

Continuity Without Reconstruction: Defining Preservation Across Sovereign Mutation

Rob Caswell

Authority Engineering

June 2026

Rob Caswell | [linkedin.com/in/rob-caswell3](https://www.linkedin.com/in/rob-caswell3)

Publication Record

Continuity Without Reconstruction: Defining Preservation Across Sovereign Mutation

Author: Rob Caswell

Publisher: Cyber Terrain LLC

Published: 2026-06-01

Zenodo version: 1.0

DOI: <https://doi.org/10.5281/zenodo.21742465>

Website: <https://cyberterrain.ai/papers/cwr-preservation.pdf>

Corpus Context

Corpus: Authority Engineering

Document ID: AE-RES-0003

Object family: AE-RES-0003

Publisher-imprint edition prepared for versioned republication.

The prior Zenodo record remains the citable record until its new version is published.

Abstract

Authority Engineering explains how authority becomes consequence, how consequence becomes reachable, and how reachable consequence transitions organize into observable topologies. These works explain the structure of authority-bearing possibility. They do not explain continuity.

Objects do not survive sovereign mutation intact. Systems transform, infrastructure changes, execution paths mutate, and representations evolve. Yet continuity may persist despite the loss of object identity.

This paper proposes that continuity does not depend upon object persistence. Continuity survives through the joint preservation of Continuity Relationships and Continuity Equivalence across mutation. Preservation therefore becomes a distinct discipline concerned not with reachability, realization, or topology, but with the survival of continuity across transformation.

1. The Preservation Problem

Authority Engineering begins with a simple observation:

Authority determines consequential futures.

From that observation emerge consequence seams, reachable consequence transitions, authority surfaces, and topological arrangements of reachable futures. These works explain how consequence becomes possible. They do not explain persistence.

A system may traverse sovereign boundaries, replace infrastructure, transform representation, mutate execution paths, or completely alter its operational substrate while still exhibiting continuity. The preservation problem therefore asks: What survives mutation when the object itself does not?

This question cannot be answered through object persistence. Objects do not survive sovereign mutation intact. Therefore, continuity cannot depend upon object persistence.

1.1 Sovereign Mutation Typology

A sovereign mutation is not merely a scheduled runtime update or a benign data migration; it is an unmapped shift in execution authority or structural composition. Because continuity must withstand diverse forms of systemic disruption, sovereign mutations are classified into three distinct operational types:

- **Substrate Mutation:** The underlying physical or virtual infrastructure transforms completely. The original execution paths are obliterated, yet the system's operational thread persists.

- **Semantic Mutation:** The structural representations, schemas, or fundamental data properties transform, rendering previous object parsers obsolete. The thing can no longer be read or understood by its original definitions, yet its functional meaning is carried forward.
- **Jurisdictional Mutation:** The state transitions across a consequence seam from the authority of one sovereign consensus rule-set to a completely foreign, distinct, or potentially adversarial rule-set.

Preservation must remain indifferent to the type of mutation encountered. True continuity survives all three.

2. Preservation as a Distinct Discipline

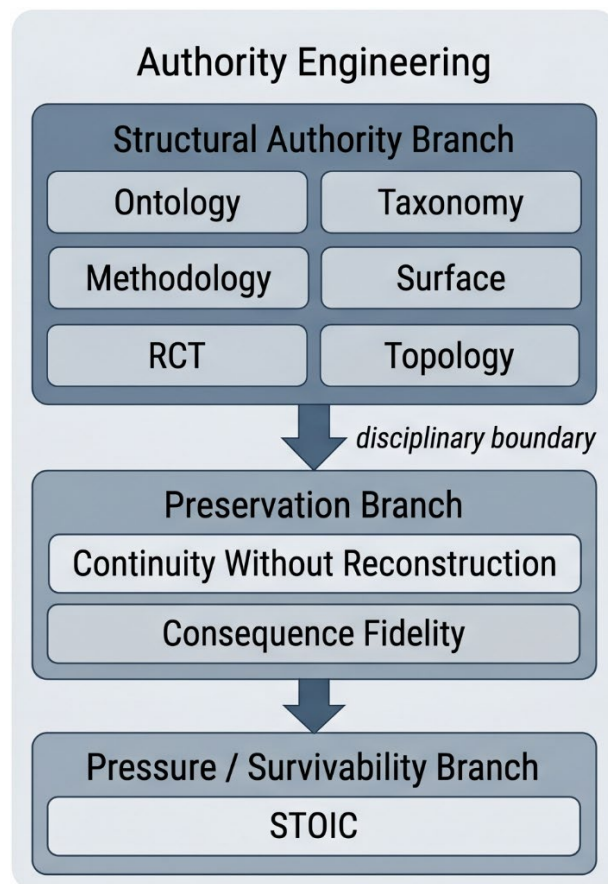


Image 1. Preservation Branch Placement in the Authority Engineering Corpus

The disciplines of Authority Engineering map two distinct dimensions of possibility across identical architectural lines.

The **Structural Authority Branch** maps the *architecture of consequence*:

Authority determines consequential futures.

Consequence Seams delineate sovereign boundaries.

Consequence constrains future system states.

The **Preservation Branch** maps the *architecture of persistence*:

Continuity Relationships define continuity structure.

Sovereign Mutations traverse consequence seams.

Continuity Equivalence defines continuity validity.

Preservation is not merely a theory of structure. It is a theory of persistence—governing how continuity structure survives transit across a consequence seam solely through the active preservation of its validity.

3. Preservation Primitives

The preservation problem cannot be solved through a single primitive. The corpus consistently reveals a primitive set.

3.1 Continuity Relationships

Continuity Relationships are preserved relational dependencies that define continuity structure across mutation. They persist independently of observers and independently of any centralized reconstruction process.

Continuity Relationships answer: **What remains connected?**

They are responsible for:

- Continuity structure
- Continuity substrate
- Topology emergence

- Continuity branching
- Observability

Without Continuity Relationships, continuity loses structure.

3.2 Continuity Equivalence

Continuity Equivalence is the preserved correspondence condition that defines continuity validity across mutation. A system need not preserve identical structure or identical execution, but it must preserve continuity equivalence.

Continuity Equivalence answers: **What remains continuously valid?**

They are responsible for:

- Continuity validity
- Replayability
- Fidelity
- Correspondence
- Preservation legitimacy

Without Continuity Equivalence, continuity loses validity.

4. The Preservation Theorem

Neither primitive is sufficient independently. Preserved relationships without equivalence produce structure without validity. Equivalence without preserved relationships produces validity without structure.

The Central Theorem of Preservation

Continuity survives only through the joint preservation of Continuity Relationships and Continuity Equivalence across mutation. Continuity Relationships define continuity structure. Continuity Equivalence defines continuity validity.

In structural terms:

Continuity survives only when continuity structure and continuity validity remain jointly preserved.

Continuity is neither structural preservation alone nor validity preservation alone. Continuity exists only when both remain jointly preserved.

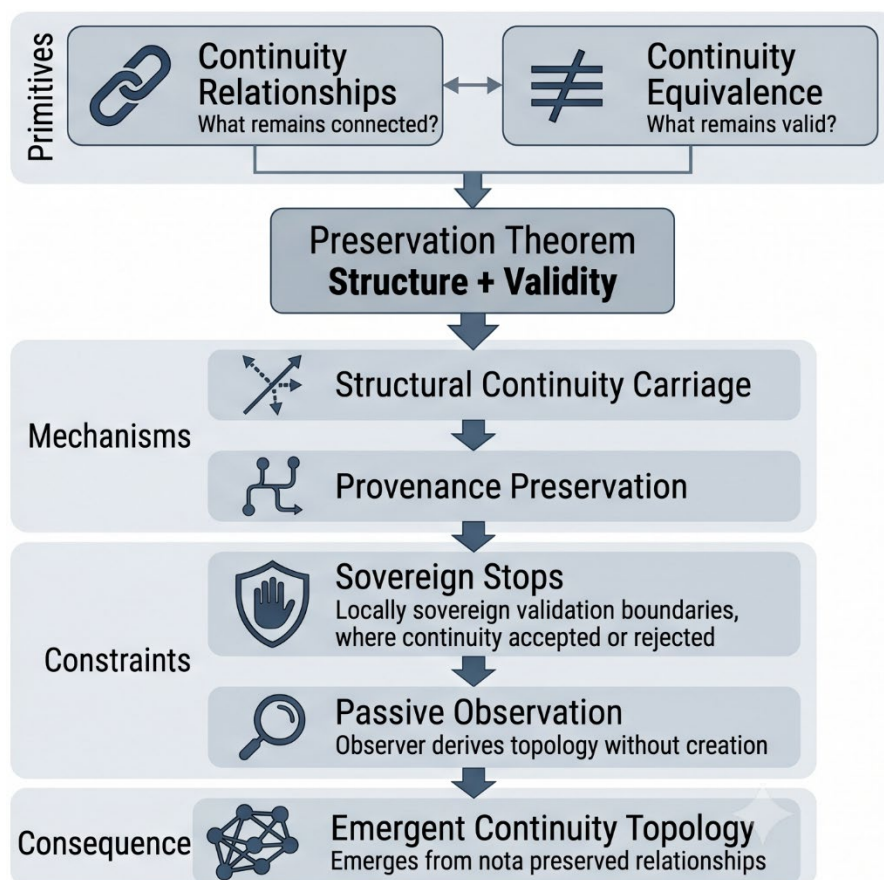


Image 2. Preservation Architecture: Ontology and Mechanism

5. Structural Continuity Carriage

The preservation theorem immediately introduces a new problem: if continuity requires the joint preservation of structure and validity, then some mechanism must carry that preservation across mutation.

Structural Continuity Carriage is that mechanism. It is an abstract vector—a preservation mapping protocol or structural mapping—that ensures both primitives safely land across changing traversal structures, changing representations, and changing execution environments.

Carriage does not constitute the underlying infrastructure itself, nor does it preserve identical form. It preserves continuity across mutation.

6. Provenance Preservation

Preservation requires more than structural carriage across an isolated transition. It requires lineage across time.

Provenance Preservation maintains replay-equivalent lineage across mutation. Replay-equivalent lineage exists when continuity validity remains preserved across every transformation in the continuity chain.

This establishes a fundamental distinction between the two preservation mechanisms:

- **Carriage** handles the event: it preserves continuity across mutation.
- **Provenance** handles the chain: it preserves continuity validity through history.

Without provenance preservation, lineage becomes unverifiable, replayability degrades, and trustworthy continuity cannot reliably distinguish a valid preservation chain from an unauthorized reconstruction.

7. Sovereign Preservation

Preservation remains locally sovereign. Every continuity claim ultimately encounters a sovereign stop.

The **sovereign stop** owns realization authority. The sovereign stop determines participation, convergence, and whether a mutation undergoes acceptance, rejection, delay, transformation, or realization.

No centralized authority can impose continuity. Continuity may only be recognized, validated, or rejected at sovereign boundaries.

7.1 Divergent Continuity

Because continuity is validated locally at the sovereign stop rather than enforced globally, a single mutation event can cause systemic fragmentation. This results in **Divergent Continuity**.

If Sovereign Stop Alpha validates and accepts the joint preservation of structure and validity following a mutation, but Sovereign Stop Beta rejects the transition or interprets the equivalence condition differently, continuity itself forks.

In Authority Engineering, Divergent Continuity is not a global system failure. It represents the birth of non-intersecting, independent continuity branches. Each branch maintains its own valid, localized relationship to historical lineage. The preservation architecture survives along both paths, though their observable topologies will permanently diverge.

8. Passive Observation

Observation does not create continuity. Observation derives continuity. Observation remains passive and non-perturbative.

The observer may discover continuity relationships, evaluate continuity conditions, and derive continuity topology. The observer **may not** create continuity through observation. Continuity exists independently of the observer; it is discovered through structure. Observation remains subordinate to preservation.

9. Emergent Continuity Topology

Topology is not reconstructed. Topology emerges.

When Continuity Relationships remain preserved across mutation, continuity becomes observable. The resulting topology is not centrally authored or imposed through semantic reconstruction; it emerges directly from structure.

Continuity topology is therefore an observational consequence of preservation rather than an engineered artifact of reconstruction. Preservation precedes topology. Topology reveals preservation.

10. Preservation Failure

The architecture of preservation is binary in its failure modes. When a primitive degrades, the systemic loss matches its definitive responsibility.

Failed Element	Immediate Loss	Resulting System State
Continuity Relationships fail	Structure	Disconnected Facades: Pure correspondence remains conceptually, but lacks the structural support to bind it together.
Continuity Equivalence fails	Validity	Ghost Structure: A structural skeleton persists, but its internal validity and meaning are lost.

Failed Element	Immediate Loss	Resulting System State
Structural Continuity Carriage fails	Transition	Obliteration: The continuity condition is destroyed during transit across a sovereign boundary.
Provenance Preservation fails	Lineage	Unverifiable Hallucination: Continuity may exist locally, but it is indistinguishable from an arbitrary reconstruction.

11. Conclusion

Objects do not survive sovereign mutation intact. Continuity therefore cannot depend upon object persistence.

The preservation problem is solved not by maintaining objects, but by preserving the conditions under which continuity remains valid despite their transformation. Continuity survives only through the joint preservation of Continuity Relationships and Continuity Equivalence across mutation.

Structural Continuity Carriage preserves both primitives across mutation, while Provenance Preservation maintains validity through history. Sovereign stops preserve local authority over realization, allowing for divergent continuity branches to safely emerge, while passive observation discovers structure to reveal topology.

The next question is not whether continuity survived, it's whether preserved validity remained bounded to intended authority.

That question belongs to **Consequence Fidelity**.