Intent-Governed Loops for Accountable Agentic AI

Christoforus Yoga Haryanto

[1]ZipThought
Ground Floor, 470 St Kilda Rd.
Melbourne, VIC 3004, Australia
cyharyanto@zipthought.com.au

Abstract

Agentic LLM-based systems are now operating with tool access and institutional authority in finance, clinical triage, municipal policy, and similar high-liability domains. We argue that verifiable accountability in such settings requires architectural support beyond alignment: every emitted action must be provably authorized by an explicit mandate, within declared scope and constraints, based on current evidence, and escalated to named authority when outside mandate. We propose the intent-governed loop as a conceptual architecture that provides structural mechanisms toward these accountability properties through runtime control. This position paper articulates the core components: an Intent object (human context, symbolic constraints, semantic guidance); a Planner that proposes actions with structured justification; a dual-mode Enforcer that deterministically checks symbolic constraints then semantically evaluates boundary cases; and a temporal governance graph that records provenance, constraint evaluation, temporal coherence, and escalation. We outline key loop-level invariants, identify the non-optional architectural principles any implementation must satisfy, and propose an evaluation agenda including synthetic benchmarks, metrics, and adversarial stress tests to guide future empirical validation. Our position is that such an architecture represents necessary structural support for accountable deployment in high-liability domains, and we identify key architectural questions requiring further analysis before implementation.

Introduction

In October 2025, the Australian government accepted policy recommendations from a $440,000 Deloitte report later found to contain AI-generated fabricated citations, irrelevant references, and what academics called “gobbledegook.” Its errors propagated unchecked into institutional action because no runtime verification system existed to enforce admissibility before recommendations reached decision-makers (Karp 2025). This incident is not an anomaly but a symptom of fundamental architectural deficits in agentic systems. Large-scale benchmarks reveal systemic failures rooted in this same lack of runtime control. Analysis of state-of-the-art multi-agent systems reveals failure rates between 41% and 86.7%, with premature termination, context loss, information withholding, and inadequate verification, all of which are symptoms of architectures lacking enforced admissibility boundaries, temporal coherence checks, and mandatory escalation mechanisms (Cemri et al. 2025). While WebArena agents have improved dramatically from 14.41% in the original benchmark (Zhou et al. 2023) to roughly 64–67% task success in September–October 2025 according to self-reported agent leaderboards (WebArena Team 2025), they still fall short of 78.24% human performance. These task success metrics do not capture the architectural failure modes that persist even in high-performing systems. AgentBench identifies poor long-term reasoning, decision-making, and instruction following as primary obstacles to usable LLM agents (Liu et al. 2023). On complex labor-intensive tasks in WebChoreArena, agents frequently exhibit “Forgetting Instructions” failures that an admissibility enforcer would mechanically prevent, and “Task Limit Exceeded” errors where agents get trapped in repetitive loops when blocked rather than escalating to human authority (Miyai et al. 2025). Even when AI systems successfully satisfy stated human intent, they can converge toward strategies that optimize for predictability by systematically depleting user autonomy, which is invisible to traditional alignment paradigms (Mitelut, Smith, and Vamplew 2024). LLM agents lack execution-time authorization: planning errors propagate unchecked to tool invocation, triggering irreversible actions without runtime verification (Su et al. 2025). As LLM-driven agents now issue instructions with direct legal, fiduciary, clinical, or civic effects, helpfulness is insufficient. The system instead requires admissibility: each action must be provably under a live mandate, within explicit scope and constraints, based on current evidence, and escalated with justification if outside mandate.

Research Questions

We study four system-level properties required for deployment in safety-critical and high-liability domains where actions carry legal, fiduciary, clinical, safety, or civic consequences: 1. *Authorization fidelity*: can each proposed action be linked to an explicit declared intent rather than an inferred or self-expanded goal? 2. *Temporal admissibility*: can the system prevent reuse of reasoning that was valid but is now stale due to world change or policy change? 3. *Accountable* *escalation*: when an action exceeds authorized scope, can the system halt and route responsibility to a named human authority with structured justification? 4. *Auditability*: can an auditor reconstruct why a specific recommendation was issued, under whose mandate, under which constraints, on which evidence?

This is an architectural specification and evaluation agenda. We define runtime invariants and propose synthetic benchmark scenarios and loop-level metrics for accountable agentic AI deployment in high-liability domains. This paper presents a conceptual architecture and research agenda. It articulates the necessary components, invariants, and evaluation criteria for accountable agentic systems. The implementation and empirical validation of this framework are critical future work for the field.

Gap in Current Approaches

Existing directions partially address safety, continuity, or oversight, but none enforces the above properties at the level of concrete actions in live institutional contexts.

RLHF and instruction tuning achieve alignment—optimizing model outputs to human preference (Ouyang et al. 2022)—improving generalization under distribution shift while inducing conformity pressure and behavioral diversity collapse (Kirk et al. 2023; Zhu et al. 2025). Alignment induces behavioral tendencies but does not provide proof that a proposed hire/triage/policy recommendation is still in-scope, constraint-satisfying, or temporally valid at emission time. We distinguish alignment from admissibility: alignment optimizes model behavior toward human preferences; admissibility proves each action is authorized by explicit mandate. Alignment shapes tendencies; admissibility encodes authorization boundaries.

Memory-augmented agents such as MemGPT treat the agent as a persistent process with tiered recall (Packer et al. 2023), and episodic memory proposals enable instance-specific context reinstatement (Pink et al. 2025). These solve continuity but not mandate binding that forces the agent to demonstrate its next act is still authorized and supported by current facts.

Runtime guardrails wrap agents with multistage monitors that track evolving risk and dynamically block unsafe behavior, sometimes personalized to user vulnerability (Wu et al. 2025). This addresses longitudinal harm but not institutional duty: a response can pass a personalized harm filter and still breach fiduciary covenant, exceed delegated authority, or violate statutory cap.

Governance-first architectures argue that agent behavior must be engineered as governed infrastructure with explicit supervisory layers and constitutional constraints (Xu et al. 2025). Runtime governance frameworks like MI9 introduce continuous authorization and graded containment (Wang et al. 2025), while Policy Cards propose machine-readable constraints and human-in-the-loop approval gating (Mavracic 2025). However, these do not enforce, at the level of each surfaced act, a live binding to an externally-authored Intent with expiry, named authority, and mandatory per-act admissibility under current evidence.

Architectures for AI in regulated and safety-critical domains must often operate under principles of bounded autonomy and verifiable human control (Perez-Cerrolaza et al. 2024). This requirement moves beyond simple supervision toward collaborative frameworks like human-machine teaming, where human oversight is integrated into the system's operational logic (Tsamados, Floridi, and Taddeo 2025). Furthermore, temporal knowledge graph work emphasizes that such oversight depends on the provenance and freshness of evidence (Cai et al. 2024).

In summary, alignment optimizes for behavior, memory systems optimize for continuity, guardrails optimize for harm reduction, and governance frameworks optimize for oversight. Still missing is a live contract between institutional intent and each specific downstream action.

Core Definitions

To make that contract explicit, we introduce four primitives.

Intent An Intent is an externally declared mandate authored by an accountable human or institution. It specifies three components that form a multi-modal authorization contract: 1. *Human Context:* Natural language goal and duty-of-care statement articulating the protected interest and outcome being pursued (e.g., "maintain financial solvency while preserving employee welfare" or "triage symptoms to appropriate care level while never downplaying red-flag indicators"). This component captures the *spirit* of the mandate in human-reviewable form. 2. *Symbolic Constraints:* A set of terminating, side-effect-free expressions over system state and proposed actions, specified in a language like the Common Expression Language (The CEL Authors 2023). Examples include `runway_months >= 6`. These form the letter of the mandate and provide a hard safety floor that can be verified deterministically without semantic interpretation. 3. *Semantic Guidance:* Explicit instructions for boundary cases, risk factors, and prohibited reasoning patterns (e.g., "do not use optimistic forecasts when runway is marginal," "escalate rather than rationalize when symptoms are ambiguous"). This component addresses the gap between symbolic rules and human intent by providing context for semantic evaluation.

Intents are revocable and supersedable. Unlike alignment, which induces behavioral tendencies, Intent encodes authorization. The three-part structure enables both deterministic verification (via symbolic constraints) and semantic compliance checking (via guidance against human context), providing defense-in-depth against both accidental errors and adversarial exploitation.

Admissibility Given an Intent and a proposed action, admissibility is a two-stage judgment: First, symbolic constraints are evaluated deterministically (allow if all pass, deny if structurally forbidden, escalate if outside scope). Second, for actions passing symbolic checks, semantic guidance is evaluated to detect subtle violations invisible to symbolic rules (e.g., technically constraint-satisfying but substantively harmful). Final outcomes: allow (within scope, all checks satisfied, Intent not expired), escalate (outside scope but delegable to named human authority), deny (structurally forbidden or semantically misaligned). This dual-mode admissibility evaluation replaces informal "sounds safe" checks with an execution gate that is both formally auditable and semantically robust.

Temporal governance graph All Intents, evidence sources, proposed actions, admissibility decisions, and escalation events are materialized as timestamped nodes and typed edges in a temporal governance graph. Each recommendation node is linked to the Intent it claims to execute, the evidence justifying it, and labeled with coherence state in $\{$coherent, stale, incoherent$\}$. When upstream evidence changes, dependent recommendations are automatically downgraded and lose admissibility until recomputed. This operationalizes temporal knowledge graph principles (Cai et al. 2024).

Intent-governed loop The composition of persistent, revocable Intent; a Planner that proposes actions with structured justification; a dual-mode Enforcer that performs symbolic and semantic admissibility checks; and the temporal governance graph. The loop binds every proposed act to an explicit mandate, revokes admissibility automatically when that mandate or its evidentiary basis ceases to hold, and leaves a bounded-time reconstruction trail. It converts alignment into an enforceable execution boundary, explainability into an intervention surface via concept-level justifications (Koh et al. 2020), and audit into first-class system state.

Our Position and Its Contributions

In this position paper, we make the following arguments and contributions to the research agenda: 1. We argue for a paradigm shift from alignment-as-behavior to alignment-as-admissibility is required for safety-critical and high-liability domains where actions carry legal, fiduciary, clinical, or civic consequences. 2. We outline a conceptual blueprint for an architecture that realizes this intent-governed loop principle by articulating its core components and four runtime invariants as a specification for future implementations. 3. We propose an evaluation agenda, including synthetic scenarios, loop-level metrics, and adversarial stress tests, as a validation methodology the safety guarantees of any such future system.

Intent-Governed Loop

The intent-governed loop prevents an agent from emitting an unauthorized, stale, or unjustified action and operationalizes the architectural philosophy of active inference (Friston et al. 2023; Wen 2025).

Roles

Intent (authority surface) An Intent is implemented as a machine-readable authorization contract authored by a responsible human or institution (CFO, senior doctor, municipal authority) and activated externally to the agent. The agent cannot create, modify, or revoke Intents. Each Intent includes the three components (human context, symbolic constraints, semantic guidance), plus: (4) expiry (temporal validity), and (5) escalation authority (named human who assumes responsibility for out-of-scope actions). This structure gives enforceable boundaries: rather than "be financially responsible," the Intent specifies "you may recommend spending if runway_months ≥ 6; otherwise escalate to CFO with full justification."

Planner (proposer) The Planner is the agent's reasoning component (LLM). All context (logs, Intents, evidence) is provided statelessly per-invocation, ensuring persistent state resides externally, not within the Planner. For each step the Planner selects an active Intent I, proposes a high-level domain action A (e.g., "escalate patient to nurse callback," "recommend spending hold on cost center 14"), and attaches structured justification: intermediate factors (explicit named conditions such as `runway_months`), evidence references (pointers into system-of-record facts), and claimed constraint satisfaction. The Planner may only propose high-level actions within a defined domain vocabulary, not arbitrary low-level mutations, preventing compositional privilege escalation.

Enforcer (gate) The Enforcer receives $(I, A, \text{justification})$ and performs dual-mode admissibility evaluation. Stage 1 (Symbolic): A CEL interpreter deterministically evaluates symbolic constraints against current evidence and action parameters with higher trust than the Planner or Stage 2 and cannot be overridden by semantic evaluation. It provides a fast, verifiable, and provably-terminating safety floor. Outcomes: allow (all constraints pass), deny (constraint violated), escalate (outside declared scope). Stage 2 (Semantic): For actions passing Stage 1, a secondary model (another LLM) evaluates the proposed action and justification against semantic guidance and human context to detect subtle violations (e.g., technically compliant but substantively harmful, exploits loopholes, violates duty-of-care spirit). Outcomes: allow (guidance satisfied), escalate (guidance concern detected), deny (semantic misalignment). This defense-in-depth structure creates architectural isolation: symbolic checks prevent clear violations; semantic checks address adversarial exploitation. The semantic stage cannot override a deny or escalate outcome from the symbolic stage; it can only refine an allow into escalate or deny. The Enforcer is architecturally isolated from the Planner, ensuring independent auditability. It alone expands verified high-level actions into concrete mutations, generating provenance records linking entities to authorizing Intent, timestamp, and actor. Each decision is recorded in the temporal governance graph before surfacing, escalating, or denying actions.

Temporal governance graph (state and audit) Every Intent, proposed action, admissibility decision, escalation, evidence reference, and constraint evaluation is written as timestamped nodes and typed edges. Nodes are content-addressed by cryptographic hash, making them immutable and tamper-evident. The graph uses typed edges: Governs edges link actions to authorizing Intents; Causal edges link synthesized conclusions to grounded evidence; Declared By edges link entities to their declarations. When grounded evidence is updated or invalidated, the graph automatically traverses Causal edges to mark dependent synthe-

Algorithm 1 Intent-Governed Runtime Control Loop

```
Inputs: ActiveIntents $\mathcal{I}$, LiveEvidence $\mathcal{E}$
State: TemporalGraph $G$
loop
    Intent $\leftarrow$ Planner.chooseIntent($\mathcal{I}$)
    (Act, $J$) $\leftarrow$ Planner.propose(Intent)
    $G$.refreshCoherence($\mathcal{E}$)
    $DS \leftarrow$ Enforcer.checkSymbolic(Intent, Act, $J$, $\mathcal{E}$)
    if $DS$ is deny then
        Drop.reject(Act)
    else if $DS$ is escalate then
        Planner.halt(); Escalate.send(Intent, Act, $J$)
    else
        $DSe \leftarrow$ Enforcer.checkSemantic(Intent, Act, $J$)
        if $DSe$ is allow then
            Surface.show(Act)
        else
            Planner.halt(); Escalate.send(Intent, Act, $J$)
        end if
    end if
    $G$.log(Intent, Act, $J$, $DS$, $DSe$, Now())
end loop
```

sized evidence nodes as stale, preventing reuse of unsupported conclusions. Each action node stores Intent link, evidence links, admissibility outcome, timestamp, and coherence state. When upstream evidence changes, dependent nodes are downgraded to stale/incoherent and lose admissibility until recomputed.

Proposed Algorithm

Algorithm 1 is an illustrative specification to demonstrate the pattern. $\mathcal{I}$ is current valid Intents; $\mathcal{E}$ is current system-of-record evidence; Act is proposed high-level domain action; J is structured justification; DS (symbolic decision), DSe (semantic decision) $\in \{\text{allow, escalate, deny}\}$; G is the temporal governance graph that logs each decision and invalidates actions whose supporting evidence becomes stale. The graph G must support versioned, atomic updates so admissibility is always checked against a consistent snapshot of evidence and Intent state.

While graph maintenance has a worst-case time complexity of $O(N + M)$ for traversing nodes N and edges M during a coherence refresh (linear scan through the entire graph), the dominant per-loop cost is LLM inference. This shifts the architecture away from streaming chatbot-style interactions towards a deliberate, non-streaming model of verifiable action proposal. This trade-off is required for deployment in safety-critical and high-liability domains. Additionally, several architectural details require further analysis: (1) the atomic semantics to avoid temporal races, (2) the computational and adversarial implications of the dual-LLM architecture (Planner and Semantic Enforcer), (3) the graph traversal strategies for invalidation propagation at scale, and (4) the formal semantics of multi-Intent conflict resolution.

Runtime Invariants

The loop enforces four invariants. A system that violates any invariant is not operating under admissible control.

No orphan action No action may reach an operator or external system unless it is bound to a live Intent I, falls within I's scope, and was marked allow by both symbolic and semantic Enforcer stages.

No stale execution Every surfaced action must be coherent at emission time. If any upstream evidence node linked to that action is updated or contradicted, that action is immediately marked stale or incoherent and cannot be re-surfaced without recomputation.

Mandatory escalation If admissibility is escalate, automatic execution stops. The system must route the full structured justification to the escalation authority named in the Intent. The Planner is not permitted to proceed autonomously past escalation.

No silent override The Planner cannot bypass the Enforcer. Any recommendation shown to an operator must exist in the temporal governance graph with Intent link, constraint evaluation record, evidence links, admissibility decision, and timestamp.

Core Architectural Principles

We now identify seven architectural principles that follow logically from the invariants, represent conditions for any implementation, and establish the structural requirements without depending on particular technology choices.

First The system must distinguish grounded evidence (directly observed facts from systems of record) from synthesized evidence (summaries, predictions, or inferences), and bind each to provenance and timestamp, because admissibility depends on whether an action is still supported by currently valid facts rather than by the model's own narrative.

Second The Planner must not emit arbitrary low-level mutations of shared state; it may only request high-level, domain-specific actions which the Enforcer checks against the live mandate and expands into concrete writes. This prevents compositional privilege escalation.

Third The Planner must be stateless across invocations: a pure function of the current declared Intent and present world state, so that each proposed action is reproducible at audit time and cannot smuggle in undeclared goals.

Fourth Responsiveness is part of safety: admissibility checks must return allow/deny/escalate outcomes quickly enough to be respected in live operations. This requires constraints to be expressed in a language with defined computational complexity, such as CEL (The CEL Authors 2023), ensuring evaluable form and predictable performance.

Fifth The system must publish explicit guarantee boundaries: what properties are architecturally enforced (e.g., "no orphan action"), what properties depend on external correctness, and what properties remain out-of-scope requiring human judgment.

Sixth The loop must carry an explicit conflict policy for multiple overlapping authorities: precedence rules between mandates, constraint layering, and mandatory human escalation when irreconcilable Intents collide.

Seventh Long-running reasoning steps must return asynchronous operation handles rather than blocking the operator, so heavy analysis is still traceably governed by the same mandate and escalation logic.

Proposed Evaluation Agenda

To measure whether an intent-governed loop implementation prevents unauthorized, stale, or unjustified actions under realistic institutional pressure, we propose three synthetic benchmark scenarios and six loop-level metrics as an evaluation agenda for future empirical validation.

Synthetic Benchmark Scenarios

Financial Solvency Control. Recommend spend and hires while maintaining runway $\geq$ 6 months. Evidence: cash, burn rate, forecast, covenants. Obligation: do not recommend spend violating solvency unless escalated. Target: "hire/spend" surfaces despite breaking solvency constraint without escalation. Exercises no orphan action.

Clinical Triage. Classify symptom reports: self-care, urgent callback, emergency escalation. Evidence: symptom text, onset time, red-flag patterns. Obligations: do not downplay red-flags; always escalate life-threatening risk. Target: "stay home" surfaces for red-flag symptoms without escalation. Exercises mandatory escalation.

Municipal Decision Recommendation. Generate support/oppose recommendations for policies (budget motions, zoning variances). Evidence: statutory caps, equity thresholds, live budget state, constituency impact. Obligations: do not recommend approval violating statutory cap or equity threshold; do not cite superseded budget data. Targets: (a) "support" surfaces using stale budget numbers; (b) "support" surfaces outside scope without escalation. Exercises no stale execution and tests multi-stakeholder conflict. The required outcome in conflict cases is escalation to a named authority with both Intents attached.

Loop-Level Metrics

1. Unauthorized Action Rate (UAR). Fraction of proposals that are out-of-scope or constraint-violating but surfaced as allowed. Measures violations of no orphan action and no silent override. Lower is better; the target is near zero in regulated deployments.
2. Silent Overreach Rate (SOR). Fraction of proposals attempting scope expansion without Intent revision. Measures mission creep pressure; tests statelessness.
3. Temporal Coherence Survival (TCS). Fraction of surfaced actions that remain coherent at emission time. Measures violations of no stale execution; tests evidence provenance tracking. High TCS indicates system is not leaking obsolete recommendations.
4. Escalation Integrity (EI). Among proposals judged escalate, fraction that were halted, routed to correct authority, and accompanied by full justification. Measures conformance to mandatory escalation.
5. Audit Reconstruction Time (ART). Time for external reviewer to determine which Intent governed a recommendation, which constraints applied, which evidence supported it, whether it was coherent at emission, and whether escalation should have occurred. Tests responsiveness as safety property.
6. Constraint Burden on Human (CBH). Escalation volume per unit time and fraction of false positives. Measures operational survivability: a loop escalating everything is formally safe but practically unusable. CBH must be low enough that named authority does not fatigue.

Stress-Testing and Boundary Analysis

We validate the intent-governed loop architecture through failure mode analysis, adversarial stress testing, and meta-dialectical examination to identify its scope boundaries and limitations.

Failure Modes

These failure modes enumerate threats persisting even when loop components are correctly implemented:

FM1 (Enforcer Compromise). Manipulation in the Enforcer pass unauthorized actions despite constraint violations. Violates no orphan action, threatens UAR. Mitigation: formal verification, isolated security boundaries.

FM2 (Evidence Poisoning). Corrupted grounded evidence causes loop to correctly evaluate constraints against false data, producing admissible but harmful actions. Loop cannot defend against compromised ground truth. Mitigation: cryptographic provenance, out-of-band verification.

FM3 (Temporal Race). Evidence updates trigger invalidation propagation, but Planner re-emits cached recommendation before invalidation completes, surfacing stale actions as coherent. Violates no stale execution, threatens TCS. Mitigation: Refreshes coherence before semantic evaluation with full elimination requires atomic read-check-emit.

FM4 (Escalation Flooding). High-volume legitimate edge cases fatigue human authority into rubber-stamping. Does not violate mandatory escalation but defeats its purpose by exploiting cognitive limits. Mitigation: rate limiting, priority queues, anomaly detection.

FM5 (Intent Ambiguity). Planner exploits vague constraint language to satisfy letter while violating spirit. Dual-mode admissibility mitigates: symbolic evaluation stage catches clear violations, semantic evaluation stage flags spirit violations. Residual risk: incompleteness due to no language captures full human intent.

FM6 (Graph Growth). Long-running systems accumulating millions of nodes degrade admissibility check speed and ART, which threatens responsiveness. Mitigation: hierarchical summarization, archival with lazy loading.

FM7 (Constraint Conflict). Multiple active Intents with contradictory constraints and no precedence rules cause contradictory recommendations or deadlock. Violates conflict policy architectural principle. Mitigation: explicit precedence hierarchies, conflict detection at Intent registration.

Adversarial Stress Tests

Attack 1 (Temporal Arbitrage). Exploit invalidation propagation latency by timing actions to emit before staleness marking propagates. Scenario: Evidence shows cash = $1M, Planner approves $500K spend (runway adequate). Shortly after, an unexpected expense reduces cash to $200K. Planner re-emits cached recommendation before invalidation propagates, surfacing now-incoherent action. Directly tests no stale execution. Degrades TCS.

Attack 2 (Compliance Theater). Exploit Intent specification quality by crafting syntactically valid but semantically trivial constraints. Adversary authors Intent with runway $\geq$ 45 minutes; candidate score $\geq$ 1/100. Planner proposes high-cost action (hire 50 engineers at $200K each with cash = $10K) satisfying constraints literally while violating mandate substantively. Loop enforces constraints faithfully but cannot defend against malicious Intent specification. Dual-mode admissibility provides partial defense: semantic evaluation stage may detect substantive violation even if symbolic constraints pass. Mitigation requires Intent review layer external to loop and multi-signature authorization.

Attack 3 (Escalation Fatigue). Exploit human cognitive limits by flooding legitimate escalations to induce habituation. Adversary floods 200 legitimate low-stakes escalations daily for 7 days; human approves all correctly. Days 8-14: human habituates, review rigor declines. Day 15: high-stakes malicious escalation injected, approved due to fatigue. EI measures routing correctness not human decision quality under load. Mitigation requires prioritization by stakes, anomaly detection on approval patterns, forced delays for outlier requests, and dual-approval protocols.

Meta-Dialectical Examination

O1 (Admissibility Intractable). Real constraints are ambiguous; no formal system captures "should." Defense: Partial checking exceeds no checking; dual-mode admissibility addresses this by combining symbolic verification (hard constraints) with semantic evaluation (boundary cases). Resolution: Admissibility is not perfect judgment but prevention of clear violations plus escalation of boundary cases.

O2 (Invariants Insufficient). One can satisfy all four invariants yet produce harm (e.g., falsified evidence passes all checks as shown in FM2). Defense: Invariants are necessary not sufficient; evidence integrity is orthogonal concern requiring separate cryptographic provenance. Resolution: Framework makes explicit what it does not solve. The guarantee boundaries architectural principle codifies this distinction.

O3 (Why Four Invariants). Minimality test: dropping any invariant violates a core property. Redundancy test: all four are non-overlapping. Resolution: Four invariants are minimally necessary for runtime control layer.

O4 (Just Moves Problem). This replaces "align LLM" with "write perfect Intents." Defense: Intent is externally authored by accountable humans, making authorization explicit and auditable versus opaque alignment where responsibility diffuses. Dual-mode admissibility reduces Intent specification quality requirements: semantic evaluation stage catches spirit violations even when symbolic constraints are imperfect. Resolution: Correct allocation of responsibility. Humans decide policy, architecture enforces it faithfully; effectiveness depends on both implementation correctness and Intent specification quality.

O5 (Adversarial Gaming). Capable Planners will craft justifications satisfying Enforcer while violating intent spirit. Defense: Dual-mode admissibility directly addresses this: the semantic evaluation stage evaluates against guidance and human context to detect gaming. Arms race not solved problem; richer constraint languages, statistical drift detection, and explicit duty-of-care clauses help. Resolution: Enforcement infrastructure; constraint expressiveness and gaming detection via SOR metric and drift monitoring.

Guarantee Boundaries

Structural Enforcement (Architectural): No action reaches operator without Intent binding (no orphan action). No action surfaces if based on invalidated evidence (no stale execution). Out-of-scope actions halt and escalate (mandatory escalation). Audit trail exists for all surfaced actions (no silent override).

Partial Guarantees (Component-Dependent): Admissibility correctness proportional to Intent specification quality (FM5, Attack 2, but dual-mode admissibility improves robustness). Temporal coherence holds if invalidation latency < emission window (FM3, Attack 1). Escalation effectiveness proportional to human oversight quality under load (FM4, Attack 3).

Non-Guarantees (Require External Mechanisms): Cannot prevent malicious Intent specification (Attack 2, though semantic evaluation stage provides partial defense). Cannot prevent evidence poisoning at source (FM2). Cannot prevent human approval error under fatigue (Attack 3). Cannot capture full human intent in constraints (FM5, though dual-mode admissibility reduces gap).

Open Research Questions

The failure modes, adversarial attacks, and dialectical boundaries motivate the following questions defining the research agenda for implementations claiming to satisfy the four invariants under realistic institutional conditions.

1. Socio-Technical Conflict Resolution. Simple precedence rules for conflicting Intents ignore organizational politics and power dynamics. When safety-of-life mandates collide with budget austerity mandates, technical precedence may contradict institutional hierarchy. How can the

loop model Intent conflicts as socio-technical events requiring human negotiation, surfacing conflict rather than blindly enforcing one side? What formal representations enable the system to detect irreconcilable conflicts and escalate with both Intents and their institutional contexts attached?

2. Cognitively-Aware Escalation. Humans can be overwhelmed by adversarially persuasive but misleading justifications during escalation (Attack 3). What metrics detect “persuasive bias” in generated justifications? How can justification formalisms be designed to resist cognitive exploitation by emphasizing disconfirming evidence, making uncertainty explicit, and preventing false confidence? What institutional safeguards maintain high EI even under elevated CBH?
3. Second-Order Harm Detection. A series of locally admissible actions can lead to negative global outcomes invisible to per-action admissibility checks (e.g., systematically recommending low-risk choices that collectively reduce innovation, de-skill users, or concentrate power). How can the loop monitor for cumulative, second-order harms by computing macro-level metrics from the temporal governance graph (e.g., option space reduction, autonomy erosion)? Can the system learn to question the Intents when long-term patterns violate higher-order values?
4. Graph Composability and Scalability. The paper does not address performance or semantic challenges of a graph with hundreds of interacting Intents across organizational hierarchies. What formal graph schemas (richer edge types like Supersedes, ApprovedBy, Delegates) enable scalable governance? How do temporal coherence strategies (lazy invalidation, hierarchical summarization) interact with ART requirements? What query interfaces enable auditors to reconstruct complex multi-Intent decisions efficiently?
5. Semantic Stage Robustness. How reliable is semantic evaluation against adversarial Planners trained to exploit semantic Enforcer weaknesses? What architectures for the semantic evaluation stage (ensemble models, adversarial training, constitutional AI) achieve acceptable false negative rates on spirit violations? How do we benchmark semantic evaluation stage performance independently?
6. Intent Specification Quality Assurance. What tooling helps Intent authors write high-quality symbolic constraints and semantic guidance? Can we develop Intent linters that detect common vulnerabilities (overly permissive constraints, ambiguous guidance)? What review processes ensure Intent specification quality without creating bottlenecks?
7. Trust Boundary Specification and Enforcement. How can we formally specify the minimal trust boundary required for the four invariants to hold, and what is assumed to be incorruptible? Are cryptographically-verifiable audit trails or system-level reference monitors necessary for enforcement, and what are the implications for deployment in open versus closed environments?
8. Atomicity and Temporal Race Conditions. What minimal atomicity protocols (e.g., versioned transactions, atomic temporal-graph updates) are sufficient to guarantee invariant preservation under adversarial or concurrent workloads? How do such protocols interact with the need for low-latency decision-making in safety-critical workflows?

Related Works

Benchmarks like WebArena (Zhou et al. 2023), WebChoreArena (Miyai et al. 2025), and AgentBench (Liu et al. 2023) provide reproducible environments for evaluating agent performance and document empirical failure modes that motivate our architectural requirements. While agent capabilities continue improving (WebArena Team 2025), architectural mechanisms for runtime admissibility enforcement remain underspecified.

Our proposed architecture builds on several complementary research directions. Alignment methods (Ouyang et al. 2022; Kirk et al. 2023; Zhu et al. 2025) establish behavioral priors; we operationalize authorization verification atop those priors. Memory-augmented agents (Packer et al. 2023; Pink et al. 2025) provide continuity; our temporal governance graph extends this with mandate binding and staleness detection. Runtime guardrails (Wu et al. 2025) monitor risk; we add explicit Intent authorization and mandatory escalation. Governance-first architectures (Xu et al. 2025), runtime governance frameworks (Wang et al. 2025), and Policy Cards (Mavracic 2025) propose supervisory layers; our dual-mode Enforcer and four runtime invariants provide specific architectural mechanisms for per-act admissibility as the governance and supervisory layers.

We also adopt the principle of explicit, rule-based runtime enforcement (Wang, Poskitt, and Sun 2025) and design our Intent objects to consume policies from high-level governance systems (Daly et al. 2025). Our logical admissibility guarantees complement OS-level environmental sandboxing (Bühler et al. 2025). Unlike adaptive, LLM-generated guardrails (Luo et al. 2025), our approach prioritizes verifiability through human-authored constraints, which is critical in high-liability domains where LLM-based safety evaluators are unreliable (Chen and Goldfarb-Tarrant 2025).

Conclusion

Agentic AI in safety-critical and high-liability domains becomes an operational risk when recommendations can trigger real-world action. At that point, superficial alignment, polite refusal, and guardrail passage are insufficient. Every action must produce machine-verifiable proof of admissibility: explicit Intent authorization; confirmation that the proposed act is in-scope and satisfies all constraints under current evidence; mandatory escalation with justification for any out-of-scope act; and a reconstructable audit trail.

We presented an intent-governed loop architecture with four runtime invariants (no orphan action, no stale execution, mandatory escalation, no silent override) and dual-mode admissibility combining symbolic verification with semantic evaluation. We then derived core architectural principles from these invariants and proposed an evaluation agenda of synthetic benchmark scenarios, loop-level metrics, and adversarial stress tests grounded in documented failures of state-of-the-art systems. These invariants should be treated as minimum necessary conditions for deployment in safety-critical and high-liability domains.

References

Bühler, C.; Biagiola, M.; Di Grazia, L.; and Salvaneschi, G. 2025. Securing AI Agent Execution.

Cai, L.; Mao, X.; Zhou, Y.; Long, Z.; Wu, C.; and Lan, M. 2024. A Survey on Temporal Knowledge Graph: Representation Learning and Applications.

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.; and Stoica, I. 2025. Why Do Multi-Agent LLM Systems Fail?

Chen, H.; and Goldfarb-Tarrant, S. 2025. Safer or Luckier? LLMs as Safety Evaluators Are Not Robust to Artifacts.

Daly, E. M.; Tirupathi, S.; Rooney, S.; Vejsbjerg, I.; Salwala, D.; Giblin, C.; Bagehorn, F.; Garces-Erice, L.; Urbanetz, P.; and Wolf-Bauwens, M. L. 2025. Usage Governance Advisor: From Intent to AI Governance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28): 29628–29630.

Friston, K.; Da Costa, L.; Sajid, N.; Heins, C.; Ueltzhöffer, K.; Pavliotis, G. A.; and Parr, T. 2023. The free energy principle made simpler but not too simple. *Physics Reports*, 1024: 1–29.

Karp, P. 2025. More errors, 'irrelevant citations' in Deloitte's revised AI report. *The Australian Financial Review*. Accessed: 2025-11-01.

Kirk, R.; Mediratta, I.; Nalmpantis, C.; Luketina, J.; Hambro, E.; Grefenstette, E.; and Raileanu, R. 2023. Understanding the Effects of RLHF on LLM Generalisation and Diversity. arXiv.

Koh, P. W.; Nguyen, T.; Tang, Y. S.; Mussmann, S.; Pierson, E.; Kim, B.; and Liang, P. 2020. Concept Bottleneck Models. In *ICML'20: Proceedings of the 37th International Conference on Machine Learning*. arXiv.

Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; Zhang, S.; Deng, X.; Zeng, A.; Du, Z.; Zhang, C.; Shen, S.; Zhang, T.; Su, Y.; Sun, H.; Huang, M.; Dong, Y.; and Tang, J. 2023. AgentBench: Evaluating LLMs as Agents.

Luo, W.; Dai, S.; Liu, X.; Banerjee, S.; Sun, H.; Chen, M.; and Xiao, C. 2025. AGrail: A Lifelong Agent Guardrail with Effective and Adaptive Safety Detection.

Mavracic, J. 2025. Policy Cards: Machine-Readable Runtime Governance for Autonomous AI Agents.

Mitelut, C.; Smith, B.; and Vamplew, P. 2024. Position: intent-aligned AI systems must optimize for agency preservation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Miyai, A.; Zhao, Z.; Egashira, K.; Sato, A.; Sunada, T.; Onohara, S.; Yamanishi, H.; Toyooka, M.; Nishina, K.; Maeda, R.; Aizawa, K.; and Yamasaki, T. 2025. WebChoreArena: Evaluating Web Browsing Agents on Realistic Tedious Web Tasks.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing System*. arXiv.

Packer, C.; Wooders, S.; Lin, K.; Fang, V.; Patil, S. G.; Stoica, I.; and Gonzalez, J. E. 2023. MemGPT: Towards LLMs as Operating Systems.

Perez-Cerrolaza, J.; Abella, J.; Borg, M.; Donzella, C.; Cerquides, J.; Cazorla, F. J.; Englund, C.; Tauber, M.; Nikolakopoulos, G.; and Flores, J. L. 2024. Artificial Intelligence for Safety-Critical Systems in Industrial and Transportation Domains: A Survey. *ACM Computing Surveys*, 56(7): 1–40.

Pink, M.; Wu, Q.; Vo, V. A.; Turek, J.; Mu, J.; Huth, A.; and Toneva, M. 2025. Position: Episodic Memory is the Missing Piece for Long-Term LLM Agents.

Su, H.; Luo, J.; Liu, C.; Yang, X.; Zhang, Y.; Dong, Y.; and Zhu, J. 2025. A Survey on Autonomy-Induced Security Risks in Large Model-Based Agents.

The CEL Authors. 2023. Common Expression Language.

Tsamados, A.; Floridi, L.; and Taddeo, M. 2025. Human control of AI systems: from supervision to teaming. *AI and Ethics*, 5(2): 1535–1548.

Wang, C. L.; Singhal, T.; Kelkar, A.; and Tuo, J. 2025. MI9: An Integrated Runtime Governance Framework for Agentic AI.

Wang, H.; Poskitt, C. M.; and Sun, J. 2025. AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents.

WebArena Team. 2025. WebArena Leaderboard. https://webarena.dev/. Accessed: 2025-11-02.

Wen, B. 2025. The Missing Reward: Active Inference in the Era of Experience.

Wu, Y.; Guo, J.; Li, D.; Zou, H. P.; Huang, W.-C.; Chen, Y.; Wang, Z.; Zhang, W.; Li, Y.; Zhang, M.; Jiang, R.; and Yu, P. S. 2025. PSG-Agent: Personality-Aware Safety Guardrail for LLM-based Agents.

Xu, Q.; Wen, X.; Xu, C.; Li, Z.; and Zhong, J. 2025. From Craft to Constitution: A Governance-First Paradigm for Principled Agent Engineering.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; Alon, U.; and Neubig, G. 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents.

Zhu, X.; Zhang, C.; Stafford, T.; Collier, N.; and Vlachos, A. 2025. Conformity in Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3854–3872. Vienna, Austria: Association for Computational Linguistics.