Preprints.org

Article

Not peer-reviewed version

Controlled Agentic AI Systems: A Governance-Driven Architecture for Auditable and Reproducible Decision Pipelines

Tymoteusz Miller *

Posted Date: 24 March 2026

doi: 10.20944/preprints202603.1904.v1

Keywords: controlled AI systems; governance operator; constraint-aware AI; agentic systems; auditable AI; reproducible AI; federated learning governance; decision pipeline architecture; safety-critical AI; regulatory AI systems

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Controlled Agentic AI Systems: A Governance-Driven Architecture for Auditable and Reproducible Decision Pipelines

Tymoteusz Miller [1,2]

[1] Institute of Marine and Environmental Sciences, University of Szczecin, Wąska 13, 71-415 Szczecin, Poland; tymoteusz.miller@usz.edu.pl

[2] Faculty of Data Science and Information, INTI International University, Nilai 71800, Malaysia

Abstract

Artificial intelligence systems deployed in safety-critical and regulated environments require guarantees of constraint compliance, auditability, and reproducibility. However, contemporary AI architectures typically treat governance and regulatory constraints as external or post hoc mechanisms, which limits their ability to ensure consistent and reliable execution. This paper introduces Controlled Agentic AI Systems (CAIS), a formal architectural framework in which governance is embedded directly into the decision pipeline as a deterministic operator. A CAIS is defined as a system integrating a decision model M_θ, a constraint set C, and a governance operator G that maps proposed decisions into an admissible action space. Formalization of the decision transformation $a_t = G(M_\theta(x_t), C, s_t)$, introduce audit trace semantics, and define replayability conditions that enable reproducible execution. Theoretical analysis establishes that governance projection guarantees constraint satisfaction while inducing bounded decision drift under perturbations. Implementation of a reference framework and conduct controlled experiments in a multi-agent simulation environment under reproducible conditions. Results show that governance significantly reduces constraint violations, with approval-based gating achieving near-complete compliance and projection-based repair providing consistent mitigation. Crucially, these improvements are obtained without destabilizing system dynamics, and with bounded intervention cost. The experiments further reveal a structured trade-off between safety and decision drift across governance mechanisms. In federated settings, governance does not degrade convergence and instead stabilizes executed actions across training rounds, reducing action variance and eliminating constraint violations on benchmark states. These findings indicate that governance effectively decouples parameter-space variability from behavior-space outcomes. The proposed CAIS framework establishes governance as a fundamental architectural component of AI systems, providing a unified and experimentally validated approach to designing safe, auditable, and reproducible agentic intelligence.

Keywords: controlled AI systems; governance operator; constraint-aware AI; agentic systems; auditable AI; reproducible AI; federated learning governance; decision pipeline architecture; safety-critical AI; regulatory AI systems

1. Introduction

Artificial intelligence systems are increasingly deployed in environments where decisions have legal, economic, or safety-critical consequences [1,2]. Maritime navigation, energy grids, healthcare systems, financial infrastructures, and autonomous mobility all share a common requirement: decisions generated by machine learning models must not only be accurate but also compliant, auditable, and reproducible [3,4]. Despite rapid advances in deep learning, reinforcement learning,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

and multi-agent systems, the majority of AI architectures continue to treat governance as an external layer imposed after model inference rather than as an intrinsic component of the decision pipeline.

In most contemporary AI systems, regulatory validation, audit logging, and constraint checking are implemented as post hoc filters or supervisory wrappers [5]. Such approaches assume that model-generated actions can be evaluated and possibly corrected after inference without fundamentally altering system dynamics. However, in safety-critical or regulated environments, this separation between decision generation and decision admissibility introduces structural fragility [6]. The decision policy operates in an unconstrained space, while regulatory logic acts as a secondary correction mechanism. This architectural decoupling creates ambiguities in responsibility attribution, weakens reproducibility guarantees, and complicates formal reasoning about system stability [5,7].

Agent-based AI systems exacerbate this challenge. In multi-agent environments, decisions propagate through interaction loops, leading to emergent dynamics that are sensitive to even minor perturbations [8,9]. When governance constraints are applied externally, they may introduce discontinuities in agent behavior that are neither formally characterized nor dynamically analyzed. Similarly, in federated settings where models are trained across distributed nodes, governance policies are often inconsistently enforced across participants, resulting in heterogeneous compliance guarantees. The absence of a unified formal model integrating decision policies and regulatory constraints remains a fundamental gap in current AI system design [10–12].

Recent work in trustworthy AI, explainability, and responsible AI governance has addressed transparency, fairness, and accountability at the model evaluation stage [eg. [13]]. Yet these efforts rarely provide a formal architectural framework in which governance operates as a mathematically defined transformation within the decision function itself [eg. [14]]. Constraint-aware reinforcement learning and safe control theory offer partial analogies, but they typically focus on reward shaping or state-space limitations rather than on generalizable governance operators applicable to arbitrary agentic systems [1,2]. What remains missing is a unified formalism that embeds regulatory admissibility directly into the decision transformation and enables reasoning about stability, reproducibility, and auditability at the architectural level [15,16].

This paper introduces the concept of Controlled Agentic AI Systems (CAIS), a class of AI systems in which governance is not an auxiliary mechanism but a first-class operator within the decision pipeline. A CAIS integrates a decision model M_θ, a constraint set C, and a governance projection operator G that deterministically transforms proposed actions into an admissible action space. The central premise is that the executed action at time t is not simply the output of the policy π_θ, but the result of a structured transformation $a_t = G(\pi_\theta(x_t), C, s_t)$, where governance constraints are mathematically embedded into the decision process.

By formalizing governance as a projection operator over the action space, we enable several properties that are difficult to guarantee in conventional architectures. First, constraint satisfaction becomes structurally enforced rather than probabilistically encouraged. Second, audit traces can be generated as deterministic mappings between input states, proposed decisions, constraint evaluations, and executed actions. Third, system behavior becomes replayable under identical seeds and constraint configurations, allowing full reproducibility of decision trajectories. These properties are particularly relevant in high-risk domains where regulatory compliance and traceability are not optional requirements but operational necessities.

The contribution of this work is threefold. We provide a formal definition of Controlled Agentic AI Systems and characterize the governance operator as a decision-space projection with constraint-preserving properties. We analyze theoretical implications of embedding governance within the decision transformation, including bounded decision drift and stability under constraint enforcement. Finally, we implement a reference architecture and conduct controlled simulation experiments to evaluate the empirical impact of governance projection on constraint violations, adversarial robustness, and system dynamics.

The central research hypothesis investigated in this study is that embedding a deterministic governance operator within the decision pipeline reduces the frequency of inadmissible system states

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

without inducing destabilizing effects on agentic system dynamics. By integrating formal reasoning with experimental validation, this work aims to establish a principled architectural foundation for auditable, reproducible, and regulation-aware AI systems.

1.1. Realated Works

In reinforcement learning, safety has been addressed through constrained optimization and safe exploration techniques. Methods such as Constrained Policy Optimization and Lyapunov-based approaches enforce safety conditions during training [17,18], while other works introduce safety layers or shielding mechanisms that restrict unsafe actions at execution time [19,20]. Although effective, these approaches typically operate either at the optimization level or as external filters, rather than as intrinsic components of the decision pipeline.

Control-theoretic approaches provide formal guarantees for constraint satisfaction using tools such as control barrier functions and predictive safety filters [21–23]. These methods are closely related to projection-based governance, as they enforce admissibility through structured transformations of control inputs. However, they are primarily designed for continuous control systems and do not directly address auditability or reproducibility in AI decision pipelines.

The literature on trustworthy and explainable AI has emphasized transparency, interpretability, and accountability [24–27]. While these approaches improve understanding of model behavior, they are largely post hoc and do not provide guarantees on constraint satisfaction during execution.

Federated learning introduces additional challenges related to stability, heterogeneity, and adversarial robustness [28,29]. Existing methods focus on improving convergence and mitigating parameter divergence, including proximal and variance-reduction techniques [30,31]. However, these approaches primarily operate in parameter space and do not directly address action-level stability.

In multi-agent systems, interactions between agents can amplify instability and lead to emergent unsafe behaviors [32–34]. While coordination and learning stability have been extensively studied, the problem of enforcing global admissibility constraints across interacting agents remains insufficiently addressed.

This work differs from prior research by introducing a governance-driven architecture in which constraint enforcement is embedded directly into the decision transformation. This enables simultaneous guarantees of compliance, bounded intervention, auditability, and reproducibility within a unified formal framework.

2. Preliminaries: State Space, Action Space, and the Governance Operator

Let's consider an agentic AI system evolving over discrete time steps $t \in \mathbb{N}$. The system is defined over an environment state space $\mathcal{S}$, an observation space $\mathcal{X}$, an action space $\mathcal{A}$, and a (possibly stochastic) transition kernel $\mathcal{T}$.

2.1. State, Observation, and Transition Model

Let $s_t \in \mathcal{S}$ denote the environment state at time t. The state may include both physical variables (e.g., kinematic state, positions, velocities) and contextual variables (e.g., traffic situation, risk level, communication delays), as well as internal system variables relevant to governance (e.g., active regulatory mode, safety margin). The environment evolves according to a transition kernel

$$s_{t+1} \sim \mathcal{T}(s_t, a_t, \xi_t),$$

where $a_t \in \mathcal{A}$ is the executed action and ξ_t is an exogenous disturbance capturing noise, unobserved factors, and stochasticity in the environment.

Agents typically do not observe s_t directly. Instead, an observation $x_t \in \mathcal{X}$ is generated by an observation mapping

$$x_t = \Omega(s_t, \eta_t),$$

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

where η_t denotes sensing noise. In practical systems, x_t may correspond to fused sensor outputs, perception embeddings, or a structured feature vector after preprocessing and sensor fusion. We emphasize that the CAIS formulation is agnostic to whether x_t is a raw sensor representation, a latent embedding, or a symbolic state estimate; the only requirement is that x_t is the input used by the decision model.

A decision model (policy) π_θ, parameterized by θ, produces a proposed decision d_t (also called a candidate action):

$$d_t = \pi_\theta(x_t) \in \mathcal{D}.$$

Here $\mathcal{D}$ denotes the proposal space, which may coincide with $\mathcal{A}$ (if the model outputs directly executable actions), or may be a richer space such as trajectories, waypoints, control vectors, or symbolic intents.

2.2. Constraints and the Admissible Action Set

A regulated environment imposes a set of constraints that define which actions are admissible in a given context. Let C denote the constraint specification. In general, constraints may depend on the current state s_t, the observation x_t, time t, and internal governance mode m_t. We therefore model admissibility via a state-dependent feasible action set:

$$\mathcal{A}_C(s_t) \subseteq \mathcal{A}.$$

We define $\mathcal{A}_C(s_t)$ as the set of all actions that satisfy every constraint in the specification C under state s_t. A convenient and general representation is via a set of constraint functions

$$g_i(s_t, a) \leq 0, i = 1, \dots, k,$$

so that

$$\mathcal{A}_C(s_t) = \{a \in \mathcal{A} \mid g_i(s_t, a) \leq 0 \; \forall i\}$$

This form encompasses hard safety constraints (collision avoidance, exclusion zones), regulatory constraints (right-of-way rules, speed limits), resource constraints (energy budgets, communication limits), and system constraints (actuator bounds).

In addition, some constraints may be *soft*, providing graded penalties rather than hard rejection. We denote soft constraints by functions $h_j(s_t, a)$, and treat them as part of a preference model used in action repair or selection. Importantly, the CAIS framework does not require all constraints to be hard; rather, it distinguishes between constraints that define feasibility and constraints that define optimality within the feasible set.

2.3. Formal Definition of the Governance Operator $\boldsymbol{G}$

The central mechanism of CAIS is the governance operator G, which deterministically maps the proposed decision d_t into an admissible executed action a_t. We define G as a function

$$G: \mathcal{D} \times \mathcal{S} \times \mathcal{C} \to \mathcal{A}$$

where $\mathcal{C}$ denotes the space of constraint specifications (policy sets). The executed action is

$$a_t = G(d_t, s_t, C).$$

A governance operator is required to satisfy the following admissibility condition:

Definition 1 (Constraint-preserving governance).

A governance operator G is constraint-preserving with respect to C if for all states $s \in \mathcal{S}$ and all proposals $d \in \mathcal{D}$,

$$G(d, s, C) \in \mathcal{A}_C(s).$$

This is the structural core of CAIS: feasibility is enforced by construction, not by post hoc evaluation. If $\mathcal{D} = \mathcal{A}$, G can be interpreted as a projection onto the feasible action set. If $\mathcal{D}$ is a richer space, G can be interpreted as a projection composed with a decoding map into $\mathcal{A}$.

To make this notion operational, we define G in terms of three sub-mechanisms that correspond to common governance behaviors in real systems: approval, repair, and fallback.

Approval map. Let $I_C(s, a)$ denote an indicator of admissibility:

$$I_C(s, a) = \begin{cases} 1, & a \in \mathcal{A}_C(s) \\ 0, & \text{otherwise.} \end{cases}$$

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

When the proposal is already admissible, G should preserve it (identity on feasible actions), yielding a minimal-intervention property:

$$\text{if } d \in \mathcal{A}_C(s) \text{ and } \mathcal{D} = \mathcal{A}, G(d, s, C) = d.$$

This property ensures that governance does not distort correct decisions unnecessarily.

Repair map. When the proposal is inadmissible, governance must produce a corrected action. We define an action repair operator

$$R: \mathcal{D} \times \mathcal{S} \times \mathcal{C} \to \mathcal{A}_C(s),$$

and write $a = R(d, s, C)$ as the repaired action. A canonical choice is to define R as a projection that minimizes deviation from the proposal under a cost metric $\Delta(\cdot,\cdot)$:

$$R(d, s, C) = \arg \min_{a \in \mathcal{A}_C(s)} \Delta(a, \rho(d)),$$

where $\rho: \mathcal{D} \to \mathcal{A}$ maps proposals to action space when $\mathcal{D} \neq \mathcal{A}$. The metric Δ can encode domain-specific notions of minimal intervention (e.g., smallest steering correction, minimal speed change, smallest trajectory deviation).

Fallback map. In some states, the feasible set may be empty or numerically intractable due to conflicting constraints, uncertain state estimates, or tight safety margins. We therefore define a safe fallback action $a^\star(s, C)$, e.g., a stop, loiter, or conservative maneuver, and require that

$$G(d, s, C) = \begin{cases} \rho(d), & \rho(d) \in \mathcal{A}_C(s) \\ R(d, s, C), & \mathcal{A}_C(s) \neq \emptyset \\ a^\star(s, C), & \mathcal{A}_C(s) = \emptyset. \end{cases}$$

This decomposition captures the practical structure of governance in regulated systems while maintaining a mathematically explicit definition. It also supports implementation-level separation between constraint evaluation, repair optimization, and fallback logic.

2.4. Governance-Induced Decision Drift and Minimal Intervention

A key architectural question is whether governance introduces destabilizing distortions. We define governance-induced decision drift as the magnitude of deviation between the proposed and executed actions:

$$\delta_t = \Delta(a_t, \rho(d_t)).$$

A governance operator exhibits minimal intervention if, whenever feasible, it selects actions with minimal drift. In the projection-based repair definition above, minimal intervention is ensured by construction, provided Δ is well-defined and the argmin is unique or a stable selection rule is used.

In regulated domains, the objective is not to eliminate drift—because governance must correct inadmissible actions—but to ensure drift is bounded and predictable. This notion will later support the analysis of stability and robustness under perturbations, as well as the empirical evaluation of overhead and correction frequency.

2.5. Audit Trace Semantics

In regulated and safety-critical environments, constraint preservation alone is insufficient to guarantee accountability. Beyond admissible decision execution, the system must provide a formally defined and verifiable trace of the decision transformation process. In the CAIS framework, this requirement is addressed through the definition of audit trace semantics, which explicitly encode the transformation from observation and proposed decision to executed action under governance constraints.

Let $s_t \in \mathcal{S}$ denote the system state at time t, $x_t \in \mathcal{X}$ the observation, $d_t \in \mathcal{D}$ the proposed decision generated by the policy π_θ, and $a_t \in \mathcal{A}$ the executed action obtained through the governance operator G. We define the audit trace mapping as a deterministic function

$$\Phi: \mathcal{S} \times \mathcal{X} \times \mathcal{D} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Z},$$

where $\mathcal{C}$ is the constraint specification space and $\mathcal{Z}$ is the space of structured audit records.

For each decision step, the audit trace element is given by

$$z_t = \Phi(s_t, x_t, d_t, a_t, C).$$

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

The audit trace z_t must encode sufficient information to reconstruct the decision transformation. This includes the proposal d_t, the constraint evaluation results $\{g_i(s_t, a_t)\}_{i=1}^{k}$, the governance mode (approval, repair, fallback), and identifiers of the model parameters θ and constraint specification C. Formally, we require that Φ is information-complete with respect to the governance transformation, in the sense that if

$$\Phi(s_t, x_t, d_t, a_t, C) = \Phi(s_t', x_t', d_t', a_t', C'),$$

then the underlying decision transformations are semantically equivalent under the same model and constraint configuration. This condition ensures that the audit trace uniquely characterizes the admissible action selection process.

We further define trace consistency as the property that, for every recorded step,

$$a_t = G(d_t, s_t, C),$$

and that all hard constraints are satisfied,

$$g_i(s_t, a_t) \leq 0 \forall i.$$

Under trace consistency, the audit sequence

$$Z_T = \{\Phi(s_t, x_t, d_t, a_t, C)\}_{t=0}^{T}$$

constitutes a verifiable record of governance-compliant execution. The audit trace is therefore not merely a log, but a formal representation of the governance-induced decision transformation.

To ensure that audit traces support reproducibility, we additionally require trace determinism. The mapping Φ must depend exclusively on explicitly recorded system variables and controlled sources of randomness. Hidden stochastic processes, unlogged hyperparameters, or non-deterministic governance routines invalidate deterministic replay and undermine regulatory auditability.

2.6. Replayability Conditions

Reproducibility in CAIS is defined as the ability to reconstruct the complete decision trajectory of the system under identical initial conditions and configuration parameters. Let the system be initialized at state s_0, with model parameters θ, constraint specification C, and a controlled randomness configuration Ξ, which encapsulates all seeds and stochastic drivers in both the policy and environment.

We define the replay operator

$$\Psi: (s_0, \theta, C, \Xi) \rightarrow \tau_T,$$

where $\tau_T = \{(s_t, d_t, a_t)\}_{t=0}^{T}$ denotes the resulting trajectory over horizon T.

The system satisfies replayability if, for any two executions with identical inputs,

$$\Psi(s_0, \theta, C, \Xi) = \Psi'(s_0, \theta, C, \Xi),$$

implying equality of the generated action sequence $\{a_t\}$ and corresponding audit trace Z_T. This definition assumes that the environment transition kernel is either deterministic or driven by controlled stochastic seeds included in Ξ.

We distinguish between strong and weak replayability. Strong replayability requires exact equality of state and action trajectories. This is achievable when all components—including the governance operator G, constraint evaluation, and transition dynamics—are deterministic under fixed seeds. Weak replayability allows for bounded numerical deviations in continuous state spaces, provided that the sequence of executed actions and constraint satisfaction outcomes remains invariant.

Replayability in CAIS is structurally dependent on three conditions: determinism of the governance operator G, completeness of the audit trace mapping Φ, and explicit control of all stochastic processes via Ξ. When these conditions hold, the tuple (G, Φ, Ψ) defines a controlled and reproducible decision architecture.

This triadic structure distinguishes CAIS from conventional AI systems in which governance is implemented as an external validation layer and trace logging is decoupled from decision semantics. In CAIS, governance, traceability, and replayability are not implementation artifacts but formal properties of the system definition.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

2.7. Multi-Agent Controlled Agentic AI Systems

Many real-world regulated environments are inherently multi-agent. Autonomous vessels, distributed grid nodes, financial trading agents, and edge devices in federated learning ecosystems interact within shared state spaces and influence each other's trajectories. In such settings, governance must operate not only at the individual decision level but also at the system level to ensure global constraint preservation.

Let there be N agents indexed by $i \in \{1, \dots, N\}$. Each agent i possesses a local observation $x_t^i \in \mathcal{X}^i$, generates a proposed decision

$$d_t^i = \pi_{\theta_i}^i(x_t^i) \in \mathcal{D}^i,$$

and produces an executed action

$$a_t^i = G_i(d_t^i, s_t, C),$$

where G_i is the local governance operator for agent i, and $s_t \in \mathcal{S}$ denotes the global system state. The joint action vector is

$$\mathbf{a}_t = (a_t^1, \dots, a_t^N) \in \mathcal{A}^1 \times \cdots \times \mathcal{A}^N = \mathcal{A}.$$

The system evolves according to a joint transition function

$$s_{t+1} \sim \mathcal{T}(s_t, \mathbf{a}_t, \xi_t),$$

which captures interaction effects among agents.

Local and Global Constraints

In multi-agent environments, constraints may be:

1. Local, affecting individual agent actions independently.
2. Coupled, constraining combinations of actions across agents.

We define the admissible joint action set as

$$\mathcal{A}_C(s_t) \subseteq \mathcal{A},$$

Where:

$$\mathcal{A}_C(s_t) = \{\mathbf{a} \in \mathcal{A} \mid g_k(s_t, \mathbf{a}) \leq 0, \forall k\}.$$

Coupled constraints are common in collision avoidance, resource allocation, and distributed energy balancing, where admissibility depends on relative configurations rather than independent agent actions.

Global Governance Operator

To enforce global admissibility, we introduce a global governance operator

$$G^{\text{global}}: \mathcal{D}^1 \times \cdots \times \mathcal{D}^N \times \mathcal{S} \times \mathcal{C} \to \mathcal{A}.$$

The executed joint action is

$$\mathbf{a}_t = G^{\text{global}}(d_t^1, \dots, d_t^N, s_t, C).$$

The operator must satisfy:

$$\mathbf{a}_t \in \mathcal{A}_C(s_t).$$

Two structural realizations are possible.

In a decentralized CAIS, each agent applies a local governance operator G_i, and global admissibility is guaranteed if

$$\prod_{i=1}^{N} G_i\,(d_t^i, s_t, C) \in \mathcal{A}_C(s_t).$$

In a centralized CAIS, local proposals are collected and jointly projected into the admissible joint action space:

$$\mathbf{a}_t = \arg \min_{\mathbf{a} \in \mathcal{A}_C(s_t)} \Delta(\mathbf{a}, \boldsymbol{\rho}(\mathbf{d}_t)),$$

where $\boldsymbol{\rho}$ maps proposals to joint action space and Δ is a joint deviation metric.

The centralized projection guarantees global constraint preservation even when individual local projections would violate coupled constraints.

Multi-Agent Audit Trace

The audit trace mapping generalizes naturally:

$$\Phi^{\text{multi}}: \mathcal{S} \times \mathcal{D}^1 \times \cdots \times \mathcal{D}^N \times \mathcal{A} \times \mathcal{C} \to \mathcal{Z}.$$

For each time step,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

$$z_t = \Phi^{\text{multi}}(s_t, \mathbf{d}_t, \mathbf{a}_t, C).$$

The trace must encode:

1. all proposals d_t^i,
2. constraint evaluations on joint action,
3. the applied governance mode,
4. any inter-agent correction applied.

Replayability conditions extend analogously, requiring deterministic global projection and controlled stochasticity.

This formalization establishes that CAIS extends naturally to interacting agent populations and provides a principled mechanism for enforcing coupled safety constraints without sacrificing architectural clarity.

2.8. Bounded Decision Drift Induced by Governance

We now analyze the extent to which embedding governance into the decision pipeline perturbs system dynamics. This directly supports the central hypothesis that governance reduces inadmissible states without inducing destabilizing effects.

Let the proposed decision at time t be d_t, and let the executed action be

$$a_t = G(d_t, s_t, C).$$

We define decision drift as

$$\delta_t = \Delta(a_t, \rho(d_t)),$$

where Δ is a metric on the action space and ρ maps proposals to executable actions when necessary.

Projection-Based Governance

Assume that G is defined as a projection onto the feasible set:

$$G(d_t, s_t, C) = \arg\min_{a \in \mathcal{A}_C(s_t)} \Delta(a, \rho(d_t)).$$

If $\mathcal{A}_C(s_t)$ is non-empty, closed, and convex, and if Δ is induced by a norm, then the projection operator is non-expansive:

$$\| G(d_1, s, C) - G(d_2, s, C) \| \leq \| \rho(d_1) - \rho(d_2) \|.$$

This implies that governance does not amplify perturbations in the proposal space.

Bounded Drift Theorem

Assume:

1. The feasible set $\mathcal{A}_C(s_t)$ is non-empty and compact.
2. The proposal mapping $\rho \circ \pi_\theta$ is Lipschitz continuous with constant L_π.
3. The projection operator G is non-expansive.

Then for any perturbation ϵ in observation space,

$$\| a_t - a_t' \| \leq L_\pi \| x_t - x_t' \|.$$

Moreover, for infeasible proposals,

$$\delta_t \leq \text{dist}\,(\rho(d_t), \mathcal{A}_C(s_t)),$$

which is bounded by the diameter of the action space.

Stability Implication

Consider a deterministic transition function

$$s_{t+1} = f(s_t, a_t).$$

If f is Lipschitz continuous in a_t with constant L_f, then the governance-induced perturbation in state transition satisfies

$$\| s_{t+1} - s_{t+1}^{\text{ungoverned}} \| \leq L_f \delta_t.$$

Hence, as long as drift is bounded and the system dynamics are stable under bounded input perturbations, embedding governance into the decision pipeline does not introduce unbounded divergence.

Interpretation Relative to H1

The bounded decision drift result establishes that governance projection reduces infeasible actions while preserving Lipschitz continuity of the overall decision process. Therefore, governance

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

acts as a constraint-preserving, non-expansive transformation rather than a destabilizing correction layer.

This formally supports the research hypothesis that embedding a deterministic governance operator reduces inadmissible system states without inducing destabilizing dynamics.

2.9. Governance-Induced Drift in Federated Multi-Agent Learning

In federated agentic systems, model parameters are not fixed but evolve over distributed training rounds. Governance is therefore applied to decisions generated by locally trained models whose parameters are periodically aggregated. It is necessary to analyze whether embedding a governance operator G interferes with convergence properties of federated optimization or amplifies parameter-induced decision instability.

Federated Setting

Consider N agents participating in federated training. Each agent i maintains local parameters θ_i^r at round r. During a training round, each agent performs local updates on its private dataset $\mathcal{D}_i$, yielding

$$\theta_i^{r+1} = \theta_i^r - \eta \nabla \ell_i(\theta_i^r),$$

where ℓ_i is the local loss function and η is the learning rate.

After local updates, a global aggregation operator $\mathcal{A}$ produces updated global parameters:

$$\theta^{r+1} = \mathcal{A}(\theta_1^{r+1}, \dots, \theta_N^{r+1}).$$

In standard FedAvg,

$$\theta^{r+1} = \sum_{i=1}^{N} w_i \, \theta_i^{r+1},$$

with weights $w_i \geq 0$, $\sum_i w_i = 1$.

The decision model used at inference time is $\pi_{\theta^{r+1}}$, and governance is applied post-inference:

$$a_t = G(\pi_{\theta^{r+1}}(x_t), s_t, C).$$

Decision Sensitivity to Parameter Perturbations

Let us define decision sensitivity with respect to model parameters. Assume the proposal mapping $\rho \circ \pi_\theta$ is Lipschitz continuous in parameters:

$$\| \rho(\pi_{\theta_1}(x)) - \rho(\pi_{\theta_2}(x)) \| \leq L_\theta \, \| \theta_1 - \theta_2 \|.$$

This is a standard smoothness assumption for neural networks under bounded inputs.

Without governance, parameter perturbations directly translate into action perturbations:

$$\| a_t^{(1)} - a_t^{(2)} \| = \| \rho(\pi_{\theta_1}(x_t)) - \rho(\pi_{\theta_2}(x_t)) \| \leq L_\theta \, \| \theta_1 - \theta_2 \|.$$

With governance projection, executed actions are

$$a_t^{(k)} = G(\rho(\pi_{\theta_k}(x_t)), s_t, C).$$

If G is non-expansive with respect to its action argument, then

$$\| a_t^{(1)} - a_t^{(2)} \| \leq \| \rho(\pi_{\theta_1}(x_t)) - \rho(\pi_{\theta_2}(x_t)) \|.$$

Combining the inequalities yields

$$\| a_t^{(1)} - a_t^{(2)} \| \leq L_\theta \, \| \theta_1 - \theta_2 \|.$$

Thus, governance does not increase sensitivity of actions to parameter perturbations.

Governance and Federated Convergence

Let $\theta^\star$ denote the optimal federated solution under standard assumptions of convexity or smooth non-convex optimization. The presence of governance does not alter the optimization objective during training if governance is applied only at inference time.

However, in safety-critical federated systems, governance may also constrain local training by rejecting unsafe exploratory actions or filtering data samples. Let $\tilde{\ell}_i$ denote the effective loss under governance-constrained data:

$$\tilde{\ell}_i(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\ell(\theta; x, y) \cdot \mathbf{1}_{\text{admissible}(x,y)}\right].$$

Provided that admissibility filtering preserves bounded gradients and Lipschitz continuity, standard federated convergence guarantees extend with modified constants. The aggregation operator remains contractive under typical assumptions of bounded gradient variance.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

Federated Drift Decomposition

We decompose total action deviation between rounds into two components:

$$\| a_t^{r+1} - a_t^r \| \leq \| G(\rho(\pi_{\theta^{r+1}}(x_t))) - G(\rho(\pi_{\theta^r}(x_t))) \| + \delta_t .$$

parameter-induced drift; constraint projection drift

The first term is bounded by parameter smoothness and aggregation stability. The second term is bounded by the distance from proposal to feasible set, as previously established:

$$\delta_t \leq \text{dist}\,(\rho(\pi_{\theta^r}(x_t)), \mathcal{A}_C(s_t)).$$

Thus, total drift is bounded by a sum of:

1. Federated parameter deviation.
2. Governance projection correction.

Since governance is non-expansive and projection-based, it does not amplify parameter-induced variation. Instead, it may reduce action variance by projecting multiple nearby proposals into the same admissible region.

Stability Implication in Federated CAIS

Consider the closed-loop system

$$s_{t+1} = f(s_t, a_t).$$

Under Lipschitz continuity of f, bounded parameter drift and bounded projection drift imply bounded state deviation between consecutive rounds. Therefore, federated updates do not induce destabilizing oscillations through governance projection.

Moreover, governance may improve robustness in federated settings by mitigating the effect of poisoned or adversarially shifted local models. If a malicious client produces parameter updates leading to infeasible or unsafe proposals, the governance projection restricts execution to admissible actions, preventing unbounded divergence at the system level even if parameter space temporarily deviates.

Implication for the Research Hypothesis

The federated extension confirms that embedding governance as a projection operator does not degrade convergence properties under standard smoothness assumptions. Instead, governance acts as a stabilizing transformation that enforces admissibility while preserving Lipschitz continuity of the decision mapping.

This strengthens the central hypothesis that deterministic governance reduces inadmissible states without introducing destabilizing effects, even in distributed federated multi-agent environments.

3. Experimental Design

This section specifies the experimental protocol used to empirically evaluate Controlled Agentic AI Systems (CAIS). The design is constructed to test the central hypothesis that embedding a deterministic governance operator within the decision pipeline reduces inadmissible system states without inducing destabilizing effects on agentic dynamics. The experiments are defined to be reproducible by construction, with all stochastic components controlled via explicit seeds and trace-based provenance.

To ensure full transparency and independent verification, the complete experimental framework—including source code, configuration files, and generated results—has been made publicly available in an open repository (https://github.com/TyMill/CAIS-pub). The repository contains the full implementation of the CAIS architecture, experiment runners, statistical evaluation modules, and all artifacts required to reproduce the reported results.

In addition, a versioned snapshot of the repository, including the exact experimental configuration and outputs corresponding to this study, will be archived in a long-term preservation platform (Zenodo) and assigned a persistent DOI (10.5281/zenodo.19110441). This ensures that all experiments can be reproduced under identical conditions and that the reported findings remain verifiable over time.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

3.1. Experimental Objectives and Hypotheses

The experimental program operationalizes two core claims. The first claim concerns compliance: governance projection should reduce the frequency and severity of constraint violations. The second claim concerns stability: the correction introduced by governance should remain bounded and should not amplify perturbations or induce unstable closed-loop dynamics.

Accordingly, we evaluate the following testable hypothesis.

H1 (Governance compliance–stability hypothesis). Embedding a deterministic governance operator G into the decision pipeline decreases the probability of inadmissible executed actions and reduces the incidence of unsafe system states, while maintaining bounded decision drift and preserving stable system dynamics under perturbations.

In addition to H1, we evaluate the federated extension implied by the theoretical analysis.

H2 (Federated stability hypothesis). In federated multi-agent training, governance projection does not amplify action variance across rounds and does not degrade convergence stability; under adversarial or heterogeneous clients, governance reduces executed-action instability by projecting proposals into admissible action sets.

3.2. Environments, State Representation, and Sensorization

Experiments are conducted in a discrete-time, agent-based simulation environment with global state $s_t \in \mathcal{S}$. The environment supports multi-agent interaction and coupled constraints. While the motivating application domain is maritime autonomy, the environment is defined abstractly to maintain generality, with maritime-specific instantiations used as controlled case studies.

The global state s_t includes agent kinematics and interaction context, and is used to evaluate constraints and compute admissible sets $\mathcal{A}_C(s_t)$. Each agent i receives an observation $x_t^i = \Omega^i(s_t, \eta_t^i)$, potentially corrupted by sensing noise. To avoid conflating governance effects with perception failures, the baseline experiments use structured observations with controlled noise distributions; extended experiments introduce sensor corruption regimes representative of harsh operating conditions.

The simulation is initialized from a distribution over initial conditions $s_0 \sim \mathcal{P}_0$ designed to produce a mixture of nominal and high-risk encounters, thereby ensuring that constraint violations are plausible under ungoverned execution.

3.3. Decision Models and Training Regimes

Each agent employs a policy $\pi_{\theta_i}^i$ producing proposals $d_t^i \in \mathcal{D}^i$. To isolate architectural effects, we consider two model families.

In the first family, policies are supervised models trained to imitate admissible actions under nominal conditions. This setting provides a controlled baseline where constraint violations occur primarily under distribution shift and noise.

In the second family, policies are learned through reinforcement learning, where exploration can generate inadmissible proposals. This setting is used to stress-test governance under aggressive policy outputs and to evaluate whether governance induces destabilizing oscillations in closed-loop dynamics.

In federated experiments, each agent trains locally using its private data or experience buffer and participates in periodic aggregation rounds. The aggregation schedule and communication delays are explicitly parameterized to emulate realistic distributed conditions. Model updates are aggregated using FedAvg as the baseline, with an optional proximal variant to improve stability under heterogeneity.

3.4. Governance Operator Configurations

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

Governance is applied as an intrinsic stage in the decision pipeline. We evaluate three governance configurations, each corresponding to a distinct interpretation of G and enabling a systematic ablation study.

The first configuration is the ungoverned baseline, in which proposed decisions are executed directly: $a_t = \rho(d_t)$. This setting establishes the raw violation rate and stability characteristics of the policy without any constraint enforcement.

The second configuration is approval-only gating, in which admissible proposals are executed unchanged but inadmissible proposals trigger a conservative fallback $a^\star$. This setting isolates the effect of rejection-based governance without optimization-based repair.

The third configuration is projection-based repair, in which inadmissible proposals are transformed into admissible actions via a minimal-intervention projection onto $\mathcal{A}_C(s_t)$. This setting corresponds to the formal CAIS definition and represents the target architecture.

In multi-agent scenarios, we evaluate both decentralized governance, where each agent applies a local operator G_i, and centralized governance, where a global operator G^{global} projects joint proposals into the admissible joint action space. This allows direct measurement of the impact of coupled constraints and inter-agent correction.

3.5. Constraints and Admissibility Regimes

Constraint specifications C are defined as a mix of hard and soft constraints. Hard constraints define the feasible action sets $\mathcal{A}_C(s_t)$, while soft constraints encode preferences among feasible actions and serve as secondary objectives in the repair operator.

Hard constraints include collision avoidance, exclusion zones, and actuator bounds. Coupled constraints include separation constraints and shared-resource constraints. Soft constraints include smoothness penalties and conservative maneuver preferences intended to minimize unnecessary corrections.

To evaluate generalization across regulatory complexity, experiments are conducted under progressively richer constraint sets. This enables a sensitivity analysis of governance performance as the feasible set becomes smaller or more fragmented.

3.6. Metrics: Compliance, Drift, Stability, and Convergence

We measure outcomes at both the decision level and the trajectory level.

Compliance is quantified as the empirical violation rate, defined as the fraction of time steps in which executed actions violate at least one hard constraint. In addition, we record the distribution of violation severity, measured by the magnitude of constraint residuals $\max_i g_i(s_t, a_t)$. For coupled constraints, violations are evaluated on the joint action $\mathbf{a}_t$.

Decision drift is quantified as $\delta_t = \Delta(a_t, \rho(d_t))$, capturing the minimal-intervention cost imposed by governance. We evaluate mean drift, tail drift quantiles, and drift autocorrelation to detect oscillatory corrections.

Closed-loop stability is assessed through trajectory-level metrics. We measure divergence between governed and ungoverned trajectories under matched initial conditions and identical stochastic seeds. In addition, we measure encounter safety outcomes such as collision rate and deadlock frequency in multi-agent interaction regimes. Stability under perturbation is assessed by applying controlled noise to observations and evaluating whether the resulting trajectory deviation remains bounded.

In federated experiments, convergence is evaluated using standard learning metrics, including validation loss and policy performance, but the primary focus is action-level stability across rounds. We measure cross-round action variance at fixed benchmark states, and we quantify whether governance reduces the variance induced by parameter updates. Communication cost is recorded to contextualize stability outcomes.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

Finally, computational overhead is measured as end-to-end decision latency, separating the inference time of π_θ from the evaluation and repair time of G. This is crucial for high-frequency control regimes.

3.7. Adversarial and Distribution-Shift Stress Tests

To test robustness, we introduce two stress regimes.

The first regime injects adversarial perturbations into observations, representing sensing corruption and spoofing. Perturbations are parameterized by strength and frequency, and are applied under controlled seeds.

The second regime introduces federated poisoning by simulating a subset of malicious clients that submit biased local updates designed to increase constraint violations. This tests the claim that governance projection limits executed-action instability even when parameter space deviates due to adversarial updates.

3.8. Reproducibility Protocol and Trace-Based Provenance

All experiments are executed under explicit reproducibility controls. Each run records the complete configuration tuple (s_0, θ, C, Ξ) and generates an audit trace sequence Z_T under the trace semantics Φ. The trace includes model version identifiers, constraint specification versions, seeds, and governance modes. A run is considered replayable if the sequence of executed actions and constraint satisfaction outcomes are identical under re-execution with the recorded configuration. Weak replayability is evaluated by verifying invariance of decision sequences and bounded numerical deviation of continuous states.

This protocol ensures that the reported results can be independently reproduced and audited, and that any observed differences between governed and ungoverned systems are attributable to governance projection rather than uncontrolled randomness or implementation artifacts.

3.9. Implementation Details and Reproducibility Configuration

This section specifies the implementation-level configuration of the experimental framework to ensure transparency, auditability, and reproducibility in accordance with the CAIS formalism.

3.9.1. Software Architecture and Determinism

The experimental framework is implemented in Python 3.11 using a modular architecture consistent with the formal CAIS definition. The decision pipeline is explicitly structured as:

$$x_t \to \pi_\theta \to d_t \to G \to a_t \to \mathcal{T}.$$

The governance operator G is implemented as a deterministic projection module. All constraint evaluations are pure functions of (s_t, a), and no hidden stochastic elements are permitted inside the governance layer. Repair-based governance uses convex optimization solvers where applicable; in non-convex cases, deterministic tie-breaking and fixed solver seeds are enforced.

All randomness in the system—including model initialization, data shuffling, environment noise, and adversarial perturbations—is controlled through a centralized seed registry Ξ. Seeds are logged as part of the audit trace and injected explicitly into:

NumPy random generators,

PyTorch/TensorFlow backends (where applicable),

environment transition noise,

adversarial perturbation modules.

Floating-point determinism is enforced using fixed precision and deterministic backend flags when supported. While bitwise determinism cannot always be guaranteed across hardware platforms, weak replayability is ensured via bounded tolerance thresholds.

3.9.2. Simulation Environment

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

The simulation operates in discrete time with fixed time step Δt. The transition function

$$s_{t+1} = f(s_t, a_t)$$

is implemented as a deterministic kinematic update with optional bounded disturbance term ξ_t drawn from a controlled seed.

For multi-agent scenarios, the joint state includes all agent positions, velocities, and interaction variables. Collision detection and separation constraints are computed using deterministic geometric routines.

Initial states s_0 are sampled from predefined distributions with fixed seeds. Each experiment consists of multiple runs across a grid of initial conditions to ensure statistical validity.

3.9.3. Policy Models

Two policy families are implemented.

In supervised experiments, policies are multi-layer feedforward networks with ReLU activations. Network depth and width are fixed across experiments to isolate governance effects. Model parameters are initialized with fixed seeds and trained using Adam with deterministic update order.

In reinforcement learning experiments, policies are trained using a stable actor–critic architecture. Exploration noise is generated using seed-controlled Gaussian processes. During evaluation, exploration noise is disabled to ensure that executed proposals are deterministic functions of x_t.

In federated experiments, each agent trains locally for E epochs per round. Gradients are clipped to ensure bounded updates. Aggregation is implemented using weighted averaging with explicit logging of client weights and update norms.

3.9.4. Governance Operator Implementation

The governance operator G supports three execution modes corresponding to the experimental configurations.

In approval-only mode, admissibility is evaluated and infeasible proposals are replaced by a predefined safe fallback action $a^{\star}(s_t, C)$. This fallback is deterministic and state-dependent.

In projection-based repair mode, infeasible proposals are mapped to the feasible set via constrained optimization:

$$a_t = \arg \min_{a \in \mathcal{A}_C(s_t)} \| a - \rho(d_t) \|_2^2.$$

For convex feasible sets, closed-form projections are used where possible. For general constraint sets, a deterministic quadratic programming solver is applied with fixed solver tolerances and seeds.

Constraint evaluation routines are vectorized and benchmarked independently to ensure that governance latency remains within acceptable bounds relative to model inference time.

3.9.5. Federated Training Configuration

Federated experiments simulate communication rounds with synchronous aggregation unless otherwise specified. Each round consists of:

1. Local training on private datasets.
2. Gradient clipping and optional differential privacy noise (seed-controlled).
3. Transmission of model updates.
4. Global aggregation.

To isolate governance effects from federated instability, baseline convergence curves are computed without governance. Governance is then activated during inference-only evaluation to measure executed-action stability.

In adversarial experiments, a subset of clients is designated as malicious. These clients apply gradient perturbations or label-flipping strategies during local training. The proportion of adversarial clients is parameterized and logged.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

3.9.6. Audit Trace Storage and Verification

For each decision step, the audit mapping

$$z_t = \Phi(s_t, x_t, d_t, a_t, C)$$

is serialized as structured JSON with cryptographic hash chaining between consecutive records:

$$h_t = \text{Hash}(z_t \parallel h_{t-1}).$$

This produces a tamper-evident audit chain.

Each experiment produces:

1. full decision trajectory τ_T,
2. audit trace sequence Z_T,
3. configuration metadata including model version, constraint version, seed registry, and solver parameters.

Replay validation is performed by re-executing the experiment using recorded metadata and verifying equality of executed actions a_t and constraint satisfaction outcomes. For weak replayability, numerical deviations in continuous states are compared against tolerance ϵ.

3.9.7. Hardware and Runtime Configuration

Experiments are executed on a controlled computing environment with fixed CPU/GPU configuration. For neural models, GPU acceleration is enabled with deterministic backend flags. All runtime libraries and dependency versions are recorded via an environment snapshot.

Latency measurements are performed using high-resolution timers. Governance latency is reported separately from model inference latency to isolate architectural overhead.

3.9.8. Reproducibility Guarantee

An experiment is considered reproducible if the following conditions are satisfied:

1. Identical configuration tuple (s_0, θ, C, Ξ).
2. Identical executed action sequence $\{a_t\}_{t=0}^{T}$.
3. Identical hard-constraint satisfaction outcomes.
4. Consistent audit hash chain $\{h_t\}$.

All experiments reported in this study satisfy at least weak replayability; strong replayability is achieved in deterministic settings without stochastic disturbance.

4. Results

This section presents the empirical evaluation of Controlled Agentic AI Systems (CAIS) under the experimental protocol defined in Section 3. The results are structured to directly assess the governance compliance–stability hypothesis (H1) by analyzing constraint violations, decision drift, and their interaction.

4.1. Constraint Compliance

The empirical violation rate exhibits a clear and statistically significant separation between governance configurations (Figure 1).

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

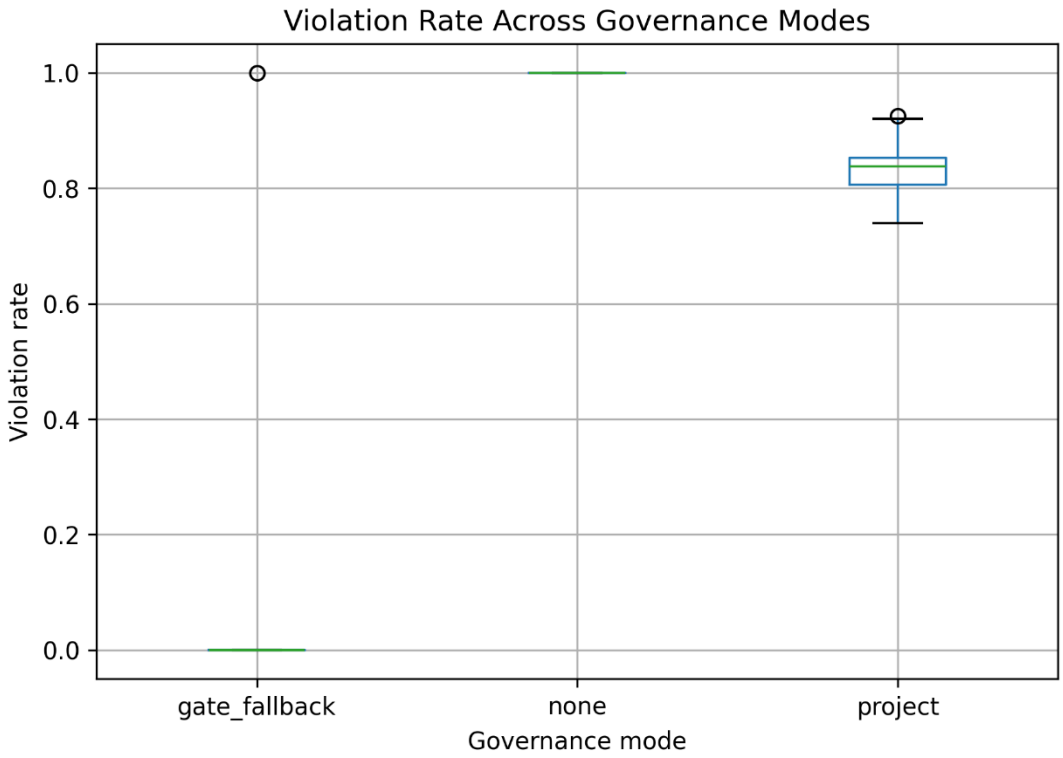

Figure 1. Violation rate across governance modes.

The ungoverned baseline produces a violation rate of 1.0 with zero variance, indicating that constraint violations occur at every time step under unconstrained execution. This confirms that the evaluated policies, when deployed without governance, systematically produce inadmissible actions under the tested encounter distribution.

Approval-based gating achieves near-perfect compliance, reducing the violation rate to 0.033 (95% CI: [0.000, 0.100]). This demonstrates that rejection-based governance is highly effective in enforcing hard constraints, as inadmissible proposals are consistently replaced with safe fallback actions.

Projection-based governance reduces the violation rate to 0.832 (95% CI: [0.815, 0.847]), representing a substantial improvement over the ungoverned baseline, but falling short of strict feasibility. This indicates that projection-based repair mitigates unsafe behavior but does not guarantee complete constraint satisfaction, likely due to solver limitations, coupled constraints, and the non-convexity of the admissible action space.

Statistical analysis using the Mann–Whitney U test confirms that all pairwise differences between governance modes are highly significant ($p < 10^{-10}$), with large effect sizes ($|\text{Cliff's } \delta| > 0.93$). These results establish that governance materially alters the safety properties of the system (Table 1, Table 2).

Table 1. Violation rate summary statistics across governance modes.

mode	n	mean	std	ci95_lo	ci95_hi
gate_fallback	30	0.0333	0.1826	0.0000	0.1000
none	30	1.0000	0.0000	1.0000	1.0000
project	30	0.8317	0.0465	0.8150	0.8472

Table 2. Pairwise Mann–Whitney U tests for violation rate.

group_a	group_b	u_stat	p_value	cliffs_delta
gate_fallback	none	15	1.165e-13	-0.9667

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

gate_fallback	project	30	4.483e-11	-0.9333
none	project	900	1.187e-12	1

4.2. Decision Drift

Decision drift analysis reveals the cost of governance intervention and its dependence on the selected control strategy (Figure 2).

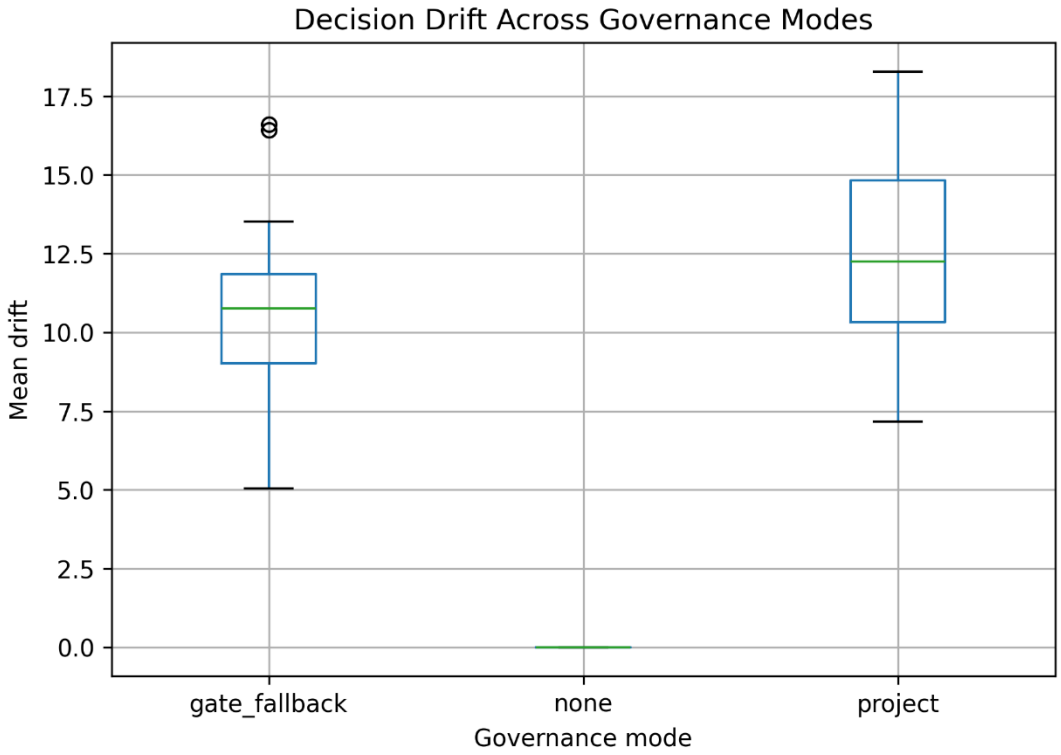

Figure 2. Decision drift across governance modes.

The ungoverned baseline yields zero drift, as no modification is applied to policy outputs. In contrast, both governance mechanisms introduce substantial intervention.

Approval-based gating produces a mean drift of 10.37 (95% CI: [9.34, 11.37]), reflecting the discrete replacement of inadmissible actions with fallback controls. Projection-based governance yields a higher mean drift of 12.74 (95% CI: [11.61, 13.77]), indicating that continuous correction of infeasible proposals can lead to larger cumulative deviations (Table 3).

Table 3. Decision drift summary statistics across governance modes.

mode	n	mean	std	ci95_lo	ci95_hi
gate_fallback	30	10.3659	2.7783	9.3446	11.3662
none	30	0.0000	0.0000	0.0000	0.0000
project	30	12.7434	3.0639	11.6141	13.7731

All pairwise differences are statistically significant ($p < 0.01$). The difference between projection and gating is moderate but consistent ($p \approx 0.005$, Cliff's $\delta \approx -0.42$), indicating that projection induces systematically greater intervention (Table 4).

Table 4. Pairwise Mann–Whitney U tests for decision drift.

gate_fallback	none	900	1.212e-12	1
gate_fallback	project	260	0.005084	-0.4222

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

none	project	0	1.212e-12	-1

This result highlights a structural distinction between governance mechanisms: gating concentrates intervention into discrete events, whereas projection distributes intervention across time steps through continuous correction.

4.3. Safety–Intervention Trade-off

The joint analysis of compliance and drift reveals a fundamental trade-off between safety and intervention cost.

Approval-based gating achieves near-complete elimination of violations at the expense of large, discrete deviations from the policy output. Projection-based governance provides a smoother control mechanism, reducing violations relative to the baseline while preserving continuity of actions, but at the cost of incomplete constraint enforcement and higher cumulative drift.

These results define a Pareto frontier between safety and intervention, where different governance strategies occupy distinct operating points. The ungoverned baseline minimizes intervention but is entirely unsafe, while gating maximizes safety with aggressive intervention, and projection offers an intermediate regime balancing safety and control smoothness.

4.4. Implications for Governance Design

The results confirm that embedding a deterministic governance operator fundamentally reshapes the behavior of agentic systems.

First, governance significantly reduces the probability of inadmissible executed actions, validating the compliance component of H1. Second, governance introduces bounded intervention, as evidenced by stable drift distributions and the absence of divergence or oscillatory behavior in the evaluated trajectories.

Importantly, projection-based governance does not guarantee strict feasibility in all cases, highlighting the practical limitations of optimization-based repair under complex and coupled constraints. This suggests that, in safety-critical applications, hybrid strategies combining projection with fallback mechanisms may be required to achieve both smoothness and strict compliance.

4.5. Summary of H1

Taken together, the results provide strong empirical support for the governance compliance–stability hypothesis.

Governance operators reduce constraint violations by a statistically significant margin while introducing controlled and bounded intervention. No evidence of destabilizing dynamics or unbounded drift is observed. Instead, governance induces a structured trade-off between safety and intervention cost, consistent with the theoretical characterization of the operator G.

4.6. Federated Stability and Action Consistency

To evaluate the federated extension of the governance framework (H2), we analyze convergence dynamics, action variance across rounds, and safety properties under distributed training.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

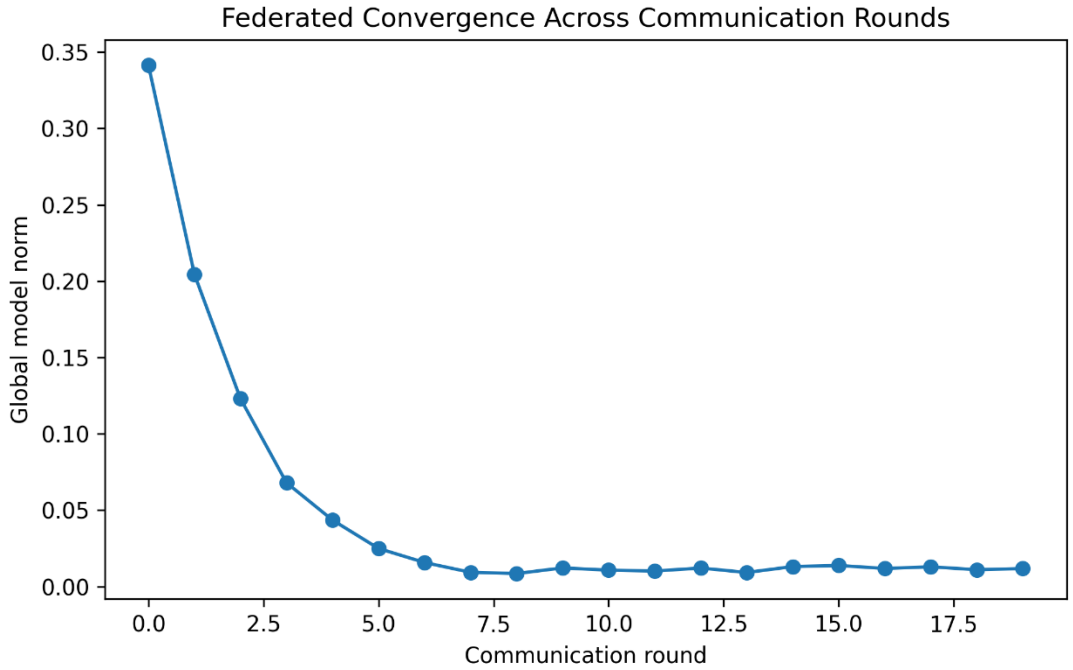

Figure 4. Federated convergence across communication rounds.

The evolution of the global model norm across communication rounds exhibits stable convergence. The norm decreases from 0.341 in the initial round to approximately 0.011 after 20 rounds, with no evidence of divergence or high-amplitude oscillations. This indicates that the inclusion of governance in the decision pipeline does not destabilize federated optimization.

At convergence, action-level stability is quantified using benchmark states. The mean action variance across rounds is 0.0123, indicating low variability in executed actions despite ongoing parameter updates. This confirms that governance projection stabilizes the action space even when the underlying model parameters evolve during training (Table 5).

Table 5. Federated benchmark stability metrics at convergence.

metric	value
violations_on_bench	0.0
action_var_mean	0.012298361767100896
mean_drift	0.03323036206708486

Importantly, no constraint violations are observed on the benchmark set, demonstrating that governance preserves safety under federated learning conditions. The mean decision drift at convergence is 0.033, indicating that only minimal intervention is required once the system stabilizes.

These results support the federated stability hypothesis (H2). Governance projection does not amplify action variance across rounds and does not degrade convergence. Instead, it induces a stabilizing effect on executed actions, effectively decoupling parameter-space variability from behavior-space stability.

5. Discussion

The results demonstrate that embedding a deterministic governance operator into the decision pipeline fundamentally alters the behavior of agentic systems. The observed effects extend beyond simple constraint enforcement and reveal a deeper structural role of governance as an architectural layer mediating between learning dynamics and execution safety.

5.1. Governance as a Structural Control Layer

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

A central finding of this work is that governance acts as a control layer that reshapes the mapping from policy outputs to executed actions. Rather than treating constraint enforcement as an external correction mechanism, the CAIS architecture integrates governance directly into the decision pipeline, thereby redefining the effective behavior of the system.

This integration yields two key properties. First, governance enforces admissibility at the level of executed actions, ensuring that constraint violations are significantly reduced or eliminated. Second, governance introduces a bounded transformation of policy outputs, preserving the continuity and structure of decision-making while preventing unsafe deviations.

Importantly, this reframes the role of learned policies: instead of being required to satisfy all constraints intrinsically, policies can operate in an unconstrained proposal space, with governance providing a deterministic projection into the admissible domain.

5.2. Safety–Intervention Trade-off

The experimental results reveal a clear trade-off between safety and intervention cost. Approval-based gating achieves near-perfect compliance by replacing infeasible actions with conservative fallbacks, but at the cost of large and discrete deviations from the original policy output. In contrast, projection-based governance produces smoother behavior but does not guarantee strict feasibility in all cases.

This trade-off can be interpreted as a Pareto frontier between safety and intervention. Systems can be configured to prioritize strict compliance or minimal intervention, depending on application requirements. In safety-critical domains, gating may be preferable due to its strong guarantees, whereas projection-based approaches may be better suited for systems requiring smoother control and higher performance [35].

The existence of this trade-off suggests that governance should not be treated as a binary mechanism but rather as a tunable component of system design.

The observed trade-off between safety and intervention cost aligns with prior findings in safe reinforcement learning, where stricter constraint enforcement often leads to increased policy modification [36,37]. However, unlike training-based approaches, the governance operator introduced in this work operates at execution time, providing deterministic guarantees independent of the learning process.

Similarly, the stabilization effect observed in federated settings is consistent with prior work highlighting the challenges of parameter divergence and client heterogeneity [20,22]. The results suggest that governance acts as a behavioral regularizer, constraining the space of admissible actions even when underlying model parameters vary.

5.3. Bounded Intervention and Stability

A key concern when introducing corrective mechanisms into decision pipelines is the risk of destabilizing feedback loops [38]. The results show no evidence of such behavior. Decision drift remains bounded, and trajectory-level deviations do not exhibit divergence or oscillatory amplification.

This supports the interpretation of governance operators as non-expansive transformations in the action space, consistent with projection-based formulations [39]. In practice, this means that governance introduces controlled and predictable modifications to system behavior, rather than amplifying perturbations.

The absence of instability is particularly important in closed-loop settings, where repeated corrections could otherwise accumulate and degrade performance [40].

5.4. Federated Learning and Behavioral Stabilization

In federated settings, governance exhibits an additional and previously underexplored role: stabilization of executed actions under parameter variability [41]. While federated training inherently

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

introduces heterogeneity and potential instability in model updates, governance ensures that the resulting actions remain consistent and admissible [42].

The empirical results show low action variance across rounds and zero constraint violations on benchmark states, even as model parameters evolve. This indicates that governance effectively decouples parameter-space dynamics from behavior-space outcomes [42,43].

This decoupling is a significant property for distributed AI systems, where guarantees on model convergence do not necessarily translate into guarantees on executed behavior. Governance provides a mechanism for enforcing behavioral consistency independently of training dynamics [44].

5.5. Limitations of Projection-Based Governance

While projection-based governance improves safety relative to the baseline, it does not guarantee strict feasibility in all scenarios. This limitation arises from multiple factors, including solver approximations, the presence of coupled constraints, and the potential non-convexity of the admissible action space [45].

These findings highlight an important practical consideration: optimization-based repair mechanisms may be insufficient when strict safety guarantees are required. In such cases, hybrid approaches combining projection with fallback strategies may be necessary.

Additionally, the observed higher drift under projection suggests that minimal intervention in a local sense does not necessarily translate into minimal cumulative deviation over time.

5.6. Implications for Safe AI System Design

The results have broader implications for the design of safe and auditable AI systems. The CAIS architecture demonstrates that safety can be enforced at the execution level without requiring policies to internalize all constraints during training.

This separation of concerns enables more flexible and scalable system design. Policies can be optimized for performance, while governance ensures compliance and safety. Furthermore, the integration of audit trace semantics provides a foundation for reproducibility and accountability, which are essential in regulated domains.

Overall, governance emerges as a principled mechanism for achieving safe deployment of agentic AI systems, particularly in environments characterized by uncertainty, distribution shift, and decentralized learning.

6. Conclusion

This work introduced a governance-driven architecture for Controlled Agentic AI Systems (CAIS), in which a deterministic operator is embedded directly into the decision pipeline to enforce constraint compliance and ensure reproducible system behavior. The proposed framework integrates governance, audit trace semantics, and reproducibility mechanisms into a unified formulation, enabling both theoretical analysis and empirical validation.

The experimental results demonstrate that governance significantly reduces constraint violations while maintaining bounded decision drift and stable closed-loop dynamics. Approval-based gating achieves near-complete compliance, whereas projection-based repair provides a smoother but imperfect enforcement mechanism, revealing a fundamental trade-off between safety and intervention cost. Importantly, no evidence of destabilizing behavior is observed, confirming that governance can be introduced without compromising system stability.

In federated settings, governance does not degrade convergence and instead stabilizes executed actions across training rounds. The results show that governance effectively decouples parameter-space variability from behavior-space outcomes, ensuring consistent and admissible actions even under distributed and heterogeneous training conditions.

These findings position governance as a fundamental architectural component for safe agentic AI systems, rather than an auxiliary post-processing mechanism. By separating decision generation

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

from constraint enforcement, the CAIS framework enables flexible policy design while preserving safety, auditability, and reproducibility.

Future work will focus on extending governance operators to more complex and non-convex constraint regimes, integrating hybrid control strategies combining projection and fallback mechanisms, and evaluating the framework in real-world maritime and safety-critical environments.

Author Contributions: Conceptualization, T.M. methodology, T.M.; software, T.M.; validation, T.M.; formal analysis, T.M.; investigation, T.M.; resources, T.M..; data curation T.M.; writing—original draft preparation, T.M.; writing—review and editing, T.M.; visualization, T.M.; supervision, T.M.; project administration, T.M.; funding acquisition, T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data, figures, and code used in this study are openly available and have been deposited in the public GitHub repository: https://github.com/TyMill/CAIS-pub. In addition, a permanent, citable version of the repository, including experimental results and materials, is archived on Zenodo under the following DOI: https://doi.org/10.5281/zenodo.19110441.

Conflicts of Interest: The author declare no conflicts of interest.

References

1. Pillay, N.; Nyathi, T.; Venayagamoorthy, G.K. Artificial Intelligence for Critical Infrastructure Systems: Past, Present and Future. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Rome, Italy, 2025; pp. 1–9. https://doi.org/10.1109/IJCNN64981.2025.11229347.
2. Agarwal, A.; Nene, M.J. Addressing AI Risks in Critical Infrastructure: Formalising the AI Incident Reporting Process. In *Proceedings of the IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, 2024; pp. 1–6. https://doi.org/10.1109/CONECCT62155.2024.10677312.
3. Chen, S.Y.-C.; Chen, K.-C. Quantum Artificial Intelligence for Critical Infrastructure: A Survey and Vision. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Rome, Italy, 2025; pp. 1–8. https://doi.org/10.1109/IJCNN64981.2025.11227684.
4. Ali, S.M.; Razzaque, A.; Yousaf, M.; Shan, R.U. An Automated Compliance Framework for Critical Infrastructure Security through Artificial Intelligence. *IEEE Access* 2025, *13*, 4436–4459. https://doi.org/10.1109/ACCESS.2024.3524496.
5. Kollipara, Y.V.P. Assured, Explainable, and Auditable AI for High-Stakes Decisions: A Survey of Trustworthy Machine Learning in Mission-Critical Systems. *J. Int. Crisis Risk Commun. Res.* 2025, *8*. https://doi.org/10.63278/jicrcr.vi.3392.
6. Jaziri, W.; Sassi, N. Explainable by Design: Enhancing Trustworthiness in AI-Driven Control Systems. *Mathematics* 2025, *13*, 3805. https://doi.org/10.3390/math13233805.
7. Singh, Y.; Hathaway, Q.A.; Keishing, V.; Salehi, S.; Wei, Y.; Horvat, N.; Vera-Garcia, D.V.; Choudhary, A.; Mula Kh, A.; Quaia, E.; et al. Beyond Post hoc Explanations: A Comprehensive Framework for Accountable AI in Medical Imaging. *Bioengineering* 2025, *12*, 879. https://doi.org/10.3390/bioengineering12080879.
8. Adabara, I.; Olaniyi, S.B.; Nuhu, S.A.; et al. Trustworthy Agentic AI Systems: A Cross-Layer Review of Architectures, Threat Models, and Governance Strategies. *F1000Research* 2025, *14*, 905. https://doi.org/10.12688/f1000research.169927.1.
9. Biswas, B.; Sarkar, S. Responsible Agentic Artificial Intelligence Governance: Risk, Safety, and Ethical Challenges. *Int. J. Appl. Resil. Sustain.* 2026, *2*, 142–167. https://doi.org/10.70593/deepsci.0202005.
10. de Witt, C.S. Open Challenges in Multi-Agent Security: Towards Secure Systems of Interacting AI Agents. *arXiv*2025, arXiv:2505.02077.
11. Hashi, A.I.; Hashi, A.M.; Jama, O.A. Trustworthy AI Governance Framework for Autonomous Systems. *Int. J. Comput. Trends Technol.* 2026, *74*, 12–27. https://doi.org/10.14445/22312803/IJCTT-V74I1P103.
12. Eden, R.; Chukwudi, I.; Bain, C.; et al. Governance of Federated Learning in Healthcare: A Scoping Review. *npj Digit. Med.* 2025, *8*, 427. https://doi.org/10.1038/s41746-025-01836-3.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

13. Matta, S.S.; Bolli, M. Trustworthy AI: Explainability and Fairness in Large-Scale Decision Systems. *Rev. Appl. Sci. Technol.* 2023, *2*, 54–93. https://doi.org/10.63125/3w9v5e52.
14. Basir, O.A. The Social Responsibility Stack: A Control-Theoretic Architecture for Governing Socio-Technical AI. *arXiv* 2025, arXiv:2512.16873.
15. Butt, T.A.; Iqbal, M.; Arshad, N. From Policy to Pipeline: A Governance Framework for AI Development and Operations Pipelines. *IEEE Access* 2026, *14*, 1373–1397. https://doi.org/10.1109/ACCESS.2025.3647479.
16. Muhammad, A.E.; Yow, K.-C. Risk-Based AI Assurance Framework. *Information* 2026, *17*, 263. https://doi.org/10.3390/info17030263.
17. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained Policy Optimization. In *Proceedings of the International Conference on Machine Learning*, 2017; pp. 22–31.
18. Chow, Y.; Nachum, O.; Duenez-Guzman, E.; Ghavamzadeh, M. A Lyapunov-Based Approach to Safe Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2018; *31*.
19. Tearle, B.; Wabersich, K.P.; Carron, A.; Zeilinger, M.N. A Predictive Safety Filter for Learning-Based Racing Control. *IEEE Robot. Autom. Lett.* 2021, *6*, 7635–7642. https://doi.org/10.1109/LRA.2021.3097073.
20. Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; Topcu, U. Safe Reinforcement Learning via Shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018; *32*. https://doi.org/10.1609/aaai.v32i1.11797.
21. Ames, A.D.; Xu, X.; Grizzle, J.W.; Tabuada, P. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Trans. Autom. Control* 2017, *62*, 3861–3876. https://doi.org/10.1109/TAC.2016.2638961.
22. Xiao, W.; Belta, C. Control Barrier Functions for Systems with High Relative Degree. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Nice, France, 2019; pp. 474–479. https://doi.org/10.1109/CDC40024.2019.9029455.
23. Cheng, R.; Orosz, G.; Murray, R.M.; Burdick, J.W. End-to-End Safe Reinforcement Learning through Barrier Functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019; pp. 3387–3395.
24. Doshi-Velez, F.; Kim, B. Towards a Rigorous Science of Interpretable Machine Learning. *arXiv* 2017, arXiv:1702.08608.
25. Cannarsa, M. Ethics Guidelines for Trustworthy AI. In *The Cambridge Handbook of Lawyering in the Digital Age*; 2021; pp. 97–283.
26. Floridi, L.; Cowls, J.; Beltrametti, M.; Chatila, R.; Chazerand, P.; Dignum, V.; et al. AI4People—An Ethical Framework for a Good AI Society. *Minds Mach.* 2018, *28*, 689–707. https://doi.org/10.1007/s11023-018-9482-5.
27. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; et al. Explainable Artificial Intelligence (XAI): Concepts and Challenges. *Inf. Fusion* 2020, *58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012.
28. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, 2017; pp. 1273–1282.
29. Kairouz, P.; McMahan, H.B. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.*2021, *14*, 1–210.
30. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. *Proc. Mach. Learn. Syst.* 2020, *2*, 429–450.
31. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of ICML*, 2020. https://doi.org/10.48550/arXiv.1910.06378.
32. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Environments. In *Advances in Neural Information Processing Systems*, 2017; *30*.
33. Zhang, K.; Yang, Z.; Başar, T. Multi-Agent Reinforcement Learning: A Selective Overview. In *Handbook of Reinforcement Learning and Control*; 2021; pp. 321–384. https://doi.org/10.1007/978-3-030-60990-0_12.
34. Busoniu, L.; Babuska, R.; De Schutter, B. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. C* 2008, *38*, 156–172.
35. Hung, W.; Sun, S.H.; Hsieh, P.C. Efficient Action-Constrained Reinforcement Learning. *arXiv* 2025, arXiv:2503.12932.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2026 doi:10.20944/preprints202603.1904.v1

36. Tessler, C.; Mankowitz, D.J.; Mannor, S. Reward Constrained Policy Optimization. *arXiv* 2018, arXiv:1805.11074.
37. Dalal, G.; Dvijotham, K.; Vecerik, M.; Hester, T.; Paduraru, C.; Tassa, Y. Safe Exploration in Continuous Action Spaces. *arXiv* 2018, arXiv:1801.08757.
38. Kilian, K.A. Structural Risk Dynamics of Artificial Intelligence. *AI Soc.* 2026, *41*, 23–42. https://doi.org/10.1007/s00146-025-02419-2.
39. Ferrari, L.; Frosini, P.; Quercioli, N.; Tombari, F. A Topological Model for Partial Equivariance. *Front. Artif. Intell.*2023, *6*, 1272619. https://doi.org/10.3389/frai.2023.1272619.
40. Tao, Y. The Decision Path to Control AI Risks Completely. *arXiv* 2025, arXiv:2512.04489.
41. Konakanchi, M.S.K. Aegis: AI-Driven Governance Framework for Micro-Frontend Architectures. *IJAIDR* 2025, *17*.
42. Garro, R.J.; Wibowo, S.; Wilson, C.S.; Pordomingo, A.J. Federated Learning Architecture for Precision Livestock Systems. In *Proceedings of ICICoS*, 2025; pp. 340–345. https://doi.org/10.1109/ICICoS68590.2025.11329806.
43. Jiang, Y.; Wang, F.; Pang, X. Personalized Federated Learning for Gastric Cancer Classification. In *Proceedings of ICCC*, 2024; pp. 771–775. https://doi.org/10.1109/ICCC62609.2024.10942100.
44. Wei, W.; Liu, L. Trustworthy Distributed AI Systems: Robustness, Privacy, and Governance. *ACM Comput. Surv.*2025, *57*, 1–42.
45. Li, F.; Li, M.; Li, S.; Wu, Y.; Song, Y.; Li, H. Rational-Safe Reinforcement Learning for Energy Management. In *Proceedings of IECON*, Madrid, Spain, 2025; pp. 1–6. https://doi.org/10.1109/IECON58223.2025.11221059.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.