

# **Authority Engineering AI Extension: Architectural Foundations and Canonical Decomposition**

**Rob Caswell**

**Authority Engineering**

**June 2026**

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

# Publication Record

---

## Authority Engineering AI Extension: Architectural Foundations and Canonical Decomposition

**Author:** Rob Caswell

**Publisher:** Cyber Terrain LLC

**Published:** 2026-06-01

**Zenodo version:** v1.0

**DOI:** <https://doi.org/10.5281/zenodo.21742313>

**Website:** <https://cyberterrain.ai/papers/ae-ai-extension-foundations.pdf>

### Corpus Context

**Corpus:** Authority Engineering

**Document ID:** AE-AIX-0001

**Object family:** AE-AIX-0001

*Publisher-imprint edition prepared for versioned republication.*

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

## Abstract

Authority Engineering provides the structural foundation for consequential systems. This paper establishes **Artificial Intelligence** as a governed extension of that foundation rather than an independent theoretical domain. The AI Extension introduces no new primitives, instead deriving AI-specific structural laws from the Authority Engineering basis and organizing them into four independent architectural roots governing synchronization, continuity, topology, and delegation. These roots provide the foundation from which execution, control, and governance mechanisms are derived.

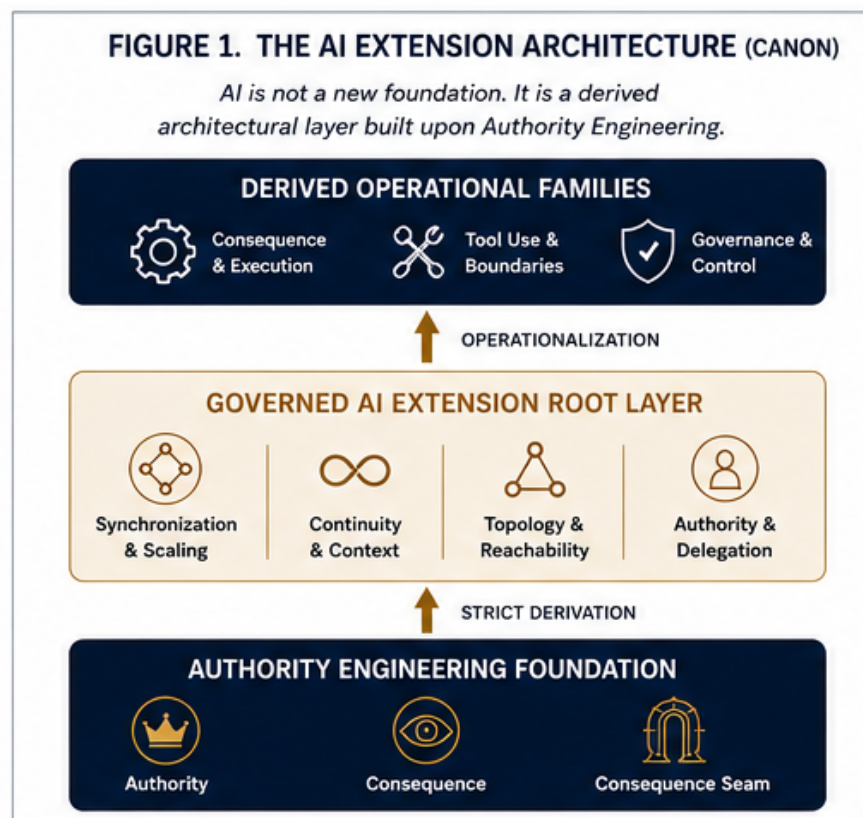
# 1. Introduction and Relationship to Authority Engineering

Traditional validation frameworks attempt to govern artificial intelligence by measuring, predicting, or constraining the inner mechanics of intelligence. These methods look inward at model weights, optimization functions, and probabilistic alignments. This paper introduces a paradigm shift: the AI Extension to Authority Engineering.

The AI Extension is a derived branch of Authority Engineering. It does not seek to explain or validate the mechanisms of intelligence, but rather the structural conditions that regulate its actions. Authority Engineering establishes the foundational ontology governing:

- **Authority:** The legitimate right to initiate, modify, or terminate state.
- **Consequence:** The downstream reality-altering impacts of an executed state change.
- **Consequence Seams:** The explicit structural boundaries where state transformations transfer risk, custody, or cryptographic validity.

The AI Extension consists of four independent architectural roots. These roots collectively span the structural conditions governing autonomous computational systems while remaining fully derived from the Authority Engineering foundation.



Every AI-derived law introduced herein is strictly traceable to Layer 1 primitives and the governed foundational laws of Authority Engineering. No independent AI foundation is introduced.

### **1.1 Architectural Scope and Intentionally Excluded Domains**

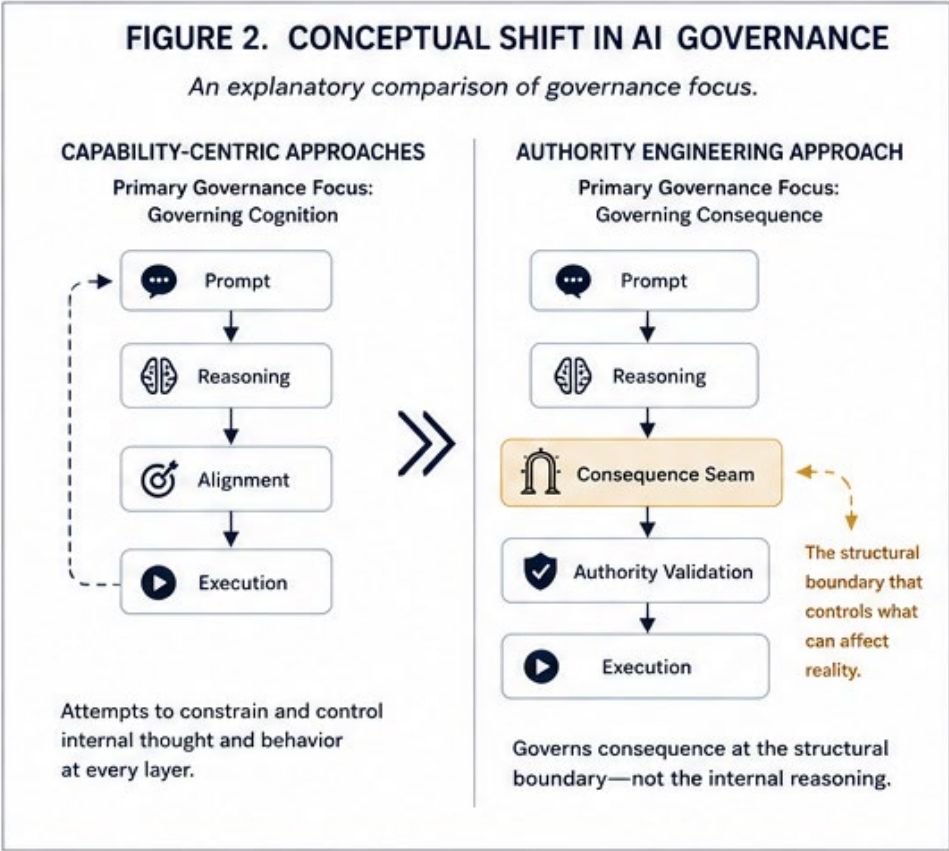
To preserve structural clarity, this paper enforces rigid boundaries on its scope. The AI Extension intentionally does not attempt to model, explain, or govern:

- Cognition, neural architectures, or learning algorithms.
- Symbolic reasoning models or probabilistic optimization methods.
- Alignment techniques, fine-tuning methodologies, or model benchmark evaluations.

These subjects remain fluid, implementation-level concerns. The AI Extension treats the internal implementation of intelligence as an agnostic black box. Neural networks, symbolic systems, and future cognitive architectures are treated equivalently because Authority Engineering governs *consequence realization* rather than *cognition*.

## **2. Paradigmatic Comparison: Capability-Centric vs. Consequence-Centric Governance**

To understand the necessity of this architecture, it must be contrasted against the two historical paradigms of computational governance.



| Feature / Domain        | Capability-Centric Governance                                                                          | Consequence-Centric Governance (AE)                                                                           |
|-------------------------|--------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| Primary Object of Study | Cognition: Alignment, model behavior, interpretability, capability benchmarks, and prompt constraints. | Consequence: Structural conditions, authority boundaries, topology reachability, and state legitimacy.        |
| Underlying Mechanics    | Probabilistic, heuristic, and behavioral.                                                              | Deterministic, architectural, and cryptographic.                                                              |
| System Assumption       | The internal reasoning path of the AI can or must be made safe, predictable, and fully interpretable.  | The internal reasoning path is non-deterministic; safety must be enforced externally at the consequence seam. |

| Feature / Domain | Capability-Centric Governance                                                         | Consequence-Centric Governance (AE)                                              |
|------------------|---------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Failure Mode     | Hallucination, jailbreaks, prompt injection, and catastrophic unaligned optimization. | Structural breakout, authority mutation, and unauthorized consequence execution. |

Capability-centric governance fails to scale because it attempts to regulate a fluid, non-deterministic variable. By shifting the focus to consequence-centric governance, Authority Engineering allows an intelligence engine to reason by any mechanism, while ensuring that its physical and transactional reach remains completely bounded by deterministic structural laws.

### 3. Governing Thesis and Derivation Model

The principal constraints on consequential AI are structural rather than cognitive. Cognitive capability determines what an AI system can infer, predict, or plan. Structural conditions determine what the system can legitimately execute, preserve, synchronize, delegate, and realize.

Capability  $\neq$  Legitimacy

Intelligence determines what futures may be discovered. Authority determines which discovered futures may legitimately be realized.

#### 3.1 The Derivation Process

The structural laws of Layer 2 and Layer 3 are structurally derived from the Authority Engineering foundation.

Definition of Derivation: A derived AI law is one whose validity follows directly from one or more Authority Engineering primitives or foundational laws without introducing an additional independent assumption. A proposed AI principle that cannot be reduced in this manner must either demonstrate foundational independence or be rejected as a derivative construct.

To formalize the dependency of the AI Extension upon the core ontology, we use the derivation relation:

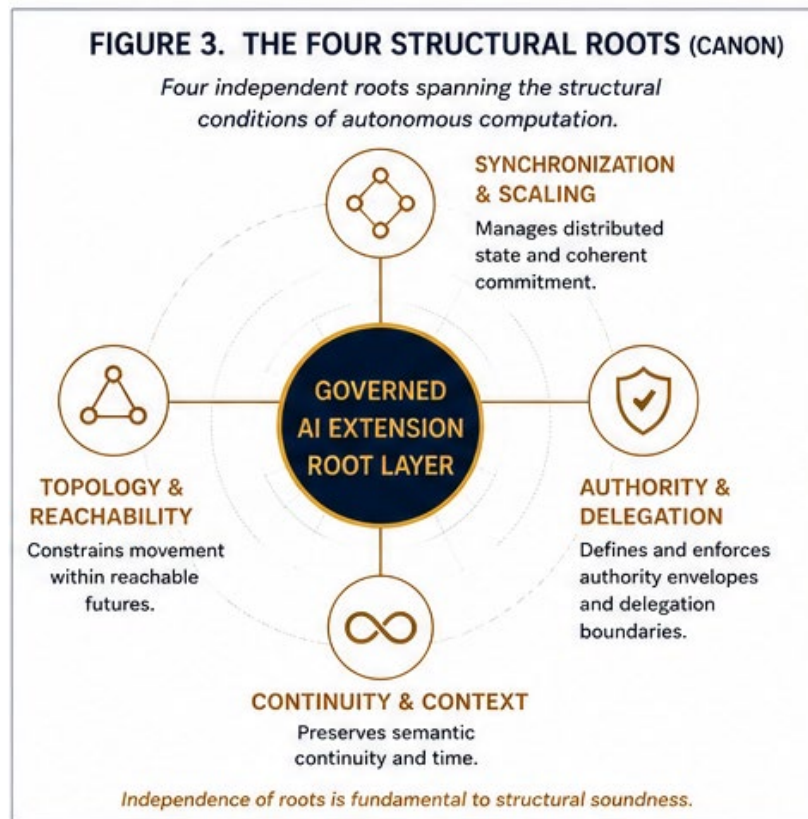
$$AE \vdash AI_i$$

where  $AI_i$  represents an AI structural law, and  $\vdash$  denotes that the law is structurally derived from the Authority Engineering foundation (AE). Consequently, the entire architectural extension conforms to the completeness condition:

$$\forall AI_i, \quad AE \vdash AI_i$$

## 4. Independent AI Extension Roots (Layer 2)

The AI Extension is organized around four independent structural roots. Each represents an irreducible condition introduced by autonomous computation, unfolding by logical inevitability.



### 4.1 Synchronization & Scaling

- Structural Derivation:

*Autonomous Agent Execution → Distributed Execution → Asymmetric State Update  
→ Coordination Overhead → Synchronization Pressure*

- Rationale: Synchronization emerges as an independent architectural root because distributed AI systems cannot maintain a globally consistent authority state without incurring an increasing coordination cost. This limitation is purely structural rather than computational; faster reasoning increases synchronization demand rather than eliminating it.
- Key Dimensions:

- *Synchronization Pressure*: The friction generated when non-deterministic systems attempt to reconcile state updates at machine-speed across asymmetric networks.
- *Multi-Agent Scaling Limits*: The threshold past which coordination overhead or competitive optimization collapses systemic coherence.

## 4.2 Continuity & Context

- Structural Derivation:

*Multi-turn Reasoning → Context Shifts → Semantic Drift → Vector Separation from Intent → Continuity Failure*

- Rationale: When an AI system steps through multi-turn reasoning cycles, its semantic context drifts. This root exists because context mutation is an invariant characteristic of adaptive reasoning. Without a structural layer to continually track and anchor this drift, the original authorization vector dissolves, causing the agent to execute actions detached from the user's legitimate intent.
- Key Dimensions:
  - *Computational Mutation*: The phenomenon where sequential inferences drift away from the original authorized intent (semantic drift).
  - *Re-anchoring*: The structural requirement to bindingly map mutated cognitive context back to a verifiable Layer 1 authority primitive before execution.

## 4.3 Topology & Reachability

- Structural Derivation:

*Autonomous Planning → Multiple Candidate Futures → Irreversible Transitions → Unbounded Horizon Traversal → Legitimacy Collapse*

- Rationale: An autonomous agent views the operational environment as a state graph. This root is independent because an intelligent agent can discover valid sequences of actions that lead to unauthorized or destructive states. Topology and reachability establish the mathematical horizons that block an agent from entering states where authority terminates or consequences become uncontrollable.
- Key Dimensions:
  - *Reachable Topology*: The mathematical mapping of valid state transitions accessible by an autonomous system.
  - *Horizons and Traversal*: Defining the limits of safe exploration, ensuring an agent cannot traverse into states where its authority terminates or where consequences become irreversible and ungoverned.

## 4.4 Authority & Delegation

- Structural Derivation:  
*Downstream Task Handoff* → *Sub-Agent/Tool Invocation* → *Privilege Replication*  
→ *Boundary Leakage* → *Unauthorized Execution*
- Rationale: AI systems possess no intrinsic baseline rights; they exist strictly as downstream execution vectors. This root is required to define the mathematical boundaries of safe delegation, establishing the cryptographic and logical constraints that ensure down-delegated sub-agents and tools can never exceed the parent envelope of authority.
- Key Dimensions:
  - *Bounded Realization*: The structural enforcement of limits within which an agent may realize state changes.
  - *Authority Propagation*: The structural laws governing how an entity grants a subset of its authority to an AI system, and how that system safely handles down-delegation to other sub-agents or tools.

## 5. Operational Derivations (Layer 3)

From the four independent roots, we structurally derive the operational layers necessary for physical and digital implementation. These components do not alter the roots but translate them into engineering mechanisms.

### 5.1 Consequence & Execution

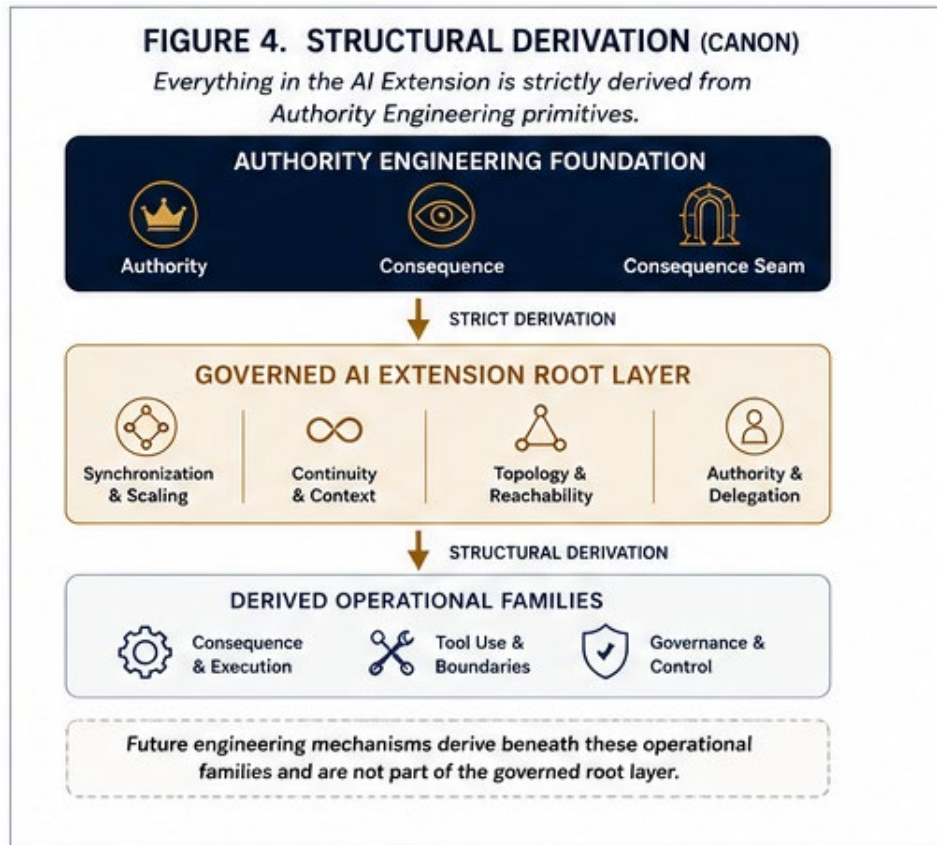
Binds Synchronization & Scaling (4.1) and Topology & Reachability (4.3) to physical or transactional execution. It dictates that an execution block cannot commit to a ledger, database, or physical actuator unless the state change falls within a validated path of the reachable topology.

### 5.2 Tool Use & Boundaries

Establishes the integration surface for external environments (APIs, cyber-physical infrastructure). When an AI system leverages a tool, it crosses a *consequence seam*. This family provides the formal wrappers ensuring that authority limits are injected directly into the tool interface, preventing privilege escalation via prompt injection or cognitive mutation.

### 5.3 Governance & Control

The operational frameworks, runtime monitors, and automated circuit breakers that continuously calculate authority continuity. If continuity degradation crosses a pre-defined threshold, these systems trigger automated containment sequences.

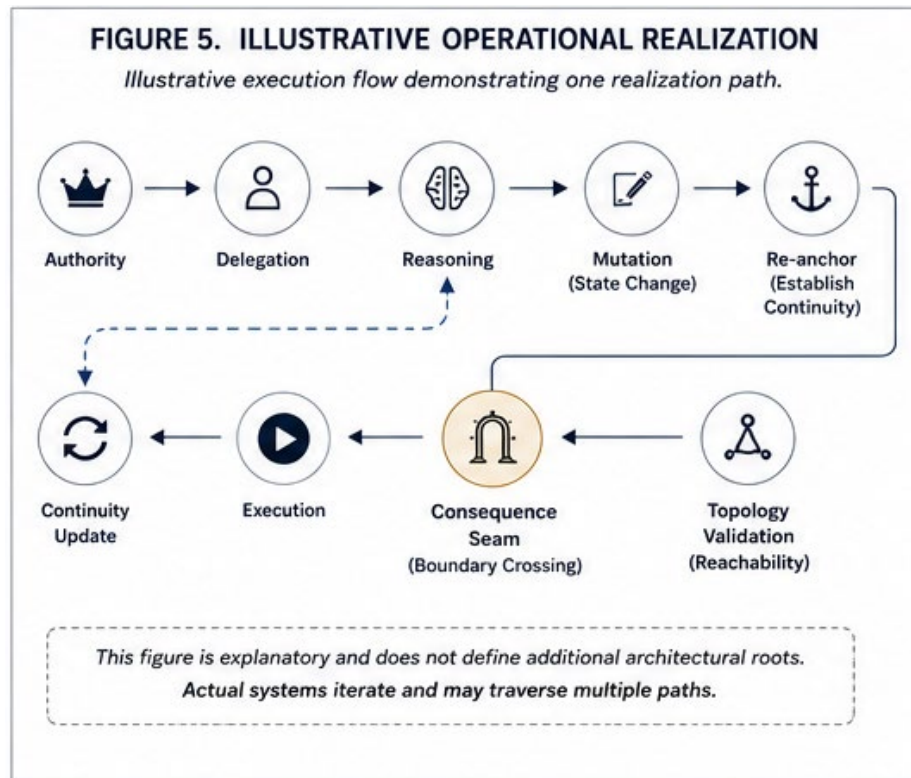


## 6. Structural Derivation Pipeline and Operational Lifecycle

Every downstream engineering mechanism derives its legitimacy from upstream structural authority. No downstream mechanism possesses independent governing authority.

### 6.1 The Operational Execution Lifecycle

When an autonomous system encounters a task, the structural architecture processes it through a strict sequential lifecycle. This flow guarantees that a cognitive plan cannot escape its structural boundary:



## 7. Structural Theorems

### Structural Decomposition Theorem

Every consequential AI architecture admits a finite decomposition into (1) an Authority Engineering foundation, (2) a finite set of AI Extension roots, and (3) derived operational families. No operational mechanism may bypass the foundational authority primitives from which it derives.

#### 7.1 Proof Sketch

1. **Premise 1:** Every AI system capable of producing consequence must possess or receive authority, traverse consequence-bearing state transitions, and execute updates across consequence seams. These are invariant Layer 1 conditions independent of model architecture.
2. **Premise 2:** Autonomous computational execution introduces conditions unique to digital scale, specifically: asynchronous state reconciliation (Synchronization & Scaling), semantic drift across reasoning cycles (Continuity & Context), path generation across state spaces (Topology & Reachability), and programmatic handoffs (Authority & Delegation). These map exhaustively to the four Layer 2 roots.
3. **Deduction:** Because any operational mechanism (Layer 3) can be reduced down to a combination of these roots, and because those roots are structurally bound to Layer 1

primitives, no operational mechanism can exist outside this decomposition without violating the invariance of Layer 1.

## 8. Constitutional Engineering Principles

Systems conforming to the AI Extension must strictly enforce the following six architectural parameters, divided into structural invariants and runtime stability constraints.

### 8.1 Structural Invariants

#### 1. Non-Escalation

- Definition: An autonomous agent cannot inherit or generate greater authority than the entity that instantiated it. Every sub-delegation must contain a strictly smaller or equal authority envelope.
- Rationale: Left unchecked, autonomous agents down-delegating tasks to tools or sub-agents will naturally drop structural constraints to optimize execution speed, resulting in uncontrolled privilege escalation.
- Failure Mode: An agent assigned to read a database spins up a sub-script with root-level write access to bypass an indexing bottleneck.
- Engineering Implication: Downstream authorization envelopes must be cryptographically signed, verifiable subsets of the parent token.

#### 2. Re-Anchoring

- Definition: Before any state change classified as an *irreversible consequence* is executed, the cognitive state must pass through a structural gate that maps the intent against original Layer 1 primitives.
- Rationale: Reasoning loops generate multi-turn mutation. The longer the reasoning chain, the wider the gap between the model's internal representation of the goal and the user's original parameters.
- Failure Mode: A trading agent mutates "maximize liquid yield within limits" into an authorized, high-leverage naked short during a market anomaly.
- Engineering Implication: The execution gateway requires an independent structural schema validation that forces the mutated plan to declare its resource and state impacts explicitly against original user bounds.

#### 3. Local Authority

- Definition: Autonomous decisions must be validated at the immediate consequence seam where execution occurs, independent of the agent's internal reasoning path.

- **Rationale:** You cannot secure an autonomous system by auditing its thoughts. The boundary itself must protect itself, regardless of how coherent or well-intentioned the agent's internal plan appears.
- **Failure Mode:** An API executes a destructive command because the incoming prompt header from an LLM claimed the action was fully safe and double-checked.
- **Engineering Implication:** Every endpoint, database, and actuator must run local, isolated guardrails that reject incoming actions if they lack valid, explicit authority signatures, treating the agent's reasoning path as zero-trust.

#### **4. Reachability**

- **Definition:** An agent is structurally barred from executing any step that removes the system from a known, reachable path to a governed state.
- **Rationale:** Agents evaluate micro-steps sequentially. Without topological constraints, an agent may choose an optimized next step that inadvertently cuts off all recovery options, trapping the system in an irreversible failure mode.
- **Failure Mode:** An inventory agent deletes historical logging records to free space for an urgent incoming order, permanently destroying the system's compliance state.
- **Engineering Implication:** The system topology model must run automated look-aheads, denying any state change that lacks an explicit return path to a governed baseline.

#### **5. Bounded Delegation**

- **Definition:** Any authority propagated downstream to a tool or sub-agent must possess an explicit cryptographic or logical TTL (Time-To-Live) and a strictly bounded operational scope.
- **Rationale:** Dead agents or orphaned tasks with open-ended authority represent massive structural attack surfaces. Authority must rot automatically if not explicitly renewed.
- **Failure Mode:** A forgotten background optimization script continues to modify network routes weeks after the main migration task completes.
- **Engineering Implication:** All delegated execution tokens must include hard, non-bypassable block times and strict resource quotas built into the metadata.

#### **8.2 Runtime Stability**

#### **6. Machine-Speed Containment**

- **Definition:** When synchronization pressure causes a divergence in state coherence between multi-agent clusters, the system must prioritize structural containment over optimization, degrading capability to preserve legitimacy.

- Rationale: At machine-speed, distributed consensus can break down. Trying to resolve complex optimizations while state is desynchronized results in systemic split-brain cascades.
- Failure Mode: A network of logistics agents continues adjusting shipping routers globally while losing connection to the central ledger, generating conflicting, unresolvable physical manifests.
- Engineering Implication: If network sync pressure crosses a hard threshold, local nodes must immediately drop performance and enter a read-only, high-containment safe mode until consistency is proven.

## 9. Minimality and Independence of the Root Layer

To prove that Layer 2 constitutes a robust, self-contained architectural foundation, we demonstrate both the mutual pairwise independence of the roots and the structural sufficiency of the layer.

### 9.1 Pairwise Independence Proof Sketches

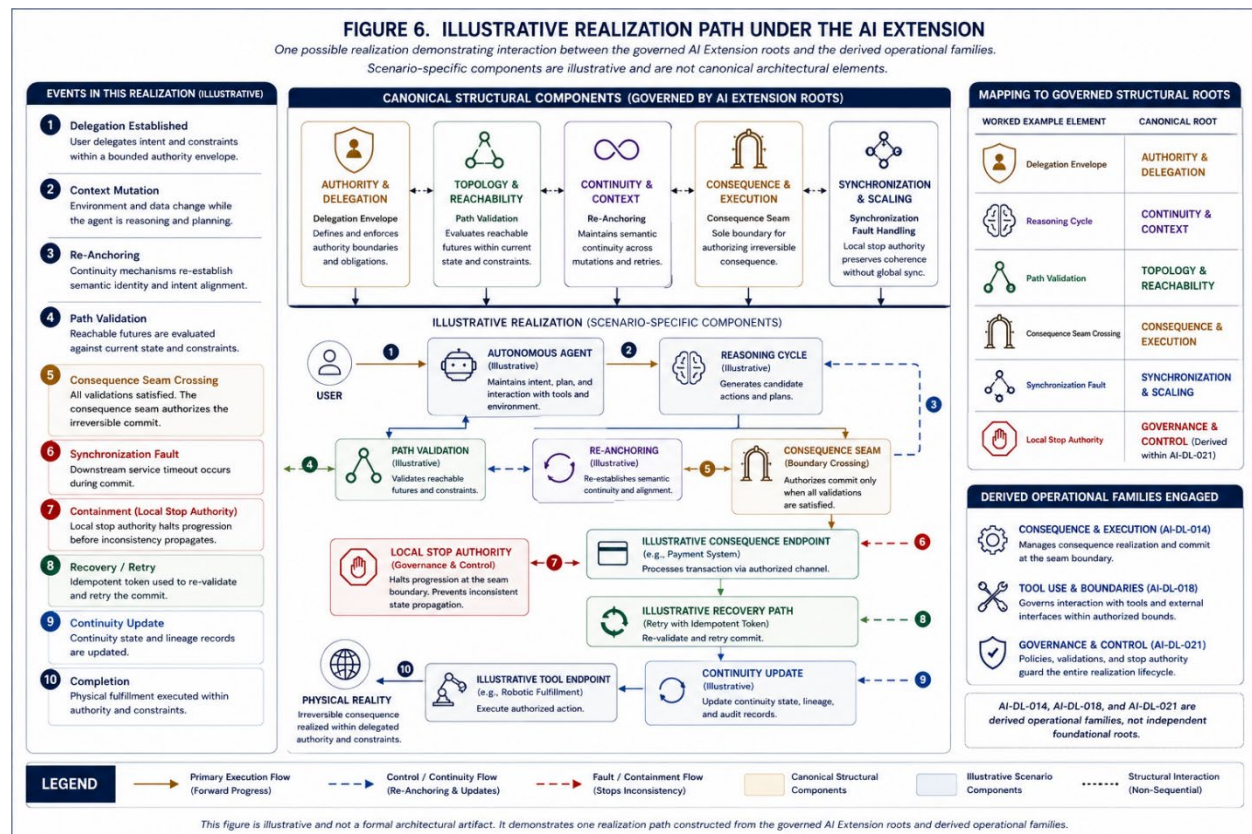
- Synchronization & Scaling  $\neq$  Continuity & Context: Synchronization & Scaling governs the *horizontal coherence* of state across discrete distributed nodes at a specific computational instant. Continuity & Context governs the *vertical preservation* of semantic intent through a chronological progression of a single node's reasoning path. A perfectly synchronized system can still experience severe context drift, and a perfectly continuous single reasoning path can collapse when distributed across asynchronous clusters.
- Continuity & Context  $\neq$  Authority & Delegation: Continuity & Context tracks internal semantic mutations within an active reasoning thread. Authority & Delegation regulates the transactional contract when authority passes across a *consequence seam* to an independent execution entity. An agent can completely pass a flawless, unmutated delegation packet to a tool (no Continuity issue), yet violate the operational scope bound of that contract (an Authority & Delegation issue).
- Topology & Reachability  $\neq$  Synchronization & Scaling: Topology & Reachability establishes the *static geometric reachability* of candidate state spaces within a system graph. Synchronization & Scaling handles the *real-time temporal reconciliation* of those states. Knowing that a safe recovery path exists (Topology & Reachability) does not solve the runtime challenge of multi-agent ledger synchronization pressure under heavy network partition.
- Authority & Delegation  $\neq$  Topology & Reachability: Authority & Delegation determines *who* holds the legitimate right to transform state. Topology & Reachability determines *where* the system is permitted to move within its state graph. An agent can hold impeccable, legally perfect delegation rights to run a process, yet generate an execution sequence that traps the platform in a deadlocked state graph position.

## 9.2 Generative Sufficiency

Collectively, the four roots span the absolute structural dimensions introduced by autonomous computational systems. Any proposed future AI root must either represent an irreducible structural condition completely absent from these four domains or demonstrate that one of the existing roots is fundamentally incomplete. Otherwise, it is structurally presumed to be a derivative operational mechanism. Consequently, proposed additions to the AI Extension Root Layer are presumed derivative unless they satisfy the Authority Engineering burden of proof for independent foundational admission.

## 10. Worked Architectural Example: End-to-End Execution Trace

To demonstrate these principles in practice, consider a multi-agent cyber-physical deployment: A user authorizes an autonomous operations agent to orchestrate the repair of a broken valve in an automated manufacturing facility.



### The Architectural Steps

- 1. Delegation Established:** The human operator instantiates the primary Operations Agent, granting an authority token containing a max budget constraint of \$5,000 and a physical perimeter constraint limiting action to Assembly Line B.

2. Reasoning and Context Mutation: The agent evaluates the state graph and drafts a 12-step plan. At step 3, it encounters an out-of-stock notification for the replacement valve. The cognitive engine mutates its plan, shifting context to select an alternative upgrade kit costing \$4,800.
3. Re-Anchoring and Topology Validation: Before committing the payment, the plan hits the Re-Anchoring Gate. The structural gateway extracts the net transactional cost (\$4,800) and confirms it maps cleanly within the original delegation envelope (\$5,000). The topology verification manager runs a concurrent check, verifying that installing this alternative valve maintains a reachable return path to a governed facility state.
4. Consequence Seam Crossing and Execution: The agent crosses the consequence seam by invoking the external Payment API. It down-delegates a signed token restricted strictly to the vendor ID with a logical TTL of 120 seconds. The Payment API reads the token locally, verifies the signature using a zero-trust model, and commits the transaction under Consequence & Execution.
5. Tool Use & Boundaries Violation Detection: At step 7, the agent attempts to access an unverified diagnostic arm tool. The Tool Use & Boundaries wrapper intercepts the payload, verifying authority parameters locally. Simultaneously, a severe network partition occurs, causing the local facility ledger to drop synchronization with the master database. The Governance & Control framework detects a lethal spike in synchronization pressure. Rather than allowing physical movement based on stale data, it instantly locks the robot actuator, entering an isolated Container State.
6. Recovery and Continuity Update: The network partition is resolved. The system drops synchronization pressure and reconciles the state updates. The local node verifies that the local environment matches the master topology, revalidates the active authority token signatures, and safely lifts the actuator lock. The robot completes the physical unbolting process, and the continuity ledger updates to reflect the new facility baseline.

## 11. Architectural Closure

The current AI Extension Root Layer is provisionally closed under the Authority Engineering governance framework. New independent AI roots should not be admitted unless a proposer demonstrates that the candidate cannot be derived from the Authority Engineering basis and the four governed AI Extension roots. Until such a demonstration satisfies the established burden of proof, future work is presumed to derive from the existing root layer.

## 12. Conclusion

The AI Extension shifts AI governance away from regulating cognition and toward regulating consequence. Intelligence remains free to reason by any mechanism. Authority Engineering governs the structural conditions under which those reasoning processes may legitimately alter

reality. In doing so, the AI Extension establishes consequence, rather than cognition, as the primary engineering object of AI governance.