Engineering Management

Open-Sourcing the Institution

February 4, 2026

Summary

Organizational opacity is becoming unsustainable. Machine intelligence is making institutional internals legible to outsiders whether institutions cooperate or not. The costs of the opaque institutions we have — excess management, bureaucratic overhead, nepotism, lost productivity — are measured in trillions. Meanwhile, the cost of defending corporate secrets is rising sharply while the cost of extracting them falls — nation-state actors demonstrated near-zero-marginal-cost espionage in the 2025 Notepad++ supply chain compromise, shifting the tradeoff against organizational secrecy.

This paper proposes an alternative: open-source the institution itself. Not just the research outputs, but the governance, resource flows, decision history, and credit assignment. The concrete architecture — an organization repo with integrity checks, an automated recognition system, and a recursive project management layer — is implementable with existing tools. The key insight is that integrity checks are to organizational state what tests are to code: they hold the institution accountable to its own stated rules.

Machine intelligence will make individual contribution legible whether institutions cooperate or not. The choice is between a raw meritocracy where legibility serves only those at the top, or a system of explicit sharing built on honest data about who does what. Transparent institutions are the infrastructure for the second option — you cannot build fair redistribution on opaque foundations. The tooling described here is open source: https://github.com/mjtomei/project_manager.

Contents

1 Introduction

Organizations are approaching a forced transition. Machine intelligence is making institutional internals legible to outsiders whether institutions cooperate or not. Nation-state actors are driving the cost of maintaining corporate secrecy far above its competitive value — the 2025 Notepad++ supply chain compromise ran undetected for six months, and the tj-actions/changed-files attack exposed secrets across 23,000 repositories through a single compromised CI/CD dependency. And intra-organization opacity results in its own inefficiencies — excess management, bureaucratic bloat, nepotism — with costs measured in trillions. The question is no longer whether organizations will become transparent, but whether they will manage the transition or have it imposed on them.

This paper proposes managing it. We describe a research lab where the lab itself is open source — not just the research outputs, but the organizational structure, decision-making processes, resource allocation, and governance, all visible, forkable, and improvable by anyone.

The core idea is that management becomes engineering. The difference is that engineering tradeoffs are made explicit: documented, versioned, debatable, auditable. Management tradeoffs currently live in the discretion of whoever holds a management title, invisible to the people affected by them. Making them explicit infrastructure — resource allocation rules, credit assignment criteria, accountability mechanisms as code in a repo — doesn't eliminate the tradeoffs. It makes them visible so the people affected can participate in deciding them. Both positive reinforcement (recognition, credit, visibility) and negative reinforcement (integrity checks, conflict detection, staleness warnings) become auditable systems rather than unaccountable discretion.

But the argument for transparency is not only about better governance. Transparent institutions will reveal what opaque ones currently hide: that contribution is unevenly distributed. This visibility is a feature, not a bug. Opaque institutions obscure both freeloading and unrecognized contribution — they protect the people coasting and punish the people carrying the load. As machine intelligence reshapes the nature of work [22], organizations will need to decouple compensation from output, but you cannot do that honestly without knowing what the output actually is. This is not a surveillance argument. It is the observation that honest conversations about contribution require honest data, and honest data requires transparent systems.

2 The End of Organizational Opacity

Machine intelligence is making organizational opacity increasingly difficult to maintain. It can aggregate public filings, leaked documents, employment records, financial data, and published research into coherent pictures of what organizations are actually doing — regardless of what those organizations say they're doing [23, 24].

Organizations that don't open up voluntarily will have their opacity pierced by others. Companies are deploying blockchain-based supply chain verification because they can't verify what opaque suppliers claim about sourcing and labor practices [8]. The era of "trust us, it works" is ending across investment — venture investors now conduct spot-checks by contacting claimed customers directly and demanding bank statements behind reported revenue, because they

can no longer take institutional self-reporting at face value [9]. Self-insured employers are building internal claims analytics because they can't trust insurer-reported numbers — nearly half of insured adults report receiving surprise medical bills, and those who dispute denied claims get them reversed roughly half the time [10, 21].

This is parallel construction at organizational scale — building an independent picture of an institution's internals from external data, analogous to the law enforcement technique of reconstructing evidence from public sources to avoid revealing classified intelligence methods. When institutions are opaque, anyone with resources builds their own apparatus to see through them. But it's not only the powerful who gain this capability. The same tools that let investors reconstruct an institution's internals are available to employees, researchers, and the public. An individual can now aggregate an organization's public filings, job postings, Glassdoor reviews, published outputs, and financial records into a picture that used to require a dedicated analyst team. The opacity that once protected mismanagement from scrutiny above also protected it from scrutiny below. Both protections are ending at the same time.

The result is a world where the people running opaque organizations are the last to know what everyone else — funders, competitors, and their own staff — already sees. The question is not whether your organization will become transparent. It is whether you will be the one who decides how.

3 The Shifting Economics of Corporate Secrecy

The previous section described how machine intelligence reconstructs organizational internals from external data. But the case against corporate secrecy is stronger than passive reconstruction: nation-state actors are actively penetrating corporate infrastructure, and the dynamics of this contest are shifting the cost-benefit tradeoff of maintaining secrets decisively against the defender.

In June 2025, a Chinese state-sponsored hacking group compromised the hosting infrastructure of Notepad++, an open-source text editor with hundreds of millions of downloads [26]. The attackers did not exploit a vulnerability in the software itself. They gained access to the shared hosting provider, intercepted update traffic, and selectively redirected targeted users to malicious servers delivering a custom backdoor codenamed Chrysalis [27]. The operation ran undetected for six months. When Notepad++ creator Don Ho disclosed the incident in February 2026, he could not even provide indicators of compromise — after a week analyzing 400 GB of server logs, his incident response team found "signs of an intrusion" but "no concrete indicators of compromise — such as binary hashes, domains, or IP addresses" [26]. The attackers were that clean.

Notepad++ is a text editor maintained by a small team. Consider what the same capability implies when directed at a corporation's proprietary infrastructure. If a nation-state actor can silently compromise a software update mechanism for six months, it can read a corporation's internal communications, source code, strategic plans, and trade secrets at leisure. The Notepad++ case is not exceptional — it is one instance in a growing pattern of supply chain compromises including SolarWinds (2020), Codecov (2021), 3CX (2023), and the tj-actions/changed-files GitHub Action (2025) [25], each demonstrating the same fundamental dynamic: infrastructure that organizations trust implicitly becomes the attack vector.

Machine intelligence is accelerating this contest, but the acceleration is asymmetric. Both attackers and defenders gain AI-assisted tools for vulnerability discovery, code analysis, and anomaly detection. But the attacker's problem is structurally easier: find one exploitable weakness. The defender must protect every surface. As machine intelligence makes both sides faster, the asymmetry does not shrink — it compounds. A system that can audit a million lines of code for vulnerabilities in hours helps the defender, but it helps the attacker more, because the attacker only needs one of those vulnerabilities to succeed.

This asymmetry also means that a corporation's home country defense budget is irrelevant. A company headquartered in the country with the world's largest military and intelligence apparatus is not thereby protected from espionage by a smaller state. The defending nation would need to secure every piece of infrastructure its corporations depend on — every hosting provider, every software dependency, every cloud service, every third-party vendor. The attacking nation needs to find one gap in that chain. The Notepad++ attackers did not breach a hardened government target; they compromised a shared hosting server running a PHP script. The supply chain extends far beyond what any defense establishment monitors or controls.

The current state of corporate infrastructure makes this worse than the asymmetry alone would suggest. Most organizations' software supply chains were never designed to resist nation-state attackers. Update mechanisms lack cryptographic verification. Dependencies are pulled from public registries without integrity checks. Hosting infrastructure is shared. Internal networks assume perimeter security. These are not obscure vulnerabilities — they are standard practice. Notepad++ lacked certificate verification in its updater until version 8.8.8, released in November 2025 [26]. This was not negligence; it was normal. The tj-actions/changed-files incident in March 2025 demonstrated the same pattern in CI/CD infrastructure: an attacker compromised a single GitHub Action used by over 23,000 repositories, injecting code that dumped CI runner secrets — AWS access keys, GitHub tokens, private RSA keys — into publicly readable workflow logs [25]. The attack cascaded through a chain of dependent actions (reviewdog/action-setup $\rightarrow$ tj-actions/eslint-changed-files $\rightarrow$ tj-actions/changed-files), illustrating how a single compromised dependency propagates through the ecosystem. This is the infrastructure that corporations rely on to build and deploy their proprietary software. The entire ecosystem is unprepared, which means there will be many entry points for sophisticated attackers, at least for the foreseeable future while the industry catches up.

The implication for organizational opacity is direct: the cost of defending corporate secrets is rising rapidly while the value of those secrets — relative to what an attacker can extract — is falling. Once automated espionage tooling is built and deployed, the marginal cost of targeting an additional organization approaches zero. The traditional justification for corporate opacity — that secrecy protects trade secrets and competitive position — assumed a world where keeping secrets was cheap relative to their value. That tradeoff is inverting for any organization of sufficient interest to a state actor.

Open organizations shift competitive advantage to ground that is not vulnerable to this dynamic. When your governance, processes, and institutional knowledge are public by design, the attack surface for espionage shrinks dramatically. The competitive advantage shifts to execution speed, talent density, and institutional culture — none of which can be stolen by compromising a hosting provider. This is not an argument that all information should be public. It is the observation that the set of information worth keeping secret is much smaller than most organizations assume, and the cost of defending even that smaller set against state-level adversaries is higher than most organizations realize.

4 The Cost of Opaque Institutions

The economic costs of organizational opacity are substantial, though precise measurement is difficult. The estimates in Table 1 draw from a mix of peer-reviewed research, industry surveys, and trade publications of varying methodological rigor.

In one Australian study, non-administrative staff reported spending 6.4 hours per week — 16% of a standard work week — complying with internal regulation, and the cost of self-imposed internal rules exceeded the cost of government regulation by more than two to one [14].

Nepotism and favoritism compound these costs. Nearly 75% of employees report having worked in a toxic workplace, with poor leadership — including favoritism and lack of accountability — cited as the top cause by 79% of them [15]. Toxic culture is 10 times more important than compensation in predicting turnover [16]. When unqualified hires hold leadership roles through connections rather than competence, the remaining staff pick up the slack, leading to burnout and further turnover. The organizational knowledge needed to actually run the institution is never written down — it lives in the heads of whoever happens to be politically connected enough to stay.

The same patterns appear in public institutions. Researchers receiving federal grants spend over 44% of their time on administrative tasks rather than research — a figure that has remained stubbornly consistent across FDP surveys in 2005, 2012, and 2018 [17]. Nepotism in public institutions creates wasteful overstaffing where unneeded positions are created to employ relatives — documented across municipalities as a drag on economic development [19]. Perceived nepotism is negatively associated with educational investment across countries, as measured by PISA scores [20]. Funding decisions happen behind closed doors. Credit assignment is political. Access to resources depends on who you know, not what you contribute.

These costs become especially dangerous heading into machine-intelligence-driven economic disruption. If we cannot see who contributes what inside our institutions, we have no foundation for the redistribution that machine-intelligence productivity gains will require. Amodei [22] argues that society will need to pay people even when they are not providing traditional economic value. That is not possible to do fairly — or even to do at all — without transparent systems that make contribution legible in the first place.

Cost	**Estimate**	**Source**
Excess management & bureaucracy (U.S.)	\$3T/yr (∼17% GDP)	[11]
Fortune 500 back-office inefficiencies	\$480B/yr	[12]
Regulatory compliance (U.S. aggregate)	\$103–289B/yr	[13]
Toxic workplace culture (U.S.)	\$50B/yr	[16]
Hospital claims overturn spending	\$19.7B (2022)	[21]
Unreimbursed university overhead	\$6.8B/yr	[18]
Markets responding to opacity		
Blockchain supply chain verification	\$2.9B → \$44.3B by 2034	[8]
Open-source intelligence (OSINT)	\$15B → \$49B by 2029	[23]

Table 1: Estimated costs of organizational opacity and markets emerging to circumvent it. Figures are U.S.-specific unless noted.

5 Open-Sourcing the Institution

Most “open science” initiatives open-source the outputs (papers, data, code) but keep the institution opaque. This lab open-sources the institution itself: governance, resource flows, decision history, credit assignment. The structure is the product.

6 Architecture

Here is what it looks like when an organization has no secrets from itself. Higher-level units (the lab itself, divisions within it) have two layers of state: an organization repo and a project management layer (managed by `pm`). Individual research projects below them may only need the pm layer and their code repo. The organizational layer is for units that need to track more than just a tech tree.

6.1 The Organization Repo

A GitHub repo that is the public record of the organization. This is where things live that don’t belong in any individual project but need to be tracked: documents, small scripts, spreadsheets, test programs, analyses, meeting notes, governance decisions, and anything else that should be visible and version-controlled but doesn’t need wide release as a standalone project.

It also includes pointers to the other repos and projects in the organization — individual project repos, external dependencies — in `docs/` so it can act as a central dispatch.

```
org-repo/
|-- docs/                          # golden copy of project state
|   |-- governance/                # decision-making processes,
    roles
|   |-- research-agenda/           # what the lab works on and why
|   |-- resources/                 # grant tracking, allocation
    records
|   `-- onboarding/                # how to join, how things work
|-- members/
|   |-- alice/                     # Alice's working directory
|   |   |-- notes/                 # her meeting notes, scratch work
|   |   |-- scripts/               # one-off analysis scripts
|   |   `-- proposals/             # drafts she's working on
|   |-- bob/
|   |   |-- notes/
|   |   |-- data/                  # small datasets, spreadsheets
|   |   `-- experiments/           # test programs, prototypes
|   `-- ...
|-- checks/                        # integrity checks (see below)
`-- archive/                       # completed work, historical
    records
```

Member directories. Every member gets their own directory — their workspace within the org. They can put anything there: notes, scripts, Excel sheets, draft proposals, experimental

code. Their work is visible to the rest of the organization without requiring them to publish it anywhere else.

When a member wants to promote something to the project-wide golden copy, they open a PR moving or copying it into `docs/`. The PR is a request for the organization to recognize and adopt the work. Other members review it, and the merge is the organization saying "yes, this is now part of our shared state."

The `docs/` directory. The golden copy — the authoritative state of the project. Nothing gets into `docs/` without a PR. The PR history in `docs/` is the lab's institutional memory.

The `checks/` directory. Automated integrity checks that project managers create and maintain. These run against the repo (via CI or on-demand) and surface issues that humans should look at.

6.2 Integrity Checks

Integrity checks are to organizational state what tests are to code. When machine intelligence produces code, tests check that the code does what it claims. When an organization produces documents, decisions, and resource allocations, integrity checks do the same thing: they verify organizational state against the organization's own stated rules and surface inconsistencies for human judgment.

This applies to any organization that manages resources — public or private, academic or corporate. The checks maintain a history of both automated and human-assisted audits. Every check run is logged. When a human reviews an issue surfaced by a check and makes a judgment call, that judgment is recorded too. Over time this builds an audit trail showing not just the current state but how the organization has responded to issues as they were discovered — an accountability mechanism that persists regardless of personnel changes.

Examples:

- **Consistency**: does the resource allocation in `docs/resources/` add up? Do budget numbers match grant amounts recorded in governance decisions?
- **Staleness**: are there proposals sitting for months with no PR to `docs/`? Governance documents referencing people who are no longer active?
- **Completeness**: does every active grant have an allocation record? Does every research project in the agenda have a corresponding pm instance?
- **Conflict**: contradictory statements across governance documents? Two proposals allocating the same funds differently?
- **Attribution**: does every work product have clear attribution? Documents referencing work without crediting the contributor?

These checks can be simple scripts, LLM-powered analyses, or structured validators. They're not gates — they surface issues for human judgment, same as a test suite surfaces failures for a developer to evaluate. The checks themselves are in the repo, subject to the same review process as everything else.

6.3 Automated Recognition

The same infrastructure that detects problems also recognizes achievements. In traditional organizations, recognition flows through managers, shaped by who is visible, who is in the right social circle, who shares the manager's background. Automated recognition replaces this with a read operation on organizational state that's already being maintained.

Examples: milestone completion across a tech tree; a new member's first PR to `docs/`; sustained contribution over time; a PR that unblocks several downstream projects; a grant's deliverables all completed with the full funding-to-output chain visible.

Because the org repo and pm state are readable text and structured data, anyone can ask an LLM to assess contributions holistically — reading a member's directory, their PRs, the projects they've touched, the discussions they've participated in. This is closer to what a thoughtful colleague would say if they'd been paying attention to everything, which no human can do at organizational scale.

This avoids the problems of fixed metrics. Lines of code, commit frequency, and citation counts reward the metric instead of the work. LLM-based assessment introduces its own risks — LLMs have predictable biases toward verbose, confident, and recently-active contributors, and their evaluation can be anticipated and gamed. The game doesn't disappear; it becomes a meta-game. But because the recognition criteria, prompts, and outputs are all in the repo, anyone can see how the game is being played. The meta-game itself is auditable — if someone is optimizing for the evaluator rather than doing useful work, that pattern is visible in the same data the evaluator reads. You can write integrity checks against the meta-game the same way you write them against the organization's primary state.

The point is that the organization notices what its members accomplish without requiring self-promotion or managerial attention. The system does the noticing. People decide what to do with it. This is not a tool for optimizing productivity — it will surface what looks like inefficiency but is actually useful exploration, and the organization should understand that distinction.

6.4 The Project Management Layer

Managed by `pm`[1], the project management layer maintains a `project.yaml` file, `plans/` directory, and a dependency graph (the "tech tree") of everything that needs to happen. `pm` tracks what needs to be built, in what order, what's blocked, what's ready, and who's working on what. It generates prompts for machine intelligence coding agents and detects when branches merge, automatically unblocking downstream work. The pm state can live inside the organization repo or in a standalone repository — the tooling is agnostic.

The pm layer integrates with a recursive tech tree system. The lab's top-level tree connects the organizational layer to all the research projects below it. In *prescriptive* mode, a parent tree suggests work to child projects (the child opts in and can accept or decline). In *descriptive* mode, a parent tree observes child projects without directing them — a read-only aggregation that requires no coordination.

[1] `pm` is an open-source CLI tool for managing dependency graphs of work items across parallel contributors. Source and documentation: https://github.com/mjtomei/project_manager

The organization repo and the pm layer serve separate concerns: the org repo is the record of what the organization is and has done; the pm layer is the plan for what it's doing next. Setting up a new organization requires initializing both: `pm init` for the pm layer, and creating the org repo with the directory structure described in Section 6.1. A walkthrough demonstrating the full setup — from an empty repo to a working organization with integrity checks, member directories, and a populated tech tree — is forthcoming as a companion to this paper.

The pm layer will expose what looks like inefficiency — blocked work, stalled plans, idle projects. Some of that is real inefficiency. Some of it is useful exploration. The visibility lets the organization have honest conversations about what's happening.

7 Relationship to Existing Institutions

The lab is additive, not exclusionary. Researchers can participate while holding positions at existing institutions. Grants can come from traditional funders and flow through traditional fiscal sponsors. The transparency is about what happens with the resources, not about requiring new channels.

As long as existing power structures persist, people will interface with governments through existing institutions for taxes, employment law, visa sponsorship, and compliance. A researcher receiving a stipend still files taxes through their employer. A grant still goes through a fiscal sponsor. The open layer sits on top of these structures, making decisions and flows visible without pretending the underlying structures don't exist.

This also matters for funders. All else equal, an organization where a machine intelligence can read and verify what's actually happening is less risky to fund than an opaque one. As this kind of transparency becomes feasible, it will increasingly be expected — not because anyone mandates it, but because the alternative becomes an obviously worse bet.

8 Limitations and Open Questions

The transparency paradox. Bernstein [6] demonstrated that factory workers were more productive when given privacy from management observation. This is a genuine finding, but the mechanism matters: Bernstein's workers had no control over the observation system and no ability to use it themselves. Transparency was asymmetric — management watched workers, not the reverse. In that configuration, transparency is surveillance, and surveillance induces hiding behavior. This project is specifically an attempt at solving the problem Bernstein identified, by making transparency symmetric. When every member has the same machine-intelligence-assisted access to organizational state that executives and project managers have, the power asymmetry that drives hiding behavior is substantially reduced. A tech worker with machine intelligence tools can read governance documents, audit resource flows, and assess organizational health with nearly the same capability as the people nominally in charge. The mechanism described here is more symmetric than prior transparency systems precisely because machine intelligence makes organizational legibility available to everyone, not just to those with management titles. Nevertheless, the risk of chilling effects on exploratory work remains real. Members might avoid speculative projects that could look unproductive. The

member directories are designed to mitigate this — they are personal workspaces where incomplete and exploratory work is expected — but the tension between organizational legibility and individual creative freedom requires ongoing attention.

Legitimate uses of opacity. Not all organizational opacity is pathological. Negotiating positions, personnel matters, and shielding individuals from external pressure are cases where some degree of opacity serves legitimate purposes. The architecture described here does not require total transparency — the org can choose what is world-readable and what is member-only. But the paper's argument is strongest for governance, resource allocation, and credit assignment, where opacity more often enables abuse than serves legitimate ends. The traditional case for trade secret protection is further weakened by the analysis in Section 3: as the cost of defending secrets against state-level adversaries rises, the net competitive value of maintaining them falls.

LLM gaming and bias. The automated recognition system inherits the risks of algorithmic management documented by Kellogg et al. [7]. LLMs have predictable biases and their assessments can be gamed by those who understand the model's preferences. Making the recognition infrastructure open and auditable means the meta-game — gaming the evaluator rather than doing the work — is itself visible and auditable. This does not eliminate strategic behavior, but it makes each layer of strategy legible to anyone who cares to look. Formal analysis of these recursive gaming surfaces is future work.

No empirical validation. This paper describes an architecture, not empirical results. The claims about cost reduction, improved fairness, and organizational health are predictions, not measurements. A pilot deployment (even at small scale: 5–10 people for 3–6 months) measuring administrative overhead, perceived fairness, and contribution distribution would substantially strengthen the argument. Formal modeling — agent-based simulation of organizational dynamics under transparent vs. opaque regimes, game-theoretic analysis of strategic behavior, principal-agent models adapted to symmetric transparency — is also future work. This is a design proposal; the empirical and theoretical validation comes next.

Cost figures. The economic estimates in Section 4 draw from sources of varying rigor. They indicate the scale of the problem but should not be treated as precise measurements. In particular, the $3 trillion figure from Hamel [11] is based on extrapolation from exemplar companies, not a comprehensive economic study.

9 Related Work

The idea of making organizational governance explicit and transparent has a substantial history. This section positions the present work relative to prior approaches and identifies what is genuinely new.

Commons governance. Ostrom [1] established eight design principles for self-governing institutions managing shared resources: clearly defined boundaries, proportional costs and benefits, collective choice arrangements, monitoring, graduated sanctions, conflict resolution mechanisms, minimal recognition of rights to organize, and nested enterprises. The architecture described in this paper instantiates several of these principles — integrity checks as monitoring, governance-as-code as collective choice arrangements, the recursive tech tree as

nested enterprises — in a version-controlled, machine-readable form. Ostrom's framework provides the theoretical grounding; we provide a concrete implementation using modern software infrastructure.

Self-management frameworks. Holacracy [2] encodes governance rules explicitly, replacing traditional management hierarchy with a constitution that defines roles, accountabilities, and decision-making processes. Laloux [3] surveys organizations operating with radical transparency, including Buurtzorg and Morning Star. Our approach shares the goal of making governance rules explicit but differs in two ways: the rules are stored in Git (not a proprietary constitution), making them forkable and diffable; and machine intelligence is used to check organizational state against the rules, rather than relying solely on human process adherence. Holacracy's adoption difficulties — Zappos being the most prominent case — also illustrate that rigid governance frameworks can fail when they don't accommodate organizational ambiguity. Our approach is deliberately flexible: governance documents are prose, not a fixed schema.

Decentralized Autonomous Organizations (DAOs). DAOs encode governance in smart contracts on a blockchain, making institutional rules machine-readable and forkable [4]. The goals overlap significantly with ours. The key differences are infrastructural: our approach uses Git, which is universally understood by developers, does not require tokens or chain infrastructure, and integrates with existing development tools. DAOs also have a well-documented history of governance failures — including the 2016 DAO exploit, where transparent machine-readable rules were used against the organization, and persistent voter apathy in token-based governance. We take these as evidence that governance transparency is necessary but not sufficient; the integrity checks and recognition system described in Section 6 are designed to address some of these failure modes.

Open-source governance. The open-source software movement provides decades of evidence on transparent organizational structures [5]. Research on open-source governance shows both the power of open contribution models and their failure modes: hidden hierarchies, burnout among maintainers, and difficulty with accountability when no one is formally in charge. Buffer's radical transparency experiment (public salaries and open financials since 2013) provides real-world data on organizational transparency in a corporate setting. Valve's nominally flat structure produced documented problems with clique-based resource allocation despite apparent openness — a cautionary tale that motivates our integrity checks, which are designed precisely to surface hidden power structures that persist even under nominal transparency.

The transparency paradox. Bernstein [6] found that factory workers were more productive when given privacy from management observation, showing that transparency can reduce performance by inducing hiding behavior. This is an important counterpoint, but the mechanism matters: Bernstein's workers had no control over the transparency system and no ability to use it themselves. The transparency was asymmetric — management watched workers, not the reverse. This project is specifically an attempt at solving that problem by making transparency symmetric: machine intelligence gives every member the same ability to read and audit organizational state that was previously reserved for management. The remaining risk — that even symmetric transparency produces chilling effects on exploratory work — is addressed in Section 8.

Algorithmic management. Kellogg et al. [7] provide a systematic review of algorithmic man-

agement in the workplace, documenting how automated systems can reproduce or amplify existing power asymmetries. The automated recognition system described in Section 6.3 is an instance of algorithmic management and inherits these risks. We address this by making the recognition infrastructure itself open and auditable — the criteria are code, not black-box algorithms — but acknowledge that this does not eliminate all risks (see Section 8).

10 Conclusion

Organizations rely on opacity — implicit systems of credit, funding, access, and decision-making understood by insiders but invisible to outsiders and unaccountable to anyone. Machine intelligence is ending this arrangement. Machine intelligence already reconstructs what institutions are actually doing regardless of what those institutions claim. This paper argues that organizations face a choice: adopt structured transparency on their own terms, or have it imposed from outside. We describe an architecture for *open-sourcing the institution itself* — governance, resource flows, decision history, and credit assignment — using version-controlled repositories, automated integrity checks, and machine-readable organizational state. Management becomes engineering: both positive reinforcement (recognition, credit, visibility) and negative reinforcement (integrity checks, conflict detection, staleness warnings) are code in the repository, auditable by anyone. We present the design of a research lab built on these principles and discuss its applicability to any organization that manages resources and produces work products.

References

[1] E. Ostrom, *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990.

[2] B. J. Robertson, *Holacracy: The New Management System for a Rapidly Changing World*. Henry Holt, 2015.

[3] F. Laloux, *Reinventing Organizations*. Nelson Parker, 2014.

[4] D. Rozas, A. Tenorio-Fornés, S. Djunadi, and J. Hassan, "Analysis of the Potentials of Blockchain Technology for Commons Governance," 2021.

[5] E. S. Raymond, *The Cathedral and the Bazaar*. O'Reilly, 1999.

[6] E. S. Bernstein, "The Transparency Paradox: A Role for Privacy in Organizational Learning and Operational Control," *Administrative Science Quarterly*, vol. 57, no. 2, pp. 181–216, 2012.

[7] K. C. Kellogg, M. A. Valentine, and A. Christin, "Algorithms at Work: The New Contested Terrain of Control," *Academy of Management Annals*, vol. 14, no. 1, pp. 366–410, 2020.

[8] Market.us, "Blockchain for Supply Chain Traceability Market Size," 2024. https://market.us/report/blockchain-for-supply-chain-traceability-market/

[9] Triunity Capital Ventures, "Due Diligence in 2025: What Investors Really Want to Know About Your AI Startup." https://triunitycapitalventures.com/due-diligence-in-2025-what-investors-really-want-to-know-about-your-ai-startup/

[10] AJMC, "Survey Exposes Pervasive Billing Errors, Aggressive Tactics in US Health Insurance," 2024. https://www.ajmc.com/view/survey-exposes-pervasive-billing-errors-aggressive-tactics-in-us-health-insurance

[11] G. Hamel, "Excess Management Is Costing the U.S. $3 Trillion Per Year," *Harvard Business Review*, 2016. https://hbr.org/2016/09/excess-management-is-costing-the-us-3-trillion-per-year

[12] SSO Network, "Fortune 500: $480 Billion in Back Office Inefficiencies a Year." https://www.ssonetwork.com/continuous-improvement-process-improvement/articles/480-billion-in-back-office-inefficiencies-how-to

[13] NBER / Cato, "The Cost of Regulatory Compliance in the United States." https://www.cato.org/research-briefs-economic-policy/cost-regulatory-compliance-united-states

[14] Deloitte Australia, "Get Out of Your Own Way — Unleashing Productivity," cited in *Consultancy.uk*. https://www.consultancy.uk/news/973/deloitte-compliance-costs-australian-economy-250-billion

[15] iHire, "Toxic Workplace Trends Report," February 2025 ($n = 2{,}285$ across 57 industries). https://www.ihire.com/about/press/ihire-toxic-workplace-trends-report-pr

[16] D. Sull, C. Sull, W. Cipolli, and C. Brighenti, “Why Every Leader Needs to Worry About Toxic Culture,” *MIT Sloan Management Review*, March 2022. $50B figure originally from SHRM, “The High Cost of a Toxic Workplace Culture,” 2019. https://sloanreview.mit.edu/article/why-every-leader-needs-to-worry-about-toxic-culture/

[17] S. L. Schneider, “2018 FDP Faculty Workload Survey: Research Report: Primary Findings,” Federal Demonstration Partnership Foundation, 2020. https://thefdp.org/wp-content/uploads/FDP-FWS-2018-Primary-Report.pdf

[18] Association of American Universities, “Frequently Asked Questions about Facilities and Administrative Costs of Federally Sponsored University Research.” https://www.aau.edu/key-issues/frequently-asked-questions-about-facilities-and-administrative-costs

[19] “Nepotism, Political Competition and Overemployment,” *Political Research Exchange*, 2020. https://www.tandfonline.com/doi/full/10.1080/2474736X.2020.1781542

[20] “Nepotism, Human Capital and Economic Development,” *Journal of Economic Behavior & Organization*, 2020. https://www.sciencedirect.com/science/article/abs/pii/S0167268120304431

[21] American Hospital Association, “Skyrocketing Hospital Administrative Costs: Burdensome Commercial Insurer Policies Are Impacting Patients and the Health Care Workforce,” September 2024. https://www.aha.org/guidesreports/2024-09-10-skyrocketing-hospital-administrative-costs-burdensome-commercial-insurer-polici

[22] D. Amodei, “The Adolescence of Technology,” 2026. https://darioamodei.com/essay/the-adolescence-of-technology

[23] Future Market Insights, “Open Source Intelligence (OSINT) Market Size, Share & Trends,” 2024. https://www.futuremarketinsights.com/reports/open-source-intelligence-market

[24] A. Kim, M. Muhn, and V. Nikolaev, “Financial Statement Analysis with Large Language Models,” *Chicago Booth Research Paper*, 2024. https://bfi.uchicago.edu/working-paper/financial-statement-analysis-with-large-language-models/

[25] Wiz Threat Research, “GitHub Action tj-actions/changed-files Supply Chain Attack (CVE-2025-30066),” March 2025. https://www.wiz.io/blog/github-action-tj-actions-changed-files-supply-chain-attack-cve-2025-30066

[26] D. Ho, “Notepad++ Hijacked by State-Sponsored Hackers,” February 2026. https://notepad-plus-plus.org/news/hijacked-incident-info-update/

[27] Rapid7, “Notepad++ Hosting Breach Attributed to China-Linked Lotus Blossom Hacking Group,” reported by *The Hacker News*, February 2026. https://thehackernews.com/2026/02/notepad-hosting-breach-attributed-to.html