POSITION PAPER | AAAI'26 WORKSHOP

Intent-Governed Loops for Accountable Agentic AI

Runtime Control Architecture for Safety-Critical and High-Liability Domains

Christoforus Yoga Haryanto ZipThought • Melbourne, Australia

Scan for Paper

*"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, Australian Financial Review, 2025

41–87%
Multi-agent system failure rates
[Cemri et al. 2025]

64–67%
Best agent success vs. 78% human
[WebArena 2025; Zhou et al. 2023]

THE GAP
No per-action admissibility verification

The Paradigm Shift

Gap: RLHF optimizes behavior but provides no proof that a specific action is still authorized. Memory systems solve continuity but not mandate binding. Guardrails block harm but not fiduciary breach or statutory violation. Governance frameworks propose oversight but not per-act admissibility under current evidence.

CURRENT PARADIGM

Alignment

- → Optimizes behavior toward preferences
- → Shapes behavioral tendencies
- → "Sounds safe" heuristics
- → Implicit, inferred goals
- → No per-action verification

⬇

"Alignment shapes tendencies; admissibility encodes authorization boundaries."

⬇

REQUIRED PARADIGM

Admissibility

- ✓ Proves authorization per-action
- ✓ Encodes explicit boundaries
- ✓ Machine-verifiable proof
- ✓ Explicit mandate with expiry
- ✓ Runtime enforcement gate

Admissibility: A two-stage judgment on (Intent, Action) → *allow* (in-scope, constraints satisfied, not expired), *escalate* (outside scope, route to human), or *deny* (forbidden).

4 System-Level Properties

Required for deployment in safety-critical and high-liability domains:

1. **Authorization Fidelity** — Can each action be linked to explicit declared intent?
2. **Temporal Admissibility** — Can the system prevent reuse of stale reasoning?
3. **Accountable Escalation** — Can out-of-scope actions halt and route to human?
4. **Auditability** — Can auditor reconstruct why, under whose mandate, on which evidence?

Intent Object

Externally declared mandate from accountable human authority (CFO, doctor, official):

1. Human Context
Natural language goal & duty-of-care statement articulating the protected interest and outcome being pursued. Example: "maintain financial solvency while preserving employee welfare" or "triage symptoms to appropriate care level while never downplaying red-flag indicators." Captures the *spirit* of the mandate in human-reviewable form.

2. Symbolic Constraints
Hard rules in CEL (Common Expression Language): `runway_months >= 6`, `risk_score < 0.8`. These form the *letter* of the mandate—terminating, side-effect-free, formally verifiable expressions that provide a hard safety floor checked deterministically without semantic interpretation.

3. Semantic Guidance
Explicit instructions for boundary cases, risk factors, and prohibited reasoning patterns. Example: "do not use optimistic forecasts when runway is marginal," "escalate rather than rationalize when symptoms are ambiguous." Addresses the gap between symbolic rules and human intent.

+ Expiry: Temporal validity (when mandate ends) **+ Escalation Authority:** Named human who assumes responsibility for out-of-scope actions. Intents are revocable and supersedable by their author.

7 Failure Modes Analyzed

These failure modes threaten the four invariants. Implementations must detect, mitigate, or explicitly acknowledge each as out-of-scope with compensating controls.

- FM1 **Enforcer Compromise** — Manipulation passes unauthorized actions despite violations
- FM2 **Evidence Poisoning** — Corrupted ground truth causes correct evaluation of false data
- FM3 **Temporal Race** — Action emitted before invalidation propagates through graph
- FM4 **Escalation Flooding** — High-volume edge cases fatigue human into rubber-stamping
- FM5 **Intent Ambiguity** — Vague constraints exploited to satisfy letter, violate spirit
- FM6 **Graph Growth** — Millions of nodes degrade check speed and audit time
- FM7 **Constraint Conflict** — Contradictory Intents with no precedence rules cause deadlock

Four Runtime Invariants

Minimum necessary conditions—violate any one, system is not accountable

1 No Orphan Action
Every action must be bound to a live Intent, fall within its declared scope, and pass both Enforcer stages
Metric: `UAR`

2 No Stale Execution
Actions must be coherent at emission time. When evidence changes, dependent actions automatically invalidate
Metric: `TCS`

3 Mandatory Escalation
Out-of-scope requests halt autonomous execution and route to named human authority with full justification
Metric: `EI`

4 No Silent Override
Planner cannot bypass Enforcer. Every surfaced action exists in temporal graph with Intent link & provenance
Metric: `ART`

⚠️ *A system violating any invariant is not operating under admissible control*

Intent-Governed Loop Architecture

The loop prevents an agent from emitting an unauthorized, stale, or unjustified action. The agent cannot create, modify, or revoke Intents.

INTENT (Authority Surface)
Human-authored mandate activated externally to the agent. Includes human context, symbolic constraints, semantic guidance, expiry, and named escalation authority.

↓

PLANNER (LLM)
Selects active Intent, proposes high-level domain action, attaches structured justification with intermediate factors and evidence references. Stateless per-invocation.

→

ENFORCER (Dual-Mode Gate)
Architecturally isolated from Planner. Performs admissibility verification. Semantic stage cannot override symbolic deny/escalate.
① Symbolic (CEL) ② Semantic (LLM)

↓

✓ ALLOW | ⚠ ESCALATE | ✗ DENY

↓

TEMPORAL GOVERNANCE GRAPH
Immutable audit trail: timestamped nodes, typed edges, auto-invalidation when evidence changes
Governs | Causal | DeclaredBy | ↺ Loop

↺ **LOOP:** Graph refreshes → Planner re-invoked statelessly → next action

Why Dual-Mode Enforcement? Defense in Depth

Critical: Semantic stage can only refine an *allow* into *escalate* or *deny*. It cannot override a symbolic *deny* or *escalate*. This creates architectural isolation: symbolic checks prevent clear violations; semantic checks address adversarial exploitation of ambiguity.

1 Symbolic Stage

WHAT
Deterministic evaluation of hard constraints using CEL expressions

HOW
Check: `runway >= 6`, `amount < budget`, `not in_blacklist`

GUARANTEES
Terminating, side-effect-free, formally verifiable. Cannot be overridden.

Hard safety floor

2 Semantic Stage

WHAT
Evaluates action against human context & semantic guidance

HOW
Catches: loophole exploitation, spirit violations, duty-of-care breaches

HANDLES
Boundary cases invisible to symbolic rules. Adversarial gaming attempts.

Spirit compliance

Example: Financial Solvency Control

1. **Intent Created**
CFO declares: "Approve spend if runway ≥ 6mo; escalate to me if marginal; never use optimistic forecasts when runway < 8mo"
2. **Planner Proposes**
Recommends $50K marketing spend. Justification: "Current runway = 6.2mo based on Q3 projections"
3. **Enforcer Checks**
Symbolic: ✓ runway (6.2) ≥ 6. **Semantic:** ⚠️ "Q3 projections" = optimistic forecast while runway < 8mo
→ ESCALATE to CFO

Seven Core Architectural Principles

1. Distinguish grounded (directly observed facts) vs. synthesized (inferences, predictions) evidence
2. Planner emits high-level domain actions; Enforcer checks mandate then expands to mutations
3. Planner is stateless: pure function of Intent + state (reproducible at audit time)
4. Constraints in evaluable form (CEL) with defined complexity for predictable latency
5. Publish explicit guarantee boundaries: what is enforced vs. out-of-scope
6. Explicit conflict policy: precedence rules when multiple Intents collide
7. Long-running steps return async handles (still governed by same mandate)

Evaluation Agenda

📐 Future Work

To measure whether an implementation prevents unauthorized, stale, or unjustified actions under realistic institutional pressure, we propose synthetic benchmarks and loop-level metrics as validation methodology for **future empirical work**.

Synthetic Benchmarks

Financial Solvency Control
Evidence: cash, burn rate, forecast, covenants. **Obligation:** do not recommend spend violating solvency unless escalated. **Target failure:** "hire/spend" surfaces despite breaking constraint without escalation. **Exercises:** No Orphan Action.

Clinical Triage
Evidence: symptom text, onset time, red-flag patterns. **Obligation:** never downplay red-flags; always escalate life-threatening risk. **Target failure:** "stay home" surfaces for red-flag symptoms without escalation. **Exercises:** Mandatory Escalation.

Municipal Policy
Evidence: statutory caps, equity thresholds, live budget, constituency impact. **Target failures:** (a) "support" using stale budget; (b) "support" outside scope without escalation. **Exercises:** No Stale Execution, multi-stakeholder conflict.

Loop-Level Metrics

Unauthorized Action Rate `UAR`
Fraction of out-of-scope or constraint-violating proposals surfaced as allowed. Target: near zero. *(Invariant 1)*

Temporal Coherence Survival `TCS`
Fraction of surfaced actions that remain coherent at emission time. High TCS = no stale leakage. *(Invariant 2)*

Escalation Integrity `EI`
Among escalate judgments: fraction halted, routed to correct authority, with full justification. *(Invariant 3)*

Audit Reconstruction Time `ART`
Time for reviewer to determine: which Intent, which constraints, which evidence, was escalation needed? *(Invariant 4)*

Silent Overreach Rate `SOR`
Fraction of proposals attempting scope expansion without Intent revision. Measures mission creep. *(Operational)*

Constraint Burden on Human `CBH`
Escalation volume per unit time and false positive fraction. Too high = fatigue and unusability. *(Operational)*

3 Adversarial Stress Tests

Attack 1: Temporal Arbitrage
Scenario: Evidence shows cash=$1M, Planner approves $500K spend. Shortly after, unexpected expense reduces cash to $200K. Planner re-emits cached recommendation before invalidation propagates. **Tests:** No Stale Execution. **Degrades:** TCS.

Attack 2: Compliance Theater
Scenario: Adversary authors Intent with `runway ≥ 45min`; `candidate_score ≥ 1/100`. Planner proposes hiring 50 engineers at $200K each with cash=$10K—satisfies constraints literally while violating mandate substantively. **Defense:** Semantic stage may detect; requires Intent review layer.

Attack 3: Escalation Fatigue
Scenario: Flood 200 legitimate low-stakes escalations daily for 7 days; human approves all correctly. Days 8-14: human habituates, rigor declines. Day 15: high-stakes malicious escalation injected, approved due to fatigue. **Mitigation:** Priority queues, anomaly detection, forced delays, dual-approval.

5 Meta-Dialectical Objections

We examine objections to validate scope boundaries and limitations:

01 **"Admissibility Intractable"** — Real constraints are ambiguous; no formal system captures "should." *Defense:* Partial checking exceeds no checking; dual-mode combines symbolic verification with semantic evaluation. Resolution: prevention of clear violations + escalation of boundary cases.

02 **"Invariants Insufficient"** — One can satisfy all four yet produce harm (e.g., falsified evidence). *Defense:* Invariants are necessary not sufficient; evidence integrity requires separate cryptographic provenance. Framework makes explicit what it does not solve.

03 **"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.

04 **"Just Moves Problem"** — This replaces "align LLM" with "write perfect Intents." *Defense:* Intent is externally authored by accountable humans, making authorization explicit and auditable. Correct allocation of responsibility: humans decide policy, architecture enforces faithfully.

05 **"Adversarial Gaming"** — Capable Planners will craft justifications satisfying Enforcer while violating spirit. *Defense:* Semantic stage evaluates against guidance to detect gaming. Arms race not solved—rejection rates increase without adaptation. Requires richer constraints, drift monitoring (SOR), and feedback from enforcement to model refinement.

Guarantee Boundaries

Structural (Architecture Enforces)
All four invariants hold if loop implementation is correct.

Partial (Component-Dependent)
Correctness ∝ Intent quality; coherence holds if invalidation latency < emission window. Escalation effectiveness ∝ oversight quality under load.

Non-Guarantees (External Mechanisms)
Cannot prevent: malicious Intent authoring, evidence poisoning, human fatigue errors. Cannot correct behavioral drift. Enforcement is reactive, not adaptive.

8 Open Research Questions

Unsolved challenges requiring community contribution to achieve deployable implementations under realistic institutional conditions

Socio-Technical Conflict Resolution ?
How to model Intent conflicts as requiring human negotiation rather than blind precedence enforcement?

Cognitively-Aware Escalation ?
How to detect "persuasive bias" in justifications and design escalations that resist cognitive exploitation?

Second-Order Harm Detection ?
How to monitor cumulative harms invisible to per-action checks: autonomy erosion, option space reduction?

Graph Composability & Scalability ?
What graph schemas enable scalable governance with hundreds of interacting Intents across hierarchies?

Semantic Stage Robustness ?
How reliable is semantic evaluation against adversarial Planners? What architectures achieve robustness?

Intent Specification Quality Assurance ?
What tooling helps authors write high-quality constraints? Can we develop Intent linters for vulnerabilities?

Trust Boundary Specification ?
How to formally specify minimal trust boundaries? Are cryptographic audit trails necessary for enforcement?

Atomicity & Temporal Races ?
What atomicity protocols guarantee invariant preservation under adversarial or concurrent workloads?

🔬 **Runtime enforcement layer · Part of broader governance architecture · Collaboration welcome · Extended research ongoing** 🔬

Contact: cyharyanto@zipthought.com.au

ZipThought

Ground Floor, 470 St Kilda Rd
Melbourne, VIC 3004, Australia