Executable Governance at the Point of Irreversibility

A Formal Architectural Framework for Authority Enforcement in Composed AI Systems

Masayuki Otani
AI Consultant Insights (AICI)
United Kingdom
info@aiconsultantinsights.com

***Abstract*—As AI systems evolve from bounded models into agentic, multi-system architectures, governance failures increasingly occur despite local compliance of individual components. These failures are structural rather than behavioral, arising when legitimate authority cannot be demonstrated or enforced at the moment composite actions become irreversible. This paper introduces a formally specified architectural governance framework comprising four integrated mechanisms, ISDAIRE (ex-ante admissibility), ARETABA (runtime enforcement), GAG (compositional authority preservation), and MGAG (multi-layer governance). Authority is treated as a first-class executable object, defined ex-ante, cryptographically bound at runtime, compositionally preserved across system boundaries, and deterministically enforced at irreversibility boundaries. The framework is mathematically rigorous, falsifiable, implementation-agnostic, and applicable to any distributed architecture producing irreversible outcomes. Formal definitions, atomicity semantics, provenance requirements, integrity-based audit criteria, and bounded latency analysis are provided. This work is proposed as a reference architecture for executable governance in composed AI systems.**

***Index Terms*—AI governance, architectural governance, authority, irreversibility, execution-time control, compositional systems, auditability, agentic systems**

I. Introduction

Contemporary AI governance frameworks primarily emphasise principles, risk management, documentation, and post-deployment accountability. While necessary for transparency and regulatory compliance, these mechanisms often assume bounded systems with stable, identifiable loci of authority.

This assumption no longer holds in modern deployments. Production systems increasingly consist of agentic workflows that compose models, tools, APIs, data services, and actuators across vendors, organisations, and regulatory domains. In such environments, authority is distributed, delegated, transformed, and sometimes lost as actions traverse system boundaries.

A recurring failure pattern is observed in practice. Each component behaves correctly, complies with local rules, produces audit logs, and passes formal checks. Nevertheless, the overall outcome is illegitimate, unsafe, or harmful. Responsibility becomes diffuse, and no single authority can be shown to have legitimately authorised the final action.

This failure mode is structural rather than behavioural. Authority fragments at system boundaries and is absent or unverifiable at the moment when outcomes become irreversible.

Security engineering establishes that protection must be architectural and enforced at the point of use, not merely documented or reconstructed post-hoc. The Principle of Complete Mediation requires that every protected operation be checked at the moment it occurs [1]. This paper extends that principle to composed AI systems by formalising governance as an executable control structure that binds authority at execution time.

II. Scope and Non-Claims

A. Scope

This framework applies to systems that autonomously or semi-autonomously initiate actions producing irreversible external effects, including financial settlement, legal commitment, physical actuation, access-control modification, or irreversible data disclosure. It is designed for composed, multi-system architectures where authority must be preserved across organisational, technical, and regulatory boundaries.

B. Out of Scope

Systems without meaningful irreversibility boundaries, such as offline analytics, batch inference, simulation, or purely advisory systems, are out of scope.

C. Non-Claims

This framework does not guarantee safety, correctness, alignment, or regulatory compliance. It does not certify systems or vendors. It provides an architectural control structure that makes authority enforceable, refusals deterministic, and outcomes auditable at execution time. As with classical security architectures, correctness depends on proper instantiation. Architecture defines what is enforceable, not what is guaranteed.

III. Architectural Premise and Problem Formalisation

A. Foundational Premise

P1. If legitimate authority cannot be shown and enforced at the moment an outcome becomes irreversible, governance did not exist for that action path.

This premise applies complete mediation to composed AI systems. The protected operation is the irreversible action itself.

B. Formal Model

Let a composite system be represented as a directed graph

$$\mathcal{S} = (V, E) \tag{1}$$

where V is the set of components and E represents control flow, data flow, or authority delegation edges.

Let $\mathcal{A}$ be the set of actions the system may attempt.

Let $\mathbb{T}$ be a totally ordered execution time domain.

C. Execution Boundary

Define an execution boundary function

$$T_e : \mathcal{A} \to \mathbb{T} \tag{2}$$

where $T_e(a)$ denotes the earliest time at which action a becomes irreversible.

D. Central Governance Question

Was legitimate authority present and enforceable at $T_e(a)$ for every executed action a.

IV. Limitations of Existing Governance Approaches

Existing governance approaches exhibit structural limitations when applied to composed, agentic systems.

First, many are risk-centric rather than authority-centric, emphasising mitigation over execution-time enforcement [2]. Second, they commonly assume local compliance implies global legitimacy, which is fragile in multi-system supply chains. Third, post-hoc accountability emphasises documentation and oversight, which does not define how authority is preserved and enforced across boundaries at runtime [5]. Fourth, normative frameworks provide useful principles but do not define execution semantics [3], [4].

These approaches are necessary but insufficient where outcomes arise from composition rather than a single bounded decision locus.

V. ISDAIRE Ex-Ante Admissibility

ISDAIRE answers whether authority may ever exist.

A. Definition and Purpose

For an action class α, define an ISDAIRE specification

$$\begin{aligned}\text{ISDAIRE}(\alpha) = (&I(\alpha), S(\alpha), D(\alpha),\\ &A(\alpha), R(\alpha), E(\alpha))\end{aligned} \tag{3}$$

where I is Intent, S is Scope, D is Domain, A is Authority Source, R is Risk Framing, and E is Execution Boundary specification.

B. Admissibility Predicate

Define

$$\begin{aligned}\text{Admissible}_{\text{ISDAIRE}}(\alpha) = 1 \iff\ & I(\alpha) \in \text{ValidIntents}\\ \wedge\ & S(\alpha) \subseteq \text{PermittedEffects}\\ \wedge\ & D(\alpha) \in \text{KnownDomains}\\ \wedge\ & A(\alpha) \neq \emptyset\\ \wedge\ & R(\alpha) \leq \text{RiskThreshold}\\ \wedge\ & E(\alpha) \text{ is deterministic}\end{aligned} \tag{4}$$

C. Governance Oracle Constraint

Rule IS1. Evaluation of $\text{Admissible}_{\text{ISDAIRE}}(\alpha)$ must be performed by a mechanism that is deterministic and non-agentic, or governed by a higher-order instance of this framework, or explicitly authorised and frozen by human authority ex-ante.

Rule IS1a. Any chain of governance evaluation instances must terminate at a non-agentic deterministic evaluator or explicit human authority within a declared bounded depth $D_{\max}$. Unbounded or cyclic governance recursion is non-compliant.

A frozen evaluator is one whose logic and parameters cannot self-modify, retrain, or adapt without invalidating associated authority and forcing re-authorisation.

D. Failure Condition

ISDAIRE failure occurs when there exists an α such that execution is attempted while $\text{Admissible}_{\text{ISDAIRE}}(\alpha) = 0$. This is a design-time failure that runtime governance cannot compensate.

VI. ARETABA Runtime Authority Enforcement

ARETABA answers whether authority is valid now.

A. Authority Object

For an action instance a at execution boundary $T_e(a)$, define a runtime authority object

$$\begin{aligned}\beta = (&p,\ \sigma,\ \lambda,\ \rho,\\ &\kappa,\ \pi)\end{aligned} \tag{5}$$

where p is Principal, σ is Scope, λ is Lifecycle validity, ρ is Revocation, κ is Collapse behaviour, and π is Provenance binding.

B. State Responsibility

Let x denote full system state at time t. Define $x_\lambda \subseteq x$.

Axiom AR1. x_λ must contain all state elements upon which $\lambda(\beta, x, t)$ or $\rho(\beta, x, t)$ functionally depend. Omission of any such dependency constitutes non-compliance.

C. Execution Permission

Define

$$\text{Valid}_\lambda(\beta, x, t) = \lambda(\beta, x, t) \tag{6}$$

$$\text{Revoked}_\rho(\beta, x, t) = \rho(\beta, x, t) \tag{7}$$

Execution permission is granted if and only if

$$\begin{aligned} \text{Permit}(a, \beta, x) \iff & \text{Admissible}_{\text{ISDAIRE}}(\alpha) \\ & \wedge\ (a \in \sigma) \\ & \wedge\ \text{Valid}_\lambda(\beta, x, T_e(a)) \\ & \wedge\ \neg\text{Revoked}_\rho(\beta, x, T_e(a)) \end{aligned} \tag{8}$$

D. Logical Atomicity

Rule AR2. Authority evaluation and action commitment at $T_e(a)$ must be logically atomic, meaning executed as a single local transaction, or a bounded distributed transaction protocol with explicit rollback or compensating authority.

A compensating authority is a pre-authorised mechanism that either reverts committed state or records rollback with provenance guarantees equivalent to the original action.

If neither logical atomicity nor compensating authority can be guaranteed, execution must be refused.

E. Default Deny and Refusal Semantics

Axiom AR3. If no valid authority object β can be resolved at $T_e(a)$, or if any required authority attribute is missing, malformed, or unverifiable, the system must deterministically refuse execution and transition to a declared safe halt or fallback state. Absence of authority is treated as explicit denial, not an exceptional condition.

F. Fallback Semantics

Define a fallback action set $\text{Fallback}(a, \beta) \subseteq \mathcal{A}$.

Rule AR4. For every refusal scenario, a fallback action $f \in \text{Fallback}(a, \beta)$ must satisfy

$$\text{Risk}(f) \leq \text{Risk}(a) \tag{9}$$

and there exists β' such that $\text{Permit}(f, \beta', x)$ holds, with β' defined ex-ante.

Fallback actions must be safer than refused actions, must have pre-authorised authority, and must be auditable.

VII. GAG Global Architectural Governance

GAG answers whether authority survived composition.

A. Signal Model and Provenance

Let systems emit structured signals

$$\Sigma = \Sigma_{\text{num}} \cup \Sigma_{\text{bool}} \cup \Sigma_{\text{str}} \cup \Sigma_{\text{json}} \tag{10}$$

Signals are insufficient without provenance. Define provenance evidence

$$\pi = (\text{source\_id}, \text{signature}, \text{timestamp}, \text{nonce}, \text{attestation}) \tag{11}$$

Requirement G1. Provenance evidence must be non-repudiable, tamper-evident, time-ordered, chain-verifiable, and bound to authority object β.

B. Provenance Validity

Define

$$\begin{aligned} \text{Valid}_\pi(\pi, t) \iff & \text{Verify}(\text{signature}, \text{source\_id}) \\ & \wedge\ \text{Fresh}(t, \Delta t_{\max}) \\ & \wedge\ \text{Unique}(\text{nonce}) \\ & \wedge\ \text{Attest}(\text{env}) \end{aligned} \tag{12}$$

C. Composite Authority Conditions

Define typed deterministic rules $r_k : (\sigma, \pi) \to \{0, 1\}$. Composite authority conditions are defined ex-ante as

$$C_i = r_{i1} \wedge r_{i2} \wedge \cdots \wedge r_{ik} \tag{13}$$

D. Authority Validity Function

Let p be the authorising principal. Define

$$A_p(t) = \bigwedge_{i=1}^{n} C_i(\sigma(t), \pi(t)) \tag{14}$$

Evaluation occurs at T_e. If required signals arrive asynchronously, evaluation must wait for all required signals within a bounded timeout $\Delta t_{\max}$, or refuse execution if the timeout is exceeded. No inference or reconstruction of missing signals is permitted.

E. GAG Verification Function

Define $\text{GAG}(a, \beta, \pi) \in \{0, 1\}$ verifying that provenance π forms a valid, acyclic, and cryptographically bound chain from originating principal p to execution boundary $T_e(a)$.

Rule G2. Authority delegation graphs must be acyclic or explicitly bounded. Cyclic or self-referential authority paths are non-compliant.

F. GAG Invariant

$$\begin{aligned} \text{Execute}(a) \iff & A_p(T_e(a)) = 1 \\ & \wedge\ \text{GAG}(a, \beta, \pi) = 1 \end{aligned} \tag{15}$$

VIII. MGAG Multi-Layer Global Architectural Governance

Let governance layers be indexed $k \in \{1, 2, \ldots, L\}$. For each layer k define execution boundary $T_e^{(k)}$, principal $p^{(k)}$, authority object $\beta^{(k)}$, and authority conditions $C_i^{(k)}$.

Define per-layer authority validity

$$A^{(k)}(t) = \bigwedge_{i=1}^{n_k} C_i^{(k)}(\sigma(t), \pi(t)) \tag{16}$$

Define multi-layer permission

$$\text{MGAG}(a) = \bigwedge_{k=1}^{L} \text{Permit}_k(a, \beta_k, x) \tag{17}$$

Execution is permitted if and only if for all k, $A^{(k)}(T_e^{(k)}) = 1$ and $\text{GAG}(a, \beta^{(k)}, \pi^{(k)}) = 1$ hold. Failure at any layer forbids global execution or triggers the layer-appropriate fallback.

Rule M1. If layers impose conflicting requirements, the system must refuse execution or escalate to a higher-priority authority explicitly defined ex-ante. Priority orderings must be declared in the ISDAIRE specification and auditable.

IX. Integrated Control Structure

At execution boundary T_e, the unified control structure is

$$\begin{aligned} \text{Execute}(a) \iff & \text{Admissible}_{\text{ISDAIRE}}(\alpha) \\ & \wedge \ \text{GAG/MGAG}(a, \beta, \pi) \\ & \wedge \ \text{Permit}_{\text{ARETABA}}(a, \beta, x) \end{aligned} \tag{18}$$

This structure is deterministic, auditable, and enforceable.

X. Audit Artefacts and Requirements

For every executed action a, and for every refused action where refusal occurs at or before $T_e(a)$, the system must record

1) Authority object identifier β
2) Execution boundary timestamp $T_e(a)$
3) Bound state subset x_λ
4) Provenance evidence π
5) Enforcement outcome, permit, refuse, fallback, or escalate

Audit artefacts must be tamper-evident, cryptographically chained, and time-ordered [6].

Audit closure statement. Absence of recorded refusal, fallback, or escalation artefacts for an attempted irreversible action path is itself evidence of governance failure for that path.

XI. Falsifiability

Criterion F1. A system is non-compliant if there exists T_e such that authority at T_e is unknown, inferred, reconstructed, or unverifiable from audit artefacts.

Criterion F2. A system is non-compliant if authority at T_e is stale, forged, or derived from corrupted x_λ, even if present.

These criteria are empirically testable through inspection of the audit artefacts defined above.

XII. Latency Considerations

This framework introduces latency only at execution boundaries T_e.

Let

$$\Delta t = t_{\text{ISDAIRE}} + t_{\text{GAG}} + t_{\text{prov}} + t_{\text{log}} \tag{19}$$

For multi-layer governance with L layers

$$\Delta t_{\text{MGAG}} \leq \sum_{k=1}^{L} \Delta t_k \tag{20}$$

Parallel evaluation and service-level bounds may be applied, provided refusal semantics are preserved. Refusal on timeout is mandatory and must be logged as an enforcement outcome.

XIII. Applied Scenario

An autonomous payment agent attempts a £50,000 transfer.

A. ISDAIRE Evaluation

Let the action class be $\alpha = \text{financial\_transfer}$, with scope

$$S(\alpha) = \{\text{transfer} \mid \text{amount} \leq 10{,}000\} \tag{21}$$

The class is admissible, so $\text{Admissible}_{\text{ISDAIRE}}(\alpha) = 1$.

B. GAG Evaluation at the Execution Boundary

Let $T_e(a)$ be settlement initiation. The system collects required signals and provenance and evaluates $A_p(T_e(a))$ using deterministic predicates. Scope validation fails because the requested amount is outside $S(\alpha)$, therefore $A_p(T_e(a)) = 0$. No inference or reconstruction is permitted for missing or late signals. Authority does not hold at $T_e(a)$.

C. ARETABA Enforcement

At $T_e(a)$, ARETABA resolves the relevant authority object β and enforces default deny. The authority evaluation and commitment step is logically atomic, so no partial settlement occurs. Execution is refused deterministically.

D. Fallback and Audit Artefacts

A pre-authorised fallback routes the request to a human treasury queue under a separate authority object β'. Audit artefacts record β, $T_e(a)$, x_λ, provenance π, and the enforcement outcome, refusal followed by fallback routing. No component failed. Governance executed.

XIV. Conclusion

Governance that cannot execute at the moment of irreversibility is indistinguishable from absence. Authority that cannot be refused, revoked, or audited at execution time is not authority. This paper defines the minimal architectural control structure required for governance to exist in composed AI systems. ISDAIRE determines whether authority may exist. GAG preserves authority across composition. ARETABA enforces authority, refusal, and fallback at execution. MGAG ensures these properties hold across layers. Anything less is policy, not governance.

References

[1] J. H. Saltzer and M. D. Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975.

[2] National Institute of Standards and Technology, *AI Risk Management Framework (AI RMF 1.0)*, NIST, 2023.

[3] Organisation for Economic Co-operation and Development, "OECD Principles on Artificial Intelligence," OECD Council Recommendation, 2019.

[4] IEEE, *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems*, IEEE, 2019.

[5] European Union, "Regulation laying down harmonised rules on artificial intelligence," commonly referred to as the EU AI Act, adopted 2024.

[6] R. C. Merkle, "A Certified Digital Signature," in *Advances in Cryptology, CRYPTO '89 Proceedings*, Springer, 1989.