MYTHOS SYSTEMS

PERMANENT LIBRARY PUBLICATION

ENGINEERING STANDARDS LIBRARY / NETWORK ENGINEERING

Network Source of Truth and Drift Reconciliation Standard

Defines authoritative state, discovery, intent, reconciliation, exceptions, and audit.

PERMANENT DOCUMENT ID
MYTHOS-NET-0005

PUBLICATION
Candidate Standard

ASSURANCE SURFACE
9 Atomic Criteria / 6 Families

PERMANENT PUBLICATION IDENTITY

Network Source of Truth and Drift Reconciliation Standard

Defines authoritative state, discovery, intent, reconciliation, exceptions, and audit.

DOCUMENT ID
MYTHOS-NET-0005

VERSION
1.0.0-candidate

STATUS
Candidate Standard

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-23

SCHEDULED REVIEW
2027-01-23

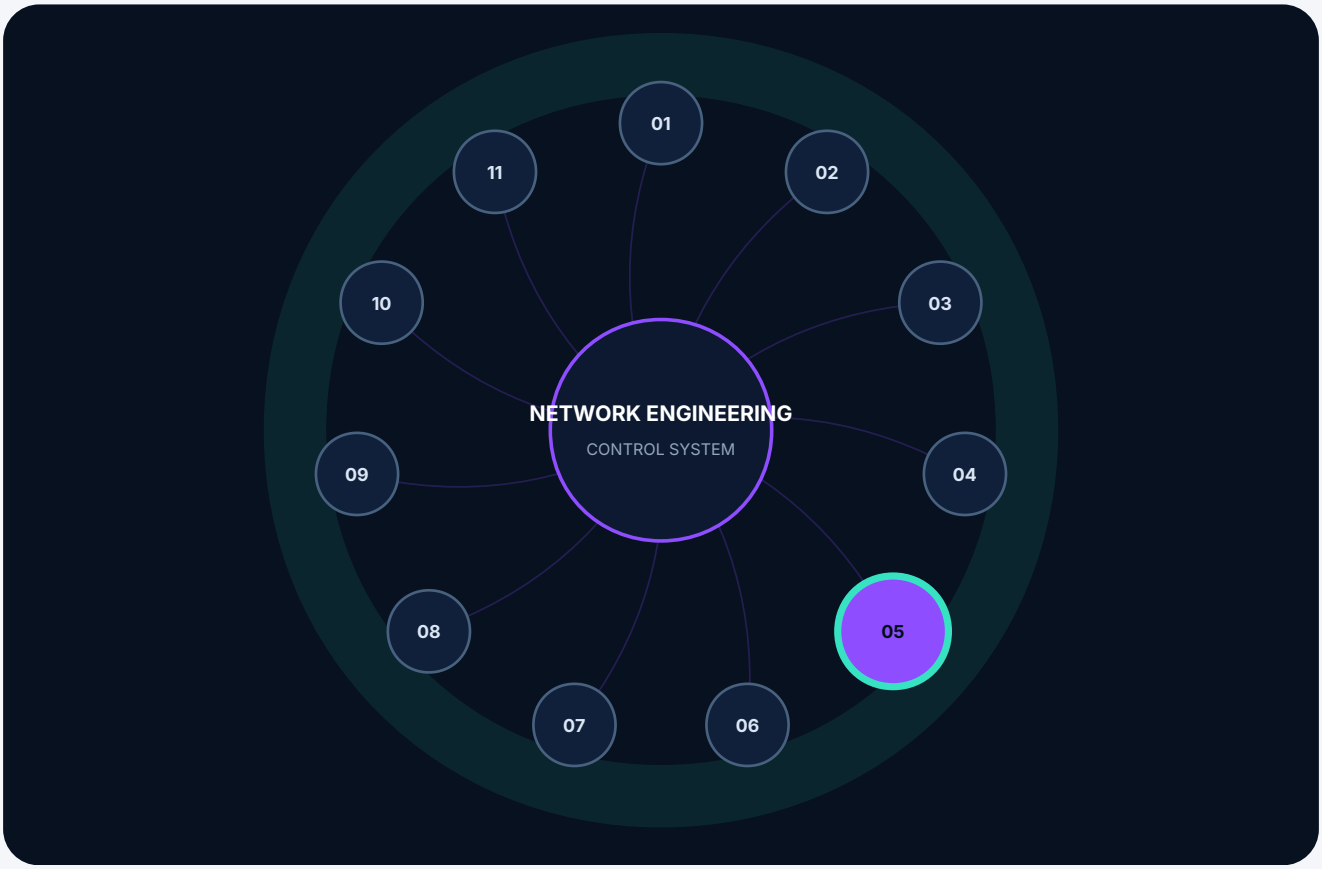
CLASSIFICATION
Public

9	6	4	1
ATOMIC CRITERIA	CAPABILITY FAMILIES	ASSESSMENT PROFILES	PERMANENT IDENTITY

Publication doctrine. This standard establishes durable requirements for its defined scope. Interpretation must preserve the permanent document identity, recorded evidence, and formal revision history.

DOMAIN CONTROL SYSTEM

MYTHOS-NET-0005 inside the network engineering standard constellation



Connected assurance

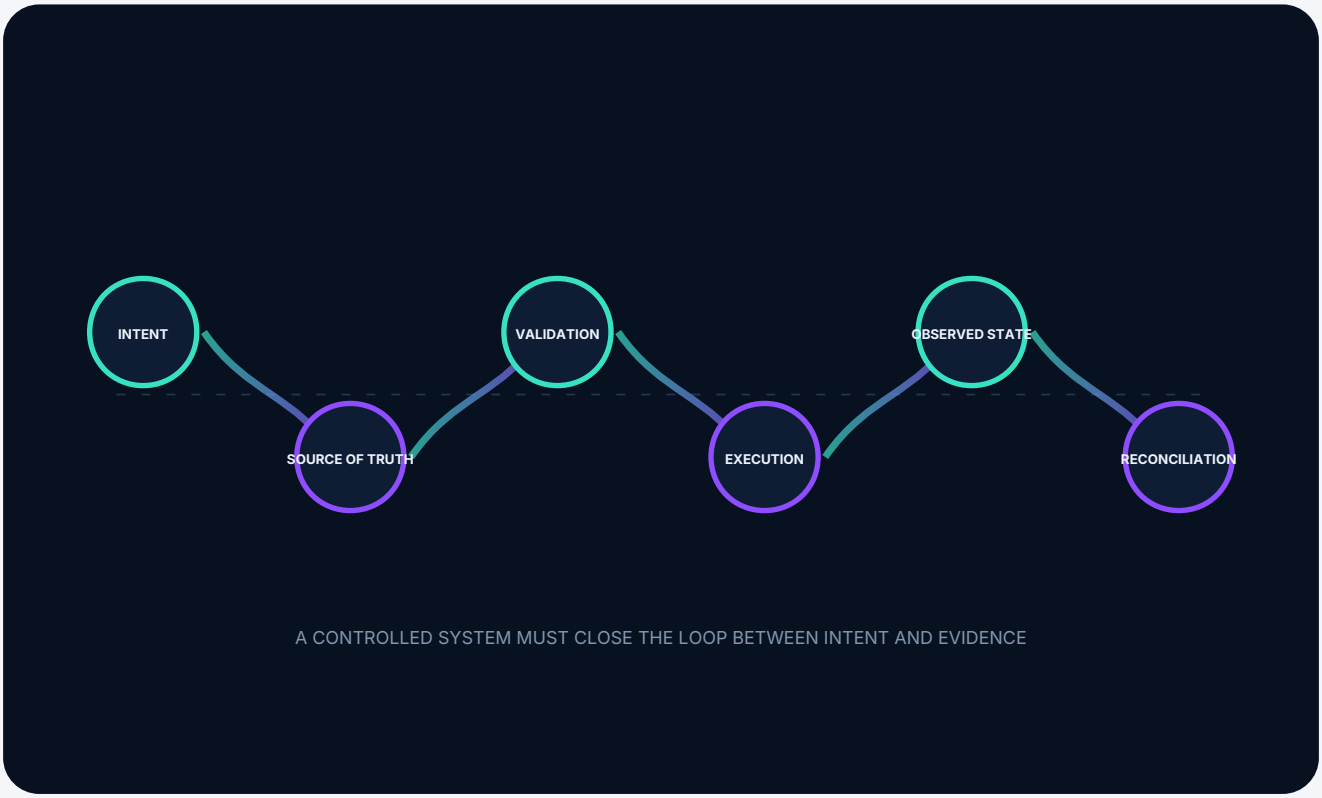
Architecture, policy, implementation, operations, and evidence are evaluated as one controlled system.

Specialization rule

Companion standards may add precision, but they cannot silently weaken this publication's requirements.

CLOSED-LOOP ARCHITECTURE

Intent becomes trustworthy only when observed state returns as evidence



State separation

Desired, approved, deployed, observed, historical, and simulated states must remain distinguishable.

Recoverable change

Material change requires validation, authorization, monitoring, containment, and a credible recovery path.

EVALUATION TOPOLOGY

The capability families and atomic criteria of MYTHOS-NET-0005

Capability families

Families expose the major engineering surfaces and preserve the source weighting model.

Atomic criteria

Each criterion carries a question, maturity spectrum, evidence expectations, assessment method, and operational signals.

NAVIGATION

Contents

0. Executive Mandate

Part I. Scope and Applicability

Part II. Definitions and Taxonomy

Part III. Evaluation Methodology

Evaluation system

4.1 - SoT Authority and Schema Integrity

- 4.1.1 - Single Source of Truth
- 4.1.2 - Data Model Completeness

4.2 - State Ingestion and Fidelity

- 4.2.1 - Automated Discovery
- 4.2.2 - Fidelity Validation

4.3 - Drift Detection and Reconciliation

- 4.3.1 - Drift Detection Scope
- 4.3.2 - Reconciliation Automation

4.4 - GitOps and Version Control Integration

- 4.4.1 - State-as-Code

4.5 - API Security and Data Governance

- 4.5.1 - SoT Access Control

4.6 - Operational Lifecycle

- 4.6.1 - State Hygiene

Part V. The Mythos State Management Suite (MSTMS)

ENGINEERING DOCTRINE

0. Executive Mandate

The management of network state has failed.

Organizations operate their networks using a fragmented patchwork of spreadsheets, tribal knowledge, and the device's running-configuration as the de facto source of truth. This results in phantom IP allocations, undocumented VLANs, and configuration drift that causes outages and security breaches. The network's actual state is a mystery until something breaks.

This standard exists because:

1. **A Spreadsheet is Not a Database:** Tracking critical infrastructure state in Excel is a liability. It cannot be queried programmatically, it cannot enforce schema, and it cannot be reconciled against reality.
2. **The Device is Not the Source of Truth:** The running-configuration on a switch is the *observed state*, not the *intent*. Treating the device as the SoT means every out-of-band CLI change becomes the new architecture, cementing drift into the fabric.
3. **Drift is a Bug, Not a Feature:** Configuration drift - any divergence between the intended state (SoT/Git) and the observed state (device) - is an operational failure. It MUST be detected, alerted, and automatically remediated.
4. **If It's Not in the SoT, It Doesn't Exist:** Any device, IP, VLAN, or circuit not recorded in the authoritative SoT is unauthorized. The SoT MUST be the absolute prerequisite for any automation, security policy, or simulation.

This document establishes the **Mythos Source-of-Truth & Drift Standard (MSTDS)**, a comprehensive, evidence-based methodology for evaluating the authority, completeness, ingestion fidelity, and reconciliation capabilities of network state management platforms.

ENGINEERING DOCTRINE

Part I. Scope and Applicability

1.1 In Scope

This standard covers the evaluation of:

- Source-of-Truth (SoT) and Source-of-Record (SoR) platforms (NetBox, Nautobot)
- IP Address Management (IPAM) and Data Center Infrastructure Management (DCIM)
- Configuration drift detection and automated remediation
- GitOps reconciliation for network state (desired-state vs. observed-state)
- Real-time state ingestion (gNMI, streaming telemetry, eBPF)
- Data integrity, schema enforcement, and validation rules
- SoT API security and audit logging

1.2 Out of Scope

- Network automation execution engines (covered in MYTHOS-NET-0002)
- Digital twin simulation (covered in MYTHOS-NET-0003)
- General database architecture (covered in MYTHOS-DBS)

1.3 Audience

- Network architects and automation engineers designing state management
- Platform engineers building Network-as-Code pipelines
- Security teams evaluating IPAM and segmentation alignment
- AI governance, risk, and compliance teams
- Standards bodies seeking a reference network state methodology

---

ENGINEERING DOCTRINE

Part II. Definitions and Taxonomy

2.1 State Management Dimensions

Dimension	Definition
Source-of-Truth (SoT)	The authoritative, programmatically accessible data store that defines the intended state of the network (IPs, VLANs, topologies, contracts).
Observed State	The actual running configuration and operational status of network devices, as reported by the control plane.
Intended State	The state declared in the SoT and version-controlled Git repositories.
Drift	Any divergence between the Intended State and the Observed State.
Reconciliation	The automated process of detecting drift and realigning the observed state with the intended state.

2.2 The State Management Doctrine

> A spreadsheet is a liability. A device is an observer, not an authority. Drift is a bug. If it is not in the SoT, it does not exist. The SoT writes the network; the network never writes the SoT.

ENGINEERING DOCTRINE

Part III. Evaluation Methodology

3.1 Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; spreadsheets, no SoT, manual state
1	Basic IPAM but incomplete, no reconciliation
2	Functional SoT but manual data entry, no drift remediation
3	Solid production performance; API-driven SoT, automated drift detection
4	Strong performance; continuous reconciliation, GitOps integration
5	Best-in-class; sets the standard for autonomous, cryptographically-verified network state

Weighting

Category	Weight	Rationale
SoT Authority & Schema Integrity	25%	An incomplete or unenforced SoT is useless
State Ingestion & Fidelity	20%	The SoT must reflect reality to be trusted
Drift Detection & Reconciliation	20%	Unremediated drift causes outages
GitOps & Version Control Integration	15%	State must be versioned and peer-reviewed
API Security & Data Governance	10%	The SoT is a critical attack surface
Operational Lifecycle	10%	State management must be continuous

Total: 100%

A system MUST score at least 3.0 in SoT Authority and Drift Detection to be considered for production use.

EVALUATION AND ASSURANCE SYSTEM

Capability claims are bounded by evidence, context, and reproducible assessment.

9 atomic criteria across 6 capability families define the evaluation surface of this standard.

4.1 / CAPABILITY FAMILY

SoT Authority and Schema Integrity

SOURCE WEIGHT 25%

4.1.1
Single Source of Truth

4.1.2
Data Model Completeness

MYTHOS-NET-0005-EVAL-4.1.1

Single Source of Truth

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
SoT Authority and Schema Integrity	**4.1.1**	**25%**	**Evidence-led**

Is the SoT the designated authority for network intent and allocation?

0
No centralized SoT; state fragmented across spreadsheets and devices

1
Centralized IPAM but not used by all teams

2
SoT exists but out-of-band changes are common

3
SoT is the required pre-requisite for all automation; no manual IP allocation

4
SoT enforces schema (no orphaned IPs, no invalid VLANs); automation refuses to run without SoT context

5
Leading authority: formally verified schema, cryptographic attestation of state, zero out-of-band allocations, and SoT-as-code

ENGINEERING INTERPRETATION

This criterion evaluates whether single source of truth is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- source-of-truth objects, Git history, observed-state snapshots, and reconciliation evidence
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. introduce controlled drift and verify detection, adjudication, and restoration
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- multiple authorities, silent manual changes, stale inventory, and destructive auto-remediation
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- drift detection time
- reconciliation success
- unowned object count
- coverage of in-scope assets or flows

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

MYTHOS-NET-0005-EVAL-4.1.2

Data Model Completeness

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
SoT Authority and Schema Integrity	4.1.2	25%	Evidence-led

Does the SoT capture the full intent of the network (physical, logical, virtual)?

0
Only tracks IP prefixes

1
Tracks IPs and basic device inventory

2
Tracks cables, interfaces, and VLANs

3
Tracks logical topologies (VRFs, overlays, routing policies)

4
Tracks security contracts (service meshes, ACL intent, microsegmentation tags)

5
Leading completeness: full infrastructure graph (physical, logical, security, AI workloads), formally verified referential integrity, and zero unmodeled state

ENGINEERING INTERPRETATION

This criterion evaluates whether data model completeness is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- approved architecture or policy record
- current configuration or platform export
- change and exception records
- monitoring, test, or assessment evidence

ASSESSMENT PROCEDURE

1. Examine authoritative design, policy, configuration, and lifecycle records.
2. Interview the accountable owner and operators responsible for the control.
3. Test a representative positive, negative, failure, and recovery scenario.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- The control exists only in diagrams or policy text.
- Observed state differs from approved intent without a governed exception.
- Evidence cannot establish scope, time, owner, or operating effectiveness.

OPERATIONAL MEASURES

- coverage of in-scope assets or flows
- age and closure rate of unresolved findings
- percentage of changes passing pre-deployment validation
- time to detect and contain control failure

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.2 / CAPABILITY FAMILY

State Ingestion and Fidelity

SOURCE WEIGHT 20%

4.2.1
Automated Discovery

4.2.2
Fidelity Validation

MYTHOS-NET-0005-EVAL-4.2.1

Automated Discovery

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
State Ingestion and Fidelity	**4.2.1**	**20%**	**Evidence-led**

How is the SoT populated and maintained?

0
Manual data entry

1
Automated network scanning (SNMP/SSH) on schedule

2
API-driven ingestion from automation tools

3
Streaming ingestion via gNMI/streaming telemetry

4
Real-time eBPF-driven state observation for fabric validation

5
Leading ingestion: continuous, real-time, multi-source correlation with cryptographic attestation of observed state

ENGINEERING INTERPRETATION

This criterion evaluates whether automated discovery is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- approved architecture or policy record
- current configuration or platform export
- change and exception records
- monitoring, test, or assessment evidence

ASSESSMENT PROCEDURE

1. Examine authoritative design, policy, configuration, and lifecycle records.
2. Interview the accountable owner and operators responsible for the control.
3. Test a representative positive, negative, failure, and recovery scenario.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- The control exists only in diagrams or policy text.
- Observed state differs from approved intent without a governed exception.
- Evidence cannot establish scope, time, owner, or operating effectiveness.

OPERATIONAL MEASURES

- coverage of in-scope assets or flows
- age and closure rate of unresolved findings
- percentage of changes passing pre-deployment validation
- time to detect and contain control failure

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

MYTHOS-NET-0005-EVAL-4.2.2

Fidelity Validation

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
State Ingestion and Fidelity	4.2.2	20%	Evidence-led

Does the system verify that the SoT matches the actual network?

0
No validation; SoT assumed correct

1
Manual spot checks

2
Automated nightly diff reports

3
Continuous parity validation between SoT and digital twin (Batfish)

4
Automated alerts on drift with suggested remediation

5
Leading fidelity: continuous mathematical proof of SoT-to-device parity, automatic flagging of undocumented state

ENGINEERING INTERPRETATION

This criterion evaluates whether fidelity validation is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- approved architecture or policy record
- current configuration or platform export
- change and exception records
- monitoring, test, or assessment evidence

ASSESSMENT PROCEDURE

1. Examine authoritative design, policy, configuration, and lifecycle records.
2. Interview the accountable owner and operators responsible for the control.
3. Test a representative positive, negative, failure, and recovery scenario.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- The control exists only in diagrams or policy text.
- Observed state differs from approved intent without a governed exception.
- Evidence cannot establish scope, time, owner, or operating effectiveness.

OPERATIONAL MEASURES

- coverage of in-scope assets or flows
- age and closure rate of unresolved findings
- percentage of changes passing pre-deployment validation
- time to detect and contain control failure

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.3 / CAPABILITY FAMILY

Drift Detection and Reconciliation

SOURCE WEIGHT 20%

4.3.1
Drift Detection Scope

4.3.2
Reconciliation Automation

MYTHOS-NET-0005-EVAL-4.3.1

Drift Detection Scope

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Drift Detection and Reconciliation	4.3.1	20%	Evidence-led

What types of drift does the system detect?

0
No drift detection

1
Detects configuration text changes only

2
Detects operational state changes (BGP neighbors, interface status)

3
Detects semantic drift (e.g., a VLAN exists but is not attached to the correct interface/subnet)

4
Detects intent drift (e.g., a device config complies with syntax but violates security policy)

5
Leading detection: formally verified semantic drift, zero-day policy violations, and cryptographic sealing of intended state

ENGINEERING INTERPRETATION

This criterion evaluates whether drift detection scope is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- telemetry coverage, detection source, test fixtures, version history, alerts, and case outcomes
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. replay benign and malicious fixtures; measure fidelity and operational response
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- untested rules, missing telemetry, alert-only metrics, and undocumented suppression
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- true-positive rate
- false-positive burden
- coverage by technique
- mean time to contain
- coverage of in-scope assets or flows

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

MYTHOS-NET-0005-EVAL-4.3.2

Reconciliation Automation

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Drift Detection and Reconciliation	4.3.2	20%	Evidence-led

How is drift remediated?

0
Manual remediation

1
Automated alerts, manual push

2
Automated remediation for specific, low-risk changes

3
Automated remediation for all configuration drift (desired-state push)

4
GitOps reconciliation: drift triggers a PR to revert/overwrite, CI/CD pushes the fix

5
Leading reconciliation: continuous self-healing, formally verified convergence, and zero human intervention for standard drift

ENGINEERING INTERPRETATION

This criterion evaluates whether reconciliation automation is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- telemetry coverage, detection source, test fixtures, version history, alerts, and case outcomes
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. replay benign and malicious fixtures; measure fidelity and operational response
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- untested rules, missing telemetry, alert-only metrics, and undocumented suppression
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- true-positive rate
- false-positive burden
- coverage by technique
- mean time to contain
- coverage of in-scope assets or flows

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.4 / CAPABILITY FAMILY

GitOps and Version Control Integration

SOURCE WEIGHT 15%

4.4.1
State-as-Code

MYTHOS-NET-0005-EVAL-4.4.1

State-as-Code

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
GitOps and Version Control Integration	4.4.1	15%	Evidence-led

Is the intended network state version-controlled in Git?

0
No version control

1
Periodic backups to Git

2
Git repository exists, but SoT (NetBox) is the primary interface

3
Git is the Single Source of Truth; SoT is a derived cache (GitOps Pull)

4
Git PRs required for all changes; PR triggers simulation + SoT validation

5
Leading State-as-Code: cryptographic signing of commits, PQC-signed merge gates, immutable history, and formally verified Git-to-SoT-to-Device pipeline

ENGINEERING INTERPRETATION

This criterion evaluates whether state-as-code is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- approved architecture or policy record
- current configuration or platform export
- change and exception records
- monitoring, test, or assessment evidence

ASSESSMENT PROCEDURE

1. Examine authoritative design, policy, configuration, and lifecycle records.
2. Interview the accountable owner and operators responsible for the control.
3. Test a representative positive, negative, failure, and recovery scenario.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- The control exists only in diagrams or policy text.
- Observed state differs from approved intent without a governed exception.
- Evidence cannot establish scope, time, owner, or operating effectiveness.

OPERATIONAL MEASURES

- coverage of in-scope assets or flows
- age and closure rate of unresolved findings
- percentage of changes passing pre-deployment validation
- time to detect and contain control failure

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.5 / CAPABILITY FAMILY

API Security and Data Governance

SOURCE WEIGHT 10%

4.5.1
SoT Access Control

MYTHOS-NET-0005-EVAL-4.5.1

SoT Access Control

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
API Security and Data Governance	4.5.1	10%	Evidence-led

Is the SoT protected against unauthorized modification?

0 Open API, no auth	**1** Basic API keys	**2** OIDC/OAuth2 with RBAC
3 Granular object-level permissions (RBAC/ABAC)	**4** mTLS with SPIFFE identity for all automation clients	**5** Leading governance: PQC-mTLS, SPIFFE identity, cryptographic audit trails, and zero standing human write access

ENGINEERING INTERPRETATION

This criterion evaluates whether sot access control is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- API inventory, authorization policy, token metadata, gateway logs, and abuse controls
- approved architecture or policy record
- current configuration or platform export

ASSESSMENT PROCEDURE

1. test object and function authorization, token misuse, replay, expiration, and rate limits
2. Examine authoritative design, policy, configuration, and lifecycle records.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- bearer tokens without audience limits, hidden APIs, and inconsistent authorization
- The control exists only in diagrams or policy text.

OPERATIONAL MEASURES

- authorized API coverage
- token age
- replay rejection
- abuse-control efficacy
- coverage of in-scope assets or flows

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

4.6 / CAPABILITY FAMILY

Operational Lifecycle

SOURCE WEIGHT 10%

4.6.1
State Hygiene

MYTHOS-NET-0005-EVAL-4.6.1

State Hygiene

FAMILY	SOURCE REFERENCE	WEIGHT CONTEXT	ASSESSMENT POSTURE
Operational Lifecycle	4.6.1	10%	Evidence-led

Does the system enforce hygiene (no stale data, no orphans)?

0
Stale data accumulates indefinitely

1
Manual cleanup

2
Automated alerts for stale objects

3
Automated garbage collection for orphaned objects

4
Strict referential integrity prevents orphaned objects from being created

5
Leading hygiene: formally verified referential integrity, automated lifecycle management (provisioning → active → decommissioned), and zero stale state

ENGINEERING INTERPRETATION

This criterion evaluates whether state hygiene is governed as an operating capability rather than a one-time configuration. Assessment must distinguish intended design, approved implementation, observed behavior, historical evidence, and unresolved risk.

REQUIRED EVIDENCE

- approved architecture or policy record
- current configuration or platform export
- change and exception records
- monitoring, test, or assessment evidence

ASSESSMENT PROCEDURE

1. Examine authoritative design, policy, configuration, and lifecycle records.
2. Interview the accountable owner and operators responsible for the control.
3. Test a representative positive, negative, failure, and recovery scenario.

EXAMINE - INTERVIEW - TEST

COMMON FAILURE PATTERNS

- The control exists only in diagrams or policy text.
- Observed state differs from approved intent without a governed exception.
- Evidence cannot establish scope, time, owner, or operating effectiveness.

OPERATIONAL MEASURES

- coverage of in-scope assets or flows
- age and closure rate of unresolved findings
- percentage of changes passing pre-deployment validation
- time to detect and contain control failure

DECISION BOUNDARY

A maturity score is valid only for the assessed scope and evidence period. Documentation alone does not establish operating effectiveness.

LIFECYCLE AND ASSURANCE

Part V. The Mythos State Management Suite (MSTMS)

Traditional network benchmarks measure protocol uptime. The MSTMS evaluates SoT authority, fidelity, and reconciliation.

5.1 Suite Composition

5.1.1 MSTMS-Authority

- **Shadow IT Injection:** 50+ tests adding devices/IPs directly to the network without SoT entry; verify automation ignores/quarantines them.
- **Schema Validation:** 100+ tests attempting to create invalid objects (overlapping IPs, missing relations); verify SoT rejects them.

5.1.2 MSTMS-Reconciliation

- **Config Drift Injection:** 100+ tests modifying device configs out-of-band; verify system detects and auto-remediates.
- **State Fidelity:** 50+ tests verifying SoT topology matches digital twin topology.

5.2 Execution Protocol

PUBLICATION BOUNDARY

Candidate standard. Permanent identity. Evidence before authority.

Approval requires designated technical, security, legal, accessibility, and governance review with recorded findings and dispositions.