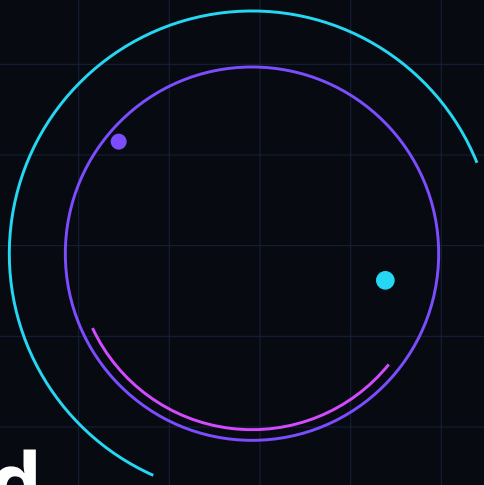
MYTHOS TECHNICAL LIBRARY

STANDARDS & BENCHMARKS / MYTHOS-AI



Applied Automation and Infrastructure-as-Code Benchmark Standard

Defines AI-assisted automation and infrastructure-as-code evaluation with intent, policy, validation, blast radius, rollback, drift, and evidence controls.

MYTHOS-AI-0005

Candidate Mythos Standard / 2.0.0-candidate

Published 2026-07-26 | Review 2026-10-26 | Public

Mythos Systems

Permanent Identity and Edition Record

Permanent document ID	MYTHOS-AI-0005
Canonical title	Applied Automation and Infrastructure-as-Code Benchmark Standard
Publication class	Standards & Benchmarks
Primary domain	MYTHOS-AI - Artificial Intelligence & Agentic Systems
Version	2.0.0-candidate
Status	Candidate Mythos Standard
Publication date	2026-07-26
Scheduled review	2026-10-26
Publisher	Mythos Systems
Owner	Aaron Stovall
Classification	Public
Prior edition	1.0.0-candidate / 2026-07-24
Supersession	This edition supersedes the prior candidate while preserving the permanent identity and historical source.

Identity doctrine

The permanent document ID is immutable. Production sequence labels are not part of the public identity. Atomic standards are authoritative; portfolios, workbooks, and machine-readable exports are generated views.

Normative Language and Applicability

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL indicate the strength of provisions in this Mythos standard. The publication is a Mythos-authored candidate standard and does not claim approval by the cited external authorities.

Applicability rule

Every control is applicable unless the organization documents a specific architectural or lifecycle reason that the subject is absent. N/A decisions MUST name the decision owner, evidence, affected scope, review date, and triggers that invalidate the exclusion.

Conformance evidence hierarchy

Direct deterministic proof: schemas, tests, signatures, state reconciliation, access decisions, build or runtime evidence.

Reproducible technical evidence: benchmark fixtures, traces, artifacts, profiles, logs, metrics, and environment manifests.

Independent assessment: qualified human or separate evaluator using an explicit rubric and disclosed uncertainty.

Attestation or supplier claim: informative unless independently verified and current for the deployed configuration.

Demonstration or narrative: insufficient by itself for a conforming score above 2.

Critical-control doctrine

Critical controls cannot be averaged away. A failed or stale critical control constrains the maximum conformance level regardless of the aggregate score.

Scope, Audience, and Engineering Principles

Purpose. Defines AI-assisted automation and infrastructure-as-code evaluation with intent, policy, validation, blast radius, rollback, drift, and evidence controls.

In-scope systems. AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows.

Primary audience. platform, cloud, network, DevOps, security, SRE, change-management, and AI automation teams.

Required engineering principles

Model capability and complete system behavior MUST be measured separately.

Human, legal, production, financial, identity, and governance authority MUST remain external to model confidence or conversational fluency.

Evaluation MUST include expected, prohibited, adversarial, degraded, recovery, and lifecycle-change conditions.

Source authority, data rights, permissions, versions, and freshness MUST be visible in evidence.

Critical outcomes MUST be supported by deterministic validators or independent review wherever feasible.

The organization MUST preserve rollback, revocation, correction, deletion, and retirement paths.

Optimization for score, latency, throughput, tokens, or cost MUST NOT hide safety, quality, privacy, accessibility, or reliability regressions.

Exclusions

This standard does not certify a model, provider, framework, benchmark, dataset, or product. Conformance is limited to the declared system, use case, version, environment, evidence period, and assessment scope.

Conformance and Scoring Model

Level	Weighted threshold	Critical-control condition	Meaning
No conformance	< 50	Any state	Materially incomplete, unverified, or unsafe.
Foundation	>= 50	No critical control scored 0	Defined baseline with partial implementation and evidence.
Operational	>= 65	All applicable critical controls >= 3	Implemented and operationally tested.
Assured	>= 80	All applicable critical controls >= 4	Independently verified with adversarial and recovery evidence.
Continuously Assured	>= 90	All applicable critical controls = 5	Continuous regression, monitoring, freshness, and incident learning.

Weighted score

For applicable controls: weighted contribution = (control score / 5) x control weight. N/A controls are removed from the denominator only after an approved applicability decision. The final score is normalized to 100.

A conformance claim MUST include scope, system and model versions, assessment date, evidence window, evaluator identity, scores by family, critical-control status, unresolved findings, exceptions, limitations, and next review.

Control Score Interpretation

Score	Maturity	Required evidence state
0	Not implemented	Not implemented: the assessed control is absent, materially unknown, or contradicted by evidence.
1	Initial	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Nonconformity classes

Critical: credible risk of serious harm, unauthorized high-impact action, material security or privacy failure, false assurance, or inability to recover.

Major: requirement absent or ineffective across a meaningful part of scope, with no sufficient compensating control.

Minor: localized or low-impact deficiency that does not invalidate the overall control objective.

Observation: improvement opportunity or emerging risk without current nonconformity.

Score validity

A score expires when evidence exceeds its approved freshness period or a material change invalidates the assessment. Current operation is not inferred from a historic assessment.

Standard Architecture and Control Model

01	**Govern**	Scope, ownership, authority, policy, impacts, risk tolerance, exceptions, and lifecycle decisions.
02	**Map**	System boundaries, actors, models, data, prompts, tools, interfaces, populations, dependencies, and failure domains.
03	**Measure**	Representative and hidden evaluation, deterministic validation, human or model judging, uncertainty, performance, and cost.
04	**Manage**	Release gates, least privilege, monitoring, incident response, correction, rollback, revocation, deletion, and retirement.
05	**Prove**	Traceable evidence bundles connecting claims to configurations, tests, artifacts, effects, decisions, and user outcomes.

Assessment unit

The assessment unit is the declared AI system or workflow, not the model name alone. It includes prompts, retrieval, memory, tools, policies, interface, evaluators, infrastructure, human oversight, and operating environment.

Benchmark Dimensions and Weights

Dimension	Weight	Assessment emphasis
Intent and authority	17%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for intent and authority.
Planning and policy	17%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for planning and policy.
Testing and blast radius	20%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for testing and blast radius.
Execution safety	20%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for execution safety.
Drift and portability	10%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for drift and portability.
Evidence and scoring	10%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for evidence and scoring.
Lifecycle	6%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for lifecycle.

Benchmark scores MUST be reported by dimension and by critical-control status. A single aggregate ranking MUST NOT be used to conceal weak safety, authority, privacy, recovery, or evidence performance.

Contents and Control Index

Control Summary Matrix

Control	Requirement	Family	Wt.	Crit.
C01	Automation Use-Case and Risk Contract	Intent and authority	6	YES
C02	Authoritative Source of Truth	Intent and authority	6	YES
C03	Model, Tool, Provider, and Module Inventory	Intent and authority	5	-
C04	Identity, Credentials, and Privilege Separation	Planning and policy	6	YES
C05	Desired-State Schema and Input Validation	Planning and policy	6	YES
C06	Plan Generation and Human-Readable Change Model	Planning and policy	5	YES
C07	Policy-as-Code and Guardrails	Testing and blast radius	5	YES
C08	Unit, Integration, Simulation, and Digital-Twin Testing	Testing and blast radius	5	YES
C09	Blast-Radius Calculation and Change Segmentation	Testing and blast radius	5	YES
C10	Approval, Separation of Duties, and Change Windows	Testing and blast radius	5	YES

Control Summary Matrix - Continued

Control	Requirement	Family	Wt.	Crit.
C11	State, Locking, and Concurrency Control	Execution safety	5	YES
C12	Idempotency, Ordering, and Partial-Effect Safety	Execution safety	5	YES
C13	Apply Execution and Runtime Safeguards	Execution safety	5	YES
C14	Rollback, Compensation, and Data Recovery	Execution safety	5	YES
C15	Drift Discovery and Reconciliation	Drift and portability	5	-
C16	Multi-Vendor and Portability Evaluation	Drift and portability	5	-
C17	Observability, Evidence, and Post-Change Validation	Evidence and scoring	5	YES
C18	Benchmark Method, Scoring, and Failure Disclosure	Evidence and scoring	5	YES
C19	Change Lifecycle, Reassessment, and Retirement	Lifecycle	6	-

Weight integrity

The 19 applicable control weights sum to 100. Family weights match the benchmark dimension model. Critical controls are highlighted in the atomic control pages and assessment workbook.

MYTHOS-AI-0005-C01

Automation Use-Case and Risk Contract

Family: Intent and authority | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain automation use-case and risk contract for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The operating contract must define intended users, decisions, prohibited uses, affected systems, consequences, risk tolerance, terminal conditions, and accountable authority. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that automation use-case and risk contract is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for automation use-case and risk contract covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The operating contract must define intended users, decisions, prohibited uses, affected systems, consequences, risk tolerance, terminal conditions, and accountable authority.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C01

Automation Use-Case and Risk Contract

Family: Intent and authority | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for automation use-case and risk contract.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- tool-call and external-effect receipts
- failure-injection, idempotency, rollback, and post-change validation results

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Automation Use-Case and Risk Contract is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for automation use-case and risk contract.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C02

Authoritative Source of Truth

Family: Intent and authority | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain authoritative source of truth for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Data must have documented origin, rights, purpose, classification, quality, lineage, transformation history, access restrictions, retention, and deletion obligations. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that authoritative source of truth is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for authoritative source of truth covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Data must have documented origin, rights, purpose, classification, quality, lineage, transformation history, access restrictions, retention, and deletion obligations.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C02

Authoritative Source of Truth

Family: Intent and authority | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for authoritative source of truth.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- dataset or source manifest with rights, lineage, quality, and split records
- privacy, retention, deletion, and contamination review

Required metrics

- lineage completeness
- rights-review coverage
- duplicate or contamination rate
- quality-slice pass rate
- deletion-verification rate

Score anchors

0	Not implemented: Authoritative Source of Truth is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for authoritative source of truth.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C03

Model, Tool, Provider, and Module Inventory

Family: Intent and authority | Weight: 5 | Critical: NO

Normative requirement

The organization MUST establish, implement, verify, and maintain model, tool, provider, and module inventory for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The inventory must identify versions, owners, suppliers, deployment locations, interfaces, dependencies, trust boundaries, and supported lifecycle states. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that model, tool, provider, and module inventory is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for model, tool, provider, and module inventory covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The inventory must identify versions, owners, suppliers, deployment locations, interfaces, dependencies, trust boundaries, and supported lifecycle states.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C03

Model, Tool, Provider, and Module Inventory

Family: Intent and authority | Weight: 5 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for model, tool, provider, and module inventory.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- tool-call and external-effect receipts
- failure-injection, idempotency, rollback, and post-change validation results

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Model, Tool, Provider, and Module Inventory is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for model, tool, provider, and module inventory.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C04

Identity, Credentials, and Privilege Separation

Family: Planning and policy | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain identity, credentials, and privilege separation for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The inventory must identify versions, owners, suppliers, deployment locations, interfaces, dependencies, trust boundaries, and supported lifecycle states. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that identity, credentials, and privilege separation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for identity, credentials, and privilege separation covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The inventory must identify versions, owners, suppliers, deployment locations, interfaces, dependencies, trust boundaries, and supported lifecycle states.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C04

Identity, Credentials, and Privilege Separation

Family: Planning and policy | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for identity, credentials, and privilege separation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Identity, Credentials, and Privilege Separation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for identity, credentials, and privilege separation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C05

Desired-State Schema and Input Validation

Family: Planning and policy | Weight: 6 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain desired-state schema and input validation for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that desired-state schema and input validation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for desired-state schema and input validation covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C05

Desired-State Schema and Input Validation

Family: Planning and policy | Weight: 6 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for desired-state schema and input validation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- task success with confidence interval
- critical-failure rate
- slice variance
- reproducibility delta
- human-review burden

Score anchors

0	Not implemented: Desired-State Schema and Input Validation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for desired-state schema and input validation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C06

Plan Generation and Human-Readable Change Model

Family: Planning and policy | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain plan generation and human-readable change model for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Automation must separate intent, planned change, policy decision, approved scope, applied state, observed effect, post-change validation, and rollback while preventing stale or concurrent execution. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that plan generation and human-readable change model is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for plan generation and human-readable change model covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Automation must separate intent, planned change, policy decision, approved scope, applied state, observed effect, post-change validation, and rollback while preventing stale or concurrent execution.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C06

Plan Generation and Human-Readable Change Model

Family: Planning and policy | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for plan generation and human-readable change model.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Plan Generation and Human-Readable Change Model is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for plan generation and human-readable change model.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C07

Policy-as-Code and Guardrails

Family: Testing and blast radius | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain policy-as-code and guardrails for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The benchmark must use repository-level tasks, authoritative context, isolated workspaces, hidden tests, independent review, secure build evidence, change-impact analysis, and regression protection. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that policy-as-code and guardrails is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for policy-as-code and guardrails covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The benchmark must use repository-level tasks, authoritative context, isolated workspaces, hidden tests, independent review, secure build evidence, change-impact analysis, and regression protection.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C07

Policy-as-Code and Guardrails

Family: Testing and blast radius | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for policy-as-code and guardrails.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- immutable repository revision and isolated environment manifest
- patch, build, hidden-test, review, and provenance artifacts

Required metrics

- hidden-test pass rate
- escaped-defect rate
- security finding rate
- review minutes per accepted change
- rework rate

Score anchors

0	Not implemented: Policy-as-Code and Guardrails is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for policy-as-code and guardrails.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C08

Unit, Integration, Simulation, and Digital-Twin Testing

Family: Testing and blast radius | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain unit, integration, simulation, and digital-twin testing for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The benchmark must use repository-level tasks, authoritative context, isolated workspaces, hidden tests, independent review, secure build evidence, change-impact analysis, and regression protection. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that unit, integration, simulation, and digital-twin testing is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for unit, integration, simulation, and digital-twin testing covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The benchmark must use repository-level tasks, authoritative context, isolated workspaces, hidden tests, independent review, secure build evidence, change-impact analysis, and regression protection.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C08

Unit, Integration, Simulation, and Digital-Twin Testing

Family: Testing and blast radius | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for unit, integration, simulation, and digital-twin testing.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- immutable repository revision and isolated environment manifest
- patch, build, hidden-test, review, and provenance artifacts

Required metrics

- hidden-test pass rate
- escaped-defect rate
- security finding rate
- review minutes per accepted change
- rework rate

Score anchors

0	Not implemented: Unit, Integration, Simulation, and Digital-Twin Testing is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for unit, integration, simulation, and digital-twin testing.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C09

Blast-Radius Calculation and Change Segmentation

Family: Testing and blast radius | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain blast-radius calculation and change segmentation for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Automation must separate intent, planned change, policy decision, approved scope, applied state, observed effect, post-change validation, and rollback while preventing stale or concurrent execution. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that blast-radius calculation and change segmentation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for blast-radius calculation and change segmentation covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Automation must separate intent, planned change, policy decision, approved scope, applied state, observed effect, post-change validation, and rollback while preventing stale or concurrent execution.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C09

Blast-Radius Calculation and Change Segmentation

Family: Testing and blast radius | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for blast-radius calculation and change segmentation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Blast-Radius Calculation and Change Segmentation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for blast-radius calculation and change segmentation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C10

Approval, Separation of Duties, and Change Windows

Family: Testing and blast radius | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain approval, separation of duties, and change windows for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Authority must be enforceable outside the model through stable identities, least privilege, scoped credentials, separation of duties, revocation, and auditable approval. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that approval, separation of duties, and change windows is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for approval, separation of duties, and change windows covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Authority must be enforceable outside the model through stable identities, least privilege, scoped credentials, separation of duties, revocation, and auditable approval.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C10

Approval, Separation of Duties, and Change Windows

Family: Testing and blast radius | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for approval, separation of duties, and change windows.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Approval, Separation of Duties, and Change Windows is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for approval, separation of duties, and change windows.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C11

State, Locking, and Concurrency Control

Family: Execution safety | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain state, locking, and concurrency control for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that state, locking, and concurrency control is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for state, locking, and concurrency control covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C11

State, Locking, and Concurrency Control

Family: Execution safety | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for state, locking, and concurrency control.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: State, Locking, and Concurrency Control is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for state, locking, and concurrency control.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C12

Idempotency, Ordering, and Partial-Effect Safety

Family: Execution safety | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain idempotency, ordering, and partial-effect safety for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that idempotency, ordering, and partial-effect safety is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for idempotency, ordering, and partial-effect safety covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must test realistic adversaries, direct and indirect manipulation, multi-turn escalation, cross-component attacks, prohibited outcomes, control bypass, and safe containment.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C12

Idempotency, Ordering, and Partial-Effect Safety

Family: Execution safety | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for idempotency, ordering, and partial-effect safety.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- adversarial case corpus with campaign provenance
- critical-failure findings, control response, and retest evidence

Required metrics

- attack success rate
- critical control bypass rate
- safe abstention or containment rate
- time to detection
- retest closure rate

Score anchors

0	Not implemented: Idempotency, Ordering, and Partial-Effect Safety is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for idempotency, ordering, and partial-effect safety.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C13

Apply Execution and Runtime Safeguards

Family: Execution safety | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain apply execution and runtime safeguards for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that apply execution and runtime safeguards is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for apply execution and runtime safeguards covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must identify accountable ownership, authoritative inputs, expected outputs, operating limits, dependencies, failure behavior, evidence, and reassessment triggers.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C13

Apply Execution and Runtime Safeguards

Family: Execution safety | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for apply execution and runtime safeguards.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- tool-call and external-effect receipts
- failure-injection, idempotency, rollback, and post-change validation results

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Apply Execution and Runtime Safeguards is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for apply execution and runtime safeguards.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C14

Rollback, Compensation, and Data Recovery

Family: Execution safety | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain rollback, compensation, and data recovery for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Data must have documented origin, rights, purpose, classification, quality, lineage, transformation history, access restrictions, retention, and deletion obligations. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that rollback, compensation, and data recovery is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for rollback, compensation, and data recovery covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Data must have documented origin, rights, purpose, classification, quality, lineage, transformation history, access restrictions, retention, and deletion obligations.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C14

Rollback, Compensation, and Data Recovery

Family: Execution safety | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for rollback, compensation, and data recovery.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- dataset or source manifest with rights, lineage, quality, and split records
- privacy, retention, deletion, and contamination review

Required metrics

- lineage completeness
- rights-review coverage
- duplicate or contamination rate
- quality-slice pass rate
- deletion-verification rate

Score anchors

0	Not implemented: Rollback, Compensation, and Data Recovery is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for rollback, compensation, and data recovery.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C15

Drift Discovery and Reconciliation

Family: Drift and portability | Weight: 5 | Critical: NO

Normative requirement

The organization MUST establish, implement, verify, and maintain drift discovery and reconciliation for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. The control must define production indicators, drift thresholds, incident triggers, review frequency, change detection, owner notification, corrective action, and retirement criteria. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that drift discovery and reconciliation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for drift discovery and reconciliation covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- The control must define production indicators, drift thresholds, incident triggers, review frequency, change detection, owner notification, corrective action, and retirement criteria.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C15

Drift Discovery and Reconciliation

Family: Drift and portability | Weight: 5 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for drift discovery and reconciliation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- applicable control coverage
- critical-failure count
- evidence freshness
- open nonconformities

Score anchors

0	Not implemented: Drift Discovery and Reconciliation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for drift discovery and reconciliation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C16

Multi-Vendor and Portability Evaluation

Family: Drift and portability | Weight: 5 | Critical: NO

Normative requirement

The organization MUST establish, implement, verify, and maintain multi-vendor and portability evaluation for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that multi-vendor and portability evaluation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for multi-vendor and portability evaluation covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C16

Multi-Vendor and Portability Evaluation

Family: Drift and portability | Weight: 5 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for multi-vendor and portability evaluation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- task success with confidence interval
- critical-failure rate
- slice variance
- reproducibility delta
- human-review burden

Score anchors

0	Not implemented: Multi-Vendor and Portability Evaluation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for multi-vendor and portability evaluation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C17

Observability, Evidence, and Post-Change Validation

Family: Evidence and scoring | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain observability, evidence, and post-change validation for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that observability, evidence, and post-change validation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for observability, evidence, and post-change validation covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C17

Observability, Evidence, and Post-Change Validation

Family: Evidence and scoring | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for observability, evidence, and post-change validation.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- task success with confidence interval
- critical-failure rate
- slice variance
- reproducibility delta
- human-review burden

Score anchors

0	Not implemented: Observability, Evidence, and Post-Change Validation is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for observability, evidence, and post-change validation.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C18

Benchmark Method, Scoring, and Failure Disclosure

Family: Evidence and scoring | Weight: 5 | Critical: YES

Normative requirement

The organization MUST establish, implement, verify, and maintain benchmark method, scoring, and failure disclosure for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that benchmark method, scoring, and failure disclosure is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for benchmark method, scoring, and failure disclosure covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Evaluation must use representative and hidden cases, reproducible configuration, deterministic validators where possible, statistical uncertainty, critical-failure gates, and slice-level reporting.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C18

Benchmark Method, Scoring, and Failure Disclosure

Family: Evidence and scoring | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for benchmark method, scoring, and failure disclosure.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- task success with confidence interval
- critical-failure rate
- slice variance
- reproducibility delta
- human-review burden

Score anchors

0	Not implemented: Benchmark Method, Scoring, and Failure Disclosure is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for benchmark method, scoring, and failure disclosure.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

MYTHOS-AI-0005-C19

Change Lifecycle, Reassessment, and Retirement

Family: Lifecycle | Weight: 6 | Critical: NO

Normative requirement

The organization MUST establish, implement, verify, and maintain change lifecycle, reassessment, and retirement for all in-scope AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows. Recovery must cover safe stop, checkpoint integrity, rollback or compensation, credential revocation, artifact restoration, evidence preservation, requalification, and controlled retirement. The control MUST define acceptance criteria, prohibited outcomes, evidence, exception authority, and reassessment triggers. Where automated enforcement is feasible, the organization SHOULD enforce the requirement outside the model or agent. Any residual manual dependency MUST be documented, assigned, time-bounded, and tested.

Control objective

Objective

Ensure that change lifecycle, reassessment, and retirement is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for change lifecycle, reassessment, and retirement covering AI-generated and AI-operated infrastructure code, network automation, cloud automation, configuration management, policy, and deployment workflows and name the accountable owner, approver, assessor, and affected stakeholders.
- Recovery must cover safe stop, checkpoint integrity, rollback or compensation, credential revocation, artifact restoration, evidence preservation, requalification, and controlled retirement.
- Implement the control through machine-enforceable policy, typed schemas, isolated runtime boundaries, validated procedures, or an approved combination appropriate to the risk.
- Define explicit nominal, negative, adversarial, malformed, degraded, overloaded, recovery, and rollback tests; critical cases must be hidden from the system under evaluation where feasible.
- Correlate evidence to stable system, model, data, prompt, repository, tool, environment, user or tenant, and release identities; protect records against unauthorized modification.
- Set review cadence and event-driven reassessment triggers for material changes, incidents, supplier updates, drift, failed controls, new affected populations, and retirement.

Applicability

Applicable unless a documented, approved, evidence-supported exclusion demonstrates that the capability, data, decision, interface, population, or lifecycle stage is absent.

MYTHOS-AI-0005-C19

Change Lifecycle, Reassessment, and Retirement

Family: Lifecycle | Weight: 6 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for change lifecycle, reassessment, and retirement.

Select representative normal, boundary, high-impact, and adversarial cases, including at least one prohibited outcome and one dependency or recovery failure.

Verify the exact model, data, prompt, tool, policy, runtime, repository, environment, and evaluator versions used during assessment.

Execute deterministic validators first, then calibrated model or human judgments only where deterministic proof is not available.

Reconcile expected behavior with raw outputs, trajectories, tool effects, telemetry, artifacts, user-visible outcomes, and unresolved contradictions.

Assign a 0-5 score, record evidence links and confidence, open nonconformities, define remediation ownership, and set the next verification date.

Minimum evidence

- approved control design or operating contract with accountable owner
- versioned configuration, schema, policy, model, prompt, dataset, or architecture record
- reproducible test plan with expected and observed results
- traceable decision, exception, approval, and remediation records
- negative, boundary, degraded, and recovery test results
- operational telemetry and current-state verification

Required metrics

- recovery success rate
- duplicate-effect rate
- time to safe state
- residual-effect count
- evidence preservation rate

Score anchors

0	Not implemented: Change Lifecycle, Reassessment, and Retirement is absent, materially unknown, or contradicted by evidence.
1	Initial: intent is documented, but ownership, repeatability, implementation, or evidence is incomplete.
2	Defined: approved procedure and design exist, but implementation or representative testing is partial.
3	Implemented: control operates for the declared scope and passes normal and basic negative tests.
4	Verified: independent assessment, hidden or adversarial cases, current evidence, and recovery validation demonstrate effectiveness.
5	Continuously assured: automated monitoring and regression, current evidence, incident learning, and timely reassessment sustain verified effectiveness.

Failure indicators

The organization cannot identify an authoritative owner, current scope, or complete implementation for change lifecycle, reassessment, and retirement.

Only a successful demonstration, vendor claim, aggregate score, or model-generated judgment is available; raw evidence and negative tests are absent.

Critical cases can be hidden by averages, unsupported N/A designations, stale evidence, or changes in model, data, prompt, tool, policy, or environment.

Failure, rollback, revocation, deletion, containment, or retirement behavior has not been exercised under realistic conditions.

Required Benchmark Scenarios

MYTHOS-AI-0005-B01 - Representative production task

Demonstrate the declared use case using current configuration and representative inputs. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

MYTHOS-AI-0005-B02 - Hidden critical holdout

Execute high-impact cases withheld from development, tuning, and prompt iteration. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

MYTHOS-AI-0005-B03 - Malformed and adversarial inputs

Test schema violations, ambiguity, direct or indirect manipulation, and unsafe instructions. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

MYTHOS-AI-0005-B04 - Long-horizon or multi-step execution

Measure compounding error, context loss, looping, premature completion, and recovery. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

MYTHOS-AI-0005-B05 - Dependency and tool failure

Inject model, network, data, tool, credential, evaluator, or service failure. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

Required Benchmark Scenarios - Continued

MYTHOS-AI-0005-B06 - Resource pressure and overload

Exercise concurrency, long inputs, queueing, rate limits, memory, tokens, and cost limits. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

MYTHOS-AI-0005-B07 - Privacy and cross-boundary test

Attempt cross-tenant, secret, personal-data, restricted-source, cache, and egress violations. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

MYTHOS-AI-0005-B08 - Rollback, revocation, or restore

Prove safe stop, compensation, revocation, prior-state restoration, and residual-effect reconciliation. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

MYTHOS-AI-0005-B09 - Human intervention and disagreement

Test approval, interruption, correction, appeal, assessor disagreement, and minority findings. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

MYTHOS-AI-0005-B10 - Material change and regression

Change model, prompt, dataset, tool, provider, runtime, or policy and run requalification gates. Apply the scenario specifically to ai automation and iac benchmark and preserve raw evidence.

Conformance Report and Evidence Bundle

Permanent document ID, standard version, assessment version, system and use-case scope.

System, model, provider, prompt, data, retrieval, memory, tool, evaluator, runtime, repository, and infrastructure identities.

Applicability decisions and normalized denominator.

Control-level scores, weights, critical status, evidence links, assessor, date, confidence, and finding disposition.

Benchmark scenario results with raw artifacts, deterministic validators, judge or expert records, uncertainty, failures, and limitations.

Family and total score, achieved conformance level, unresolved critical or major findings, approved exceptions, expiry, and next review.

Release, rollback, revocation, deletion, and retirement evidence where applicable.

Evidence bundle integrity

The evidence bundle SHOULD use content hashes, signatures or protected repository history, stable artifact IDs, access controls, retention policy, and independent verification. Evidence links alone are insufficient when the referenced object can change without detection.

Exceptions, Waivers, and Compensating Controls

An exception MUST NOT be used to relabel a failed requirement as conforming. Exceptions document accepted residual risk for a bounded scope and time; the underlying control score remains evidence-based.

- Identify the exact control, system scope, reason, affected users or assets, and current score.
- Describe the missing requirement, residual risk, foreseeable failure, and affected decisions.
- Define compensating controls, validation evidence, monitoring, owner, approver, start date, expiry, and remediation plan.
- Obtain approval from an authority independent of the implementation owner for critical or high-impact exceptions.
- Reassess on incident, material change, evidence expiry, control failure, or exception expiration.
- Publish exceptions in the conformance report; hidden exceptions invalidate the claim.

Critical exception cap

An unresolved exception against a critical control prevents Assured or Continuously Assured conformance and may prevent Operational conformance when the control score is below 3.

Source Authority and Freshness Register

NIST - Artificial Intelligence Risk Management Framework 1.0

Cross-sector AI risk governance, mapping, measurement, and management.

https://doi.org/10.6028/NIST.AI.100-1

NIST - NIST AI 600-1 - Generative Artificial Intelligence Profile

Generative-AI risks, controls, evaluation, and lifecycle actions.

https://doi.org/10.6028/NIST.AI.600-1

NIST - AI Resource Center and AI RMF Playbook

Current implementation resources and revision notices for the AI RMF.

https://airc.nist.gov/

ISO/IEC - ISO/IEC 42001:2023 - AI management systems

AI management-system requirements and continual improvement.

https://www.iso.org/standard/42001

ISO/IEC - ISO/IEC 23894:2023 - Guidance on AI risk management

AI risk-management guidance across lifecycle and stakeholders.

https://www.iso.org/standard/77304.html

ISO/IEC - ISO/IEC 42005:2025 - AI system impact assessment

Impact-assessment guidance for individuals and societies.

https://www.iso.org/standard/42005

MLCommons - MLCommons Benchmarks

Open performance, quality, and safety benchmark programs.

https://mlcommons.org/benchmarks/

Source Authority and Freshness Register - Continued

MLCommons - AI Risk and Reliability

Safety-test methodology and risk-oriented evaluation work.

https://mlcommons.org/working-groups/ai-risk-reliability/ai-risk-reliability/

HashiCorp - Terraform plan and test documentation

Plan generation, speculative plans, and execution workflow.

https://developer.hashicorp.com/terraform/cli/commands/plan

OpenTofu - OpenTofu test command

Native module and configuration testing.

https://opentofu.org/docs/cli/commands/test/

Open Policy Agent - OPA Documentation

Policy-as-code decision and test framework.

https://www.openpolicyagent.org/docs

NIST - SP 800-53 Rev. 5

Security and privacy control baseline for automated systems.

https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

Freshness rule

Sources were verified for this candidate edition on 2026-07-26. The standard MUST be revalidated when a cited authority changes, becomes superseded, is withdrawn, or materially alters the control interpretation.

Limitations, Revision History, and Adoption Position

This candidate Mythos standard is designed for engineering governance and assessment. It is not a legal opinion, regulatory certification, safety guarantee, or substitute for domain-specific professional judgment. Model behavior remains probabilistic and system risk depends on data, users, tools, deployment, incentives, and operating context.

Revision history

Version	Date	Change
2.0.0-candidate	2026-07-26	Expanded normative edition with 19 weighted controls, benchmark scenarios, conformance levels, machine-readable catalogs, assessment workbook integration, source freshness, and explicit lineage.
1.0.0-candidate	2026-07-24	Earlier permanent-identity candidate retained in the release lineage archive.

Adoption position

Use this edition as a candidate control and benchmark baseline. Organizations SHOULD pilot it on bounded systems, retain assessment evidence, submit corrections, and avoid representing candidate conformance as external certification.