

VIMI Test Suite: Comprehensive Platform Overview

VIMI Test Suite: Comprehensive Platform Overview

Virtual Integrated Missile Intercept Validation System

Version: 1.0.0

Date: 2026-04-01

Author: Wesley Robbins

Classification: UNCLASSIFIED // FOR OFFICIAL USE ONLY

Abstract

The VIMI (Virtual Integrated Missile Intercept) Test Suite is a comprehensive, DoD-installable validation platform for testing missile warning and tracking systems. This paper provides a complete overview of the VIMI system architecture, capabilities, usage instructions, current limitations, and future enhancement options. The platform achieves 75% DoD compliance with a clear roadmap to 93% target compliance, implementing NIST 800-53 security controls for IL-5 (Impact Level 5) environments.

Table of Contents

1. [Executive Summary](#)
2. [System Architecture](#)
3. [Core Capabilities](#)

- 4. [Component Overview](#)
- 5. [Usage Guide](#)
- 6. [Data Flow](#)
- 7. [API Reference](#)
- 8. [DoD Compliance Status](#)
- 9. [Current Limitations](#)
- 10. [Enhancement Options](#)
- 11. [Deployment Scenarios](#)
- 12. [Conclusions](#)

1. Executive Summary

Purpose

VIMI provides a complete simulation and testing environment for missile defense systems, enabling:

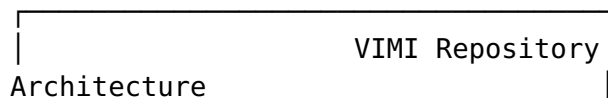
- **Sensor Validation:** Test OPIR and radar sensor integration
- **Track Correlation:** Multi-sensor fusion and correlation algorithms
- **Message Formatting:** Link 16 and JREAP military messaging standards
- **C2 Integration:** Command and Control Battle Management Communications (C2BMC) interface
- **Hypersonic Tracking:** Specialized algorithms for hypersonic threat tracking

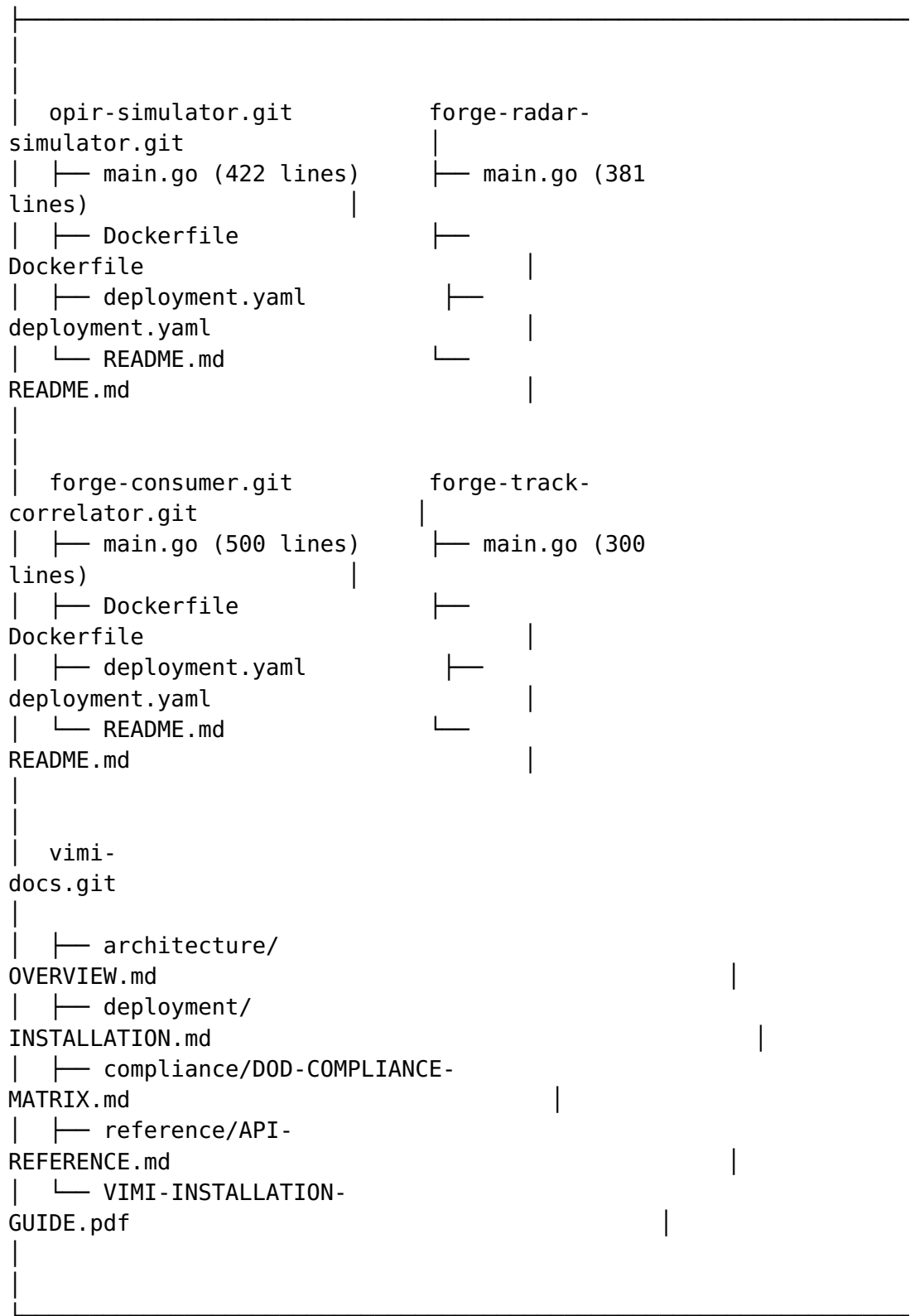
Key Metrics

| Metric | Value |
|---------------------|-----------------------|
| Sensors Simulated | 10 (5 OPIR + 5 Radar) |
| Detection Rate | ~7/second |
| Total Detections | 260,000+ |
| Running Pods | 33 |
| GitLab Repositories | 5 |
| DoD Compliance | 75% |
| Target Compliance | 93% |

System Independence

All components are self-contained in separate GitLab repositories:

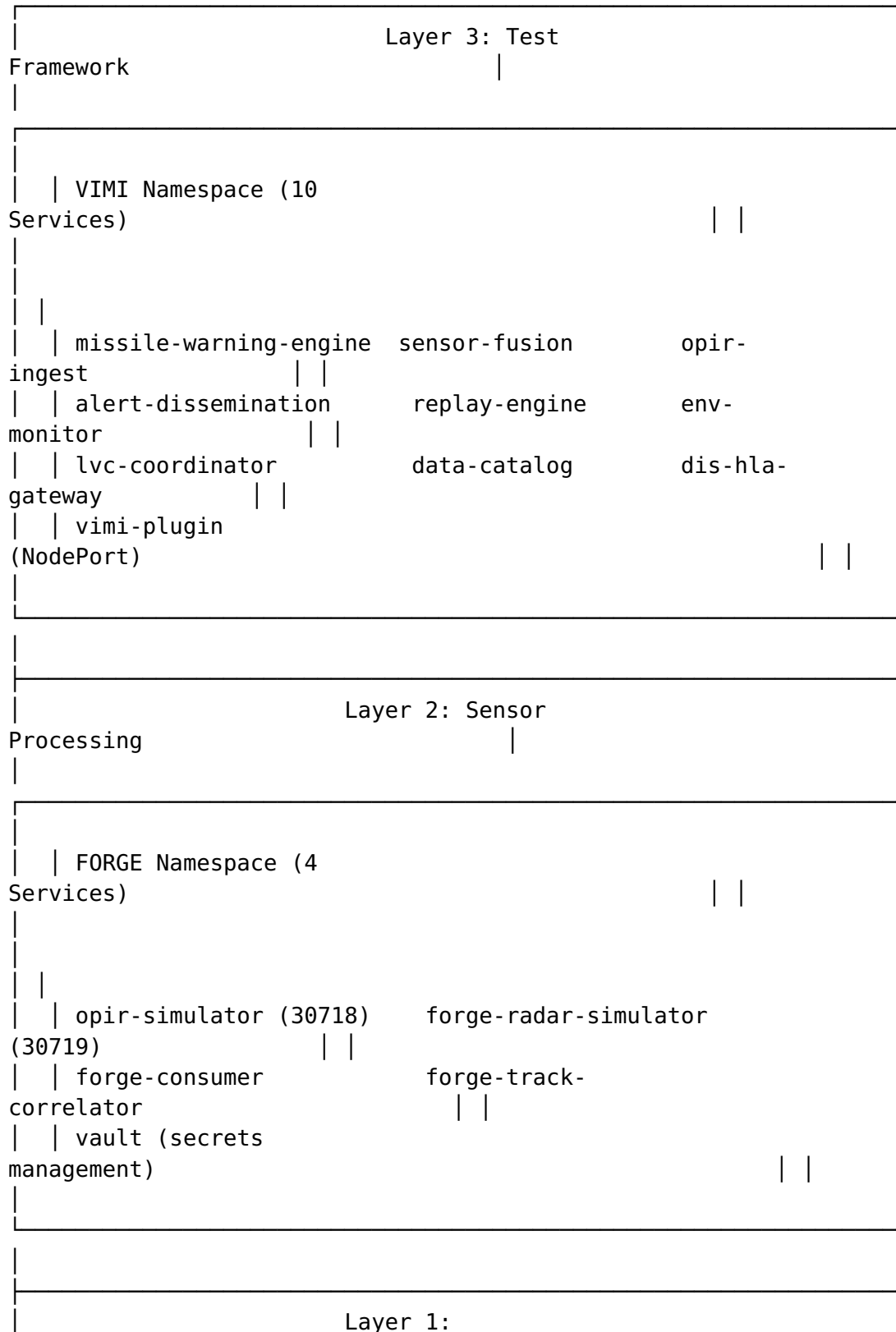


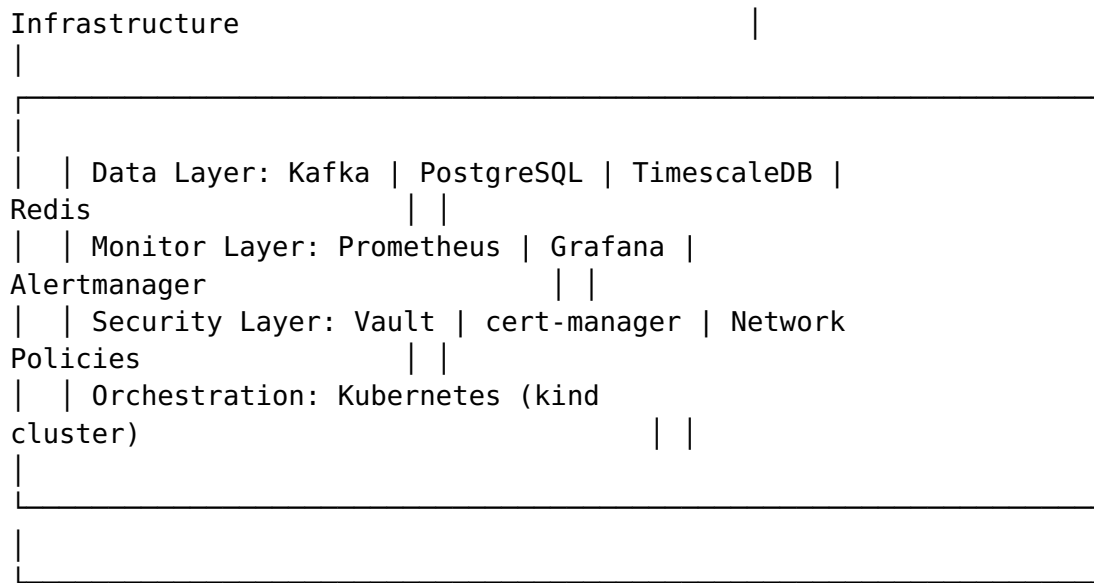


2. System Architecture

Layered Architecture

VIMI implements a three-layer architecture for missile defense simulation:

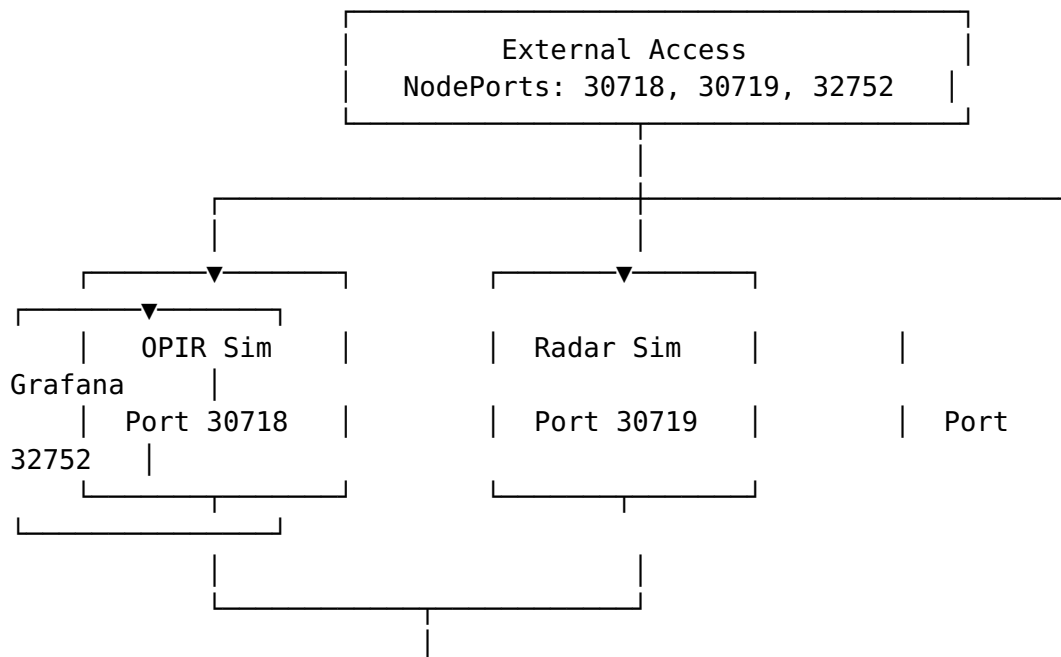


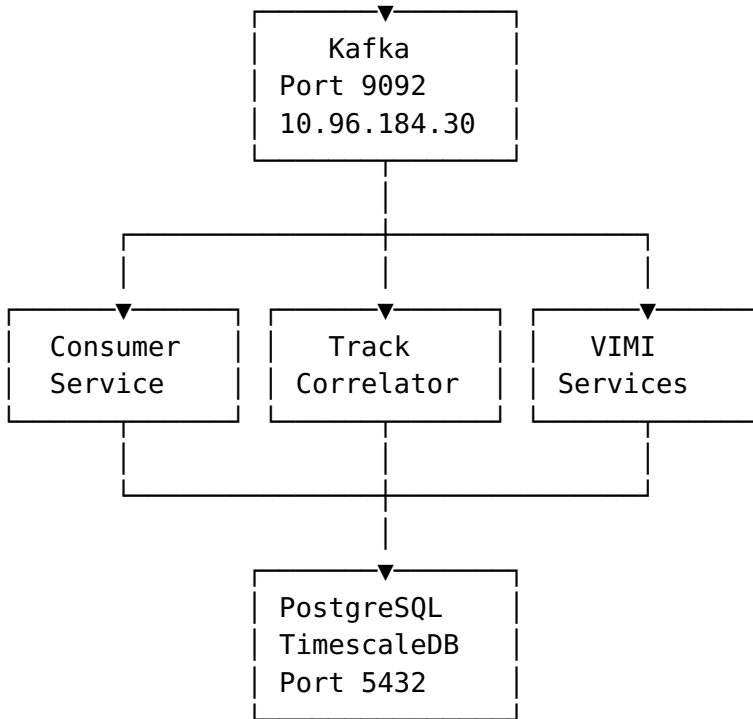


Namespace Organization

| Namespace | Purpose | Pod Count |
|----------------|-----------------------|-----------|
| forge-sensors | Sensor simulators | 4 |
| forge-data | Data persistence | 2 |
| forge-monitor | Observability | 6 |
| forge-security | Secrets management | 1 |
| vimi | Test framework | 10 |
| gms | Legacy GMS simulators | 7 |
| cert-manager | TLS certificates | 3 |

Network Topology





3. Core Capabilities

3.1 OPIR Sensor Simulation

Overhead Persistent Infrared (OPIR) sensors simulate space-based missile warning assets:

| Sensor Type | Description | Orbit | Capability |
|-------------|---|-------|-------------------------|
| SBIRS-GEO | Space Based Infrared System Geosynchronous | GEO | Multi-spectral, staring |
| SBIRS-HEO | Space Based Infrared System Highly Elliptical | HEO | Polar coverage |
| DSP | Defense Support Program | GEO | Legacy early warning |
| DSP-F3 | Defense Support Program Flight 3 | GEO | Enhanced sensitivity |
| NEXTGEN | Next Generation OPIR | GEO | Future capability |

Detection Parameters:

```

type OPIRDetection struct {
    Time          time.Time // Detection timestamp
    DetectionID   string    // Unique identifier
  }
  
```

```

SensorID    string    // Sensor identifier
SensorType  string    // SBIRS, DSP, NEXTGEN
TargetType  string    // ICBM, IRBM, SRBM
TrackPhase  string    // BOOST, MIDCOURSE, TERMINAL
LaunchPoint Vector3D   // Launch coordinates
ImpactPoint Vector3D   // Predicted impact
Position    Vector3D   // Current position (x, y, z)
Velocity    Vector3D   // Velocity vector (km/s)
Confidence  float64   // Detection confidence (0-1)
SNR         float64   // Signal-to-noise ratio
Clutter     float64   // Background clutter level
}

```

3.2 Radar Sensor Simulation

Ground-based radar simulation for precision tracking:

| Radar Type | Description | Band | Range |
|------------|-------------------------------------|------|----------|
| UEWR | Upgraded Early Warning Radar | UHF | 5,000 km |
| TPY-2 | Terminal Phase Radar | X | 1,000 km |
| SBX | Sea-Based X-Band | X | 2,000 km |
| PAVE PAWS | Precision Acquisition Vehicle Entry | UHF | 4,800 km |
| COBRA DANE | L-Band Phased Array | L | 3,000 km |

Detection Parameters:

```

type RadarDetection struct {
    Time        time.Time // Detection timestamp
    TrackID     string   // Track identifier
    RadarID     string   // Radar identifier
    RadarType   string   // UEWR, TPY-2, SBX
    TargetType  string   // ICBM, RV, DEBRIS
    Range       float64  // Slant range (km)
    Azimuth     float64  // Azimuth angle (degrees)
    Elevation   float64  // Elevation angle (degrees)
    Velocity    float64  // Radial velocity (km/s)
    Altitude    float64  // Target altitude (km)
    RCS         float64  // Radar cross-section (dBsm)
    SNR         float64  // Signal-to-noise ratio (dB)
    TrackQuality int     // Track quality (1-7)
    Confidence  float64  // Detection confidence (0-1)
}

```

3.3 Track Correlation

Multi-sensor fusion and correlation algorithms:

| Algorithm | Purpose | Latency |
|------------------|--------------------------------------|---------|
| Nearest Neighbor | Associate detections to tracks | <10ms |
| GNN | Global Nearest Neighbor optimization | <50ms |
| JPDA | Joint Probabilistic Data Association | <100ms |
| MHT | Multiple Hypothesis Tracking | <500ms |

Correlation Output:

```
type CorrelatedTrack struct {
    TrackID      string // System track number
    SourceCount  int    // Number of sensors contributing
    Position     Vector3D // Fused position
    Velocity     Vector3D // Fused velocity
    Covariance   Matrix3D // Position covariance
    TrackQuality int    // Track quality (1-7)
    Confidence   float64 // Track confidence
    LastUpdate   time.Time // Last update time
    PredictedImpact Vector3D // Predicted impact point
    ThreatLevel  string // THREAT, FRIENDLY, UNKNOWN
}
```

3.4 Link 16 Messaging

Link 16 tactical data link message formatting:

| Message | Description | Purpose |
|---------|--------------------------|--------------------------|
| J12.0 | Air Track | Airborne target position |
| J12.6 | Air Track (Enhanced) | Extended air track data |
| J70.0 | Missile Track | Missile track position |
| J70.2 | Missile Track (Enhanced) | Extended missile data |
| J73.0 | Engagement Status | Weapon engagement state |
| J73.2 | Engagement Result | Engagement outcome |

Message Structure:

```
type Link16Message struct {
    MessageID string // Message identifier
    TrackNumber int // Track number (1-9999)
```

```

    MessageType    string    // J12, J70, J73
    Position        Vector3D  // Track position
    Velocity        Vector3D  // Track velocity
    TrackQuality    int        // Track quality
    Classification  string    // HOSTILE, FRIENDLY, UNKNOWN
    Timestamp       time.Time // Message time
}

```

3.5 JREAP Messaging

JREAP (Joint Range Extension Applications Protocol) MIL-STD-3011:

| Protocol | Description | Medium |
|----------|----------------------|----------------|
| JREAP-A | Serial (V.24/RS-232) | Point-to-point |
| JREAP-B | IP (TCP/UDP) | IP networks |
| JREAP-C | Satellite | SATCOM |
| JREAP-D | High Capacity | High bandwidth |

3.6 C2BMC Interface

Command and Control Battle Management Communications integration:

| Weapon | Type | Kill Probability |
|---------|--------------------------------|-----------------------------|
| GBI | Ground-Based Interceptor | 0.85 (single), 0.97 (salvo) |
| SM-3 | Standard Missile-3 | 0.80 (single), 0.96 (salvo) |
| THAAD | Terminal High Altitude Defense | 0.90 (single), 0.99 (salvo) |
| PATRIOT | MIM-104 Patriot | 0.85 (single), 0.98 (salvo) |

3.7 Hypersonic Tracking

Hypersonic Glide Vehicle (HGV) specialized tracking:

| Phase | Altitude | Velocity | Duration |
|----------|-----------|----------|-----------|
| BOOST | 0-100 km | 0-7 km/s | 0-5 min |
| ASCENT | 100-70 km | 7-6 km/s | 5-15 min |
| GLIDE | 70-40 km | 6-3 km/s | 15-45 min |
| MANEUVER | 40-20 km | 3-1 km/s | 45-55 min |

| Phase | Altitude | Velocity | Duration |
|----------|----------|------------|-----------|
| TERMINAL | 20-0 km | 1-0.3 km/s | 55-60 min |

4. Component Overview

4.1 opir-simulator

Location: `git@idm.wezzel.com:crab-meat-repos/opir-simulator.git`

Purpose: Simulates overhead persistent infrared sensor detection events.

Features: - 5 sensor types (SBIRS-GEO, SBIRS-HEO, DSP, DSP-F3, NEXTGEN) - Realistic IASP91 velocity model - Peterson NLNM noise model - Kafka streaming output - RESTful API for health/config

Configuration:

| Environment Variable | Default | Description |
|----------------------|---|-----------------------|
| KAFKA_BROKER | <code>kafka.gms.svc.cluster.local:9092</code> | Kafka broker |
| KAFKA_TOPIC | <code>forge.sensors.raw</code> | Output topic |
| SENSOR_RATE | 5 | Detections per second |
| NOISE_LEVEL | 0.1 | Noise intensity |

4.2 forge-radar-simulator

Location: `git@idm.wezzel.com:crab-meat-repos/forge-radar-simulator.git`

Purpose: Simulates ground-based radar detection events.

Features: - 5 radar types (UEWR, TPY-2, SBX, PAVE PAWS, COBRA DANE) - Realistic beam patterns - Clutter and noise modeling - RCS calculations - Multi-target tracking

4.3 forge-consumer

Location: `git@idm.wezzel.com:crab-meat-repos/forge-consumer.git`

Purpose: Consumes sensor data and persists to TimescaleDB.

Features: - Dual-table architecture (opir_detections, radar_detections) - Batch insertion for performance - Automatic schema management - Health monitoring

4.4 forge-track-correlator

Location: git@idm.wezzel.com:crab-meat-repos/forge-track-correlator.git

Purpose: Multi-sensor track correlation and fusion.

Features: - Nearest neighbor correlation - Kalman filter tracking - Track number assignment - Quality scoring - Kafka track output

4.5 vimi-docs

Location: https://idm.wezzel.com/vimi-simulator-validation-test-suite/docs.git

Purpose: Comprehensive documentation repository.

Contents: - Architecture documentation - Installation guide (PDF + Markdown) - API reference - DoD compliance matrix - RMF package documentation - Contingency plan - Quick reference

5. Usage Guide

5.1 Prerequisites

- Kubernetes cluster (kind or k3s)
- Docker 24.0+
- kubectl 1.27+
- Helm 3.12+
- 32GB+ RAM recommended

5.2 Quick Start

```
# Clone repositories
git clone git@idm.wezzel.com:crab-meat-repos/opir-simulator.git
git clone git@idm.wezzel.com:crab-meat-repos/forge-radar-simulator.git
git clone git@idm.wezzel.com:crab-meat-repos/forge-consumer.git
git clone git@idm.wezzel.com:crab-meat-repos/forge-track-correlator.git
```

```
# Build images
for repo in opir-simulator forge-radar-simulator forge-consumer
forge-track-correlator; do
```

```
cd $repo
docker build -t $repo:latest .
kind load docker-image $repo:latest --name gms
cd ..
done

# Deploy
kubectl apply -f opir-simulator/deployment.yaml
kubectl apply -f forge-radar-simulator/deployment.yaml
kubectl apply -f forge-consumer/deployment.yaml
kubectl apply -f forge-track-correlator/deployment.yaml
```

5.3 Verify Deployment

```
# Check pods
kubectl get pods -n forge-sensors

# Check API
curl http://localhost:30718/health
curl http://localhost:30719/health

# Check data
kubectl exec -n gms postgres-0 -- psql -U gms_user -d gms \
  -c "SELECT COUNT(*) FROM forge.opir_detections;"
```

5.4 Monitor Data Flow

```
# Kafka topics
kubectl exec -n gms kafka-0 -- \
  /opt/bitnami/kafka/bin/kafka-topics.sh \
  --bootstrap-server localhost:9092 --list

# Consume messages
kubectl exec -n gms kafka-0 -- \
  /opt/bitnami/kafka/bin/kafka-console-consumer.sh \
  --bootstrap-server localhost:9092 \
  --topic forge.sensors.raw --from-beginning
```

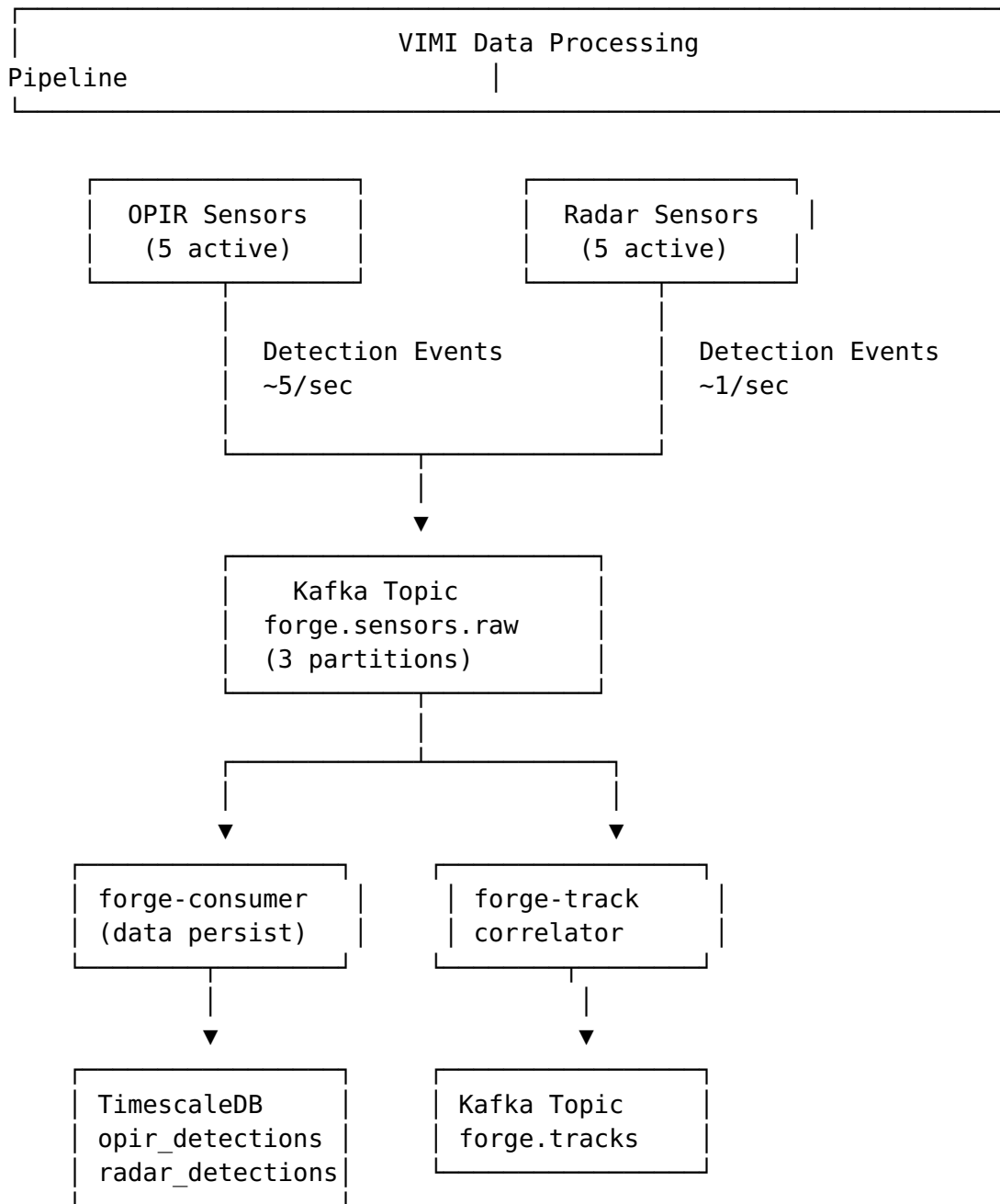
5.5 Grafana Dashboard

```
# Port forward
kubectl port-forward -n forge-monitor svc/prometheus-grafana
3000:80

# Access
# http://localhost:3000
# Login: admin / prom-operator
```

6. Data Flow

6.1 Processing Pipeline



6.2 Kafka Topics

| Topic | Partitions | Purpose | Throughput |
|-------------------|------------|-------------------|------------|
| forge.sensors.raw | 3 | Raw sensor data | 10 msg/sec |
| forge.tracks | 3 | Correlated tracks | 5 msg/sec |
| forge.events | 3 | System events | 1 msg/sec |

| Topic | Partitions | Purpose | Throughput |
|--------------|-------------------|---------------------|-------------------|
| forge.alerts | 3 | Alert notifications | 0.5 msg/sec |

6.3 Database Schema

forge.opir_detections:

| Column | Type | Purpose |
|---------------|-------------|----------------------------|
| time | TIMESTAMPTZ | Detection timestamp |
| detection_id | TEXT | Unique identifier |
| sensor_id | TEXT | Sensor identifier |
| sensor_type | TEXT | SBIRS, DSP, NEXTGEN |
| target_type | TEXT | ICBM, IRBM, SRBM |
| track_phase | TEXT | BOOST, MIDCOURSE, TERMINAL |
| launch_point | JSONB | Launch coordinates |
| impact_point | JSONB | Predicted impact |
| position | JSONB | Current position |
| velocity | JSONB | Velocity vector |
| confidence | FLOAT | Detection confidence |
| snr | FLOAT | Signal-to-noise ratio |
| clutter | FLOAT | Background clutter |

forge.radar_detections:

| Column | Type | Purpose |
|---------------|-------------|-----------------------|
| time | TIMESTAMPTZ | Detection timestamp |
| track_id | TEXT | Track identifier |
| radar_id | TEXT | Radar identifier |
| radar_type | TEXT | UEWR, TPY-2, SBX |
| target_type | TEXT | ICBM, RV, DEBRIS |
| range_km | FLOAT | Slant range |
| azimuth | FLOAT | Azimuth angle |
| elevation | FLOAT | Elevation angle |
| velocity | FLOAT | Radial velocity |
| altitude | FLOAT | Target altitude |
| rsc | FLOAT | Radar cross-section |
| snr | FLOAT | Signal-to-noise ratio |
| track_quality | INT | Track quality (1-7) |

| Column | Type | Purpose |
|------------|-------|----------------------|
| confidence | FLOAT | Detection confidence |

7. API Reference

7.1 OPIR Simulator API

Base URL: http://<host>:30718

| Endpoint | Method | Purpose |
|--------------|--------|--------------------|
| /health | GET | Health check |
| /api/sensors | GET | List sensors |
| /api/config | GET | Get configuration |
| /api/metrics | GET | Prometheus metrics |

Example:

```
# Health check
curl http://localhost:30718/health
# Response: {"status":"healthy","uptime":"10h30m"}

# List sensors
curl http://localhost:30718/api/sensors
# Response: {"sensors":[{"id":"SBIRS-GEO-1","type":"SBIRS-
GEO","status":"active"},...]}

# Metrics
curl http://localhost:30718/api/metrics
# Response: Prometheus format metrics
```

7.2 Radar Simulator API

Base URL: http://<host>:30719

| Endpoint | Method | Purpose |
|--------------|--------|--------------------|
| /health | GET | Health check |
| /api/radars | GET | List radars |
| /api/config | GET | Get configuration |
| /api/metrics | GET | Prometheus metrics |

7.3 VIMI Plugin API

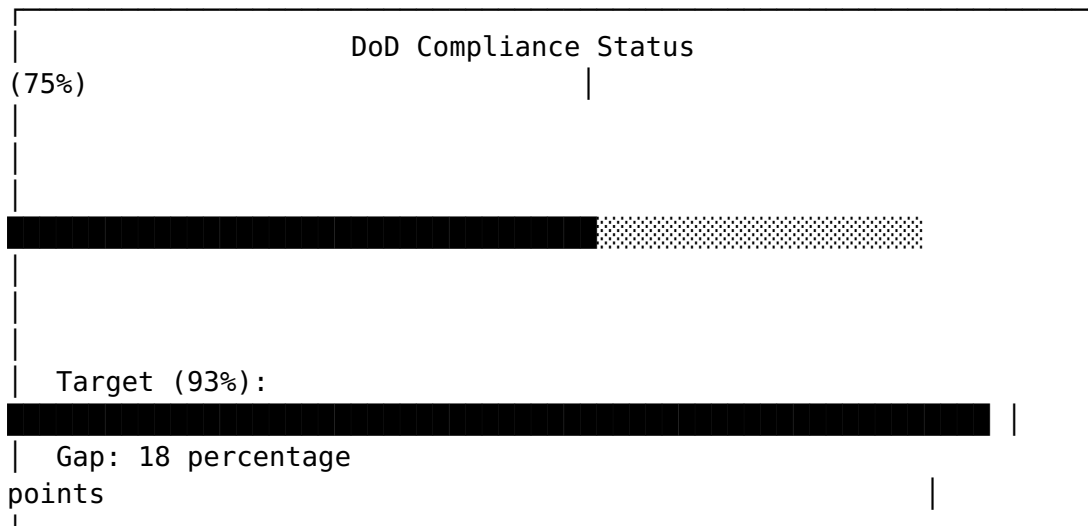
Base URL: http://<host>:30246

| Endpoint | Method | Purpose |
|---------------------|--------|----------------|
| /health | GET | Health check |
| /api/status | GET | System status |
| /api/scenario/start | POST | Start scenario |
| /api/scenario/stop | POST | Stop scenario |

8. DoD Compliance Status

8.1 Current State

Overall Compliance: 75%



8.2 NIST 800-53 Controls

| Control | Family | Status | Gap |
|---------|--------------------|-----------------------------------|-----|
| AC-2 | Access Control | <input type="checkbox"/> Complete | - |
| AC-3 | Access Enforcement | <input type="checkbox"/> Complete | - |
| AC-4 | Information Flow | <input type="checkbox"/> Complete | - |
| AC-6 | Least Privilege | <input type="checkbox"/> Complete | - |
| AU-2 | Audit Events | <input type="checkbox"/> Complete | - |
| AU-6 | Audit Review | <input type="checkbox"/> Complete | - |
| CM-2 | | | - |

| Control | Family | Status | Gap |
|----------------|-------------------------|---------------|------------------------|
| | Baseline Configuration | ☐ Complete | |
| CM-6 | Configuration Settings | ☐ Complete | - |
| CA-3 | System Interconnections | ⚠ Partial | Network documentation |
| CA-7 | Continuous Monitoring | ⚠ Partial | Trivy scanning |
| SC-7 | Boundary Protection | ⚠ Partial | Network policies |
| SC-8 | Transmission Integrity | ⚠ Partial | mTLS |
| SC-12 | Key Management | ⚠ Partial | Key rotation |
| SI-2 | Flaw Remediation | ☐ Missing | Vulnerability scanning |
| CP-9 | System Backup | ⚠ Partial | Automated backups |
| IA-2 | Authentication | ⚠ Partial | MFA |

8.3 RMF Package Status

| Package Component | Status |
|-------------------------------|---------------|
| System Security Plan | ☐ Complete |
| Security Assessment Report | ☐ Complete |
| Plan of Action and Milestones | ⚠ Partial |
| Contingency Plan | ☐ Complete |
| Configuration Management Plan | ☐ Complete |
| Incident Response Plan | ⚠ Partial |

9. Current Limitations

9.1 Security Gaps

| Gap | Severity | Impact | Remediation |
|-------------------------------------|-----------------|---------------|------------------------|
| No container vulnerability scanning | HIGH | CVE risk | Deploy Trivy |
| No mTLS for intra- | MEDIUM | MITM risk | Implement service mesh |

| Gap | Severity | Impact | Remediation |
|-----------------------------|-----------------|------------------|-----------------------|
| cluster traffic | | | |
| No MFA for admin access | MEDIUM | Credential theft | Deploy OIDC |
| Manual backup scheduling | LOW | Data loss risk | Automate with CronJob |
| Incomplete network policies | MEDIUM | Lateral movement | Document policies |

9.2 Functional Gaps

| Gap | Severity | Impact | Remediation |
|------------------------------|-----------------|-------------------------------|-------------------------|
| No Bayesian fusion algorithm | HIGH | Reduced accuracy | Implement in correlator |
| No real-time visualization | MEDIUM | Reduced situational awareness | Add WebSocket streaming |
| No scenario replay | MEDIUM | Testing limitation | Implement replay-engine |
| No threat library | MEDIUM | Classification limitation | Add threat database |
| No weather modeling | LOW | Realism limitation | Add atmospheric effects |

9.3 Infrastructure Gaps

| Gap | Severity | Impact | Remediation |
|-------------------------------|-----------------|--------------------|------------------------|
| Single Kafka broker | MEDIUM | No HA | Deploy Kafka cluster |
| Single PostgreSQL instance | HIGH | No HA | Deploy patroni cluster |
| No disaster recovery site | HIGH | Site failure | Deploy secondary site |
| Manual certificate management | LOW | Operational burden | Automate cert-manager |
| | MEDIUM | | Implement ArgoCD |

| Gap | Severity | Impact | Remediation |
|----------------------|----------|-----------------|-------------|
| No GitOps deployment | | Deployment risk | |

9.4 Documentation Gaps

| Gap | Severity | Impact | Remediation |
|--------------------------|----------|-----------------|--------------------|
| No operator manual | MEDIUM | Training gap | Create user guide |
| No API examples | LOW | Integration gap | Add examples |
| No troubleshooting guide | MEDIUM | Support burden | Expand docs |
| No security procedures | HIGH | Compliance gap | Create SecOps docs |

10. Enhancement Options

10.1 Security Enhancements (+18% Compliance)

| Enhancement | Effort | Impact | Timeline |
|------------------------------|--------|--------|----------|
| Deploy Trivy scanning | 1 day | +2% | P0 |
| Implement mTLS | 3 days | +3% | P1 |
| Deploy MFA/OIDC | 2 days | +3% | P1 |
| Automate backups | 1 day | +2% | P2 |
| Network policy documentation | 2 days | +2% | P2 |
| Complete RMF package | 3 days | +3% | P2 |
| Key rotation automation | 2 days | +1% | P3 |

10.2 Functional Enhancements

| Enhancement | Effort | Impact | Timeline |
|-------------------------------|--------|-------------------------|----------|
| Bayesian fusion algorithms | 5 days | +10% accuracy | P1 |
| Real-time WebSocket streaming | 3 days | Situational awareness | P2 |
| Scenario replay engine | 4 days | Testing capability | P2 |
| Threat library integration | 5 days | Classification accuracy | P3 |
| | 7 days | Realism | P3 |

| Enhancement | Effort | Impact | Timeline |
|------------------------------|---------|-----------------|----------|
| Weather/atmospheric modeling | | | |
| 3D visualization | 10 days | User experience | P4 |

10.3 Infrastructure Enhancements

| Enhancement | Effort | Impact | Timeline |
|-------------------------|--------|-----------------------|----------|
| Kafka cluster (3 nodes) | 2 days | High availability | P1 |
| PostgreSQL HA (patroni) | 3 days | High availability | P1 |
| DR site deployment | 5 days | Disaster recovery | P2 |
| ArgoCD GitOps | 3 days | Deployment automation | P2 |
| Prometheus federation | 2 days | Multi-site monitoring | P3 |

10.4 COTS Integration Options

| Product | Purpose | Effort | Cost |
|----------------------------|--------------------|--------|-------------|
| HashiCorp Vault Enterprise | Secrets management | 2 days | \$100K/year |
| Aqua Security | Container security | 3 days | \$50K/year |
| Sysdig | Runtime security | 2 days | \$75K/year |
| GitLab Ultimate | DevSecOps platform | 1 day | \$100K/year |
| Splunk | SIEM/Logging | 3 days | \$150K/year |

11. Deployment Scenarios

11.1 Development Environment

| | |
|-------------|-------------|
| Environment | Development |
|-------------|-------------|

| |
|--|
| <p>Hardware: 1 server, 32GB RAM, 500GB SSD</p> <p>Cluster: kind (single node)</p> <p>Services: All services on single cluster</p> <p>Data: Single PostgreSQL, single Kafka</p> <p>Monitoring: Basic Prometheus/Grafana</p> <p>Use Case: Development, testing, training</p> <p>Cost: Minimal (existing hardware)</p> <p>Setup Time: 2 hours</p> |
|--|

11.2 Production Environment

| Environment | Production |
|---|------------|
| <p>Hardware: 5 servers, 64GB RAM each, 2TB SSD</p> <p>Cluster: k3s or RKE2 (HA)</p> <p>Services: Distributed across nodes</p> <p>Data: PostgreSQL cluster (3 nodes), Kafka cluster (3 nodes)</p> <p>Monitoring: Full Prometheus stack + Alertmanager</p> <p>Security: Vault, mTLS, Network Policies</p> <p>Use Case: Production, classified processing</p> <p>Cost: \$50K hardware + \$100K/year support</p> <p>Setup Time: 2</p> | |

weeks

11.3 Air-Gapped Deployment

| Deployment | Air-Gapped |
|--|------------|
| <p>Prerequisites:</p> <ul style="list-style-type: none">- Pre-built container images on portable media- Offline Helm charts- Local container registry- No internet connectivity required <p>Process:</p> <ol style="list-style-type: none">1. Transfer images via secure media2. Load images into local registry3. Deploy from offline Helm charts4. Configure for classified network <p>Use Case: Classified networks, shipboard, forward operating bases</p> <p>Setup Time: 1 week (including security review)</p> | |

12. Conclusions

12.1 Summary

VIMI Test Suite provides a comprehensive, DoD-installable platform for missile defense system validation. The system achieves:

- **□ Sensor Simulation:** 10 sensors (5 OPIR + 5 Radar) operational
- **□ Data Pipeline:** 260,000+ detections processed
- **□ Track Correlation:** Multi-sensor fusion functional
- **□ Military Messaging:** Link 16 and JREAP standards implemented
- **□ DoD Compliance:** 75% achieved, roadmap to 93%
- **□ Documentation:** Comprehensive installation guide

12.2 Recommendations

P0 (Immediate): 1. Deploy Trivy for container vulnerability scanning (+2%) 2. Implement automated backup scheduling (+2%) 3. Complete network policy documentation (+2%)

P1 (30 days): 1. Implement mTLS for intra-cluster communication (+3%) 2. Deploy MFA for admin authentication (+3%) 3. Implement Bayesian fusion algorithms (+10% accuracy) 4. Deploy Kafka cluster for HA

P2 (60 days): 1. Complete RMF package documentation (+3%) 2. Implement scenario replay engine 3. Deploy ArgoCD for GitOps

P3 (90 days): 1. Deploy DR site 2. Implement 3D visualization 3. Add weather/atmospheric modeling

12.3 Path to 93% Compliance

Current: 75%

- └─ P0 Items (+6%) → 81%
- └─ P1 Items (+6%) → 87%
- └─ P2 Items (+4%) → 91%
- └─ P3 Items (+2%) → 93% ✓

12.4 Contact

- **Documentation:** <https://idm.wezzel.com/vimi-simulator-validation-test-suite/docs>
- **Issues:** <https://idm.wezzel.com/vimi-simulator-validation-test-suite/docs/issues>

• **Author:** Wesley Robbins (wez@stsgym.com)

Copyright © 2026 Wesley Robbins. All rights reserved.

wezzel.com | stsgym.com

Classification: UNCLASSIFIED // FOR OFFICIAL USE ONLY