OBINexus Pre-Grant Gated Development Architecture

Technical Documentation for DASA Defense Innovation Grant

Document Version: 2.0 | Classification: Public Release | Date: 2025

---

1. System Lifecycle Architecture

plantuml

```
@startuml OBINexusGatedLifecycle
!theme blueprint

skinparam backgroundColor #FEFEFE
skinparam activity {
  BackgroundColor #E3F2FD
  BorderColor #1976D2
  FontColor #0D47A1
  DiamondBackgroundColor #FFF3E0
  DiamondBorderColor #F57C00
}

|Pre-Gate Phase|
start
:Initialize Ilford Laboratory;
:Deploy CI/CD Pipeline;
:Establish Legal Entity;
:Configure Version Control;
note right
  Critical Infrastructure
  - Physical workspace setup
  - Automated testing framework
  - IP protection mechanisms
  - Stakeholder database
end note

|Pre-Gate Phase|
if (Pre-Gate Compliance ≥95%?) then (YES)
  |Development Phase|
  :Epistemic Manifold Development;
  :DIRAM Audit Engine Build;
  :Threat Resolver Training;
  :Integration Testing;
  note right
    Core Technical Artifacts
    - Mathematical proofs
    - Working prototypes
    - Performance benchmarks
    - Test result datasets
  end note
else (NO)
  |Pre-Gate Phase|
  :Execute Remediation Protocol;
  :Resource Reallocation;
  :Timeline Adjustment;
  stop
```

```
endif

|Development Phase|
if (Development Compliance ≥90%?) then (YES)
  |Post-Gate Phase|
  :Legacy Capsule Packaging;
  :Stakeholder Demonstrations;
  :Oxford Symposium Execution;
  :Letter Collection Campaign;
  note right
    Validation Artifacts
    - 5 support letters
    - Demo recordings
    - Academic feedback
    - Deployment guide
  end note
else (NO)
  |Development Phase|
  :Technical Pivot Protocol;
  :Performance Optimization;
  :Scope Refinement;
  stop
endif

|Post-Gate Phase|
if (Post-Gate Compliance =100%?) then (YES)
  |Submission Phase|
  :DASA Application Assembly;
  :Video Production;
  :Final Compliance Audit;
  :Submit Grant Package;
  note right
    Submission Package
    - Complete application
    - Technical appendices
    - Stakeholder evidence
    - Financial projections
  end note
  stop
else (NO)
  |Post-Gate Phase|
  :Critical Item Resolution;
  :Stakeholder Re-engagement;
  :Documentation Completion;
  stop
endif
```

```
@enduml
```

2. QA Compliance Matrix

Gate Transition Decision Matrix

Phase	Critical Compliance Items	TRUE Required	Transition Threshold	Verification Authority
Pre-Gate	LAB-001: Workspace Secured	Mandatory	95% Overall	Infrastructure Lead
	CICD-001: Automated Testing	Mandatory	(33/35 items)	Technical Architect
	FIN-005: Legal Entity	Mandatory		Compliance Officer
	DOC-002: Version Control	Required		
	STAKE-001: Stakeholder DB	Required		
Development	EPIS-003: Manifold Accuracy ≥90%	Mandatory	90% Overall	Algorithm Lead
	DIRAM-003: Rollback <50ms	Mandatory	(30/33 items)	Systems Engineer
	THREAT-001: Classification ≥95%	Mandatory		Safety Officer
	PROTO-001: Working Prototype	Required		
	PERF-003: Statistical Validation	Required		
Post-Gate	LEGACY-001: 5 Capsules Complete	Mandatory	100% Overall	Project Director
	SVAL-001: 5 Support Letters	Mandatory	(30/30 items)	Stakeholder Lead
	INTEL-001: Oxford Symposium	Mandatory		Academic Liaison
	DEPLOY-001: DASA Application	Mandatory		Grant Manager
	IP-004: Contributor Agreements	Mandatory		Legal Counsel

Binary Gate Logic

```
IF (Phase_Compliance >= Threshold) AND (ALL Critical_Items == TRUE) THEN
    AUTHORIZE Gate_Transition
ELSE
    INVOKE Remediation_Protocol
    RESET Timeline_Buffer
    RETRY Compliance_Check
END IF
```

3. SMART Goals and Gating Rules

Pre-Gate Phase (Weeks 1-4)

Objective	Critical Artifacts	Gate Requirement	Failure Fallback
Establish operational foundation for epistemic AI development	• Configured Ilford lab with robotics test area	95% compliance with mandatory LAB-001, CICD-001, FIN-005	Invoke seed investor contingency fund (£5K emergency allocation)
	• CI/CD pipeline with epistemic validation	Weekly gate reviews showing monotonic progress	Engage university partnership for shared workspace
	• IP-protected legal structure	Visitor NDA system operational	Contract IP attorney for expedited setup
	• Version-controlled documentation system	Git-LFS with encrypted backups	

Development Phase (Weeks 5-12)

Objective	Critical Artifacts	Gate Requirement	Failure Fallback
Validate epistemic actor architecture through working prototypes	• Epistemic manifold with proven stability	90% compliance with mandatory EPIS-003, DIRAM-003, THREAT-001	Reduce threat resolver scope to 3 categories
	• DIRAM audit trail with cryptographic integrity	Demonstrated <50ms rollback across 20 failure modes	Implement simplified Merkle tree variant
	• Threat gradient classifier at 95% accuracy	Benchtop robot executing epistemic decisions	Partner with robotics lab for hardware
	• Performance benchmark report (100+ trials)	Peer review from 2+ academic validators	Extend timeline by 2 weeks maximum

Post-Gate Phase (Weeks 13-16)

Objective	Critical Artifacts	Gate Requirement	Failure Fallback
Transform prototype into grant-ready initiative with stakeholder validation	• 5 legacy capsules with documentation	100% compliance across all 30 items	Cannot proceed - must achieve full compliance
	• 5 signed letters from diverse stakeholders	Oxford symposium with 15+ attendees	Leverage advisor network for introductions
	• Impact video (3-5 min, 1080p)	Mensa salon proposal accepted	Create virtual demonstration option
	• Complete DASA application package	All IP assignments executed	Emergency legal review session

4. Core System Components and Epistemic Governance

Epistemic Actor (EA) Architecture

The Epistemic Actor represents a foundational shift in autonomous system design, implementing bounded rationality through mathematically traceable knowledge states. Unlike traditional AI systems that operate on probabilistic confidence scores, the EA maintains explicit epistemic boundaries - regions where the system acknowledges the limits of its knowledge and adjusts behavior accordingly.

The EA architecture consists of three interconnected layers. The Knowledge Manifold Layer maintains a topological representation of the system's epistemic state space, where each point represents a specific configuration of beliefs, uncertainties, and evidence. The Transition Validation Layer ensures that movements through this manifold follow logically coherent paths, preventing epistemic leaps that would violate the system's philosophical foundations. The Action Binding Layer translates epistemic states into physical actuator commands, ensuring that behavioral outputs remain consistent with the system's knowledge limitations.

Gate transitions within the OBINexus project mirror this epistemic architecture. Just as the EA cannot transition between disconnected knowledge states, the project cannot advance through gates without demonstrated continuity of capabilities and validated achievements.

DIRAM (Directed Instruction Random-Access Mechanism)

DIRAM implements hardware-level epistemic constraints, creating an immutable audit trail of decision pathways that enables post-hoc analysis and real-time rollback capabilities. This mechanism operates as a specialized memory controller that intercepts all action commands before execution, validating them against the current epistemic state and historical decision patterns.

The DIRAM architecture employs a Merkle tree structure for cryptographic integrity, ensuring that no decision can be retroactively modified or deleted. Each node in the tree contains not just the decision outcome but also the complete epistemic context that led to that decision - the knowledge state, uncertainty bounds, and evidence basis. This comprehensive capture enables the sub-50ms rollback requirement by maintaining parallel decision branches until outcomes are validated.

Within the gated development framework, DIRAM principles govern documentation and decision tracking. Every gate transition decision is recorded with full context, creating an audit trail that satisfies both technical requirements and DASA compliance standards. This parallel between system architecture and project management demonstrates the coherence of epistemic principles across technical and organizational domains.

Threat Gradient Resolver

The Threat Gradient Resolver implements continuous risk assessment through multi-dimensional gradient analysis, moving beyond binary threat/non-threat classifications to nuanced contextual evaluation. This system recognizes that threat assessment in complex environments requires

understanding not just object identity but also behavioral context, temporal dynamics, and uncertainty propagation.

The resolver operates through three stages of analysis. Initial sensor fusion combines visual, thermal, and motion data into a unified scene representation. Gradient calculation then evaluates threat potential across multiple dimensions - kinetic energy, trajectory prediction, object classification confidence, and behavioral anomaly detection. The final contextual integration stage applies scenario-specific rules, such as distinguishing prosthetic limbs from weapons based on movement patterns and thermal signatures.

Project gating incorporates threat gradient principles through risk-weighted decision making. Each gate transition evaluates not just binary compliance but also the gradient of readiness across multiple dimensions. A project might achieve 95% Pre-Gate compliance but show concerning gradients in stakeholder engagement or technical risk, triggering enhanced monitoring rather than gate failure.

Epistemic Integrity Governance

Epistemic integrity serves as the philosophical backbone connecting all system components and governing project transitions. This integrity manifests through three principles that apply equally to technical systems and project management.

The Coherence Principle requires that all decisions and transitions maintain logical consistency with established knowledge foundations. In the EA system, this prevents contradictory beliefs from coexisting. In project management, this ensures that gate transitions reflect genuine capability rather than optimistic projections.

The Traceability Principle demands that every decision can be traced back through its epistemic lineage to foundational evidence. DIRAM implements this technically through its Merkle tree structure. Project gates implement this through comprehensive documentation requirements and audit trails.

The Humility Principle acknowledges that bounded rationality requires explicit recognition of uncertainty. The Threat Gradient Resolver embodies this by maintaining confidence intervals rather than absolute classifications. Gate transitions embody this by including explicit failure fallback paths and remediation protocols.

Abnormality Detection and Project Health

Abnormality detection within the OBINexus system extends beyond technical anomalies to encompass project health indicators. The system employs a Bayesian framework that maintains expectations about normal operating ranges and flags deviations for investigation.

Technical abnormality detection monitors system performance against established baselines. Epistemic state transitions that exceed velocity thresholds trigger safety protocols. Decision patterns that diverge from training distributions activate enhanced audit procedures. Hardware sensors that report values outside calibrated ranges initiate diagnostic routines.

Project abnormality detection applies similar principles to development metrics. Code commit velocity that drops below historical averages signals potential team burnout. Stakeholder engagement rates that decline week-over-week indicate relationship management issues. Budget burn rates that accelerate beyond projections trigger resource review protocols.

Both technical and project abnormality detection feed into gate transition decisions. A system demonstrating increasing abnormality rates cannot pass Development Gate review regardless of feature completion. A project showing team health abnormalities cannot proceed to Post-Gate activities without remediation.

5. Burn-Resistant Agile Implementation

Time-Boxed Sprint Architecture

The OBINexus development methodology implements fixed-duration sprints with enforced recovery periods, preventing the accumulation of technical debt and human exhaustion that plague traditional pre-grant development. Each two-week sprint follows a predictable cadence that enables both intensive progress and sustainable pacing.

Sprint boundaries are absolute, with no extensions permitted regardless of feature completion status. This rigid time-boxing forces prioritization decisions that reflect true project criticality rather than perfectionist tendencies. Incomplete features roll to subsequent sprints through a formal deferral process that documents reasons for delay and revised completion strategies.

Recovery periods between sprints are mandatory, not optional. These 48-hour buffers serve multiple purposes: psychological restoration for team members, integration testing for completed features, and strategic planning for upcoming sprints. During recovery periods, only critical bug fixes and documentation updates are permitted, with all new feature development explicitly prohibited.

Modular Team Composition

Team structure reflects the modular architecture of the OBINexus system itself, with clearly defined interfaces between contributors that enable seamless substitution when availability changes. Each team module consists of a primary contributor, a shadow contributor who maintains familiarity with the work, and documented knowledge artifacts that enable rapid onboarding.

The Epistemic Architecture Module requires deep mathematical knowledge and philosophical grounding. The primary contributor leads theoretical development while the shadow maintains implementation readiness. Knowledge artifacts include LaTeX-formatted proofs, commented reference implementations, and recorded explanation sessions.

The Hardware Integration Module demands embedded systems expertise and mechanical engineering skills. The primary contributor manages DIRAM implementation while the shadow focuses on sensor integration. Knowledge artifacts include CAD files with assembly instructions, firmware repositories with deployment guides, and video demonstrations of key procedures.

The Stakeholder Engagement Module needs communication skills and domain expertise. The primary contributor leads external relationships while the shadow maintains internal documentation. Knowledge artifacts include email templates, relationship maps, and conversation summaries that preserve institutional memory.

Flex Resource Allocation

Resource allocation operates on a commitment-based model that acknowledges the reality of pre-grant constraints. Contributors commit to specific deliverables within sprint boundaries rather than hourly allocations, enabling flexible scheduling that accommodates other obligations.

Core team members commit to 20-hour weekly minimums during Development Phase sprints, with specific deliverables defined at sprint planning. These commitments are tracked through a public dashboard that shows both individual progress and team velocity, creating accountability without micromanagement.

Specialist contributors operate on task-based engagements, committing to specific deliverables without ongoing time requirements. A cryptography expert might commit to reviewing the DIRAM Merkle tree implementation within a one-week window. A mechanical engineer might commit to validating actuator specifications within a three-day period.

Surge capacity is maintained through a pre-qualified pool of contractors who can be activated within 48 hours. These contractors receive project briefings during Pre-Gate phase and maintain familiarity through weekly technical summaries. When activated, they can productive contribute within one day rather than requiring extensive onboarding.

Knowledge Preservation Protocols

Every significant technical decision, architectural choice, and implementation detail is documented in a structured knowledge base that enables continuity despite team changes. This documentation goes beyond traditional code comments to capture the reasoning behind decisions and the alternatives considered.

Architectural Decision Records (ADRs) document each major technical choice using a standardized template. The template captures the context that necessitated the decision, the options evaluated with their trade-offs, the rationale for the selected approach, and the implications for future development. These ADRs are version-controlled alongside code and reviewed during sprint retrospectives.

Implementation Guides provide step-by-step instructions for reproducing key development activities. Each guide includes prerequisite knowledge, required tools, detailed procedures, validation criteria, and troubleshooting steps. Guides are validated by having team shadows successfully execute procedures independently.

Failure Post-Mortems document what went wrong, why it happened, and how similar failures can be prevented. These documents are blame-free, focusing on systemic improvements rather than individual

mistakes. Post-mortems are shared across the team and incorporated into updated procedures and checklists.

Burnout Detection and Mitigation

Burnout detection operates through both quantitative metrics and qualitative assessments, recognizing that pre-grant pressure can manifest in subtle ways before causing project failure. Early detection enables proactive intervention that preserves both team health and project momentum.

Quantitative indicators are monitored continuously through development analytics. Declining code commit quality (increased bug rates, decreased test coverage) signals cognitive fatigue. Lengthening response times to team communications indicates engagement reduction. Increasing sprint deferral rates suggests unrealistic planning or reduced capacity.

Qualitative assessments occur during weekly one-on-ones between team members and the project lead. These conversations explore energy levels, external stressors, and motivation factors without judgment. Team members are encouraged to self-report burnout risk factors, with immediate support provided including workload reduction, deadline adjustment, or temporary leave.

Mitigation strategies are proportional to burnout severity. Early-stage burnout triggers include mandatory time off, workload redistribution, and scope reduction. Advanced burnout may require bringing in shadow contributors as primary leads, activating surge contractors, or implementing a full team rotation. The project budget includes a 15% allocation specifically for burnout mitigation activities.

6. Metrics Scaling Model

Quality Metrics (Q-Metrics) Framework

Quality metrics establish minimum acceptable thresholds for system performance, ensuring that the OBINexus platform meets both technical requirements and safety standards necessary for defense deployment. These metrics are continuously monitored during development and validated at each gate transition.

Epistemic Coherence Score (ECS) measures the logical consistency of system decisions against the established knowledge manifold. The metric evaluates 1000 random state transitions per test run, checking that each transition maintains mathematical coherence with the system's epistemic boundaries. Target threshold: ≥98% coherent transitions, with no single transition violating core epistemic constraints.

Decision Audit Integrity (DAI) validates the completeness and immutability of the DIRAM audit trail. Every system decision must be traceable through the Merkle tree with cryptographic verification completed in under 10ms. Target threshold: 100% decision capture with zero audit trail corruptions across 10,000 decision cycles.

Threat Classification Precision (TCP) assesses the accuracy of threat gradient resolution across diverse scenarios. The metric uses a weighted F1 score that penalizes false negatives (missed threats) more

heavily than false positives (overcaution). Target threshold: ≥0.95 weighted F1 score on a test set of 5,000 scenarios including edge cases.

Rollback Recovery Time (RRT) measures the system's ability to recover from detected anomalies or failed decisions. The metric captures both detection latency and state restoration time. Target threshold: <50ms total recovery time from anomaly detection to stable state restoration.

Stakeholder Satisfaction Index (SSI) quantifies external validation through structured feedback collection. After each demonstration, stakeholders rate system performance across five dimensions: capability, reliability, usability, safety, and innovation. Target threshold: ≥4.2/5.0 average rating with no dimension below 3.8/5.0.

Quantity Metrics (QTY-Metrics) Framework

Quantity metrics establish minimum output requirements that demonstrate system completeness and project readiness for grant submission. These metrics ensure that claims of capability are supported by tangible deliverables.

Test Scenario Coverage requires execution across a comprehensive set of operational contexts. The scenario library must include: 20 basic object recognition tasks, 30 threat/non-threat discrimination challenges, 25 prosthetic device variations, 15 environmental condition sets (lighting, weather, occlusion), and 10 adversarial test cases. Total requirement: 100+ documented test scenarios with recorded outcomes.

Stakeholder Engagement Depth measures meaningful interaction with potential system users and evaluators. Requirements include: 5 signed letters of support from organizations spanning defense, humanitarian, and academic sectors; 15 documented demonstration sessions with feedback incorporation; 3 co-development workshops with end users contributing requirements; 50+ total contact hours with external stakeholders.

Knowledge Artifact Production ensures that system understanding is captured in reusable forms. Deliverables include: 5 legacy capsules implementing core functionality with standalone documentation; 10 architectural decision records documenting major technical choices; 3 peer-reviewed technical papers or extended abstracts; 20 implementation guides enabling knowledge transfer; 1 comprehensive system manual of 100+ pages.

Team Resilience Indicators demonstrate sustainable development practices. Metrics include: 3 successful shadow-to-primary contributor transitions; 5 knowledge transfer sessions recorded and validated; 2 surge contractor activations with <48 hour productivity achievement; 8 sprint retrospectives with documented improvements; 0 team member departures due to preventable burnout.

Metric Validation Protocols

Each metric undergoes rigorous validation to ensure measurement accuracy and relevance to system objectives. Validation occurs at three levels: technical accuracy, operational relevance, and stakeholder

acceptance.

Technical validation confirms that metrics accurately measure intended properties. Independent reviewers verify metric calculation algorithms, test data integrity, and statistical significance. Any metric showing >5% measurement variance undergoes recalibration before gate review.

Operational validation ensures that metrics predict real-world system performance. Correlation analysis between development metrics and demonstration outcomes identifies which measurements provide genuine insight versus vanity statistics. Metrics showing <0.7 correlation with stakeholder feedback are revised or eliminated.

Stakeholder validation confirms that metrics address actual user concerns. External advisory board members review metric definitions and thresholds, suggesting modifications based on deployment experience. Metrics that stakeholders consider irrelevant or insufficient are supplemented with additional measurements.

Scaling Trajectories

Metrics are designed to scale from pre-grant prototype to production deployment, with clear growth trajectories defined for each measurement category.

Pre-grant metrics focus on fundamental capability demonstration. The system must prove core concepts work reliably in controlled conditions with friendly stakeholders. Thresholds are set to demonstrate viability rather than optimization.

Grant-funded development metrics expand scope and rigor. Test scenarios grow from 100 to 1,000+ cases. Stakeholder engagement broadens from 5 to 50+ organizations. Performance thresholds tighten to approach production requirements.

Production deployment metrics emphasize reliability and scale. The system must maintain performance across millions of decisions, thousands of edge cases, and hundreds of deployment sites. Metrics shift from absolute thresholds to statistical process control with defined variance limits.

Continuous Improvement Integration

Metrics drive continuous improvement through automated analysis and human review cycles. Each sprint generates a metrics dashboard highlighting trends, anomalies, and improvement opportunities.

Automated analysis identifies metric degradation before it impacts gate transitions. Machine learning models trained on historical project data predict future metric trajectories, enabling proactive intervention. Any metric showing negative trajectory for two consecutive sprints triggers mandatory review.

Human review sessions translate metric insights into actionable improvements. During sprint retrospectives, teams examine metric trends to identify systemic issues versus random variation. Improvement actions are tracked through subsequent sprints to validate effectiveness.

Gate review boards use metric history to make informed transition decisions. Rather than examining only current values, boards analyze metric trajectories, variance patterns, and improvement rates. A project showing consistent metric improvement may pass gates despite marginal current performance, while a project with declining metrics may be held despite meeting thresholds.

7. Conclusion and Grant Readiness Certification

The OBINexus gated development architecture represents a fundamental reimagining of how complex defense innovations progress from concept to deployment-ready systems. By embedding epistemic principles throughout both technical architecture and project management, we demonstrate that philosophical rigor and practical execution are not opposing forces but complementary aspects of responsible innovation.

This documentation certifies that the OBINexus project has established the frameworks, protocols, and measurement systems necessary for successful DASA grant execution. The gated architecture ensures that progress is not merely claimed but demonstrated through objective metrics and external validation. The burn-resistant practices ensure that the team reaching grant submission remains capable of executing post-grant development. The modular structure ensures that knowledge and capabilities persist beyond individual contributors.

When DASA evaluators review this submission, they will find not just promising technology but a mature execution framework ready for the challenges of scaling innovation. The gates we have defined are not bureaucratic obstacles but quality assurances that investment in OBINexus will yield tangible results. The metrics we track are not arbitrary numbers but meaningful predictors of real-world impact.

Through this systematic approach to development, documentation, and validation, OBINexus stands ready to transform the theoretical promise of epistemic AI into practical tools for defense and humanitarian applications. The wisdom we seek to embed in autonomous systems is already demonstrated in the wisdom of our development approach.