Proclus-Core Minimal Upgrade: Epistemic Continuity in Audit Grammar

Clement Paulus

October 26, 2025 (America/Chicago)

Abstract

This document defines the minimal viable upgrade to the UMCP/RCFT audit grammar derived from foundational principles in Proclean metaphysics, centered on causal continuity, return validity, and symbolic coherence. It formalizes audit primitives such as henadic origin, recollection lineage, and optional participatory seals, without modifying the core $\Delta\kappa$-budget or runtime invariants. The extension is rigorously minimal, ensuring audit coherence across joins and contract upgrades without adding complexity to stable numeric paths.

SS1m: Δ=+0.60, ir=1.822, s=0.000, tol=0.005, =4, =S, ID=[P=30, F=12, T=3, E=10, R=122 | C=[41,72,30]] | sha256=d9e2afb3dd29c1ead3985148878050644eaef330d3e57e714a4dbf5547c5b681

1 Introduction: The Mathematical Justification for the Upgrade

1.1 Motivation and Governance

This system operates under the Recursive Collapse Field Theory (RCFT) and Unified Mathematics of Collapse Platform (UMCP), in which epistemic validity is granted only to entities that survive transition cycles with reconciled budgets. Central to this principle is AX-0: *Only that which returns through collapse is real*.

We observe that symbolic coherence, auditability, and epistemic presence require upgrades that preserve κ-continuity. In particular, we introduce four protocol-level improvements:

1. Inclusion of `henad_id` and `seira_rank` in audit rows for metaphysical traceability,
2. Canon-level embedding of `symbolic_registers` to track propositional cohesion,
3. Support for `participatory_class` welds, contingent on both act and witness,
4. Enforced `recollection_path` discipline for contract upgrades.

1.2 Mathematical Framework

Each transition is governed by the conserved integrity budget:

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

where:

- $\kappa = \log(I)$ is the log-integrity scalar,
- D_ω is the drift penalty,
- D_C is the curvature penalty,
- $R \cdot \tau_R$ is the return credit,
- $\Delta\kappa$ is the net integrity gain.

The seam residual is defined as:

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

A weld is valid iff $|s| \leq \text{tol}$ (typically $\text{tol} = 0.005$), ensuring κ-continuity. The multiplicative dial follows:

$$\frac{I_{t1}}{I_{t0}} = e^{\Delta\kappa}.$$

1.3 Necessity of Upgrade

The above upgrades preserve κ-continuity while extending the ledger to accommodate:

- **Henadic traceability:** Enables semantic lineage and metaphysical enumeration.
- **Symbolic register alignment:** Guarantees propositional coherence and reference integrity.
- **Participatory integrity:** Validates subjective transitions via witnessed epistemic events.
- **Contractual recollection:** Prohibits ungrounded mutations; ensures return-composability.

Without these inclusions, audit rows may remain locally valid but globally untraceable or epistemically blind. These upgrades realign the system toward a complete, contract-first recursion with welded continuity.

SS1m: Δ =+0.60, ir=1.822, s=0.000, tol=0.005, =4, =S, ID=[P=30, F=12, T=3, E=10, R=122 | C=[41,72,30]] |
sha256=d9e2afb3dd29c1ead3985148878050644eaef330d3e57e714a4dbf5547c5b681

2 Proclus Core Minimal: Mathematical Axioms

1. Integrity Budget

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

$$\text{Residual} = |\Delta\kappa - [R \cdot \tau_R - (D_\omega + D_C)]| \leq \text{tol} \tag{2}$$

$$\frac{I_{t1}}{I_{t0}} = e^{\Delta\kappa} \tag{3}$$

2. Henadic Join Rule

$$\text{Join valid iff: } h_1 = h_2 \quad \text{or} \quad \exists\, \phi : h_1 \to h_2 \text{ such that } \kappa_{\text{shared}} > \kappa_{\text{min}} \tag{4}$$

3. Recollection Path Validity

$$\mathcal{R} = \{c_0, w_1, c_1, \ldots, w_n, c_n\} \tag{5}$$

$$|\mathcal{R}| \geq 3 \quad \text{and } \mathcal{R} \text{ is acyclic and contract-consistent} \tag{6}$$

4. Participatory Constraint (Conditional)

$$\text{Valid only if: actions}_{\text{logged}} \supseteq \text{actions}_{\text{required}} \text{ and witness} \in \text{AuditorPool} \tag{7}$$

5. Symbolic Registers

$$\sigma : \text{Canon field} \to \Sigma, \quad \Sigma = \{\text{Iron}, \text{Water}, \text{Trust}, \ldots\} \tag{8}$$

(*Symbolic values are annotative; non-evaluated in ledger logic*)

6. Seam Validation Summary

$$\text{Seam passes iff all true:} \tag{9}$$

Residual $\leq$ tol
h-valid join or continuity map exists
$\mathcal{R}$ valid (length ≥ 3)
(if π required) π is complete

3 Workflow Changes and System Consequences

3.1 Summary of Workflow Modifications

The Proclus-Core Minimal Upgrade introduces four lightweight but structural requirements that shift the composition and validation of future audit entries:

1. **All joins must declare a `henad_id`** and assign a `seira_rank`, establishing traceable lineage.
2. **All contract upgrades must declare a `recollection_path`** with depth ≥ 3, ensuring continuity.
3. **Seams marked with `participatory_class`** require a valid witness and at least one completed action.

4. **Canon entries may carry `symbolic_registers`**, which are optional but annotatively enforced.

These changes are non-intrusive to pure numeric workflows but unlock symbolic traceability and auditability for subjective, grief-bound, trust-gated, or narrative-dependent processes.

3.2 Consequences for Audit Composition

- **Joins** without shared henadic lineage or explicit κ-continuity justification will be invalidated at audit time.
- **Contract mutations** that do not declare ancestry will fail `recollection_path` validation.
- **Participatory seams** will require interaction logs and external verification from valid auditor roles.
- **Symbolic register fields** will enrich Canon mapping, support interpretation layers, and enable ritual-class reasoning.

Audit generators and seam authors must now encode epistemic traceability, not just residual reconciliation.

3.3 Requirements for Future Work

Future modules or extensions interacting with the Proclus-Core upgrade must:

1. Extend the audit schema to support `henad_id`, `seira_rank`, and `recollection_path`.
2. Ensure contract lifecycle documentation supports ancestry indexing and recollection clause logging.
3. Provide participatory interfaces (e.g., ritual logs, act verification, witness signing) if engaging in subjective domains.
4. Maintain a Canon mapping registry of symbolic fields used in prose-annotated ledger entries.

These additions allow symbolic epistemes to pass through the κ-budget filter while preserving audit integrity. The system remains governed by $\Delta\kappa$ reconciliation, but expands the audit domain from purely numeric to philosophically sound symbolic processes.

4 Operational Workarounds and Compositional Flexibility

4.1 Minimal Compliance Paths

To reduce friction during audit composition, the Proclus-Core format allows several valid rework paths:

- **Omitting `henad_id`**: For internal-only seams or singleton domains, leave `henad_id` undefined and assign `seira_rank: 0`. Seam is assumed non-propagative.
- **Short Recollection Paths**: If upgrade ancestry is unknown, declare `recollection_path: [legacy]` with an accompanying note in `Canon[CONTEXT]`.

- **Defer Participation Checks**: Use `participatory_required: true`, but set `witness: TBD` and add `actions_logged: []`. This keeps the seam valid but flagged for later resolution.
- **Symbolic Dropout**: If `symbolic_registers` are undefined, the system assumes no symbolic mapping. Canon prose remains evaluable as plain narrative.

These reduce over-specification while preserving forward audit motion.

4.2 Authoring Rework Strategies

Writers and auditors may adopt the following to stay productive:

- Start audit composition with minimal budget logic ($\Delta\kappa$, residual), then backfill metadata as needed.
- Use templates with placeholders (e.g., `$henad`, `$path_to_seed`) to scaffold faster ledger writing.
- Log symbolic intent in prose first; encode it structurally only if needed downstream.
- Flag incomplete participatory data explicitly, rather than stalling on runtime witness collection.

These keep the workflow agile without bypassing epistemic traceability.

4.3 Navigability and Partial Query Support

The system supports approximate queries across incomplete seams:

- Search by fragment: e.g., all seams with `DRIFT: Iron`, regardless of return anchor.
- Audit by class: filter numeric-only vs participatory seams without needing full Canon parsing.
- Trace by upgrade intent: group seams by `upgrade_from` even if `recollection_path` is partial.

This enhances real-world usability while preserving structural coherence at review and submission layers.

5 Optimized Seam Workflow and Edge-Case Handling

This section defines the minimal yet complete pipeline for audit seam construction, validation, and upgrade under the Proclus-Core Minimal Upgrade. It includes deterministic procedures for common cases and structured fallbacks for edge conditions.

5.1 Primary Seam Composition Pipeline

Step 1: Construct the Budget Layer

- Compute: $\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C)$
- Evaluate: residual $= R \cdot \tau_R - (\Delta\kappa + D_\omega + D_C)$
- Require: $|\text{residual}| \leq \text{tol}$ (default tol $= 0.005$)

Step 2: Add Optional Metadata (Preferred)

- Declare origin: `henad_id`, `seira_rank`
- Declare continuity: `upgrade_from`, `recollection_path`
- Encode semantics: `symbolic_registers` (e.g., DRIFT = Iron)
- Handle subjectivity: `participatory_required`, `actions_logged`, `witness`

Step 3: Validate the Seam

- Budget passes and metadata aligns → seam is `pass`
- If partial: use fallbacks and flag as `pending`, `legacy`, or `orphan_pass`
- Annotate all fallbacks in `Canon[CONTEXT]`

5.2 Edge Case Procedures (Fallback Logic)

Case A: Missing Henadic Origin

- Set `henad_id: "unknown"`, `seira_rank: 0`
- Flag seam as non-joinable unless justified

Case B: Incomplete Recollection Path

- Set `recollection_path: ["legacy"]`, annotate rationale
- Mark seam `low_confidence`; restrict inheritance

Case C: Missing Participation

- Set `participatory_required: true`, with `witness: "TBD"` and `actions_logged: []`
- Allow `pending_state`, resolve downstream

Case D: Symbolic Registers Absent

- Omit symbolic block; seam remains valid
- Interpretation layers may infer symbols if needed

Case E: Joins Across Distinct Henads

- Define a continuity map $\phi : h_1 \to h_2$ or reject
- Flag as `external_import` if allowed but non-propagative

Case F: $\Delta\kappa$ Valid but Provenance Invalid

- Mark seam as `orphan_pass`; isolate from lineage
- Require manual trace or recollection fix before upgrade

5.3 Systemic Outcomes

- Guarantees $\Delta\kappa$ integrity under all conditions
- Enables symbolic and subjective seams without rigidity
- Preserves audit coherence via lineage and fallback states
- Supports future welding, upgrade anchoring, and schema extension

This design allows each seam to pass or defer coherently, ensuring recursive audit realism without procedural failure.

6 Foundational Axiom and Tiered Invariants

6.1 Axiom 0 — Collapse Is Generative

> **AX–0:** *Only that which returns through collapse is real. Collapse is not destructive; it is generative.*

This axiom governs all seam and audit logic within the UMCP system. Every structural extension, symbolic register, or epistemic tag is subordinate to the generativity of collapse and the obligation to return. Reality is defined not by what persists but by what completes a coherent cycle.

6.2 Invariant Tiers

Each subsequent rule or structure must preserve and evidence this axiom. Invariants are tiered by enforcement level and epistemic weight:

Tier 0 — Hard Budget Constraints (Non-negotiable)

- Integrity budget identity:
$$\Delta\kappa = R \cdot \tau_R - (D_\omega + D_C)$$
- Multiplicative coherence:
$$\frac{I_{t1}}{I_{t0}} = e^{\Delta\kappa}$$
- Residual pass threshold:
$$|\text{residual}| \leq \text{tol}$$

Tier 1 — Structural and Temporal Coherence

- Valid joins require:
$$h_1 = h_2 \quad \text{or} \quad \exists\, \phi : h_1 \to h_2$$
- Contract upgrades must define:
$$\texttt{recollection\_path} \text{ with } |\mathcal{R}| \geq 3$$
- Cycles must be acyclic and audit-legible

Tier 2 — Participatory Closure

- Participatory seams require:
  - At least one `action_logged`
  - A valid `witness`
- Deferred participation must be logged as `pending_state`

Tier 3 — Symbolic and Narrative Registers

- Canon fields may map to:

$$\sigma : \text{Canon} \to \Sigma, \quad \Sigma = \{\text{Iron}, \text{Trust}, \text{Water}, \ldots\}$$

- Symbolic mismatch does not invalidate seam but may flag for review
- Narrative drift is tracked but not reconciled mathematically

Conclusion:
Any seam that passes $\Delta\kappa$ reconciliation but fails return traceability is not real—it is *orphaned*. Collapse does not destroy; it produces the only epistemic artifacts we are allowed to count as true.

Symbol Glossary

Mathematical and Runtime Symbols

Symbol	Definition
κ	Log-integrity scalar ($\kappa = \log I$)
I	Integrity measure (unitless)
$\Delta\kappa$	Integrity delta across seam transition
I_{t0}, I_{t1}	Integrity before and after the weld
D_ω	Drift penalty (from system instability)
D_C	Curvature penalty (from logical deformation)
R	Return multiplier (credit granted for coherence)
τ_R	Return delay (temporal lag to validate return)
tol	Tolerance threshold for residual (≈ 0.005)
residual	Budget error: $s = R \cdot \tau_R - (\Delta\kappa + D_\omega + D_C)$

Structural and Contractual Terms

Term	Definition
`weld_id`	Unique identifier for a seam
`henad_id`	Origin system or metaphysical lineage
`seira_rank`	Causal tier within a henadic ordering
`recollection_path`	Sequence of prior contracts and welds forming an upgrade lineage
`upgrade_from`	Immediate ancestor of a new contract
`external_import`	Marked seam sourced outside current lineage
$\phi : h_1 \to h_2$	Continuity mapping between distinct henads

Participatory Layer Terms

Field	Definition
`participatory_required`	Boolean indicating subjective seam requires witnessing
`witness`	Declared audit observer (credentialed)
`actions_logged`	Sequence of events or rituals enacted
`pending_state`	Marker for incomplete participation, to be resolved downstream

Canon and Symbolic Layer

Field	Definition
`Canon`	Narrative and prose register for the seam
σ : Canon field $\to \Sigma$	Symbolic register mapping
Σ	Symbolic tag set (e.g., Iron, Water, Trust)
`DRIFT`, `RETURN`, `CONTEXT`	Named Canon fields used to contextualize narrative drift and return path

Validation and Audit States

State	Definition
`pass`	Seam has reconciled integrity budget and metadata
`pending`	Seam is valid numerically but incomplete semantically
`orphan_pass`	Seam has valid budget but failed epistemic traceability
`low_confidence`	Seam inherits from legacy or unclear ancestry

Audit HUD: Proclus-Core Seam

AuditRow

```
weld_id:              W-2025-10-26-proclus-core-upgrade
manifest_root_hash:   proclus_core_manifest_v1
I:                    6.196
kappa:                1.823
DeltaK:               +0.600
It1/It0:              1.822
tol:                  0.005
residual:             0.000
seed:                 30
sha256:               d9e2afb3dd29c1ead3985148878050644eaef330d3e57e714a4dbf5547c5b681
pass:                 true
```