



ENGINEERING STANDARDS LIBRARY / CLOUD ARCHITECTURE AND COMPUTE

Kubernetes, Container Orchestration, and Operator Standard

Cluster governance, API and admission control, workload isolation, image provenance, operators, fleet lifecycle, recovery, and evidence.

PERMANENT DOCUMENT ID

MYTHOS-CLD-0007

PUBLICATION

Candidate Standard
Version 1.0.0-candidate

ASSURANCE PROFILE

Standard / Advanced
High Assurance

PERMANENT PUBLICATION IDENTITY

Kubernetes, Container Orchestration, and Operator Standard			DOCUMENT ID MYTHOS-CLD-0007 VERSION 1.0.0-candidate STATUS Candidate Standard PUBLISHER Mythos Systems PUBLICATION DATE 2026-07-24 SCHEDULED REVIEW 2027-01-24 CLASSIFICATION Public
18 ATOMIC CRITERIA	6 ASSURANCE VIEWS	4 IMPLEMENTATION PROFILES	1 PERMANENT IDENTITY

Publication doctrine. This publication establishes durable requirements for its defined scope. Interpretation must preserve the permanent document identity, required evidence, controlled exceptions, source currency, and formal revision history.

INFRASTRUCTURE AND CONTROL TOPOLOGY

Cloud Architecture and Compute

A trustworthy Kubernetes and container orchestration system is a governed chain of ownership, identity, desired state, enforcement, workload or device behavior, telemetry, recovery, and independently reviewable evidence.

OWNERSHIP

Purpose, service boundary, accountable owner, suppliers, users, and retained responsibility remain explicit.

ENFORCEMENT

Identity, policy, segmentation, configuration, and local safety mechanisms constrain every trust transition.

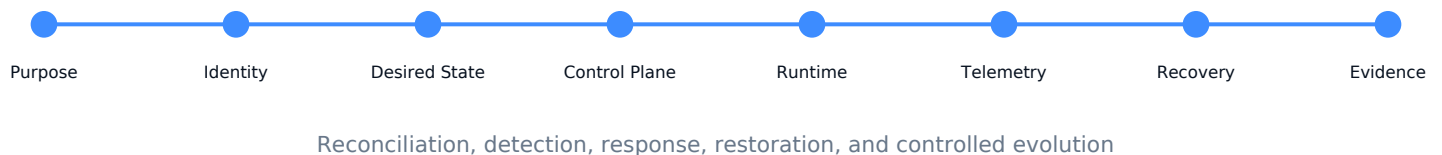
CONTINUITY

Capacity, dependency loss, safe degradation, backup, restoration, rollback, and exit are tested before reliance.

EVIDENCE

Inventory, desired state, runtime observation, tests, incidents, exceptions, and change records form the assurance chain.

GOVERNED INFRASTRUCTURE FLOW



INTERPRETATION AND ASSURANCE

How to apply this publication

Normative intent

Requirements define outcomes and constraints. Implementations may vary, but evidence must demonstrate equivalent control.

Evidence over assertion

Vendor documentation and design intent are not equivalent to reproduced behavior. Record evidence grade and uncertainty.

Risk-proportional depth

Higher consequence, autonomy, sensitivity, and irreversibility require stronger isolation, review, verification, and recovery.

Controlled exception

Exceptions require an owner, rationale, compensating control, residual-risk acceptance, expiration, and reevaluation trigger.

Normalized assessment scale

0	1	2	3	4	5
ABSENT	AD HOC	PARTIAL	OPERATIONAL	ADVANCED	LEADING

A numeric score must never hide a failed mandatory gate, missing evidence, untested workload, or unacceptable residual risk.

INTERACTIVE NAVIGATION

Contents

Executive mandate	6
Scope and applicability	7
Definitions and taxonomy	8
Architecture and trust model	9
Governance and accountability	10
Normative control system	11
Evidence and assessment	16
Assurance views and metrics	17
Failure modes and adversarial scenarios	19
Implementation profiles and lifecycle	20
Adoption roadmap and conformance	22
Source authority and revision control	24

Navigation doctrine

Bookmarks and internal links are conveniences. Permanent document identity, criterion identifiers, evidence references, and revision history remain authoritative.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

0. Executive Mandate

Purpose, strategic outcome, and non-negotiable operating doctrine

MANDATE

Organizations adopting Kubernetes and container orchestration capabilities **MUST** define an owned service boundary, establish enforceable identity and configuration, protect data and safety, test failure and recovery, and preserve evidence sufficient for independent review.

Required outcomes

- Create a durable governance boundary for Kubernetes and container orchestration across design, acquisition, deployment, operation, change, and retirement.
- Require risk-proportional segmentation, identity, configuration, telemetry, resilience, recovery, and supplier controls.
- Prevent central automation, administrative tooling, or provider services from becoming unbounded authority.
- Prove operation with representative workloads, devices, failure modes, and restoration exercises.
- Define measurable conformance, exceptions, reassessment triggers, and a viable exit path.

Decision boundary: conformance supports a documented assurance claim for the named scope. It does not prove universal availability, safety, regulatory acceptance, vendor fitness, or immunity from cyber-physical failure.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

1. Scope and Applicability

Boundary definition, audience, exclusions, and triggering conditions

IN SCOPE

- Architecture, acquisition, configuration, integration, and operation of Kubernetes and container orchestration capabilities.
- Human, workload, device, service, network, data, facility, provider, supplier, and evidence dependencies.
- Design, deployment, monitoring, support, incident response, recovery, migration, and retirement.
- On-premises, hosted, managed, carrier, manufacturer, integrator, and externally operated components.

OUT OF SCOPE

- Claims of legal, safety, carrier, or regulatory approval without the responsible authority.
- Vendor certification treated as proof of the adopter configuration or operating effectiveness.
- Theoretical or laboratory behavior presented as production evidence without a declared boundary.
- Inherited controls without documented scope, owner, period, limitations, and shared responsibility.

Applicability triggers

- The capability supports a business-critical, safety-relevant, regulated, public, or externally dependent service.
- The environment can expose sensitive data, identity, control, physical process, communications, or shared infrastructure.
- A management plane, automation system, carrier, provider, or supplier can affect broad operational scope.
- Failure, compromise, or change can be difficult to detect, reverse, isolate, or recover.
- The system depends on hardware, firmware, radio, browser, cloud, endpoint, IoT, OT, or real-time media behavior.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

2. Definitions and Taxonomy

Controlled language for architecture, governance, and assessment

AUTHORITY

The right to approve, deny, constrain, or execute an action. Model output is not authority.

EVIDENCE

A reproducible source, trace, measurement, test, record, signed artifact, or documented observation supporting a claim.

ASSURANCE VIEW

A defined perspective such as governance, architecture, security, operations, privacy, or human oversight.

ATOMIC CRITERION

A uniquely identified requirement that can be assessed, evidenced, excepted, and revised independently.

IMPLEMENTATION PROFILE

A risk-proportional set of required depth, evidence, and gates for a class of use.

CONTROLLED EXCEPTION

A time-bounded deviation with owner, rationale, compensating control, residual risk, expiration, and review trigger.

DETERMINISTIC MECHANISM

A component whose inputs, policy, state transition, and side effects can be constrained and verified.

SOURCE AUTHORITY

The documented basis for treating a source as reliable for a defined claim, context, jurisdiction, and time.

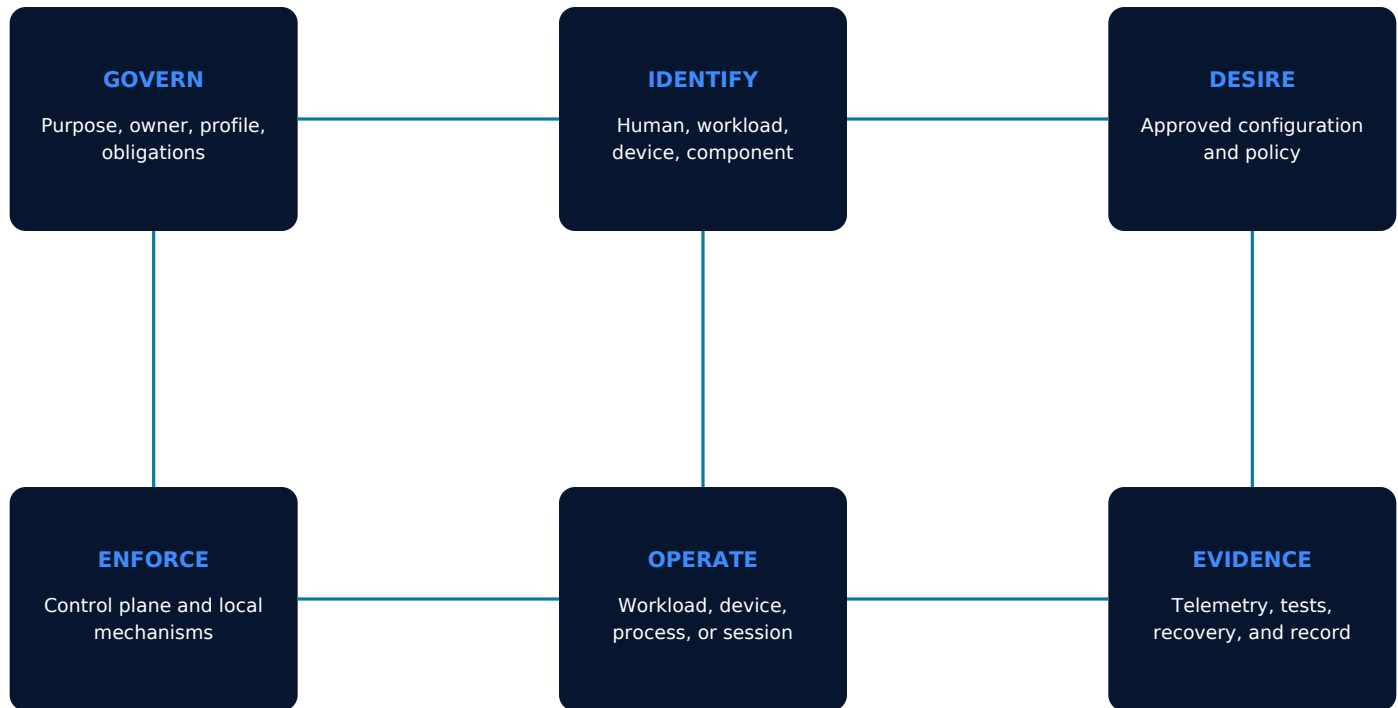
REASSESSMENT TRIGGER

A change, incident, drift signal, dependency revision, or time boundary requiring renewed evaluation.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

3. Architecture and Trust Model

Separation of governance, management, enforcement, runtime, telemetry, and recovery



Mandatory trust boundaries

- Management planes **MUST** be isolated, strongly authenticated, monitored, and recoverable.
- Identity and policy **MUST** be evaluated at every provider, network, device, workload, and administrative transition.
- Runtime state **MUST** be compared with approved desired state and material drift **MUST** trigger response.
- Safety and continuity mechanisms **MUST** remain available when cloud, WAN, identity, DNS, time, carrier, or management dependencies fail.
- Evidence **MUST** be protected from the systems and administrators whose behavior it is intended to assess.

4. Governance and Accountability

Decision rights, separation of duties, and lifecycle ownership

ACCOUNTABLE OWNER

Accepts scope, risk tolerance, funding, and residual risk.

SYSTEM OWNER

Maintains architecture, inventory, operating boundaries, and change control.

SECURITY / PRIVACY

Defines threat, identity, data, isolation, monitoring, and incident requirements.

EVALUATION AUTHORITY

Designs representative tests, gates releases, and preserves reproducibility.

OPERATIONS

Maintains availability, rollback, recovery, evidence, and incident handling.

HUMAN OVERSIGHT

Reviews consequential decisions, exceptions, disputed outcomes, and harm signals.

DECISION PRINCIPLE

No single actor should be able to define the objective, grant authority, execute the consequential action, suppress evidence, and approve the result. Separation must be technical where consequence requires it.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

5. Normative Control System

Atomic criteria 01-04 of 18

MYTHOS-CLD-0007-C01 Cluster Inventory, Ownership, and Criticality

The organization **MUST** define, implement, verify, and maintain cluster inventory, ownership, and criticality for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE approved service contract; architecture decision; owner and risk sign-off

MYTHOS-CLD-0007-C02 API Server Exposure and Administrative Access

The organization **MUST** define, implement, verify, and maintain api server exposure and administrative access for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE asset, service, dependency, supplier, region, and lifecycle inventory

MYTHOS-CLD-0007-C03 Authentication, RBAC, and Service-Account Governance

The organization **MUST** define, implement, verify, and maintain authentication, rbac, and service-account governance for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE desired-state configuration; policy evidence; privileged-access and negative tests

MYTHOS-CLD-0007-C04 Admission Control and Policy Enforcement

The organization **MUST** define, implement, verify, and maintain admission control and policy enforcement for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE network, data-flow, zone, conduit, radio, device, or trust-boundary diagrams

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

5. Normative Control System

Atomic criteria 05-08 of 18

MYTHOS-CLD-0007-C05**Pod Security and Workload Isolation**

The organization **MUST** define, implement, verify, and maintain pod security and workload isolation for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE runtime logs, traces, metrics, alarms, quality, capacity, and drift evidence

MYTHOS-CLD-0007-C06**Network Policy, Service Mesh, and Egress Control**

The organization **MUST** define, implement, verify, and maintain network policy, service mesh, and egress control for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE failure, failover, rollback, restoration, revocation, and recovery exercise

MYTHOS-CLD-0007-C07**Secrets, Certificates, and Key Lifecycle**

The organization **MUST** define, implement, verify, and maintain secrets, certificates, and key lifecycle for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE safety, privacy, residency, consent, or physical-impact assessment

MYTHOS-CLD-0007-C08**Image Provenance, Signing, and Registry Governance**

The organization **MUST** define, implement, verify, and maintain image provenance, signing, and registry governance for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE supplier, carrier, firmware, license, support, portability, and exit evidence

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

5. Normative Control System

Atomic criteria 09-12 of 18

MYTHOS-CLD-0007-C09**Node Operating System and Runtime Hardening**

The organization **MUST** define, implement, verify, and maintain node operating system and runtime hardening for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE approved service contract; architecture decision; owner and risk sign-off

MYTHOS-CLD-0007-C10**Storage Classes, Persistent Data, and Encryption**

The organization **MUST** define, implement, verify, and maintain storage classes, persistent data, and encryption for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE asset, service, dependency, supplier, region, and lifecycle inventory

MYTHOS-CLD-0007-C11**Operator, Controller, and Custom-Resource Governance**

The organization **MUST** define, implement, verify, and maintain operator, controller, and custom-resource governance for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE desired-state configuration; policy evidence; privileged-access and negative tests

MYTHOS-CLD-0007-C12**Multi-Tenancy and Namespace Boundary Assurance**

The organization **MUST** define, implement, verify, and maintain multi-tenancy and namespace boundary assurance for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE network, data-flow, zone, conduit, radio, device, or trust-boundary diagrams

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

5. Normative Control System

Atomic criteria 13-15 of 18

MYTHOS-CLD-0007-C13**GitOps, Desired State, and Drift Reconciliation**

The organization **MUST** define, implement, verify, and maintain gitops, desired state, and drift reconciliation for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE runtime logs, traces, metrics, alarms, quality, capacity, and drift evidence

MYTHOS-CLD-0007-C14**Observability, Audit, and Runtime Detection**

The organization **MUST** define, implement, verify, and maintain observability, audit, and runtime detection for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE failure, failover, rollback, restoration, revocation, and recovery exercise

MYTHOS-CLD-0007-C15**Backup, Restore, and Cluster Reconstitution**

The organization **MUST** define, implement, verify, and maintain backup, restore, and cluster reconstitution for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE safety, privacy, residency, consent, or physical-impact assessment

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

5. Normative Control System

Atomic criteria 16-18 of 18

MYTHOS-CLD-0007-C16**Version, Upgrade, and Deprecation Management**

The organization **MUST** define, implement, verify, and maintain version, upgrade, and deprecation management for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE supplier, carrier, firmware, license, support, portability, and exit evidence

MYTHOS-CLD-0007-C17**Incident Containment and Forensic Preservation**

The organization **MUST** define, implement, verify, and maintain incident containment and forensic preservation for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE approved service contract; architecture decision; owner and risk sign-off

MYTHOS-CLD-0007-C18**Fleet Evidence and Periodic Conformance**

The organization **MUST** define, implement, verify, and maintain fleet evidence and periodic conformance for in-scope Kubernetes and container orchestration capabilities. The control boundary **MUST** identify accountable ownership, affected services and assets, authoritative desired state, trust transitions, failure and recovery behavior, minimum evidence, inherited responsibility, and reassessment triggers.

MINIMUM EVIDENCE asset, service, dependency, supplier, region, and lifecycle inventory

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

6. Evidence and Assessment

Examine, interview, test, observe, restore, and trace

EXAMINE

Policies, inventories, diagrams, configuration, code, firmware, contracts, provider evidence, logs, exceptions, and lifecycle records.

INTERVIEW

Accountable owners, architects, engineers, operators, safety personnel, security teams, vendors, carriers, integrators, and responders.

TEST

Identity, segmentation, policy, negative paths, malformed input, capacity stress, dependency loss, failover, rollback, and revocation.

RESTORE

Independent restoration of configuration, data, service, device, controller, cluster, media, or browser state from protected evidence.

OBSERVE

Production-like performance, quality, drift, resource use, physical behavior, alarms, user impact, and incident execution.

TRACE

End-to-end linkage from approved intent and identity through change, runtime behavior, telemetry, outcome, exception, and review.

Evidence grades

A

Independently reproducible, complete, current, integrity-protected, and representative.

B

Reproduced by the implementing organization with strong traceability and controlled scope.

C

Partially reproduced or indirect; limitations, inherited responsibility, and uncertainty recorded.

D

Assertion, diagram, design intent, certificate, or vendor statement without reproduced behavior.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

7. Assurance Views

Six perspectives required for a complete assurance claim

GOVERNANCE

Ownership, purpose, risk tolerance, policy, exception, and decision rights.

ARCHITECTURE

Boundaries, dependencies, failure modes, state, data flow, and portability.

SECURITY

Identity, authorization, isolation, secrets, abuse resistance, and incident response.

OPERATIONS

Reliability, performance, cost, observability, change, recovery, and support.

PRIVACY / RIGHTS

Purpose limitation, minimization, consent, access, correction, deletion, and egress.

HUMAN / SOCIETAL

Usability, accessibility, disclosure, contestability, impact, and human control.

Conformance requires a defensible position across every applicable view. A strong aggregate score cannot compensate for an unowned system, a failed mandatory gate, untested recovery, or evidence below the accepted grade.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

7.2 Evaluation Metrics and Decision Gates

Measures must expose service behavior, control effectiveness, failure, uncertainty, and cost

SERVICE

Availability, correctness, coverage, quality, latency, throughput, and user or process outcome.

CONTROL

Unauthorized attempt, policy denial, drift, segmentation escape, privilege, and exception exposure.

RESILIENCE

Failover success, recovery time, data loss, safe degradation, restore validity, and dependency survival.

SECURITY

Exploitability, credential exposure, supply-chain integrity, attack detection, containment, and revocation.

CAPACITY

Utilization, saturation, queueing, radio or fabric headroom, thermal margin, quota, and forecast error.

OPERATIONS

Change failure, mean time to detect, repair, support burden, vendor escalation, and evidence completeness.

PRIVACY / SAFETY

Unauthorized egress, retention breach, consent coverage, unsafe action, physical consequence, and contestability.

LIFECYCLE

Patch debt, end-of-support exposure, portability, replacement time, retirement completion, and reassessment age.

Decision gate: thresholds **MUST** be declared before assessment, cover representative load and failure conditions, and identify mandatory safety, identity, recovery, and evidence failures that block promotion.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

8. Failure Modes and Adversarial Scenarios

Challenge assumptions before the system is trusted with consequential work

CONTROL-PLANE COMPROMISE

An administrative or orchestration plane can alter broad production scope.

REQUIRED: PREVENT | DETECT | CONTAIN | RECOVER | EVIDENCE

PROVIDER OR REGION DEPENDENCY

A provider, region, identity, DNS, or service dependency fails without a tested alternative.

REQUIRED: PREVENT | DETECT | CONTAIN | RECOVER | EVIDENCE

CONFIGURATION DRIFT

Runtime state diverges from approved desired state without timely detection.

REQUIRED: PREVENT | DETECT | CONTAIN | RECOVER | EVIDENCE

SOVEREIGNTY BREACH

Data or keys move through an unapproved jurisdiction or subprocess.

REQUIRED: PREVENT | DETECT | CONTAIN | RECOVER | EVIDENCE

COST AND QUOTA RUNAWAY

Unbounded scaling, abuse, or configuration error exhausts budget or service limits.

REQUIRED: PREVENT | DETECT | CONTAIN | RECOVER | EVIDENCE

EXIT FAILURE

Workloads, data, identity, or evidence cannot be migrated within the required time.

REQUIRED: PREVENT | DETECT | CONTAIN | RECOVER | EVIDENCE

Adversarial testing MUST protect people and production systems. Use isolated environments, synthetic or minimized data, bounded authority, kill switches, explicit legal and ethical review, and documented stop conditions.

9. Implementation Profiles

Risk-proportional implementation without weakening mandatory boundaries

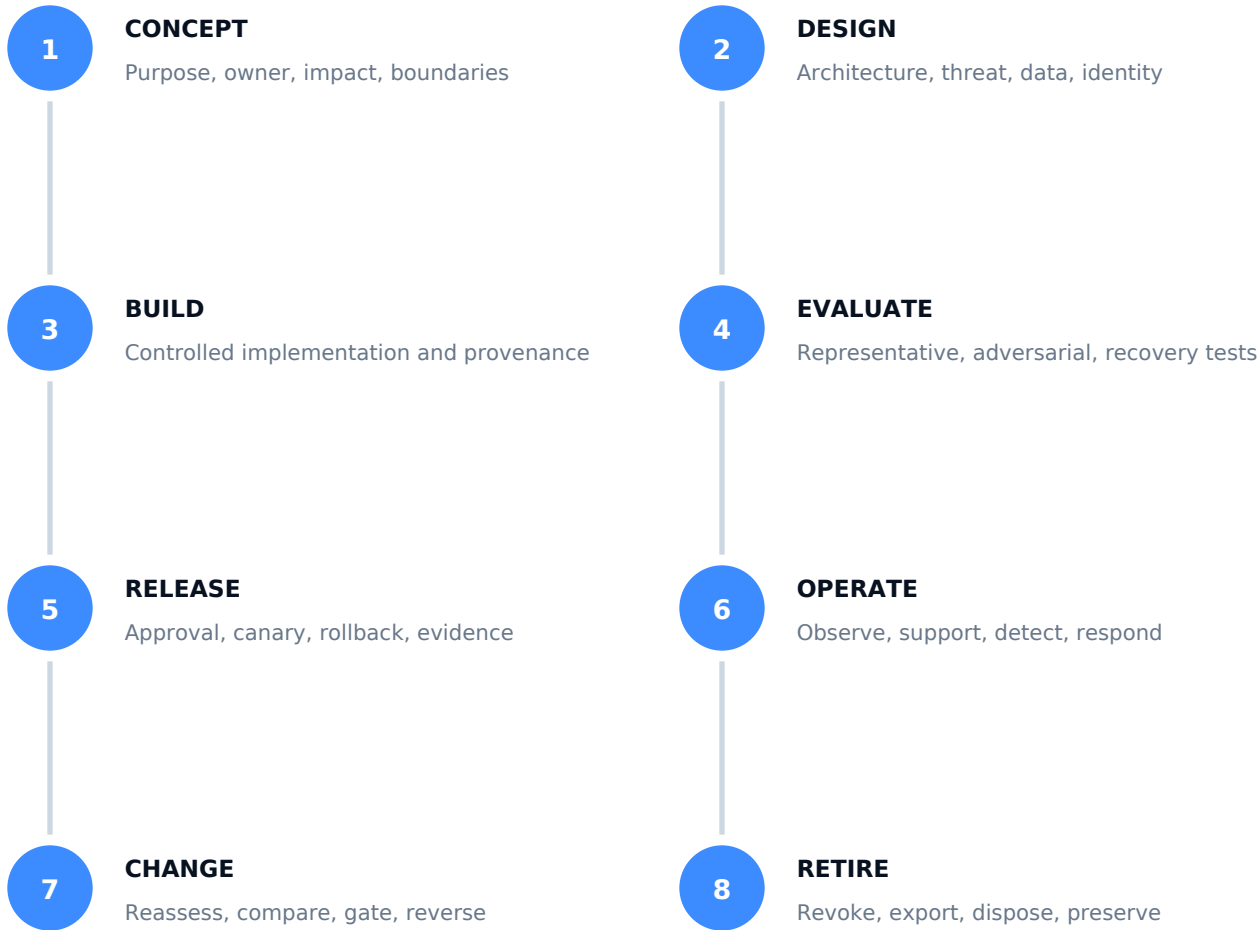
FOUNDATIONAL	ADVANCED	HIGH ASSURANCE
TYPICAL SCOPE Low consequence, limited autonomy, non-sensitive data, reversible outputs.	TYPICAL SCOPE Business-critical workflow, sensitive data, multiple tools or providers, moderate autonomy.	TYPICAL SCOPE Safety, security, regulated, financial, identity, or broadly consequential use.
MINIMUM DEPTH Named owner; inventory; basic evaluation; access control; logging; rollback; annual review.	MINIMUM DEPTH Formal threat model; representative evaluation; approval gates; isolated execution; tested recovery; quarterly review.	MINIMUM DEPTH Independent challenge; strongest identity; deterministic enforcement; comprehensive evidence; canary release; continuous monitoring.

Profile selection rule

Choose the highest profile triggered by consequence, autonomy, sensitivity, scale, irreversibility, external dependency, or inability to rapidly detect and recover. Tailoring may strengthen controls; it may not erase a mandatory gate without a controlled exception.

9.2 Operational Lifecycle

Evidence-bearing operation from concept through retirement



LIFECYCLE INVARIANT

Every stage **MUST** leave sufficient evidence for the next authority to understand what was intended, what changed, what was tested, what failed, what was accepted, what remains uncertain, and how to recover or exit.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

10. Adoption Roadmap

A sequenced path from uncontrolled capability to evidence-bearing operation

0-30 DAYS**Establish ownership and truth**

Inventory systems and dependencies; classify risk; freeze unsupported claims; identify immediate privilege, data, and evidence gaps.

31-90 DAYS**Create enforceable boundaries**

Define policy; isolate execution; implement identity and approval gates; establish representative evaluation and baseline telemetry.

3-6 MONTHS**Prove recovery and operating control**

Exercise failure, rollback, revocation, incident, provider exit, and state-recovery scenarios; mature evidence packaging.

6-12 MONTHS**Scale assurance**

Automate control checks; expand workload coverage; independent challenge; portfolio metrics; controlled exception expiry; continuous reassessment.

Adoption is complete only when operating evidence demonstrates the selected profile in the actual deployment context. Documentation alone is not conformance.

10.2 Conformance and Exception Statement

What an assurance claim must contain

SYSTEM / SERVICE Named, versioned capability and deployment boundary.	PROFILE Foundational, Advanced, or High Assurance with trigger rationale.
SCOPE Workloads, users, data, tools, regions, providers, and exclusions.	CRITERIA Pass, partial, fail, not applicable, and exception status for every criterion.
EVIDENCE Artifact references, evidence grade, date, environment, and reproducibility.	LIMITATIONS Known failure modes, unsupported workloads, uncertainty, and residual risk.
EXCEPTIONS Owner, rationale, compensating control, expiration, and approval.	VALIDITY Assessment date, change boundary, review date, and reassessment triggers.

Prohibited claim: “compliant” without the named scope, profile, evidence date, limitations, exceptions, and authority accepting residual risk.

MYTHOS-CLD-0007 / CLOUD ARCHITECTURE

11. Source Authority and Reference Register

Primary sources informing interpretation; current obligations and provider behavior must be verified at assessment time

NIST CSF 2.0

Cybersecurity Framework 2.0

<https://doi.org/10.6028/NIST.CSWP.29>

Enterprise governance and cybersecurity outcome framework.

NIST SP 800-53 Rev. 5

Security and Privacy Controls for Information Systems and Organizations

<https://doi.org/10.6028/NIST.SP.800-53r5>

Broad control baseline for organizational systems and services.

CSA CCM v4

Cloud Controls Matrix

<https://cloudsecurityalliance.org/research/cloud-controls-matrix>

Cloud control domains and shared-responsibility assessment structure.

FinOps Framework

FinOps Framework

<https://www.finops.org/framework/>

Cloud value, cost, usage, ownership, and operating-practice guidance.

NIST SP 800-204A

Building Secure Microservices-based Applications Using Service-Mesh Architecture

<https://doi.org/10.6028/NIST.SP.800-204A>

Microservice and service-mesh security architecture.

CNCF Cloud Native Security

Cloud Native Security Whitepaper

<https://github.com/cncf/tag-security/tree/main/security-whitepaper>

Cloud-native lifecycle, supply-chain, runtime, and operations guidance.

Source currency rule: authorities, specifications, legal obligations, provider services, firmware, browser behavior, carrier requirements, and operational evidence MUST be checked at assessment time. This publication records a 2026-07-24 source snapshot.

11.2 Revision History and Decision Register

Permanent identity, controlled change, and publication integrity

VERSION	DATE	STATE	SUMMARY
1.0.0-candidate	2026-07-24	Candidate	Permanent-publication edition; source refresh; expanded criteria, evidence, assurance, and lifecycle guidance.
Next review	2027-01-24	Scheduled	Review source currency, threat evolution, operating evidence, adoption feedback, and proposed revisions.

Publication decisions

- PERMANENT IDENTITY** The document ID remains stable across revisions; batch or production sequence metadata is not public identity.
- CHANGE CONTROL** Normative changes require rationale, impact analysis, affected-criterion mapping, and approval.
- SUPERSESSION** A superseded edition remains traceable; current routes and catalogs identify the active edition.
- CORRECTION** Material errors require a recorded correction, affected-evidence analysis, and notification appropriate to impact.
- ARCHIVAL INTEGRITY** Published artifacts receive hashes, metadata, and a release manifest sufficient for independent verification.

END OF PERMANENT PUBLICATION / MYTHOS-CLD-0007
MYTHOS SYSTEMS / ENGINEERING STANDARDS LIBRARY