arXiv:2603.09875v1 [cs.MA] 10 Mar 2026

速度的官僚主义：内存一致性模型与多智能体授权撤销之间的结构等价

Vladyslav
ParakhinOkta 高级数据工
程师

摘要

支撑传统身份与访问管理的时间性假设，在智能体式执行模式下会彻底崩塌。对于人类操作者而言，60 秒的撤销窗口在操作层面微不足道，但在每刻 100 次操作的情况下，却会导致约 6×10^3 次未经授权的 API 调用；在 AWS Lambda 的规模下，这一数字接近 6×10^5。这是一致性问题，而不仅仅是延迟问题；授权撤销领域的研究在很大程度上未能认识到这一点。

我定义了一个能力一致性系统（CCS）——一个由以下元素构成的元组 $\langle A,\ C,\ \Sigma,\ \delta,\ \alpha,\ \mathcal{B}\rangle$——并构造一个状态映射函数 $\varphi:\ \Sigma_{\mathrm{MESI}} \to \Sigma_{\mathrm{auth}}$，使其在有界陈旧语义下保留转换结构。安全性定理（定理3.1）对执行次数型释放一致性导向的一致性机制（RCC）策略下未经授权的操作数量给出上界$D_{\mathrm{rcc}} \le n$，且该上界独立于智能体速度 v——这与时间界定策略的 $O(v \cdot$生存时间）缩放方式有本质区别。

评估通过基于时钟刻度的离散事件模拟进行：三个业务情境化场景、四种策略，以及每种配置十个确定性随机种子（种群规模 σ）。急切失效产生 500.0 ± 0 次未授权操作，耗时 $v = 100$，$\Delta_{\mathrm{network}} = 5$；基于租约的 TTL 产生 6 次，耗时 000.0 ± 0；延迟按需检查产生 2 次，耗时 400.0 ± 0；RCC 耗时 $n = 50$，产生 50.0 ± 0 次（$\sigma = 0$，精确验证了定理 3.1）。在异常触发的撤销下，与租约 TTL 相比，RCC 的未授权操作减少了 $184\times$（16.0 ± 3.7 次，而非 2 次，950.8 ± 3.6）。

贡献： (1) 建立缓存一致性与授权撤销语义之间的形式化状态映射；(2) 提出速度脆弱性指标 $V_v = v \cdot \mathrm{TTL}$，将智能体速度确立为一个基本安全维度；(3) 提出一个以释放一致性为基础的操作次数有界凭证模型，在同步边界强制实现一致性；(4) 开展可复现的多轮评估，并在 https://github.com/hipvlady/prizm 发布源代码。

主题： 多智能体系统 (cs.MA)；密码学与安全 (cs.CR)；分布式、并行与集群计算 (cs.DC)

目录

1 The　一致性框架 **3**

2 理论基础 **3**

2.1 缓存一致性要点 .32.2 故障遏制 .42.3 信息论框架 .4

3 形式化模型 **4**

3.1 能力一致性系统 .43.2 状态映射函数 .53.3 损害界限函数 .6

4 相关工作 **7**

4.1 多智能体故障模式 .74.2 组织可靠性模型 .74.3 信任与安全治理 .74.4 委托与授权标准 .8

5 授权即一致性问题 **8**

5.1 与一致性无关（急切）撤销 .85.2 一致性导向（宽松）撤销 .8

6 MESI映射 **9**

7 架构 **9**

7.1 将执行次数界限作为释放一致性 .97.2 权威服务 .97.3 异构一致性 .10

8 评估 **11**

8.1 仿真方法 .118.2 场景配置 .118.3 场景 1 — 银行业务级联撤销 .118.4 场景 2 — CRM 高速代理 .128.5 场景3 — 异常自动撤销 .138.6 成本效益分析.14

9 讨论 **14**

9.1 范围与适用性 .149.2 局限性 .14

10 结论 **15**

1 一致性框架

该智能体已经执行了 47 次未经授权的 API 调用。授权机构于 $t_r = 0$ 撤销了其凭证。生存时间将于 $t = 60$ 过期。当前授权栈中没有任何机制能检测到这一点。

受感染的智能体系统在采用 生存时间的凭证管理机制时，会在自我失效前累积 $V_v = v \cdot$生存时间内 的未授权操作，其数量与速度和窗口长度均成线性关系。在 $v = 10,000$ 每秒事务数（AWS Lambda 规模）和 60 秒的 生存时间下，$V_v = 6 \times 10^5$。Replit 事件 [13]——一个删除了生产数据库的 AI 编码智能体——是这一故障类别的具体实例，而非边缘案例。损害上界并非授权架构中的偶然因素，而是系统所处一致性机制的一种结构性属性。

为人类操作员开发的 IAM 协议沿袭了隐含假设：以分钟为单位衡量的会话超时、在人类交互速率下可容忍的撤销窗口（~ 1 请求/秒$\rightarrow 60$ 未授权操作），以及作为设计权衡而可接受的最终一致性。自主智能体同时违背了所有这些假设。OAuth 2.0、OIDC 及其衍生协议并非为能够递归委派、高频决策循环并在数千个并发会话中并行执行的实体而设计。

OpenID 基金会 [12] 指出，离线衰减委派链中的撤销问题“在很大程度上尚未解决”。Chan 等人 [6]认为，人工智能系统需要独立且可验证的身份，而非改用人类凭证。Nagabhushanaradhya [11] 提出了 OIDC-A，这是一种面向智能体身份的 OpenID Connect 扩展。Mei 等人 [10] 将大语言模型智能体视为 AIOS 抽象下的可调度进程。我将这一推理延伸至其逻辑结论：如果智能体是进程，那么它们在授权层就需要一致性协议，而不仅仅是身份验证或速率限制。

核心论点是，在有界陈旧语义下，多智能体委托链中的授权撤销，在操作层面等同于共享内存多处理器中的缓存一致性。我对这种等价关系进行了形式化，为四种撤销策略推导出有界陈旧性保证，并通过仿真对其进行评估。

贡献。
1. 形式等价。 定义能力一致性系统（CCS），并给出状态映射函数 φ，将 MESI 状态映射为授权状态，并在有界陈旧语义下保持转换结构（第3节）。**2. 速度脆弱性指标。** 将基于 TTL 的方法所具有的损害潜力形式化为 $V_v = v \cdot \mathrm{TTL}$，并证明该上界依赖于速度，而操作次数上界独立于速度（第 3节）。**3. 操作次数有界凭证模型。** 将 OpenID 基金会提出的执行次数方案 [12] 置于释放一致性理论的基础上 [19], ，证明其在一致性边界处与获取/释放同步原语形式等价（第7节）。**4. 可复现评估。** 在三个场景中评估四种策略，并进行多次运行的统计聚合（运行 10 次，随机种子为 0–9）： $120\times$ 在CRM 场景中，执行次数方案相较于 TTL 所减少的未授权操作数； $184\times$ 异常场景中的结果（第8节）。

2 理论基础

2.1 缓存一致性基础

Sorin、Hill 和 Wood [19] 将一致性定义为：对某个内存位置的读取应返回最近写入的值，且写操作应按序串行化。 MESI 协议——包含四种稳定状态（已修改、独占、共享、无效）及定义明确的转换——是典型的

实现方式。在稳定状态之间，瞬态状态 [19, 第 6 章，§6.4。1] (记作 XYZ：“原为X，正在转换为 Y，等待事件 Z”）用于描述进行中的操作；其持续时间恰好对应于我在授权类比中所说的损害窗口。

一致性策略分为两类 [19, 第2章，第2节。3]： 与一致性无关 (单写多读- 强制执行、同步失效）和一致性导向 (宽松、有界陈旧性）。GPU架构 [18, 1] 采用后者，实现时间一致性（基于租约的自失效）和释放一致性（同步点一致性）。第8 节评估的授权策略与这些类别完全对应。

2.2 故障遏制

Reason 的瑞士奶酪模型 [15] 认为，当相互独立的防御层具有相互对齐的故障模式时，就会发生故障。Vijayaraghavan 等人 [21]将其应用于多智能体可靠性研究，证明具有正交故障模式的级联评议层能够捕获 92.1% 的错误。对于生存时间- 基授权而言，对齐问题在结构上完全相同：故障模式（时间窗口）与威胁向量（操作速度）精确对齐。操作计数上限与速度正交——它们构成了瑞士奶酪模型所要求的屏障。

2.3 信息论框架

香农 [17] 指出，通过噪声信道进行可靠通信需要以有效带宽为代价引入冗余。授权机构与代理之间的授权信道存在噪声（延迟、有损且可能遭受拜占庭故障）。一致性强制开销——重新验证检查、心跳信号——相当于授权层的纠错码：以吞吐量成本换取撤销可靠性。 RCC 开销公式开销$_{\mathrm{RCC}} = \Delta_{\mathrm{revalidation}}/n$ （第 8.6 节）以数值方式精确地刻画了这种对应关系。

3 形式化模型

3.1 能力一致性系统

定义 3.1（能力一致性系统）。 一个 能力一致性系统 （CCS）是一个元组 $\langle A,\ C,\ \Sigma,\ \delta,\ \alpha,\ \mathcal{B}\rangle$ 其中：

- $A = \{a_1, \ldots, a_m\}$ 是一个有限代理集合；
- $C = \{c_1, \ldots, c_k\}$ 是一个有限的能力集合（类似于内存位置）；
- $\Sigma = \{M, E, S, I\}$ 是稳定授权状态的集合；
- $\delta\colon\ \Sigma \times \mathcal{E} \to \Sigma$ 是事件 $\mathcal{E} =$ {授予、撤销、委托、内省、耗尽、过期}上的状态转移函数；
- $\alpha\colon\ A \times C \to \Sigma$ 将每个代理-能力对映射到其当前授权状态；
- $\mathcal{B}\colon\ \Sigma \to 2^{\mathcal{O}}$ 是操作 $\mathcal{O}$上的许可操作函数。

能力有效性映射到缓存行有效性： $s \in \{M, E, S\}$ 是带有有效数据的缓存块； I 是失效缓存行——若未经授权机构进行一致性填充，则不得进行任何读或写。形式上， $\mathcal{B}(I) = \emptyset$精确地镜像了该硬件约束。

SWMR 适配。 对于任意能力 c在逻辑时间 t：

$$\left|\{a \in A : \alpha_t(a, c) = M\}\right| \leq 1.$$

至多一个智能体可同时持有可委托状态；多个智能体可以S 并发处于该状态。这与 [19, 第 2 章第 2 节4]中的单写多读不变量相呼应。

定义 3.2 (授权状态机)。 授权状态机 $\mathcal{M} = (\Sigma \cup \Sigma_T,\ \mathcal{E},\ \delta)$ 在定义 *3.1* 的基础上，加入瞬态状态 $\Sigma_T = \{EIA, SIA, MIC, ISG, IED\}$，遵循 *[19,* 第*6]*章的记法。

有效转换列于表1。

表 1：授权状态机的有效转换。

From	事件	To	语义
I	授予_共享	S	已签发能力（基于角色的）
I	授予 _ 独占	E	已签发即时凭证
E	委托	M	代理子委托
E	撤销	EIA	撤销进行中，确认待处理
EIA	ack	I	撤销已确认
S	撤销	SIA	撤销进行中（共享）
M	撤销 _ 级联	MIC	级联撤销，子代理待处理
MIC	all_acks	I	所有受委托者已确认
S	耗尽	I	操作计数耗尽（RCC）
S	过期	I	TTL 已过期（时间一致性）

无效转换——$I \to M$（未事先授予的委派）、$S \to M$（绕过独占性升级）——会被 δ拒绝。瞬态状态并非装饰：EIA 和MIC 的持续时间，正是未授权操作不断累积的时间区间。将瞬态状态压缩为原子转换的形式化分析，会系统性地低估损害窗口。

3.2 状态映射函数

定义 3.3（状态映射）。设 $\Sigma_{MESI} = \{M_{hw}, E_{hw}, S_{hw}, I_{hw}\}$ *[19]*和 $\Sigma_{\text{auth}} = \{M, E, S, I\}$。映射 φ：$\Sigma_{MESI} \to \Sigma_{\text{auth}}$ 为：

$$\varphi(M_{hw}) = M \text{（可写 } \mapsto \text{ 可委托）} \varphi(E_{hw}) = E \text{（独占读取 } \mapsto \text{ 唯一凭证持有者）} \varphi(S_{hw}) = S \text{（共享读取 } \mapsto \text{ 基于角色的池化访问）} \varphi(I_{hw}) = I \text{（无效 } \mapsto \text{ 已撤销）}$$

命题 3.1（结构等价）。映射 φ 保留转换结构：对于硬件 *MESI* 协议中的每个有效转换 (s_1, e_{hw}, s_2)，在 $\mathcal{M}$中都存在一个对应的有效转换 $(\varphi(s_1)\ e_{\text{auth}}, \varphi(s_2))$，其中 e_{auth} 是与硬件事件 e_{hw}相对应的授权域事件。

证明概要。通过对照 [19, 表 63]进行了穷举验证。硬件 *BusRd* 映射为 授予_共享；*BusRdX* 映射为 授予_独占；*Snoop-Invalidate* 映射为 撤销；写回 映射为委托（范围衰减）。单写多读 按构造得到保留（定义 3.1）。

□

我采用结构等价 而非同构，因为授权域引入了硬件 MESI 中不存在的状态转换——尤其是耗尽 和过期——并施加了额外约束：子级 .范围 $\subseteq$父级.范围。硬件模型是授权模型的严格子集，这在架构层面很方便：已知的硬件界限可作为授权的下界估计。

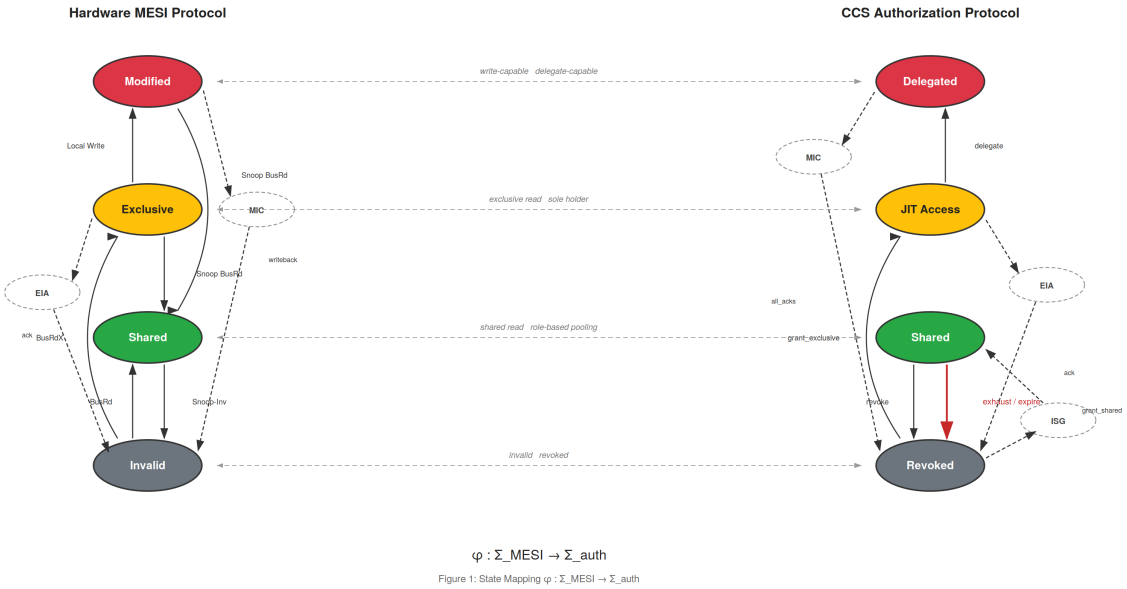

图 1：状态映射 φ ： $\Sigma_{\text{MESI}} \to \Sigma_{\text{auth}}$。左侧：硬件 MESI 状态机及其转换触发条件（$BusRd$、$BusRdX$、$Snoop\text{-}Inv$）。右侧：授权状态机及其对应的触发条件（授予、委托、撤销）。瞬态状态以虚线节点表示（EIA、 MIC、 ISG）。连接箭头上标注了事件对应关系。

等价范围。 CCS 在有界陈旧语义下实现了操作层面 的等价，而非语义同构。硬件 MESI 通过物理互连属性保证内存可见性；CCS 则通过协议层面的策略约束实现类似的授权可见性。特定于领域的扩展——信任评分、委托有向无环图、范围衰减——使授权模型超出了硬件域的范围。CCS 是一种受一致性机制启发的授权协议，而非完整的 MESI 实例。

3.3 损害上界函数

定义 3.4 （速度脆弱性）。对于速度为 v （操作/刻）的智能体和生存时间为 Δt （时步）的凭证：

$$V_v(v, \Delta t) = v \cdot \Delta t.$$

其意义不在于算术本身——这很简单——而在于代理速度成为了一项一等安全参数：这是以人为中心的身份与访问管理文献中所缺少的维度。

定义 3.5 （损害上界）。对于撤销策略 π，损害上界 D_π 是授权机构发起撤销后可执行的未授权操作数量上限：

策略 π	D_π	上界类型
急切 （一致性无关）	$D_{\text{eager}} \le v \cdot \Delta_{\text{network}}$	有时间界限的、依赖速度的
惰性 （使用时检查）	$D_{\text{lazy}} \le v \cdot (\Delta_{\text{network}} + \Delta_{\text{check}})$	有时间界限的、依赖速度
租约 （时间一致性）	$D_{\text{lease}} \le v \cdot$生存时间	有时间界限的、依赖速度
执行计数 （*RCC*）	$D_{\text{rcc}} \le n$	操作次数有界， 速度-独立的

定理 3.1 (RCC 安全性界限)。 在采用执行计数策略且预算为 n 的 *CCS* 中，撤销启动后每项能力的未授权操作上限受 n 限制，且与代理速度无关 v。

证明概要。 设 t_r 为授权机构撤销智能体 a持有的能力 c 的时间。设 k为自最近一次获取以来， a 针对 c 已执行的操作数。预算耗尽时（$k = n$，即释放点）， a 必须联系授权机构以获取新的凭证（获取）。授权机构已于 t_r撤销 c，因此拒绝此次获取。在 t_r之后，未授权操作的最大数量为 $n - k \leq n$。关键在于，此界限针对的是操作，而非时间：速度更快的智能体会更早耗尽 n，但无法超过 n。无论 v如何，该证明均成立。 □

备注 1 (定理 3.1适用范围)。 该上界 $D_{\text{rcc}} \leq n$ 适用于 每项能力。在委托链中，多个智能体持有各自独立预算的不同能力，级联过程中未授权操作的总数可能随随机调度而变化——银行业务场景（σ =13。0 表示执行计数，第*8.3*节）便是一个具体例证——但每项能力都遵守确定性上界。在全部 *120* 次实验运行中均未出现界限违规（$4 \times 10 \times 3$种配置），这毫无例外地证实了这一点。

备注 2 (与速率限制的区别)。 速率限制器限制吞吐量； RCC 则在操作边界 使凭证本身失效。在速率限制下，已撤销的智能体会受到限流，但仍持有其认为有效的凭证。在 RCC下，智能体会在同步边界发现撤销，并转换为 $\alpha(a,\ c) \to I$。没有新授权，智能体便无法恢复——这就是获取语义。表 *5*中可以观察到这种行为差异： RCC 是唯一会强制重新获取（1.0 ± 0 重新验证），而非悄然使状态过期的策略。

推论 3.1.1。 对于急切策略： $D_{\text{eager}} = v \cdot \Delta_{\text{network}}$。在 $v = 100$, $\Delta_{\text{network}} = 5$ 时： D_{eager} =500。只有当 $\Delta_{\text{network}} < 1/v$ 时，急切失效才能使未授权操作趋近于零——而在现实的高速部署中，这一条件无法实现。

4 相关工作

4.1 多智能体故障模式

在 ReAct [25] 和 AutoGen [24]的基础上——这些工作记录了从单一 LLM向复合型智能体系统的转变——但两个框架都没有处理各个智能体的授权状态，这一缺口会直接带来安全后果。思维链 [23]提高了推理的可靠性，却未触及授权状态。Cemri 等人 [5] 提出了迄今最全面的故障分类体系：MAST (1,600+ 条轨迹、七个框架）识别出三个类别中的十四种故障模式。他们所说的“智能体间失调”——智能体偏离共享预期——与我形式化定义的一致性违规直接对应：智能体认为自身处于状态 S，但授权机构已将其转移至 I。这就是授权层面的陈旧读取语义。

4.2 组织可靠性模型

基于 [21]——“对手团队”架构——我将他们的组织隐喻从语义一致性（生成器/批评器分离）扩展到安全一致性（执行/授权分离）。Huang 等人 [9] 提供了实证依据：分层监督导致 5.5% 的性能下降，而扁平拓扑为 23.7%，其检查器机制可恢复 96.4% 的故障智能体错误。权威服务在CCS 中于授权层扮演检查器角色。2–10% 的RCC 开销（第8.6节）与 Huang 等人的基准相当。

4.3 信任与安全治理

Raza 等人 [14]将 TRISM应用于智能体多智能体系统，并定义监控指标（组件协同评分、工具使用效能）。他们的框架规定了需要监控的内容；CCS则界定了执行机制：信任分数下降会触发具体的

状态转换 $S \to I$，发生在状态机中。CSA 框架 [7] 阐述了“持续授权”——权限会持续接受评估，而非仅在会话启动时授予一次。我的架构将其实现为一个协议层，信任信号通过该层转化为一致性操作。

4.4 委派与授权标准

Schwenkschuster 等人 [16] 形式化了跨域身份链；South 等人 [20] 提供了密码学委托原语。两者都关注链的建立。我关注链的拆除——撤销在委托有向无环图中的一致传播。建立是这个问题中较容易的一半。

范围衰减有两种方式：在线方式（RFC 8693 令牌交换 [4]）和离线方式（Biscuits [3]，Macaroons [2]）。两者都是无状态的范围管理机制。CCS 提供了互补的有状态一致性层：即使委派经过了正确的身份验证和范围衰减，撤销的时间一致性问题依然存在；正如 [12] 所承认的，这个问题“在很大程度上尚未解决”。这两种机制可以组合使用，并非相互竞争。

5 将授权视为一致性问题

在集中式系统中，授权是原子的：一次数据库检查就能得出二元且实时的结果。在分布式智能体系统中，授权是缓存的——每个令牌和签名断言都是某项曾在 t_0存在的权限的缓存副本。分布式授权文献探讨了这种缓存对性能的影响，却在很大程度上忽视了其对一致性的影响。

智能体系统难以保障安全，根源在于将陈旧令牌问题视为延迟问题，而从结构上看，它其实是一致性问题。在 Sorin 等人的分类法中， [19],一致性协议负责管理写入（撤销）向所有读取者（智能体）的传播。“写入”指撤销事件；“读取者”指持有缓存凭证的智能体。一致性策略决定了写入传播的速度及其成本。

5.1 与一致性无关的（急切）撤销

此类别通过同步方式强制执行单写多读：授权机构会先使所有已持有的令牌无效，然后才确认撤销。在硬件术语中，这就是基于总线的侦听。这种脆弱性是结构性的，而非偶然的：只要有一个智能体离线，流程就会挂起或故障开放。Cemri 等人的MAST 分类法 [5] 将“系统设计问题”列为主要故障类别；同步撤销恰恰会在授权层引入这种故障模式。推论3.1.1进一步指出了一项限制：当$\Delta_{\text{network}} > 0$——在 $v = 100$时，急切策略无法实现未授权操作为零，因为 5 个时钟周期的传播窗口内会出现 500 次未授权操作，无论广播有多激进都无济于事。

5.2 一致性导向（宽松）撤销

系统接受在有界陈旧性条件下进行异步撤销传播。问题在于——有趣的设计空间究竟在哪里——是什么限定了陈旧度。时间一致性（生存时间）以时间限定陈旧度，从而产生 $O(v \cdot$生存时间）定义中的漏洞3.4。释放一致性（执行次数）以操作限定陈旧度，根据定理3.1消除了速度依赖性。这两种限定类型之间的定性差异，是本文的核心实证主张。

6 MESI映射

我根据定义 φ 3.3，将其应用于代理授权生命周期（图1）。每个稳定状态都有具体的操作语义：

已修改 $\rightarrow$ **委托授权**。 智能体持有委托权——以Biscuit/Macaroon的术语而言 [3, 2]，它可以附加衰减限制。来自 M 的撤销需要级联通知：$M \xrightarrow{revoke_cascade} MIC$。进行中的 MIC 间隔正是级联损害累积之处。

独占 $\rightarrow$ **即时访问**。 一次性凭证；不可共享。映射到CSA“即时访问” [7]。该凭证的存在时间精确等于任务时长——这是时间界定策略下最短的 D_π 窗口。

共享 $\rightarrow$ **基于角色的池化**。 多个代理同时持有只读副本（例如，“读取S3存储桶”）。这是大多数OAuth访问令牌的默认状态。陈旧性风险具有乘法效应：在被逐一置为无效之前，所有副本均为陈旧。

无效 $\rightarrow$ **已撤销**。 $\mathcal{B}(I) = \emptyset$。若无一致性事务，该智能体无法继续 .

瞬态状态 EIA——已发送撤销请求，确认待处理——是急切策略部署中大部分损害发生的阶段。将撤销视为瞬时完成的形式化分析，因遗漏这一时间段而系统性地低估了损害窗口。

7 架构

7.1 将执行次数界限视为释放一致性

在 GPU 释放一致性 [19, 1, 第10章，第10.1节4]，中，一致性在同步点强制实施：获取 操作会加载最新状态，释放 操作会提交并公开写入。在这些边界之间，处理器使用本地缓存运行，不产生一致性流量。

每个凭证都会附带操作预算 n 。每执行一次操作，计数器就递减一次。在 $k = n$——预算耗尽、到达释放点时，智能体会联系授权机构以获取新凭证（获取）。如果能力在 n次操作的临界区内已被撤销，则获取请求会被拒绝。根据定理 3.1，损害上限为 n ，无论 v如何。

OpenID 基金会 [12] 另行提出了“受执行次数约束的凭证”。我证明，这在形式上等价于释放/获取周期：每个 n次操作构成一个临界区，其边界分别是获取（凭证刷新）和释放（预算耗尽）。据我所知，授权领域文献尚未阐明这种等价关系；这是将硬件一致性界限引入授权域的关键理论贡献。

7.2 权威服务

权威服务将策略决策点（PDP） [8] 与一致性目录控制器相结合：

1. **能力注册表**。 所有有效能力、委派DAG以及范围衰减链的基准事实。强制执行子.范围 $\subseteq$ 父.范围作为硬性不变量。（在网络分区下，注册表可能与代理的缓存视图发生偏离；见第9.2节。）2.**信任评分器**。 实现CSA 持续授权 [7] 以及 Raza 等人的TRISM监控 [14]。行为分析产生动态信任分数；分数低于阈值$\tau < 0.4$时触发自动撤销： $\alpha(a,\ c) \rightarrow I\ \forall c \in C_a$。阈值 τ被视为场景固定；跨 $\tau \in [0.2, 0.6]$ 的敏感性分析予以延后。

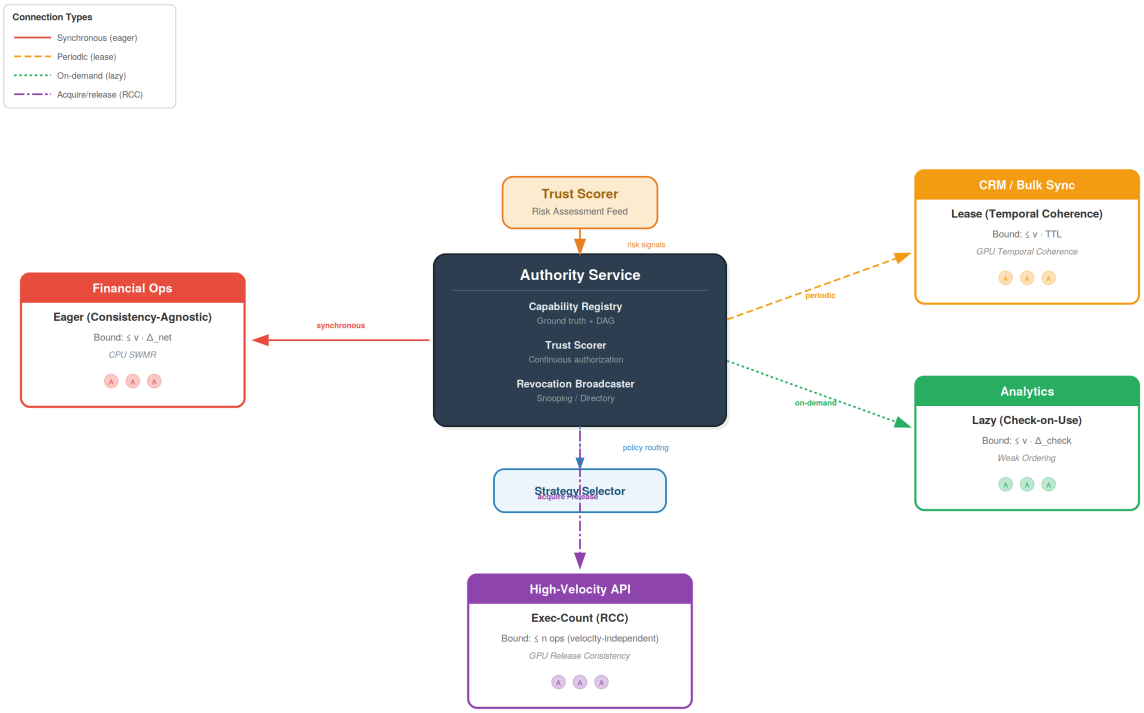

Figure 2: Heterogeneous Coherence Architecture

图 2：异构一致性架构。授权机构（中央）连接四个代理集群，每个集群均标注了一致性策略。消息类型：同步（急切，实线）、周期性（租约，虚线）、按需（惰性，点线）、获取/释放（执行计数，双线）。信任评分器将风险评估结果传递给策略选择器。

3. **撤销广播器**。面向 ≤ 25 代理的监听式广播；对更大规模群体采用基于目录的单播 [19, 第6章，§6.4。3]。监听到目录的转换点在本研究中未经实证验证——已标记为待定校准事项。

7.3 异构一致性

在 CPU+GPU 系统中的复合一致性模型基础上 [19, 第10章，§10.2]，不同代理类别在不同策略下运行。策略分配直接映射自一致性类别分类法：高完整性操作（金融交易）需要一致性无关语义（急切）；高吞吐量操作（批量 CRM 更新）可容忍一致性导向的松弛（租约、执行计数）。每台设备在其自身的一致性模型下运行；权威服务提供跨域排序——即全局一致性控制器。

表 2：按代理上下文分配的异构一致性策略。

代理上下文	一致性策略	损害上界	硬件类比
金融业务	急切（与一致性无关）	$\leq v \cdot \Delta_{\text{net}}$	CPU 单写多读
CRM／批量同步	基于租约的（时间性的）	$\leq v$ 生存时间	GPU 时间性的
分析	惰性（使用时检查）	$\leq v \cdot \Delta_{\text{check}}$	弱排序
高吞吐量 API	执行计数 (RCC)	$\leq n$ 次操作	GPU 释放一致性

8 评估

8.1 仿真方法

我们用 Python 3.11 实现了一个基于时钟周期的离散事件模拟器。每个时钟周期都是一个不可分割的调度单位；同一时钟周期内的所有操作在逻辑上同时发生。时钟刻度抽象将一致性动态与部署特定延迟隔离开来。要映射到特定部署，可将时钟周期计数乘以该部署的调度间隔（例如，一个时钟周期 $= 10$ ms 意味着 5 个时钟周期的延迟为 $= 50$ ms）。

方法论注意事项。 当操作集中在时钟周期边界时，时钟周期模拟器会引入轻微的时间混叠伪影。在概率调度场景（银行业务、异常）中，每个智能体每个时钟周期都会通过独立的伯努利抽样来处理 $p = 0.5$ 和 $p = 0.7$ 处的操作。对于确定性 CRM 场景（每时钟周期恒定执行 $v = 100$ 操作），时钟周期对齐完全消除了这一问题，并使所有随机种子下都得到 $\sigma = 0$ ——这虽是正确的结果，但值得明确说明，而不应将其作为出人意料的结果报告。

多次运行聚合。 每种策略-场景配置均使用确定性种子 0–9 执行 10 次。报告总体均值和总体标准差 σ （而非样本 s），因为在指定的种子范围内，这 10 次运行构成了完整的实验总体。所有场景配置均以 YAML 文件的形式与源代码一同发布，地址为 https://github.com/hipvlady/prizm。

8.2 场景配置

表 3：三个业务情境化场景的仿真参数。

参数	场景 1（银行业务）	场景 2（CRM）	场景 3（异常）
代理	10	1	5
委派深度	3	1	1
动作模型	概率（$p = 0.5$）	确定性（100/时钟周期）	概率（$p = 0.7$）
种子	0–9（10运行次数）	0–9（10运行次数）	0–9（10运行次数）
网络延迟	10时步 时钟周期 100	5时步 时钟周期 0	10 个时步
撤销触发			自动（信任分数）
生存时间（租约，时步）	120	60	3,000
执行计数 n	60	50	100
惰性检查间隔	40 时步	23 时步	10 时步
异常突发	N/A	N/A	12 操作/刻
信任阈值 τ	0.8	0.8	0.4
信任衰减	0.3	0.3	0.5
持续时间（时钟周期）	200	120	300

8.3 场景 1 —BankingCascade 撤销

支付处理代理在时钟周期 100 遭到入侵。三级委托链（用户 →A → B → C）需要进行级联撤销。概率调度（$p = 0.5$）会产生真实的跨随机种子方差。

急切策略达到观测到的最紧界限（14.9 ± 3.2）——但这个非零值值得明确说明。在一个包含 10 个智能体、深度为 3 的委托树中，10 个时钟周期的网络延迟允许传播期间仍有操作发生，这与推论3.1.1一致。急切策略并未实现零值；它实现的是期望意义上的 $v \cdot \Delta_{\text{network}}$ 。

表 4：银行级联结果（均值 $\pm\,\sigma$，运行次数 10 次，随机种子 0–9）。

指标	急切	租约（120 个时钟周期）	惰性（40 个时钟周期）	RCC（$n = 60$）
未授权操作	14.9 ± 3.2	29.9 ± 2.6	35.3 ± 2.9	32.3 ± 13.0
最大陈旧度（时步）	10.0 ± 0	20.0 ± 0	23.0 ± 0	37.0 ± 21.6
已发送消息	3.0 ± 0	3.0 ± 0	3.0 ± 0	3.0 ± 0
界限违规	0	0	0	0

RCC 表现出的方差最高（$\sigma = 13.0$）。这并不违反定理 3.1：该定理为 $D \le n$ 每项能力设定了界限，而界限违规次数为零，证实每项能力都遵守了其预算。方差源于不同委派深度上的释放/获取边界与随机动作调度之间的相互作用——视随机种子而定，不同代理会在级联传播波前的不同阶段耗尽预算。这一现象是在界限之内的随机陈旧度，而非违反界限的随机情况。

所有策略的结果相近（范围为 14.9–35.3），反映出概率调度下的有效速度较低。CRM 场景（第 8.4 节）旨在最大限度地展现速度依赖性。

8.4 场景 2——CRM 高速代理

销售同步代理以每时钟周期 100 次操作的速度运行。凭证在时钟周期 0 时被撤销——最坏情况下，整个模拟窗口都处于撤销之后。

表5：CRM 高吞吐量结果（平均值 $\pm\,\sigma$，运行 10 次，随机种子 0–9）。

指标	急切	租约（60 时步）	惰性（23 时步）	RCC（$n = 50$）
未授权操作	500.0 ± 0	**6，000.0 ± 0**	2，400.0 ± 0	$\mathbf{50.0 \pm 0}$
最大陈旧度（时步）	5.0 ± 0	60.0 ± 0	24.0 ± 0	0.0 ± 0
重新验证次数	0.0 ± 0	0.0 ± 0	0.0 ± 0	1.0 ± 0
界限违规	0	0	0	0

$\sigma = 0$ 在所有策略和所有随机种子下——这是确定性调度的直接结果（$v = 100$ 恒定）。零方差是一项验证性结果：它证明定义 3.5中的损害界限是精确的，而非随机近似。

表 6：预测损害界限与观测损害界限（CRM 场景）。

策略	预测值 D_π	观测值	匹配
急切：$v \cdot \Delta_{\text{net}} = 100 \times 5$	500	500.0 ± 0	精确
惰性：$v \cdot (\Delta_{\text{net}} + \Delta_{\text{check}}) = 100 \times (1 + 23)$	2,400	$2{,}400.0 \pm 0$	精确
租约：$v \cdot$生存时间 $= 100 \times 60$	6,000	$6{,}000.0 \pm 0$	精确
RCC：n	50	50.0 ± 0	精确（定理3.1）

从 $120\times$ 租约到 RCC $(6,\,000 \to 50)$ 的降幅是核心实证结果（图 3）。生存时间 造成的损害随 $O(v \cdot$生存时间$)$变化；RCC 造成的损害则封顶于 $O(n)$，与速度无关。急切策略相比租约实现了 $12\times$ 的改进，但其损害仍比RCC高一个数量级，而且会引入同步耦合，带来可用性风险（第5节）。

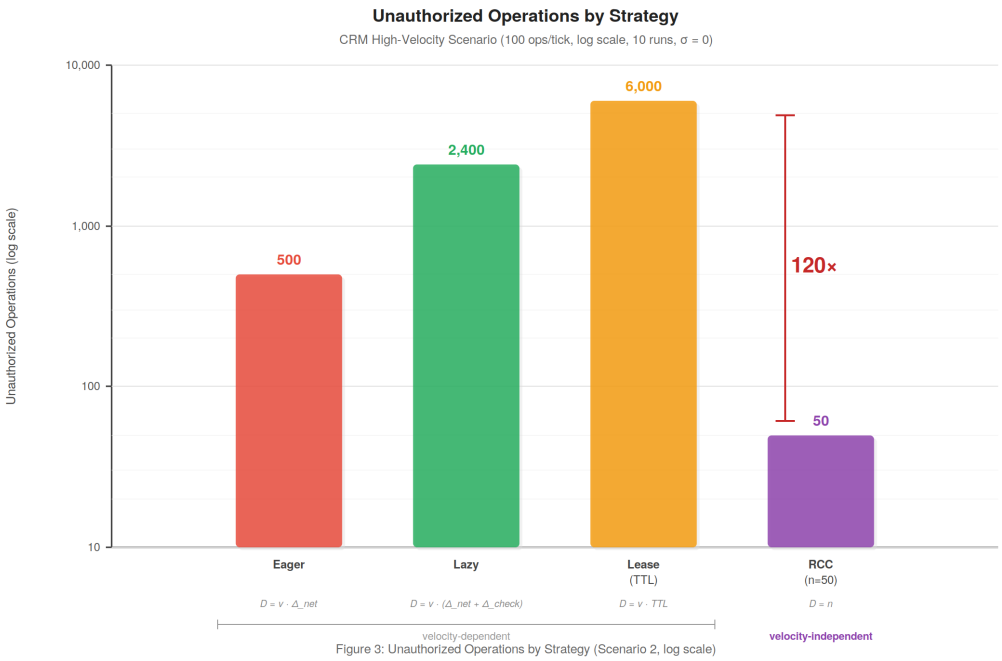

Figure 3: Unauthorized Operations by Strategy (Scenario 2, log scale)

图3：按策略统计的未授权操作（场景 2，对数刻度）。CRM 高速度场景（100 操作/刻，10 次运行，$\sigma = 0$）。Lease 与 RCC 之间的 $120\times$ 差距已标注。纵轴采用对数刻度；每个柱形下方显示损害上界公式。

8.5 场景3——异常自动撤销

智能体行为在时钟周期 50 时分化为突发模式（12 操作/刻）。没有发出明确的撤销指令；信任评分器检测到异常，将信任分数降至 $\tau = 0.4$ 以下，并触发自动撤销。

表7：异常自动撤销结果（均值 $\pm\ \sigma$，10 次运行，随机种子 0–9）。

指标	**急切**	**Lazy**	**租约（3000 个时钟周期）**	**RCC**（$n = 100$）
未授权操作	108.0 ± 0	10.8 ± 3.6	$\mathbf{2,950.8 \pm 3.6}$	$\mathbf{16.0 \pm 3.7}$
最大陈旧度（时步）	10.0 ± 0	1.9 ± 0.3	245.9 ± 0.3	1.8 ± 0.4
级联完整性	1.0	1.0	0.0	1.0

RCC 实现了 $184\times$ 相对于租约生存时间（16.0 ± 3.7 对比$2{,}950.8 \pm 3.6$）的降低，与定理3.1 在动态撤销触发器下的速度无关保证一致。

反直觉的结果：**惰性优于急切**（10.8 ± 3.6对比 108.0 ± 0）。急切模式在撤销传播期间的同步阻塞延迟了撤销信号处理，而智能体仍继续使用其缓存凭证进行操作。信任评分器触发撤销；惰性智能体的下一次检查落在其 10 时步检查间隔内——陈旧度仅为 1.9 ± 0.3 时步。相比之下，急切智能体在跨 5 个代理进行的 10 时步广播期间累积了 108 次未授权操作，而异常窗口期间每时钟周期 12 次操作的突发速率更使情况雪上加霜。

这一结果表明，四种策略并不构成一种简单的线性排序，其顺序也不会在各场景中保持不变。策略有效性是拓扑相关的。短于网络传播延迟的惰性检查间隔在信任触发的撤销下具有优势；同样的检查间隔在高速确定性调度下将是灾难性的（CRM：惰性 $= 2{,}400$ vs. RCC $= 50$）。RCC 结果（16.0 ± 3.7）是可靠后备机制：无论哪种策略在特定拓扑中表现更好，RCC 都提供有界上限。

租约策略的级联完整性为 0.0，这是一个清晰的结果：在 3,000 个时钟周期的生存时间和 300 个时钟周期的仿真下，没有任何租约在模拟窗口内过期。该智能体从不会自行失效。

8.6 成本效益分析

RCC 的理论重新验证开销：

$$\text{Overhead}_{\text{RCC}} = \frac{\Delta_{\text{revalidation}}}{n}.$$

使用 $\Delta_{\text{revalidation}} = 1$ 时钟周期（本地重新认证）和 $n = 50$时：开销为 $= 2.0\%$，即有此比例的操作需要进行重新验证往返。在 $n = 10$时：开销为 $= 10.0\%$。开销与 n成反比，因此形成了可调节的安全性能调节参数。在 $n < 5$时，开销接近急切策略的水平，消除了一致性导向优势。

表 8：策略比较摘要。

策略	损害上界	确定性?	与速度无关?	耦合
急切	$v \cdot \Delta_{\text{net}}$	Yes	No	同步
租约	v 生存时间	Yes	**No**	None
Lazy	$v \cdot (\Delta_{\text{net}} + \Delta_{\text{check}})$	Yes	**No**	按需
RCC	n	**Yes**	**Yes**	获取/释放

RCC 是唯一一种损害上界不受代理速度和网络延迟影响，同时又避免同步耦合的策略。

9 讨论

9.1 范围与适用性

主张3.1 在核心MESI 状态和转换的有界陈旧语义下成立。授权域的扩展——范围衰减、基于信任的撤销、分层委派DAG——严格扩展了硬件模型。硬件界限可作为下界估计，而非精确刻画。

Huang 等人 [9] 的实证支持——分层监督以 5.5% 的开销恢复了 96.4% 的故障错误——为一致性强制执行中固有的开销与安全权衡提供了依据。2–10% 的RCC开销是授权层等效项。这种对应关系具有启发性；我避免过度声称，因为 Huang 等人的场景（语义错误恢复）与 CCS 场景（授权状态一致性）只是类比相关，并非形式等价。

9.2 局限性

权限集中化。 CCS 假设权限一致性严格强于代理一致性——这相当于 NUMA 一致性中的可靠目录控制器假设 [19, 第 6 章，§6.4.3]。发生网络分区时，使用缓存凭证的代理仍受其策略界限约束：无论能否访问授权机构，执行计数代理的操作次数都不会超过剩余操作数。授权机构恢复时遵循目录控制器故障转移机制：从持久化事件日志中重放待处理撤销。脑裂场景留待未来对复制式权限服务的研究。

对抗性调度。 评估采用公平调度下的确定性随机种子。RCC 限制未授权操作的数量，而非其语义影响。策略性对手可以选择执行哪些 n 操作，以造成最大损害——这种能力在硬件中没有对应的类比，但对于依据学习到的目标行动的受感染智能体而言，却完全有可能。执行预算内的对抗性操作调度仍是一个开放问题。

仿真规模。 三个场景，1–10 个智能体，委托深度为 1–3。生产系统还会引入拜占庭故障、网络分区和异构身份提供商。对 $100+$ 个智能体进行压力测试是未来的工作；从侦听到目录的转换启发式需要在该规模下进行经验校准。

时钟刻度抽象。 逻辑时钟刻度用于抽象墙上时钟时间。相对比较（减少 $120\times$、$184\times$ ）不受时钟刻度到时间映射的影响；绝对时间方面的结论则需要针对具体部署进行映射。

形式化验证。 主张3.1 通过穷举转换枚举而非机械化证明得到验证。一个TLA+ 规范可用于通过模型检验验证安全性（获取后不存在未授权操作）和活性（所有智能体最终都达到一致状态）。这是未来工作的主要形式化方向。

上下文污染。 凭证撤销不会撤销知识。读取过 50 条记录的智能体会保留这些上下文。撤销虽有必要，但不足以实现完全遏制。

新兴标准。 与OIDC-A [11], SCIMAgenticSchema [22], 和身份链 [16] 的集成，需要随着这些标准逐渐成熟而进行协议层形式化。

10 结论

多智能体委托链中的授权撤销，在有界陈旧语义下，操作上等同于共享内存多处理器中的缓存一致性。形式化模型（第 3 节）通过明确的状态映射、安全性定理和损害上界分析确立了这一点。基于 生存时间 的方法会产生依赖速度的损害扩展 $O(v \cdot$生存时间$)$。执行次数界限（RCC）则给出独立于速度的确定性上界，即每项能力凭证 n 次操作（定理3.1）——在同步边界强制实现一致性，而非拒绝单次操作。

涵盖三个场景的基于时钟刻度的仿真（共运行 120 次）证实：定义 3.5 得出的预测上界在所有确定性配置中均与观测值完全匹配；RCC 在 $120\times$方面优于生存时间，在 CRM 场景中如此，在异常场景中则为 $184\times$ ；所有 120 次运行均未出现界限违规，证实了定理 3.1 的每项能力凭证适用范围。高速智能体——任何运行速度约高于每秒 10 次操作的自主系统——都不应持有限时凭证。操作次数受限的凭证会在由 n 定义的边界处强制定期重新授权，确保无论智能体运行速度多快，都能在有限次操作内发现撤销。

参考文献

[1] J. Alsop 等人。图形处理器的惰性释放一致性。载于 第 *49* 届 *IEEE/ACM* 国际微体系结构研讨会（*MICRO*）论文集，2016年。

[2] A. Birgisson、J. G. Politz、A. Taly、M. Vaziri 与 M. Liskov。Macaroons：带有上下文限制条件的 Cookie，用于云中的去中心化授权。载于第21届网络与分布式系统安全研讨会（NDSS）论文集，2014年。[3] Biscuit Security。Biscuit：具备去中心化验证的授权令牌。https: //www.biscuitsec.org/，2024年。[4] B. Campbell、J. Bradley、N. Sakimura 与 D. Tonge。OAuth 2.0 令牌交换。RFC 8693，IETF，2020年。[5]M. Cemri、M. Z. Pan、S. Yang、L. A. Agrawal、B. Chopra、R. Tiwari、K. Keutzer、A. Parameswaran、D. Klein、K. Ramchandran、M. Zaharia、J. E. Gonzalez 与 I. Stoica。多智能体大语言模型系统为何会失败？arXiv 预印本 arXiv:2503.13657，2025年。[6] A. Chan、N. Kolt、P. Wills、U. Anwar、C. Schroeder de Witt、N. Rajkumar、L. Hammond、D. Krueger、L. Heim 与 M. Anderljung。人工智能系统的身份标识。arXiv 预印本 arXiv:2406.12137，2024年。[7]云安全联盟。智能体人工智能身份管理方法。https://cloudsecurityalliance.org/blog/2025/03/11/agentic-ai-identity-management-approach，2025年。[8] V. C. Hu、D. Ferraiolo、R. Kuhn、A. Schnitzer、K. Sandlin、R. Miller 与 K. Scarfone。基于属性的访问控制（ABAC）定义与考量指南。特别出版物800-162，美国国家标准与技术研究院，2014年。[9]J. Huang、J. Zhou、T. Jin、X. Zhou、Z. Chen、W. Wang、Y. Yuan、M. R. Lyu 与 M. Sap。存在故障智能体时多智能体协作的韧性。arXiv 预印本 arXiv:2408.00989，2024年。[10] K. Mei、X. Zhu、W. Xu、W. Hua、M. Jin、Z. Li、S. Xu、R. Ye、Y. Ge 与 Y. Zhang。AIOS：大语言模型智能体操作系统。arXiv 预印本 arXiv:2403.16971，2024年。[11] S. Nagabhushanaradhya。面向代理的 OpenID Connect（OIDC-A）1.0：面向基于大语言模型的智能体身份与授权的标准扩展。arXiv 预印本 arXiv:2509.25974，2025年。[12]OpenID 基金会。智能体人工智能的身份管理。https://openid.net/wp-content/uploads/2025/10/Identity-Management-for-Agentic-AI.pdf，2025年。[13] Oso。为人工智能智能体设置权限。https://www.osohq.com/learn/ai-agent-permissions-delegated-access，2025年。[14] S. Raza、R. Sapkota、M. Karkee 与 C. Emmanouilidis。智能体人工智能治理的 TRiSM：基于大语言模型的智能体多智能体系统中信任、风险与安全管理综述。arXiv 预印本 arXiv:2506.04133，2025年。[15] J. Reason。人为错误：模型与管理。BMJ，320(7237):768–770，2000年。[16] A. Schwenkschuster、P. Kasselman、K. Burgin、M. J. Jenkins 与 B. Campbell。跨域的身份与授权链式委托。IETF互联网草案，https://datatracker. ietf.org/doc/draft-ietf-oauth-identity-chaining/，2024年。[17] C. E. 香农。通信的数学理论。《贝尔系统技术期刊》，27(3):379–423，1948年。[18] I. Singh 等人。GPU 架构的缓存一致性。IEEE Micro，33(4)，2013年。

[19] D. J. Sorin、M. D. Hill 和 D. A. Wood。《内存一致性与缓存一致性入门》。《计算机体系结构综合讲座》。Morgan & Claypool，第 2 版，2020年。[20]T. South、S. Marro、T. Hardjono、R. Mahari、C. Deslandes Whitney、D. Greenwood、A. Chan 和 A. Pentland。经身份验证的委托与获授权的人工智能代理。arXiv 预印本 arXiv:2501.09674，2025年。[21] G. Vijayaraghavan 等人。如果你想要一致性，就组织一个对手团队：组织智能的多智能体模型。arXiv 预印本 arXiv:2601.14351，2026年。[22] M. Wahl。跨域身份管理系统：智能体身份架构。IETF互联网草案，https://datatracker.ietf.org/doc/draft-wahl-scim-agent-schema/，2024年。[23] J. Wei 等人。思维链提示能够在大语言模型中激发推理能力。载于神经信息处理系统进展（*NeurIPS*），2022年。[24] Q. Wu 等人。AutoGen：通过多智能体对话实现下一代大语言模型应用。arXiv 预印本 arXiv:2308.08155，2023年。[25] S. Yao 等人。ReAct：协同语言模型中的推理与行动。载于国际学习表征会议（*ICLR*），2023年。

算法 1 仿真引擎

要求： 场景配置 S，策略 π，最大 _时步 T
确保： 指标 M

从 S 初始化 AuthorityService 和 AgentSet
初始化 MetricsCollector
对于 $t = 1$ **到** T **执行**
    对于所有 智能体 $a \in$AgentSet **执行** ▷阶段 1：智能体操作
        尝试次数 $\leftarrow$ 确定_操作_计数(a， S， t)
        对于所有尝试 **执行**
            上限 $\leftarrow$随机_能力(a)
            结果 $\leftarrow$ a.尝试_操作(能力)
            如果 能力。实际上_已撤销_于_授权机构()**那么**
                M.未授权_计数 $+= 1$
            结束如果
            如果 $\pi =$ RCC **且** a.操作_剩余 $= 0$ **那么**
                a.请求_重新验证() ▷ 获取
            结束条件判断
        结束循环
    结束循环
    对于所有待处理撤销 r **执行** ▷第二阶段：授权机构处理
        如果 $\pi =$急切 **则**
            广播_并_等待_确认(r)
        否则如果 $\pi =$惰性 **则**
            标记_已撤销(r)
        否则如果 $\pi =$租约 **则**
            标记_已撤销(r)
        否则如果 $\pi =$RCC **则**
            标记_已撤销(r)
        结束如果
    结束循环
    传递_消息(延迟 $= S$.延迟_时步) ▷ 阶段 3：网络传递
    对于所有智能体 a**执行** ▷阶段 4：异常检测
        如果 信任_scorer.check_异常(a)**则**
            撤销_所有_能力(a)
        结束如果
    结束循环
    对于所有智能体 a具有瞬态状态 **执行** ▷ 阶段 5：解决瞬态
        如果 确认_已收到(a)**则**
            a.状态 $\leftarrow$稳定_目标(a.瞬态_状态)
        结束如果
    结束循环
    M.记录_时钟周期(t， AgentSet) ▷ 阶段 6：指标
结束循环
返回 M.聚合()