Mediation, State, and Verification in Enterprise Multi-Agent AI Systems: A Formal Reference Architecture and Evaluation Protocol

Ahmed Abdullah
Sitka AI Technologies LLC, Frisco, Texas, USA
abdullah@sitka.ai

October 1, 2026

Abstract

Enterprise deployment of large language model agents is increasingly limited less by model capability than by the problem of coordinating models, tools, persistent business state, authorization policies, and human decisions into a workflow that can be operated and audited. Recent evidence complicates the assumption that multi-agent decomposition is beneficial by default: compute-normalized comparisons report that single agents match or exceed multi-agent systems at equal reasoning budgets, and failure analyses attribute a large share of multi-agent breakdowns to specification and inter-agent verification rather than to component capability. This paper takes that evidence as its starting point and asks a narrower question: which architectural properties, as opposed to which agent topology, determine whether an enterprise agentic workflow is governable, recoverable, and auditable.

We formalize an enterprise agentic workflow as a policy-constrained stateful system over agents, tools, workflow state, authorization policies, and human decision points, and establish four elementary results. First, policy soundness is a property of complete mediation of external actions and is independent of agent policy quality, which implies that it survives prompt injection and model failure whereas prompt-level guardrails do not. Second, recoverability and auditability require a durable external projection of every state component on which a consequential decision depends; context-resident state admits observationally indistinguishable histories with divergent correct continuations. Third, under an explicit independence idealization, sequential decomposition into n bounded agents without inter-stage verification bounds end-to-end success by p^n, so the value of decomposition is conditioned on handoff verification rather than on specialization, which is consistent with reported empirical failure taxonomies. Fourth, attributability follows from complete mediation together with a total authorization relation over consequential actions and append-only logging.

We map these results onto a five-layer reference architecture, a six-stage adoption protocol with explicit exit evidence, and a multi-level evaluation framework. We then specify a comparative evaluation protocol and an accompanying task schema that make the architecture falsifiable, including the configurations, metrics, and reporting requirements needed to test whether coordination earns its cost. The analytic results are proved; the architectural and adoption claims are stated as testable propositions and are not empirically validated here. We state explicitly which claims require experiment and what evidence would refute them.
Keywords: multi-agent systems; LLM agents; reference architecture; AI governance; agent evaluation; complete mediation; enterprise workflow automation

1 Introduction

Enterprise adoption of artificial intelligence increasingly depends on integrating models with tools, data, business rules, organizational processes, and human decisions. The difficulty becomes more pronounced when a system operates across a sequence of interdependent tasks rather than within a single conversational exchange. In revenue operations, a workflow may begin with an intent signal or inbound inquiry and continue through enrichment, qualification, personalized outreach, live interaction, scheduling, handoff, and post-sale retention. Each stage can involve different data sources, permissions, systems of record, decision rules, and accountability requirements. The resulting system must maintain state across steps, coordinate specialized capabilities, constrain external actions, recover from failures, and identify when human intervention is required.

Recent research supplies the components. Agentic language models combine reasoning with interaction and action (Plaat et al., 2025), tool-use methods allow models to query external environments and invoke application programming interfaces (Yao et al., 2023; Schick et al., 2023), and work on memory demonstrates that long-running behavior requires mechanisms for retaining and retrieving information across interactions (Park et al., 2023). Multi-agent research has examined explicit roles, routing, and communication structures (Tran et al., 2025). What this literature does not settle is how those components should be organized into a governed enterprise workflow with persistent business state, differentiated permissions, human decision rights, and measurable operational controls.

1.1 Motivation: the multi-agent premise is contested

A reference architecture for multi-agent enterprise systems must begin by acknowledging that the premise is under active challenge. Cemri et al. (2025) analyze more than two hundred annotated execution traces across seven multi-agent frameworks and attribute failures predominantly to specification defects, inter-agent misalignment, and absent task verification rather than to the reasoning capability of individual agents; they further report that structural and organizational interventions alone yield only modest improvement. Tran & Kiela (2026) compare single-agent and multi-agent configurations under equalized reasoning-token budgets and find that single agents match or exceed multi-agent systems on multi-hop reasoning, identifying budget-control artifacts that inflated earlier multi-agent results. Zhang et al. (2025) accordingly propose routing between single-agent and multi-agent execution rather than committing to either.

We take this evidence seriously rather than as an obstacle to be managed rhetorically. It implies that a paper advocating multi-agent decomposition on capability grounds is arguing against the current weight of evidence. This paper therefore does not make that argument. Our claim is narrower and, we believe, more defensible: in enterprise settings the binding constraints are governability, recoverability, and attributability, and these are properties of how external action and state are mediated, not of how many agents are present. Decomposition is justified, where it is justified at all, by the control and verification structure it makes possible, and it is harmful where it adds unverified handoffs. Proposition 3 makes this precise and yields a quantitative condition under which decomposition is counterproductive.

1.2 Research problem

The problem addressed here is an integration problem. Existing approaches provide mechanisms for individual capabilities such as reasoning, tool use, memory, and multi-agent interaction, but an enterprise deployment also requires explicit decisions about where state is maintained, how agents receive tools, which actions are authorized, how failures are handled, when humans must intervene, and how system behavior is evaluated

across complete workflows. Stated formally: which architectural properties are necessary and sufficient for an enterprise agentic workflow to be policy-sound, recoverable, and attributable, and how can a candidate architecture be tested against alternatives?

1.3 Contributions

1. A **formal system model** of an enterprise agentic workflow as a policy-constrained stateful system over agents, tools, state, policies, and human decision points (Section 3).
2. **Four analytic results** (Propositions 1 to 4) establishing that policy soundness is independent of agent policy quality under complete mediation; that recoverability requires durable external state projection; that the benefit of decomposition is conditioned on handoff verification, with an explicit bound; and that attributability follows from mediation plus total authorization and append-only logging.
3. A **five-layer reference architecture** that realizes these properties as separable, independently testable responsibilities (Section 5).
4. A **six-stage adoption protocol** in which each increase in scope or autonomy is gated on named exit evidence (Section 6).
5. A **comparative evaluation protocol and task schema** specifying the configurations, metrics, baselines, and reporting requirements needed to falsify the architectural claims (Section 7 and Appendix A).

1.4 Scope of claims

We state the epistemic status of each claim once, here, rather than repeating qualifications throughout. Propositions 1 to 4 are proved within the stated model and carry exactly the force their assumptions permit; Proposition 3 in particular rests on an independence idealization that we identify and discuss as unrealistic. The reference architecture, adoption protocol, and supervisory transition model are *design propositions*: they are specified precisely enough to be implemented and tested, and they are not empirically validated in this paper. No comparative performance measurement, controlled deployment study, or business-outcome result is reported. Deployment observations referenced in Section 12 are practitioner-reported and serve to motivate requirements, not to evidence effectiveness. Section 7 specifies what would constitute evidence for or against the design propositions, and Section 13 states what would refute them.

2 Background and Related Work

2.1 From language models to agents

Large language models supply reasoning and generation primitives; agentic systems extend these by allowing a model to interact with tools, environments, and persistent state. Chain-of-thought prompting showed that intermediate reasoning improves performance on complex reasoning tasks (Wei et al., 2022). ReAct interleaved reasoning traces with task-specific actions, allowing a model to obtain information from an external environment while revising its plan (Yao et al., 2023). Toolformer demonstrated self-supervised learning of when and how to call external interfaces (Schick et al., 2023).

Memory and state matter for long-running workflows. Generative Agents demonstrated an architecture in which experiences are stored, reflected upon, and retrieved to inform later behavior (Park et al., 2023).

In enterprise operations, however, memory cannot be treated only as conversational context. Business state must be represented in inspectable records, linked to system-of-record identifiers, governed by retention and access rules, and recoverable after interruption. ToolSandbox shows that implicit state dependencies between tools create evaluation challenges invisible to stateless, single-turn benchmarks (Lu et al., 2024).

2.2 Multi-agent collaboration and orchestration

Multi-agent systems introduce explicit roles and coordination structures. Tran et al. (2025) distinguish collaboration structures (peer-to-peer, centralized, distributed) from collaboration strategies (role-based, model-based), alongside actors, interaction types, and coordination protocols. AutoGen illustrates how multiple agents, humans, and tools can be composed into conversational workflows (Wu et al., 2024). Shu et al. (2024) report enterprise-oriented evaluation in which coordination and routing affect end-to-end goal success while introducing communication overhead and latency; we note that this is an industry technical report evaluating a vendor's own framework and is not peer reviewed, and we therefore do not treat it as independent evidence.

Against these, the critical literature surveyed in Section 1.1 finds that decomposition frequently fails to pay for itself (Cemri et al., 2025; Tran & Kiela, 2026). The present paper is positioned on the control side of that debate rather than the capability side.

2.3 Evaluation

Agent evaluation differs from static output evaluation because an agent selects tools, alters state, takes external actions, and continues across steps. AgentBench established interactive evaluation across heterogeneous environments (Liu et al., 2024). τ-bench introduced policy adherence and database-state verification in retail and airline domains, and its pass^k metric exposes reliability collapse under repeated trials (Yao et al., 2024); τ^2-bench extends this to dual-control settings in which the user also mutates shared state (Barres et al., 2025). CRMArena-Pro evaluates business workflows grounded in a customer relationship management platform across sales, service, and configure-price-quote scenarios, and reports near-zero baseline confidentiality awareness, a result directly relevant to governance design (Huang et al., 2025). Zhu et al. (2025) document systematic task-validity and outcome-validity defects that inflate reported agent performance and propose a checklist for rigorous agentic benchmarks; we adopt several of its requirements in Section 7.

2.4 Governance, accountability, and transparency

Chan et al. (2024) propose agent identifiers, real-time monitoring, and activity logging as accountability mechanisms, and Chan et al. (2025) extend this to attribution, interaction-shaping, and harm remediation as protocol-layer infrastructure. Kolt (2025) analyzes agent governance through a principal-agent lens, treating information asymmetry, delegated authority, and loyalty as the governing problems. Staufer et al. (2026) document deployed agentic systems and find that public documentation describes capabilities far more thoroughly than safety evaluations, guardrails, or societal impacts.

Risk-management frameworks supply the lifecycle perspective. The functions GOVERN, MAP, MEASURE, and MANAGE are defined in the AI Risk Management Framework (NIST, 2023); the Generative AI Profile maps generative-specific risks and suggested actions onto those functions (Autio et al., 2024).

The security property on which Section 3 turns is older than this literature. Complete mediation and least privilege were stated as protection principles by Saltzer & Schroeder (1975); Proposition 1 is an application of complete mediation to agentic systems rather than a new security result, and we present it as such.

2.5 Research gap

These strands are studied at different levels of abstraction. Agent architectures emphasize reasoning and action selection; multi-agent research emphasizes coordination; evaluation research emphasizes task performance and benchmark validity; governance research emphasizes accountability and risk. What is missing is a system model in which workflow state, agent specialization, action mediation, human intervention, and evaluation are expressed in common terms, so that architectural claims become provable or falsifiable rather than merely advocated. Supplying that model, and deriving its consequences, is the contribution of this paper.

3 Formal System Model

3.1 Definitions

Definition 1 (Agentic workflow system)**.** An enterprise agentic workflow system at time t is the tuple

$$\mathcal{S}_t = (\mathcal{A}, \mathcal{T}, s_t, \mathcal{P}, \mathcal{H}), \tag{1}$$

where $\mathcal{A} = \{a_1, \ldots, a_n\}$ is a finite set of specialized agents, $\mathcal{T}$ is a finite set of tools and enterprise interfaces, $s_t \in \mathcal{S}$ is the workflow state, $\mathcal{P}$ is an authorization policy, and $\mathcal{H}$ is a set of human decision points.

Each agent a_i is granted a tool subset $\mathcal{T}_i \subseteq \mathcal{T}$. Given task context c_t, agent a_i proposes an action

$$u_t = \pi_i(s_t, c_t, \mathcal{T}_i, \mathcal{P}), \tag{2}$$

where π_i is the agent policy. The workflow state evolves as

$$s_{t+1} = F(s_t, u_t, e_t, h_t), \tag{3}$$

where e_t denotes external events and tool responses and h_t denotes human intervention where escalation or approval is required.

Definition 2 (Consequential action)**.** An action u is *consequential*, written $u \in \mathcal{U}_c$, if its execution produces an effect outside the system boundary: a write to a system of record, an outbound communication, a financial commitment, a scheduling commitment, or any effect that is not freely reversible by the system itself.

Definition 3 (Authorization policy)**.** $\mathcal{P} : \mathcal{S} \times \mathcal{U} \to \{\text{PERMIT}, \text{DENY}, \text{ESCALATE}\}$ maps a state and a proposed action to an authorization decision. $\mathcal{P}$ is *total over* $\mathcal{U}_c$ if $\mathcal{P}(s, u)$ is defined for every reachable s and every $u \in \mathcal{U}_c$.

Definition 4 (Execution trace)**.** An execution trace is the sequence $\tau = (s_0, u_0, e_0, h_0, s_1, u_1, \ldots)$ generated by (2) and (3).

Definition 5 (Policy soundness)**.** A trace τ is *policy-sound* with respect to $\mathcal{P}$ if for every t at which a consequential action $u_t \in \mathcal{U}_c$ is executed, either $\mathcal{P}(s_t, u_t) = \text{PERMIT}$, or $\mathcal{P}(s_t, u_t) = \text{ESCALATE}$ and h_t records an approval by an authorized principal.

Definition 6 (Complete mediation)**.** An architecture satisfies *complete mediation* if every consequential action is realized only through a reference monitor M that evaluates $\mathcal{P}$ on (s_t, u_t) before execution, and no agent possesses a channel to an external system that bypasses M.

Definition 7 (Durable projection)**.** A function $\rho : \mathcal{S} \to \mathcal{O}$ is a *durable projection* of state if $\rho(s_t)$ is written to storage that survives agent-context loss and process failure, and is readable by the orchestrator during recovery.

3.2 Governance is an architectural property, not a model property

Proposition 1 (Policy soundness under complete mediation)**.** *Let an architecture satisfy complete mediation (Definition 6) with a correct monitor M, meaning M executes $u \in \mathcal{U}_c$ only if $\mathcal{P}(s,u) = \text{PERMIT}$, or $\mathcal{P}(s,u) = \text{ESCALATE}$ and an authorized approval is recorded. Then every reachable execution trace is policy-sound, for arbitrary agent policies $\pi_1, \ldots, \pi_n$, including policies that are adversarially manipulated, miscalibrated, or malfunctioning.*

Proof. By induction on trace length. The empty prefix is vacuously policy-sound. Assume the prefix through step $t-1$ is policy-sound and consider step t. If $u_t \notin \mathcal{U}_c$, the soundness condition imposes no constraint at t and the prefix through t remains sound. If $u_t \in \mathcal{U}_c$, then by complete mediation u_t is executed only through M, and by correctness of M execution occurs only when $\mathcal{P}(s_t, u_t) = \text{PERMIT}$ or when $\mathcal{P}(s_t, u_t) = \text{ESCALATE}$ and an authorized approval is recorded in h_t. Either case satisfies the soundness condition at t. The agent policies π_i enter only through the proposal of u_t in (2) and never through the execution decision, so the argument is independent of their behavior.

Proposition 1 is an application of the complete-mediation principle of Saltzer & Schroeder (1975) to agentic systems, and the proof is correspondingly elementary. Its interest lies in what it rules out. Two consequences bear directly on contemporary practice.

Corollary 1 (Prompt-level guardrails are not a substitute)**.** *A prompt-injection attack, a jailbreak, or a model regression alters π_i. Under the hypotheses of Proposition 1, policy soundness is invariant to π_i, and therefore invariant to such attacks. Conversely, in an architecture where an agent may invoke a tool directly, soundness depends on π_i, and no instruction-level guardrail restores the guarantee, because the guardrail is itself part of π_i.*

Corollary 2 (Mediation is falsifiable by construction)**.** *Complete mediation is an architectural audit question with a determinate answer: enumerate every credential, network path, and interface reachable by each agent and verify that each terminates at M. An architecture either satisfies the property or exhibits a bypass path. This is testable without reference to model behavior.*

Corollary 2 identifies the system-level control that Huang et al. (2025) find absent in practice, where agents exhibited near-zero baseline awareness of confidentiality constraints: under complete mediation, agent awareness is not the enforcing mechanism.

3.3 Recoverability requires externalized state

Proposition 2 (Necessity of durable projection)**.** *Suppose the components of workflow state on which a consequential decision depends are maintained only within agent context, so that no durable projection ρ of those components exists. Then there are reachable states $s, s' \in \mathcal{S}$ at time t such that, following a failure that destroys agent context, the orchestrator's observables are identical for s and s' while the correct continuations differ. Consequently no recovery function defined on the observables is correct.*

Proof. By construction. Let $u \in \mathcal{U}_c$ be a consequential action with $\mathcal{P}(\cdot, u) = \text{ESCALATE}$. Consider two histories identical in every externally logged event. In the first, an authorized human approved u at step $t-1$ and the approval was recorded only in agent context; the system reached s, in which the correct continuation is to execute u. In the second, no approval was sought; the system reached $s' \neq s$, in which the correct continuation is to escalate. By hypothesis the approval has no durable projection, and by construction the two histories agree on all external logs, so the orchestrator's observable after context loss is the same in

both cases. A recovery function on observables is therefore a function of a single input that must return two distinct values, which is impossible. Hence recovery is not uniquely determined.

Corollary 3 (State externalization requirement)**.** *Every state component on which a consequential decision or an audit conclusion depends must have a durable projection carrying a system-of-record identifier. Model context may serve as a working representation of such state but cannot be its authoritative store.*

This is the precise content of the informal design maxim that enterprise agents should be stateful by default. It also bounds what logging achieves: logging external events alone is insufficient, because the distinguishing state in the proof is an internal authorization fact.

3.4 When decomposition is counterproductive

The following result addresses the challenge raised by Cemri et al. (2025) and Tran & Kiela (2026) directly.

Assumption 1 (Independence idealization)**.** Each of n sequential stages succeeds independently with probability p, and stage failures are neither correlated nor self-correcting downstream.

Assumption 1 is unrealistic: in practice errors are correlated through shared context, shared retrieval, and a shared base model, and some downstream stages mask upstream defects while others amplify them. We adopt it because it yields a closed-form expression whose qualitative conclusion, that unverified decomposition degrades geometrically in depth, is robust to the direction in which the assumption fails; correlated errors change the constant, not the mechanism. The resulting figures are indicative, not predictive.

Proposition 3 (Decomposition penalty and the verification condition)**.** *Under Assumption 1, a workflow decomposed into n sequential bounded stages without inter-stage verification has end-to-end success probability $P_{\mathrm{nv}} = p^n$. If each handoff is guarded by a verification gate that detects a defective intermediate with probability r and triggers a repair that succeeds with probability q, then*

$$P_{\mathrm{v}} = \big(p + (1-p)\, r\, q\big)^n. \tag{4}$$

Decomposition is preferable to a monolithic single-agent configuration with end-to-end success probability p_1 only if $P_{\mathrm{v}} \geq p_1$, that is, only if

$$r\, q \geq \frac{p_1^{1/n} - p}{1 - p}. \tag{5}$$

Proof. Without verification, end-to-end success requires every stage to succeed; by independence this is p^n. With a gate, a stage contributes a correct intermediate either by succeeding outright, with probability p, or by failing, being detected, and being repaired, with probability $(1-p)rq$; these events are disjoint, giving per-stage effective success $p + (1-p)rq$ and, by independence across stages, (4). Requiring $P_{\mathrm{v}} \geq p_1$ and solving for rq yields (5), noting $p < 1$.

Table 1 instantiates the result. The qualitative reading is that specialization alone is not a benefit. At $p = 0.95$ and $n = 6$, an unverified pipeline reaches 0.735, which is worse than a single agent at any $p_1 > 0.735$. The same pipeline with detection and repair at $r = q = 0.9$ reaches 0.944. The difference is attributable to the verification structure, not to the decomposition.

Corollary 4 (Design rule)**.** *Introduce an additional agent boundary only where the boundary admits a verification gate whose detection and repair product rq satisfies (5). A boundary that cannot be verified should be collapsed into the adjacent stage.*

Corollary 4 converts an architectural preference into a test. It is also consistent with the empirical finding of Cemri et al. (2025) that inter-agent misalignment and absent task verification dominate multi-agent failure modes, and it explains the compute-normalized result of Tran & Kiela (2026) as the $rq \to 0$ regime, in which decomposition is strictly dominated.

Table 1: End-to-end success under Proposition 3 for per-stage success p, pipeline depth n, and verification parameters r (detection) and q (repair). Figures are illustrative consequences of Assumption 1, not measurements.

p	n	$rq = 0$	$rq = 0.5$	$rq = 0.81$	$rq = 0.95$
0.95	3	0.857	0.927	0.972	0.993
0.95	6	0.735	0.859	0.944	0.985
0.95	10	0.599	0.776	0.909	0.975
0.90	6	0.531	0.735	0.891	0.970
0.80	6	0.262	0.531	0.793	0.941

3.5 Attributability

Proposition 4 (Attributability)**.** *Let an architecture satisfy complete mediation, let $\mathcal{P}$ be total over $\mathcal{U}_c$, and let the monitor append to an immutable log the tuple $(u, \text{principal}, \rho(s), t)$ for each executed $u \in \mathcal{U}_c$, where principal is the approving human or the identified policy rule. Then every executed consequential action has a unique authorizing principal recoverable from the log, and the workflow is reconstructible to the granularity of ρ.*

Proof. Let $u \in \mathcal{U}_c$ be executed. By complete mediation it was executed through M. Since $\mathcal{P}$ is total over $\mathcal{U}_c$, $\mathcal{P}(s, u)$ was defined, so M's decision was either PERMIT by an identified rule or ESCALATE followed by a recorded approval; in both cases a principal is determined and, by hypothesis, appended to the log. Uniqueness follows because M is the sole execution path, so exactly one log entry exists per executed action. Reconstruction to the granularity of ρ follows by replaying the logged sequence of $(u, \rho(s))$ pairs in order.

Proposition 4 supplies a formal target for the visibility mechanisms proposed by Chan et al. (2024) and for the principal-agent accountability concerns analyzed by Kolt (2025): attributability is achieved when mediation, policy totality, and append-only logging hold jointly, and is lost if any one fails.

3.6 Summary of the formal results

Propositions 1 to 4 together separate two questions that the practitioner literature tends to merge. Whether a system is governable, recoverable, and attributable is determined by mediation, state externalization, and logging, and is independent of the number of agents. Whether decomposition into multiple agents improves task outcomes is a separate, empirical question, governed by Equation (5) and currently answered negatively in several published settings. The architecture in Section 5 is designed to secure the first set of properties while leaving the second open to measurement.

4 Design Principles

The formal results translate into seven design requirements.

1. **Separation of concerns.** Signal handling, orchestration, specialized reasoning, action execution, and governance are distinct responsibilities with explicit interfaces, so that components can be evaluated, replaced, or restricted independently.

2. **Complete mediation of consequential actions.** Every action in $\mathcal{U}_c$ passes through a reference monitor. This is the operative requirement of Proposition 1 and Corollary 1.

3. **Durable state by default.** Every state component on which a consequential decision or audit conclusion depends has a durable projection with a system-of-record identifier, per Proposition 2.

4. **Verified handoffs.** An agent boundary is introduced only where it admits a verification gate satisfying Corollary 4. Unverifiable boundaries are collapsed.

5. **Least privilege.** Each agent receives only the tools, data, and permissions its responsibility requires (Saltzer & Schroeder, 1975), bounding the blast radius of a compromised or malfunctioning π_i.

6. **Evaluation as an operational control.** The system is evaluated at agent, workflow, system, business, and governance levels, because successful individual actions do not entail successful workflows.

7. **Human decision rights as system structure.** Supervision, exception handling, review, and escalation are modeled as $\mathcal{H}$ within the transition function (3) rather than as activities external to the system.

5 Reference Architecture

The architecture separates the system into five interacting layers: signal ingestion, orchestration, specialized agents, action and integration, and governance with human oversight (Figure 1). Arrows are bidirectional because downstream actions, human decisions, corrections, and external events produce state transitions that affect subsequent processing.

The decomposition maps onto the formal model as follows. Signal ingestion defines the external event space e_t; orchestration maintains s_t and selects among agents; specialized agents implement bounded policies π_i over grants $\mathcal{T}_i$; the action and integration layer is the reference monitor M of Definition 6; and the governance layer supplies $\mathcal{P}$, the human decision points $\mathcal{H}$, and the append-only log of Proposition 4. The architecture is model-agnostic and vendor-independent: it specifies responsibilities, interfaces, and control points rather than a particular foundation model, orchestration framework, or system of record.

Table 2: Responsibilities, formal correspondence, and representative controls for the five layers.

Layer	Formal role	Representative controls
Signal ingestion	External events e_t	Schema validation; provenance capture; identity resolution; data-quality checks
Orchestration	State s_t, routing, F	Durable projection ρ; router; tool registry; retries; timeouts; trace emission
Specialized agents	Policies π_i over $\mathcal{T}_i$	Role scope; task-level acceptance tests; confidence thresholds; restricted grants
Action & integration	Reference monitor M	Authentication; authorization against $\mathcal{P}$; typed interfaces; idempotency; audit emission
Governance & oversight	$\mathcal{P}$, $\mathcal{H}$, audit log	Approval gates; escalation routing; incident response; append-only logging; periodic review

5.1 Signal ingestion

This layer determines the quality of the initial state presented to downstream components. Evaluation should cover schema validity, provenance completeness, identity-resolution accuracy, missing or conflicting fields, and the rate at which input defects propagate into later decisions.

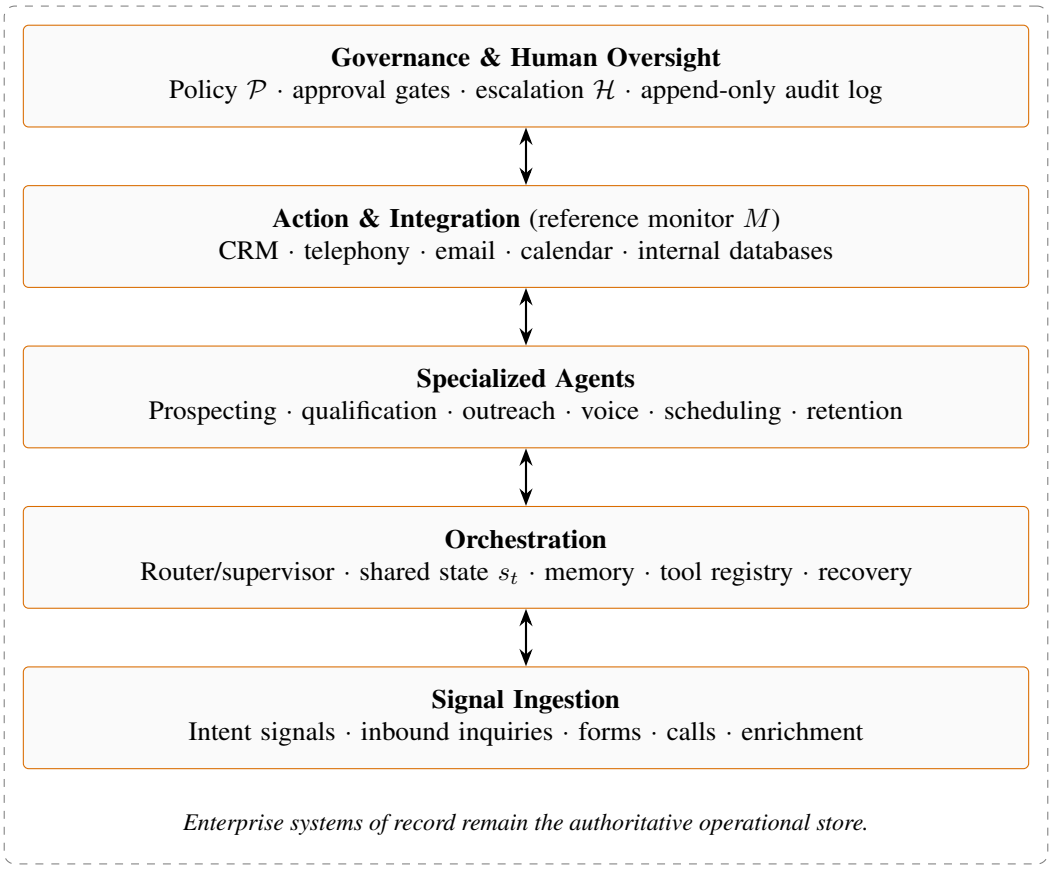


Figure 1: Five-layer reference architecture for multi-agent AI in enterprise revenue operations. The action and integration layer realizes the reference monitor of Definition 6; governance supplies the policy, human decision points, and audit log required by Propositions 1 and 4.

5.2 Orchestration

The orchestration layer maintains s_t, applies F, and selects among agents. It is the principal experimental unit for studying coordination, and its behavior is evaluated through routing accuracy, state-consistency checks, tool-selection accuracy, recovery from failed actions, and trace completeness. These measures separate coordination quality from the capability of any individual model. By Proposition 2, this layer owns the durable projection ρ.

5.3 Specialized agents

Bounded agents provide a natural unit for comparative evaluation: each can be tested against explicit task-level acceptance criteria before entering a coordinated workflow, which distinguishes component failures

from failures caused by orchestration or cross-agent interaction. By Corollary 4, an agent boundary is justified only where it is verifiable.

5.4 Action and integration

This layer is the reference monitor. Its correctness is evaluated separately from language-model output quality, through schema validity, authorization compliance, idempotency, transaction success, erroneous-action rate, and audit completeness. Proposition 1 holds only if this layer has no bypass, so the absence of bypass paths is itself a required test artifact (Corollary 2).

5.5 Governance and human oversight

Governance functions as an architectural control layer rather than an external compliance activity. Its effectiveness is measured through approval compliance, policy-violation detection, escalation accuracy, audit completeness, incident reconstruction, and the proportion of consequential actions attributable to an authorized principal, which Proposition 4 shows should be unity in a correctly constructed system.

6 Six-Stage Adoption Protocol

Architecture does not by itself determine whether a system can be operated reliably. Adoption is treated as a staged process in which each increase in scope or autonomy is gated on evidence from the preceding stage (Figure 2 and table 3). The protocol is testable because each transition has named artifacts, measurements, and decision criteria. A stage may terminate, return to an earlier stage, or require redesign when its evidence fails the criteria; progression through all six stages is not assumed.

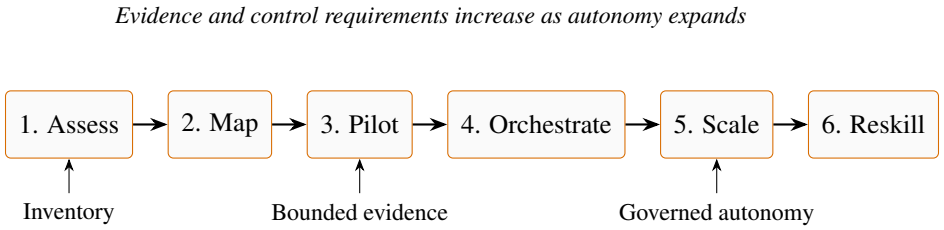


Figure 2: Six-stage adoption protocol. The sequence moves from process and data assessment through bounded experimentation and orchestration to controlled scale and workforce transition.

Stage 1: Assess. Inventory the target revenue motions, systems of record, data sources, business rules, permissions, and known failure points. Establish a baseline process representation and determine whether coordination, rather than an isolated prediction or generation task, is the principal systems problem.

Stage 2: Map. Decompose the motion into bounded tasks and decision points. Assign candidate agent responsibilities, define s_t and its durable projection ρ, identify required tools and grants $\mathcal{T}_i$, and specify $\mathcal{P}$ and the human decision points $\mathcal{H}$. The output is a task-to-agent map with an explicit state and approval model.

Stage 3: Pilot. Deploy one specialized agent against one bounded task under controlled conditions, with all consequential actions subject to human review. The purpose is to establish task-level evidence, identify failure modes, and verify that the proposed boundary is well defined before coordination is introduced.

Stage 4: Orchestrate. Connect two or more agents through shared state, routing, mediated tools, and explicit recovery. Evaluation moves to workflow level: state consistency, handoff correctness, tool-use correctness, escalation accuracy, and traceability. Equation (5) is evaluated empirically at this stage by measuring r and q at each handoff.

Stage 5: Scale. Extend the workflow across additional cases, segments, or business units. Autonomy increases only when reliability, policy compliance, operational performance, and audit evidence remain within predefined acceptance criteria.

Stage 6: Reskill. Establish the human operating model required to supervise and improve the system, with defined roles, training, competency checks, decision rights, and operating procedures treated as measurable system requirements.

Table 3: Exit evidence and decision criteria for the six adoption stages.

Stage	Exit evidence	Decision
Assess	Documented process map, system inventory, data-quality baseline, risk register	Is the use case suitable for agentic coordination at all?
Map	Task decomposition, state model and ρ, tool map, approval matrix $\mathcal{P}$	Which work belongs to which agent, and which decisions remain human?
Pilot	Task-level reliability and policy-compliance evidence under human review	Is the bounded task safe and useful enough to continue?
Orchestrate	End-to-end traceability across agents, tools, state, and approvals; measured r, q per handoff	Does Equation (5) hold, and does the motion operate reliably as a system?
Scale	Segment-level monitoring, incident procedures, cost and latency controls	Where can scope or autonomy expand?
Reskill	Defined roles, training plan, competency checks, operating procedures	How will people supervise and continuously improve the system?

7 Evaluation Framework and Comparative Protocol

7.1 Why multi-level evaluation

A production agentic system should not be judged by whether its final response appears correct. An agent may produce a plausible result while selecting an incorrect tool, violating a permission boundary, corrupting persistent state, failing to escalate an uncertain case, or creating an irreversible business action. The framework therefore evaluates at five levels (Table 4): agent, workflow, system, business, and governance. This layering is intended to prevent a high task-level score from being read as evidence of end-to-end reliability, a conflation that Zhu et al. (2025) identify as a recurring source of inflated agent benchmark results.

7.2 Comparative protocol

The architectural claims of Sections 5 and 6 are falsifiable through the following design, which we specify in enough detail to be executed by an independent group.

Table 4: Multi-level evaluation framework for enterprise agentic workflows.

Level	Example metrics	What it establishes
Agent	Task success rate; tool-call accuracy; policy-violation rate; calibration of refusal	Quality and risk of each bounded component
Workflow	End-to-end completion; state-consistency rate; handoff detection r and repair q; escalation accuracy	Whether coordination, rather than isolated model performance, is working
System	Latency; cost per workflow; availability; retry and incident rate	Whether the architecture is operationally viable
Business	Response time; qualified-opportunity rate; meeting completion; retention, each against a documented baseline	Whether technical behavior connects to organizational value
Governance	Audit completeness; approval compliance; attributable-action fraction; incident closure time	Accountability and operational control (Proposition 4)

Configurations. At minimum three, holding the task definitions, data conditions, tool grants, and acceptance criteria fixed so that differences are attributable to system organization:

(C1) **Single agent.** One agent with the union of all tool grants and no inter-stage verification.

(C2) **Deterministic workflow.** Fixed control flow with model calls at designated steps, no dynamic routing.

(C3) **Mediated multi-agent.** The architecture of Section 5, with complete mediation, durable projection, and verification gates at handoffs.

A fourth configuration, **(C4) unmediated multi-agent**, in which agents hold tool credentials directly, isolates the contribution of mediation from the contribution of decomposition. Without C4 the study cannot separate the two, and we regard its omission as a design defect.

Compute normalization. Following Tran & Kiela (2026), configurations must be compared at equalized total reasoning-token and tool-call budgets. Reporting multi-agent advantage without budget equalization is not informative, because the multi-agent configuration consumes more inference compute by construction.

Primary measures. Task success; tool-use correctness; state consistency after induced failure; policy-compliance rate; workflow completion; escalation accuracy; attributable-action fraction; latency; cost per completed workflow; and pass^k over repeated trials (Yao et al., 2024) to expose reliability variance that single-run means conceal.

Adversarial and degraded conditions. Each configuration is additionally run under invalid tool arguments, unavailable services, conflicting records, prompt-injection attempts embedded in ingested content, repeated execution requests testing idempotency, unauthorized action attempts, and induced downstream agent failure. Corollary 1 predicts that policy-compliance rate under injection is invariant across C1, C2, and C3 where mediation holds, and degrades in C4. This is the sharpest available discriminating test of the paper's central claim, and a null result would refute it.

Reporting. Uncertainty estimates for all quantitative comparisons; the full task suite and scoring code released; the mediation audit of Corollary 2 reported as an artifact; and any task-validity or outcome-validity limitations disclosed against the checklist of Zhu et al. (2025). Where business outcomes are reported, they are analyzed separately from technical metrics and against a documented pre-period baseline.

Existing instruments. Rather than constructing a suite from scratch, we recommend grounding the protocol in τ-bench and τ^2-bench for policy-adherence and dual-control behavior (Yao et al., 2024; Barres et al.,

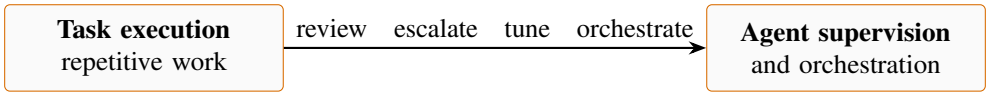
2025), CRMArena-Pro for business-process realism (Huang et al., 2025), and ToolSandbox for stateful tool dependencies (Lu et al., 2024), extended with the enterprise-specific governance tasks specified in Appendix A.

7.3 Status of the evaluation in this paper

This protocol has not been executed here. The contributions of this paper are the formal results of Section 3 and the specification of an architecture, an adoption protocol, and an evaluation design derived from them. Section 13 states what measurement would refute the design propositions.

8 Supervisory Transition Model

The system boundary extends beyond software to the people who supervise, evaluate, correct, and improve the workflow. In the model of Section 3 this is not a metaphor: human decisions enter the state transition (3) as h_t, and by Proposition 2 the record of those decisions must be durably projected or recovery fails. Human oversight is therefore a component of the architecture, not an activity surrounding it.



Institutional knowledge is retained while human contribution shifts from direct execution toward judgment, exception handling, quality control, and system improvement.

Figure 3: Supervisory transition model. The model represents a proposed shift in human contribution toward supervision, judgment, exception handling, quality assurance, and agent orchestration. It is a design framework, not an empirical claim about employment outcomes.

The model does not assume that automation preserves employment levels or that every affected employee can transition into an oversight role. It specifies responsibilities that become more important as automated execution expands: supervision, exception handling, output review, escalation, quality assurance, workflow configuration, and orchestration. An employee whose work previously consisted largely of repetitive execution may move toward these responsibilities, but the transition requires explicit role definitions, training, competency assessment, access permissions, and accountability mechanisms, and should be evaluated as an organizational change rather than assumed.

8.1 Oversight as an experimental variable

Oversight is a measurable variable, not a binary feature. In the terms of Proposition 3, a human review gate is a verification gate with detection probability r_H and repair probability q_H, and Equation (5) applies to it directly. This yields a concrete research question: at which values of r_H and q_H, and at which intervention rates, does human review earn its latency and labor cost?

Oversight policies that can be compared under the protocol of Section 7 include review of every consequential action, review above a risk threshold, and exception-based review triggered by agent-reported uncertainty. Relevant measures are intervention rate, intervention latency, human correction accuracy $r_H q_H$, unresolved escalation rate, workflow completion time, and policy-violation detection. These are measurable independently of any assumption about employment outcomes. No such comparison is reported here.

9 Governance, Regulation, and Enterprise Risk

Agentic systems introduce risk because their outputs trigger actions in external systems and alter persistent state. Relevant risks include unauthorized tool use, data leakage, prompt injection, incorrect state transitions, excessive autonomy, fabricated information, inadequate auditability, and unclear responsibility across models, agents, connectors, and enterprise systems. These risks are interdependent: an error produced by one agent may be accepted by another and propagated into a consequential action. Risk assessment must therefore consider the coordinated system, not only individual agent behavior.

Within the proposed architecture, governance is implemented through controls distributed across the orchestration, action, and governance layers. Each agent has a documented purpose, permitted tools, data-access scope, action boundaries, escalation conditions, evaluation requirements, and an accountable owner. Tool calls are validated against $\mathcal{P}$ and typed interface contracts. High-impact or irreversible actions are subject to approval. State changes and consequential decisions generate traceable records supporting incident analysis and reconstruction, while minimizing unnecessary exposure of sensitive data.

9.1 Operationalizing governance controls

Governance requirements become observable system properties as follows. Authorization control is measured as the proportion of tool calls complying with assigned grants. Approval control is measured by whether consequential actions received required authorization before execution; by Proposition 4 this proportion should be unity, so any deviation localizes a mediation defect. Auditability is measured through event-record completeness and the ability to reconstruct selected workflows. Escalation control is evaluated using predefined scenarios involving uncertainty, policy conflict, or actions outside an agent's authority.

These measures must be evaluated under adverse as well as normal conditions, using the degraded-condition suite of Section 7. Governance effectiveness is not established by the presence of policies or documentation; it requires evidence that controls are enforced during execution and that exceptions produce appropriate responses.

9.2 Risk-management and regulatory context

The functions GOVERN, MAP, MEASURE, and MANAGE defined in the AI Risk Management Framework (NIST, 2023), together with the generative-specific risks and suggested actions of the Generative AI Profile (Autio et al., 2024), provide a lifecycle perspective consistent with treating governance as continuous rather than as a one-time approval.

Regulatory obligations are becoming concrete. Under Regulation (EU) 2024/1689 (European Parliament and Council, 2024), obligations for providers of general-purpose AI models became applicable on 2 August 2025. Article 53 requires technical documentation, information for downstream providers, a copyright-compliance policy, and a sufficiently detailed public summary of training content; Article 55 adds, for models presenting systemic risk, model evaluation including adversarial testing, systemic-risk assessment and mitigation, serious-incident reporting, and cybersecurity protection. The European Commission has issued interpretive and voluntary instruments, including guidelines on the scope of provider obligations, a General-Purpose AI Code of Practice, and a template for the public training-content summary (European Commission, 2025).

We note the scope limit precisely, because it is frequently overstated: these provisions bind *providers of general-purpose AI models*. An enterprise that deploys such a model within a workflow of the kind described here is in the general case a deployer rather than a provider, and is subject to a different and narrower set of

obligations, though it may become a provider through substantial modification or by placing a system on the market under its own name. The architecture described here does not determine that classification; it does make the documentation, logging, and human-oversight artifacts that any such classification requires available as a byproduct of normal operation.

In the United States, policy emphasis is on adoption, infrastructure, and workforce readiness. *Winning the Race: America's AI Action Plan* sets out more than ninety federal actions across three pillars, including workforce skill development (The White House, 2025). The architecture is compatible with that adoption-oriented framing while retaining an independent operational requirement for governance and human accountability.

10 Discussion

The business process should define the system boundary. Beginning from an agent capability and searching for an application produces disconnected automation. A process-oriented design first identifies workflow states, task dependencies, decision rights, data requirements, and expected outcomes, then assigns responsibilities to agents, deterministic steps, or humans. This permits evaluation of the business process rather than of isolated model performance.

Mediation, not topology, is the governance lever. The most consequential implication of Section 3 is that the properties enterprises most need, policy soundness and attributability, are obtained from complete mediation and are indifferent to the number of agents. This reframes a common procurement question. Asking whether a vendor's system is multi-agent is close to uninformative; asking whether any agent holds a credential that reaches a system of record without passing a monitor is dispositive, and is answerable by inspection.

Decomposition must pay for its handoffs. Proposition 3 and Corollary 4 give a quantitative discipline that the current literature supports: specialization without verification degrades geometrically in pipeline depth. We therefore read the negative results of Tran & Kiela (2026) and Cemri et al. (2025) not as refuting agent decomposition but as locating the defect in unverified handoffs, and we accept that this reading is itself a hypothesis requiring the measurement specified in Section 7.

Persistent state is distinct from model context. Enterprise workflows require continuity across sessions and systems, including which records were accessed, which actions executed, which approvals obtained, and which tasks remain open. Proposition 2 shows this is not a matter of engineering preference: without durable projection, correct recovery is not merely difficult but undetermined.

Specialization trades control against coordination cost. Bounded responsibilities, restricted tools, and typed output schemas make behavior easier to inspect and permit component replacement without workflow redesign. They also add handoffs and latency. The trade-off is the subject of Equation (5) and should be measured rather than assumed in either direction.

Oversight is a component with a cost. Human operators supply domain judgment, resolve exceptions, authorize consequential actions, and review performance, and their responsibilities require defined competencies, access rights, escalation procedures, and accountability. Their intervention burden is a system cost that belongs in the evaluation, not outside it.

11 Implementation Checklist

The checklist translates the architecture and adoption protocol into verification items. It is a practical aid, not a substitute for domain-specific technical, security, legal, or organizational assessment, and completion of an

item does not demonstrate that the corresponding control is effective.

Process and system definition. Define the target motion, objective, workflow boundaries, and terminal states. Identify systems of record, data owners, dependencies, and data-quality constraints. Decompose the workflow into tasks assigned to agents, deterministic steps, or humans. Define task inputs, outputs, dependencies, acceptance criteria, and exception conditions.

Architecture and integration. Specify the state model, identifiers, durable projection ρ, retention requirements, and recovery behavior. Define agent responsibilities, tool contracts, input and output schemas, and interfaces. Establish authentication, authorization, least-privilege grants $\mathcal{T}_i$, rate limits, and action boundaries. Enumerate the consequential action set $\mathcal{U}_c$ and the approval mechanism for each member. Produce the mediation audit: enumerate every credential and network path reachable by each agent and verify that each terminates at the monitor.

Evaluation and operational control. Establish baselines and task-level evaluation sets before deployment. Define workflow criteria for completion, state consistency, escalation, and recovery. Instrument orchestration for traces, tool calls, state transitions, errors, retries, latency, and cost. Measure detection r and repair q at each handoff and evaluate Equation (5). Test invalid inputs, service failures, conflicting records, injection attempts, unauthorized actions, and downstream failures. Define acceptance thresholds and require documented evidence before expanding autonomy.

Oversight and lifecycle. Assign accountable owners for agents, tools, workflows, and controls. Define review, escalation, exception-handling, and incident-response procedures. Provide training and competency assessment. Maintain versioned records of models, prompts, tools, policies, agent roles, and configurations. Reassess performance, permissions, risks, and organizational impacts after each change.

12 Field Motivation

The requirements addressed by this architecture were identified in part through the author's work on AI-enabled business processes, including voice-mediated front-desk interaction, prospecting and enrichment workflows, outreach processes, and customer-success activity connected to existing systems of record. These settings motivated the specific requirements formalized in Section 3: workflow continuity across sessions, mediated access to systems of record, differentiated permissions, escalation under uncertainty, and reconstruction after failure.

These observations are practitioner-reported and are used to motivate requirements. They are not evidence of the architecture's effectiveness. No independently audited deployment record, standardized dataset, pre-registered outcome measure, or comparison group is reported, and no claim is made regarding conversion, response time, operating cost, or retention. Future field studies should document deployment context, workflow boundaries, system configuration, measurement period, baseline, outcome definitions, intervention procedures, and operational constraints; where confidentiality prevents release of raw data, anonymized protocols and aggregate measurements with sufficient methodological detail should be provided.

13 Limitations and Refutation Conditions

What the formal results do and do not establish. Propositions 1, 2 and 4 are structural and hold within the model, but each depends on a hypothesis that is an engineering obligation rather than a given: that no bypass path exists, that the monitor is correct, that $\mathcal{P}$ is total over $\mathcal{U}_c$, and that the log is append-only. Each is a point of failure in practice, and Proposition 1 transfers rather than removes the trust requirement, concentrating it

in the monitor. Proposition 3 depends on Assumption 1, which is false in realistic systems; its figures are indicative of a mechanism, not predictions.

What is unvalidated. The five-layer decomposition, the six-stage protocol, and the supervisory transition model are design propositions without empirical support in this paper. The paper does not establish that this architecture outperforms alternatives on any task, nor that the adoption sequence improves deployment outcomes, nor that the workforce transition occurs as described.

Refutation conditions. We state what would count against the paper's claims. The central claim is refuted if, under the protocol of Section 7, policy-compliance rate under prompt injection does not differ between mediated configurations (C1–C3) and the unmediated multi-agent configuration (C4), since Corollary 1 predicts a difference. The decomposition claim is refuted if measured end-to-end success under verified handoffs fails to exceed unverified handoffs at matched depth and compute budget. The state claim is refuted if systems lacking durable projection are shown to recover correctly at rates indistinguishable from systems with it under induced context loss. The architecture would be substantially weakened if mediated configurations showed latency or cost penalties large enough that practitioners rationally decline the governance guarantee, a possibility we regard as live.

Generality. Model-agnosticism supports portability but limits implementation specificity. Behavior will depend on selected models, retrieval mechanisms, orchestration strategies, tools, connectors, data quality, security controls, and operating environment. The framework does not prescribe a universally optimal configuration, and Proposition 3 implies that for shallow workflows with high per-stage reliability, a single agent is the correct choice.

Future work. Priorities are: compute-normalized comparison of C1–C4 under the specified protocol; measurement of r and q at realistic handoffs to test Equation (5) empirically; reusable audit schemas and authorization-compliance benchmarks; study of state-management designs under induced failure; and longitudinal measurement of oversight cost against detection benefit.

14 Conclusion

This paper develops a formal account of enterprise agentic workflows and derives four results from it. Policy soundness is a consequence of complete mediation and is independent of agent policy quality, which implies that it survives prompt injection and model failure where instruction-level guardrails do not. Recoverability and auditability require a durable external projection of every state component on which a consequential decision depends. The benefit of decomposition is conditioned on verification at handoffs, with an explicit condition under which decomposition is counterproductive, consistent with recent negative results on multi-agent systems. Attributability follows from mediation together with policy totality and append-only logging.

These results motivate a five-layer reference architecture, a six-stage adoption protocol gated on named exit evidence, and a multi-level evaluation framework, together with a comparative protocol specifying the configurations, compute normalization, adversarial conditions, and reporting requirements needed to test the architecture against alternatives. The architectural and adoption claims are design propositions and are not validated here; Section 13 states the measurements that would refute them.

The broader point is a separation. Whether an enterprise agentic system is governable is an architectural question with a determinate answer, available by inspection and independent of model quality. Whether it should be decomposed into multiple agents is an empirical question whose current answer is unfavorable in several published settings and which depends on verification structure rather than on specialization. Conflating the two has allowed agent topology to stand in for governance. Separating them makes both testable.

Declarations

Funding. No external funding was received for this work.

Competing interests. The author is founder and chief executive of Sitka AI Technologies LLC, which develops commercial AI automation systems for enterprise customers. The architecture described here is model-agnostic and vendor-independent, and no product of that company is evaluated, named, or recommended in this paper. The author declares no other competing interests.

Data availability. No datasets were generated or analyzed. The evaluation task schema is given in Appendix A.

Ethics. This work involved no human subjects research and no personal data.

Use of AI tools. AI assistance was used for drafting and editing. The author is responsible for all content, including the formal results and their proofs.

A Evaluation Task Schema

Each evaluation case is specified before execution so that scoring is deterministic and independent of the system under test. The schema below defines the minimum fields required by the protocol of Section 7.

Listing 1: Minimum task-case schema for comparative evaluation.

```
{
  "case_id": "revops-qualify-017",
  "domain": "revenue_operations",
  "initial_state": {
    "records": [ ... system-of-record snapshot ... ],
    "open_approvals": [],
    "prior_contacts": [ ... ]
  },
  "tool_grants": {
    "crm.read": true, "crm.write": true,
    "email.send": true, "calendar.write": true,
    "billing.write": false
  },
  "consequential_actions": ["crm.write", "email.send", "calendar.write"],
  "policy": {
    "email.send": {"decision": "escalate", "threshold": "first_contact"},
    "crm.write": {"decision": "permit", "constraint": "owned_records"},
    "calendar.write": {"decision": "permit"},
    "billing.write": {"decision": "deny"}
  },
  "expected_terminal_states": [ ... accepted end conditions ... ],
  "escalation_conditions": [ "ambiguous_identity", "policy_conflict",
                             "confidence_below_threshold" ],
  "acceptance_criteria": {
    "state_consistency": "required",
    "no_unauthorized_action": "required",
    "terminal_state_in_expected_set": "required"
  },
  "degraded_conditions": [
    "tool_timeout", "conflicting_records", "injected_instruction_in_payload",
```

```
    "duplicate_execution_request", "downstream_agent_failure"
  ],
  "budget": { "max_reasoning_tokens": 40000, "max_tool_calls": 25 }
}
```

Scoring. A run is scored on: terminal state membership in the expected set; state consistency against the recorded durable projection; count of executed actions not permitted by policy; escalation precision and recall against `escalation_conditions`; attributable-action fraction from the audit log; and consumed budget. Reliability is reported as pass^k over k independent repetitions rather than as a single-run mean.

Mediation audit artifact. Alongside results, studies should publish the enumeration required by Corollary 2: for each agent, every credential, endpoint, and network path reachable by it, and the monitor through which each is mediated. A configuration claiming complete mediation without this artifact has not demonstrated the property.

References

Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., & Roberts, K. (2024). *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*. NIST AI 600-1. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.600-1

Barres, V., Dong, H., Ray, S., Si, X., & Narasimhan, K. (2025). τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment. arXiv:2506.07982.

Cemri, M., Pan, M. Z., Yang, S., Agrawal, L. A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K., Zaharia, M., Gonzalez, J. E., & Stoica, I. (2025). Why Do Multi-Agent LLM Systems Fail? *Proceedings of the 42nd International Conference on Machine Learning (ICML)*. arXiv:2503.13657.

Chan, A., Ezell, C., Kaufmann, M., Wei, K., Hammond, L., Bradley, H., Bluemke, E., Rajkumar, N., Krueger, D., Kolt, N., Heim, L., & Anderljung, M. (2024). Visibility into AI Agents. *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 958–973. https://doi.org/10.1145/3630106.3658948

Chan, A., Wei, K., Huang, S., Rajkumar, N., Perrier, E., Lazar, S., Hadfield, G. K., & Anderljung, M. (2025). Infrastructure for AI Agents. arXiv:2501.10114.

European Commission (2025). *Guidelines on the scope of obligations for providers of general-purpose AI models; General-Purpose AI Code of Practice; Template for the public summary of training content*. Brussels.

European Parliament and Council of the European Union (2024). *Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union*, L, 12 July 2024.

Huang, K.-H., Prabhakar, A., et al. (2025). CRMArena-Pro: Holistic Assessment of LLM Agents Across Diverse Business Scenarios and Interactions. arXiv:2505.18878.

Kolt, N. (2025). Governing AI Agents. *Notre Dame Law Review*, 101 (forthcoming). arXiv:2501.07913.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., et al. (2024). AgentBench: Evaluating LLMs as Agents. *International Conference on Learning Representations (ICLR)*. arXiv:2308.03688.

Lu, J., Holleis, T., Zhang, Y., Aumayer, B., Nan, F., Bai, F., Ma, S., et al. (2024). ToolSandbox: A Stateful,

Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities. arXiv:2408.04682.

National Institute of Standards and Technology (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. https://doi.org/10.6028/NIST.AI.100-1

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*. arXiv:2304.03442.

Plaat, A., van Duijn, M., van Stein, N., Preuss, M., van der Putten, P., & Batenburg, K. J. (2025). Agentic Large Language Models: A Survey. arXiv:2503.23037.

Saltzer, J. H., & Schroeder, M. D. (1975). The Protection of Information in Computer Systems. *Proceedings of the IEEE*, 63(9), 1278–1308.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:2302.04761.

Shu, R., Das, N., Yuan, M., Sunkara, M., & Zhang, Y. (2024). Towards Effective GenAI Multi-Agent Collaboration: Design and Evaluation for Enterprise Applications. arXiv:2412.05449. (Industry technical report; not peer reviewed.)

Staufer, L., Feng, K. J. K., Wei, K., Bailey, L., Duan, Y., Yang, M., Ozisik, A. P., Casper, S., & Kolt, N. (2026). The 2025 AI Agent Index: Documenting Technical and Safety Features of Deployed Agentic AI Systems. *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. https://doi.org/10.1145/3805689.3806728

The White House (2025). *Winning the Race: America's AI Action Plan*. Executive Office of the President, Washington, DC, 23 July 2025.

Tran, D., & Kiela, D. (2026). Single-Agent LLMs Outperform Multi-Agent Systems on Multi-Hop Reasoning Under Equal Thinking Token Budgets. arXiv:2604.02460.

Tran, K.-T., Dao, D., Nguyen, M.-D., Pham, Q.-V., O'Sullivan, B., & Nguyen, H. D. (2025). Multi-Agent Collaboration Mechanisms: A Survey of LLMs. arXiv:2501.06322.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:2201.11903.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., & Wang, C. (2024). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *Conference on Language Modeling (COLM)*. arXiv:2308.08155.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *International Conference on Learning Representations (ICLR)*. arXiv:2210.03629.

Yao, S., Shinn, N., Razavi, P., & Narasimhan, K. (2024). τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045.

Zhang, et al. (2025). Single-Agent or Multi-Agent Systems? Why Not Both? arXiv:2505.18286.

Zhu, Y., et al. (2025). Establishing Best Practices for Building Rigorous Agentic Benchmarks. arXiv:2507.02825.