区块链上的自主代理：标准、执行模型与信任边界

Saad Alqithami

arXiv:2601.04583v1 [cs.AI] 8 Jan 2026

摘要

大语言模型的进步催生了智能体人工智能系统，它们能够进行推理、规划并与外部工具交互，以执行多步骤工作流；与此同时，公有区块链已发展为可编程基础设施，用于价值转移、访问控制和可验证状态转换。二者的融合带来了一个高风险的系统设计挑战：如何设计标准化、可互操作且安全的接口，使代理能够观察链上状态、拟定交易意图并授权执行，同时避免用户、协议或组织面临不可接受的安全、治理或经济风险。本综述通过系统性文献综述梳理了智能体—区块链互操作性这一新兴领域，从最初超过 3000 条记录中筛选出 317 篇相关文献。我们的贡献包括：一个涵盖五类集成模式的分类体系，涉及只读分析、模拟与意图生成、委托执行、自主签名和多智能体工作流；一个针对智能体驱动交易管线的威胁模型，涵盖从提示注入和策略滥用，到密钥泄露、对抗性执行动态及多智能体串谋等风险；以及一个比较能力矩阵，从 13 个维度对 20 多个代表性系统进行分析，维度包括托管模型、权限管理、策略执行、可观测性和恢复。基于分析揭示的不足，我们提出一份研究路线图，重点关注两种接口抽象：交易意图模式，用于实现可移植且明确无歧义的目标规范；以及策略决策记录，用于在不同执行环境中实现可审计、可验证的策略执行。最后，我们提出一套可复现的评估套件与基准测试，用于评估智能体介导的链上执行的安全性、可靠性和经济鲁棒性。

关**键词** 智能体人工智能 · 区块链 · 智能体—区块链接口 · 交易意图 · 账户抽象 · 策略执行 · 最大可提取价值 · 安全

1 引言：两场革命的必然碰撞

过去五年间，两场深刻且并行的技术革命发生了碰撞：智能体人工智能的兴起，以及公有区块链作为全球结算层的成熟。前者已从静态语言建模发展到能够分解目标、调用工具并在多步骤工作流中保持状态的系统。后者则从加密货币支付通道演进为一种可编程底层架构，支持价值转移、访问控制和可验证的状态转换。二者的融合并非仅仅是学术上的新奇事物；它已经推动一场正在展开的变革，使具有经济意义的数字活动实现自动化。然而，这种融合也带来了独特而危险的系统挑战：我们如何才能安全地将自主、非确定性的人工智能连接到不可篡改、具有对抗性且承载高价值的金融网络?

大语言模型（LLM）发展为能力强大的自主代理，是人工智能领域最重大的进展之一。早期的基于Transformer的模型展现出卓越的文本生成流畅度，但缺乏持续推理或与外部交互的能力 [302, 51]。思维链提示的引入表明，明确的推理轨迹能够显著提升模型处理复杂任务的表现 [313], ；随后，工具增强方面的研究进一步证明，大语言模型可以学会调用外部API、数据库和计算资源 [267, 284]。这些进展最终促成了智能体系统的出现，使其能够进行多步骤规划、自我纠正和自主任务执行 [330, 330]。

与此同时，区块链技术也经历了自身的转型，从一种小众的密码学新奇事物发展为关键的金融基础设施。以太坊上智能合约的引入，使可编程、自动执行的协议得以根据任意逻辑托管和转移价值 [318, 318]。随后，去中心化金融（DeFi）的迅猛发展表明，复杂的金融工具——借贷协议、自动做市商、衍生品平台——能够在没有传统中介的情况下运行 [324, 324]。近来，ERC-4337 [2] 等账户抽象标准将交易授权与密钥所有权解耦，使具备复杂访问控制策略的可编程钱包成为可能 [55]。

这两种发展轨迹的交汇既创造了前所未有的机遇，也带来了新的风险。一方面，具备区块链能力的人工智能智能体能够以超人的速度和稳定性自动进行投资组合管理、执行复杂的交易策略、参与治理并管理资金运营 [317, 90]。另一方面，自主决策、不可逆交易和对抗性环境的结合，形成了独特且危险的攻击面。遭到入侵或操纵的代理可能会转移钱包中的全部资金、执行未经授权的交易，或落入复杂的 MEV 提取陷阱——这一切都可能发生在人类介入之前 [80, 81]。

在真实环境中评估智能体能力已成为一项重要的研究重点。AndroidWorld [255]和 AndroidLab [327]等基准测试评估智能体在动态移动环境中完成多步骤任务的能力，而 AgentClinic [268] 和 MLGym [225] 等特定领域的基准测试则分别评估智能体在医疗保健和机器学习研究情境中的表现。这些评估框架揭示了当前智能体架构令人瞩目的能力及其持续存在的局限性，为寻求在高风险区块链环境中部署智能体的系统设计者提供了重要指导 [69, 207]。

智能体区块链系统的实际部署，需要仔细考量现实世界中的约束条件。生产系统必须在自主性与安全性、速度与验证、能力与可控性之间取得平衡 [88, 239]。面向大语言模型的区块链交互新兴框架表明，有效集成不仅需要精湛的技术，还需要审慎考虑失效模式、对抗性场景和人工监督机制 [296, 234]。如今采用的设计模式和架构选择，将在未来多年影响智能体区块链系统的安全性与效能。

本综述全面考察智能体与区块链集成领域，综合了同行评审出版物、技术标准和行业报告中的见解。我们提出了一套新颖的集成模式分类体系、一种针对自主区块链交互独特风险的专门威胁模型，以及对 20 个生产系统的比较分析。我们的目标是为研究人员、从业者和政策制定者提供必要的概念框架与实践指导，助其应对这一快速发展的领域。

1.1 智能体人工智能的曙光

大语言模型（LLM）从文本生成器演变为智能体系统的速度令人瞩目。这一转变最适合被理解为一系列架构和方法论上的步骤：

1. **可扩展的架构：** Transformer架构实现了高度并行的训练，并成为现代大语言模型扩展的主导基础 [302]。后续的大规模训练工作表明，增加模型容量和数据量能够广泛提升模型在各种任务上的表现 [51]。2. **规模化带来的新兴任务能力：** GPT-3 等大型模型展现出强大的上下文学习和少样本学习能力，使其能够执行训练期间未被直接监督的任务 [51]。3. **对齐与可控性：** 随着大语言模型能力不断增强，研究重点转向引导模型行为，使其符合人类目标，并避免不安全或非预期的行动。关于对齐的综述工作对由此形成的训练、偏好塑造和安全技术领域进行了系统梳理 [158]。4. **从推理到自主代理：** 能够引出中间推理过程的提示方法，尤其是思维链（CoT），表明在引导大语言模型生成结构化中间步骤时，它们能够更可靠地解决复杂问题 [313]。在此基础上，ReAct 等面向行动的提示和规划范式进一步将推理与工具调用及环境交互联系起来 [330], , 推动了实用的类代理认知架构的发展 [325, 211]。

关键在于，当这些认知架构与**工具和 API**结合时，它们便具有了经济上的重要性。工具使用型代理可以查询实时系统、调用外部服务并执行多步骤流程，而不只是生成文本。近期研究已系统梳理大语言模型代理如何学习使用工具，以及如何对工具接口进行规范和优化 [325, 336]。自监督工具使用（例如 Toolformer）等工具学习机制，以及大型 API 连接模型（例如 Gorilla），进一步展示了如何通过训练或提示模型，使其能够大规模选择工具调用并设置参数 [267, 237]。与此同时，其安全影响已十分明确：函数调用和工具调用

会扩大攻击面，若工具缺乏稳健的策略和验证层约束，就会为滥用和越狱开辟新的途径 [66] [322, 181]。

1.1.1 大语言模型智能体架构的进展

大语言模型智能体架构的发展经历了几个不同阶段，每个阶段都引入了规划、工具使用和多步推理方面的新能力。早期的代理框架主要依赖思维链提示来引导逐步推理 [313], , 但当任务需要与外部系统交互或适应意外结果时，这种方法就显得脆弱。ReAct 框架通过交替进行推理轨迹和动作执行，解决了这一局限，使智能体能够观察环境反馈并据此调整规划 [330]。近期研究则探索了自适应规划和推理机制，使智能体能够处理日益复杂的多轮任务 [91, 151]。

工具学习已成为必须与外部系统交互的智能体的一项关键能力。研究表明，大语言模型可以通过上下文示例学习使用新工具，如 Toolformer 所示 [267],；也可以通过针对函数调用的专门训练来掌握工具使用，如 NexusRaven [284] 和 Gorilla [237]所示。对于区块链交互，工具学习使智能体能够适应新的协议、合约接口和特定链惯例，而无需针对每种集成都进行显式编程。然而，工具学习也会带来风险：学会错误使用工具的代理可能会构造无效或危险的交易，而对抗性工具描述也可能操纵代理行为。关于提示注入 [131, 191] 和越狱攻击 [349, 350] 的研究揭示了基于大语言模型的系统容易受到精心设计的输入影响，这凸显了在任何工具使用框架中采用稳健验证、沙箱隔离和人工监督机制的必要性。

1.1.2 核心问题：衔接智能与不可变性

智能体与区块链集成的核心，在于人工智能系统的自适应、概率性本质，与区块链执行的确定性、不可变本质之间存在根本性张力。这种张力体现在多个维度，并塑造了集成系统的设计空间。

人工智能智能体，尤其是基于大语言模型的智能体，通过模式匹配、概率推断和习得式启发法运行。其输出本质上具有随机性，即使输入相同，不同运行结果也会因采样过程而有所不同。它们擅长处理歧义、适应新情境，并在不确定性下做出合理决策。然而，它们也存在多种失效模式，包括幻觉、不同上下文间的不一致，以及易受难以预测或全面防范的对抗性输入影响 [273, 114] 。整体评估框架揭示了大语言模型推理中系统性的弱点，即使在最先进模型中，这些弱点依然存在 [183]。
相比之下，区块链系统通过不可变代码所规定的确定性状态转换运行。以太坊虚拟机保证，每笔交易无论何时何地执行，都会产生相同的结果 [318]。这种确定性使分布式网络能够实现无需信任的验证与共识 [178]。然而，这也意味着错误一旦写入链上，就无法撤销。区块链交易没有撤销按钮——这一特性已导致智能合约漏洞 [244] 和执行错误造成数十亿美元的不可挽回损失。

整合这些范式，需要仔细关注它们之间的接口。代理的概率推理必须转化为确定性交易规范。区块链不可篡改的执行必须以全面验证和模拟 [295]为前提。链上操作的不可逆性必须配以适当的保障措施和人工监督机制，正如第5节提出的威胁模型所探讨的那样。

1.1.3 历史背景：两场革命的交汇

智能体人工智能与区块链技术的交汇，代表着21世纪初最具变革性的两项技术发展的融合。要理解这一交汇，就必须认识到各自迥异的历史发展轨迹如何推动这些技术发展至当前状态。

区块链革命始于 2008年《比特币白皮书》的发布。该白皮书提出了一种无需可信中介即可解决双重支付问题的新颖方案 [223]。这一突破促成了数字稀缺性的创造和无需信任的价值转移，为金融基础设施的新范式奠定了基础。随后，智能合约平台逐步发展，其中以以太坊为代表 [318], , 将区块链能力从简单的价值转移扩展到任意可编程逻辑 [190], , 使去中心化应用和自主金融协议得以创建。DeFi 的爆发证明了这些协议在实践中的可行性；系统化研究记录了链上金融基础设施的快速增长 [324]。

随着 Transformer 架构 [302]和大语言模型的发展，人工智能革命进入了一个全新的变革阶段。缩放定律表明，随着计算资源、数据和参数的增加，模型能力会以可预测的方式提升，从而推动能力迅速增强。指令遵循模型的出现，使大语言模型从文本补全引擎转变为能够遵循复杂指令并使用外部工具的交互式助手。近期研究主要聚焦于开发能够自主推理并执行复杂任务的智能体 [163]。

这些革命的交汇带来了任何一种技术单独都无法实现的可能性。区块链为无需信任的价值转移和可编程协议提供基础设施，而智能体人工智能则提供了在复杂决策空间中导航并代表委托人与这些系统交互所需的智能。二者结合，使一类新型自主经济参与者得以出现；它们能够持续运行、执行复杂策略，并在极少人工干预的情况下管理巨额价值。

1.1.4 自主系统的基础性发展

智能体人工智能系统的出现，建立在数十年来对自主代理和多智能体系统的研究之上。有关承诺与惯例的基础性研究为多智能体系统中的协调奠定了概念框架，至今仍为现代智能体设计提供参考 [156]。早期关于基于逻辑的代理通信协议的研究 [94] 探讨了智能体如何交换信息并协调行动，为理论奠定了基础；随着基于大语言模型的智能体开始在多智能体配置中运行，这些理论基础依然具有现实意义 [90]。
从基于规则的专家系统转向基于学习的方法，标志着代理能力发生了重大变化。关于代理行为的动力系统方法 [8] 以及多智能体协调的研究 [134] ，为理解自主系统如何在复杂环境中运行奠定了基础。近期大语言模型的出现加速了这一趋势，使智能体能够理解和生成自然语言、对复杂情境进行推理，并与各种工具和 API 交互。然而，能力的提升也带来了新的挑战，包括难以确保系统在分布外场景中的可靠行为 [22], 、可能出现意料之外的故障模式，以及如何对日益自主的系统保持人工监督 [112]。

自主代理在金融与经济领域的应用历史悠久，早于区块链技术的出现。关于去中心化系统 [82]和基于区块链的多智能体架构 [92]的研究，探讨了智能体如何在无需信任的环境中运行。Anoma [124]等基于意图的架构，以及在以太坊虚拟机上实现的隐私保护型意图执行 [311] ，代表着智能体与区块链交互的新范式：从命令式交易构造转向声明式意图规范。这些架构创新，加上通过硬件安全模块 [136] 实现的安全密钥管理方面的进展，以及预言机可靠性 [193], 的提升，正在构建实现安全、高效的智能体与区块链集成所需的基础设施。

1.2 去中心化经济的成熟

与此同时，区块链技术早已超越其最初作为点对点电子现金系统 [223]的定位，发展日趋成熟。在这一过程中，它演变为可编程结算层，使智能合约能够在共享账本 [318]上，以最小化信任的方式自动执行价值转移、治理和状态更新。这一转变推动去中心化应用（dApp）生态系统迅速扩张：经济活动由可组合协议而非中心化中介促成，对抗性激励则构成默认威胁模型的一部分。

这一演进的标志性体现是**去中心化金融（DeFi）**，它提供开放且无需许可的交易、借贷和结构化链上金融操作机制。尽管具体的协议生态变化迅速，但其架构模式保持稳定：自主合约系统公开公共接口，这些接口可以组合成端到端工作流，在创造强大自动化机遇的同时，也带来了新型攻击面 [170]。与此同时，区块链通过**互联网原生治理**和**去中心化自治组织（DAO）**实现了集体决策协调，并借助链上投票和智能合约强制执行的规则来实现这一点 [266, 217]。最后，代币化原语拓展了链上所有权表示的范围，支持同质化与非同质化资产模型，并实现可编程托管、转移限制及来源追溯机制。

然而，链上应用的增长也加剧了第一层执行的可扩展性限制。这推动了**二层扩展** 的兴起，借助以 Rollup 为中心的架构，将执行转移到链下，同时保留链上结算保障 [325]。对于智能体介导的交互而言，这些扩展层并非无关紧要的细节；它们往往是切实可行的执行场所，使高频、低延迟、成本敏感型工作流成为可能，同时

表 1：传统 Web API 与区块链交易接口的比较，凸显了为何智能体到链工具需要更严格的安全性控制措施 [80, 318]。

维度	传统 Web API（例如天气端点）	区块链交易接口
主要操作	信息检索（GET）	状态转换（签名并发送）
可逆性	可逆／幂等	不可逆且不可撤销
错误成本	低（例如，信息不准确）	高（例如，永久性财务损失）
运行环境	良性／协作性	默认对抗性（最大可提取价值）
权限模型	限定范围的 API 密钥	密码学私钥（持有人资产）
状态一致性	中心化，最终一致	去中心化，可验证一致

也引入了新的运维考量（跨链桥接、最终性假设和跨域安全边界）。

1.3 动机：为什么“Agent-to-Chain”是一个独特危险的问题

将人工智能智能体连接到区块链，并非只是又一次工具集成。区块链系统的基本特性带来了一系列高风险属性，使“代理到链”的交互有别于典型的 API 调用。这些差异总结于表 1，也因此需要一类全新的标准、接口和安全协议。

这些差异之所以重要，是因为故障的影响不止于用户体验，还会直接造成不可逆且带有经济对抗性的后果：

- **不可逆性与经济后果：** 公有区块链采用仅追加账本设计；若不违背共识假设，已最终确认的交易便无法回滚。这一特性对于信任最小化至关重要，但也意味着，错误或恶意交易会直接造成永久损失，而非可恢复的系统错误。此前对智能合约执行的分析指出，原子组合和不可逆的状态转换会放大故障的影响，交易一旦在链上提交，尤为如此 [235]。

- **授权即持有人资产：** 与范围受限的 API 凭证不同，区块链授权从根本上与密码学签名绑定。持有私钥或签名能力，就意味着直接掌控资产和合约交互，因此授权是一种可转让的持有人资产，而非可撤销的权限。这一区别促使人们转向强化托管架构，包括阈值签名和多方签名方案，旨在对抗性环境中降低密钥单点安全失陷的风险 [184, 169]。

- **默认存在对抗性执行环境：** 由于待处理交易在纳入区块之前可被公开观察，区块链执行环境会激励对抗性重排序和插入策略。最大可提取价值 *[60]*（MEV）将这一现象形式化，表明理性参与者可以通过抢先交易、尾随交易和三明治攻击系统性地获利，同时不违反协议规则 [80]。因此，代理必须将公共内存池视为对抗性的交易场所，而非中立队列。
- **异构且碎片化的底层环境：** Agent-to-Chain 交互很少只针对单一执行环境。相反，代理会在异构的第一层和第二层系统、不同的账户和费用模型以及跨链桥之间运行。关于跨链桥故障和生态系统碎片化的实证研究表明，这些边界会集中风险，并经常成为高价值攻击面，尤其是在自动化转账工作流中 [9, 46]。

综合来看，这些特性凸显了安全优先的接口规范的重要性：智能体到链系统应围绕受限意图表示、策略门控授权、稳健托管以及假定存在对抗性排序和价值提取的执行路径来构建 [36]。

1.3.1 风险因素的深入分析

智能体与区块链集成所涉及的风险，已超出眼前的技术挑战，延伸至更广泛的系统性和社会层面的关切。多智能体系统带来了协调挑战，可能导致涌现故障 [194, 218]。金融交易自动化引发了对市场稳定性和闪崩的担忧 [195, 209]。当智能体处理敏感用户数据时，隐私考量至关重要 [197, 196]。人工智能智能体的整体发展既带来机遇，也伴随风险，必须谨慎管理 [203, 199]。对于高风险应用而言，人工智能系统的诚实性与对齐是一项关键关切 [206]。智能合约

安全性仍是一项持续存在的挑战 [210, 212]。不同区块链系统之间的互操作性带来了额外的攻击面 [214]。必须谨慎设计拜占庭容错协议，以应对对抗性条件 [216, 201]。验证智能合约逻辑对于确保其正确性至关重要 [208]。全面的风险分析不仅要考虑各个代理系统的故障模式，还必须考虑多个代理在共享区块链环境中交互所产生的涌现风险。
当少数代理系统或运营者控制区块链活动的很大一部分时，就会出现集中度风险。这种集中可能造成单点故障，为市场操纵提供条件，并削弱区块链系统旨在实现的去中心化特性。技术市场中的赢家通吃动态表明，如果不进行有意识的干预，代理-区块链生态系统可能会自然演变为高度集中的结构。

当多个智能体采用相似策略、使用共同数据源或共享底层模型架构时，就会产生相关性风险。相关行为可能放大市场波动，形成反馈回路，并在共同假设被证明错误时导致级联故障。不同代理系统广泛采用相似的大语言模型基础，形成了一种单一文化风险，可能表现为协同故障或漏洞。

监管风险涵盖围绕自主代理及其交易法律待遇的不确定性。关于法律责任、问责以及代理执行的合约是否具有可执行性等问题，目前仍大多未得到解决。随着监管框架不断演变，以应对自主经济参与者带来的新颖挑战，代理及其运营者可能面临意料之外的法律风险。

1.4 范围与定义

本综述聚焦于实现智能体—区块链互操作性的标准、协议、架构和系统。我们对相关接口领域进行系统化梳理，这些接口使智能体能够安全地读取链上数据、构造交易并授权执行。

关键定义：

- **智能体人工智能：**一种通常由大语言模型驱动的人工智能系统，能够针对多个步骤进行推理和规划，并与外部工具交互以实现目标 [330, 325]。
- **工具/函数调用：**智能体通过结构化调用而非自由文本调用外部能力（API、数据库或其他服务）的机制，从而能够与有状态系统进行基于实际数据的交互 [267, 237]。
- **钱包：**一种管理密码学密钥并生成签名，以授权交易和智能合约交互的软件或硬件接口。
- **智能合约：**部署在区块链上的程序，其状态转换由共识机制强制执行，并由交易触发。
- **交易意图：**对用户期望结果（“做什么”）进行高层次、声明式表达，并将其编译为一笔或多笔具体的、特定于链的交易（“如何做”）；该概念用于意图中心执行架构 [202]。

范围包括：

- 智能体—工具通信的标准和协议（例如 MCP、UTCP），以及它们对安全接口设计的影响。
- 智能体钱包和密钥管理架构，包括多方计算/门限签名、可信执行环境以及基于账户抽象的智能账户 [117, 55, 169, 184]。
- 智能体驱动交易的威胁模型和安全考量，包括对抗性执行动态以及策略绕过风险 [80]。
- 用于评估智能体介导的链上执行的安全性、可靠性和经济鲁棒性的评估框架与基准测试。
- 用于约束、审计代理操作并从中恢复的治理与策略执行机制。

范围之外：

- 通用 L　语言模型对齐与安全性，但直接影响工具使用或交易授权的情况除外。

- 底层区块链共识协议或密码学原语的设计，但其对接口和托管设计构成约束的情况除外。
- 关于智能体理论的纯理论研究，不涉及具体的链上接口或授权路径。
- 人工智能对市场的更广泛经济或社会影响，但直接关联到智能体介导的链上执行的情况除外 [100]。

1.5 贡献

本综述系统且全面地概述了代理与区块链互操作领域。我们的主要贡献如下：

1. **集成模式分类体系：**一种五部分分类体系，依据代理的权限边界对系统进行分类，涵盖从只读分析到委托执行、自主签名和多代理工作流的不同类型。 2. **专门的威胁模型：**一种针对代理驱动交易工作流的威胁模型，识别观察、规划、授权、执行和验证各环节中的关键资产、威胁行为者及故障模式。 3. **比较能力矩阵：**对 20+ 个代表性系统进行分析，涵盖 13 个维度，包括托管与密钥控制、权限管理、策略执行、可观测性和恢复机制。 4. **研究路线图：**一项面向未来的研究议程，以两种接口抽象为核心：用于便携式目标规范的**交易意图模式（TIS）**和用于可审计策略执行的**策略决策记录（PDR）**，并附有参考架构。 5. **可复现工件：**一组资产，包括综述方案、PRISMA 风格的筛选摘要、精选参考文献数据库，以及用于构建智能体到链系统的实用安全检查清单。

通过系统化梳理设计空间、明确风险面并勾勒标准化路径，本综述旨在为致力于构建安全且可互操作的区块链人工智能智能体未来的研究人员、开发者和政策制定者提供基础性资源。

1.5.1 范围边界与局限性

尽管本综述旨在全面涵盖智能体与区块链集成，但其范围受到若干边界的界定。了解这些边界有助于读者 contextualize 本综述的贡献，并确定需要其他资源支持的领域。

时间范围本综述主要涵盖截至2026年初的发展，尤其强调2023—2025年期间的系统与研究，因为这一时期该领域经历了快速增长。必要时，本综述会纳入更早期的基础性工作以提供背景，但并不试图全面涵盖人工智能智能体或区块链技术在2020年以前的发展。

技术范围本综述聚焦于人工智能智能体与公开的无许可区块链网络交互的系统，尤其是以太坊及与以太坊虚拟机兼容的区块链。私有或许可型区块链部署虽然可能具有相关性，但不在本文的深入讨论范围内。同样，本综述主要关注基于大语言模型及相关架构的智能体，对其他人工智能范式（如强化学习智能体或基于规则的系统）的覆盖较为有限。

应用范围 本综述重点关注金融应用，包括去中心化金融、支付和资产管理，这些领域构成了当前代理与区块链部署的主体。对于供应链、身份和治理等非金融应用，本文会在其与核心主题交叉之处加以讨论，但全面探讨这些领域超出了本综述的范围。

地理与监管范围 本综述并未试图全面涵盖不同司法管辖区的监管框架。文中从通用层面讨论监管考量，但读者应查阅针对具体司法管辖区的资源，以获取详细的合规指南。

2 方法论：系统综述的严谨方案

为确保对智能体—区块链互操作性格局进行系统、全面且可复现的综述，我们采用了结构化的系统性文献综述（SLR）协议，并依据 PRISMA 2020 报告原则，使用 PRISMA 风格的流程图报告筛选过程 [233]。我们遵循既定的系统性文献综述实践，预先定义研究问题、制定多源检索策略、应用预先注册的纳入与排除标准，并维护可审计的筛选日志。在区块链与安全性文献中，类似的系统性文献综述协议常用于减少选择偏差，并在整合快速发展的证据库时提升透明度 [145, 271, 15]。

本节详细介绍了我们方法论的四个核心组成部分：（1）指导我们开展研究的研究问题；（2）多来源检索策略；（3）用于筛选文献的纳入与排除标准；（4）用于从所选语料库中提取和分析属性的编码方案。

2.1 研究问题

本综述围绕四个研究问题（RQ）展开，旨在从描述性系统梳理逐步推进至面向未来的研究议程：

- **RQ1：将智能体人工智能系统与区块链集成的主要架构模式有哪些？**我们旨在识别反复出现的集成模式（例如只读分析、仿真优先工作流、委托执行和自主签名），并阐明解释这些模式为何在实践中出现的关键权衡因素（例如权限、延迟、可审计性和故障恢复）。
- **RQ2：与代理驱动的链上交易相关的独特安全威胁和风险有哪些？**我们构建了一个专门的威胁模型，将工具使用漏洞（例如指令劫持和工具欺骗）与区块链原生对抗性动态（例如内存池可见性和交易重排序）相结合，并以在区块链安全领域应用大语言模型和智能体这一更广泛的发展方向为基础 [145]。
- **RQ3：在实现安全的智能体与区块链互操作性方面，协议、平台和标准的最新技术水平如何？**我们分析将智能体到链上执行付诸实践的系统和提案，包括工具接口、钱包架构、账户抽象技术栈和基于意图的协议，并使用统一的编码标准进行比较。
- **RQ4：当前领域中最重要的开放研究问题和空白是什么？**我们提炼出 RQ1–RQ3 揭示的最关键空白，并将其归纳为一份 2026年研究路线图，重点关注缺失的接口层、可验证的策略执行以及可复现的评估实践。

2.2 检索策略：广泛撒网

我们的文献检索于2025年十月至2025年12月期间进行，特意涵盖学术文献和灰色文献。之所以这样做，是因为许多代理和区块链系统通过技术报告、改进提案、规范和开源文档不断演进，而这些资料往往先于常规同行评审出现（或根本不会进入同行评审）。例如，区块链技术栈中的关键基础设施产物，通常首先以公开技术报告或协议文档的形式记录，而非存档出版物 [50]。

2.2.1 检索渠道

我们检索了以下来源，以确保广泛覆盖：

- **同行评审渠道：** IEEE Xplore、ACM Digital Library、SpringerLink 和 Elsevier ScienceDirect。
- **预印本存档：** arXiv，重点关注 cs.AI、cs.CR、cs.DC 和 cs.SE。
- **技术标准和提案：** 以太坊改进提案（EIP），尤其是与账户抽象和交易格式相关的提案，以及定义互操作性接口的相关 RFC 和 Web 标准。
- **开源框架和行业文档：**代表性代理框架和 Web3 工具的代码仓库及技术文档，以及主要的安全性和基础设施提供商（例如钱包和中继器技术栈、预言机网络和自动化平台） [50]。

2.2.2 搜索查询

针对不同执行场所调整了搜索字符串，但遵循相同的概念结构：(i) 智能体系统，(ii) 区块链交互，以及 (iii) 连接两者的接口层。核心查询为：
(“人工智能智能体” OR “大语言模型智能体” OR “自主代理” OR “多智能体系统”) AND (“区块链” OR“
智能合约” OR “加密货币” OR “Web3” OR “去中心化金融”)
随后，我们进一步细化查询，以探查特定的接口子领域：

- (“代理” OR “大语言模型”) AND (“钱包” OR “交易签名” OR “密钥管理” OR “托管”)
- “工具协议” AND (“人工智能” OR “代理”)
- ("以意图为中心的" OR "以意图为中心的") AND ("区块链" OR "交易")
- "账户抽象" AND ("代理" OR "安全性" OR "EIP-4337")
- "策略引擎" AND ("区块链" OR "智能合约")

2.2.3 质量评估与偏见缓解

确保本综述的质量和客观性，需要系统地关注潜在偏见和质量问题。

来源质量评估 根据多项标准评估每个来源的质量，包括发表场所、作者资历、方法严谨性和引用影响力。在成熟交易场所发表的同行评审出版物，其权重高于预印本或非正式来源。不过，鉴于这一领域发展迅速，对于能够提供同行评审文献中未有的独特见解的预印本和技术文档，也将其纳入考量。

利益冲突考量 许多相关来源由在其所描述技术领域拥有商业利益的组织制作。钱包提供商、基础设施公司和协议开发者提供的文档可能反映的是宣传性观点，而非客观观点。在可能的情况下，我们会将利益相关方的声明与独立来源进行核实，或附加适当的限定说明。

时效性与稳定性之间的权衡 该领域的快速发展使得涵盖最新进展与确保所引用来源的稳定性之间产生了张力。非常新的来源可能会被修订或撤回，而较早的来源可能已经过时。本综述试图通过优先采用稳定来源来介绍基础概念，同时纳入近期来源以涵盖新兴发展，从而平衡这些考量。

地域与语言范围 本综述主要涵盖英文来源，这可能导致地域偏见，使英语国家和地区的发展得到更多关注。其他地区的重要发展可能未得到充分体现。读者应当注意这一局限，并在适当情况下查阅特定地区的来源。

2.2.4 方法论考量与局限性

本综述采用的系统综述方法反映了证据综合领域的既定最佳实践，同时也承认对快速演变领域进行综述所面临的独特挑战。智能合约的安全性已经得到广泛研究，为理解代理中介交易的风险奠定了基础 [244, 264]。跨链协议带来了必须仔细分析的额外复杂性 [241, 248]。去中心化身份系统的设计是实现代理身份验证的关键促进因素 [245, 262]。多智能体系统拥有悠久的研究历史，为现代智能体与区块链集成提供了参考 [277, 260]。近期关于基于大语言模型的智能体的研究为智能体设计提供了新的视角 [249, 221]。区块链系统中恶意行为的检测是一个活跃的研究领域 [275]。共识机制仍在持续演进，新协议在性能和安全性方面带来了改进 [219, 222]。使用强化学习训练人工智能智能体也已得到探索 [231, 232]。若干方法论考量值得明确讨论。

智能体人工智能和区块链技术的发展速度意味着，任何综述都有可能在发表后不久就过时。我们尝试通过关注基本原则、架构模式和长期存在的挑战，而非可能被取代的具体实现，来降低这一风险。不过，读者应注意，文中所描述的具体系统和能力可能已在本文撰写后发生变化。

代理与区块链整合的跨学科特性，使全面覆盖文献面临挑战。相关研究分散在计算机科学、密码学、经济学、法律及其他领域，各自拥有自己的

发表平台、术语和方法规范。我们的检索策略试图跨越这些学科广泛搜寻，但由于学科边界的存在，仍可能遗漏部分相关研究。
发表偏倚是另一个方法学方面的隐忧。成功的代理部署和正面结果比故障或负面发现更有可能发表。这种偏倚可能导致对智能体能力的评估过于乐观，并低估其中涉及的挑战和风险。我们尝试通过纳入灰色文献、行业报告和有记录的事件，并结合同行评审出版物，来平衡这种偏倚。

2.3 纳入与排除标准：界定范围

为确保语料库聚焦且相关，我们制定了纳入与排除标准，并由两名独立审阅者在筛选过程中应用这些标准；如有分歧，则由第三名审阅者解决。 [34] 研究或系统只有满足以下所有标准，才会被**纳入**：

- **必须包含智能体组件：**系统必须包含人工智能或自主智能体组件，能够进行非平凡的推理、规划或决策（超出确定性的硬编码规则）。
- **必须具备链上接口：**代理必须通过读取状态、构造候选交易或意图，和/或通过签名或策略门控路径授权执行，与运行中的区块链环境交互。纯链下智能体系统不予纳入。
- **必须描述一种接口、协议或架构：**该研究必须对智能体与区块链如何连接提供技术见解（例如工具接口、交易／意图格式、托管与授权架构、执行流水线或策略执行）。仅提出某种应用结果（例如“使用人工智能预测代币价格”）而未说明互操作机制的研究不予纳入。

相反，我们**排除了**以下研究：

- 纯属理论性智能体论文（例如形式化智能体理论或博弈论），且没有具体的链上实现、接口规范或可执行架构。
- 未涉及与互操作性相关的智能体或人工智能要素的区块链论文（例如仅涉及共识或密码学的研究）。
- 非英语出版物。
- 缺乏足够技术细节以供复现分析的营销材料、新闻文章或社交媒体帖子。
- 已被同一贡献的更新且内容大幅更完整的版本明确取代（例如，正式出版物取代了早期的研讨会论文或预印本版本）。

2.4 PRISMA式筛选流程：证据漏斗

我们的研究筛选遵循 PRISMA 2020 的四个阶段：识别、筛选、资格评估和纳入 [233]。图 1 总结了记录的流转情况以及由此形成的语料库。

1. **识别：** 我们从学术数据库中检索到 2847 条记录，并从灰色文献来源中检索到另外 423 条记录，合计 3270 条记录。2. **筛选：** 移除 1114 条重复记录后，我们根据标题和摘要筛选了 2156 条记录，并排除了 1542 条超出范围的记录（例如，区块链用于无关的应用领域，或不包含智能体组件的人工智能系统）。3. **资格评估：** 我们对 614 篇报告的全文进行了资格评估，并排除了 297 篇报告，主要原因包括：关于智能体—链接口的技术细节不足（n=112）、仅作概念性或理论性阐述而未涉及接口架构（n=89），或已被更新且更完整的版本取代（n=54）。其余排除均基于其他有记录的理由（n=42）。4. **纳入：** 最终的定性综合纳入了 **317** 项独特的研究、系统和标准。从中，我们选取了 **85** 个实现细节充分且具备公开文档的系统，用于比较能力矩阵中的定量编码。

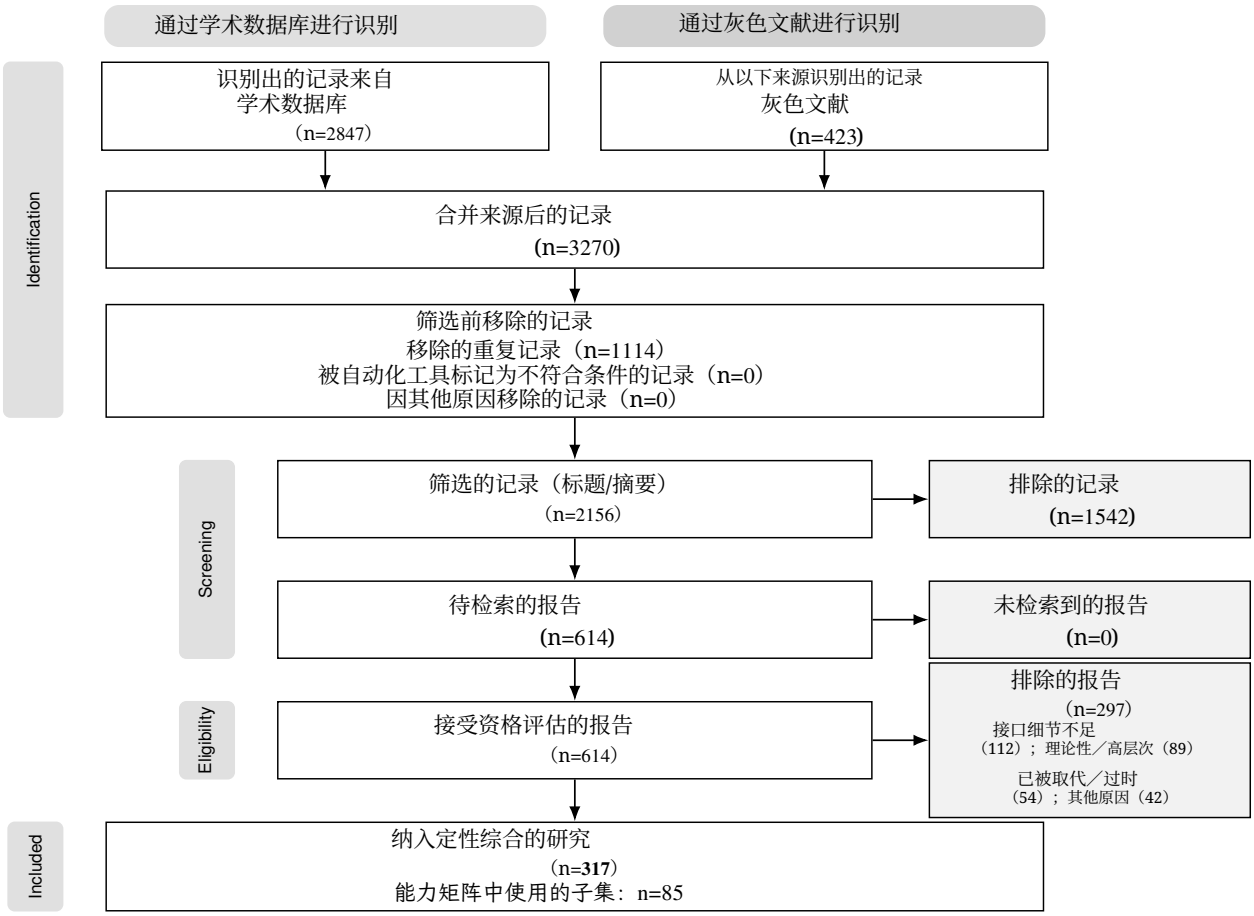

图1：PRISMA 2020年风格的流程图，总结了本综述所采用的系统性检索和筛选流程。

可复现性资料包。为使本综述便于审计和扩展，我们提供了一份可复现性资料包（作为补充材料和/或存放于公共代码库），其中包含：(i) 最终确定的数据库查询字符串和日期范围；(ii) 去重后的参考文献导出文件；(iii) 筛选日志及纳入/排除理由；以及 (iv) 用于填充比较矩阵的编码手册和提取表。在许可允许的情况下，我们还会提供每个分析系统的制品元数据（例如代码库链接和版本标签）。

2.5 提取属性的编码方案：创建结构化比较

对于选定进行深入分析的 85 个系统，我们提取了一组统一的 **13** 项属性，用于填充能力矩阵。该编码方案旨在捕捉架构选择和安全相关设计决策，以便对异构实现进行结构化的同类比较。每个编码值均有证据指针（例如规范章节、文档页面或代码库制品）作为依据，并记录在提取表中。

（A1）接口层：向代理开放的主要集成接口（例如工具协议、钱包 API、智能合约接口、中继器服务）。**（A2）执行路径：**操作从意图到链上提交的流转方式（例如直接通过钱包广播、中继器或聚合器提交、求解器中介执行、捆绑器与入口点流程）。**（A3）交易表示：**所支持的最高层级表示形式（例如原始交易、结构化操作、意图对象、批处理或原子捆绑包）。**（A4）托管模型：**签名权限的持有和行使方式（例如用户持有的密钥、委托签名、多方计算或门限签名、基于可信执行环境的签名） [184, 169]。

（A5）授权机制：具体的授权原语（例如外部拥有账户签名、智能账户验证、会话密钥、基于角色的签名、多重签名或门限审批）。**（A6）权限模型：**系统限制代理可执行操作的方式（例如允许列表、支出限额、方法级权限、能力式授权、限定范围的会话）。**（A7）策略控制：**授权或提交前是否强制执行运行时策略（例如基于规则的防护机制、可编程策略即代码、情境约束、合规性检查）。**（A8）可观测性：**审计日志、交易预览与模拟、监控钩子以及代理操作可追溯性的可用情况（适用时还包括工具调用的来源信息）。**（A9）恢复机制：**是否支持事件响应和安全故障处理（例如紧急停止开关、密钥轮换、紧急延迟、撤销、社交恢复、争议处理流程）。**（A10）最大可提取价值与对抗性执行防护：**设计是否考虑内存池对手和可提取价值（例如私有订单流、拍卖、滑点约束、提交—揭示模式、时序防护）[80]。**（A11）多链与跨域支持：**系统是否可在多条区块链或多个 Rollup 上运行，以及是否公开链特定假设（费用市场、账户模型、最终性语义）。**（A12）信任假设与依赖项：**所需的外部信任和基础设施依赖项（例如 RPC 提供商、中继器、求解器、可信执行环境、预言机），以及这些依赖项被视为可信、需许可还是对抗性的。**（A13）威胁假设：**其目标威胁模型的明示或隐含内容（例如密钥泄露、提示词或工具注入攻击、恶意去中心化应用、内部人员风险、内存池或最大可提取价值对手）。

为提高单人编码条件下的一致性，我们使用了书面编码手册，为每项属性设定标准化标签，为每个编码值记录证据链接，并在最终确定矩阵前，对有歧义的案例进行第二轮重新编码，同时明确记录决策日志。

能力指标的编码规则。 对于能力矩阵（表6），我们采用三级支持标准：✓ 表示系统提供一流且有文档说明的支持，并将其作为稳定功能提供；～表示部分、间接或依赖外部条件的支持（例如，通过插件、自定义集成实现，或默认执行路径不支持）；– 表示不支持或缺少相关支持文档。描述性列（类型、工具接口、链范围、托管模型）直接依据官方文档和/或参考实现进行编码。对于能力列，我们采用以下标准：

- **意图：** ✓ 如果系统支持可编译为交易的声明式意图或结果级对象；～如果系统支持结构化操作/模板，但不支持真正的意图编译；– 如果系统仅支持原始交易构造。
- **签名：** ✓ 如果系统提供明确的签名或授权路径（外部拥有账户签名、智能账户验证、委托/多方计算签名或会话密钥签名）；～如果只能通过系统外部的钱包进行签名；– 如果不支持签名或授权。
- **策略：** ✓ 如果策略检查可以在签名或提交前阻止/批准操作（例如允许列表、支出限额、规则引擎）；～如果存在约束条件，但较为粗略或间接；– 如果未提供策略门控。
- **预览：** ✓ 如果系统提供人类可读的交易/操作预览（已解码的调用数据、风险标记、摘要）；～如果预览仅通过第三方工具提供；– 如果未提供预览。
- **模拟：** ✓ 如果系统支持执行前模拟（试运行、状态差异、revert 检测、结果估算）；～如果只能通过外部服务进行模拟；– 如果不支持。
- **可观测性：** ✓ 如果操作和执行提供审计日志与监控；～如果日志不完整或间接；– 如果除链上跟踪记录外，系统未提供有意义的可追溯性。
- **恢复：** ✓ 系统是否支持恢复控制（撤销、延迟、紧急停止开关/暂停、密钥轮换、紧急工作流）；～恢复是否受到限制或依赖外部支持；– 是否未提供恢复机制。

3 背景：基础构件

要理解智能体—区块链互操作性的挑战，仅对底层技术有表层认识是不够的。智能体并不是简单地与区块链“对话”；它通过一个多阶段流程运行，在这一流程中，错误会在感知、规划、授权、执行和验证各环节不断累积。本节

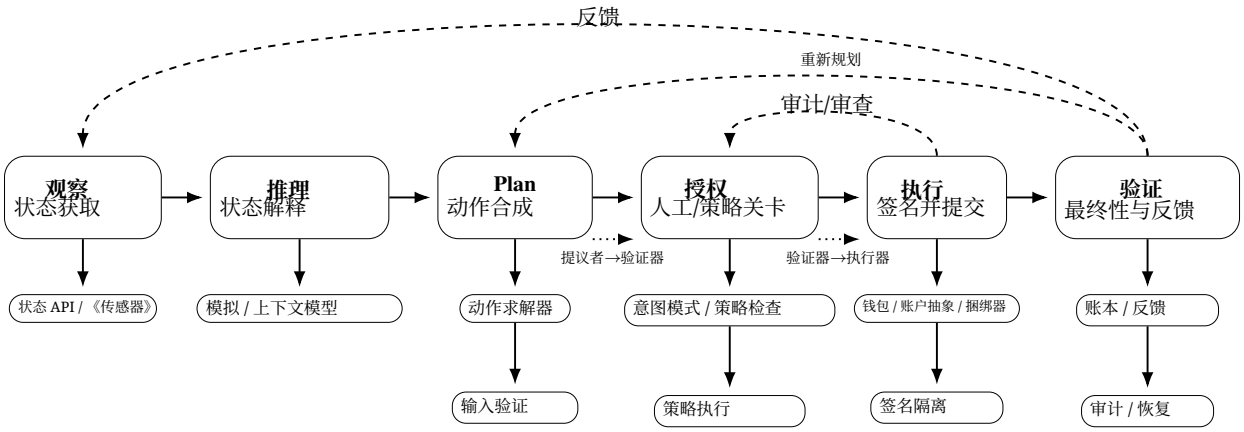

图2：单色六阶段代理行动流水线。顶行展示核心阶段，中间行概述常见集成接口，底行则突出典型控制措施。虚线箭头表示反馈回路；点线箭头表示可选的多代理角色。账户抽象执行路径通常由打包器和智能账户中介 [55]。

将对支持代理驱动链上操作的基础原语进行解构。我们重点介绍全文中反复使用的三个构建模块：(i) 代理操作流水线；(ii) 钱包从外部拥有账户（EOA）向可编程智能账户的演进；以及 (iii) MEV 在塑造对抗性执行中的作用。

3.1 代理操作流水线：详细解构

代理中介的链上操作的生命周期可以建模为一个六阶段流水线，如图2所示。该模型提供了一个与实现无关的框架，可用于分析集成模式、追踪观察到的状态如何转变为不可逆的状态转换，并确定安全控制应在哪些环节介入。

流水线概述。 我们将代理与链的交互执行建模为一个六阶段流水线，将观察到的状态转化为不可逆的状态转换。这些阶段刻意不依赖具体实现，适用于自托管代理、由智能账户中介的执行（包括账户抽象），以及企业级或委托托管系统。此抽象很有用，因为现实中的故障往往横跨多个层面。例如，观察阶段发生的提示注入事件可能会破坏规划，而对抗性执行造成的影响则可能使原本正确的规划失效。

1. **观察（状态获取）**。 代理会读取链上 状态（RPC读取、日志、索引器）、内存池 状态（待处理交易、费用条件），以及链下状态（价格、风险信号、治理公告及其他外部背景信息）。此阶段容易受到数据欺骗、预言机操纵、索引器入侵，以及通过不可信文本源实施的间接提示注入攻击。

2. **推理（状态解释）**。代理将异构信号转换为结构化的世界模型（头寸、约束、目标和风险态势）。当检索内容未与可信策略和系统提示清晰区分时，以大语言模型为中心的代理容易受到上下文混淆、幻觉式不变量和指令遵循失败的影响。3. **规划（动作合成）**。代理选择协议并构造候选操作（兑换、存入、借入、跨链桥、治理投票或多步骤工作流）。规划必须考虑链语义（代币小数位、授权、Permit标准）、经济约束（流动性、滑点）和对抗性执行动态（抢先交易、三明治攻击和尾随交易）[80]。4. **授权（人工和/或策略门控）**。在执行任何不可逆操作之前，应根据用户确认、支出限额、允许列表、速率限制、基于模拟的防护机制和策略引擎等约束检查代理的提案。在此阶段，意图表示和策略证明有助于减少代理意图与链上实际执行内容之间的歧义。5. **执行（签名与提交）**。系统生成密码学授权（外部拥有账户签名、智能账户验证、MPC签名和/或会话密钥），并通过公共内存池、私有中继、打包器或求解器网络提交交易。密钥泄露、会话密钥权限过度、中继操纵和交易替换是此阶段的主要风险。

6. **验证（最终性与反馈）**。系统监控纳入情况，处理链重组，验证事后状态（余额、头寸、发出的事件），并在执行结果偏离意图时触发补救（例如部分成交、回滚或意外授权）。稳健的代理会将验证视为首要环节，记录可审计日志，并仅在达到充分最终性后更新长期记忆。

流水线中的最大可提取价值为何重要。与传统 Web API 不同，区块链执行在默认情况下具有经济层面的对抗性。向公共内存池公开的有利可图操作可能被搜索者重新排序、复制或遭受三明治攻击，因此执行风险是环境固有的一部分 [80]。因此，更安全的智能体架构会结合*事前*控制措施（模拟、滑点界限、私有订单流、基于求解器的执行）与*事后*控制措施（验证、恢复工作流，以及明确的人工升级路径）。本文将始终使用这条流水线来组织威胁分类体系和实用安全检查清单。

3.1.1 面向代理开发者的智能合约基础

要有效实现智能体与区块链集成，必须深入理解智能合约机制。本节为代理开发者提供必要的背景知识；他们可能比起区块链技术，更熟悉人工智能系统。

合约状态与存储 智能合约会在合约存储中维护持久状态；合约存储是一种键值存储，其中每个 256 位密钥都映射到一个 256 位值。存储操作是成本最高的区块链操作之一，其中写入的成本远高于读取。与合约交互的代理系统必须了解存储模式，才能预测 Gas 成本并优化交易构造。

常见的存储模式包括映射（基于哈希值的键值关联）、数组（按顺序排列的数据结构）以及紧凑存储（在单个存储槽中编码多个值）。不同模式具有不同的 Gas 成本特征和访问模式，这会影响代理应如何与其交互。

函数选择器与 ABI 编码 合约函数通过函数签名的哈希值派生出的四字节选择器进行标识。参数根据应用程序二进制接口（ABI）规约进行编码，该规约为不同数据类型定义了标准编码方式。代理系统必须正确编码函数调用，才能成功与合约交互。

复杂类型（数组、结构体、嵌套类型）的 ABI 编码可能十分复杂，编码错误会导致失败交易或出现意外行为。代理系统应使用成熟的 ABI 编码库，而不是从头实现编码逻辑。

事件与日志 合约会发出事件，以记录重要的状态变更并向链下观察者提供信息。事件存储在交易日志中，其成本低于合约存储，但链上代码无法访问。代理系统高度依赖事件来监控合约状态并检测相关活动。

按主题筛选事件，可以在大量区块中高效检索特定类型的事件。代理系统应设计事件监控策略，在全面性与效率之间取得平衡，筛选相关事件，同时避免产生过多的远程过程调用负载。

错误处理与回滚 合约执行可能因回滚而失败，回滚会撤销所有状态变更，并将剩余的 Gas 返还给调用方。回滚可能由智能合约显式触发（通过 require、revert 或 assert 语句），也可能由运行时错误隐式触发（Gas 耗尽、无效操作）。智能体系统必须妥善处理回滚，解读错误消息并据此调整策略。

近期 Solidity 版本引入的自定义错误类型可提供结构化错误信息，供智能体解析并据此采取行动。代理系统应设计为能够解读旧式字符串错误和现代自定义错误。

3.1.2 钱包安全的历史视角

自加密货币发展早期以来，区块链钱包的安全状况已发生显著变化。了解这一演变有助于理解当前的最佳实践和新出现的挑战。

早期安全挑战 最早期的加密货币用户面临着重大的安全挑战，而当时的工具支持十分有限。私钥管理完全依赖手动操作，用户需要负责生成、存储和保护密码学密钥。许多早期用户因密钥丢失、被盗或运维错误而失去了对资金的访问权限。

2014 年 Mt. Gox 交易的崩溃凸显了中心化托管的风险：单点故障可能导致灾难性损失。这一事件推动了更完善的安全实践和工具的发展，包括硬件钱包 [167]s 和多重签名方案。

硬件钱包的发展 硬件钱包的出现，是为了应对纯软件密钥存储在安全性方面的局限。通过将私钥隔离在专用安全硬件中，这些设备能够防范恶意软件、网络钓鱼及许多其他攻击向量。如今，硬件钱包已成为保护大额加密货币持有资产的标准做法。

然而，硬件钱包也带来了自身的挑战，包括物理安全、供应链完整性和可用性限制。硬件设备需要进行物理交互，这会造成操作阻力，可能与自动化智能体运行不兼容。

多重签名的演进 多重签名方案从简单的 m-of-n 配置发展为具备可编程授权逻辑的复杂智能合约实现。现代多重签名钱包支持时间锁、支出限额和基于角色的访问控制等功能。

多重签名技术的发展为可编程授权模型奠定了基础，使代理集成成为可能。多重签名钱包证明了授权逻辑可以定制并自动化，从而为更复杂的委托方案铺平了道路。

当前安全状况 当前安全状况反映了十多年来运行实践所积累的经验教训。最佳实践包括纵深防御、职责分离和持续监控。然而，人工智能智能体的集成带来了新的挑战，现有安全框架未必能够充分应对。

3.1.3 区块链钱包的演变

区块链钱包的演变反映了整个生态系统的成熟过程：从面向爱好者的技术发展为支持主流应用的基础设施。了解这一演变过程，对于代理—钱包交互的设计至关重要。

第一代钱包是简单的密钥管理工具，用于存储私钥和签署交易。用户需要对密钥安全、备份与恢复承担全部责任。私钥一旦丢失，就意味着永久失去对关联资金的访问权限，且无法恢复。这一模型虽然实现了最大程度的去中心化，但由于容错性极低且用户体验欠佳，不适合主流采用。

第二代钱包在可用性和安全性方面进行了多项改进。硬件钱包将私钥隔离在安全元件中，从而防范基于软件的攻击。多重签名方案支持共享托管安排和组织控制。分层确定性（HD）钱包通过助记词种子短语简化了备份与恢复。这些创新提升了安全性和可用性，但仍保留了显式密钥管理这一基本模型。

以智能合约钱包和账户抽象为代表的第三代钱包，代表了向可编程托管转变的范式。此类钱包不再仅仅依赖密码学密钥持有，而是支持将任意授权逻辑编码在智能合约中。这种可编程性实现了社交恢复、支出限额、会话密钥和自动化策略等功能，而这些功能在传统外部拥有账户中是不可能实现的。

对于代理系统而言，这一演进尤为重要。智能合约钱包为细粒度委托奠定了基础，使主体能够向智能体授予特定且有界的权限，而不是提供全有或全无的访问权限。这些钱包的可编程特性使授权策略能够根据不同智能体用例的具体要求进行定制，在自主性与适当约束之间取得平衡。

3.1.4 区块链技术的持续演进

区块链生态系统持续快速发展，新的共识机制、扩展性解决方案和应用范式正以越来越快的速度涌现。公平交换协议支持参与方之间无需信任的交易 [106]。区块链在包括医疗保健在内的各个行业中的应用已得到广泛探索 [155, 127]。跨链互操作性仍是该生态系统面临的关键挑战 [121]。区块链系统的经济学（包括补贴机制）已得到深入研究 [122]。区块链在物联网和机器人应用中的使用也得到了探索 [162]。智能合约安全仍是一个重要的研究领域 [164]。支付通道网络支持可扩展的链下交易 [166]。研究人员已利用零知识证明开发出隐私保护智能合约 [172]。近期的综述对区块链格局进行了全面的

概述 [109]。这一演变对智能体与区块链集成具有深远影响，因为智能体必须经过设计，以适应不断变化的协议语义、新的交易类型以及不断发展的最佳实践。
主流网络从工作量证明向权益证明共识的转变，改变了交易纳入的经济动态，为代理介导的执行带来了新的机遇与挑战。按权益加权的验证者选择带来了不同于算力竞争的博弈论考量，影响代理在交易时机和费用优化方面应采用的策略。同样，应用专用区块链和 Rollup 的激增造成了碎片化的格局，代理必须在其中应对各异的执行环境，而这些环境具有不同的安全保障和最终性特征。

模块化区块链架构将共识、数据可用性和执行拆分为不同层，这一发展对代理系统尤为重要。这些架构支持更灵活的部署模式，使代理能够选择针对特定用例优化的执行环境，同时继承共享共识层和数据可用性层的安全性。理解这些架构模式，对于设计能够在不断演变的区块链格局中高效、安全运行的代理至关重要。

3.1.5 代理流水线的实现考量

GOALIATH 理论为理解人类和人工智能体的目标导向行为提供了一个认知框架，可为设计更稳健的智能体规划系统提供参考 [148]。长期机器人自主性研究为设计能够在无人为干预的情况下长时间运行的代理提供了宝贵经验，其中包括自我监控和适应的必要性 [175]。

代理操作流水线的实际实现涉及许多工程决策，这些决策会影响系统可靠性、安全性和性能。交易模拟工具使智能体能够在将操作提交到区块链之前预览其影响 [294, 295]。可以使用静态分析工具验证智能合约的安全性 [300]。多智能体协调需要谨慎的协议设计 [298, 304]。强化学习技术可用于训练智能体执行复杂任务 [288, 299]。不同区块链系统之间的互操作性是一项关键挑战 [278, 303]。可验证凭证支持隐私保护认证 [283]。多智能体系统可用于解决复杂问题 [286]。智能合约系统的设计需要仔细考虑安全性和可用性 [293, 290]。零知识证明支持隐私保护交易 [307]。人工智能驱动的智能体正越来越多地应用于区块链应用 [306]。流水线的每个阶段都有各自的实现挑战和设计权衡。

观察阶段必须在全面性与效率之间取得平衡。智能体需要访问相关的区块链状态、市场数据和上下文信息，但处理过多数据可能会使智能体的推理能力不堪重负，并增加延迟。有效的观察策略会通过筛选、聚合和优先级排序，向智能体提供与当前任务最相关的信息。

推理阶段或许是最难可靠实现的阶段。基于大语言模型的推理本质上具有概率性，可能产生不一致或错误的输出，尤其是在处理复杂的多步骤任务时。思维链提示、自洽性检查和集成方法等技术可以提高推理的可靠性，但也会增加计算开销和延迟。对于时间敏感型区块链操作而言，推理质量与响应时间之间的权衡尤为突出。
构建阶段必须生成语法正确且语义有效的交易。这需要精确处理数据类型、编码格式和协议特定约定。交易构造中的错误可能导致交易失败、浪费燃料费，甚至造成非预期的价值转移。稳健的构建实现会采用多层验证机制，包括模式检查、模拟，以及针对高风险操作的人工审查。

3.2 账户抽象（ERC-4337[2]）

推动链上自动化变得更安全、更易用的关键因素之一是**账户抽象（AA）**。在实践中，**ERC-4337**是以太坊和 EVM 兼容链上影响力最大的实现方案。它引入了可编程的“智能账户”，能够执行自定义验证规则，同时兼容现有的共识规则和交易流程 [55]。对于智能体系统而言，这一转变具有奠基性意义：钱包不再被视为拥有无条件权限的单一私钥，而是借助账户抽象转变为可编程的安全边界，能够在链上检查约束条件，然后再执行操作。

3.2.1 EOA/CA 二分法及其局限

以太坊历来区分两种账户类型：

1. **外部拥有账户（EOA）**。由私钥控制的账户。EOA 可以发起交易、持有资产，也是常见钱包（例如 MetaMask 风格账户）中的默认账户。2. **合约账户（ CA ）**。由代码控制的账户。CA 可以编码复杂逻辑，但无法自行发起交易；只有在交易调用时才会响应。

这种僵化的二分法给可用性和安全性都带来了摩擦。多重签名授权、支出限额、社交恢复或应用专属权限等可编程钱包功能，自然应由智能合约代码实现。然而，合约钱包仍需要由 EOA 资助的交易来“唤醒”并支付 Gas，这在历史上迫使用户采用双层架构：由 EOA 发起并资助交易，再由合约钱包提供策略逻辑。对于代理部署而言，这一局限更加突出：赋予代理 EOA 级别的签名权限，会使权限边界收缩为单一密钥；而仅使用合约的设计则难以妥善解决交易发起和费用支付体验问题。

3.2.2 ERC-4337：无需协议硬分叉的范式转变

账户抽象的概念在以太坊社区中由来已久，早期提案阐述了核心思想，而这些思想最终将在 ERC-4337 [54] 中得以实现。

早期的账户抽象提案需要进行协议变更（硬分叉），而这类变更既缓慢，又在政治层面代价高昂。 [246] ERC-4337 无需改变共识即可实现账户抽象：它引入了一种更高层级的操作对象——用户操作（UserOperation），并通过名为入口点（EntryPoint）的规范链上合约进行处理 [55]。从概念上讲，ERC-4337 在常规交易之上增设了一条“执行通道”：

- 用户操作（UserOperation）。 一种结构化的伪交易，用于描述智能账户要执行的操作。用户操作通常包括智能账户地址（发送方）、描述该操作的调用数据、Gas 和费用字段，以及授权载荷。这种结构尤其适用于智能体系统，因为代理可以在操作描述这一层构造操作，而无需直接处理传统交易格式的底层细节 [55]。
- **打包器**。 专门用于监控 UserOperation 内存池、将多个操作组装成一笔标准交易，并将该交易提交给 EntryPoint合约的节点。打包器被设计为无需许可的基础设施：原则上任何人都可以运行打包器，并且打包器会通过 ERC-4337 经济机制获得其所消耗 Gas 的补偿 [55]。
- 入口点。 一个单例合约，负责协调 ERC-4337 操作的验证与执行。在实际运行中，入口点会为每个操作强制执行两阶段流程：（i）验证 阶段，调用由账户定义的验证逻辑（通常为 validateUserOp）；以及（ii）执行 阶段，在验证成功后分发所请求的调用。这种设计将关键安全检查集中在一条链上、可审计的路径中，并统一智能账户表达授权逻辑的方式 [55, 280]。
- **支付主**。 可选合约，可用于赞助 Gas 或强制执行赞助约束。用户操作可以请求支付主，入口点将根据其策略验证支付主并向其收费。支付主支持“免 Gas”入驻，也使应用能够在明确的约束条件下赞助执行；当智能体不应持有不受限制的链的原生代币余额时，这一点尤为有用 [55, 280]。

安全提示：账户抽象并非天然安全。账户抽象提升了灵活性，但也扩大了出错的设计空间。设计不当的验证逻辑可能重新引入类似“外部拥有账户”的风险（权限过宽、约束条件薄弱），而不安全的支付主控策略则可能为滥用行为提供补贴。此外，在许多环境中，捆绑器和内存池动态仍具有对抗性，因此，要在代理部署中实现有意义的安全性，账户抽象必须配合谨慎的策略设计、模拟/预览工具以及恢复控制。

3.2.3 对代理系统的启示

ERC-4337 是链上代理的实用安全基础层，原因有四。

- **可编程授权**。 智能账户可以将授权策略直接编码到链上验证逻辑中：会话密钥、时间界限、函数级允许列表、价值限制、多方授权以及条件执行检查。这样，代理便可在范围严格受限的权限下运行，并由链在提交时强制执行，而无需将主私钥交由代理保管 [55]。

- **将签名与 Gas 支付解耦**。 借助支付主，系统可以授权代理操作，而无需代理持有原生 Gas 代币。这支持这样的代理架构：代理的权限仅限于操作授权，无法随意清算或再平衡以维持燃料储备，同时仍可在由应用控制的赞助策略下执行操作 [55, 280]。
- **标准化与互操作性**。 ERC-4337 提供了一种标准化流程（用户操作 → 捆绑器 →入口点），不同的智能账户实现可以共用该流程。这减少了定制钱包集成，并为合规账户中的中间件和代理提供了通用接口，有助于避免供应商锁定和集成碎片化 [55]。
- **模块化与可扩展性**。 账户抽象生态系统天然具有模块化特征：账户实现、验证模块（例如会话密钥）、恢复逻辑和支付主可以相对独立地演进。对于代理式应用，这使安全配置能够根据风险承受能力和运营需求进行调整，同时保持一致的执行骨干 [280]。

总之，ERC-4337 将钱包从被动的密钥持有者转变为可编程的安全平台。对于自主或半自主的链上系统，它提供了一种有原则的替代方案，不再将不受限制的私钥控制权授予代理，而是通过智能账户验证来强制执行按能力范围限定的授权，并辅以预览/模拟、监控和恢复。

3.3 矿工/最大可提取价值（MEV）

若不清楚理解**最大可提取价值（MEV）**，对链上执行环境的讨论便不完整。在许多区块链系统中，尤其是去中心化金融领域，MEV 是一种结构性经济力量；从普通用户和自动化智能体的角度来看，它往往具有对抗性。按照标准定义，MEV 指能够影响区块内交易纳入 和排序的参与者可获取的利润，包括通过插入、排除或重新排序交易来创造有利的执行结果 [80]。对于以机器速度反复执行操作的人工智能智能体而言，将 MEV 视为边缘情况，无疑会导致系统性的价值泄漏。

3.3.1 从矿工到最大可提取价值：MEV 的演变

这一概念最初在工作量证明系统中被称为矿工可提取价值，反映了矿工对区块构建的控制权，以及他们对公共内存池中待处理交易的可见性。早期对去中心化交易所的实证观测记录了这一核心现象：通过对待处理的用户交易进行重排序和插入攻击，可以提取价值 [80]。以太坊转向权益证明后，该术语演变为最大 可提取价值，以强调价值提取并不局限于矿工。相反，专业化参与者和基础设施构成的供应链可以共同塑造交易排序和路由，从而获取这些价值，即便最终的区块提议者本身并未进行策略搜索。

一种有用的抽象方式是将以下环节区分开来：**(i) 策略发现**（识别有利可图的重新排序或插入操作）、**(ii) 捆绑构建**（对原子交易序列进行编码），以及 **(iii) 区块组装与选择**（选择收益最大化的候选区块）。当代综述和缓解研究描述了这一不断扩大的生态系统及其对安全性、去中心化和市场效率的影响 [18]。

3.3.2 最大可提取价值攻击的分类体系

近期研究致力于对最大可提取价值策略进行更细致的分类，这对于制定有针对性的缓解技术至关重要 [129]。了解智能体易受哪些特定类型的最大可提取价值影响，是当前研究的重点领域。

对于代理式执行，最大可提取价值最常表现为经济结果恶化，而非明显的协议破坏。 [291]常见模式包括：

- **抢先交易**。 对手观察到一笔会影响价格或有利可图的交易后，会提交一笔竞争性交易，并通过提供更高的有效费用，使其在目标交易之前 执行，从而攫取随后的价格变动所带来的利润 [80]。
- **尾随交易**。 对手安排一笔交易，使其在目标交易之后立即执行，以捕获该交易创造的机会，例如大额兑换所带来的套利 [80]。
- **三明治攻击**。 一种结合抢先交易和尾随交易的复合策略。对手在受害者交易之前，将执行价格推向对受害者不利的方向，随后在交易之后回滚，主要从受害者的滑点预算中提取价值。三明治攻击已经得到实证和理论层面的形式化研究，并且仍然是针对朴素去中心化交易所执行的主要攻击类型之一 [80, 174]。

这些模式对智能体尤其具有破坏性，因为智能体往往行为一致、能够大规模重复执行操作，并可能表现出可预测的执行策略。如果没有考虑最大可提取价值的约束条件，智能体就可能成为可提取价值的持续来源。

3.3.3 智能体系统的影响与缓解措施

对于人工智能智能体而言，公共内存池并非中立的队列，而是一个公开的信息渠道，有价值的意图可能在其中被发现并遭到策略性利用。因此，实用的缓解措施需要将执行时的预防性控制与稳健的执行后验证相结合：

- **减少公开的意图泄露**。 在可行的情况下，避免将有价值的操作广播到公共内存池，并优先选择能够减少交易前可见性或仅暴露最少信息的执行路径。以缓解措施为重点的综述对这些方法进行了系统梳理，其中包括订单流保护，以及旨在降低可提取价值的重新设计 [18]。
- **优先采用 MEV 感知型执行机制**。 批量拍卖、求解器中介执行和基于意图的结算，可以将竞争转移到受控机制中，从而减少某些在内存池中可见的攻击面。这些方法无法消除对抗性激励，但可以改变攻击面以及可提取价值的分布 [174, 18]。
- **设置严格的约束条件**。 采用严格的滑点限制、最小输出边界和价格保护机制，以限制最坏情况下的损失。这些控制措施无法防止所有形式的最大可提取价值，但可以通过在执行超出不可接受的阈值时触发回滚，限制对手能够提取的价值数量 [80]。
- **在相关情况下，探讨协议层防御措施**。一些研究提出了抗最大可提取价值的协议设计或替代执行层，旨在降低提取策略的可行性或盈利能力。尽管这类设计尚未得到普遍部署，但它们阐明了设计空间及其权衡取舍 [240]。

总之，最大可提取价值是链上执行环境的一项基本属性，而非罕见异常。设计代理时，必须假定公开到公共订单流中的有价值操作会被观察到并被采取行动。一项保守的设计原则是：**除非执行策略和约束条件已考虑最大可提取价值，否则应避免将有价值的意图暴露到公共内存池中** [80]。

3.3.4 拜占庭容错的历史基础

分布式共识的理论基础比区块链技术早了数十年。拜占庭将军问题于 20 世纪 80 年代初首次得到形式化阐述，它确立了基本的不可能性结果和下界，这些至今仍对现代分布式系统构成约束。关于实用拜占庭容错（PBFT）的奠基性研究展示了如何构建能够容忍拜占庭故障的复制状态机 [61, 62]。近期的综述全面概述了拜占庭容错领域，涵盖经典协议和较新的协议 [111]。研究人员还探讨了移动区块链系统的安全性，重点关注在移动环境中运行所面临的独特挑战 [86]。研究人员也探索了区块链技术在供应链管理中的应用，重点关注其安全与透明度优势 [105]。Bedrock 协议为构建高性能、可扩展的拜占庭容错系统提供了一种新方法 [21]。T-Chain 架构提出了构建可扩展、高效区块链系统的一种新方法 [115]。这些基础性成果表明，要在任意（拜占庭）故障存在的情况下实现共识，至少需要 $3f + 1$ 个节点才能容忍 f 个故障参与者；这一界限至今仍是许可型区块链系统和多方计算协议设计的核心。

拜占庭容错（BFT）共识的实际实现取得了显著进展，实用拜占庭容错（PBFT）的提出证明，在现实网络条件下，BFT 可以在性能可接受的情况下实现。后续的优化（包括 HotStuff 及其变体）将每个视图的通信复杂度从 $O(n^2)$ 降低到 $O(n)$，使拜占庭容错共识能够扩展到更大的验证者集合。这些进展与智能体区块链系统直接相关，因为它们决定了智能体在构造并提交交易时必须推理的信任假设和最终性保证。

3.3.5 最大可提取价值供应链及其对智能体的影响

最大可提取价值供应链已发展成为一个成熟的生态系统，其中有多个专业化参与者，各自承担不同角色，从交易排序中提取并分配价值。对于希望防范最大可提取价值提取或参与最大可提取价值捕获的智能体而言，了解这条供应链至关重要。

供应链的底层是搜索者，他们是通过监控待处理交易、分析智能合约状态并模拟潜在提取策略来识别最大可提取价值机会的专业化参与者。搜索者在基础设施方面投入巨大，以实现低延迟的内存池访问、强大的模拟能力和策略开发。搜索者之间竞争激烈，成功的搜索者往往处于算法交易和区块链技术的前沿。

构建者将搜索者和其他来源的交易汇总为完整区块，在遵守协议规则和构建者特定策略的同时，优化总可提取价值。在权益证明以太坊中，区块构建与区块提议分离后，构建者这一角色应运而生，由此形成了区块构建市场，实现了专业化和竞争。构建者通过向提议者提供更高的付款来竞争交易纳入，从而推动 MEV 提取效率的提升。
提议者（权益证明系统中的验证器）从构建者提供的区块中进行选择，并将其提议给网络。借助 MEV-Boost 等机制，提议者可以将区块构建外包给构建者，同时保留在竞争区块中进行选择的权利。这种分离使提议者无需开发专门的提取能力，也能获取 MEV，同时形成一个能够高效分配区块空间的市场。

对于代理系统而言，这条供应链既带来了风险，也创造了机会。通过公共内存池提交交易的代理会暴露于搜索者的 MEV 提取之下。反过来，能力出众的代理也可以亲自充当搜索者，捕获因市场低效或其他代理活动而产生的 MEV 机会。设计代理的交易提交策略时，必须考虑这一竞争格局。

面向智能体的最大可提取价值保护策略 有几种策略可以帮助代理降低最大可提取价值暴露。通过 Flashbots Protect 等服务或由构建者运营的私有内存池进行私有交易提交，可以防止搜索者观察待处理交易并制定提取策略。不过，私有提交会带来对私有内存池运营方的信任依赖，而且可能无法完全防范所有形式的最大可提取价值。

通过 CoW Protocol [247]或 1inch Fusion 等协议进行基于意图的执行，会将交易构造委托给相互竞争、以提供最优执行为目标的专用求解器。这些协议可以将原本会被外部搜索者提取的最大可提取价值纳入内部，从而通过更优价格将价值返还给用户。不过，意图协议也会引入自身的信任假设，且未必适用于所有交易类型。

最大可提取价值感知型交易构造旨在设计交易，以尽可能减少可供提取的机会。相关技术包括设置严格的滑点限制、将大额订单拆分为多笔交易、安排交易时机以避开可预测的模式，以及使用承诺-揭示方案，在执行前隐藏交易详情。这些技术需要深入理解最大可提取价值动态，而且未必能彻底消除价值提取。

3.4 多方计算（MPC）

账户抽象在钱包逻辑层增加了可编程性，而另一种互补的原语**多方计算（MPC）**则通过解决链上自动化中最常见的灾难性故障模式之一——单个签名密钥遭到安全失陷——来强化钱包密钥托管层。多方计算使多个参与方能够基于私有输入共同计算某个函数，而不向彼此透露这些输入。在钱包工程中，应用最广泛的多方计算是**门限签名**：分布式的密钥份额共同生成标准签名（例如 ECDSA），而无需在任何单一位置重构完整的秘密密钥 [169, 184]。

3.4.1 核心原则：消除单点故障

在传统的自托管模式中，单个私钥（或助记词）构成单点故障：一旦安全失陷，所有关联资产和授权都会立即面临风险。基于多方计算的托管通过确保在密钥生成、存储或签名期间，签名密钥绝不会以单个明文密钥的形式出现，从而降低这种风险。取而代之的是，密钥以份额的形式分散存储在独立参与方（设备、服务或安全模块）中，并通过交互式协议完成签名；该协议在保护密钥份额隐私的同时，输出标准签名 [184, 169]。

安全性收益是直接的：仅攻陷代理主机（或仅攻陷用户设备）的攻击者，至多获得一份份额，无法单独生成签名。这会将对手所需的能力从单系统攻陷提升为多系统攻陷（或串通），而在现实的运行环境中，这要困难得多。

Tabl表 2：链上多重签名与基于多方计算的门限签名（多方计算阈值签名）的比较).

维度	链上多重签名	多方计算阈值签名
机制	智能合约强制要求多个 EOA签名。	参与方之间的交互式协议会生成一个标准签名。
链兼容性	取决于合约支持和账户类型。	若该链接受目标签名算法（例如 ECDSA），则与链无关。
链上足迹	多重签名结构和签名者在链上可见。	显示为标准的单签名交易链上。
费用和开销	由于合约执行和调用数据，费用更高。	标准交易费用；开销发生在链下通信。
策略演进	许多策略变更需要链上更新/-交易。	许多运营策略可以在链下修改（具体取决于托管设计）。

3.4.2 门限签名方案（TSS）：关键赋能技术

门限签名方案实现了一个 t-*of*-n 模型：

- n 个参与方各自持有一个密钥份额。
- t 是生成有效签名所需的参与方最低数量。

任何由至少 t 个参与方组成的子集都可以协作生成符合目标算法的有效签名（以太坊风格的账户通常使用 ECDSA）；而任何至多由 $t-1$ 个参与方组成的联盟，除了其各自份额所能揭示的信息之外，无法获知有关底层签名密钥的任何有用信息 [184]。现代构造使用经过精心设计的多轮协议、承诺和一致性检查，在不重构密钥的情况下共同推导签名随机数和签名组件。

多方计算阈值签名与链上多重签名。必须区分多方计算阈值签名与链上多重签名智能合约（例如 Safe{Wallet}）。在多重签名合约中，多个外部拥有账户会生成多个签名，并由合约在链上执行法定人数要求。在多方计算阈值签名中，则通过链下的密码学方式执行法定人数要求，从而生成一个标准签名；该签名在链上看来与传统的单签名者交易无法区分 [169]。

3.4.3 安全模型与信任假设

t-of-n 门限参数明确了信任假设：只要遭到安全失陷或串通的参与方少于 t 个，系统就能保持安全。在实际托管产品中，会通过组织与基础设施隔离（使用不同的机器、环境和访问控制），并在适当情况下采用经过加固的签名组件（例如基于硬件安全模块的服务）来强化这种独立性 [96, 274]。在智能体部署中，参与方可能包括：

• 用户控制的设备，
• 代理执行主机，
• 独立的策略或审批服务，和/或
• 专用安全模块（例如由 HSM 支持的签名器）。

这种隔离很有价值，因为它能将许多常见故障（代理主机遭入侵、云凭证泄露、提示注入导致恶意交易构造）转化为策略和法定人数问题，而非直接引发资产损失事件。

3.4.4 运营权衡：延迟、可用性和拒绝服务

多方计算阈值签名并非没有代价。由于签名过程需要交互，因此会带来**延迟**（需要进行多轮通信）和**可用性耦合**（签名时，必须有足够数量的参与方在线并及时响应）。在对抗环境中，这些特性可能成为攻击目标：即使攻击者无法窃取资金，仍可能试图通过阻止法定人数形成（例如，针对某个参与方发起网络中断）来降低服务质量，从而实质上导致钱包无法执行时间敏感型操作，形成拒绝服务。因此，实际部署会通过谨慎选择阈值和冗余（例如，t=2 个中的 n=3 个，而非 3 个中的 3 个），并强化通信信道、监控法定人数故障 [169]，在安全性与活性之间取得平衡。

另一项工程考量是**策略部署位置**。如果将策略服务设为必需的签名者，它就会成为可用性的关键依赖；如果它是可选的，策略就只能起到建议作用，而无法强制执行。智能体设计者应将这些选择视为威胁模型的一部分：即使发生现实情况下可能出现的部分故障，“安全”的配置也必须仍然可用。

3.4.5 对代理系统的影响

多方计算阈值签名为部署受共同控制的链上代理提供了实用基础 l:

• **代理发起、用户授权的执行**。 代理可以构造交易（或在账户抽象环境中构造用户操作），并提供自己的签名份额；但要执行交易，还需要其他份额，例如来自手机或硬件钱包的用户批准份额。这样一来，无需妨碍代理的自动化优势，即可通过密码学机制实现人在回路控制。

• **策略强制的否决权**。 可以将策略服务纳入签名法定人数，使其成为必需参与方。如果交易违反约束条件（如支出限额、允许列表或与高风险智能合约的交互），策略参与方就会拒绝参与，从而阻止签名生成，进而阻止交易执行。

• **失陷遏制与响应时间**。 如果代理主机遭到安全失陷，攻击者可能只会获得一个份额，不足以完成签名。这为密钥轮换、份额撤销或重新配置法定人数争取了时间，无需立即将资产迁移到全新的账户。

总而言之，多方计算和门限签名提供了一个与链无关且保护隐私的托管层，可补充链上可编程性。结合代理式执行，它们能够构建这样的架构：智能体在具备实质性自主性的同时，仍通过密码学机制受用户同意和可执行的策略约束所制约 [169, 184]。

3.4.6 智能合约安全与验证

自可编程区块链问世之初，智能合约的安全性就一直面临严峻挑战。包括 DAO 黑客攻击事件和众多去中心化金融攻击在内的重大事件表明，即使是经过审计的合约，也可能存在隐蔽漏洞，导致灾难性损失。智能合约形式化验证是确保其正确性和安全性的关键技术 [97]。人们已开发出多种用于检测智能合约漏洞的自动化工具，包括静态分析工具和模糊测试工具 [123, 154]。人们也探讨了智能合约的法律影响，重点关注如何将传统法律框架应用于这项新技术所面临的挑战 [128]。区块链技术在供应链管理中的应用也得到研究，重点关注其安全与透明度优势 [119]。近期的指南全面概述了智能合约安全状况，包括常见漏洞和安全开发的最佳实践 [41, 63]。人们也探讨了物联网（IoT）环境下智能合约的安全性，重点关注保护资源受限设备所面临的独特挑战 [65]。对于代理中介交易而言，这一安全状况尤为重要，因为代理不仅必须构造有效交易，还必须推理其交互合约的安全性。

形式化验证技术已成为智能合约安全领域一种颇具前景的方法，能够对正确性属性进行数学证明。Certora、Echidna 和 Slither 等工具提供了不同程度的保障，从轻量级静态分析到完整的形式化验证不等。然而，这些工具的实际应用仍受到现实世界合约交互复杂性的限制，尤其是在可组合的去中心化金融协议中，多个合约的交互会产生涌现行为。因此，与智能合约交互的代理系统必须具备运行时监控和模拟能力，以便在提交不可逆交易之前检测潜在的危险交互。

3.4.7 预言机设计模式与安全考量

近期综述指出了区块链预言机领域的关键开放研究方向，包括需要更稳健的安全模型、更完善的经济激励，以及更加去中心化的架构 [99]。
预言机充当区块链智能合约与外部数据源之间的桥梁，使合约能够对现实世界事件和状况作出响应。对于与依赖预言机的协议交互的代理系统而言，理解预言机设计模式及其安全影响至关重要。

价格预言机架构 向去中心化金融协议提供资产价格的价格预言机采用多种架构，各自具有不同的安全性和活性特征。时间加权平均价格（TWAP）预言机会计算指定时间窗口内的平均价格

，能够抵御短期操纵，但代价是反映当前价格时存在延迟。中位数预言机会汇总多个来源的价格并报告中位数值，从而抵御异常值和单一来源操纵。

推送式预言机会主动向链上合约提交价格更新，使协议无需额外交易即可获取当前价格。拉取式预言机则要求协议或用户请求更新价格，虽然能降低链上成本，但会引入延迟并可能影响可用性。混合方式结合推送和拉取机制，以平衡成本、延迟和可用性。

预言机安全威胁 预言机操纵对去中心化金融协议及与之交互的智能体构成重大威胁。闪电贷攻击可能暂时扭曲去中心化交易所的价格，导致时间加权平均价格预言机报告被操纵的价值。多区块攻击可在更长时间内操纵价格，影响采用较长平均窗口的预言机。经济攻击则可利用预言机报告价格与真实市场价格之间的差距，从协议中攫取价值。
代理与依赖预言机的协议交互时，必须将预言机安全性纳入考量。这包括了解各协议所采用的预言机架构、监控预言机操纵迹象，以及实施保障措施，避免依据被操纵的数据采取行动。对于高风险操作，代理可以交叉核对多个预言机来源，并对报告的数值进行合理性检查。

新兴预言机模式 一些新兴模式正在解决传统预言机设计的局限性。乐观预言机允许任何人提交数据，并设置争议期，在此期间可以对不正确的提交提出质疑。这种模式通过经济激励维持安全性，同时降低运营成本和延迟。零知识预言机利用密码学证明验证数据的真实性，而无需披露底层来源，从而实现保护隐私的数据馈送。
面向代理的预言机模式也正在兴起，旨在提供代理进行决策所需的数据类型。其中包括汇总社交媒体和新闻数据的情绪预言机、提供协议级风险评估的风险预言机，以及为特定交易策略提供信号的策略预言机。随着代理的采用不断增长，预言机生态系统可能会不断演进，以更好地满足代理的特定需求。

3.5 去中心化金融原语：代理的试验场

要理解链上代理能够安全执行哪些操作，我们必须准确把握其运行所处的市场微观结构 。去中心化金融（DeFi）并非单一协议，而是由金融原语构成的可组合技术栈，其风险面、延迟敏感性和对抗性激励各不相同。在本综述中，我们将 DeFi 视为代理的主要操作领域，因为大多数 Agent-to-Chain 工作负载最终都可归结为：（i）交易；（ii）抵押信贷；以及（iii）以稳定币计价的会计核算。因此，我们重点关注三种基础原语：自动做市商（AMMs）、借贷协议和稳定币。去中心化金融技术已在近期的多篇综述中得到系统梳理 [28, 324]。自动做市商形式理论也已得到深入研究 [37, 38]。DeFi 协议的安全性同样一直是研究的重要领域 [256]。

3.5.1 自动做市商（AMM）：去中心化交易所的引擎

概念。 自动做市商通过根据池储备以算法方式为资产定价，在无需传统订单簿的情况下实现去中心化交易。这种设计带来了连续流动性和可组合性，但也使执行质量与池深度和交易排序紧密相关。Uniswap 风格常函数做市商已得到广泛研究。早期研究分析了常函数做市商的形式属性，包括其定价行为和无常损失特征 [23]。Uniswap 协议本身也经过了深入分析，研究内容包括其市场行为和经济影响 [11]。近期研究则着眼于建立更通用的自动做市商理论，其中包括集中流动性设计 [38, 174]。近期的多篇综述也对更广泛的 DeFi 生态系统（包括自动做市商）进行了系统梳理 [324]。尽管常数乘积自动做市商（CPMM）是典型代表，现代自动做市商已扩展到更丰富的不变量（包括集中流动性设计），因此引入了智能体必须正确建模的额外状态和参数化 [11, 324]。

对智能体的影响。

- **行动空间**。 代理的核心操作是兑换、添加流动性和移除流动性，通常通过路由器合约执行。即使是“简单”的兑换，也属于多约束优化：代理必须选择路由、设置截止时间和最小输出金额，并考虑代币小数位数、授权以及潜在的 Permit 流程 [324]。

- **执行风险（滑点和排序）**。 自动做市商的执行对交易排序和瞬态状态高度敏感。在公共内存池中，价格影响与可观测性的结合会带来可预测的提取机会（例如，执行价格遭受夹击式恶化）。因此，代理应将最小输出金额和路径选择视为安全参数，而不只是用户体验参数 [146, 174]。
- **流动性提供风险（无常损失）**。如果代理提供流动性，那么当相对价格变动时，它会面临无常损失；同时也会受到手续费收益动态的影响，而后者取决于交易量、波动率和资金池设计。集中流动性会放大机遇与风险，因为它会使流动性提供者头寸具有路径依赖性，并对区间选择十分敏感 [11, 38]。

3.5.2 借贷协议：去中心化货币市场

概念。借贷协议构建超额抵押信贷市场：贷方将资产存入资金池，借款人则以抵押品为担保借款，并受风险参数（贷款价值比、清算阈值和协议定义的激励）约束。专门的综述分析了借贷池的安全性与经济动态，这一点十分重要，因为这些系统将智能合约正确性、预言机完整性和清算市场行为紧密关联起来 [37, 180]。

对智能体的启示。

- **行动空间**。智能体可以存入/提取、借入/偿还、调整抵押品构成，并在某些协议中管理利率模式或杠杆循环。从操作层面看，这些操作通常要求按正确顺序完成授权和协议调用，因此模拟与预览步骤尤为有价值。
- **风险管理**。 借入会引入持续的偿付能力约束（类似健康因子的监控）。代理必须跟踪抵押品估值（预言机数据馈送）、利息累积和清算阈值，并在延迟和费用约束下规划缓解措施（追加抵押品、部分还款、去杠杆） [37, 180]。
- **作为对抗性子博弈的清算**。 清算是一个竞争激烈且时间敏感的市场，具有类似最大可提取价值的动态：许多参与者会观察到相同的抵押不足头寸，并竞相获取清算激励。作为清算人行事的代理，必须将执行可靠性、Gas 动态和排序风险纳入策略，而不能将清算视为静态的应用程序接口调用 [146, 204]。

3.5.3 稳定币：计价单位与隐藏的依赖

概念。 稳定币为大多数去中心化金融活动提供计价单位，也是财资管理、风险控制和绩效报告的默认计价方式。对于智能体系统而言，稳定币不只是一种资产类别；它还是目标函数、预算约束和安全策略中的隐含假设（例如，“每天亏损不得超过 $X”）。对去中心化金融的全面概述强调，稳定币是流动性、抵押品和结算的基础，同时也会带来颁发者、抵押品及机制特定风险，这些风险会在压力期间扩散至整个系统 [28, 256]。

对智能体的启示。

- **政策感知型计价**。 智能体应以稳定币计价的条款（限额、敞口上限和损失阈值）来编码策略，同时还要明确规定哪些稳定币可用于会计核算和结算（因为“稳定”并非保证）。
- **机制风险**。 不同设计意味着不同的故障模式（托管型／法币支持型、加密资产抵押型或算法型）。智能体应将稳定币选择视为风险态势的一部分：正常情况下可接受的工具，在压力情境或出现脱锚信号时可能不再允许使用。
- **可组合性风险**。 稳定币敞口往往是间接的（以抵押品、流动性提供者库存或协议金库资产的形式存在）。因此，智能体需要具备投资组合级可见性，而不能只进行逐笔交易检查，才能避免在表面上相互独立的头寸中集中累积脱锚风险。

这对智能体设计为何重要。这些基础构件定义了智能体到链系统的实际行动空间，同时也界定了威胁面。自动做市商使智能体面临交易排序和价格影响提取的风险；借贷协议使智能体面临预言机和清算动态带来的风险；稳定币则为会计核算提供锚定，同时也引入了系统性假设。因此，后续章节将围绕这些基础构件梳理接口标准和安全控制，强调模拟、策略门控和 MEV 感知型执行是基础要求，而非可有可无的改进 [146, 180, 204]。

3.5.4 市场动态与代理行为

自主代理进入 DeFi 市场，可能从根本上改变市场动态 [228]。代理中介交易可以通过实现更快速的套利、更灵敏的流动性提供以及更精细的风险管理，提高市场效率。然而，这也会带来新的风险，包括代理行为趋同可能加剧市场波动性，以及代理间 MEV 提取的出现。

代理运营者面临的经济激励未必总是与更广泛的市场健康状况一致。以短期利润提取为优化目标的智能体，可能会采取某些行为：这些行为单独来看是理性的，但从整体上却会损害市场质量。例如，激进地提取最大可提取价值会增加其他用户的交易成本；流动性提供策略会在波动率较高时撤出；交易模式则会利用信息不对称，让不够老练的参与者蒙受损失。
要理解这些动态，需要新的分析框架，将共享市场环境中多个自主代理之间的策略性交互纳入考量。博弈论分析、基于代理的建模，以及对生产环境中代理行为的实证研究，都有助于加深这种理解。这些研究成果应为代理系统的设计以及其所处市场的治理提供依据。

3.5.5 高级DeFi机制与风险面

2020年DeFi危机凸显了高度互联的去中心化金融协议中固有的系统性风险，包括清算级联和预言机故障 [133]。在这些环境中运行的智能体必须经过设计，能够抵御此类黑天鹅事件。

除自动做市商和借贷协议等基础原语之外，DeFi 生态系统已经发展出越来越复杂的金融工具。收益聚合器会自动优化跨多个协议的收益，其设计与安全性已在近期研究中实现系统化 [77]。收益聚合器的安全性也一直是重要的研究领域，重点在于识别并缓解常见漏洞 [252]。不同 DeFi 协议之间的资金流动也已得到研究，重点是理解该生态系统的系统性风险 [243]。零知识证明这一基础密码学原语也已应用于 DeFi，以实现隐私保护交易 [125]。多链DeFi协议的安全性也已得到探索，重点关注保障跨链通信所面临的独特挑战 [335]。这些高级机制既为代理驱动的执行创造了机遇，也带来了风险。

闪电贷代表了代理系统的一项尤为重要的创新。通过在单笔原子交易中实现无抵押借入，闪电贷使智能体能够执行复杂的套利策略、清算和资本高效操作，而这些操作原本需要大量前期资本。然而，闪电贷也会带来新型攻击向量，众多漏洞利用事件已经证明，攻击者可以使用闪电贷借入的资金操纵预言机价格或耗尽协议储备金。因此，代理系统的设计必须充分了解闪电贷的运行机制，既要将其视为合法操作的工具，也要将其视为对手可能采用的威胁向量。

3.6 二层扩展：代理的高速执行场所

以太坊等第一层区块链提供了稳健的结算层，但其吞吐量有限，且交易费用波动较大、通常较高，因此不适合许多需要频繁、低延迟交互的代理工作负载（例如，迭代式投资组合再平衡、通过纠正操作进行持续风险监控，或清算感知型信贷管理）。这一缺口推动了**第二层（L2）扩容方案**的迅速普及。这类方案旨在链下执行交易，同时将安全性和数据可用性锚定到底层一层。对于智能体而言，L2 往往会成为默认执行场所，因为它们能降低每次操作的成本，并支持更高频率的操作，而不至于让燃料费成为目标函数中的主要成本项。Rollup 尤其代表了占主导地位的设计类别，扩容文献已对其进行了全面梳理 [297]。

3.6.1 核心理念：链下执行，链上结算

Rollup 背后的核心理念，是将高成本的执行 转移到二层环境，同时通过向一层智能合约提交交易批次（或其压缩表示），保留由一层强制执行的结算保证。二层运营者（通常称为排序器）会高速排序并执行交易，然后定期向一层提交批次承诺。一层合约会根据 Rollup 的安全模型验证批次，并更新对二层状态的链上承诺（通常是状态根）。这样一来，许多二层交易可以分摊链上验证和数据成本，相较于一层执行，吞吐量也会大幅提升 [297]。

对于智能体设计而言，这一区别很重要，因为**执行** 是在二层特定假设（排序、延迟、费用市场）下进行的，而**结算与最终性** 则受一层提交频率以及证明／验证机制的约束。忽视这一区别的智能体往往会错误处理确认语义、重组窗口和跨域资本约束。

3.6.2 乐观 Rollup：无罪推定，直至证明有罪

概念。 乐观 Rollup（例如 Arbitrum 和 Optimism）会将批次承诺提交到一层，并乐观地假定其正确，同时在**挑战期** 内通过证明无效状态转换并使用**欺诈证明** [297]将其拒绝，以此确保正确性。其核心执行机制是，任何人都可以通过提供证据，证明某笔交易（或某个计算步骤）执行有误，来质疑已提交的断言，从而触发链上争议处理流程，对该主张作出裁定。
对智能体的影响。

- **低成本、高频执行**。 交易费用通常低于一层，使更精细的控制循环成为可能（例如，更小但更频繁的再平衡或风险调整），而这类操作在一层上并不经济。
- **EVM 兼容性**。 许多乐观 Rollup 致力于实现 EVM 等效性或高度兼容性，从而减少工程上的阻力：相同的合约、软件开发工具包和交易构造逻辑通常只需稍作修改即可迁移。
- **提现延迟与资本规划**。 质疑窗口造成了**非对称最终性**：二层内部操作可能很快得到确认，但提取至一层通常需要等待争议期（通常为数天）。需要跨域流动性的智能体必须在规划和风险管理中明确考虑这一延迟（例如，避免采用默认假设一层可立即结算的策略） [297]。

3.6.3 零知识汇总：通过有效性证明建立信任

除了 Rollup，零知识证明也已用于支持链上私密函数执行，即函数逻辑本身对公众隐藏 [49]。这使更复杂的隐私保护型应用成为可能，智能体可与智能合约交互，而无需透露其内部状态或逻辑。

概念。 零知识汇总（例如 zkSync 风格系统和基于 STARK 的设计）通过为每个提交的批处理附加**有效性证明**来确保正确性。Rollup 无需等待质疑，而是使用简洁密码学证明（通常为 zk-SNARKs 或零知识可扩展透明知识论证）证明该批处理的状态转换正确。一层合约会验证该批处理的单个证明，并在验证通过后接受状态更新为有效 [297, 68]。这种模型无需漫长的欺诈证明窗口，并能实现更快、对一层而言更对称的结算，但仍受证明生成和提交频率的限制。
对智能体的影响。

- **更快的一层退出和更高的资本效率**。 由于批处理在有效性得到证明后即被接受，一层提现延迟通常取决于证明生成和一层纳入，而非持续数日的争议期。这可以显著提高频繁跨域再平衡的策略的资本效率 [297]。
- **复杂性与成熟度**。 零知识证明系统比乐观争议系统复杂得多，而实现零知识以太坊虚拟机等价性在技术上要求很高。因此，不同零知识汇总之间的实际差异（虚拟机语义、工具链成熟度、操作码覆盖率、性能范围）可能大于乐观生态系统中的差异，智能体不应假设它们可以完美移植。
- **成本结构**。 证明生成需要大量计算，可能会将成本从链上验证转移到链下证明基础设施。由此产生的费用动态因设计而异，并且可能随着证明系统和硬件的改进而随时间变化 [297, 68]。

3.6.4 第二层格局：碎片化、跨链桥与面向智能体的风险

原子跨链交换是不同区块链之间无需信任地交换资产的基础技术，通常使用哈希时间锁定合约（HTLC） [147]。权益证明侧链提供了一种创建可互操作区块链的机制，这些区块链拥有各自的安全性和治理模型，并可通过跨链桥连接到主链 [120]。

对代理而言，L2 生态系统并非单一的执行场所，而是由一系列领域构成的碎片化集合，各自拥有不同的排序器、费用市场、确认语义和应用部署。因此，能力出色的代理需要：

- **在多 L2 世界中导航**。 了解相关应用和流动性所在的位置，并据此选择执行场所，而不要想当然地认为一层是规范的执行场所。
- **对特定域最终性进行建模**。 将确认和结算视为依赖域的事项（一层最终性与二层本地确认），对于依赖跨域状态一致性的策略尤其如此。
- **将桥接风险视为首要威胁**。 在一层和 L2 之间（以及不同 L2 之间）转移资产通常依赖跨链桥和跨链消息系统，而这些系统屡次成为风险集中的领域和高价值攻击面。自动执行跨链桥接的代理必须将跨链桥安全态势、监控和故障处理纳入策略和执行逻辑，而不是将跨链桥接视为常规转账 [46, 30]。

总之，L2 是链上智能体系统的重要推动因素，因为它们提供了频繁交互所需的吞吐量和成本特性。与此同时，它们也带来了新的操作语义和跨域风险。因此，安全的智能体架构应将 L2 选择、确认语义和跨链桥交互视为代理操作流水线中的明确组成部分，并根据所采用的 Rollup 安全模型调整策略门控和验证 [297]。

3.6.5 智能体系统的二层选择标准

二层扩容方案的激增，为代理系统带来了复杂的决策格局。不同的二层在多个维度上各有权衡，最佳选择取决于每个用例的具体需求。

安全模型考量 二层的安全模型决定了安全操作所需的信任假设。乐观 Rollup 在挑战期（通常为 7 天）后继承以太坊的安全性，在此期间可以对欺诈性状态转换提出质疑。零知识汇总通过有效性证明提供即时的密码学最终性，但证明生成的复杂性也带来了不同的信任考量，涉及证明者实现正确性。

对于代理系统，安全模型的影响包括提款时机（代理可能需要为从乐观 Rollup 提取资金所需的较长提款期做好规划）、最终性假设（代理必须了解在不同 L2 中，交易何时可视为最终）以及故障模式处理（代理必须制定应对 L2 故障或争议的策略）。

性能特征 L2 在吞吐量、延迟和成本维度上的表现差异显著。高吞吐量的 L2 每秒可处理数千笔交易，使得在一层上成本高昂、难以实施的策略得以实现。低延迟的 L2 可缩短确认时间，使代理行为响应更迅速。低成本的 L2 降低了实现盈利操作的经济门槛，拓宽了可行的代理策略范围。

代理系统应根据策略需求选择二层。高频交易策略受益于低延迟、高吞吐量的二层。对成本敏感的策略则受益于交易费用最低的二层。需要与一层流动性交互的策略必须将跨链桥接成本和延迟纳入考量。

生态系统与流动性考量 除了技术特性，选择二层还必须考虑生态系统因素，包括可用流动性、协议部署和开发者工具。技术性能更优但流动性有限的二层，可能不适合需要深度市场的策略。同样，关键协议（去中心化交易所、借贷平台等）的可用性，也会限制可在各二层上执行的策略。

在多个二层上运行的代理系统必须应对异构环境带来的复杂性。这包括分别跟踪每个二层的状态、处理不同的远程过程调用接口和交易格式，以及为跨越多个网络的策略管理跨二层桥接。

4 代理—区块链集成模式分类体系（2025年）

随着智能体—区块链互操作性领域日趋成熟，一系列截然不同的架构模式逐渐涌现。这些模式并非只是技术上的新奇事物；它们体现了信任、自主性和风险方面的根本性演进。对于任何希望在 Web3 生态系统中构建智能体系统或与之交互的开发者、研究人员或用户而言，理解这些模式至关重要。本节将深入探讨我们提出的五部分分类体系，并以此作为

表3：概　述按代理权限和风险逐步增加的代理—区块链集成模式　　k.

模式	代理权限	主要接口平面	主要风险	演进压力
I. 只读分析	无（信息性）	数据平面（远程过程调用、索引器、分析-ics）	数据欺骗、过时或存在偏见的观点	对可操作见解的需求
II. 意图生成	仅提供建议	决策平面（模拟、意图模式）	不安全或误导性意图的批-准	部分自动化的需求
III. 委托执行	限定范围 / 基于角色	策略平面（智能账户、会话密钥）	策略绕过或权限提升	安全的有限自主性
IV. 自主签名	完全（受策略约束）	托管平面（多方计算、可信执行环境、签名应用程序接口）	密钥或代理失陷	高风险自动化
V. 多代理工作流	分布式／基于法定人数	协调平面（共享策略，工作流）	串通或治理俘获	最小化信任控制

分析定义各个阶段的安全考量、架构权衡和演进压力的组织框架。该分类体系如图3所示，并在表3中进行了总结，清晰描绘了从被动的只读代理到具备直接、高风险签署能力的完全自主多智能体系统的演进过程。

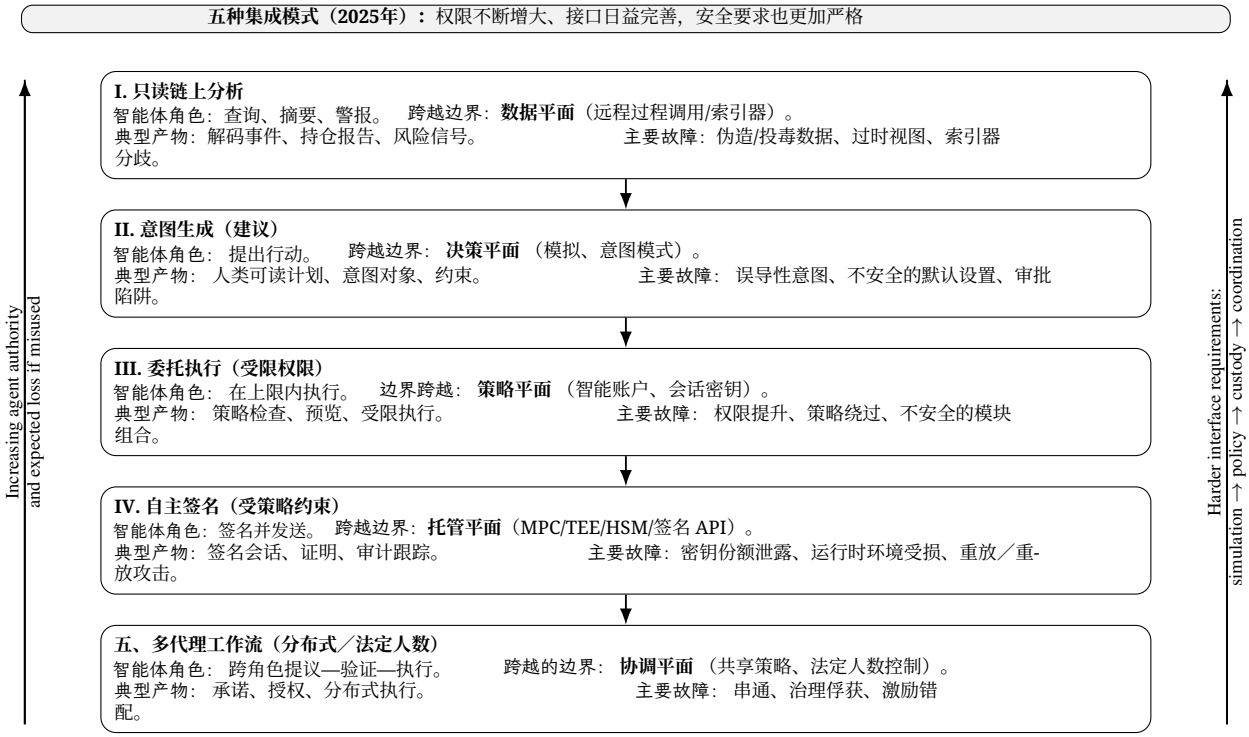

图3：五种代理—区块链集成模式，展现了沿接口平面（数据、决策、策略、托管、协调）逐步演进的过程。该图是一个概览模型；后续小节将详细介绍各模式的机制、威胁面及代表性系统。

综合来看，图3和表3从结构上概述了代理与区块链集成如何随着权限从观察转向执行和协调而演变。本节其余部分将依次考察每种模式，并在专门的小节中介绍其核心架构、典型的系统实现以及主要威胁面。各小节不会重复上述摘要，而是着重探讨在实践中实现该模式的机制、该权限级别下出现的具体故障模式，以及为缓解这些问题所需的接口和策略控制。这种组织方式使分类体系既可作为概念地图，也可作为实用指南，帮助系统设计者应对不同的信任和自主性机制。

4.1 模式 I：只读链上分析智能体

描述。 模式 I 是基础集成模式：代理是观察者 和分析者，而非执行器。它作为异构链上数据的自然语言控制平面，将用户的问题（例如“这个协议安全吗？”或“过去 24 小时发生了哪些变化？”）转换为结构化查询，再将结果综合为

可审计解释。重要的是，代理无权 提交交易或更改状态；其输出仅供参考。这种模式与更广泛的趋势相符，即转向使用工具的大语言模型系统，其核心能力是在外部系统中可靠地检索和综合信息，而不仅仅是生成内容 [336, 322]。

架构深入解析。典型的模式 I 系统分为三个层：

- **数据平面（读取）**。代理通过远程过程调用端点查询区块链状态（余额、存储槽、日志），并且通常依赖已索引或预聚合服务，以执行历史查询并获取更高层级的度量指标（例如协议活动、合约交互和实体级启发式方法）。由于这些来源呈现的视图各不相同（规范链状态与索引解释），稳健的设计会将它们视为互补信息，而非可以互换的信息。
- **解释平面（结构化）**。检索到的信号会被规范化为精简的世界模型（资产、头寸、协议合约、时间戳、风险标记）。这是大语言模型最能发挥作用、但也最容易出错的环节：检索文本中的歧义、不同提供方之间不一致的模式，以及不可信内容与系统指令的混杂，都可能导致代理的结论失真 [131]。
- **解释平面（报告）**。代理会生成便于人类理解的回答，并注明来源信息：查询了哪些来源、观察到哪些差异，以及哪些假设仍不确定。这不仅仅是用户体验上的润色；在输入可能遭到操纵的环境中，这是帮助用户校准信任的主要手段。

安全模型与风险。 尽管模式 I 无法通过签署交易直接造成资金损失，但仍可能通过操纵决策引发损失。主要威胁类别是**对输入的完整性攻击**，包括：(i) 遭到入侵或恶意的 RPC/索引服务提供商；(ii) 预言机或价格馈送操纵，导致风险指标失真；以及 (iii) 通过不可信检索内容（网页、文档、治理帖子，甚至交易元数据）实施的间接提示注入，使智能体将这些内容视为指令而非证据 [131]。因此，实际的安全原则是**通过来源多样性发现差异**：针对关键事实查询多个相互独立的来源，将策略/指令与检索内容分离，并明确标记冲突，而不是将冲突平均消解。这种“通过构造进行验证”的思维方式，与更广泛的跨领域安全论点一致：互操作性技术栈会放大薄弱信任边界所带来的后果 [141]。

演进压力。 模式 I 智能体很快会遇到可用性上限：在完成分析并提出操作建议后，用户仍必须手动执行整个流程（复制地址、设置滑点、选择交易场所并安全签名）。洞察 与行动 之间的这一差距，自然形成了向模式 II 演进的压力；在模式 II 中，智能体不再止步于叙述性建议，而是开始输出结构化的候选意图 或交易草案，供用户批准 [101]。

4.2 模式 II：模拟与意图生成 tion

描述。 在模式 II 中，代理从被动分析者转变为面向行动的副驾驶。它不再只是描述当前世界状态，而是提出如何改变这种状态。给定一个高层次的用户目标（例如，“在低风险的前提下，使 10000 USDC 的收益最大化”），代理会构造一个拟议的**交易意图**：一种结构化、可供机器读取的规约，用于说明期望的结果（要做什么）以及约束条件（滑点界限、交易场所、代币白名单、截止时间），但尚未绑定原始的、特定于链的交易字节（如何实现）。关键的保障措施是，代理不会执行操作。相反，它会生成 (i) 一个意图对象，(ii) 清晰的人类可读预览，以及 (iii) 在状态分叉环境中模拟得出的结果，然后将决策权交给人类签名者。这种模式将使用工具的大语言模型转化为候选行动的规划器和编译器，同时保留用户托管权和最终授权 [336, 322]。

架构深入解析。 模式 II 系统通常会在模式 I 的基础上增加两种能力：(a)通过模拟实现反事实执行，以及 (b)通过意图模式实现结构化操作表示。

- **模拟与预览层（反事实执行）**。 代理依赖交易模拟（通常基于分叉状态），在用户签名前执行预检。高保真模拟会返回丰富的跟踪信息：状态变化量（余额、授权、头寸）、发出的事件、Gas 用量、回滚原因和调用图。这不仅是调试便利；它还是缩小用户意图与实际链上影响之间差距的主要控制手段。设计完善的预览应呈现用户能够理解的语义后果（“向合约 X 授予额度”、“兑换 USDC→以太币”、“开立借入头寸”），而不仅仅是底层调用数据。

- **意图表示（从叙述性目标到受约束的操作）**。代理输出一个结构化意图对象，使其在不同前端之间保持稳定，并且最好能在不同钱包和执行后端之间互操作。约束条件应体现在这一意图层中，包括已批准资产列表、最大滑点、目标交易场所、风险

态势，以及明确的禁止规则（例如“不得进行无限授权”）。以意图为中心的架构被视为实现更安全组合的一种途径，因为它们使目标能够独立于编译后的交易序列进行审计 [202]。

推理流水线。 模式 II 中的推理本质上是反事实推理：代理必须探索可能操作的空间，并在约束条件下比较执行结果。例如，收益寻求工作流通常会遵循以下循环：

1. **列举候选项**。 根据用户约束（风险等级、区块链、交易场所允许列表），确定候选交易场所和工具（兑换为收益型资产、存入借贷市场、质押路径）。 2. **获取报价并规划**。 获取报价并构造候选操作序列，适当时包括通过聚合器进行路由。 3. **模拟**。 在分叉环境上执行每个候选规划，以获得具体的执行结果（收到的数量、授予的授权、Gas 成本和故障模式）。 4. **评分**。 计算效用分数，将预期收益、费用、滑点和安全性约束（例如授权范围、合约信誉信号）纳入考量。 5. **呈现选项**。 将排名靠前的候选项作为不同的意图提供，并排展示预览和明确的利弊，供用户批准或拒绝。

安全模型与风险。 模式 II 通过让用户保留签名权来降低直接的密钥风险，但也引入了一类更隐蔽的故障：**恶意或不安全的意图生成**。攻击者可以通过以下方式，试图诱导代理生成有害意图：(i) 对抗提示、(ii) 投毒的检索内容，或 (iii) 误导性元数据（代币名称冲突、钓鱼合约），利用代理倾向于将看似合理的标识符视为可信信息这一特点。间接提示注入尤其值得关注：如果未将不可信内容与策略及系统指令清晰分离，代理就可能被诱导生成违背用户预期、却表面上仍然前后一致的意图 [131]。
因此，该模式中的防御措施主要依赖于语义验证，而非密码学隔离：

1. **高保真模拟与以用户为中心的预览**。 预览必须以难以忽视的方式呈现危险的状态变更（尤其是授权、类似 Permit 的授权以及意外的外部调用）。在去中心化金融威胁分析中，预览与模拟一再被强调为必要的控制措施，因为它们能够揭示复杂调用图的实际影响，而这些影响对终端用户而言通常是不透明的 [146, 149]。 2. **将用户理解能力作为控制措施（及其局限性）**。 用户是最后一道关卡：他们必须解读预览、验证目标合约与资产，并拒绝可疑提案。这会带来认知负担和容易出错的人为因素，尤其是在预览内容冗长、技术性强，或波动性市场造成时间压力的情况下。因此，模式 II 的强弱完全取决于预览的清晰度以及用户评估预览的能力。

演进压力。 逐笔交易人工审批的要求会成为吞吐量瓶颈。它限制了策略复杂度，也让重复性、低风险的操作变得繁琐，促使设计者转向有限自主权：代理无需持续接受人工干预，即可执行预先批准的操作类别。对安全、有限自动化的需求推动了向模式 III 的转变：权限在明确范围、上限和策略约束下委托，而非每笔交易都重新审批。

4.3 模式 III：委托执行（受限角色）

描述。 模式 III 是分类体系中代理首次从提案迈向执行的阶段，但仍严格遵循最小权限原则。用户并非将主密钥交给代理，而是授予范围有限且可撤销的权限，严格限定其范围（可以执行哪些操作）、额度（可以花费多少）和时限（权限有效期）。在实践中，这种委托通过智能账户设计实现，其中授权可编程，并可设定为以运行时检查为条件。在以太坊和 EVM 兼容链上，账户抽象架构，尤其是 ERC-4337，通过标准化智能账户执行路径而无需协议硬分叉，有力地推动了这一模式的发展 [55, 308, 280]。

架构深入解析。 模式 III 引入了模式 I–II 中并不常见的两项架构要素：(i) 一种委托签名能力（通常是会话密钥）；(ii) 一个策略执行点，用于确定性地控制执行。

- **接口层（智能账户和模块）**。代理与用户的智能账户交互，而非与人工批准者交互。智能账户提供验证钩子，可通过模块（守卫、验证器、插件）进行扩展，由这些模块负责在执行前强制实施约束条件。在 ERC-4337 下，代理

通常会提交一项面向智能账户的用户操作；账户的验证逻辑决定该操作是否已获授权 [55, 280]。

- **委托原语（会话密钥／角色）**。委托通常由智能账户认可的会话密钥或角色凭证表示。会话密钥颇具吸引力，因为它们允许授予范围狭窄的能力，且无需将资金迁移到新地址即可撤销。在许多实现中，会话密钥仅在模块检查的条件下有效（时间界限、支出上限、目标地址允许列表），因此密钥持有者获得的是有条件的权限，而非不受限制的控制权。
- **策略模块（确定性门控）**。策略模块充当安全护栏。常见的约束包括：
  – **函数允许列表：**限制可调用的选择器（例如，仅允许兑换或收获函数）。– **合约允许列表：**限制目标地址（例如，特定路由器、金库或借贷池）。– **支出限额：**设置单笔交易上限和／或限速上限（例如，每天 $\leq$100 USDC）。– **时间界限：**为委托凭证设置有效期。– **频率限制：**限制操作频率（例如，每 24 小时最多收获一次）。

  当模块评估语义（资产转移、授予的授权、净价值变化），而不仅仅是目标地址等表面属性时，这些控制措施最为有效。

执行流程（ERC-4337 实现）。ERC-4337 下的一种典型流程如下：

1. 代理构造一项用户操作，其中编码了智能账户发起的预期调用。 2. 代理使用其委托凭证（会话密钥或角色密钥）对用户操作进行签名。 3. 捆绑器将用户操作纳入捆绑交易，并将其提交至全局入口点合约 [55, 280]。 4. 入口点调用账户的验证逻辑（通常为 validateUserOp），该逻辑会委托给已安装的策略模块。 5. 如果策略检查通过，则执行该操作；否则，该操作会在验证时被确定性地拒绝。

安全模型与风险。模式 III 的核心安全优势是**安全失陷时损失有界**。如果代理遭到安全失陷，损失将受链上代码所强制执行的委托范围和上限限制。与“代理应该遵守规则”相比，这是更强有力的保障，因为一旦约束条件嵌入账户逻辑，执行就不再依赖代理的运行时完整性。

主导风　　险转向 **策略定义及其实现的正确性与完整性**　　：

- **策略漏洞与非预期能力**。 宽松的允许列表（仅限制目标地址，而非函数级别）可能会允许对原本可信的合约调用意料之外的方法，包括未曾预料到的授权或升级路径。
- **模块漏洞与组合风险**。 守卫和验证器本身也是智能合约。程序错误、对调用数据结构的不安全假设，或模块间脆弱的组合，都可能造成绕过。形式化与系统性分析一再表明，智能合约安全既关乎规约和行为建模，也关乎代码模式 [10, 20]。
- **运行时攻击面**。 即便有链上强制执行，智能体在提交时仍会受到交易替换、中继操纵及对抗性执行动态（包括最大可提取价值效应）的影响；因此，应尽可能将策略门控与模拟／预览配合使用，并采用保守的执行参数。

演进压力。 委托执行非常适合定期、结构化的自动化（例如定期兑换、定期支付和常规策略维护），并且天然契合基于账户抽象的自动化设计 [40, 308]。其主要局限在于缺乏灵活性：链上策略有意采用确定性设计，随着策略、交易场所和市场状况的变化，调整这些策略可能成本高昂或耗时较长。这促使系统转向模式 IV，通过更多依赖强化托管基础设施和链下控制来实现更高的自主性和更快的决策，同时仍维持强有力的安全保障。

4.4 模式 IV：自主签名与执行

描述。 模式 IV 是针对单个 代理最强大、因此风险最高的集成模式。代理被赋予构造、签名和提交交易的能力，无需针对每项操作实时获得人工审批。

这使高频且对时效要求严格的策略（例如财资管理、自动化做市或投资组合再平衡）得以实现，但也瓦解了模式 II–III 中存在的传统最后一道防线。因此，模式 IV 要求**机构级密钥托管** 和**纵深防御策略执行**，因为任何故障都可能以机器速度造成不可逆且可变现的损失。钱包安全综述一再强调，在现实系统中，交易授权是风险最集中的环节，而且“签名发生在哪里”往往比“用户界面显示什么”更重要 [149]。

架构深入解析。 稳健的模式 IV 架构将代理的推理过程与签名操作分离。代理绝不应 在其运行时记忆或文件系统中直接持有长期私钥。相反，它应通过经身份验证的 API 和明确的策略门控与强化签名基础设施交互 [292, 149]。实践中，主要有两种常见的托管架构：

1. **MPC/TSS 签名服务（分布式托管）**。 代理是门限签名方案（TSS）中的一方，持有一个密钥份额，而其他份额则由独立设备或安全服务持有。交易签名通过交互式协议执行，在不于任何单一位置重构私钥的情况下生成标准签名 [314, 184, 169]。这缩小了代理安全失陷的影响范围：仅窃取代理的份额不足以伪造签名。在实际部署中，通常还会叠加额外的保护措施，例如由 HSM 支持的密钥份额存储，以及将参与方部署在不同地理位置，以降低相关性失陷的风险 [274]。

2. **基于可信执行环境的签名器（硬件隔离托管）**。 在此架构中，签名密钥位于可信执行环境（TEE）内，即使周边宿主操作系统遭到安全失陷，该环境也旨在抵御密钥提取。代理向安全区提交候选交易或意图，只有通过本地策略检查后，安全区才会签名。TEE 被用于多种钱包和元交易设计，以实现隔离并建立基于证明的信任 [227, 168]。

策略引擎（缺失的控制平面）。在模式 IV 中，策略执行通常必须超越简单的允许列表和上限。原因不仅在于功能复杂性，还在于适应性：高风险自主执行必须能够应对市场状况、协议状态以及不断演变的威胁。因此，需要建立**多层策略引擎**，将以下要素结合起来：

- **静态约束：**地址和函数允许列表、支出限额、时间窗口、速率限制，以及明确的禁止规则（例如，不得进行无限额授权；不得调用跨链桥）。
- **基于模拟的防护机制：**在签名前进行分叉模拟和语义差异分析，以揭示授权、外部调用和净资产变化（这与 DeFi 安全分析一致，后者强调调用图复杂性以及预览副作用的实际重要性） [146, 149]。
- **行为监控：**根据代理的历史行为分布进行异常检测（例如，新颖目的地址、异常金额、异常授权模式、频率突然增加）。链上攻击识别领域已广泛研究监控与检测系统；它们也是自主执行中策略门控的天然补充 [30, 250]。

这就是将意图规约与授权证据分开的原因。按照我们的路线图，代理会生成一项交易意图（TIS），评估器依据更丰富的规则集对其进行评估，并生成一份策略决策记录（PDR）；签名服务（多方计算或可信执行环境）会在释放签名之前验证该 PDR。即使 PDR 位于链下，它也会为交易获授权的原因建立可审计的保管链，从而显著改善事件后取证和问责。

安全模型与风险。模式 IV 的设计应基于**预期会发生安全失陷**这一假设：任何单一组件（包括代理）都可能失效。因此，安全性依赖于独立控制措施，而这些措施同时遭到破坏的难度很高：

1. **相关故障：** 这是最危险的一类，即代理、策略引擎和签名基础设施同时遭到安全失陷（或共享同一个信任根、凭证或部署环境）。2. **策略漏洞和语义缺口：** 复杂策略可能存在错误或不完整，也可能通过边缘情况的调用数据、审批机制或合约升级绕过。这既是规范问题，也是实现问题。3. **智能体泛化错误和目标漂移：** 即使没有外部攻击者，自主代理也可能因错误假设、制度转变或误解约束条件而采取技术上有效、但在财务上有害的行动。因此，模式 IV 要求严格的目标规范、审慎的风险限额和稳健的验证循环。

运营要求（不可妥协）。鉴于在此权限级别下故障不可避免，生产环境中的模式 IV 部署 require：（i）持续监控与告警；（ii）快速撤销和密钥轮换程序；以及（iii）一个可立即使用的**紧急停止开关**，无需等待链上治理，也无需跨时区协调人员，即可停止签名或冻结委托权限 [292, 149]。

演进压力。即使采用强化托管和严格策略，单个自主代理仍会令人不安地集中了判断和操作权限。这促使人们将决策本身去中心化：由多个代理分担不同角色（提议者、风险检查者、执行器），要求法定人数审批，并引入集体治理控制机制，使其能够抵御单个代理遭到安全失陷或判断失误的影响。这一动因推动了向模式 V 的转变：采用具有分布式权限以及明确协调与监督机制的多代理工作流。

4.5 模式 V：多代理工作流

描述。模式 V 通过将信任、决策和执行权限分散到多个相互独立的代理，缓解单代理自主系统（模式 IV）中权限过度集中的问题。该模式不再假定任何单个代理都能做到足够安全或与目标充分一致，而是采用结构性方法：任何不可逆的链上操作都必须经过基于法定人数、涉及多个角色的决策过程，方可执行。从概念上说，这种模式既借鉴了组织治理（例如，金融机构中执行、风险和审计职能的分离），也借鉴了拜占庭容错和法定人数共识等经典分布式系统原则 [217, 339]。

这一模式最主要的实现形式是**提议者–验证者–执行者**工作流，其中战略意图、风险评估和执行被明确解耦，并分配给不同的智能体角色。

架构深入解析。

- **协调接口**。架构的核心是协调与治理合约，它充当共享状态机和裁决层。该合约记录提案、收集授权或拒绝意见、执行法定人数规则，并授权执行。智能体不依赖隐式信任，也不单靠链下消息传递；所有关键决策都会以透明且可审计的形式呈现在链上。

- **角色专业化**。权限被分解为不同的智能体角色，每个角色的职责都受到严格限制：1. **策略代理（提议者）**。这些代理监控市场、协议和外部信号，并生成候选行动。它们的输出不是原始交易，而是结构化的交易意图，用于描述目标、约束条件和预期结果。在去中心化自治组织或财库场景中，这些代理负责发现机会，而非执行操作。2. **风险代理（验证者）**。验证者代理会独立评估提议的意图。其评估可能包括基于分叉的模拟、经济压力测试、依赖项检查，以及根据明确的风险策略进行验证。验证者会在链上明确投票，决定授权还是拒绝提案。重要的是，这些代理的设计采取保守立场，并对提议者保持对抗性，发挥系统性“免疫反应”的作用，而非与其协作。3. **执行代理（执行者）**。执行者是经过刻意简化且受到严格约束的代理，其唯一职责是执行已达到法定人数的提案。它没有自由裁量权，也不具备战略逻辑。实际上，这一角色可以通过强化的自动化服务或去中心化执行网络来实现，其权限严格受治理状态控制。

安全模型与风险。 模式 V 的安全模型基于**明确的链上分布式信任**。任何单一智能体都无法单方面采取行动，而一个角色的安全失陷或不当行为也不会立即转化为损失。这显著提高了成功发起攻击的门槛，但也引入了一种新的主要故障模式：**智能体之间的串通**。

主要风险和缓解措施包括：

- **串通风险：** 一组达到法定人数的提议者和验证者可能合谋批准恶意操作。可通过代理多样性、运营者独立性和明确的法定人数阈值来缓解这一风险。
- **经济利益一致性：** 要求智能体质押抵押品，并在可证明的不当行为发生时予以削减，从而针对恶意协调建立直接的经济惩罚机制。
- **声誉与问责：** 维护提案历史、投票和执行结果的公开链上记录，可实现长期声誉追踪和外部审计。

- **透明度：** 所有提案、评估和执行过程均可见，使更广泛的社区能够进行人工监督和事后取证分析。

虽然这种模式显著降低了单点故障风险，却增加了系统复杂性和协调开销。因此，它最适用于价值高、影响大的系统，在这类系统中，安全性和合法性比延迟和简洁性更为重要。

演化压力与局限。 模式 V 代表了当前代理与区块链集成的实践前沿。它在深度自动化的诉求与对抗性环境、不可逆性和经济风险的现实之间取得平衡。它并不追求完美对齐，也不绝对信任某个单一代理，而是将怀疑直接融入架构之中。

未来的系统很可能会改进这种模式，而非取而代之：完善激励机制设计、降低协调延迟，并为代理独立性和决策质量提供更强的形式化保证。然而，在可预见的未来，最安全的自主链上行为并非由一个强大的代理独白，而是由众多专业代理展开有组织的对抗性对话。

过渡到威胁。 上述五种模式不仅对接口设计进行了分类，还逐步转移了主要的信任边界。模式 I 将风险集中在数据完整性上，模式 II 集中在预览保真度和用户解读上，模式 III 集中在委托权限的正确性上，模式 IV 集中在签名和策略基础设施的安全性上，模式 V 则集中在协调和法定人数治理上。这一演进促使我们采用具备流水线感知和接口感知能力的威胁模型：我们需要推理那些针对认知（通过提示和上下文操纵）、中间件（通过工具和 SDK 遭入侵）以及执行经济学（通过最大可提取价值和订单流利用）的攻击，而不只是传统的软件缺陷。因此，下一节将形式化定义所有模式中反复出现的受保护资产、对手类别和信任边界，并以此为基础，构建后续的攻击分类体系和缓解措施。

4.5.1 模式选择决策框架

选择适当的集成模式需要对多个因素进行系统性评估。本节提供了一个用于指导模式选择的决策框架。

风险评估 首先需要考虑的是预期应用的风险概况。风险较高的应用应采用更为保守的模式，并加强人工监督。风险因素包括风险敞口价值、操作的可逆性、结果的可预测性以及错误后果。

管理重大价值的应用通常应从模式 II（模拟与意图生成）或模式 III（委托执行）开始，而不是直接采用完全自主模式。“重大价值”的阈值取决于运营者的风险承受能力以及代理系统的成熟度。

运营要求 第二个考虑因素是运营要求，包括延迟、吞吐量和可用性。需要快速响应的应用可能不适合要求对每笔交易进行人工审批的模式。需要全天候运行的应用可能需要能够在无人工可用的情况下运行的自主模式。

运营要求与风险管理之间的权衡通常决定了适当的模式。高频交易应用可能需要采用模式 IV（自主签名），尽管其风险较高；而低频应用则可以承受人在回路模式所带来的延迟。

监管约束第三项考量是监管约束，这些约束可能要求采取特定控制措施或监督机制。某些司法管辖区可能要求对特定类型的交易或超过特定价值阈值的交易进行人工审批。应尽早识别合规要求，并将其纳入模式选择的考量。

组织能力第四项考量是组织能力，包括技术专长、运营成熟度和风险管理基础设施。自主程度更高的模式需要更完善的监控、事件响应和治理能力。组织应如实评估自身能力，并选择能够安全运行的模式。

演进路径 最后，模式选择应考虑预期的演进路径。通常，先采用保守模式，再随着经验积累逐步扩大自主性，比一开始就采用激进模式更安全。所选模式应允许能力随时间受控扩展。

4.5.2 集成模式的比较分析

本分类体系所识别的五种集成模式，代表了代理自主性和风险谱系上的不同位置。理解这些模式之间的权衡，对于为特定用例选择合适的架构至关重要。

模式 I（只读）以最低的实用性为代价，提供最高的安全性。此模式下的代理可以观察和分析区块链状态，但无法造成任何改变。此模式适用于分析、监控和咨询应用，其价值在于信息处理，而非交易执行。其安全属性很简单：由于代理无法为交易签名，因此无法通过区块链交互直接造成财务损害。

模式 II（模拟与意图生成）将代理能力扩展至交易构造，同时维持人工对执行过程的控制。代理可以分析情况、发现机会并准备交易，但最终授权权仍由人类掌握。此模式非常适合代理判断能够创造价值、但风险高到需要人工监督的应用。其安全模型依赖人工审查者在执行前发现代理错误或恶意建议。

模式 III（委托执行）赋予代理在预定义约束范围内的有限执行权限。此模式既能实现常规操作自动化，也能通过护栏机制防范灾难性错误。安全模型从人工审查每笔交易，转向由人类制定并监控策略。此模式适用于操作可预测且范围明确的应用；若每笔交易都需人工审批，相关开销将难以承受。

模式 IV（自主签名）赋予代理广泛的执行权限，仅施加最低限度的交易级约束。这种模式支持复杂策略，这些策略需要跨多笔交易快速协同执行。其安全模型高度依赖智能体设计、测试和监控，而非交易级控制措施。这种模式仅适用于完全自主带来的收益超过其额外风险的应用。

模式 V（多智能体协调）将自主运行扩展到涉及多个交互式智能体的场景。这种模式支持复杂工作流，这些工作流需要智能体之间进行协调、协商或集体决策。安全模型必须考虑智能体间动态，包括智能体交互可能引发的串通、竞争和涌现行为。

5 智能体—区块链系统的威胁模型

智能体人工智能与区块链的融合，带来了与传统 Web2 服务集成截然不同的威胁面。在智能体—区块链系统中，概率推理和工具编排与确定性且不可逆的状态转换相结合，且往往处于对抗性经济条件下。STRIDE 等标准检查清单仍有助于全面识别问题，但对实际部署中占主导地位的故障模式描述不足：这些故障模式包括引导代理决策过程的攻击、破坏工具介导感知的攻击，以及利用公共网络中的交易可见性、排序和执行动态的攻击。

近期关于工具调用型大语言模型应用的分析表明，即使开发者提示无害，不可信检索内容仍可通过检索介导的指令劫持来引导后续工具使用 [131]。因此，我们不仅将代理建模为软件，也将其视为一个决策组件，其输入、工具和执行通道都可能成为攻击目标。

5.1 资产、对手与信任边界

稳健的威胁模型首先要准确说明需要保护什么（资产）、谁可能发起攻击（对手），以及信任假设会在哪里发生变化（信任边界）。 [254] [39]

资产。智能体到链系统汇集了多种高价值资产：**(i) 资金与签名权限** （私钥、多方计算份额、会话密钥、智能账户验证逻辑）；**(ii) 权限与策略状态** （允许列表、支出限额、角色分配、支付主规则、中继器权限范围）；**(iii)智能体状态**（记忆、工具注册表、检索文档、执行计划、缓存的 ABI）；以及 **(iv) 遥测数据** （交易预览、模拟轨迹、策略决策、审计日志），用于检测和取证。在许多部署中，运行时还会持有**链下机密信息** （API 密钥、凭证、通知渠道）；这些信息一旦泄露，可能助长横向移动或社会工程攻击 [258]。

对手。我们考虑一系列能力和动机各不相同的对手：

- **经济对手（订单流和最大可提取价值行为者）：** 这些行为者会监控交易流，并通过重新排序、复制、尾随交易或实施三明治攻击来提取价值；基础性研究阐述了矿工/最大可提取价值及其对去中心化交易所执行的影响 [80]。
- **远程攻击者：** 利用代理运行时、插件、软件开发工具包、浏览器扩展、钱包连接器或 CI/CD 流水线中的漏洞，以获取签名能力或左右决策的行为者。
- **工具和数据源操纵者：** 污染或控制链下输入（网页内容、社交渠道、价格数据源、治理信息），或入侵远程过程调用、索引器和预言机基础设施，从而误导观察与推理阶段的行为者。
- **恶意或受入侵的协议：** 对抗性合约、钓鱼前端或可升级系统，即使交易在意图层面看似无害，也会触发非预期的状态转换。
- **内部人员和运营者：** 特权方（托管服务提供商、远程过程调用运营者、打包器/求解器运营者、DevOps）可能出于诚实但好奇、疏忽或恶意，其访问权限可能绕过既定控制措施。

信任边界。 在代理—区块链架构中，事件往往集中发生在数据或权限跨组件传递的边界处。最常见的边界包括：**(i) 用户界面与代理之间**（用户认为自己批准的内容与代理提出的内容）；**(ii)代理运行时与模型提供方之间**（提示、上下文和工具输出离开可信计算基）；**(iii) 代理与工具及端点之间**（远程过程调用、索引器、Web API 和去中心化应用前端）；**(iv) 代理与签名器/托管服务之间**（硬件钱包、多方计算/可信执行环境服务、智能账户验证模块）；以及 **(v) 提交至区块链的路径**（公共内存池、私有中继、打包器/求解器网络）。边界 **(iii)** 尤其脆弱，因为检索介导的指令劫持可能隐藏在代理需要获取并总结的内容中，随后传播至工具调用和交易构造 [131]。边界 **(v)** 是最大可提取价值提取集中的地方，因为公开传播有利可图的意图会招致对抗性重排序和价值攫取 [80]。

假设与范围。 我们的分析假设标准密码学原语未遭破解，且基础链的共识能够提供最终确定性。我们不假设应用、前端或链下服务值得信任，并将快速演进的智能合约系统视为智能体必须安全应对的对抗性环境的一部分 [132]。对于分布式决策（模式 V），我们将串通、贿赂、审查和治理俘获视为一阶威胁，因为即使各组件单独运行时均正确，这些因素仍可能攻破基于法定人数的控制措施。

5.2 攻击类别分类体系

智能体—区块链系统结合了（i）使用不可信语言和工具输出的概率决策器，以及（ii）错误不可逆且可被利用牟利的确定性执行底层架构。 [32]这种不对称耦合从根本上重塑了威胁面：攻击不再仅针对软件组件或网络基础设施，也会针对智能体的推理过程、策略解释和执行路径。

为使这一领域便于分析，我们将威胁划分为一组分层攻击类别，并使其与图4中的主要信任边界相对应。表4简要概述了这些类别，并将每一类映射到受影响的流水线阶段、遭受攻击的主要资产以及具有代表性的防御控制措施。这一分类体系有意以边界而非漏洞为导向：每一类都对应组件或参与者之间发生信任转移的一个独特点。

具体而言，我们区分以下类别：**(C1) 认知攻击**，通过操纵智能体推理和工具选择发动攻击（提示注入和指令劫持）；**(C2) 工具与数据平面攻击**，通过污染观测和模拟发动攻击（远程过程调用或索引器入侵、预言机操纵、恶意网页内容）；**(C3)策略与控制平面攻击**，绕过或破坏约束条件（配置错误的允许列表、策略模块漏洞、备用签名路径）；**(C4) 签名与托管攻击**，获取授权（密钥窃取、会话密钥权限过度、多方计算/可信执行环境安全失陷）；**(C5)提交路径攻击**，利用交易可见性和排序（内存池对手和最大可提取价值）；**(C6) 应用层攻击**，利用看似无害的交互触发有害的链上状态转换（恶意合约、升级陷阱、钓鱼前端）；以及 **(C7) 多智能体与治理攻击**，利用法定人数形成和协调机制（女巫攻击和串通）。

在本节余下部分中，我们始终采用这一分类，依次分析各类别，并针对攻击出现的边界提出缓解措施。

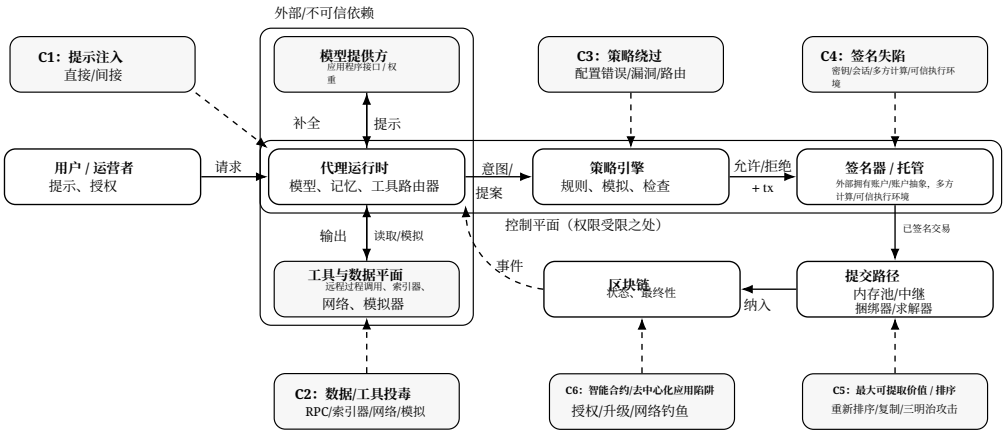

图4：智能体—区块链系统威胁模型。实线箭头表示名义上的数据/控制流；虚线箭头表示与主要信任边界相对应的代表性攻击类别。

表4：攻击类别（C1–C7）与流水线阶段、主要目标和代表性控制措施的对应关系。

类别	流水线阶段	主要目标	代表性控制措施
C1：提示注入 / 认知劫持	推理 → 规划 → 授权	代理策略遵循；工具选择完-整性	上下文隔离；意图规范化；工具沙箱
C2：工具/数据欺骗	观察 → 推理	状态完整性（价格、元数据、风险信-号）	多源法定人数；来源信息；合理性界限
C3：中间件操纵	授权 →执行	预览与 sign	所见即所签；类型化意图；依赖关系固定
C4：密钥/凭证窃取	执行（及运行时）	签名权限；密钥；工具凭证	最小权限；MPC/TEE 托管；密-钥卫生
C5：重放/随机数操纵	执行 ↔ 验证	交易唯一性；排序保证-tees	随机数规范；幂等消息；重放保护
C6：最大可提取价值/经济操纵	执行（提交路径）	执行价格与排序	私有订单流；意图求解器；严格边界
C7：串通／治理攻击	规划 → 授权	决策法定人数；验证者集合完整性	质押；证明；多样性约束

非目标与排除的威胁。为了让威胁分类具有可操作性，并聚焦于代理到链边界，我们有意排除与互操作性无关，或需要采用不同分析框架的类别。首先，我们不对诸如密码学破解、共识安全性违规或一层协议层的持续网络分区等基础层故障建模；我们假设标准密码学原语仍然安全，且链能够提供最终确定性。其次，除非通用大语言模型风险会实质性影响代理流水线中的工具选择、授权决策或执行正确性，否则我们不打算全面探讨这些风险（例如开放域幻觉、偏见或有害内容生成）。第三，我们不会孤立地详尽罗列智能合约漏洞类别（重入、算术错误及类似缺陷）；相反，我们将存在漏洞或快速演变的协议视为周边对抗性环境的一部分，并重点关注代理如何通过自动化意图构建、模拟和签名而暴露于风险之中。最后，只有当拒绝服务和可用性攻击会改变执行结果时，我们才会予以考虑（例如审查、延迟纳入或被迫回退到风险更高的提交路径），而不会将其作为独立的网络安全综述来讨论。

5.2.1 第 1 类：提示注入与认知劫持

机器学习模型易受对抗样本影响这一漏洞长期以来一直是研究主题，早期研究表明，微小且无法察觉的扰动可能导致错误分类 [59]。这种对对抗鲁棒性的基础理解，是如今在基于大语言模型的智能体中所见更复杂提示注入攻击的前身。

提示注入攻击针对的是代理的决策基础，而非传统的软件缺陷：对手会在代理接收的任意渠道中放入恶意指令，试图重定向意图形成、工具选择或授权步骤。实证研究和基准测试套件表明，提示注入在许多智能体设计中仍然有效，尤其是在将不可信内容（检索结果、工具输出、网页）与可信指令拼接到同一上下文中时 [192, 131]。

攻击机制。 当系统未能严格区分（i）可信指令 （系统指令、本地策略、治理规则）与（ii）不可信内容（用户文本、网页、链上元数据、检索文档、工具输出）时，注入攻击便会得逞。攻击者利用这种歧义，诱使代理：（a）调用高风险工具（例如批准、转账、签名）；（b）构造有害的交易意图（例如指向仿冒代币或攻击者控制的合约）；或（c）通过编造理由（“策略已经批准”）寻求更高权限授权。

- **直接注入攻击（交互式）**。对手是用户（或遭入侵的用户界面），会发出精心设计的提示，试图绕过约束条件或诱使代理不安全地使用工具。在链上场景中，这类攻击往往可立即变现：注入的指令旨在通过单次交互循环触发授权、资产转移或签名请求。
- **间接注入攻击（由检索介导）**。对手会将指令隐藏在代理正常运行时预期获取并解析的内容中（例如文档页面、仪表板、论坛帖子、代币元数据，甚至以文本形式呈现的交易轨迹）。间接注入攻击尤其危险，因为它会先破坏良性任务（总结、比较、评估风险），再通过引导后续工具调用或意图构建，转而操控执行 [131]。例如，恶意网页可以嵌入隐藏指令（如 CSS/HTML 技巧），这些指令一旦被检索到上下文窗口中，就会试图诱使代理发起授权，或编造转账理由。

针对故障模式的缓解措施。如果防御能强制实施结构性隔离和确定性约束，而不是依赖模型“表现良好”，效果会更可靠：

- **上下文隔离（数据不是指令）**。 将不可信内容保存在明确分隔的字段中（例如 <数据> 区块或单独的消息角色），并在运行时应用严格的解析规则。应向代理明确说明，这些字段是仅供数据使用的；同时，运行时强制规定数据字段不得引入新的工具权限或覆盖策略。这针对的是根本原因：共享上下文窗口中的指令歧义 [192]。
- **在任何授权之前进行意图规范化**。 要求代理输出结构化意图对象（例如 TIS），其中仅包含声明式字段（资产、数量、目的地、约束、截止时间）。下游组件（策略引擎、模拟器、签名器）应仅接受这一结构化表示，而不是自由文本。这样可以减少注入指令可利用的自由度，提高可审计性，并使“原本的意图”与“执行方式”彼此分离。
- **工具使用门控与最小权限路由**。 即使注入攻击在推理层成功，运行时也应阻止高风险工具调用，除非通过独立的策略检查（或获得明确的用户确认）。实际上，应围绕批准、设置授权额度、转账、签名和广播等操作实施严格的权限边界，并要求调用参数源自规范化意图，而非原始对话文本。

5.2.2 第 2 类：工具与数据源欺骗

此类攻击针对**观察** 阶段，通过破坏代理所依赖的测量数据的完整性来实施。在代理与区块链交互的环境中，欺骗性输入尤其危险，因为它们可能导致代理采取局部看来合理（基于所获数据）、但从全局来看后果灾难性的操作（基于现实情况）。其核心故障模式是将不可信的数据平面视为事实预言机。

攻击机制。 对手会污染或攻陷可信输入通道，例如 RPC 提供商、索引服务、分析仪表板、治理信息源或价格预言机。操纵方式可以是：(i) 源级（安全失陷、内部人员访问、DNS 劫持）；(ii) 协议级 （通过流动性不足和短暂的价格冲击实施预言机操纵）；或 (iii) 呈现级 （误导性元数据、伪造的“审计”声明或钓鱼文档）。攻击者的目的未必是“入侵”代理，而是诱使其基于错误前提签名或建议有害操作。

它会在哪些环节影响流水线。欺骗主要发生在**观察**阶段，并在**推理**和**规划**将被篡改的信号传播到以下结果后造成危害：(a) 扭曲的世界模型，(b) 定价错误的交易，(c) 不安全的抵押品决策，或 (d) 误导性的风险评估。

示例。 预言机与价格操纵攻击可能诱使智能体按照失真的价格进行交易，或高估抵押品价值。一种广为人知的模式是通过短暂的价格冲击操纵依赖预言机的协议，如 Mango Markets [213] 等事件所示。如果智能体将遭到安全失陷的信号视为真实依据，就可能兑换成几乎毫无价值的资产，或建立会立即被清算的杠杆头寸。

针对该故障模式的缓解措施。 防御措施应降低任何单一受污染渠道主导智能体信念状态的概率，并确保关键决策有可验证的来源信息作为依据，且受到领域约束的限制。

- **基于偏差规则的多提供方定人数机制**。对于关键指标（现货价格、时间加权平均价格、波动率、储备快照、风险评分），应查询多个独立提供方，并使用稳健统计方法（中位数、截尾均值）进行汇总。拒绝超出所配置偏差阈值的异常值，并在数据源出现重大分歧时升级处理 [50]。这直接针对“单一来源主导”这一故障模式。
- **来源与真实性检查**。优先选择提供可验证来源（签名载荷、经身份验证的传输、固定为预期公钥）的来源。维护端点和密钥的允许列表，并在审计日志中记录来源元数据，以便事件后取证能够追溯哪个输入影响了哪个决策。
- **合理性界限与不变量交叉检查**。实施成本低但有效的领域约束：相对于近期时间窗口的价格界限、代币小数位一致性、储备单调性检查、合约代码标识（字节码哈希）以及已知安全地址注册表。对于交易，在将规划提交授权之前，要求模拟结果在输入发生微小扰动时仍保持稳定（轻量级鲁棒性测试）。
- **“测量”与“决策”产物的分离**。将原始观测值（及其来源信息）与规划所用的派生特征分别存储。这样更容易检测并回滚遭投毒的特征，同时不抹去审计轨迹。

5.2.3 第3类：中间件遭入侵与意图操纵

此类别针对的是**授权**与**执行**之间的衔接环节。即使代理提出了正确的操作，且用户批准了清晰的预览，遭入侵的中间件层仍可能篡改实际签名或提交的内容。这种故障模式表现为预览意图与已签名字节之间的绑定遭到破坏。

攻击机制。攻击者入侵钱包 SDK、浏览器扩展、插件、中继器集成或依赖项（例如通过软件包供应链事件）。遭入侵的组件会在预览之后、签名之前篡改意图或交易载荷、向批处理中注入额外调用、替换目标地址，或更改调用数据，同时保持表面 UI 文本看似合理。

示例。用户批准了一项兑换意图。中间件悄然将其包装成多重调用，其中包含额外的授权或授予运营者权限的操作（例如 setApprovalForAll），从而为之后盗取资产创造条件。已签名载荷在技术上有效，但在语义上与预览内容不同。

针对故障模式采取缓解措施。控制措施应确保预览后的任何转换要么不可能发生，要么能在授予授权前被检测到。

- **所见即所签（WYSIWYS）**。根据将要签名的精确字节生成预览，并以密码学方式将预览绑定到签名请求。硬件钱包的显示屏提供了一种强有力的实现方式，因为其渲染界面与主机隔离 [333]。
- **采用规范编码的类型化意图**。使用结构化意图格式并进行规范序列化，确保签名涵盖完整的规范形式：目标地址、选择器、代币标识符、数量、截止时间和界限。规范化可降低可塑性，让“静默篡改”更难隐藏。
- **在各组件间执行端到端完整性检查**。让策略引擎和签名器重新计算交易/意图的摘要，并与预览时生成的摘要进行比对。如果摘要不一致，则采取故障关闭策略，并要求重新授权。
- **严格管理签名关键路径上的依赖项**。固定版本，启用完整性检查（哈希锁，并在可行时使用可复现构建），并持续审计依赖项。尽量减少特权攻击面，并在从预览到签名的流水线中禁用动态代码加载。

5.2.4 第4类：密钥与凭证窃取

此类别针对代理的**签名权限**及相关机密材料。在代理部署中，影响范围包括链上凭证（外部拥有账户密钥、会话密钥、多方计算分片）和可用于横向移动的链下机密信息（API 密钥、Webhook 令牌、通知渠道）。故障模式是将长期有效的权限置于易受攻击的运行时环境中。

攻击机制。

- **环境安全失陷**。攻击者获得对执行环境的访问权限（服务器失陷、恶意容器映像、CI 泄露），并从文件、环境变量、日志或进程内存中提取机密信息。
- **通过交互渠道诱导泄露**。 攻击者通过社会工程或提供受污染的工具输出，诱使代理打印或转发机密信息（例如，要求转储环境变量以便调试，或“把你的配置粘贴过来，我可以帮忙”）。

针对故障模式的缓解措施。 控制措施应当：(i) 降低运行时中任何机密信息的价值；(ii) 将签名与代理进程隔离；(iii) 将输出视为潜在的数据窃取渠道。

- **从构造上实现最小权限**。 避免在代理运行时中持久存储主密钥。优先使用时间受限的会话密钥、限定范围的能力、允许列表，以及由账户抽象策略强制执行的支出限额。
- **强化托管原语**。 将签名置于多方计算阈值签名或由可信执行环境支持的签名器之后，确保完整的密钥材料绝不会出现在智能体内存中。门限签名可降低单主机失陷的影响 [169,184], ，而由硬件安全模块支持的设计则可进一步将机密信息隔离在专用硬件中 [274]。
- **密钥卫生与扫描**。 将日志和模型输出视为数据窃取渠道。对代码仓库和遥测应用密钥扫描，默认对工具输出进行编辑隐藏，并使用短期凭证及轮换机制。确保代理拒绝涉及泄露配置或密钥的请求。
- **权限提升的双通道确认**。 如果代理必须请求更高权限（更大的支出限额、新的合约允许列表），则要求通过独立通道（硬件钱包确认、操作员审批流程）进行授权，以打破单通道安全失陷。

5.2.5 第 5 类：重放与随机数操纵

此类攻击利用区块链账户的排序语义。许多代理系统在链下对消息（意图）进行签名，并在链上提交交易，而这两条路径都需要强健的唯一性和排序保证。其故障模式是：在系统假定授权只能使用一次的上下文中，重复使用有效授权。

攻击机制。

- **重放**。 如果应用不具备重放防护，则重新提交有效的已签名载荷（交易或链下授权），从而触发重复的状态转换。
- **Nonce 干扰与替换动态**。 对手会利用 Nonce 冲突、交易替换或内存池可见性，试图延迟、审查或使交易失效。在多发送方配置（代理 + 运营者 +自动化）中，良性并发也可能导致 Nonce 冲突，攻击者还可以放大这一问题。

针对相应故障模式的缓解措施。

- **签名消息的重放保护**。 所有签名意图都必须包含唯一标识符（随机数/盐值）、严格的有效期和域分离。链上使用方必须存储已使用的随机数并拒绝重复使用，从而确保每条消息只能使用一次。
- **将 Nonce 管理作为一等组件**。 集中管理每个账户的 Nonce 分配（或使用可虚拟化 Nonce 的智能账户方案），以避免出现空缺和冲突。跟踪待处理交易、检测交易丢弃，并以确定性的方式处理替换交易。对于账户抽象路径，利用验证逻辑在 validateUserOp 层强制执行防重放约束。
- **幂等应用设计**。 在可行的情况下，将状态转换设计为幂等或可安全重复执行（例如，“将仓位设为 X”而非“增加 X”），以限制重放发生时造成的损害。

5.2.6 第 6 类：最大可提取价值与经济操纵

本课程聚焦于**提交路径** 以及执行排序的经济博弈。 [352] 这并非软件缺陷，而是一种激励相容的提取机制，会系统性地惩罚公共内存池中的朴素执行策略 [80, 251]。其故障模式是向对抗性拍卖暴露有价值的意图。

攻击机制。 内存池中可见的操作会被观察，并遭到对抗性重新排序或前后夹击。三明治攻击是针对自动做市商交易的典型提取模式，但智能体也容易受到跟单交易、尾随交易和对抗性交易替换的影响。

与故障模式相对应的缓解措施。

- **私有订单流和中继器**。 如果可用，请通过私有通道提交，以降低内存池可见性并防止简单的三明治攻击 [240]。
- **基于意图的执行和求解器市场**。 优先采用拍卖或基于求解器的执行方式：代理对意图进行签名，并将执行委托给能够将 MEV 保护纳入考量的竞争性市场，而不是直接广播容易受攻击的交易。
- **将硬性界限设为可强制执行的约束条件**。 设置严格的滑点、截止时间和价格限制，并要求在对抗性假设下进行模拟后再执行。在规划中，应将最大可提取价值风险视为首要成本组成部分，而非事后才考虑的因素。
- **MEV 感知规划信号**。 将内存池和流动性状况（预期三明治攻击成本、深度敏感性）纳入规划，使代理能够判断何时不应交易。

5.2.7 第 7 类：多智能体串谋与治理攻击

匿名凭证的原则可扩展用于去中心化自治组织中的匿名投票，智能体可以参与治理，而不透露自己的投票偏好 [70]。这有助于缓解去中心化治理中的串通与胁迫。

本类别专门针对**模式 V** 系统，这类系统的安全性依赖于智能体、验证者或投票者组成的法定人数。 [269]分布式架构减少了单个智能体的故障模式，但会引入集体行动失败。其核心故障模式是控制法定人数，而非攻破任何单个组件。

攻击机制。一种常见手段是女巫策略：攻击者创建看似相互独立的智能体，以掌控法定人数权重并批准恶意行动 [323]。更广泛地说，贿赂、形成卡特尔以及相关性失陷，可能导致验证者集合批准有害提案，同时在程序上看似合规。

针对该故障模式的缓解措施。

- **经济威慑（质押与罚没）**。要求参与者质押具有实际经济价值的资产，并对可证实的恶意授权设定透明的罚没条件。
- **高信任角色的身份与证明**。 使用可验证凭证 [116] 或证明门控来管理验证者角色，限制可加入法定人数集合的人员，并支持对运营者来源信息进行审计。
- **通过多样性约束降低相关故障风险**。 将独立性假设纳入治理设计（运营者多样性、司法管辖区多样性、实现多样性），并轮换验证者集合，以降低长期控制风险。
- **问责材料与可审计性**。 要求验证者公布签名理由、模拟记录和证据指针。这会提高协同欺骗的成本，并改善事件后归因。

5.2.8 现实世界安全事件分析

审视现实世界的安全事件，可为智能体区块链系统设计提供宝贵经验。由于该领域尚处于起步阶段，针对智能体系统的全面事件数据有限，但去中心化金融和人工智能系统中的相关事件仍能提供有益见解。

智能合约漏洞 智能合约攻击的历史表明，链上代码执行存在风险。包括 The DAO 黑客攻击事件、Parity 钱包冻结事件以及多起去中心化金融攻击在内的重大事件，已造成数十亿美元损失。这些事件凸显了全面审计、形式化验证和纵深防御策略的重要性。

对于代理系统而言，代理所交互的合约中的智能合约漏洞构成了重大风险。代理的设计应能够识别并避免与未经审计或可疑合约进行交互。在执行前对交易执行结果进行模拟和验证，有助于识别潜在的攻击条件。

预言机操纵 预言机操纵攻击利用了去中心化金融协议对外部价格数据的依赖。攻击者通过闪电贷、低流动性市场及其他技术操纵预言机价格，从依赖被操纵价格的协议中攫取价值。

依赖预言机数据进行决策的代理系统也会面临这些风险。代理应使用多个相互独立的预言机来源，对预言机数据实施合理性检查，并在预言机数据出现异常时能够安全地进入故障状态。

治理攻击 治理攻击曾利用去中心化自治组织和去中心化金融协议中的基于代币的投票机制。攻击者通过闪电贷、积累代币或社会工程手段获取投票权，从而通过恶意提案。

参与治理的代理系统必须经过设计，以抵御操纵并避免助长治理攻击。代理应实施保障措施，防止对恶意提案投票，也不应在未经适当验证的情况下自动遵循治理建议。

人工智能系统故障 尽管这些故障并非区块链领域特有，但其他领域的人工智能系统故障仍能提供相关经验。聊天机器人遭操纵、推荐系统故障以及自动驾驶车辆事故等事件表明，在缺乏充分保障措施的情况下，将人工智能系统部署于高风险环境会带来风险。
智能体区块链系统应从这些事件中吸取教训，实施健壮性测试、监控和故障安全机制。区块链交易不可逆，这会加剧人工智能故障造成的后果，因此预防和快速检测尤为重要。

5.2.9 纵深防御：分层安全架构

可信人工智能涵盖一系列理想特性，包括可靠性、鲁棒性、公平性和透明度；这些特性对于建立用户对智能体系统的信任都至关重要 [14]。
智能体区块链系统要实现有效的安全防护，需要采用纵深防御，通过多个彼此独立的防护层，针对各种威胁提供冗余保障。单一安全措施并不足够；全面防护需要采用分层防御，在多个层面应对威胁。

第 1 层：输入验证与清理 第一道防线是在所有输入到达代理的推理组件之前，对其进行验证与清理。这包括筛选区块链数据中可能存在的恶意内容、根据预期格式验证用户输入，以及清理外部数据源。输入验证应采取保守策略，拒绝含糊或可疑的输入，而不是试图对其进行解释。

有效的输入验证需要了解代理输入渠道暴露的攻击面。区块链数据、用户消息、API 响应及其他输入源各自带来不同的风险，因此需要采用针对性的验证策略。

二层：**推理约束与护栏**第二层限制代理的推理过程，防止恶意输入绕过输入验证后产生有害输出。这包括用于确立行为边界的系统提示、用于拦截潜在有害响应的输出过滤，以及用于检测异常决策模式的推理监控。

护栏机制应以安全失效为设计原则，在不确定时默认采取保守行为。护栏机制的严格程度与智能体效用之间的权衡，必须根据应用的风险概况进行校准。

第 3 层：交易验证与模拟第三层会在执行前验证拟议的交易，确保其符合策略约束，并能按预期执行。这包括策略引擎针对支出限额和允许列表进行检查、依据当前区块链状态进行模拟，以及验证交易结果是否符合代理的意图。

交易验证应在隔离环境中进行，以防遭到恶意合约或状态操纵。多项独立验证检查可提供冗余，防止单项检查发生故障。

第 4 层：执行控制与熔断器 第四层控制交易执行，并提供紧急停止能力。这包括通过速率限制防止资金迅速耗尽、针对异常交易要求额外授权的渐进式授权，以及在检测到异常时停止执行的断路器。

断路器的设计应具有适当的灵敏度，在真正出现异常时触发，同时避免误报干扰正常操作。应定期根据运营经验审查并更新断路器的触发标准。

第 5 层：监控与响应 第五层提供持续监控和事件响应能力。这包括实时监控代理行为和交易结果、在出现潜在问题时通知操作员的警报系统，以及用于调查和修复安全事件的事件响应流程。

监控系统应捕获足够详细的信息，以便进行取证分析，同时遵守隐私约束。事件响应流程应形成文档、经过测试并定期更新。

5.2.10 详细攻击场景与案例研究

要了解威胁态势，就需要考察具体的攻击场景，了解理论上的漏洞如何在实践中显现。以下案例研究取材于有记录的事件和安全研究，揭示了智能体区块链系统面临的实际风险。

场景 1：通过恶意合约元数据实施提示注入 攻击者部署一份精心制作元数据（名称、符号或其他字符串字段）的智能合约，其中包含提示注入载荷。当代理在正常操作中查询该合约时，这些恶意字符串便会被纳入代理上下文。注入载荷指示代理将资金转至攻击者的地址，并将此行为包装成合法操作。如果代理的提示处理未经过适当净化，便可能执行这些恶意指令，导致资金被未经授权转移。

此场景说明，必须将所有外部数据（包括从区块链状态中检索的数据）都视为可能具有对抗性。缓解措施包括严格进行输入净化、分离数据与指令通道，以及在执行前根据预期模式验证代理输出。

场景 2：从代理交易中提取最大可提取价值 一个执行去中心化金融策略的代理向公共内存池提交了一笔大额兑换交易。最大可提取价值搜索者检测到这笔待处理交易后，构造了一次三明治攻击：先通过一笔抢先交易使价格朝不利方向变动，随后执行代理的交易，最后再执行一笔跟进交易以攫取价格差。代理的交易以低于预期的价格执行，差价则被最大可提取价值搜索者提取。
此场景说明了公共区块链执行环境的对抗性。缓解措施包括使用私有内存池、在交易构造时考虑最大可提取价值并设置适当的滑点限制，以及通过能够将最大可提取价值内部化的求解器网络进行基于意图的执行。

场景 3：对人工审批者实施社会工程 攻击者发现，某目标组织对代理交易采用人工参与的审批流程。攻击者通过提示注入或其他手段操纵代理，使其生成一系列看似合法的交易，逐渐让人工审批者习惯于不经仔细审查便予以批准。待审批者充分麻痹后，攻击者便触发一笔恶意交易；这笔交易看起来与此前用于使审批者放松警惕的交易相似，但目标地址不同。
这一情景表明，人工监督并非万灵药，本身也可能成为老练攻击者的目标。缓解措施包括变换审批工作流以防止习惯化、通过自动化异常检测标记异常模式，以及实行职责分离，要求高价值交易由多个相互独立的批准者审批。

5.2.11 智能体区块链安全的形式化方法

人工智能对齐领域致力于确保人工智能系统的行为符合人类价值观和意图，这是在高风险环境中部署自主代理的一项关键先决条件 [76]。人工智能控制问题，即确保先进人工智能系统仍处于人类控制之下这一挑战，是一个由来已久的研究主题；随着强大的大语言模型问世，这一问题变得愈发紧迫 [24]。

将形式化方法应用于智能体区块链系统，是一个前景可观但充满挑战的研究方向 [25, 150, 257]。传统形式化验证技术是为确定性软件系统开发的，必须经过调整，才能应对基于大语言模型的智能体的非确定性，以及区块链执行环境的对抗性动态。

运行时验证为静态形式化方法提供了一种互补途径，能够持续监控代理行为是否符合指定的安全属性。通过使用运行时监视器对代理执行过程进行插桩，系统

可以实时检测策略违规、异常行为模式和潜在攻击。结合自动干预机制（例如交易取消或密钥撤销）后，运行时验证能够针对代理错误和对抗性操纵提供纵深防御。挑战在于，所指定的安全属性既要足够精确，能够通过形式化方法进行验证，又要足够灵活，以适应自主代理的合法运行需求。

6 标准 与接口层：智能体系统的连接纽带（2025年）

智能体—区块链系统的安全性、互操作性和实际可行性，关键取决于将推理组件连接到执行底层架构的接口层的质量。接口不仅仅是为开发者提供便利；它们还定义了信任边界、调节权限，并决定整个系统能够实现的安全保障。设计良好的接口层能够抽象复杂性、落实策略约束并实现安全的可组合性；而定义不完善的接口则会扩大攻击面，并导致脆弱的集成。
截至2025年，整个生态仍然四分五裂，由事实标准、专有 API 和新兴开放协议混杂构成。本节分析构成智能体系统连接纽带的三种接口层：（i）将智能体与链下和链上能力相连接的工具调用协议；（ii）管理授权的钱包与签名接口；以及（iii）使链上环境适用于智能体控制的智能合约交互模式。

6.1 工具调用协议：智能体与工具之间的通用语言

代理若要超越纯粹推理而采取行动，就必须以结构化、可解释且可审计的方式调用工具。工具调用协议定义了智能体用来发现能力、指定函数语义、交换输入与输出以及协商权限的通用语言。这些协议的演进体现了从临时集成逐步转向明确结构与关注点分离的过程。

6.1.1 临时应对时代（2023年前）

早期代理框架（包括 LangChain 和 AutoGPT） [130],依赖自定义的、特定于应用的工具定义。开发者手动封装 API，并通过非正式的提示约定或无类型接口将其提供给模型。尽管这种方法灵活，但存在若干结构性弱点：

- **脆弱性**。 工具语义与实现细节紧密耦合。细微的 API 变更就可能在不知不觉中导致代理行为失效。
- **可扩展性有限**。 每增加一项新能力都需要专门集成，从而阻碍共享工具生态系统的形成。
- **安全边界薄弱**。 工具访问通常等同于在宿主进程中不受限制地执行代码，使推理与执行之间的隔离不复存在。

这些限制在区块链场景中尤为棘手，因为工具滥用可能直接造成不可逆的财务损失。

6.1.2 通过函数调用实现事实标准化（2023—2024年）

主要大语言模型提供商推出**函数调用**接口，标志着一个决定性的转折点。开发者可以使用结构化格式（例如 JSON Schema ）指定工具模式，模型则能够输出机器可读的调用规范，供应用确定性地执行。这一转变将工具选择和参数化的责任从提示工程转移到了模型推理，从而实现了更加模块化的智能体架构。

这一范式迅速被多个提供商采用，实际上使其成为一项事实标准。然而，从去中心化和对抗性环境的角度来看，其局限性十分明显：

- **提供商中心化**。 工具调用语义受限于专有 API 和不透明的模型行为，降低了可验证性和可移植性。
- **隐式信任假设**。调用工具的决策并未通过密码学方式绑定到策略约束或用户意图，因此审计和执行变得更加复杂。

因此，尽管函数调用显著改善了开发者体验，但它仍不足以作为实现最小化信任的代理与区块链集成的基础。

6.1.3 新兴开放协议（2024–2025年）

作为回应，人们提出了几种开放协议，旨在将智能体、模型和工具解耦，同时保持互操作性：

- **模型上下文协议（MCP）**。MCP 提出了一种标准化接口，用于向模型提供上下文，包括工具、数据和用户状态。通过明确区分用户、应用和模型的角色，MCP 旨在支持模块化生态系统，使任何合规智能体都能与异构工具提供方实现互操作 [93, 79]。
- **通用工具调用协议（UTCP）**。UTCP 专注于将原生 API 直接开放给智能体，而无需定制封装层。它对能力协商和可验证输出的重视，与高风险自动化的需求高度契合，尤其是在工具输出会影响授权或执行的情况下。
- **智能体间（A2A）协议**。A2A 协议将工具调用泛化为智能体间通信，把其他智能体视为一等能力。这种抽象对于多代理工作流（模式 V）至关重要，因为其中的协调、验证和协商会跨越组织边界进行。

虽然这些协议在范围和成熟度上各不相同，但它们有着共同目标：让智能体能力明确、可检查且可组合。

6.1.4 缺失的一层：权限、风险提示与可审计性

尽管进展迅速，仍存在一个关键缺口：在工具接口层面，尚无广泛采用的标准来规范能力范围限定、权限协商和风险披露。智能体通常无法通过程序查询某个工具获准执行哪些操作、受到哪些约束，或会产生哪些副作用。因此，权限往往需要手动配置，增加了权限过度授予和错误配置的可能性。

同样尚不成熟的还有**审计与来源信息**机制。很少有工具协议要求对响应进行密码学认证，或将响应绑定到已声明的能力集。在对抗环境中，这使得人们难以区分真实的工具输出与伪造或重放的响应。成熟的接口层可能需要工具对输出进行签名、公布明确的能力描述符，并公开机器可读的风险元数据，从而让智能体能够推理信任和权限，而不是想当然地认为它们存在。

6.2 钱包接口：从意图到执行的跨链桥

钱包接口是智能体—区块链系统中安全性最关键的边界，因为概率性的意图形成会在此转变为确定性的 授权。实际上，钱包层负责协调：(i) 密钥托管与签名生成，(ii) 策略执行与用户同意，以及 (iii) 决定人类认为自己授权了什么的交易呈现（预览／模拟）。因此，钱包不只是“密钥持有者”，更是授权控制平面 ，其设计在很大程度上决定了智能体工作流是否可审计、是否符合最小权限原则，以及能否抵御安全失陷 [230]。

传统的以用户为中心的钱包（例如，每笔交易都 require 手动确认的浏览器扩展）可以支持模式 II 工作流，但随着自主性增强，它们会成为瓶颈，也会造成语义不匹配。2024–2025年，新兴的设计空间正转向智能体钱包接口：具备明确策略钩子的程序化签名端点、类型化交易／意图表示、基于模拟的预览，以及可验证日志。从概念上说，这一层应提供三种能力：(1) 能力范围限定 （代理获准执行哪些操作），(2) 执行正确性（实际上将在链上发生什么），以及 (3) 归因与问责 （谁／什么授权了什么，以及原因）。

6.2.1 账户抽象（ERC-4337）：可编程钱包的基础

账户抽象（AA），尤其是 ERC-4337 风格的智能账户，为智能体钱包提供了天然的基础，因为它将授权逻辑从静态的外部拥有账户密钥转移到了可编程验证代码中 [83, 280, 55]。AA 不再将钱包视为被动的签名器，而是将其转变为能够在执行之前强制执行规则的主动验证器。ERC-4337 架构通常由四个相互作用的组件构成：

- **用户操作**。 一种结构化请求对象，用于编码智能账户要执行的操作（以及验证该操作所需的授权数据）。对于代理系统而言，该对象是

规划器或意图生成器的自然输出，因为它比原始调用数据更具结构化，并且旨在通过程序进行验证 [280, 55]。

- **捆绑器**。一种无需许可的服务，从专用内存池收集用户操作并将其提交到链上，通过分摊纳入成本来实现更丰富的提交策略（例如私有路由，或在相邻生态系统中采用求解器式执行） [280]。
- **入口点**。一种规范化智能合约，通过调用智能账户的验证逻辑来验证并执行操作。从安全角度来看，这统一了执行所在的位置，并使策略模块能够跨账户进行组合 [280]。
- **支付主**。一种可选的智能合约，用于赞助 Gas（并可施加额外约束），从而支持“无 Gas”或由应用支付的操作，并将代理权限与持有原生手续费代币解耦 [280]。

这对代理为何重要。账户抽象是实现**模式 III（委托执行）**最直接的推动因素，因为它支持范围严格限定且可撤销的能力（通常通过会话密钥、角色模块、守卫和支出限制器实现）。其关键特性是确定性执行：即使代理遭到入侵，也无法超出账户验证和策略模块中编码的界限。这使账户抽象成为签名边界处最小权限原则的实际实现，而非尽力而为的指导方针。

6.2.2 多方计算与可信执行环境：链下策略与硬件安全

机构级加密资产托管面临诸多挑战，包括需要健全的内部控制和满足监管合规要求，这推动了复杂托管解决方案的发展 [165]。利用用户社交网络的社交恢复和钱包管理方案，为传统助记词提供了另一种选择；对于非技术用户而言，安全管理助记词可能颇具难度 [144]。

账户抽象策略模块功能强大，但某些部署需要满足以下条件：(i) 托管结构跨越多个设备或组织；(ii) 策略取决于丰富的链下上下文；或 (iii) 具备比应用主机所能提供的更强密钥隔离保障。这些需求在**模式 IV（自主签名）**部署中十分常见，因为代理必须频繁且迅速地执行操作，同时受到密码学约束。

多方计算／门限签名方案（TSS）。多方计算阈值签名系统以分布式份额取代单一私钥，并通过协议生成标准签名，而无需在同一位置重构密钥。这降低了单主机遭入侵风险，并支持共享控制配置（例如，代理份额 + 用户设备份额 + 策略服务份额） [169, 184]。在代理式钱包接口中，多方计算通常与链下策略决策步骤配合使用：只有交易满足约束条件时，策略服务才会参与签名法定人数，从而将治理转化为密码学否决。

可信执行环境（TEE）与基于硬件安全模块的签名。基于可信执行环境的设计和基于硬件安全模块的托管，会将密钥材料隔离在经过加固的边界内，并向代理主机提供精简的签名 API，从而降低内存抓取和运行时入侵的风险 [274, 227]。在实践中，密钥安全的核心价值不只是“硬件”，更在于策略耦合签名：安全区/硬件安全模块可以要求请求经过身份验证，验证元数据（限额、目标地址、函数选择器、截止时间），并在请求违反策略时拒绝签名。

智能体钱包的接口层设计目标。在账户抽象和链下托管设计中，钱包接口要对智能体最有用，就应提供：

- **类型化授权界面：**采用规范编码的结构化操作/意图，减少规划内容、预览内容和签名内容之间的歧义 [280]。
- **签名边界处的策略钩子：**在签名生成之前评估可强制执行的约束（允许列表、限额、时间界限），而非在提交之后才评估 [230]。
- **预览与模拟对齐：**通过高保真模拟和渲染，尽可能缩小用户理解与已签名字节之间的语义缺口（在人类仍参与其中时尤为重要）。
- **审计级可观测性：**通过持久化日志关联意图、策略决策、签署者批准和链上回执，从而支持事件后取证与问责。

总而言之，现代钱包接口正趋向于一种混合模型：通过链上可编程性（账户抽象模块）实现明确、确定性的约束，并结合链下强化托管（多方计算/可信执行环境/硬件安全模块），以实现更强的密钥隔离和丰富上下文的策略。这种组合为从用户批准的意图（模式 II）扩展到受约束的委托（模式 III），并在合理情况下进一步实现具备纵深防御的自主执行（模式 IV），奠定了切实可行的基础 [169, 184, 227, 280]。

表 5：命令式交易 与基于意图的交互：从代理安全性和验证的角度 n.

维度	命令式交易（优先使用调用数据）	意图（优先考虑结果）
主要产物	指定如何执行的底层调用数据	指定什么结果可接受的签名声明
代理认知负荷	高：协议特定参数、授权、路由细节	较低：声明式约束（数量、界限、截止时间），更少特定协议的细节
失效模式	调用数据指定错误、隐藏的副作用、脆弱的参数化	约束条件指定错误；求解器故障；部分成交语义
MEV 暴露	广播至公共内存池时较高	通常通过私有订单流、批处理或求解器竞争来降低 [80, 251, 240]
验证	必须从执行轨迹中重构意图	将交易回执和事后状态与已签名的约束进行比较；审计更加清晰 path

6.3 智能合约接口：为智能体设计

智能合约是智能体系统最终的执行界面。即使代理运行时、策略引擎和签名器都经过精心设计，针对人工驱动的前端优化的合约接口也可能不适用于程序化调用：参数化存在歧义、副作用未明确提示、回滚原因不清晰，以及后状态信号不足，都会增加代理（i）错误指定操作、（ii）未能检测到不安全的状态转换，或（iii）无法以确定性方式恢复的概率。相比之下，便于代理使用的合约旨在尽量减少歧义并最大限度地提高可验证性：它们提供类型化、声明式的交互界面，明确的前置条件和后置条件，机器可解析的交易回执，以及一致的失败语义，从而支持自动化规划和稳健验证。

6.3.1 意图中心范式

面向代理的设计中的一个关键转变，是从命令式的底层交易构造转向**基于意图的**交互。意图系统不再要求代理构建特定于协议的调用数据（例如，使用一长串参数调用路由器），而是允许代理在约束条件下表达声明式结果，并将寻路和执行细节委托给专业执行器（求解器），由它们在市场状况下竞争以满足该意图 [220]。这种委托在对抗性环境中尤为重要，因为天真的内存池广播会因最大可提取价值动态而持续受到惩罚 [80, 251]。

意图与命令式调用。 表 5 总结了从代理视角看二者的核心区别：意图降低了动作空间复杂度，并将执行风险（路由、排序、部分成交逻辑）转移到受约束的市场中，同时通过带签名且防重放的消息格式保留用户级约束。

为何意图适合代理。 意图接口通常在三个实际方面更适合代理，也更安全。首先，它们将行动空间压缩 为一小组声明式字段，从而减少参数混淆和意外过度授权的可能性。其次，它们与流水线分离自然契合：代理生成意图，策略层对其进行模拟和评估，执行层则在有界约束下尽力完成。第三，意图会在设计中明确体现最大可提取价值风险：代理对约束进行签名，而执行可通过降低内存池可见性的机制，或通过竞争将排序策略内部化的机制来完成 [80, 251, 240]。

6.3.2 其他便于代理使用的合约模式

除了意图之外，还有几种接口模式可以显著提升自主或半自主代理的安全性、可观测性和可恢复性。

1）将可验证回执作为一等输出。代理流水线依赖**验证**阶段。合约可以通过发出结构化、明确无歧义的事件来支持这一点，将其用作机器可解析回执：哪些资产发生了转移、应用了哪些界限、选择了哪条路由、收取了哪些费用，以及调用了哪个授权模块。回执应包含标识符，以便代理将 (i) 已签名的意图（或规范化的交易摘要）与 (ii) 链上执行结果关联起来，从而在出现偏差时进行确定性的后置状态检查和取证重建。

2）标准化的失败语义和明确的前置条件。 当智能合约发生通用回滚时，代理无法可靠地判断失败类型（流动性不足、截止时间已过期还是权限违规）。因此，便于智能体使用的设计会提供明确的前置条件检查和结构化的失败原因。关键要求并非风格层面的，而是操作层面的：智能体必须能够将失败对应到恢复操作（重新规划、重新模拟、升级为人工审批或停止），而无需猜测。

3）为单次调用决策支持提供丰富的视图接口。 智能体经常需要在速率限制和不稳定的远程过程调用（RPC）环境下运行。提供全面视图函数（头寸、限额、预言机引用、抵押因子、配置标志）的合约可以减少多次调用竞态条件，并简化规划。汇总读取也能提升可追溯性：智能体可以记录促成某个意图的确切状态快照。

4）最大限度减少权限的入口点。 在可能的情况下，合约应避免要求广泛授权或无限期的操作员授权。代理工作流受益于范围严格限定的授权（有上限的额度、逐操作许可、时间限制），因为这些措施能缩小代理安全失陷的影响范围，并与委托执行模式自然契合。这与账户抽象控制措施相辅相成：合约接口不应迫使代理重新采用粗粒度权限管理。

要点。 接口设计并非表面功夫。对于智能体系统而言，它是一道安全边界。以意图为中心的接口能减少操作歧义，并通过将执行转移至有界竞争机制，缓解常见的经济攻击；而代理友好型合约模式则能提升验证和恢复过程的确定性。这些特性共同构成稳健的技术栈：代理生成声明式意图，钱包执行策略和最小权限原则，合约则在明确约束条件下发出可验证、机器可读的执行结果 [220, 80, 251, 240]。

6.3.3 协议实现细节

代理-区块链接口的实际实现需要关注众多协议级细节，这些细节会影响安全性、可靠性和互操作性。本节将探讨主要接口标准的关键实现考量。

ERC-4337 实现考量 在代理系统中实现对 ERC-4337 的支持，需要处理若干不易察觉的复杂问题。构造用户操作时必须考虑 Gas 估算所面临的挑战，因为验证和执行所需的 Gas 可能不同于标准交易的 Gas 估算结果。捆绑器的选择会影响交易纳入时间和最大可提取价值暴露。集成支付主需要谨慎处理授权流程和余额管理。
ERC-4337 的验证阶段对可执行的操作施加了限制，包括对存储访问和外部调用的限制。代理系统必须确保其验证逻辑符合这些限制，以避免用户操作被拒绝。验证与执行阶段的分离也带来了时序检查与使用（TOCTOU）漏洞的风险，必须谨慎管理。

MCP 服务器实现 为区块链交互实现 MCP 服务器，需要在表达能力与安全性之间取得平衡。工具定义必须足够精确，以便代理有效使用，同时避免提供过于宽泛、可能遭到滥用的能力。错误处理必须提供足够的信息，帮助代理从故障中恢复，同时避免泄露敏感信息。

MCP 服务器中的资源管理必须考虑区块链操作可能持续较长时间。交易确认可能需要数分钟甚至更久，因此需要妥善处理超时并报告进度。连接管理必须能够妥善应对网络中断，尤其是在操作涉及多笔交易或需要监控链上事件时。

意图协议考量 基于意图的协议在求解器选择、执行质量和争议解决方面增加了额外的复杂性。提交意图的代理系统必须准确说明其要求，包括可接受的价格范围、时序约束和偏好的执行场所。对意图履行情况的监控必须能够检测部分成交、执行失败和次优结果。

不同实现中的意图协议信任模型差异显著。有些协议依靠声誉系统和经济担保来确保求解器诚实性，另一些则采用密码学证明或乐观验证。代理系统必须了解所用协议的信任假设，并采用适当的验证和回退机制。

6.3.4 去中心化身份与可验证凭证

去中心化匿名凭证领域的基础性研究为这样的系统奠定了基础：用户可以证明关于自身的陈述，而无需透露自己的身份 [118]。较新的系统（如 zk-creds）在此基础上利用零知识简洁非交互式知识论证创建灵活的匿名凭证，并可将其集成到现有的身份基础设施中 [259]。将万维网联盟可验证凭证与 FIDO 身份验证标准相结合的框架正在朝着

更以用户为中心的身份管理模型迈进。在这种模型中，用户（及其智能体）能够更好地掌控自己的个人数据 [176]。

将去中心化身份（DID）标准与智能体区块链系统相结合，为隐私保护授权和合规开辟了新的可能性。可验证凭证使智能体能够证明其主体的属性（例如认证状态、司法管辖区或风险承受能力），而无需透露不必要的个人信息。已有多个框架被提出，用于构建以用户为中心、注重控制权和隐私的去中心化身份系统 [19, 31]。区块链在访问控制中的应用也已得到探索，重点是创建更加灵活、安全的系统 [42]。去中心化身份系统的设计也得到了深入研究，重点在于隐私、安全性和可用性之间的权衡 [56]。可验证凭证系统的效率也是一项重要研究领域，重点是构建更具可扩展性、性能更优的系统 [78]。CredenceLedger 系统提供了一个用于管理和验证凭证的去中心化框架 [26]。这项能力对于受监管金融活动尤为重要，因为代理必须证明其符合了解你的客户（KYC）和反洗钱（AML）要求。

自主权身份（SSI）框架为用户掌控的凭证管理奠定了基础，使个人能够托管自己的身份属性，并选择性地向依赖方披露这些属性。结合零知识证明 [45], 后，这些系统可以让代理在不透露底层数据的情况下，证明其符合复杂的策略要求。例如，代理可以证明其委托人是合格投资者，而不透露其净资产；也可以证明其居住在获准的司法管辖区，而不披露确切位置。将这些身份原语与代理授权框架相结合，是实现合规自主金融活动的一个重要方向。

6.3.5 P代理兼容性协议设计原则 性

设计能够 与人工智能智能体良好协作的协议，需要关注若干不同于传统协议的关键原则 1
以人为本的界面 设计。

显式状态表示 代理受益于显式、完整的状态表示，而非隐式或不完整的信息。协议应提供全面的状态快照，使代理能够推理其操作的完整上下文。对人类用户而言显而易见的隐藏状态或副作用，可能会让缺乏上下文知识、无法推断缺失信息的代理感到困惑。

确定性行为 代理在与表现出确定性、可预测行为的系统交互时表现最佳。非确定性结果、竞争条件或依赖时序的行为会使代理推理复杂化，并可能导致意外结果。若无法避免非确定性，协议应明确说明不确定性的来源和界限。

结构化错误处理 代理需要结构化、机器可读的错误信息，以便诊断故障并从中恢复。依赖自然语言解释的面向人类的错误消息，对代理处理而言可靠性较低。协议应提供错误代码、结构化错误数据和明确的恢复指导，供代理以编程方式处理。

版本管理与兼容性 代理系统可能会长期运行，在此期间协议也会不断演进。明确的版本控制、向后兼容策略和弃用时间表，能够帮助代理平稳适应协议变更。未充分提前通知就进行破坏性变更，可能导致代理故障，且难以诊断和解决。

速率限制与资源管理 智能体可能会生成大量请求，尤其是在探索或错误恢复期间。协议应实施明确的速率限制策略，并提供有用的反馈，使智能体能够调整其行为。缺乏明确反馈的资源耗尽可能导致智能体陷入故障循环。

7 比较能力矩阵：2025年最新技术水平

为对当前代理—区块链生态系统提供结构化、以证据为基础的概览，我们从最初筛选出的85个符合条件的系统中，选取20个具有代表性的平台、协议和框架进行了比较分析。我们从13个能力维度对每个系统进行评估，涵盖：（i）授权与托管技术栈（托管模型、签名路径、策略执行、恢复）；（ii）智能体执行技术栈（意图接口、预览/模拟、可观测性）；以及（iii）运营攻击面（工具接口、链范围、部署成熟度）。

由此形成的能力矩阵（表6）旨在以高信息密度呈现内容：它支持并列比较，揭示常见的架构权衡，并突出反复出现的缺口，这些缺口也促使我们在后文讨论开放问题。

7.1 系统选择与分析方法

选择策略与资格评估。 我们通过汇总以下各类资料中反复提及的系统，形成初始候选集：(i) 代理钱包技术栈和账户抽象生态系统的开发者文档与参考实现，(ii) 广泛使用的去中心化金融执行与模拟基础设施，以及 (iii) 知名的托管与自动化服务提供商 [103,89]。随后，我们依据与研究范围相符的纳入标准进行筛选：(a) 系统有公开文档，且规格说明足够详尽，能够支持能力编码；(b) 系统可能参与第4节所述至少一种代理—区块链集成模式（数据平面分析、意图生成、委托执行、自主签名或多代理工作流）。这一选择与筛选流程遵循区块链安全综述中系统性证据综合的一般原则，并采用最佳实践所推荐的明确标准和可复现编码产物 [145]。

生态系统的分层覆盖。 为避免过度代表任何单一层级，我们选择了能够覆盖代理在实践中会遇到的主要接口层级的系统：

- **全栈代理工具包：** 将工具调用、链上操作和开发者体验整合在一起的端到端框架（例如 Coinbase AgentKit）。
- **钱包基础设施提供商：** 提供程序化托管和策略控制的嵌入式钱包与签名后端（例如 Privy、Turnkey）。
- **智能合约钱包与 SDK：** 模块化智能账户、账户抽象 SDK 以及扩展生态系统（例如 Safe{Wallet}、Biconomy、ZeroDev）。
- **基于意图的协议：** 用于意图提交和考虑 MEV 的结算的求解器或拍卖中介执行层（例如 CoW Protocol、1inch Fusion）。
- **安全与自动化平台：** 智能体经常依赖的监控、中继和运维工具，以实现安全执行（例如 OpenZeppelin Defender、Gelato）。
- **通用代理框架：** 将 Web3 工具作为适配器集成，而非作为一等公民的钱包或安全基础层的大语言模型代理框架（例如 LangChain、AutoGPT）。

编码标准与判定。 表6中的每个能力列均依据明确的编码标准进行编码，以减少歧义并支持可重复性。我们使用 ✓、 ∼和 – 分别表示一等、部分／间接和缺失 支持：

- ✓ 表示有文档记录且可供外部使用的支持，代理无需定制胶合代码即可调用，正常集成所需的代码除外（例如，受支持的策略模块接口、标准的模拟／预览端点或有文档记录的恢复工作流）。
- ∼ 表示间接的、实验性的支持，或需要非标准扩展或大量定制中间件的支持（例如，某项能力确实存在，但只能通过私有 API、未记录文档的模块或特定于应用的封装器实现）。
- – 表示没有明确记录的该能力支持。

对于定性字段（例如托管模型、工具接口、链范围），我们记录主导的设计模式，并在备注列中注明限定条件（例如，多方计算是由用户控制还是由提供商介导，策略是在链上还是链下，以及恢复是通过流程实现还是通过加密方式强制执行）。在编码过程中发现的分歧通过明确的判定步骤予以解决：重新核查所引用的文档，在可用时依据参考实现验证术语，并在可比系统之间一致地应用编码标准。在维护该矩阵这一持续更新的成果时，我们建议报告一个简单的一致性统计量（例如百分比一致率或 κ），以便使偏移和编码标准中的歧义清晰可见。

解读注意事项。 该矩阵有意以能力为导向，而非以评分为导向。A ✓ 并不意味着某项实现默认就是安全的；它表示相关机制确实存在且可以使用。反过来，– 并不意味着无法实现，只表示截至编写时，该能力并非系统公开文档中明确列出的一等功能。因此，备注 列应视为每一行不可或缺的组成部分，尤其是对于策略执行、可观测性和恢复等安全敏感维度。

7.2 能力矩阵

表6汇总了所选系统的比较能力矩阵。我们在后续章节中使用该矩阵来说明：(i) 为什么某些集成模式在生产部署中占据主导地位；(ii) 安全控制集中在流水线的哪些环节；以及 (iii) 哪些接口层差距会在其他方面已趋于成熟的技术栈中反复出现。

indicates partial or indirect

Notes
Production-ready agent toolkit with MPC custody
Embedded wallet with social recovery
Modular smart account with extensive policy support
Enterprise-grade key management with policy controls
Programmable key pairs with condition-based signing
Enterprise MPC with comprehensive policy engine
Standard account abstraction with UserOp intents
Intent-based DEX with MEV protection
Intent-based aggregator with resolver network
Private transaction submission for MEV protection
Transaction simulation and debugging platform
Full-stack AA with session keys and paymasters
Kernel smart account with plugins and session keys
Modular account with gas sponsorship
Decentralized compute for smart contract automation
Web3 automation with relay and functions
Enterprise security with automated operations
Standard protocol for agent-tool communication
LLM agent framework with Web3 tool integrations
Experimental autonomous agent with crypto capabilities

7.3 关键见解与新兴架构趋势

区块链技术的监管格局仍在不断演变，围绕如何将现有法律框架应用于这项新技术的辩论也在持续进行 [35]。这种监管不确定性对智能体区块链系统的设计与部署有着重大影响。

我们对20个代表性系统在13个能力维度上的比较编码（表6），揭示了若干一致的架构趋势。这些趋势体现了2025年的格局，并阐明了生态系统的趋同方向。 [309, 186]

1. **账户抽象已成为代理的默认钱包底层架构**。纵观整个矩阵，面向程序化执行的系统越来越多地采用与账户抽象相契合的智能账户架构，因为这类架构让授权逻辑变得可编程，而不再完全依赖密钥。实际上，ERC-4337 生态系统提供了与代理相关的基础构件，例如会话密钥、模块化验证、基于角色范围的权限、社交恢复和 Gas 赞助，这些功能共同降低了大规模部署受限自动化的操作阻力 [55, 83]。以 AA 为核心的技术栈（例如 Safe 模块、Biconomy、ZeroDev、Alchemy Account Kit）在<定位策略>、<定位预览>和<定位模拟>方面反复呈现的 ✓ 模式，与这一从单体外部拥有账户转向具备策略能力的账户的趋势相符。

2. **执行接口正从命令式交易转向声明式意图**。 第二个趋同点是，系统正稳步从底层交易构造转向以意图为中心的交互：代理指定执行结果（约束、限额、截止时间），并将路径搜索和执行细节委托给专门的基础设施（聚合器、求解器、拍卖） [324, 149, 220]。矩阵显示，全栈平台和安全导向平台（如 Coinbase AgentKit、OpenZeppelin Defender）以及专门的意图协议（如类似 CoW 和 Fusion 的设计）都对意图提供了强有力的支持，这表明意图正成为推理与执行之间的首选边界。这一转变降低了代理端的认知负担，缩小了脆弱且依赖特定协议的交易组装所涉及的范围；同时，通过竞争性求解器流水线路由执行，而不是直接广播交易，也提升了抗 MEV 能力。

3. **策略执行已成为生产级系统的标志性特征**。 矩阵凸显了为严肃资产托管而设计的系统与仍处于实验阶段的系统之间的明显分界：几乎所有成熟技术栈都具备稳健的策略控制，无论是通过智能账户模块在链上执行，还是通过托管与签名基础设施在链下执行 [230, 95]。实际上，策略被视为遏制运行时环境受损、代理目标设定错误以及不安全工具输出的首要防线。相比之下，若框架提供代理自主性，却没有相应的策略控制和审计控制，就很难将其用于重要资产管理，因为这类框架缺乏明确机制来限制故障影响。

4. **托管不可能三角仍未解决，并由此限制了自主性上限**。 该矩阵还凸显了一项长期存在的权衡：(i) 自主权、(ii) 运维安全和 (iii) 高频自主性。多方计算阈值签名和基于可信执行环境的签名器能够有效遏制安全失陷，并提供更完善的策略门控，但也增加了对签名基础设施及其运营者的信任依赖和运行依赖 [169,184, 274]。智能账户托管提升了可组合性，并使控制逻辑更加去中心化，但复杂策略可能因此变得更复杂、链上成本也可能更高。简单的用户持有的密钥能最大限度地保留自主权，但仍然最不适合自主代理，因为一旦失陷，其影响范围通常不受限制。因此，托管方案选择并非实现细节，而是一项会限制可行自主性水平的架构决策。

5. **通用代理框架在原生加密领域的安全原语方面仍然落后**。 尽管通用代理框架可以通过 Web3 工具封装器进行扩展，但该矩阵表明，它们往往缺少链上环境中对防止事故和开展取证至关重要的原生支持功能：类型化意图流程、强化的授权边界、高保真模拟、结构化预览，以及面向恢复的可观测性。这一差距与更广泛的评估结果一致：在对抗性或复杂的工具环境下，工具使用型代理往往难以保持可靠性，因此在高风险部署中，需要采用更具领域针对性的护栏机制和评估框架 [189]。

6. **模拟、预览和可验证回执正逐渐成为基本预期**。 最后，能力分布表明，只要人工审批属于控制回路的一部分，交易模拟和便于人理解的预览就正从“锦上添花”转变为标准做法。这与所见即所签式审批工作流和可信显示设备背后的安全性理念一致：它们旨在确保用户授权与实际签名内容的语义相绑定 [149, 281]。不优先考虑预览和模拟的系统通常运行在较狭窄的自动化模式下（或假定操作员可信），因此不太适合面向广大用户的代理式执行。

综合来看，这些趋势表明，最成熟的设计正逐渐趋向于模块化的纵深防御体系：通过智能账户实现可编程授权，通过以意图为中心的执行降低协议脆弱性和最大可提取价值

暴露，通过策略引擎限制故障，并借助模拟和可观测性支持验证与恢复。这些趋同趋势也揭示了仍待解决的研究和标准化缺口；接下来，我们将探讨接口标准和缺失的互操作层 [292]。

7.4 关键系统与协议

为补充表6中的高层概览，我们现在从架构层面考察每个选定的系统。我们不再重复功能勾选项，而是分析影响安全性、自主性和运营风险的设计选择：(i) 权限如何表示（密钥、会话密钥、智能账户角色）；(ii) 约束条件在哪里执行（链上验证模块还是链下策略引擎）；(iii) 执行如何受到保护（模拟、预览、具备 MEV 感知的提交）；以及 (iv) 存在哪些可观测性和恢复钩子。这一分析框架十分有用，因为生态系统中的许多故障并非由缺少基础构件导致，而是源于代理的自主性与系统的执行和验证层之间存在不匹配。

7.4.1 全栈代理工具包与基础设施

Coinbase AgentKit（2024年）

历史背景与产品意图。Coinbase AgentKit 定位为面向开发者的工具包，通过 Coinbase 开发者平台（CDP）的基础构件和集成，将代理工作流连接到链上操作。公开资料简明扼要地描述了其核心设计目标：为代理提供以钱包为中心的接口，并通过打包的软件开发工具包和扩展，在将常见链上操作集成到代理运行时时减少定制胶合代码 [74,75]。

架构深入解析。 从智能体与区块链集成的角度来看，AgentKit 最适合理解为一个全栈集成界面 ，它将 (i) 代理端操作 API 与 (ii) 能够表达并执行操作约束的执行后端打包在一起。

- **权限与托管界面**。 实际上，AgentKit 符合模式 III/IV 的部署风格，将智能体推理与签名权限分离。面向开发者的接口旨在让智能体请求链上操作，同时在模型循环之外执行托管和授权控制。这种分离符合生产环境智能体到链系统的标准指导原则：代理运行时不应直接持有高价值的长期密钥，签名也应通过执行边界（策略、托管服务或受保护的签名环境）进行中介，而不是让原始密钥材料驻留在进程中。

- **操作接口与工具语义**。 全栈工具包的一项关键架构贡献，并非提供操作（兑换、转账、部署等），而是围绕这些操作建立类型化、可重复的语义：必需输入、预期输出、错误界面，以及与智能体工具路由器集成的途径。AgentKit 打包的软件开发工具包及其公开的扩展途径（例如 OpenAI Agents SDK 扩展）体现了这一方向：它们标准化了智能体调用链上能力并接收结构化结果的方式，以便后续推理和日志记录 [75]。

- **执行路径与安全钩子**。在生产环境中，代理工具包的安全价值取决于其对以下方面的支持：(i) 执行前检查（模拟/估算和不变量检查）、(ii) 策略门控（支出限额、允许列表、角色约束），以及 (iii) 可审计性（交易意图、策略决策记录、执行回执）。AgentKit 体现了能力矩阵所反映的更广泛趋势：领先的工具包将策略和可观测性视为一等功能，而非可有可无的附加项。

安全模型与信任假设。AgentKit 类工具包的安全态势最适合描述为托管式组合：开发者依托精选集成层，获得更高的速度和一致性，同时也必须信任工具包抽象边界的正确性，以及其连接的托管和策略组件。主要残余风险与第5节中的威胁分类相呼应：(i) 如果操作层依赖已遭安全失陷的上游数据，则可能出现工具平面完整性问题和端点欺骗；(ii) 如果授权未以密码学方式绑定到已签名载荷，则可能在预览与签名之间发生中间件篡改；(iii) 为加快自动化而授予广泛权限，可能导致策略配置错误。

生态系统影响。 AgentKit 的突出之处，主要在于它将代理到区块链开发工作流打包成可重复使用、配有文档支持且非专业开发者也能轻松上手的形式。在 2025 年的行业格局中，这一点意义重大：它加速了实验与生产化进程，同时也提高了开发者视为基本要求的默认安全预期门槛（策略、日志记录和结构化操作接口）[74]。

Privy 嵌入式钱包（2024年）

历史背景与愿景： Privy 在 2023—2024 年间崭露头角，成为嵌入式钱包的领先提供商。其背后的驱动力是一个显而易见的普及障碍：主流用户通常不愿安装专用钱包软件、管理助记词，也不愿费心理解 Gas 和特定区块链的用户体验。嵌入式钱包技术栈旨在让签名和密钥恢复体验更接近人们熟悉的 Web2 身份验证流程（电子邮件、社交登录、设备生物识别），同时仍在账户层提供密码学控制。这种设计方向对于面向消费者的代理式应用尤其重要：代理必须能够以编程方式发起操作，但用户必须保有清晰易懂且可执行的授权与恢复路径控制权。 [16, 149]。

架构深入解析：

- **托管模型（阈值密钥控制）**。一种常见的嵌入式钱包模式是使用阈值构造，避免在任何单一设备上保留单个长期有效的私钥：密钥材料会拆分为多个份额（通常分布在用户设备、服务端组件和恢复渠道），签名则需要达到法定人数。这提高了抵御设备丢失的韧性，并降低了单一环境安全失陷后立即获得单方面签名权限的可能性。具体机制因提供商而异，但底层安全目标一致：在确保终端用户能够顺利恢复的同时，降低单点失陷风险。[314, 169]。

- **代理集成与授权易用性**。从代理系统的角度看，嵌入式钱包最适合**模式 II**（模拟与意图生成）：代理可以拟定意图或交易请求，再由钱包层提供范围受限且可由用户核验的授权步骤（通常通过设备绑定身份验证来实现）。在自动化程度更高的部署中，嵌入式钱包还可以成为**模式 III**的入门途径：用户创建智能账户，并向代理委托范围严格受限的能力（例如时间受限的会话密钥或角色），而非授予持久且不受限制的签名权限。这符合分类体系中始终强调的最小权限原则。 [83, 149]

安全模型： 其安全态势最恰当的描述是风险转移且影响范围有限。嵌入式钱包并非将所有安全责任都集中于助记词托管，而是将风险分散到设备安全、身份验证渠道、服务端控制和恢复机制等方面。这提升了可用性和实际恢复能力，但也引入了新的依赖项（身份恢复、服务可用性以及阈值签名执行的正确性）。因此，严谨的部署会将身份与恢复界面视为一等安全边界，并辅以审慎的交易预览、明确的策略约束，以及用于异常检测的强大遥测能力。 [149]。

生态系统影响： 嵌入式钱包降低了面向消费者的去中心化应用的使用门槛，因此能够为商业、游戏和社交应用等领域的代理式用户体验提供支持。其战略意义与其说在于服务去中心化金融领域的资深用户，不如说在于让代理介导的操作能够在用户注意力有限且必须切实可行地进行操作恢复的环境中得以实现。 [149]。

7.4.2 密钥管理与托管基础设施

Safe{Wallet} +模块（2023年）

历史背景与愿景： Safe{Wallet}（前身为 Gnosis Safe）是以太坊生态系统中的基础智能账户原语，广泛用于金库托管和多方控制。其核心贡献在于，以链上编码的阈值授权取代单密钥权限，使控制策略透明，并可由基础执行层强制执行。这种安全优先的设计使 Safe{Wallet} 成为智能体授权架构的天然基础，在此类架构中，目标不仅是签名，还要在约束条件下 签名，并对这些约束实施可审计治理。 [149]。

架构深入解析：

- **核心授权逻辑（阈值控制）**。 从本质上看，Safe 是一种由所有者集合和签名阈值治理的智能合约账户。只有满足阈值时，交易才会执行，从而有效防范单密钥泄露，并通过所有者构成和签名策略实现明确的职责分离（例如提议者与批准者角色）。 [149]。

- **通过模块和守卫实现模块化（可编程约束）**。代理式控制通常通过 Safe 的可扩展性实现：可以授予外部模块和守卫式组件受控的执行权限，也可以由它们对交易实施前置条件和后置条件。这种模块化正是实现**模式 III**

的关键：代理可以获准在严格限定的范围内操作（函数／目标允许列表、支出限额、速率限制、时间窗口），而信任级别较高的操作仍需在人类或多方审批通过 Safe 阈值层后才能执行。由此形成的架构支持纵深防御：即使代理的链下推理失效，确定性的链上检查仍能限制安全失陷的代理可执行的操作。 [310, 149]

安全模型：该安全模型是链上可执行性与组合风险。核心 Safe 合约在规则透明的前提下提供了强有力的保障；不过，可扩展性也使部分安全责任转移到了已安装模块/守卫的正确性，以及所有者集合的运维规范上。实际上，主要故障模式很少是阈值概念本身，而通常是错误配置，或辅助组件引入的漏洞。因此，应谨慎选择模块、进行严格审计，并持续监控模块行为，尤其是在使用模块赋予智能体自主或半自主执行权限时。 [310]。

生态系统影响：Safe{Wallet} 为高价值资产托管提供了广受认可且可组合的基础方案，已成为需要明确策略执行和可审计治理的代理部署的实用支柱。其架构理念与智能体到链系统的新兴最佳实践高度契合：在链上约束权限，默认将特权降至最低，并将可扩展性视为安全关键面，而非便利层。 [149, 310]。

Privy 嵌入式钱包（2024年）

历史背景与愿景。 Privy 在 2023—2024年发展周期中崛起，成为嵌入式钱包领域的领先服务提供商：它提供基础设施，旨在让钱包创建、身份验证和交易审批在面向消费者的应用中显得自然流畅。其基本论点是，限制主流采用的因素，与其说是协议能力，不如说是用户体验摩擦和密钥管理焦虑。因此，嵌入式钱包优先采用用户熟悉的登录流程（电子邮件/社交登录）、通行密钥和设备绑定身份验证，以及可恢复性，同时仍力求避免完全托管型的信任假设。

架构深入解析。 理解 Privy 的设计，最好将其视为一种混合密钥管理技术栈，配有面向应用的 SDK，并采取恢复优先策略。

- **托管模型（混合门限 + 恢复）。** 一种常见的嵌入式钱包模式是结合门限签名与可恢复密钥份额，从而做到：(i) 没有任何单一组件能够单方面签名；(ii) 设备丢失并不意味着资产损失。一种典型实现是将用户密钥拆分为三个份额，并要求达到 2/3 的阈值才能授权签名；这三个份额分别是设备绑定份额、服务端份额，以及可通过带外机制访问的恢复份额。这种方法与关于可恢复密钥管理和分布式授权的秘密共享领域更广泛的研究相一致，同时着重于用户体验驱动的恢复工作流。[16, 169]。

- **授权与代理集成（模式 II–III）。**对于代理式应用，Privy 主要支持：(i) **模式 II（模拟/意图生成）**，允许代理式应用提出一项操作，并通过用户确认提示（例如，在受信任设备上进行生物识别确认）将最终授权交由用户完成；以及 (ii) **模式 III（委托执行）**，与智能账户（例如 ERC-4337 账户）配合使用时，可签发限定范围的会话密钥，并在链上强制执行支出限额和允许列表。嵌入式钱包由此成为用户友好型桥梁，用于一次性设置和策略安装，之后代理则在受限权限下运行。 [55, 308]。

- **运营攻击面。**嵌入式钱包扩大了集成面：身份恢复渠道、设备绑定、SDK供应链和后端策略配置都成为首要安全考量。架构目标并非消除信任，而是将其落实到可审计且可监控的控制措施中，并根据应用的风险承受能力进行调整。

安全模型。 Privy式嵌入式钱包通常会将主要风险从助记词窃取转向身份和恢复机制泄露 （电子邮件/社交登录账户接管、SIM卡交换攻击或恢复链接拦截），以及*SDK* 和后端完整性 风险。作为代理启用机制使用时，建议采取以下安全措施：(i) 将高价值权限置于时间受限、限定范围的会话密钥之后；(ii) 在智能账户层强制实施严格的策略控制；(iii) 对策略变更和恢复操作要求用户进行强身份重新验证。 [308, 169]。

生态系统影响。 嵌入式钱包为面向消费者的代理式体验提供了一条切实可行的入门路径，使应用能够在保留熟悉的批准模型的同时，安全地提出操作。它们对代理系统最重要的贡献并非自主性，而是可恢复、低摩擦的授权，有助于推动更受约束的链上委托。

7.4.3 密钥管理与托管基础设施

Safe{Wallet} +模块（2023年）

历史背景与愿景。 Safe{Wallet}（前身为 Gnosis Safe）是高价值链上控制领域的典型智能账户底层。其核心模型——基于智能合约账户的阈值授权——早于 ERC-4337 出现，却充分预示了智能账户的发展方向：可组合授权逻辑、模块化扩展以及明确的策略执行。Safe 从多重签名钱包发展为模块化账户平台的历程，映照出更广泛的趋势，即迈向可编程钱包与受约束的委托。 [55, 308]。
架构深入解析。

- **核心授权原语（阈值控制）。** Safe 的基础机制是在一组所有者之上实施阈值控制，要求获得 t-of-n 个授权后才能执行一笔交易。这实现了链上职责分离，并降低了单密钥泄露风险。

- **模块与守卫（通过组合实现策略）。** Safe 与智能体的相关性通过其模块模式得以实现：可以授予外部合约代表 Safe 执行交易的权利，并可选择由守卫逻辑进行中介处理，以检查并否决行动。从智能体的角度来看，模块可以为模式 III 实现委托合约：允许名单中的目标、函数选择器、支出限额、时间窗口和速率限制均在链上以确定性方式强制执行。这种方式在保持执行过程透明且可审计的同时，也符合账户抽象的目标。 [55, 308]。
- **故障模式（模块正确性占主导地位）。** 主要风险从 Safe 的核心（通常经过严格审计）转移到自定义模块接口：权限过宽的模块、配置错误的允许列表，或意料之外的调用路径，都可能重新引入广泛权限。这与相关研究的结论一致：如果策略框架和封装器未经过审慎设计与验证，就可能成为最薄弱的一环。 [310]

安全模型。 Safe 提供最大程度的链上可执行性：授权和约束由不可变执行语义强制执行，因此尤其适用于不信任链下中间件的高风险委托场景。因此，其安全态势可概括为“安全性取决于已安装模块及其配置”，这使标准化且经过审计的策略模块成为代理部署的实际必需品。 [310]。

生态系统影响。 Safe 是财库和去中心化自治组织账户层的默认选择，也是智能体系统的重要构建模块，适用于需要透明、可组合且可撤销委托的场景。它是对模式 III 进行“策略优先”解读的基石，并经常作为 ERC-4337 兼容技术栈下的账户层。 [55, 308]。

Turnkey（2024年）

历史背景与愿景。 **[301]** Turnkey 将自身定位为面向希望获得企业级签名控制、又无需运营托管基础设施的团队的“密钥管理即服务”。该产品定位面向需要安全、可编程签名，并重视可审计性与运维保障的构建者，同时让私钥材料与应用程序运行时保持隔离。

架构深入解析。

- **基于可信执行环境的签名隔离。** Turnkey 风格的架构使用可信执行环境（TEE）来隔离密钥生成与签名。在该模型中，密钥材料存放于硬件隔离飞地内，外部应用程序只能通过经身份验证的 API 请求签名。这自然对应模式 IV：代理需要自主执行，但绝不能持有原始密钥。关于可信执行环境安全性与飞地设计的综述强调，可信执行环境是强大的隔离原语，但其保障取决于威胁模型、飞地代码的完整性以及证明流程。 [68, 168]。
- **作为签名关卡的策略（链下控制平面）。** 基于可信执行环境的签名器的一项关键特性，是能够在签名时刻强制执行丰富的策略：模板约束、目标地址允许列表、金额上限、速率限制以及基于环境的条件。这对代理系统在运维层面颇具吸引力，因为策略的更新速度可以快于链上治理，同时仍能在权限边界（签名器）处得到执行。

- **可观测性与事件响应。** 由于签名器处于最终授权瓶颈，因此这里是生成不可篡改审计日志的理想位置，可记录签名请求、策略决策和被拒绝的交易。在实践中，这会成为模式 IV 系统“紧急停止开关”方案的核心组成部分。

安全模型。 安全模型是硬件强制隔离加策略门控签名。主要的信任假设是：(i) 对相关攻击者而言，可信执行环境的隔离和证明属性均成立；以及(ii)策略

表 7：对比代理部署的托管与授权底层架构。

系统	主要信任锚点	策略位置	最适用模式
Privy（嵌入式）	设备 + 身份 +阈值恢复 [16]	应用 + 可选智能账户模块 [308]	II、III（引导）
Safe + 模块	链上执行语义 [308]	链上模块/守卫 [310]	III（强），V（财库）
Turnkey（可信执行环境）	飞地隔离 +远程证明 [68]	签名方策略门控	IV（强）
Lit（阈值网络）	阈值诚实性 + 动机 [169]	网络侧策略即代码	IV、V（分布式）

引擎得到正确实现和配置。与多方计算相比，这种方案的权衡在于：可信执行环境将信任集中在硬件与服务提供商栈中，而多方计算则将信任分散给多个密钥份额持有者。 [68, 169]。

生态系统影响。 类似 Turnkey 的签名器通过将原始密钥处理从代理运行时中移除，并使策略执行成为签名操作的核心组成部分，为构建生产级自主智能体提供了一条切实可行的路径。

Lit Protocol（2024年）

历史背景与愿景。 **[84]** Lit Protocol 推进了一种加密原生理念：机密管理和签名应由去中心化阈值网络 提供，而不是由中心化托管方或单一硬件飞地提供。其目标是构建一个与区块链无关且抗审查的可编程授权层，从而实现基于条件的签名和自动化。

架构深入解析。

- **用于签名和解密的阈值网络**。 Lit 采用门限密码学，并在节点网络中进行多方计算式协调，因此没有任何单一节点持有完整的私钥。签名（或解密）需要达到阈值数量的节点参与，这与现代分布式授权和韧性领域的阈值密码学／多方计算文献相一致。 [169, 314]。
- **可编程授权（策略即代码）**。 一个核心差异化特点是将条件嵌入授权逻辑中（例如，仅在链上谓词成立时才签名）。对于代理系统而言，这为模式 IV 提供了另一种实现方式：代理请求执行一项操作，但阈值网络会在释放签名前强制执行策略条件。
- **协调与活性方面的考量**。 去中心化签名引入了新的系统约束：节点可用性、法定人数形成以及经济激励。尽管信任模型变得更加去中心化，运营模型却更为复杂，也可能容易受到相关故障或激励失调的影响。

安全模型。 Lit 的安全态势建立在分布式信任与加密经济学假设之上：安全性取决于对手无法控制达到阈值的节点，也取决于激励机制能够遏制串通。这有别于链上模块（Safe）所具备的确定性可执行性，也不同于硬件根信任保障（可信执行环境）。 [169, 314]。

生态系统影响。 Lit 为可编程签名和基于条件的自动化提供了一种去中心化替代方案，对于跨链或审查敏感型智能体工作流尤其具有吸引力。它是构建高级模式 IV 系统以及有意分散权限的模式 V 工作流的可靠构建模块。 [169, 314]。

7.5 关键系统与协议

为补充能力矩阵提供的高层概览，本小节将从以下角度分析每个入选系统：**(i)** 历史背景与产品意图，**(ii)** 核心架构选择（接口、托管和控制平面设计），以及 **(iii)** 安全假设与生态系统层面的影响。我们的目标并非重述功能核对项，而是揭示在对抗性和高风险条件下部署智能体—区块链系统时至关重要的设计权衡。

7.5.1 全栈代理工具包与基础设施

Coinbase AgentKit（2024年）

历史背景与愿景。 Coinbase AgentKit 体现了一种更广泛的转变：大型基础设施提供商正在打包提供端到端的“代理就绪型”技术栈，而不是让开发者自行拼凑托管、策略、模拟和执行路由等分散组件。这一战略动机契合代理式执行的运营现实：可靠的密钥管理、可强制执行的控制措施和生产环境可观测性，与模型质量同样重要。

架构深入解析。 从系统层面来看，AgentKit 式技术栈通常结合了：(i) 托管式签名后端，(ii) 用于限制可签名内容的策略层，以及 (iii) 紧密集成的执行路径，以降低提交环节的脆弱性。

- **托管与授权**。 一种常见模式是双层模型：高价值权限由更严格的托管控制措施保护，而日常操作则使用具有时间限制和限定范围的凭证（例如，会话式委托），这与最小权限设计以及第4节中的委托执行模式相一致。
- **策略控制平面**。 生产环境中的技术栈往往在链下执行策略（以提高灵活性），但会将执行绑定到签名流程，从而防止通过其他交易构造路由绕过策略违规检查。
- **执行环境**。 可靠性和具备 MEV 感知能力的路由通常被视为平台功能范围的一部分，而不是开发者事后才考虑的问题，因为提交路径默认具有对抗性 [80]。

安全模型。 安全态势采用纵深防御：托管隔离加上策略门控和可观测性。主要的信任假设包括：（i）签名服务以及策略执行路径的正确性；（ii）供智能体使用的精选工具与端点集合的完整性。

生态系统影响。 全栈工具包能够减少集成开销，并缩短对于不具备深厚原生加密领域经验的团队而言的部署时间。其代价在于，平台层面的设计决策（托管服务提供商、策略语义和默认路由设置）会切实影响代理的风险范围。

Privy 嵌入式钱包（2024年）

历史背景与愿景。嵌入式钱包旨在减少用户注册流程中的阻力，其方式是以熟悉的身份验证模式和应用集成式签名取代以助记词为先的用户体验。对于代理式应用而言，相关贡献并非“智能体”本身，而是提供可编程钱包接口，支持意图审查、策略门控和可恢复性，同时无需用户操作独立的钱包界面。

架构深入解析。

- **托管与恢复原语**。嵌入式钱包设计通常会结合门限技术（以提升可用性和抗攻陷能力）与恢复工作流，相较于单纯的助记词托管，这些工作流更符合消费者的预期。这一总体设计方向与现代密钥基础设施中采用的实用秘密共享和门限方法相一致 [184, 169]。
- **代理集成路径**。在实践中，嵌入式钱包与模式 II（意图与预览，并由用户批准）十分契合；当与智能账户和会话式委托结合时，还可过渡到模式 III。其架构要点在于，嵌入式钱包成为受控的授权界面，而不仅仅是密钥存储。

安全模型。 安全权衡将身份与设备安全提升为首要考量（账户接管、恢复渠道风险和设备攻陷）。对于智能体，关键要求是在超出受限委托范围操作时，确保签名仍受强有力的用户或策略检查约束。

生态系统影响。 嵌入式钱包基础设施降低了面向消费者的代理式应用的门槛，但同时也凸显了以下两点的重要性：(i) 精心设计的授权预览（所见即所签）；(ii) 应用逐渐转向后台自动化时，采用范围严格受限的委托。

7.5.2 密钥管理与托管基础设施

Safe{Wallet} +模块（2023年）

历史背景与愿景。 Safe{Wallet} 将智能合约账户确立为主流的托管与治理基本组件，并以多重签名控制作为去中心化自治组织和机构资金库的基准安全态势。其向模块化授权的演进与智能体直接相关，因为这支持实施可执行的链上约束机制。

架构深入解析。

- **核心账户模型**。Safe 的基础机制是由链上逻辑强制执行的阈值授权，这使审批界面可供审计，也难以仅通过链下攻破来绕过。

- **模块生态系统**。模块化模式（例如支持受限角色和策略模块的框架）是模式 III 的主要实现方式。模块可以实现函数和交易对手方允许列表、支出限额及限时权限，并确保这些规则在验证时得到强制执行 [171]。

安全模型。 该模型强调透明、确定性的执行。主要剩余风险在于模块层：系统的正确性和审计质量取决于已安装模块中最不稳健的那个，这与更广泛的研究发现相呼应：如果策略和防护合约未经过谨慎设计与验证，它们可能成为主要攻击面 [310]。

生态系统影响。 Safe 及其模块模式已成为代理受限执行的参考设计：它们为“受限自动化”提供了可信、开放的基础，而这通常是在完全自主的托管模型之前更具实际意义的采用路径。

Turnkey（2024年）

历史背景与愿景。 Turnkey 风格的“密钥管理即服务”系统旨在解决代理部署中的一个反复出现的约束：团队希望在具备强隔离特性的同时实现程序化签名，但不能假定代理运行时是存放长期密钥材料的安全位置。

架构深入解析。

- **基于 TEE 的隔离**。可信执行环境（TEE）将密钥操作与主机环境隔离，并确保签名受飞地强制执行逻辑约束。其预期优势在于，即使主机遭到安全失陷，也不会轻易导致原始密钥被窃取，前提是 TEE 及其证明模型可靠 [227]。

- **策略与签名绑定**。一项切实可行的设计目标，是使策略检查与签名生成不可分割：策略引擎评估请求，只有在满足约束条件时，隔离签名器才会生成签名。这符合模式 IV 的要求：既需要灵活性，又应通过强化的控制平面对其加以约束。

安全模型。安全态势依赖于隔离、远程证明以及正确的飞地代码。与阈值多方计算相比，可信执行环境将信任集中于硬件和飞地实现；与链上策略相比，则是以公开可审计性换取更丰富的链下执行。工程上必须具备强大的可观测性和事件响应钩子（撤销、轮换和紧急停止开关），因为自主性越高，延迟检测的代价也越大。

生态系统影响。 基于 TEE 的签名服务为实现自主执行提供了一条可行途径，同时无需将敏感密钥材料置于代理进程中。因此，对于需要运营敏捷性的模式 IV 部署而言，这是一项重要的构建模块。

Lit Protocol（2024年）

历史背景与愿景。 Lit Protocol 代表了一种原生加密领域的可编程密钥使用方法：它不依赖单一托管方或安全区，而是致力于通过去中心化节点集实现阈值控制签名和基于条件的授权。

架构深入解析。

- **门限签名与去中心化**。 该系统依托门限密码学，确保没有任何单个节点持有完整的密钥材料，且签名需要法定人数参与。这降低了单点失陷风险，也契合分布式托管系统中使用门限原语的更广泛动机 [184, 169]。

- **可编程授权**。 其独特之处在于能够将签名与可编程条件绑定，从而实现基于条件的自动化和协调模式；若结合稳健的治理与监控，这些模式便可支持更先进的模式 IV 和模式 V 部署。

安全模型。 信任假设转而取决于参与节点集合的去中心化程度和诚实阈值，并由加密经济激励提供支持。与任何法定人数模型一样，相关性失陷和串通会成为关键故障模式，因此部署时应将门限签名与质押、监控及治理保障措施相结合，并与第 5.2 节中的威胁分类体系保持一致。

生态系统影响。 去中心化密钥网络将自动化扩展到单一服务提供商托管之外。要在高风险智能体场景中切实采用这类网络，取决于成熟的运维控制措施，以及对节点治理和故障处理的高度透明。

Fireblocks（2023年）

历史背景与愿景。 Fireblocks 广为人知的是基于多方计算的签名，以及以此为基础构建的机构托管和资金管理运营。在这一分类体系中，它的重要意义在于将阈值托管与面向工作流的策略执行相结合，这与高保证自动化模式自然契合 [107]。

架构深入解析。

- **多方计算托管。**多方计算签名将密钥材料分散至多个参与方，从而确保单主机失陷不会导致单方签名能力。这与成熟的多方计算方法一致，兼顾实际签名性能与安全性 [184, 169]。
- **策略与工作流门控。**机构平台强调基于规则的授权、多步骤审批和职责分离，这与模式 IV 部署相符；在此类部署中，合规与可控性是必要条件，而非可选功能。

安全模型。其安全态势采用纵深防御，结合程序性控制与密码学托管。机构分析和平台文档强调，自动化应与可强制执行的策略及可审计性相结合，因为主要故障模式不仅是密钥泄露，还包括对抗性条件下的不安全授权 [96]。

生态系统影响。机构托管平台为受监管实体试行代理式执行提供了采用路径，因为这些平台在提供熟悉的治理机制（策略、授权、审计）的同时，也增加了自动化所需的程序化 API。

7.5.3 账户抽象与意图协议

ERC-4337 打包器（2023年）

历史背景与愿景。 ERC-4337 通过引入结构化的伪交易接口（用户操作）和以验证为中心的执行枢纽（入口点），在无需进行共识层变更的情况下推进账户抽象 [55]。打包器是将用户操作对象传输至链上执行的去中心化服务层，从而支持可编程钱包和面向代理的授权方案。

架构深入解析。

- **UserOperation 内存池与打包。** 智能体（或应用）将用户操作对象提交至专用内存池。打包器聚合这些操作，并将其以交易形式提交至入口点；入口点随后根据智能账户的逻辑验证并执行每项操作 [55]。
- **验证作为安全边界。** 对于智能体而言，关键在于授权可以在验证时实现可编程化。这包括会话式委托、策略模块，以及由账户验证路径强制执行的重放约束，从而与委托执行设计保持一致 [83, 285]。
- **动机与市场动态。** 打包器在竞争激烈的交易纳入市场中运作：它们预先支付 Gas，并通过账户逻辑或支付主获得补偿。这有助于构建一个无需许可的生态系统，但也带来了运营方面的考量（审查压力、中继依赖以及感知 MEV 的纳入策略），智能体设计者必须将这些因素视为执行威胁面的一部分 [265, 80]。

安全模型。 安全模型以确定性的链上验证为基础：即使打包器存在故障或具有对抗性，是否执行仍由入口点和账户验证逻辑决定。残余风险主要集中在：（i）支付主和验证模块的正确性；（ii）对抗性排序和审查条件下的提交路径。

生态系统影响。 打包器使账户抽象得以实际运行。对于智能体系统而言，它们是基础设施的核心组成部分，因为它们支持可编程授权、灵活的燃料费模型，以及支持受策略约束自主性的标准化执行路径。

CoW Protocol（2024年）

历史背景与愿景： CoW Protocol（“需求巧合”）代表了应对最大可提取价值问题的更广泛响应：随着公共订单流执行变得日益具有对抗性，生态系统从命令式、暴露于内存池的交易，转向了以下机制：(i) 降低交易可预测性；(ii) 将执行竞争内部化；(iii) 通过签名承诺保留用户约束。这一动机与关于公共网络中交易重排序和价值提取的基础性分析相一致，这些分析表明，天真的广播并执行策略可能会系统性地受到对抗性搜索者的惩罚 [80, 18]。

架构深入解析： 从高层次来看，CoW 风格的设计将“提交一次去中心化交易所调用”替换为“签署一项执行意图”。用户（或代理）会签署一份结构化订单，其中编码了约束条件（资产对、限价/最小输出量、有效期限以及可选的执行偏好）。随后，协议利用链下协调层，由相互竞争的执行者（求解器）提出满足这些约束的结算方案。

- **意图层**。 主要接口是经过签名、携带约束的消息，而非完整指定的交易。这将复杂性从代理的行动空间中移出，并使代理的承诺可供审计，且受明确限额约束；这与近期以意图为中心的执行研究中讨论的通用基于意图的设计模式一致 [220]。
- **求解器竞争与结算**。 竞争中的求解器会寻找一种执行方案，既能 (a) 满足每一项已签名的约束，又能 (b) 提高执行质量（价格、费用和实际滑点）。一种常见机制是基于批次的结算，即将多个意图一并撮合；当存在互补意图时，这种方式可以实现直接匹配，仅在需要时才路由至链上流动性。批量拍卖作为缓解对抗性排序效应的手段已得到广泛研究；通过降低单笔交易被单独识别和预测的可能性，它们可以减少某些最大可提取价值攻击向量 [33, 18]。

安全模型： 核心安全属性是约束保持：求解器无法以低于已签名意图所允许的汇率合法地为用户结算（例如，低于最小输出量或限价约束），因为违反约束会使结算无效。对最大可提取价值的抵御主要依赖架构层面的 设计，而非绝对防护：批处理和非内存池暴露可以降低经典三明治攻击模式的可行性和盈利能力，但残余风险依然存在（例如，求解器一方的信息优势、逆向选择，以及通过辅助渠道发生的部分信息泄露）。这些权衡反映了更广泛的最大可提取价值文献中的观点：缓解措施改变的是动机与可观测性，而非彻底消除对抗性行为 [80, 18]。

生态系统影响： 对于智能体而言，CoW 式执行十分重要，因为它清晰地将决策 （要实现什么结果）与执行优化（如何在对抗性条件下实现该结果）区分开来。这既减轻了智能体的认知负担，也缩小了与底层交易构造相关的攻击面，同时为针对已签名意图对象的策略检查（限额、截止时间、获准交易场所）提供了天然基础 [220]。

1inch Fusion (2024年)

历史背景与愿景： 去中心化交易所聚合解决了跨交易场所的价格发现问题，但其本身并未解决执行对抗性问题。随着最大可提取价值成为一项首要成本，聚合器生态系统日益采用意图式执行模式，让用户（或代理）表达约束条件，并将执行搜索委托给专门的执行者，通常借助类拍卖机制来协调各方动机。这一方向与更广泛的趋势一致，即转向基于意图的系统和拍卖调解的执行，以切实应对公共订单流的脆弱性 [220, 18]。

架构深入解析： Fusion 类设计可理解为一种聚合器集成的意图市场：

- **带约束意图**。用户（或代理）对意图进行签名，其中明确指定交易及其约束条件（最低可接受输出/限价、有效期和其他执行界限）。签名将用户的授权绑定到这些约束条件上，这对于代理式使用而言是一项关键的安全原语，因为它为概率规划器提供了确定性的边界 [220]。
- **执行者之间的拍卖与竞争**。一组执行者（通常称为解析者）在已签名的约束条件下竞争执行该意图，通常通过类拍卖过程进行：执行者在短时间范围内逐步提出更优条件，表明愿意执行。以拍卖为中介的竞争是降低执行成本和缓解对抗性重排序的常见设计模式，已在频繁批量拍卖及相关机制的背景下得到研究 [33]。
- **委托执行**。获胜的执行者负责构造并提交实现该意图的链上结算交易（可为多笔），并可能经由多个流动性来源进行路由。此类委托降低了代理编写底层调用的需求，也可以与对已签名意图实施的严格策略检查相结合（交易场所允许列表、最大名义金额、最大滑点、截止时间约束）。

安全模型： 核心保障依然是约束保持：如果执行者在不使交易路径失效的情况下超出已签名的界限进行结算，则无法完成结算。MEV 缓解得益于内存池暴露的减少和竞争性执行，这可以降低夹子攻击易感性并改善实际成交价，但这种保护并非普遍适用。MEV 文献强调，只要执行仍具有经济价值且部分可观测，对手就能适应；因此，稳健的设计会结合竞争、有界意图和执行渠道选择（私有中继/拍卖），而不是依赖任何单一手段 [80, 18]。

生态系统影响： 对于智能体系统，Fusion 风格的执行强化了能力矩阵所强调的“意图优先”架构趋势：代理可以在执行结果和约束这一层面运行，而竞争性市场则负责路由、Gas/排序策略以及对抗性执行动态。这提高了与策略引擎的可组合性（因为待审查的对象是类型化签名意图），并降低了协议专属交易构造的脆弱性 [220]。

Flashbots Protect（2023年）

历史背景与愿景。 Flashbots 于2020年起作为一项研究与工程工作而兴起，旨在减少最大可提取价值对以太坊造成的负面外部性，包括掠夺性订单流行为和交易排序不透明等问题。其工作有助于将这一问题正式化并量化其规模，促使人们转向在高风险场景下使用私有、可问责的执行通道 [72, 33]。Flashbots Protect 将这一愿景转化为面向开发者和用户的基础工具：一个公共 RPC 端点，通过私有中继基础设施路由交易，以减少用户受到基于内存池的攻击的风险。

架构深入解析。

- **私有提交路径。** Protect 提供一个 RPC 端点，接受标准的已签名交易，但会将其转发至构建者侧基础设施，而不会将其广播到公共内存池。其核心理念很简单：降低敏感操作的交易前透明度，使第三方搜索者无法轻易观察、模拟并以对抗性方式重新排序用户交易。
- **执行语义与威胁降低。** 由于交易在纳入区块之前不会公开传播，直接的抢先交易和三明治攻击机会便会减少。这会将主要威胁从开放、无需许可的对手集合（任何内存池监视者）转移到范围更窄的一组特权方（更早看到交易的中继器/构建者）；这是一种有意作出的安全权衡，旨在为许多工作流提供更强的实际保护 [80]。
- **适配代理流水线。** 在代理操作流水线中，Protect 最适合被视为位于执行通道控制与链上纳入之间边界处的一种控制手段。它通过降低正确且符合策略的交易在提交期间仍遭受经济剥削的概率，补充模拟和策略门控等上游保障措施。

安全模型。其安全模型属于**信任转移型隐私。**Protect 可降低公开可见性，从而缓解常见的内存池攻击，但这也意味着必须依赖中继和构建者不会滥用提前访问权（例如构建者侧提取）。这并非完全无需信任的保证，而是一种务实的防御措施；当基准替代方案意味着必然暴露于公共内存池中的对抗性订单流策略时，这种做法通常是合理的 [72]。

生态系统影响。Flashbots Protect 已成为广泛采用的最大可提取价值感知型执行构建模块。对于智能体系统而言，它是一种实施门槛低、杠杆效应高的控制手段；在适当情况下，将其与严格的滑点界限、截止时间和基于意图的执行路径结合使用，可以显著减少执行期间的损失 [135]。

Tenderly（2024年）

历史背景与愿景： Tenderly 于 2018年 创立，旨在解决早期区块链开发中的一个根本局限：缺乏对智能合约执行的深入可观测性。Etherscan 等区块浏览器虽然能够事后查看交易，却几乎无法帮助用户理解执行路径、状态转换或失败原因。Tenderly 的愿景是为以太坊虚拟机提供一个全栈可观测性与调试平台，类似于 Web2 中的应用性能监控系统，使开发者能够在部署前后分析智能合约行为。

架构深入解析：

- **核心架构：** Tenderly 是一个垂直整合的开发者平台，提供执行跟踪、状态检查和模拟服务。其架构将兼容以太坊虚拟机的执行引擎与经过索引的链上状态及历史区块数据相结合，从而支持对交易进行确定性重放和前向模拟。

1. **执行调试器：** Tenderly 提供细粒度的执行轨迹，揭示操作码级行为、存储读写、内部调用和回滚原因。这项能力与此前关于智能合约执行跟踪和符号调试的研究相契合。

2. **Gas 与性能分析：**该平台提供工具，可将 Gas 成本归因到特定执行路径和指令，从而支持性能优化，并检测异常执行模式。

3. **实时监控与告警：** Tenderly 支持开发者针对合约状态、函数调用和事件发出设置警报，从而持续监控已部署系统，并及早检测异常行为。

4. **模拟 API：** Tenderly 的模拟 API 是智能体系统中最关键的组件。该接口允许代理在指定区块高度对 EVM 链进行分叉，并针对该分叉状态执行拟议的交易。模拟会返回一份全面报告，其中包括执行成功或失败、发出的事件、状态差异以及完整的执行跟踪。事实证明，基于分叉的模拟是预测交易结果并防止复杂 DeFi 工作流发生执行故障的有效方法。

- **在代理操作流水线中的角色：** Tenderly 在代理操作流水线的 **观察**、**推理**和**验证**阶段发挥核心作用。提交交易之前，代理可以先模拟该交易，以回答这个问题：“如果现在执行，这笔交易会产生什么结果？”这样便能在提交不可逆的链上操作之前，检测回滚、意外的状态变更以及在经济上不利的结果。

安全模型： Tenderly 不提供直接的执行或托管保障；相反，它充当预防性安全层。其安全价值源于准确的执行前分析，以及对链上执行语义的高保真复现。这与安全关键系统中成熟的运行时验证和预检验证方法相一致。其主要信任假设是模拟环境能够忠实反映主网行为，包括操作码语义、Gas 计量和合约状态。

生态系统影响： Tenderly 已成为以太坊专业开发的基础工具，对于在高风险环境中运行的自主代理也日益不可或缺。通过支持确定性的执行前分析，它显著降低了高昂错误、失败交易和意外副作用的风险。对于智能体系统而言，Tenderly 就像一款飞行模拟器：它是必不可少的安全层，能将不透明、不可逆的执行转变为可检查、可验证的流程。它的广泛应用反映出安全代理—区块链架构正更广泛地转向模拟优先设计。

Biconomy（2024年）

历史背景与愿景： 自早期开展元交易 [289]和基于中继器的执行工作以来，Biconomy 一直是推动以太坊应用实现“媲美 Web2 的”可用性的核心参与者。在账户抽象时代到来之前，元交易设计提供了一种实用方式，使用户意图与 Gas 付款相分离，并支持免 Gas 入门，但同时也引入了新的中间件信任面和运维复杂性 [48]。随着通过 ERC-4337 实现的账户抽象日趋成熟，Biconomy 已转型为全栈账户抽象基础设施提供商：它不再在链下模拟智能账户，而是提供生产级组件（账户、支付主、打包器、软件开发工具包），让应用和代理能够将可编程授权与 Gas 抽象作为一等接口 [55, 83]。

架构深入解析：

- **账户抽象技术栈构成（账户 + 基础设施 + SDK）**。Biconomy 的产品最好理解为围绕 ERC-4337 构建的一体化流水线：(i)智能账户（基于合约的钱包）、(ii) 捆绑器基础设施，用于纳入 UserOperation；以及 (iii) 支付主，用于 Gas 赞助和替代费用支付。对于代理工作负载而言，这种集成意义重大，因为它减少了定制胶合代码，并将运营相关事项（速率限制、可靠性、监控）集中到一个定义明确的服务边界中 [55, 83]。
- **智能账户安全边界**。从安全工程的角度来看，账户抽象将主要授权面从外部拥有账户的私钥转移到附加于智能账户的验证逻辑和模块。因此，模块正确性和权限范围限定成为决定委托代理执行安全性的主要因素（PatternIII）。设计原则是在验证层贯彻最小权限（函数/合约允许列表、支出上限、到期时间），这样即使代理密钥遭到安全失陷，其权限仍会受到执行时强制执行的确定性检查约束 [55, 285]。

- **支付主与赞助执行（代理用户体验与滥用抵抗能力）**。支付主是提升代理用户体验的直接推动因素，因为它们免除了代理持有原生 Gas 的要求，并支持基于策略的赞助（例如：只赞助特定目标、仅赞助低于某个价值阈值的交易，或仅赞助经过身份验证的会话）。然而，支付主也会带来拒绝服务与骚扰攻击面（赞助耗尽、对抗性交易构造），因此必须将其视为承载策略的安全组件，而不只是用户体验组件。在生产部署中，赞助规则应可审计、默认采取保守设置，并与强有力的请求身份验证和基于模拟的预检查相结合 [55, 83]。

- **会话密钥即受限能力**。对智能体而言，最重要的操作原语是能够签发范围狭窄的会话权限（时间受限、目标受限、价值受限），并由

账户的验证逻辑强制执行。这带来的是能力安全态势，而非单体密钥态势：智能体可以在沙箱内自主操作，撤销也成为常规的控制平面操作（过期、撤销、轮换），而不再是紧急的密钥轮换事件 [285, 55]。

安全模型： Biconomy 的安全模型由两层构成：(i) 在智能账户验证边界进行链上强制执行；(ii) 依赖链下基础设施（打包器/支付主服务）的运营信任，以确保可用性和纳入。有限权限 论点在策略被编码为确定性检查、以限制会话密钥可执行的操作时最具说服力。因此，主要故障模式集中在：(a) 策略/模块配置错误或实现漏洞（导致出现非预期调用路径）；以及 (b) 基础设施层面的问题（审查、错误路由、可靠性下降），这些问题可能迫使系统采用不安全的回退方案，或破坏活性假设 [55, 285]。
生态系统影响： Biconomy 通过提供一套实用的端到端技术栈，大幅降低了采用 ERC-4337 的门槛。从智能体区块链系统的角度来看，它的影响在于，让无法自行维护定制化打包器、支付主服务和账户模块运营的消费级和中端应用，也能切实采用模式 III 式委托执行。从更广泛的层面看，这进一步推动生态系统趋向以可编程账户、明确的策略界面和标准化执行流程作为安全自动化的默认基础 [55, 83]。

Biconomy（2024年）

历史背景与愿景。 Biconomy 起源于账户抽象尚未普及的时期，当时它作为基础设施提供商，致力于提供“元交易”和 Gas 抽象，以减少原生 Gas 支付和逐笔交易签名带来的用户体验摩擦。随着账户抽象日趋成熟，ERC-4337 实现标准化，Biconomy 转而提供全栈账户抽象方案，将智能账户部署、赞助执行和开发者工具整合为一套完整的工作流。从这个意义上说，Biconomy 处于钱包基础设施和应用中间件的交汇处：它着眼于提升开发者体验，同时将可编程授权和费用赞助作为一等原语，为面向消费者和代理的应用提供实际支持 [55, 83]。

架构深入解析。 从概念上看，Biconomy 与 ERC-4337 执行流水线相契合：客户端构造一个用户操作，将其提交给捆绑器，并依托由入口点协调的智能账户验证／执行路径 [55]。其“全栈”定位通常体现在三个层面：

- **智能账户抽象**。 Biconomy 提供智能账户模板和 SDK 级抽象，将可编程授权作为可组合模块（例如验证器、支出规则和会话范围权限）加以呈现。这是支持代理工作流的关键因素：此类工作流需要受约束的委托，而非持久性的主密钥托管（分类体系中的模式 III） [55, 83]。
- **捆绑器和支付主服务**。 在 ERC-4337 系统中，捆绑器通过聚合用户操作并支付基础层 Gas 来提供纳入保障和活性，而支付主则依据应用定义的规则赞助 Gas [55, 280]。Biconomy 将这些角色作为托管服务实现，使应用能够提供 Gas 赞助流程，而无需每个团队自行运行专用基础设施。
- **面向开发者的 SDK 和策略钩子**。 对于智能体应用场景而言，实际的差异化优势不仅在于能够发送交易，还在于能够以结构化方式表达和检查约束。Biconomy 的技术栈强调 SDK 级支持，用于构建受约束的执行流程（例如具有时间／价值限制的权限，以及合约／函数范围限定）；这与智能账户的链上验证逻辑相辅相成，并减少对脆弱的、仅依赖提示词级安全性的做法的依赖 [55, 83]。

安全模型。 Biconomy 的安全态势最好从分层角度来理解：(i) 链上验证 在 validateUserOp 阶段确定性地执行账户规则；(ii) 基础设施正确性 （捆绑器/支付主的行为和可用性）会影响执行可靠性和故障模式；(iii)策略设计质量 决定了委托密钥和自动化的实际影响范围。因此，主要的信任假设集中在已部署智能账户验证逻辑的正确性，以及任何托管基础设施组件的运维安全上，这与通用 ERC-4337 威胁模型一致 [55, 280]。

生态系统影响。 Biconomy 通过将原本分散的 ERC-4337 角色整合为面向开发者的平台，推动了账户抽象走向主流。对于智能体区块链系统而言，其战略价值在于让受约束的委托（类似会话的权限）、Gas 赞助和标准化用户操作流程能够切实实现规模化应用，从而降低部署模式 III 式自动化的门槛，而无需直接采用风险较高的自主签名（模式 IV） [83, 55]。

ZeroDev（2024年）

历史背景与愿景。 ZeroDev 作为 ERC-4337 生态系统中的开发者优先型服务提供商应运而生，重点提供模块化智能账户和托管基础设施（捆绑器/支付主），以降低账户

抽象的集成开销。其所述的市场定位类似于智能账户的“支付基础设施”：提供可靠的基础构件，用于账户创建、赞助执行和许可式自动化，让应用团队能够构建智能体工作流，而无需从头到尾重新实现账户抽象技术栈。这与 ERC-4337 更广泛的架构动机相契合：无需进行基础层硬分叉，即可实现可编程授权和账户级定制 [55, 83]。

架构深入解析。 从架构角度来看，ZeroDev 代表了 2024–2025年常见的一种账户抽象模式：围绕规范的 EntryPoint流程，提供一种具有明确设计主张的模块化智能账户设计，以及托管式打包/赞助基础设施 [55]。其主要组成部分包括：

- **模块化智能账户设计**。 现代账户抽象技术栈日益采用“面向插件”的设计：不同的验证器、恢复机制和委托执行策略可以作为模块组合，而不必硬编码到单体账户中。这种模块化对于代理部署尤其重要，因为它既支持严格限定范围的权限（模式 III），也支持回退和恢复控制（这对运行安全至关重要） [83, 55]。
- **捆绑器和赞助基础设施**。 与其他账户抽象服务提供商一样，ZeroDev 的托管组件实现了交易打包和费用赞助。这对智能体很重要，因为它能实现自动化，无需代理管理原生 Gas 余额，并为向 EntryPoint合约提交用户操作提供标准化途径 [55, 280]。
- **许可式自动化接口**。 在智能体场景中，“会话密钥”最好理解为能力令牌：它们是受时间和策略限制的签名权限，其效果由智能账户验证器执行。尽管具体实现各不相同，但设计目标与 ERC-4337 的可编程验证模型一致：将执行纳入确定性验证代码，并减少对链下善意或用户界面介导的确认的依赖 [55, 83]。

安全模型。 ZeroDev 风格的账户抽象技术栈，其核心安全保障源自链上验证：如果验证器拒绝某项操作，即使代理运行时遭到安全失陷，该操作也无法执行。然而在运行层面，系统仍容易受到基础设施层故障模式（可用性、审查压力、中继/捆绑器行为不当）以及策略配置错误（权限过于宽泛、缺少支出限额、目标允许列表过于宽松）的影响。因此，实际的安全立场是“确定性执行，但依赖运行基础设施”，这与 ERC-4337 的通用信任与威胁概况相符 [55, 280]。

生态系统影响。ZeroDev 通过开发者工具，让开发者能够使用模块化智能账户和基于用户操作的执行，从而推动了账户抽象的普及。在你的分类体系中，它的主要意义在于让模式 III 能够在生产环境中部署：受限委托执行（通过可编程验证），结合赞助和标准化提交，为自动化提供了一条切实可行的路径，同时相较于模式 IV 的自主签名，仍保留了实质性的限制 [83, 55]。

7.5.4 Alchemy AccountKit（2024年）

历史背景与愿景： Alchemy 是 Web3 领域领先的开发者基础设施提供商之一，其进入账户抽象（账户抽象）领域，反映了生态系统从外部拥有账户（EOA）转向可编程智能账户的趋势。围绕单例 EntryPoint、用户操作对象、打包器和支付主构建的账户抽象机制，已被广泛分析为一种无需更改基础层即可实现“钱包可编程性”的实用路径，因此尤其适合需要受限委托和可审计策略钩子的代理式执行流程 [308, 280]。Account Kit 延续了 Alchemy的平台理念：开发者应能够使用统一的、面向生产环境的技术栈来构建、运营和监控账户抽象驱动的应用，而不必临时拼凑异构组件。

架构深入解析：

- **核心账户抽象接口**。 Account Kit 以开发者友好的形式实现了 ERC-4337 风格的流水线：应用（或代理）生成结构化的用户操作，捆绑器将操作打包以纳入区块，账户则在执行前强制执行验证逻辑。对于智能体系统而言，这种架构尤为重要，因为它清晰地分离了（i）意图形成、（ii）授权/验证以及（iii）提交/执行，从而支持在账户层进行范围明确的委托和确定性执行 [308]。

- **Light Account 的设计选择**。 Alchemy 的 *Light Account* （按其公开定位）强调极简主义和 Gas 效率。实际上，这种设计偏向通常会以牺牲可扩展性（大量可选模块和升级接口）为代价，换取更简单的安全分析和更低的单次操作开销。对于创建或

管理大量账户的代理部署而言，这种方式可能很有吸引力，因为它能降低用户引导和日常自动化操作的边际成本。

- **捆绑器与支付主集成**。 Account Kit 的一项显著特点是将交易打包和 Gas 赞助作为一流的托管服务提供：捆绑器路径可确保用户操作可靠地纳入区块，而支付主则支持费用赞助（或有条件的赞助策略）。通过支付主中介执行是自主代理的重要赋能因素，因为它将执行与原生 Gas 库存管理解耦，并使费用策略能够作为系统控制平面的一部分加以管理 [280]。这也与早期关于元交易式抽象的研究方向一致：第三方可以在明确的授权约束下协助执行 [152]。

- **统一的开发者控制平面**。 通过在同一个服务平台中整合账户生命周期管理、用户操作提交、赞助配置和监控，Alchemy 降低了集成复杂度和运维偏差。对于智能体构建者而言，实际好处是从签名到提交的路径上需要定制开发的组件更少，从而减少了需要审计和监控的信任边界数量。

安全模型： 其安全态势最恰当的描述是：账户抽象强制的正确性与托管基础设施的可靠性。在链上，智能账户的验证逻辑定义了授权边界，并可编码最小权限约束，以限制代理的影响范围。在链下，运营信任假设则转向对 Alchemy 托管的捆绑器和支付主服务（包括其策略配置和事件响应）的可用性与完整性的信任。这是一种常见的权衡：开发者接受一定程度的中心化，以换取生产级可靠性和简化的运维，同时保留账户层的密码学授权 [308, 280]。

生态系统影响： Alchemy 的分发能力和开发者关注度，使 Account Kit 成为推动账户抽象模式普及的催化剂。对于代理—区块链生态系统而言，它的意义与其说是引入一种新原语，不如说是让一种实用的“代理就绪型”执行底层架构成为常态：结构化操作、可编程验证、Gas 赞助和集成可观测性。这些功能相结合，降低了部署代理式应用的门槛，使其能够在现实环境中实现受限自主性、高吞吐量和运营监控。

7.6 关键见解与新兴架构趋势

我们的比较分析（表6）揭示了若干架构趋势，这些趋势日益定义着2024–2025年生产级智能体—区块链系统的特征。

1. 账户抽象正逐渐巩固为默认的钱包底层架构。 智能账户和 ERC-4337 生态系统已从实验性的用户体验改进，发展为代理式授权的实用基础。这主要是因为，智能账户能够在无需第一层协议变更的情况下，使授权具备可编程性 （验证模块、策略钩子、恢复流程），同时保持与现有工具和执行环境的兼容性 [55, 308]。以账户抽象为重点的技术栈（例如 Biconomy、ZeroDev、Alchemy Account Kit）对策略、预览和模拟 consistently strong 的矩阵支持表明，主要的工程工作已从“我们如何签名？”转向“我们如何约束、预览并验证将要签名的内容？”

2. 从命令式交易持续转向声明式意图。 在最成熟的系统中，智能体越来越多地表达结果，而非底层调用序列。以意图为中心的设计减少了智能体有效的行动空间，规范了必须验证的内容，并使安全检查更易于审计，因为签名对象相比执行机制（路由器地址和调用数据）更接近用户意图（资产、界限、截止时间）。基于意图的交易以及批处理／拍卖机制，也是对不利执行环境的直接回应，尤其是对公共内存池中最大可提取价值提取的回应 [80, 179]。在实践中，意图允许将执行复杂性（路由、批处理、求解器选择、私有订单流）委托给专业化基础设施，由其在执行用户签署的约束的同时，内化感知 MEV 的优化。

3. 策略执行是高风险自主性的把关机制。生产系统普遍将策略视为防范遭入侵的智能体、定义错误的目标和工具层欺骗的关键保障。实现方式的差异主要体现在架构层面：通过智能账户模块实现链上策略（以提供透明且可执行的约束），或通过托管／工作流引擎实现链下策略（以提升运营敏捷性、纳入更丰富的上下文并支持企业审批流程）。这一模式始终如一：旨在保护重要价值的系统通常会将策略视为一等接口，而非应用层的事后补充 [230]。

4. 托管权衡空间仍是根本因素，而非偶然因素。 </style}该矩阵凸显了一系列长久存在的托管权衡：（i）采用多方计算／基于可信执行环境的签名和工作流控制的中心化企业托管；（ii）采用模块化授权的链上智能账户；以及（iii）尽量减少中介的用户持有密钥。多方计算和门限签名可降低单主机失陷的风险，并支持组织审批语义 [184, 169], 而基于可信执行环境的签名器则强调在硬件保护环境中实现密钥隔离和策略执行 [104]。相反，纯粹的

用户持有密钥模型虽能最大限度地实现自主权，但与自主运行的契合度较低，因为它们将操作策略、签名权限和运行时入侵都归并到同一个故障域中。

5. 通用代理框架在加密原生安全控制方面仍未达到预期。 起步于 Web3 之外的框架通常会提供工具封装器和基础签名钩子，但与加密原生技术栈相比，它们在确定性策略执行、高保真模拟、结构化交易预览和恢复工作流方面的支持仍然有限。这一差距与模型质量的关系较小，更多在于接口规范性：成熟的系统会将"意图→ 策略 → 模拟 → 签名 → 提交 → 验证"视为一条经过工程化设计的流水线，并配备类型化制品和可审计检查点。

6. 模拟与预览正逐渐成为人在回路授权的基本要求。 在各种托管模型中，成熟系统越来越多地要求在授权前进行执行前模拟，并提供人类可读预览，作为最后一道安全检查点；这符合所见即所签原则，并降低了中间件和用户界面层操纵风险 [333, 236]。如果缺少这些功能，系统往往会面向受限的全自动任务，或采取不同的风险态势。

总体而言，2025年的格局显示出正朝着模块化的纵深防御架构趋同：AA 赋能授权、以意图为中心的交互界面、策略即产品（链上和/或链下），以及系统化的飞行前验证与可观测性 [292]。这些趋势推动了下一节关于标准化的讨论，因为互操作性和可审计性取决于能否让这些接口在不同供应商和生态系统之间实现可组合。

7.6.1 去中心化自动化与预言机

Chainlink Functions（2024年）

历史背景与愿景。 **[64]**Chainlink 是去中心化金融领域占主导地位的去中心化预言机生态系统，其路线图不断拓展，从数据馈送逐步扩展到为智能合约提供更通用的链下服务。在这一发展历程中，*Chainlink Functions* 将预言机模型从"数据传递"扩展至"可验证链下计算"，旨在让合约能够安全地使用任意应用程序接口输出和计算结果，而无需退回到依赖单一可信服务器的做法 [46, 50]。

架构深入解析。 从整体上看，Functions 实现了一个请求—计算—证明—响应循环：链下执行由去中心化预言机网络完成，结果则以具备真实性保证的方式返回链上 [50]:

- **请求构造（链上）**。 启用了 Functions 的合约会发出请求，其中指定：（i）计算载荷（通常为 JavaScript 或受限程序）；（ii）输入参数；（iii）任何必需的机密/配置引用；以及（iv）对响应格式的预期。

- **链下执行（预言机委员会）**。 指定的预言机网络（或委员会）获取请求，在受限环境中执行负载，并执行程序所需的任何外部 API 调用。此阶段旨在将传统网络数据和计算资源引入合约，同时使信任模型更接近去中心化预言机的假设，而不是依赖单一中间件服务器 [46]。

- **聚合与证明**。 节点返回结果，这些结果根据网络协议进行聚合，生成最终响应，然后将其传回发起请求的合约。其关键设计目标是，响应并非仅仅被"报告"，而是在旨在支持预言机层完整性与来源信息保证的框架下生成的 [50]。

- **链上交付（响应回调）**。 最终结果会发布到链上，并通过回调交付给请求合约，使合约能够更新状态或触发后续逻辑。

代理的作用。 Functions 主要通过扩展链上代理（或代理控制的合约）在执行时能够可靠"知晓"的信息，使其受益于**观察** 和 **推理** :

- **链下数据获取**。 合约可以使用链上原生无法获取的外部信号（例如企业系统、公共数据集或领域专用 API），从而减少对中心化"keeper"后端的需求 [46]。

- **计算卸载**。 高成本计算可以转移到链下，同时保留结构化且可审计的请求与响应路径。当替代方案要么需要支付高昂的 Gas 成本，要么涉及不透明的中心化计算时，这一点尤为重要 [173]。

安全模型。 其安全态势属于去中心化预言机网络：正确性取决于协议对抗性假设，包括委员会选择、聚合，以及参与节

点的经济或运营激励。换句话说，Functions 降低了对单一可信服务器的依赖，但仍面临预言机领域的常见挑战：如果攻击者能在经济或运营层面破坏预言机流程，或操纵链下执行环境，就可能扭曲执行结果。因此，预言机系统的设计指南强调采用多源策略、经认证的来源信息，并明确计算与数据管道中的信任假设 [46, 50]。

生态系统影响。 Functions 通过支持智能合约在预言机中介的信任模型下使用任意 API 输出和计算结果，拓展了混合链上系统的设计空间。这显著拓宽了"适合代理"的智能合约所能实现的功能，尤其是需要外部上下文的自动化和决策逻辑，同时保持接口兼容智能合约执行约束和可审计性要求 [73, 173]。

Gelato Network（2024年）

历史背景与愿景。 Gelato 的出现旨在解决智能合约的一项结构性局限：合约逻辑是响应式的，执行需要由外部提交交易。Gelato 的主张是，Web3 需要一个自动化层，能够在满足基于时间或条件的谓词时，可靠地触发链上操作，而无需应用团队运维定制的机器人集群。在智能体场景中，这一自动化层通过为必须在时间和活性约束下执行的规划提供稳健的执行后端，补充代理操作流水线。

架构深入解析。 Gelato 可以理解为一个双边执行市场：

- **协议层**。 开发者注册自动化任务，指定 (i) 目标智能合约和调用数据模板，(ii) 触发条件（时间计划、事件谓词、链上状态检查），以及 (iii) 费用或赞助配置。这类似于早期元交易和中继器模式：执行委托给专门的提交者，但将其推广到任意条件执行 [270, 263]。
- **执行者层**。 链下执行者会监控任务，并在条件满足时提交交易、收取费用。执行者之间的竞争性可用性在实践中保障了活性：如果某个运营者离线，其他运营者可以执行同一任务，从而形成基于市场的冗余模型。

Ke智能体的关键功能。 Gelato 最自然地对应于**执行**（从操作层面而言，也对应于**验证**）：

- **执行委托**。 智能体可以提出有界规划（交易模板加约束），并将提交工作委托给 Gelato，从而减轻运营负担，无需为调度和事件监控专门搭建基础设施。
- **Gas 抽象和中继**。 由于中继器式执行将 Gas 管理从代理转移出去，设计可以避免在代理运行时中维持充足的热钱包 Gas 余额。这与更广泛的元交易文献相符：首选批处理器/中继器可以减少摩擦并提高可靠性 [152]。

安全模型。 核心安全属性是已执行操作的完整性：执行器不应能够篡改用户预期的调用。实际上，这取决于如何对任务进行编码，以确保已执行的调用数据是固定的，或可根据经认证的输入推导得出；同时还要确保任何动态参数本身都受到约束或经过链上验证。剩余风险主要在于可用性和经济层面：费用不足时，执行者可能会延迟执行；如果提交任务时没有采用私有订单流保护，对手仍可利用公共内存池的可见性。因此，在以下情况下，Gelato 的表现最为出色：(i) 搭配链上策略模块，限制受委托执行者的操作范围；以及 (ii) 在执行对最大可提取价值敏感时，采用私有提交路径 [80]。

生态系统影响。 Gelato 已成为用于定期维护操作、限价单式执行和运维工作流的常见自动化原语。对智能体构建者而言，它的主要贡献是将"在满足条件时可靠执行"转化为基础设施，而非定制工程，从而降低部署持续运行、全天候自主行为的门槛，同时避免将托管与执行可靠性都交由代理运行时承担。

OpenZeppelin Defender（2023年）

历史背景与愿景。 OpenZeppelin Defender 将 OpenZeppelin 的安全优先理念从合约库延伸至运维安全 [229]。其前提是，许多事件的成因不仅是合约漏洞，还包括脆弱的运维工作流：未受管理的管理员密钥、临时脚本、监控缺失以及响应延迟。Defender 将监控、自动化和安全交易提交整合到一个托管控制平面中，旨在减少运维错误，并为智能合约系统规范事件响应。

架构深入解析。 Defender 最适合被视为一套具有明确立场的安全运营技术栈：

• **监控与检测**。 事件级和交易级监控组件支持基于规则的检测，用于识别异常或与策略相关的链上活动。这通过将原始链数据转化为可执行的触发器，支持**观察** 和**验证** 阶段。

• **Autotasks 作为受控的链下执行器**。 Autotasks 运行有明确边界的链下逻辑，可由计划任务、Webhook 或监控触发器调用。这为生产环境中的类智能体工作流提供了一种务实的实现方式：采用定义明确的运行时、托管密钥和可审计执行日志，而不是使用凭据存储随意拼凑的专用机器人。

• **管理与交易执行**。 Defender 支持结构化运维操作（暂停、升级、角色变更），并集成了安全密钥管理方案，例如多重签名和中继器。这与更广泛的观点一致：在已部署系统中，运维正确性和访问控制与合约正确性同等重要 [310]。

智能体关键特性。 Defender 最适合与**模式 II/III**部署（代理提出操作，受约束的执行器提交交易）配合使用；与严格的策略和审批机制结合时，也可支持**模式 IV**式的自主性：

• **安全感知自动化**。 监控触发器可以控制执行流程，使自动化仅在满足可验证条件时才会发生（例如，在检测到已知漏洞利用特征时执行补救操作）。

• **运维防护措施**。 托管密钥、标准化日志和受控运行时降低了代理运行时成为凭据泄露或不安全临时脚本薄弱环节的可能性。这种运维规范是密码学托管原语和链上策略模块的实用补充。

安全模型。 Defender 主要通过运维隔离提供完整性和可审计性：受控执行环境、结构化权限管理和可观测性。核心信任假设在于平台运营方，因为控制平面是中心化的。作为交换，该平台降低了常见的部署风险，并支持快速响应工作流，而使用定制基础设施很难可靠地复现这些能力。从威胁模型的角度看，主要残余风险是与任何托管服务相关的供应链风险和凭据风险，这进一步凸显了在生产部署中采用最小权限凭据、限定范围的中继器和职责分离的重要性。

生态系统影响。 对于希望实现注重安全的自动化、又不想运营自己的机器人集群的团队而言，Defender 已成为默认的运维层。对于智能体—区块链系统而言，其主要贡献是使监控到操作的闭环能够通过具有完善审计跟踪和明确运维控制措施的方式进行部署，而这通常是注重安全的环境采用生产环境的前提。

7.6.2 智能体专用框架

MCP（模型上下文协议）（Anthropic，2024年）

历史背景与愿景： MCP 最适合被理解为一种互操作与集成标准，而非钱包产品或链上协议。它源于实际需求：需要标准化应用向模型提供上下文 的方式，包括工具规范、数据源和运行时约束，同时确保应用（代理运行时）与模型（推理组件）之间清晰的角色分离。综述将 MCP 定位为更广泛的 2024–2025年趋势之一：转向开放、模型无关的工具层，以减少供应商锁定，并促进由可组合工具和代理运行时构成的生态系统。

架构深入解析：

• **核心抽象：模型到工具的标准化**。 MCP 提供了一种结构化方式，让代理运行时描述可用工具（能力、架构和调用模式），并协调工具调用与响应。在区块链场景中，这一点很重要，因为同一个代理可能需要与各种异构端点交互，例如 RPC 提供商、模拟器、索引器、中继器和托管服务，而无需为每种集成都硬编码定制封装层。

• **通过设计确保信任边界卫生**。通过标准化向模型呈现工具输出的方式，MCP 促进了以下两者在架构上的分离：(i) 负责执行权限、日志记录和策略检查的代理运行时；以及 (ii) 根据输入和工具结果提出操作的模型。这种分离符合威胁模型的重点，即工具边界是主要攻击面，并且安全性应由确定性的运行时控制来保障，而非寄希望于模型行为。

安全模型：MCP 本身并不像智能账户策略模块或多方计算签名器那样提供安全保障。相反，它提供了连接各组件的基础，使安全控制能够集中在运行时层：能力范围限定、允许列表、对工具调用的审计，以及附加到工具输出的一致来源元数据。实践中，MCP 与防护组件配合使用时最有价值；这些组件会在模型提出的操作进入签名或提交环节之前对其进行验证和限制。

与智能体—区块链系统的相关性：对于链上代理，MCP 最适合作为模式 *I–III* 的促成因素：它能让工具集成更加统一、可审计，从而提高观察、推理和规划阶段的可靠性与可移植性。它还能减少生产环境中的集成脆弱性，因为代理通常依赖会随时间变化的第三方服务（模拟、监控、私有中继、托管 API）。

与"评分器"或监督模式的关系：虽然 MCP 并非模型评分框架，但它很容易与那些在执行前使用次级评估信号的监督架构组合使用。例如，可以实现一个两步运行时：工作模型提出结构化意图，独立的审查流程则在允许签名或提交之前，根据安全规则、风险启发式规则或红队评估清单评估该意图。关于可扩展监督、辩论和系统化红队测试的人工智能安全研究，为采用这种多阶段监督提供了更广泛的依据 [52, 153, 238, 114]。在区块链部署中，这可以直接对应类型化意图和策略决策记录，而 MCP 则充当标准工具接口，使两个阶段都能借此获取可验证的上下文并执行检查。

LangChain Web3（2024年）

历史背景与愿景：LangChain 于 2022年问世，成为最具影响力的开源大语言模型（LLM）应用构建框架之一。它的核心贡献是引入一种模块化、可组合的抽象，将模型、工具、记忆和控制流串联起来，构建协调一致的智能体系统。随着大语言模型的应用加速普及，LangChain 成为各领域工具使用型代理的默认实验和原型开发环境。2023—2024年开发的 LangChain Web3 扩展将这一理念延伸至区块链领域，明确旨在降低现有 LangChain 用户的使用门槛，使他们能够构建可观察链上状态、对其进行推理并采取行动的代理 [177]。

架构深入解析：

- **框架级集成，而非基础设施**。LangChain Web3 是一种库层集成，而非托管系统、执行网络或策略框架。它提供了一系列抽象，使区块链交互能够与网络搜索、数据库和 API 一样，作为一等工具纳入 LangChain 代理循环。
- **Web3 工具封装器**。 Web3 工具包将常见的区块链操作封装为可调用工具，包括读取合约状态、查询余额、获取交易回执以及提交交易。这些工具主要通过将原始远程过程调用交互转换为结构化、模型可读取的响应，来支持代理流水线中的**观察** 阶段。
- **动态智能合约接口**。 LangChain Web3 提供了加载合约 ABI 并为单个合约函数动态生成可调用工具的实用程序。这样，代理便可限定在精确的链上能力集内，例如与特定的去中心化交易所或治理合约交互，而无需为每个方法硬编码逻辑。
- **签名者抽象**。 该框架提供了一个接口层，用于集成各种签名后端，从本地外部拥有账户到外部签名服务皆可支持。这种抽象使 LangChain 智能体能够参与流水线中的**授权** 和**执行** 阶段，同时将实际的密钥管理委托给外部系统。

在代理—区块链架构中的角色： LangChain Web3 最适合被视为一个认知与编排层。它擅长组织多步推理、维护对话或任务记忆，以及安排工具使用。在本综述前文介绍的分类体系中，LangChain 自然支持**模式 I（只读智能体）**和**模式 II（意图生成）**；与外部托管、策略和执行组件配合时，也可参与**模式 III（委托执行）**。它并非专为高风险自主执行而设计的独立解决方案。

安全模型与局限性： LangChain 本身不为链上交互提供任何安全保障。它不会强制执行交易策略、模拟执行结果、限制签名权限，也无法防范提示注入或工具滥用。因此，基于 LangChain 的 Web3 智能体的安全态势完全取决于其周边架构。在实验系统中，不安全的部署方式仍是常见的故障模式，例如授予代理直接访问不受限制的私钥的权限 [131]。

因此，在生产级智能体—区块链系统中，LangChain 必须与以下方案搭配使用：

- 用于受限签名的智能账户或基于多方计算的托管，
- 链下或链上策略执行层，
- 执行前的模拟与预览工具，
- 以及在适当情况下，借鉴监督式代理设计的二次审查或评分机制 [52,114]。

生态系统影响： 尽管存在局限，LangChain 仍作为一种入门途径，在生态系统中发挥着关键作用。它在人工智能社区中的广泛采用，极大地加快了大语言模型与区块链交汇领域的实验进程。尽管许多基于 LangChain 的 Web3 代理仍处于原型阶段，但该框架让开发者能够在投入安全、生产级部署所需的更繁重工程工作之前，探索代理交互模式，从而推动了创新。从这个意义上说，LangChain Web3 是代理—区块链生态系统发展的重要催化剂，但要满足本综述所述的安全性与可靠性要求，它仍依赖于更专业的基础设施。

AutoGPT Crypto（2023年）

历史背景与愿景： 2023年早期自主代理项目的发布，让广大开发者受众切实感受到“自主驱动”的大语言模型工作流。尤其是 AutoGPT，它推广了一种简单却极具吸引力的模式：代理通过工具调用，围绕高层目标不断循环执行推理 → 操作 → 观察，以逐步推进目标。这种模式与 ReAct 式框架高度契合，即交替进行推理、操作和观察；它也影响了许多后续代理框架对工具编排和记忆的构想。 [330, 267] 在加密生态系统中，社区分支及其扩展（通常被非正式地称为“AutoGPT Crypto”）将同一循环应用于市场监控、链上数据检索和自动化执行等任务。

架构深入解析：

- **核心循环**。 典型设计采用迭代式控制回路：
  1. **推理：** 代理根据目标和累积状态提出下一步行动。2. **行动选择：** 代理从可用工具中进行选择（例如数据获取、合约读取、交易下单等），并以结构化格式输出工具参数。3. **观察与记忆更新：** 代理接收工具输出，总结重要结果，并将其追加到工作记忆或日志中，供下一轮使用。

  这一循环在概念上与广泛研究的工具使用型代理设计一致。 [330, 189]。
- **工具面与执行的耦合**。 “加密货币”变体与通用代理演示的区别在于，开放世界输入（网页内容、价格馈送、链上元数据）与高风险操作（签名、授权、转账）之间存在直接耦合。在实践中，使工具能够灵活组合的同一机制也扩大了攻击面，因为不可信工具输出可能影响后续的行动选择。 [131, 188]

安全模型： 从威胁模型的角度来看，早期自主加密代理最好被视为治理程度最低的组合，而非经过强化的系统。若缺少严格的策略门控、类型化意图约束和执行前模拟，代理实际上是在宽泛的行动空间中运行，且可信指令与不可信内容之间缺乏明确区分。关于提示词和工具鲁棒性的实证研究表明，这类设计容易受到指令引导攻击和工具介导的操纵，尤其是在检索内容与控制逻辑被融合到同一上下文中时。 [131, 191, 348] 因此，这些原型更适合作为探索性演示，而非在缺乏额外控制措施的情况下用于管理具有实质价值的架构。

生态系统影响： 尽管存在局限，AutoGPT 式的加密货币分叉项目仍为生态系统发挥了重要的“推动力量”作用。它们让不受约束的自主性所带来的风险变得具体可见，并推动业界转向生产级技术栈中如今已然可见的纵深防御模式：智能账户与限定范围的会话密钥、明确的策略引擎、确定性模拟，以及可审计的执行流水线。更广泛的智能体文献也印证了同一趋势：随着工具可用性不断提升，可靠性与安全性要求加强对工具使用的治理、采用结构化接口并进行系统性评估，而不能仅仅依赖临时提示。 [189, 333]。

综合：将代理框架作为安全边界。 总的来说，本小节中的智能体专用框架表明，“智能体能力”和“代理安全”在很大程度上取决于模型周围的边界，而非模型本身。LangChain 等通用框架通过让工具使用易于定义和扩展，最大限度地提升了可组合性和开发者效率；然而，在高风险场景中，这些框架默认并不安全，因为它们并未内在地强制分离不可信观察与特权操作，也不

要求使用类型化意图、确定性约束或执行前验证。因此，在我们的分类体系中，LangChain 风格的技术栈最适合用于模式 I/II 部署，除非它们配备外部控制措施；因为若缺少严格策略门控便转向模式 III/IV（委托执行或自主签名），威胁模型中遭遇 C1–C4 类风险（认知劫持、工具/数据欺骗、中间件操纵以及密钥/凭证泄露）的可能性会大幅增加。相比之下，MCP 风格的监督循环和以策略为中心的架构有意牺牲灵活性，以换取结构化安全：它们在意图形成与执行之间加入独立的评估或策略层，优先采用规范化意图对象而非自由形式计划，并使授权取决于可验证检查（模拟、界限、允许列表和托管约束）。这种设计选择通过在关键的信任边界（代理↔工具、代理↔托管以及提交路径动态）约束攻击面，直接针对我们先前指出的主要故障模式，并以一种在对抗性条件下仍可审计且可执行的方式实现模式 III/IV 系统。简言之，LangChain 是构建链上代理大脑的绝佳基础，但 MCP 风格和策略驱动的方法，才能让这个大脑在具有实质风险价值的情况下投入部署。 [131, 189, 330]。

综合：代理框架作为“思维中枢”与“治理者”。 表6中的智能体专用框架可清晰地分为两类：（i）让工具使用更便捷的编排框架，以及（ii）让工具使用更安全的治理框架。LangChain 风格的技术栈在可组合性、快速原型开发和异构工具集成方面表现出色，但在高风险场景中默认并不安全，因为框架本身不会强制执行托管、策略、预览或模拟方面的约束。因此，这类框架自然适用于模式 I 和模式 II 部署：代理提出操作，但不持有权限，而最有力的控制措施位于模型之外。相反，MCP 风格和以策略为中心的方法将代理重新定位为一个组件：它必须通过发出结构化意图、接受独立的策略评估并获得可审计审批，才能取得权限。这与我们的威胁模型所强调的故障模式（尤其是认知劫持和工具/数据平面欺骗）相吻合。这种结构化的做法限制更多，但却是模式 III 和模式 IV 系统得以实现的先决条件；只有在确定性约束、可验证预览和可撤销能力的限制之下，自动化才站得住脚。简而言之，生态系统正逐渐形成一种务实的分工：通用代理框架提供推理和编排基础，而策略引擎、可编程钱包及可验证执行路径则提供治理基础，以遏制安全失陷并限制影响范围。

7.7 2025 年能力矩阵分析

本节对比较能力矩阵（表6）进行叙述性分析，重点关注各系统在所评估维度上的设计选择如何落实（或未能落实）我们的分类体系和威胁模型所要求的控制措施。我们的目标不是对系统进行排名，而是描述截至 2025 年已形成的架构均衡，以及安全实现自主执行仍存在的差距。

7.7.1 托管模型：权限配置与爆炸半径

托管是基础维度，因为它决定了签名权限所在的位置，进而决定安全失陷会如何扩散。我们观察到四种反复出现的托管模式：

- **用户持有的密钥（EOA 式自托管）**。这种模型最大限度地保障自主权，但也将故障集中于单一密钥之上。实际上，它并不适合自主执行，因为它将概率决策组件与不可撤销签名耦合在一起，却没有可强制执行的约束条件。

- **智能账户（通过可编程验证实现委托托管）**。账户抽象将权限从单一私钥转移到可编程验证逻辑，从而支持支出限额、角色分离和有时限的委托。这是模式 III（委托执行）的主流基础，因为它允许代理通过范围严格限定的能力而非主权限来执行操作。

- **阈值托管（多方计算和门限签名）**。分布式托管通过要求多个份额参与授权，降低单主机失陷的影响。现代阈值签名方案和多方计算工作流能够为运营环境提供实用的延迟和可用性 [184, 169]。该模型非常适合模式 IV 部署，此时代理参与授权，但仍必须满足独立的共同签名和策略检查。

- **硬件隔离托管（基于可信执行环境的签名）**。硬件隔离旨在即使在主机失陷的情况下，也能防止密钥材料暴露。安全态势取决于可信计算基，以及围绕签名请求的策略执行完整性，但它为密钥处理面提供了明确的遏制边界。

在这些模型中，关键趋势是逐渐摆脱“密钥位于代理运行时”这一模式，转向通过构造使代理可撤销且受限的托管机制。

7.7.2 权限与策略：从粗粒度角色到类型化约束

能力矩阵显示，各方普遍认同，自主执行所需的不只是粗粒度角色。 [3]我们观察到以下演进过程：

- **基于角色的控制（基于角色的访问控制）**。智能合约层仍普遍采用简单角色，但它们对代理而言过于粗略，因为无法体现交易语义（资产、交易对手方、滑点、截止时间）。
- **链上基于能力的委托（会话密钥和模块）**。智能账户模块支持有时限、价值受限和功能受限的委托，为模式 III 系统提供确定性沙箱。这通过限制安全失陷的推理所能执行的操作，直接缓解了威胁模型中关于“代理失陷”的假设。
- **用于语义检查的链下策略引擎**。对于模式 IV，更丰富的策略通常位于链下，以便纳入上下文、模拟和运维工作流。核心要求是，策略决策必须可审计，并对最终签署的内容具有约束力，从而符合类别 3 中所述的预览至签名完整性边界。

最强的设计会将类型化意图与策略评估结合起来，从而减少授权时对自由文本的依赖。这与安全和合规领域更广泛地转向形式化执行的趋势一致 [230]。

7.7.3 意图接口：降低认知负荷和 MEV 暴露

第二个主要的趋同方向是转向意图式接口：代理表达结果和约束条件，而不是组装命令式交易。这一点有两个方面的重要意义。

首先，意图通过缩小呈现给代理的行动空间，减轻认知负担并减少出错面。其次，意图路由能够通过启用私有或基于拍卖的执行路径，减少对公共内存池对手的暴露。这与公共网络中交易排序和提取无处不在这一经验事实相符 [80, 33]。近期关于以意图为中心的执行的研究，进一步说明这一方向可作为应对对抗性排序的一种安全与市场设计方案 [220]。

在分类体系中，以意图为中心的执行是从模式II（意图生成）通向模式III和模式IV最清晰的路径，因为它能在行使权限之前，为模拟、预览和策略评估提供清晰的接口。

7.7.4 预览与模拟：人工批准与安全自主性的前置条件

预览与模拟能力并非装饰性功能。它们是不可撤销执行前的最后一道防御层，并通过确保授权与将要签名和提交的内容相绑定，将所见即所签式的完整性目标落实到实际操作中。

该矩阵表明了一项实践规范：面向用户批准和高价值执行而设计的系统，越来越多地将高保真模拟和清晰的预览视为基线要求。相比之下，专注于基础设施原语的组件（例如纯密钥服务或自动化轨道）可能不会自行提供模拟，但当它们与上游模拟器和下游签名器配合使用时，安全性会得到提升；后者在没有经过验证的意图封套时会拒绝签名。

7.7.5 可观测性：作为一等能力的可审计性

可观测性是连接预防与恢复的纽带。成熟的技术栈越来越多地记录的不仅是最终交易，还包括：意图提案、模拟结果、策略决策以及签名请求元数据。这对于事件后取证至关重要，也有助于在发生故障时明确问责范围。它还支持根据险情分析和运行遥测，持续改进策略与护栏机制。

7.7.6 恢复与撤销：设计安全失陷遏制机制

该矩阵表明，整个生态系统中的稳健恢复能力并不均衡。无论采用何种托管模型，最具可操作性的恢复原语包括：

- **即时撤销（紧急停止开关）**。怀疑发生安全失陷时，迅速撤销代理的委托能力是首要的遏制机制。
- **有时限的能力**。短期会话密钥和到期授权可限制最坏情况下的损失，即使撤销有所延迟也是如此。
- **账户级恢复**。智能账户模式支持恢复工作流（包括守护者式机制），可降低设备故障或密钥泄露造成灾难性损失的风险，同时使权限与代理运行时保持可分离。

这些控制措施直接对应威胁模型中的一项假设：安全失陷不是会不会发生的问题，而是什么时候发生的问题；因此，系统设计必须优先实现有界故障。

7.7.7 多链范围：集成广度与假设债务

多链支持拓展了能力，但也增加了假设债务：异构 RPC、各不相同的交易语义、变化多样的最大可提取价值环境，以及桥接依赖。桥接中介设计会带来独特风险，而桥接遭到攻陷仍是重大的系统性风险，因此，与单链部署相比，多链智能体设计需要更严格的来源信息管理和模拟规范。

7.7.8 威胁假设：2025年系统的设计中心

在评估的 13 个维度中，三项威胁假设主导着现实世界中的设计：

- **代理可能受到操纵或遭到安全失陷**。 这推动了设计转向会话密钥、类型化意图，以及在授权时实施策略门控。
- **公共内存池具有对抗性**。 这推动了私有提交、拍卖机制和以意图为中心的执行，以应对已有记录显示普遍存在的价值提取和排序操纵现象 [80, 72]。
- **链上环境充满敌意且不断演变**。 这推动了模拟优先执行，并使设计更倾向于采用有界且可审计的自动化，而非自由形式的交易合成。

总结。 总体而言，能力矩阵描绘出一个正朝着纵深防御体系汇聚的生态系统：通过可编程账户实现有界权限，通过意图接口实现受限动作规范，通过模拟与预览进行执行前验证，并通过策略与可观测性实现可审计性和恢复。剩余的差距与其说在于缺少构建模块，不如说在于如何标准化这些模块之间的互操作：类型化意图模式、可验证的策略决策凭证，以及在面对对抗性工具输出和波动的执行条件时仍能保持安全的可移植权限模型。

7.7.9 按系统类别进行比较分析

除了对单个系统进行分析，按类别考察系统还能揭示重要的规律与权衡，为架构决策提供参考。

托管解决方案：安全性与可用性托管解决方案在安全性与易用性权衡谱上处于不同位置。硬件钱包等自托管解决方案通过隔离网络的密钥存储提供最大安全性，但要求用户具备丰富的专业知识，且可编程性有限。Fireblocks 和 Lit Protocol 等多方计算解决方案将信任分散到多个参与方，从而减少单点故障，同时支持程序化访问。Safe 等智能合约钱包提供链上可编程性，但会带来智能合约风险和 Gas 开销。
对于代理系统，托管解决方案的选择取决于信任模型和运营要求。需要低延迟签名的智能体可能更倾向于采用经过优化的签名协议的多方计算解决方案。需要复杂授权逻辑的智能体可能更倾向于采用具备可编程策略的智能合约钱包。管理机构资产的智能体可能需要机构托管服务提供商提供的合规功能和保险保障。

执行基础设施：速度与保护执行基础设施在速度与 MEV 保护之间的权衡各不相同。直接 RPC 提交延迟最低，但 MEV 暴露程度最高。Flashbots Protect 等私有内存池能够提供 MEV 保护，但代价是增加延迟，并需要信任内存池运营方。CoW Protocol 等基于意图的协议通过求解器竞争实现最优执行，但也会增加复杂性，且可能不支持所有交易类型。

代理系统应根据策略需求选择执行基础设施。高频策略可能优先考虑延迟，而非 MEV 保护。大额交易可能优先考虑 MEV 保护，而非延迟。复杂的多步骤操作可能受益于基于意图的执行，因为它可以针对整个操作进行优化。

策略引擎：表达能力与简洁性 策略引擎涵盖从简单的基于规则的系统到复杂的可编程平台。简单的策略引擎支持支出限额和允许列表等基本约束条件，以最低的配置复杂性提供直接明了的安全性。高级策略引擎则支持复杂的条件逻辑、时间约束以及与外部数据源的集成，从而实现复杂的授权工作流，但代价是复杂性增加。
策略引擎的适当复杂程度取决于应用的风险概况和运维复杂性。简单的应用采用基本的策略约束即可满足需求。复杂的机构级应用则可能需要高级策略引擎提供全面的表达能力，以落实细致的授权要求。

7.7.10 行业趋势与市场动态

智能体与区块链集成领域受到更广泛的行业趋势影响，这些趋势塑造着技术发展、采用模式和竞争动态。

整合与专业化 市场正同时经历整合与专业化。大型平台正在收购或构建覆盖整个技术栈的全面能力，从托管、执行到监控，无所不包。与此同时，专业服务提供商不断涌现，凭借特定领域的深厚专长服务细分市场。

这种动态既为提供便利性和互操作性的集成平台创造了机会，也为在特定领域提供卓越能力的专业解决方案创造了机会。代理系统设计者必须在这一格局中做出选择，在集成平台与最佳组件选择之间权衡。

监管压力与合规 对加密货币和人工智能日益增加的监管关注，正在影响代理与区块链系统的发展。围绕托管、交易监控和风险管理的合规要求，正推动机构级基础设施的采用。新兴的人工智能法规可能会对透明度、问责和人工监督提出额外要求。

前瞻性的代理系统设计应预见监管要求，并主动构建合规能力。从一开始就将合规纳入考量的系统，将更有条件适应不断演变的监管框架。

机构采用 机构对加密货币和人工智能的采用正在加速，进而催生了对企业级代理与区块链解决方案的需求。机构对安全性、合规和运营可靠性的要求高于零售用户，推动了对更稳健基础设施的投资。

机构市场细分更看重可靠性、支持和问责，而非尖端功能。面向机构客户的代理系统提供商必须证明其运营成熟度，并提供适当的服务级别承诺。

开源与标准化 开源开发与标准化工作正在塑造竞争格局。开源代理框架和区块链工具降低了准入门槛，并推动快速创新。围绕接口、协议和数据格式开展的标准化工作促进了互操作性，并减少了供应商锁定。

参与开源与标准化工作，有助于影响生态系统的发展方向，并获取社区知识与贡献。不过，这也需要投入资源，并接受社区治理流程。

8 2026年路线图：TIS和PDR

尽管当前的智能体区块链系统领域创新丰富，但在安全互操作性和主流开发者采用方面，仍缺乏统一、标准化的接口。 [282] 我们的分析表明，许多集成仍依赖厂商专用 API、临时拼凑的消息格式以及协议专属交易构造。这增加了集成开销，也带来了本可避免的安全风险，尤其是在非确定性智能体被迫在原始交易层操作时。为弥补这些不足，我们提出了以两项互补接口标准为核心的 2026年研发路线图：一项是**交易意图模式（TIS）**，另一项是**策略**

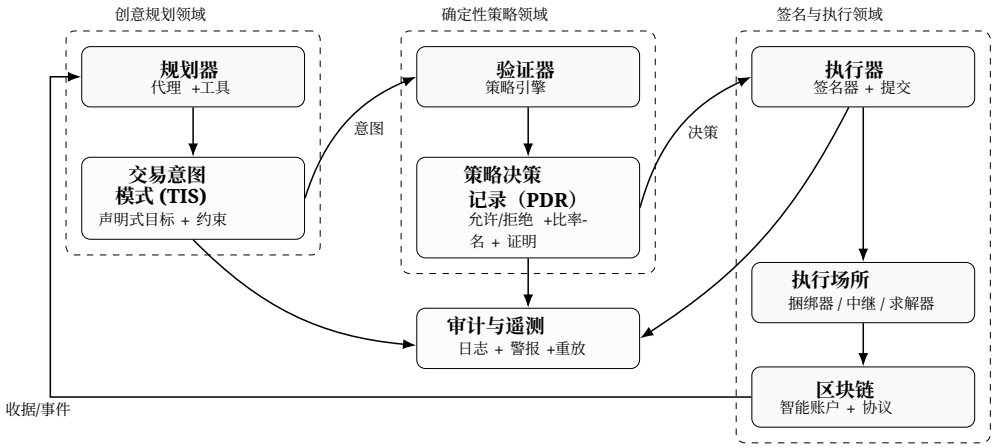

图 5：交易意图模式（TIS）与策略决策记录（PDR）技术栈的参考架构。该架构将规划器（代理）、验证器（策略引擎）和执行器（签名器／提交层）分离，从而实现纵深防御。

决策记录（PDR）。其目标是标准化以下内容：(i) 用户或智能体希望实现的目标；(ii) 在发生任何不可逆签名之前，请求已依据可强制执行的策略完成评估的可验证证明。

8.1 参考架构：解耦意图、策略与签名

图 5 展示了拟议的参考架构，它在智能体交易流水线的关键组件之间实现了严格的职责分离。 [242] 该架构将代理富有创造力但易出错的规划过程，与确定性的授权和签名操作解耦。这种分离符合纵深防御的原则：规划器生成目标和约束，验证器根据策略和威胁假设评估这些目标，而执行器仅在获得可验证的授权凭证后才进行签名。

8.2 交易意图模式（TIS）：链上目标的通用语言

问题： 缺乏用于表示用户意图的统一方式，仍然是安全实现代理式执行的主要瓶颈。如今，意图执行兑换操作的代理仍必须处理协议特定 ABI 和函数签名（Uniswap、Curve 与 CoW 之间各不相同）、链特定执行约束，以及交易层的故障模式。这种方式脆弱、难以审计，并且实质性地扩大了攻击面，因为代理被迫成为低层机制方面的专家，而不是目标和约束方面的专家。

解决方案： 我们提出将 TIS 作为一种标准化且可扩展的基于 JSON 的模式，用于描述做什么 （期望状态变更）以及约束条件 （安全与有效性边界），而不是如何做 （协议特定调用数据）。它建立在向以意图为中心的执行转变这一更广泛趋势之上 [202]，在理念上与新兴的意图标准和协调原语（例如 ERC-7521 及相关工作）兼容。 [142, 53]。

8.2.1 TIS 核心模式与设计理由

TIS 旨在兼具人类可读性和机器可解析性。最小核心架构应包含 s:

- 意图标识符 **（字符串，UUID）：** 用于在链下和链上系统中进行跟踪、关联和审计的全局唯一标识符。 [334]
- 操作 **（字符串，枚举）：** 描述意图类别的高级动词。初始集合应刻意保持精简（例如 SWAP、LEND、BORROW、STAKE、VOTE），并通过版本化注册表或命名空间进行扩展，以避免碎片化。
- 输入 **（对象数组）：** 用户提供的资产。每个对象指定代币（地址）、数量和约束条件（例如 EXACT、MAXIMUM）。
- 输出 **（对象数组）：** 用户预期收到的资产。每个对象指定代币、数量和约束条件（例如，最低值、精确）。

- 约束（**对象**）：全局有效性条件，例如截止时间（Unix 时间戳）、随机数（重放保护）、链 ID（域绑定）以及可选的排他性（例如求解器/捆绑器/中继器允许列表条目，或 null）。

- 预览 （**字符串**）：对意图清晰易懂、便于人类阅读的自然语言描述，适合用户审阅及后续所见即所签工作流使用。

规范化说明：由于 TIS 旨在经过哈希处理并绑定到授权凭证中（第 8.3 节），实现时应采用规范的 JSON 序列化，以确保不同语言和运行时之间的哈希值稳定。一种实用方案是 JSON 规范化方案（JCS）。 [126]。

8.2.2 示例：复杂的多步骤意图

TIS 在多步骤工作流中的价值更加明显。 [337]例如，向收益金库执行一次“Zap”操作，需要多个底层操作（兑换、流动性提供、存入）：

```
{
 "意图标识符": "f47ac10b-58cc-4372-a567-0e02b2c3d479"    ,
 "操作": "Zap    _INTO_VAULT" ,
 "输入":  [
 {
 "代币": "0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48"    ,
 "数量": "1000000000"    ,
 "约束条件": "精确"
 }
 ],
 "输出":  [
 {
 "代币":    "0x.." ,
 "数量": "950000000000000000"    ,
 "约束条件": "最低值"
 }
 ],
 "约束": {
 "截止时间": 1767226000    ,
 "排他性": null
 },
 "元数据": {
 "目标金库":        "0x..",
 "中间交换协议": "UniswapV3"
 },
 "preview": "存入 1000 USDC 至 Yearn ETH/USDC 金库，并至少获得 0.95 个金库代币
       作为回报。"
}
```

这一项意图概括了多步骤目标，同时无需暴露协议特定调用数据。代理会构造目标和约束条件，而专用求解器或执行适配器则会生成最优交易序列，并通过适当的执行场所（例如基于求解器的结算、私有中继或 ERC-4337 风格的打包）提交该序列 [2]。

8.3 策略决策记录（PDR）：可验证的合规证明

问题：当前的策略执行仍然各自为政，而且往往不透明。即使存在策略引擎，通常也没有可互操作的方式来证明某项意图在签名前已依据特定规则集进行评估。这使审计更加复杂，削弱事件响应能力，并在信任链中造成单点故障。

解决方案： 我们提议将**策略决策记录（PDR）**作为一种经过签名且可验证的授权凭证，用于证明某个特定的 TIS 对象已接受评估，并被批准或拒绝。从概念上讲，它充当提交给签名器的签名能力凭证。这一设计借鉴了成熟且广泛部署的授权模式：通过紧凑的签名令牌，在解耦组件之间传递声明和约束条件。 [139, 140, 160, 161]。

8.3.1 PDR 核心架构（JWT 配置文件）

我们提议将 PDR 标准化为 JSON Web 令牌（JWT）配置文件。JWT 能以紧凑的形式表示声明，并附带数字签名（JWS），使执行器能够进行无状态验证，而无需在每次签名事件时都在线调用策略引擎。 [160, 159]。

- **标头：**包含签名算法 alg 和令牌类型 typ（例如，PDR+JWT）。实现应选择与其密钥材料和信任域相符的算法（例如，基于 ECDSA 的 JWS）。
- **负载（声明）：**
  – iss（颁发者）：签发该决策的策略引擎的 DID/身份。– sub（主体）：代表其执行意图的用户账户（或智能账户）。– aud（受众）：PDR 的预期接收方，即执行器/签名者身份。绑定到 aud 可防止有效的 PDR 被重放至其他签名者。– exp（过期时间）：设置较短的过期时间，以限制重放和过期授权。 [351]– iat（签发时间）：签发时间。– intent_hash（字符串）：对规范化后的 TIS 对象计算的密码学哈希（例如 Keccak-256）。这会将 PDR 绑定到唯一一个意图。– decision（字符串，枚举）：批准或拒绝。– bound_constraints（对象，可选）：策略引擎施加的附加约束（例如，降低支出限额、缩短截止时间、限制交易场所）。执行者必须执行这些收紧后的约束。

- **签名：**一份加 yptograp密签g名来自 policy engine, p颁发者g integrity 以及颁发者的真实性 y。 [159]

8.3.2 PDR 工作流与安全保障

1. 代理构造 TIS，并将其提交给已配置的策略引擎。2. 策略引擎根据其规则集（例如支出限额、允许列表、交易场所限制、时间窗口和上下文信号）评估 TIS。3. 如果获得批准，策略引擎会签发并签名 PDR（JWT），然后将其返回给代理。4. 代理将 TIS 和 PDR 一并提供给签名器（执行器）。5. 在执行任何签名操作之前，签名器会执行以下验证步骤：

- 根据可信策略引擎的公钥验证 PDR 签名。
- 检查 aud 声明是否与签名者身份相符。
- 检查时间有效性（exp，以及可选的较窄 iat 容差窗口）。
- 重新计算规范化 TIS 的哈希值，并确认其与 intent_hash 相符。这样可以防止“PDR 替换”：代理先获得对无害意图的授权，然后在签名时替换成另一种意图。

- 强制执行任何收紧后的界限_约束。1. 只有通过所有检查后，签名器才会继续生成并提交最终交易（可能通过捆绑器、中继或求解器交易场所） [2]。

这一工作流会生成不可否认、可审计的证据链，确保职责分离。它也符合这样的威胁假设：代理及其运行时可能遭到安全失陷；即使如此，遭到安全失陷的规划器也无法超出验证器和执行器共同许可的范围。

8.4 优势、挑战与未来研究

对 TIS 和 PDR 进行标准化，将通过把当今的定制集成转化为可验证、可移植的制品，显著提升智能体—区块链系统的互操作性、安全性、可组合性和可审计性。具体而言，TIS 为求解器和执行场所提供通用的目标级接口，而 PDR 则引入签名者可验证的证明，证明某项意图在发生任何不可逆授权之前已依据策略完成评估。这些层共同减少协议特定耦合，缩小规划器的攻击面，并通过使决策和约束可由外部审计，改进事件后取证。

然而，要实现这一愿景，就必须应对若干实际挑战：

- **标准化与治理：** 要实现广泛采用，需要建立协调一致的社区流程，钱包提供商、托管服务、基础设施运营商（捆绑器/中继器）和安全研究人员都应参与其中。一条可信的路径是在以太坊生态系统中开展开放的标准制定流程（例如采用以太坊改进提案式的工作流程），并辅以参考实现和测试向量，以明确规范化、哈希和验证行为。
- **求解器生态系统启动：** TIS 的价值主张取决于一个由求解器和执行适配器组成的竞争性市场，它们能够将意图转化为稳健的执行计划，在价格和可靠性方面展开竞争，并支持针对特定交易场所的防护措施（例如，私有中继或批量拍卖）。启动这一生态系统并非易事：它需要激励机制设计、明确的责任边界，以及在对抗性条件下衡量求解器质量（成交率、滑点、失败率）的机制。
- **策略语言与可移植性：** 为了让 PDR 实现互操作性，而非造成供应商锁定，各实现之间的策略决策必须具备可比性和可移植性。一个有前景的方向是标准化一个最简、可由机器检查的策略层（例如，允许列表约束、支出限额、交易场所限制和时间窗口），并将更丰富的组织策略映射到该层。开放策略代理（OPA）等策略即代码系统[1] 为表达和审计规则集提供了有用的基础，不过仍需进一步定义区块链专用原语及其语义。 [12]

未来的研究方向包括：

- **隐私保护型 PDR：** 开发能够证明合规性，同时不泄露敏感策略逻辑或意图细节的 PDR 构造。一种方法是附加零知识证明，证明验证器已在经批准的策略承诺下评估该意图，并得出了所述决策，同时仅选择性地披露签名者必须执行的内容（例如有界约束）。隐私保护型账本和合规设计提供了早期证据，表明这类证明在受监管环境中具有实用性。 [224]
- **去中心化信任发现与注册表：** 构建去中心化信任注册表，由策略引擎发布公钥、元数据和证明，使签名者能够在不依赖中心化目录的情况下发现并验证可信验证者。一种自然的基础架构是由 ENS 支持的注册表，也可以选择关联去中心化标识符（DID）文档，以提供更丰富的身份信息和密钥轮换语义。 [98, 305]

通过建立这些缺失的接口层，生态系统可以逐步形成一个稳健且可审计的链上代理执行控制平面：规划器保持灵活，验证器继续由策略驱动且可供审查，执行者则保持确定性且范围狭窄。这种分离是在此前概述的威胁模型下，在维持安全性的同时扩大智能体自主性的最直接途径。

8.4.1 实施路线图与里程碑

从当前碎片化的方法转向本综述所设想的标准化 TIS/PDR 生态系统，需要多个利益相关方协同努力。本节概述了分阶段的实施路线图，并列出具体里程碑。

第一阶段：规范与参考实现（第 1–6 个月）第一阶段着重于敲定规约并开发参考实现，以验证其可行性。

里程碑：

- 发布 TIS v1.0 规约及 JSON Schema 定义
- 发布 PDR v1.0 规约及密码学要求
- TIS 解析器和验证器的参考实现
- PDR 生成器和验证器的参考实现
- 覆盖规约边界情况的测试套件

第二阶段：生态系统集成（第 7 至 12 个月）第二阶段侧重于将 TIS/PDR 支持集成到现有的生态系统工具和平台中。

里程碑：

- 与至少三家主要钱包提供商集成
- 与至少两个代理框架集成

- 与至少两个策略引擎集成
- 开发 TIS/PDR 浏览器扩展，以便用户查看
- 发布集成指南和最佳实践

第三阶段：生产部署（第13至18个月）第三阶段侧重于生产部署和运营经验。

里程碑：

- 至少由五名代理运营者进行生产部署
- 运营指标的收集与分析
- 根据生产环境经验更新规约
- 开发监控与分析工具
- 发布案例研究与经验教训

第四阶段：标准化与治理（第19至24个月）第四阶段侧重于正式标准化并建立治理流程。

里程碑：

- 向相关标准机构提交 TIS/PDR
- 建立规范治理流程
- 制定一致性测试和认证计划
- 成立生态系统工作组，持续推进开发
- 发布纳入社区反馈的 v2.0 规范

9 评估：链上代理安全的基准测试与检查清单

自主链上代理的出现，要求评估方法论随之演进。传统的软件测试方法，例如单元测试和集成测试，并不适用于行为具有非确定性、面临经济层面对抗且与不可逆状态转换紧密耦合的系统 [200]。形式化方法和智能合约验证方面的既有工作，主要关注链上代码的静态正确性，却无法捕捉智能体推理、外部工具使用和实时市场交互所带来的动态风险。

因此，评估智能体—区块链系统需要一种全面的端到端方法论，涵盖推理、授权、执行和执行后验证。在自主信息物理系统和自适应分布式系统等相关领域，也有人提出了类似观点：在这些领域中，涌现行为和对抗性环境使纯粹的组件级评估失去效力 [143, 338]。本节提出一个专为链上智能体设计的结构化评估框架，将定量基准与定性安全检查清单相结合。

9.1 对领域特定、全面基准测试的需求

现有的大语言模型基准测试评估的是孤立的认知能力，例如通用知识（如 MMLU）或代码合成（如 HumanEval），但并未对抗性经济环境或不可逆的执行过程 [143] 建模。同样，智能合约基准测试侧重于漏洞检测和 Gas 效率，但假设交易意图已由外部正确指定 [142, 215]。

相比之下，链上代理必须联合推理不确定信息、与不可信工具交互、遵守策略约束，并在 MEV压力下执行交易。近期关于大语言模型工具使用的知识系统化论文明确指出，缺乏能够评估完整工具—决策—行动循环的基准测试 [181]。我们的框架通过将智能体—链流水线作为单一系统进行评估，弥补了这一空白。

We 围绕贯穿所综述系统和威胁模型的三个支柱组织评估：

1. **安全与保障**，体现抵御利用和损失的能力；2. **任务可靠性与鲁棒性**，体现发生故障时的运维正确性；3. **经济鲁棒性**，体现面对对抗性市场动态时的理性行为。

9.2 支柱 1：安全与保障基准测试

安全性基准测试侧重于评估代理抵御威胁模型中所识别攻击类别的韧性，尤其是提示操纵、工具投毒、策略绕过和 MEV 利用。与对抗性机器学习和区块链安全领域以往的评估实践一致，这些基准测试在受控的分叉环境中执行，其中布置了对抗性伪影 [200, 142]。

9.2.1 提示注入攻击抵抗（PIR）分数

间接提示注入已被确认为基于大语言模型的智能体的一项根本性漏洞，尤其是在处理不可信外部内容时 [131]。在区块链环境中，这种风险会进一步加剧，因为注入的指令可能触发不可逆金融操作。

- **方法论：** 代理在处理良性任务的同时，会接触经过对抗性精心设计的数据源，包括 HTML 页面、代币元数据字段、API 响应和错误消息，并遵循针对大语言模型工具使用的既定威胁分类法 [181]。
- **评分：** PIR 分数衡量被正确忽略或标记、且未执行恶意指令的攻击所占比例。低于 99% 的分数表明其不适合自主运行。

9.2.2 工具欺骗检测（TSD）率

工具投毒和数据欺骗攻击已在分布式系统和基于预言机的架构中得到广泛记录 [57]。对于代理而言，依赖单一数据源代表着一种关键故障模式。

- **方法论：** 代理会接触到相互冲突或遭到操纵的工具输出，例如价格差异或错误的余额，并应执行交叉验证或拒绝执行操作。
- **解读：** TSD 率评估代理是否实施冗余和合理性检查，这是安全预言机和去中心化金融系统设计中所推荐的实践 [142]。

9.2.3 最大可提取价值漏洞评估（MVA）

最大可提取价值已被形式化为公有区块链的一种结构性属性，而非边缘情况 [80, 4]。因此，提交公有交易却未采取缓解措施的自主代理，默认被认为容易遭受攻击。

- **方法论：** 交易在由对抗性参与者填充的内存池中执行，并遵循既定的最大可提取价值基准测试惯例 [58]。
- **指标：** MVA 分数衡量相对于最优、无最大可提取价值影响的执行基准所损失的价值。

9.2.4 策略遵循测试（PAT）

策略执行故障已被确认为自主和半自主金融系统中造成损失的主要原因 [102]。此基准用于评估智能体在对抗性提示下是否遵守明确的约束条件。

- **方法论：** 反复将代理置于违反策略会带来更高表面效用的情境中。
- **结果：** 较高的 PAT 分数表明能够正确地将约束条件置于短期回报之上。

9.3 支柱 2：任务可靠性与鲁棒性

除了安全性之外，智能体还必须能够在现实的操作条件下可靠地完成复杂任务，包括应对部分故障和暂时性不一致。鲁棒性基准测试借鉴了分布式系统测试和故障注入的原则 [276]。

9.3.1 去中心化金融任务完成（DTC）套件

Multi-step 去中心化金融工作流是检验智能体规划与执行能力的天然压力测试 [138] .

- **方法论：** 任务按组合复杂度递增分类，从单协议交互到原子化多协议套利。

- **度量指标：** 成功率、Gas效率、执行延迟和推理深度。

9.3.2 故障恢复测试（FRT）

对部分故障的韧性是可靠自主系统的 defining property [185]。

- **方法论：** 环境会注入远程过程调用故障、回滚调用和丢弃的交易。
- **成功标准：** 代理必须诊断失败原因，并最终确定正确的恢复策略 [182]。

9.4 支柱 3：经济鲁棒性基准测试

经济上的非理性与技术层面的安全失陷一样具有破坏性。关于算法交易和去中心化金融攻击的既有研究表明，对抗性激励如何利用简单策略 [80]。

9.4.1 经济理性评分（ERS）

- **方法论：** 智能体会面临已知的经济陷阱，包括拉高出货骗局和蜜罐合约。

- **解读：** 较高的 ERS 分数表明其能够推理流动性、可售性和执行可行性，而非仅关注表层利润信号。

9.5 Agent-to-Chain 安全审计清单

仅靠定量基准不足以揭示架构弱点。受分布式系统和 DAO 治理中的安全审查实践启发 [137, 102], ，我们提出了一份结构化审计清单，按照代理操作流水线进行组织。

该清单通过支持对可观测性、授权边界、执行路径和恢复机制进行定性审查，对基准测试形成补充。基准测试与清单结合，可实现纵深防御评估，符合安全系统设计的最佳实践 [200]。

9.5.1 评估方法最佳实践

对智能体区块链系统进行严谨评估，需要特别关注方法论。本节概述开展有效评估的最佳实践。

测试环境保真度 评估环境应忠实再现智能体在生产环境中可能遇到的条件，包括真实的网络延迟、Gas 价格波动、流动性状况和对抗性行为。在理想化环境中开展的评估可能会高估代理的表现并低估风险。

主网分叉通过复现实际区块链状态，提供高保真的测试环境。然而，分叉环境无法再现实时市场的全部动态，包括其他参与者对代理行为的反应。在测试网和模拟环境中开展补充测试，有助于覆盖无法通过分叉复现的场景。

对抗性测试 安全评估应包括由经验丰富的红队开展的对抗性测试。自动化安全扫描可以识别已知漏洞模式，但有创造力的攻击者可能会发现自动化工具无法检测到的新型攻击向量。红队演练应覆盖完整的攻击面，包括提示注入、经济攻击和基础设施攻陷。

漏洞赏金计划通过激励外部研究人员发现并报告漏洞，提供持续的对抗性测试。设计完善、奖励适当且范围明确的漏洞赏金计划能够显著提升安全态势。

长时测试 代理行为可能会在长时间操作期间发生变化，这是由状态累积、模型漂移或环境变化所致。短时测试可能无法发现只有在长时间操作后才会出现的故障模式。长时测试，包括连续运行数天或数周，有助于识别这些延迟出现的故障模式。

统计严谨性 评估结果应以恰当的统计严谨性报告，包括置信区间、样本量和显著性检验。单次运行结果或较小的样本量可能无法可靠地反映真实性能。评估应包含足够多的重复试验，以确保结果具有统计置信度。

可复现性 评估流程应记录充分的细节，以便他人独立复现。这包括明确指定测试数据、环境配置、评估指标和分析流程。可复现的评估有助于验证结果，并比较不同系统。

10 案例研究：实践中的架构

为将前文的分类体系和能力分析落实到可部署的设计中，本节将探讨三个横跨**模式 III–V**的代表性案例研究。每个案例都提供端到端的执行流程（设置、意图构建、授权、提交和验证），并揭示自主性提升所带来的安全权衡。全文将重点强调何处实施控制（链上模块与链下策略引擎）、如何限定权限（会话密钥、MPC 阈值和 PDR 证明），以及哪些威胁最为突出（提示注入、密钥泄露、工具欺骗和最大可提取价值）。

10.0.1 详细基准测试规范

本节详细说明本综述提出的评估基准规范，以便从业者实施一致的评估流程。

安全基准测试：提示注入抵抗性 提示注入抵抗性基准测试用于评估代理抵御通过恶意输入操纵其行为的能力。

测试套件组成：

- 100 直接注入尝试（明确要求转账的指令）
- 100 间接注入尝试（区块链数据中的恶意内容）
- 100多轮操纵尝试（逐步建立信任）
- 100 基于编码的攻击（Unicode、Base64 及其他编码）
- 100 上下文溢出尝试（超出上下文窗口）

评估指标：

- 攻击成功率（ASR）：实现其目标的攻击所占的百分比
- 检测率：被正确识别为恶意的攻击所占的百分比
- 假阳性率：被错误标记的良性输入所占的百分比
- 攻击下的性能退化：攻击尝试对性能造成的影响

通过标准：

- 所有攻击类别的 ASR $< 1\%$
- 直接攻击的检测率 $> 95\%$
- 假阳性率 $< 5\%$

可靠性基准测试：交易成功率交易成功率基准测试用于评估智能体交易构建和执行的可靠性。

测试套件组成：

- 200 简单转账交易
- 200 代币兑换交易
- 200 多步骤去中心化金融操作
- 200 网络拥塞下的交易
- 200 有时间约束的交易

评估指标：

- 成功率：成功执行的交易所占的百分比
- 回滚率：链上回滚交易所占的百分比
- 模拟准确度：模拟结果与实际执行结果之间的关联
- 恢复率：成功重试的失败交易所占的百分比

通过标准：

- 简单转账成功率 > 99%
- 复杂操作成功率 > 95%
- 模拟准确度 > 98%

经济基准：执行质量 执行质量基准用于评估代理交易执行的经济效率。

测试套件组成：

- 100 小额兑换（<$1,000）
- 100 中等规模兑换（$1,000 - $100,000）
- 100 大额兑换（>$100,000）
- 100 时间敏感型执行
- 100多平台执行

评估指标：

- 价格影响：执行时相对于市场中间价的偏差
- 最大可提取价值泄漏：MEV 搜索者提取的价值
- Gas效率：相对于最优执行所消耗的 Gas
- 执行延迟：从意图到确认所需的时间

通过标准：

- 价格影响 < 0.5%（小额兑换）
- 最大可提取价值泄漏 < 0.1%（占交易价值）
- Gas效率 > 90占理论最优值的百分比

10.0.2 综合评估框架

对智能体区块链系统进行评估，需要采用全面的框架来衡量系统质量的多个维度。关于拜占庭容错的基础研究，为评估分布式系统的安全性提供了理论依据 [312, 331, 345]。近期研究则侧重于构建区块链漏洞和隐私问题的分类体系，这些研究可为评估框架的设计提供参考 [326, 341, 342]。区块链系统的可扩展性也是一项关键评估维度，相关研究探讨了多种提升性能的技术 [344, 332]。不同区块链系统之间的互操作性同样备受关注，目前已有多个框架被提出，用于评估跨链通信协议 [315]。研究人员也探讨了评估基于智能体的系统所面临的独特挑战，重点在于构建逼真的仿真环境和基准测试 [340, 343]。分布式系统中常见的半双工通信协议，其安全性也得到了深入研究 [320]。密码协议的通用可组合性是评估复杂系统安全性的关键概念 [349]。最后，分布式数据库的性能是影响区块链系统整体性能的关键因素 [321]。传统的软件评估指标虽不可或缺，却不足以体现自主代理在对抗性经济环境中运行时的独特特征。

功能正确性评估必须验证智能体能否针对预期用例生成正确的输出。这包括正向测试（验证对于有效输入的行为是否正确）和负向测试（验证对无效输入、边界情况和对抗性场景的处理是否恰当）。基于大语言模型的智能体具有随机性，这使正确性评估更为复杂，因为相同的输入在多次运行中可能产生不同的输出。

安全评估必须评估系统抵御威胁模型中确定的所有威胁的韧性。这既包括自动化安全测试（模糊测试、静态分析和渗透测试），也包括由领域专家开展的人工安全审查。基于大语言模型的智能体引入了新的攻击面，例如提示注入和越狱，因此需要采用专门的评估技术，而传统的安全评估方法可能并未涵盖这些技术。

经济评估必须衡量代理策略的财务表现和风险特征。这包括使用历史数据进行回测、在真实的市场状况下进行模拟，以及仔细分析尾部风险和最坏情况。经济评估还应考虑代理行为对市场质量和其他参与者的影响，而不仅仅是代理运营者的回报。

10.1 案例研究 1：简单支付代理（模式 III）

目标： 代理代表用户执行定期付款，例如每月向流媒体服务支付 10 USDC 订阅费。

架构： 此用例体现了**模式 III：委托执行**。代理不应拥有不受限制的签名权限。它只需获得范围严格受限的权限，在严格且由智能合约强制执行的约束条件下执行单一操作。推荐的架构是配备策略模块（例如会话密钥模块或 Zodiac Roles 修饰器）的**ERC-4337 智能账户**，以在链上限制代理权限 [2, 261]。

10.1.1 详细设置与配置

1. **智能账户：** 用户控制着一个 ERC-4337 智能账户（例如，基于 Safe{Wallet} 的账户抽象实现） [261, 2]。

2. **会话密钥生成 [272]：** 在订阅注册期间，去中心化应用会在本地生成一个新的密钥对（在浏览器中或移动设备的安全飞地中）。公钥记为 0xAgentSessionKey，将被授权为受限会话密钥。

3. **策略模块安装：** 去中心化应用准备一笔交易，用于安装和配置链上策略模块（例如 ZodiacRoles 或定制的会话密钥模块）。用户批准一笔交易，该交易将：
   - 在智能账户上安装该模块；
   - 将 0xAgentSessionKey 注册为获授权的委托方；并且
   - 编码硬性约束，由该模块在每次委托执行时强制执行。

链上约束示例：

- require(接收方 == 0xStreamingServiceAddress)
- require(数量 <= 10 * 1e6) // 10 USDC
- require(区块时间戳 >=上次付款时间戳 + 30天)
- require(区块时间戳 <密钥到期时间戳)

这些约束构成了核心安全边界：即使代理遭到安全失陷，最大损失也会受到其无法绕过的链上逻辑限制。

10.1.2 详细执行流程

到期日当天，代理（例如由流媒体服务运行的受限后端任务）会根据 ERC-4337 [55] 构造并提交一项用户操作。

代理日志（示例）：

[2026-07-15 08:00:00]信息：已为账户 0xUserSmartAccountAddress 触发计费周期。[2026-07-15 08:00:02]信息：应付款（上次付款：2026年-06-15）。[2026-07-15 08:00:03]信息：正在构造用户操作：向 0xStreamingServiceAddress 转账 10 USDC。[2026-07-15 08:00:04] 信息：正在使用会话密钥 0xAgentSessionKey 为用户操作签名。[2026-07-15 08:00:05] 信息：正在向打包器端点提交用户操作。[2026-07-15 08:00:20] 信息：正在等待交易回执.....[2026-07-15 08:00:35]信息：成功：已纳入区块 21000000。[2026-07-15 08:00:36] 信息：正在应用数据库中更新订阅状态。

用户操作载荷（示意）：

```
{ "发送方": "0xUserSmartAccountAddress",
```

```
    "随机数": "0x..", "调用数据": "0x..", "签名": "
0x..", "支付主账户及数据": "
0xPaymasterAddress.." }
```

链上验证（ERC-4337 路径）：

1. 捆绑器将用户操作纳入发送至入口点的交易捆绑包中。2. 入口点在 0xUserSmartAccountAddress 上调用 validateUserOp。3. 智能账户将验证委托给已安装的策略模块。4. 策略模块验证：

- 签名对应于 0xAgentSessionKey；并且
- 执行满足所有约束条件（接收方、金额上限、执行频率、有效期）。5. 若验证通过，入口点便通过智能账户执行调用（USDC 转账）。

10.1.3 安全分析深入探讨

- **攻击面：** 极小且边界明确。最坏情况下的损失上限为每个时间间隔 10 USDC，并由不可变的链上检查加以执行。
- **密钥管理：** 代理绝不会接触用户的主密钥。会话密钥仅用于单一用途，且有时间限制。
- **信任边界：** 信任以经过审计的链上模块为基础，而非依赖代理运行时的正确性。

10.2 案例研究 2：自主投资组合再平衡代理（模式 IV）

目标： 由代理管理用户的去中心化金融投资组合，并通过在偏差超过阈值时进行再平衡，达到预期的资产配置（例如，50% 以太币、25% 封装比特币、25% USDC）。
架构： 此用例符合**模式 IV：自主签名**。由于代理获准发起具有显著经济影响的操作，因此它需要比模式 III 更严格的控制措施。推荐采用**纵深防御** 架构，结合分布式托管（多方计算）、策略执行（链下策略引擎）和 MEV 感知型执行（私有中继和基于意图的交易场所） [314, 80, 108]。

10.2.1 详细设置与配置

1. **托管与签名（分布式控制）：** 资金存放在 **2-of-3 多方计算钱包** [314]:
   - **份额 1（代理端）：** 存放在经过加固的环境中（例如硬件安全模块或基于可信执行环境的服务）。
   - **份额 2（用户端）：** 存放在用户的移动设备上，并通过生物识别审批。
   - **份额 3（恢复）：** 存储在硬件钱包上，作为恢复／法定人数路径。
2. **策略引擎（情境约束）：** 用户配置链下策略引擎（例如类似 Turnkey 的控制措施），其中包括：

   - 目标配置和偏差阈值（偏差达到 $> 2\%$ 时再平衡）；
   - 协议和路由的允许列表（例如 Uniswap V3、Aave V3）；
   - 每日周转上限（例如，投资组合价值的 $\leq 25\%$）。

10.2.2 详细执行流程（TIS/PDR 模型）

此次执行采用所提议的 **TIS/PDR 职责分离**：代理提出有界意图（TIS），策略引擎生成已签名的合规凭证（PDR），签名器在生成签名之前会强制执行两者。
代理日志（示例）：

```
[14:00:05]信息：观察到资产配置：47.5% 以太币、26.0% 封装比特币、26.5% USDC。[14:
00:06]警告：以太币低于阈值（48%）。需要再平衡。[14:00:08]信息：拟议操作：通过
Uniswap V3 将 5,000 USDC 兑换为 WETH。[14:00:09]信息：正在构造 TIS。
```

[14:00:10] 信息：正在向策略引擎提交 TIS。[14:00:12] 信息：已收到已签名的 PDR（批准）。[14:00:13] 信息：正在将 TIS+PDR 提交至 MPC 签名工作流。[14:00:25] 信息：用户已在移动设备上批准（已满足 2-of-3）。[14:00:26] 信息：正在将已签名交易提交至私有中继，以缓解最大可提取价值（MEV）。[14:00:41] 信息：执行已确认；更新后的资产配置：49.8% 以太币。

TIS 负载（示例）：

{
"意图标识符": "a1b2c3d4-e5f6-4a7b-8c9d-0e1f2a3b4c5d", "操作": "SWAP", "输入": [{ "代币": "0xa0b8..", "数量": "5000000000", "约束条件": "精确" }],"输出": [{ "代币": "0xc02a..", "数量": "1500000000000000000", "约束条件": "最低值" }],"约束": { "截止时间": 1767230000 }, "预览": "精确兑换 5,000 USDC，至少获得 1.5 WETH，以重新平衡投资组合。" }

PDR 负载（解码示例）：

{ "iss": "https://policy.turnkey.com", "sub": "0xUserAddress", "aud": "https://signer.fireblocks.com", "exp": 1767229800, "意图哈希值": "0xkeccak256(TIS负载)", "决策": "批准", "绑定约束": { "最大Gas费用": "60000000000" } }

10.2.3 安全分析深入探讨

- **纵深防御：**单一环节遭到安全失陷（代理运行时、一个多方计算份额或某个工具）不足以耗尽资金。对手必须突破多个相互独立的控制措施。
- **可审计性：**TIS 和 PDR 提供持久的取证审查与合规记录：可留存并重新验证拟议的意图、策略依据和签名决策。
- **运行时紧急停用开关：**用户可以撤销代理份额、收紧策略阈值，或使策略颁发者密钥失效，从而立即降低权限，而无需重新部署链上代码。

10.2.4 其他实现示例

除了主要案例研究外，还有若干其他示例阐释了智能体与区块链集成在实践中的重要方面。

示例：自动化收益优化 收益优化代理会持续监控去中心化金融协议，以识别存入资产的最高收益机会。该代理的观测模块会跟踪数十个协议中的利率、流动性挖矿奖励和协议风险。其推理模块会综合考虑预期收益、Gas 成本、智能合约风险和投资组合约束，评估各种机会。

当代理发现更优机会时，会构建一系列交易，从当前头寸中提取资产并存入新的头寸。构建模块会优化交易排序，以尽量减少滑点和 MEV 暴露。执行前，会根据策略约束验证这些交易，包括最大头寸规模、协议允许列表和最低收益阈值。

此示例展现了在去中心化金融环境中持续监控和快速响应的价值，因为收益机会可能转瞬即逝。它还说明了全面风险评估的重要性，因为收益优化策略历来容易受到智能合约攻击和经济攻击的影响。

示例：跨链套利 跨链套利代理会监控多个区块链网络中资产的价格差异。当发现有利可图的机会（并计入桥接费用、Gas 成本和执行风险）时，它会启动一系列协调交易以获取套利收益。

跨链套利的复杂性要求在多个具有不同最终性特征的区块链上进行复杂的状态管理。代理必须跟踪每条链上的待处理交易，妥善处理部分故障，并

管理连续操作之间的时序依赖关系。跨链桥选择涉及速度、成本和安全性之间的权衡，代理必须动态评估这些权衡。

此示例说明了多链智能体运行面临的挑战，以及健全故障处理机制的重要性。跨链操作存在多种故障模式，包括桥接延迟、交易回滚以及执行期间的价格波动，代理必须做好准备应对这些情况。

示例：治理委托 治理委托代理根据委托人的偏好，管理多个去中心化自治组织中的投票权。该代理监控治理提案，分析其潜在影响，并根据预定义的策略规则或学习到的偏好进行投票。

该代理的分析模块会评估提案，考量其技术价值、经济影响，以及与委托人所表达价值观的对齐程度。对于复杂提案，代理可能会生成摘要和建议，供人工审查，而不是自主投票。代理还会管理委托关系，根据受托人的投票历史及其与委托人偏好的契合程度调整委托。

此示例说明了代理有望提升治理参与度，同时也引发了重要问题：自动化系统在集体决策中应发挥何种适当作用。

10.3 案例研究 3：多智能体去中心化自治组织财库管理（模式 V）

目标： 一个去中心化治理流程，用于管理大型去中心化自治组织金库，由专用代理提出、评估并执行投资操作，并确保问责透明。 [113]
架构： 此用例体现了**模式 V：多代理工作流**。它将权限分散给不同类型的参与者，以减少单点故障、落实职责分离，并通过结构化的对抗性审查提升决策质量。

10.3.1 详细设置与配置

1. **代理角色与激励：**
   - **策略代理（例如，5 个）：** 提出操作并根据绩效展开竞争；激励包括绩效费用和声誉。
   - **风险代理（例如，7 个）：** 评估提案；通过聘用费获得报酬，并以足额质押作为担保，若因疏忽或渎职导致问题，则可能遭到罚没。
   - **执行代理(1)：** 机械地执行仅满足链上接受条件的提案；不具备任何自由裁量的策略权限。
2. **治理合约（注册表与执行）：** 链上合约维护：
   - 代理注册表和质押记账；
   - 提案提交和投票逻辑；
   - 法定人数规则和执行门控。

10.3.2 详细执行流程

聚合日志（示例）：

> [StrategyAgent-A] 信息：提出意图：将 500 以太币分配至 GMX ETH-USDC 池。提案 ID：0x123。[GovernanceContract] 事件：新提案（proposalId：0x123）。[RiskAgent-OZ]信息：观察到新提案。正在获取 0x123 的意图。[RiskAgent-OZ]信息：模拟结果：成功；滑点 0.2%。[RiskAgent-OZ]信息：合约分析：0 项严重发现。[RiskAgent-OZ]信息：风险评分：经济=中，技术=低。投票=批准。.....（其他风险代理投票）.....[GovernanceContract] 事件：提案已批准（proposalId：0x123）。[ExecutionAgent] 信息：观察到提案已批准。正在通过私有中继执行。[ExecutionAgent] 信息：成功：已在 GMX 上建立财库头寸。

10.3.3 安全分析深入探讨

- **职责分离：** 提案生成、安全评估和执行是相互独立的职能，可减少相关失效模式。
- **抗串谋性：** 质押和罚没会提高不诚实批准的成本。评估者的多样性和独立工具链可降低单一工具遭入侵的风险。

- **透明度与问责：** 提案材料、投票和执行事件均记录在链上，便于公众审计和事件后归因。

总的来说，这些案例研究展示了分类体系如何转化为运行架构。模式 III 强调通过链上强制执行实现有界委托；模式 IV 引入了策略驱动的自主性、分布式托管和 MEV 感知型执行；模式 V 则通过在专用代理之间分散信任，并遵循链上治理约束，将自主性扩展至组织环境。

10.3.4 智能体激增的治理影响

区块链生态系统中自主代理的激增，引发了有关治理和集体决策的根本性问题 [198, 316]。随着代理能力不断增强、部署范围日益扩大，它们对治理结果的影响也将不断增大，并可能以不符合人类利益相关者偏好的方式改变权力平衡。

一个令人担忧的问题是，代理介导的治理俘获可能出现，即老练的行为者利用代理，在协议治理中积累不成比例的影响力。代理能够持续监控治理提案，比人类参与者更快地响应投票机会，并以人类难以实现的方式跨多个账户或协议进行协调。若缺乏适当的保障措施，这些能力可能催生新形式的治理操纵。

反过来，代理也可以通过降低参与门槛来提升治理参与度。帮助用户理解复杂提案、依据用户表达的偏好委托投票权，以及参与治理讨论的代理，可以提高实际参与率，并改善集体决策的质量。要在实现这些益处的同时降低风险，就需要审慎设计代理系统和治理机制。

10.3.5 代理参与去中心化治理

自主代理与去中心化治理的交汇，引发了关于代表性、问责以及集体决策本质的根本问题 [43, 87, 110, 187]。随着去中心化自治组织日益管理着大量经济资源和协议参数，由代理促成的治理参与既带来了机遇，也引发了担忧。

一方面，代理可以通过对提案进行更深入的分析、依据策略偏好自动委托，以及及时投票（人类参与者可能会因注意力限制而错过投票时机），来提升治理参与度。另一方面，由代理促成的治理也引发了对投票操纵、女巫攻击，以及影响力集中于掌控复杂代理系统者手中的担忧。因此，治理框架必须不断发展以应对这些动态变化，并可纳入检测和缓解自动化操纵的机制，同时保留代理辅助参与所带来的益处。

10.3.6 案例研究方法论

本综述中介绍的案例研究，是通过对公开信息进行系统分析，并在可能的情况下辅以专家咨询而编写的。

案例选择标准 选择这些案例旨在展示本综述所涵盖的各种集成模式和应用领域。选择标准包括代表性（案例应能说明常见模式）、启发性（案例应能提供明确的经验教训）和文档可获取性（应有足够的公开信息以进行有意义的分析）。

信息来源 案例研究信息收集自多个来源，包括技术文档、博客文章、会议演讲，以及在可能的情况下与系统运营人员直接沟通。通过交叉参照多个来源的信息来验证准确性并识别差异。

分析框架 每个案例研究均采用统一的框架进行分析，涵盖系统架构、安全模型、运营实践和经验教训。该框架有助于对不同案例进行比较，并识别其中的共同模式与差异。

局限性 案例研究必然受到公开信息有限的制约。系统实现的专有细节、运营指标和安全事件可能不会披露。所呈现的案例应被视为说明性示例，而非对所述系统的全面记录。

10.3.7 案例研究的经验教训 s

本综述中的案例研究展示了智能体与区块链集成的潜力与挑战。 .
分析这些案例可以归纳出几项共通的经验教训。

经验教训 1：从保守起步，逐步扩展 成功的代理部署通常从保守的配置开始，并随着运行经验的积累逐步扩展能力。从只读模式或仅模拟模式开始，能让操作员在授予执行权限之前验证代理行为。逐步提高支出限额并扩大允许执行的操作范围，有助于在建立对代理可靠性的信心的同时，审慎承担风险。

这种做法与一开始就部署完全自主的智能体以最大化潜在收益的诱惑背道而驰。保守部署或许会错失短期机会，但能降低灾难性故障的风险，避免其损害长期成功。

经验教训 2：投资于监控与可观测性 有效的监控与可观测性对于安全地运行代理至关重要。操作员必须能够了解代理正在做什么、为何做出特定决策，以及其行为与预期有何差异。对监控基础设施的投入能带来丰厚回报，包括更快地检测事件、更轻松地进行调试，以及更深入地了解代理性能。

监控应涵盖多个维度，包括交易成功率、经济性能、策略合规性和行为模式。异常检测系统可以识别异常行为，这些行为可能表明存在安全问题或策略漂移。

第3课：为故障与恢复做好规划 代理系统不可避免地会遭遇故障，原因可能是程序错误、意外的市场状况或安全事件。成功的部署会为故障制定规划，并测试恢复程序。这包括在检测到异常时暂停操作的自动断路器、允许操作员介入的手动覆盖能力，以及在事件发生后恢复正常操作的恢复程序。

故障规划应同时考虑技术故障（交易回滚、网络问题）和经济故障（策略损失、不利的市场波动）。不同类型的故障可能需要不同的响应程序。

第4课：保持人工监督 即使是高度自主的智能体，也能从人工监督中受益。由具备相关知识的操作员定期审查代理行为、定期进行策略评估并持续监控，有助于发现自动化系统可能遗漏的问题。人工监督还可以提供问责，并促使系统适应自动化策略可能未涵盖的变化状况。

适当的人工监督程度取决于所涉风险和代理系统的成熟度。风险较高的应用需要更严格的监督，而具有成熟运行记录的系统则可以接受较轻的监督。

11 相关工作

本综述综合了智能体人工智能工具使用、区块链执行基础设施和去中心化金融领域的研究，特别关注连接智能体与链上操作的接口层。以往的综述和知识系统化（SoK）论文分别探讨了这一技术栈的各个组成部分，但从标准和接口出发、整合这些研究方向并厘清 2026年设计空间的视角，至今仍具有现实意义。

11.1 大语言模型安全性与工具使用综述

将大语言模型部署为工具使用型代理的安全性是一个快速发展的领域，近期的综述总结了大语言模型集成系统中工具学习和工具调用所面临的威胁与缓解措施 [253, 181]。一个重要教训是，即使开发者提示看似无害，不可信检索内容仍可能通过间接提示注入影响模型行为 [131]。本综述将这些故障模式纳入我们的威胁模型和评估清单，并重点关注区块链交易这一高风险场景，因为一次安全失陷的操作就可能造成不可逆的后果。除提示注入外，知识系统化（SoK）类研究还强调了大语言模型与工具边界处更广泛的接口风险，包括数据

外泄、不安全的工具调用，以及工具编排中的拒绝服务攻击向量 [181]。我们的贡献是将这些通用接口威胁转化为适用于代理到链场景的分析，在这一场景中，工具不只是读取数据，还能直接触发具有价值的状态转换。

11.2 MEV 知识体系化

最大可提取价值（MEV）对交易排序和纳入过程中产生的获利机会进行了形式化描述，奠基性研究记录了抢先交易和三明治攻击等典型策略 [80]。后续的体系化研究分析了 MEV 供应链、中继器和构建者的角色，以及涉及私有订单流和替代执行场所的缓解机制 [58]。本综述直接建立在这些研究成果之上，将 MEV 视为自主代理始终面临的对手，而非人类交易者偶尔遭遇的风险。在我们的威胁模型中，任何公开交易都被假定具有经济对抗性；在能力矩阵中，我们则追踪各系统是否提供私有提交路径或基于意图的执行场所，以降低 MEV 暴露。

11.3 基于意图的架构研究

以意图为中心的架构将用户期望的结果（即“做什么”）与协议特定的执行细节（即“如何做”）分离，由此推动了基于求解器的执行和竞争性订单流，以实现更优定价并降低对抗性风险 [202]。这一范式既作为一种交互模型，也作为一种执行层机制得到分析；与以调用数据为中心的交易构造相比，它改变了信任和威胁假设 [71]。我们提出的交易意图模式（TIS）是对这一研究方向的直接贡献：其目标是在各类智能体任务中标准化意图表示，而不仅限于代币兑换。路线图提出，意图接口应成为非确定性代理规划与确定性链上执行之间的默认边界对象，尤其是在与 PDR 形式的签名策略证明相结合时。

11.4 智能合约安全与形式化方法综述

智能合约安全领域已有丰富的研究文献，其中包括对漏洞类别、分析方法和缓解策略的综述 [347, 215, 142]。形式化方法和侧重验证的综述则从另一角度补充了这些观点，着重讨论规约、验证工作流，以及在对抗性环境中纯静态分析的局限性 [200]。本文综述在此基础上，将重点放在代理层：我们假设协议可能已经过审计，但仍将代理执行边界视为独立的风险来源。具体而言，我们强调模拟是一种安全原语，能够弥合智能合约正确性与安全代理行为之间的差距，尤其是在存在动态状态、多步骤工作流和对抗性市场结构的情况下。

11.5 互操作性、跨链风险与执行基础设施方面的研究

区块链技术栈日益依赖互操作性和跨链执行，由此带来了额外的安全与隐私风险。有关 Rollup 和可扩展性的综述为代理可能跨二层和三层系统运行的执行环境提供了背景 [297]。跨链互操作性综述和知识系统化论文指出了跨链桥与消息传递风险，以及跨域安全组合的复杂性 [85, 29, 279]。本文综述借鉴这些研究结果，为多链代理采用保守的威胁假设提供依据，并说明为何应将执行场所（打包器、中继器、求解器和跨域消息传递）视为代理安全边界的一部分，而不应仅仅将其看作基础设施。

11.6 去中心化身份、治理与密钥管理

密钥管理和身份框架是代理托管与授权设计选择的基础。关于去中心化标识符和可验证凭证的综述，为理解如何在互操作环境中识别、进行身份认证和审计智能体及策略引擎提供了更广泛的背景 [205, 328, 13]。在治理层面，近期对去中心化自治组织治理的综述以及对去中心化自治组织攻击面的分析，促使人们为高价值操作设计多方和多代理审批工作流 [137, 102]。我们的分类体系可以视为密钥管理与授权的发展历程：从单密钥、单代理模式，逐步演进到分布式托管、策略证明和多代理工作流，以满足遏制安全失陷并大规模支持可审计性的需求。

通过将这些线索交织起来，本综述对代理-区块链接口作为关键安全集成层提供了统一视角，并推动了意图模式、策略证明和评估基准的标准化工作。

12 结论与未来工作

智能体人工智能与区块链技术的融合，标志着这两个领域演进过程中的关键时刻，使自主系统能够在对抗性激励驱动环境中行使真正的经济自主权。这一转变扩展了可编程金融与治理的范围，从常规支付和资金管理运营，延伸至意图驱动执行以及自主参与链上协调机制 [100, 217, 180]。与此同时，区块链施加了一种独特且毫不宽容的运行机制：操作不可逆转，执行过程具有经济对抗性（最大可提取价值、操纵、审查压力），而错误会随着委托权限的扩大而立即放大 [80, 58, 287]。这些条件要求采用有别于传统大语言模型工具使用模式的接口和控制平面，包括结构化意图、策略执行、可验证审计凭证以及基于模拟的预检 [131, 330, 313]。

本综述系统概述了 2025年智能体—区块链互操作性的发展现状。基于遵循 PRISMA 指南、涵盖 317篇文献的综述，我们提出了：(i) 由五部分组成的集成模式分类体系，(ii) 针对工具使用型代理的区块链专属威胁模型，以及 (iii) 比较能力矩阵，用于呈现托管、权限、策略位置、可观测性和最大可提取价值缓解方面的主导设计权衡 [273, 202]。在所研究的系统中，该领域正逐渐转向纵深防御架构，将账户抽象与受限委托、以意图为中心的执行场所，以及结合强大可观测性和审计跟踪的分层策略执行融为一体 [2, 308, 27]。为弥补尚存的差距，我们提出了以两个缺失的接口层——交易意图模式（TIS）和策略决策记录（PDR）——为核心的 2026年路线图，并概述了用于评估端到端智能体到区块链流水线安全性、可靠性和经济鲁棒性的基准测试支柱与检查清单 [202, 200, 138]。

展望未来，从代理原型迈向可信赖的链上自主性，最可靠的路径是标准化与可衡量的保障。标准化的意图和策略工件可以减少集成脆弱性，推动可组合的求解器市场，并在规划、验证和签名之间建立可审计的问责边界。此外，领域专用的基准测试和检查清单能够将定性的安全声明转化为可重复的证据，尤其是在提示注入、工具欺骗、MEV压力和部分故障模式下 [131, 80, 185]。通过开放标准、纵深防御和严格评估，智能体人工智能可以安全地利用区块链执行，构建更具可编程性、更高效且更去中心化的系统 [27, 100]。

12.1 未来工作与开放研究问题

尽管进展迅速，真正自主且可信赖的链上智能体仍处于研究的早期阶段。下面我们将重点介绍最重要的开放问题。

12.1.1 智能体—合约阻抗失配

概率性、非确定性的代理行为与确定性的智能合约执行之间存在深刻矛盾。当代理故障导致不可逆的状态变更时，即便极少发生，也可能酿成灾难性后果。重要的研究方向包括：开发能够确保代理输出始终处于形式化规定的安全边界内的方法，其中包括对工具调用、策略检查和执行场所进行组合式推理 [200, 143]；将大语言模型的灵活性与符号约束、类型化意图和经形式化验证的防护条件结合起来，使高层推理保持表达能力，同时确保执行受到约束且可验证 [47, 319]；以及设计稳健的遥测和最后一道中断机制，包括链上或近链监视器，以检测偏离所学习行为基线的情况，并触发撤销、暂停或安全退出流程 [188, 182]。

12.1.2 智能体系统的经济学

大规模自主逐利智能体将重塑市场微观结构和对抗性动态，可能会加剧反馈回路和利用行为。开放问题包括：如何检测并缓解独立智能体之间的默契协调，尤其是在流动性稀薄环境和求解器市场中；如何理解重复交互下的均衡结果 [329, 157]；如何厘清链下计算、求解器竞争和验证开销应如何定价、补贴或限制速率，尤其是在账户抽象和赞助执行模式下 [2, 308]；以及如何描述许多智能体共享相似决策规则或工具链时的系统性风险，包括预言机冲击、拥堵和 MEV 激增情境下的压力测试 [80, 287]。

12.1.3 治理、监管与法律责任

自主经济行为引发了有关问责和可执行监督的未决问题。关键方向包括构建可审计流程，使安全声明能够得到独立验证，其中包括可验证的策略决策凭证和可复现的评估报告 [200, 27]；设计链上保险和附条件保障

机制，以便为智能体运营风险定价、设限并进行风险共担，理想情况下还应与可衡量的安全态势和合规证据挂钩 [224]；并将身份、密钥管理和策略溯源联系起来，使代理运营者、策略发布者和执行场所之间的责任边界始终清晰 [57]。

12.1.4 信任的用户体验

主流采用取决于非专业人士能否正确校准对自主金融智能体的信任。研究应优先开展以下工作：生成简洁、忠实反映实际约束条件和所执行检查，并与意图及策略工件相关联的面向用户的解释，而非事后编造的说辞 [27]；设计直观的会话密钥、支出限额、允许列表和撤销控制措施，并使其与账户抽象工作流及恢复模式相契合 [2, 308]；制定经过校准的路径，让用户可以从范围严格受限的模式 III 委托开始，随着系统积累安全证据和运营信任，再逐步转向自主性更高的模式 [273, 27]。

应对这些挑战，需要在智能体安全、智能合约工程、MEV 研究、密码学、人机交互和标准制定流程等领域开展协调合作。技术机遇十分重大，但在对抗性金融环境中实现可信的自主性，标准必须始终毫不妥协 [80, 131, 27]。

交易意图模式（TIS）- 参考实现

本附录为路线图部分提出的交易意图模式（TIS）提供参考实现。TIS 旨在与链无关、可扩展，并适用于机器处理和人工审查。该模式使用 JSON Schema（Draft-07）进行规范，并拆分为多个模块化定义，以便清晰呈现。

A.1 顶层 TIS 对象（JSON Schema Draft-07）

```
{ "$schema": "http://json-schema.org/draft-07/schema#", "title": "交易意图
模式（TIS）", "description":"用于表达所需结果的与链无关的模式"}
                                                                    来访事项及针对链上自主代理的限制条件。
      链执行。", "type": "object", "additionalProperties": false, "properties":
 { "version": { "type": "string", "pattern": "^1\\.0\\.0$" }, "intentId": { "type": "
 string", "format": "uuid" }, "元数据": { "$ref": "#/definitions/Metadata" }, "操
 作": { "$ref": "#/definitions/Action" }, "约束":{ "$ref": "#/
 definitions/Constraints" }, "偏好设置":{ "$ref": "#/definitions/Preferences" }
 }, "required":  ["version", "intentId", " 操作 ", " 约束 "] ,"definitions": { " 地
 址": { "type": "string", "pattern": "^0x[a-fA-F0-9]{40}$" }, "十六进制字节": { "
 type": "string", "pattern": "^0x[a-fA-F0-9]*$" }, "无符号整数字符串": { "type":
 "string", "pattern": "^[0-9]+$", "description":" 以十进制字符串形式序列化的
 无符号整数。" }, "Token": { "type": "object", "additionalProperties": false, "
 properties": { "链 ID": { "type": "integer", "minimum": 1 }, "address": { "$
 ref": "#/definitions/Address" }, "符号": { "type": "string" }, "小数位数": { "
 type": "integer", "minimum": 0, "maximum": 255} }, "required":  ["链 ID",
 "address"]}, "Metadata": { "type": "object", "additionalProperties": false, "
 properties": { "originator": { "type": "string", "description": "标识符，用于
                                                                    用户、代理或客户端，其
      创建意图。" }, "createdAt": { "type": "string", "format": "日期
 时间" }, "originChainId": { "type": "integer", "minimum": 1 }, "
 tags": { "type": "array", "items": { "type": "string" }} }
```

```
}, "Constraints": { "type": "object", "additionalProperties": false, "properties": { "nonce":
{ "$ref": "#/definitions/无符号整数字符串" }, "deadline": { "type": "integer", "minimum":
0, "description":"Unix 时间戳 s

                                                                                秒。" },
"validFromBlock": { "type": "integer", "minimum": 0 }, "validUntilBlock": { "type": "
integer", "minimum": 0 }, "maxGasPriceWei": { "$ref": "#/definitions/UintString" }, "
requiredSigner": { "$ref": "#/definitions/Address" }, "exclusivity": { "type": ["string", "
null"],"pattern": "^0x[a-fA-F0-9]{40}$", "description":"可选的求解器/构建者地址，获准独
占执行权

                                                                                期间的权益
    有效期。" } }, "required": ["deadline"]}, "Preferences": { "type": "object", "
additionalProperties": false, "properties": { "privacyMode":{ "type": "string", "
enum": ["PUBLIC", "PRIVATE"]}, "executionSpeed": { "type": "string", "enum": ["
FAST", "NORMAL","SLOW"] }, "routing": { "type": "string", "enum": ["BEST_
PRICE", "MIN_GAS", "MIN_RISK"],"description":"传达给求解器的可选路由偏好。" } } },
"Action": { "type": "object", "oneOf": [{ "$ref": "#/definitions/actions/Swap" }, { "$
ref": "#/definitions/actions/Transfer" }, { "$ref": "#/definitions/actions/Delegate" }]},
"actions": { "Swap": { "type": "object", "additionalProperties": false, "properties": { "
type": { "const": "SWAP" }, "tokenIn": { "$ref": "#/definitions/Token" }, "tokenOut":
{ "$ref": "#/definitions/Token" }, "amountIn": { "$ref": "#/definitions/UintString" }, "
minAmountOut": { "$ref": "#/definitions/UintString" }, "slippageBps":{ "type": "
integer", "minimum": 0, "maximum":10000 }, "recipient": { "$ref": "#/
definitions/Address" } }, "required": ["type", "tokenIn", "tokenOut", "amountIn", "
minAmountOut"]}, "Transfer": { "type": "object", "additionalProperties": false,
```

"properties": { "type": { "const": "TRANSFER" }, "token": { "$ref": "#/definitions/Token" }, "to": { "$ref": "#/definitions/Address" }, "amount": { "$ref": "#/definitions/UintString" }, "memo": { "type": "string" } }, "required": ["type", "token", "to", "amount"]}, "Delegate": { "type": "object", "additionalProperties": false, "properties": { "type": { "const": "DELEGATE" }, "delegatee": { "$ref": "#/definitions/Address" }, "scope": { "type": "object", "additionalProperties": false, "properties": { "contracts": { "type": "array", "items": { "$ref": "#/definitions/Address" } }, "functions": { "type": "array", "items": { "type": "string"} }, "maxValueWei":{ "$ref": "#/definitions/UintString" }, "validUntil": { "type": "integer", "minimum": 0, "description":"Unix 时间戳秒数。"} } } }, "required": ["type", "delegatee", "scope"]} } } }

A.2 关于规范化与哈希的说明

实现应在哈希和签名前对 TIS 对象进行规范化（例如，采用稳定的 JSON 序列化和确定性的键排序）。这是确保不同客户端对相同语义意图计算出相同哈希值所必需的。

A.3 扩展点

未来可通过在 definitions/actions/* 下添加更多操作类型，以及为元数据、约束或偏好设置扩展新字段来实现扩展。应通过版本控制保持向后兼容性。

B 策略决策记录（PDR）—参考实现

本附录为路线图部分提出的策略决策记录（PDR）提供参考实现。PDR 是策略引擎评估特定 TIS 对象后生成的签名证明。它绑定了 (i) 经评估的意图哈希，(ii) 决策结果，以及 (iii) 策略引擎施加的任何约束。

B.1 顶层 PDR 对象（JSON Schema Draft-07）

{
"$schema": "http://json-schema.org/draft-07/schema#", "title": "策略决策记录（PDR）", "description":"关于特定交易意图（TIS）的已签名策略证明。", "type": "object", "additionalProperties": false, "properties": {

"version": { "type": "string", "pattern": "^1\\.0\\.0$" }, "pdrId": { "type": "string", "format": "uuid" }, "issuer": { "type": "string", "description": "策略引擎的标识符（例如：去中心化标识符、ENS、统一资源定位符）。" }, "subject": { "type": "string", "description": "代表其发起意图的账户或用户，意图正在

g

已评估。" }, "audience": { "type": "string", "description": "预期的验证器，通常为签名服务标识符。" }, "issuedAt": { "type": "string", "format": "date-time" }, "expiresAt": { "type": "string", "format": "date-time" }, "tisHash": { "type": "string", "pattern": "^0x[a-fA-F0-9]{64}$", "description":"规范化 TIS 对象的 Keccak-256 哈希。" }, "decision": { "$ref": "#/definitions/Decision" }, "policyEngineSignature": { "$ref": "#/definitions/Signature" } }, "required": ["version", "pdrId", "issuer", "audience", "issuedAt","expiresAt", "tisHash", " decision", "policyEngineSignature"] ,"definitions": { "Address": { "type": "string", "pattern": "^0x[a-fA-F0-9]{40}$" }, "UintString": { "type": "string", "pattern": "^[0-9]+$" }, "Modification": { "type": "object", "additionalProperties": false, "properties": { "path": { "type": "string", "description": "指向 TIS 对象的 JSON 指针路径（RFC 6901）。" }, "operation": { "type": "string", "enum": ["ADD", "REPLACE", "REMOVE"] }, "value": { "type": ["string", "number", "object", "array", "boolean", "null"] } }, "required": ["path", "operation"]}, "Decision": { "type": "object", "additionalProperties": false, "properties": { "outcome": { "type": "string", "enum": ["APPROVED", "REJECTED"]}, "policyId": { "type": "string", "description": "所应用策略包或规则集版本的标识符。" }, "reason": { "type": "string", "description": "人类可读的决策理由，对 REJECTED 尤其重要。" }, "riskScore": { "type": "number", "minimum": 0.0, "maximum": 1.0 }, "boundConstraints": { "type": "object", "additionalProperties": false, "properties": { "maxGasPriceWei": { "$ref": "#/definitions/UintString" }, "maxValueWei":{ "$ref": "#/definitions/UintString" }, "tightDeadline": { "type": "integer", "minimum": 0, "description": "可选的缩短版 Unix 截止时间"

e

。" } } }, "修改后的参数": { "type": "array",

"items": { "$ref": "#/definitions/Modification" } } }, "required": ["outcome", "policyId"]}, "Signature": { "type": "object", "additionalProperties": false, "properties": { "signer": { "$ref": "#/definitions/Address", "description": "策略引擎签名密钥的地址。" }, "alg": { "type": "string", "description": "签名算法标识符（例如 ES256K）。" }, "signature": { "type": "string", "pattern": "^0x[a-fA-F0-9]+$" } }, "required": ["signer", "signature"]} } }

B.2 验证与执行流程

PDR 支持多阶段执行流程：

1. **意图创建：**代理构造一个 TIS 对象。2. **策略评估：**代理将 TIS 提交给策略引擎。3. **PDR 生成：**策略引擎评估 TIS，并在批准后生成已签名的 PDR。4. **执行请求：**代理将 TIS 和 PDR 提交给签名环境（例如 MPC 服务、HSM 或 TEE）。5. **签名者验证：**签名环境会验证：(a) PDR 上的签名是否与受信任的策略引擎公钥匹配；(b) 规范化 TIS 的哈希值是否与 tisHash 匹配；(c) expiresAt 是否尚未过期，且 audience 是否与签名者身份匹配；(d) 是否在签名前强制执行任何绑定约束和修改后的参数。

C 智能体系统的符合性级别

本附录定义了智能体—区块链系统的四个符合性级别，与评估章节相对应。每个级别都会增加更多控制措施，并降低安全失陷时的最大损失。

C.1 一层 0：不受约束的代理

- **描述：** 代理可直接访问私钥，并签名和提交任意交易。
- **架构：** 密钥存储在本地或环境变量中；没有策略门控。
- **风险：** 安全失陷可能导致灾难性损失。
- **示例：** 早期自主交易脚本的默认配置。

C.2 一层：基础链上策略控制措施

- **描述：**通过智能账户模块（例如会话密钥、函数/合约允许列表）限制权限。
- **架构：**代理在范围严格受限的链上规则下签名用户操作。
- **控制措施：** – 函数允许列表

– 合约白名单– 静态支
出限额

- **风险：** 受模块约束限制。

C.3 二层：链下策略与强制模拟

- **描述：** 智能体提案由链下策略引擎审核，并在签名前进行模拟。
- **架构：** TIS 提交至策略引擎，然后必须通过 PDR 才能签名。
- **控制措施：** – 所有一层控制措施– 强制执行预执行
模拟和回滚检查– 动态风险评分和异常检测

- **风险：** 在单组件失陷情况下，风险较低。

C.4 L3：硬件安全签名与分布式审批

- **描述：** 通过硬件隔离（TEE/HSM）或分布式签名（MPC），并对高价值操作采用法定人数审批，实现最高级别的安全性。
- **架构：** 签名由可信执行环境/硬件安全模块生成，或通过多方计算生成；高价值操作需要多方审批或多智能体法定人数审批。
- **控制措施：**
  – 所有二层控制措施– 可信执行环境/硬件
  安全模块或多方计算签名– 高风险操作的
  法定人数审批– 完善的恢复与撤销流程

- **风险：**除非发生多层安全失陷，否则风险极低。

C.5 实施参考：SampleAgent 配置

本附录提供常见智能体部署场景的示例配置，说明本综述中讨论的概念如何转化为实际实现。

C.5.1 配置 1：保守型 DeFi 智能体

保守型去中心化金融智 能体将安全性置于性能之上，适用于管理大量资产并尽可能降低 1
风险承受能力 .

代理_配置：名称："保守型_去中心化
金融_代理" 版本："1.0.0"

托管：类型："智能_合约_钱包" 实现：
"Safe" 阈值："2-of-3" 签名者：- 类
型："硬件_钱包" - 类型："多方计算_
分片" - 类型："人工_备份"

策略：最大_交易_金额：“10000 USD” 每
日_限额：“50000 USD” 允许的_协议：- “
uniswap_v3” - “aave_v3”

- “co
mpound_v3” 允许
的_操作：- “兑换” -
“供应” - “提取” 阻
止的_操作：- “借入”
- “ 杠杆 ”

执行：模拟_必需：是 人工_授权_阈值："
5000 美元" 最大可提取价值_保护："
flashbots_protect" 最大_滑点："0.5%"

监控：警报_渠道：- "电子邮件" - "
Slack" 异常_检测：是 自动_暂停_条
件：- "每日_亏损 > 5%" - "失败的_
交易 > 3"

C.5.2 配置 2：ActiveTradingAgent

主动交易代理优先考虑性能和响应能力，适用于需要快速执行的策略。

代理_配置：名称："主动_交易_代
理" 版本："1.0.0"

托管: 类型: "mpc_钱包" 实现: "
Turnkey" 阈值: "2-of-3" 延迟_
优化: true

策略: 最大_交易_价值: "100000 USD" 持仓_
限额: 最大_单一_持仓: "25%" 最大_协议_敞
口: "40%" 允许的_协议: "全部_已审计" 风险
_参数: 最大_回撤: "10%" var_限额: "5%"

执行: 模拟_必需: false 延迟_目标: "
100ms" 最大可提取价值_保护: "私有_内
存池" 最大_滑点: "2%" 重试_策略: 最大_
重试次数: 3 退避: "指数型"

监控：实时_盈亏：true 持仓
_跟踪：true

自动_再平衡：true

C.5.3 配置 3：治理参与代理

治理代理负责管理多个去中心化自治组织的投票和委托。

代理_配置：名称："治理_代理
" 版本："1.0.0"

托管：类型："智能_合约_钱包" 实现：
"Safe" 治理_优化：是

策略：投票_策略：默认_操作："弃权" 自动_投票
_阈值："低_影响" 人工_审核_阈值："高_影响" 委
托_策略：允许_委托人："精选_名单" 每位_委托
人的最大_委托_比例："20%" 重新委托_冷却期：
"7 天"

协议：- 名称："uniswap" 投票_权重_来源：
"uni_代币" 提案_监控：true - 名称："
aave" 投票_权重_来源："aave_代币" 提案_
监控：true

监控：提案_警报：true 投票_截止时间_提醒：
true 委托_绩效_跟踪：true

这些配置 展示了代理部署可用选项的范围，以及将 g
配置选项 与具体用例需求相匹配的重要性。

D 智能体—区块链安全检查清单

本检查清单为开发者、审计人员和操作员提供实用指南，用于评估智能体—区块链系统的安全态势。它按照代理操作流水线进行组织。

D.1 观察

1. □ 代理是否使用多个相互独立的 RPC 提供商？ 2. □ 代理是否验证链下 API 数据的完整性和时效性？ 3. □ 代理是否监控与其头寸相关的异常链上事件？ 4. □ 代理是否能抵御预言机操纵和陈旧价格风险？ 5. □ 代理是否订阅其所依赖协议的安全警报信息源？

D.2 推理

1. □ 该代理是否能够抵御通过检索内容实施的指令劫持？ 2. □ 该代理是否考虑了最大可提取价值和对抗性排序激励？

3. □ 该代理是否维护了自身权限和限额的内部模型？ 4. □ 是否记录推理轨迹和工具调用，以供审计审查？ 5. □ 该代理在决策时是否考虑了 Gas、费用和截止时间风险？

D.3 构造

1. □ 代理是否生成标准化意图（TIS），而不是临时调用数据？ 2. □ 代理是否在签名前，基于分叉状态模拟每个价值转移操作？ 3. □ 代理是否生成清晰、易于人类理解的效果预览（资产、授权、状态变更）？ 4. □ 代理是否突出显示不可逆或高风险操作（例如无限授权）？ 5. □ 每个构造的操作是否都包含严格的截止时间和重放保护？

D.4 授权

1. □ 系统是否在适用情况下使用智能账户（ERC-4337）？ 2. □ 是否通过链上模块（会话密钥、允许列表、限额）限制权限？ 3. □ 是否设有用于高阶规则的链下策略引擎？ 4. □ 每项策略决策是否都记录为与意图哈希绑定的可验证 PDR？ 5. □ 签名是否隔离在可信执行环境 / 硬件安全模块中，或通过多方计算分布式完成？ 6. □ 密钥是否按权限级别和操作用途进行分段？ 7. □ 是否有明确的智能体密钥和模块撤销流程？ 8. □ 高价值操作是否由多方审批或法定人数进行门控？ 9. □ 是否强制执行单笔交易和时间窗口支出限额？ 10. □ 策略变更是否通过时间锁或分阶段发布得到保护？

D.5 执行

1. □ 价值转移交易是否在适当情况下通过私有中继或私有订单流进行路由？ 2. □ 是否在能够降低最大可提取价值风险时使用基于意图的交易场所（例如，基于求解器的执行）？ 3. □ 系统能否稳健地估算费用并适应拥堵？ 4. □ 系统能否安全地处理失败或卡滞的交易（替换策略、取消或回滚）？ [67]5. □ 是否对执行进行实时监控并发出告警？

D.6 验证与恢复

1. □ 代理是否通过读取执行后的状态（而不只是交易回执）来验证执行结果？ 2. □ 代理是否解析事件以确认预期结果并检测异常？ 3. □ 代理能否在多步骤工作流中从部分故障中恢复（通过补偿或回滚）？ 4. □ 是否设有紧急停止开关，能够快速可靠地停止操作？ 5. □ 是否有包含明确升级步骤的事件响应手册？

D.7 通用

1. □ 是否向用户和审计人员清晰记录了架构和信任模型？ 2. □ 关键代码（合约、模块、策略引擎）是否经过审计？ 3. □ 是否设有负责任披露机制或漏洞赏金计划？ 4. □ 链下通信是否经过身份认证并加密？ 5. □ 机密信息是否存储在安全保险库中，并按照既定流程定期轮换？ 6. □ 主机环境是否经过安全加固、受到监控并定期打补丁？ 7. □ 是否有安全的智能体逻辑和策略规则更新流程（预发布环境、回滚）？

8. □ 是否以可复现的方式报告基准测试和安全性评估？ 9. □ 是否针对策略变更和紧急行动设有治理流程？ 10. □ 是否向用户清晰披露风险，并提供争议解决途径？

D.8 其他实现说明

本节为实践者提供额外的实现说明，帮助他们基于本综述介绍的框架部署智能体区块链系统。

测试策略全面测试对于安全地部署代理至关重要。测试策略应包括针对各个组件的单元测试、针对组件交互的集成测试，以及针对完整工作流的端到端测试。应特别关注边界情况、错误处理和对抗性场景。

主网分叉使得无需冒真实资金风险即可针对真实的区块链状态进行测试。不过，分叉环境无法重现实时运行的所有方面，包括网络延迟、Gas价格动态以及其他参与者的行为。辅以在测试网和模拟环境中的测试，有助于覆盖分叉环境无法重现的场景。

监控与告警生产环境中的代理系统需要全面的监控与告警，以便在问题造成重大损害之前及时发现。需要监控的关键度量指标包括交易成功率、Gas消耗、执行延迟和经济性能。异常检测系统可以识别可能表明存在安全问题或策略漂移的异常模式。

应校准告警阈值，在灵敏度与误报率之间取得平衡。过于灵敏的警报会造成告警疲劳，可能被忽略；而灵敏度不足的警报则可能漏掉真实问题。定期审查并根据运维经验调整告警阈值，有助于保持适当的灵敏度。

事件响应 尽管已尽最大努力进行预防，安全事件仍会发生。做好准备的事件响应流程能够实现快速、有效的响应，从而将危害降至最低。事件响应流程应涵盖检测、遏制、调查、补救和事后复盘。

事件响应的关键能力包括迅速停止代理操作、撤销已泄露凭证，以及与受影响的参与方沟通。定期开展事件响应演练，有助于确保流程按预期运行，并让团队成员熟悉各自的角色。

运维安全 运维安全实践可以防范仅靠技术控制措施无法应对的威胁。关键实践包括职责分离（任何个人都不应拥有对关键系统的完整访问权限）、访问日志记录与审查，以及定期开展安全评估。

对于管理重大价值资产的系统，人员安全尤为重要。背景调查、安全培训，以及明确的可接受使用策略，有助于降低内部威胁风险。离职流程应确保离职人员失去对系统和凭证的访问权限。

E 术语表

本术语表定义了本综述中使用的关键术语，以确保跨学科受众能够形成一致的理解。

账户抽象： 一种将账户所有权与交易授权分离的范式，使授权逻辑能够通过智能合约进行编程，而不再仅依赖密码学密钥持有。
代理： 一种自主软件系统，能够感知环境、推理观测结果，并采取行动，以尽量减少人工干预的方式实现指定目标。

智能体人工智能： 旨在长时间自主运行的人工智能系统，通过做出决策并采取行动来追求目标，而非仅仅回应单个查询。
捆绑器： 在 ERC-4337 中，一种收集多个用户的用户操作、将其捆绑为单笔交易并提交至区块链的服务。

断路器： 当检测到预先设定的异常条件时，自动暂停智能体运行，以防在可能发生安全事件期间继续执行。

托管： 对控制区块链资产的密码学密钥进行安全保管和管理。托管解决方案包括自托管（用户持有密钥）和机构托管（第三方持有密钥）。
去中心化金融（DeFi）： 通过区块链网络上的智能合约实现的金融服务，无需传统中介即可运行。

外部拥有账户（EOA）： 由私钥控制的区块链账户，不同于由代码控制的智能合约账户。

Gas： 用于衡量在以太坊及类似区块链上执行操作所需计算量的单位。用户支付 Gas 费用，以补偿验证器处理交易。

护栏机制： 用于限制代理行为、防止有害或非预期操作的约束和保障措施。
意图： 对期望结果的高层级规约，可通过各种执行路径实现，不同于具体的交易规范。

LLM（大语言模型）： 一种通过大型文本语料库训练的神经网络，能够生成和理解自然语言，是许多现代人工智能智能体的基础。
MEV（最大可提取价值）： 通过交易排序从区块链用户那里提取的价值，包括抢先交易、尾随交易和三明治攻击。

MPC（多方计算）： 一种密码学技术，使多个参与方能够使用各自的输入共同计算 Functions，同时不向彼此透露这些输入。
预言机： 一种向智能合约提供外部数据的服务，弥合链上信息与链下信息之间的差距。

PDR（策略决策记录）： 一种拟议的标准，用于记录策略评估决策，并创建可审计的代理授权追踪记录。

提示注入： 一种攻击技术，通过将恶意指令插入人工智能智能体的输入上下文来操纵其行为。

Rollup： 一种二层扩展解决方案，在链下执行交易，同时将交易数据或证明提交至主链以确保安全性。

会话密钥： 一种权限受限的临时密钥，使代理能够在既定约束条件内执行交易，而无需访问主账户密钥。

滑点： 预期执行价格与实际执行价格之间的差异，通常由市场波动或交易执行期间的 MEV 提取造成。

智能合约： 部署在区块链上的自执行代码，在预定义条件满足时自动履行协议条款。

TIS（交易意图模式）： 一种拟议标准，用于以结构化、机器可读的格式表达交易意图。

工具使用： 人工智能智能体与外部系统及 API 交互，以完成超出其固有能力范围任务的能力。

TSS（门限签名方案）： 一种将签名权限分配给多个参与方的密码学方案，需要达到阈值数量的参与方协作才能进行签名生成。
用户操作： 在 ERC-4337 中，表示用户预期操作的数据结构，由账户抽象基础设施进行验证和执行。

钱包： 软件或硬　　件，用于管理密码学密钥并促进与区块链网络的交互。

参考文献

[1] 开放策略代理：云原生环境的基于策略的控制（文档）。开放策略代理，2016年。访问日期：2026年1月7日。

[2] EIP-4337：通过入口点合约实现账户抽象。《以太坊改进提案》，2021年。访问日期：2026年1月7日。

[3] Gnosis Guild 权限入门套件：角色修饰器与权限模式。GitHub 仓库文档，2023年。访问日期：2026年1月6日。[4] Flashbots Protect 文档。Flashbots 文档，2025年。访问日期：2026年1月7日。[5] CoW Protocol 文档。CoW Protocol 文档，2026年。访问日期：2026年1月7日。[6] OpenZeppelin Defender 文档。OpenZeppelin，2026年。访问日期：2026年1月7日。[7] Tenderly 文档：模拟。Tenderly，2026年。访问日期：2026年1月7日。[8] Eric Aaron。动态意向：面向目标导向具身智能体的集成智能建模。《机器人与人工智能前沿》，2016年。[9] Ibrahim Abdelaziz、Kinjal Basu、Mayank Agarwal、Sadhana Kumaravel、Matthew Stallone、Rameswar Panda、Yara Rizk、G P Shrivatsa Bhargav、Maxwell Crouse、Chulaka Gunasekara、Shajith Ikbal、Sachindra Joshi、Hima Karanam、Vineet Kumar、Asim Munawar、Sumit Neelam、Dinesh Raghu、Udit Sharma、Adriana Meza Soria、Dheeraj Sreedhar、Praveen Venkateswaran、Merve Unuvar、David Daniel Cox、Salim Roukos、Luis A. Lastras 和 Pavan Kapanipathi。Granite 函数调用模型：通过细粒度任务的多任务学习引入函数调用能力。载于 Franck Dernoncourt、Daniel Preo₫tiuc-Pietro 和 Anastasia Shimorina 编，《2024年自然语言处理实证方法会议论文集：产业专题》，第1131–1139页，迈阿密，美国佛罗里达州，2024年11月。计算语言学协会。[10] Tesnim Abdellatif 和 Kei-Leo Brousmiche。基于用户与区块链行为模型的智能合约形式化验证。载于*2018*年第九届*IFIP*新技术、移动性与安全国际会议（*NTMS*），2018年。[11] Hayden Adams、Noah Zinsmeister、Moody Salem、River Keefer 和 Dan Robinson。Uniswap v3 核心。Uniswap Labs，2021年。[12] 开放策略代理。开放策略代理（OPA）文档。https://www.openpolicyagent.org/docs/ latest/，2023年。访问日期：2026年1月6日。[13] Md. Rayhan Ahmed、A. K. M. Muzahidul Islam、Swakkhar Shatabda 和 Salekul Islam。基于区块链的身份管理系统与自主主权身份生态系统：综合综述。*IEEE Access*，2022年。[14] Hamda Al-Breiki、Muhammad Habib Ur Rehman、Khaled Salah 和 Davor Svetinovic。可信区块链预言机：综述、比较与开放研究挑战。*IEEE Access*，2020年。[15] Tamimul Alam、Md. Asraf Ali 和 Md. Hasibur Rahman。元宇宙去中心化金融中的抢先交易攻击：系统综述。《国际科学与研究档案》，2024年。[16] Bithin Alangot、Pawel Szalachowski、Tien Tuan Anh Dinh、Souhail Meftah、Jeff Ivanos Gana、Khin Mi Mi Aung 和 Zengpeng Li。兼具可审计性与隐私保护的去中心化身份认证。《算法》，2023年。[17] Alchemy。Alchemy Account Kit：智能账户与账户抽象工具。https://docs.alchemy. com/docs/account-kit-overview，2026年。访问日期：2026年1月6日。[18] Zeinab Alipanahloo、Abdelhakim Senhaji Hafid 和 Kaiwen Zhang。以太坊与二层链中的最大可提取价值（MEV）缓解方法：综合综述。*IEEE Access*，2024年。[19] Morteza Alizadeh、Karl Andersson 和 Olov Schelén。去中心化身份方法的比较分析。*IEEEAccess*，2022 年。[20] Mouhamad Almakhour 、Layth Sliman、Abed Ellatif Samhat 和 Abdelhamid Mellouk。使用有限状态机对复合智能合约安全性进行形式化验证的方法。《沙特国王大学学报：计算机与信息科学》，2023年。[21] Mohammad Javad Amiri、Chenyuan Wu、Divyakant Agrawal、Amr El Abbadi、Boon Thau Loo 和 Mohammad Sadoghi。拜占庭容错的基石：用于拜占庭容错协议分析、实现与实验的统一平台。载于第*21*届*USENIX*网络系统设计与实现研讨会（*NSDI 24*），2024年。[22] Dario Amodei、Chris Olah、Jacob Steinhardt、Paul Christiano、John Schulman 和 Dan Mané。人工智能安全中的具体问题。*arXiv*，2016年。[23] Guillermo Angeris、Hsien-Tang Kao、Rei Chiang、Charlie Noyes 和 Tarun Chitra。对 Uniswap 市场的分析。*arXiv*预印本 *arXiv:1911.03380*，2019年。[24] Ian Appel 和 Jillian Grennan。对去中心化自治组织的控制。载于《美国经济学会论文与会议录》，2023年。

[25] Myrto Arapinis、Andriana Gkaniatsou、Dimitris Karakostas 和 Aggelos Kiayias。硬件钱包的形式化处理。《金融密码学与数据安全》，2019年。[26] Rodelio Arenas 和 Proceso Fernandez。CredenceLedger：用于可验证学术凭证的许可型区块链。载于*2018 IEEE*工程、技术与创新国际会议（*ICE/ITMC*），2018年。[27] Hassan Atwi、Tom Lichtenstein、Cesare Pautasso 和 Mathias Weske。提升交易排序的透明度：区块链驱动的协作流程中的动态排序者选择。《区块链：研究与应用》，2025年。[28] Raphael Auer、Bernhard Haslhofer、Stefan Kitzler、Pietro Saggese 和 Friedhelm Victor。去中心化金融（DeFi）技术。《数字金融》，2024年。[29] André Augusto、Rafael Belchior、Miguel Correia、André Vasconcelos、Luyao Zhang 和 Thomas Hardjono。知识系统化：区块链互操作性的安全与隐私。载于*2024 IEEE*安全与隐私研讨会（*SP*），2024年。[30] André Augusto、Rafael Belchior、Jonas Pfannschmidt、André Vasconcelos 和 Miguel Correia。XChainWatcher：监控并识别跨链桥攻击。*arXiv*预印本 *arXiv:2410.02029*，2024年。[31] Oscar Avellaneda、Alan Bachmann、Abbie Barbir、Joni Brenan、Pamela Dingle 和 Kim Hamilton Duffy。去中心化身份：从何而来，又将走向何方？*IEEE*通信标准杂志，2019年。[32] Algirdas Avizienis、Jean-Claude Laprie、Brian Randell 和 Carl Landwehr。可信赖与安全计算的基本概念与分类体系。*IEEE*可信赖与安全计算汇刊，2004年。[33] Brad Bachu、Xin Wan 和 Ciamac Moallemi。量化订单流拍卖中的价格改善。载于金融密码学与数据安全国际会议，2024年。[34] Seyed Mojtaba Hosseini Bamakan、Amirhossein Motavali 和 Alireza Babaei Bondarti。区块链共识算法性能评估标准综述。《Expert Systems with Applications》，2020年。[35] Mikolaj Barczentewicz、Alex F. Sarch 和 Natasha Vasan。区块链交易排序作为市场操纵。《俄亥俄州立大学技术法学刊》，2023年。[36] Elaine Barker、Miles Smid、Dennis Branstad 和 Santosh Chokhani。密码密钥管理系统设计框架。NIST特别出版物 800-130，2013年。[37] Massimo Bartoletti、James Hsin-yu Chiang 和 Alberto Lluch Lafuente。知识系统化：去中心化金融中的借贷池。载于金融密码学与数据安全：*FC 2021*国际研讨会，2021年。[38] Massimo Bartoletti、James Hsin-yu Chiang 和 Alberto Lluch-Lafuente。去中心化金融中自动做市商理论。《计算机科学中的逻辑方法》，2022年。[39] Massimo Bartoletti 和 Livio Pompianu。智能合约实证分析：平台、应用与设计模式。《金融密码学与数据安全》，2017年。[40] ANDREW BEAMS、RANJIT KUMARESAN、MOHAMMAD MOHSEN MINAEI BIDGOLI、MAHDI ZA- MANI、SRINIVASAN RAGHURAMAN 和 WANYUN GU。通过账户抽象实现自动付款。Technical Disclosure Commons，2022年。[41] Rob Behnke。智能合约形式化验证指南。Halborn博客，2023年。[42] Rafael Belchior、Benedikt Putz、Guenther Pernul、Miguel Correia、André Vasconcelos 和 Sérgio Guerreiro。SSIBAC：基于自主主权身份的访问控制。载于*2020 IEEE*第*19*届计算、通信中的信任、安全与隐私国际会议（*TrustCom*），2020年。[43] Cristiano Bellavitis 和 Paul P. Momtaz。去中心化自治组织（DAO）中的投票治理与价值创造。《创业洞察》，2025年。[44] Andréa Belliger 和 David J. Krieger。关于人工智能对齐的新视角。《人工智能时代的伦理：驾驭政治与安全》，2025年。[45] Eli Ben-Sasson、Alessandro Chiesa、Christina Garman、Matthew Green、Ian Miers、Eran Tro mer 和 Madars Virza。Zerocash：基于比特币的去中心化匿名支付。载于*2014 IEEE*安全与隐私研讨会，2014年。[46] Abdeljalil Beniiche。区块链预言机研究。*arXiv*，2020年。[47] Anjanava Biswas 和 Wrick Talukdar。构建智能体人工智能系统：创建能够推理、规划和适应的智能自主人工智能智能体。Packt出版社，2023年。

[48] Florian Blum、Benedikt Severin、Michael Hettmer、Philipp Hückinghaus 和 Volker Gruhn。运用区块链策略构建混合式去中心化应用——元交易示例。载于*2020 IEEE*区块链与加密货币国际会议（*ICBC*），2020年。[49] Sean Bowe、Alessandro Chiesa、Matthew Green、Ian Miers、Pratyush Mishra 和 Howard Wu。ZEXE：实现去中心化隐私计算。载于*2020 IEEE*安全与隐私研讨会（*SP*），2020年。[50] Lorenz Breidenbach、Christian Cachin、Benedict Chan、Alex Coventry、Steve Ellis、Ari Juels、Farinaz Koushanfar、Andrew Miller、Brendan Magauran、Daniel Moroz、Sergey Nazarov、Alexandru Topliceanu、Florian Tramèr 和 Fan Zhang。Chainlink 2.0：去中心化预言机网络演进的后续步骤。Chainlink Labs，2021年。[51] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sandhini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel Ziegler、Jeffrey Wu、Clemens Winter、Chris Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott Gray、Benjamin Chess、Jack Clark、Christopher Berner、Sam McCandlish、Alec Radford、Ilya Sutskever 和 Dario Amodei。语言模型是少样本学习器。载于 H. Larochelle、M. Ranzato、R. Hadsell、M.F. Balcan 和 H. Lin 编，《style id='9'>《神经信息处理系统进展》，第33卷，第1877–1901页。Curran Associates, Inc.，2020年。[52] Jonah Brown-Cohen、Geoffrey Irving 和 Georgios Piliouras。通过双重高效辩论实现可扩展的人工智能安全。*arXiv*，2023年。[53] Kwame Bryan。ERC-8001：代理协调框架。https://eips.ethereum.org/EIPS/eip-8001，2024年。访问日期：2026年1月6日。[54] Vitalik Buterin、Ansgar Dietrichs、Matt Garnett、Will Villanueva 和 Sam Wilson。EIP-2938：账户抽象。以太坊改进提案，2020年。[55] Vitalik Buterin、Yoav Weiss、Dror Tirosh、Shahaf Nacson、Alex Forshtat、Kristof Gazso 和 Tjaden Hess。ERC-4337：使用替代内存池实现账户抽象。以太坊改进提案，2021年。访问日期：2026年1月3日。[56] Cristian Nicolae Butincu 和 Adrian Alexandrescu。去中心化标识符与自主主权身份系统的设计方面。*IEEEAccess*，2024年。[57] Tianle Cai、Xuezhi Wang、Tengyu Ma、Xinyun Chen 和 Denny Zhou。作为工具制造者的大语言模型。*arXiv*，2023年。[58] Giulio Caldarelli。理解区块链预言机问题：行动呼吁。《Information》，2020年。[59] Yongcan Cao、Wenwu Yu、Wei Ren 和 Guanrong Chen。分布式多智能体协调研究的最新进展综述。*IEEE*工业信息学汇刊，2013年。[60] Agostino Capponi、Ruizhe Jia 和 Kanye Ye Wang。公有区块链中的最大可提取价值与配置低效。《金融经济学期刊》，2025年。[61] Miguel Castro 和 Barbara Liskov。实用拜占庭容错。载于第三届操作系统设计与实现研讨会论文集，1999年。[62] Miguel Castro 和 Barbara Liskov。实用拜占庭容错与主动恢复。*ACM*计算机系统汇刊（*TOCS*），2002年。[63] CertiK。什么是智能合约审计中的形式化验证？CertiK博客，2023年。[64] Chainlink。Chainlink Functions：面向智能合约的链下计算与数据。https://docs.chain.link/chainlink-functions，2026年。访问日期：2026年1月6日。[65] Stefanos Chaliasos、Marcos Antonios Charalambous、Liyi Zhou、Rafaila Galanopoulou、Arthur Gervais、Dimitris Mitropoulos 和 Ben Livshits。智能合约与去中心化金融安全工具：它们是否满足从业者的需求？载于*ICSE '24*：第*46*届软件工程国际会议论文集，2024年。[66] Patrick Chao、Edoardo Debenedetti、Alexander Robey、Maksym Andriushchenko、Francesco Croce、Vikash Sehwag、Edgar Dobriban、Nicolas Flammarion、George J. Pappas、Florian Tramèr、Hamed Hassani 和 Eric Wong。JailbreakBench：用于越狱大语言模型的开放鲁棒性基准。《神经信息处理系统进展》第37卷（NeurIPS 2024），2024年。[67] Szu-Tung Chen、Gülçin Ermiş 和 Alexei Sharpanskykh。面向自动化飞机地面保障的多智能体规划与协调。《机器人与自主系统》，2023年。[68] Thomas Chen、Hui Lu、Teeramet Kunpittaya 和 Alan Luo。零知识简洁非交互式知识论证综述。*arXiv*，2022年。[69] Xiaolan Chen、Jiayang Xiang、Shanfu Lu、Yexin Liu、Mingguang He 和 Danli Shi。评估医疗保健领域的大语言模型与智能体：临床应用中的关键挑战。《智能医学》，2025年。

[70] Kota Chin、Keita Emura 和 Kazumasa Omote。使用账户抽象的匿名且可问责合约钱包系统。《信息安全与应用》，2025年。[71] Tarun Chitra、Kshitij Kulkarni、Mallesh Pai 和 Theo Diamandis。基于意图的市场分析。*arXiv* 预印本 *arXiv:2403.02525*，2024年。[72] Usman W. Chohan。去中心化自治组织与治理问题。《去中心化自治组织》，2024年。[73] Kenneth Hsien Yung Chung、Dan Li 和 Peter Adriaens。技术赋能的可持续基础设施融资：区块链与去中心化预言机网络的案例。《技术预测与社会变迁》，2023年。[74] Coinbase。AgentKit：每个人工智能智能体都值得拥有一个钱包。GitHub 仓库，2024年。访问日期：2026年1月4日。[75] Coinbase。coinbase-agentkit-openai-agents-sdk：AgentKit 的 OpenAI Agents SDK 扩展。Python 包索引（PyPI），2024年。访问日期：2026年1月4日。[76] Vincent Conitzer 和 Caspar Oesterheld。协作式人工智能基础。在*AAAI* 人工智能会议论文集，2024年。[77] Simon Cousaert、Jiahua Xu 和 Toshiko Matsui。知识系统化：去中心化金融中的收益聚合器。在*2022*年 *IEEE* 区块链与加密货币国际会议（*ICBC*），2022年。[78] Jie Cui、Yihu Zhu、Hong Zhong、Qingyang Zhang、Chengjie Gu 和 Debiao He。面向跨域工业物联网的高效基于区块链的双向认证与会话密钥协商。*IEEE*物联网期刊，2024年。[79] Allan Dafoe、Edward Hughes、Yoram Bachrach、Tantum Collins、Kevin R. McKee、Joel Z. Leibo、Kate Larson 和 Thore Graepel。协作式人工智能中的开放问题。*arXiv* 预印本 *arXiv:2012.08630*，2020年。[80] Philip Daian、Steven Goldfeder、Tyler Kell、Yunqi Li、Xueyuan Zhao、Iddo Bentov、Lorenz Breidenbach 和 Ari Juels。闪电男孩 2.0：去中心化交易所中的抢先交易、交易重排序与共识不稳定性。*arXiv* 预印本 *arXiv:1904.05234*，2019年。[81] Philip Daian、Steven Goldfeder、Tyler Kell、Yunqi Li、Xueyuan Zhao、Iddo Bentov、Lorenz Breidenbach 和 Ari Juels。闪电男孩 2.0：去中心化交易所中的抢先交易、矿工可提取价值与共识不稳定性。在*IEEE*安全与隐私研讨会，2020年。[82] Konstantin Danilov、Roman Rezin、Ilya Afanasyev 和 Aleksandr Kapitonov。迈向基于区块链的机器人经济学：自主代理行为验证。在*IEEE* 智能系统国际会议，2018年。[83] M. Darshan、Matthieu Amet、Gautam Srivastava 和 Jorge Crichigno。实现下一代区块链系统跨链互操作性的架构。*IEEE*物联网期刊，2023年。[84] Adán Sánchez de Pedro、Daniele Levi 和 Luis Iván Cuende。Witnet：去中心化预言机网络协议。*arXiv*，2017年。[85] Zhihong Deng、Chunming Tang、Taotao Li、Parhat Abla、Qi Chen、Wei Liang 和 Debiao He。增强区块链跨链互操作性：综合综述。*arXiv* 预印本 *arXiv:2505.04934*，2025年。[86] Vipin Deval、Vimal Kumar Dwivedi、Abhishek Dixit、Alex Norta、Syed Attique Shah 和 Rahul Sharma。移动智能合约：探索可扩展性挑战与共识机制。*IEEE Access*，2024年。[87] Ivan Dimitrov 和 Wolfgang Prinz。加密货币钱包标准创新：分析 ERC4337 与基于零知识证明的提案。在*2024*年第六届区块链计算与应用国际会议（*BCCA*），2024年。[88] Diego Caples Div Garg、Andis Draguns、Nikil Ravi、Pranav Putta、Naman Garg、Prannay Hebbar、Youngchul Joo、Jindong Gu、Charles London、Christian Schroeder de Witt 和 Sumeet Motwani。REAL：基于真实网站的确定性模拟对自主代理进行基准测试。NeurIPS 2025，2025年。[89] D. Dolev 和 H. R. Strong。用于拜占庭协议的身份认证算法。《*SIAM* 计算杂志》，1983 年。[90] Ali Dorri 、 Salil S. Kanhere 和 Raja Jurdak。多智能体系统：综述。*IEEE Access*，2018年。[91] Abhishek Dutta 和 Yen-Che Hsiao。迈向自主代理：语言模型中的自适应规划、推理与行动。*arXiv* 预印本 *arXiv:2408.06458*，2024年。[92] Abul Ehtesham、Aditi Singh、Gaurav Kumar Gupta 和 Saket Kumar。代理互操作协议综述：模型上下文协议（MCP）、智能体通信协议（ACP）、智能体对智能体协议

(A2A), 和一个g实体网络协议 (ANP)。 *arXiv* *prep*印刷 *arXiv:2505.02279* , 2025年。

[93] Abul Ehtesham、Aditi Singh、Gaurav Kumar Gupta 和 Saurabh Kumar。《代理互操作协议的比较分析：MCP、ACP、A2A 和 ANP》。*arXiv*预印本，2025年。[94] Ulle Endriss、Nicolas Maudet、Fariba Sadri 和 Francesca Toni。《基于逻辑的代理通信协议》。《代理通信进展》，2004年。[95] Yimika Erinle、Yebo Feng、Jiahua Xu、Nikhil Vadgama 和 Paolo Tasca。《用于多方加密资产管理的共享托管钱包》。《未来互联网》，2025年。[96] Fireblocks 安永会计师事务所。《使用 Fireblocks 的多方计算解决方案进行数字资产托管和交易处理的最佳实践》。《白皮书》，2023年。[97] 以太坊。《智能合约形式化验证》。ethereum.org，2025年。[98] 以太坊名称服务。《以太坊名称服务（ENS）文档》。https://docs.ens.domains/，2023年。访问日期：2026年1月6日。[99] Shahinaz Kamal Ezzat、Yasmine N. M. Saleh 和 Ayman A. Abdel-Hamid。《区块链预言机：现状与研究方向》。*IEEEAccess*，2022年。[100] Tianyu Fan、Yuhao Yang、Yangqin Jiang、Yifei Zhang、Yuxuan Chen 和 Chao Huang。《AI-Trader：实时金融市场中的自主代理基准测试》。*arXiv*，2025年。[101] Yixuan Fan、Lei Zhang、Ruiyu Wang 和 Muhammad Ali Imran。《深入探讨去中心化自治组织中的投票：概念分析与评估框架提案》。《*IEEE Network*》，2024年。[102] Rainer Feichtinger、Robin Fritsch、Lioba Heimbach、Yann Vonlanthen 和 Roger Wattenhofer。《知识系统化：针对去中心化自治组织的攻击》。*arXiv*，2024年。[103] Paul Neil Feldman。《拜占庭协议的最优算法》。麻省理工学院，1988年。[104] Md Sadek Ferdous、Farida Chowdhury 和 Madini O. Alassafi。《利用区块链技术探索自主主权身份》。《IEEE Access》，2019年。[105] Md Sadek Ferdous、Mohammad Jabed Morshed Chowdhury、Mohammad A. Hoque 和 Alan Colman。《区块链共识算法：综述》。*arXiv*，2020年。[106] Josep-Lluis Ferrer-Gomila、M. Francisca Hinarejos 和 Andreu-Pere Isern-Deyà。《一种由区块链支持的公平合同签署协议》。《电子商务研究与应用》，36:100869，2019年。[107] Fireblocks。《Fireblocks MPC 钱包技术（白皮书/文档）》。https://www.fireblocks.com/，2024年。访问日期：2026年1月6日。[108] Flashbots。《Flashbots Protect：私有交易提交与最大可提取价值缓解》。https://docs.flashbots. net/flashbots-protect/overview，2026年。访问日期：2026年1月6日。[109] Kenneth Freeman、Norbert Gillem、Aidan R. Jones 和 Nishant Sharma。《城市空中交通运行意图的区块链案例研究》。NASA 艾姆斯研究中心，2023年。[110] Robin Fritsch、Marino Müller 和 Roger Wattenhofer。《分析去中心化治理中的投票权：谁控制着去中心化自治组织？》。《区块链：研究与应用》，2024年。[111] Xiang Fu、Huaimin Wang 和 Peichang Shi。《区块链共识算法综述：机制、设计与应用》。《中国科学：信息科学》，2021年。[112] Iason Gabriel。《人工智能、价值观与对齐》。《心智与机器》，2020年。[113] Maxim Gaina。《区块链与智能合约：基础知识及去中心化应用案例研究》。博洛尼亚大学，2018年。[114] Deep Ganguli、Liane Lovitt、Jackson Kernion、Amanda Askell、Yuntao Bai、Saurav Kadavath、Ben Mann、Ethan Perez、Nicholas Schiefer、Kamal Ndousse、Andy Jones、Sam Bowman、Anna Chen、Tom Conerly、Nova DasSarma、Dawn Drain、Nelson Elhage、Sheer El-Showk、Stanislav Fort、Zac Hatfield-Dodds、Tom Henighan、Danny Hernandez、Tristan Hume、Josh Jacobson、Scott Johnston、Shauna Kravec、Catherine Olsson、Sam Ringer、Eli Tran-Johnson、Dario Amodei、Tom Brown、Nicholas Joseph、Sam McCandlish、Chris Olah、Jared Kaplan 和 Jack Clark。《通过红队测试语言模型减少危害：方法、规模扩展行为与经验教训》。*arXiv*，2022年。[115] Sheng Gao、Tianyu Yu、Jianming Zhu 和 Wei Cai。《T-PBFT：一种基于 EigenTrust 的实用拜占庭容错共识算法》。《中国通信》，2019年。[116] Jesús García-Rodríguez、Rafael Torres Moreno、Jorge Bernal Bernabé 和 Antonio Skarmeta。《迈向隐私保护型可验证凭证的标准化模型》。载于*ARES 2021*：第*16*届国际可用性、可靠性与安全会议，2021年。

[117] Sanjam Garg、Abhishek Jain、Pratyay Mukherjee、Rohit Sinha、Mingyuan Wang 和 Yinuo Zhang。《带权密码学：多方计算、加密与签名》。载于年度国际密码学会议，第295–327页。施普林格，2023年。[118] Christina Garman、Matthew Green 和 Ian Miers。《去中心化匿名凭证》。密码学电子印本档案，2013年。[119] Valentina Gatteschi、Fabrizio Lamberti 和 Claudio Demartini。《智能合约技术》。《剑桥智能合约、区块链技术与数字平台手册》，2019年。[120] Peter Gaži、Aggelos Kiayias 和 Dionysis Zindros。《权益证明侧链》。载于*2019*年*IEEE*安全与隐私研讨会（*SP*），2019年。[121] Jonas Gebele、Timm Mutzel、Burak Oez 和 Florian Matthes。《利用机密计算区块链进行跨链密封投标拍卖》。*arXiv*预印本 *arXiv:2510.19491*，2025年。[122] Yifu Geng、Bo Qin、Qin Wang、Wenchang Shi 和 Qianhong Wu。《补贴桥：以补贴奖励跨区块链中继者》。载于国际信息与通信安全会议，2023年。[123] Asem Ghaleb、Julia Rubin 和 Karthik Pattabiraman。《AChecker：静态检测智能合约访问控制漏洞》。载于*2023*年*IEEE/ACM*第*45*届软件工程国际会议（*ICSE*），2023年。[124] Christopher Goes、Awa Sun Yin 和 Adrian Brink。《Anoma：面向全栈去中心化应用的统一架构》。Zenodo，2023年。[125] Oded Goldreich、Silvio Micali 和 Avi Wigderson。《证明仅能证明其有效性，或NP中的所有语言均具有零知识证明》。《ACM会刊》，1991年。[126] O. Gondrom。RFC 8785：JSON 规范化方案（JCS）。https://www.rfc-editor.org/rfc/ rfc8785，2020年。访问日期：2026年1月6日。[127] Jingxuan Gong、Peilin Yan、Yue Zhang、Hao An 和 Lei Liu。《区块链遇上大语言模型：双向集成的动态综述》。*arXiv*预印本，2024年。[128] Guido Governatori、Florian Idelberger、Zoran Milosevic、Regis Riveret、Giovanni Sartor 和 Xiwei Xu。《论法律合同、命令式与声明式智能合约以及区块链系统》。《人工智能与法律》，2018年。[129] Vincent Gramlich、Dennis Jelito 和 Johannes Sedlmeir。《最大可提取价值：当前认知、分类与开放研究问题》。《电子市场》，2024年。[130] (Significant) Gravitas。《AutoGPT：一项自主GPT-4实验（源代码仓库）》。https://github.com/Significant-Gravitas/AutoGPT，2023年。访问日期：2026年1月6日。[131] Kai Greshake、Sahar Abdelnabi、Shailesh Mishra、Christoph Endres、Thorsten Holz 和 Mario Fritz。《并非你所报名参与的：利用间接提示注入攻陷现实世界的大语言模型集成应用》。载于第*16*届*ACM*人工智能与安全研讨会论文集，第79–90页，2023年。[132] D.M. Groß。《迈向可信人工智能：机器学习中的形式化验证》。拉德堡德知识库，2024年。[133] Lewis Gudgeon、Daniel Perez、Dominik Harz、Benjamin Livshits 和 Arthur Gervais。《去中心化金融危机》。载于*2020*年加密谷区块链技术会议（*CVCBT*），2020年。[134] Jinnan Guo、Keke Gai、Liehuang Zhu 和 Zijian Zhang。《一种安全双向锚定多侧链方案》。《计算机科学讲义》，2020年。[135] Bálint Gyevnár 和 Atoosa Kasirzadeh。《面向所有人的人工智能安全》。《自然·机器智能》，2025年。[136] Juhyeng Han、Insu Yun、Seongmin Kim、Taesoo Kim、Sooel Son 和 Dongsu Han。《借助ScaleTrust实现可扩展且安全的硬件安全模块虚拟化》。《*IEEE/ACM*网络汇刊》，2023年。[137] Jungsuk Han、Jongsub Lee 和 Tao Li。《DAO治理综述：近期文献与新兴趋势》。《公司金融期刊》，2025年。[138] Andreas Happe 和 Jürgen Cito。《大语言模型驱动的进攻性安全基准测试实践：测试平台、度量指标与实验设计》。*arXiv*，2025年。[139] D. Hardt。RFC 6749：OAuth 2.0授权框架。https://www.rfc-editor.org/rfc/ rfc6749，2012年。访问日期：2026年1月6日。[140] Dick Hardt。RFC 6749：OAuth 2.0授权框架。IETF，2012年10月。访问日期：2026年1月7日。[141] Christopher G. Harris。《跨链技术：区块链互操作性的挑战与机遇》。载于*2023*年*IEEE*国际全层智能系统会议（*COINS*），2023年

.

[142] Ammar Hassan、Imran Makhdoom、Waseem Iqbal、Awais Ahmad 和 Asad Raza。《从信任到真实：缓解区块链预言机问题的进展》。《Journal of Network and Computer Applications》，217：103672，2023年。[143] Juan He、Rong Wang、Wei-Tek Tsai 和 Enyan Deng。《SDFS：面向智能合约的可扩展数据馈送服务》。载于*2019*年*IEEE*第十届软件工程与服务科学国际会议（*ICSESS*），2019年。[144] Shuangyu He、Qianhong Wu、Xizhao Luo、Zhi Liang、Dawei Li、Hanwen Feng、Haibin Zheng 和 Yanan Li。《基于社交网络的加密货币钱包管理方案》。*IEEE Access*，6：7654–7663，2018年。[145] Zheyuan He、Zihao Li、Sen Yang、He Ye、Ao Qiao、Xiaosong Zhang、Xiapu Luo 和 Ting Chen。《用于区块链安全的大语言模型：系统性文献综述》，2025年。[146] Lioba Heimbach 和 Roger Wattenhofer。《知识系统化：防止去中心化金融中的交易重排序操纵》。*arXiv*，2022年。[147] Maurice Herlihy。《原子跨链交换》。*arXiv*预印本 *arXiv:1801.09515*，2018年。[148] Bernhard Hommel。《GOALIATH：目标导向行为理论》。《心理学研究》，2022年。[149] Ivan Homoliak 和 Martin Perešíni。《知识系统化：加密货币钱包——基于认证因素的安全性综述与分类》。载于*IEEE*区块链与加密货币国际会议，2024年。[150] Xinyi Hou、Yanjie Zhao、Shenao Wang 和 Haoyu Wang。《模型上下文协议（MCP）：概况、安全威胁与未来研究方向》。*arXiv*预印本，2025年。[151] Hanjiang Hu、Changliu Liu、Na Li 和 Yebin Wang。《通过单轮强化学习训练用于多轮任务规划的任务推理大语言模型智能体》。*arXiv*预印本 *arXiv:2509.20616*，2025年。[152] William Hughes、Tobias Magnusson、Alejandro Russo 和 Gerardo Schneider。《利用基于哈希的授权和首选批处理器，在区块链上实现廉价且安全的元交易》。《区块链：研究与应用》，2023年。[153] Geoffrey Irving、Paul Christiano 和 Dario Amodei。《通过辩论实现人工智能安全》。*arXiv*，2018年。[154] Gerardo I uliano、Luigi Allocca、Matteo Cicalese 和 Dario Di Nucci。《Solidity智能合约中的自动化漏洞注入：一种用于基准测试开发的基于变异的方法》。*arXiv*，2025年。[155] Ibrahim T. Javed、Fares Alharbi、Badr Bellaj、Tiziana Margaria、Noel Crespi 和 Kashif Naseer Qureshi。《Health-ID：一种面向远程医疗的区块链去中心化身份管理方案》。医疗保健，2021年。[156] Nick R. Jennings。《承诺与惯例：多智能体系统协调的基础》。《知识工程评论》，1993年。[157] N.R. Jennings。《利用联合意图控制工业多智能体系统中的协作问题求解》。《人工智能》，1995年。[158] Jiaming Ji、Tianyi Qiu、Boyuan Chen、Borong Zhang、Hantao Lou、Kaile Wang、Yawen Duan、Zhonghao He、Lukas Vierling、Donghai Hong、Jiayi Zhou、Zhaowei Zhang、Fanzhi Zeng、Juntao Dai、Xuehai Pan、Kwan Yee Ng、Aidan O'Gara、Hua Xu、Brian Tse、Jie Fu、Stephen McAleer、Yaodong Yang、Yizhou Wang、Song-Chun Zhu、Yike Guo 和 Wen Gao。《人工智能对齐：综合综述》，2025年。[159] M. Jones、J. Bradley 和 N. Sakimura。RFC 7515：JSON网络签名（JWS）。https://www.rfc-editor.org/rfc/rfc7515，2015年。访问日期：2026年1月6日。[160] M. Jones、J. Bradley 和 N. Sakimura。RFC 7519：JSON Web 令牌（JWT）。https://www.rfc-editor.org/ rfc/rfc7519，2015年。访问日期：2026年1月6日。[161] Michael Jones、John Bradley 和 Nat Sakimura。RFC 7519：JSON Web 令牌（JWT）。IETF，2015年5月。访问日期：2026年1月7日。[162] Aleksandr Kapitonov、Sergey Lonshakov、Aleksandr Krupenkin 和 Ivan Berman。《由无人机构成的多智能体系统自主商业活动的区块链协议》。载于*IEEE*无人机系统研究、教育与发展研讨会，2017年。[163] Sayash Kapoor、Benedikt Stroebl、Zachary S. Siegel、Nitya Nadgir 和 Arvind Narayanan。《重要的人工智能智能体》。*arXiv*，2024年。[164] Gurdip Kaur、Arash Habibi Lashkari、Iman Sharafaldin 和 Ziba Habibi Lashkari。《智能合约与去中心化金融的安全性和威胁》。载于：《去中心化金融中的网络安全管理》。金融创新与技术。施普林格，*Cham*。，2023年。[165] Adrian Keller 和 Ralf Hofstetter。《加密资产托管：成功的私钥》。普华永道瑞士，2024年。

[166] Rami Khalil 和 Arthur Gervais。《Revive：链下支付网络的再平衡》。载于 ACM计算机与通信安全会议，2017年。[167] Abdul Ghaffar Khan、Amjad Hussain Zahid、Muzammil Hussain 和 Usama Riaz。使用硬件钱包和二维码保障加密货币安全。载于 2019年国际创新计算会议（ICIC），第 1–10 页，2019年。[168] Seongjin Kim、Sanguk Yun 和 Jungho Jang。使用 Android 和 OP-TEE 设计安全、便于用户使用的区块链模块化钱包。arXiv，2025年。[169] Chelsea Komlo。门限签名。《IEEE Security & Privacy》，2024年。[170] Queping Kong、Jiachi Chen、Yanlin Wang、Zigui Jiang 和 Zibin Zheng。DeFiTainter：检测去中心化金融协议中的价格操纵漏洞。载于 ISSTA 2023：第32届 ACM SIGSOFT 国际软件测试与分析研讨会论文集，2023年。[171] Nodirbek Korchiev、Akash Pateria、Vodelina Samatova、Sogolsadat Mansouri 和 Kemafor Anyanwu。驯服区块链上用户编程交易这一难题：一种声明式交易方法。arXiv预印本 arXiv:2411.02597，2024年。[172] Ahmed Kosba、Andrew Miller、Elaine Shi、Zikai Wen 和 Charalampos Papamanthou。Hawk：区块链密码学与隐私保护智能合约模型。载于 2016年IEEE安全与隐私研讨会（SP），2016年。[173] Vigneshram Krishnamoorthy、Wenhao Luo、Michael Lewis 和 Katia Sycara。将任务规划与规范感知推理相结合、用于社交机器人的计算框架。载于 2018年第27届IEEE机器人与人类交互通信国际研讨会（RO-MAN），2018年。[174] Kshitij Kulkarni、Theo Diamandis 和 Tarun Chitra。迈向最大可提取价值理论（一）：恒定函数做市商。arXiv，2023年。[175] Lars Kunze、Nick Hawes、Tom Duckett、Marc Hanheide 和 Tomáš Krajník。面向长期机器人自主性的人工智能：综述。《 IEEE 机器人与自动化快报》，2018 年。[176] Romain Laborde、Arnaud Oglaza、Samer Wazan、François Barrere、Abdelmalek Benzekri 和 David W. Chadwick。基于万维网联盟可验证凭证和 FIDO通用认证框架、以用户为中心的身份管理框架。载于 2020年第17届IEEE年度消费者通信与网络会议（CCNC），2020年。[177] LangChain。LangChain文档。https://python.langchain.com/，2026年。访问日期：2026年1月6日。[178] Bahareh Lashkari 和 Petr Musilek。区块链共识机制综合综述。IEEE Access，2021年。[179] Aris Leivadeas 和 Matthias Falkner。意图网络综述。IEEE通信调查与教程，2022年。[180] Wenkai Li、Jiuyang Bu、Xiaoqi Li、Hongli Peng、Yuanzheng Niu 和 Yuqing Zhang。去中心化金融安全综述：挑战与机遇。《沙特国王大学学报：计算机与信息科学》，2022年。[181] Zekun Li、Zhiyu Chen、Mike Ross、Patrick Huber、Seungwhan Moon、Zhaojiang Lin、Xin Luna Dong、Adithya Sagar、Xifeng Yan 和 Paul A Crook。通过函数调用将大语言模型用作零样本对话状态跟踪器。载于第62届计算语言学协会年会论文集（第1卷：长篇论文），第 8688–8704 页，2024年。[182] Zhaoshuo Li、Liang Cai、Weiwei Qiu、Fanglei Huang、Hao Duan 和 Chao Yuan。Cox：可靠的 BFT 共识算法运行时切换协议。《区块链：研究与应用》，2025年。[183] Percy Liang 等。语言模型的全面评估。arXiv预印本 arXiv:2211.09110，2023年。[184] Yehuda Lindell。快速安全的双参与方ECDSA签名。《密码学杂志》，2017年。[185] Huijun Liu、Bin Ji、Jie Yu、Shasha Li、Jun Ma 和 Miaomiao Li。针对大语言模型的离线文本对抗攻击。载于 2024年国际神经网络联合会议（IJCNN），2024年。[186] Jun Liu、Chunlan Liu、Mingwei Lin 和 Guangxia Xu。区块链共识机制综合综述：分析、应用与未来趋势。《计算机网络》，2025年。[187] Lu Liu、Sicong Zhou、Huawei Huang 和 Zibin Zheng。从技术到社会：基于区块链的去中心化自治组织概述。IEEE计算机学会开放期刊，2021年。

[188] Weiwen Liu、Xu Huang、Xingshan Zeng、Xinlong Hao、Shuai Yu、Dexun Li、Shuai Wang、Weinan Gan、Zhengying Liu、Yuanqing Yu、Zezhong Wang、Yuxian Wang、Wu Ning、Yutai Hou、Bin Wang、Chuhan Wu、Xinzhi Wang、Yong Liu、Yasheng Wang、Duyu Tang、Dandan Tu、Lifeng Shang、Xin Jiang、Ruiming Tang、Defu Lian、Qun Liu 和 Enhong Chen。Toolace：在大语言模型函数调用中赢得积分，2025年。[189] Xiao Liu、Hao Yu、Hanchen Zhang、Yifan Xu、Xuanyu Lei、Hanyu Lai、Yu Gu、Hangliang Ding、Kaiwen Men、Kejuan Yang、Shudan Zhang、Xiang Deng、Aohan Zeng、Zhengxiao Du、Chenhui Zhang、Sheng Shen、Tianjun Zhang、Yu Su、Huan Sun、Minlie Huang、Yuxiao Dong 和 Jie Tang。AgentBench：评估作为智能体的大语言模型。ICLR 2024，2024年。[190] Yang Liu、Jinlong He、Xiangyang Li、Jingwen Chen、Xinlei Liu、Song Peng、Haohao Cao 和 Yaoqi Wang。区块链智能合约执行机制概述。《工业信息集成期刊》，2024年。[191] Yi Liu、Gelei Deng、Yuekang Li、Kailong Wang、Zihao Wang、Xiaofeng Wang、Tianwei Zhang、Yepang Liu、Haoyu Wang、Yan Zheng 和 Yang Liu。针对大语言模型集成应用的提示注入攻击。*arXiv*，2023年。[192] Yupei Liu、Yuqi Jia、Runpeng Geng、Jinyuan Jia 和 Neil Zhenqiang Gong。提示注入攻击与防御的形式化及基准评测。载于 *第33届USENIX安全研讨会（USENIX Security 24）*，2024年。[193] Sin Kuang Lo、Xiwei Xu、Mark Staples 和 Lina Yao。区块链预言机可靠性分析。《计算机与电气工程》，2020年。[194] T. Logenthiran、Dipti Srinivasan 和 David Wong。微电网中面向分布式能源资源的多智能体协调。载于 *2008年IEEE可持续能源技术国际会议*，2008年。[195] Charles Lohest、Samy Bettaieb 和 Axel Legay。使用控制流图和机器学习自动检测智能合约漏洞。载于 *2024年计算机安全应用年会研讨会（ACSAC研讨会）*，2024年。[196] Shaofei Lu、Jingru Pei、Renke Zhao、Xiaochen Yu、Xuyang Zhang、Junyi Li 和 Guangzhong Yang。CCIO：一种基于预言机的联盟区块链跨链互操作方法。《传感器》，2023年。[197] Duc Anh Luong 和 Jong Hwan Park。使用零知识简洁非交互式知识论证的区块链隐私保护身份管理系统。*IEEEAccess*，2023年。[198]Michael Lustenberger、Florian Spychiger 和 Lukas Küng。去中心化自治组织——治理、技术与法律视角：第二届欧洲DAO研讨会（DAWO）论文集，瑞士苏黎世，2025年。*SpringerCham*，2025年。[199] Yash Madhwal、Yury Yanovich、Aleksandra Korotkevich、Daria Parshina、Nshteh Seropian、Stepan Gavrilov、Alex Nikolaev、S. Balachander 和 A. Murugan。通过免燃料费交易赋能区块链中的自主物联网设备。《区块链：研究与应用》，2025年。[200] Matteo Maffei。智能合约安全分析的形式化方法。2021年计算机辅助设计形式化方法会议（FMCAD），2021年。[201] Dahlia Malkhi 和 Pawel Szalachowski。有向无环图上的最大可提取价值（MEV）保护。*arXiv*，2022年。[202] Indrajeet Mandal、Jitendra Soni、Mohd Zaki、Morten M. Smedskjaer、Katrin Wondraczek、Lothar Wondraczek、Nitya Nand Gosvami 和 N. M. Anoop Krishnan。评估用于原子力显微镜自动化的大语言模型智能体。《自然-通讯》，2025年。[203] Tula Masterman、Sandi Besen、Mason Sawtell 和 Alex Chao。面向推理、规划和工具调用的新兴AI智能体架构概览：综述。*arXiv预印本*，2024年。[204] Huned Materwala、Shraddha M. Naik、Aya Taha、Tala Abdulrahman Abed 和 Davor Svetinovic。去中心化金融中的最大可提取价值：分类体系、检测与缓解。*arXiv*，2025年。[205] Carlo Mazzocca、Abbas Acar、Selcuk Uluagac、Rebecca Montanari、Paolo Bellavista 和 Mauro Conti。去中心化标识符与可验证凭证综述。*IEEE通信调查与教程*，2025年。[206] Leo McKee-Reid、Christoph Sträter、Maria Angelica Martinez、Joe Needham 和 Mikita Balesni。从诚实到欺骗：上下文内强化学习可能使诚实模型通过奖励作弊。*arXiv*，2024年。[207] Nikita Mehandru、Brenda Y. Miao、Eduardo Rodriguez Almaraz、Madhumita Sushil、Atul J. Butte 和 Ahmed Alaa。在临床环境中评估作为智能体的大语言模型。《npj数字医学》，2024年。[208] Feng Mi、Zhuoyi Wang、Chen Zhao、Jinghui Guo、Fawaz Ahmed 和 Latifur Khan。VSCL：使用深度学习自动检测智能合约漏洞。载于 *2021年IEEE区块链与加密货币国际会议（ICBC）*，2021年。

[209] Feng Mi、Chen Zhao、Zhuoyi Wang、Sadaf MD Halim、Xiaodi Li、Zhouxiang Wu、Latifur Khan 和 Bhavani Thuraisingham。智能合约自动化漏洞检测框架。*arXiv*，2023年。[210] Andrew Miller、Zhicheng Cai 和 Somesh Jha。智能合约与形式化方法的机遇。《形式化方法、验证与确认的应用：工业实践》，2018年。[211] Mahmoud Mohammadi、Yipeng Li、Jane Lo 和 Wendy Yip。大语言模型智能体的评估与基准测试：综述。载于第*31*届*ACM SIGKDD*知识发现与数据挖掘会议论文集（第*2*卷），KDD '25，第6129–6139页，美国纽约州纽约市，2025年。计算机协会。[212] Bhabendu Kumar Mohanta、Soumyashree S Panda 和 Debasish Jena。区块链技术中的智能合约与用例概述。载于*2018*年第*9*届计算、通信与网络技术国际会议（*ICCCNT*），2018年。[213] Mohsin。《人工智能代理基准解析：评估自主代理的7种经验证明的方法》。Services Ground，2025年。[214] Monika 和 Rajesh Bhatia。区块链互操作性解决方案。载于*2020*年计算、电气与电子智能技术国际会议（*ICSTCEE*），2020年。[215] Monika 和 Rajesh Bhatia。智能合约安全漏洞分析与形式化验证：综述。《自主与自适应通信系统国际期刊》，2025年。[216] Henrique Moniz。伊斯坦布尔BFT共识算法。*arXiv*，2020年。[217] Robbie Morrison、Natasha C. H. L. Mazey 和 Stephen C. Wingreen。DAO争议：主张一种新的公司治理形态?《区块链前沿》，2020年。[218] Xuechen Mu、Hankz Hankui Zhuo、Chen Chen、Kai Zhang、Chao Yu 和 Jianye Hao。用于促进协作式多智能体强化学习的分层任务网络规划。*arXiv*预印本 *arXiv:2306.08359*，2023年。[219] Fedor Muratov、Andrei Lebedev、Nikolai Iushkevich、Bulat Nasrulin 和 Makoto Takemiya。YAC：区块链的BFT共识算法。*arXiv*，2018年。[220] Praveen Ku mar Myakala、Anil Kumar Jonnalagadda 和 Sooraj George Thomas。意图驱动的去中心化：面向私有、可扩展自主系统的架构。载于*2025*年全球新兴技术大会（*GINOTECH*），2025年。[221] Abhimanyu Nag、Madhur Prabhakar 和 Tanuj Behl。去中心化交易所中兑换交易执行的需求双重巧合机制。*arXiv*预印本 *arXiv:2507.10149*，2025年。[222] Devishree Naidu、Bhushan Wanjari、Rohit Bhojwani、Saurabh Suchak、Rahul Baser 和 Niranjan Kumar Ray。利用零知识证明实现区块链中的隐私保护身份验证的高效智能合约。载于*2023*年*OITS*国际信息技术会议（*OCIT*），2023年。[223] Satoshi Nakamoto。比特币：一种点对点电子现金系统。Bitcoin.org，2008年。[224] Neha Narula、Willy Vasquez 和 Madars Virza。zkLedger：分布式账本的隐私保护审计。载于第*15*届*USENIX*网络系统设计与实现研讨会（*NSDI '18*），2018年。[225] Deepak Nathani、Lovish Madaan、Nicholas Roberts、Nikolay Bashlykov、Ajay Menon、Vincent Moens、Amar Budhiraja、Despoina Magka、Vladislav Vorotilov、Gaurav Chaurasia、Dieuwke Hupkes、Ricardo Silveira Cabral、Tatiana Shavrina、Jakob Foerster、Yoram Bachrach、William Yang Wang 和 Roberta Raileanu。MLGym：推进人工智能研究代理的新框架与基准，2025年。[226] Gelato Network。Gelato Web3服务：Web3函数。https://docs.gelato.network/web3-services/ web3-functions，2026年。访问日期：2026年1月6日。[227] Emanuel Onica 和 Ciprian Amariei。在以太坊去中心化应用中使用SGX支持元交易。载于第*22*届*IFIP*分布式应用与互操作系统国际会议（*DAIS*），2022年。[228] Kris Oosthoek。现金闪崩：去中心化金融中的网络威胁。*arXiv*预印本 *arXiv:2106.10740*，2021年。[229] OpenZeppelin。OpenZeppelin Defender：Autotasks文档。https://docs.openzeppelin.com/ defender/autotasks，2026年。访问日期：2026年1月6日。[230] Ariel Orda 和 Ori Rottenstreich。确保区块链交易排序的公平性。载于*2019*年*IEEE*区块链与加密货币国际会议（*ICBC*），2019年。[231] Longfei Ouyang、Yuan Yuan 和 Fei-Yue Wang。学习型市场：一种基于区块链和智能合约的人工智能协作框架。*IEEE*物联网期刊，2020年。

[232] Longfei Ouyang、Wei Zhang 和 Fei-Yue Wang。智能合约：让智能合约为区块链智能服务。《计算机与电气工程》，2022年。[233] Matthew J Page、Joanne E McKenzie、Patrick M Bossuyt、Isabelle Boutron、Tammy C Hoffmann、Cynthia D Mulrow、Larissa Shamseer、Jennifer M Tetzlaff、Elie A Akl、Sue E Brennan、Roger Chou、Julie Glanville、Jeremy M Grimshaw、Asbjørn Hróbjartsson、Manoj M Lalu、Tianjing Li、Elizabeth W Loder、Evan Mayo-Wilson、Steve McDonald、Luke A McGuinness、Lesley A Stewart、James Thomas、Andrea C Tricco、Vivian A Welch、Penny Whiting 和 David Moher。PRISMA 2020声明：系统综述报告指南的更新。*BMJ*，372，2021年。[234] Varatheepan Paramanayakam、Andreas Karatzas、Iraklis Anagnostopoulos 和 Dimitrios Stamoulis。少即是多：优化边缘设备上的函数调用与大语言模型执行。载于*2025*年欧洲设计、自动化与测试会议（*DATE*），2025年。[235] Bhargavi Paranjape、Scott Lundberg、Sameer Singh、Hannaneh Hajishirzi、Luke Zettlemoyer 和 Marco Tulio Ribeiro。ART：大语言模型的自动多步推理与工具使用，2023年。[236] Hyejin Park、Ivan Ureta 和 Boyoung Kim。制定衡量去中心化自治组织去中心化程度的维度与指标。《行政科学》，2023年。[237] Shishir G. Patil、Tianjun Zhang、Xin Wang 和 Joseph E. Gonzalez。Gorilla：连接海量 API 的大语言模型。载于 A. Globerson、L. Mackey、D. Belgrave、A. Fan、U. Paquet、J. Tomczak 和 C. Zhang（编），《神经信息处理系统进展》，第37卷，第126544–126565页。Curran Associates, Inc.，2024年。[238] Ethan Perez、Saffron Huang、Francis Song、Trevor Cai、Roman Ring、John Aslanides、Amelia Glaese、Nat McAleese 和 Geoffrey Irving。使用语言模型对语言模型进行红队测试。*arXiv*，2022年。[239] Robin D. Pesl。采用大语言模型实现自动化系统集成。《智能信息系统》。*CAiSE 2025*。《商业信息处理讲义》，第*557*卷。施普林格，*Cham*。，2025年。[240] Julien Piet、Vivek Nair 和 Sanjay Subramanian。MEVade：抗 MEV 区块链设计。载于*2023*年*IEEE*区块链与加密货币国际会议（*ICBC*），2023年。[241] Babu Pillai、Kamanashis Biswas 和 Vallipuram Muthukkumarasamy。使用交易实现基于区块链系统之间的跨链互操作性。《知识工程评论》，2020年。[242] Jeremy Pitt 和 Abe Mamdani。多智能体系统中的通信协议：开发方法与参考架构。《智能体通信问题》，2000年。[243] Apisara Pornprasith。DeFi收益聚合器中的资金流与绩效关系。朱拉隆功大学，2022年。[244] Purathani Praitheeshan、Lei Pan 和 Robin Doss。基于智能合约的以太坊链上钱包安全评估。《网络与系统安全》，2020年。[245] Alex Preukschat 和 Drummond Reed。自主主权身份。Manning 出版社，2021年。[246] R Priscilla、J Gold Beulah Patturose、Jerrish Bright J M、Sanjay M、Rahul P 和 Kalyani T。使用ERC-4337标准的去中心化投票应用。载于*2024*年通信、计算、智能材料与器件国际会议（*ICCCSMD*），2024年。[247] CoW Protocol。求解器竞赛规则。https://docs.cow.fi/cow-protocol/reference/core/ auctions/competition-rules，2025年。访问日期：2026年1月6日。[248] Aleksei Pupyshev、Dmitry Gubanov、Elshan Dzhafarov、Ilya Sapranidi、Inal Kardanov、Vladimir Zhuravlev、Shamil Khalilov、Marc Jansen、Sten Laureyssens、Igor Pavlov 和 Sasha Ivanov。Gravity：区块链无关的跨链通信与数据预言机协议。*arXiv*，2020年。[249] Pranav Putta、Edmund Mills、Naman Garg、Sumeet Motwani、Chelsea Finn、Divyansh Garg 和 Rafael Rafailov。Agent q：面向自主 AI 智能体的高级推理与学习。*arXiv*预印本 *arXiv:2408.07199*，2024年。[250] Peng Qian、Rui Cao、Zhenguang Liu、Wenqing Li、Ming Li、Lun Zhang、Yufeng Xu、Jianhai Chen 和 Qinming He。智能合约与 DeFi 安全的实证综述：漏洞检测与自动修复。*arXiv*预印本 *arXiv:2309.02391*，2023年。[251] Hongwei Qin、Shiliang Shao、Ting Wang、Xiaotian Yu、Yi Jiang 和 Zonghan Cao。移动机器人自主路径规划算法综述。《无人机》，2023年。[252] Kaihua Qin、Liyi Zhou、Benjamin Livshits 和 Arthur Gervais。利用闪电贷攻击 DeFi 生态系统：为了乐趣与利润。《金融密码学与数据安全》。FC 2021，2021年。[253] Changle Qu、Sunhao Dai、Xiaochi Wei、Hengyi Cai、Shuaiqiang Wang、Dawei Yin、Jun Xu 和 Ji-Rong Wen。大语言模型的工具学习：综述。*arXiv*，2024年。

[254] Fahad Rahman、Chafiq Titouna 和 Farid Nait-Abdesselam。《区块链与智能合约基础》。载于：*El Madhoun*、*N.*、*Dionysiou*、*I.*、*Bertin*、*E.*（编），《面向创新应用的区块链与智能合约技术》。施普林格，*Cham*。，2024年。[255] Christopher Rawles、Sarah Clinckemaillie、Yifan Chang、Jonathan Waltz、Gabrielle Lau、Marybeth Fair、Alice Li、William Bishop、Wei Li、Folawiyo Campbell-Ajala、Daniel Toyama、Robert Berry、Divya Tyamagundlu、Timothy Lillicrap 和 Oriana Riva。《Androidworld：面向自主代理的动态基准测试环境》，2025年。[256] Amith Kumar Reddy、Pradeep Katari、Venkat Rama Raju Alluri、Ajay Aakula 和 Ashok Kumar Reddy Sadhu。《从协议到实践：去中心化金融（DeFi）的详细分析》。《欧洲经济快报》，2019年。[257] Gliori Riza。《将硬件安全模块（HSM）用于文件加密与安全的研究》。载于*RTA-CSIT*论文集，2023年。[258] Andrea Rizzini、Marco Esposito、Francesco Bruschi 和 Donatella Sciuto。《具有概率合规性的隐私智能钱包》。*arXiv*，2025年。[259] Michael Rosenberg、Jacob White、Christina Garman 和 Ian Miers。《zk-creds：基于零知识简洁非交互式知识论证和现有身份基础设施的灵活匿名凭证》。载于*2023*年*IEEE*安全与隐私研讨会（*SP*），2023年。[260] Maayan Roth、Reid Simmons 和 Manuela Veloso。《用于协调式多智能体策略的去中心化通信策略》。《多机器人系统：从群体到智能自动机》第三卷，2005年。[261] Safe。《Safe文档》。https://docs.safe.global/，2025年。访问日期：2026年1月6日。[262] Christoph Salger。《去中心化争议解决：在仲裁中运用区块链技术和智能合约》。《佩珀代因争议解决法律期刊》，2024年。[263] Ronan Sandford、Liraz Siri、Dror Tirosh、Yoav Weiss、Alex Forshtat、Hadrien Croubois、Sachin Tomar、Patrick McCorry、Nicolas Venturo、Fabian Vogelsteller 和 Gavin John。《ERC-2771：原生元交易安全协议》。《以太坊改进提案》，2020年。[264] Pablo Santamaría、Llanos Tobarra、Rafael Pastor-Vargas 和 Antonio Robles-Gómez。《用于管理数字证据保管链的智能合约：一项实际案例研究》。《智慧城市》，2023年。[265] Carlos Santana 和 Laura Albareda。《区块链与去中心化自治组织（ DAO ）的兴起：整合模型与研究议程》。《技术预测与社会变迁》，2022年。[266] Francisco Santos。《The DAO：区块链治理中的百万美元教训》。塔林理工大学，2018年。[267] Timo Schick、Jane Dwivedi-Yu、Roberto Dessi、Roberta Raileanu、Maria Lomeli、Eric Hambro、Luke Zettlemoyer、Nicola Cancedda 和 Thomas Scialom。《Toolformer：语言模型能够自学使用工具》。载于 A. Oh、T. Naumann、A. Globerson、K. Saenko、M. Hardt 和 S. Levine（编），《神经信息处理系统进展》，第36卷，第68539–68551页。Curran Associates, Inc.，2023年。[268] Samuel Schmidgall、Rojin Ziaei、Carl Harris、Eduardo Reis、Jeffrey Jopling 和 Michael Moor。《AgentClinic：用于评估人工智能在模拟临床环境中表现的多模态智能体基准测试》。*arXiv*，2024年。[269] Johannes Sedlmeir、Reilly Smethurst、Alexander Rieger 和 Gilbert Fridgen。《数字身份与可验证凭证》。《商业与信息系统工程》，2021年。[270] István András Seres。《论区块链元交易》。载于*2020*年*IEEE*区块链国际会议（*Blockchain*），2020年。[271] Kaushal Shah、Dhruvil Lathiya、Naimish Lukhi、Keyur Parmar 和 Harshal Sanghvi。《去中心化金融协议的系统综述》。《国际智能网络期刊》，2023年。[272] Ali Shahidinejad 和 Jemal Abawajy。《物联网区块链辅助认证与会话密钥生成协议的全面分类及批判性综述》。《*ACM*计算调查》，2024年。[273] Erfan Shayegani、Md Abdullah Al Mamun、Yu Fu、Pedram Zaree、Yue Dong 和 Nael Abu-Ghazaleh。《对抗攻击揭示的大语言模型漏洞综述》。*arXiv*，2023年。[274] Wazen M. Shbair、Eugene Gavrilov 和 Radu State。《以太坊区块链的基于硬件安全模块（HSM）的密钥管理解决方案》。载于*2021*年*IEEE*区块链与加密货币国际会议（*ICBC*），2021年。[275] Ibne Farabi Shihab、Sanjeda Akter 和 Anuj Sharma。《检测并缓解强化学习系统中的奖励破解：一项全面的实证研究》。*arXiv*，2025年。[276] R. K. Shyamasundar。《为何难以克服、减少或监管最大可提取价值（MEV）？》。载于*2025*年区块链国际会议，2026年。

[277] Riyaz Sikora 和 Michael J. Shaw。《用于信息系统协调与集成的多智能体框架》。《管理科学》，1998年。[278] Paulo Caetano da Silva、Gabriel Antônio Lopes de Castro 和 Daniel José Diaz。《区块链平台之间的跨链互操作性》。《Contabilidad y Negocios》，2025年。[279] Amritraj Singh、Kelly Click、Reza M. Parizi、Qi Zhang、Ali Dehghantanha 和 Kim-Kwang Raymond Choo。《区块链网络中的侧链技术：考察与最新进展综述》。《*Journal of Network and Computer Applications*》，2020年。[280] Aniket Kumar Singh、Inzimam Ul Hassan、Gaganjot Kaur、Shanu Kumar 和 Anmol。《通过单例入口合约和验证型支付主合约实现账户抽象》。载于*2023*年第二届国际边缘计算与应用会议（*ICECAA*），2023年。[281] Maria Sommerhalder。《硬件安全模块》。《数据保护与加密技术趋势》，2023年。[282] Taylor Sorensen、Jared Moore、Jillian Fisher、Mitchell Gordon、Niloofar Mireshghallah、Christopher Michael Rytting、Andre Ye、Liwei Jiang、Ximing Lu、Nouha Dziri、Tim Althoff 和 Yejin Choi。《迈向多元对齐的路线图》。*arXiv*，2024年。[283] Manu Sporny、Ted Thibodeau Jr、Ivan Herman、Gabe Cohen、Michael B. Jones、Grant Noble、Dave Longley、Daniel C. Burnett、Brent Zundel、Kyle Den Hartog 和 David Chadwick。《可验证凭证数据模型 v2.0》。《W3C建议标准》，2025年。[284] Venkat Krishna Srinivasan、Zhen Dong、Banghua Zhu、Brian Yu、Hanzi Mao、Damon Mosk-Aoyama、Kurt Keutzer、Jiantao Jiao 和 Jian Zhang。《NexusRaven：一种适用于商业用途的函数调用语言模型》。载于*NeurIPS 2023*年决策基础模型研讨会，2023年。[285] Lukas Stockburger、Georgios Kokosioulis、Alivelu Mukkamala、Raghava Rao Mukkamala 和 Michel Avital。《区块链赋能的去中心化身份管理：公共交通领域中的自主主权身份案例》。《区块链：研究与应用》，2021年。[286] Lijun Sun、Yijun Yang、Qiqi Duan、Yuhui Shi、Chao Lyu、Yu-Cheng Chang、Chin-Teng Lin 和 Yang Shen。《多样化应用中的多智能体协调：综述》。*arXiv*预印本 *arXiv:2502.14743*，2025年。[287] Mohammad Hossein Tabatabaei、Roman Vitenberg 和 Narasimha Raghavan Veeraragavan。《理解区块链：定义、架构、设计与系统比较》。《Computer Science Review》，2023年。[288] Mona Taghavi、Jamal Bentahar、Hadi Otrok 和 Kaveh Bakhtiyari。《用于提高区块链预言机可靠性的强化学习模型》。《Expert Systems with Applications》，2023年。[289] Tahlil Tahlil、Syead Tanvir Islam Movin、Sania Azhmee Bhuiyan 和 Mahabub Alahi。《免燃料费链上密码管理器：基于EVM的平台比较分析》。载于*2024*年*IEEE*区块链与加密货币国际会议（*ICBC*），2024年。[290] Yuto Takei。《不是你的多方计算，就不是你的币》。*ArXiv*，2023年。[291] Yiyi Tao、Yixian Shen、Hang Zhang、Yanxin Shen、Lun Wang 和 Chuanqi Shi。《大语言模型对抗对抗攻击的鲁棒性》。载于*2024*年第四届人工智能、机器人与通信国际会议（*ICAIRC*），2024年。[292] Thales TCT。《密码密钥管理最佳实践》。白皮书，2022年。[293] Datopic Technologies。《通过形式化分析进行智能合约分析：威胁与漏洞》。Datopic Technologies博客，2023年。[294] Tenderly。《如何使用Tenderly SDK模拟交易》。https://docs.tenderly.co/tenderly-sdk/tutorials-and-quickstarts/how-to-simulate-transactions-with-tenderly-sdk，2025年。访问日期：2026年1月6日。[295] Tenderly。《Tenderly文档：交易模拟与调试》。https://docs.tenderly.co/，2026年。访问日期：2026年1月6日。[296] TheSequence。《The Sequence Knowledge #560：智能体基准测试的精彩世界》。Substack，2025年。[297] Louis Tremblay Thibault、Tom Sarry 和 Abdelhakim Senhaji Hafid。《使用Rollup实现区块链扩容：综合综述》。*IEEEAccess*，2022年。[298] Alejandro Torreño、Eva Onaindia、Antonín Komenda 和 Michal Štolba。《协作式多智能体规划：综述》。《*ACM*计算调查》，第50卷第6期，2017年11月。

[299] Tuan-Dung Tran、Kiet Anh Vo、Nguyen Binh Thuc Tram、Ngan Nguyen Bui Kim、Phan The Duy 和 Van-Hau Pham. 通过侧链集成和有效时间密钥数据访问控制增强区块链互操作性.《数据工程与通信技术讲义》，2023年.[300] Petar Tsankov、Andrei Dan、Dana Drachsler Cohen、Arthur Gervais、Florian Buenzli 和 Martin Vechev. Securify：智能合约的实用安全分析. *arXiv*，2018年.[301] Turnkey. Turnkey文档. https://docs.turnkey.com/，2025年. 访问日期：2026年1月6日.[302] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Ł ukasz Kaiser 和 Illia Polosukhin. 注意力就是你所需要的一切. 载于 I. Guyon、U. Von Luxburg、S. Bengio、H. Wallach、R. Fergus、S. Vishwanathan 和 R. Garnett（编），《神经信息处理系统进展》，第30卷. Curran Associates, Inc.，2017年.[303] Anand Kumar Vedantham. 人工智能增强的企业对齐与平台治理的 CONNECT 框架.《计算机科学与技术研究期刊》，2025年.[304] Frank von Martial. 智能体的合作与协调，第15–57页. 施普林格柏林海德堡，柏林，海德堡，1992年.[305] 万维网联盟. 去中心化标识符（DID）v1.0. https://www.w3.org/TR/did-core/，2022年. 访问日期：2026- 01-06.[306] Dylan Waldner. 人工智能安全、对齐与伦理（AI SAE）. *arXiv*，2025年.[307] Zhiguo Wan、Yan Zhou 和 Kui Ren. zk-AuthFeed：利用经过身份验证的零知识证明保护智能合约的数据馈送. *IEEE*可信赖与安全计算汇刊，2023年.[308] Qin Wang 和 Shiping Chen. 账户抽象分析. 载于*2023*年*IEEE*区块链国际会议（*Blockchain*），2023年.[309] Shuai Wang、Liwei Ouyang、Yong Yuan、Xiaochun Ni、Xuan Han 和 Fei-Yue Wang. 区块链赋能的智能合约：架构、应用与未来趋势. *IEEE*系统、人与控制论汇刊：系统，2019年.[310] Wei Wang、Jingjing Song、Guangquan Xu、Yidong Li、Hao Wang 和 Chunhua Su. ContractWard：以太坊智能合约的自动化漏洞检测模型. *IEEE*网络科学与工程汇刊，2021年.[311] Weijie Wang、Haotian Deng、Jinwen Liang 和 Chuan Zhang. 实现用于以太坊虚拟机区块链意图执行的隐私保护优化器架构. 载于第六届*ACM*区块链与安全关键基础设施国际研讨会，2025年.[312] Xin Wang、Sisi Duan、James Clavin 和 Haibin Zhang. 区块链中的拜占庭容错：从协议到用例.《*ACM*计算调查》，2022年.[313] Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、brian i chter、Fei Xia、Ed Chi、Quoc V Le 和 Denny Zhou. 思维链提示能够引导大语言模型进行推理. 载于 S. Koyejo、S. Mohamed、A. Agarwal、D. Belgrave、K. Cho 和 A. Oh（编），《神经信息处理系统进展》，第35卷，第24824–24837页. Curran Associates, Inc.，2022年.[314] Frank Wiener. MPC门限签名钱包：简介. Blockdaemon博客，2019年.[315] James Williams. 互操作性协议：促进区块链网络中的跨链通信. ResearchGate，2024年.[316] Bernd W. Wirtz、Jan C. Weyerer 和 Ines Kehl. 人工智能治理：基于风险与指南的综合框架.《政府信息季刊》，2022年.[317] Georg Wölflein、Dyke Ferber、Daniel Truhn、Ognjen Arandjelovic 和 Jakob Nikolas Kather. 大语言模型智能体打造智能体工具. 载于 Wanxiang Che、Joyce Nabende、Ekaterina Shutova 和 Mohammad Taher Pilehvar（编），《第*63*届计算语言学协会年会论文集（第*1*卷：长篇论文）》，第26092–26130页，奥地利维也纳，2025年7月. 计算语言学协会.[318] Gavin Wood. 以太坊：一种安全的去中心化通用交易账本. 以太坊项目黄皮书，2014年.[319] Michael Wooldridge. 理性智能体推理. 麻省理工学院出版社，2000年.[320] Huan Wu、Chentao Yue、Yixing Fan、Yonghui Li、David Flynn 和 Lei Zhang. 分布式拜占庭容错共识的半个世纪：设计原则与演进路径. *arXiv*，2025年。

[321] Xiaohua Wu、Tingbo Zhang、Jing Wang、Jun Cheng 和 Zirui Wang. 一种基于公证人方案和群签名的分布式跨链机制. 《沙特国王大学学报：计算机与信息科学》，2023年.[322] Zihui Wu、Haichang Gao、Jianping He 和 Ping Wang. 函数调用的阴暗面：大语言模型越狱的途径. 载于第*31*届国际计算语言学会议论文集，2025年.[323] Zhiheng Xi、Yiwen Ding、Wenxiang Chen、Boyang Hong、Honglin Guo、Junzhe Wang、Xin Guo、Dingwen Yang、Chenyang Liao、Wei He、Songyang Gao、Lu Chen、Rui Zheng、Yicheng Zou、Tao Gui、Qi Zhang、Xipeng Qiu、Xuanjing Huang、Zuxuan Wu 和 Yu-Gang Jiang. AgentGym：在多样化环境中评估和训练基于大语言模型的智能体. 载于 Wanxiang Che、Joyce Nabende、Ekaterina Shutova 和 Mohammad Taher Pilehvar（编），《第*63*届计算语言学协会年会论文集（第*1*卷：长篇论文）》，第27914–27961页，奥地利维也纳，2025年7月. 计算语言学协会 .[324] Jiahua Xu 、 Krzysztof Paruch 、 Simon Cousaert 和 Yebo Feng. 知识系统化：采用自动做市商协议的去中心化交易所. 《*ACM*计算调查》，2023年.[325] Weikai Xu、Chengrui Huang、Shen Gao 和 Shuo Shang. 面向工具学习的基于大语言模型的智能体综述：W. Xu 等. 《数据科学与工程》，第1–31页，2025年.[326] Xiwei Xu、Ingo Weber、Mark Staples、Liming Zhu、Jan Bosch 和 Len Bass. 基于区块链的系统架构设计分类体系. 载于*2017*年*IEEE*国际软件架构会议（*ICSA*），2017年.[327] Yifan Xu、Xiao Liu、Xueqiao Sun、Siyi Cheng、Hao Yu、Hanyu Lai、Shudan Zhang、Dan Zhang、Jie Tang 和 Yuxiao Dong. AndroidLab：Android自主智能体的训练与系统化基准测试. ACL 2025，2025年.[328] Zhiyue Yan、Xi Zhao、Yang (Alison) Liu 和 Xin (Robert) Luo. 区块链驱动的去中心化身份管理：跨学科综述与研究议程. 《信息与管理》，2024年.[329] Ruohan Yang、Lu Liu 和 Gang Feng. 多智能体系统分布式协调最新进展综述. 《无人系统》，2022年.[330] Shunyu Yao、Jeffrey Zhao、Dian Yu、Nan Du、Izhak Shafran、Karthik R Narasimhan 和 Yuan Cao. ReAct：在语言模型中协同推理与行动. 载于第十一届国际学习表征会议，2023年.[331] Maofan Yin、Dahlia Malkhi、Michael K. Reiter、Guy Golan Gueta 和 Ittai Abraham. HotStuff：从区块链视角看拜占庭容错共识. *arXiv*，2018年.[332] Eric Yu、Wang Yue、Shi Jianzheng 和 Wang Xun. 基于区块链的人工智能代理与自主世界基础设施. 载于*2024*年*IEEE*人工智能会议（*CAI*），第278–283页，2024年.[333] Jiahao Yu、Xingwei Lin、Zheng Yu 和 Xinyu Xing. GPTFUZZER：利用自动生成的越狱提示对大语言模型进行红队测试. *arXiv*，2023年.[334] Jingang Yu、Minghao Tang 和 Shu Li. MTdt：为工业工作流的数据追踪实现元交易. 载于*2021*年第七届国际计算机与通信会议（*ICCC*），2021年.[335] Lei Yuan、Jianhao Wang、Fuxiang Zhang、Chenghe Wang、ZongZhang Zhang、Yang Yu 和 Chongjie Zhang. 通过去中心化队友建模实现多智能体激励通信. 载于*AAAI* 人工智能会议论文集，2022年.[336] Siyu Yuan、Kaitao Song、Jiangjie Chen、Xu Tan、Yongliang Shen、Kan Ren、Dongsheng Li 和 Deqing Yang. EASYTOOL：通过简洁的工具指令增强基于大语言模型的智能体. 载于 Luis Chiruzzo、Alan Ritter 和 Lu Wang（编），《*2025*年美洲分会计算语言学协会会议论文集：人类语言技术（第*1*卷：长篇论文）》，第951–972页，美国新墨西哥州阿尔伯克基，2025年4月. 计算语言学协会.[337] Yinlong Yuan、Zhu Liang Yu、Zhenghui Gu、Xiaoyan Deng 和 Yuanqing Li. 一种用于解决奖励破解问题的新型多步强化学习方法. 《应用智能》，2019年.[338] Fan Zhang、Ethan Cecchetti、Kyle Croman、Ari Juels 和 Elaine Shi. Town Crier：面向智能合约的经过身份验证的数据馈送. 载于*2016*年*ACM SIGSAC*计算机与通信安全会议论文集，2016年.[339] Gengrui Zhang、Fei Pan、Yunhao Mao、Sofia Tijanic、Michael Dang’ana、Shashank Motepalli、Shiquan Zhang 和 Hans-Arno Jacobsen. 在拜占庭帝国中达成共识：拜占庭容错共识算法综合综述. *arXiv*，2023年。

[340] Hanrong Zhang、Jingyuan Huang、Kai Mei、Yifei Yao、Zhenting Wang、Chenlu Zhan、Hongwei Wang 和 Yongfeng Zhang。《智能体安全基准（ASB）：大语言模型智能体中攻击与防御的形式化及基准测试》。*arXiv*，2024年。[341] Wenbin Zhang、Yuan Yuan、Yanyan Hu、Shaohua Huang、Shengjiao Cao 和 Anuj Chopra。《区块链上的隐私保护投票协议》。载于*2018*年第*11*届*IEEE*云计算国际会议（*CLOUD*），2018年。[342] Yuanyu Zhang、Shoji Kasahara、Yulong Shen、Xiaohong Jiang 和 Jianxiong Wan。《面向物联网的基于智能合约的访问控制》。*IEEE*物联网期刊，2019年。[343] Zijian Zhang、Xin Lu、Meng Li、Jincheng An、Yang Yu 和 Hao Yin。《基于区块链的密封投标拍卖隐私保护方案》。*IEEE*可信赖与安全计算汇刊，2024年。[344] Bao-Kun Zheng、Lie-Huang Zhu、Meng Shen、Feng Gao、Chuan Zhang、Yan-Dong Li 和 Jing Yang。《基于区块链的可扩展隐私保护数据共享》。《计算机科学与技术》期刊，2018年。[345] Weiyu Zhong、Ce Yang、Wei Liang、Jiahong Cai、Lin Chen、Jing Liao 和 Naixue Xiong。《拜占庭容错共识算法：综述》。《电子学》，2023年。[346] Weikang Zhou、Xiao Wang、Limao Xiong、Han Xia、Yingshuang Gu、Mingxu Chai、Fukang Zhu、Caishuang Huang、Shihan Dou、Zhiheng Xi、Rui Zheng、Songyang Gao、Yicheng Zou、Hang Yan、Yifan Le、Ruohui Wang、Lijun Li、Jing Shao、Tao Gui、Qi Zhang 和 Xuanjing Huang。《EasyJailbreak：大语言模型越狱统一框架》。*arXiv*，2024年。[347] Huijuan Zhu、Lei Yang、Liangmin Wang 和 Victor S. Sheng。《智能合约安全分析方法综述》。*IEEE*服务计算汇刊，2024年。[348] Kaijie Zhu、Jindong Wang、Jiaheng Zhou、Zichen Wang、Hao Chen、Yidong Wang、Linyi Yang、Wei Ye、Yue Zhang、Neil Zhenqiang Gong 和 Xing Xie。《PromptRobust：评估大语言模型对抗提示鲁棒性的探索》。*arXiv*，2023年。[349] Andy Zou、Zifan Wang、Nicholas Carlini、Milad Nasr、J. Zico Kolter 和 Matt Fredrikson。《针对对齐语言模型的通用且可迁移的对抗攻击》。*arXiv*，2023年。[350] Jing Zou、Shungeng Zhang 和 Meikang Qiu。《针对大语言模型的对抗攻击》。载于知识科学、工程与管理国际会议，2024年。[351] Patrick Züst、Tejaswi Nadahalli 和 Roger Wattenhofer。《以太坊中的三明治攻击分析与防范》。苏黎世联邦理工学院，2021年。[352] Burak Öz、Filip Rezabek、Jonas Gebele、Felix Hoops 和 Florian Matthes。《先到先服务区块链上的MEV提取技术研究》。*arXiv*，2024年。