

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

ARCHITECTURE REFERENCE

# ODT-CDA



---

## AVT25 Assessment Tools: Exponential Work Multiplication

---

*Quick Reference - The Same Enclave Problem*

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: 3 title: "AVT25 Assessment — Exponential Work Multiplication"

---

## AVT25 ASSESSMENT TOOLS: EXPONENTIAL WORK MULTIPLICATION

---

### QUICK REFERENCE - THE SAME ENCLAVE PROBLEM

#### The Problem in Numbers

---

**5 tools in SAME ENCLAVE, all doing assessment per ADP 5-0:**

1. Strategic Assessments App
2. IMO/LOO Tool
3. Product Tracker
4. Problem Intake Form
5. Lessons Learned System

**Each tool independently implemented:** - Assessment object - Finding/Observation object - Recommendation object

**Total implementations:** 5 tools × 3 primitives = **15 separate implementations** of the same doctrinal concepts

---

#### Time Cost Per Assessment

---

##### Before DBO (AVT25 Pattern)

**Analyst Workflow:** 1. Conduct assessment (60 min) 2. Enter in Strategic Assessments App (30 min) 3. Re-enter in IMO/LOO Tool (30 min) 4. Re-enter in Product Tracker (20 min) 5. Re-enter in Problem Intake Form (20 min) 6. Re-enter in Lessons Learned (20 min) 7. **Total: 180 minutes (3 hours)**

**Data Quality:** Degrades with each re-entry - Typos introduced - Different phrasing causes confusion - Fields omitted due to fatigue - Inconsistent severity assessments

**Maintenance Burden:** - Bug in assessment logic? Fix 5 times - New field needed? Add to 5 schemas - Schema change? Test 5 tools

---

### After DBO (Target)

**Analyst Workflow:** 1. Conduct assessment (60 min) 2. Enter once using DBO Assessment primitives (5 min) 3. **Total: 65 minutes**

**Time Savings:** 115 minutes per assessment (**64% reduction**)

**Data Quality:** Perfect - Single entry, zero transcription errors - Consistent across all tools - All fields captured once

**Maintenance Burden:** - Bug fix: Once, all tools benefit - New field: Add to one schema - Schema change: Test unified backend

---

### Scaling Problem

---

#### Scenario: Add 6th Assessment Tool

**Before DBO (AVT25 Pattern):** - New tool must implement: Assessment, Finding, Recommendation - Now have 6 tools × 3 primitives = **18 implementations** - Analyst workflow now 6 tools × 30 min = **210 minutes (3.5 hours)** - Maintenance burden: **6x** (fix bugs 6 times)

**Scaling Factor: Exponential ( $O(n^2)$ )** - Each new tool adds burden to ALL existing tools - Each new primitive must be replicated n times - Query "across all assessment tools" becomes UNION across n tables

**After DBO:** - New tool consumes existing Assessment primitives - Still have 3 implementations total (shared backend) - Analyst workflow: Still **65 minutes** (enter once, available to 6 tools) - Maintenance burden: **1x** (fix once)

**Scaling Factor: Linear ( $O(n)$ )** - New tools are additive, not multiplicative - Primitives remain constant - Query "across all assessment tools" is one SELECT

---

### Query Capability

---

**"Show me all critical findings from last 30 days"**

**Before DBO (AVT25):**

```
-- IMPOSSIBLE without manual aggregation
-- Would need to:
-- 1. Query Strategic Assessments App database
```

```
-- 2. Export to Excel
-- 3. Query IMO/LOO Tool database
-- 4. Export to Excel
-- 5. Query Product Tracker database
-- 6. Export to Excel
-- 7. Query Problem Intake database
-- 8. Export to Excel
-- 9. Query Lessons Learned database
-- 10. Export to Excel
-- 11. Manually combine 5 Excel files
-- 12. Deduplicate (same finding entered 5 times)
-- 13. Format for leadership brief
```

Time: 2-4 hours of manual work  
Result: Inconsistent, error-prone

### After DBO:

```
SELECT
  a.title,
  af.finding_text,
  af.severity,
  tac.tool_id as source_tool,
  a.assessed_by,
  a.assessment_date
FROM assessments a
JOIN assessment_findings af ON a.assessment_id = af.assessment_id
LEFT JOIN tool_assessment_contexts tac ON a.assessment_id = tac.assessment_id
WHERE af.severity = 'critical'
      AND a.assessment_date >= CURRENT_DATE - INTERVAL '30 days'
ORDER BY a.assessment_date DESC;
```

**Time:** 30 seconds **Result:** Comprehensive, accurate, real-time

---

## Maintenance Cost

### Bug Fix Example: "Assessment date validation is wrong"

**Before DBO:** 1. Fix in Strategic Assessments App (4 hours dev + 2 hours test) 2. Fix in IMO/LOO Tool (4 hours dev + 2 hours test) 3. Fix in Product Tracker (4 hours dev + 2 hours test) 4. Fix in Problem Intake (4 hours dev + 2 hours test) 5. Fix in Lessons Learned (4 hours dev + 2 hours test)

**Total:** 30 hours (nearly 1 work week)

**After DBO:** 1. Fix in Assessment primitive (4 hours dev + 2 hours test) 2. All 5 tools inherit the fix automatically

**Total:** 6 hours **Savings:** 24 hours (80% reduction)

---

## Feature Addition Example: "Add confidence level to assessments"

**Before DBO:** 1. Update Strategic Assessments schema + UI (8 hours) 2. Update IMO/LOO schema + UI (8 hours) 3. Update Product Tracker schema + UI (8 hours) 4. Update Problem Intake schema + UI (8 hours) 5. Update Lessons Learned schema + UI (8 hours) 6. Migrate existing data in all 5 databases (12 hours)

**Total:** 52 hours (1.3 work weeks)

**After DBO:** 1. Add confidence\_level to Assessment primitive (2 hours) 2. Update shared backend (2 hours) 3. Migrate existing data once (2 hours) 4. Update tool UIs to display new field (5 tools × 1 hour = 5 hours)

**Total:** 11 hours **Savings:** 41 hours (79% reduction)

## THE MATH OF EXPONENTIAL WORK

### Cost Model

**Variables:** - **n** = number of tools - **p** = number of primitives (Assessment, Finding, Recommendation) - **t** = time per entry - **m** = maintenance time per change

**Before DBO (AVT25 Pattern):** - **Implementations:**  $n \times p$  (grows with both dimensions) - **Entry time:**  $n \times t$  (analyst enters  $n$  times) - **Maintenance:**  $n \times m$  (fix in  $n$  places) - **Queries:**  $O(n)$  complexity (UNION across  $n$  tables)

**After DBO:** - **Implementations:**  $p$  (constant, regardless of  $n$ ) - **Entry time:**  $t$  (analyst enters once) - **Maintenance:**  $m$  (fix once) - **Queries:**  $O(1)$  complexity (single SELECT)

### Real Numbers (AVT25 Assessment Tools)

Metric	Before (n=5)	After (DBO)	Savings
Primitive implementations	15	3	80%
Time per assessment	180 min	65 min	64%
Time per bug fix	30 hrs	6 hrs	80%
Time per feature add	52 hrs	11 hrs	79%
Time for cross-tool query	2-4 hrs manual	30 sec	>99%

## Projected: If we add 5 more assessment tools (n=10)

Metric	Before (n=10)	After (DBO)	Savings
Primitive implementations	30	3	90%
Time per assessment	330 min	65 min	80%
Time per bug fix	60 hrs	6 hrs	90%
Time per feature add	102 hrs	11 hrs	89%

**Pattern:** Savings increase as n increases (exponential vs. linear scaling)

## SCALING ASSESSMENT

Propagating assessment tools without a shared backend does not scale linearly — it multiplies the work exponentially. Every new tool added under the AVT25 pattern increases burden across all existing tools. This is mathematically unsustainable.

**Principle:** Every new assessment tool adds burden to ALL existing tools. The work doesn't scale linearly — it scales exponentially.

## DOCTRINE AS THE STARTING POINT

The distinction between what end users request and what doctrine requires must be maintained. Doctrine is the authoritative starting point — not individual tool team preferences.

**Principle:** Don't ask each tool team "what Assessment object do you want?" (results in 5 different answers). Ask ADP 5-0 "what is Assessment?" (one answer). Build it once. All tools consume it.

## THE SOLUTION: DBO CONVERGENCE

### Three-Phase Fix

**Phase 1: Build Assessment Primitives from ADP 5-0 (30 days)** - Assessment (identity) - AssessmentFinding (identity) - AssessmentRecommendation (identity) - ToolAssessmentContext (classification/relationship)

**Phase 2: Migrate Strategic App + IMO/LOO Tool** (90 days) - Demonstrates shared backend working - Proves 64% time savings - Validates cross-tool queries

**Phase 3: Migrate Remaining 3 Tools** (90 days) - Product Tracker - Problem Intake Form - Lessons Learned System

**Result:** - 5 tools → 1 backend - 180 min → 65 min per assessment - Exponential scaling → Linear scaling - Manual queries → SQL queries - Unsustainable → Sustainable

---

## BOTTOM LINE

**AVT25 Assessment Tools are the smoking gun:** Same enclave. Same doctrinal function. Zero shared foundation. Work multiplying exponentially.

**This isn't a cross-enclave problem (CDS). This isn't a cross-workshop problem (doctrine drift). This is a fundamental failure to build shared primitives from doctrine.**

**If we can't get assessment tools in the same enclave to share a backend, we've already lost.**

**Doctrine is gospel. Start there.**