Accountable Delegation: Why Organizations Need Agent-Aware Endpoint Control for AI Agents

Ali Mohammadi Ruzbahani

PubArx:2026.00017v1 [cs.AI] 26 Nov 2026 · Preprint

***Abstract*—Organizations increasingly delegate work to AI agents that run on employees' computers, where they read and change files, execute programs and contact external services, while employees paste business and personal data into AI chat services. Operating systems, identity systems and network controls authorize all of this as the employee, because they know no principal smaller than the user. Data protection law, the EU AI Act, sector regulation and the NIST frameworks nonetheless hold the organization accountable for what its AI does, and they call for least privilege, human oversight, the ability to intervene, and records that withstand scrutiny. From these sources we derive six requirements for governing agents on endpoints. We prove that no control deciding by account, role, device or program can enforce a policy that treats an agent differently from its user, whereas attribution along the chain of processes suffices; that the workload of human oversight is bounded by the number of distinct situations, with the probability that the next request needs a person below $K/(en)$ after n requests; and that rarity of unsafe behaviour offers no protection at organizational scale. After assessing the established control families, we present IOArx, which names the agent behind each file operation, decides it by a policy written for agents, holds the agent while a person decides, gates network connections, inspects submissions to AI services and keeps a hash-chained record, all on the device. In measurements, IOArx attributed processes to their agents without a misattribution, held a violating agent on Linux within a median of 3.1 ms, and raised no false alarm on 32,771 real prompts.**

***Index Terms*—AI agents, AI governance, accountability, human oversight, access control, data protection, regulatory compliance, endpoint security**

I. Introduction

ORGANIZATIONS increasingly delegate work to AI agents that run on their employees' own computers. A coding agent reads a repository, edits files, installs packages, runs commands and opens network connections; a desktop assistant searches documents and drafts messages; and in the browser, employees paste contracts, logs and customer records into AI chat services. All of it happens under the employee's account. The operating system, the identity provider and the network see the employee, so every agent holds every right the employee holds, and the controls the organization operates were not built to tell what an agent did from what the person did. The law, meanwhile, treats that difference as the organization's business. Data protection law makes the controller responsible for the security of personal data and requires it to demonstrate compliance; the EU AI Act obliges deployers of high-risk systems to assign human oversight to competent people and to keep the logs those systems generate; and the NIST AI Risk Management Framework expects mechanisms that can supersede, disengage or deactivate an AI system whose outcomes depart from its intended use [1]. An organization can write a policy that forbids agents to read credentials or to send customer data to outside services, but on the endpoint it cannot enforce that policy, cannot have a person decide when it matters, and cannot show afterwards that the policy held.

What is missing is not another rule but a principal. Access control on the endpoint has no notion of an agent, so a policy about agents has nothing to attach to. IOArx supplies that principal [2]. It names the agent behind every file operation by following the chain of processes back to a recognized agent program, so that a shell or compiler started by an agent is known as the agent's. It decides each operation by a policy written for agents, lets a person decide while the agent waits, pauses or stops the agent together with every process it started, gates network connections before data leaves, inspects what employees submit to AI services in the browser, and appends every decision to a hash-chained record. All of it runs on the device. Each function answers an obligation the organization already carries, and together they turn an AI-use policy on paper into one that is enforced and evidenced where agents act.

This paper develops the case in the manner of a governance analysis, supported by formal results and measurements. Its contributions are the following.

- A synthesis of provisions of European, Canadian and United States law and of NIST frameworks into six requirements for governing AI agents on endpoints, with a proportionality condition on the control itself (Section III).
- A formal account showing that controls deciding by account, role, device or program cannot enforce any agent-specific policy, that attribution along the process chain suffices, that the workload of human oversight is bounded by the number of distinct situations, and that rarity of unsafe behaviour gives no protection at organizational scale (Section IV).
- An assessment of the established control families against these requirements (Section V).
- An account of IOArx as an organizational control, with measured evidence and an operating model, followed by recommendations for organizations, agent vendors and policy makers (Sections VI and VII).

II. AI Agents in the Organization

Generative AI has entered ordinary work because it pays. In data from 5,172 customer-support agents of a firm that

A. Mohammadi Ruzbahani is Director of the IOArx Development Team. Calgary, Alberta, Canada (e-mail: info@ioarx.com).

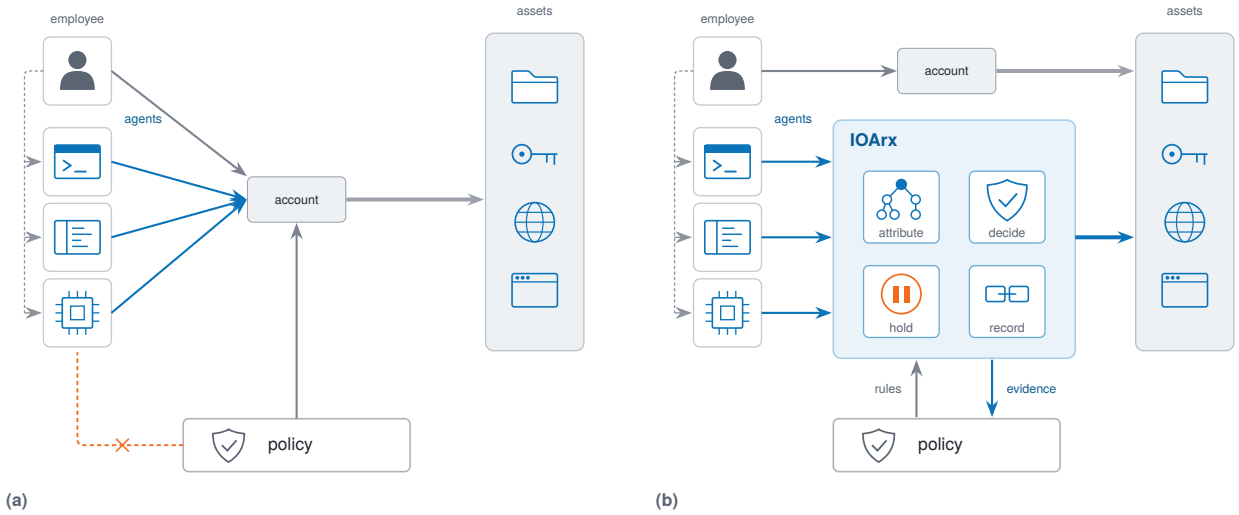

Fig. 1. The delegation gap. (a) Policy reaches the employee's account, through which every agent acts. (b) IOArx makes each agent a principal and returns evidence to governance.

introduced an AI assistant in stages, access to the assistant raised the number of issues resolved per hour by 15% on average and by 30% among less skilled and less experienced workers [3]. Systems built on the same models increasingly act instead of answering. Chan et al. trace this shift to four characteristics that raise the agency of an algorithmic system, namely underspecification, directness of impact, goal-directedness and long-term planning, and they insist that recognizing such agency does not shift responsibility for the resulting harms away from the people and organizations behind the system [4]. On an employee's computer the shift is tangible: coding agents, AI code editors, desktop assistants and local model runtimes read and write files, start programs and contact services as part of their ordinary work.

Every such agent is a program the employee started, and the operating system authorizes programs by the account that runs them. The agent thus receives the whole of the employee's authority, from SSH keys and cloud credentials to customer files, mail archives and the network. The principle of least privilege, that every program should operate using the least set of privileges necessary to complete its job [5], fails here by construction, because the operating system has no smaller principal to give the agent (Fig. 1a).

Two kinds of failure turn this breadth into exposure. One is adversarial. Indirect prompt injection places instructions in content that an agent retrieves, so that an attacker without any access to the organization can steer the agent through a web page, an issue or a shared document [6]. On the InjecAgent benchmark, a ReAct-prompted GPT-4 agent followed injected instructions in 24% of the cases and in 47% once the attack was reinforced [7], and AgentDojo turns the question into 97 realistic tasks and 629 security test cases on which attacks and defenses can be compared [8]. The other kind needs no adversary: in an emulated sandbox of high-stakes tool use, even the safest agent evaluated failed in risky ways 23.9% of the time [9]. Neither failure announces itself to the employee who started the agent, since both arise inside the agent while it works.

People carry the same exposure into the browser. Users of chat assistants disclose personal information about themselves and about others to obtain the help they want [10], and personal data in a model's training text can later be extracted from the model [11]. Prohibition does not settle the matter. Employees use hardware and software that never received formal approval from their IT department, a practice known as shadow IT [12], and they rationalize violations of security policy through neutralization techniques, which Siponen and Vance found to be an important factor in such violations when tested alongside the threat of sanctions [13]. An AI-use policy that exists only as a document leaves the decision to the person at the keyboard and, once an agent acts, to the agent.

Proposals for governing agents emphasize visibility. Chan et al. propose agent identifiers, real-time monitoring and activity logs as measures that make the actions of AI agents visible and thereby support accountability, and they note that visibility into business uses of agents may matter more than visibility into personal uses [14]. Each of these measures needs a party that can observe the agent. For agents that run on employees' computers that party can only be the organization, and it can observe only through a control on the endpoint.

III. Governance Obligations

Accountability, in Bovens's widely used formulation, is a relationship between an actor and a forum in which the actor has an obligation to explain and to justify its conduct, the forum can pose questions and pass judgement, and the actor may face consequences [15]. Locating the actor is already difficult when many people contribute to an outcome, which Bovens calls the problem of many hands; an agent acting under an employee's account adds a hand that the organization cannot see. Organizational AI governance builds on that relationship as a system of rules, practices, processes and technological tools that keeps an organization's use of AI aligned with its strategy and values, with legal requirements and with the

ethical principles it follows [16]. Both notions presuppose that the organization can establish what its AI did. Table I collects the provisions of law and of widely adopted frameworks that bear on agents running on employees' computers. They differ in scope and in legal force, yet they converge on a small set of capabilities.

The AI Act addresses the organization as a deployer, the natural or legal person using an AI system under its authority (Art. 3(4)). Every deployer must take measures to support the AI literacy of the staff who operate AI systems on its behalf (Art. 4, as amended by Regulation (EU) 2026/1744). For high-risk systems, the Act requires that natural persons can oversee the system effectively, including by intervening in its operation or interrupting it through a stop button or a similar procedure that brings it to a halt in a safe state (Art. 14(1), 14(4)(e)), and it obliges deployers to assign that oversight to people with the necessary competence, training and authority, to monitor operation and to keep the automatically generated logs for at least six months (Art. 26(2), (5), (6)). These obligations bind high-risk systems only and apply to them from December 2027 or August 2028, depending on how the system is classified (Art. 113 as amended). Their substance is nonetheless the substance of oversight anywhere. Analysing the drafts of the Act, Enqvist showed that its oversight provisions addressed the providers of systems and left open who performs oversight at the user's end and with what powers [19]; the final text answers part of this through Article 26(2). Reviewing 41 policies that require human oversight of government algorithms, Green finds that people cannot reliably perform the oversight functions these policies expect, so that such policies can legitimize flawed algorithms instead of correcting them [20]. Sterz et al. name the conditions under which a person exercises oversight effectively: causal power over the system and its effects, epistemic access to the relevant aspects of the situation, self-control, and intentions that fit the role [21].

Data protection law applies whatever the risk class of the AI. The GDPR requires that personal data be limited to what is necessary and protected against unauthorised processing (Art. 5(1)(c), (f)), that the controller be responsible for and able to demonstrate compliance (Art. 5(2), 24(1)), that protection be built into processing by design (Art. 25(1)), and that processing systems remain confidential and intact (Art. 32(1)(b)). A breach of security that leads to the unauthorised disclosure of, or access to, personal data is a personal data breach (Art. 4(12)), whether a person or an agent brought it about. Unless it is unlikely to result in a risk to people, the controller must notify it without undue delay and, where feasible, within 72 hours of becoming aware of it, describing where possible the categories and approximate number of data subjects and records concerned, and it must document every breach (Art. 33(1), (3)(a), (5)). Fines reach EUR 10 million or 2% of worldwide annual turnover for infringements of Articles 25 to 39, and EUR 20 million or 4% for infringements of the basic principles (Art. 83(4), (5)). Canada's PIPEDA makes an organization responsible for the personal information under its control, requires safeguards appropriate to its sensitivity, and requires breaches that create a real risk of significant harm to be reported and every breach to be recorded (Sch. 1, cl. 4.1, 4.7; ss. 10.1(1), 10.3(1)). In the United States, the HIPAA Security Rule requires covered entities and their business associates to allow access to electronic health information only to authorized persons and programs and to record and examine activity in the systems that hold it (45 CFR 164.312(a)(1), (b)).

Cybersecurity regulation adds duties of management. NIS2 requires essential and important entities to take measures that include access control policies and incident handling, makes their management bodies approve and oversee these measures, and sets deadlines of 24 hours for an early warning and 72 hours for a notification of a significant incident (Art. 20(1), 21(2), 23(4)). DORA requires financial entities to limit logical access to information assets to what legitimate and approved functions require (Art. 9(4)(c)). Voluntary frameworks, which shape audits and procurement, point the same way. The NIST AI Risk Management Framework asks that processes for human oversight be defined, assessed and documented, and that mechanisms exist to supersede, disengage or deactivate AI systems [1]; its profile for generative AI counts information security, data privacy, human-AI configuration, and value chain and component integration among the risks to manage [22]. SP 800-53 already extends least privilege to processes acting on behalf of users and asks for irrefutable evidence that an individual, or a process acting on behalf of an individual, performed an action (AC-6, AU-10) [17]; an agent and every process it starts are precisely such processes. The zero trust architecture of SP 800-207, finally, grants access to each resource per session, by dynamic policy over the observable state of client identity, application and requesting asset [18].

Read together, these provisions ask six things of an organization whose agents work on its endpoints. *R1, attribution:* each action can be traced to the agent that performed it and to the person it acted for. *R2, least privilege for agents:* the organization can restrict what agents may access independently of what their users may access. *R3, oversight at the point of action:* a competent person can approve or refuse an agent's action while the agent waits, and the absence of an answer grants nothing. *R4, intervention:* an agent that acts against the policy or against a person's decision can be halted together with everything it started. *R5, outbound control:* data flows from agents to the network, and from people to AI services, are governed before data leaves. *R6, evidence:* decisions are recorded with the agent's identity, in a form whose alteration can be detected, for as long as audits, investigations and breach notification require. A seventh condition constrains the control itself, which is processing of personal data in its own right and a form of monitoring at work. Workplace surveillance affects employees' well-being, work culture and motivation, and whether it harms or benefits them depends on whether it is conducted in a balanced way [23]. *P, proportionality:* the control observes agents rather than people, processes what it observes on the device, and records only what the requirements need.

IV. A Formal Account of Agent Governance

The requirements say what an organization must be able to do. This section asks what a control must observe to do it,

TABLE I
PROVISIONS BEARING ON AI AGENTS THAT ACT ON EMPLOYEES' COMPUTERS

Source	Provision	What it asks of the organization	Req.
EU AI Act, Regulation (EU) 2024/1689, as amended by Regulation (EU) 2026/1744	Art. 4	Measures that support the AI literacy of staff who operate AI systems on its behalf	R3
	Art. 14(1), 14(4)(e)	High-risk systems can be overseen by natural persons, who can intervene or interrupt them so that they halt in a safe state	R3, R4
	Art. 26(2), (5), (6)	Deployers of high-risk systems assign oversight to competent people, monitor operation and keep the logs for at least six months	R1, R3, R6
GDPR, Regulation (EU) 2016/679	Art. 5(1)(c), (f), 5(2), 24(1)	Data minimisation; protection against unauthorised processing; responsibility for, and demonstration of, compliance	R1, R2, R5, R6
	Art. 25(1), 32(1)(b)	Data protection by design; ongoing confidentiality and integrity of processing systems	R2, R5
	Art. 33(1), (3)(a), (5)	Breach notified, where feasible within 72 hours, with the categories and number of records concerned; every breach documented	R1, R6
NIS2, Directive (EU) 2022/2555	Art. 20(1), 21(2)(a), (b), (i), 23(4)	Management approves and oversees measures that include access control policies and incident handling; early warning of a significant incident within 24 hours, notification within 72 hours	R2, R6
DORA, Regulation (EU) 2022/2554	Art. 9(4)(c)	Logical access limited to what legitimate and approved functions require	R2
PIPEDA, S.C. 2000, c. 5	Sch. 1, cl. 4.1, 4.7; ss. 10.1(1), 10.3(1)	Accountability for personal information under its control; safeguards matched to sensitivity; breach reports and records	R1, R2, R6
HIPAA Security Rule, 45 CFR 164.312	(a)(1), (b)	Access only for authorized persons and programs; mechanisms that record and examine activity	R1, R2, R6
NIST AI RMF 1.0 [1]	MAP 3.5, MANAGE 2.4	Defined processes for human oversight; mechanisms to supersede, disengage or deactivate AI systems	R3, R4
NIST SP 800-53 Rev. 5 [17]	AC-4, AC-6, AU-9, AU-10	Information flow enforcement; least privilege for users and processes acting on their behalf; protection of audit information; non-repudiation	R1, R2, R5, R6
NIST SP 800-207 [18]	Tenets 3, 4	Access to each resource granted per session by dynamic policy over the observable state of client identity, application and requesting asset	R1, R2

and what human oversight and enforcement cost at the scale of an organization. The model refers to no product; Section VI returns to IOArx.

Let H be the employees of an organization and A the agents they run, each agent being a process that runs the program of an AI agent and acting for the employee who started it, $\delta\colon A \to H$; we set $\delta(h) = h$ for $h \in H$. A process p runs a program $\mathrm{img}(p)$ under an account $\mathrm{acct}(p) \in H$, in a session with attributes $\kappa(p)$ such as the device, its network location and the time. Its principal $\pi(p) \in H \cup A$ is the agent nearest to p among p and its ancestors, the processes that led to it, or $\mathrm{acct}(p)$ if there is none. Agents, and the processes they start, run under the account of the employee who started the agent, so $\mathrm{acct}(p) = \delta(\pi(p))$ for every process. A request $r = (p, o, \omega)$ asks for an operation $\omega \in \Omega$ on an object $o \in O$, which may be a file, a network destination or an AI service.

Definition 1 (Governance policy)**.** *A governance policy is a map* $G\colon (H \cup A) \times O \times \Omega \to \{\mathsf{Allow}, \mathsf{Ask}, \mathsf{Block}\}$*, and the decision it intends for* $r = (p, o, \omega)$ *is* $G(\pi(p), o, \omega)$*. The policy is agent-sensitive if* $G(a, o, \omega) \neq G(\delta(a), o, \omega)$ *for some* $a \in A$*,* $o \in O$ *and* $\omega \in \Omega$*.*

A policy that lets employees open their SSH keys but forbids agents to do so is agent-sensitive, and so is one that asks a person before an agent uploads a file the employee may upload freely. Every agent-specific clause of an AI-use policy has this form.

Definition 2 (Control)**.** *A control observes each request through a view* $\theta(r)$ *and decides it by a rule* f*. It enforces* G *on a set of requests* R *if* $f(\theta(r)) = G(\pi(p), o, \omega)$ *for every* $r \in R$*. The view is lineage-blind if* $\theta(p, o, \omega) = \theta(p', o, \omega)$ *whenever* $\mathrm{acct}(p) = \mathrm{acct}(p')$*,* $\mathrm{img}(p) = \mathrm{img}(p')$ *and* $\kappa(p) = \kappa(p')$*.*

File system permissions, identity and access management, role- and attribute-based access control over employee attributes, and network controls keyed to users, devices or applications all decide through lineage-blind views. Agents, for their part, do their work through the general-purpose programs their users also run: shells, interpreters, package managers and version control.

Assumption 1 (Shared programs)**.** *For every agent* a *and every* (o, ω) *that* a *can request, there are processes* p_a *and* p_h *with* $\pi(p_a) = a$*,* $\pi(p_h) = \delta(a)$*,* $\mathrm{img}(p_a) = \mathrm{img}(p_h)$ *and* $\kappa(p_a) = \kappa(p_h)$*.*

Theorem 1 (Lineage-blind controls)**.** *Under Assumption 1, let* G *be agent-sensitive with witness* (a, o, ω)*. Every control with a lineage-blind view decides* (p_a, o, ω) *and* (p_h, o, ω) *identically and therefore errs on at least one of them. In particular, it refuses the agent's request only if it also refuses the identical request of the employee.*

Proof. We have $\mathrm{acct}(p_a) = \delta(a)$ and $\mathrm{acct}(p_h) = \delta(\delta(a)) = \delta(a)$. With equal programs and sessions, lineage-blindness gives $\theta(p_a, o, \omega) = \theta(p_h, o, \omega)$, so f returns one decision d for both requests. The intended decisions $G(a, o, \omega)$ and $G(\delta(a), o, \omega)$ differ, so d differs from at least one of them, and $d = \mathsf{Block}$ applies to both requests alike. □

Figure 2 shows the two requests of the proof. A lineage-blind control has two choices for each such pair: let the agent

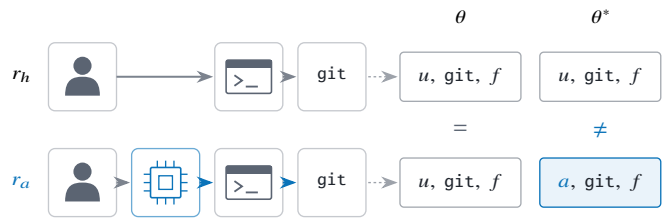

Fig. 2. The employee's request r_h and the agent's request r_a under a lineage-blind view θ and a lineage view θ^*.

act with the employee's full authority, or take that authority from the employee as well. Two consequences follow at once.

Corollary 1 (Records)**.** *A record whose entries are computed from a lineage-blind view assigns the same entry to* (p_a, o, ω) *and* (p_h, o, ω)*. It cannot establish whether an agent performed an action, let alone which agent.*

Corollary 2 (Price of interception)**.** *For any distribution over the pairs of Assumption 1, a lineage-blind control blocks the agent's request with exactly the probability with which it blocks the employee's.*

The positive counterpart needs only the chain of processes and evidence that ties a program to an agent.

Proposition 1 (Lineage suffices)**.** *Let a view* θ^* *contain, for each request, the chain of processes from* p *back to the start of its session and, for each process* q *in it, evidence* e_q *about its program, and let* ρ *map evidence to the agent program it recognizes, or to* $\perp$*. Set* $\hat{\pi}(p) = q$ *for the nearest* q *in the chain with* $\rho(e_q) \neq \perp$*, and* $\hat{\pi}(p) = \mathrm{acct}(p)$ *if there is none. If the chain is complete and* $\rho(e_q) \neq \perp$ *exactly when* q *runs the program of an AI agent, then* $\hat{\pi} = \pi$*, and the rule* $f^*(\theta^*(r)) = G(\hat{\pi}(p), o, \omega)$ *enforces every policy* G*. If* $\hat{\pi}(p) \neq \pi(p)$ *on a fraction* ε *of requests,* f^* *errs on at most that fraction.*

Proof. Under the two conditions, the nearest process in the chain whose evidence ρ recognizes is the nearest agent among p and its ancestors, so $\hat{\pi}(p) = \pi(p)$; if there is none, both equal $\mathrm{acct}(p)$. The rule is well defined because $\hat{\pi}(p)$ is computed from $\theta^*(r)$ alone. Then $f^*(\theta^*(r)) = G(\pi(p), o, \omega)$ for every request. In general f^* is correct wherever $\hat{\pi}(p) = \pi(p)$, which confines its errors to the fraction ε. □

The exactness of ρ is where the strength of evidence matters. A program's name or folder can be imitated by any program, whereas a verified code signature or an installed application package is vouched for by the operating system.

Human oversight comes next, and its cost decides whether it is exercised in substance or waved through [24], [20]. Suppose that the requests needing a person fall into K situations, each the class of requests that one recorded answer decides, such as one agent writing to one folder.

Theorem 2 (Workload of oversight)**.** *Let requests arrive one at a time and independently, each in situation* k *with probability* w_k*,* $\sum_k w_k = 1$*. Let a person be asked at the first request of each situation and record the answer before the next request arrives, and let the recorded answer decide every later request of that situation. Let* P_n *be the number of times the person is asked among the first* n *requests and* q_{n+1} *the probability that request* $n+1$ *needs the person. Then*

$$\mathbb{E}[P_n] = \sum_{k=1}^{K} \big(1 - (1 - w_k)^n\big) \leq \min(n, K), \tag{1}$$

$$q_{n+1} = \sum_{k=1}^{K} w_k (1 - w_k)^n \leq K \, \frac{n^n}{(n+1)^{n+1}} < \frac{K}{e\, n} \quad (n \geq 1), \tag{2}$$

and the middle bound in (2) *is attained when* $K = n + 1$ *and* $w_k = 1/K$ *for every* k*.*

Proof. Request m needs the person exactly when its situation did not occur among the first $m-1$ requests, so $q_m = \sum_k w_k (1-w_k)^{m-1}$ and $\mathbb{E}[P_n] = \sum_{m=1}^{n} q_m = \sum_k w_k \sum_{j=0}^{n-1} (1-w_k)^j = \sum_k \big(1 - (1-w_k)^n\big)$. Each term is at most 1, and by Bernoulli's inequality at most nw_k; summing gives $\min(n, K)$. For (2), the function $g(w) = w(1-w)^n$ has derivative $(1-w)^{n-1}\big(1-(n+1)w\big)$ on $[0,1]$ and therefore its maximum $n^n/(n+1)^{n+1}$ at $w = 1/(n+1)$; summing over k gives the middle bound. Since $\ln(1+x) > x/(1+x)$ for $x > 0$, we have $(n+1)\ln(1+1/n) > 1$, that is $(1+1/n)^{n+1} > e$, and hence $n^n/(n+1)^{n+1} = 1/\big(n(1+1/n)^{n+1}\big) < 1/(en)$. For $K = n+1$ and $w_k = 1/K$, both sides of the middle bound equal $\big(n/(n+1)\big)^n$. □

The bound (1) restates a result on remembered answers in [2], while (2) and its attainment are new. Together they bound the workload of oversight by the number of distinct situations, however much the agents do, and make the probability that the next request needs a person fall at least as fast as $K/(en)$, whatever the frequencies of the situations (Fig. 3b, c). Since the bound is attained, no assumption about frequencies beyond their number can improve it. Rare situations still reach a person when they first arise, which is where judgement is needed.

The last question is how much protection the rarity of unsafe behaviour provides. Let the agents of an organization carry out N tasks in a period, each of which independently attempts an action the policy forbids with probability p, and let a control stop each such attempt before it takes effect with probability c, independently.

Proposition 2 (Scale)**.** *Let* U *be the number of forbidden actions that take effect and* $x = (1-c)p$*. Then* $\mathbb{E}[U] = xN$ *and*

$$1 - e^{-xN} \leq \Pr[U \geq 1] = 1 - (1-x)^N \leq xN. \tag{3}$$

Consequently, $\Pr[U \geq 1] \leq \beta$ *requires* $x \leq \ln\big(1/(1-\beta)\big)/N$ *and holds whenever* $x \leq \beta/N$*.*

Proof. Each task yields a forbidden action that takes effect with probability $p(1-c) = x$, independently of the other tasks, so U is binomial with parameters N and x. Then $\Pr[U = 0] = (1-x)^N$, which is at most e^{-xN} because $1 - x \leq e^{-x}$, and at least $1 - xN$ by Bernoulli's inequality. Solving $1 - e^{-xN} \leq \beta$ and $xN \leq \beta$ for x gives the two conditions. □

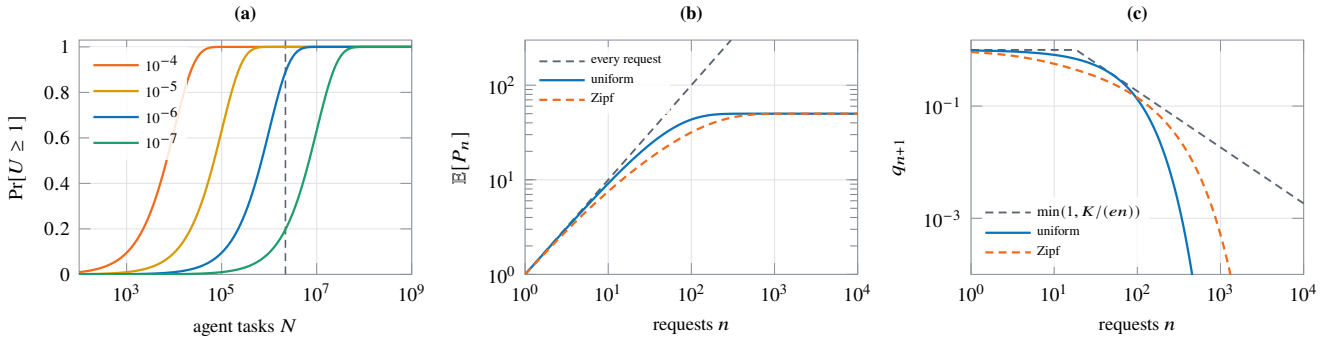

Fig. 3. (a) Probability that a forbidden action takes effect within N agent tasks, for residual rates $x = (1-c)p$ per task; the dashed line marks a year of ten tasks a day by 1,000 employees. (b) Expected questions to a person after n requests in $K = 50$ situations. (c) Probability that request $n+1$ needs a person.

TABLE II
CONTROL FAMILIES AGAINST THE REQUIREMENTS

Control family	R1	R2	R3	R4	R5	R6
Identity, roles and permissions	○	○	○	○	○	◐
Endpoint data loss prevention	○	○	◐	○	◐	◐
Endpoint detection and response	◐	○	○	◐	○	◐
Network security	○	○	○	○	◐	◐
Agents' own safeguards	◐	◐	◐	◐	◐	◐
Isolation of the agent	●	●	○	●	◐	◐
IOArx	●	●	●	●	◐	●

● met by design, ◐ met in part, ○ not met

Consider 1,000 employees who each delegate ten tasks a working day over a year of 220 working days, so that $N = 2.2\times 10^6$. Were unsafe behaviour in practice more than a hundred thousand times rarer than in the benchmarks of Section II, $p = 10^{-6}$, the probability that at least one forbidden action takes effect within the year without enforcement would be 0.89 (Fig. 3a). Keeping it below 5% requires $x \le 2.3 \times 10^{-8}$. The organization controls neither p, which belongs to the agent's vendor and to whoever writes the content the agent reads, nor N, which measures the value it draws from its agents. It controls c. A control must therefore stop practically every attempt within the scope of the policy, and by Corollary 2 a lineage-blind control can do so only by refusing the employees the same actions.

V. Established Controls Against the Requirements

Organizations already operate controls that touch each requirement. The question is which of them can reach an agent as a principal. Table II summarizes the assessment, which rests on what each family observes and where it acts in its standard form, not on any particular product.

Identity and access management, with role- and attribute-based access control, decides by the identity and attributes of the user, the device and the session, and file system permissions decide by account and group. Where an agent calls a service with credentials of its own, these systems can tell it apart. On the employee's computer, however, the agent inherits the session, the files and the stored credentials of the account; the view is lineage-blind, so by Theorem 1 no agent-sensitive policy is enforceable there, and by Corollary 1 the records name the employee.

Endpoint data loss prevention inspects content at egress channels such as removable media, mail, uploads and the clipboard, and decides by content, channel, user and application. It covers part of R5 and can ask a person to justify a transfer, which covers part of R3. Its policies are keyed to applications rather than to the chain of processes, so an agent that uploads through a shell utility or the browser is, to the control, the employee using that utility. Endpoint detection and response records process trees, which carries the lineage that R1 needs, and it can terminate processes. Its decisions, however, look for malicious behaviour, whereas an agent is legitimate software doing what it was asked; it offers no authorization policy over what agents may access and no decision by a person before an action takes effect. Network controls, from firewalls to secure web gateways and cloud access security brokers, decide by destination, user, device and sometimes application. They stop traffic before it leaves, but they see the employee's computer as one source and cannot hold the process that produced the traffic.

Agents' own safeguards, such as confirmation before a command runs, permission settings and sandboxes, address every requirement in part. They operate inside the software whose behaviour they are meant to constrain, which a successful injection steers [6]; they differ from vendor to vendor, so no organization-wide policy reaches them all; and their transcripts stay with the agent. The principle of complete mediation asks that every access to every object be checked for authority [5], and a check inside the agent sees only the actions the agent routes through it. Isolation in a virtual machine or a container, finally, gives the agent its own principal by construction and meets R1, R2 and R4. It does so by separating the agent from the employee's working files and credentials, which are the reason the agent was started, and it provides no decision by a person at the moment of access; organizations therefore apply it to selected workloads rather than to the assistants employees use every day.

VI. IOArx as an Organizational Control

IOArx consists of a desktop application for Windows and Linux, built on one decision core, and IOArx for Browsers, an extension for Chrome, Edge and Firefox. It follows the design that Section IV prescribes: the agent is the principal of every

decision, the view contains the chain of processes, a person decides while the agent is held, and the record names the agent. Its architecture, its policy semantics and their proofs, and its evaluation are given in [2]. This section maps its functions to the requirements, reports the evidence that matters to an organization and describes how an organization operates it.

A. From Requirements to Mechanisms

Attribution (R1). IOArx recognizes 28 agents on Windows, from command-line coding agents and AI code editors to desktop assistants and local model runtimes, and an organization can add its own. Recognition uses five kinds of evidence. A code signature of a named publisher, optionally of a named product, and the identity of an installed application package are verified by the operating system; the program's file name, part of its installation path and text in its command line are not. Each process is resolved along its chain of ancestors and named by the nearest recognized program, as in Proposition 1, and descendants are followed from their first event, so that a compiler or shell started by an agent is attributed to it. Every record states which evidence matched and whether it was verified.

Least privilege for agents (R2). A rule names a file or folder, the operations it covers among reading, writing, deleting and renaming, a subject and a decision. The subject is any program, any agent, one named agent, or one ordinary program identified by its executable, and a rule for an ordinary program never applies to an agent. Among the rules that apply, the longest path prevails, then the most specific subject, then the strictest decision, so the outcome never depends on the order in which rules were written. On first run, IOArx places Documents, Desktop, Pictures and Downloads under an Ask rule for every agent. It also recognizes credential stores from their paths in eight categories, among them SSH and other private keys, cloud credentials, environment files, password databases, browser login and cookie files and access tokens, and it reports every agent access to them that was allowed as a critical leak, which shows the policy owner where a rule is missing.

Oversight at the point of action (R3). An Ask decision suspends the agent together with every process it started and shows the person the agent, whether its identity was verified, the file and the operation, with a countdown. Allow resumes the processes and Block stops them. If no one answers before the deadline, 30 seconds by default and adjustable from 15 seconds to 5 minutes, the request is refused and the processes are stopped, so the absence of a person never grants access, as the principle of fail-safe defaults asks [5]. The person may answer once or record the answer as a rule for the agent, the operation and the file or its folder, which are the situations of Theorem 2. In the terms of Sterz et al., the person holds causal power, because the agent cannot continue before the answer, and epistemic access, because the alert names the agent and what it asked for [21].

Intervention (R4). A Block decision stops the agent and every process it started at once, and the application names the agent, the operation and the file. Processes of the operating system that must keep running are never paused or stopped. Once the agent is held, it performs no further operation. If its operations are at least $1/r$ apart and the hold takes effect a time L after the violating operation, at most $\lfloor rL \rfloor$ further operations complete in between [2].

Outbound control (R5). On the endpoint, IOArx gates the network at the first packet of each connection: programs on a block list cannot connect, and in ask mode only the programs a person has allowed can. The gate identifies programs by their executable file. In the browser, IOArx for Browsers checks prompts when they are sent, text when it is pasted and files when they are uploaded on 19 AI services, among them ChatGPT, Claude, Gemini, Microsoft Copilot and DeepSeek. Its 29 detectors for credentials and personal data pair each format with validation, such as a checksum, the structure of a token, required context or a minimum entropy. Depending on the policy, the person may cancel, send the submission with the findings removed, or, in warning mode, send it unchanged, and whole services can be blocked.

Evidence (R6). Every decision is appended to an activity log on the device. Each record names the program, the agent, the evidence that identified it, the chain of processes, the file and the operation, the decision and whether a rule, a person or the deadline made it. Each record is chained to its predecessor by a SHA-256 hash, so that a change to any record, or its removal, insertion or reordering, is detected relative to any later chain value an auditor has kept [25], [2]. The application checks the chain whenever it reads the log, and reports over a day, a week, a month or the whole log export the records with the result of that check as CSV or JSON.

Proportionality (P). IOArx observes agents, not people. Events of processes outside the lineage of every agent are discarded at the first lookup, so the employee's own work is neither decided nor recorded. Observation, decisions and the record stay on the device; the application contacts its vendor only to validate its licence and to look for updates, and neither exchange carries files, paths, prompts or activity. The extension keeps the kind and number of its findings, never their content. IOArx decides by rules that people define and draws no inferences from data, which makes it one of the systems based on rules defined by natural persons that Recital 12 of the AI Act places outside the definition of an AI system.

B. Measured Evidence

Table III summarizes measurements of IOArx 1.0 reported in [2]. Attribution was measured on a developer workstation running Windows, where it named 36 of 314 processes as belonging to four agents, 16 of them through an ancestor, without a misattribution; among 3,449 executables on disk, it recognized 69, all of them parts of agent products. Containment was measured on Linux with a stand-in agent that violated a rule at rates of up to about 2×10^5 operations per second. Prompt inspection was measured on 32,771 real user prompts from a public dataset of conversations with AI assistants, into which 14,500 credentials and personal data values of 29 kinds were also planted.

TABLE III
MEASURED PROPERTIES OF IOARX 1.0

Property	Setting	Result
Attribution	314 processes, Windows	36 attributed, 16 through an ancestor, none misattributed
Recognition	3,449 executables	69 recognized, all in agent products
Containment	Linux, Block, 120 trials	medians 3.1 and 3.2 ms for deletion and writing, all within 16 ms
Oversight hold	Linux, Ask, 90 trials	89 within 16 ms; median 4.0 ms at full speed
Decision	complete decision	median 19 μs
Inspection	32,771 real prompts	no false alarm
Detection	14,500 planted values	99.65% found
Record	10^6 entries	chain verified in 2.3 s

C. Operating IOArx in an Organization

The four functions of the NIST AI Risk Management Framework give the operating model its structure [1]. To govern, a policy owner, typically the security or data protection function, decides which locations and operations agents may use freely, which need a person and which are closed, and the organization informs employees about what is observed, namely the agents' activity and not their own. To map, each device shows which recognized agents are running and which folders and files they use, so that the organization learns which AI is actually at work, including tools that never received approval [12]. To measure, reports count allowed and blocked requests per agent and single out critical leaks, and the exported records feed the organization's reviews and the breach assessments that Table I requires. To manage, the rules, the alerts and the network gate act on each request, and a rule tester runs sample accesses through the rules and alerts before the policy takes effect, without opening any file.

The formal results also suggest an order of adoption. Rules that close credential stores and regulated data to agents come first, because they are agent-sensitive and thus beyond the reach of every lineage-blind control (Theorem 1), and because the actions they prevent become near certain at organizational scale (Proposition 2). Ask rules then go where judgement is needed; Theorem 2 shows that, once answers are recorded, the questions they raise concentrate at the beginning and then fade. The network gate in ask mode limits connections to programs someone has allowed, and the record is reviewed at intervals that match the deadlines of Table I.

VII. RECOMMENDATIONS

The analysis leads to recommendations for three groups. Organizations should treat AI agents as principals of their access policy and of their records, distinct from the employees they act for. In practice that means agent-sensitive rules for credential stores and regulated data, a person's decision where judgement matters, refusal when no one answers, and records that name the agent. Agents belong in data protection impact assessments and in incident response, since the description of a breach that Article 33 of the GDPR requires presupposes knowing what the agent accessed. Employees, and where applicable their representatives, should know what is observed, and the observation should stay with the agents and on the device.

Vendors of agents can make attribution verifiable. When every executable of an agent product, including helper programs shipped inside editor extensions, carries a code signature that names the publisher and the product, recognition rests on evidence the operating system checks. In our measurements, 12 of the 69 recognized executables, among them a search tool bundled inside an editor extension, could be recognized only by the folder they were installed in, and agents that run inside a general-purpose interpreter carry no signature of their own; a signed launcher would make both verifiable.

Standard setters and regulators can name the agent as a subject. SP 800-53 already speaks of processes acting on behalf of users [17], and zero trust policy already weighs the application next to the identity [18]; an agent acting through the programs of a user's session, however, is distinguishable from that user only by lineage (Theorem 1). Guidance that names AI agents, with every process they start, as subjects of access control and of audit would let organizations implement, and auditors test, agent-sensitive policies. Guidance on human oversight under Articles 14 and 26 of the AI Act and on the MANAGE function of the NIST framework can state that oversight belongs where the system can be held, since that is what gives the overseer causal power [21]. Data protection authorities can make explicit that agents on employees' devices fall under the controller's duties of security and accountability like any other processing, and that records at the level of the agent serve the demonstration of compliance that Article 5(2) of the GDPR requires.

VIII. CONCLUSION

AI agents act as the employees who start them, while the law holds organizations accountable for what their AI does. The two meet on the endpoint, and there the controls that decide by account, role, device or program cannot tell agent from employee, whatever their rules say. Attribution along the chain of processes removes the obstacle, oversight with remembered answers stays within the attention people can give, and enforcement that acts on every request is the only lever that scale leaves to the organization. IOArx implements these conclusions on the employee's device and keeps both the decisions and the evidence there. Organizations that delegate work to agents can thereby delegate it accountably.

ACKNOWLEDGMENT

AI tools were used to polish the text, correct grammar, improve readability.

REFERENCES

[1] E. Tabassi, "Artificial intelligence risk management framework (AI RMF 1.0)," National Institute of Standards and Technology, NIST AI 100-1, Jan. 2023, doi: 10.6028/NIST.AI.100-1.

[2] A. Mohammadi Ruzbahani, "IOArx: Agent-aware policy enforcement between AI agents and sensitive data on the endpoint," PubArx preprint 2026.00016v1 [cs.CR], Nov. 2026. [Online]. Available: https://pubarx.com/abs/2026.00016

[3] E. Brynjolfsson, D. Li, and L. Raymond, "Generative AI at work," *The Quarterly Journal of Economics*, vol. 140, no. 2, pp. 889–942, 2025, doi: 10.1093/qje/qjae044.

[4] A. Chan *et al.*, "Harms from increasingly agentic algorithmic systems," in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT '23)*. Chicago, IL, USA: ACM, Jun. 2023, pp. 651–666, doi: 10.1145/3593013.3594033.

[5] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975, doi: 10.1109/PROC.1975.9939.

[6] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*. Copenhagen, Denmark: ACM, Nov. 2023, pp. 79–90, doi: 10.1145/3605764.3623985.

[7] Q. Zhan, Z. Liang, Z. Ying, and D. Kang, "InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents," in *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, 2024, pp. 10 471–10 506, doi: 10.18653/v1/2024.findings-acl.624.

[8] E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr, "AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents," in *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024, pp. 82 895–82 920, doi: 10.52202/079017-2636.

[9] Y. Ruan *et al.*, "Identifying the risks of LM agents with an LM-emulated sandbox," in *The Twelfth International Conference on Learning Representations (ICLR 2024)*, 2024, doi: 10.48550/arXiv.2309.15817.

[10] Z. Zhang *et al.*, "'It's a fair game', or is it? Examining how users navigate disclosure risks and benefits when using LLM-based conversational agents," in *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. Honolulu, HI, USA: ACM, May 2024, pp. 1–26, doi: 10.1145/3613904.3642385.

[11] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," in *2023 IEEE Symposium on Security and Privacy (SP)*, May 2023, pp. 346–363, doi: 10.1109/SP46215.2023.10179300.

[12] M. Silic and A. Back, "Shadow IT: A view from behind the curtain," *Computers & Security*, vol. 45, pp. 274–283, 2014, doi: 10.1016/j.cose.2014.06.007.

[13] M. Siponen and A. Vance, "Neutralization: New insights into the problem of employee information systems security policy violations," *MIS Quarterly*, vol. 34, no. 3, pp. 487–502, 2010, doi: 10.2307/25750688.

[14] A. Chan *et al.*, "Visibility into AI agents," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24)*. Rio de Janeiro, Brazil: ACM, Jun. 2024, pp. 958–973, doi: 10.1145/3630106.3658948.

[15] M. Bovens, "Analysing and assessing accountability: A conceptual framework," *European Law Journal*, vol. 13, no. 4, pp. 447–468, 2007, doi: 10.1111/j.1468-0386.2007.00378.x.

[16] M. Mäntymäki, M. Minkkinen, T. Birkstedt, and M. Viljanen, "Defining organizational AI governance," *AI and Ethics*, vol. 2, no. 4, pp. 603–609, 2022, doi: 10.1007/s43681-022-00143-x.

[17] Joint Task Force, "Security and privacy controls for information systems and organizations," National Institute of Standards and Technology, NIST Special Publication 800-53, Rev. 5, Sep. 2020, doi: 10.6028/NIST.SP.800-53r5.

[18] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," National Institute of Standards and Technology, NIST Special Publication 800-207, Aug. 2020, doi: 10.6028/NIST.SP.800-207.

[19] L. Enqvist, "'Human oversight' in the EU artificial intelligence act: What, when and by whom?" *Law, Innovation and Technology*, vol. 15, no. 2, pp. 508–535, 2023, doi: 10.1080/17579961.2023.2245683.

[20] B. Green, "The flaws of policies requiring human oversight of government algorithms," *Computer Law & Security Review*, vol. 45, p. 105681, 2022, art. no. 105681, doi: 10.1016/j.clsr.2022.105681.

[21] S. Sterz *et al.*, "On the quest for effectiveness in human oversight: Interdisciplinary perspectives," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24)*. Rio de Janeiro, Brazil: ACM, Jun. 2024, pp. 2495–2507, doi: 10.1145/3630106.3659051.

[22] National Institute of Standards and Technology, "Artificial intelligence risk management framework: Generative artificial intelligence profile," National Institute of Standards and Technology, NIST AI 600-1, Jul. 2024, doi: 10.6028/NIST.AI.600-1.

[23] K. Ball, "Workplace surveillance: An overview," *Labor History*, vol. 51, no. 1, pp. 87–106, 2010, doi: 10.1080/00236561003654776.

[24] R. Böhme and J. Grossklags, "The security cost of cheap user interaction," in *Proceedings of the 2011 New Security Paradigms Workshop (NSPW '11)*. Marin County, CA, USA: ACM, Sep. 2011, pp. 67–82, doi: 10.1145/2073276.2073284.

[25] B. Schneier and J. Kelsey, "Secure audit logs to support computer forensics," *ACM Transactions on Information and System Security*, vol. 2, no. 2, pp. 159–176, May 1999, doi: 10.1145/317087.317089.