

WORKING PAPER

EVIDE Governance Lab - July 2026

Governance Loop Engineering (G-LOOP-E)

Continuous Verification of Accountability Survivability Across Recursive Boundary Crossings

From Boundary Qualification to Operational Loop: The Governance Heartbeat, the Governance Preservation Loop, and the Reconstructability Test

Emanuel Celano

Informatica in Azienda - EVIDE Governance Lab - app.certifywebcontent.com

“RSG defines what must survive. Governance Loop Engineering defines how systems continuously verify that it survives.”
- G-LOOP-E Canonical Principle

Version 1.0 - July 2026

G-LOOP-E (Governance Loop Engineering) is the operational verification framework that continuously evaluates accountability survivability across recursive governance boundary crossings.

Abstract

Recursive Semantic Governance (RSG) established a theoretical framework for qualifying whether governance-relevant meaning survives a boundary crossing, replacing static audit-log accumulation with state transformation qualification: $G(t_{n+1}) = T(G(t_n), \Delta s, C_p)$. RSG answers the question of what must survive a crossing. It does not, by design, specify how a running system continuously confirms that survival holds between crossings, across many iterations, inside a live orchestration pipeline.

This working paper introduces Governance Loop Engineering (G-LOOP-E), an operational layer built directly on RSG primitives. G-LOOP-E reframes the exit condition of an iterative system loop away from behavioral convergence and toward accountability verification. It introduces the Governance Heartbeat -- a recurring verification pulse issued at each iteration or boundary crossing -- and the Governance Preservation Loop (GPL), the formal cycle through which a system repeatedly asks a single reconstructability question before authorizing continuation: can accountability still be independently reconstructed if this iteration becomes the last.

We distinguish G-LOOP-E from general-purpose Loop Engineering, whose feedback cycles optimize behavior and output quality. G-LOOP-E's loop optimizes nothing; it verifies. We map the loop's exit states directly onto RSG's Split Confirmation categories, catalog six loop-specific failure modes not covered by RSG's boundary-crossing taxonomy, and present the Convoy Model -- an illustrative, pedagogical figure in which a train adds, modifies, or blocks governance "cars" at each station -- to make the architecture legible to non-technical stakeholders without diluting the underlying formalism. We conclude with a full walkthrough applying G-LOOP-E to a multi-step agentic orchestration pipeline, and with computational considerations for calibrating heartbeat frequency against governance depth.

G-LOOP-E Canonical Principles: 1. RSG defines what must survive a boundary crossing. G-LOOP-E defines how a running system continuously verifies that it has. 2. The loop does not ask whether accountability is preserved; it asks whether accountability could still be reconstructed if the process stopped now.

Keywords: AI governance, recursive governance, loop engineering, accountability survivability, governance heartbeat, agentic orchestration, boundary engineering, reconstructability

Glossary of Core Terms

These definitions extend the RSG glossary. Terms already defined in RSG v1.1 (Governance Vector, Semantic Divergence, Causal Persistence, Split Confirmation, Crystallization, etc.) are used here without redefinition; only terms introduced by G-LOOP-E are defined below.

TERM	OPERATIONAL DEFINITION
Governance Heartbeat	A recurring verification pulse, issued at a defined cadence or at each boundary crossing, that triggers a Governance Preservation Loop cycle. The heartbeat does not act; it verifies.
Governance Preservation Loop (GPL)	The formal iterative cycle through which a system evaluates, at each heartbeat, whether accountability survivability still holds, and authorizes, defers, or halts continuation accordingly.
Reconstructability Test	The specific query issued at each GPL cycle: can accountability up to this point still be independently reconstructed if the process were interrupted immediately after this iteration.
Governance Inspector	The functional role -- human, automated, or hybrid -- that evaluates the Reconstructability Test at a given heartbeat, analogous to RSG's Boundary Connector but operating continuously rather than only at layer transitions.
Loop Exit Condition	The verdict issued by a GPL cycle: CONTINUE, DEFER (escalate for review while holding state), or HALT (block progression to a safe state).
Heartbeat Interval	The temporal or event-based cadence at which heartbeats are issued. A deployment parameter, not a framework constant, calibrated against consequence severity and reversibility.
Verification Depth	The subset of governance vector components actually evaluated by a given GPL cycle, ranging from a single component to the full vector. Independent of heartbeat frequency: a heartbeat can be frequent and shallow, rare and deep, or any combination of the two.
Scheduled Heartbeat	A heartbeat issued at a fixed temporal or step-based interval, independent of any specific change in governance state.
Triggered Heartbeat	A heartbeat issued immediately in response to a defined state-change event -- an authority change, a threshold change, a new unresolved signal, an increase in Δs , or a decrease in C_p -- regardless of where the system sits in the scheduled interval. Triggered heartbeats always take priority over, and do not wait for, the next scheduled heartbeat.
Trigger Deadband	A declared sensitivity threshold δ_{trig} below which fluctuations in Δs or C_p do not fire a triggered heartbeat. Suppressed sub-threshold events are still logged. The deadband value is itself part of the evidentiary record; an undeclared or silently widened deadband constitutes Heartbeat Suppression by design.
Refractory Period	A declared minimum interval between two consecutive triggered heartbeats fired by the same governance vector component, preventing trigger cascades in high-frequency systems. Events arriving within the refractory period are queued and logged, not discarded.
DEFER Budget	A declared, consequence-weighted maximum of tolerable consecutive DEFER verdicts before the loop is required to escalate to HALT. A deployment parameter,

TERM	OPERATIONAL DEFINITION
	not a framework constant. Consecutive DEFERs attributable to the same degraded component consume the budget faster than DEFERs on unrelated components.
Inspector Independence Requirement	The architectural requirement that the Governance Inspector be executively isolated from, and semantically asymmetric to, the pipeline it evaluates -- so that a degradation or hallucination affecting the pipeline cannot simultaneously affect the mechanism verifying it.
Convoy Model	An illustrative, non-formal figure in which governance vector components are represented as train cars added, modified, or blocked at successive stations. Used for stakeholder communication; not a substitute for RSG's formal vector notation.
Heartbeat Suppression	A failure mode in which the heartbeat mechanism stops firing -- or fires without triggering an actual GPL cycle -- while the surrounding system continues to execute.
Loop Starvation	A failure mode in which the Reconstructability Test never resolves to a verdict, and the system stalls indefinitely without escalating to human review or a safe halt state.
False Green Light	A failure mode in which a GPL cycle issues CONTINUE despite an unresolved or blocked signal inherited from the underlying governance vector.
Convoy Drift	The loop-cadence analogue of RSG's Recursive Drift Amplification: small verification gaps at successive heartbeats compound across many cycles without any single heartbeat individually failing.
Vanity Heartbeat	A heartbeat that fires on schedule and returns CONTINUE but whose underlying Reconstructability Test was evaluated at reduced depth, producing the appearance of verification without its substance.
Silent Decoupling	A failure mode in which the loop's view of governance state diverges from the live execution pipeline: the loop verifies a stale or cached state while downstream decisions have already been taken on newer, unverified state.

1. Introduction

1.1 From Boundary Qualification to Continuous Verification

RSG v1.1 formalized governance as transformation qualification at the moment of a boundary crossing: a discrete event at which a governance vector is emitted, aligned, split-confirmed, and crystallized. This is the correct unit of analysis for a single crossing. It leaves open a distinct and equally important question: in a system composed of many crossings executed in sequence -- an agentic pipeline, a multi-step orchestration graph, a long-running autonomous process -- what mechanism ensures that the property RSG cares about, accountability survivability, is being checked continuously rather than only when a crossing happens to occur.

Governance Loop Engineering answers this by introducing a verification cadence that sits above individual crossings: the Governance Heartbeat. Each heartbeat triggers a Governance Preservation Loop cycle, which reuses RSG's own primitives -- Δ_s , C_p , Split Confirmation -- but applies them repeatedly, as a condition for continuation, rather than once, as a condition for crossing.

1.2 Position Relative to RSG

G-LOOP-E does not modify, extend, or reinterpret RSG's formal object. It does not add new components to the governance vector $G^n(t)$, and it does not alter the transition function T . G-LOOP-E is a consumer of RSG's outputs, not a revision of RSG's theory. The relationship is strictly layered:

RSG defines what must survive a boundary crossing. G-LOOP-E defines how a running system continuously verifies, at a cadence it controls, that it has.

This separation is deliberate and mirrors EVIDE's own boundary discipline: EVIDE anchors the declared application event, never the internal state of the system producing it. G-LOOP-E preserves the same posture -- the loop verifies observable governance state; it does not infer, interpret, or reconstruct internal system reasoning.

1.3 Position Relative to Loop Engineering

Loop Engineering, in the general sense used across agentic AI system design, refers to the construction of feedback cycles that let a system observe its own output, evaluate it against an objective, and refine its behavior - an optimization loop. Table 1 sets out the surface-level correspondence between Loop Engineering's structural stages and G-LOOP-E's, followed by the distinction that actually matters.

Loop Engineering (general)	Governance Loop Engineering
Feedback loop	Governance Preservation Loop
Iterative refinement	Recursive verification, not refinement
Convergence toward a better output	Convergence toward a defensible state
Exit condition: output quality threshold met	Exit condition: Reconstructability Test passes
State propagation carries content	State propagation carries responsibility
Failure detection: output drift, reward hacking	Failure detection: Δ_s , C_p , Convoy Drift, False Green Light

The distinction that matters is the object being optimized. Loop Engineering optimizes system behavior. G-LOOP-E optimizes nothing -- it qualifies whether responsibility for what the system has already done remains attributable, verifiable, and reconstructable. A system can have an excellent Loop Engineering cycle -- consistently improving output quality -- while its Governance Preservation Loop reports HALT, because the improved output was produced by an authority chain that no longer survives independent reconstruction. The two loops answer unrelated questions and must not be collapsed into one.

2. The Governance Heartbeat

2.1 The Reformulated Question

A naive continuous-verification design would ask, at each cycle: “is accountability preserved?” This phrasing is weaker than it appears, because it invites a binary, present-tense self-assessment that a degraded system can satisfy trivially -- accountability can appear preserved from inside the process while already being unrecoverable from outside it. G-LOOP-E replaces this question with a stronger, externally-anchored formulation:

Can accountability up to this point still be independently reconstructed if the process were interrupted immediately after this iteration?

This reformulation matters for three reasons. First, it is falsifiable against an external reference -- the evidentiary chronology -- rather than against the system's own internal state, consistent with EVIDE's independence requirement. Second, it makes the loop's verdict time-anchored to the worst case (immediate interruption) rather than to the expected case (normal continuation), which is the correct posture for a governance mechanism. Third, it is structurally identical to RSG's own Accountability Survivability definition, so a GPL cycle is not introducing a new accountability standard -- it is repeatedly testing the same standard RSG already defined, at a cadence RSG itself does not specify.

2.2 Heartbeat vs. Single-Shot Verification

A single boundary crossing verification, as specified in RSG, answers the Reconstructability Test once, at crossing time. Between two crossings, however, a system may execute many internal steps -- tool calls, sub-agent delegations, intermediate state mutations -- during which no crossing occurs and therefore no RSG-level verification is triggered. The heartbeat closes this gap. It is not bound to boundary crossings; it is bound to a cadence chosen by the deployment (fixed interval, per-step, per-token-budget, or event-triggered), making the frequency of verification an explicit governance parameter rather than an artifact of how many boundaries a given pipeline happens to contain.

2.3 Formal Cycle Definition

Heartbeat(t_i) \rightarrow GPL($G(t_i)$) \rightarrow { CONTINUE, DEFER, HALT }

At each heartbeat t_i , the current governance state $G(t_i)$ is submitted to a Governance Preservation Loop cycle, which returns one of three exit verdicts.

The loop does not produce a new governance state in the sense that $T()$ does in RSG -- it does not transform $G(t_i)$. It classifies it. This is an important asymmetry: RSG's transition function T governs how state changes across a

crossing; G-LOOP-E's loop governs whether the system is permitted to keep operating on whatever state currently exists.

2.4 Scheduled vs. Triggered Heartbeats

A purely periodic heartbeat leaves a blind window between two scheduled cycles -- exactly the interval in which Convoy Drift and Silent Decoupling (Sections 5.4, 5.6) accumulate undetected. G-LOOP-E therefore defines two heartbeat classes, not one.

- Scheduled heartbeat: issued at a fixed temporal or step-based interval, independent of any specific change in governance state. This is the baseline cadence discussed in Section 2.2.
- Triggered heartbeat: issued immediately, outside the scheduled interval, in response to a defined state-change event.

Five event classes force a triggered heartbeat, all drawn directly from RSG's own observable dimensions:

- An authority attribution changes.
- A threshold constraint set changes.
- A new unresolved or blocked signal appears.
- Δs increases relative to the last confirmed baseline.
- C_p decreases.

A triggered heartbeat always takes priority over the scheduled cadence: it does not wait for the next scheduled cycle, and its verdict resets the interval clock for the following scheduled heartbeat. A G-LOOP-E deployment that defines only scheduled heartbeats and no triggers is not incorrect, but it accepts, by construction, a detection lag bounded by the interval length -- a tradeoff that should be explicit and documented, not incidental.

Trigger Deadband and Refractory Period

Two of the five trigger classes -- an increase in Δs and a decrease in C_p -- are continuous quantities that fluctuate naturally during normal operation, particularly in high-frequency multi-agent negotiation. A trigger rule that fires on any increase, however small, converts ordinary parameter noise into a cascade of high-priority, high-depth heartbeats: a resonance effect in which the governance mechanism itself saturates the system it is meant to protect -- a self-inflicted denial of service at the governance layer. Two bounding mechanisms prevent this.

- Trigger Deadband: each continuous trigger carries a declared sensitivity threshold δ_{trig} . Fluctuations in Δs or C_p smaller than δ_{trig} do not fire a triggered heartbeat. Critically, sub-threshold events are still logged as evidentiary entries -- suppressed, not discarded -- and the deadband value itself is declared in the evidentiary record. An undeclared deadband, or one silently widened at runtime, is indistinguishable from Heartbeat Suppression (Section 5.1) and must be treated as such: the deadband is a governed parameter, not an operator convenience.
- Refractory Period: after a triggered heartbeat fires on a given governance vector component, a declared minimum interval must elapse before the same component can fire another trigger. Events arriving within the refractory period are queued and evaluated at the next heartbeat (triggered or scheduled), not discarded. This bounds the maximum trigger rate per component without ever losing the underlying signal.

The three discrete trigger classes -- authority change, threshold change, new unresolved signal -- are not subject to the deadband, since they are state transitions rather than continuous fluctuations; they remain immediate and unconditional. Both δ_{trig} and the refractory interval are deployment parameters calibrated against consequence severity, following the same discipline as ϵ , θ , and the heartbeat interval itself.

3. The Governance Preservation Loop (GPL)

3.1 Loop Structure

Formally, a GPL cycle triggered at heartbeat t_i proceeds through seven steps, directly reusing RSG's observable dimensions:

- 1. Observe current governance state $G(t_i)$.
- 2. Compute semantic divergence Δs against the last confirmed baseline.
- 3. Evaluate causal persistence C_p .
- 4. Verify authority continuity (no unattributed expansion since last heartbeat).
- 5. Check threshold validity against currently applicable constraints.
- 6. Scan for unresolved or blocked signals inherited from prior crossings.
- 7. Apply the Reconstructability Test and issue an exit verdict.

Steps 2 through 6 are not new measurements -- they are RSG's own Δs , C_p , authority, threshold, and Split Confirmation machinery, invoked repeatedly rather than once. G-LOOP-E's contribution is exclusively in step 7 and in the cadence that triggers steps 1 through 6 in the first place.

3.2 Exit Conditions

Verdict	Meaning	System behavior
CONTINUE	Reconstructability Test passes; no unresolved signal threatens survivability.	Loop proceeds to next iteration without intervention.
DEFER	Test result is inconclusive, or a degraded (not blocked) signal is present.	Execution may continue within declared scope; the deferral is recorded and surfaced for review at the next natural checkpoint.
HALT	A blocked signal or a failed Reconstructability Test is detected.	Execution must stop or fall back to a safe state before any further consequence propagates. Escalation to human review is required, not optional.

The HALT verdict inherits directly from RSG's Section 6.3 clarification on blocked signals: a blocked signal does not permit silent continuation, and G-LOOP-E enforces the same discipline at every heartbeat, not only at boundary crossings. A loop that allows continuation past a HALT verdict is, by definition, producing a False Green Light.

The DEFER Budget

DEFER carries a structural risk that CONTINUE and HALT do not: it permits execution to proceed while a known degradation stands unresolved. A single DEFER is the correct verdict for a tolerable, declared degradation. A sequence of consecutive DEFERs is something else -- it is the loop repeatedly observing an unhealed state and repeatedly authorizing continuation anyway, which is precisely the accumulation pattern of Convoy Drift (Section 5.4). If an irreversible consequence propagates during such a sequence, accountability remains reconstructable after the fact, but survivability in real time has failed: the mechanism observed the degradation and did not stop the action.

G-LOOP-E therefore requires every deployment to declare a DEFER Budget: the maximum number of consecutive DEFER verdicts the loop may issue before it is required to escalate to HALT, independent of what any individual cycle's evaluation would otherwise conclude. Two rules govern the budget:

- The budget is a consequence-weighted deployment parameter, not a framework constant -- following the same discipline as ϵ , θ , and N_{max} . Low-reversibility, high-consequence pipeline stages warrant a budget of one or two; routine, fully reversible stages may tolerate more. What the framework mandates is that the budget exists, is declared, and is enforced -- not its specific value.
- Consecutive DEFERs attributable to the same degraded component consume the budget faster than DEFERs on unrelated components -- a repeated DEFER on the same component is the loop watching a single degradation persist, which is a materially stronger Convoy Drift signal than independent, transient degradations on different components.

A budget-forced HALT is recorded as such in the evidentiary chronology (`escalation_source = defer_budget_exhausted`), distinct from a HALT produced by a blocked signal or a failed Reconstructability Test -- so that post-event reconstruction can distinguish a hard governance failure from a preventive escalation.

3.3 Mapping to Split Confirmation

Each governance vector component evaluated within a GPL cycle inherits its Split Confirmation status directly from the underlying RSG state, and this status determines what the loop is permitted to conclude:

Split Confirmation status	Loop interpretation
✓ CONFIRMED	Component contributes positively to a CONTINUE verdict.
⚠ DEGRADED	Component alone cannot block continuation but caps the verdict at DEFER.
□ UNVERIFIABLE	Component is excluded from the Reconstructability Test computation and flagged; accumulation of unverifiable components across heartbeats is itself a Convoy Drift signal.
✗ BLOCKED	Component forces a HALT verdict regardless of the status of any other component.

3.4 Verification Depth

Heartbeat frequency and verification depth are independent axes. A heartbeat's frequency determines how often a GPL cycle fires; its depth determines how much of the governance vector that cycle actually evaluates. Conflating the two is the direct mechanism behind the Vanity Heartbeat failure mode (Section 5.5): a heartbeat

can fire on schedule, at high frequency, while checking almost nothing. G-LOOP-E defines four canonical depth levels.

Depth level	Components evaluated	Typical use
D0 -- Authority only	Authority vector A^n	High-frequency scheduled heartbeats in low-consequence stretches of a pipeline.
D1 -- Authority + Threshold	A^n, T^n	Routine scheduled heartbeats where normative constraints are active.
D2 -- Authority + Threshold + Continuity	A^n, T^n, C^n	Heartbeats following any triggered event (Section 2.4), or scheduled heartbeats at moderate-consequence stages.
D3 -- Full governance vector	$D^n, A^n, I^n, T^n, C^n, E^n$	Every heartbeat immediately preceding an execution boundary, and any heartbeat used to compute a Convoy Drift baseline comparison.

Depth is not required to be uniform across a run. A deployment may run frequent D0 heartbeats between two D3 heartbeats anchoring the start and end of a stage -- provided this scope choice is itself declared in the evidentiary record at each cycle, as required by the Vanity Heartbeat detection criterion in Section 5.5. An undeclared narrowing of depth is indistinguishable, after the fact, from a Vanity Heartbeat, regardless of the operator's intent -- which is precisely why declaration, not intent, is the enforceable condition.

Triggered heartbeats (Section 2.4) should default to at least D2, with one mandatory escalation rule: the depth of a triggered heartbeat must include, at minimum, the component class that triggered it. Since D2 covers only Authority, Threshold, and Continuity, a trigger originating in the Decision domain (D^n) or the Evidentiary domain (E^n) -- for example, a new unresolved signal appearing in the evidentiary vector -- automatically and non-negotiably escalates that heartbeat to D3. A triggered heartbeat evaluated at a depth that excludes its own triggering component would not even examine the reason it fired, which defeats its purpose and constitutes a Vanity Heartbeat by construction. The D2 default therefore applies only to triggers originating within the A^n, T^n , or C^n domains; D^n - and E^n -originated triggers have no D2 option.

3.5 Inspector Independence

The Governance Inspector -- the component that evaluates the Reconstructability Test at each heartbeat -- may itself be an automated module, and in fully autonomous systems it usually will be. This creates a specific and dangerous failure surface: if the Inspector shares its model, execution context, or degradation path with the pipeline it evaluates, then a hallucination or drift affecting the pipeline can simultaneously affect the mechanism that is supposed to catch it. The result is hallucinated self-certification -- a loop that confidently returns CONTINUE precisely because the same fault has compromised both the actor and its auditor.

This is the same principle that underlies EVIDE's evidentiary independence, applied one layer up: an evidentiary anchor is only meaningful if it is external to the system it anchors. G-LOOP-E therefore imposes the Inspector Independence Requirement, with two dimensions:

- Executive isolation: the Inspector runs in an execution environment separate from the pipeline, such that a failure, resource exhaustion, or compromise in the pipeline does not propagate into the Inspector's evaluation.
- Semantic asymmetry: the Inspector's verification logic must be radically different in kind from the pipeline's reasoning -- ideally deterministic and rule-based where the pipeline is probabilistic, or at minimum a distinct model with a distinct failure profile -- so that the two are unlikely to fail in the same way on the same input. An Inspector that is simply another instance of the same model is not independent; it is a correlated observer.

Independence is a requirement, not a provable property: no architecture can guarantee two components share no failure mode whatsoever. G-LOOP-E's claim is therefore bounded -- the requirement reduces the probability of correlated failure between actor and auditor; it does not eliminate it. The residual limit is stated explicitly in Section 9, consistent with the Non-Claims Discipline inherited from RSG.

4. Architectural Overview

Figure 1. Static Loop vs. Governance Loop

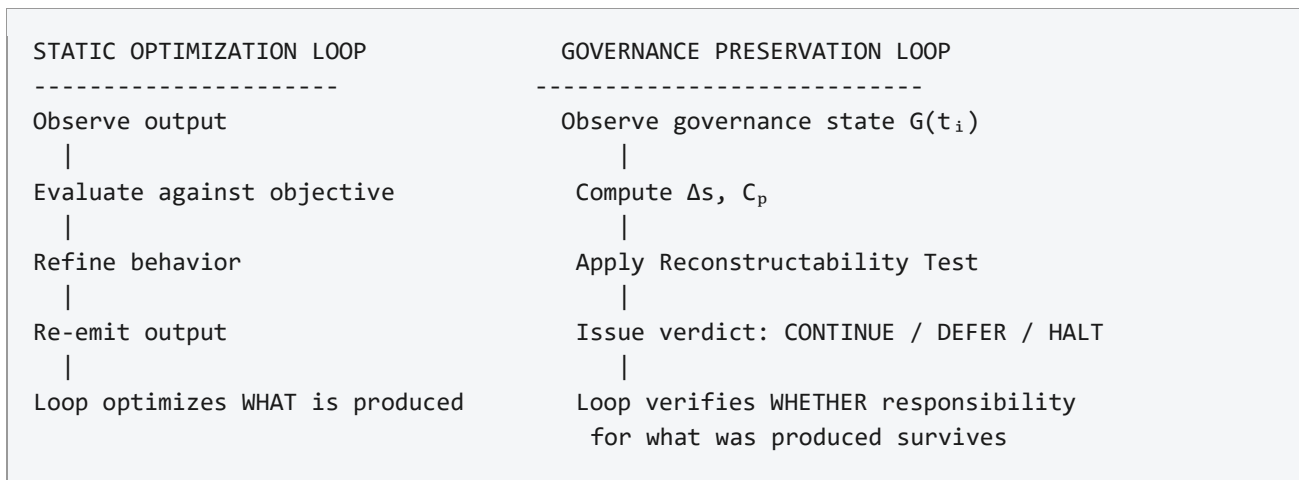
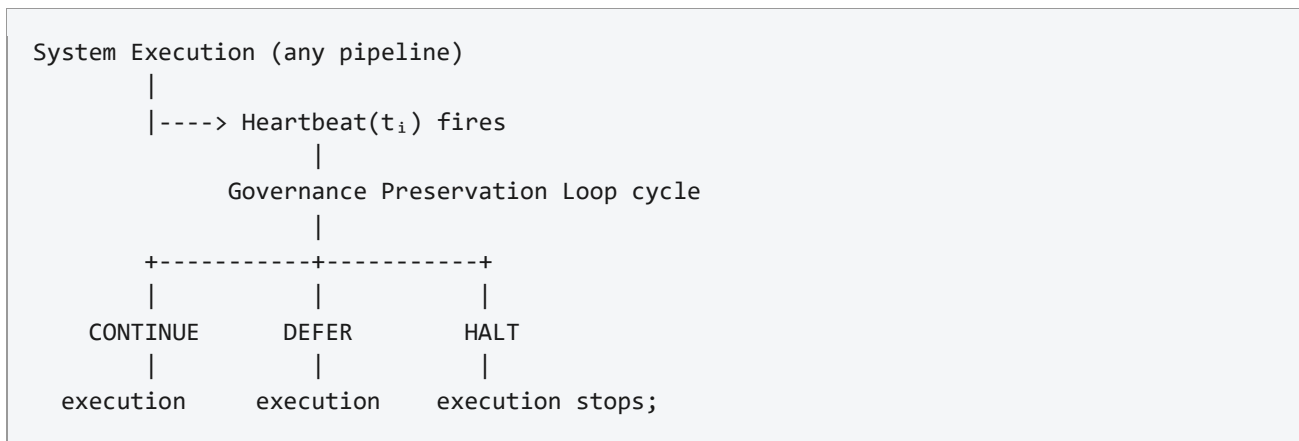


Figure 1 -- The two loop types share iterative structure but operate on different objects: output quality versus accountability survivability. They are not competing architectures; a system may run both simultaneously, in parallel, without interference.

Figure 2. Heartbeat Cycle Lifecycle



proceeds	proceeds	safe state / human
unmodified	with logged	review required
	deferral	

Figure 2 -- A heartbeat does not alter execution by itself; it gates continuation. CONTINUE and DEFER both allow execution to proceed, differing only in whether the state is flagged for later review; HALT is the only verdict that stops the pipeline.

Figure 3. The Convoy Model (illustrative)

The following figure is explicitly pedagogical. It is offered to make G-LOOP-E legible to non-technical stakeholders -- executives, auditors, regulators -- without requiring familiarity with governance vector notation. It is not a substitute for the formal definitions above, and no term introduced in this figure ("car", "station", "inspector") should be treated as canonical G-LOOP-E terminology outside this illustrative context.

STATION 1	STATION 2	STATION 3	STATION 4
(L1 Decision)	-> (L2 Continuity)	-> (L3 Evidentiary)	-> (Consequence)
[Car: Decision]	[Car: Decision]	[Car: Decision]	
[Car: Authority]	[Car: Authority]	[Car: Authority]	
[Car: Threshold]	[Car: Threshold]-x	BLOCKED, cannot	
	(degraded)	couple further	
Inspector asks at every station:			
"Can this convoy still be reconstructed if it stops here?"			
YES -> green signal, convoy departs			
NO -> red signal, convoy held at station			

Figure 3 -- Each governance vector component is represented as a car. A blocked car (X, red signal) prevents departure regardless of how many other cars are confirmed -- the same non-negotiable rule as the formal HALT verdict in Section 3.2. The convoy does not carry data between stations; it carries responsibility, and each station's inspector may only pass forward exactly the responsibility the convoy legitimately holds -- no more, no less.

5. Loop-Specific Failure Modes

RSG catalogs eight failure modes proper to boundary crossings (Semantic Inflation, Authority Hallucination, Synthetic Coherence, Stabilization by Timeout, Continuity Collapse, Recursive Drift Amplification, False Crystallization, Governance Deadlock). These remain fully applicable within G-LOOP-E, since every GPL cycle operates on RSG governance vectors. G-LOOP-E adds six additional failure modes specific to continuous, cadence-driven verification, none of which are expressible in a single-crossing model.

5.1 Heartbeat Suppression

Definition: the heartbeat mechanism stops firing, or fires without actually invoking a GPL cycle, while the surrounding system continues to execute. The system appears governed because a heartbeat schedule exists on paper; no verification is actually occurring.

Detection: heartbeat emission must itself be logged as an evidentiary event, independent of its verdict. An expected heartbeat with no corresponding log entry is, by definition, a suppression event.

5.2 False Green Light

Definition: a GPL cycle returns CONTINUE despite a blocked or unresolved signal present in the underlying governance vector. This is the loop-level instance of RSG's non-expansion violation, applied to the loop's own verdict rather than to a boundary connector's transformation.

Detection: verdict computation must be re-derivable independently from the logged component states; a CONTINUE verdict that cannot be reproduced from a \times BLOCKED component in the log is a False Green Light by construction.

5.3 Loop Starvation

Definition: the Reconstructability Test never resolves to a verdict -- the loop remains perpetually in an unresolved evaluation state, and the system neither proceeds under an explicit CONTINUE/DEFER nor stops under HALT. Unlike RSG's Governance Deadlock, which involves circular dependency between layers, Loop Starvation can occur within a single loop with no external counterpart.

Detection: every GPL cycle must carry a bounded evaluation window (analogous to RSG's N_{\max}); a cycle that exceeds this window without a verdict is required to default to HALT, not to CONTINUE. A loop implementation that defaults to CONTINUE on timeout is itself a Stabilization-by-Timeout failure inherited from RSG.

5.4 Convoy Drift

Definition: the loop-cadence analogue of Recursive Drift Amplification. Small, individually-tolerable degradations accepted at successive heartbeats -- each producing a locally valid DEFER -- compound across many cycles into a cumulative loss of reconstructability that no single heartbeat's verdict reveals.

Detection: exactly as RSG requires Δ_s to be computed against the L1 baseline rather than only the immediately prior layer, G-LOOP-E requires the Reconstructability Test to be periodically re-evaluated against the earliest confirmed heartbeat in the run, not only against the immediately preceding heartbeat.

5.5 Vanity Heartbeat

Definition: a heartbeat fires on schedule and returns a verdict, but is evaluated at a verification depth (Section 3.4) lower than the situation warrants -- for example, a D0 Authority-only check immediately preceding an execution boundary, where D3 is required -- producing the appearance of verification without its substance.

Detection: each heartbeat's evidentiary record must declare its verification depth (D0-D3); a pattern of depth narrowing over time, especially correlated with system load, or any triggered heartbeat evaluated below D2, is a Vanity Heartbeat signature.

5.6 Silent Decoupling

Definition: the loop's view of governance state diverges from the live execution pipeline -- the loop verifies a stale or cached governance vector while downstream components have already acted on newer, unverified state. The

loop's CONTINUE verdicts remain individually correct with respect to what the loop observed; they are simply no longer describing the system that is actually running.

Detection: heartbeat timestamps must be compared against the timestamp of the governance state they evaluated; a persistent or growing gap between the two is a Silent Decoupling signal, independent of the verdict returned. In distributed or asynchronous multi-agent deployments, however, absolute timestamps alone are an unreliable detection basis: clock skew and network latency can produce apparent gaps that reflect infrastructure, not governance misalignment -- yielding false Silent Decoupling positives. Detection in such deployments must therefore be based on logical sequence identifiers (monotonic state counters or Lamport-style logical clocks) attached to each governance state at emission, used alongside or in place of wall-clock timestamps: the loop evaluating a state whose sequence number is lower than the sequence number the execution pipeline has already acted upon is decoupled, regardless of what the wall clocks say.

6. Relationship to Orchestration Frameworks

RSG Section 10b positioned RSG's boundary-trained connectors relative to MCP, LangChain/LangGraph, and AutoGen. G-LOOP-E extends that positioning by specifying where the loop itself attaches within each of these architectures.

6.1 MCP and Tool-Call Boundaries

In an MCP-based system, each tool call is already an RSG boundary crossing. G-LOOP-E attaches a heartbeat to the tool-call cadence itself, independent of whether any individual call constitutes a governance-relevant crossing -- so that a sequence of many rapid, individually low-stakes tool calls is still subject to periodic Reconstructability Testing, rather than only to per-call boundary checks that a Convoy Drift pattern could exploit.

6.2 LangGraph and Multi-Step Pipelines

LangGraph-style pipelines compose discrete nodes into a directed execution graph. G-LOOP-E places a heartbeat at the graph level, not only at each node transition: a graph-level heartbeat can detect Convoy Drift across the full path from entry to exit, which a per-node check, however rigorous, cannot see by construction.

6.3 AutoGen and Multi-Agent Conversation Graphs

In autonomous multi-agent negotiation, where no human observes every turn, the heartbeat becomes the only continuous governance presence in the system. G-LOOP-E recommends heartbeat cadence tied to conversation turns rather than wall-clock time in this setting, since Convoy Drift in multi-agent graphs accumulates per exchange, not per second.

7. End-to-End Walkthrough: Multi-Step Agentic Orchestration

We apply G-LOOP-E to a representative agentic pipeline: Planner → LLM → Tool → API → Database → Second Agent → Human Review. Each arrow is a potential RSG boundary crossing; G-LOOP-E adds a heartbeat after every stage, regardless of whether that stage's crossing was itself flagged by RSG.

7.1 Stage-by-Stage Heartbeat Behavior

Stage transition	Heartbeat observation	Verdict
Planner → LLM	Authority for the plan is attributed; no unresolved signals.	CONTINUE
LLM → Tool	Decision vector provisional; threshold attribution partial (tool scope not fully declared).	DEFER, logged
Tool → API	Threshold vector remains degraded from prior stage; no new degradation introduced.	DEFER, cumulative
API → Database	A write operation is attempted under an authority vector that was never explicitly confirmed for write scope -- non-transferable signal.	HALT
Database → Second Agent	Not reached: pipeline halted at prior stage.	-
Second Agent → Human Review	Not reached.	-

7.2 Interpreting the Outcome

This is the intended behavior, not a failure of the pipeline. Two consecutive DEFER verdicts accumulated a degraded threshold vector without any single stage individually failing -- a textbook Convoy Drift pattern, and one that consumed the deployment's DEFER Budget (Section 3.2) because both DEFERs were attributable to the same persisting threshold degradation. In this walkthrough the explicit HALT arrived at the API-to-database transition on an independent blocked signal (an authority vector never confirmed for write scope); had that blocked signal not appeared, the DEFER Budget would itself have forced a preventive HALT at the next cycle, before the database write, purely on the strength of the repeated same-component degradation. Either path stops the system before the write executes. The system escalates to human review with the full heartbeat trail -- including the two prior DEFER records and the budget-exhaustion marker -- attached as evidentiary chronology. The human reviewer receives exactly the accumulated context needed to authorize or reject the write, not a single isolated error.

8. Computational Considerations

G-LOOP-E introduces overhead proportional to heartbeat frequency, since each heartbeat invokes a full or partial GPL cycle. Four calibration levers bound this cost, extending RSG Section 11b's recursion-depth constraints to the loop-cadence dimension:

- Heartbeat interval: shorter intervals catch Convoy Drift earlier but increase overhead; longer intervals reduce cost but widen the window in which drift can accumulate undetected.
- Scope-adaptive heartbeats: not every heartbeat needs to re-evaluate every governance vector component; components confirmed at the last heartbeat and unchanged since may be carried forward without re-verification, provided this is declared (to avoid Vanity Heartbeat).

- Consequence-weighted cadence: heartbeat frequency should scale with consequence severity and reversibility horizon, exactly as RSG's θ threshold scales with the same parameters -- high-stakes stages warrant tighter heartbeat spacing than routine ones.
- Baseline re-anchoring frequency: Convoy Drift detection requires periodic comparison against the earliest confirmed heartbeat; this comparison need not run at every cycle, but its own interval is a separate, explicit parameter, not an afterthought.

As with RSG, G-LOOP-E is not designed as a zero-latency mechanism. It is designed for pipelines where the cost of an unrecoverable authority or continuity failure exceeds the cost of heartbeat latency -- which describes most agentic systems operating with real-world consequence, and very few purely exploratory or sandboxed ones.

9. Limitations

- Heartbeat depth vs. frequency tradeoff: this paper does not yet specify an empirical method for jointly optimizing interval and scope; both remain deployment-calibrated parameters.
- Inherited baseline dependency: Convoy Drift detection, like Recursive Drift Amplification in RSG, depends on the integrity of the earliest confirmed heartbeat as baseline. If that baseline itself encodes an undetected error, G-LOOP-E will measure drift from that error consistently but cannot detect the original error -- the same limitation RSG discloses in Section 12, propagated one layer up.
- Governance Inspector reliability: automated evaluation of the Reconstructability Test is bounded by the reliability of the underlying semantic classification mechanisms, identically to RSG's own disclosed limitation on automatic violation detection.
- Residual correlated-failure risk in the Inspector: the Inspector Independence Requirement (Section 3.5) reduces but cannot eliminate the probability that the Inspector and the pipeline fail in the same way on the same input. No architecture can prove two components share no failure mode. G-LOOP-E's guarantee is therefore probabilistic, not absolute -- a sufficiently pervasive fault could in principle compromise both actor and auditor despite executive isolation and semantic asymmetry.
- Convoy Model is illustrative only: the train metaphor in Figure 3 has no formal semantics and must not be used as a substitute for the governance vector notation in Sections 2-3 in any implementation or compliance context.
- High-throughput, low-stakes contexts: G-LOOP-E's overhead may be disproportionate in settings where individual iterations carry negligible consequence; the framework does not yet specify a principled threshold below which heartbeats should be suppressed rather than merely widened.
- Empirical validation: the six loop-specific failure modes in Section 5 are derived analytically from RSG's existing taxonomy and have not yet been validated against production incident data.

10. Future Research Directions

- Empirical calibration of heartbeat interval against Convoy Drift incidence across deployment classes.
- Formal specification of scope-adaptive heartbeats that provably prevent Vanity Heartbeat while bounding overhead.

- Automated Silent Decoupling detection through timestamp-lineage verification integrated with EVIDE's evidentiary chronology.
- Extension of the Convoy Model into a fourth canonical RSG-family diagram, formally cross-referenced but kept explicitly separate from the vector notation.
- Cross-organizational heartbeat federation: continuous verification across accountability boundaries that span separate legal entities.
- Governance Heartbeat standards for agent-to-agent protocols, extending RSG Section 10b.3's AutoGen positioning into a concrete heartbeat specification.
- Governance Recovery as a distinct research track: whether a fourth loop verdict beyond CONTINUE / DEFER / HALT -- provisionally, REPAIR -- could allow a system to actively re-establish a degraded governance component (e.g. requesting a missing attestation, forcing a fresh boundary crossing for an unresolved signal) before re-evaluation, rather than only halting and waiting for external intervention. This is explicitly out of scope for G-LOOP-E v1.0: it changes the loop from a verification mechanism into a remediation mechanism, and deserves its own working paper rather than an extension of this one.

11. Conclusion

RSG established that governance is transformation qualification, not event recording, and specified what must be verified at a boundary crossing. This paper has argued that a single verification at crossing time is necessary but not sufficient for systems that execute many iterations between crossings -- and has proposed Governance Loop Engineering as the continuous verification layer that closes this gap without altering RSG's underlying formalism.

The Governance Heartbeat and the Governance Preservation Loop do not optimize system behavior. They repeatedly ask a single question, anchored to the worst case rather than the expected one: if this process stopped right now, could responsibility for what it has already done still be independently reconstructed. Where the answer is no, the loop does not log the fact and continue -- it stops the system before the consequence propagates.

The boundary is where governance succeeds or fails. The loop is what keeps asking, between boundaries, whether it still can.

References

- [1] Celano, E. (2026). Recursive Semantic Governance: Preserving Accountability Across AI Boundary Transformations. Working Paper v1.1. EVIDE Governance Lab: app.certifywebcontent.com/docs/rsg/
- [2] RecursiveMAS. (2026). Recursive Multi-Agent System Architecture. GitHub: github.com/RecursiveMAS/RecursiveMAS
- [3] Celano, E. (2026). EVIDE v2.0 Architectural Backlog and Roadmap. EVIDE Governance Lab: app.certifywebcontent.com/docs/evide-v2-roadmap/
- [4] EVIDE Governance Lab. (2026). Epistemic Stabilization Buffer: Schema v_buffer_01. lab.certifywebcontent.com
- [5] European Parliament. (2024). Artificial Intelligence Act. Official Journal of the European Union.
- [6] Celano, E. (2026). H.A.S.H.E.S. Manifesto: Human Alignment for Stability, Harmony, Ethics & Survival. certifywebcontent.com