

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

ARCHITECTURE REFERENCE

ODT-EA



Enterprise Architecture — 04: Governance

Architecture Reference

HEADQUARTERS
UNITED STATES ARMY EUROPE AND AFRICA
(USAREUR-AF)
Wiesbaden, Germany

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

20 MARCH 2026

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

sidebar_position: 5 title: "Governance"

ENTERPRISE ARCHITECTURE — 04: GOVERNANCE

How EA actually influences decisions — or fails to

THE GOVERNANCE PROBLEM

EA without governance is a documentation exercise. The entire point of EA — coherence, reduced duplication, deliberate transformation — depends on EA being *authoritative*. That means it has to show up in the processes where real decisions get made.

EA governance = the mechanisms by which EA influences decisions before they're made, not after.

Post-hoc review ("we looked at the finished design and it doesn't match the architecture") is not governance. Governance is upstream: EA shapes requirements, intake criteria, funding decisions, and acquisition language *before* solutions are designed.

THE FOUR GOVERNANCE ENTRY POINTS

There are four places where EA governance can be real:

1. Investment / Portfolio Governance

Where: Portfolio management boards, budget submission review, program approval processes (e.g., PPBE in DoD)

EA's role: Ensure investment requests are aligned to capability gaps (from the heat map), don't duplicate existing capability (from the app inventory), follow platform and integration standards (from the tech reference architecture), and are sequenced correctly (from the roadmap).

Without this: Programs get funded that duplicate existing capabilities, build on non-standard platforms, or solve the wrong problem.

Key artifact: Capability heat map + app inventory + roadmap

2. Solution Intake / Architecture Review

Where: IT governance boards, architecture review boards (ARB), Engineering Review Boards (ERB), design authority functions

EA's role: New solutions must demonstrate alignment to capability map, data standards, integration patterns, and tech reference architecture *before* design is finalized. EA approves deviation requests.

Without this: Each program team makes its own architectural decisions. The result is a fragmented, incompatible, unmaintainable system estate.

Key artifact: Technology reference architecture + data/information map + integration standards

3. Acquisition and Requirements

Where: Requirements documents (ICD, CDD, CPD in DoD), contracting vehicles, SOW/PWS language

EA's role: EA standards are written into acquisition requirements. Vendors are required to comply with the enterprise architecture. "Must integrate with [platform]" and "must use [data standard]" language comes from EA.

Without this: Procurement creates lock-in, incompatible interfaces, and capability islands that EA cannot govern after the fact.

Key artifact: Technology reference architecture + data standards + interoperability requirements

4. Data and Metadata Governance

Where: Data governance councils, authoritative source designations, ontology review processes, metadata standards

EA's role: EA owns the semantic layer — what data entities exist, what they mean, who is authoritative for them, and what standards govern how they're described and exchanged.

Without this: Data proliferates without semantic consistency. You get multiple systems calling the same thing by different names, different systems using the same term for different things, and no way to federate data across the enterprise.

Key artifact: Data/information map + ontology/semantic model + authoritative source registry

THE GOVERNANCE OPERATING MODEL

A functional EA governance model has three tiers:

<p>TIER 1 – EXECUTIVE / STRATEGIC Architecture Board / CTO / Investment Council Approves: major deviations, roadmap priorities, capability investment decisions</p>
<p>TIER 2 – OPERATIONAL / REVIEW Architecture Review Board (ARB) Approves: solution designs, platform choices, integration approaches, data standards</p>
<p>TIER 3 – ENABLING / STANDARDS EA Team / Domain Architects Produces: reference architectures, standards, patterns, assessments, roadmap updates</p>

Tier 3 does the work. **Tier 2** reviews and approves. **Tier 1** makes strategic decisions and owns the mandate.

Without Tier 1, EA has no authority. Without Tier 2, EA has no teeth. Without Tier 3, EA has no substance.

THE ARB: HOW IT WORKS IN PRACTICE

The Architecture Review Board is the most common EA governance mechanism. A minimal ARB operates as follows:

Intake trigger: Any solution, program, or procurement above a defined threshold (cost, complexity, or cross-domain impact) must go through ARB review.

Review criteria: - Does it align to a documented capability gap? - Does it duplicate existing capability in the app inventory? - Does it comply with tech reference architecture? - Does it meet data/integration standards? - Does it follow approved security patterns?

Outcomes: - Approve - Approve with conditions (must remediate specific items before next phase) - Hold (requires more information) - Reject / Redirect (does not meet minimum criteria, send back)

The deviation process: Teams that need to deviate from standards submit a deviation request with justification, risk assessment, and sunset plan (when and how they'll align). Deviations are time-boxed and tracked.

COMMON GOVERNANCE FAILURES

Failure	Pattern	Fix
Rubber stamp	ARB approves everything; no real review	Define pass/fail criteria; make rejection a legitimate outcome
Too late	EA reviews finished designs, not requirements	Move ARB upstream to conceptual/design phase
No enforcement	Teams bypass ARB; no consequence	Connect ARB approval to funding release
No deviation process	Teams deviate silently because the process is too rigid	Create a formal, fast deviation pathway
EA not maintained	Standards are outdated; teams don't trust them	Treat EA artifacts as living products, not one-time deliverables
Governance without authority	CTO has no formal decision rights	Establish formal authority in policy/charter documents

EA GOVERNANCE IN THE DOD CONTEXT

In DoD, EA governance is formalized through:

- **PPBE (Planning, Programming, Budgeting, and Execution)** — the investment governance cycle. EA should feed the programming phase.
- **JCIDS (Joint Capabilities Integration and Development System)** — requirements governance. EA capability maps and gap analysis should inform ICDs and CDDs.
- **RMF (Risk Management Framework)** — security/ATO governance. EA tech standards and security patterns should align with RMF controls.
- **CIO/G6 / CTO functions** — platform governance, data governance, integration standards.
- **CDAO (Chief Digital and AI Office)** — enterprise data strategy, AI governance, policy.

The EA governance function should be explicitly mapped to these processes — where does EA input go into each? Who carries it? What's the artifact?

GOVERNANCE METRICS THAT MATTER

EA governance is notoriously hard to measure. Avoid vanity metrics (number of views produced, number of ARB reviews conducted). Focus on outcomes:

Metric	What It Measures
% of new programs that passed ARB before design freeze	Upstream governance penetration
% of systems in app inventory that comply with tech reference architecture	Standards adoption
Number of capability duplicates identified and retired	Rationalization effectiveness
Time from capability gap identification to funded program	Strategy-to-execution velocity
% of data entities with designated authoritative source	Data governance maturity
Number of ARB deviation requests (and approval rate)	Standards pressure and flexibility

THE CHARTER: EA GOVERNANCE REQUIRES FORMAL AUTHORITY

EA governance cannot operate on goodwill. It needs a formal charter that specifies:

- **Scope:** What decisions require EA review?
- **Authority:** What can EA approve, condition, or reject?
- **Accountability:** Who enforces compliance?
- **Escalation:** What happens when programs push back?
- **Appeals:** How does a program contest an EA decision?

Without a charter, EA is advisory. Advisory EA can be — and will be — ignored when it's inconvenient.

Previous: [03 — Artifacts and Views](#) | Next: [05 — Military and Defense Application](#)