

AUTHORITY ARCHITECTURE

Designing Authority in Complex Organizational Systems

Rob Caswell

Framework Paper

March 2026

Authority Architecture © Rob Caswell

All rights reserved.

Publication Record

Authority Architecture: Designing Authority in Complex Organizational Systems

Author: Rob Caswell

Publisher: Cyber Terrain LLC

Published: 2026-03-01

Zenodo version: v1.0

DOI: <https://doi.org/10.5281/zenodo.21742255>

Website: <https://cyberterrain.ai/papers/authority-architecture.pdf>

Corpus Context

Corpus: Authority Engineering

Document ID: AE-ARCH-0001

Object family: AE-ARCH-0001

Publisher-imprint edition prepared for versioned republication.

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

Table of Contents

Executive Summary

- 1. The Authority Gap**
 - 2. Definition of Authority**
 - 3. Authority Architecture**
 - 4. Authority Surface**
 - 5. Authority Topology**
 - 6. Authority Primitives**
 - 7. Authority Mapping**
 - 8. Authority Entropy**
 - 9. Failure Modes**
 - 10. Applied Architectures**
 - 10.1 CISO Authority Architecture**
 - 10.2 Authority in Autonomous Systems**
 - 10.3 Authority in AI Systems**
 - 11. Conclusion**
-

List of Figures

- Figure 1 — Authority Architecture Model**
- Figure 2 — Authority Surface vs Authority Execution**
- Figure 3 — Authority Mapping Exercise**
- Figure 4 — Authority Failure Modes**
- Figure 5 — Authority Architecture and AI Execution**

Appendices

- Appendix A — Authority Mapping Exercise**
- Appendix B — Authority Primitive Reference**
- Appendix C — Authority Architecture Diagnostics**

Executive Summary

Organizations routinely design the structures that support their operations. Technical architectures govern how systems are built and operated. Financial architectures determine how resources are allocated and controlled. Risk management frameworks provide methods for identifying and managing uncertainty.

Authority, however, is rarely designed with the same structural rigor.

In many organizations authority is assumed to emerge naturally from hierarchy, titles, or reporting relationships. When authority structures function poorly, the resulting problems are often attributed to leadership failures, communication breakdowns, or organizational culture. In many cases the underlying issue is structural: the organization has never deliberately designed how authority operates within the system.

This paper introduces Authority Architecture, a framework for analyzing and designing the structure of authority within bounded systems. Authority is defined as the legitimate capacity to cause a bounded system to change state. From this definition the framework establishes several structural concepts used to examine how authority operates within complex organizations.

The authority surface identifies the locations within a system where decisions can produce consequential change. Authority topology describes how authority is distributed across those locations. Authority primitives define the structural roles required for authority to function within the system. The process of authority mapping provides a method for examining how these primitives are placed across the authority surface.

Because organizations evolve over time, authority structures also change. The concept of authority entropy describes the tendency for authority architectures to drift away from their original design as organizations grow, adopt new technologies, and adapt to operational pressures. When this drift occurs, structural failure modes may appear in the form of authority absence, authority collision, and authority drift.

The framework is then applied to several real-world contexts, including cybersecurity leadership structures, autonomous operational systems, and artificial intelligence environments in which algorithmic systems influence consequential decisions.

Authority Architecture provides a structural method for examining how decision authority operates within complex systems. By making authority visible as an architectural element of organizational design, the framework allows organizations to identify structural weaknesses and design governance systems that remain coherent as technology, automation, and artificial intelligence increasingly shape how decisions occur.

1. The Authority Gap

Modern organizations operate as complex systems that must produce outcomes under conditions of uncertainty, scale, and increasing operational speed. Within these systems, decisions must be made continuously by individuals, teams, and increasingly by automated mechanisms. Each decision carries the potential to change the state of the organization through operational, financial, regulatory, or technological consequences.

For these systems to function coherently, the ability to make consequential decisions must exist within a structure that recognizes who may legitimately act. That structure is rarely designed explicitly.

Instead, organizations commonly assign responsibility and assume that the authority required to fulfill that responsibility will exist somewhere within the organizational structure. Over time, this assumption produces gaps between those who are accountable for outcomes and those who actually possess the authority necessary to produce them. These gaps are not merely administrative inconveniences; they introduce structural instability into the governance of the organization.

When authority is unclear, missing, or fragmented, several recognizable patterns appear:

- Decisions stall because no actor possesses legitimate authority to act.
- Multiple actors assert authority over the same decision space, producing conflict or veto cycles.
- Authority gradually migrates away from formal governance structures into informal networks, operational teams, or technical systems.

Organizations often interpret these conditions as leadership failures, communication breakdowns, or cultural weaknesses. In many cases, the underlying problem is structural rather than behavioral. The organization has not deliberately designed how authority operates within the system.

This gap between responsibility and authority is becoming more visible as organizations rely increasingly on complex technological infrastructures. Automated processes, distributed platforms, and algorithmic systems now initiate actions that can produce significant operational, financial, or regulatory consequences. These systems frequently operate at speeds that exceed traditional human governance cycles, forcing organizations to confront the question of where legitimate decision authority actually resides.

In this environment, the question is no longer limited to who is responsible for outcomes. A more fundamental question emerges: where within the system does legitimate authority exist to initiate consequential change?

Organizations have developed mature disciplines for designing technology architectures, financial controls, and risk management frameworks. Yet the architecture of authority itself has historically remained implicit. Roles, reporting relationships, and governance committees are often assumed to represent authority structures, even when the ability to produce consequential decisions exists elsewhere.

This paper introduces **Authority Architecture**, a framework for understanding how authority operates within organizations and how it can be intentionally designed. Authority Architecture examines where authority exists within a system, how it is distributed across organizational structures, how decisions translate into consequential system change, and how authority structures degrade over time if they are not actively maintained.

By examining authority as an element of organizational architecture rather than as a characteristic of individuals or roles, structural weaknesses that remain hidden within conventional governance models become easier to identify and correct. The purpose of this paper is to define the foundational elements of Authority Architecture and to provide a structured approach for analyzing how authority functions in modern organizations.

2. Definition of Authority

Any discussion of authority must begin with a clear definition. In many organizations the term is used interchangeably with influence, responsibility, or accountability. While these concepts are related, they are not equivalent. Treating them as interchangeable often obscures how decisions actually occur within complex systems.

For the purposes of this framework, authority is defined as follows:

Authority is the legitimate capacity to cause a bounded system to change state.

This definition establishes several important characteristics that distinguish authority from other forms of organizational power. They are:

- **Capacity**
- **Legitimacy**
- **Bounded system**
- **State change**

Capacity refers to the practical ability to produce a decision that can alter the state of the system. Without the ability to produce such change, authority exists only in a symbolic or advisory form.

Legitimacy refers to the recognized validity of that capacity within the governing structure of the organization. Legitimacy may originate from law, governance structures, formal delegation, or established institutional design. Capacity without legitimacy produces power, but it does not constitute authority within a governed system.

A bounded system defines the domain within which authority operates. Every organization functions within operational, legal, and structural boundaries. Authority therefore cannot exist in the abstract; it is always exercised within a domain whose limits determine where decisions can legitimately occur.

State change describes the practical effect of authority. Decisions become meaningful when they alter some aspect of the system's condition. These changes may include operational actions, financial commitments, policy decisions, technical configurations, or the acceptance of risk. In each case, authority represents the recognized ability to initiate a change that affects how the system functions.

Distinguishing authority from related concepts clarifies why governance structures often fail when these concepts are conflated. Responsibility describes the obligation to perform work or achieve outcomes. Accountability describes the requirement to answer for those outcomes. Influence refers to the ability to persuade others to act. Authority, by contrast, describes the legitimate ability to initiate decisions that produce consequential system change.

When these elements are aligned, organizations are able to execute decisions coherently. When they diverge, responsibility may exist without the authority required to act, or authority may be exercised by actors who are not accountable for the resulting consequences.

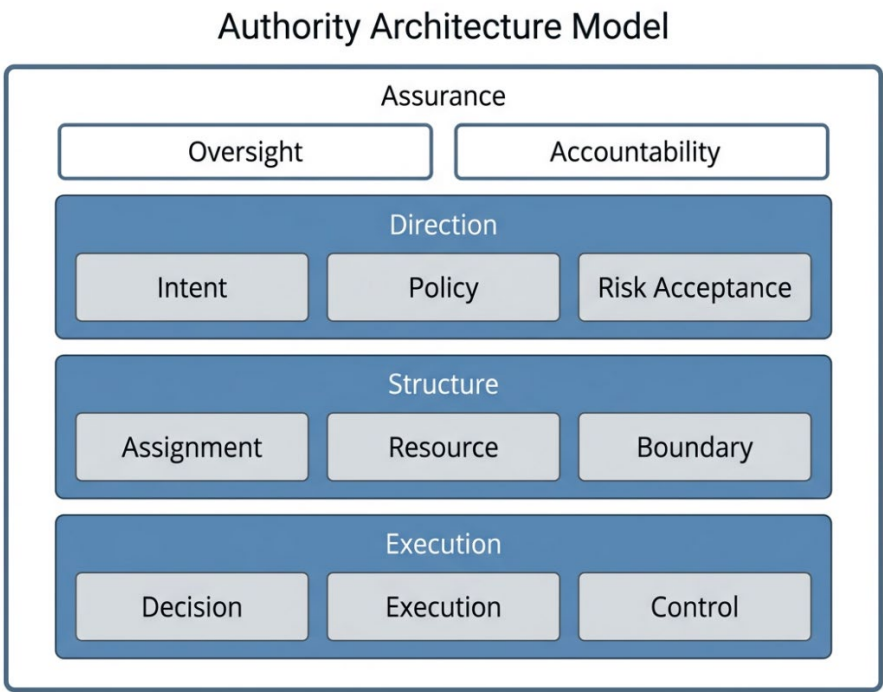
Understanding authority in this structural sense provides the foundation for examining how authority operates within complex organizations. Once authority is defined in terms of legitimate capacity to produce system change, it becomes possible to analyze where authority exists, how it is distributed across organizational structures, and how authority relationships evolve as organizations grow, automate, and operate at increasing speed.

3. Authority Architecture

Once authority is defined in structural terms, the next question is how authority should be organized within a functioning system. Organizations routinely design technical

architectures, financial control structures, and operational processes with deliberate intent. Authority, however, is often treated differently. It is assumed to emerge naturally from titles, reporting relationships, or institutional norms rather than from explicit structural design.

Authority Architecture refers to the intentional design of how authority operates within a bounded system. It concerns the placement, distribution, and coordination of legitimate decision capacity so that consequential changes to the system occur in a coherent and governed manner. The core domains of Authority Architecture are illustrated in Figure 1.



In practice, most organizations possess authority structures whether or not they have designed them consciously. Authority inevitably exists wherever decisions can alter the state of the system. The absence of deliberate design does not eliminate authority; it simply allows authority to develop through informal pathways. Over time these informal arrangements may diverge from the organization’s formal governance structures, producing gaps between where authority is assumed to exist and where it actually operates.

Authority Architecture addresses this problem by treating authority as an element of organizational structure that can be analyzed and designed with the same rigor applied to other architectural domains. This perspective shifts attention away from individual

personalities or leadership styles and toward the underlying structure through which decisions are legitimized and executed.

Several characteristics distinguish Authority Architecture from traditional governance models. Governance frameworks typically focus on roles, responsibilities, reporting lines, and oversight mechanisms. While these elements remain important, they do not necessarily reveal how authority is actually exercised within the system. Authority Architecture instead focuses on the structural conditions that determine whether a decision can legitimately produce a consequential change.

In this sense, Authority Architecture concerns itself with questions such as:

- Where within the organization can decisions alter the state of the system?
- Which actors possess legitimate authority to initiate those decisions?
- How is authority distributed across different domains of operation?
- How are authority structures maintained as organizations evolve?

These questions move the analysis of authority away from assumptions embedded in organizational charts and toward a structural understanding of how decision power operates.

Organizations that fail to design authority architectures often encounter recurring governance problems. Responsibility may be assigned without the corresponding authority required to act. Multiple actors may claim authority over the same decision space. In other cases, authority may migrate toward operational teams, technical systems, or automated processes that were never intended to serve as decision authorities.

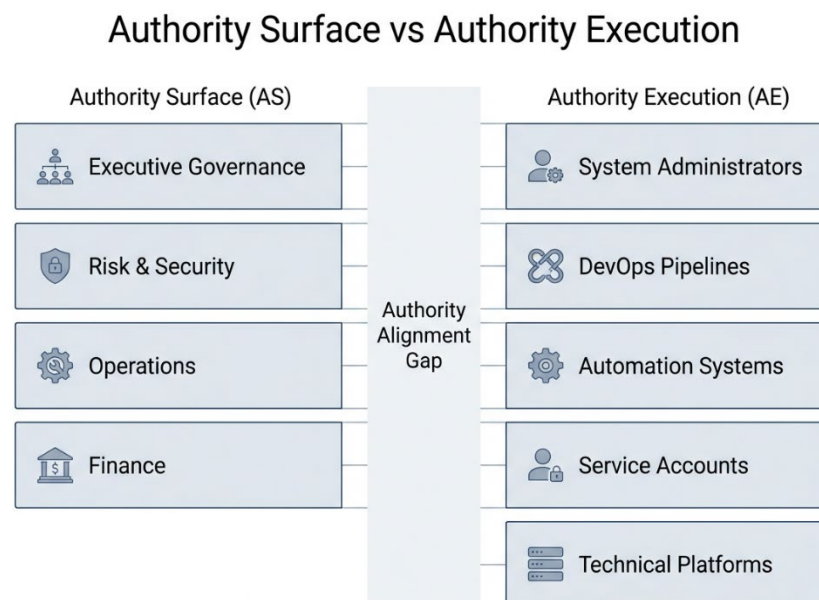
These conditions do not arise from individual misconduct or leadership weakness. They emerge from the absence of deliberate structural design.

Authority Architecture provides a framework for identifying where authority exists within a system, how it is distributed across organizational structures, and how authority relationships evolve as systems grow more complex and operate at increasing speed. By making authority visible as an architectural element of organizational design, the framework enables organizations to diagnose structural weaknesses and design governance systems that align responsibility, accountability, and legitimate decision capacity.

4. Authority Surface

Once authority is defined and authority architecture is recognized as a structural design problem, the next question concerns where authority actually exists within a system. Organizations frequently assume that authority resides primarily within formal reporting structures, executive roles, or governance committees. In practice, the locations where consequential decisions occur often extend far beyond these visible structures.

The concept of the **Authority Surface** describes the set of locations within a system where decisions can legitimately produce consequential change. These locations may exist within formal governance roles, operational teams, technical systems, or automated mechanisms. Wherever a decision can alter the state of the system in a meaningful way, authority must exist in some form. The relationship between organizational authority locations and technical execution environments is illustrated in Figure 2.



The authority surface of an organization is therefore not limited to senior leadership or formally designated decision bodies. Authority appears anywhere that system state can be altered through recognized action. Examples of such locations commonly include:

- financial commitments or budget allocations
- operational decisions affecting service delivery
- policy changes or governance determinations
- configuration of technical infrastructure or software systems
- acceptance of operational, financial, or regulatory risk

Each of these decision points represents a location where authority is exercised because each carries the ability to change the condition of the system.

Understanding the authority surface of an organization requires examining the practical consequences of decisions rather than relying solely on formal reporting relationships. Organizational charts describe lines of supervision and accountability, but they often fail to reveal where consequential decisions actually occur. In many cases, operational teams, technical administrators, or automated systems possess the ability to produce system changes that carry significant organizational impact.

As organizations adopt increasingly complex technologies and automated decision mechanisms, the authority surface expands beyond traditional managerial structures. Configuration changes in software platforms, automated deployment systems, algorithmic decision engines, and distributed operational processes can all produce meaningful system changes. In these environments, authority may be exercised through technical mechanisms that operate at speeds and scales that differ significantly from conventional governance processes.

Mapping the authority surface therefore becomes an essential step in understanding how authority operates within a system. By identifying the locations where consequential system changes can occur, organizations can determine whether legitimate authority has been intentionally placed at those locations or whether authority has emerged informally through operational necessity.

This distinction is critical. When authority is deliberately designed, the actors capable of producing system change possess recognized legitimacy and clear accountability. When authority emerges informally, the organization may find that consequential decisions are occurring in locations that were never intended to hold decision authority.

The authority surface thus provides a structural lens for examining how authority operates in practice. By identifying where system state changes can occur, organizations gain visibility into the true locations of decision power and can begin aligning authority with responsibility, accountability, and governance objectives.

5. Authority Topology

Identifying the authority surface of an organization reveals where consequential decisions can alter the state of the system. The next question concerns how authority is distributed across those locations. The pattern through which authority is arranged within a system is described as **Authority Topology**.

Authority topology refers to the structural distribution of legitimate decision capacity across the authority surface. It describes how authority is concentrated, distributed, or fragmented within the organization and how these patterns influence the stability and effectiveness of governance.

In many organizations, authority topology is assumed to mirror the hierarchical structure shown in organizational charts. Hierarchies are often interpreted as maps of decision authority, with authority presumed to flow downward from senior leadership through successive layers of management. While hierarchical structures can represent formal accountability relationships, they frequently fail to describe how authority actually operates across the authority surface.

Operational realities often produce authority distributions that diverge from formal hierarchies. Technical administrators may possess the ability to alter system configurations that carry operational or regulatory consequences. Financial control systems may permit operational teams to commit resources within defined limits. Automated deployment systems may execute changes to production environments without requiring real-time managerial approval. Each of these situations represents a location within the authority surface where authority is exercised independently of traditional managerial hierarchy.

Understanding authority topology therefore requires examining how decision capacity is distributed across the locations identified in the authority surface. Several structural patterns commonly appear within organizations:

- **Concentrated authority**, where decision capacity is centralized within a small number of actors or governing bodies.
- **Distributed authority**, where decision capacity is intentionally placed across multiple operational locations within the system.
- **Fragmented authority**, where decision capacity exists across multiple actors without clear structural coordination.

Each pattern carries different implications for governance stability. Highly concentrated authority can provide strong control but may slow decision processes in complex operational environments. Distributed authority can increase responsiveness and operational effectiveness when authority placement is deliberate and well defined. Fragmented authority, by contrast, often produces conflict, ambiguity, and inconsistent decision outcomes because authority boundaries are not clearly structured.

Authority topology therefore provides a structural lens for evaluating how authority is arranged across an organization's authority surface. By examining where authority resides

and how it is distributed, organizations can identify whether authority structures support coherent decision making or whether they produce patterns of conflict, delay, or informal authority emergence.

Understanding authority topology is a necessary step before authority structures can be intentionally designed. Once the distribution of authority across the authority surface is visible, organizations can begin evaluating whether that distribution aligns with operational requirements, governance objectives, and the legitimate capacity to produce consequential system change.

6. Authority Primitives

The authority surface identifies where consequential system change can occur, and authority topology describes how authority is distributed across those locations. Designing an authority architecture requires a structural vocabulary capable of describing how authority functions within the system. That vocabulary is provided through **Authority Primitives**.

Authority Primitives represent the fundamental structural roles required for authority to exist and operate within a bounded system. They do not describe job titles, reporting relationships, or specific organizational positions. Instead, they define the structural functions that must exist in order for authority to be established, exercised, constrained, and stabilized.

Within the Authority Architecture framework, eleven primitives describe these functions:

- Intent
- Policy
- Risk Acceptance
- Assignment
- Resource
- Boundary
- Decision
- Execution
- Control
- Oversight

- Accountability

Organizations implement these primitives through a variety of mechanisms. In some cases the primitive may correspond to a formal governance role. In others it may be embedded within operational procedures, technical systems, or automated mechanisms. The primitives themselves remain structural concepts; the specific mechanisms used to implement them may vary across organizations.

Intent

The Intent primitive defines the objectives the system is designed to pursue. It establishes the strategic direction that guides decision making throughout the organization. Intent determines what outcomes the authority system is ultimately meant to produce.

Policy

The Policy primitive translates intent into governing rules. Policy establishes the constraints and expectations that shape how authority may be exercised within the system.

Risk Acceptance

The Risk Acceptance primitive determines who may legitimately accept exposure on behalf of the system. Decisions that introduce operational, financial, or regulatory consequences require an actor who possesses the recognized capacity to accept that exposure.

Assignment

The Assignment primitive determines who is granted authority roles within the system. Through assignment, individuals, roles, or mechanisms are formally designated to exercise specific authority functions.

Resource

The Resource primitive governs the allocation of the capabilities required to produce system change. Resources may include financial capital, personnel, infrastructure, or technological capacity. Without access to resources, authority cannot produce meaningful outcomes.

Boundary

The Boundary primitive defines the limits within which authority may operate. Boundaries may be defined by jurisdiction, operational scope, regulatory requirements, or system design.

Decision

The Decision primitive represents the legitimate capacity to select among available courses of action. It defines who determines what action the system will take.

Execution

The Execution primitive represents the ability to carry out decisions in the operational environment. Execution authority may exist in human actors, technical systems, automated processes, or operational mechanisms.

Control

The Control primitive provides the capacity to interrupt, halt, or redirect execution when necessary. Control functions become particularly important in systems that operate at high speed or through automated processes.

Oversight

The Oversight primitive provides visibility into how authority is exercised across the system. Oversight functions include monitoring, auditing, and governance review mechanisms that observe authority in operation.

Accountability

The Accountability primitive establishes the ability to enforce consequences when authority is misused or when decisions produce unacceptable outcomes. Accountability stabilizes the authority system by linking authority to responsibility for results.

Meta Structure of the Primitives

Although each primitive performs a distinct structural function, the primitives naturally organize into four broader domains that describe how authority operates within the system. These domains represent the lifecycle through which authority is established, exercised, and stabilized.

The four domains are:

- Direction
- Structure
- Execution
- Assurance

The **Direction** domain establishes the intent, governing rules, and risk posture that guide the system. The **Structure** domain defines how authority roles are assigned, resourced, and bounded within the organization. The **Execution** domain governs how decisions are made, implemented, and controlled. The **Assurance** domain stabilizes the system by ensuring that authority remains observable and accountable over time.

Unlike the other domains, the Assurance domain is intentionally asymmetrical. While the Direction, Structure, and Execution domains each contain three primitives, Assurance contains two. This asymmetry reflects the distinct function of assurance within the authority architecture.

Direction, Structure, and Execution together produce system action. Assurance does not generate decisions or operational change directly. Instead, it operates across the entire architecture to observe authority in use and enforce consequences when authority is misapplied. Because assurance functions stabilize the authority system rather than produce operational decisions, fewer primitives are required to perform this role.

Authority Primitive Meta Structure

Domain	Primitives
--------	------------

Direction	Intent, Policy, Risk Acceptance
------------------	---------------------------------

Structure	Assignment, Resource, Boundary
------------------	--------------------------------

Execution	Decision, Execution, Control
------------------	------------------------------

Assurance	Oversight, Accountability
------------------	---------------------------

Authority Primitives provide the structural language used to design authority architectures. By identifying which primitives exist at each location on the authority surface and examining how those primitives are distributed across the authority topology, organizations gain the ability to analyze whether authority structures are coherent or structurally misaligned.

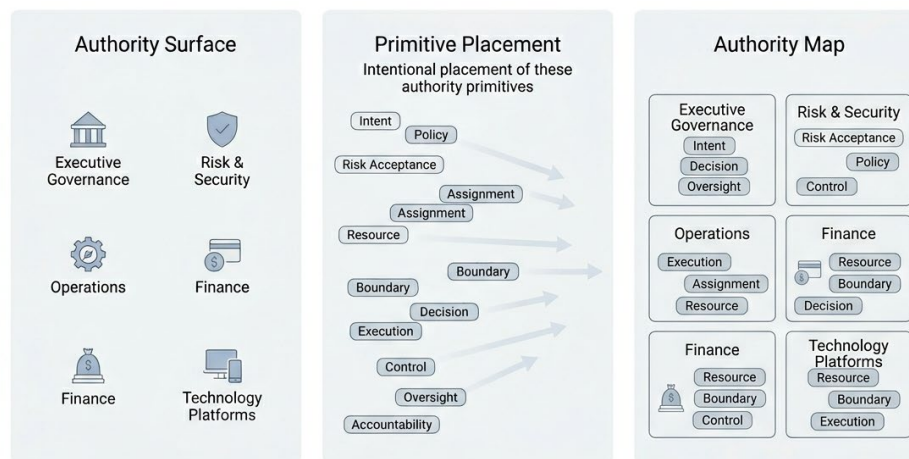
7. Authority Mapping

Once authority primitives are defined and the authority surface and topology of a system are understood, the next step is to determine how these elements interact in practice.

Authority Mapping is the process used to visualize and evaluate how authority is structured across a system.

Authority Mapping identifies where authority exists on the authority surface and determines which primitives are present at those locations. The mapping process is illustrated in Figure 3. By examining how primitives are placed across the surface, organizations can determine whether authority is coherent, fragmented, or incomplete.

Authority Mapping Exercise



Authority Mapping differs from responsibility assignment models such as RACI. RACI frameworks clarify who is responsible, accountable, consulted, or informed within a defined process. These models are useful for coordinating work and clarifying operational roles. Authority Mapping addresses a different structural question: where legitimate decision capacity exists within a system and how the primitives required for authority are distributed across the authority surface.

While responsibility models describe participation in work, Authority Mapping examines the architecture of decision authority itself. This process moves authority analysis from abstract governance discussions to a structural examination of how authority operates in practice.

Authority Mapping begins by identifying the relevant **system boundary**. Every mapping effort must define the scope of the system being examined. In some cases the boundary may encompass an entire organization. In others it may focus on a particular operational

domain, such as financial decision making, technology operations, or regulatory compliance.

Once the system boundary is defined, the next step is to identify the **authority surface** by locating the points within the system where decisions can produce consequential change. These locations may include operational processes, governance bodies, technical systems, or automated mechanisms that initiate or execute actions affecting the state of the system.

After the authority surface has been identified, the primitives are mapped across those locations. This step examines which primitives are present at each location and how they interact. Rather than relying on an exhaustive checklist, Authority Mapping typically begins with a small set of structural questions aligned with the primitive domains.

For example:

- **Direction:** Where is intent defined for this portion of the system, and what policies or risk acceptance decisions constrain it?
- **Structure:** Who assigns authority in this domain, what resources enable it, and what boundaries define its limits?
- **Execution:** Who makes decisions, how are those decisions executed, and what mechanisms exist to control execution if necessary?
- **Assurance:** What mechanisms observe authority in operation, and who enforces accountability for its outcomes?

These questions illustrate how primitives are examined across the authority surface. The objective is not to interrogate every location within the system, but to determine whether the primitives required for coherent authority are present and properly aligned.

In many organizations, authority mapping exposes structural conditions that are not visible through traditional governance models. Execution mechanisms may exist without clear decision authority. Decision authority may exist without defined boundaries or without actors capable of accepting the associated risks. In other cases, oversight and accountability functions may be weak or entirely absent.

The purpose of Authority Mapping is therefore both diagnostic and architectural. It reveals how authority is currently structured within the system and provides a method for designing authority architectures that align authority, responsibility, and accountability with the locations where consequential decisions occur.

8. Authority Entropy

Authority architectures do not remain static over time. Even when authority is intentionally designed and properly aligned with organizational objectives, the distribution and operation of authority tend to change as systems evolve. This tendency for authority structures to drift away from their original design is referred to as **Authority Entropy**.

Authority Entropy describes the gradual degradation of alignment between the intended authority architecture and the way authority is exercised in practice. As organizations grow, adopt new technologies, reorganize operational structures, and respond to changing external conditions, authority relationships naturally shift. Unless these changes are examined and corrected, authority structures that were once coherent can become fragmented or misaligned.

Several forces commonly contribute to this process.

Organizational growth introduces new roles, teams, and operational domains. As responsibilities expand, authority is often delegated incrementally to support operational efficiency. Over time these delegations may accumulate without corresponding adjustments to governance structures, producing decision authority in locations that were not originally intended to hold it.

Technological change also alters authority relationships. Technical systems increasingly automate actions that previously required human decision making. Configuration systems, automated deployment pipelines, algorithmic decision engines, and other automated mechanisms can initiate consequential system changes at speeds and scales that traditional governance processes were not designed to oversee. As these systems expand, authority may effectively migrate into technical mechanisms that operate outside the awareness of formal governance structures.

Operational pressures can produce similar effects. Teams responsible for maintaining operational continuity may develop informal practices that allow them to act quickly in response to incidents or performance demands. While these practices may initially emerge as temporary workarounds, they can become embedded within operational routines, gradually shifting authority away from formally designated decision structures.

Authority Entropy does not imply failure or mismanagement. It reflects a structural property of complex systems. As systems evolve, the locations where consequential decisions occur and the actors capable of producing those decisions inevitably change. Authority architectures that are not periodically examined and adjusted will therefore drift over time.

The practical implication is that authority architecture requires ongoing maintenance. Just as technical systems require monitoring and adjustment to remain stable, authority

systems must be periodically reexamined to ensure that authority remains aligned with responsibility, accountability, and the locations where consequential decisions occur.

Understanding Authority Entropy provides the foundation for recognizing the structural conditions that emerge when authority architectures degrade. The following section examines the common failure modes that arise when authority structures become misaligned or unstable.

9. Failure Modes

Authority architectures are designed to align responsibility, decision capacity, and accountability across the authority surface of an organization. When this alignment is maintained, authority operates coherently and decisions can be executed in a predictable and governed manner. When authority structures degrade or become misaligned, the resulting conditions tend to follow recognizable structural patterns.

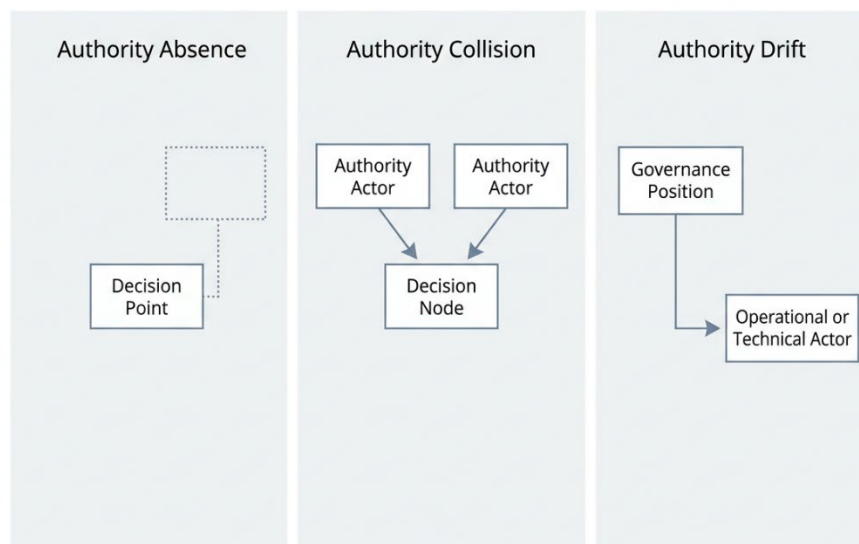
Within the Authority Architecture framework, these conditions are described as **failure modes**. Failure modes represent structural states in which the placement or interaction of authority primitives no longer produces a coherent authority system.

Three failure modes appear consistently across organizations when authority architectures become unstable:

- **Authority Absence**
- **Authority Collision**
- **Authority Drift**

These failure modes are illustrated in Figure 4.

Authority Failure Modes



Each failure mode reflects a different type of structural misalignment within the authority architecture.

Authority Absence

Authority Absence occurs when a location on the authority surface is capable of producing consequential system change, but no actor possesses legitimate authority to initiate the decision required to act.

In these situations responsibility may exist, but the corresponding authority required to act is missing. Decisions stall because actors lack recognized legitimacy to proceed. Work may be escalated repeatedly through organizational layers in search of a legitimate decision authority, delaying action and increasing operational uncertainty.

Authority Absence often emerges when new operational domains or technologies create decision points that were not anticipated by existing governance structures. Because the authority surface has expanded without corresponding adjustments to authority architecture, the system contains locations where decisions must occur but no legitimate authority has been assigned.

Authority Collision

Authority Collision occurs when multiple actors possess overlapping authority over the same location on the authority surface. In this condition more than one actor possesses legitimate decision capacity within the same domain.

Collisions often produce conflicting directives, veto cycles, or prolonged negotiation over decision ownership. Actors may each possess legitimate authority but lack clear structural boundaries defining when their authority applies.

Authority Collision commonly arises when governance structures are layered without structural integration. Regulatory oversight bodies, operational leadership, and specialized governance committees may each possess authority within the same domain, creating overlapping decision authority that produces uncertainty and delay.

Authority Drift

Authority Drift occurs when authority gradually migrates away from the locations where it was originally intended to reside. Over time operational practices, technological systems, or informal decision processes begin exercising authority that was not deliberately assigned within the authority architecture.

Drift often emerges as a consequence of Authority Entropy. As organizations grow, adopt new technologies, or adapt to operational pressures, authority may accumulate in locations that were originally designed only to execute decisions rather than make them. Automated systems, operational teams, or technical administrators may begin exercising decision authority because the system requires decisions to be made at the locations where operational change occurs.

In these conditions the formal authority architecture no longer reflects how authority actually operates within the system. Governance structures may continue to assume authority exists in one location while decisions are effectively being made elsewhere.

Failure modes rarely appear in isolation. Authority Absence, Collision, and Drift frequently interact, producing complex governance conditions in which authority is unclear, contested, or exercised informally. The presence of these conditions signals that the authority architecture has become misaligned with the authority surface and requires structural reassessment.

Understanding these failure modes allows organizations to diagnose structural weaknesses within their authority systems and to redesign authority architectures so that authority, responsibility, and accountability remain aligned with the locations where consequential decisions occur.

10. Applied Architectures

The preceding sections have introduced the structural concepts of Authority Architecture and the mechanisms through which authority can be analyzed and designed. While the framework is general and applicable across many domains, its value becomes clearer when applied to specific organizational contexts.

The following examples illustrate how authority architectures can be examined in practice. Each example represents a different environment in which consequential decisions occur: executive governance roles, automated operational systems, and artificial intelligence systems that influence decision making.

10.1 CISO Authority Architecture

Many organizations assign responsibility for cybersecurity to a Chief Information Security Officer (CISO) or an equivalent executive role. The CISO is typically charged with protecting the organization's information assets, managing cybersecurity risk, and ensuring that security practices align with regulatory and operational expectations.

In practice, however, the authority architecture surrounding the CISO role is often structurally misaligned. The CISO may be responsible for managing cybersecurity risk while lacking direct authority over the systems, infrastructure, and operational teams that control the technical environment in which those risks emerge.

In these situations the authority surface for cybersecurity decisions extends across multiple operational domains, including infrastructure management, software development, technology operations, procurement, and business process design. Each of these domains contains decision points capable of producing security consequences. Yet the primitives required to manage security risk—such as decision authority, resource control, or risk acceptance—may be distributed across multiple actors who do not report through the security function.

This configuration frequently produces the failure modes described earlier. Authority Absence may appear when security leaders are responsible for managing risk but lack legitimate authority to enforce corrective actions. Authority Collision may arise when operational leaders and security leaders both claim authority over security decisions within overlapping domains. Authority Drift may occur when operational teams begin exercising security-related decision authority simply because they control the systems where those decisions must occur.

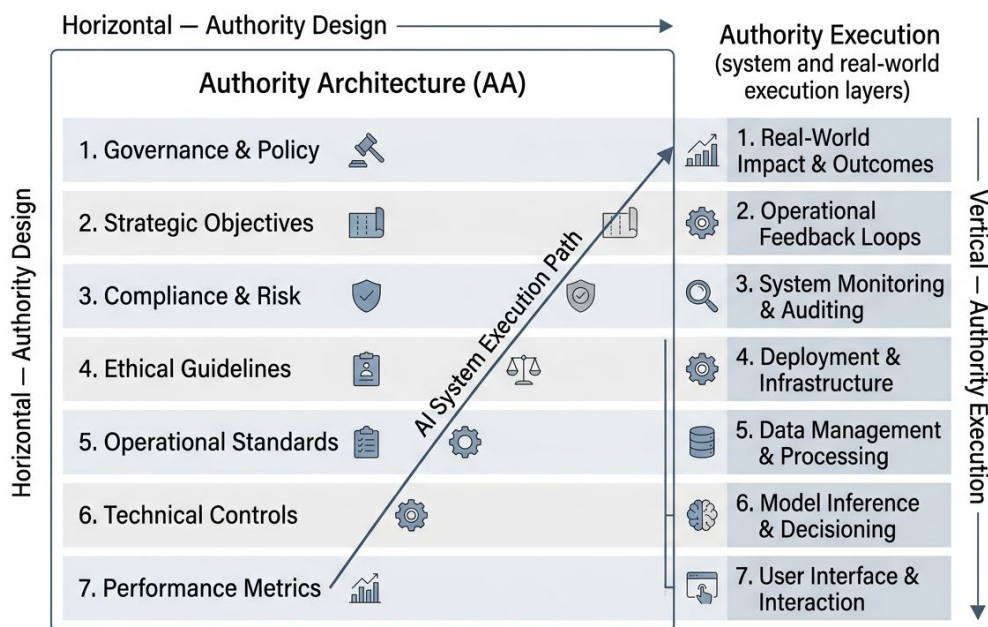
Examining the authority surface and topology surrounding cybersecurity operations allows organizations to identify these structural conditions and redesign authority placement so that security responsibility aligns with legitimate decision capacity and accountability.

10.2 Authority in Autonomous Systems

Automated and autonomous systems introduce additional complexity into authority architectures. As systems incorporate automated decision mechanisms, the locations where consequential system changes occur expand beyond traditional human decision points.

Autonomous systems do not create a new category of authority. Instead, they expose weaknesses in existing authority architectures. The way automated and AI-driven execution traverses authority design and execution layers is illustrated in Figure 5.

Authority Architecture Layers and AI Execution



When systems initiate actions, adapt operational logic, or operate without real-time human approval, decision capacity becomes embedded within algorithmic and technical structures. In these environments the question is not whether the system “has” authority, but whether the organization has deliberately designed the authority architecture surrounding the system.

Three structural conditions become particularly important in environments involving automated or autonomous decision processes.

First, the **authority surface** of the system must be clearly bounded. Organizations must identify the operational states that the system is capable of altering and understand the financial, regulatory, operational, or physical consequences those changes may produce.

Second, **control mechanisms** must exist that allow authority holders to interrupt or terminate automated execution when necessary. In systems operating at machine speed, these control mechanisms must operate at the same operational layer and tempo as the automated processes themselves.

Third, **risk acceptance** must be explicitly established before automated decision processes are deployed. Autonomous systems can generate cumulative consequences across large numbers of actions, meaning that risk acceptance cannot be evaluated solely at the level of individual transactions.

As organizations increasingly rely on automated systems to perform operational functions, authority architectures must account for the locations where automated mechanisms can produce consequential system change. Authority must be deliberately placed so that intent, policy, risk acceptance, execution control, oversight, and accountability remain aligned with the system's ability to act.

10.3 Authority in AI Systems

Artificial intelligence systems introduce a distinct challenge for authority architecture. Unlike traditional automated systems that execute predetermined instructions, many AI systems operate through probabilistic models that generate predictions, classifications, or recommendations based on learned patterns. These systems may influence decisions that carry operational, financial, legal, or societal consequences.

In these environments the authority question extends beyond execution. Organizations must determine where legitimate authority resides when decisions are influenced or partially produced by systems whose internal reasoning may not be fully transparent.

AI systems frequently shape decisions in domains such as credit approval, hiring, fraud detection, operational optimization, and medical diagnosis. Even when a human actor remains formally responsible for the final decision, the recommendations generated by an AI system may strongly influence the outcome. In practice, this means the authority surface expands to include locations where algorithmic outputs affect consequential decisions.

This expansion of the authority surface introduces new structural requirements for authority architecture. Organizations must determine who possesses the legitimate

authority to deploy AI systems, who accepts the risks associated with their outputs, and who can intervene when those outputs produce undesirable or unintended consequences.

Several primitives become particularly important in these environments. **Risk Acceptance** must be explicitly assigned before AI systems are deployed, as probabilistic systems can produce outcomes that were not anticipated at design time. **Control** mechanisms must exist to interrupt or override system influence when necessary. **Oversight** functions must be capable of monitoring how AI systems influence decisions across the authority surface, and **Accountability** must remain attached to actors who possess legitimate authority to accept the consequences of those decisions.

Authority Architecture provides a structural framework for examining these conditions. By identifying where AI systems influence decision outcomes and ensuring that the appropriate primitives surround those locations, organizations can maintain coherent authority structures even as decision processes become increasingly influenced by machine learning systems.

11. Conclusion

Organizations have long developed formal disciplines for designing the structures that support their operations. Technical architectures determine how systems are built and operated. Financial architectures govern the allocation and control of resources. Risk management frameworks establish mechanisms for identifying and managing uncertainty. Each of these domains recognizes that complex systems require deliberate structural design in order to function coherently.

Authority has rarely been treated in the same way.

In many organizations, authority is assumed to emerge naturally from hierarchy, role descriptions, or institutional tradition. As long as reporting lines appear clear and governance bodies exist, authority is presumed to be present. The preceding analysis demonstrates that this assumption is often incorrect. Authority exists wherever a system can be changed through legitimate decision capacity, and the locations where those decisions occur frequently extend beyond formal organizational structures.

Authority Architecture provides a framework for examining and designing how authority operates within a bounded system. By defining authority as the legitimate capacity to cause a bounded system to change state, the framework establishes a structural foundation for analyzing where authority exists, how it is distributed, and how it evolves over time.

The concepts introduced in this paper form a coherent analytical structure. The **authority surface** identifies the locations where consequential system change can occur. **Authority topology** describes how authority is distributed across those locations. **Authority primitives** define the structural roles required for authority to function within the system. **Authority mapping** provides a method for examining how those primitives are placed across the authority surface. **Authority entropy** explains why authority structures drift over time, and the failure modes of **authority absence**, **authority collision**, and **authority drift** describe the structural conditions that emerge when authority architectures become misaligned.

Together these elements provide a structural approach to understanding authority within complex organizations.

The importance of this approach becomes more apparent as organizations operate in environments characterized by increasing technological complexity, automation, and the growing influence of artificial intelligence. Decisions that once occurred within clearly defined managerial structures are now distributed across operational teams, automated systems, and algorithmic processes. In these environments, the locations where consequential system changes occur may not correspond to traditional organizational boundaries.

Without a deliberate approach to authority design, authority will still exist, but it will emerge informally through operational necessity rather than through intentional governance. Informal authority structures may function for a time, but they often diverge from the organization's formal accountability structures, producing the instability described throughout this paper.

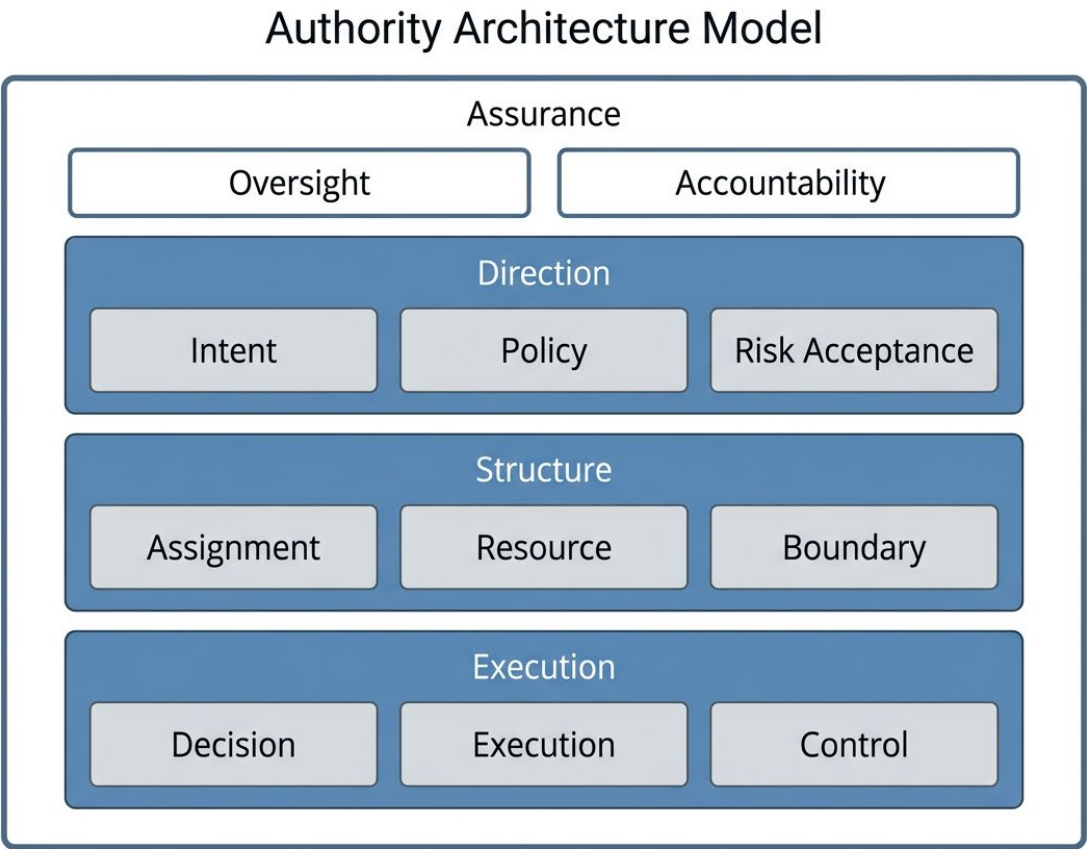
Treating authority as an architectural element of organizational design allows these conditions to be examined and corrected. By identifying where authority exists, how it is distributed, and how it interacts with responsibility and accountability, organizations can design authority architectures that remain coherent even as systems grow more complex.

Authority Architecture therefore represents an emerging discipline concerned with the structural design of decision capacity within complex systems. As organizations continue to rely on distributed technologies, automated processes, and AI-driven decision support, the need to deliberately design authority structures will only become more significant.

The framework presented in this paper provides a foundation for that work.

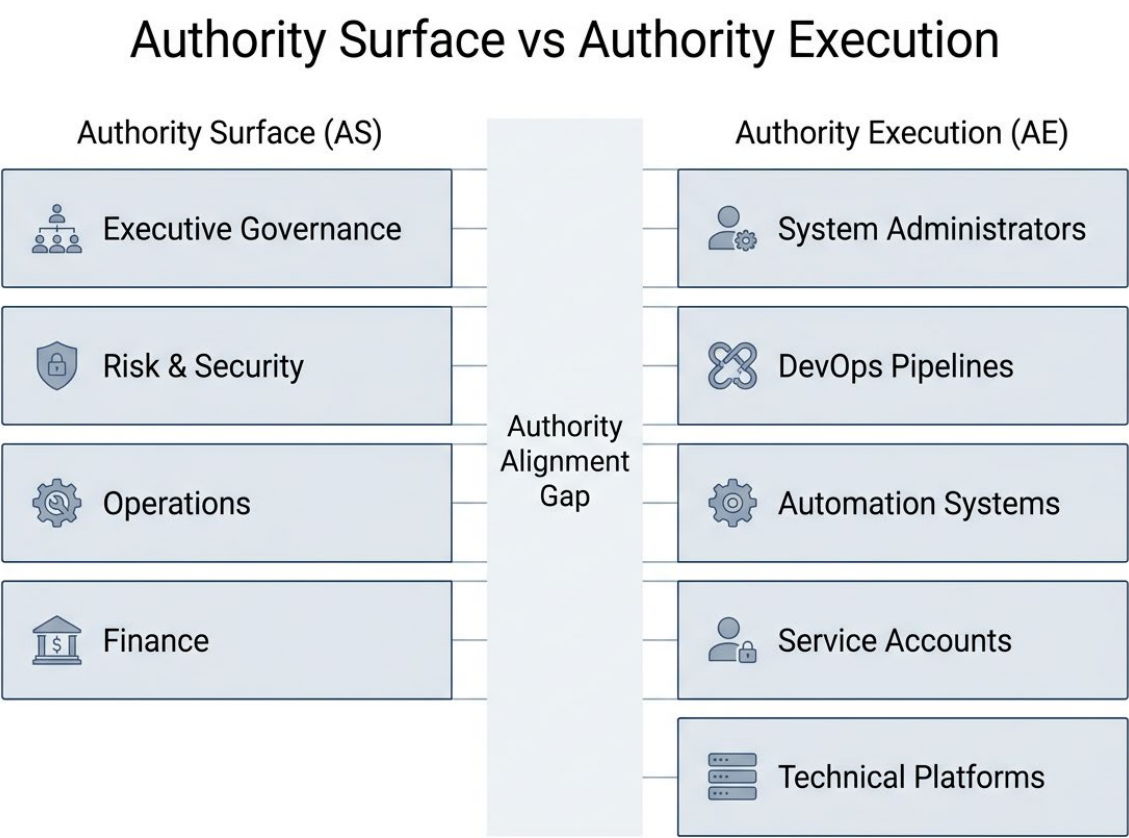
List of Figures:

Figure 1: Authority Architecture Model



The Authority Architecture Model describes how authority operates within a bounded system. Direction establishes intent, policy, and risk posture. Structure assigns authority roles, resources, and operational boundaries. Execution determines and implements system action. Assurance stabilizes the architecture through oversight and accountability.

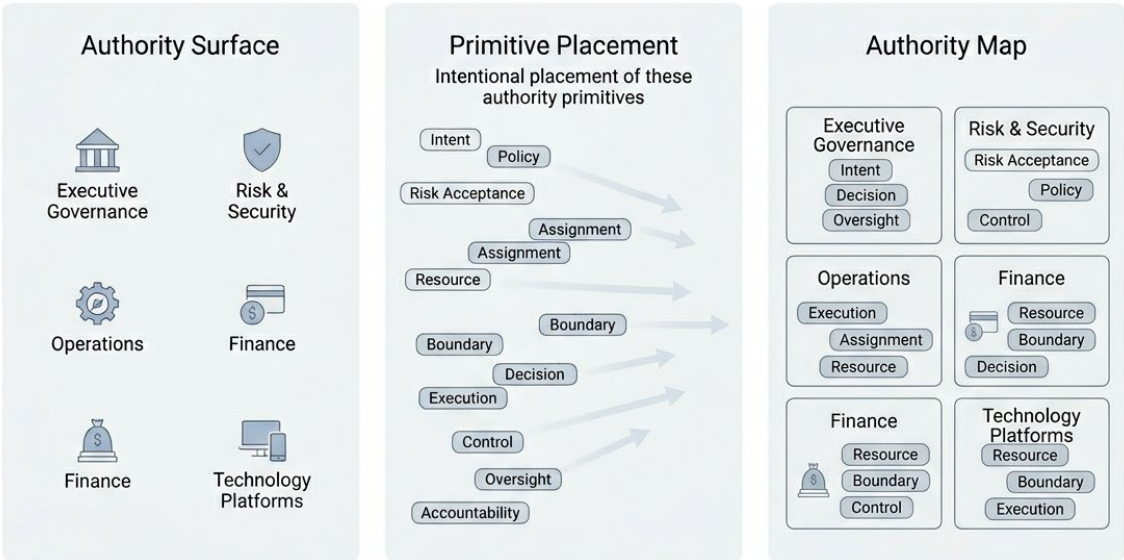
Figure 2: Authority Surface vs. Authority Execution



Organizational structures define where authority is expected to exist (Authority Surface), while technical systems determine where consequential actions are executed (Authority Execution). The difference between these structures creates an Authority Alignment Gap, where governance responsibility and system control may diverge.

Figure 3: Authority Mapping Exercise

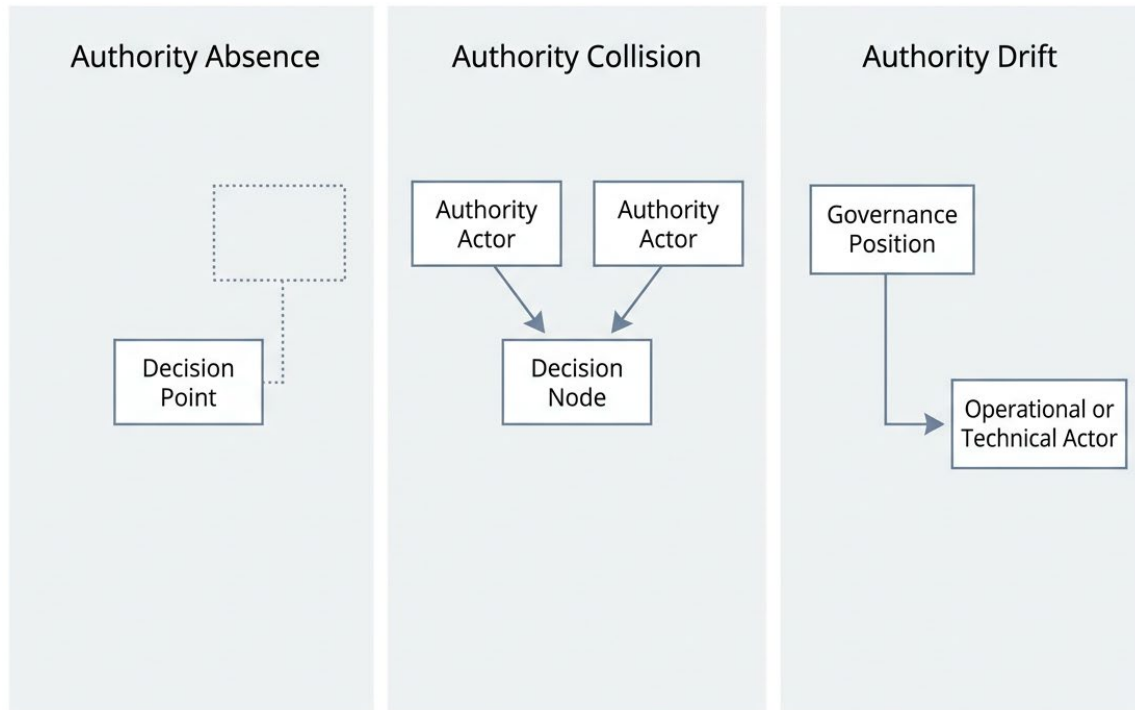
Authority Mapping Exercise



Authority Mapping places structural authority primitives across the authority surface of an organization. The resulting map reveals where authority is formally assigned, where it is absent, and where potential collisions or drift may occur.

Figure 4: Authority Failure Modes

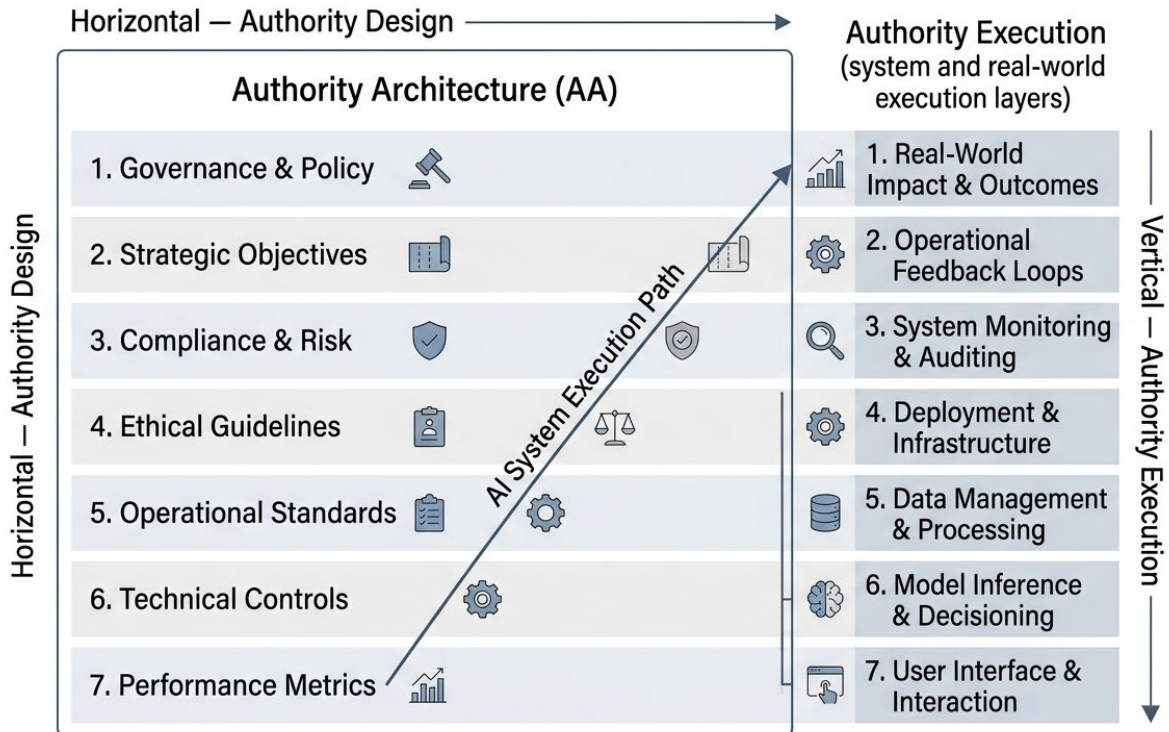
Authority Failure Modes



Authority architectures may fail in three structural ways. Authority Absence occurs when a decision exists but no actor possesses legitimate authority to act. Authority Collision occurs when multiple actors possess overlapping authority over the same decision space. Authority Drift occurs when authority gradually migrates away from its intended governance location.

Figure 5: Authority Architecture Layers and AI Execution

Authority Architecture Layers and AI Execution



Autonomous systems traverse multiple governance and execution layers simultaneously. While authority is designed across governance layers, system execution compresses these layers into real-time decision paths that propagate through technical and physical systems. This relationship is illustrated in Figure 5: Authority Architecture and AI Execution.

Appendix A — Authority Mapping Exercise

This appendix provides a structured method for applying Authority Architecture to a real organization or system. The purpose of the exercise is to identify where authority exists within a bounded system, determine how authority primitives are distributed across the authority surface, and evaluate whether the resulting authority architecture is coherent.

The exercise may be conducted using an organizational chart, system architecture diagram, or operational process map. The objective is not to document every operational detail of the system, but to identify the locations where consequential decisions can alter the state of the system.

Step 1 — Define the System Boundary

Begin by defining the system that will be examined. The system boundary determines the scope of the authority architecture under review.

Examples of system boundaries include:

- an entire organization
- a functional domain such as cybersecurity, finance, or operations
- a technical environment such as a cloud platform or production system
- a specific automated or AI-driven decision process

Clearly defining the system boundary ensures that the authority surface being examined corresponds to a coherent operational domain.

Step 2 — Identify the Authority Surface

Next identify the locations within the system where decisions can produce consequential state change. These locations form the authority surface.

Authority surface locations may include:

- executive decision bodies
- operational teams
- technical system controls
- automated processes
- algorithmic decision systems

The purpose of this step is to identify **where consequential decisions occur**, not simply where authority is assumed to exist.

Step 3 — Map the Authority Primitives

Once the authority surface has been identified, examine each location and determine which authority primitives are present.

The primitives to evaluate are:

Direction

- Intent
- Policy
- Risk Acceptance

Structure

- Assignment
- Resource
- Boundary

Execution

- Decision
- Execution
- Control

Assurance

- Oversight
- Accountability

For each location on the authority surface, determine which primitives are exercised there and which actors or mechanisms perform those roles.

Step 4 — Examine Primitive Alignment

After primitives have been mapped across the authority surface, examine how they interact.

Questions to consider include:

- Are the primitives required for coherent authority present at each decision location?
- Are authority roles clearly assigned?
- Are the resources and boundaries necessary for execution defined?
- Do oversight and accountability functions observe and stabilize authority in operation?

This step reveals whether authority structures are aligned or structurally incomplete.

Step 5 — Identify Failure Modes

Finally examine the mapped authority architecture for the structural failure modes described in this paper.

Look for evidence of:

Authority Absence

Locations where consequential decisions must occur but no actor possesses legitimate authority to initiate them.

Authority Collision

Locations where multiple actors possess overlapping authority within the same decision space.

Authority Drift

Locations where authority has migrated away from the structures originally intended to govern those decisions.

These conditions indicate that the authority architecture may require redesign.

Step 6 — Evaluate Authority Topology

The final step is to examine the overall distribution of authority across the system.

Determine whether authority is:

- concentrated in a small number of decision bodies
- intentionally distributed across operational domains
- fragmented across actors without clear structural coordination

Understanding the authority topology allows organizations to evaluate whether authority placement supports coherent governance and operational effectiveness.

Authority Mapping does not prescribe a single correct authority architecture. Different systems may require different distributions of authority depending on operational requirements, regulatory obligations, and organizational design.

The purpose of the exercise is to make authority visible as an architectural element of the system so that authority structures can be intentionally designed rather than assumed.

Appendix B — Authority Primitive Reference

This appendix provides a concise reference for the eleven Authority Primitives described in this paper. The primitives represent the fundamental structural roles required for authority to operate within a bounded system. They are organized according to the four domains introduced in the Authority Architecture framework: Direction, Structure, Execution, and Assurance.

The primitives themselves are not job titles or organizational positions. Instead, they describe structural functions that must exist somewhere within the system in order for authority to be defined, exercised, constrained, and stabilized.

Organizations implement these primitives through a variety of mechanisms, including executive roles, governance bodies, operational procedures, technical systems, and automated processes.

Authority Primitive Reference Table

Domain	Primitive	Structural Function
Direction	Intent	Defines the objectives and outcomes the system is designed to pursue
Direction	Policy	Establishes the governing rules that constrain how authority may be exercised
Direction	Risk Acceptance	Determines who may legitimately accept exposure resulting from consequential decisions
Structure	Assignment	Determines who is granted authority roles within the system
Structure	Resource	Allocates the capabilities required to produce system change
Structure	Boundary	Defines the limits within which authority may operate
Execution	Decision	Determines which course of action the system will take
Execution	Execution	Implements decisions within the operational environment
Execution	Control	Interrupts, halts, or redirects execution when necessary

Domain	Primitive	Structural Function
Assurance Oversight		Observes and evaluates how authority is exercised across the system
Assurance Accountability		Enforces consequences when authority is misused or decisions produce unacceptable outcomes

The four domains together describe the lifecycle through which authority operates within a system.

The **Direction** domain establishes the intent, rules, and risk posture that guide the system. The **Structure** domain defines how authority roles are assigned, resourced, and bounded. The **Execution** domain governs how decisions are made, implemented, and controlled. The **Assurance** domain stabilizes the authority architecture by ensuring that authority remains observable and accountable over time.

Together these primitives form the structural vocabulary used to design and analyze authority architectures.

Appendix C — Authority Architecture Diagnostics

Authority architectures rarely fail suddenly. Structural weaknesses typically emerge gradually as authority relationships drift away from their intended design. The diagnostic indicators in this appendix provide a practical method for recognizing when authority architectures may be misaligned.

These indicators correspond to the three structural failure modes described in this paper: Authority Absence, Authority Collision, and Authority Drift.

The presence of one or more of these indicators does not necessarily indicate organizational failure. Instead, they signal that the authority architecture may require examination through the Authority Mapping process described in Appendix A.

Authority Absence Indicators

Authority Absence occurs when decisions capable of producing consequential system change must occur, but no actor possesses legitimate authority to initiate those decisions.

Common indicators include:

- responsibility assigned without corresponding decision authority
- repeated escalation of routine decisions
- operational delays while organizations determine who is authorized to act
- decisions deferred to informal consensus rather than legitimate authority
- risk acceptance occurring implicitly rather than through designated authority

These conditions suggest that the authority surface has expanded beyond the locations where authority primitives have been intentionally placed.

Authority Collision Indicators

Authority Collision occurs when multiple actors possess overlapping authority within the same decision space.

Common indicators include:

- conflicting directives from multiple leaders or governance bodies
- repeated veto cycles or decision reversals

- operational teams receiving incompatible instructions from different authorities
- committees or oversight groups exercising authority within the same domain without clear boundaries
- decision processes that require negotiation between actors who each claim legitimate authority

These conditions indicate that boundaries between authority domains may be unclear or structurally misaligned.

Authority Drift Indicators

Authority Drift occurs when authority gradually migrates away from the locations where it was originally intended to reside.

Common indicators include:

- operational workarounds that bypass formal governance processes
- technical administrators making decisions that carry strategic or regulatory consequences
- automated systems initiating consequential actions without clearly assigned oversight
- governance structures that assume authority exists in one location while decisions are effectively made elsewhere
- informal decision practices becoming institutionalized over time

These indicators often appear in environments experiencing rapid technological change or operational growth.

Using the Diagnostics

When multiple indicators from any category appear within an organization, the authority architecture should be examined through the Authority Mapping process.

The objective of the diagnostic indicators is not to assign blame or identify individual failures. Instead, they provide an early signal that the structural placement of authority primitives across the authority surface may no longer align with the way decisions occur within the system.

Authority architectures that are periodically examined and adjusted are more likely to remain coherent as organizations evolve.

About the Author

Rob Caswell is an executive leader and security architect focused on the structural design of authority within complex organizational systems. His work examines how decision authority operates across executive governance, operational environments, and increasingly automated technological systems.

Through his research and applied work, Caswell has developed the **Authority Architecture** framework to address a recurring problem in modern organizations: responsibility for outcomes is frequently assigned without corresponding authority to produce them. The framework provides a structural method for examining how authority exists within bounded systems, how it is distributed across organizations and technologies, and how authority architectures evolve as systems become more complex.

His work explores the intersection of governance, cybersecurity leadership, automation, and artificial intelligence, with particular focus on how organizations can intentionally design authority structures that remain coherent as decision processes become increasingly distributed across humans, systems, and algorithmic mechanisms.