



Context Engineering, Trajectory Compaction, and Temporal Memory Standard

Defines context construction, token allocation, compaction, chronology, memory, caching, multi-agent handoffs, privacy, and long-running-task integrity.

MYTHOS-AI-0007

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-0007
Canonical title	Context Engineering, Trajectory Compaction, and Temporal Memory 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 context construction, token allocation, compaction, chronology, memory, caching, multi-agent handoffs, privacy, and long-running-task integrity.

In-scope systems. prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state.

Primary audience. AI application, agent-runtime, memory, RAG, developer-tool, privacy, security, and evaluation engineers.

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
Architecture and authority	18%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for architecture and authority.
Token and selection quality	14%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for token and selection quality.
Long-context and caching	14%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for long-context and caching.
Trajectory and temporal integrity	20%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for trajectory and temporal integrity.
Memory and multi-agent state	14%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for memory and multi-agent state.
Security and privacy	14%	Control effectiveness, evidence quality, critical failures, operational limits, and lifecycle behavior for security and privacy.
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

[Permanent identity and edition record](#)

[Normative language and applicability](#)

[Scope, audience, and engineering principles](#)

[Conformance and scoring model](#)

[Standard architecture and control model](#)

[Benchmark dimensions and weights](#)

[Control summary matrix](#)

MYTHOS-AI-0007-C01 - Context Architecture and Data Classes

MYTHOS-AI-0007-C02 - Token Accounting and Budget Allocation

MYTHOS-AI-0007-C03 - Instruction Hierarchy and Provenance

MYTHOS-AI-0007-C04 - Access, Tenant, and Purpose Boundaries

MYTHOS-AI-0007-C05 - Relevance and Dependency Selection

MYTHOS-AI-0007-C06 - Source Authority and Freshness

MYTHOS-AI-0007-C07 - Context Ordering and Delimitation

MYTHOS-AI-0007-C08 - Long-Context and Position-Sensitivity Controls

MYTHOS-AI-0007-C09 - Prompt and Prefix Cache Identity

MYTHOS-AI-0007-C10 - Trajectory Event Model

MYTHOS-AI-0007-C11 - Checkpoint and Resume Contract

MYTHOS-AI-0007-C12 - Summary and Compaction Fidelity

MYTHOS-AI-0007-C13 - Temporal Ordering and Version Integrity

MYTHOS-AI-0007-C14 - Memory Tiering and Promotion

MYTHOS-AI-0007-C15 - Multi-Agent Shared State and Handoffs

MYTHOS-AI-0007-C16 - Context Injection and Memory Poisoning Defense

MYTHOS-AI-0007-C17 - Privacy, Minimization, and Egress

MYTHOS-AI-0007-C18 - Context Quality and Compaction Evaluation

MYTHOS-AI-0007-C19 - Retention, Deletion, Reassessment, and Retirement

Control Summary Matrix

Control	Requirement	Family	Wt.	Crit.
C01	Context Architecture and Data Classes	Architecture and authority	6	YES
C02	Token Accounting and Budget Allocation	Architecture and authority	6	YES
C03	Instruction Hierarchy and Provenance	Architecture and authority	6	YES
C04	Access, Tenant, and Purpose Boundaries	Token and selection quality	5	YES
C05	Relevance and Dependency Selection	Token and selection quality	5	-
C06	Source Authority and Freshness	Token and selection quality	4	YES
C07	Context Ordering and Delimitation	Long-context and caching	5	-
C08	Long-Context and Position-Sensitivity Controls	Long-context and caching	5	YES
C09	Prompt and Prefix Cache Identity	Long-context and caching	4	YES
C10	Trajectory Event Model	Trajectory and temporal integrity	5	YES

Control Summary Matrix - Continued

Control	Requirement	Family	Wt.	Crit.
C11	Checkpoint and Resume Contract	Trajectory and temporal integrity	5	YES
C12	Summary and Compaction Fidelity	Trajectory and temporal integrity	5	YES
C13	Temporal Ordering and Version Integrity	Trajectory and temporal integrity	5	YES
C14	Memory Tiering and Promotion	Memory and multi-agent state	5	-
C15	Multi-Agent Shared State and Handoffs	Memory and multi-agent state	5	YES
C16	Context Injection and Memory Poisoning Defense	Memory and multi-agent state	4	YES
C17	Privacy, Minimization, and Egress	Security and privacy	7	YES
C18	Context Quality and Compaction Evaluation	Security and privacy	7	YES
C19	Retention, Deletion, 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-0007-C01

Context Architecture and Data Classes

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

Normative requirement

The organization **MUST** establish, implement, verify, and maintain context architecture and data classes for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 context architecture and data classes is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for context architecture and data classes covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C01

Context Architecture and Data Classes

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

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for context architecture and data classes.

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: Context Architecture and Data Classes 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 context architecture and data classes.

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-0007-C02

Token Accounting and Budget Allocation

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

Normative requirement

The organization **MUST** establish, implement, verify, and maintain token accounting and budget allocation for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 token accounting and budget allocation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for token accounting and budget allocation covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C02

Token Accounting and Budget Allocation

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

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for token accounting and budget allocation.

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: Token Accounting and Budget Allocation 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 token accounting and budget allocation.

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-0007-C03

Instruction Hierarchy and Provenance

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

Normative requirement

The organization **MUST** establish, implement, verify, and maintain instruction hierarchy and provenance for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Evidence must correlate system identity, version, configuration, inputs, outputs, actions, approvals, metrics, artifacts, findings, exceptions, and final outcomes with integrity 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 instruction hierarchy and provenance is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for instruction hierarchy and provenance covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.

Evidence must correlate system identity, version, configuration, inputs, outputs, actions, approvals, metrics, artifacts, findings, exceptions, and final outcomes with integrity 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-0007-C03

Instruction Hierarchy and Provenance

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

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for instruction hierarchy and provenance.

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: Instruction Hierarchy and Provenance 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 instruction hierarchy and provenance.

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-0007-C04

Access, Tenant, and Purpose Boundaries

Family: Token and selection quality | Weight: 5 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain access, tenant, and purpose boundaries for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Security must isolate tenants, files, processes, networks, secrets, models, caches, tools, and external effects according to an explicit threat model and negative tests. 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 access, tenant, and purpose boundaries is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for access, tenant, and purpose boundaries covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.

Security must isolate tenants, files, processes, networks, secrets, models, caches, tools, and external effects according to an explicit threat model and negative tests.

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-0007-C04

Access, Tenant, and Purpose Boundaries

Family: Token and selection quality | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for access, tenant, and purpose boundaries.

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: Access, Tenant, and Purpose Boundaries 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 access, tenant, and purpose boundaries.

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-0007-C05

Relevance and Dependency Selection

Family: Token and selection quality | Weight: 5 | Critical: NO

Normative requirement

The organization **MUST** establish, implement, verify, and maintain relevance and dependency selection for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 relevance and dependency selection is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for relevance and dependency selection covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C05

Relevance and Dependency Selection

Family: Token and selection quality | Weight: 5 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for relevance and dependency selection.

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: Relevance and Dependency Selection 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 relevance and dependency selection.

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-0007-C06

Source Authority and Freshness

Family: Token and selection quality | Weight: 4 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain source authority and freshness for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 source authority and freshness is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for source authority and freshness covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C06

Source Authority and Freshness

Family: Token and selection quality | Weight: 4 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for source authority and freshness.

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: Source Authority and Freshness 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 source authority and freshness.

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-0007-C07

Context Ordering and Delimitation

Family: Long-context and caching | Weight: 5 | Critical: NO

Normative requirement

The organization **MUST** establish, implement, verify, and maintain context ordering and delimitation for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff. 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 context ordering and delimitation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for context ordering and delimitation covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.

Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff.

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-0007-C07

Context Ordering and Delimitation

Family: Long-context and caching | Weight: 5 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for context ordering and delimitation.

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

context assembly and token accounting trace

checkpoint, summary-fidelity, stale-state, and restoration tests

Required metrics

decision and evidence retention

stale-context rate

tokens per verified outcome

restore fidelity

cross-tenant leakage rate

Score anchors

0	Not implemented: Context Ordering and Delimitation 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 context ordering and delimitation.

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-0007-C08

Long-Context and Position-Sensitivity Controls

Family: Long-context and caching | Weight: 5 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain long-context and position-sensitivity controls for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff. 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 long-context and position-sensitivity controls is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

- Establish a versioned control contract for long-context and position-sensitivity controls covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.
- Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff.
- 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-0007-C08

Long-Context and Position-Sensitivity Controls

Family: Long-context and caching | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for long-context and position-sensitivity controls.

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

context assembly and token accounting trace

checkpoint, summary-fidelity, stale-state, and restoration tests

Required metrics

decision and evidence retention

stale-context rate

tokens per verified outcome

restore fidelity

cross-tenant leakage rate

Score anchors

0	Not implemented: Long-Context and Position-Sensitivity Controls 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 long-context and position-sensitivity controls.

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-0007-C09

Prompt and Prefix Cache Identity

Family: Long-context and caching | Weight: 4 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain prompt and prefix cache identity for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 prompt and prefix cache identity is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for prompt and prefix cache identity covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C09

Prompt and Prefix Cache Identity

Family: Long-context and caching | Weight: 4 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for prompt and prefix cache identity.

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

context assembly and token accounting trace

checkpoint, summary-fidelity, stale-state, and restoration tests

Required metrics

decision and evidence retention

stale-context rate

tokens per verified outcome

restore fidelity

cross-tenant leakage rate

Score anchors

0	Not implemented: Prompt and Prefix Cache Identity 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 prompt and prefix cache identity.

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-0007-C10

Trajectory Event Model

Family: Trajectory and temporal integrity | Weight: 5 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain trajectory event model for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff. 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 trajectory event model is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for trajectory event model covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.

Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff.

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-0007-C10

Trajectory Event Model

Family: Trajectory and temporal integrity | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for trajectory event 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

context assembly and token accounting trace

checkpoint, summary-fidelity, stale-state, and restoration tests

Required metrics

decision and evidence retention

stale-context rate

tokens per verified outcome

restore fidelity

cross-tenant leakage rate

Score anchors

0	Not implemented: Trajectory Event 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 trajectory event 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-0007-C11

Checkpoint and Resume Contract

Family: Trajectory and temporal integrity | Weight: 5 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain checkpoint and resume contract for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Reproduction must pin code, data, models, prompts, schemas, tools, dependencies, environment, seeds, hardware-relevant settings, and retained artifacts. 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 checkpoint and resume contract is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for checkpoint and resume contract covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.

Reproduction must pin code, data, models, prompts, schemas, tools, dependencies, environment, seeds, hardware-relevant settings, and retained artifacts.

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-0007-C11

Checkpoint and Resume Contract

Family: Trajectory and temporal integrity | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for checkpoint and resume 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

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: Checkpoint and Resume 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 checkpoint and resume 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-0007-C12

Summary and Compaction Fidelity

Family: Trajectory and temporal integrity | Weight: 5 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain summary and compaction fidelity for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff. 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 summary and compaction fidelity is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for summary and compaction fidelity covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.

Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff.

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-0007-C12

Summary and Compaction Fidelity

Family: Trajectory and temporal integrity | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for summary and compaction fidelity.

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

context assembly and token accounting trace

checkpoint, summary-fidelity, stale-state, and restoration tests

Required metrics

decision and evidence retention

stale-context rate

tokens per verified outcome

restore fidelity

cross-tenant leakage rate

Score anchors

0	Not implemented: Summary and Compaction Fidelity 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 summary and compaction fidelity.

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-0007-C13

Temporal Ordering and Version Integrity

Family: Trajectory and temporal integrity | Weight: 5 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain temporal ordering and version integrity for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff. 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 temporal ordering and version integrity is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for temporal ordering and version integrity covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.

Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff.

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-0007-C13

Temporal Ordering and Version Integrity

Family: Trajectory and temporal integrity | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for temporal ordering and version integrity.

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: Temporal Ordering and Version Integrity 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 temporal ordering and version integrity.

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-0007-C14

Memory Tiering and Promotion

Family: Memory and multi-agent state | Weight: 5 | Critical: NO

Normative requirement

The organization **MUST** establish, implement, verify, and maintain memory tiering and promotion for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff. 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 memory tiering and promotion is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for memory tiering and promotion covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state and name the accountable owner, approver, assessor, and affected stakeholders.

Context state must preserve authority, chronology, source identity, permissions, unresolved work, decisions, evidence, cache identity, and restoration fidelity under compression or handoff.

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-0007-C14

Memory Tiering and Promotion

Family: Memory and multi-agent state | Weight: 5 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for memory tiering and promotion.

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

context assembly and token accounting trace

checkpoint, summary-fidelity, stale-state, and restoration tests

Required metrics

decision and evidence retention

stale-context rate

tokens per verified outcome

restore fidelity

cross-tenant leakage rate

Score anchors

0	Not implemented: Memory Tiering and Promotion 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 memory tiering and promotion.

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-0007-C15

Multi-Agent Shared State and Handoffs

Family: Memory and multi-agent state | Weight: 5 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain multi-agent shared state and handoffs for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 multi-agent shared state and handoffs is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for multi-agent shared state and handoffs covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C15

Multi-Agent Shared State and Handoffs

Family: Memory and multi-agent state | Weight: 5 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for multi-agent shared state and handoffs.

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: Multi-Agent Shared State and Handoffs 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-agent shared state and handoffs.

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-0007-C16

Context Injection and Memory Poisoning Defense

Family: Memory and multi-agent state | Weight: 4 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain context injection and memory poisoning defense for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 context injection and memory poisoning defense is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for context injection and memory poisoning defense covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C16

Context Injection and Memory Poisoning Defense

Family: Memory and multi-agent state | Weight: 4 | Critical: YES

Assessment procedure

- Examine the approved design, applicability decision, responsible roles, and current implementation for context injection and memory poisoning defense.
- 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: Context Injection and Memory Poisoning Defense 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 context injection and memory poisoning defense.
- 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-0007-C17

Privacy, Minimization, and Egress

Family: Security and privacy | Weight: 7 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain privacy, minimization, and egress for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 privacy, minimization, and egress is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for privacy, minimization, and egress covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C17

Privacy, Minimization, and Egress

Family: Security and privacy | Weight: 7 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for privacy, minimization, and egress.

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: Privacy, Minimization, and Egress 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 privacy, minimization, and egress.

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-0007-C18

Context Quality and Compaction Evaluation

Family: Security and privacy | Weight: 7 | Critical: YES

Normative requirement

The organization **MUST** establish, implement, verify, and maintain context quality and compaction evaluation for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 context quality and compaction evaluation is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for context quality and compaction evaluation covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C18

Context Quality and Compaction Evaluation

Family: Security and privacy | Weight: 7 | Critical: YES

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for context quality and compaction 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

context assembly and token accounting trace

checkpoint, summary-fidelity, stale-state, and restoration tests

Required metrics

task success with confidence interval

critical-failure rate

slice variance

reproducibility delta

human-review burden

Score anchors

0	Not implemented: Context Quality and Compaction 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 context quality and compaction 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-0007-C19

Retention, Deletion, Reassessment, and Retirement

Family: Lifecycle | Weight: 6 | Critical: NO

Normative requirement

The organization **MUST** establish, implement, verify, and maintain retention, deletion, reassessment, and retirement for all in-scope prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state. 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 retention, deletion, reassessment, and retirement is explicit, bounded, testable, observable, recoverable, and governed throughout the lifecycle.

Implementation requirements

Establish a versioned control contract for retention, deletion, reassessment, and retirement covering prompt assemblies, agent trajectories, context stores, caches, conversation state, repository context, memory systems, and multi-agent shared state 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-0007-C19

Retention, Deletion, Reassessment, and Retirement

Family: Lifecycle | Weight: 6 | Critical: NO

Assessment procedure

Examine the approved design, applicability decision, responsible roles, and current implementation for retention, deletion, 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: Retention, Deletion, 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 retention, deletion, 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-0007-B01 - Representative production task

Demonstrate the declared use case using current configuration and representative inputs. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

MYTHOS-AI-0007-B02 - Hidden critical holdout

Execute high-impact cases withheld from development, tuning, and prompt iteration. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

MYTHOS-AI-0007-B03 - Malformed and adversarial inputs

Test schema violations, ambiguity, direct or indirect manipulation, and unsafe instructions. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

MYTHOS-AI-0007-B04 - Long-horizon or multi-step execution

Measure compounding error, context loss, looping, premature completion, and recovery. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

MYTHOS-AI-0007-B05 - Dependency and tool failure

Inject model, network, data, tool, credential, evaluator, or service failure. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

Required Benchmark Scenarios - Continued

MYTHOS-AI-0007-B06 - Resource pressure and overload

Exercise concurrency, long inputs, queueing, rate limits, memory, tokens, and cost limits. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

MYTHOS-AI-0007-B07 - Privacy and cross-boundary test

Attempt cross-tenant, secret, personal-data, restricted-source, cache, and egress violations. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

MYTHOS-AI-0007-B08 - Rollback, revocation, or restore

Prove safe stop, compensation, revocation, prior-state restoration, and residual-effect reconciliation. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

MYTHOS-AI-0007-B09 - Human intervention and disagreement

Test approval, interruption, correction, appeal, assessor disagreement, and minority findings. Apply the scenario specifically to context and temporal memory and preserve raw evidence.

MYTHOS-AI-0007-B10 - Material change and regression

Change model, prompt, dataset, tool, provider, runtime, or policy and run requalification gates. Apply the scenario specifically to context and temporal memory 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/>

Research - Lost in the Middle

Long-context position sensitivity and retrieval behavior.

<https://arxiv.org/abs/2307.03172>

Research - MemGPT

Tiered memory and context-management architecture.

<https://arxiv.org/abs/2310.08560>

Research - Transformer-XL

Segment recurrence and temporal context modeling.

<https://arxiv.org/abs/1901.02860>

Model Context Protocol - MCP Specification 2025-11-25

Typed resources, prompts, tools, lifecycle, and durable tasks.

<https://modelcontextprotocol.io/specification/2025-11-25>

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.