Deterministic Governance Enforcement for Autonomous Infrastructure

The Petersen Governance Plane and Governance Invariant Compilation

Author:
J. D. “Pepper” Petersen
Aristotle Agentic
Helena, Montana, USA

Abstract

Autonomous artificial intelligence systems are increasingly deployed within operational infrastructure including telecommunications networks, robotics platforms, distributed energy systems, and cyber-physical control environments. These systems generate probabilistic outputs derived from statistical inference while the infrastructure they influence must operate within deterministic safety constraints and institutional authority boundaries.

Existing AI safety approaches rely primarily on runtime policy reasoning and software guardrails. While these mechanisms improve model reliability, they introduce latency and are susceptible to temporal drift, where authority granted during request evaluation becomes invalid by the time execution occurs.

This paper introduces the Petersen Governance Plane Architecture, a layered governance control framework designed to enforce institutional authority and safety constraints within the execution path of autonomous systems. A central contribution of the architecture is Governance Invariant Compilation, which transforms high-level governance artifacts into deterministic runtime invariants suitable for constant-time validation at the execution boundary.

By shifting governance from runtime reasoning to compiled constraint enforcement, the architecture enables autonomous systems that remain governable, auditable, and deployable within safety-critical infrastructure environments.

I. Introduction

Artificial intelligence systems are transitioning from advisory software to operational infrastructure. Autonomous agents now participate directly in the control loops of telecommunications networks, robotics fleets, distributed energy systems, and financial transaction infrastructure.

Machine learning systems generate probabilistic outputs based on statistical inference. However, the environments they influence operate under deterministic constraints including safety limits, regulatory requirements, and institutional authority structures.

Current AI governance frameworks focus primarily on model alignment and software-based policy enforcement. These mechanisms cannot guarantee that autonomous actions remain authorized and safe at the moment execution occurs.

A new architectural approach is therefore required in which governance enforcement operates within the execution path of autonomous systems.

II. Governance as a Control Plane

The Petersen Governance Plane introduces a control-plane architecture positioned between autonomous decision systems and operational infrastructure.

The governance stack is structured as follows:

Institutional Authority
│
Meta Authority Envelope
│
Authority Envelope
│
Governance Plane
│
Governance Invariant Compiler
│
Runtime Invariant Registers
│
Commit-Point Execution Gate
│
Physical Invariant Gater
│
Operational Infrastructure

Within this architecture, autonomous actions must satisfy governance constraints before reaching infrastructure systems.

Governance artifacts within this architecture originate from institutional authority structures rather than ad-hoc system configuration. Authority envelopes are therefore treated as governed artifacts with explicit provenance, including specification authority, review lineage, and delegation scope. Each envelope represents a structured declaration of operational authority derived from higher-level institutional mandates and may be cryptographically signed or otherwise formally attributed to its issuing authority.

This structure establishes a verifiable authority chain in which institutional mandates define meta-authority envelopes that constrain the creation of operational authority envelopes. These artifacts are subsequently compiled into runtime governance invariants enforced at execution time. By embedding provenance and delegation lineage into governance artifacts prior to compilation, the governance plane can validate not only whether an action satisfies operational constraints, but whether those constraints themselves originate from legitimate authority.

This mechanism prevents unauthorized policy modification and ensures that enforcement reflects institutional governance rather than mutable runtime configuration.

III. Governance Invariant Compilation

High-level governance artifacts such as Authority Envelopes are designed for policy expression rather than real-time execution enforcement.

The architecture introduces a Governance Invariant Compiler that transforms governance artifacts into deterministic runtime invariants.

These invariants may be represented using several implementation mechanisms including:

capability registers
permission bitmasks
numeric bound registers
lookup tables
comparator circuits
deterministic constraint logic

This compilation process eliminates the need for runtime policy reasoning during action execution.

IV. Commit-Point Authorization

Execution authorization occurs at a deterministic validation point referred to as the Commit Boundary.

At this boundary, a proposed action must satisfy three simultaneous conditions.

Identity Verification confirms that the acting model possesses valid lineage credentials.

Authority Validation confirms that the proposed action lies within compiled governance invariants derived from valid authority envelopes.

State Validation confirms that real-time telemetry indicates the environment remains safe.

Only when all three conditions are satisfied does the execution gate permit the action.

V. Temporal Drift Mitigation

Traditional governance systems validate authority during request evaluation.

However, infrastructure environments may experience changes between request and execution.

The architecture mitigates this risk through commit-point validation, intersecting compiled authority constraints with real-time telemetry at the moment execution occurs.

This ensures that authorization reflects the actual system state rather than stale authorization decisions.

VI. Deterministic Safety Enforcement

Software-based governance enforcement cannot guarantee physical safety under all conditions.

The architecture therefore introduces a Physical Invariant Gater, a deterministic enforcement component located between the execution gate and infrastructure systems.

This mechanism enforces hardware-level safety constraints such as:

actuator torque limits
RF transmission limits
electrical thresholds
environmental safety boundaries

These safeguards ensure that infrastructure cannot execute unsafe commands even if higher-level systems fail.

VII. Infrastructure Deployment Context

The architecture applies to multiple infrastructure environments including:

telecommunications switching systems
robotics and automated logistics
distributed energy control networks
autonomous vehicle systems
financial transaction infrastructure

In each case the governance plane establishes a deterministic boundary between probabilistic AI systems and operational infrastructure.

VIII. Conclusion

Autonomous infrastructure requires governance enforcement mechanisms capable of operating at execution speed rather than the speed of policy reasoning.

The Petersen Governance Plane architecture accomplishes this by compiling governance artifacts into deterministic runtime invariants and validating them at the commit boundary of execution.

By structuring governance artifacts within institutional authority hierarchies and enforcing them through deterministic runtime validation, the architecture enables autonomous systems that remain governable, auditable, and compatible with the operational requirements of real-world infrastructure.