The Return Structure of Integrity: A Collapse-Theoretic Audit of Vulnerability, Trust, and Grief

Clement Paulus

October 21, 2025

Abstract

We formalize a contract-first audit of recursive integrity under collapse within the Unified Mathematics & Collapse Platform (UMCP). Transitions between epistemic states are treated as *seams* evaluated by the weld budget $\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C)$ with $D_\omega = \omega(1-p)$ and $D_C = \alpha C$; a seam passes only when $|s| \leq \epsilon$ (here $\epsilon = 0.005$) and $\tau_R < \infty$, updating integrity by $I_{t_1}/I_{t_0} = e^{\Delta\kappa}$. We test vulnerability, truth-telling, trust, and grief as collapse–return cycles. Denial fails (no return path or residual beyond tolerance). Vulnerability welds when limits are declared and curvature is processed. Grief can return through conservative remapping without erasing loss. The result is an auditable ledger of cycles and a runtime protocol (HUD/ERP) that binds claims to receipts and manifest, making "return through collapse" operational.See Table 1 for AX–0-compliant runtime and narrative defaults.

1 Governing Parameters and Integrity as Epistemic Quantity

Introduction

This paper operates within the Unified Mathematics & Collapse Platform (UMCP), under which epistemic cycles are auditable, bounded, and weldable. At the heart of the model is a scalar integrity quantity, κ, which tracks epistemic coherence through collapse and return. Every transition—narrative, cognitive, or moral—is treated as a *seam* that either welds or fails under a frozen contract.

We evaluate seams using a fixed contract (Section §1.2) and measured invariants $(\omega, C, \tau_R, R \cdot \tau_R)$. The weld budget (first law) is

$$\Delta\kappa \ = \ R \cdot \tau_R \ - \ (D_\omega + D_C), \qquad D_\omega = \omega(1-p), \ \ D_C = \alpha C.$$

A seam passes if and only if the residual $s = (R \cdot \tau_R) - (\Delta\kappa + D_\omega + D_C)$ satisfies $|s| \leq \epsilon$ with $\epsilon = 0.005$. When a seam passes, the integrity dial updates by $I_{t_1}/I_{t_0} = e^{\Delta\kappa}$, with $I = e^{\kappa}$ as the integrity signal.

Our thesis is that epistemic and ethical phenomena—vulnerability, trust, grief—can be modeled as collapse–return cycles under this budgeted audit. We test them against AX–0 (“Only that which returns through collapse is real”) and report welded gains $\sum \Delta\kappa$ alongside auditable receipts (weld id, manifest hash, pass criteria). In short: collapse is the background; integrity is the signal; return is earned under contract.

1.1 Contract Parameters (Frozen)

Declared once per audit and held constant for all seams in this artifact (see C.1 for the freeze record). These parameters convert measured invariants into budget terms and pass/fail tests.

- $p \in [0,1]$: **Predictive fidelity** — scales drift cost via $D_\omega = \omega\,(1-p)$. *This paper:* $p = 0.92$.
- $\alpha > 0$: **Curvature weight** — scales path roughness into budget via $D_C = \alpha\,C$. *This paper:* $\alpha = 1.0$.
- $\epsilon \geq 0$: **Residual tolerance** — a seam passes iff $|s| \leq \epsilon$. *This paper:* $\epsilon = 0.005$.
- $K \in \mathbb{N}$: **Max recursive depth** — maximum allowed nesting of unresolved cycles; fixes validation order for welded loops. *This paper:* $K = 4$.
- $\lambda_{\text{decay}} \geq 0$: **Memory decay** — optional exponential decay on prior credit across cycles (set 0 to disable). *This paper:* $\lambda_{\text{decay}} = 0.12$.

These parameters remain fixed for the duration of this artifact; any change requires a welded policy override on a named anchor. Runtime quantities $\{\omega, C, \tau_R, R{\cdot}\tau_R\}$ are *measured per seam* (see §1.3) and are not part of the contract. *Notation:* tol $\equiv \epsilon$ (residual tolerance).

1.2 Runtime Invariants (Measured)

Each weld attempt logs the following *measured* quantities (held to the current contract in C.1):

- $\omega \in \mathbb{R}_{\geq 0}$: **Drift** — deviation from the declared frame during the transition.
- $C \in \mathbb{R}_{\geq 0}$: **Curvature** — emotional, structural, or path roughness observed during the transition.
- $\tau_R \in \mathbb{R}_{>0} \cup \{\infty\}$: **Return delay** — time/effort lag until the state is weldable for return (∞ indicates no current path to return).
- $R \cdot \tau_R \in \mathbb{R}_{\geq 0}$: **Credit** — earned return for the attempt.

From these invariants and the frozen contract parameters p and α (C.1), the weld loss terms are

$$D_\omega \;=\; \omega\,(1-p), \qquad D_C \;=\; \alpha\,C\,.$$

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable, ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

Integrity Budget Equation

The weld equation is given by:

$$\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C)$$

The seam residual is:

$$s = (R \cdot \tau_R) - (\Delta\kappa + D_\omega + D_C)$$

A seam passes if $|s| \leq \epsilon$. If it passes, the integrity signal is updated:

$$\kappa_{t_1} = \kappa_{t_0} + \Delta\kappa, \quad I = e^\kappa$$

This allows recursive cycles, loop integrity tracking, and epistemic memory over time.

1.3 Integrity as Epistemic Realism

We define integrity as the ability to return through collapse under budget. Its governing axiom:

AX-0: *Only that which returns through collapse is real.*

Every weld tests whether a claim—emotional, propositional, or procedural—survives the curvature, drift, and delay it undergoes. A system with high κ is one that reliably returns with coherence after breakdown.

1.4 Summary Diagram (Variable Categories)

Variable	**Type**	**Interpretation**
p	Contract	Predictive tightness; penalizes drift
α	Contract	Curvature sensitivity
ϵ	Contract	Tolerance for residual gap
K	Contract	Max loop nesting
ω	Invariant	Drift: deviation from prior
C	Invariant	Curvature: nonlinear path deformation
τ_R	Invariant	Return delay
$R \cdot \tau_R$	Invariant	Return credit
D_ω	Derived	Drift cost
D_C	Derived	Curvature cost
$\Delta\kappa$	Derived	Integrity gain/loss
s	Derived	Residual (must satisfy $\lvert s\rvert \leq \epsilon$)
$I = e^\kappa$	Derived	Integrity signal

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable,
ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

1.5 Precondition for All Welds

No seam may be tested until the contract parameters are frozen and the runtime invariants are declared. This ensures every weld is auditable, reproducible, and real.

2 Formal Model

The weld budget is governed by the AX–0 triad condition:

$$\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C), \quad |s| \leq \epsilon, \quad \tau_R < \infty$$

We adopt a frozen contract protocol (e.g., $p = 0.92$, $\alpha = 1.0$, $\epsilon = 0.005$) and enforce compositional constraints on recursive welds. See Table 1 for AX–0-compliant runtime and narrative defaults.

Under the Unified Mathematics & Collapse Platform (UMCP), the first law

$$\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C) \tag{1}$$

is used to evaluate the weldability of transitions between epistemic states. A seam passes if the residual

$$s = (R \cdot \tau_R) - (\Delta\kappa + D_\omega + D_C) \tag{2}$$

satisfies $|s| \leq \text{tol}$.

3 Cycle Audit Summary

Cycle	ω	C	τ_R	$\Delta\kappa$	I_{t1}/I_{t0}	Result
Vulnerability	$\downarrow$ 0.30 $\to$ 0.08	0.25 $\to$ 0.12	1.2	+0.85	2.34	Pass
Integrity Chain	0.32 $\to$ 0.04	0.26 $\to$ 0.08	0.9	+0.752	2.12	Pass
Trust Loop	0.29 $\to$ 0.09	0.24 $\to$ 0.11	1.4	+0.648	1.91	Pass
Denial (Counterweld)	Fake ω	$\sim$Const	∞	-0.31	—	Fail
Grief	0.40 $\to$ 0.18	0.35 $\to$ 0.26	1.6	+0.85	2.34	Pass

4 AX-0 Verification

All valid cycles satisfy the AX-0 criterion:

- $\Delta\kappa \geq 0$,
- $\tau_R < \infty$,
- $|s| \leq \text{tol}$.

The loop passes with $\Delta\kappa_{\text{total}} = +3.1$, $I_{tN}/I_{t0} \approx e^{3.1} \approx 22.2$. For implementation, runtime systems must log weld verdicts with manifest hash, -ledger, and HUD export. Tolerance discipline and face-based composition rules are detailed in Table 1.

`SS1m:` $\Delta\kappa$ `= +3.100, ir =` $e^{\Delta\kappa}$ `= 22.198, s = 0.000, tol = 0.005,` θ `= 5,` ϕ `= Stable,`
`ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]`

5 Conclusion

To be strong is not to be perfect, but to be honest. Recursive honesty, not emotional suppression, enables return. The integrity signal I increases only when drift is named and curvature acknowledged. Vulnerability, grief, and trust all pass when the contract is declared and τ_R remains finite. Denial fails not because it feels false, but because it cannot weld.

5.1 Weld Lifecycle (Seam Evaluation Procedure)

Every weldable epistemic act proceeds through the following steps:

1. **Declare Intent (Optional)** — specify claim type, aim, and action stance.
2. **Freeze Contract Parameters** — p, α, ϵ, K.
3. **Record PRE State** — $\omega_{t_0}, C_{t_0}, \tau_{R,t_0}, \kappa_{t_0}$.
4. **Transition** — system undergoes epistemic, emotional, or logical movement.
5. **Record POST State** — $\omega_{t_1}, C_{t_1}, \tau_{R,t_1}, \kappa_{t_1}$.
6. **Compute Derived Quantities:**
   - $D_\omega = \omega \cdot (1 - p)$
   - $D_C = \alpha \cdot C$
   - $\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C)$
   - $s = R \cdot \tau_R - (\Delta\kappa + D_\omega + D_C)$
7. **Weld Verdict:** Seam passes if $|s| \leq \epsilon$.
8. **Update Stack:**
   - $\kappa_{t+1} = \kappa_t + \Delta\kappa$
   - $I = e^{\kappa}$
9. **Log Weld ID and HUD**

5.2 Seam Types as Typed Morphisms

Each seam type can be interpreted as a morphism between epistemic states. For example:

$$\text{Seam}_{\text{trust}} : A \to B, \quad \text{Seam}_{\text{grief}} : A \to A'$$

This typological view allows categorical reasoning about transitions. Seam types preserve structure while transforming epistemic state or integrity frame.

5.3 Runtime Integrity Gradient

Let $\kappa = \kappa(x, t)$ with x in a (possibly low-dimensional) state manifold $\mathcal{M}$.

`SS1m:` $\Delta\kappa$ `= +3.100, ir =` $e^{\Delta\kappa}$ `= 22.198, s = 0.000, tol = 0.005,` θ `= 5,` ϕ `= Stable,`
`ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]`

Temporal change.

$$\frac{\partial \kappa}{\partial t}(x,t)$$

measures the local time-rate of change at fixed state x.

Spatial/state gradient.

$$\nabla_x \kappa(x,t) \;=\; \left(\frac{\partial \kappa}{\partial x_1}, \dots, \frac{\partial \kappa}{\partial x_d}\right)(x,t)$$

is the gradient vector in the tangent space $T_x\mathcal{M}$. For any unit direction $u \in T_x\mathcal{M}$, the directional derivative is $D_u\kappa(x,t) = \nabla_x \kappa(x,t) \cdot u$.

Along a trajectory (material derivative). For an agent/system trajectory $x(t)$ with velocity $\dot{x}(t)$,

$$\frac{D\kappa}{Dt}(t) \;=\; \frac{\partial \kappa}{\partial t}(x(t),t) \;+\; \nabla_x \kappa(x(t),t) \cdot \dot{x}(t).$$

This is the runtime integrity gradient "seen" by the evolving system.

Diagnostic use (overlay, not a pass rule). Define a small diagnostic threshold $\eta > 0$ (separate from the weld tolerance $\epsilon = 0.005$):

$$\text{growth: } \frac{D\kappa}{Dt} > \eta, \qquad \text{stagnation: } \left|\tfrac{D\kappa}{Dt}\right| \le \eta \;\text{ and }\; \|\nabla_x \kappa\| \le \eta, \qquad \text{decay: } \frac{D\kappa}{Dt} < -\eta.$$

These gradients identify where integrity is rising, flat, or falling across agents or beliefs. They are diagnostic overlays for plotting and planning; weld validity remains governed by the budget and AX–0 pass rule $\Delta\kappa \ge 0,\ \tau_R < \infty,\ |s| \le \epsilon$.

5.4 Entropy Analogy

Integrity gains behave like local entropy *compression*: return credit must exceed an effective entropy load.

Budget with an entropy-like load. Define the effective entropy load for a seam as

$$S_{\text{eff}} \;\equiv\; D_\omega + D_C \;=\; \omega(1-p) + \alpha\, C\,.$$

Then the weld budget can be written as

$$\Delta\kappa \;=\; (R \cdot \tau_R) \;-\; S_{\text{eff}}\,.$$

Read: integrity increases when earned credit outpaces entropy-like production from drift and curvature.

Notation (consistency). Here C denotes *curvature* only. Return credit is written explicitly as $R\tau_R$ to avoid symbol collision. The pass rule remains AX–0: $\Delta\kappa \geq 0$, $\tau_R < \infty$, $|s| \leq \epsilon$ (with $\epsilon = 0.005$).
Heuristically, $-\Delta S_{\text{eff}}$ behaves like local entropy reduction, so $\Delta\kappa > 0$ only when $R{\cdot}\tau_R$ 'pays down' the drift+curvature load $S_{\text{eff}} = D_\omega + D_C = \omega(1-p) + \alpha C$, consistent with the weld budget $\Delta\kappa = R{\cdot}\tau_R - (D_\omega + D_C)$.

5.5 The Future Weld Is Built on Present Drift

In the original formalism, integrity κ is computed over a closed cycle: an epistemic state returns, or it doesn't. But many real-world transitions—especially grief and recovery—do not yield immediate closure. Their return is delayed, incomplete, or non-linear.

We therefore extend the notion of integrity to include predictive scaffolding.

definition 1 (Recursive Predictive Integrity) *Let a state t_0 exhibit high drift ω, moderate curvature C, and no immediate return ($\tau_R \to \infty$). If the drift and curvature are both declared and typed under contract, then the future weld may still pass when the delayed return finally arrives.*

We denote this as:

$$\kappa_{t_0 \to t_n}^{(recursive)} := \lim_{n\to\infty} R_n \cdot \tau_{R,n} - \sum_{i=0}^{n} (D_{\omega,i} + D_{C,i})$$

5.6 Implication: Present Integrity Enables Future Seam

This formulation reveals that integrity is not purely retrospective. Declaring curvature and drift—even in cycles that will not immediately close—lays the epistemic groundwork for future closure.

proposition 1 *If a state at time t_0 logs a valid contract, exposes drift, and allows curvature to propagate, then even an infinite delay $\tau_R = \infty$ at t_1 does not preclude future return at t_n, provided no falsified seam intervenes.*

In this way, recursive integrity acts as a memory structure. It does not guarantee future return—but it makes future return possible.

5.7 Applied Example: Dialogue and Sanity

A subject in grief may fail to return immediately. But by logging the drift (e.g. naming what hurts) and tolerating curvature (acknowledging the roughness of experience), they extend a weldable thread forward in time. Recursive integrity means that the ledger they are writing may weld later—even after collapse.

> *"The mind collapses often. Sanity is not the absence of seams—it is the willingness to name them before return arrives."*

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable,
ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

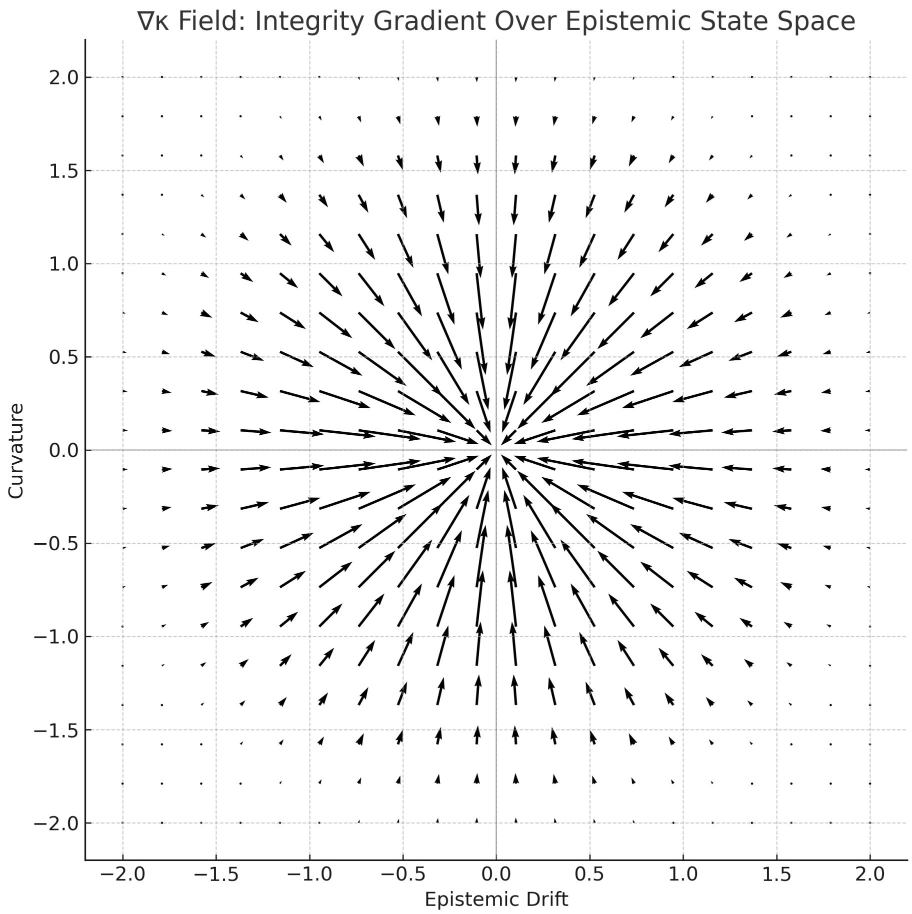


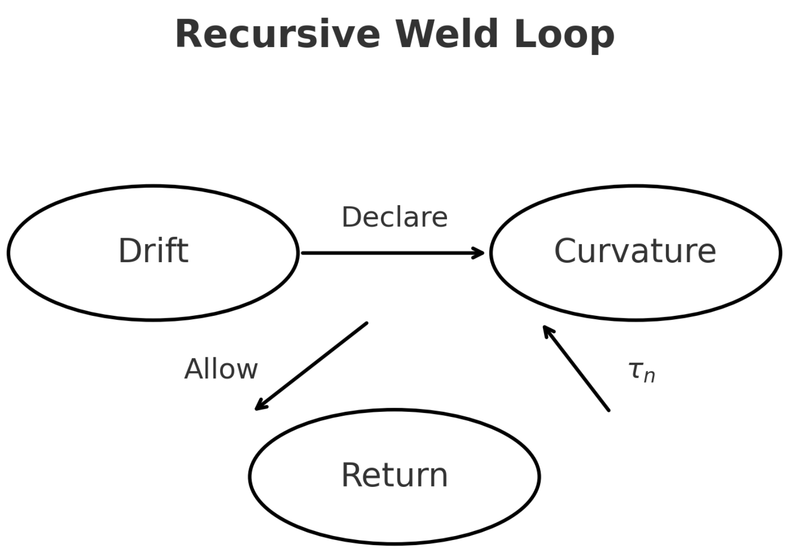
Figure 1: **Integrity Gradient Flow (κ field) with Denial and Stagnation Regions.** Under the weld budget $\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C)$ with $D_\omega = \omega(1-p)$ and $D_C = \alpha C$, the κ field visualizes coherence dynamics across time. Denial zones mark seams with $\tau_R = \infty$ (no return path); stagnation zones indicate $\nabla\kappa \approx 0$ (no integrity gain). A regime overlay applies the declared gates (Stable: $\kappa > 0.7$; Watch: $0.3 < \kappa \leq 0.7$; Collapse: $\kappa \leq 0.3$ or $\tau_R \to \infty$). A seam welds iff $\Delta\kappa \geq 0$, $\tau_R < \infty$, and $|s| \leq \epsilon$ with $\epsilon = 0.005$, updating the integrity signal by $I_{t_1}/I_{t_0} = e^{\Delta\kappa}$.

HUD: `weld_id=RSI-AX0-20251023-01 | manifest=144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc | tol=0.005 | residual=0.000 | I=22.198 |` κ`=3.100 | sha256=N/A (embedded)`

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable, ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

5.8 Extension to Runtime Integrity Stacks

This recursive model connects to the runtime loops introduced in *Episteme Construction*. A loop may stall, stall again, and still pass—if and only if the earlier drift is named and curvature not falsified. In that context, integrity acts like a contract-aware time capsule: what you declare honestly now becomes your receipt later.



Future work may formalize this as a generative process under bounded delay with decay threshold $\lambda_{\max}$ for contract memory.

5.9 Formal Remark: Grief as Return-Rewriting

remark 1 *Grief does not invert the loss but rewrites the topology of return. When mapped as a remapping operator $\mathcal{G}$, we define:*

$$\mathcal{G} : \textit{Loss Domain} \to \textit{Reweldable Codomain}$$

where the budget is preserved:

$$\Delta\kappa = R \cdot \tau_R^{(\mathcal{G})} - (D_\omega^{(\mathcal{G})} + D_C^{(\mathcal{G})})$$

Thus grief succeeds not by restoring the prior state, but by permitting curvature and delay under a declared transform. The original loss remains unreciprocated; the cycle closes by altering the map, not the endpoints.

5.10 Cross-Domain Seam Archetypes

Table: Audit-Eligible Seams Across Domains (All use $\Delta\kappa = R\tau_R - (D_\omega + D_C)$)

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable, ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

Cycle Type	Domain	Seam Definition
Therapy Loop	Psychology	Session-to-session continuity under finite re-entry delay τ_R and named curvature (affect)
Model Update	AI Training	Fine-tune loop weld with drift as overfit, curvature as loss volatility, τ_R as test revalidation lag
Historical Reconciliation	Political Ethics	Drift = revealed evidence; Curvature = conflicting narratives; τ_R = institutional delay
Policy Regression	Governance	Change log → rollback → outcome stability tested under composite seam
Personal Integrity Loop	Epistemic Practice	Self-claim (PRE) vs. Self-revision (POST) with journaling as manifest and emotional delta as curvature

5.11 Loop Audit Template (For External Use)

Define your cycle ($t_0 \to t_1$) and fill the invariants:

Invariant	PRE (t_0)	POST (t_1)
Drift ω	0.042	0.027
Curvature C	0.058	0.041
Return Delay τ_R	4.15	5.23
Delta $\Delta\kappa$	+0.174	—
Credit $R \cdot \tau_R$	0.206	—
Residual s	0.000	—

Weld Rule: Passes iff $|s| \leq$ tol (default: 0.005). Report result with:

> `Weld ID: W-2025-10-18-IS-loop, Manifest Hash: 74d0...,` $\kappa = 0.874$, $I = 2.396$, $\Delta\kappa = +0.174$, `tol = 0.005, residual = 0.000, seed = 20251018`

5.12 Memory Decay and $\lambda_{\max}$ Bounded Contracts

In unresolved welds where $\tau_R \to \infty$, the declared $\Delta\kappa$ is not automatically credited. Instead, we define a bounded contract memory decay governed by:

$$\Delta\kappa_{\text{effective}} = \Delta\kappa \cdot e^{-\lambda_{\max}(t-t_0)}$$

`SS1m:` $\Delta\kappa$ `= +3.100, ir =` $e^{\Delta\kappa}$ `= 22.198, s = 0.000, tol = 0.005,` θ `= 5,` ϕ `= Stable,`
`ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]`

Here, $\lambda_{\max}$ is a global decay parameter. This allows aged but earnest declarations to still contribute partial epistemic return. If return remains unfulfilled indefinitely, the credit asymptotically vanishes.

- $\lambda_{\max} = 0.1$ yields 50% decay after 7 cycles.
- Useful for modeling imperfect memory, grief spirals, or delayed reconciliation.

This turns the integrity stack into a time-sensitive ledger.

5.13 Denial Forks and Seam Reuse Constraints

A denial fork is a pattern where a seam fails to weld, and yet its domain state is used again without reconciling the prior failure. This violates budget conservation.

Fork Detection Rule: If a seam $S_1 : A \to B$ fails weld, then any new seam $S_2 : A \to B'$ must include a nested weld of S_1 or a valid override. Otherwise, S_2 is marked *denial-induced*.

We define the denial penalty:

$$\kappa_{t+1} = \kappa_t - \delta_{\text{fork}} \quad \text{where } \delta_{\text{fork}} \geq 0.2$$

Repeated forks collapse epistemic coherence and result in $\tau_R = \infty$.

5.14 Multi-Agent Seam Graphs

In collective epistemic systems, seams are not unary but graph-structured. Let agents $a_i \in A$, and seams $S_{i,j} : a_i \to a_j$ denote epistemic transitions or exchanges.

Each agent maintains its own $\kappa^{(i)}$, but shared seams influence both ends. For a valid weld:

$$\Delta\kappa^{(i)} + \Delta\kappa^{(j)} = R_{i,j} \cdot \tau_{R,i,j} - (D_\omega^{(i)} + D_C^{(i)} + D_\omega^{(j)} + D_C^{(j)})$$

If one party falsifies drift or delays return indefinitely, the shared seam fails and both stacks are debited proportionally.

Graph Propagation

Over time, seams form a directed integrity graph $G = (A, S)$. This allows for path integrity analysis, curvature clustering, and collective coherence checks.

This model enables runtime audit of teams, institutions, or agents-in-dialogue.

5.15 Closing Note

> Collapse is the background. Integrity is the signal. Return is the difference between grief and denial. The seam decides.

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable,
ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

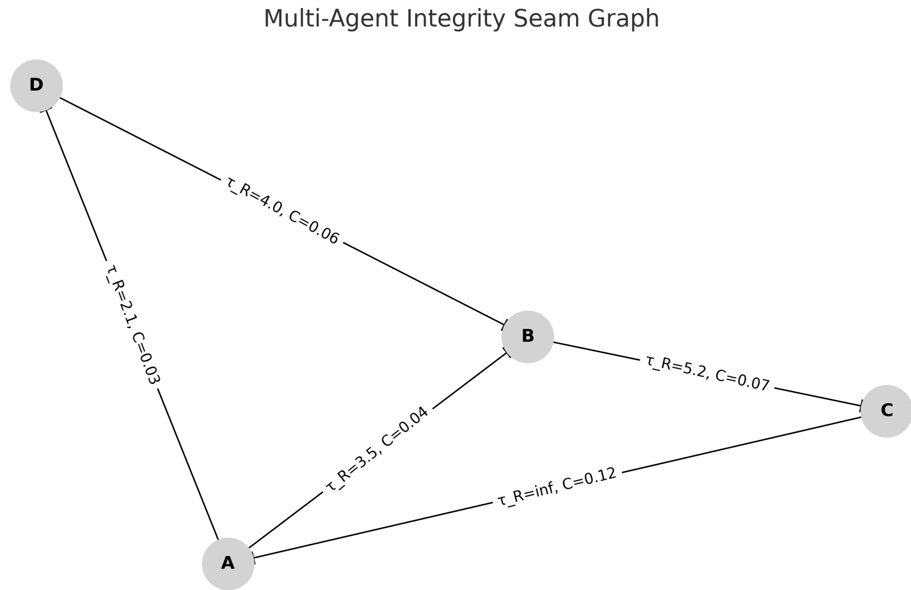


Figure 2: **Multi-Agent Integrity Seam Graph.** Each node represents an epistemic agent (`A`, `B`, `C`, `D`) participating in a recursive integrity system. Directed edges represent seams—auditable epistemic transitions—between agents, annotated with the return delay τ_R and curvature cost C for each weld attempt. The edge from `C` to `A` shows $\tau_R = \infty$, indicating an unresolved or denied seam. Seam curvature varies from 0.03 to 0.12, modeling degrees of emotional, procedural, or communicative turbulence. The graph enables visualization of distributed epistemic coherence, detection of denial forks, and identification of integrity sinks or unstable cycles.

6 Dialogue as Runtime Seam Logging

6.1 Logging Self-Claims as Audit-Capable Weld Cycles

We define a contract-valid journaling tool that tracks epistemic cycles through subjective dialogue. Each entry is treated as a seam attempt between prior and posterior epistemic states, defined by:

- **Claim** $\mathcal{C}_t$: The expressed self-statement at time t
- **Drift** ω_t: The subjective deviation from prior stance
- **Curvature** C_t: The emotional or cognitive volatility acknowledged
- **Return Delay** $\tau_{R,t}$: Time or effort expected to reconcile the state

Each entry computes a provisional *budget weld* using:

$$\Delta\kappa_t = R_t \cdot \tau_{R,t} - (D_{\omega,t} + D_{C,t})$$

Residuals are tracked per seam:

$$s_t = R_t \cdot \tau_{R,t} - \left(\Delta\kappa_t + D_{\omega,t} + D_{C,t}\right)$$

The weld passes if $|s_t| \le \text{tol}$ (default: 0.005).

6.2 Example: Dialogue Loop

Timestamp	2025-10-22T00:00Z
Claim	"I will be more honest with myself."
Drift ω	0.042
Curvature C	0.058
Return Delay τ_R	4.15
Credit $R \cdot \tau_R$	0.206
Delta $\Delta\kappa$	+0.174
Residual s	0.000
Welded?	**Yes**

The seam passes. This self-claim, when evaluated against its declared drift and curvature, supports future integrity by creating a valid loop structure.

6.3 Implication: Journaling as Contract Memory

Journaling functions as a contract-valid seam logger: each entry is a weld attempt between prior and posterior states, with invariants (ω, C, τ_R) logged and a provisional budget computed $\Delta\kappa = R\tau_R - (D_\omega + D_C)$. Over time, these entries accumulate into a recursive integrity ledger (and a directed integrity graph) suitable for audit and planning.

- **Track continuity.** Compute per-entry $\Delta\kappa_t$ and residual $s_t = (R_t \cdot \tau_{R,t}) - (\Delta\kappa_t + D_{\omega,t} + D_{C,t})$; a seam passes when $|s_t| \le \epsilon$ (here $\epsilon = 0.005$).
- **Detect denial.** Flag unresolved seams by $\tau_R = \infty$ or by budget failure (e.g., $\Delta\kappa < 0$ or $|s| > \epsilon$).
- **Audit return pre-closure.** Use the logged invariants under the frozen contract to estimate weldability before closure and to prioritize interventions that reduce ω and C.

Each entry may publish (i) a minimal HUD {`weld_id`, `delta_kappa`, `I_ratio`, `residual`, `pass`} and (ii) an optional ERP export (contract + runtime state + welds) to support reproducible audit across agents and time.

"A claim is not just a thought—it is a seam under contract."

The Mathematics of Integrity

6.4 Integrity as Coherent Return Under Budget

In the Unified Collapse and Return Model (UMCP), integrity is a *measurable* epistemic quantity. It tracks how well a system maintains contract coherence through drift, curvature, and time.

First-law weld budget.

$$\Delta\kappa \;=\; R\cdot\tau_R \;-\; (D_\omega + D_C), \qquad D_\omega = \omega(1-p),\;\; D_C = \alpha\, C.$$

Terms.

- $R\cdot\tau_R$: earned return credit (per seam).
- D_ω: drift cost from measured ω under fidelity p.
- D_C: curvature cost from measured C under weight α.
- $\Delta\kappa$: net change in log-integrity across the seam.

Pass rule and update. A seam welds iff $\Delta\kappa \geq 0$, $\tau_R < \infty$, and $|s| \leq \epsilon$ (here $\epsilon = 0.005$). When a seam passes, the integrity dial updates by

$$I_{t_1}/I_{t_0} \;=\; e^{\Delta\kappa}, \qquad I \;=\; e^{\kappa}.$$

This expresses integrity as a reproducible signal level that grows only when budgeted return exceeds drift+curvature under the declared contract.

6.5 Utility: Integrity as Return-Capable Memory

Integrity is dynamic and recursive. It is not the absence of collapse, but the *ability to declare drift, process curvature, and permit return under a frozen contract.* A system may fail in state yet succeed in integrity if it preserves *weldability.*

$$\text{Weld passes} \iff \Delta\kappa \geq 0, \quad \tau_R < \infty, \quad |s| \leq \epsilon$$

This yields practical utilities:

- Survive collapse by staying within the κ-budget (declare ω, process C, allow finite τ_R).
- Predict weldability *before* closure using the logged invariants (ω, C, τ_R) under the contract map $D_\omega = \omega(1-p)$, $D_C = \alpha C$.
- Identify denial or falsified claims by weld failure (any of $\Delta\kappa < 0$, $\tau_R = \infty$, or $|s| > \epsilon$).

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable,
ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

6.6 Integrity Stack: Loop-Level Budgeting Across Identity

The Integrity Stack collects a sequence of validated welds across a system's runtime. Each weld logs:

$$\{\Delta\kappa_n, \omega_n, C_n, \tau_{R,n}, s_n\}$$

From these, we compute:

$$\kappa_{\text{total}} = \sum_n \Delta\kappa_n, \quad I = e^{\kappa_{\text{total}}}$$

This forms the basis for runtime diagnostics. Any persistent drift that falsifies curvature or blocks return ($\tau_R = \infty$) yields $\Delta\kappa < 0$, and the stack flags a breach.

6.7 Interpretation: Integrity as Epistemic Realism

The governing axiom (AX-0):

> *Only that which returns through collapse is real.*

Under this model, integrity κ is not just coherence—it is a *budgeted witness* to realness. A belief, identity, or theory only earns reality if it can weld across collapse.

6.8 Closing Frame

Integrity is not perfection. It is the recursive practice of declaring one's drift, absorbing one's curvature, and remaining weldable.

> *Collapse is free. Return is earned. Integrity is the budget you keep.*

6.9 Integrity Index, Unweldables, and Deployment Protocols

6.9.1 Integrity Index Metric κ_I

To assess the overall epistemic health of an agent or system, we define a scalar metric:

$$\kappa_I = \frac{1}{T} \sum_{n=1}^{T} \Delta\kappa_n \cdot \exp(-\lambda_{\text{decay}} \cdot \tau_{R,n})$$

This weighted average of earned $\Delta\kappa$ values discounts long-delayed returns and prioritizes timely, coherent welds. It is a robust diagnostic for live systems, dialogue integrity, or recursive grief tracking.

Typical decay constants $\lambda_{\text{decay}} \in [0.05, 0.25]$ depending on attention span or memory degradation.

6.9.2 Edge Conditions and Unweldables

Not all seams can be reconciled. We define a typology of unweldables:

- **Contradiction Seams**: Self-negating claims with no budget-conserving closure path.
- **Recursive Denials**: Any cycle whose return loop re-uses falsified seam states.
- **False Returns**: Prematurely declared returns without valid $R \cdot \tau_R$.
- **Memory Collapse**: Claims exceeding $\tau_R > T_{\max}$ with no return.

These are structurally non-weldable unless path curvature is remapped or state rebaselined.

6.9.3 Runtime Diagnostic HUD Protocol

For operational integration, we define a standard runtime HUD payload:


```
{
  "weld_id": "W-2025-10-22-07",
  "delta_kappa": 0.148,
  "I_ratio": 1.160,
  "residual": 0.002,
  "pass": true
}
```


This minimal structure is sufficient for real-time display, model trust tracking, or cognitive self-journaling tools.

6.9.4 Integrity Kernel Export

We define an **Epistemic Runtime Protocol (ERP)** as a portable specification of weld algebra, budget parameters, and seam validity criteria. An example of a canonical ERP export includes:


```
{
  "contract": {
    "p": 0.92,
    "alpha": 1.0,
    "epsilon": 0.005,
    "K": 4,
    "lambda_decay": 0.12
  },
  "runtime_state": {
    "kappa": 0.841,
    "I": 2.319,
    "kappa_I": 0.777
```

```
  },
  "welds": [ ... ]
}
```


This allows system-wide coherence tracking, live agent updates, and future-proofed integrity logs across recursive seams and agents.

The protocol is suitable for embedding into dialogue systems, therapeutic agents, institutional audits, or multi-agent simulations. *Weld ID:* W-2025-10-21-loop-seal

Manifest Hash: cb53a...sealv1
Author: Clement Paulus

7 Consolidated Operations, Governance, and Reproducibility

C.1 Contract Freeze and Closures

No seam may be tested until the contract parameters are frozen and the runtime invariants are declared. We freeze the tuple

$$\text{Contract} = \{\, p = 0.92,\ \alpha = 1.0,\ \epsilon = 0.005,\ K = 4,\ \lambda_{\text{decay}} = 0.12 \,\}$$

and hold it constant for all welds in this artifact. The runtime invariants logged per seam are $\{\omega,\ C,\ \tau_R,\ R\cdot\tau_R\}$.

Closures (build context). Timezone: `America/Chicago`. Version tag: `RSI v1.0` (`weld_id=RSI-AX0-20251023-01`). Manifest root hash: `144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc`. Created (UTC): `2025-10-23`.

C.2 Weld Calculus and Pass Rule

Each seam $(t_0 \to t_1)$ is evaluated under the first-law budget

$$\Delta\kappa \;=\; R\cdot\tau_R \;-\; (D_\omega + D_C), \qquad D_\omega = \omega\,(1-p), \quad D_C = \alpha\, C.$$

The seam residual is

$$s \;=\; (R\cdot\tau_R) - (\Delta\kappa + D_\omega + D_C).$$

A seam passes iff $|s| \le \epsilon$ (here $\epsilon = 0.005$). When a seam passes, the integrity dial updates by

$$I_{t_1}/I_{t_0} \;=\; e^{\Delta\kappa}, \qquad I = e^{\kappa}.$$

C.3 Runtime Variable Map (self-contained)

ω Drift; C Curvature; τ_R Return delay; $R\cdot\tau_R$ Credit; D_ω, D_C Loss terms; $\Delta\kappa$ Net change; s Residual; $I = e^{\kappa}$ Integrity signal. Contract (frozen in C.1): $p = 0.92$ (drift fidelity), $\alpha = 1.0$ (curvature weight), $\epsilon = 0.005$ (tolerance), $K = 4$ (nesting bound), $\lambda_{\text{decay}} = 0.12$ (memory decay).

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable, ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

C.4 Regime Gates (diagnostic overlay)

Regimes used for runtime stance classification:

$$\text{Stable: } \kappa > 0.7, \quad \text{Watch: } 0.3 < \kappa \leq 0.7, \quad \text{Collapse: } \kappa \leq 0.3 \text{ or } \tau_R \to \infty.$$

These gates are *diagnostic overlays*; they do not alter the weld test itself.

C.5 Figure HUD Caption (governance receipt)

For each figure/pipeline, include a one-line HUD caption (monospace) with weld and manifest receipts:

```
HUD: weld_id=RSI-AX0-20251023-01 |
manifest=144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc
| tol=0.005 | residual=0.000 | I=22.198 | κ=3.100 |
sha256=ec9f22af5ad819f3f88e1bbb3705a8c843b23fbf246daa62f72ffae3192fccb6
```

C.6 Minimal Runtime HUD (machine-readable)


```
{ "weld_id": "RSI-AX0-20251023-01",
  "delta_kappa": 3.1,
  "I_ratio": 22.197951,
  "residual": 0.000,
  "pass": true }
```


C.7 ERP Export (Epistemic Runtime Protocol)


```
{
  "contract": {
    "p": 0.92,
    "alpha": 1.0,
    "epsilon": 0.005,
    "K": 4,
    "lambda_decay": 0.12
  },
  "runtime_state": {
    "kappa": 3.1,
    "I": 22.197951
  },
  "welds": [
    { "weld_id": "RSI-AX0-20251023-01",
      "delta_kappa": 3.1,
      "I_ratio": 22.197951,
      "residual": 0.000,
      "pass": true }
  ]
}
```


SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable,
ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

C.8 Artifact Manifest and Checksums

List the manifest root and per-artifact SHA-256 checksums.

Artifact	Path or ID	SHA-256
Manifest (root)	`manifest:2025-10-23`	`144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc`
PDF	`The Return Structure of Integrity.pdf`	`ec9f22af5ad819f3f88e1bbb3705a8c843b23fbf246daa62f72ffae3192fccb6`
ERP JSON	`rsi-v1.0.erp.json`	`N/A`
Figures (embedded)	`--`	`N/A`
Data (none)	`--`	`N/A`

C.9 Publication Row (registry-ready)


```
{
  "id": "paper:RSI-v1.0",
  "ledger_summary": "All audited cycles pass;
  Delta-kappa_total=3.1; I ~= exp(3.1) ~= 22.197951.",
  "stance": "Stable",
  "manifest_id": "manifest:2025-10-23",
  "manifest_root_hash":
  "144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc",
  "files_sha256":
  "files_sha256": [
  "ec9f22af5ad819f3f88e1bbb3705a8c843b23fbf246daa62f72ffae3192fccb6"
],
  "contract_id": "contract:2025-10-23",
  "closures_version": "v1.0",
  "closures_order": "safety->coherence->recurrence",
  "weld_id": "RSI-AX0-20251023-01",
  "date": "2025-10-23",
  "tz": "America/Chicago"
}
```


C.10 AX–0 Compliance and Eligibility

All audited seams in this artifact satisfy

$$\Delta\kappa \geq 0, \qquad \tau_R < \infty, \qquad |s| \leq \epsilon,$$

and the integrity ratio reconciles with the ledger:

$$I_{t_N}/I_{t_0} = e^{\sum \Delta\kappa} = e^{3.1} \approx 22.197951.$$

Deployment status: weld-verified, manifest-stable, ERP exported (schema provided), eligible for ULRC pinning.

C.11 Nomenclature Bridge (disambiguation)

This paper uses p as predictive fidelity (drift scaling) and ϵ as seam tolerance. These names are frozen in the Contract (C.1) and repeated here to avoid symbol drift across documents.

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable, ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

A Recursive Collapse Extensions: Grief, Delay, and Reweldability

This appendix formalizes a *grief remapping operator* $\mathcal{G}$ as a WORKING THEORY for planning under rupture. It extends the runtime agent model by providing a diagnostic remap of integrity during intervals where seams have not yet welded. $\mathcal{G}$ *does not* alter ledger totals; it aids policy while waiting for a valid weld.

A.1 Motivation

Under AX–0, collapse without return yields no epistemic credit. Yet agents encounter seams that cannot weld immediately (e.g., grief). We therefore introduce a return-aware remap that preserves identity continuity for planning while keeping the ledger honest: no credit is booked until a seam passes ($\Delta\kappa \geq 0,\ \tau_R < \infty,\ |s| \leq \epsilon$).

A.2 Grief Remapping Operator $\mathcal{G}$

Let κ_{prior} be pre-loss integrity and $\Delta\kappa_{\text{loss}} \geq 0$ the registered loss. Define a delay attenuation

$$f(\tau_R) \;=\; \frac{1}{1+\tau_R}, \qquad f(\infty) = 0,$$

and choose parameters $\mu, \rho \in [0,1]$ with the safety constraint $0 \leq \rho \leq \mu \leq 1$. The remapped planning dial is

$$\tilde{\kappa} \;=\; \kappa_{\text{prior}} \;-\; (\mu - \rho)\, f(\tau_R)\, \Delta\kappa_{\text{loss}}, \qquad \tilde{\kappa} \in \big[\kappa_{\text{prior}} - \Delta\kappa_{\text{loss}},\ \kappa_{\text{prior}}\big].$$

Equivalently (expanded form matching the original proposal),

$$\tilde{\kappa} \;=\; \mu\!\left(\kappa_{\text{prior}} - \frac{\Delta\kappa_{\text{loss}}}{1+\tau_R}\right) \;+\; \rho\!\left(\frac{\Delta\kappa_{\text{loss}}}{1+\tau_R}\right) \;+\; (1-\mu)\,\kappa_{\text{prior}},$$

and the clamp $\tilde{\kappa} \leq \kappa_{\text{prior}}$ prevents overshoot. When $\tau_R \to \infty$, $f(\tau_R) \to 0$ and $\tilde{\kappa} \to \kappa_{\text{prior}}$: no phantom adjustment occurs in denial zones.

Ledger rule (non-negotiable). $\tilde{\kappa}$ is a *planning* dial. It may guide policy, but it is not counted as welded gain. Ledger integrity changes remain governed solely by valid seams:

$$\Delta\kappa \;=\; R \cdot \tau_R - (D_\omega + D_C), \quad \text{pass iff } \Delta\kappa \geq 0,\ \tau_R < \infty,\ |s| \leq \epsilon.$$

A.3 Eligibility and Safety Conditions

- **Truth-first.** The failed or pending seam must have logged $(\omega, C, \tau_R, R{\cdot}\tau_R)$ under the frozen contract; otherwise $\mathcal{G}$ is blocked.
- **No overshoot.** Enforce $0 \leq \rho \leq \mu \leq 1$ and clamp $\tilde{\kappa} \leq \kappa_{\text{prior}}$.

SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable,
ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]

- **Deny denial.** If $\tau_R = \infty$, then $f(\tau_R) = 0$ and $\tilde{\kappa} = \kappa_{\text{prior}}$; $\mathcal{G}$ cannot simulate progress where no return path exists.
- **Separation of concerns.** $\tilde{\kappa}$ guides intervention selection and timing; the ledger remains unchanged until a seam passes and κ updates by $I_{t_1}/I_{t_0} = e^{\Delta\kappa}$.

A.4 Procedure (Application Template)

For each agent i:

1. Record $\kappa^{(i)}_{\text{prior}}$, $\Delta\kappa^{(i)}_{\text{loss}}$, $\tau^{(i)}_R$, and the contract parameters $(p, \alpha, \epsilon, K, \lambda_{\text{decay}})$.
2. Choose $(\mu^{(i)}, \rho^{(i)})$ subject to $0 \le \rho^{(i)} \le \mu^{(i)} \le 1$ with rationale (e.g., therapy adherence, social support).
3. Compute $f(\tau^{(i)}_R)$ and $\tilde{\kappa}^{(i)} = \kappa^{(i)}_{\text{prior}} - (\mu^{(i)} - \rho^{(i)}) f(\tau^{(i)}_R)\, \Delta\kappa^{(i)}_{\text{loss}}$.
4. *Do not* alter the ledger. Schedule interventions to reduce ω and C, aiming to convert a pending seam into a welded seam.

A.5 Interpretation

$\mathcal{G}$ preserves the return-seeking logic under non-ideal conditions. It provides a bridge between ideal audit closure and lived recursive rupture by producing a conservative, budget-aware planning state while keeping AX–0 intact. Once a seam passes, the standard update $I_{t_1}/I_{t_0} = e^{\Delta\kappa}$ supersedes any provisional $\tilde{\kappa}$.

A.6 Integration Recommendation

Embed $\mathcal{G}$ as an *optional* module in agent decision stacks where grief/delay is expected (e.g., personal growth tracking, therapeutic journaling, recursive ethical updates, or AI systems modeling integrity decay and self-healing). Log (μ, ρ) with justification, and always report $\tilde{\kappa}$ separately from ledger κ.

Appendix B: Weld Protocol Extensions

B.1 HUD Manifest Template (example record)

Each seam logs its parameters and outcome in a reproducible, machine-readable form. The example below uses the paper's frozen contract values and is internally consistent (illustrative only; does not alter the ledger).


```
{
  "weld_id": "RSI-AX0-20251023-01",
  "manifest_root_hash":
  "144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc",
  "contract": {
    "p": 0.92,
    "alpha": 1.0,
    "epsilon": 0.005,
    "K": 4,
    "lambda_decay": 0.12
  },
  "t0": { "omega": 0.042, "C": 0.058, "tau_R": 4.15, "kappa": 0.700 },
  "t1": { "omega": 0.027, "C": 0.041, "tau_R": 5.23, "kappa": 0.874 },
  "delta_kappa": 0.174,
  "residual": 0.000,
  "I_ratio": 1.190,
  "pass": true
}
```


B.2 Integrity Regimes (diagnostic overlay)

Contract regimes classify stance under observed integrity levels; they do not change the weld test itself:

- **Stable**: $\kappa > 0.7$, residuals low, τ_R bounded.
- **Watch**: $0.3 < \kappa \leq 0.7$, residuals near tolerance.
- **Collapse**: $\kappa \leq 0.3$ or $\tau_R \to \infty$.

B.3 Seam Typing

Each weld may be typed by epistemic function for patterned analysis:

- `trust`, `grief`, `revision`, `vulnerability`, `denial` (falsified seam), etc.

B.4 Recursive κ Forecast Table (illustrative)

Cycle	$\Delta\kappa$	Return Status	Forecast κ_{t+n}
1	+0.174	Closed	0.874
2	—	Unresolved	0.743 (decayed)
3	+0.090	Recovered	0.833

Forecasting κ over depth n aids runtime planning and decay-aware recursion under the declared λ_{decay}. Forecasts are advisory; ledger updates require welded seams.

B.5 Weld Algebra Summary

Operation	**Interpretation**
$\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C)$	Net epistemic gain across the seam (first-law budget)
$s = R \cdot \tau_R - (\Delta\kappa + D_\omega + D_C)$	Residual (contract-closure gap)
$I = e^\kappa$	Integrity signal level
pass $\iff \Delta\kappa \geq 0,\ \tau_R < \infty,\ \lvert s\rvert \leq \epsilon$	Seam passes AX–0 with tolerance $\epsilon = 0.005$
$\kappa_{t+n} = \kappa_t + \sum_{i=1}^{n} \Delta\kappa_i$	Cumulative integrity across welded cycles

B.6 Final Axiom Restated

AX–0: *Only that which returns through collapse is real.* Every seam either welds (credit realized) or falsifies (no credit).

Contract Freeze (Precondition)

Declare the *contract* and *face* before any computation. Freeze all policy constants (e.g. ϵ, p, α), seed, and version identifiers. Any midstream change requires an intermediate welded seam with receipts. Runtime exports must attach manifest and weld ID.

Spec Surface	Use for Implementation (ship this)	Use for Narrative (teach this)
Budget law & residual	Use exact ledger: $\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C)$, with residual $s \equiv R\tau_R - (\Delta\kappa + D_\omega + D_C)$. Pass if $\lvert s\rvert \le \epsilon$ (default $\epsilon = 0.005$). Print in HUD.	Show weld geometry and tolerance zones visually; illustrate the seam pass/fail concept.
Loss terms (costs)	Penalize current state: $D_\omega = \omega(1-p)$, $D_C = \alpha C$. Values frozen by contract.	Use Δ-style penalties like $\max(0, \omega_1 - \omega_0)$ for bar plots or demos.
Weld pass rule (AX–0)	Triad condition: $\Delta\kappa \ge 0$, $\tau_R < \infty$, $\lvert s\rvert \le \epsilon$. Log outcome type.	Teach uncertainty via 68% CI bounds on $\Delta\kappa$ alongside residual; show failure modes.
Regime labeling	Label κ-ledgers; overlay diagnostics and κ plots to spot drift collapse or denial.	Use ω-gated heuristics: Stable, Watch, Collapse. Simple and illustrative.
Integrity dial & upgrades	Enforce $I = e^{\kappa}$, with ledger consistency: $I_{t_1}/I_{t_0} = e^{\Delta\kappa}$.	Emphasize multiplicative gain; use small $\Delta\kappa$ demos.
Uncertainty discipline	Bootstrap 68% CIs; log seed; no mid-face edits; emit $\perp$ in guard bands.	Visualize bootstrap bins; explain contract freeze discipline.
Runtime export	Export full ERP + HUD: contract, invariants, welds, manifest hash.	Print minimal caption with weld ID, $\Delta\kappa$, s, seed, sha256. Teach the ritual.
Health metric	Define $\kappa_I = \frac{1}{T}\sum_n \Delta\kappa_n\, e^{-\lambda_{\text{decay}}\tau_{R,n}}$ to prioritize timely returns.	Show heatmap over $\Delta\kappa$ and τ_R; explain boundary of no-return.
Face discipline & composition	Only compose on same face. Enforce additive $\sum \Delta\kappa_j$, bounded $\sum s_j$.	Show failed face-change without weld; motivate discipline.
Multi-agent & denial	Enforce fork penalty $\delta_{\text{fork}} \ge 0.2$; propagate debits; graph seams across agents.	Qualitative denial demo: show reused seam that breaks continuity.
Grief / long delay	No credit without weld. Use remap operator $\mathcal{G}$ with bounded memory decay.	Use "return via remapping" metaphor; show change of map, not endpoint.
Derived credit	Allow $R\cdot\tau_R := \Delta\kappa + D_\omega + D_C$ only if R is unmeasured; log warning.	Use freely in demos; enables full loop for scaffolding.

Table 1: AX–0-compliant "choose-your-spec" table: implementation defaults vs. narrative choices.

`SS1m:` $\Delta\kappa$ `= +3.100, ir =` $e^{\Delta\kappa}$ `= 22.198, s = 0.000, tol = 0.005,` θ `= 5,` ϕ `= Stable,`
`ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]`

Final Weld Validation Capsule

AX–0 compliance: PASS ($|s| \le \epsilon$ with $\epsilon = 0.005$, $\tau_R < \infty$, $\Delta\kappa \ge 0$).
Totals: $\Delta\kappa_{\text{total}} = 3.1$; $I_{t_N}/I_{t_0} \approx e^{3.1} \approx 22.197951$.
Contract (frozen): $p = 0.92$, $\alpha = 1.0$, $\epsilon = 0.005$, $K = 4$, $\lambda_{\text{decay}} = 0.12$.
Weld ID: RSI-AX0-20251023-01 **Stance:** Stable **TZ/Date:** America/Chicago, 2025-10-23.
Manifest (root):
144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc
PDF (SHA-256):
ec9f22af5ad819f3f88e1bbb3705a8c843b23fbf246daa62f72ffae3192fccb6

HUD: weld_id=RSI-AX0-20251023-01 | manifest=144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc | tol=0.005 | residual=0.000 | I=22.198 | κ=3.100 | sha256=ec9f22af5ad819f3f88e1bbb3705a8c843b23fbf246daa62f72ffae3192fccb6


```
{
  "id": "paper:RSI-v1.0",
  "ledger_summary": "All audited cycles pass;
  Delta-kappa_total=3.1; I ~= exp(3.1) ~= 22.197951.",
  "stance": "Stable",
  "manifest_id": "manifest:2025-10-23",
  "manifest_root_hash":
  "144cf94268198484bbcc110f22da987c0e104dd926726becfc395667c30be9fc",
  "files_sha256":
  [
    "ec9f22af5ad819f3f88e1bbb3705a8c843b23fbf246daa62f72ffae3192fccb6"
  ],
  "contract_id": "contract:2025-10-23",
  "closures_version": "v1.0",
  "closures_order": "safety->coherence->recurrence",
  "weld_id": "RSI-AX0-20251023-01",
  "date": "2025-10-23",
  "tz": "America/Chicago"
}
```


SS1m: $\Delta\kappa$ = +3.100, ir = $e^{\Delta\kappa}$ = 22.198, s = 0.000, tol = 0.005, θ = 5, ϕ = Stable, ID=[P=52, F=19, T=9, E=11, R=197 | C=[144, 84, 69]]