AEA by Ivory & Co.

Adaptive Enterprise Architecture --- Governance System Design

v1.0.1 --- Published Baseline — Editorial Correction

Status: Published — v1.0.1. This editorial correction clarifies present-state design language without changing the approved Governance System design.

1. Purpose

This document defines the AEA Governance System as it is designed to operate.

The Governance System is a governed system that can be applied wherever governed consequence exists. It is not an AEA Architecture and is not confined to a single layer, artefact type, product, implementation or organisational context.

The Governance System uses the eleven AEA Architecture Standards to make Governance itself architecturally legible. The resulting design is expressed here as an integrated operating design. The architectural lenses explain the dimensions of that design; they do not describe a build sequence.

The Governance Standards define the normative protections through which this Design is governed and applied. The Governance System Design and Governance Standards therefore have different functions:

· Governance System Design defines the structure, operating logic, relationships and design properties of the Governance System.
· Governance Standards define the normative protection requirements that apply wherever Governance is instantiated.

2. Design boundary

Governance is not a twelfth AEA Architecture and is not exclusive to the canonical AEA layer. Governance is a subject that can itself be architected using AEA and then applied wherever governed consequence exists.

The design objective is therefore not identical governance machinery everywhere. It is a coherent governing system capable of preserving common protections across very different governed contexts.

> Governance standardises protection, not implementation.

3. Governance System design through the eleven AEA architectural dimensions

Directional Architecture — Intent

The Directional dimension establishes the governing intent: protect the integrity, legitimacy, traceability and safe evolution of governed subjects without suppressing legitimate innovation, contextual specialisation or downstream divergence. Governance must preserve consequence-

appropriate protection while keeping execution as simple as possible.

Design implications include: govern consequence rather than creativity; preserve canonical authority without assuming downstream authority; permit legitimate retained baselines and variants; and make protection durable across change.

Domain Architecture — Scope

The Domain dimension establishes the capability scope the Governance System must possess. The current thirteen Governance concerns collectively cover system governance; governed objects/records; identity/addressing; authority/decision governance; source/provenance; IP; lifecycle/status; change; dependency/blast radius/integrity; version/release/publication; information/schema/machine readability; assurance/conformance; and repository/access/retention.

The thirteen-parent set remains flat. Additional complexity or implementation detail does not justify a new parent concern unless a necessary governance phenomenon has no coherent owner.

Ecosystem Architecture — Interconnectedness

The Ecosystem dimension establishes that Governance operates across relationships among governed subjects, authorities, sources, owners, implementers, commercial interpreters, contractual parties, machines/agents and downstream derivatives. Rights and obligations may originate outside Governance; Governance consumes them as conditions.

Governance therefore needs explicit relationship, exchange, dependency and lineage visibility without assuming ownership of the actors or subjects involved.

Execution Architecture — Instructions

The Execution dimension establishes that Governance requires bounded executable mechanisms for handoff, alignment, determination, change, release, assurance, exception handling, downstream consequence assessment and re-entry. These mechanisms can be recursive, composable and specialised.

Activity is not progression. Evidence is not determination. Approval is not embedment. Assessment is not mutation. Operational stages or funnel visualisations are mechanisms/renders, not the ontology of Governance itself.

Expression Architecture — Individuality

The Expression dimension establishes that the same governing truth may require different consumable forms for Standards, products, software, SKUs, customer implementations, humans, machines, boards, certification and public material.

Renders and publications must not manufacture governing truth. Expression may vary while the governing protection and proposition lineage remain coherent.

Possibility Architecture — Optionality

The Possibility dimension establishes the deliberate boundary between exploratory possibility and governed embedment. Innovation may remain messy, contradictory and unconverged. The originating innovation context determines when thinking is sufficiently formed to be intentionally handed forward. Governance does not pull from or prescribe the innovation environment.

Once handed forward, possibilities may progress, hold, return, split, merge, redirect, converge,

terminate or re-enter. Most possibilities need not become governed change.

Reality Architecture — Actuality

The Reality dimension establishes that Governance cannot determine consequence solely from intended design or current canon. It must be able to establish the actual position of governed subjects: what exists, what baseline or variant is operating, what has actually changed, what is unknown and where evidence is incomplete.

This becomes particularly important downstream. Blast radius identifies potential consequence; Reality supplies the actual state against which consequence can be established. Unknown actuality must remain explicit rather than being invented.

Intelligence Architecture — Understanding

The Intelligence dimension establishes a Governance-specific movement pattern:

> **Innovation --- divergent understanding !’ Enablement --- convergent understanding !’ Absorption --- divergent downstream understanding**

In Innovation, possibility expands. In Enablement, submitted thinking is aligned, interpreted, challenged and progressively reduced toward consequential determination. In Absorption, an embedment or source evolution is interpreted against actual existing conditions to understand different downstream consequences.

This is a Governance System Design pattern expressed through Intelligence Architecture. It is specific to the Governance subject unless separately established as a universal Intelligence Architecture proposition. It is not a universal linear lifecycle: nested, recursive and interacting loops may operate throughout.

Intelligence owns the integration/understanding, not the subjects being understood. Governance consumes that understanding to protect consequential transitions and determinations.

Workforce Architecture — Participation

The Workforce dimension establishes that Governance must distinguish roles, participants and authority. Humans, agents and organisations may participate in governance work, but technical capability or task execution does not itself create authority.

Governance therefore needs explicit role requirements, participant assignment, delegated bounds and escalation conditions while preserving the established AEA distinction between Role, Participant, Actor and Authority.

Environment Architecture — Enablement

The Environment dimension establishes the physical and technological enablement required for Governance to operate: authoritative structured knowledge, stable identity/addressing, history/version preservation, controlled change, machine-readable relationships, monitoring/testing and reliable human/machine publication.

This use of Environment enablement is distinct from the Governance-specific Enablement movement label expressed through the Intelligence dimension. Machine semantics and renders must disambiguate them by context.

Trust Architecture — Reliance

The Trust dimension establishes that Governance exists partly to make warranted reliance

possible. Reliance cannot be created by assertion, completed forms or protocol presence alone. The protections relevant to the consequence must actually be effective, their authority and currency knowable, and uncertainty visible.

Assurance therefore tests protection effectiveness, not governance theatre. Externality alone does not prove independence, and a known defect or unknown condition may qualify reliance rather than forcing false certainty.

4. Integrated Governance System design

The Governance System has the following integrated design properties:

· Orthogonal applicability: Governance can apply to any governed subject or layer; it is not itself an AEA Architecture.

· Common protection, contextual implementation: protective intent is inherited; implementation may specialise.

· Consequence-based engagement: Governance protects consequential movement without colonising innovation.

· Applicability-based inheritance: rules propagate according to applicability and lineage, not fixed hierarchy.

· Destination-specific composition: each governed destination resolves its applicable protection set and authority.

· Traceable divergence: source evolution triggers consequence assessment, not forced synchronisation.

· Actuality before downstream determination: Reality informs what actually exists; uncertainty is preserved.

· Portfolio legibility without authority collapse: cumulative and relational consequence can be seen without centralising all decisions.

· Machine/human addressability: governing truth, rationale, lineage, status and applicability remain traversable.

· Effective assurance: conformance is protection effectiveness proportionate to consequence, not box completion.

5. Governance Standards structure

The Governance System is governed through thirteen parent Governance Standards:

1. GOV-01 AEA Governance System Standard 2. GOV-02 Governance Object & Record Standard 3. GOV-03 Identity & Addressing Standard 4. GOV-04 Authority & Decision Governance Standard 5. GOV-05 Source & Provenance Governance Standard 6. GOV-06 Intellectual Property Governance Standard 7. GOV-07 Lifecycle & Status Standard 8. GOV-08 Change Management Standard 9. GOV-09 Dependency, Blast Radius & Integrity Standard 10. GOV-10 Version, Release & Publication Standard 11. GOV-11 Information, Schema & Machine-Readability Standard 12. GOV-12 Assurance & Conformance Standard 13. GOV-13 Repository, Access & Retention Standard

These Standards are not the eleven architectural lenses repeated. They are the **normative governance concerns that operationalise those lenses to Governance as a subject**.

6. Standard !’ protection protocol !’ specialisation !’ implementation

The semantic relationship is:

> **Governance Standard !’ owns governing protection(s) !’ protection protocol expresses the applicable protection for a governed subject !’ profile/specialisation resolves contextual requirements !’ implementation mechanism realises the protection operationally**

A protection protocol is therefore **not a fourteenth Standard and not another architectural layer**. It is the applicable governing expression of protection owned by one or more Governance Standards. Cross-standard composition is permitted where a destination requires several protections; ownership remains with the relevant Standards.

Specialisation may alter implementation detail, thresholds, evidence, timing, participants or mechanisms where permitted, but may not silently defeat inherited protective intent.

7. Design-to-Standard traceability rule

Each material normative Governance proposition should be traceable to the Design requirement(s) that justify it. Traceability does not make a Design proposition normative; it makes the relationship between design intent and governing protection legible. Conversely, an architectural observation should not be promoted into a source AEA Standard merely because it was useful in designing Governance. Universal source-Standard change requires its own architectural case.

8. Open implementation decisions

The Design does not yet lock final canonical identifier grammar, universal lifecycle vocabulary, final disposition terms, certification model, technical schema, repository technology, assurance scoring, contractual terms or a universal protection checklist. Those remain implementation, profile or governed design decisions to be resolved when their applicable context requires them.