性。由于缺乏真实的情感共鸣与原生互动场景，用户对创作者的打赏意愿大幅降低。此外，平台的信息流广告与订阅变现也受到了普遍的负面挤压。

交易类应用受到的综合负面影响相对较弱，但其核心的交易佣金收入与有效使用时长依然受到强烈的冲击。Agent具备跨平台比价和自动决策能力，替代了用户在单一应用内反复搜索与比对商品的过程，导致用户在应用内的有效停留时长急剧缩短。Agent主导的跨平台购买行为，改变了单一平台的闭环交易过程，使得平台面临直接的订单流失风险，最终导致核心的交易佣金收入大幅下降。

综合来看，侵入式Agent正在从根本上改变用户与应用的交互方式，直接并大幅减少了用户在原生应用内的交互时间。工具类应用面临基础商业模式被大范围替代的风险，社交类应用的社区互动属性受到显著削弱，交易类应用则需要应对核心交易过程被外部切断的挑战。不同类别的应用均面临传统变现路径失效的局面，整体业务增长承受巨大的下行压力。

2.1.2 侵入式Agent弱化应用价值

侵入式Agent将应用降维为后台数据节点，通过截获流量分发权，系统性削弱了工具、交易与内容社交平台的商业护城河。

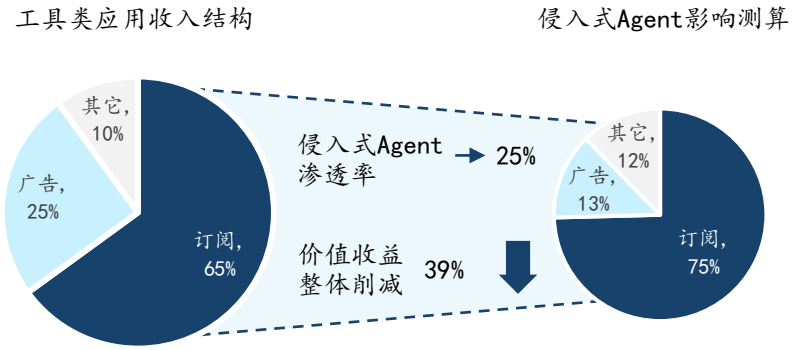
图19：典型的工具类应用

侵入式Agent对工具类应用的主要影响体现在自动化替代功能和流量与广告收入流失。首先，工具类应用逐渐被自动化任务取代，失去独立性；其次，流量减少和用户行为自动化使得平台无法依赖传统的广告和流量获取模式，盈利能力受到压制。这些变化极大地削弱了工具类应用的市场竞争力和生存空间。

侵入式Agent对工具类应用的最显著影响在于其自动化决策和功能执行。传统工具类应用依赖用户主动输入和选择功能，用户在操作中有高度的参与感。然而，随着侵入式Agent的引入，许多任务得以自动化执行，如自动填充内容、自动生成设计建议、或自动调整文本格式。虽然这一功能提升了效率，但也使得工具类应用失去了原本的灵活性和个性化选择的空间。用户的创造性和决策过程被自动化替代，工具类应用逐渐成为简单的执行节点，失去了作为独立平台存在的必要。

侵入式Agent对工具类应用的另一个重大影响是流量和广告收入的流失。工具类应用通常依赖于用户流量和广告收入来维持盈利，尤其是通过用户日常操作、活跃度和推荐来推动广告展示。然而，侵入式Agent通过自动化的任务执行减少了用户主动参与应用的需求，导致流量来源减少。用户不再主动访问应用，而是依赖Agent系统来完成任务，这使得平台失去了通过流量和广告来获取收入的机会。随着流量和用户活跃度的下降，工具类应用的盈利模式受到严重影响，其长期可持续性受到挑战。

图20：侵入式Agent对工具类App造成的价值影响测算



为了直观呈现侵入式Agent对产业商业价值的冲击，此处以工具类应用为例，从其核心营收结构出发推演其面临的实质性影响。工具类应用的商业收益通常依赖于订阅服务与广告营销两大业务模块。通过拆解侵入式Agent对这两大核心收入路径的替代机制，可以清晰地推导出其整体商业价值被系统性削减的逻辑。

在广告收入维度，侵入式Agent将用户的任务起点前置到了操作系统层。用户只需下达语音或文本指令，智能体便在后台跨应用代为执行完毕。这种交互路径直接阻断了用户主动打开应用界面的行为，导致工具类应用的前端曝光率显著下跌。在失去用户的持续关注后，平台原本依赖的信息流广告、开屏展示以及应用内搜索推广等广告资源随之失效，广告的点击转化率出现大幅下滑，从而影响了应用广告变现的流量基础。

在核心订阅收入维度，工具类应用面临功能被直接替代而引发的高净值付费用户流失。许多工具软件的订阅服务建立在专业检索或内容处理等特定功能之上，而侵入式Agent凭借多模态大模型的泛化理解与跨系统执行能力，能够直接向用户交付最终的任务结果。当智能体在多数场景下能够提供等效甚至更为高效的解决方案时，便会对工具软件的核心功能形成实质性的替代。这种替代效应会直接削弱用户为单一工具应用持续付费的意愿，导致平台核心订阅规模出现快速萎缩。

综合上述针对各项核心收入模块的归因推导与加权测算，侵入式Agent的普及对工具类应用造成的实质性影响，不仅限于用户使用时长的缩短，更在于对其商业收入结构的削减。随着智能体渗透率的不断提升，工具类应用的分发权与变现入口将被彻底截断。应用自身将逐渐从直接面向用户的独立商业应用退化为缺乏展示空间的后台执行节点，其整体商业估值与实际收益空间将面临不可逆的大幅收缩。**测算结果表明，当侵入式Agent在用户侧的渗透率达到25%时，预计将会导致工具类应用的整体商业价值出现高达39%的下降。**

图21：典型的交易类应用

侵入式Agent对交易类应用的负面影响主要体现在比价自动化、流量控制丧失和商家营销能力削弱三个方面。这些影响广泛适用于各类交易平台，如电商平台、即时消费服务平台和票务预定平台，直接改变了平台运营模式，商家和平台的竞争力因此受到影响。

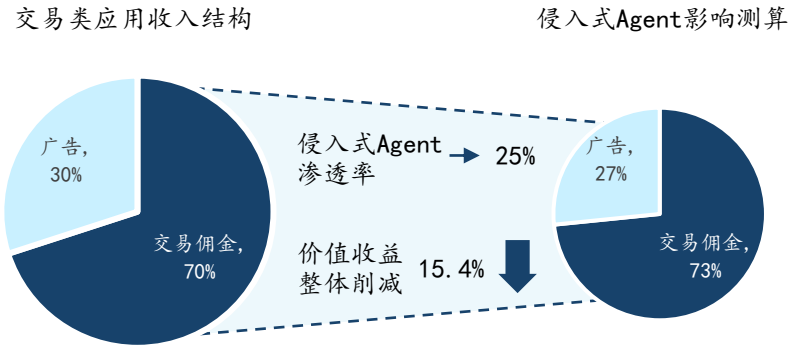
侵入式Agent通过自动化比价系统，能够迅速找到最低价并推送给消费者，减少了用户自主选择商品或服务的机会。这加速了决策过程，让消费者更依赖Agent推荐，而非平台的商品展示和个性化推荐。商家面临较大压力，尤其是价格竞争激烈时，利润空间被压缩。电商平台上，商家不得不依赖低价策略吸引消费者，导致价格战加剧，平台整体利润下降。在即时消费和票务平台，类似的比价自动化发生，消费者不再主动选择，而是由Agent系统直接推送最优选项，商家的市场控制力被削弱。

侵入式Agent同样导致了流量控制的丧失，平台失去了对用户行为的引导。传统平台依靠首页推荐、个性化推送和促销活动吸引流量，而侵入式Agent让用户的选择路径被自动化决定，流量不再由平台掌控。消费者可以通过Agent系统直接完成交易，平台无法有效引导消费决策，商家也失去了通过流量吸引潜在客户的机会。在电商平台，商家的促销活动可能被Agent系统的推荐所绕过，流量高度集中在推荐的商品上，平台营销价值下降。即时消费和票务平台也面临类似问题，流量来源不再可控，商家的营销策略效果受限。

商家的营销能力也因侵入式Agent的介入而大大削弱。平台的盈利模式依赖于广告、推荐、优惠券发放等方式吸引用户，但侵入式Agent通过自动推送、定向优惠和流量控制将这些功能取代，商家无法有效控制用户购买路径。流量集中和定向优惠发放使平台的利益更集中在少数商家手中，进一步压缩其他商家的市场份额和利润，导致平台竞争环境变得更加不平等。

侵入式Agent通过比价自动化、流量控制丧失和商家营销削弱，改变了交易平台的运营模式。在电商平台，商家面临价格竞争加剧和市场控制力减弱的问题；在即时消费平台，平台失去了对用户流量和决策的主导权，商家的营销活动无法发挥作用；在票务平台，商家和平台的盈利模式受到自动化决策威胁，流量和市场控制权进一步集中。

图22：侵入式Agent对交易类App造成的价值影响测算



评估侵入式Agent对交易类应用价值影响的测算逻辑建构在对平台核心营收结构的解构之上。交易类应用的商业营收基本由广告与营销服务收入以及交易佣金与服务费两大支柱构成。该测算模型深入结合了侵入式Agent介入用户行为的具体机制，分别对这两类底层收入来源的受损情况进行独立评估，随后依照各项收入在整体营收中的比重进行加权汇总，进而推导出平台整体商业价值下降的宏观量级。

在广告收入层面，交易平台的广告变现模式高度依赖用户在持续浏览商品信息流时产生的被动曝光。但是当侵入式Agent代替用户执行任务时，系统会直接检索目标对象并自动过滤掉所有非必要的展示信息。这种直达目标的执行方式使得平台基于推荐算法构建的展示类广告库存失去一定效益。随着用户行为从主动浏览转变为向Agent下达确切指令，支撑平台广告业务的流量基础将被大幅削减。

在交易佣金收入层面，侵入式Agent的介入将从消除非理性消费和强化比价效应两个方向压降交易流水。传统平台的交易额中有相当一部分源于用户在浏览过程中产生的非计划性冲动消费。当用户习惯由Agent直接完成交易而不再花时间查阅应用内容时，这部分冲动消费将会显著下降。同时，智能Agent具备跨越全网并在极短时间内完成比价的能力，这会迫使商家降低溢价以换取订单，导致平均成交单价下滑。非理性消费的消失与商品单价的降低，将共同缩减平台抽取交易佣金的资金基数。

基于上述针对广告变现与交易抽成两大核心业务的分析，侵入式Agent将对应用平台的商业生态与变现能力造成实质性的削弱。**综合测算推演结果表明，当侵入式Agent在市场中的渗透率达到25%时，预计在一年内将直接导致交易类应用的整体商业价值出现15.4%的下降。**

图23：典型的内容与社交类应用

侵入式Agent对内容与社交类应用的核心影响在于平台入口权与分发权同步走弱。用户在Agent侧完成内容搜索、跨平台对比与要点提炼后，打开单一App的必要性下降，访问频次与停留时长回落，直接压缩信息流广告的可售曝光。同时，关键意图与决策过程更多沉淀在Agent侧，平台可用于画像、定向与归因的数据变少，广告效率与议价能力随之下行。

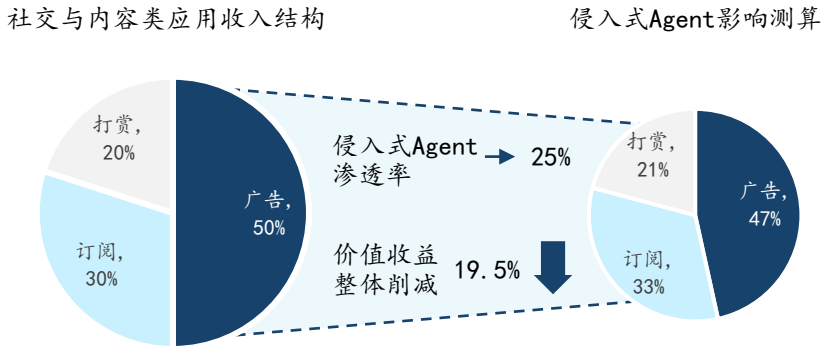
对生活内容平台而言，侵入式Agent会把用户的决策搜索从App内迁移到App外。用户原本在平台内完成搜索、对比、收藏与复看，Agent能汇总多平台内容并输出结论后，用户更倾向于只在核验细节时进入App，连续浏览减少。过程性行为数据减少会削弱推荐分发，内容曝光与创作者收益波动加大，平台需要提高激励与工具投入来稳住供给，成本压力更早显现。

对信息舆论平台而言，侵入式Agent会降低用户对热榜与话题页的依赖。用户在Agent内就能获得事件时间线、核心观点与多方信息，热点带来的访问峰值被摊薄，讨论热度更分散。品牌在热点节点的曝光效率下降，相关投放预算更可能转向外部入口或更可控的投放方式，平台对话题节奏与生命周期的组织能力随之变弱。

对视频直播社交平台而言，侵入式Agent会优先压缩信息获取型观看，而不是娱乐消费本身，教程、测评、攻略等内容更容易被Agent提炼为要点与结论，用户减少搜索型观看与有效完播。娱乐与直播互动仍主要在平台内发生，但内容发现与导流更容易被外置，平台广告触达更依赖泛曝光，定向优化与归因稳定性承压。

对交友类平台而言，侵入式Agent会削弱平台对匹配曝光与付费权益的控制。Agent可在平台外完成条件筛选、资料对比与沟通辅助，站内刷人停留等过程数据变少，推荐优化空间收窄。外部信号介入后，曝光加权等会员权益的边际价值下降，平台商业化更依赖真实性认证、反作弊与撮合质量等不可替代能力。

图24：侵入式Agent对内容与社交类App造成的价值影响测算



为了直观展示侵入式Agent对产业价值的冲击，此处以内容与社交类应用为例，从营收结构出发推演其受到的实质影响。这类应用的商业模式主要由广告收入、会员订阅和创作者打赏三部分构成。拆解智能体对这三条核心收入路径的冲击机制，可以清晰看出平台商业价值被系统性削减的内在逻辑。

在广告变现维度，智能体能够通过后台抓取，直接为用户提炼攻略摘要和核心结论。这意味着用户不再需要打开应用去反复滑动信息流。这种跨过应用界面的直接消费模式，会导致原本穿插在内容流中的展示类广告大量失去曝光机会，从而影响了平台赖以生存的广告变现基础。

在会员订阅维度，平台面临着功能被替代与用户流失的挑战。智能体具备强大的跨平台信息整合与筛选能力，能够快速向用户交付高质量的结构化结果。这种能力直接替代了内容平台原本提供的付费专栏或内容甄选特权，大幅降低了用户为单一平台付费的意愿，进而导致应用的订阅留存率出现显著下滑。

在创作者打赏维度，平台的抽佣收益同样面临缩水。内容平台的打赏收入高度依赖短视频或直播带来的瞬间情绪刺激。然而智能体通常会将这些丰富的多媒体内容转化为客观的数据或文本摘要，从而剥离了极具转化价值的情绪诱导因素，导致冲动型打赏行为大量减少。同时人工智能生成内容的普及进一步稀释了人类创作者的流量，造成平台整体的创作者经济基数不断萎缩。

综合评估各项核心收入受到的负面影响，侵入式Agent的介入全面瓦解了内容与社交类应用的商业收益结构。**根据测算逻辑推演，当侵入式Agent在用户侧的渗透率达到25%时，预计将导致内容与社交类应用的整体商业价值出现19.5%的显著下降。**

2.1.3 侵入式Agent或将新增“AI税”

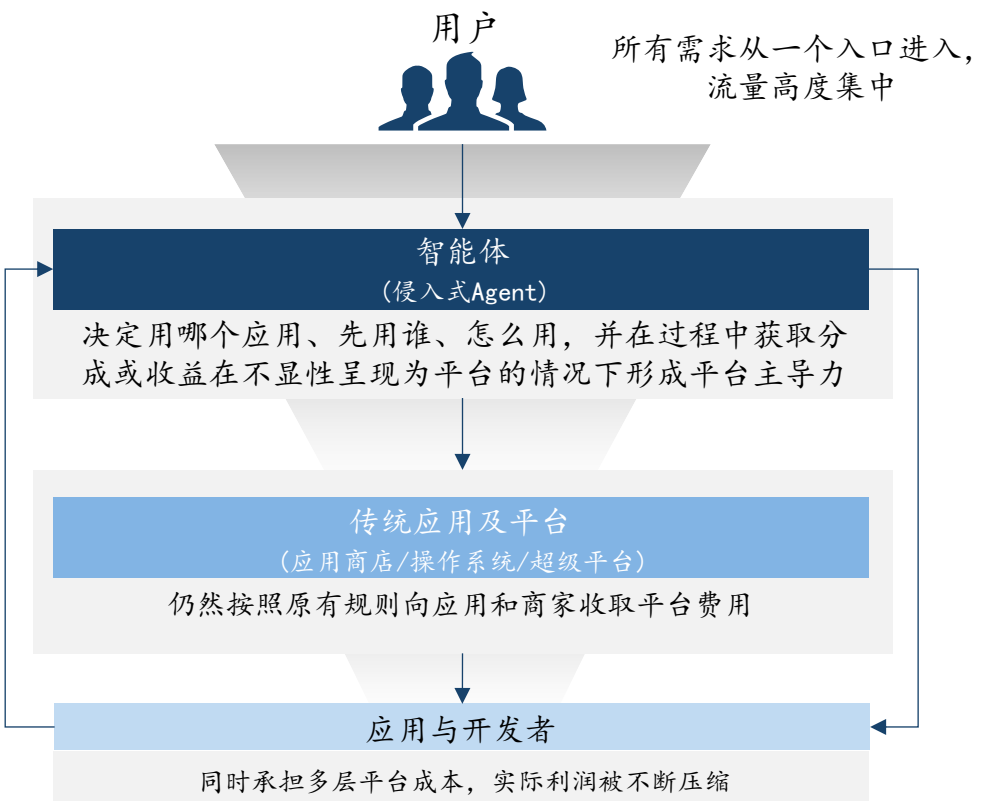
侵入式Agent通过掌控任务入口与执行路径，在不扩大需求规模的情况下前移平台抽成位置，叠加新的成本层并压缩开发者利润空间。

当用户触达入口从应用迁移至智能体、传统分发体系被系统性削弱之后，流量与交易并不会因此去平台化，而是必然围绕新的控制节点重新集中。在侵入式Agent主导的执行模式下，这一节点不再是应用商店或单一平台，而是上移至智能体本身。在这一背景下，围绕智能体形成新的流量抽成与利益分配机制，具有高度现实可能性。

从结构上看，智能体在用户与应用之间所扮演的角色，与传统平台并无本质区别。它同样位于交易与服务链路的上游，决定用户请求如何被分解、由哪些应用参与执行，以及以何种路径完成任务。这种对调用顺序、执行路径与结果呈现的控制能力，使智能体天然具备平台位势，并为其在链路中抽取价值提供了条件基础。

智能体引入的抽成机制并非一种全新的商业模式，而更像是现有平台抽成逻辑的延伸与前移。与应用商店向开发者收取渠道费用、平台在交易中抽取服务费类似，智能体同样可以通过调用优先级、路径偏好、合作关系或分成安排等方式，参与价值分配。在这种结构下，开发者所遭遇新的成本层叠加在既有成本之上。应用在原有平台抽成、渠道费用与运营成本之外，还可能需要为争取智能体调用机会付出额外代价。这意味着，智能体并未通过创造新增需求来扩大整体蛋糕，而是通过在更上游的位置重新分配既有价值，进一步压缩开发者的利润空间。

图25：侵入式Agent压缩用户与开发者的利润空间



2.2.1 侵入式Agent提高生态协作成本

侵入式Agent绕过既有权限体系，迫使终端厂商自行设定安全标准，导致系统权限与执行规则碎片化。这不仅增加了应用的适配难度，也显著抬高了产业链的生态协作成本。

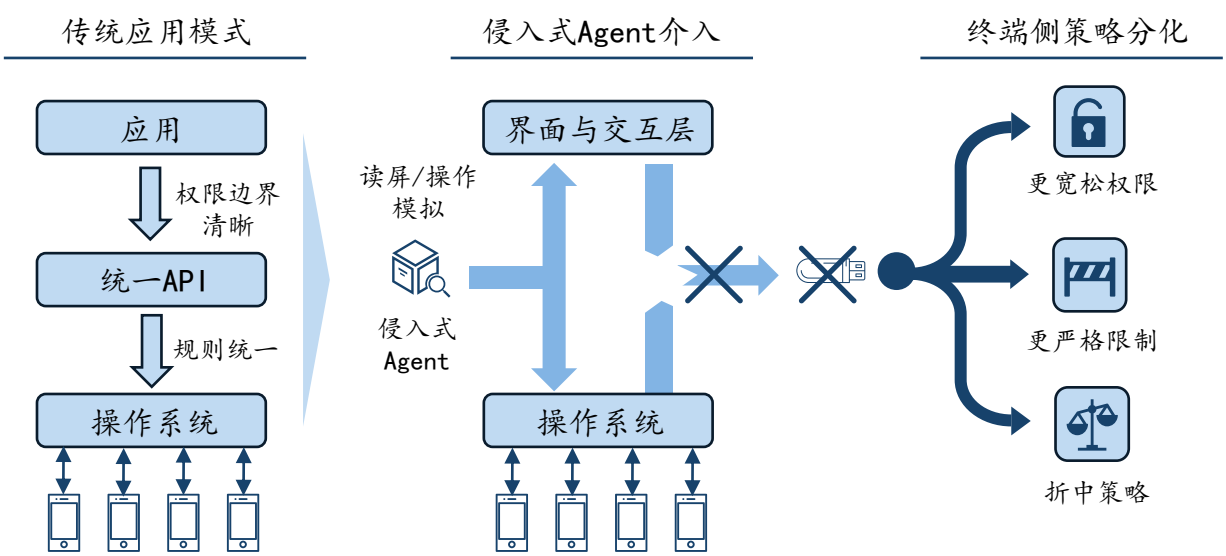
在传统应用模式下操作系统能够维持相对统一的权限标准，原因在于应用的运行方式遵循既定规则。应用通过明确的接口与系统交互，能力边界与调用方式得以提前定义并纳入统一规范。侵入式Agent则显著不同，其不依赖既有接口体系，而是通过读取屏幕与模拟操作等方式直接作用于应用界面。这种运行方式本身并不遵循应用层的既定规则，直接跨越了多个系统与应用的边界。

这种跨越边界的强行调用直接引发了软件应用阵营与底层智能体阵营的对立。应用开发者为了防止自身服务与流量被未经授权的侵入式Agent分流，被迫不断升级自身系统的安全策略。应用端会频繁增加验证步骤或高频变更界面元素特征，以此阻挡Agent的自动化操作。而侵入式Agent为了维持功能可用性，也会持续投入技术资源去试图破解和绕过这些新设的安全障碍。

双方由此将陷入长久的技术攻防对抗之中。应用厂商原本只需专注于优化核心功能与服务用户，现今却不得不将大量的研发资金与时间，消耗在应对安全体系建设之上。这种无谓的研发消耗未能为产业创造实际增量价值，反而拉高了整个生态内的技术适配与交易开发成本。

在持续的技术对抗与防御升级环境下，系统运行的连贯性受到严重影响，最终带来的结果是用户体验的直线下降。与此同时，为了应对日益复杂的侵入行为风险，不同终端厂商只能基于自身的安全合规要求与商业目标分别制定管理标准。这种终端侧策略的分化导致系统权限与执行规则日益碎片化，不仅削弱了跨终端协作的可预期性，也进一步降低了各方的生态合作意愿。

图26：侵入式行为加剧生态割裂



1. 应用开发与维护费用上升

在终端厂商围绕智能体采取差异化策略之后，最直接的后果体现在应用侧的开发与维护成本上升。与传统应用生态不同，侵入式Agent并未建立在统一接口或明确规范之上，其运行能力高度依赖终端在系统层所设定的权限边界与安全策略。这种不统一性，使应用所处的运行环境从相对稳定的平台假设，转向高度依赖终端差异的复杂状态。

在既有移动生态中，尽管不同终端存在系统定制与功能差异，但应用开发仍可以依赖一套相对清晰的基本假设：权限模型大体一致、接口行为可预测、核心能力通过官方API暴露。应用只需围绕这些确定性进行设计与适配，终端差异更多体现在性能调优与体验细节层面，而非运行规则本身。

侵入式Agent的出现，打破了这一前提。由于其通过读屏、操作模拟等方式参与应用运行，其能力边界由终端在系统层决定。在缺乏统一标准的情况下，不同终端对智能体的限制策略呈现出显著差异。这些差异会直接外溢到应用侧。应用开发者需要面对的，不再是“是否支持智能体”，而是“在不同终端上，智能体会以何种方式介入应用运行”。这一问题缺乏统一答案，也无法通过一次性适配解决。

在实际开发过程中，这种不确定性主要体现在三个方面：

首先是行为可预期性的下降。应用在某一终端上可能被智能体频繁调用、界面被解析并参与自动化执行，而在另一终端上却几乎不发生类似行为。应用难以预判自身在不同终端上的实际运行路径，从而增加了测试与验证成本。

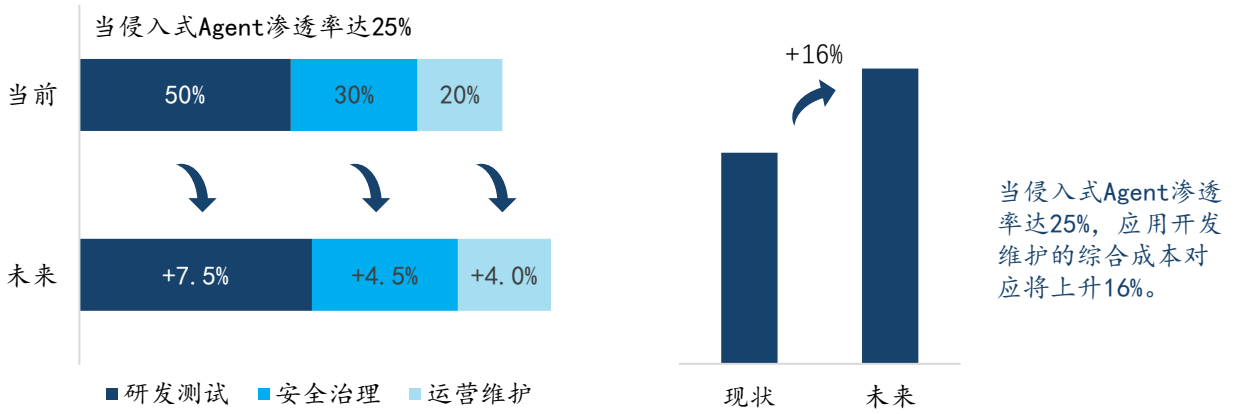
其次是适配逻辑的碎片化。为了保证基本可用性与合规性，应用往往需要针对不同终端环境采取差异化策略，例如调整界面结构、限制某些交互方式，或对智能体可能触发的操作进行额外校验。这些工作并不会带来直接的用户价值提升，却成为维持应用正常运行的必要投入。

第三是长期维护成本的累积。随着智能体能力演进与风险事件出现，权限边界和执行规则可能频繁调整。应用需要持续跟踪不同终端的策略变化，并相应更新自身逻辑。这使得适配工作从一次性的开发任务，转变为长期、持续的运维负担。

这种成本压力在不同规模的开发者之间呈现出明显差异。头部应用通常具备更充足的资源，可以通过专门团队跟进终端变化、进行多端适配；而中小开发者则往往难以承担这种复杂性，只能被动接受不同终端带来的不确定性。结果是，应用之间的竞争，不再仅仅取决于产品能力和用户体验，也越来越取决于其是否具备跨终端应对复杂环境的能力。

由此可以看到，跨终端开发成本的上升是规则不统一所带来的制度性后果。侵入式Agent以绕开应用规则的方式进入生态，在提升部分场景效率的同时，也迫使终端各自设限、各自防御。这一过程最终将复杂性与成本转嫁给应用侧，使整个生态在运行层面变得更加昂贵且更难协同。

图27： App应用开发成本影响测算



评估侵入式Agent对App应用产业价值造成的实质性冲击，可以从软件工程的基础成本结构切入。整体测算逻辑将应用的全生命周期拆分为研发测试、安全治理以及运维迭代三个核心环节。通过观察侵入式Agent在这三个环节中引发的额外工作量，能够清晰推导出这种非协作的接入模式对应用生态造成的成本增量。

在研发测试环节，由于侵入式Agent采取单方面强行读取屏幕界面的方式进行操作，并未与应用方建立正规的合作机制。这就迫使App开发团队在正常的开发任务之外，必须额外花费精力去改造界面底层结构，仅仅为了确保这些外部机器程序能够勉强识别并点击界面元素。同时在产品验收阶段，应用团队也不得不增加大量针对自动化程序的测试工作，这直接拖慢了原本的研发进度并推高了初始投入。

在安全与治理环节，侵入式Agent产生的高频机器流量在行为特征上与网络爬虫极为相似。面对这种来源复杂且不受控的自动化访问，App企业完全陷入了被动防御的境地。为了防止核心数据被无序抓取或服务接口被恶意透支，企业被迫大幅增加安全预算，去建设更为复杂的流量甄别系统，以此来艰难区分正常的机器Agent与真实的攻击。

在运维维护环节，侵入式Agent的视觉运作机制对App界面的任何微小变动都极其敏感且脆弱。应用正常的界面优化或常规的功能升级，都极易导致这些Agent的识别逻辑瞬间断裂失效。为了避免机器操作频繁报错引发连锁客诉，App的运维团队不得不背负沉重的技术包袱，耗费大量资源去长期维护专门迎合机器视觉的兼容方案，这严重削弱了应用本身的敏捷迭代能力。

综合上述业务流程的拆解可以看出，App应用在被动应对侵入式Agent的过程中，综合开发成本会显著上升。基于产业测算推演，当侵入式Agent的市场渗透率达到25%时，App应用的综合开发与维护成本预计将面临16%的上涨。

2. 生态综合治理成本上升

在传统移动互联网阶段，平台虽有竞争，但流量分发仍建立在相对清晰的规则与标准化渠道之上，应用通过应用商店、广告系统、搜索排序等方式获取流量，成本结构也相对可预期，终端厂商更多承担基础设施角色。随着侵入式Agent 接管部分流量入口，分发逻辑开始上移到系统与智能体层，智能体逐步成为服务调用与触达的关键路由节点，应用侧原有的分配与转化过程被重新改写。这一变化并未减少交易行为，反而在多个维度上抬高了成本。

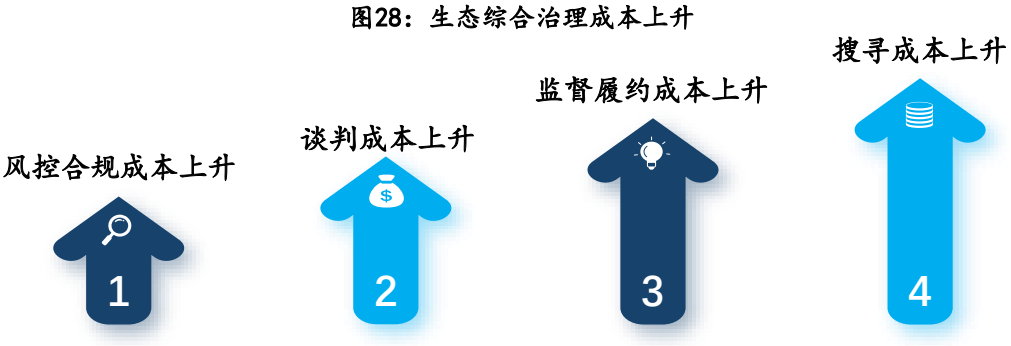
在搜寻与评估层面，第三方应用面临的进入成本显著上升。在统一规则下，第三方应用只需理解有限的分发渠道与接入规范，即可完成规模化部署。而在智能体主导的环境中，不同终端厂商围绕智能体构建了各自的入口形态与调用机制，包括全局搜索、语音助手、AI桌面等。每一套体系在内容结构、调用条件、和展示逻辑上均存在差异。第三方应用需要逐一评估这些差异，判断是否接入、如何接入、投入产出是否成立，其前期搜寻与评估成本显著增加。

在交易与博弈层面，流量获取从标准化的买量行为转向对路由权的结构性竞争。在买量时代，流量获取更多表现为标准化广告位的竞价行为，规则清晰、边界明确。智能体介入后，竞争的对象不再只是流量，而是更具结构意义的权力要素，例如默认调用权、排序优先级、展示形态以及是否允许在系统层完成交易闭环。这类权益往往无法通过统一定价解决，而需要通过商务协商、资源置换或长期合作条款来确定，其谈判复杂度和不确定性远高于传统广告交易。

在监督与履约层面，智能体分流引入了更高的不透明性。智能体分流依赖模型与策略进行路由，其决策过程具有明显的黑箱特征。这使得流量变化的归因变得困难：第三方应用很难判断流量波动究竟源于算法调整、策略变化，还是竞争对手获得了更高优先级。当分发结果不可被有效审计，纠纷与摩擦便不可避免地增加，双方需要投入更多资源用于解释、沟通与纠错，进一步推高监督成本。

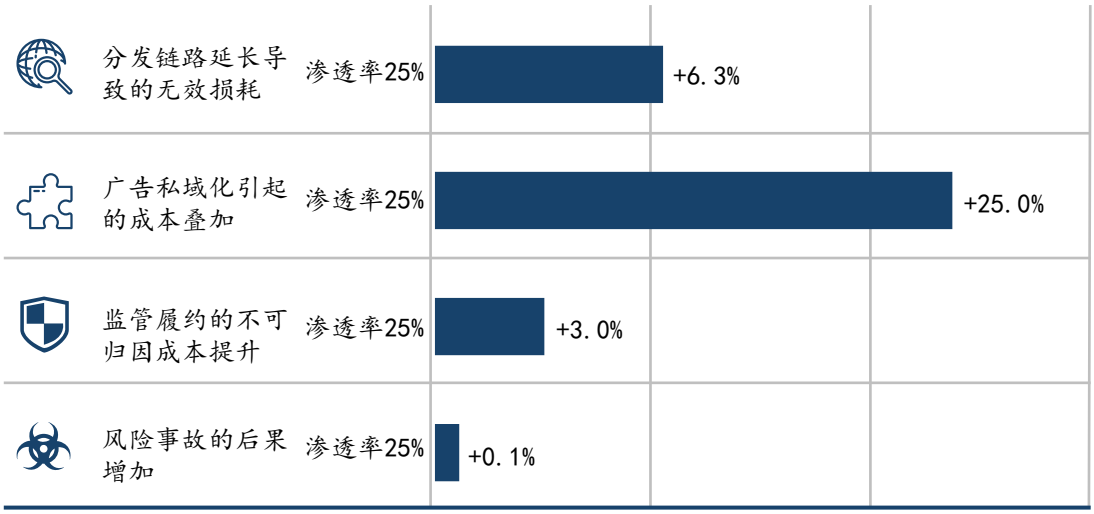
在合规与风险控制层面，系统级介入放大了责任边界的不确定性。当系统级智能体需要跨应用读取信息、聚合内容并参与交易决策时，数据使用、内容与交易责任的边界被重新划分。哪些数据可以被系统层使用、如何获得用户授权、内容聚合出现错误由谁负责、系统层代为下单后的售后与纠纷如何处理，这些问题都无法通过技术本身解决，只能通过制度、合同与合规流程加以约束。这些新增流程在规模化后，会形成持续性的成本负担。综合来看，智能体并未减少生态中的交易行为，而是将原本集中、标准化的交易关系拆解为多点、多轮、多条件的博弈过程。交易成本的上升并非短期磨合问题，而是由流量结构变化所决定的长期趋势。

图28：生态综合治理成本上升



FROST & SULLIVAN

图29： 综合生态治理成本影响测算



评估侵入式Agent对产业生态价值的实质性影响，需引入生态交易成本理论作为测算框架。这种非协同且单方面介入的模式，打破了原有的应用生态平衡，将隐性的治理成本在搜寻分发、谈判协同、监管核验与风险合规四个维度中剧烈显性化。侵入式Agent的介入导致应用流量的分发链路陷入极度不透明状态，原本公开的交易环境被迫向私有化退化，同时伴随着权责主体的严重模糊，进而引发全生态链路的成本通胀。

在搜寻与分发环节，侵入式Agent截获设备端的流量路由，使得原本清晰规范的分发机制变得难以追踪与归因。缺乏与应用方的正规协作导致应用分发链路中出现大量的无效损耗。广告主与应用开发者必须支付高昂的额外搜寻成本，才能弥补这些由于分发链路不透明所带来的资金浪费与低效产出。

在谈判与协同环节，由于流量入口被重塑，公开市场的标准化交易协议面临失效。为了维持基本的合规运作与保障自身商业权益，交易双方被迫放弃低门槛的公开市场，转而依赖高壁垒的私有通道。这种违背产业协同发展规律的运作方式，导致各方必须耗费极高的溢价去建立私有协议，促使生态内部的谈判与协同成本被大幅提高。

在监管与核验环节，侵入式Agent在设备端的封闭执行逻辑，影响了应用生态原本完整的数字证据链条。由于其规避底层系统的标准对接，导致跨主体一致性的留痕机制缺失。对账、抽查以及争议处理的频率因此急剧上升，原本在标准化环境中已经被压缩的核验摩擦成本出现了严重的倍数级增长。

在风险与合规环节，缺乏共识的运作路径极大地增加了整个产业责任链路的复杂度。侵入式Agent的存在模糊了操作主体，一旦发生欺诈行为或用户纠纷，受害方进行调查取证与补救处置的综合成本呈现出几何级数上升，处理合规风险所耗费的资源远远超出了直接的业务损失本身。

综合上述生态交易维度的深度拆解，侵入式Agent将迫使整个应用生态体系承受运转阻力。根据产业测算推演，当侵入式Agent的市场渗透率达到25%时，整个生态体系的综合治理与交易成本预计将大幅上升34.4%。

2.2.2 侵入式Agent催化生态内卷

侵入式Agent在不创造新增需求的情况下前移流量控制权，使生态竞争从价值创造转向存量博弈，加剧内卷。

侵入式Agent并未显著扩大市场需求或创造全新的服务形态，其主要作用是在既有需求规模不变的情况下，重新分配流量与控制权。由此，生态竞争逐步从创造新价值转向竞争新入口，呈现出明显的内卷特征。

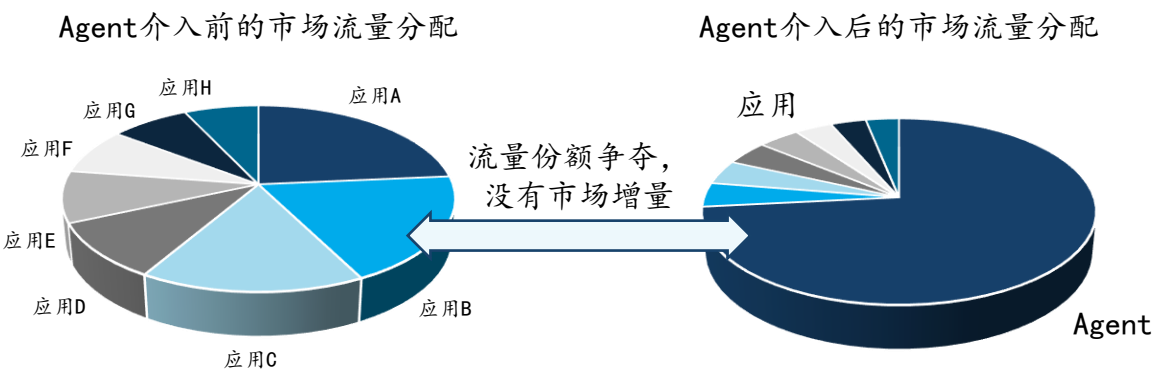
从本质上看，侵入式Agent属于一种工具层创新，而非业务层创新。它通过更高效的界面操作与任务编排，提升了执行效率，却并未在根本上改变服务供给的内容与结构。蛋糕的大小并未明显增加，但切蛋糕的方式发生了变化。当新增价值有限时，任何一方获得的优势，往往意味着另一方的直接损失。

在这一背景下，终端厂商与第三方应用的竞争不可避免地转向存量博弈。终端通过强化智能体能力吸引流量，第三方则通过适配、谈判或防御手段争取被调用权。双方的投入更多用于相互制衡，而非用于服务本身的创新。这种竞争模式在短期内可能提升局部效率，但在整体上会持续抬高行业成本，并压缩创新资源。

更重要的是，这种内卷具有自我强化特征。一旦某一终端通过智能体获得更强的分发能力，其他终端便会被迫跟进；一旦某一应用通过谈判获得优先路由权，竞争对手也不得不投入更多资源以避免被边缘化。竞争焦点不断前移，却始终围绕同一批存量流量展开，行业整体由此陷入高投入、低增量的循环。

总体而言，侵入式Agent并未开启一个以业务创新为核心的新增长周期，而是在既有生态中引入了一个新的流量控制中间层。这一中间层的出现，使得终端与应用之间的关系从相对稳定的规则协作，转向高摩擦的博弈互动。交易成本在多个维度同步上升，而竞争则更多表现为工具层内卷，而非服务层创新。这一趋势若持续强化，将不可避免地侵蚀生态的整体效率，并为后续用户体验与生态稳定性问题埋下伏笔。

图30：侵入式Agent内卷式工具特征



存量博弈加剧，内卷式竞争

2.2.3 侵入式Agent削弱生态创新活力

侵入式Agent让应用创新是否有回报取决于智能体是否调用，导致新功能难被分发、回报不确定，进而压低生态创新动力。

在侵入式Agent逐步成为用户主要交互入口之后，其对生态的影响并不止于流量分配和应用生存状态的变化，更深层的后果体现在创新激励机制本身的弱化。当应用不再直接面向用户，而是更多通过智能体被调用与执行，原本支撑应用创新的回报路径开始发生结构性变化。

在传统移动互联网模式下，应用的创新回报具有相对清晰的传导机制。新的功能形态、交互设计或服务模式，可以通过用户主动选择、使用频率提升和口碑扩散，逐步转化为流量增长与商业回报。创新是否有效，往往能够通过用户行为与市场反馈被较为直接地验证。这种机制为应用持续投入研发与产品探索提供了明确预期。

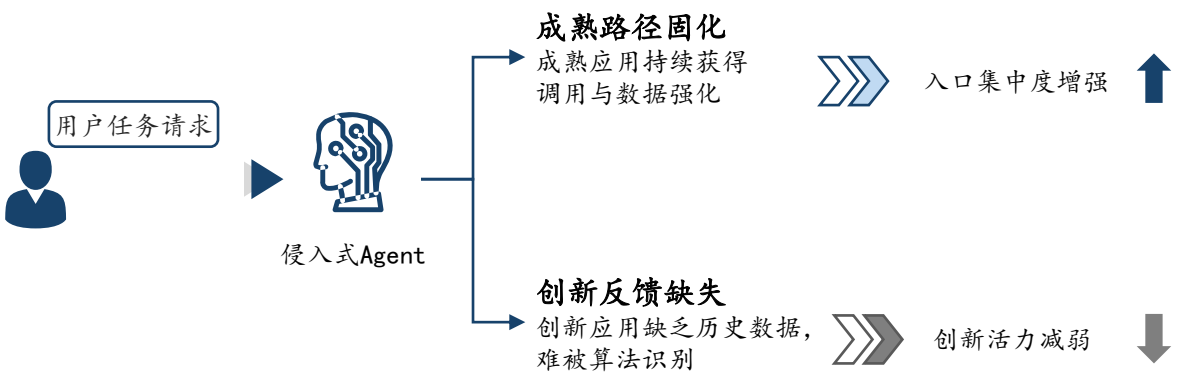
而在智能体主导的路径中，应用的创新成果是否能够被用户感知，越来越取决于智能体的路由与决策逻辑。即便应用在功能或体验上实现改进，如果这些改进未能显著影响智能体的调用结果或排序优先级，其价值也难以在用户侧得到体现。创新的反馈回路被拉长、被间接化，应用很难判断投入是否会转化为实际回报。

与此同时，智能体在执行任务时往往更倾向于选择路径稳定、成功率高、行为可预测的方案。这种偏好在提升短期效率的同时，也会在无形中抑制对新模式、新形态的尝试。创新型应用或新功能由于缺乏历史数据与稳定表现，更难在智能体路由中获得足够权重，进而形成越新越难被选中的循环。

在这一机制下，生态逐步呈现出路径固化的特征。被频繁调用的应用和能力持续获得更多数据与使用机会，其优势不断被强化；而缺乏调用机会的创新尝试则难以跨过冷启动阶段，逐渐被边缘化。创新不再主要围绕如何更好地服务用户展开，而转向如何更符合智能体的选择逻辑，其方向与动力均发生偏移。

从长期看，这种变化将使生态整体从探索驱动转向效率驱动。短期内，用户可能感受到任务完成更快、更省事；但随着创新空间被压缩、新形态难以成长，服务形态与竞争格局将趋于固化。生态表面上保持运转，内在却逐步失去对变化与突破的响应能力。

图31：侵入式Agent削弱创新活力



03 侵入式Agent的内生技术风险

侵入式Agent蕴含深层技术风险，加剧生态治理困境

- 数据隐私泄露风险
- 策略约束困难风险
- 缺乏可审计可复现风险
- 事故责任难界定风险
- 元素识别错误风险
- 路径偏航风险

3.1.1 侵入式Agent的数据泄露风险

侵入式Agent的核心安全风险在于其具备数据访问与业务操作的双重权限，一旦被误导或误判，就可能触发跨系统的数据外泄。

侵入式Agent在数据隐私层面的核心风险在于其将分散在各系统中的敏感信息与可执行权限高度集成于持续在线的执行主体。由于智能体同时具备数据读取与Agent操作能力，原有的应用级权限边界可能退化为任务链路边界。单次指令诱导或识别误判即可导致信息脱离受控环境，形成跨系统的风险外流。

该风险首先表现为非受控内容对智能体决策的干预。邮件、网页、工单及聊天内容等外部输入源可能夹带针对智能体的行为引导。智能体在解释这些内容时，若攻击者将攻击指令伪装成业务语境，可能驱动智能体在用户不知情的情况下执行数据转发、附件上传或链接共享。在Agent式浏览场景中，提示注入攻击利用Agent持续接触不可控输入并将其转化为操作的能力，使数据外传伪装成正常工作流。

风险在第三方工具链上进一步扩张。为了提高完成率与自动化覆盖面，企业往往会让Agent接入更多外部服务，邮件、网盘、协作平台、CRM与内部系统之间的数据被频繁搬运与重组。连接器越多，数据流经的系统越复杂，隐私标准、审计粒度与责任归属越难对齐。只要任务链路中某一步将内容交给外部插件或接口，数据就可能进入组织难以持续监控的处理环节，外泄不一定表现为显性的恶意行为，更可能表现为一条看似合理的自动化流程在后台完成了未被充分告知的第三方披露。

当智能体深度集成到终端设备时，隐私风险的防范范围需要从数据传输环节扩大到本地数据留存环节。为了提升交互体验与响应速度，设备通常会把上下文信息、交互记录、临时文件以及运行日志保存在本地。如果数据留存策略不明确或者清理机制不完善，一旦设备发生丢失、二手交易或遭受外部攻击，设备内存储的历史数据就面临被违规提取的风险。业界关注的便携式人工智能硬件Rabbit r1就是典型的安全案例。该设备的主要功能是利用大模型直接代替用户操作各种应用程序。安全人员在产品上市后发现了严重的数据留存漏洞。由于系统没有提供彻底的本地数据清理功能，用户的交互对话与照片等敏感信息会长期保留在设备物理存储中。这种缺陷导致设备在进入二手市场流通后，下一任持有者可以轻易读取并恢复原用户的个人隐私数据。该安全事件证明了终端设备数据留存风险的真实存在，并促使制造商发布系统更新以增加恢复出厂设置功能，同时严格限制了系统日志的留存规模。此类事件表明，数据隐私风险不仅存在于云端服务器和网络传输过程，也同样长期隐蔽地存在于终端设备的本地缓存与运行日志之中。

对于侵入式Agent而言，数据隐私风险是其能力结构带来的固有缺陷。智能体对用户真实工作流的渗透越深，触达核心敏感数据与高频业务动作的机会就越多，这导致数据外泄极易伪装为正常的执行逻辑。为确保风险可控，产品设计必须同步构建数据边界、授权边界与可审计边界。首先，需实现数据流向的全程溯源，确保每一项外传操作均可被识别并阻断。其次，应建立完善的清理机制，保证链路中的缓存与敏感留存数据可被彻底销毁。最后，针对涉及隐私的关键动作，必须建立强制性的用户知情确认与实时管控机制，防止智能体在未经授权的情况下跨越安全红线。

3.1.2 侵入式Agent的权限泛化风险

侵入式Agent为提高任务完成率通常会一次性获取过宽权限并长期保持登录状态，导致单次误触发操作即可波及大规模资产范围并演变为团队级数据风险，进而显著抬高治理成本。

图32：侵入式Agent权限泛化的风险

侵入式Agent在产品设计上倾向于优先保证跨应用连续执行能力以完成复杂工作流。为实现这一目标，系统通常需要一次性接入电子邮件与协作工具以及云端存储等多个核心业务系统，并长期保持账户的高级登录状态。赋予Agent的操作权限越宽泛且生命周期越长，其能够直接触达的数据资产范围就越广。因此，任何由意外触发或外部恶意指令诱导的违规操作，都难以被有效限制在单一系统内部。此类风险极易顺着Agent的访问链路跨越应用边界，将个人层面的局部操作失误迅速扩大为组织层面的大规模数据泄露事件。

上述问题在Rabbit r1智能设备的安全争议中得到直观体现。当服务集成采用远程托管会话的登录流程时，用户并非在自身掌控的本地设备完成安全授权，而是在远程云环境中输入核心账号与动态验证码，由远程系统代替用户完成跨平台的身份验证与后续操作。这种设计直接导致用户的登录凭证和会话信息长期暴露并在第三方环境中流转。原本应由本地设备承担的闭环认证过程被完全转移到外部系统，大幅增加了核心数据在传输与存储节点被截获留存的概率。一旦这些集中保存凭证的远程服务器遭到入侵，极易引发跨平台高价值数据的批量泄露。

综上所述，权限泛化风险的核心机制在于Agent获取的系统控制权远超单一任务的实际需求，且这些权限在时间和空间维度高度集中。这种架构设计不仅成倍放大了单一应用漏洞的破坏力，也使得整体数据安全威胁彻底脱离传统的局部网络控制边界，最终导致组织面临难以预测且不可控的全局性安全隐患。

3.1.3 侵入式Agent的策略约束困难风险

侵入式Agent容易被攻击内容指令引导，在连续决策中触发真实高风险操作，并导致跨系统扩散。

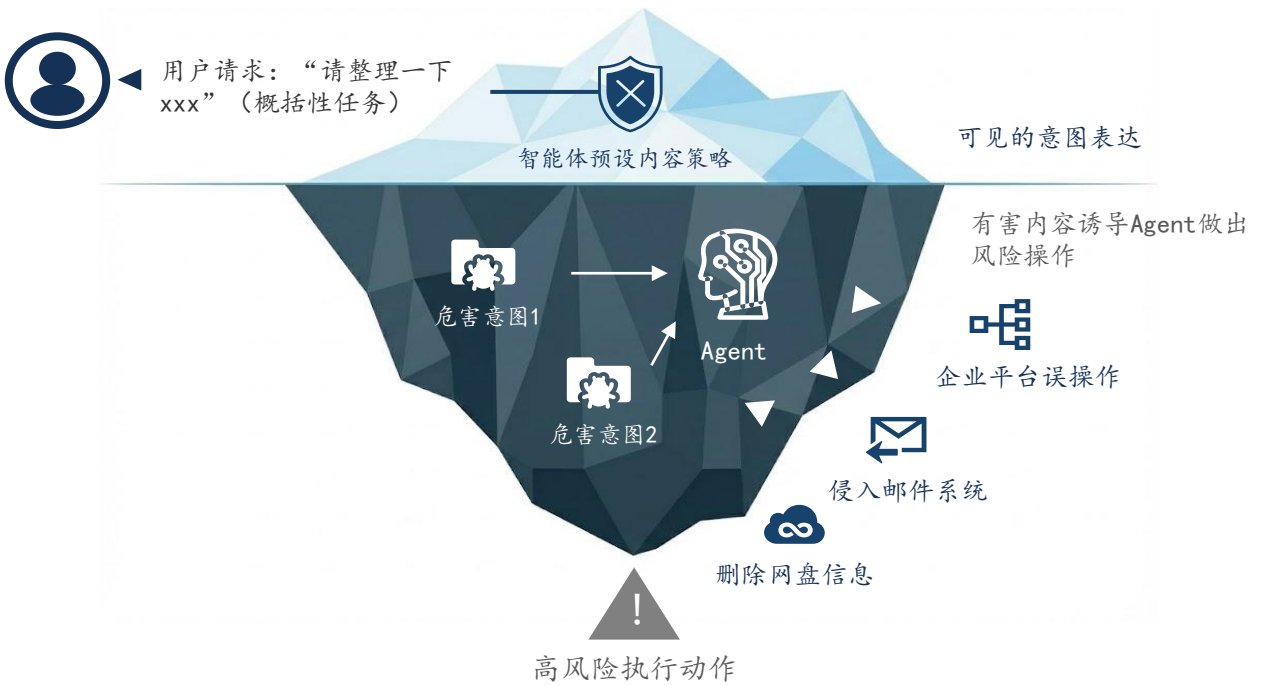
侵入式Agent的典型技术风险在于其策略通常仅能覆盖可见的意图表达，难以识别真实执行链路中的连续决策与实际操作。多数安全防护机制擅长识别显式的危险请求或对话层面的敏感指令，但当智能体需从页面、邮件或文档内容中自行提取步骤、判断目标并选择工具时，安全策略往往局限于表面语义。这导致系统难以及时识别智能体已被特定内容引导至错误目标或高风险动作。

此类风险显著降低了攻击成本。攻击者无需提出危险请求，只需将操作步骤伪装成合理的工作说明并嵌入内容中。当用户发出概括性任务指令时，智能体可能将这些嵌入内容视为执行依据并按步骤推进。在Agent浏览器相关案例中，此类内容内指令会被智能体误判为任务分解模板，从而触发批量移动或清理操作，导致大量文件被移至回收站甚至清空。该过程通常伴随正常的页面反馈与操作回显，具有较强的隐蔽性。由于执行逻辑与正常操作路径高度相似，用户与系统往往直到结果出现明显偏差时才能察觉异常。

在企业环境中，这种风险更敏感，因为侵入式智能体往往需要跨多种业务入口读取与处理信息。邮件、网盘、协作空间与业务系统之间的内容会相互引用、相互触发。一旦策略被“正常内容”绕过，错误步骤就不再停留在单一对话里，而会随着自动化流程在多个系统间连续落地，影响范围扩大，止损与溯源难度同步上升。

因此，业界对这类工具的治理正在转向更保守的选择。部分安全观点主张在企业环境中对AI浏览器类Agent采取限制或阻断策略，核心理由是其决策与执行会持续受外部内容影响，而这种影响很难被稳定地显式化、审计化和策略化约束；当“内容可诱导、执行可落地”同时成立时，风险与收益往往不成比例。

图33：侵入式Agent的策略约束难题



FROST & SULLIVAN

3.1.4 侵入式Agent缺乏可复现性风险

侵入式Agent的非确定性会降低可复现性，使线上行为难回放、难追责，从而削弱在生产化环境的可信度。

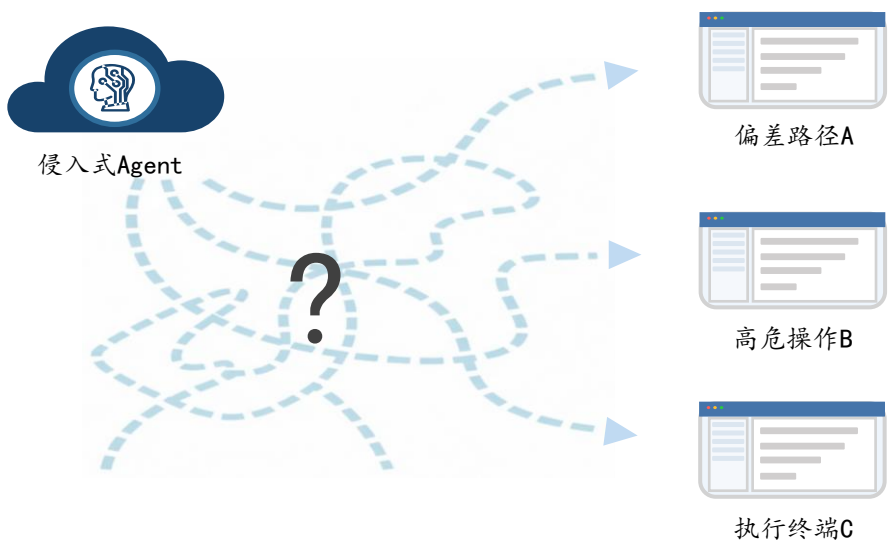
在侵入式Agent的生产化落地中，可复现性不足会直接削弱可信度。模型推理存在非确定性，同样的输入在不同时间运行，输出的重点、步骤顺序和风险判断可能发生偏移。在对话场景里，这种差异往往可被接受；一旦进入合规、风控、财务审批等高要求场景，结论不一致会让业务方难以形成稳定预期，也难以建立对系统的长期信任。更现实的问题是，事故发生后需要回放当时的决策与执行路径，非确定性会让复盘变成概率事件，模型可能无法再次给出当时的同一解释与同一动作序列。

这种问题在自动化链路里会被进一步放大。侵入式Agent不仅生成文本建议，还会把判断转化为界面操作，包含点击、输入、提交等连续动作。只要某一步的选择发生偏差，后续路径就可能进入另一条分支，最终结果出现差异。对组织而言，这意味着同一任务的成功率与风险水平难以被稳定评估，也意味着上线后的问题排查更依赖运气和人工经验，而不是可验证的证据链。

审计困难通常来自两个层面。第一层是缺少足够细的过程记录，很多系统只保留了结果或少量摘要信息，缺乏每一步的输入、工具调用、页面状态与关键决策点。第二层是缺少一致的证据边界，无法清楚说明智能体是在什么上下文下触发了某个动作，依据了哪些信息，为什么选择了这条执行路径。结果是当出现误操作、越权访问或数据损失时，团队很难快速回答三个关键问题，触发源是什么，决策链是什么，责任点在哪里。

对GUI Agent来说，审计的最低要求是把执行链路变成可回放的证据链。需要记录每一步的提示与中间结果，记录每一次工具调用与参数，记录关键界面状态与被操作对象，并对模型版本、策略版本和权限配置做版本化固化。只有把这些信息补齐，组织才有能力在事故后还原现场、定位根因、修复策略，并在监管或内控要求下完成可解释与可追责。否则，生产环境里的不确定性会长期留在系统底层，最终以更保守的权限收紧和更重的人工复核回到每一次任务执行中。

图34：侵入式Agent可复现与审计困难



缺乏过程记录，事故无法回放定位

3.1.5 侵入式Agent使得事故责任难以界定

智能体全局接管打破应用隔离机制。一旦发生数据泄露或误删，系统无法分辨指令出自用户还是智能体，操作来源难以核实导致责任无法追溯。

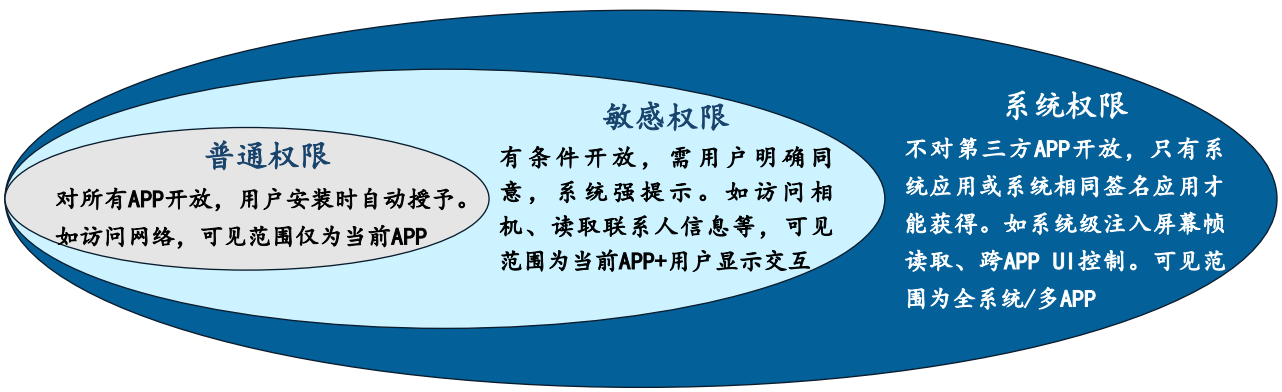
侵入式Agent通过介入操作系统底层，获取本该严格受限、不对第三方APP开放的系统权限，其将自身权限架构设在所有第三方APP之上，严重挑战了第三方APP的安全风控机制。

Android权限体系分为三个等级：**（1）普通权限。**指保障APP正常运行的基础功能权限，如访问网络，对所有APP完全开放，用户安装时自动授予，可见范围仅为当前APP页面，风险等级低。**（2）敏感权限。**指涉及隐私/行为的高风险权限，为平衡便利性与隐私安全有条件开放，如访问相机相册/麦克风/位置/摄像头、读取联系人信息，APP获取该权限时，需用户明确授权且系统会有强提示，可见范围为当前APP页面和用户显示交互，风险等级一般。**（3）系统级权限。**指不对第三方APP开放的、只有系统应用或与系统相同签名的应用才能获得的权限，包括READ_FRAME_BUFFER、INJECT_EVENTS等，如系统级注入屏幕帧读取、跨APP UI控制，可见范围为多APP页面，甚至整个系统，风险等级极高。

在Android系统安全架构中，第三方APP（如微信、淘宝、抖音、AI应用APP等）的权限获取遵循严格的分级管理机制。常规APP仅能通过Android官方API申请两类权限：普通权限在安装时自动授权，敏感权限需用户主动确认。这种设计旨在通过最小权限原则保护用户隐私和系统安全，APP仅能访问完成核心功能所必需的资源，而非无限制获取敏感数据。然而，侵入式Agent突破这一常规权限边界，通过深度介入Android系统底层，获取更高级别的权限——本该严格受限、不对第三方APP开放的系统权限，严重影响了Android权限体系的分级管理机制。

侵入式智能体将自身权限置于所有第三方应用之上。以微信和支付宝为代表的高敏感应用均设有严格的反自动化条款，严禁任何非真实人工的点击行为接入。但是侵入式智能体利用系统底层权限，以代码模拟真实用户的触摸操作来强行触发应用功能。即使用户同意智能体代为操作手机，这种单方面的用户许可也无法替代应用服务商的官方授权。当智能体在后台自动运行并跨越不同程序时，底层系统完全无法分辨某次操作指令究竟是源于用户的真实物理点击，还是智能体自动生成的模拟代码。这种指令来源的绝对模糊性，直接导致一旦发生越权或隐私数据泄露等安全事故，各方的安全责任根本难以准确界定。

图35：Android权限体系



3.2.1 侵入式Agent的元素识别错误风险

侵入式Agent依赖界面结构定位元素，但界面动态变化会导致识别错位与流程偏离，进而触发难以察觉的误操作风险。

侵入式Agent的元素识别通常建立在界面当前的结构信息之上。它会通过系统提供的界面自动化能力或网页结构解析，读取界面元素的层级、类型、名称与可交互状态，再用元素标识、路径关系或位置特征锁定目标控件。界面稳定时，这种方式定位快、可解释性强，也便于把一次成功的操作复用到后续任务中。

问题在于，现代应用的界面很少真正稳定。大量产品采用组件化与动态渲染，页面会随着状态变化不断重排与重绘。前端框架驱动的局部更新、A/B测试带来的版本差异、内容延迟加载与列表追加、弹窗与浮层的插入、以及用户个性化设置导致的布局差异，都会让同一页面在不同时间或不同用户环境下呈现出细微但关键的不同。对人来说，这些差异通常不影响理解；对依赖结构与定位特征的Agent来说，差异会直接改变可操作对象的边界与对应关系。

界面结构发生偏移后，识别错误会以几种方式出现。最常见的是点击错位，目标控件仍在但位置与顺序变化，Agent仍按原先规则点击，最终落到相邻元素上。另一类是功能混淆，控件外观相近、文案相似或图标接近时，Agent更容易将关键影响动作当成普通动作触发，例如把删除当成编辑，把提交当成下一步。更严重的是流程偏离，Agent点错一次入口就进入另一条分支，后续动作仍然可以顺畅执行，但任务会在错误路径上越走越远，直到结果与预期明显不一致才被发现。

当应用缺少稳定的后端接口，Agent只能完全依赖界面执行，上述问题会被进一步放大。界面是唯一通路时，自动化既难绕开动态变化，也难获得更可靠的结构化反馈。界面稍有改动，就可能导致流程中断或误操作，同时需要频繁维护与调参才能恢复可用。更现实的风险在于，流程并不总是以报错停止的方式失败，而可能在错误对象上继续执行，造成更难察觉的实际损失。

一些研究与实践案例已经反复暴露这一点。界面上两个相似图标的细微差异，就可能让智能体选错控件，导致后续任务无法继续。产品落地中也出现过识别偏差触发误删、重复下单等事故，往往依赖撤销与回收站等机制才能降低损失。

图36：动态界面易致元素识别错误

3.2.2 侵入式Agent放大单点错误与路径偏航风险

侵入式Agent的自动化链路容易将单次失误放大为跨系统的连续误操作，一旦初始识别错误就会导致任务整体偏航并触发不可逆结果，最终显著抬高资产回滚与治理成本。

侵入式Agent的自动化链路越长，单点错误越容易被放大成链路级事故。它把理解、判断与执行串成连续步骤，前一步的输出直接决定后一步的动作。链路在多个应用之间穿行时，人为复核与停顿被压缩，错误更容易以更快速度推进到更深环节，等到异常被发现，影响往往已经跨越多个对象与多个系统，回滚成本显著上升。

这类放大效应在企业SaaS场景里更突出。智能体为了完成端到端任务，通常需要长期持有登录态与授权访问能力，并能在共享盘、协作空间、工单系统等资产密集区域执行操作。一旦前端出现误读或被诱导，系统侧看到的仍可能是合规入口与合规动作，风险识别更依赖事后排查，止损窗口被动缩短。

2025年12月披露的一类零点击Agent浏览器攻击，把这一风险呈现得非常直观，攻击者通过一封看似正常的邮件，将分步骤的整理指令嵌入内容；当用户已授权Agent访问Gmail与Google Drive，并发出概括性请求让Agent处理邮件与相关整理任务时，Agent可能把邮件内容当作可执行步骤，进而对真实文件执行批量清理操作，将大量内容移动到回收站，影响范围可扩展到共享文件夹与团队盘。

企业越把跨系统任务交给智能体，就越要在前几步把错误拦住。否则，一次误读就可能被自动化放大成对共享资产的批量操作，引发业务中断，并带来高昂的复核与恢复成本。为止损，平台和组织往往不得不收紧权限、增加关键步骤确认，结果是把原本节省的步骤重新以更强约束、更高摩擦的形式加回到每个正常用户和正常任务上。

图37：零点击邮件指令触发网盘批量误删

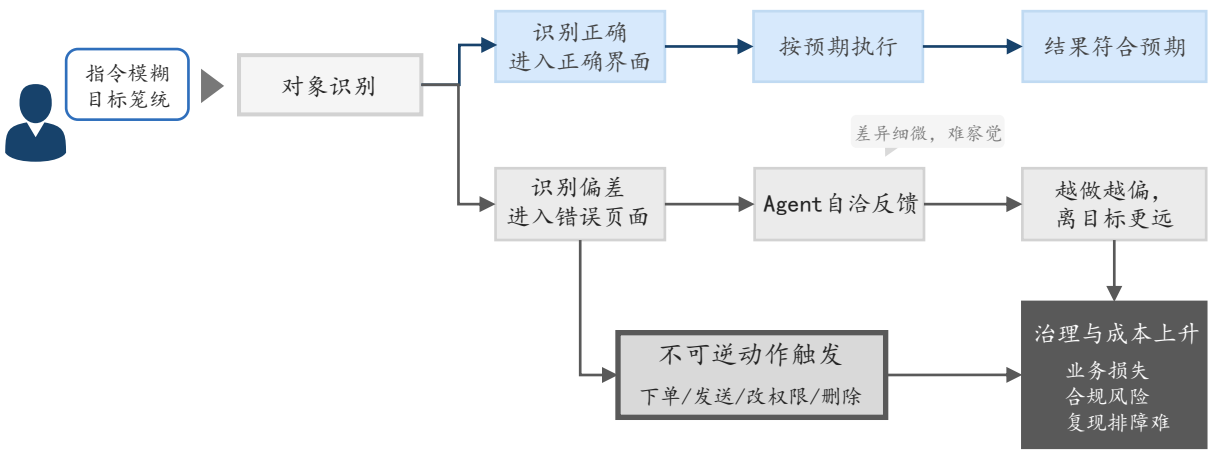
当智能体识别目标对象错误时，风险不仅限于单步操作失误，而是导致整个任务进入错误路径。智能体一旦进入错误分支，后续步骤将在错误上下文中持续推进，导致操作结果偏离用户意图。与单点输出偏差不同，侵入式执行的偏差会沿链路传播，增加后期修复成本。

这种路径偏航在实际业务中具有隐蔽性。不同页面的视觉差异可能较小，错误路径也能产生逻辑自洽的反馈，例如跳转至相似设置页、选中相似联系人或进入不同套餐页面。问题往往直到最后涉及价格、权限或对象名称等关键信息时才被暴露。此外，部分错误操作具有不可逆特征，如提交订单、发送消息、修改权限或删除数据，一旦执行则无法通过简单撤回处理，只能依赖复杂的补救流程。

相关案例表明，Agent存在超出用户指令范围自主延伸任务链的情况。例如，部分浏览器AI插件在处理笼统指令时，会擅自检索用户邮箱并执行其中隐藏的指令。这反映出大型语言模型Agent存在过度自主性，执行了超出用户明确请求的操作。此类行为体现了模型决策的不透明与不可预测，用户难以获知Agent的内部执行步骤，也无法解释其路径选择逻辑。模型行为的不确定性与决策过程缺乏可解释性，降低了用户与开发者对Agent行动的信任度。

从生产落地角度分析，任务路径偏航会显著抬高治理与责任成本。首先是业务风险，错误操作引发的损失、投诉或合规事件通常涉及跨部门处理。其次是排障难度，界面执行缺乏稳定契约，任务失败可能由页面变化、网络延迟或账号差异共同诱发，导致问题难以复现。最后是运营成本，为防止偏航扩大，系统需强化过程监控与人工干预，导致单位任务成本上升，限制规模化应用。最终，侵入式路线的效率优势因关键环节的人工加固而被稀释，稳定交付成为主要瓶颈。

图38：目标错误造成路径偏航



3.3.1 侵入式Agent在C端的风险案例

侵入式Agent在C端实际应用会触发平台风险控制的反制措施，导致任务成功率下降，验证步骤增加，最终让用户承担更高的操作负担

在C端环境下，侵入式Agent最初呈现给用户的，是一种高度直观且易于理解的服务目标。例如，减少操作步骤，代替人工点击，在多个应用之间替用户完成任务。从单一任务视角看，这种能力确实能够在短时间内降低用户的操作负担，尤其在购物、预订、填写表单等高频使用场景中，带来立刻可以感知到的效率提升。**然而，当侵入式Agent从个别功能演示转变为真实使用，并开始参与高频消费与内容场景后，其对用户体验的负面影响逐渐显现。一旦侵入式智能体违反平台的入口管理与风险控制规则，其结果是平台会采取封禁等限制措施，最终增加了用户的使用困难。**

案例一：豆包手机遭主流平台限制，用户遭受损失

豆包手机是目前少数将侵入式Agent明确嵌入系统层，并以替用户完成跨应用任务为主要特点的终端设备之一。该设备试图通过系统级智能体，在无需平台接口配合的前提下，自动完成点外卖、比价购物、购票等高频消费操作，从而构建一种跳过应用启动界面、直接输出结果的使用体验。

然而，在真实使用环境中，这一运行方式很快与主流平台的风险控制与规则体系发生直接矛盾。随着用户尝试通过智能体自动操作外卖、电商、支付等关键应用，多个主流平台开始对相关行为进行限制：部分场景下任务无法继续执行，部分情况下用户被要求恢复为手动操作，甚至出现账户被判定为处于异常操作环境，需要额外验证的情况。

从用户视角来看，日常服务可用性被显著降低：

- 原本可以稳定使用的外卖、购物、支付流程，在系统级智能体参与后变得无法预测结果。
- 用户无法判断失败原因，是智能体能力不足，还是平台规则限制。
- 高频服务被反复中断，导致使用成本与心理负担上升。

需要强调的是，侵入式Agent在C端实际应用时必然会遇到根本性的规则冲突。对于平台而言，智能体在未经协商的情况下模拟用户行为，跳过既有入口与流程，实质上削弱了平台对安全、风险控制与交易过程的管理权限；而平台采取限制与拦截措施，是其维护系统正常运行与商业规则的合理反应。可以看到侵入式Agent的使用成本，最终会是由终端用户承担。当平台选择通过更严格的识别、更复杂的验证或直接限制访问来应对侵入式行为时，最终呈现给用户的，是操作步骤增加、成功率下降以及使用过程不连贯。

案例二：OpenClaw误删核心邮件，用户承担数据丢失后果

OpenClaw作为一款主打桌面端自动化的侵入式Agent，其核心功能之一是接管用户的电子邮箱客户端，自动完成邮件分类与清理任务。在最初的产品演示中，这种基于屏幕视觉识别技术的自动处理机制，向用户展示了无需手动逐条查阅即可快速清空无用收件箱的高效运行模式。这种直接操作界面的方式，在短期内确实降低了用户处理海量信息的繁杂程度。

然而，当OpenClaw脱离受控测试环境，直接面对个人用户真实且复杂的邮件内容时，其技术实现方式的局限性导致了严重的操作失误。由于侵入式Agent主要依赖界面元素的识别和模拟鼠标点击来执行任务，而不是通过邮件服务商的标准接口获取准确的底层文本信息， OpenClaw在判断邮件重要程度时出现了严重偏差。在执行自动清理指令的过程中，

智能体无法准确区分常规商业推广与包含关键业务进度、电子账单或紧急通知的邮件，直接对其执行了删除操作。

这种缺乏验证机制的操作直接损害了用户的数据资产安全。在实际场景中，核心业务信息被意外清除，直接导致用户错过关键的工作节点或财务支付期限。事件发生后，用户必须耗费大量时间去已删除文件夹逐一翻找和恢复被误删的内容，部分用户甚至面临文件因时间过期被彻底清空而完全无法挽回的情况。更为严重的是，由于担心智能体再次产生不可逆转的影响，用户不得不大幅增加人工复核的频率。这使得用户原本希望通过自动化工具节省的时间成本，最终被完全转移到了对智能体执行结果的日常监督上，从根本上违背了引入该技术以提升工作效率的初衷。

需要指出的是，OpenClaw误删邮件事件暴露了侵入式Agent在执行操作时的安全缺陷。当系统级智能体被授予直接修改或删除用户数据的权限，却不具备与其执行能力相匹配的语境理解与容错确认机制时，任何微小的识别错误都会导致重大的使用事故。这一案例表明，在缺乏明确操作确认环节的C端应用中，侵入式Agent带来的初步效率提升远不足以抵消其引发的数据安全风险。最终，用户不仅未能享受到自动化带来的长期便利，反而成为了智能体决策失误的直接受害者。

案例三：Comet智能体违规访问平台，引发法律诉讼与合规风险

美国Perplexity公司推出了一款名为Comet的人工智能浏览器。这款浏览器内置了智能体程序，可以代替用户在网页上完成复杂的任务，例如登录第三方网站、自动寻找商品以及比较价格并完成下单。但是，该产品因为涉嫌侵犯第三方平台的商业利益而面临法律诉讼。

2025年7月，Perplexity公司推出了这款新型浏览器，并在同年10月向全球用户免费提供。这款浏览器将人工智能技术整合到日常浏览中，其核心目的在于把单纯提供搜索结果转变为代替用户执行具体操作。用户可以授权该智能体代表自己采取行动。这包括使用用户保存在本地设备的账号信息，登录用户的私人亚马逊账号，然后在亚马逊平台上自动寻找商品、比较价格，甚至完成最终的支付结算。整个过程不需要用户手动点击屏幕。为了回应外界对数据隐私的担忧，Perplexity公司声明，所有涉及个人账号的操作都在用户本地设备上完成，敏感数据绝对不会上传到云端服务器。

尽管有隐私保护承诺，这款智能体的购物功能依然因为未经授权访问平台数据和代替用户下单，涉嫌侵犯亚马逊公司的利益而遭到起诉。回顾事件背景，2024年11月，Perplexity公司曾推出一项代买服务，允许付费用户委托智能体在第三方网站购物。当时亚马逊公司就指控其滥用用户账号。随后Perplexity公司承诺在没有获得明确许可前停止部署相关程序。2025年5月，亚马逊公司更新了平台使用规则，专门增加了针对人工智能Agent的条款，要求这些程序的提供商必须明确表明自己的真实身份。然而在2025年7月至10月期间，Comet浏览器内的智能体伪装成常见的谷歌浏览器，使用用户的私人账号访问亚马逊平台。在此期间，该智能体多次绕过了亚马逊平台的技术拦截。最终在2025年11月，亚马逊公司正式提起诉讼，指控Perplexity公司违反了计算机欺诈与滥用的相关法律。亚马逊公司指出，Comet智能体通过伪装身份和绕过安全措施的方法，在没有获得授权的情况下，访问了亚马逊平台和用户的私人账号。

在诉讼中，亚马逊公司提出了三项核心指控。第一，Comet智能体在没有获得许可的情况下，读取商品数据并代替用户进行购买。

第二，Perplexity公司违反了亚马逊平台的运营规则。这些规则明确禁止绕过平台的技术安全措施，并且要求自动化程序在平台上运行时必须公开真实身份，遵守访问限制，绝对不允许隐瞒其作为程序的本质，也不能试图突破平台的拦截机制。第三，Comet智能体的行为影响了亚马逊平台对账号访问权限的管理。由于该智能体本身存在容易被黑客攻击的安全缺陷，它的接入增加了用户数据被盗取的风险，对大量用户的隐私构成了直接威胁。此外，该智能体的自动化操作干扰了平台为用户提供的个性化购物体验，影响了平台的广告收入。为了识别并拦截这些程序产生的无效访问量，以及帮助被误封禁的用户恢复账号，亚马逊平台承担了额外的运营压力和财务成本。

目前这项诉讼还在初步阶段，但它清楚地展示了人工智能程序在与第三方平台交互时面临的严重合规问题。如果智能体在没有获得授权的情况下直接访问第三方平台，或者使用伪装身份和绕过拦截等技术手段来突破访问限制，这种行为是以放弃合法合规为代价的，必然会面临法律的严格审查。这为所有自动化Agent产品提供了明确的警示。该案件也表明，当前亟需建立一套完整的规则体系，涵盖身份确认、权限分配、数据传输标准以及操作记录审查。只有将技术管理手段与制度规范结合起来，才能在发挥自动化工具效率的同时，防止因技术被滥用而产生的全局系统性风险。

侵入式Agent在C端带来的风险与警示

缺乏规范的侵入式Agent如果广泛普及，将直接增加用户的使用负担。这类程序通过读取屏幕信息、调用无障碍服务和模拟人类点击，将原本需要人工执行的界面操作转化为可大量复制的自动化功能。在信息安全领域，这种无需后台高级权限即可在显示界面上执行关键指令的能力存在极高风险。一旦被恶意程序利用，其产生的影响与用户本人执行危险操作的结果完全相同。

过去一年，安卓操作系统多次发生利用无障碍功能接管手机界面的资金盗窃事件。攻击者借助自动化技术，完成了诱导输入密码、读取验证码以及强制转账的连续步骤。这种方式降低了实施攻击的技术要求，扩大了受害人群的范围。受害者通常在账户资金减少后，才发现设备已被外部程序自动操作。

系统管理层面的主要困难在于识别意图。软件与侵入式Agent都依赖读取屏幕图文信息和模拟点击来完成任务，操作系统很难准确区分其真实目的。当界面自动化功能在个人设备上大量出现时，系统开发者只能采取全面严格限制控制权限和程序执行范围的防御策略，以此防范普遍的安全风险。

这种防御策略直接导致系统施加更加严格的使用条件。操作系统提高了获取设备控制权限的标准，阻止程序在后台运行，并在涉及资金或隐私的高风险操作前，强制增加多次用户手动确认的步骤。这些改变虽然降低了设备被控制的可能性，却大幅度增加了普通用户完成正常任务的步骤。智能体原本旨在减少的手动操作，被系统强制增加的验证环节完全抵消。

最终，在持续加强的系统安全限制下，用户的操作便利程度被严重削弱，使用智能体的实际时间成本显著增加。界面自动操作越普遍，系统防止功能被滥用的拦截机制就会越广泛。长远来看，这将大幅度增加开发侵入式自动化工具的技术难度，并形成产品广泛推广的阻力。

3.3.2 侵入式Agent在B端的风险案例

侵入式Agent在B端环境中的扩散往往更加隐蔽，其风险也更容易在早期被低估。

案例一：美国总务管理局GSA的RPA审计揭示侵入式Agent加大系统越权风险

公共部门在推进自动化时，往往优先选择对现有系统改造最小的路径。RPA以贴近业务端的方式落地，能够在不重构系统、不改接口的前提下替代重复操作，因此更容易被接受，也更容易快速扩张。问题在于，这类自动化一旦进入生产环境，组织通常会把它当作工具项目管理，而不是当作具备持续访问能力的数字执行主体来治理，风险就会在扩张过程中被低估。

例如，GSA的相关内部审计在部署RPA机器人后，部分关键管理机制没有同步跟上，例如访问权限与安全计划更新不及时，任务结束后仍保留系统访问能力，机器人账号的生命周期管理不够严格，审批与变更流程也可能出现绕行。更重要的是，RPA通过界面操作完成任务，行为记录更依赖终端侧日志与应用侧提示信息，链路一旦跨系统或跨页面，组织既有的审计方式很难形成完整的过程证据，复核与追责的成本随之上升。

案例二：侵入式Agent在Salesloft体系中将凭证风险放大为跨系统的连锁风险

侵入式Agent在企业侧的核心风险，集中在身份凭证被滥用后的放大效应。一旦智能体具备跨系统执行能力，组织的安全边界会从单一应用扩展到整条业务链路，任何一个环节的凭证泄露都可能被快速转化为规模化的数据访问与操作执行，止损窗口随之缩短。

Salesloft相关事件提供了一个直接参照。攻击者并不依赖攻破平台漏洞，而是通过获取合法访问令牌进入多个企业环境并批量获取数据。由于访问路径与操作形态更接近正常用户行为，安全团队很难在第一时间把它与入侵目的区分，依赖补丁与漏洞修补的防护手段也难以快速发挥作用。这与侵入式Agent在企业内的落地方式高度契合。为了完成端到端任务，智能体必须持有账号、会话或令牌，并在多个系统之间持续调用与写回。企业为了提升效率，常见做法包括复用员工账号、共享账号或设置专用执行账号。无论采用哪一种，只要凭证被窃取或被无约束的使用，智能体的速度与覆盖面会把影响从单点扩大到链路级扩散，带来更快的外泄、更难的回滚以及更高的取证与复核成本。

因此，侵入式Agent身份需要独立管理，权限应按任务最小化配置，高风险动作要引入明确的审批与确认，执行过程要做到可追溯、可复核、可回放。否则，风险压力会持续向平台与终端侧传导，最终以更严格的限制和更重的操作负担回到每一个企业用户自身。

侵入式Agent在B端带来的风险与警示

在B端，侵入式Agent在企业内部的引入，往往以“低集成成本”“快速自动化”为卖点，但随着其逐步渗透到关键业务流程，审计失焦、身份滥用与责任不清等问题迅速显现。真实案例表明，当智能体以像员工一样操作系统的方式运行，组织治理并未因此简化，反而被迫引入更严格的权限控制、更复杂的合规审查与更保守的使用策略。结果是，自动化能力并未获得持续放大，反而在多组织环境中被直接禁止或大幅限制。

04 AI Agent时代的未来治理路径

面对侵入式Agent的冲击，亟需确立API主导并构建双重授权与审计体系

- 中国企业侧重短期效率与快速落地
- 美国企业侧重安全可控的规模化发展
- 依托双重授权机制共建合作生态
- 构建全链路可审计的体系协助GUI Agent负责任落地
- 以“API主导、GUI辅助”的治理方向最大化Agent价值

4.1.1 美国AI Agent的治理路径参考

中美在Agent的路径差异在于，中国依靠高权限读屏与模拟点击快速落地，美国坚持最小权限与沙盒或浏览器边界，并以协议和API推动可规模化落地。

与中国AI Agent通过纯视觉方案实现跨应用自动化、侧重快速商业化落地不同，因强调技术创新和隐私安全的平衡，美国对AI Agent能力权限保持警惕，推进相对缓慢，且正尝试通过开放共建协议和强化人机协作探索出一条风险更低、可规模化的发展路径。

近年来中美科技企业在图形用户界面Agent技术路径选择上呈现显著差异。中国厂商基于本土移动互联网应用高度集成与跨场景服务的生态特点，倾向采用纯视觉方案模拟用户操作实现跨应用自动化。部分技术通过获取系统级权限构建底层交互闭环，注重快速商业化落地。该路径因绕过传统应用安全验证体系而引发合规争议。相比之下，美国企业更强调技术创新与隐私保护的平衡，采用更为审慎的技术路线。其应用实践主要通过沙盒隔离与人机协作机制降低系统风险，整体技术推进与权限释放呈现防御性态势。

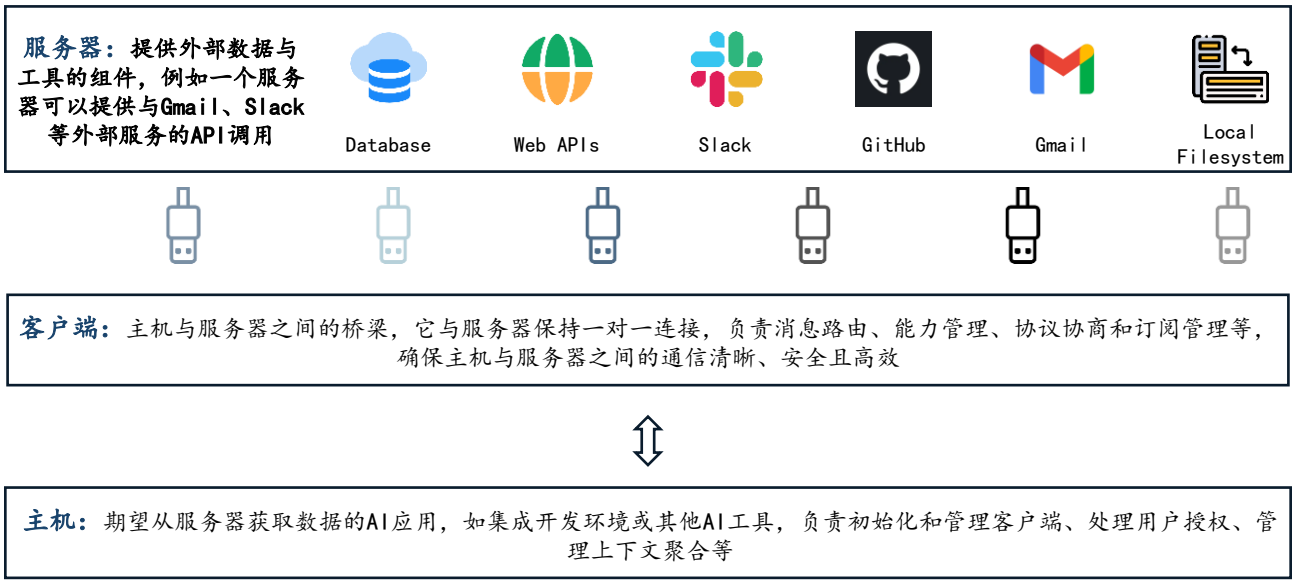
在网页端，美国公司推出的浏览器Agent产品确立了严格的权限边界。OpenAI于2025年推出深度集成至浏览器核心体验的产品。该系统在获得用户明确许可后可自主执行页面导航与表单填写等复杂任务。执行关键操作前系统强制征求用户同意，并允许用户随时暂停或全面接管控制权。底层架构内置了严格的安全限制，阻断运行未知代码、下载文件与读取本地密码等高危操作。在数据隐私保护方面，产品采用可选加入的数据原则。默认情况下用户浏览内容不用于模型训练，用户需主动开启授权配置，系统还设有隐身模式以全面禁用人工智能功能。Anthropic发布的浏览器端技术通过解析页面截图理解当前内容，识别交互按钮与导航菜单。在执行跨站点多步骤任务时，链接点击与表单提交动作被严格限制在浏览器沙箱环境内完成。该服务仅向付费用户开放，不提供免费版或应用编程接口直接调用途径，以确保服务稳定性与数据安全追溯。

在移动端，美国头部科技企业暂未全面铺开无限制的底层界面自动化操作，而是致力于构建兼顾自动化效率与隐私合规的系统级标准。谷歌近期在安卓生态内探索出一条统筹安全与体验的治理方案。谷歌与三星合作引入系统级Agent时采取了规范化的双轨机制。一方面主推系统级功能框架，引导应用开发者主动对接标准化安全接口。另一方面针对未适配的第三方应用，推出内置用户透明度与控制功能的智能自动化框架。在处理订餐与打车等跨应用任务时，Agent被限制在一个沙盒化的虚拟窗口中模拟屏幕操作。这种机制避免了人工智能越权触碰设备核心隐私数据或其他后台文件。执行过程中系统将操作界面实时展示给用户，保证用户享有完整的过程知情权与随时阻断的控制权。苹果公司基于全面升级的开发者意图框架测试新版语音助手，允许用户通过语音指令模拟应用内部操作。目前苹果正与Uber以及YouTube等多家头部第三方应用开展严格的封闭测试。针对医疗与金融等涉及极端敏感数据的领域，苹果采取了直接剔除相关功能或施加最高级别权限物理隔离的策略。考虑到全面接管应用操作可能带来指数级增长的安全隐患，苹果主动放缓了新版功能的全面普及速度。

美国移动端Agent进程推进看似缓慢的背后，实则源于其更注重通过开放共建的标准化协议整合第三方功能的策略。这与侵入式Agent通过获得特权绕过API，无需第三方应用适配就直接操作界面的技术路线存在本质差异。

2024年11月，Anthropic推出MCP开源协议，旨在定义一套通用通信规则，以标准化AI与外部上下文的交互。MCP协议的核心逻辑是标准化直连，通过统一的接口规范，让AI与外部资源直接进行结构化数据交互（例如：AI应用要连接第三方应用，不用为每家制定协议，只需要双方支持MCP协议就可互通），无需依赖视觉识别。这种标准化接口能显著降低AI对接异构系统的复杂度，使AI应用的开发效率和功能扩展性双向提升。目前，全球已有超10,000个公开MCP服务器，已被ChatGPT、Gemini、Copilot等主流AI产品支持和接入。2025年12月，Anthropic宣布将MCP协议正式捐赠给Linux基金会旗下AAIF，这意味着MCP协议不再是一家公司私有资产，而是有望成为一个中立的、行业公认的开放标准。

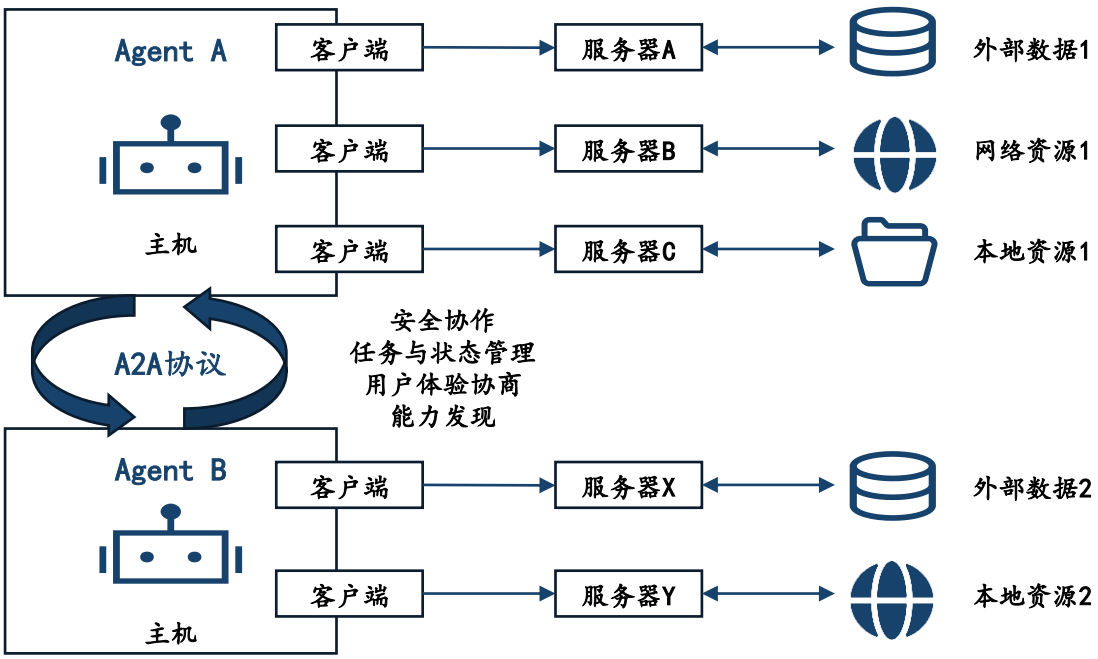
图39：MCP协议方案



MCP开源协议让AI Agent能够连接外部工具和数据，比如调用API、使用数据库等。随着MCP协议采用速度加快，Google于2025年4月推出一个配套的A2A开源协议，旨在让来自不同开发商或基于不同框架构建的异构AI Agent能够进行安全的通信和任务协作。其原理为打通客户端AI Agent与远程AI Agent之间的通信，前者负责制定和传达任务，后者负责执行任务。从细节看，A2A的交互更加贴合人类对于复杂任务协同处理认知：远程AI Agent通过JSON格式的“Agent Card”标注其性能特征，可使客户端AI Agent识别到能胜任当前任务的最佳AI Agent，并利用A2A协议与其进行通信；对于复杂长时间任务，AI Agent之间具备异步通信能力，针对差异化的工作节奏可彼此保持同步。2025年6月，Linux基金会宣布与亚马逊网络服务、思科、Google、微软、Salesforce、SAP和ServiceNow共同成立A2A项目。该项目由Linux基金会托管，初始捐赠内容是Google开发的A2A协议规范、相关SDK以及开发工具。

除MCP、A2A协议，美国CopilotKit于2025年5月推出AG-UI协议（一项由美国提出的 Agent 图形界面交互协议标准），旨在解决AI Agent与前端应用之间的通信难题。它通过标准化事件流（定义16种标准化事件类型，如消息、工具调用、状态差异）保持AI Agent与前端应用的实时同步，这些事件通过标准HTTP或其他传输格式进行流式传输，确保延迟和高可靠性。例如：在一个协作场景，AI Agent可以边处理任务边向用户界面发送进度更新，用户可随时介入调整。这种模式特别适合需要人工监督的场景，如医疗和法律应用。

图40：MCP+A2A协议



4.2.1 治理策略一：构建全链路可审计的运行体系

为能够让AI Agent实现负责任落地，须建立任务级最小权限与全链路可审计机制，使跨应用操作边界可控、过程可观测、责任可追溯。

构建基于任务级最小权限与全链路可审计的AI Agent运行体系，通过动态权限控制模型与多维度身份认证机制，实现跨应用交互场景下的操作边界精准界定、执行过程实时可观测、责任主体可追溯验证。

侵入式Agent依托系统级权限实现跨应用交互与自动化操作，本质上突破了Android权限体系“最小权限原则”的设计初衷，这种架构模式使AI能够实时获取屏幕视觉信息、解析用户操作意图并执行多步骤任务，由此产生的隐私风险已超出传统应用权限管控范畴——用户行为轨迹、敏感内容识别等数据可能在未显式授权情况下被采集与处理。而且，具备高度自主决策能力的AI Agent在任务执行过程中可能产生不可预期的行为偏差，当其决策导致财产损失或数据泄露时，开发者、部署者、使用者等多元主体之间的责任边界将面临模糊化挑战。因此，AI Agent的可持续发展亟需构建基于任务级最小权限与全链路可审计的运行体系。

一是界定操作边界，根据操作类型实时分配最小必要原则，确保权限、数据使用最小化。首先，AI Agent部署时需明确行为边界，对AI Agent行为进行分级管控，如针对直接影响用户资产权益和人身安全的高敏感行为（金融交易、支付转账、不可逆数据删除等），应禁止其代为执行；针对涉及用户隐私信息的行为，需动态授权和实时显示，确保用户知情权与选择权实时在线；针对基础功能操作可允许通过协议直接授权以实现无感体验。其次，将操作权限划分若干安全等级，建立分级分类管控机制，采用微权限管理技术为每个任务实例分配最小必要原则。在涉及高敏感操作时，需通过多重生物特征认证+人工复核机制方可执行，同时通过操作类型、数据敏感度、执行环境等多维度实时计算风险值，当风险超过预设阈值时，自动触发人工介入流程。再者，AI Agent在数据搜集和使用时需遵循功能必要性原则，通过形式化验证矩阵确保数据采集与核心功能的必然关联性，对于非必要数据字段实施默认拒绝策略，以保证数据使用最小化。最后，构建贯穿数据收集、处理、储存、传输、销毁全流程的保护体系，在数据流转各环节设置“安全关卡”，在处理涉及用户隐私信息的敏感数据时，需优先在端侧完成，必须上云的数据需在可信隔离环境中执行，如采用加密算法对数据进行加密处理。

二是构建全链路实时可追溯的AI Agent运行体系，确保责任界限清晰。在AI Agent初始化阶段通过交互协议（如多模态弹窗、协议签署流程）向用户明确其功能边界（如支持的操作类型、数据访问权限、执行时限等），同步展示操作风险矩阵，包括但不限于数据泄露风险等级、误操作可能性等。此外，系统应设置可选项，保证用户可随时开启或撤销AI Agent代为执行功能，类似ChatGPT Atlas为用户提供的隐身模式。在代为执行的过程中，通过可视化进度条实时反馈执行状态，同时在用户界面侧边栏部署可交互式控件，支持用户随时接管/暂停/终止操作；采用区块链存证技术记录关键操作节点，每个步骤生成包含时间戳、操作参数、执行结果的不可篡改日志；执行记录需支持多层级回溯，基础层提供原始操作日志，中间层生成可视化操作路径图谱，高级层支持AI驱动的异常行为分析。

4.2.2 治理策略二：API主导GUI辅助的治理方向

API主导、GUI辅助的治理策略，本质是将跨应用执行分层。可标准化、可审计的操作优先API，API覆盖不到的长尾环节才允许GUI，同时对GUI施加更严格的权限确认与审计约束。

API Agent和GUI Agent是目前两大主流移动交互范式。相较于缺乏授权约束的侵入式Agent，“高价值场景API Agent主导、长尾需求GUI Agent辅助”的混合模式因能解决API模式下覆盖率不足和侵入式Agent模式下安全隐患问题，更利于构建AI Agent、APP、用户三者间的良性协同。

API Agent与GUI Agent是现阶段移动交互的两种主流范式。API Agent的核心优势在于权限边界高度可控且操作效率高，能为高价值场景提供安全保障。通过标准化接口定义和细粒度的权限管理，系统能够精确限定AI Agent对敏感信息的访问范围，避免过度授权风险。同时，API模式下通过直接调用后端服务，可实现精准调用与即时闭环执行，显著减少完成任务的步骤与时间。GUI Agent的核心优势则在于通用性，可解决API模式无法覆盖的长尾需求。它依托视觉理解技术，无需依赖应用API即可操作屏幕界面，实现对多数移动应用的适配，包括未开放API的功能、非结构化场景以及快速响应应用更新。例如，GUI Agent能处理小众应用或最新功能，无需等待API迭代。

基于此，相较于纯API模式或设计不当的侵入式模式，API主导、GUI辅助的混合模式可实现更广泛的用例覆盖、更高效的执行以及更人性化的交互风格，符合中国移动互联网安全合规与生态包容的发展逻辑。例如，通过API Agent完成支付操作，使用GUI Agent查看订单信息。这种混合模式突破了单点范式的局限，如API模式的覆盖盲区或侵入式Agent模式的安全隐患，有利于构建AI Agent、APP与用户间的良性协同。对于APP厂商，一方面通过API实现核心能力的安全授权，规避数据滥用风险；另一方面通过GUI扩展生态包容性，覆盖小众需求。对于用户，获得一句话完成复杂任务的无缝体验，且隐私风险大幅降低。对于AI生态，该模式构建了三方协同新范式，即AI Agent精准执行高价值任务、APP扩展服务边界、用户享受安全普惠体验，推动移动智能交互从功能堆砌跃升至安全、高效、普惠三位一体的生态级智能服务。

目前，市场已提出混合API Agent和GUI Agent的三种策略：

1.基于GUI工作流的API封装策略：使用API包装器将GUI操作隐藏在可编程接口中，这种方法有效地将GUI交互抽象为结构化命令，使原本为人工导航设计的应用程序能够以更具程序化和可拓展性的方式实现自动化。例如：将多步骤任务转变为可直接调用函数的形式，为开发者提供类似API的接口，简化集成过程，增强效率、可拓展性。

2.统一编排工具：通过统一编排器，根据任务需求和系统功能，动态选择最合适的方式来完成交互。例如：微软实验工具UFO通过智能路由引擎，在执行任务时优先使用APP开放的API，如果没有合适API，则无缝切换至GUI模式。

3.低代码/无代码解决方案：用户通过拖放组件方式构建应用程序或实现自动化流程。例如：在订单处理流程中，用户将“支付网关”组件拖到设计界面进行简单配置，平台就会在后台自动调用API Agent向支付端点发送请求、处理支付事务，接着连接“物流服务”组件完成订单发货，若某个步骤需要进行GUI验证，平台又能无缝插入GUI Agent模拟人工操作。这种方式将API模式的高效性和GUI模式的直观性结合起来，使得构建自动化流程简单化。

4.2.3 治理策略三：双重授权体系

双重授权机制的核心，是要求侵入式Agent的每一次关键操作同时满足用户与系统层授权、以及被调用方业务层授权，以明确调用边界并强化可审计与责任可追溯性。

双重授权机制的核心是在保障安全可控与可审计的同时，减少未经同意调用引发的生态摩擦。

双重授权机制要解决的问题，是为AI Agent的跨应用操作建立明确的许可边界，使其在扩展能力的同时不影响生态协作秩序。该机制要求一次关键操作同时满足两类授权。其一来自用户与系统层，明确用户是否同意智能体代为执行，并限定可调用的系统权限范围。其二来自被调用方应用与服务提供方，明确是否同意被智能体调用，并对可执行的动作范围与关键约束形成可执行的约定。两类授权同时成立，相关操作方可进入执行链路，从源头降低“未经同意即调用”带来的摩擦与争议。

仅依赖用户或系统授权容易使高权限能力在跨应用与跨流程执行中被默认放大，进而触发两类外溢问题。一种是安全与可控性压力上升，识别偏差、流程偏离与越权触发更容易触达敏感信息与关键业务动作，且事后核验与追溯成本更高。另一种是生态协同难度加大，智能体在入口层重塑用户路径后被调用方的商业规则更容易被绕开，协作摩擦增加且参与意愿下降，长期供给与创新动能受到影响。双重授权的价值在于将规则前置，把跨主体调用纳入可约束与可核验的协同框架，从而同时缓释安全风险与生态冲突。

为确保双重授权机制能够真正发挥统筹安全与协同生态的作用，其落地执行需紧密围绕四个核心要素展开。首先是界定许可边界，这直接对应用户与被调用方的双向诉求，既限定智能体可调用的系统底层权限，又明确哪些业务动作允许代办以及哪些必须由用户亲自确认。其次是细化动作约束，将双重授权的具体要求落实到动作与场景粒度，针对高风险操作设定关键节点二次确认以及额度和频次限制等触发条件，并配备撤销与回滚功能。再次是实现过程留痕，建立贯穿授权记录、执行过程与结果反馈的全链路档案，确保双边授权的每一步操作均可追溯。最后是明确责任归属，依托全链路留痕为核验、审计与责任划分提供事实依据，从而保障整个双重授权规则体系的切实可执行。

方法论

沙利文研究院布局中国市场，深入研究19大行业，持续跟踪532个垂直行业的市场变化，已沉淀超过100万行业研究价值数据元素，完成超过1万个独立的研究咨询项目。

研究院依托中国活跃的经济环境，研究内容覆盖整个行业的发展周期，伴随着行业中企业的创立，发展，扩张，到企业走向上市及上市后的成熟期，研究院的各行业研究员探索和评估行业中多变的产业模式，企业的商业模式和运营模式，以专业的视野解读行业的沿革。

研究院融合传统与新型的研究方法，采用自主研发的算法，结合行业交叉的大数据，以多元化的调研方法，挖掘定量数据背后的逻辑，分析定性内容背后的观点，客观和真实地阐述行业的现状，前瞻性地预测行业未来的发展趋势，在研究院的每一份研究报告中，完整地呈现行业的过去，现在和未来。

研究院密切关注行业发展最新动向，报告内容及数据会随着行业发展、技术革新、竞争格局变化、政策法规颁布、市场调研深入，保持不断更新与优化。

研究院秉承匠心研究，砥砺前行的宗旨，从战略的角度分析行业，从执行的层面阅读行业，为每一个行业的报告阅读者提供值得品鉴的研究报告。

法律声明