

BEYOND HUMAN APPROVAL

When AI Decides, Accountability Must Stay Human

A governance architecture for autonomous agents before delegated action becomes delegated blame

Giovanni Leonardi

July 2025



When AI Decides, Accountability Must Stay Human

Transformation · No. 354
July 2025

"Autonomy changes the speed of action, not the ownership of consequence."

1. The accountability gap is now an operating risk

Giovanni Leonardi

31 July 2025

Autonomous agents are moving from recommendation to action. They can interpret an objective, construct a plan, call tools, exchange information with other agents and complete a sequence of tasks with limited human intervention. The attraction is obvious: work can move at machine speed across boundaries that traditional automation could not cross.

The governance problem is equally clear. Most organisations still allocate accountability through people, roles, committees and legal entities. An agent has none of these attributes. It cannot hold a duty, absorb a sanction, explain a compromise to a customer or repair institutional trust. Yet it can now influence decisions that create financial, operational, regulatory and reputational consequences.

As at 31 July 2025, this is not a distant design question. Early deployments are already exposing a gap between the authority delegated to agents and the controls surrounding their decisions. The gap is not caused by intelligence alone. It arises because organisations are connecting probabilistic systems to real processes without redesigning the ownership of those processes.

Autonomy changes the speed of action, not the ownership of consequence.

The case for change is therefore not that every agent requires constant human approval. That would remove much of the value. The case is that each delegated action needs an accountable human owner, an explicit boundary of authority and evidence sufficient to reconstruct what happened.

2. Why conventional governance is failing

Traditional technology governance assumes a relatively stable object: a system is specified, tested, released and then operated within a defined process. Change is

controlled through projects and releases. Accountability is assigned to system owners, process owners and operational teams.

An autonomous agent behaves differently. It can select an approach at run time, use context that was not present during testing and combine tools in sequences that designers did not enumerate. The same agent can produce different actions from apparently similar situations. It may also depend on a chain of models, data sources, prompts, permissions and external services. Responsibility becomes diffused across the chain just when the action becomes harder to predict.

Four organisational conditions make the gap wider.

- Authority is granted through technical access rather than business mandate. An integration gives the agent permission to update a record, issue a message or trigger a transaction. The organisation then mistakes technical capability for authorised judgement.
- Ownership is fragmented. Technology owns the platform, a business function owns the process, risk owns the policy and a supplier may operate part of the stack. Each party controls an input, but nobody owns the decision as a whole.
- Control evidence is designed for deterministic workflows. Logs show that a tool was called, but not why the agent selected it, what alternatives it considered or which policy boundary shaped the result.
- Escalation is treated as an exception path. In autonomous operation, recognising uncertainty and handing work to a person is a core capability. If escalation is slow, socially discouraged or technically awkward, the agent is pushed to act beyond its competence.

These conditions explain why a technically successful pilot can become a governance failure in production. The agent completes tasks, cycle time falls and users are impressed. Meanwhile, unowned judgement accumulates inside the workflow.

3. What organisations have tried

The first response has often been to extend familiar controls. These approaches are understandable, and each has value, but none is sufficient on its own.

3.1 Human approval for every material action

Approval gates create a visible point of accountability and work well for infrequent, high-impact decisions. They fail when applied indiscriminately. Reviewers become a queue, approval becomes habitual and the person clicking “accept” lacks the time or context to challenge the recommendation. Accountability is performed rather than exercised.

The useful lesson is not to remove humans. It is to place human judgement where consequence, novelty or uncertainty exceeds a defined threshold.

3.2 Policy documents and acceptable-use rules

Written policy establishes intent and can prohibit clearly unacceptable uses. It works when translated into operating constraints: permitted data, prohibited actions, transaction limits, separation of duties and mandatory escalation. It fails when it remains prose that neither the agent nor the surrounding workflow can enforce.

A policy that cannot be observed at the moment of action is guidance, not control.

3.3 Model testing before release

Testing can identify recurring failure modes, unsafe outputs and weak instructions. It is necessary, particularly for high-impact use cases. It cannot prove the safety of every future action because the operating context, data and tool sequence will change. A strong test result is evidence about a bounded evaluation, not a permanent licence to act.

3.4 Central AI committees

A central forum can set standards, compare risks and stop duplicated experimentation. It works when it defines common control requirements and resolves cross-enterprise questions. It fails when every use case must wait for a distant committee or when the committee “approves AI” without owning the business outcome.

The centre should govern the system of delegation. The business should remain accountable for each delegated decision.

3.5 Detailed logging

Logs are indispensable for investigation, assurance and improvement. They are not accountability by themselves. A perfect record of an unowned decision merely allows the organisation to see its failure more clearly.

What has worked across these approaches is the combination of clear ownership, bounded authority, observable decisions and rehearsed intervention. What has failed is control by artefact: the signed checklist, the policy PDF, the approval click or the raw event log standing in for judgement.

4. The strongest case against heavier governance

There is a serious opposing argument. Autonomous agents are valuable precisely because they reduce coordination cost. If organisations impose boards, approvals and documentation on each action, they will reproduce the bureaucracy that agents were meant to remove. Existing accountability already sits with process owners and executives; adding an “AI governance” layer may create another fragmented function. Excessive caution may also drive experimentation into unofficial channels where control is weaker.

This argument is right about the danger of governance theatre. It is wrong to conclude that the answer is minimal governance. The choice is not between autonomy and control. It is between controls designed around the risk of the decision and controls inherited from the delivery of conventional systems.

Good governance should make low-risk action faster. It should pre-authorise routine decisions inside narrow boundaries, reserve human attention for material exceptions and make ownership unambiguous. The objective is not more approval. It is better delegation.

5. An accountability architecture for autonomous agents

A practical architecture has six connected elements.

5.1.1. Name the accountable decision owner

Every agent-enabled process needs one named role accountable for its outcomes. That owner must have authority to define acceptable performance, approve the scope of delegation, suspend operation and fund corrective action.

The owner is not necessarily the developer, platform team or day-to-day user. It is the person who would already answer for the business decision if no agent existed. Delegating execution does not delegate accountability.

5.2.2. Define an authority envelope

The agent should operate inside a written and technically enforced envelope covering:

- the decisions it may make;
- the data it may access and retain;
- the tools and counterparties it may use;
- financial, legal and operational limits;
- conditions that require escalation;
- actions that are always prohibited.

The envelope should be specific enough to test. “Act in the customer’s best interest” is a principle. “Do not alter payment terms, disclose protected information or make an irreversible commitment without approval” is an operable boundary.

5.3.3. Tier decisions by consequence and reversibility

Not every action deserves the same control. A useful tiering model considers impact, reversibility, detectability, time sensitivity and the vulnerability of affected people.

1. Observe: the agent gathers or summarises information but cannot change the operational state.
2. Recommend: the agent proposes an action and a person decides.
3. Act with notification: the agent takes a reversible, bounded action and creates prompt visibility.
4. Act autonomously: the agent completes pre-authorised, low-variance work within enforced limits.

5. Escalate or stop: the agent encounters uncertainty, conflict, novelty or material consequence outside its envelope.

The tier belongs to the decision, not to the agent as a whole. One agent may act autonomously when classifying routine correspondence and only recommend when changing a customer commitment.

5.4 4. Preserve decision evidence

Assurance requires more than a transcript. For each material action, the organisation should be able to reconstruct:

- the objective and instruction in force;
- the relevant input data and its provenance;
- the model, tools and permissions used;
- the policy checks and authority limits applied;
- the action taken, its confidence or uncertainty indicators and any escalation;
- the human interventions and subsequent outcome.

Evidence should be proportionate. The purpose is to support review, challenge and learning, not to create an unreadable archive.

5.5 5. Monitor outcomes, not only behaviour

Technical monitoring asks whether the agent ran, produced errors or breached a rule. Governance monitoring asks whether the delegated process remains acceptable.

Measures should include erroneous actions, near misses, escalation quality, reversals, complaints, unequal effects, control overrides and the time taken to detect and correct harm. A process can remain technically available while its decisions deteriorate.

Thresholds should trigger predefined responses: narrow the authority envelope, increase sampling, return a decision tier to recommendation, pause a tool or suspend the use case.

5.6 6. Rehearse intervention

An agent that cannot be stopped safely is not governed. Teams need a tested way to identify affected actions, revoke permissions, preserve evidence, notify decision owners and recover the process. Intervention should be possible at the level of a tool, decision type or use case rather than requiring a platform-wide shutdown.

Rehearsal reveals practical weaknesses that policy review misses: unclear contacts, incomplete logs, dependent automations and manual workarounds that cannot absorb the returning volume.

6. A composite operating example

Consider a service organisation using an agent to resolve routine supplier discrepancies. The pilot initially allowed the agent to read invoices, compare contract terms, correspond with suppliers and update the payment workflow. Accuracy appeared high, but accountability was divided: procurement owned supplier relationships, finance owned payment controls and technology owned the agent.

A material discrepancy exposed the weakness. The agent interpreted an ambiguous tolerance rule, accepted a credit against a future invoice and updated the workflow. No single action breached a technical permission, yet the combined decision created an unauthorised commercial commitment.

The response was not to abandon the agent. The organisation named the head of procurement operations as decision owner, separated comparison from commitment, limited autonomous adjustments to reversible amounts below a defined threshold and required escalation when terms were ambiguous. Finance retained control of payment release. The evidence record linked the instruction, contract clause, proposed resolution and tool actions. A weekly sample examined outcomes rather than wording quality.

Cycle time remained lower than the manual process, while exceptions reached people with the authority and context to decide. The improvement came from redesigning delegation, not from making the agent more eloquent.

7. The management decisions required now

Leaders do not need to predict the final form of autonomous technology. They need to make five decisions before authority spreads invisibly through pilots and integrations.

1. Identify every use case in which an agent can change an operational state, communicate externally or commit resources.
2. Assign a business decision owner and document the authority envelope for each use case.
3. Match decision tiers to consequence and reversibility, reserving human attention for genuine judgement.
4. Require decision evidence and outcome monitoring before scaling autonomy.
5. Test suspension, investigation and recovery while the use case is still small.

The central question is not whether an agent made the decision alone. The question is whether the organisation deliberately delegated that decision, bounded the authority and retained a human owner capable of answering for the result.

Autonomous agents can remove friction from work. They cannot remove institutional responsibility. Organisations that understand this will scale autonomy with confidence. Those that do not will discover, after the first serious failure, that a chain of sophisticated systems still ends with a person being asked: who allowed this to happen?