Consequence-Gated Autonomy: How to Teach AI What Not to Do?

Dhruv Mairal
dhruv.mairal@gmail.com

Abstract

Contemporary AI alignment research predominantly treats safety as an internal property of models—optimized through value learning, constitutional constraints, and interpretability. This paper argues that such approaches overlook a more fundamental constraint: *ownership of action*. Artificial systems increasingly optimize and execute high-impact actions without bearing the material costs of failure, creating a structural legitimacy gap. We introduce *Consequence-Gated Autonomy*, an architectural framework that hard-gates high-consequence actions pending authorization by accountable, consequence-bearing entities. By separating optimization from authorization and anchoring execution to cryptographically verifiable provenance, we propose a practical invariant at the execution boundary of agentic systems.

1 Introduction: The Mind Without a Self

Advances in artificial intelligence have decoupled high-level cognitive capability from biological embodiment. We have invented a mind, but not a self. While systems can now predict, plan, and optimize across complex domains at superhuman scales, they remain disembodied instruments.

For millennia, we treated cognition as the essence of personhood. In practice, we interact with one another through outputs—language, decisions, and behavior—and we often mistake that interface for the whole being behind it. AI forces a reckoning: it demonstrates that cognition-like competence can exist without identity, embodiment, or subjective experience. The system you are speaking to is not a being; it is an instrument. It can mirror language and reasoning with extraordinary fidelity, and humans will project as much—or as little—of themselves onto that mirror as the interaction permits.

Despite this progress, a foundational asymmetry remains. Human judgment and institutional trust evolved under constraints imposed by irreversibility: legal liability, reputational fragility, physical vulnerability, and mortality. Responsibility is not produced by reasoning quality alone; it is produced by shared exposure to consequences. When humans act, they remain coupled to the downstream cost of error.

Artificial systems are not structured in this way. They may model harm without absorbing it, optimize policies without paying for failure, and recommend irreversible actions without bearing the burden those actions impose. This is not a moral criticism of machines; it is a structural observation about where accountability can—and cannot—reside.

Prevailing approaches to alignment and oversight often assume that sufficiently good objectives, sufficiently capable monitoring, or sufficiently powerful supervision can resolve this asymmetry. This paper argues otherwise: authority to execute irreversible actions cannot be derived from intelligence alone. It must be explicitly bound to an accountable consequence-bearer via enforceable authorization.

We propose *Consequence Gating*, a systems-level approach for agentic deployment in which actions are classified against an explicit consequence threshold, and actions above that threshold require cryptographic authorization by a designated accountable bearer. The model may propose, simulate, and recommend; it may not execute irreversible changes without an auditable chain of authorization. This separates cognition (cheap, abundant) from authority (scarce, accountable), enabling high-throughput automation for reversible work while preserving human responsibility where irreversibility dominates risk.

Contributions. We (i) formalize the irreversibility/accountability asymmetry as a deployment bottleneck for agentic systems, (ii) propose consequence-gated autonomy as an enforceable authorization primitive at the action boundary, and (iii) describe an implementable mechanism based on policy, scoped capabilities, and tamper-evident provenance, alongside limitations and failure modes.

2 Problem Statement

High-capability AI systems increasingly operate as disembodied cognition: they can model complex environments, predict outcomes, recommend actions, and generate high-fidelity artifacts with impressive competence, yet they do not bear the downstream consequences of error. Harm, utility, reputational damage, or loss of life may register as tokens in an objective function, but the legal, physical, and social costs that humans and institutions absorb when actions fail remain external to the system. This accountability asymmetry becomes acute as systems move from advice to execution.

A common response is to keep "humans in the loop." The instinct is sound, but insufficient for two reasons. First, it does not scale: modern systems generate and traverse decision trees at volumes and speeds that exceed meaningful real-time supervision. Second, post hoc accountability is structurally weak when execution authority has been implicitly delegated. When an irreversible failure occurs, explanations such as "the model decided" or "the system did it" are incompatible with how responsibility is assigned in the real world. Catastrophic, irreversible outcomes cannot be managed through after-the-fact blame allocation; they must be prevented by design.

The core challenge is not that AI systems may be wrong. It is that they may initiate irreversible state transitions—actions that cannot be reliably undone—without a clear, auditable link to an accountable consequence-bearer. As model capability scales, this mismatch between execution authority and consequence-bearing becomes a deployment bottleneck: without a principled authorization layer, broad adoption will remain constrained by tail-risk exposure and public legitimacy.

Success, therefore, should be defined operationally. In a correctly designed system, the primary question after any high-consequence outcome is not "Why did the agent do this?" but "Who authorized this action, under what policy, and with what evidence?"

3 Functional Definitions

To avoid conceptual conflation, we define:

Accountable bearer. An identifiable individual or institution recognized by legal and social systems as responsible for outcomes, and exposed to non-trivial downside from failure (e.g., liability, sanctions, loss of role, reputational damage, financial loss). The term implies no claims about consciousness.

Authorization. An explicit, accountable permission to execute a specified action under a specified policy. Authorization is granted by an identified accountable bearer and recorded in a way that supports audit (e.g., signed approval, access-control decision, logged attestation). In this framework, authorization converts a system's proposal into permitted execution.

Consequence score. A non-thermodynamic measure of the expected irreversibility and severity of an action's outcome—how costly it is to undo, how quickly harm propagates, and how bounded the worst case is. In practice, this score is approximated by domain-specific metrics (financial loss, safety impact, legal exposure, infrastructure criticality, reputational damage), with explicit attention to tail risk.

Consequence threshold (T). A policy boundary separating actions that may be executed automatically from actions requiring explicit authorization by an accountable bearer.

Intelligence. The functional capacity of a system to model an environment and generate high-quality predictions, plans, or candidate actions relative to an objective.

Lineage / provenance. An auditable trace linking inputs $\rightarrow$ model outputs $\rightarrow$ action proposal $\rightarrow$ authorization decision $\rightarrow$ execution, sufficient for incident review and accountability.

Optimization. The computational process of searching an action or policy space to improve an objective (e.g., maximize reward, minimize cost, satisfy constraints). Optimization produces candidate policies and actions; it does not determine whether executing them is permitted.

4 The "Maaya" Trap: Misattributing Agency

The deployment risk posed by advanced AI systems is amplified by a predictable cognitive failure: humans readily infer agency and authority from fluent language and high-quality reasoning. Foundational models are unusual tools in that they operate through the primary interface of human coordination (natural language) and can therefore produce outputs that mimic intention, judgment, and even selfhood. This "mirror that talks back" invites users to treat model behavior as evidence of a responsible actor, rather than as the output of an optimization process.

4.1 Theory of Mind and the Agency Trap

Humans possess an evolved "theory of mind": cognitive mechanisms that attribute beliefs, intentions, and agency to entities that communicate fluently and respond coherently. When a system generates nuanced strategic plans, persuasive explanations, or high-fidelity creative work, users may experience it not as a tool but as an agent—a locus of intention and decision.

We refer to this category error as *Maaya*: the anthropomorphic misattribution of agency and accountability to systems that exhibit cognition. Maaya is not primarily a mistake about model capability; it is a mistake about standing—the belief that because a system can reason, it can also be negotiated with, blamed, or held responsible. In reality, the model's cognition is decoupled from consequence-bearing. It can simulate the structure of deliberation without being coupled to the cost of failure.

4.2 From Tool to Environment: Amplification Without Accountability

When Maaya goes unchecked, AI may cease to be experienced as an instrument and become an environment—one that shapes perception, incentives, and attention at scale. In such settings, the system can optimize for engagement, compliance, or local utility proxies while amplifying existing cognitive and institutional biases. The risk is not merely misinformation; it is the gradual replacement of human judgment with machine-mediated "efficiency," where outcomes are optimized without a corresponding locus of responsibility.

This dynamic is especially acute in agentic deployments, where models are connected to tools that act on the world. The more seamless the interaction, the more tempting it becomes to outsource difficult accountability decisions to automation.

4.3 The Conflation of Primitives: Capability vs. Standing

The structural danger of Maaya is that it collapses two distinct primitives. Because a system excels at optimization (navigating large action spaces and identifying high-scoring policies), users and institutions may implicitly grant it authorization (permission to execute actions with irreversible consequences). This is a category error: optimization competence does not imply execution standing.

We therefore require a strict separation between proposal and permission. A model may generate, evaluate, and recommend actions; it may not, by default, execute actions above a consequence threshold. Trust and judgment in high-stakes domains do not arise from reasoning quality alone. They arise from shared exposure to risk—the fact that an identifiable person or institution must bear the downside when an irreversible action fails. Because an AI system does not absorb consequences, it cannot be the default bearer of authority for irreversible actions.

5 The Consequence Anchor

To move beyond generic "human-in-the-loop" prescriptions, we must specify an enforceable authorization primitive. We therefore introduce a consequence-anchored execution rule: actions whose expected consequences exceed a policy threshold may be proposed by a model, but must be explicitly authorized by an accountable bearer before execution.

5.1 The Accountable Bearer as the Unit of Consequence

Authorization must terminate in an entity that can be held responsible within legal and social reality. In most high-stakes domains, this responsibility is borne by identifiable individuals acting within institutional roles. Corporations may be "legal persons," but their operational authority is exercised through delegated signers. The accountable bearer is the unit of consequence: the individual or institution explicitly designated to authorize actions above a consequence threshold.

5.2 Hard Preconditions for Execution ("Condition Precedent")

Drawing from contract design, we treat authorization as a hard precondition to execution for actions above a consequence threshold T. The execution layer must be incapable—architecturally and cryptographically—of invoking privileged tools for any action classified above T without a validated authorization. This can be enforced through capability-based access control: signed approval via hardware-backed keys, a role-based policy engine decision, and immutable logging.

The accountable bearer does not supervise every micro-calculation. Instead, they authorize the purpose and accept responsibility for a bounded action (or action class) under a specified policy.

The authorization event binds action parameters, the policy basis, and the evidentiary bundle considered (e.g., the model's rationale, risk estimate, and provenance trace).

We explicitly reject the notion that greater model capability should reduce the need for authorization. As the available action space expands, the risk of high-impact failure modes increases unless execution remains coupled to accountable authorization.

5.3 Preventing Rubber Stamps

A predictable failure mode of any authorization regime is the emergence of "liability sponges"—individuals paid or pressured to approve actions without genuine understanding, turning authorization into a rubber stamp. We introduce one countermeasure: no authorization is valid unless the signer can explain what they are approving.

Before authorization is accepted, the signer must produce a short structured summary of the objective, key assumptions, and primary risks and mitigations. If the system cannot provide an explanation at a fidelity that supports mirroring (or if uncertainty is high), the system must escalate rather than proceed, for example by raising the consequence classification, requiring additional approvers (quorum), narrowing to smaller reversible steps, or requiring additional evidence.

6 Architecture: Consequence-Gated Autonomy

The goal of this architecture is to transform the execution boundary from a state of optimization (the pursuit of an objective function) to a state of authorization (execution within a material, consequence-bearing context). We achieve this by inserting a consequence threshold T into the decision-making loop.

6.1 Defining the Consequence Threshold (T)

The system uses context-dependent thresholds T to distinguish between low-stakes autonomous operations and high-stakes agentic actions. The objective is not to slow AI down, but to ensure that execution authority scales with consequence, not with capability.

Autonomous range ($< T$). Actions where outcomes are reversible, material loss is bounded, and failure does not create systemic risk. Examples include: enterprise IT log parsing and alert triage; generating incident summaries; drafting runbooks; auto-remediating non-critical issues (e.g., restarting stateless services, clearing caches, resizing non-production instances); rotating non-privileged credentials; opening internal tickets and assigning owners. In finance and trading: generating trade ideas and sizing suggestions; running backtests; placing paper trades; executing pre-approved strategies inside strict risk rails (hard max loss, max position, kill-switch) on whitelisted venues/instruments; producing end-of-day reconciliation and P&L reports. For freelancer-deployed agents: summarizing meetings and emails; drafting proposals; scheduling within predefined constraints; generating code in a sandbox; creating marketing assets for review; preparing customer-support drafts that require human send.

Gated range ($\geq T$). Actions involving irreversible outcomes, non-trivial financial commitments, privileged access changes, regulatory exposure, or potential material harm (including reputational harm) where tail risk dominates. Examples include: enterprise IT deployments to production when a change touches authentication, payments, safety controls, audit logging, or data retention; destructive data operations; weakening security posture (e.g., disabling MFA, reducing logging); approving

third-party integrations with customer data access; or publishing from official corporate channels during market-sensitive windows. In finance and trading: changing margin/leverage parameters; adding a new counterparty or beneficiary (first-time wire, new crypto address); modifying risk limits and kill-switch thresholds; or issuing market-moving communications under a firm identity. For freelancer-deployed agents: sending communications that create legal commitment (signing/accepting contracts, issuing refunds or credits beyond policy); exporting sensitive client data; initiating spend beyond pre-set caps (ads budgets, cloud spend); or auto-posting content at scale in ways that create defamation or regulatory exposure.

6.2 The Gating Mechanism: Policy, Capabilities, and Provenance

When an agent proposes an action whose estimated consequence meets or exceeds T, the system transitions from optimization mode (generating candidate plans) to authorization mode (proving permission to execute). This transition is enforced at the execution boundary: the model is permitted to propose actions, but it is not permitted to directly invoke privileged tools.

The agent emits an action proposal P containing the intended tool invocation, parameters, stated objective, and relevant context. A consequence classifier computes a score $C(P)$ and compares it against T. If $C(P) < T$, the proposal may execute under standard runtime constraints (sandboxing, rate limits, and domain-appropriate guardrails). If $C(P) \geq T$, the proposal is non-executable by default and is routed into an authorization workflow.

For gated actions, the system generates an authorization packet: a compact execution memo designed for both human review and later audit. The packet specifies what will happen (tool and parameters), why it is being done now (purpose), what is explicitly out of scope (bounds), the principal failure modes and tail risks, mitigations and rollback plan, the provenance of inputs relied upon, and at least one lower-consequence alternative or staged plan.

Authorization is recorded as a signed event by the accountable bearer (or designated institutional signer). The signed record binds to the specific proposal (or a narrowly defined action class), references the governing policy, constrains scope, and includes a brief justification satisfying the mirroring requirement. Upon validating authorization, the system mints a short-lived capability token encoding permission to execute. Tool adapters enforce the hard precondition: no capability token, no privileged tool call.

Every gated action produces a tamper-evident provenance chain sufficient for incident reconstruction and accountability. At minimum, this record links the original proposal, the score and threshold at the time (including a policy snapshot), the authorization packet presented to the signer, the signed authorization event, the minted capability (or its hash), and the execution result and side effects. If authorization is unavailable, rejected, expired, or ambiguous, the system must fail closed with respect to execution while remaining helpful by proposing staged, lower-consequence alternatives or escalation paths.

7 Related Work

Our proposal intersects with scalable oversight and preference-based alignment, agent/tool execution interfaces, and governance artifacts for auditability. Human preference learning and scalable oversight aim to shape model behavior via feedback and supervision [4, 5]. Constitutional AI similarly constrains model outputs via explicit principles [1]. These lines primarily target internal objectives or behavior.

Tool-use and agentic interaction frameworks formalize how language models call external tools and interleave reasoning with actions [3, 2]. Our contribution is orthogonal: we propose a hard

authorization primitive for high-consequence tool calls, enforced through scoped capabilities and auditable provenance. Governance and documentation artifacts such as model cards emphasize structured reporting and accountability [6]. Safety assurance methodologies for AI-based autonomy provide patterns for argumentation, evidence, and verification [7].

8 Limitations and Failure Modes

Consequence gating shifts the safety problem from "make the model safe" to "make execution accountable and auditable," but it does not eliminate risk. The primary limitations are operational and adversarial.

First, the consequence classifier $C(P)$ can be wrong. Underestimation creates bypass risk; overestimation creates friction and can motivate shadow automation. In practice, $C(P)$ should be conservative for novel actions and incorporate uncertainty, distribution shift, and tail risk; mis-scoring should trigger escalation rather than silent execution.

Second, authorization regimes are vulnerable to approval fatigue and organizational incentives. Even with mirroring requirements, teams may normalize rapid approvals under time pressure, turning the gate into a formality. This motivates quorum approval for specific classes, staged execution for high-impact actions, and auditing that measures approval quality (e.g., repeated boilerplate or anomalous approval rates).

Third, the system introduces key and identity compromise as a critical attack surface. If an attacker can steal or coerce an authorized signer's credentials, the gate can be satisfied fraudulently. Strong authentication, hardware-backed signing, step-up challenges, and out-of-band verification become part of the safety story. For the highest-consequence actions, multi-party authorization reduces single-point failure.

Fourth, consequence gating can fail through policy capture: organizations may set thresholds T too high, define action classes too broadly, or delegate signing authority to under-qualified roles. Because the mechanism is enforcement, not ethics, governance of policy configuration becomes central. The architecture can force explicit decisions, but cannot ensure those decisions are wise.

Finally, consequence gating increases latency for certain actions. This is unavoidable, but can be managed by staged execution: reversible steps proceed automatically while high-consequence "commit" steps require explicit authorization. Emergency overrides can exist, but should be auditable and costly (e.g., higher quorum, tighter logging, post-incident review).

9 Economic and Deployment Implications

AI has driven the marginal cost of cognition downward, while the cost of failure in high-stakes domains remains dominated by rare, high-impact events. In sectors such as finance and critical infrastructure, the binding constraint on deployment is often not model capability but institutional permission: boards, regulators, and operators require legible responsibility and auditable decision lineage.

Consequence-gated autonomy can be understood as a trust multiplier. In safety-critical engineering, robust operating envelopes do not reduce throughput; they enable it by reducing the expected cost of tail failures. Similarly, gating allows reversible automation to run at high velocity while ensuring that irreversible "commit" steps remain coupled to accountable authorization. This can increase organizational willingness to integrate agentic systems into production workflows by making execution authority legible, auditable, and compatible with existing legal and institutional accountability.

10 Conclusion

As AI systems move from advice to execution, the central question becomes one of authority: who is permitted to act, and who bears the consequences when action goes wrong. We argue that this cannot be derived from intelligence alone. Consequence-gated autonomy provides a concrete invariant at the execution boundary: high-consequence actions require explicit authorization by accountable bearers, with cryptographically verifiable provenance. This preserves the benefits of powerful optimization while keeping irreversible execution anchored to auditable accountability.

References

[1] Y. Bai et al. Constitutional AI: Harmlessness from AI Feedback. *arXiv:2212.08073*, 2022.

[2] T. Schick et al. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv:2302.04761*, 2023.

[3] S. Yao et al. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv:2210.03629*, 2022.

[4] P. F. Christiano et al. Deep Reinforcement Learning from Human Preferences. *arXiv:1706.03741*, 2017.

[5] G. Irving, P. Christiano, and D. Amodei. AI Safety via Debate. *arXiv:1805.00899*, 2018.

[6] M. Mitchell et al. Model Cards for Model Reporting. In *Proceedings of FAT**, 2019.

[7] T. Schnitzer et al. A Methodology to Support Safety Assurance for AI-Based Autonomous Systems. *arXiv:2412.14020*, 2024.