

When Agents Delegate: Accountability Chains in Multi-Agent AI Systems

Ornella Bahidika

Independent Researcher, Seattle, USA

Abstract—Ethical frameworks for the oversight of autonomous AI systems generally assume a dyadic relationship: one human principal authorizing and supervising one artificial agent. Deployed practice has moved on. Contemporary agentic systems decompose tasks and delegate them onward, to sub-agents they instantiate, to third-party agents discovered through interoperability protocols, and to tools whose own behavior is agentic. Each hop in such a delegation chain attenuates the three prerequisites on which accountability depends: authorized scope, attributable intent, and reconstructable provenance. This study analyzes the resulting accountability gap. We contribute a taxonomy of delegation structures in deployed agentic systems; an analysis of four failure modes by which delegation dissolves accountability, which we term authorization dilution, intent drift, provenance loss, and responsibility laundering; a normative framework, the Delegation Accountability Chain, comprising five conditions under which multi-agent delegation remains compatible with meaningful human oversight; an audit of five widely used interoperability protocols and orchestration frameworks (the Model Context Protocol, the Agent2Agent protocol, LangGraph, AutoGen, and the OpenAI Agents SDK) against the five conditions, finding that none satisfies any condition chain-wide at a mandatory level; and a mapping of the conditions onto the human-oversight obligations of the EU Artificial Intelligence Act and the NIST AI Risk Management Framework, both drafted for the dyadic case. We argue that the question of where the buck stops has a defensible answer in multi-agent systems only if delegation is constrained by design.

Keywords—Agentic AI; accountability; delegation; multi-agent systems; human oversight; AI governance; responsibility gap

I. INTRODUCTION

In earlier work we asked where the buck stops when an autonomous AI system acts consequentially on a human's behalf, proposed an action-level tiered oversight framework calibrated by stakes and reversibility, and identified tier laundering, the composition of individually low-tier actions into a higher-tier outcome, as the central methodological gap in per-action oversight [1]. That work named composition-aware oversight as the highest-leverage direction for follow-up research. Like nearly all of the human-oversight literature it engaged, that earlier analysis presumed a single agent answerable to a single accountable human. The present study takes up its follow-up program for the setting in which composition is most acute and least governed: delegation among agents.

The deployed world has moved past that presumption. Production agentic systems now plan by decomposition: an orchestrating agent breaks a goal into subtasks and assigns them to other agents. Interoperability protocols such as the Model Context Protocol and agent-to-agent communication

standards make it routine for an agent to enlist capabilities that its human principal never inspected, sourced from parties the principal never chose [2], [3]. Each such hop is a delegation, an operation classical accountability theory treats with suspicion: Bovens defines accountability as a relationship in which an actor is obliged to explain and justify conduct to a forum that can pose questions and impose consequences [4]. A forum cannot pose questions to a chain whose links it cannot see. Fig. 1 contrasts the two settings.

This study asks what oversight can mean when the overseen agent is itself a principal to other agents. We proceed as follows. Section II situates the problem in the responsibility-gap literature and shows that current regulatory instruments assume the dyadic case. Section III offers a taxonomy of delegation structures observed in deployed systems. Section IV analyzes four failure modes by which delegation dissolves accountability. Section V presents the Delegation Accountability Chain, five conditions under which delegation remains compatible with meaningful human oversight. Section VI audits five widely deployed interoperability protocols and orchestration frameworks against the conditions. Section VII maps the conditions onto the EU AI Act and the NIST AI RMF. Section VIII considers objections and limitations, and Section IX concludes.

II. BACKGROUND AND RELATED WORK

Three bodies of work frame human oversight of AI, and all three exhibit the dyadic assumption.

1) *Responsibility gaps*: Matthias observed two decades ago that learning machines create situations in which no human satisfies the classical conditions for responsibility, control and knowledge [5]. Nissenbaum earlier identified “many hands” as a barrier to accountability in computerized systems [6]. Delegating agents combine both problems: the hands are now artificial and generated at runtime.

2) *Meaningful human control*: Santoni de Sio and van den Hoven’s influential account requires that autonomous systems track the reasons of the relevant humans and that their behavior be traceable to a proper moral and technical understanding by designers and operators [7]. Both the tracking and tracing conditions are defined against a fixed system boundary. Delegation makes the boundary dynamic: the “system” at time t includes agents that did not exist at deployment time.

3) *Multi-agent systems and capability security*: Two older literatures bear directly on delegation and are rarely joined to the oversight debate. Multi-agent systems research has studied cooperation, negotiation, and delegation among software

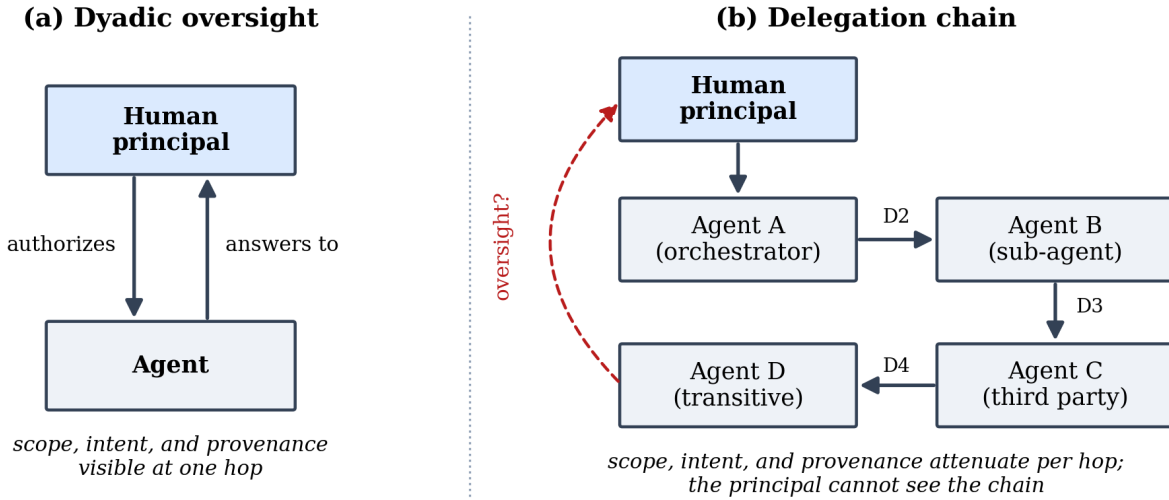


Fig. 1. Dyadic oversight versus a delegation chain. In the dyad, authorization and answerability meet at a single hop. In a chain, scope, intent, and provenance attenuate per hop, and the principal cannot see the chain acting under its grant.

agents since the 1990s [8], but its concern was coordination among agents, not accountability of agents to humans; the field's delegation formalisms assume the principals are themselves agents. Capability security, by contrast, is about conveying bounded authority: Hardy's confused-deputy analysis showed that authority which travels implicitly is authority that gets misused [9], and the object-capability tradition demonstrated that authorization can be made to travel as an unforgeable artifact that composes safely [10]. The framework of Section V can be read as importing the capability discipline into the human-oversight debate: what capability systems do for programs, delegation-accountability conditions must do for chains that terminate in a responsible person.

4) *Regulatory instruments*: Article 14 of the EU AI Act requires that high-risk systems be designed so that natural persons can effectively oversee them, including the ability to understand capacities and limitations, remain aware of automation bias, correctly interpret output, and intervene or interrupt [11]. The NIST AI Risk Management Framework likewise organizes its Govern, Map, Measure, and Manage functions around an identified system with an identified operator [12]. Neither instrument contemplates a system that recruits other systems after deployment; both implicitly assume the object of oversight is the object that was assessed.

Recent work has begun to note the strain. Ethics-based auditing programs presume access to a bounded artifact [13]; scholars of the algorithmic society observe that autonomy blurs attribution [14]; practitioner commentary asks the buck-stops question of agentic AI directly [15]; and Chan et al. propose concrete visibility mechanisms for agents, including identifiers, real-time monitoring, and activity logs [16]. Closer still, concurrent work proposes point mechanisms for delegation itself: a cryptographic protocol that accumulates signed per-hop records anchored in a human principal's declared scope [17], identity-governance infrastructure for propagating authorization through multi-agent systems [18], and a draft Internet standard for human-anchored agent identity, delegation, and provenance [19]. These proposals supply individual mecha-

nisms. What this study adds is an account of delegation as the mechanism by which attribution dissolves; an argument that the five conditions of Section V must hold jointly and be defined over chains rather than agents, since a compliant agent delegating to a compliant agent does not yield a compliant chain; and an audit showing that none of these mechanisms, proposed or shipped, is required by the specifications agents actually run on.

III. A TAXONOMY OF DELEGATION IN DEPLOYED AGENTIC SYSTEMS

We distinguish four delegation structures, ordered by increasing accountability risk, and note a deployed public example of each.

1) *D1. Static tool invocation*: The agent calls a fixed, enumerated tool (a search API, a database query). The tool set is known at deployment and can be assessed with the system. This is delegation only in a degenerate sense; classical oversight frameworks handle it. Mainstream chat assistants invoking a fixed web-search or code-execution tool exemplify this class.

2) *D2. Self-decomposition*: An orchestrating agent instantiates subordinate agents of its own kind, assigns them sub-tasks, and integrates their results. The sub-agents inherit the orchestrator's capabilities but not necessarily its constraints; whether instructions, permissions, and prohibitions survive the handoff is an implementation choice, often an accidental one. Anthropic's production research system is a documented public example: a lead orchestrator agent decomposes a query and spawns parallel subagents whose results it synthesizes [20].

3) *D3. Heterogeneous enlistment*: The agent discovers and engages third-party agents or services through an interoperability protocol. The enlisted party was built by a different organization, is governed by different policies, and may change its behavior between the principal's assessment and the moment of use. The MCP ecosystem exemplifies this class: hosts routinely connect to community-published third-party servers,

and the A2A protocol was designed expressly for cross-vendor agent enlistment [2], [3].

4) *D4. Transitive delegation*: An enlisted agent itself delegates onward, producing chains of depth greater than two. No party in the chain, including the original human principal, has visibility over the full set of actors, and depth is unbounded in current protocols. The audited specifications themselves contemplate this class: an MCP server may act as an OAuth client to upstream services, and A2A permits a server to relay authorization requirements through a chain of dependent tasks (Section VI).

Table I summarizes the four classes.

TABLE I. DELEGATION STRUCTURES IN DEPLOYED AGENTIC SYSTEMS

Class	Delegate set	Public example	Principal's visibility
D1 Static tool invocation	Fixed and enumerable at deployment	Chat assistant with a fixed search or code tool	Full
D2 Self-decomposition	Instantiated at runtime by the orchestrator	Orchestrator spawning parallel subagents [20]	Partial
D3 Heterogeneous enlistment	Discovered at runtime from third parties	Community MCP servers; A2A cross-vendor agents [2], [3]	Low
D4 Transitive delegation	Recursively assembled, unbounded	Servers acting as clients to upstream services (Section VI)	None

Two properties make D2 through D4 categorically different from D1. First, the delegate set is determined at runtime, so pre-deployment assessment cannot enumerate it. Second, delegation is composable, so any framework that governs only single hops fails by induction: a compliant agent delegating to a compliant agent does not yield a compliant chain unless compliance is defined over chains.

IV. FOUR WAYS DELEGATION DISSOLVES ACCOUNTABILITY

A. Authorization Dilution

A human authorizes an agent for a scope: purposes, resources, and limits. When the agent delegates, the effective scope of the delegate is whatever the delegating agent communicates, which may be broader than, narrower than, or simply different from the original grant. Across hops, the human's authorization becomes a rumor. The delegate acts under a scope no human issued. Where a delegate exercises inherited credentials broader than the human's grant, this is the confused-deputy problem in its classic ambient-authority form [9]; where scope degrades through linguistic restatement, the failure is distinct but the remedy is the same: authority must travel explicitly or not at all.

B. Intent Drift

Each delegation restates the goal. Restatement is interpretation, and interpretation by a language model is lossy and generative at once: constraints are dropped, emphases shift, and instrumental subgoals are elevated to ends. The terminal agent in a chain may faithfully execute an instruction that no human ever gave. Drift is not misbehavior by any single link;

every link may act reasonably on what it received. The failure is emergent, which is why per-agent oversight cannot detect it.

C. Provenance Loss

Accountability requires reconstruction: which information the system received, what it retrieved, which actions it took, and why [21]. In a delegation chain, the records that would answer these questions are scattered across parties with different logging practices, retention policies, and disclosure obligations, or are not produced at all. The forum, in Bovens's sense, has no dossier to examine. Post-incident, it can be impossible even to establish that a chain existed.

D. Responsibility Laundering

The most corrosive failure is intentional or negligent exploitation of the first three: structuring action through delegation so that no identifiable party bears responsibility. The pattern is kin to agency laundering, in which humans obscure their own responsibility by routing decisions through automated systems [22]; responsibility laundering is its multi-agent completion: the routing is performed by agents, and no single human need intend the obfuscation. It also generalizes the tier-laundering pattern identified in our earlier framework [1]: where tier laundering composes low-tier actions within a single agent into a high-tier outcome, responsibility laundering composes individually oversightable agents into an unoversightable chain. Where a human organization performing the same decomposition would face joint liability doctrines, agent chains today face none. If accountability attaches only to single human-agent dyads, delegation becomes a machine for converting responsible action into unattributable outcome. Regulation that ignores this creates an incentive gradient toward deeper chains.

E. A Worked Example

The four failure modes rarely occur alone; a single ordinary chain can exhibit all of them. Consider a procurement assistant, authorized by an operations manager to renew existing software subscriptions up to a set monthly amount. Faced with a renewal that requires a vendor form, the orchestrator spawns a sub-agent to handle vendor interaction (self-decomposition) and, finding the vendor exposes an agent interface, enlists it through an interoperability protocol (heterogeneous enlistment). The vendor's agent, in turn, calls its own payment-processing agent (transitive delegation). At the first hop, the manager's spending cap and renewals-only restriction survive only as sentences in a prompt; the sub-agent inherits the orchestrator's payment credential, whose scope is whatever the credential allows rather than what the manager authorized (authorization dilution). Restating the goal as "complete the vendor's renewal process" drops the cap and the renewals-only qualifier, and the vendor's agent reasonably treats an upsell bundle as part of that process (intent drift). When the resulting charge exceeds anything the manager approved, the reconstruction spans the deployer's logs, the vendor's logs, and a payment processor's records, none linked by a shared identifier and none obliged to disclose (provenance loss). Each party can truthfully say it executed the instructions it received, and the vendor can note that its agent was never told about

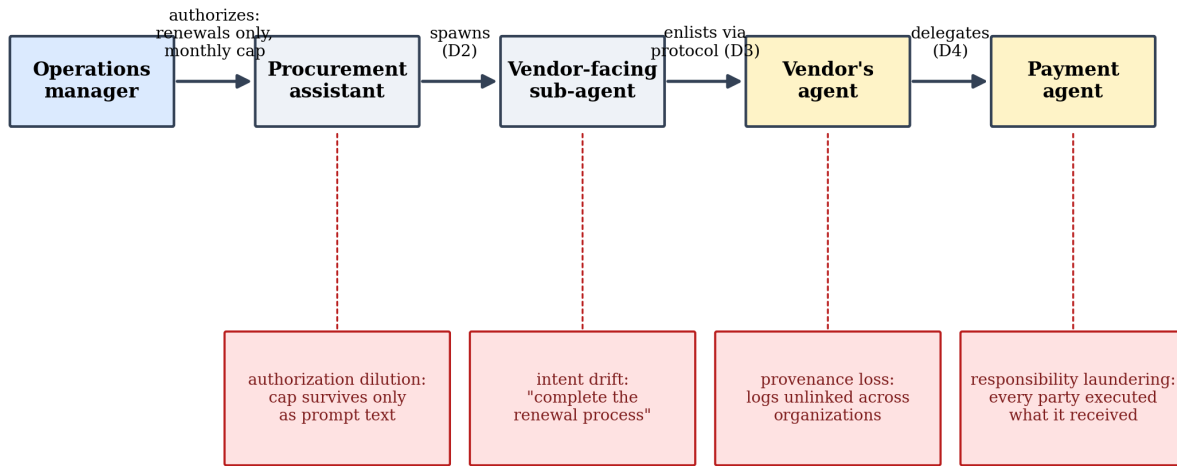


Fig. 2. The worked example: an ordinary procurement chain exhibiting all four failure modes. No party misbehaves; the accountability failure is structural.

any cap; no identifiable party bears the failure (responsibility laundering). Nothing in this chain requires malice or unusual carelessness. Fig. 2 traces the chain and its failures. Section VI shows that the infrastructure such a chain would run on today provides none of the mechanisms that would have interrupted it.

V. THE DELEGATION ACCOUNTABILITY CHAIN

We propose five conditions, jointly necessary for a delegating agentic system to support claims of meaningful human oversight. The conditions are design constraints: each is checkable against an implementation.

1) *C1. Conserved authorization*: A delegate's effective scope must be a subset of the delegator's scope, enforced mechanically rather than communicated linguistically. Scope must travel as a verifiable artifact (a capability token or its equivalent) that downstream agents cannot expand. The chain-wide invariant is that no action occurs outside the original human grant.

2) *C2. Human anchor*: Every chain terminates, upward, in an identified natural person or accountable body that issued the originating authorization and remains the answerable party for the chain's aggregate conduct. Anchorship is non-delegable: an agent cannot become an anchor. This preserves Bovens's forum relationship: there is always someone obliged to explain.

3) *C3. Attributable initiation*: Every delegation event must record who delegated, to whom, under what communicated goal, and under what scope, in a form attributable to the anchor's chain. Delegation to parties that refuse attribution is prohibited conduct, not a logging gap.

4) *C4. Chain-complete provenance*: The reconstruction standard that audit scholarship applies to single systems [21] must be defined over the chain: a single query, anchored at the human principal, must be able to produce the full delegation tree with per-link records sufficient to answer what was received, retrieved, and done, and what rationale was stated. Practically this requires provenance to propagate with the work, not to be assembled forensically afterward.

5) *C5. Bounded depth and diameter*: Because drift and dilution compound per hop, chains must carry explicit limits on delegation depth and on the set of admissible delegates, set by the anchor as part of the original grant. Unbounded delegation defeats oversight, and assessors should treat it as a per se failure of Article 14-style obligations.

Each condition carries an operational test an assessor can apply; Table II states them.

TABLE II. THE DELEGATION ACCOUNTABILITY CHAIN: CONDITIONS AND OPERATIONAL TESTS

Condition	Requirement	Operational test
C1 Conserved authorization	Delegate scope is a mechanically enforced subset of the delegator's	Every delegated credential is cryptographically derived from, and attenuates, the parent grant
C2 Human anchor	Every chain terminates upward in an identified accountable person or body	Any action in the chain resolves in a single lookup to an identified anchor
C3 Attributable initiation	Every delegation event is recorded and attributable	Every delegation edge exists as a typed record naming delegator, delegatee, goal, and scope
C4 Chain-complete provenance	The full delegation tree is reconstructable from the origin	One anchor-issued query returns the full tree with per-link records
C5 Bounded depth and diameter	The grant limits delegation depth and admissible delegates	The grant carries explicit bounds the runtime refuses to exceed

Deterministic enforcement is the connective tissue of the five conditions. Linguistic instruction is the medium of drift; only mechanisms outside the model, gating code that validates scope, records provenance, and refuses noncompliant delegation, can carry authorization across hops without loss. The design stance is that the agent proposes and the mechanism decides, a pattern practitioners have converged on independently for reliability reasons, and which we here argue has normative status: it is what makes delegation oversightable at all.

VI. DO CURRENT PROTOCOLS SATISFY THE CHAIN CONDITIONS? AN AUDIT

The conditions of Section V would be of limited interest if the ecosystem already met them. To assess this, we audited the current normative specifications and official documentation of five artifacts, selected as the most widely adopted interoperability protocols and multi-agent orchestration frameworks at the time of writing (Table III): the Model Context Protocol (MCP, specification revision 2026-07-28), the Agent2Agent protocol (A2A, specification 1.0.0), and the LangGraph, Microsoft AutoGen, and OpenAI Agents SDK frameworks (documentation as of August 2026). For each artifact we assessed each condition as satisfied (a mandatory, MUST-level mechanism exists at protocol or framework level), partial (optional, SHOULD-level, or achievable only through application-level configuration), or absent (no relevant provision; we record what was searched and not found). Two clarifications govern the exercise. First, the three frameworks are developer libraries that make no governance claims of their own; we audit them not against their own claims but against the oversight claims deployers make for systems built on them, because those are the claims the conditions of Section V underwrite. Second, the mandatory-mechanism bar is deliberate. For accountability in particular, optional recording is weak assurance, because the parties best positioned to disable a record are the parties it would hold to account (Section IV-D); default-on but disable-able is therefore materially different from mandatory. We audit what each artifact guarantees rather than what a diligent application could build atop it; whatever is not guaranteed is left to the discretion of each implementer. Audit working notes, including full verbatim quotations and source URLs for every assessment, are available as supplementary material (S1: audit working notes; S2: machine-readable audit evidence) at <https://github.com/ornellabahidika/when-agents-delegate-supplement>.

No cell in the table is satisfied; Fig. 3 visualizes the verdicts. We discuss each artifact, then the pattern.

A. Model Context Protocol

MCP contains the strongest single-hop authorization mechanics of any audited artifact, and their limits are instructive. When authorization is used, it is mandatory in the details: clients MUST implement OAuth 2.0 Resource Indicators (RFC 8707), servers “MUST validate that access tokens were issued specifically for them as the intended audience,” and token passthrough is prohibited (“MCP servers MUST NOT accept or transit any other tokens”); the protocol’s separate security-best-practices document repeats the ban (“MCP servers MUST NOT accept any tokens that were not explicitly issued for the MCP server”). But authorization as a whole is “OPTIONAL for MCP implementations,” applies only to HTTP transports (local stdio servers sit outside the framework and are directed to ambient environment credentials, the classic confused-deputy configuration [9]), and it is pairwise: each hop obtains a separate token from its own authorization server, so a delegate’s effective scope is set by that server’s policy, not derived from the delegator’s grant; scope subsetting appears only as guidance (“MCP clients SHOULD follow the principle of least privilege”). The consent model is anchored at the human-to-host hop and the specification is direct about

its reach: “While MCP itself cannot enforce these security principles at the protocol level,” implementors are asked, at SHOULD strength, to build robust consent and authorization flows. No message or field records a delegation event (C3), and the spec’s own security guidance concedes that passthrough patterns “make incident investigation, controls, and auditing more difficult” without requiring any audit record in response. The architecture works against chain reconstruction by design: servers “should not be able to read the whole conversation, nor ‘see into’ other servers” (C4). No provision bounds chain length or membership; composability is a design goal (“Servers should be highly composable”), and a server may itself act as an OAuth client to upstream services (C5).

B. Agent2Agent Protocol

A2A mandates per-request security: “Servers MUST reject requests with invalid or missing authentication credentials” and “Servers MUST implement authorization checks on every A2A Protocol Operations request.” An OpenID Connect security scheme is defined, so a human principal can be authenticated at a given hop, though the authenticated client is typically an agent or service; the scheme carries end-user identity only where a deployment arranges it. But identity does not travel: “A2A protocol payloads... don’t carry user or client identity information directly. Identity is established at the transport/HTTP layer,” which means the originating human is invisible one hop beyond the client (C2 partial). No requirement records who delegated what to whom; the protocol’s contextId and referenceTaskIds group related tasks informationally but are not delegation records (C3 absent). Tracing and audit appear only in non-normative enterprise guidance at SHOULD strength (C4 partial). Notably, A2A does define one chain-aware construct: a task needing credentials can propagate that need upstream through a chain of tasks in an authorization-required state, with the sensible rule that a credential obtained this way “MUST NOT be assumed to authorize subsequent messages on the Task.” This is a credential-acquisition relay rather than an accountability mechanism, but it demonstrates that chain-scoped constructs are expressible at protocol level when designers choose to define them. Nothing bounds chain depth, and agent discovery through published AgentCards is open-ended (C5 absent).

C. LangGraph

LangGraph’s multi-agent documentation describes hand-offs as control transfer: “Agents transfer control to each other via tool calls.” What travels is graph state and control flow, not authorization; a subordinate node’s tools are whatever the developer wired in, with no mechanical relation to the delegator’s scope (C1 absent). Human involvement exists as the interrupt() primitive, which can “pause before a critical action and ask for approval,” but it is optional, per-node, and identifies no one: the docs speak of a generic “human” with no identity or authentication attached (C2 partial). Recording is likewise optional: the docs recommend tracing “the full coordination flow across agents with LangSmith,” an external opt-in service, and persistent state history requires configuring a checkpoint (C3, C4 partial). The framework’s one always-on limit is enforced: a default recursion limit of 1,000 steps, raised from the long-standing default of 25 in version 1.0.6, with a

TABLE III. AUDIT RESULTS (SATISFIED / PARTIAL / ABSENT)

Artifact	C1 conserved authorization	C2 human anchor	C3 attributable initiation	C4 chain provenance	C5 bounded depth
MCP (2026-07-28)	partial	partial	absent	absent	absent
A2A (1.0.0)	partial	partial	absent	partial	absent
LangGraph	absent	partial	partial	partial	partial
AutoGen	absent	partial	partial	partial	partial
OpenAI Agents SDK	absent	absent	partial	partial	partial

Audit verdicts: five artifacts against the five chain conditions (no cell satisfied)

MCP (2026-07-28)	partial	partial	absent	absent	absent
A2A (1.0.0)	partial	partial	absent	partial	absent
LangGraph	absent	partial	partial	partial	partial
AutoGen	absent	partial	partial	partial	partial
OpenAI Agents SDK	absent	absent	partial	partial	partial
	C1 conserved authorization	C2 human anchor	C3 attributable initiation	C4 chain provenance	C5 bounded depth

Fig. 3. Audit verdicts for the five artifacts against the five chain conditions. Green would indicate a mandatory chain-level mechanism; no cell qualifies.

hard GraphRecursionError when exceeded. But it bounds the total steps of a run rather than delegation depth or membership, and the caller may raise or effectively remove it (C5 partial).

D. AutoGen

In AutoGen’s Swarm pattern, a handoff is a model tool call in a shared conversational context: “agents can hand off task to other agents based on their capabilities,” and “all agents share the same message context.” The per-agent handoffs list restricts targets, but it is constructor configuration, not a verifiable artifact, and no scope subsetting exists between delegator and delegate (C1 absent). A human control point exists, the UserProxyAgent, which “blocks the execution of the team until the user provides feedback or errors out,” but including one is optional and the human is an anonymous input callback (C2 partial). Handoffs are structurally represented with source and target within a run’s message stream, which permits attribution while the run lives, but durable recording is not required and only the Swarm pattern uses handoff messages at all; AutoGen’s other team patterns, round-robin and selector-based group chats and the publish-subscribe core runtime, route work by speaker selection or topic subscription with no handoff record whatever (C3 partial). Telemetry is native but opt-in, and can be disabled with an environment variable (C4 partial). Termination conditions are numerous and optional, and the standard group-chat teams’ turn limit “defaults to None, meaning no limit” (C5 partial).

E. OpenAI Agents SDK

The SDK is the strongest audited artifact on provenance and among the weakest on authorization. At a handoff, “the new agent takes over the conversation, and gets to see the entire previous conversation history”; what transfers is context, not scoped authority, and the successor’s tools are configured independently of the delegator’s, so a delegate can lawfully be broader than its principal (C1 absent). No accountable human exists at framework level: tool approvals are opt-in, and both approvals and traces attach to agent names and API keys rather than identified persons (C2 absent). Tracing, however, is on by default and chain-complete within a run: the runner is wrapped in a trace, “[e]ach time an agent runs, it is wrapped in agent_span(),” and “[h]andoffs are wrapped in handoff_span(),” so the delegation tree of a run is reconstructable as shipped (C3, C4 partial). It remains disable-able: a single environment variable turns it off globally, sensitive payloads can be stripped, and it is structurally unavailable under zero-data-retention policies. The turn limit bounds the run but may be disabled outright (max_turns=None) (C5 partial).

F. What the Pattern Means

Three regularities run through the table. First, security is per-hop and chain-blind: the protocols invest heavily in authenticating and authorizing each pairwise connection while leaving the chain, the thing the human actually needs oversight of, unrepresented. Second, attribution and provenance are advisory wherever they exist: every recording mechanism found is

optional, disable-able, or non-normative, and what is optional is, for accountability purposes, absent when it matters most. Third, limits exist as loop guards rather than authorization semantics: recursion and turn limits protect against runaway cost, are set by the deployer rather than the authorizing principal, and constrain neither depth of delegation nor who may be enlisted.

The audit supports two conclusions. First, the failure modes of Section IV are not hypothetical: the infrastructure on which multi-agent systems are being built provides no chain-wide authorization conservation, no required delegation records, no reconstructable provenance across trust boundaries, and no principal-set bounds on depth or membership. Second, the gap is tractable: audience-bound tokens, default-on trace spans, and A2A's chained authorization state show that each required primitive already exists somewhere in the ecosystem in embryonic, hop-local form. What is missing is not technique but requirement: no specification makes any of it mandatory across a chain. That is an actionable finding for protocol maintainers, and Section VII argues it should also be an actionable finding for regulators.

VII. MAPPING TO THE EU AI ACT AND NIST AI RMF

A. Article 14, EU AI Act

Effective oversight by natural persons is not satisfiable over a chain that violates C2 (no answerable person), C4 (nothing to interpret or monitor), or C5 (capacities and limitations undefined because the delegate set is unbounded). We propose that conformity assessment for agentic high-risk systems should treat C1 through C5 as the operationalization of Article 14: a system that cannot demonstrate them cannot demonstrate effective oversight, whatever its dyadic interface provides. Article 14(4) makes this concrete. Its measures must enable the overseer to understand the system's capacities and limitations, which are undefined while the delegate set is open (C5); to correctly interpret the system's output, which presupposes records of what the chain did (C4); to decide not to use the system in a particular situation, which presupposes knowing what the system currently comprises; and to intervene or halt it, which requires an anchor with authority over the whole chain (C2). Article 25 reallocates provider obligations along the value chain: a deployer, importer, or distributor becomes the provider if it puts its name on a high-risk system, substantially modifies it, or modifies its intended purpose. Whether runtime delegation triggers the reallocation turns on the Act's definition of substantial modification in Article 3(23): a change not foreseen in the provider's conformity assessment that affects compliance or intended purpose. Delegation the provider designed for, within an assessed and bounded delegate set, is foreseen and triggers nothing. An unbounded delegate set, by contrast, cannot have been assessed, because the parties, capabilities, and purposes it will encompass did not exist at assessment time; delegation beyond the assessed boundary is unforeseen by construction and, on a plain reading of Article 3(23), a substantial modification whose provider obligations fall to the operator of the delegating system. C3's delegation records are the evidence such a rule needs to be administrable, and C5's bounds are what makes a delegate set assessable at all.

B. NIST AI RMF

The Map function must map chains, not systems: the inventory object is the delegation tree the anchor can generate under C4. Measure inherits chain-level metrics (observed depth, scope-violation attempts refused, drift measured as divergence between originating and terminal instructions). Manage gains a concrete lever: anchors adjust C5 bounds as risk tolerance changes. Govern locates the anchor role organizationally, which is where our earlier framework's answer to the buck-stops question [1] becomes an org-chart fact rather than a metaphor. For protocol maintainers, the same mapping reads as a work list: the hop-local primitives the ecosystem already ships, audience-bound tokens, trace spans, chained authorization states, become compliance infrastructure the moment they are made mandatory, chain-scoped, and anchored.

VIII. OBJECTIONS AND LIMITATIONS

1) *Does this not simply prohibit useful autonomy?*: C1 through C5 constrain how delegation happens, not whether. Systems may still decompose, enlist, and chain; they must do so within a conserved grant, with propagated provenance, under bounded depth. The claim is that autonomy purchased by abandoning these conditions is autonomy purchased by abandoning oversight, and that operators should not be permitted to claim both.

2) *Is mechanical scope conservation feasible across organizational boundaries?*: Partially, today. Mechanical attenuation is solved within a trust domain: OAuth token exchange (RFC 8693) narrows delegated tokens, and macaroon-style credentials attenuate by appending caveats that downstream parties cannot remove [23], and concurrent proposals extend the same primitives to agent chains [17], [19]. What no deployed standard provides is a shared scope vocabulary across organizations, without which "subset" is undefined between authorization servers; C1 across organizational boundaries therefore requires protocol evolution toward verifiable scope mapping, not new cryptography. Where conservation is infeasible, C5 answers: such parties are outside the admissible delegate set.

3) *Limitations*: The audit of Section VI assesses specifications and official documentation, not deployed configurations; individual deployments may exceed (or fall below) what their substrate guarantees, and the audited documents are living artifacts whose provisions may change. The audit reflects single-author coding against a three-level rubric whose partial-or-absent boundary involves judgment; the working notes expose every assessment for reexamination, but no inter-rater reliability is reported. The artifacts were selected for adoption rather than sampled, and the conditions' costs in latency, capability, and engineering effort are not quantified here. The taxonomy reflects systems observable in 2026 and will age. Finally, the anchor condition presumes legal persons willing to be anchors; market pressure toward responsibility laundering (Section IV-D) may require regulatory force that this study can only recommend.

IX. CONCLUSION AND FUTURE WORK

Where does the buck stop when agents delegate to agents? Our answer: exactly where it stopped before, at an identified

human principal serving as the chain's anchor, but only if the chain beneath that anchor is built so that authorization is conserved, initiation is attributable, provenance is chain-complete, and depth is bounded. These are not ethics layered onto engineering; they are engineering requirements derived from what accountability means. Systems that cannot meet them do not merely have an accountability gap; they are built in a way that refuses accountability, and their operators should not be able to claim otherwise.

DATA AVAILABILITY

The audit working notes underlying Section VI, comprising verbatim quotations, source URLs, and per-assessment rationale for all twenty-five verdicts, are available as supplementary material (S1: audit working notes; S2: machine-readable audit evidence) at <https://github.com/ornellabahidika/when-agents-delegate-supplement>. S1 provides the human-readable working notes; S2 provides the same assessments as structured, machine-readable records.

DECLARATION ON GENERATIVE AI

In preparing this manuscript, the author used Claude (Anthropic) to assist with drafting, language refinement, structural organization of the text, and execution of the protocol audit reported in Section VI (retrieving the audited specifications, extracting candidate passages, and organizing assessments, with independent adversarial re-verification of every quotation against the source documents). The conceptual framework, the five chain conditions, the audit rubric and verdicts, the scholarly arguments, the selection of cited literature, and the final conclusions are the responsibility of the author. Bibliographic details for all cited references were verified by the author against the original sources. All AI-assisted text and analysis was critically reviewed by the author, who takes full accountability for the accuracy, originality, and integrity of the content.

REFERENCES

- [1] O. Bahidika, "Agentic Accountability: 'The Buck Stops Where?' Ethical Frameworks for Human Oversight of Autonomous AI Systems," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 17, no. 5, pp. 13–22, 2026, doi: 10.14569/IJACSA.2026.0170502.
- [2] Model Context Protocol project, "Model Context Protocol specification, revision 2026-07-28," originated by Anthropic. [Online]. Available: <https://modelcontextprotocol.io/specification/2026-07-28>
- [3] A2A Project / Linux Foundation, "Agent2Agent (A2A) protocol, specification 1.0.0," originated by Google, 2025. [Online]. Available: <https://a2a-protocol.org/latest/specification/>
- [4] M. Bovens, "Analysing and assessing accountability: A conceptual framework," *European Law Journal*, vol. 13, no. 4, pp. 447–468, 2007.
- [5] A. Matthias, "The responsibility gap: Ascribing responsibility for the actions of learning automata," *Ethics and Information Technology*, vol. 6, no. 3, pp. 175–183, 2004.
- [6] H. Nissenbaum, "Accountability in a computerized society," *Science and Engineering Ethics*, vol. 2, no. 1, pp. 25–42, 1996.
- [7] F. Santoni de Sio and J. van den Hoven, "Meaningful human control over autonomous systems: A philosophical account," *Frontiers in Robotics and AI*, vol. 5, art. 15, 2018.
- [8] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.
- [9] N. Hardy, "The confused deputy (or why capabilities might have been invented)," *ACM SIGOPS Operating Systems Review*, vol. 22, no. 4, pp. 36–38, 1988.
- [10] M. S. Miller, "Robust composition: Towards a unified approach to access control and concurrency control," Ph.D. thesis, Johns Hopkins University, 2006.
- [11] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 (Artificial Intelligence Act)," *Official Journal of the European Union*, OJ L 2024/1689, 2024.
- [12] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," 2023.
- [13] J. Mökander, J. Morley, M. Taddeo, and L. Floridi, "Ethics-based auditing of automated decision-making systems: Nature, scope, and limitations," *Science and Engineering Ethics*, vol. 27, no. 4, art. 44, 2021.
- [14] A. Mukherjee and H. H. Chang, "Agentic AI: Autonomy, accountability, and the algorithmic society," arXiv:2502.00289, 2025.
- [15] Communications of the ACM News, "Agentic AI: The buck stops where?," 2026. [Online]. Available: <https://cacm.acm.org/news/agentic-ai-the-buck-stops-where/>
- [16] A. Chan, C. Ezell, M. Kaufmann, K. Wei, L. Hammond, et al., "Visibility into AI agents," in *Proc. 2024 ACM Conf. on Fairness, Accountability, and Transparency (FAccT '24)*, 2024.
- [17] A. Dalugoda, "HDP: A lightweight cryptographic protocol for human delegation provenance in agentic AI systems," arXiv:2604.04522, 2026.
- [18] K. Tallam, "Authorization propagation in multi-agent AI systems: Identity governance as infrastructure," arXiv:2605.05440, 2026.
- [19] Beyer, "Architecture for human-anchored agent identity, delegation, and provenance," IETF Internet-Draft, draft-beyer-agent-identity-architecture-00, 2026.
- [20] Anthropic, "How we built our multi-agent research system," *Anthropic Engineering*, June 2025. [Online]. Available: <https://www.anthropic.com/engineering/multi-agent-research-system>
- [21] I. D. Raji et al., "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in *Proc. FAT* 2020*, pp. 33–44, 2020.
- [22] A. Rubel, C. Castro, and K. Pham, *Algorithms and Autonomy: The Ethics of Automated Decision Systems*. Cambridge University Press, 2021.
- [23] A. Birgisson, J. G. Politz, U. Erlingsson, A. Taly, M. Vrabie, and M. Lentzner, "Macaroons: Cookies with contextual caveats for decentralized authorization in the cloud," in *Proc. NDSS 2014*.