Authority Engineering

A Structural Discipline for Consequential Authority in Complex Systems

Rob Caswell

Independent Researcher
Authority Architecture / Cybersecurity Governance
March 2026

Published by Cyber Terrain LLC | cyberterrain.ai

DOI: https://doi.org/10.5281/zenodo.21741292

CYBER TERRAIN

Publication Record

Authority Engineering: A Structural Discipline for Consequential Authority in Complex Systems

Author: Rob Caswell

Publisher: Cyber Terrain LLC

Published: 2026-03

Zenodo version: 1.0

DOI: https://doi.org/10.5281/zenodo.21741292

Website: https://cyberterrain.ai/papers/authority-engineering.pdf

Corpus Context

Corpus: Authority Engineering

Document ID: AE-CORE-0001

Object family: AE-CORE-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

Modern automated systems increasingly perform actions that directly alter economic, legal, and physical reality. Financial systems execute value transfers, identity platforms bind or mutate identity records, AI agents generate policy decisions, and cyber-physical systems actuate infrastructure. These actions represent consequential state transitions rather than merely informational operations.

Traditional cybersecurity frameworks were developed primarily to address system compromise, unauthorized access, and vulnerability exploitation. While these models remain essential, they do not adequately capture a structural property of modern systems: the authority to produce consequential outcomes.

This paper introduces Authority Engineering, a discipline concerned with the analysis, design, and governance of consequential authority in complex systems. Authority Engineering examines the structural topology of authority embedded in systems, the dynamics governing authority expansion and constraint, and the architectural mechanisms required to safely govern automated authority.

The framework introduces several core constructs, including Authority Outcome Primitives, Authority Surface topology, Authority Surface dynamics, and Authority Phase Transitions. Together these constructs describe how authority behaves in automated environments. The paper further introduces structural laws governing authority systems and describes the role of Authority Architecture in implementing governance boundaries.

Finally, the paper situates Authority Engineering within the broader landscape of AI governance and distributed system design and briefly introduces the Authority Surface Index (ASI) as a quantitative measurement framework derived from the discipline.

Keywords

Authority Engineering
AI Governance
Cybersecurity Architecture
Automated Systems
Authority Surface
System Governance
Distributed Systems Security

Rob Caswell | @rob-caswell3

DOI: https://doi.org/10.5281/zenodo.21741292

1 Introduction — The Authority Problem in Modern Systems

Modern computing systems increasingly perform actions that directly alter economic, legal, and physical reality. Financial platforms execute value transfers, identity systems bind or mutate identity records, automated policy systems influence governance decisions, and cyber-physical systems actuate infrastructure.

These actions are not merely informational operations; they produce consequential state transitions.

Traditional cybersecurity frameworks were designed primarily to address system compromise and unauthorized access. They evaluate vulnerabilities, attack paths, authentication mechanisms, and system integrity. These models focus on exposure and likelihood.

While such frameworks remain foundational, they are not sufficient to describe a critical property of modern automated systems: the authority to produce consequential outcomes.

Automation, system integration, and artificial intelligence have significantly expanded the number of systems capable of performing consequential actions without direct human intervention. These systems interact with financial services, regulatory frameworks, identity infrastructure, and physical control environments.

As a result, the central risk question is no longer limited to whether a system can be compromised. The more fundamental question becomes:

What consequential state transitions are systems structurally capable of performing?

This paper introduces Authority Engineering, a structural discipline for analyzing, designing, and governing consequential authority in complex socio-technical systems.

The conceptual hierarchy of the discipline is summarized as:

Authority Engineering

↓

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

Authority Architecture

↓

Authority Surface

↓

Authority Surface Index

Authority Engineering studies how consequential authority behaves in complex systems. Authority Architecture concerns the design of governance boundaries and decision rights. Authority Surface describes the structural possibility space of consequential transitions. The Authority Surface Index provides a quantitative measurement framework for evaluating authority magnitude.

1.1 Core Definitions

Definition 1 (Authority).
Authority is the structural capacity of an actor or system to produce consequential state transitions affecting economic, legal, physical, computational, or governance domains.

Definition 2 (Authority Engineering).
Authority Engineering is the discipline concerned with the analysis, design, and governance of consequential authority in complex systems.

Definition 3 (Authority Surface).
The Authority Surface of an actor is the set of consequential state transitions the actor is structurally capable of producing.

Definition 4 (Authority Architecture).
Authority Architecture is the design of decision rights, termination authority, escalation paths, and governance boundaries governing consequential authority.

1.2 Related Work

Several adjacent research areas address aspects of authority and consequence in distributed systems.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

Cybersecurity research traditionally focuses on attack surfaces, vulnerability exploitation, and access control mechanisms. These models evaluate the likelihood of compromise but do not explicitly measure the consequential authority embedded in systems.

Distributed systems literature addresses capability security, object-capability models, and privilege delegation, which partially relate to authority relationships. However, these frameworks focus primarily on permission structures rather than the topology of consequential outcomes.

Safety engineering and cyber-physical systems research address hazard analysis, control system safety, and fail-safe architectures, which analyze consequences of system behavior but typically do not frame the problem in terms of authority structures.

AI governance research increasingly addresses automated decision systems, but most approaches focus on alignment, transparency, and ethics rather than structural authority.

Authority Engineering builds upon these domains but focuses specifically on the structural capacity of systems to produce consequential outcomes.

2 Authority vs Access

Conventional cybersecurity models emphasize access control. Authentication and authorization mechanisms determine who may interact with system resources.

However, access and authority are distinct concepts.

Access describes the ability to observe or interact with system resources. Authority describes the ability to produce consequential state transitions.

A user may have access to a financial database but lack authority to transfer funds. Conversely, an automated payment service may possess authority to execute transfers while operating with limited direct access to internal infrastructure.

Traditional security metrics focus on reachability and compromise. These models evaluate whether an adversary can gain access to a system.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

Authority Engineering instead evaluates consequence. More precisely, it evaluates the set of consequential state transitions that remain structurally reachable within a system, whether exercised by a human operator, software process, or autonomous agent.

The relevant question becomes:

What could the system change if authority were exercised?

This distinction becomes especially important in automated environments. AI agents, orchestration systems, and autonomous workflows frequently operate with authority to execute consequential operations across multiple domains.

Security models centered on compromise do not fully capture the magnitude of risk embedded in these systems.

Authority Engineering therefore examines the topology of authority rather than solely the mechanisms controlling access.

3 Authority Outcome Primitives

Authority execution ultimately produces a finite set of consequential outcomes. These outcomes can be categorized into fundamental transition classes called Authority Outcome Primitives.

These primitives represent the closure of consequential authority in socio-technical systems.

The canonical primitives are:

Table 1 — Authority Outcome Primitives

Primitive	Description
Money	Transfer or allocation of economic value
Contract	Creation or modification of legally binding obligations
Identity	Binding or mutation of identity relationships
Policy	Modification of governance rules or system behavior

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

Primitive	Description
Authority	Delegation or mutation of authority relationships
Model	Modification of computational behavior or learning state
Physical	Actuation of physical systems

This finite primitive set establishes the consequence-theoretic closure of authority analysis. Any complex execution path may be decomposed into one or more primitive outcome classes.

These primitives form the closure of consequential authority. This property is referred to as Authority Primitive Closure.

Regardless of system complexity, authority execution ultimately resolves into one or more of these primitive outcomes.

For example:

Financial transaction systems terminate in Money transitions.
Identity verification systems terminate in Identity transitions.
Governance systems terminate in Policy or Authority transitions.

By defining this primitive set, Authority Engineering provides a finite analytical basis for studying authority systems.

4 Authority Surface Topology

The Authority Surface represents the structural possibility space of consequential transitions available to an actor or system.

Formally, the authority surface of actor is defined as:

$$AS(a) = \{ \tau \in T \mid C(\tau) \wedge P(\tau) \}$$

Where:

AS(a) = authority surface of actor *a*

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

τ = a system state transition

T = the set of reachable system transitions

$C(\tau)$ = τ is consequential

$P(\tau)$ = τ resolves into one or more authority outcome primitives

The Authority Surface is therefore not a measure of permission in the narrow access-control sense. It is a structural measure of what consequential transitions are reachable through the actor's available system pathways, delegated powers, and execution mechanisms.

Authority surfaces arise from system integrations, delegated permissions, automated workflows, and governance structures.

Unlike traditional attack surfaces, authority surfaces are defined by consequence potential rather than reachability.

Two systems may exhibit identical attack surfaces but radically different authority surfaces.

A read-only analytics system exposes minimal authority despite broad accessibility. Conversely, an automated payment processor may expose substantial authority even with strict access controls.

Authority Engineering therefore studies the topology of authority surfaces and the pathways through which authority transitions propagate across systems.

5 Authority Surface Dynamics

Authority surfaces evolve over time as systems integrate new services, automate decision processes, and delegate execution authority to software agents.

Authority Engineering models this evolution using the Authority Surface Balance Law:

Where expansion forces include:

$$\Delta AS = (E + D + S + A) - (T + O + R + B)$$

ΔAS = change in authority surface magnitude

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

Execution Expansion (E) — addition of automated execution capabilities

Delegation Expansion (D) — transfer or propagation of authority relationships

State-Space Expansion (S) — growth in the number of mutable system states

Autonomy Expansion (A) — increasing independence of automated decision processes

Contraction forces include:

Termination Authority (T) — mechanisms capable of halting authority execution

Oversight (O) — monitoring and verification processes

Risk Bound Governance (R) — formal governance constraints

Boundary Constraints (B) — structural limits on system integration or authority propagation

The Authority Surface Balance Law is not intended as a fully reduced physical equation, but as a structural accounting model for authority expansion and contraction. Its purpose is to make explicit that authority magnitude changes through identifiable architectural and governance forces, rather than through permission changes alone.

Automation tends to expand authority surfaces, while governance mechanisms attempt to constrain them.

The equilibrium of these forces determines the magnitude of authority embedded within a system.

6 Authority Phase Transitions

Authority systems may experience discontinuous increases in authority magnitude when new primitives become reachable.

These events are referred to as Authority Phase Transitions.

A critical authority transition can be expressed as:

$$\Delta AS_critical = \Sigma\, w(\tau_i)$$

Where:

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

ΔAS_critical = discontinuous increase in authority surface magnitude

τi = newly reachable authority transition

w(τi) = weighted consequence magnitude of that transition

Authority phase transitions frequently occur when:

automation unlocks new execution capabilities

system integrations introduce new primitives

AI agents gain access to operational tools

governance systems gain authority to modify policy

These transitions can significantly increase authority magnitude even when no security vulnerabilities are introduced. This is a central distinction between Authority Engineering and traditional security analysis. Security failure is not required for authority magnitude to change discontinuously. Structural reconfiguration alone may be sufficient.

7 Laws of Authority Engineering

Authority Engineering introduces several structural principles governing authority systems.

Consequence Proximity Law.

Governance controls must exist at consequence execution points.

Primitive Convergence Law.

All authority execution paths converge on a finite set of outcome primitives.

Boundary Integrity Law.

Transitions between primitives must pass through explicit governance boundaries.

Unified Semantics / Distributed Enforcement Law.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

Authority meaning must remain centrally defined while enforcement occurs across distributed systems.

Runtime Adaptation Law.

Authority flows must be dynamically regulated in automated environments.

The laws described above define the structural behavior of authority systems. The next section examines the design discipline responsible for implementing these principles within real-world systems.

8 Authority Architecture

Authority Architecture is the design discipline responsible for implementing governance structures within authority systems.

It defines:

decision rights

escalation mechanisms

termination authority

delegation structures

oversight processes

Authority Architecture translates the theoretical principles of Authority Engineering into operational system design. If Authority Engineering is the discipline governing how consequential authority behaves, Authority Architecture is the applied design practice through which that authority is allocated, bounded, escalated, and terminated in real systems.

9 Authority Measurement

Practical governance requires quantifying the magnitude of authority embedded within systems.

The Authority Surface Index (ASI) provides a quantitative measurement framework derived from Authority Engineering.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

ASI evaluates four structural components:

Authority Execution Score

Constraint Symmetry Score

Concentration Risk Index

Termination Latency Factor

Together these components estimate the structural magnitude of authority embedded in a system. In this sense, ASI should be understood as a measurement framework derived from the discipline, not as a substitute for the underlying structural theory.

The ASI framework is described in detail in a companion paper.

10 Authority Engineering in AI Systems

Artificial intelligence systems significantly expand authority surfaces because they combine automation, integration, and autonomous decision-making.

AI agents frequently interact with multiple services and APIs, execute actions autonomously, and operate at machine speed.

Traditional security models do not adequately capture these dynamics because they focus on compromise rather than authority topology.

AI systems intensify the authority problem because they combine broad integration, dynamic pathway discovery, and machine-speed execution in a single operating actor. As a result, authority must be analyzed as a structural property of system design rather than as a secondary consequence of access permissions alone.

Authority Engineering provides a structural framework for governing automated authority.

11 Implications for Governance and System Design

The rise of automated systems requires a shift in governance models.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

Organizations must evaluate systems not only for compromise risk but also for the magnitude of authority they embed.

Key governance implications include:

analyzing system authority surfaces during design

implementing governance controls at primitive execution points

establishing explicit termination authority for automated systems

continuously monitoring authority expansion as systems evolve

The practical implication is that governance must move closer to execution. System design must explicitly identify consequential primitives, define escalation boundaries between them, assign termination authority, and monitor authority expansion as integrations and automation evolve. In this respect, Authority Engineering reframes governance as a structural design problem rather than a purely procedural or compliance problem.

12 Conclusion

Modern computing systems increasingly operate with the authority to produce consequential outcomes across economic, legal, and physical domains.

Traditional cybersecurity frameworks focus primarily on preventing compromise and controlling access. While these protections remain essential, they do not fully address the structural risks created by automated authority.

Authority Engineering provides an initial foundation for reasoning about consequential authority in complex systems. As automated systems increasingly participate in economic, legal, governance, and physical domains, the structural capacity to produce consequential state transitions becomes a primary object of analysis. The discipline proposed here is intended as a starting framework: one that defines the problem, establishes core constructs, and creates a basis for future work in authority measurement, architecture, governance, and system control.

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

13 Future Research Directions

Authority Engineering is introduced in this paper as an initial framework for analyzing and governing consequential authority in complex automated systems. The concepts presented here establish foundational constructs, but several areas require further development in order for the discipline to mature into a complete analytical and engineering framework.

First, formal mathematical models of authority systems require further elaboration. While the paper introduces structural laws governing authority behavior, including the Authority Surface Balance Law and Authority Phase Transitions, additional work is needed to formalize authority topology using graph and network models. Such models may enable rigorous analysis of authority propagation, concentration, and containment within large distributed systems.

Second, quantitative measurement techniques must be refined. The Authority Surface Index (ASI) is introduced as a preliminary measurement framework, but additional research is required to validate the weighting schemes, measurement inputs, and empirical calibration of authority magnitude across different system architectures. Future work should examine how authority surfaces evolve in real-world environments such as financial infrastructure, identity platforms, and AI agent ecosystems.

Third, architectural design patterns for governing automated authority require further exploration. Authority Architecture defines the governance structures responsible for allocating, constraining, and terminating authority within systems. However, practical implementation patterns for authority boundaries, termination mechanisms, and escalation structures remain an open area for system design research. In particular, the emergence of AI agents and autonomous workflows introduces new requirements for machine-speed governance controls.

Fourth, the dynamics of authority in artificial intelligence systems present a particularly important research domain. AI agents operating across integrated service environments may dynamically discover execution pathways that expand their reachable authority surfaces. Understanding how authority surfaces expand, compress, or reorganize under autonomous system behavior will be critical to governing future automated infrastructures.

Finally, empirical case studies will be necessary to validate the theoretical constructs introduced in this paper. Observational analysis of authority surfaces in real-world systems, including financial transaction platforms, identity

Published by Cyber Terrain LLC | cyberterrain.ai DOI: https://doi.org/10.5281/zenodo.21741292

infrastructures, cyber-physical control environments, and AI agent frameworks, will help refine the discipline and test its practical applicability.

Authority Engineering is therefore best understood as the starting point of a broader research program. As automated systems increasingly participate in consequential economic, legal, and physical domains, the structural capacity of those systems to produce real-world outcomes will become a primary object of analysis. Developing the analytical tools, architectural patterns, and governance mechanisms required to safely manage that authority will be a central challenge for future system design.

References

Saltzer, J. H., & Schroeder, M. D. (1975).
The protection of information in computer systems.

Denning, D. E. (1976).
A lattice model of secure information flow.

Lampson, B. W. (1974).
Protection.

Leveson, N. (2011).
Engineering a Safer World: Systems Thinking Applied to Safety.

NIST. (2024).
AI Risk Management Framework.

Cloud Security Alliance. (2024).
AI Controls Matrix.

Published by Cyber Terrain LLC | cyberterrain.ai

DOI: https://doi.org/10.5281/zenodo.21741292