ENVELO REQUIREMENTS

Enforcer for Non-Violable Execution & Limit Oversight — Sentinel Authority

v1.1

1. Overview

1.1 Definition

ENVELO (Enforcer for Non-Violable Execution & Limit Oversight) defines the runtime enforcement requirements for ODDC conformance. ENVELO specifies architectural and behavioral requirements for the enforcement layer that constrains autonomous system actions to declared operational boundaries.

ENVELO is a method designation describing non-bypassable enforcement requirements. It is not a product, platform, software package, or implementation specification. Multiple implementation approaches may satisfy ENVELO requirements.

1.2 Relationship to ODDC

ENVELO-compliant enforcement is mandatory for ODDC conformance. An Operational Design Domain specification without runtime enforcement is unverifiable. ENVELO provides the enforcement layer that makes ODD boundaries real and auditable.

1.3 Implementation Responsibility

Sentinel Authority does not implement, operate, or monitor runtime systems. Operators and their licensed implementers are solely responsible for ENVELO-compliant implementation. Sentinel Authority verifies conformance through CAT-72 evidentiary demonstration; it does not certify implementation correctness, validate code, audit systems, or warrant behavior.

2. Core Requirements

The following requirements are mandatory for ENVELO compliance. All requirements must be satisfied; partial compliance does not meet ENVELO specifications.

2.1 Non-Bypassable Interlock (REQUIRED)

The enforcement mechanism must validate all autonomous actions before execution. This is the foundational ENVELO requirement.

2.1.1 Architectural Isolation

The interlock must be architecturally isolated from the decision logic it constrains:

- Separate execution context (process, container, or hardware boundary)
- No shared memory or direct variable access with control model
- Communication only through defined interface (action proposals and approval/denial)
- Independent failure modes (interlock failure does not compromise control model, and vice versa)

2.1.2 No Bypass Pathways

No pathway may exist for unvalidated action execution:

- All actuator commands must pass through enforcement layer

- No "debug modes" or "override flags" that disable enforcement
- No timing windows during which actions execute without validation
- No privilege escalation mechanisms accessible to control model

2.1.3 Verification Evidence

CAT-72 evidence must demonstrate non-bypassability through architectural documentation and runtime telemetry showing all actions validated prior to execution.

2.2 Fail-Closed Behavior (REQUIRED)

Out-of-domain actions must be blocked or overridden. The system must transition to a defined safe state upon enforcement activation.

2.2.1 Safe State Definition

Safe states must be formally specified in ODD documentation:

- Explicit definition of safe state for each operational mode
- Transition procedures from any operating state to safe state
- Maximum transition time from enforcement activation to safe state achievement
- Verification criteria for safe state confirmation

2.2.2 Enforcement Actions

Upon detection of out-of-domain action proposals, the enforcement layer must:

- Block the proposed action (prevent execution)
- Log the enforcement event with full context
- Initiate safe state transition if continued operation is unsafe
- Alert operators through defined notification channels

2.2.3 Response Time

Enforcement response time must be specified and verified:

- Maximum latency from boundary detection to action blocking
- Maximum latency from enforcement activation to safe state initiation
- Response times must be validated during CAT-72 fail-closed verification

2.3 Runtime Comparison (REQUIRED)

The enforcement layer must continuously evaluate system state and proposed actions against declared ODD constraints.

2.3.1 Continuous Evaluation

State evaluation must be continuous during autonomous operation:

- Minimum evaluation frequency: 1 Hz for all ODD boundary conditions

- Higher frequency recommended for fast-dynamics systems (10 Hz or greater)
- No evaluation gaps during operational transitions

2.3.2 Boundary Evaluation

For each proposed action, the enforcement layer must evaluate:

- Predicted state after action execution
- Proximity to all declared ODD boundaries
- Trajectory analysis for multi-step action sequences (if applicable)
- Margin verification against declared tolerances

2.3.3 Evaluation Independence

The enforcement layer evaluation must be independent of control model predictions. The enforcement layer must maintain its own state estimation or receive state information from sensors independent of control model inputs.

2.4 Tamper-Evident Logging (REQUIRED)

All boundary interactions and enforcement events must be recorded with cryptographic integrity for audit.

2.4.1 Mandatory Log Contents

Each log entry must include:

- Timestamp (UTC ISO 8601, millisecond precision)
- System state at time of evaluation
- Proposed action details
- Evaluation result (approved/denied/modified)
- Action taken (executed/blocked/modified)
- Boundary proximity metrics

2.4.2 Integrity Requirements

Log integrity must be maintained through:

- Hash-chain linking (SHA-256 minimum) - each entry includes hash of previous entry
- Cryptographic sealing at defined intervals (minimum hourly)
- Tamper detection capability (hash verification on read)
- Append-only storage (no modification or deletion of historical entries)

2.4.3 Retention

Logs must be retained for conformance term plus minimum 3 years. Logs must be available for Sentinel Authority inspection upon request.

2.5 Architectural Consonance (REQUIRED)

The safe operational envelope constraining autonomous actions must be derived from the same physics-based

simulation used to generate synthetic training data for the control model.

2.5.1 Rationale

Architectural consonance ensures that the learned control policy operates within a state space inherently bounded by the same physics model that defines safety constraints. When training data and safety envelope derive from the same simulation:

- The control model learns behavior consistent with the physics that define safety boundaries
- Safety margins are meaningful (same physical units and relationships)
- Model behavior near boundaries is predictable (trained on consistent physics)

2.5.2 Non-Compliant Architectures

The following architectures do not satisfy ENVELO requirements:

- Training data from one simulation, safety envelope from independent simulation
- Training data from physical testing, safety envelope from simulation
- Safety envelope derived from manufacturer specifications without simulation validation
- Control model trained on data from multiple inconsistent simulations

2.5.3 Evidence Requirements

See CAT-72 Procedure Section 6 for detailed evidentiary requirements demonstrating architectural consonance.

3. Enforcement Architecture

3.1 Three-Stage Pattern

ENVELO requires a non-bypassable control boundary between decision logic and action execution. The reference architecture follows a three-stage pattern:

PREDICT

Control model generates proposed action based on system state and objectives. The control model operates within its learned policy to determine optimal actions. Proposed actions are submitted to the enforcement layer for validation.

EVALUATE

Enforcement layer compares proposed action against ODD envelope boundaries. Evaluation includes current state assessment, predicted state after action, and boundary proximity analysis. Evaluation is independent of control model predictions.

EXECUTE

Action proceeds only if within envelope; otherwise fail-closed behavior activates. Approved actions are forwarded to actuators. Denied actions are blocked and logged. Enforcement activation triggers safe state transition if required.

3.2 Architectural Constraints

The enforcement layer must satisfy the following architectural constraints:

- The control model cannot modify, disable, or bypass the enforcement layer
- The enforcement layer has exclusive control over actuator interfaces
- Enforcement layer failure results in safe state transition (fail-safe)
- No shared failure modes between control model and enforcement layer

4. Implementation Guidance

4.1 Hardware vs. Software Enforcement

ENVELO requirements may be satisfied through hardware enforcement, software enforcement, or hybrid approaches:

Hardware enforcement (e.g., dedicated safety PLC, hardware interlocks) provides stronger isolation guarantees but less flexibility. Software enforcement (e.g., separate process with privilege isolation) provides flexibility but requires careful implementation to ensure non-bypassability.

Sentinel Authority does not mandate implementation approach. CAT-72 evaluates enforcement behavior regardless of implementation method.

4.2 Redundancy Considerations

ENVELO does not mandate redundancy requirements. Operators should consider redundancy based on:

- Consequence severity of enforcement layer failure
- Regulatory requirements applicable to operational domain
- Underwriting requirements
- Industry best practices

4.3 Testing Requirements

Prior to CAT-72, operators should verify:

- Enforcement layer correctly blocks all out-of-domain actions
- Safe state transitions complete within specified time
- Logging captures all required data elements
- Hash-chain integrity is maintained under load
- No bypass pathways exist under any configuration

5. Liability and Responsibility

ODDC conformance does not transfer liability from the operator to Sentinel Authority.

Operators remain solely responsible for:

- ENVELO-compliant implementation correctness
- System behavior within and outside declared ODD
- Operational compliance with all applicable regulations

- Incident response and remediation
- All consequences of system operation

Sentinel Authority is an independent conformance determination body. It is not a regulator, insurer, implementation partner, or guarantor of system behavior.