Judgment, Delegation, Termination, Verification:
Toward a Minimal Accountability Grammar for Human-AI Agent Decision Chains

Yuqiang Wang
Cognitive Emergence
yuqiang@humanjudgment.org

Ruoxi Ran
Zhejiang University
ranruoxi@zju.edu.cn

Abstract

As AI agents are increasingly deployed in organizational and decision-making settings, responsibility becomes harder to reconstruct in practice. In human-agent workflows and agent-to-agent task chains, actions can propagate quickly, authority can shift quietly, and the connection back to human judgment may be thin or broken. Existing accountability practice usually relies on two familiar tools: system logs, which record machine behavior, and governance frameworks, which specify what human oversight should occur. Both matter, but neither provides a stable event structure for representing the accountability-relevant events that connect human roles, agent actions, and auditable records.

This paper proposes a candidate minimal action grammar for modeling accountability-relevant events in AI agent decision chains. The framework uses four primitives: Judgment, Delegation, Termination, and Verification. We argue that these primitives capture four distinct kinds of change in an accountability chain: decision recording, authority transfer, responsibility-chain closure, and record-chain validation. On that basis, we advance two claims as testable propositions rather than settled conclusions: expressive adequacy and conditional minimality. We also compare the framework with alternative representation schemes and define an evaluation protocol for coverage, minimality, accountability utility, and independent verification.

The paper is conceptual, formal, and methodological in character. Its aim is not to settle legal liability or ethical correctness, but to provide a clearer and more portable representation layer for accountability in human-agent and agent-to-agent environments.

Keywords: AI agent accountability; human oversight; multi-agent systems; verifiable records; human-AI collaboration; audit grammar

1 Introduction

AI agents are no longer confined to suggesting, scoring, or ranking. More and more, they act. They call tools, spawn subtasks, hand work to other agents, negotiate with services, execute workflows, and carry out tasks that once sat squarely in human hands. Sometimes a person reviews the result. Sometimes a person only reviews a sample. Sometimes one agent passes work to another before any human looks in at all.

That shift creates a practical headache, not just a philosophical puzzle. Agents do things that matter. People still answer for the outcome. So where, exactly, is the record of human responsibility? Where do we see the moment someone exercised judgment, transferred authority, halted a chain, or confirmed that the relevant record still holds together? In many systems, we do not see it clearly at all.

The tools we already have are useful, but they leave a conspicuous gap in the middle. System logs tell us what happened inside the machine layer: prompts, outputs, tool calls, triggers, execution traces, handoffs. Governance frameworks tell us what organizations are expected to do: maintain human oversight, preserve intervention capacity, document high-risk decisions. What sits between those two layers is much thinner than it should be. We often have agent events on one side and institutional expectations on the other, but no stable event structure for the decision, control, and record-layer events that make accountability legible. Without that layer, reconstruction turns into a scavenger hunt through tickets, trace logs, chat threads, policy language, and scattered timestamps.

This paper argues for a thinner and more explicit representation layer. More specifically, it proposes a *candidate* accountability grammar for AI agent decision chains built around four primitives: Judgment, Delegation, Termination, and Verification. Judgment records a decision, including who made it, what it concerned, when it was made, and in what context. Delegation records a transfer of authority with an explicit scope and expiry condition. Termination records the closing of an active responsibility chain. Verification records whether the event record and its predecessor chain remain structurally valid.

The claim here is deliberately narrower than "this solves AI accountability." It does not. The wager is smaller and, we think, more useful. If we want to reconstruct responsibility in human-agent and agent-agent systems, what kinds of decision, control, and record-validation events do we need to represent as first-class events? And how small can that vocabulary be before it starts dropping information that matters?

We do not present the framework as settled doctrine. It is a conceptual, formal, and methodological proposal. It offers a candidate representation layer, makes its assumptions visible, and names the places where it could fail. That is important. A framework like this should not survive because it sounds neat. It should survive only if other researchers can push on it and still find it useful.

This paper makes four contributions. It introduces a four-primitive framework for representing accountability-relevant events in AI agent decision chains. It develops the causal semantics of those primitives by placing them on distinct layers of an accountability state model. It turns expressive adequacy and conditional minimality into claims that can actually be challenged, rather than leaving them as elegant intuitions. And it sketches a multi-method evaluation agenda for testing the framework's coverage, parsimony, practical utility, and verifiability.

2 The Accountability Gap in AI Decision Chains

The phrase "accountability gap" gets used in a lot of ways. Here we use it in a narrow sense. We do not mean that human responsibility disappears when AI agents are involved. We mean that responsibility becomes hard to locate in the record.

In older organizational settings, the line from action to responsibility was rarely perfect, but it was often readable enough. A person gathered information, weighed options, made a call, acted, and answered for the consequences. AI agents change the shape of that chain. Humans still set goals, define guardrails, approve deployments, review samples, or retain final authority. But agents now perform many intermediate steps that carry real weight: they rank, infer, prioritize, route, call tools, delegate subtasks to other agents, and sometimes execute. By the time an outcome emerges, the causal path may be technically rich and institutionally murky.

System logs help, of course. They can tell us when an agent emitted an output, when a workflow advanced, when a tool call fired, or when a downstream agent took over. Still, ask a slightly different set of questions and the logs often go quiet. Who held authority over this matter at that moment? Who actually made the judgment that counted? Which agent was acting under delegated scope, and when did that scope end? Who confirmed that the relevant chain still held together? Those are not machine-state questions. They are accountability questions.

Governance frameworks approach the problem from the opposite direction. They state what institutions should do. They require oversight, documentation, intervention capability, review procedures, and retention periods. That language matters because it anchors accountability normatively. Yet norms do not automatically turn into evidence. A policy saying "a human must review high-risk agent outputs" is not the same thing as a record showing which human reviewed which output, when they did so, under what authority, and with what consequence.

That is the gap we care about here. It is not only a legal gap or an ethical gap. It is also a representational one. The human part of the chain is often under-specified in the event structure itself. When that happens, institutions end up narrating accountability after the fact instead of recording it as they go.

3 Related Work and Positioning

A number of research traditions already circle this terrain, though they tend to stop short of the specific representational problem at issue here.

Work on human oversight, human-in-the-loop systems, and algorithm-in-the-loop decision making has long insisted that meaningful human involvement matters in consequential settings [3, 7]. That literature is persuasive on an essential point: it matters whether a person is in the loop, and it matters where they sit in the loop. What it usually does not provide is a compact event-level grammar for representing interventions across human-agent workflows and agent-to-agent chains in a way that travels across domains.

The literature on algorithmic accountability and auditing pushes on adjacent concerns. It looks at traceability, auditability, institutional responsibility, internal review, external challenge, and the limits of transparency discourse [1, 2, 5, 8]. These scholars have been especially sharp about responsibility drift and audit theater. They show, convincingly, that institutions can produce the appearance of accountability without much substance. Still, most of that literature studies the processes and institutions of accountability rather than asking what the smallest stable vocabulary of accountability-relevant events might be.

There is also a well-developed technical tradition around provenance, audit trails, append-only structures, and verifiable logging [4, 6, 10]. This work gives us strong tools for tamper detection, event reconstruction, and multi-party verification. It solves an important downstream problem: how to trust the integrity of the chain. But these mechanisms are, in a sense, happily indifferent to content. They can help ensure that something was logged without telling us which decisions, control changes, or record-layer validations most deserve to be logged as accountability events.

Sociotechnical and institutional research adds another piece. It reminds us that fairness, responsibility, and oversight are not properties that live inside the model alone. They are embedded in workflows, role hierarchies, organizational incentives, and regulatory settings [9]. Organization theory, especially work on authority and delegation, also matters here [11]. Those traditions help us see why judging, authorizing, escalating, or signing off are not just workflow niceties. They are institutional acts. Even so, those insights do not typically arrive in a form that a technical system can adopt as a small, operational semantics.

So where does this paper sit? Not above these literatures, and not in place of them. If anything, it sits awkwardly between them. We are after a missing middle layer: a representation of accountability-relevant events that is thinner than institutional theory, more semantic than system logging, and more operational than governance principle alone, especially in environments where humans supervise agents and agents hand work to other agents.

We also want to avoid the old academic trick of renaming familiar things and pretending that the new label does all the work. The four-primitive framework only earns attention if it helps separate what nearby alternatives often blur together. We return to those alternatives in Section 6.

4 A Minimal Accountability Grammar

We propose that accountability-relevant events in AI agent decision chains can be represented using four primitives: Judgment, Delegation, Termination, and Verification.

Judgment records a decision. At minimum, it should capture who made the decision, what the decision concerned, when it was made, and the immediate context in which it was made. In many domains, that record may also include the decision outcome itself, the artifact reviewed, and any applicable rationale or policy hook.

Delegation records a bounded transfer of authority. A human actor grants another actor—human or artificial—the right to decide or act within a specified scope, and that grant should carry an expiry condition, whether time-based, event-based, or revocation-based. Delegation does not make responsibility evaporate. It redistributes it under stated limits.

Termination records the closing of a responsibility chain. It captures the act that ends an authorization path, halts a process, revokes a live delegation, or otherwise closes the chain through which responsibility was flowing.

Verification records the validation of record integrity and predecessor chains. Verification does not reopen the underlying decision on the merits. Its job is narrower: to confirm that the event record is structurally intact and that it remains linked to a valid chain of prior records.

This vocabulary follows a minimality principle. We keep only those action types that seem to have independent significance for accountability analysis. If a behavior can already be represented by an existing primitive, it should not be promoted to foundational status just because institutions happen to use a different local verb. "Approve," for instance, is often better treated as a judgment outcome. "Escalate" usually combines ending one handling path with assigning authority elsewhere. "Acknowledge receipt," when it matters for accountability, often works as a lightweight form of verification.

That said, minimality can turn sloppy if the category boundaries stretch too far. We therefore treat an intervention as accountability-relevant only when it does at least one of four things: it changes the normative outcome of a matter; it changes the authority boundary around that matter or class of matters; it changes whether a process or authorization remains active; or it changes whether a record or its predecessor chain remains valid. If none of those happens—say a human supervisor merely reads information, or one agent emits an intermediate note without institutional effect—then the act sits outside the event space we are trying to model.

5 The Causal Semantics of the Four Primitives

The four primitives are not best understood as a checklist or a fixed workflow. They behave more like different kinds of state transition inside the same accountability chain.

It helps to make that concrete. We write the accountability state at time t as

$$S_t = \langle A_t, D_t, P_t, E_t \rangle$$

where A_t is the authority state, D_t is the decision state, P_t is the process-validity state, and E_t is the record-validity state. In plain language: who may act, what has been decided, whether the relevant path is still live, and whether the record chain remains intact and verifiable.

This is a thin model on purpose. It does not try to absorb the whole institution. It only asks: what changed, in accountability terms?

Judgment primarily updates D_t. It records that a decision has been made and anchors that decision to an actor, object, time, and context. Delegation primarily updates A_t, because it changes who may decide or act within a bounded scope and for how long. Termination primarily updates P_t, because it revokes or closes an active path of authorization or execution. Verification primarily updates E_t, because it changes whether a record and its predecessors can still be treated as structurally intact and chain-valid.

Once we look at the framework this way, a few things become easier to say cleanly. A single real-world action can be composite. A senior reviewer may both record a new Judgment and issue a Termination on an earlier delegated path. That does not break the model; it just means one observed action may need more than one event record.

The model also shows why institutional language can be deceptive. Organizations talk about sign-off, escalation, override, clearance, attestation, and authorization. Those labels matter in local

workflow. Still, they do not always mark distinct foundational actions. Sometimes they are just different names for the same underlying state change. Sometimes they are bundles.

That point matters because the grammar is not trying to mirror every institutional verb one by one. It is trying to isolate the smallest set of state-changing actions that accountability analysis keeps circling back to.

The limits of the model follow from the same choice. It does not encode the full content of a decision, the whole social meaning of a role, or the legality of an action. Those questions remain important. They just sit in surrounding institutional context, domain-specific metadata, and downstream normative analysis. The grammar operates at a thinner layer. It asks what kind of accountability-relevant change occurred, not whether the world ought to endorse it.

Seen this way, the primitives are not loose synonyms. They are different causal operations on different layers of the chain. Delegation often creates the authority basis for a later Judgment record. Judgment may trigger Termination when a process must be stopped. Verification can attach to Judgment, Delegation, or Termination and test whether the resulting record chain remains intact. The relation is causal, then, not merely temporal. And it is certainly not a universal sequence. Real institutions are messier than that.

This is why we treat the four primitives as semantically orthogonal. Judgment is not a special case of Delegation. Termination is not merely a negative decision. Verification is not just "deciding again." Collapse those distinctions and the record becomes flatter precisely where accountability needs structure.

6 Theoretical Claims, Scope, and Criteria for Minimality

We advance two core research claims, but we present them as testable and revisable propositions rather than as theorems already proven.

First, the four primitives have **expressive adequacy**. By this we mean that, within the target scope of the paper, any accountability-relevant event that changes authority state, decision state, process-validity state, or record-validity state should be representable as one or more Judgment, Delegation, Termination, or Verification events. Purely informational browsing, informal conversation, or comments without institutional effect do not fall within the target event space.

Second, the four primitives have **conditional minimality**. This does not mean that no alternative representation could ever exist under any legal system, theoretical framework, or granularity of analysis. It means only that, under the representational goal and state partition used here, the four primitives satisfy three criteria.

The first is a **necessity criterion**. If one primitive is removed, a class of events with independent accountability significance becomes inexpressible or can be represented only through a lossy approximation. Remove Judgment, and a decision record no longer has an independent representation. Remove Delegation, and authority transfer becomes opaque. Remove Termination, and continuing authorization cannot be cleanly distinguished from a closed responsibility chain. Remove Verification, and structurally valid records cannot be cleanly distinguished from broken or unvalidated

chains.

The second is a **non-derivability criterion**. If one primitive can be derived losslessly from the others, it should not remain primitive. Our claim is that the four primitives correspond to distinct types of state update and therefore cannot be reduced to one another without distortion. For example, reducing Termination to a negative form of Judgment wrongly folds chain closure into decision recording. Reducing Verification to another Judgment event wrongly folds record-integrity validation into substantive decision content.

The third is a **redundancy criterion for candidate extensions**. If a proposed fifth primitive is only an outcome label, a context-specific variant, or a recurrent combination of existing primitives, it should not count as foundational. We treat Approve, Escalate, Override, Monitor, Acknowledge, and Sign-off as typical candidates. Approve is usually a Judgment outcome. Escalate often means ending local handling and delegating authority upward. Override often looks like a new Judgment record that conflicts with a prior one, sometimes paired with Termination. Monitor enters the accountability record only when linked to a recorded decision or a later verification step. Acknowledge and Sign-off, when they function as chain-validation acts, usually belong under Verification.

In this sense, minimality is conditional. It is limited by scope and by evaluative criteria. Its validity ultimately depends on ablation studies, candidate-primitive comparisons, and scenario coding results. If future work consistently finds a stable class of accountability events in human-agent or agent-agent settings that cannot be represented by the four primitives and also cannot be reasonably decomposed, then the minimality claim should be revised.

We therefore treat minimality not as a matter of intuition but as a comparative claim. The four-primitive framework must perform at least as well as, and ideally better than, several plausible alternatives. These include: (1) a two-primitive model built only from judgment and delegation; (2) a three-primitive model that folds verification into judgment; (3) workflow-style approval labels such as approve, escalate, reject, and sign-off; and (4) non-semantic combinations of system logs and governance checklists. If those alternatives match or outperform the four-primitive framework in coverage, interpretability, or responsibility reconstruction, then the claim of conditional minimality fails.

We also make the boundaries of the framework explicit. It does not decide whether a judgment was correct. It does not decide whether an authorization was justified. It does not settle legal liability. And it does not try to represent the full internal behavior of AI agents. It is a framework for representing events that should count as accountability-relevant records, not a framework for deciding whether those events were normatively good.

6.1 Why Not More or Fewer Primitives?

One reasonable objection is that four primitives may still be too many. Why not use a thinner scheme such as "decision" and "delegation," then infer the rest from context? The short answer is that inference is exactly what breaks down under audit pressure. If Termination is not recorded distinctly, reviewers must infer from missing actions or changed outcomes that a responsibility chain was closed. If Verification is not recorded distinctly, they must infer from document presence

or workflow completion that the record chain is intact. In high-stakes systems, those inferences are often fragile, contested, or simply wrong.

The opposite objection is that four primitives may be too few. Why not include approval, override, escalation, monitoring, attestation, acknowledgment, and escalation-to-human as separate foundational categories? Here the issue is different. A larger vocabulary can certainly describe more surface variation, but a foundational grammar should distinguish between recurrent institutional labels and genuinely independent state transitions. "Override," for instance, often reduces to a new Judgment record and, sometimes, a Termination event affecting the prior path. "Escalate" often looks primitive in workflow language, but structurally it is usually a composite of ending local handling and reassigning authority upward.

That is why the paper frames minimality as conditional instead of absolute. We are not claiming that all communities must always speak in exactly four categories. We are claiming something narrower: if the representational goal is to capture accountability-relevant state changes in AI agent decision chains, then four categories appear to be a strong candidate for the smallest stable basis.

6.2 Comparison Targets

Because the argument is comparative, the framework should be read against plausible alternatives rather than in isolation. We highlight four comparison targets.

The first is a **decision-plus-delegation model**. This is the most compact alternative. It can capture many ordinary workflow events, but it tends to blur together stopping a process and merely deciding against an outcome. It also struggles to express the difference between an internal action and an action that has become institutionally confirmable.

The second is a **three-primitive model that folds verification into judgment**. This alternative is attractive when institutions treat sign-off as the final decision itself. But it becomes problematic once the same system needs to distinguish substantive review from evidentiary confirmation. In many regulated environments, those are not the same act and may not even be performed by the same person.

The third is a **workflow-label model** built from ordinary business language such as approve, reject, escalate, sign-off, acknowledge, and override. This model often matches organizational practice well at the local level. The downside is that it can be hard to compare across sectors, because the same label may carry different accountability meanings in different workflows, while different labels may encode the same underlying state change.

The fourth is a **logs-plus-checklists approach**. This is common in real organizations: system logs capture machine execution, and governance templates capture institutional requirements. The strength of this approach is pragmatic familiarity. The weakness is that responsibility reconstruction still requires stitching together heterogeneous artifacts and inferring accountability-relevant events from indirect evidence.

The four-primitive framework should ultimately be judged by whether it improves on these alternatives in ways that matter: cleaner event typing, fewer unresolved "other" categories, better

Scheme	Authority transfer	Process revocation	Chain validation	Cross-domain comparability	Typical weakness
Four-primitive framework	Explicit	Explicit	Explicit	High	Requires disciplined event typing and coding guidance
Decision + delegation model	Partial	Weak / inferred	Weak / inferred	Medium	Collapses stopping and confirming into context-dependent interpretation
Three-primitive model (verification folded into judgment)	Explicit	Partial	Partial	Medium	Blurs decision recording with chain validation
Workflow-label scheme (approve, escalate, sign-off, etc.)	Variable	Variable	Variable	Low to medium	Tracks local practice well but travels poorly across domains
Logs + governance checklists	Indirect	Indirect	Indirect	Low	Reconstruction depends on stitching heterogeneous artifacts and inference

Table 1: Comparison between the proposed four-primitive framework and alternative representation schemes.

reconstruction performance, and clearer separation between authority, decision, process validity, and record-chain validity.

7 Research Questions

To turn these theoretical claims into a testable research agenda, we ask four questions.

- **RQ1:** Within the scope defined in this paper, are the four primitives sufficient to cover most accountability-relevant events across human-agent and agent-agent decision chains?
- **RQ2:** Do the four primitives satisfy conditional minimality, such that removing any one primitive makes important interventions inexpressible, while common candidate extensions turn out to be redundant?
- **RQ3:** Compared with system logs alone or governance documents plus logs, do records based on the four primitives significantly improve third-party reconstruction of authority, intervention, and oversight?
- **RQ4:** Can records expressed through the four primitives be independently validated across implementations and support interoperable auditing?

These four questions correspond to expressive adequacy, minimality, accountability utility, and interoperability with independent verification.

8 Evaluation Agenda and Operationalization

This section does not report completed results. Its purpose is narrower: to translate the paper's claims into a concrete and refutable evaluation agenda. The thresholds below are provisional, but naming them now is better than leaving all criteria implicit.

8.1 Expressive Adequacy

The goal of expressive adequacy testing is to examine whether the four primitives can cover accountability-relevant events in real or high-fidelity scenarios. A practical way to do this is to build a scenario corpus drawn from settings such as agent-assisted research, multi-agent task execution, agent-mediated customer support, agent-based procurement, and human-supervised agent operations. Each scenario would be decomposed into key event points, and multiple independent coders would classify each event as Judgment, Delegation, Termination, Verification, or Other.

The main metrics would be coverage rate, residual Other rate, and intercoder agreement, for example Cohen's kappa or Krippendorff's alpha. As a first benchmark, the framework would look provisionally well-supported if coverage exceeds 90%, residual Other remains below 10%, intercoder agreement reaches at least 0.75, and no single recurring uncoded event type accounts for more than 5% of the corpus. On the other side, the framework would be in trouble if residual Other stays high, if agreement remains below about 0.67 after coder training, or if one stable and substantively important class of events keeps escaping the scheme.

For FAccT or JRC in particular, a stronger version of this study would introduce alternative coding ontologies as baselines, such as workflow approval labels or simplified two-primitive and three-primitive schemes. Expressive adequacy would then mean not only that the four-primitive framework works, but that it works more robustly than nearby alternatives.

8.2 Minimality

Testing minimality requires both ablation analysis and candidate-extension analysis. In an ablation analysis, we remove one primitive at a time—Judgment, Delegation, Termination, or Verification—and observe which scenarios or event structures become inexpressible, or can be expressed only by losing important state information. A useful working threshold is that each removed primitive should make at least 10% of the corpus lossy or inexpressible, and that those failures should not cluster only in marginal edge cases. If an ablated scheme performs nearly as well as the full four-primitive scheme, then the necessity claim weakens quickly.

Candidate-extension analysis asks whether common "fifth primitive" proposals really introduce a new independent state transition. Candidates might include Approve, Escalate, Override, Monitor, Acknowledge, and Sign-off. The key question is whether a candidate marks a stable kind of

accountability state change that the existing primitives cannot capture. If a candidate primitive consistently reduces residual Other by more than 5 percentage points, achieves coder agreement above 0.70, and captures a semantically coherent class that is not merely a bundle of existing primitives, the framework should be revised rather than defended by stubbornness.

To make this analysis persuasive, later empirical work should include a failure-case library showing which scenarios fail under each ablation and whether those failures cluster in particular agent interaction patterns or institutional contexts.

8.3 Accountability Utility

Accountability utility is not about whether the framework looks elegant on paper. It is about whether it helps third parties reconstruct responsibility chains more effectively. One way to test this is through a comparative study with three conditions: system logs only, governance documents plus logs, and accountability records structured with the four primitives. Participants might include auditors, compliance professionals, trained graduate students, or other reviewers with relevant expertise in agent operations.

Participants would answer a fixed set of questions for the same cases: who held what authority at a given time, whether a human or an agent actually exercised judgment, whether override, revocation, or termination occurred, and whether the available records are enough to show that the required supervisory control took place. The core metrics would be reconstruction accuracy, time to reconstruct, confidence scores, and agreement across reviewers. As a rough benchmark, the utility claim would look credible if the four-primitive condition improves reconstruction accuracy by at least 15 percentage points or reduces median reconstruction time by about 20%, without lowering reviewer agreement. Smaller gains may still matter, of course, but then the argument has to become more contextual and less declarative.

For normatively oriented follow-up work, a qualitative layer could ask participants when they regard a record as sufficient to support responsibility attribution or meaningful human oversight. That would connect structured representation to normative judgment rather than leaving the analysis at task efficiency alone.

8.4 Interoperability and Verifiability

Interoperability and verifiability concern whether records generated by different organizations or systems can be independently validated and linked into an auditable chain. A natural next step would be to build several reference implementations, or simulated outputs from different implementations, and then test them with independent validators that check signatures, predecessor references, and structural integrity. Fault injection could include record tampering, missing predecessor links, invalid scope definitions, and signature mismatches.

The main metrics would be verification success rate, tamper-detection rate, and compatibility across implementations. A sensible early benchmark would require valid records to verify successfully more than 95% of the time, deliberate tampering to be detected more than 99% of the time, and cross-

implementation compatibility to remain above 90%. If compatibility drops below that, or if even a small share of critical tampering faults slip through undetected, the implementation story starts looking shaky. Importantly, this part of the evaluation concerns record integrity, not the truth of the underlying event. A record can be structurally valid and still describe a dishonest or mistaken event.

One point matters especially here: verifiability and truth should remain separate. We therefore interpret Verification as a test of record integrity and chain validity, not as a final certification of factual truth. That keeps the framework from promising too much and makes it easier to connect with existing audit and compliance practice.

8.5 What a First Empirical Package Could Look Like

A reasonable pilot would not need to be massive. It could start with a corpus of 80 to 120 scenario units drawn from five settings: agent-assisted research, multi-agent task execution, agent-mediated customer support, agent-based procurement, and human-supervised autonomous operations. Each unit would describe a bounded decision episode with enough context to identify accountability-relevant events and authority shifts.

Three to five coders could annotate those units under multiple schemes: the four-primitive scheme, a simplified alternative such as decision-plus-delegation, and a workflow-label scheme. The resulting comparison would already allow the paper to report early evidence on coverage, residual "Other" rates, and coding agreement.

In parallel, a small reconstruction study could recruit trained graduate students, compliance professionals, or auditors-in-training. Participants would receive the same cases in different representational formats and answer responsibility-tracing questions under time constraints. Even a modest study of this kind could reveal whether the four-primitive format reduces ambiguity in practice. A lightweight validator prototype could then complete the package by demonstrating that the framework is not only conceptually tidy but also implementable in a minimally credible way.

None of this would close the case. But it would change the status of the manuscript. Instead of saying "here is a framework and a future agenda," the paper could say "here is a framework, here is how to test it, and here is initial evidence that the tests are worth taking seriously." By the same token, the framework should be revised, or potentially abandoned, if later work finds a stable new primitive category, fails to show improved responsibility reconstruction, or cannot achieve consistent independent validation across implementations.

9 Illustrative Cases

To show how the four primitives operate in agent environments, we sketch three simplified cases.

In a human-supervised research setting, a planning agent proposes a synthesis of sources and a recommended answer path. The human operator reviews the proposal, accepts part of it, and revises the final direction before execution. That intervention becomes a Judgment record because

it captures who decided, what was decided, when, and in what task context. A compliance or quality role later checks that the record remains intact and correctly linked to the prior chain of agent outputs. That is a Verification event.

In a multi-agent execution setting, a lead agent is granted authority to distribute subtasks to specialized retrieval, coding, and reporting agents within a bounded scope and validity window. That is a Delegation event. Later, the supervising human detects drift or unsafe tool use and revokes that authority, restoring direct approval before any further delegation occurs. That is Termination. If an auditor later confirms that the revocation record is intact and correctly chained before a disputed downstream action, that becomes Verification.

In a multi-agent moderation setting, one agent ranks incoming reports and another recommends account actions. A trust-and-safety analyst reviews a proposed suspension and decides that the account should receive only a warning because the agent chain overgeneralized from prior behavior. That is Judgment. Earlier, a policy lead may have delegated low-severity enforcement authority to the moderation agent chain under defined conditions. That is Delegation. If a spike in false positives leads the lead to suspend automated enforcement until thresholds are recalibrated, that is Termination. If internal audit later confirms that the suspension record was preserved intact and linked correctly across the chain, that is Verification.

These cases do not exhaust the space. Variants in procurement, hiring, and other agent-mediated settings fit the same pattern: authority can be granted with scope, decisions can be recorded by either a human or an authorized agent, active chains can be closed, and the resulting records can be structurally validated. The point is not to catalog every workflow. It is to show that the same four-event grammar recurs across differently shaped agent environments.

9.1 A Small Public Governance Anchor

The cases above are illustrative and admittedly stylized. It helps, then, to check the framework against at least one public institutional text. The NIST AI Risk Management Framework 1.0 is useful for that purpose [12]. It is not an event schema, and it was not written to endorse Judgment, Delegation, Termination, and Verification. That is precisely why it is a useful test.

Read through the AI RMF with this paper's lens and a pattern starts to show. Governance responsibilities and assigned decision rights sit close to Delegation. Human review, contestability, and risk response decisions lean toward Judgment. Incident response, rollback, and suspension pathways look a good deal like Termination. Documentation, traceability, and chain validation push toward Verification. The mapping is not perfect. It should not be. But it is close enough to suggest that the four-primitive grammar can read a real governance framework without too much forcing.

That does not validate the theory on its own. It does something smaller. It shows that the framework is not limited to author-invented examples and that its categories can travel, at least tentatively, into a widely used public risk-management document.

10 Discussion

The proposed grammar matters, we think, for three practical reasons. It gives "human oversight" a more operational shape. It resists collapsing accountability into one local agent workflow, since research agents, support agents, coding agents, moderation chains, and procurement agents all ask some version of the same structural questions: who had authority, who exercised judgment, who could halt the process, and who confirmed the record. And it speaks to protocol design, where cryptographic integrity is not enough unless we also know what belongs in the record.

There is also a design-time use here. Teams building or deploying AI agents could ask, before rollout: where exactly is authority being delegated? Who still retains the power to terminate? Which decisions still require human judgment? Which events would need verification if the system were later challenged? Framed that way, the grammar becomes a design aid as much as an audit artifact.

Still, better records do not make weak institutions strong. A system can be beautifully documented in Judgment/Delegation/Termination/Verification terms and remain unsafe, unjust, or unlawful. Documentation is not redemption. What it can offer, more modestly, is a sharper picture of where responsibility was exercised, delegated, ended, or structurally validated.

11 Limitations

The limitations are real and not especially subtle.

The paper's main contribution is still conceptual modeling and research design. Expressive adequacy, conditional minimality, and accountability utility all require empirical testing. Until that testing exists, the framework remains a strong candidate rather than an established standard.

Even a well-built multi-domain scenario corpus will not fully capture the ambiguity and conflict of real institutions. Workflows drift. Roles overlap. Records are incomplete. People act through unofficial channels. Some of the hardest accountability problems emerge precisely in those messy edges.

Boundary cases also pose trouble. "Escalation," for example, may combine termination of local handling with delegation elsewhere. In other words, classification itself sometimes becomes interpretive work. That is not fatal, but it does mean the framework will live or die partly on the quality of its coding manual and the discipline of its users.

There is also the perennial problem of performative compliance. A verifiable record shows that something was recorded in a structurally valid way. It does not guarantee that the underlying action was honest, competent, or normatively sound. No grammar can solve that on its own.

Finally, the institutional meaning of verification is likely to vary. In one setting, Verification may involve predecessor-chain checking. In another, it may involve documentary completeness. In another still, it may involve external attestation. So this fourth primitive should be read as a record-layer operation, even when a human or institutional role performs it. The framework probably needs contextual extensions if it is ever used in earnest.

At present, the paper also lacks systematic evidence from real deployments, incident reports, or compliance archives. So any claim about cross-sector portability should be taken as a working hypothesis. That is not a flaw to hide. It is the current state of the argument.

12 Conclusion

As AI agents increasingly act and decide on behalf of humans, the central accountability question shifts. It is no longer only what the agent did. It is also when, how, and under what authority people entered the chain, and how agents handed work to other agents along the way.

Existing system logs capture machine behavior. Governance frameworks capture institutional expectations. What they often miss is a stable representation of accountability-relevant events. This paper has proposed a candidate minimal grammar for that missing layer, built from four primitives: Judgment, Delegation, Termination, and Verification.

We have argued that these primitives track four different kinds of accountability state change: decision recording, authority allocation, process validity, and record-chain validity. We have also argued, more cautiously than triumphantly, that this four-part grammar may be both expressively strong and conditionally minimal. Whether that argument holds will depend on comparative testing, ablation, reconstruction studies, and implementation work. It should.

The aim here is not to settle final responsibility in law or ethics. It is to make accountability in human-agent and agent-agent systems easier to see, easier to test, and harder to blur into the background. If the framework succeeds, it will do so not because it sounds elegant, but because it helps institutions record the moments when judgment, delegation, termination, and verification actually mattered.

References

[1] Mike Ananny and Kate Crawford. Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society*, 20(3):973–989, 2018. doi: 10.1177/1461444816676645. doi: `https://doi.org/10.1177/1461444816676645`.

[2] Nicholas Diakopoulos. Accountability in algorithmic decision making. *Communications of the ACM*, 59(2):56–62, 2016. doi: 10.1145/2844110. doi: `https://doi.org/10.1145/2844110`.

[3] Ben Green and Yiling Chen. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):50:1–50:24, 2019. doi: 10.1145/3359152. doi: `https://doi.org/10.1145/3359152`.

[4] John Kelsey and Bruce Schneier. Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security*, 2(2):159–176, 1999. doi: 10.1145/317087.317089. doi: `https://doi.org/10.1145/317087.317089`.

[5] Joshua A. Kroll, Joanna Huey, Solon Barocas, Edward W. Felten, Joel R. Reidenberg, David G. Robinson, and Harlan Yu. Accountable algorithms. *University of Pennsylvania Law Re-*

view, 165(3):633–705, 2017. official URL: https://scholarship.law.upenn.edu/penn_law_review/vol165/iss3/3/.

[6] Ben Laurie, Adam Langley, and Emilia Kasper. Rfc 6962: Certificate transparency. Technical Report RFC 6962, IETF, 2013. doi: https://doi.org/10.17487/RFC6962.

[7] Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 30(3):286–297, 2000. doi: 10.1109/3468.844354. doi: https://doi.org/10.1109/3468.844354.

[8] Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 33–44, 2020. doi: 10.1145/3351095.3372873. doi: https://doi.org/10.1145/3351095.3372873.

[9] Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and abstraction in sociotechnical systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 59–68, 2019. doi: 10.1145/3287560.3287598. doi: https://doi.org/10.1145/3287560.3287598.

[10] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *ACM SIGMOD Record*, 34(3):31–36, 2005. doi: 10.1145/1084805.1084812. doi: https://doi.org/10.1145/1084805.1084812.

[11] Herbert A. Simon. *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations*. Free Press, New York, 4 edition, 1997. ISBN 9780684835822. publisher URL: https://www.simonandschuster.com/books/Administrative-Behavior-4th-Edition/Herbert-A-Simon/9780684835822.

[12] Elham Tabassi et al. Ai risk management framework (ai rmf 1.0). Technical Report NIST AI 100-1, National Institute of Standards and Technology, 2023. doi: https://doi.org/10.6028/NIST.AI.100-1.