

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

ARCHITECTURE REFERENCE

# ODT-EA



---

## Enterprise Architecture — 01: Foundation

---

*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: 2 title: "EA Foundation"

---

---

## ENTERPRISE ARCHITECTURE — 01: FOUNDATION

---

What it is, what it isn't, and why it exists

---

### THE ONE-SENTENCE DEFINITION

**Enterprise Architecture is the discipline of making enterprise change deliberate, coherent, and governable.**

Everything else — frameworks, diagrams, capability maps, technology standards — is in service of that sentence. EA is not a product. It is not a diagram. It is not an IT department. It is a *decision-support function* for transformation.

---

### WHY IT EXISTS

Organizations change constantly: new strategies, new technology, new threats, new regulations, new structures. Without EA, each change is made locally, with local logic, optimizing for local outcomes. The result is:

- **Fragmentation** — systems that don't talk to each other
- **Duplication** — three teams solving the same problem independently
- **Opacity** — nobody knows what the enterprise actually looks like right now
- **Decision debt** — future changes become harder and costlier because of hidden dependencies

EA exists to interrupt that pattern. It gives leadership a *shared model of the enterprise* so that change decisions are made with visibility into consequences, dependencies, and trade-offs.

---

## WHAT "ENTERPRISE" ACTUALLY MEANS

This is where most EA efforts go wrong. "Enterprise" does not mean "IT systems." It means the **full sociotechnical system**:

Domain	What It Covers
People	Roles, responsibilities, org structures, culture
Processes	How work actually gets done, end-to-end
Information	What data exists, what it means, who owns it
Technology	Systems, infrastructure, tools, integrations
Strategy	Goals, outcomes, priorities, constraints

EA spans all of these. An architecture that only describes IT systems is *solution architecture*, not enterprise architecture. The difference matters because most enterprise problems are cross-domain — they involve people, process, data, AND technology simultaneously.

## THE SCOPE BOUNDARY

Scope	Name	Example
One solution, bounded	Solution Architecture	Design a data pipeline for a specific mission
Cross-solution, cross-org	Enterprise Architecture	Define how all data pipelines relate to each other and to mission outcomes

EA operates at the **coherence layer** — it doesn't design individual solutions, it defines the rules, patterns, and structures that make individual solutions work together.

## WHAT EA ACTUALLY PRODUCES

EA produces **coherence mechanisms**, not just documents:

- **Principles** — what we optimize for (e.g., "data is a shared asset, not a system-owned asset")
- **Views and models** — shared understanding of the current and target state

- **Standards and patterns** — reusable building blocks so teams aren't starting from scratch
- **Roadmaps** — sequenced, dependency-aware change plans
- **Governance hooks** — the processes by which EA actually influences decisions

The last one is the most important. EA that doesn't show up in investment decisions, acquisition requirements, and solution reviews is shelfware.

---

## EA IS ONLY REAL IF IT SHOWS UP HERE

- Investment and portfolio decisions (what gets funded)
- Solution intake and review (what gets built)
- Acquisition and requirements (what gets procured)
- Data and metadata standards (what gets trusted)
- Operational adoption (what gets used)

If you can't point to a decision that EA influenced, EA doesn't exist — it's just documentation.

---

## THE RELATIONSHIP TO OTHER DISCIPLINES

EA is not a standalone function. It is the **integration layer** between:

```
graph TD; A[Strategy & Governance] --> B[Enterprise Architecture]; B --> C[Portfolio, Programs, Projects]; C --> D[Operations and Execution];
```

Strategy & Governance  
↓  
Enterprise Architecture  
↓  
Portfolio, Programs, Projects  
↓  
Operations and Execution

EA actively connects to: performance management, portfolio governance, risk analysis, compliance, information/metadata management, org design, systems thinking, and knowledge management. It is the connective tissue, not the muscle.

---

## COMMON FAILURE MODES

Failure Mode	What It Looks Like
Artifact-first	Producing diagrams and frameworks with no connection to decisions
IT-only scope	Treating EA as a technology standards function
No governance hook	EA exists but has no formal role in intake, funding, or review
Too abstract	Models that nobody can use operationally
Too rigid	EA that can't adapt to changing strategy or environment

## ONE-LINE SUMMARY PER AUDIENCE

- **To a CTO:** EA is how you govern transformation coherently across domains.
- **To a program manager:** EA is the map that tells you what your program depends on and what depends on you.
- **To a data engineer:** EA is the authoritative source of truth for how data, systems, and processes are supposed to relate.
- **To leadership:** EA is how you make investment decisions that compound instead of conflict.

Next: [02 — Schools of Thought](#)