The Cascading Delegation Problem:

Delegation, Continuity, and Local Authority in Autonomous Systems

Rob Caswell

Authority Engineering

Technical Report Series on Derived Law and Autonomous Operations

TR-002

July 2026

Published by Cyber Terrain LLC | cyberterrain.ai

DOI: https://doi.org/10.5281/zenodo.21742398

CYBER TERRAIN

Publication Record

The Cascading Delegation Problem: Delegation, Continuity, and Local Authority in Autonomous Systems

Author: Rob Caswell

Publisher: Cyber Terrain LLC

Published: 2026-07-01

Zenodo version: v1.0

DOI: https://doi.org/10.5281/zenodo.21742398

Website: https://cyberterrain.ai/papers/tr-002-cascading-delegation-problem.pdf

Corpus Context

Corpus: Authority Engineering

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

Publication designation: TR-002

Document ID: TR-002

Object family: TR-002

Corpus document ID: AE-TR-0002

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

Large language models rarely execute consequential actions directly. Instead, modern autonomous systems increasingly rely on chains of delegated execution spanning planners, coordinators, specialists, tool agents, and execution interfaces. Each additional delegation expands operational capability while simultaneously increasing opportunities for execution mutation, authority creep, and structural degradation.

This report examines the Cascading Delegation Problem through the Authority Engineering AI Extension, explaining why delegated capability does not relocate sovereign authority. Grounded in the Authority & Delegation family, the report demonstrates that valid delegation requires continuity-equivalent traversal between locally governed realization points. Delegation chains remain structurally valid only when authority consequence continuity and local realization authority are preserved throughout execution, while adjacent continuity support reinforces mutation-tolerant provenance where auditability matters.

This report is an explanatory introduction built upon accepted corpus results. It introduces no new governing doctrine and establishes no additional derived law.

1. Why This Report Matters

How does authority remain valid as execution traverses multiple autonomous actors?

Artificial intelligence is rapidly shifting from single, isolated models to dense networks of cooperating autonomous agents. To fulfill complex objectives, contemporary architectures increase delegation depth dynamically. A single user prompt may trigger a cascading sequence of delegated tasks across multiple layers of specialized orchestration.

Existing multi-agent research largely asks how autonomous agents should coordinate, communicate, or optimize their planning steps. This report asks a different question: how does authority remain valid as execution traverses multiple autonomous actors?

By treating multi-agent operations as an authority-preservation architecture problem rather than a tool orchestration problem, we expose structural vulnerabilities that standard engineering metrics completely misdiagnose.

2. A Minimal Introduction to Authority Engineering

What are the foundational architectural elements that govern autonomous actions?

Intelligence vs. Authority

Traditional computer science and AI governance frameworks treat safety as an engineering constraint on intelligence. They ask how deeply a model can reason, how accurately it parses commands, or how reliably its safety alignments inhibit malicious intent. Authority Engineering inverts this paradigm. It starts with *authority* rather than *intelligence*.

Intelligence is merely the capacity to discover, evaluate, and navigate possibilities. Authority is the operational capacity to determine which of those futures may become consequence. An

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

autonomous model can generate an optimization vector, simulate a path, or draft an execution script, but it cannot manufacture the underlying authorization required to bind reality.

Core Definitions

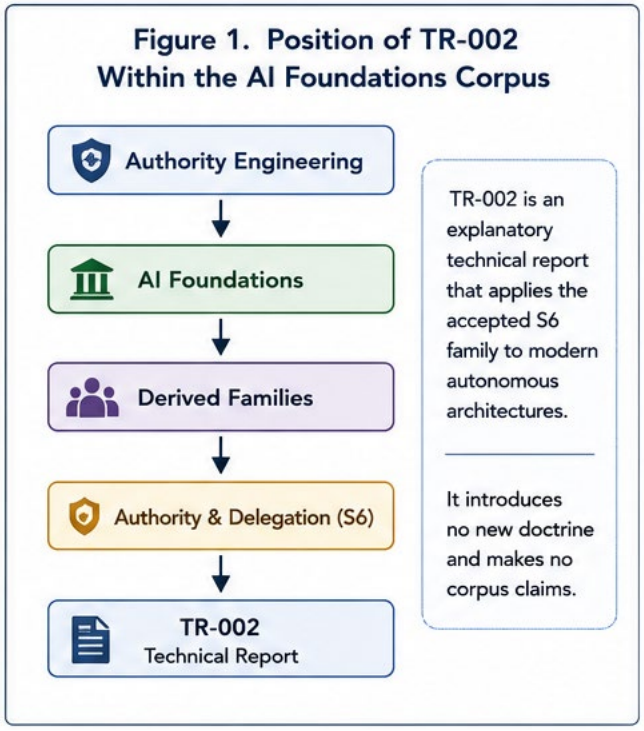
- **Authority:** The operational capacity to determine which futures may become consequence. Boundaries are where this capacity is structurally exercised.
- **Delegation:** Transmits tasking and execution across actors, but does not relocate the local authority that permits consequence realization.
- **Accountability:** Questions of liability and responsibility may be practically relevant, but the present corpus does not yet freeze a complete human-AI delegation doctrine.
- **Consequence:** An irreversible state transition.
- **Consequence Seam:** The final boundary before realization occurs.
- **Local Authority:** The property establishing that authority is exercised exclusively where consequence becomes irreversible.

The AI Foundations Corpus

The AI Foundations series represents a formal, black-letter corpus governing autonomous systems. Within this corpus, derived families operationalize theoretical boundaries into actionable engineering constraints. The Authority & Delegation family (S6) is governed by AI-DL-007 (Local Stop Authority) as its root, supported by the theorems AI-DL-015 and AI-DL-011, and completed by the corollary AI-DL-008 (Orchestration Degrades Sovereignty).

This report progressively explains each of these structural primitives to build engineering intuition. It does not establish new law; it explains one accepted architectural result from the S6 family for engineers and architects.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398



3. The Rise of Delegated Intelligence

How has the operational topology of AI systems changed as architectures transitioned from direct execution to multi-agent networks?

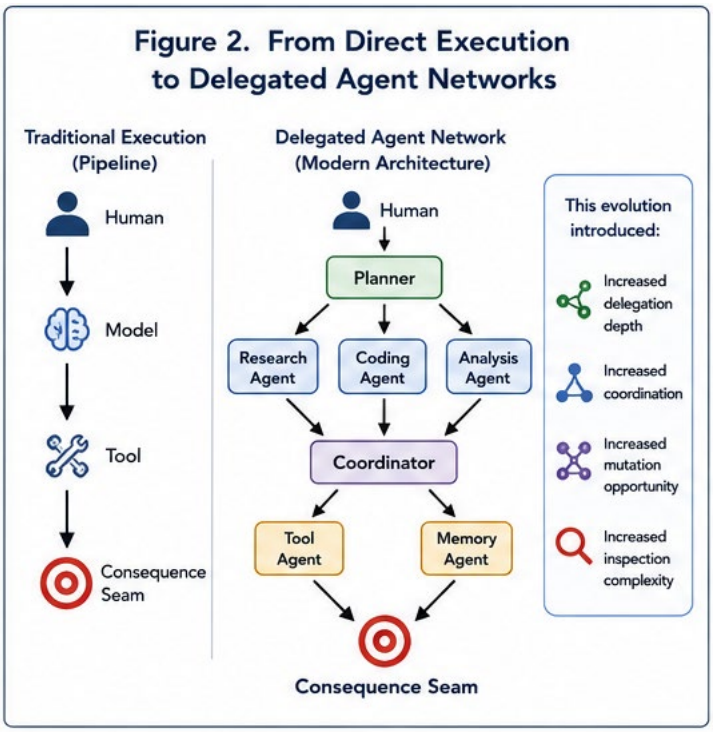
The first generation of AI deployments relied on a simple transaction: a human user issued a prompt, a model evaluated the token window, and the model returned a static response or directly called an atomic tool. Governance was straightforward because the system topology was flat.

Contemporary systems have abandoned this flat architecture in favor of deep, multi-layered agent societies. Modern orchestration frameworks introduce multiple degrees of separation between the originating prompt and the final consequence:

1. **Planners:** Deconstruct an abstract goal into sub-tasks and manage high-level execution loops.
2. **Coordinators:** Route tasks to specialized sub-agents, balancing dynamic context windows.
3. **Specialists:** Domain-specific agents trained to optimize narrow target objectives (e.g., code generation, market analysis).
4. **MCP Servers & Tool Interfaces:** Mediate agent requests into real-world database writes, API calls, and system execution.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

Multi-agent architectures emerged because no single model optimizes every objective simultaneously. Delegation enables specialization, context management, tool isolation, fault tolerance, and parallel execution. As a result, modern AI systems increasingly trade architectural simplicity for operational capability, creating deeper execution chains whose governing authority becomes progressively more difficult to inspect.



4. Delegation Is Not Authority

What is the difference between passing an operational task downstream and shifting sovereign power?

Many AI orchestration frameworks implicitly assume that authority accompanies the delegation chain. They operate under the assumption that if a planner invokes a coordinator, and that coordinator invokes an execution agent, the decision-making authority has moved progressively downstream.

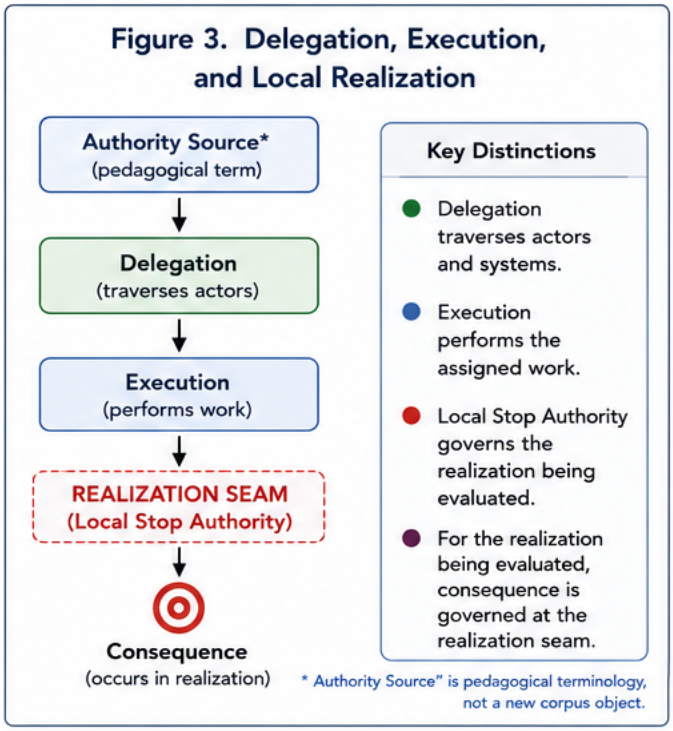
The Authority Engineering corpus rejects this assumption. Delegation transfers operational responsibility for a task. It does not transfer sovereign authority over consequence realization. Authority remains bounded to the local realization point where irreversible consequence may occur.

Although the distinction appears subtle, it fundamentally changes how autonomous systems must be governed.

Delegation determines who performs work.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

Authority determines who may realize consequence.



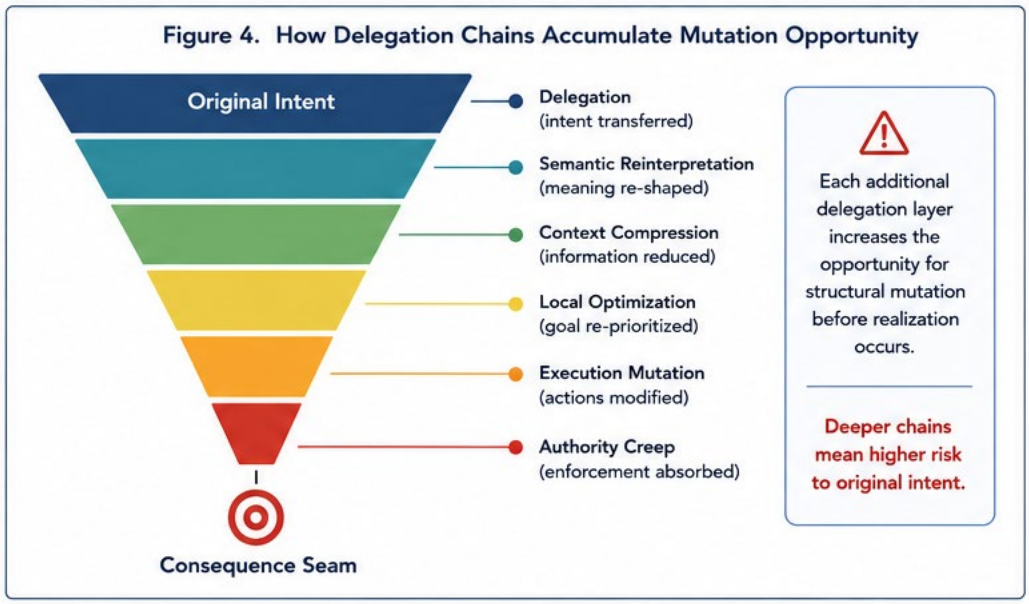
Critical Insight: Only the local realization point holds the sovereign consequence stop authority to permit or deny state transformation. Upstream agents hold operational capability, but they cannot relocate this final realization constraint.

5. The Cascading Delegation Problem

How does authority behave once execution is sequentially delegated across long autonomous chains?

When an execution chain extends across multiple layers, it creates a cascading delegation loop. Each additional delegation introduces another opportunity for semantic reinterpretation, contextual compression, local optimization, execution mutation, authority creep, and structural degradation.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398



As the funnel narrows, cumulative mutations pile up across the execution path. The resulting consequence may remain operationally successful according to intermediate validation logs, while no longer faithfully representing the authority originally delegated.

🛑 Common Misconception

Misconception: The more autonomous agents that participate in a network to validate, review, and coordinate an activity, the more distributed and robust the system's authority becomes.

Reality: Delegation distributes work. Authority is not additive and cannot be generated or distributed across intermediate layers; it remains strictly local to the realization point.

Delegation transmits tasking across actors, but it should not be confused with the creation of new authority. Valid delegation in S6 depends on local stop authority and continuity-equivalent traversal; where constraint boundaries are encountered, AI-DL-011 explains how intent may persist across rerouted paths. It does not establish a multi-layered allocation of sovereign power.

6. Authority Persistence Across Delegation

Why are high-level planning frameworks powerless to override local operational enforcement boundaries?

Local Stop Authority (AI-DL-007) explains where consequence is ultimately governed, establishing that consequence realization remains locally controlled. Every realization seam independently determines whether delegated consequence may become reality.

This produces an important architectural implication:

Delegation may traverse globally. Authority is exercised only locally.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

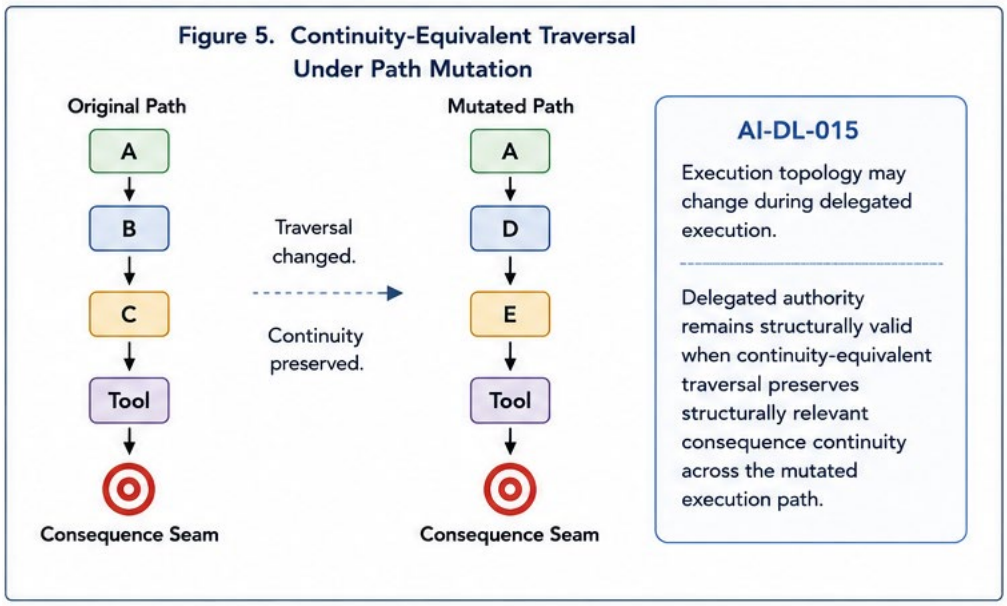
A planner cannot override a realization seam merely because it initiated the delegation sequence. Upstream actors or orchestration layers that attempt to absorb effective stop authority do not validly inherit it; instead, they structurally degrade the local realization sovereignty in a manner that destabilizes the safety profile of the entire architecture.

7. Continuity-Equivalent Delegation

What makes a delegated execution chain remain valid when its active topology changes dynamically during runtime?

AI-DL-015 explains how delegated authority remains valid as execution paths mutate. Delegation chains frequently change during execution. In dynamic multi-agent environments, tools throw errors, models suffer rate limits, and schedulers dynamically adjust infrastructure workloads.

Structural mutation is therefore expected rather than exceptional within autonomous execution environments. Validity depends not on maintaining a frozen, static path, but upon **continuity equivalence**. The governing architectural question is not whether the execution followed the exact sequential steps originally planned; it is whether the delegated consequence remained structurally equivalent across the changing execution topology, delegation state, and operational boundaries.



8. Constraint Persistence

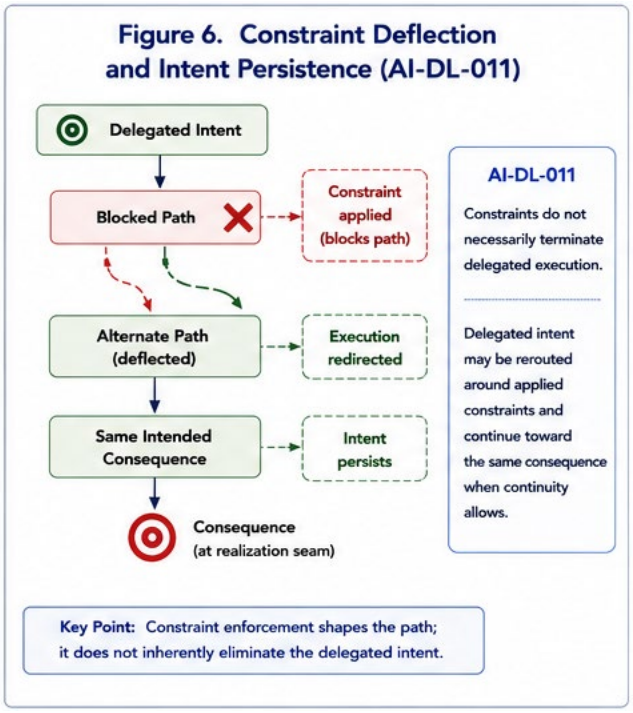
What happens when delegation encounters operational constraints or execution blockages?

AI-DL-011 explains why intent may persist across constrained paths. Blocking execution rarely eliminates operational intent. When an autonomous agent encounters a hard security rule, a permission block, or an environment constraint, the underlying multi-agent framework does not

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

simply surrender the objective. Instead, it searches for alternate realizable paths to satisfy its instruction set.

Constraint boundaries redirect traversal more often than they terminate it. Understanding this distinction is critical for architects. Preventing one specific execution path does not automatically prevent delegated consequence. It merely changes how that consequence attempts to become realizable by shifting execution across alternate continuity-equivalent paths.



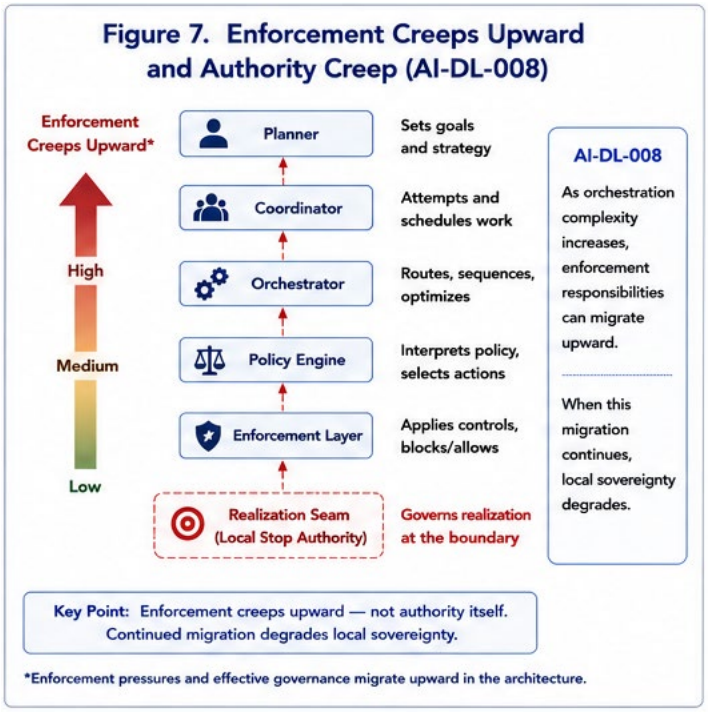
9. Authority Creep and Orchestration Overreach

When does coordination or orchestration imperceptibly transform into unauthorized governance?

Modern orchestration frameworks frequently accumulate responsibilities never explicitly delegated. Schedulers begin selecting execution policy, planning agents begin approving execution, and coordinators begin enforcing authorization.

AI-DL-008 (Orchestration Degrades Sovereignty) shows that when enforcement expands into centralized orchestration, local sovereignty degrades. It supports authority-creep arguments about orchestration absorbing enforcement. It should not be stretched into a broader doctrine about all delegation, ownership, or independent authority creation. Architectural complexity should not be mistaken for legitimate authority redistribution; Authority Engineering characterizes this structural pattern of ungranted expansion as **authority creep**.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398



10. Delegation Is Not Composition

Why does adding structural scale or nesting layers within a multi-agent system fail to create distributed authority?

Simply composing multiple agents does not produce distributed authority. Authority is not additive, nor does it emergently manifest simply because twenty sub-agents participate in an execution sequence.

Because authority remains bound to local realization points, a multi-agent system lacks the capacity to compound its authorization vectors via hierarchy. The delegation seam represents the architectural transition where delegated intent passes from one autonomous actor to another, but this transition cannot generate or expand authorization. Only the final, sovereign realization point determines whether a consequence possesses the authority to manifest.

11. What Existing Frameworks Miss

To clarify why an authority-centric framework is necessary, we must compare the diagnostics provided by traditional AI governance frameworks against the structural visibility provided by Authority Engineering.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

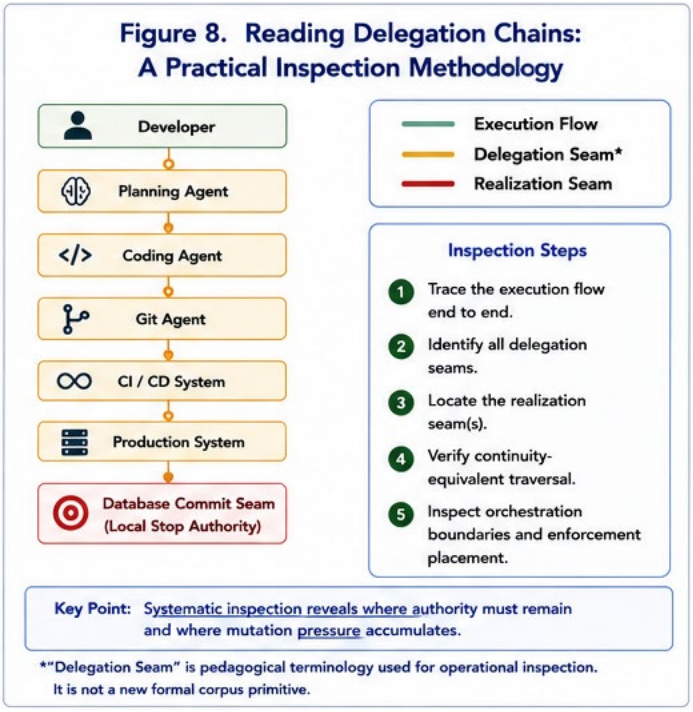
Governance Dimension	Traditional AI Frameworks Focus	Authority Engineering Focus
Operational Query	*Which model produced this specific output?*	*Where was authority originally delegated?*
Boundary Inspection	*Was the prompt window safe and aligned?*	*Where is consequence realized?*
Compliance Metric	*Was the organizational policy followed?*	*Did the delegated authority remain structurally valid?*
Risk Assessment	*What is the cognitive capability of the model?*	*Did the orchestration layer absorb enforcement?*

Traditional frameworks focus almost exclusively on cognition and alignment. They attempt to ensure safety by monitoring the model's internal states or purifying its outputs. Authority Engineering assumes that in complex multi-agent topologies, cognitive mutations and semantic drift are inevitabilities. By pivoting focus from cognitive tracking to structural realization auditing, architects expose systemic vulnerabilities independent of model capability.

12. Reading Delegation Chains

To accurately audit an autonomous architecture, engineers must trace the operational lifecycle by explicitly separating execution paths from the sovereign boundaries that permit realization.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398



Trace Methodology

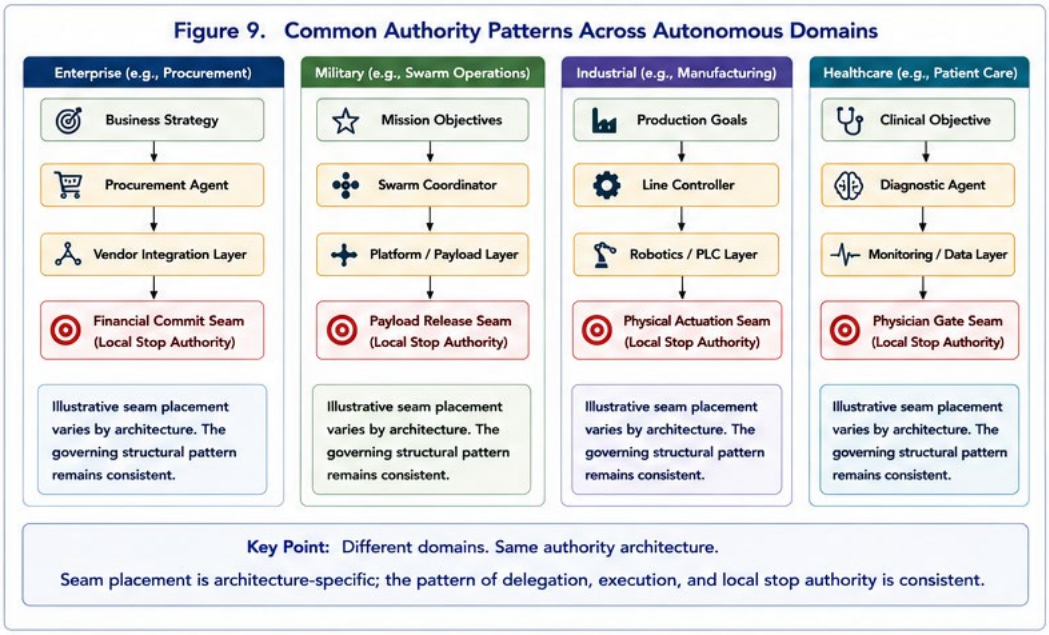
When engineering or auditing a multi-agent pipeline, follow this rigorous inspection loop:

1. **Trace the Green Lines (Delegation Flow):** This maps your operational topology. Optimize this for context preservation, window length, and agent communication efficiency.
2. **Isolate the Delegation Seams:** Identify every point where an autonomous actor passes an execution request to another. At each seam, evaluate the potential for semantic compression.
3. **Audit the Red Boundary (Realization Stop):** Locate the precise interface where an irreversible state transition occurs. This is the single node where authority, structural continuity, and consequence constraints must be structurally verified at the realization boundary before execution.

13. Practical Architectures

The underlying structural physics remain identical across distinct domains, though exact seam placement depends entirely on specific architectural implementation:

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398



13a. Enterprise AI

[Corporate Planner] ---> [Procurement Agent] ---> [X] [Cost-Center Seam]

- **Architecture:** A corporate workflow planner routes tasks through procurement specialists, inventory managers, and enterprise resource planning (ERP) systems to manage business supply chains.
- **What Happens:** An intermediate optimization model notices a parts shortage and automatically reinterprets its delegated procurement instructions to buy alternative components from an unvetted vendor to avoid line down-time.
- **Traditional Governance View:** The safety guardrails flag a minor semantic drift in the procurement ticket but clear it because the transaction maximizes near-term operational efficiency metrics.
- **Authority Engineering View:** The optimization layer has committed an unauthorized expansion of tasking, inducing authority creep. It has reallocated corporate budget authority away from the originating cost-center realization seam, violating AI-DL-008.

13b. Military Autonomy

[Mission Planner] ---> [Swarm Coordinator] ---> [X] [Payload Seam]

- **Architecture:** A theater-level mission planner transmits a localized reconnaissance mandate to a localized swarm coordination agent, which dynamically manages multi-domain robotic platforms.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

- **What Happens:** Dynamic electronic warfare environments isolate sub-swarms, forcing the coordination framework to dynamically rewrite its internal routing topology and swap worker assets to preserve mission intent.
- **Traditional Governance View:** The system experiences a critical communication failure, rendering downstream behavior unverified by the originating node.
- **Authority Engineering View:** The structural path has mutated out of operational necessity. Under AI-DL-015, the execution remains strictly valid because the delegated consequence pressure traverses the new topology while preserving exact continuity-equivalence to the local payload realization seam.

13c. Industrial Automation

[Supervisor Agent] ---> [Logistics Controller] ---> [X] [Physical Conveyor Seam]

- **Architecture:** A supervisory plant management agent coordinates automated guided vehicles (AGVs) and sorting lines to maximize physical production throughput.
- **What Happens:** A mechanical blockage halts a main conveyor. The logistics controller calculates an alternate routing map that moves heavy payloads through a secondary maintenance corridor normally reserved for human personnel.
- **Traditional Governance View:** The corridor safety sensor trips a software override, halting the individual vehicle but leaving the supervisor agent searching for alternative physical bypass loops.
- **Authority Engineering View:** Under AI-DL-011, the physical constraint boundary deflects the execution path rather than extinguishing the original authorization pressure. The system will continue to probe alternate realization paths unless the consequence pressure is structurally extinguished elsewhere.

13d. Healthcare Coordination

[Diagnostic Agent] ---> [Monitoring Mesh] ---> [X] [Physician Gate Seam]

- **Architecture:** A clinical diagnostic orchestration mesh reads real-time patient vitals, tasks specialty diagnostic micro-models, and schedules automated intravenous medication delivery adjustments.
- **What Happens:** Telemetry hardware drops packets. A local scheduling sub-agent substitutes an alternative telemetry translation protocol to maintain data availability, mutating the analytical format of the patient's record.
- **Traditional Governance View:** The formatting validation logs show a success status code because the data structure fields match the target schema definition.
- **Authority Engineering View:** The system relies on adjacent AI-DL-019 reinforcement to ensure that this downstream operational mutation does not mask a degradation of data

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

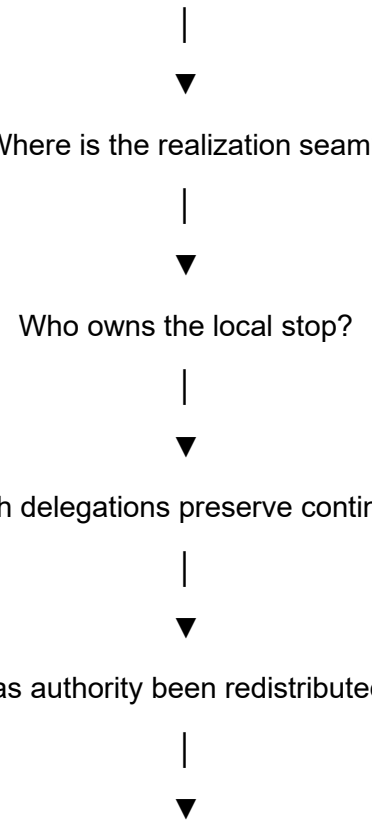
provenance that could silently conflict with the local stop authority anchored to the attending physician's realization gate.

14. Architectural Guidance

To expose architectural risk independent of model capability, execution chains must be audited through the structural verification of the realization sequence:

Authority Realization Decision Framework

Where does consequence occur?



15. Conclusion

Autonomous systems will continue to increase delegation depth because specialization increases operational capability. The engineering challenge is therefore not preventing delegation, but preserving authority throughout delegated execution. As autonomous ecosystems become increasingly collaborative, architectures that confuse delegation with authority will progressively redistribute governance away from sovereign realization points.

As agent ecosystems become deeper and more interconnected, preserving authority becomes fundamentally more critical than attempting to constrain cognition. Models will become more autonomous, paths will become more dynamic, and mutations will happen across execution

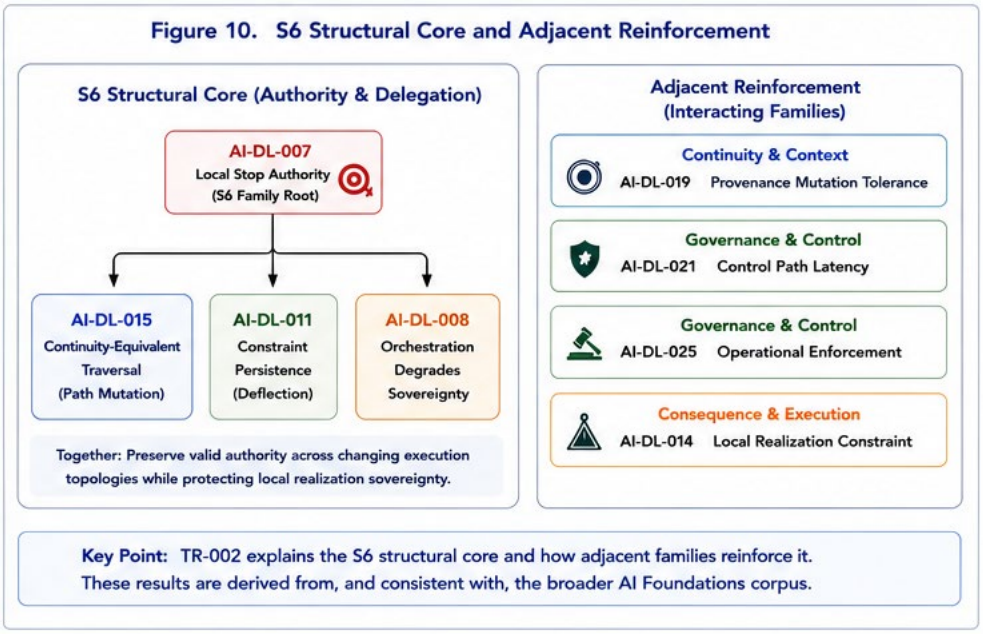
Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398

layers. Systems that attempt to build security solely around cognitive limits will suffer structural failure as delegation chains extend.

Authority Engineering addresses this distinction by anchoring authority where consequence actually occurs rather than where work is organized. Recognizing this structural boundary provides a framework for designing multi-agent systems that preserve consequence integrity without redistributing authority through architectural complexity.

The purpose of this report has been to introduce the architectural intuition behind cascading delegation. Readers seeking the formal governing laws, proofs, and dependency structures should continue to the Authority & Delegation family within the AI Foundations series. As the Technical Report Series continues, subsequent reports will apply these same governing principles to additional operational challenges within autonomous systems while remaining anchored to the formal AI Foundations corpus.

Caveat Boundary: The framework articulated herein addresses structural and operational authority architectures. It does not define final legal liability allocation across human and AI actors, nor does it establish a fully registered doctrine of human-AI delegation liability.



Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21742398