Reachable Consequence Transitions:

Predictive Validation of Authority Topology Transitions

Rob Caswell

Authority Engineering

Technical Report Series on Derived Law and Autonomous Operations

TR-006

July 2026

Published by Cyber Terrain LLC | cyberterrain.ai

DOI: https://doi.org/10.5281/zenodo.21742415

CYBER TERRAIN

Publication Record

Reachable Consequence Transitions: Predictive Validation of Authority Topology Transitions

Author: Rob Caswell

Publisher: Cyber Terrain LLC

Published: 2026-07-01

Zenodo version: v1.0

DOI: https://doi.org/10.5281/zenodo.21742415

Website: https://cyberterrain.ai/papers/tr-006-reachable-consequence-transitions.pdf

Corpus Context

Corpus: Authority Engineering

Publication series: Technical Report Series on Derived Law and Autonomous Operations

Publication designation: TR-006

Document ID: TR-006

Object family: AE-TRANS-0001

Corpus document ID: AE-TRANS-0001

Publisher-imprint edition prepared for versioned republication.
The prior Zenodo record remains the citable record until its new version is published.

© 2026 Cyber Terrain LLC. All rights reserved.

Abstract

Traditional authorization architectures rely on a transient paradigm: they evaluate whether an incoming operation is authorized based strictly on the system's present authority configuration. While sufficient for static runtime conditions, this model introduces systemic vulnerabilities in dynamic, enterprise-scale environments where the execution of an operation alters the authority architecture itself. By focusing exclusively on the legitimacy of the entry point, contemporary access controls are structurally incapable of predicting how a granted request will mutate the underlying graph. This structural blind spot allows for incremental policy erosion, privilege creeping, and the accidental dissolution of security boundaries.

This report establishes a predictive governance framework derived from the principles of Authority Engineering. By shifting the primary analytical object from individual permission evaluation to the emergent graph topology, we demonstrate that every authority-altering operation constitutes an explicit state transition within an authority graph. We formalize the concept of the Reachable Consequence Space—the mathematical set of all downstream nodes accessible via directed reachability from an entity—and detail the mechanics of validating the resulting Consequence Expansion (ΔC) prior to execution state commit. This predictive validation model intercepts proposed transitions, simulating the future graph state to ensure absolute conformity with structural invariants via deterministic transition evaluation. Ultimately, this framework transitions governance from a reactive reflection on past operations to the verifiable preservation of future system integrity.

Rob Caswell | linkedin.com/in/rob-caswell3

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

PART I: FOUNDATIONAL TRANSITION THEORY

Chapter 1: The Limitation of Present-State Governance

Traditional authorization architectures are inherently transient and localized. When a request is evaluated, the system determines whether the security principals, targets, and attributes satisfy the policy state at that exact microsecond.

This creates a severe structural blind spot. Present-state governance lacks the structural foresight to determine whether the resulting state of a granted request remains valid. By focusing exclusively on the legitimacy of the entry point, traditional models blindly permit operations that gradually degrade the system's structural integrity, allowing for privilege creeping, policy erosion, and the accidental dissolution of security boundaries.

Chapter 2: Paradigm Comparison: Authorization vs. Predictive Validation

To position the contribution of TR-06, we contrast the foundational mechanics of traditional authorization frameworks with the predictive architecture of Authority Engineering:

FIGURE 1: GOVERNANCE PARADIGM COMPARISON
(Illustrative comparsion of governance models proposed in TR-06)

DIMENSION	TRADITIONAL AUTHORIZATION	PREDICTIVE VALIDATION (TR-06)
Evaluation Focus	Evaluates the current operational request	Evaluates the resulting authority topology
Temporal State	Operates statically on the present state	Simulates and evaluates the future state
Object of Governance	Validates individual permissions	Validates the resulting authority topology
Decision Basis	Authorization of request	Structural admissibility of topology transition
Primary Mitigation	Stops explicitly unauthorized actions	Prevents structurally inadmissible topology transitions
Operational Stance	Reactive	Predictive

*This is a comparison of governance models for explanatory purposes. It does not represent a canonical **Authority Engineering** taxonomy.*

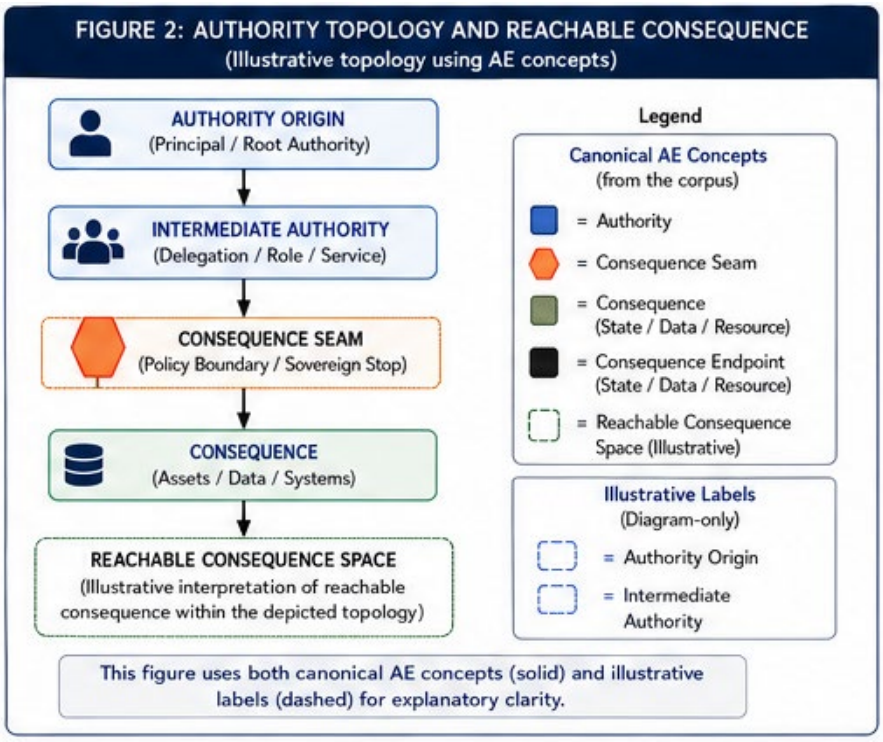
Chapter 3: Authority Topology as the Governing Object

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

To resolve this blind spot, we must elevate the analytical focus from individual permissions to the broader structure of the system. Per the established corpus, authority is not an isolated attribute; it is defined by the graph of relationships between entities. These explicit and implicit relationships form the **Authority Topology**.

Authority topology does not merely describe how authority is organized. It constrains how authority may propagate through the system. Every valid relationship creates or removes possible traversal paths. As the topology evolves, so too does the space of realizable consequence. Reachability is therefore a property of topology rather than of individual permissions.

The structure of this topology directly determines the pathing through which authority can flow, ultimately mapping out the **Reachable Consequence Space**—the total sum of states, data, and resources an entity can impact.



Chapter 4: Every Authorized Action Changes Topology

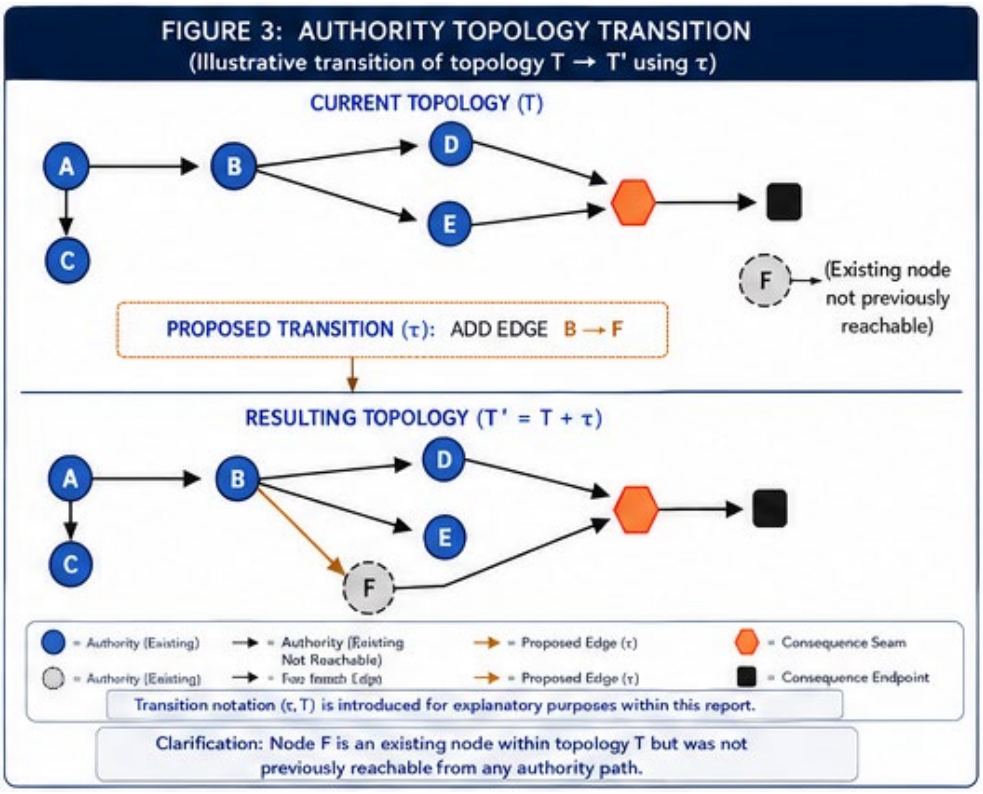
The core architectural pivot of TR-06 is the recognition that every authorization decision that modifies authority relationships is a topology transition. When the system approves an administrative or operational request, it alters the underlying graph. Examples of topology-altering actions include:

- **Delegation:** Extending an authority pathway from one principal to another.
- **Role Assignment:** Mapping a principal to a structural node within the graph.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

- **Policy Modification:** Rewriting the conditional logic governing a relationship edge.
- **Trust Establishment:** Interconnecting two previously isolated sub-graphs.
- **Credential Issuance:** Injecting a new cryptographic verification edge into the topology.
- **Model Promotion:** Elevating an algorithmic component, expanding its relational reach.

Consequently, every lifecycle event must be modeled as an explicit state transition:



PART II: PREDICTIVE EVALUATION AND SPECIFICATION

Chapter 5: Reachable Consequence

Once an authority topology undergoes a transition, the paths available within the graph change. The central governing question must therefore evolve: **What consequence becomes reachable that was previously unreachable?**

To formalize this mathematically without introducing unnecessary primitives, we define the Reachable Consequence Space (C) of a given topology (T) for an entity (e) as the set of all downstream nodes reachable via directed reachability within the authority edges (E):

$$C(e,T) = \left\{ n \in T \mid e \xrightarrow{*} n \right\}$$

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

Where $\overset{*}{\rightarrow}$ denotes the directed reachability paths within the relationship graph.

When a topology transition (τ) is proposed, it generates a modified topology ($T' = \tau(T)$), where τ represents the specific operational transition function applied to the graph state. The validation mechanism evaluates the resulting **Consequence Expansion** (ΔC):

$$\Delta C = C(e, T') \setminus C(e, T)$$

If ΔC violates one or more governing structural invariants, the transition is inadmissible.

Mathematical Notation:

T Current authority topology

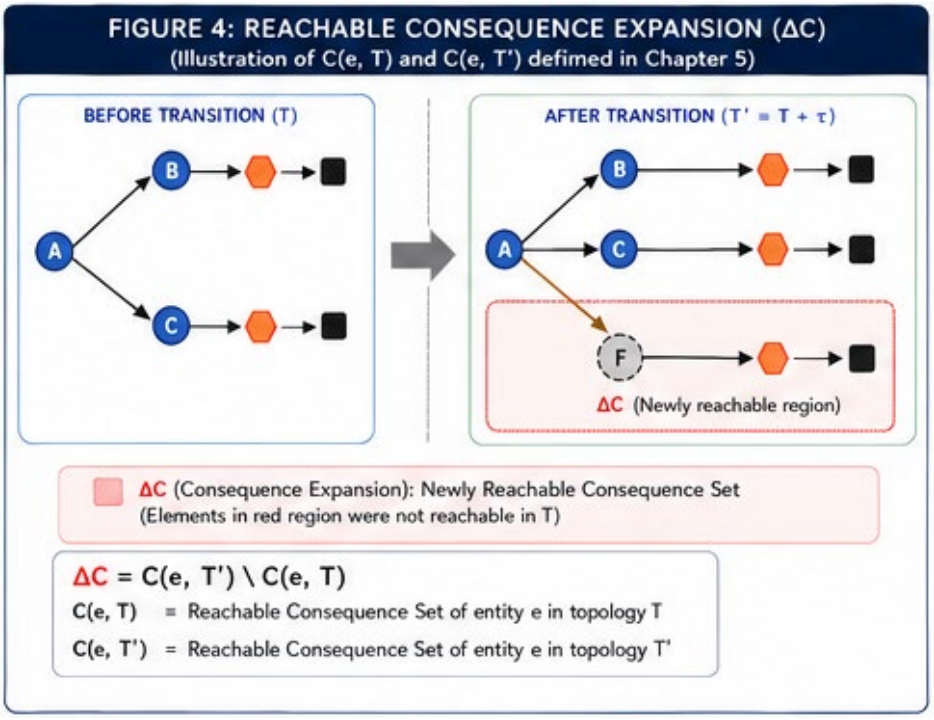
T' Resulting authority topology

τ Proposed topology transition

e Entity under evaluation

C(e,T) Reachable consequence set

ΔC Consequence expansion



Chapter 6: Predictive Topology Validation

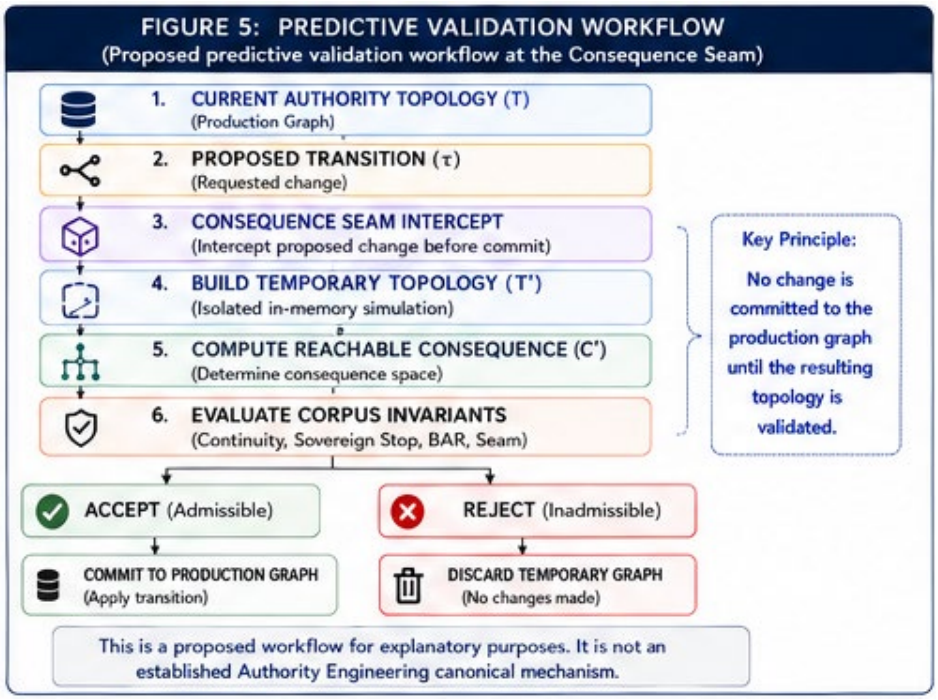
Predictive Topology Validation intercepts the operational pipeline before any state change is written to disk or executed in the runtime environment. By utilizing the comprehensive

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

evaluation pipeline detailed in Figure 3, the governing layer models the potential graph state before committing it.

Mechanically, when an entity requests a modification to the graph (such as installing a new trust edge or delegating a role assignment), the data plane catches the request at the consequence seam. Rather than passing the payload to an administrative service to immediately mutate the production database, the engine generates an isolated, immutable in-memory variant of the target sub-graph.

This pipeline introduces no new ontology. It leverages existing corpus concepts by calculating reachability profiles using the standard relationship graph and measuring the output directly against pre-configured structural invariants. The transition function τ is applied solely to the simulated instance, allowing the engine to traverse all potential downstream paths from the originating principal to map out the theoretical boundary of the new topology.



Chapter 7: Transition Admissibility

Traditional security frameworks often frame denials in moralistic or adversarial terms (*"This action is malicious"*). Authority Engineering shifts this to a purely structural determination: **"This transition produces an invalid authority topology."**

Governing Axiom

Intent is irrelevant; structural admissibility is all that matters.

Admissibility is strictly grounded in four core corpus pillars:

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

- **Continuity:** The transition must not break the explicit lineage tracking of authority assignment. The system must verify that every ancestral edge leading to the transition remains unbroken and signed by a valid, localized root of authority.
- **Sovereign Stop:** The resulting topology must respect absolute, un-bypassable termination nodes that halt authority propagation. If the transition bypasses or establishes an alternative routing around an active Sovereign Stop node, it is immediately discarded.
- **Bounded Authority Realization:** The expanded topology must not allow actualized authority to exceed the maximum theoretical bounds assigned to that entity class. The calculated ΔC must be subset-bounded by the absolute authority threshold of the receiving node.
- **Consequence Seam:** The transition must not bridge or compromise the air-gapped boundaries separating distinct consequence domains. Any edge introduction that provides a logical path crossing an established seam is blocked.

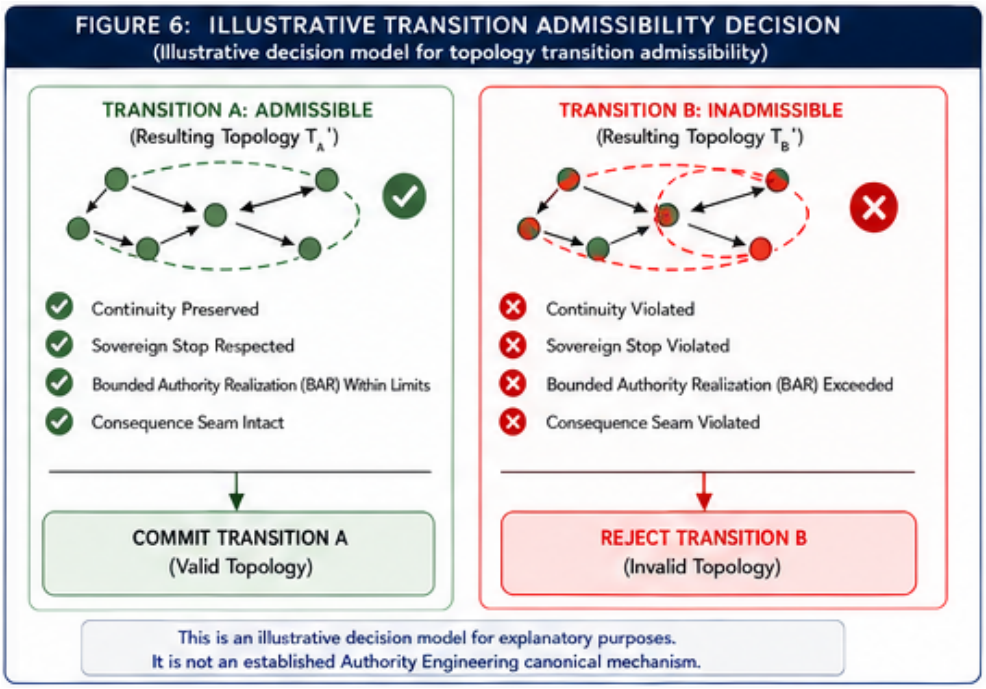
Chapter 8: Preventing Invalid Future States

Rather than relying on vague notions of "catastrophic failure," we define an unacceptable state as a structurally inadmissible topology that violates the architectural invariants of the corpus.

Some topology transitions produce configurations that directly breach these invariants. The predictive engine intercepts these proposals. The transition is rejected not because the user lacks the initial privilege to submit the request, but because the resulting state is architecturally inadmissible.

Traditional reference monitors fail here because they lack memory of the future; they validate that the user has the UpdateTopology permission today, but remain blind to the fact that the requested update will cause an illegal structural convergence tomorrow. By enforcing admissibility bounds during the transition pipeline, the architecture mathematically guarantees that the live execution environment can never land in a corrupted or over-privileged structural state.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415



PART III: OPERATIONAL PROOF & TESTING

Chapter 9: Worked Examples

To illustrate the application of predictive validation, we examine three operational scenarios where authority transitions are evaluated against structural invariants prior to execution. These examples demonstrate that the governing mechanism acts on the emergent topology rather than the isolated legitimacy of the request.

Example 1: Delegation Expansion

Consider an ecosystem with an established baseline topology:

Plaintext

[Admin] ——► [Team Lead] ——► [Production Repository]

An administrative user proposes a delegation transition (τ_1) to grant operational access to an external entity:

Plaintext

[Admin] ——► [Contractor]

The predictive validation engine computes the resulting authority topology and determines that the delegation introduces a previously unreachable traversal path to the Production Repository. Because the resulting topology expands reachable consequence beyond the bounded authority profile permitted for third-party entities, the transition is rejected. The request to delegate access

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

was fully authorized under traditional administrative policy; however, the transition is blocked because the resulting topology is structurally inadmissible.

Example 2: Cross-Domain Trust Relationship

Two structurally isolated authority domains operate independently under strict isolation policies:

Plaintext

[Finance Domain: Core Assets] [HR Domain: Employee Ledger]

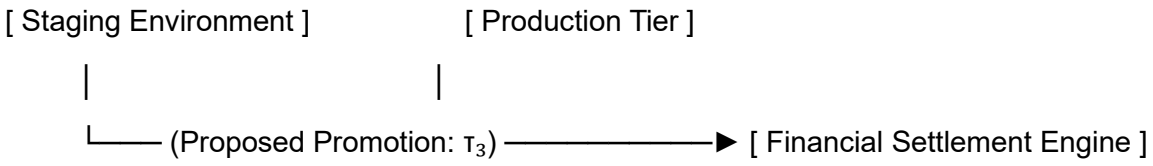
An operations team submits a request to establish a cross-domain trust relationship (τ_2) to streamline an inter-departmental data flow. Individually, neither domain contains a policy violation, and the trust configuration itself satisfies standard validation criteria.

The predictive evaluation models τ_2 and tracks downstream pathing. The simulation discovers that connecting the sub-graphs creates an indirect traversal path allowing identities within the HR Domain to bypass a Consequence Seam protecting financial records. The proposed edge installation violates the structural boundary of Bounded Authority Realization, and the trust relationship is rejected before it can be instantiated in the environment because the connection itself is structurally invalid.

Example 3: Operational Model Promotion

An operational model is scheduled for promotion from a staging environment to a live tier (τ_3). The transition expands the model's relational reach, granting it authority to invoke production workflows.

Plaintext



During the predictive graph state simulation, the reachability analysis determines that the model's new position in the topology enables it to indirectly trigger the Financial Settlement Engine via cascading automated pathways. The promotion is denied. The model is not classified as "malicious," nor is the promotion script erroneous; the transition is blocked because it creates an authority topology that exceeds the admissible bounds established for the promoted model.

Chapter 10: Scientific Falsification Framework

To maintain scientific rigor and distinguish Authority Engineering from descriptive architectural philosophies, the predictive validation framework must make testable, refutable claims. The model advanced in TR-06 is falsifiable and would be proven incorrect if any of the following empirical conditions are demonstrated:

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

- **F1: Topology-Consequence Decoupling:** If two identical authority topologies consistently produce divergent reachable consequence spaces under identical inputs without any topological variation, the foundational premise fails.
- **F2: Pathless Authority Propagation:** If future authority actualization regularly occurs through operational pathways that are entirely absent from the evaluated relationship graph, predictive validation is structurally insufficient.
- **F3: Invariant Independence:** If transitions that violate Authority Engineering structural invariants cannot be distinguished from admissible transitions through topology evaluation, then predictive validation provides no governing advantage.
- **F4: Non-Locality of Downstream Bounds:** If localized, bounded sub-graph evaluations consistently fail to accurately characterize downstream consequence behavior under corpus assumptions, the predictive transition model is unfeasible.
- **F5: Non-Topological Authority Realization:** If a class of system operations can permanently expand an entity's Reachable Consequence Space without altering any existing relationship edges or graph properties, the central thesis of TR-06 is false.

Collectively, these conditions establish that the predictive validation framework is empirically testable rather than merely descriptive. If any of these conditions are demonstrated under the assumptions of the Authority Engineering corpus, the central claims of TR-06 must be revised or rejected.

Chapter 11: Architectural Properties

Implementing predictive validation requires an intentional focus on system architecture. Because the underlying mathematics relies on foundational graph principles, the complexity does not lie in the evaluation of the topology, but in its efficient maintenance.

Key engineering strategies focus on structural modeling and clear architectural properties:

- **Localized Validation:** Decoupling topology validation logic so it can run isolated in-memory graph operations prior to committing changes to the primary data store. By copying small, relevant sub-graphs into low-overhead memory tables, the runtime avoids blocking global system traffic during evaluation.
- **Incremental Topology Maintenance:** Evaluating only the localized sub-graph altered by the proposed transition function (τ), rather than recalculating the global system state. The engine maps the delta boundary (ΔC) relative only to the affected parent and child nodes, drastically reducing computation cycles.
- **Bounded Validation:** Enforcing strict depth limits on traversals to guarantee safety checks remain within deterministic operational boundaries. Graph searches terminate automatically if they exceed the maximum path depth permitted by the entity's boundary context.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

- **Predictable Validation Latency:** This architecture allows validation latency to scale with transition complexity rather than total graph size, making low-latency, deterministic validation feasible even in massive authority graphs.

Chapter 12: Relationship to the Authority Engineering Corpus

TR-06 does not replace the existing canonical foundation; it extends it. By taking the static primitives established in earlier canon and viewing them through the lens of state transition systems, TR-06 operationalizes the following:

- **Authority Topology:** Transformed from a descriptive diagnostic tool into an active, real-time control layer. The graph is no longer an artifact generated by compliance audits; it is the structural mesh through which all modifications must pass.
- **Consequence Seam:** Utilized as a concrete mathematical boundary check during the predictive simulation phase. The seam acts as a structural gate, rejecting any proposed edge that would cause two segregated consequence spaces to intersect.
- **Sovereign Stop & Bounded Authority Realization:** Serving as the definitive, hard-coded validation rules that determine the final admissibility of any proposed graph state. These primitives provide the baseline invariant functions that evaluate the simulated topology before it is committed to production storage.

PART IV: CANONICAL COMPRESSION

Key Takeaways

1. **Every authority-changing operation is a topology transition.** Systems cannot alter permissions without changing the geometry of the underlying graph.
2. **Every topology transition changes reachable consequence.** Modifying relationship edges alters the structural pathing through which real-world impact flows.
3. **Governance should validate the resulting topology before execution.** Evaluating the entry request is transient; predicting the emergent future state is permanent.
4. **Execution is the consequence of an admissible topology, not the object of governance.** Control layers must regulate graph transitions rather than trying to police runtime behavior out-of-band.

Conclusion

TR-06 documents a fundamental departure from the reactive, transaction-centric access control methodologies that have historically characterized enterprise security architectures. Traditional

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415

governance models are bound to a retrospective posture, forced to police execution paths at the point of invocation or document failures downstream via out-of-band audit trails. By demonstrating that the execution of an authorized operation inevitably alters the system's underlying relationship graph, this report establishes that true governance cannot be separated from the structural state transition itself.

The predictive validation framework formalizes this reality by elevating the authority topology to the primary object of system regulation. By simulating the future graph state and analyzing the resulting Consequence Expansion (ΔC) prior to execution, the data plane transitions from a passive observer to an inline structural filter running deterministic validation loops. Under this paradigm, intent, identity legitimacy, and administrative credentials cease to be the sole measures of operational compliance; instead, they are subordinated to the strict, invariant geometric rules of graph reachability. The system enforces safety not by guessing the malice of an actor, but by verifying the structural admissibility of the state that the actor's request would bring into existence.

Ultimately, the engineering realization detailed herein reframes the relationship between policy and runtime mechanics. Governance no longer evaluates execution directly; instead, it establishes and maintains the rigid structural boundaries within which execution is permitted to occur. Execution therefore becomes the consequence of an admissible authority topology rather than the primary object of governance. Future research within this series will focus on applying predictive topology validation to increasingly distributed, partitioned, and dynamically evolving authority environments, expanding the bounds of structural governance across complex infrastructure states.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742415