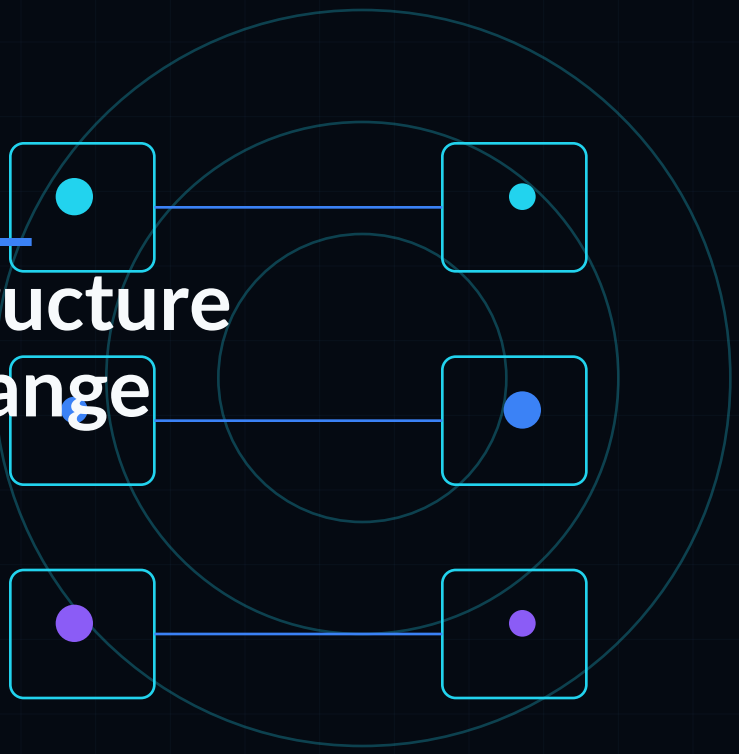
REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Digital Twin, Infrastructure Automation, and Change Assurance



A digital-twin and automation architecture that combines authoritative state, simulation, policy, controlled execution, telemetry, reconciliation, and evidence across infrastructure domains.

PERMANENT DOCUMENT ID
RA-011

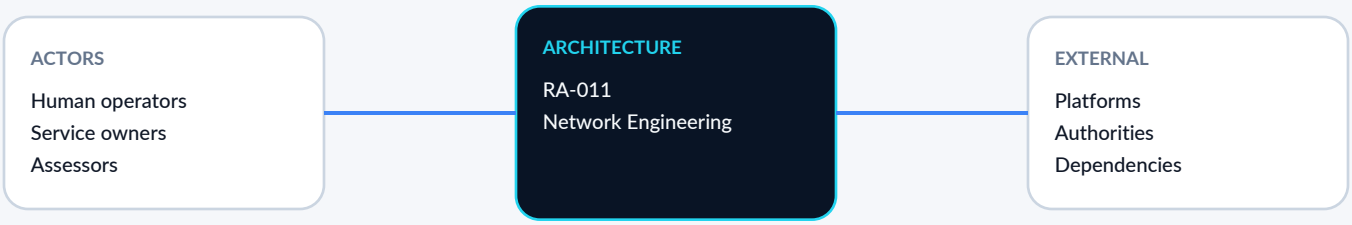
PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Network Engineering	6	6	18	0

SYSTEM CONTEXT



EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for digital twin, infrastructure automation, and
2. change assurance.
3. Make ownership, trust boundaries, data and control flows, failure handling, and evidence explicit.
4. Support multiple implementation profiles without weakening mandatory assurance outcomes.

ARCHITECTURE DOCTRINE

Interpret the architecture as a bounded operating model: explicit ownership, least authority, separable desired/runtime/observed/evidence state, safe degradation, reversible change, measurable outcomes, and evidence-backed acceptance.

Layered architecture and component model

LAYER 01
Authoritative infrastructure graph

LAYER 02
Digital twin and simulation

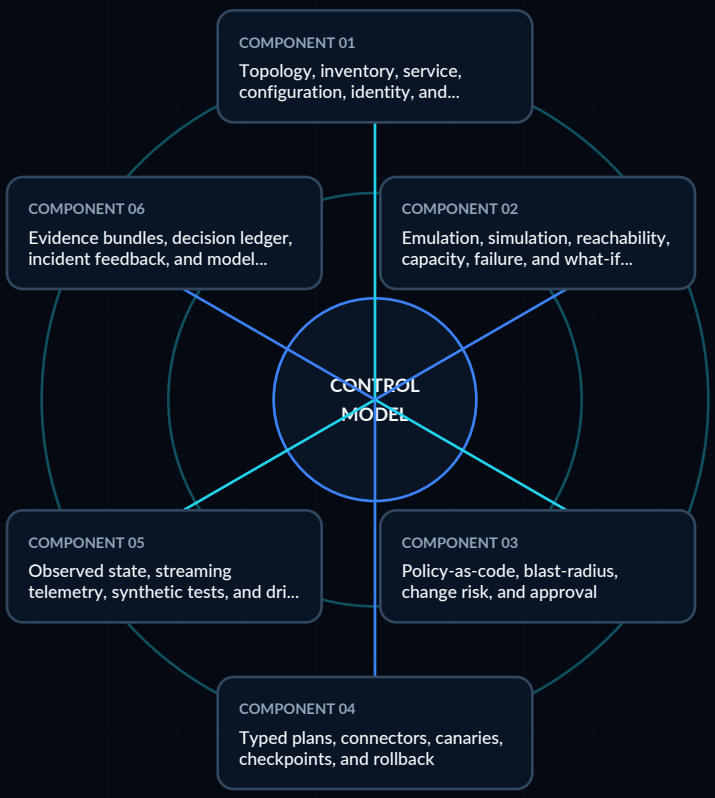
LAYER 03
Policy and risk analysis

LAYER 04
Plan and execution engine

LAYER 05
Telemetry and reconciliation

LAYER 06
Evidence, learning, and rollback

CORE COMPONENTS



Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

Primary sequence and control flow

EVIDENCE THREAD Identity + authorization + version + state transition + output + validation + error + timing + accountable disposition.

State, data, authority, and trust flow

TRUST BOUNDARY REGISTER

TB-01
Human or machine actor versus authorization service

TB-02
Control plane versus workload, device, or enforcement point

TB-03
Tenant or project versus shared platform

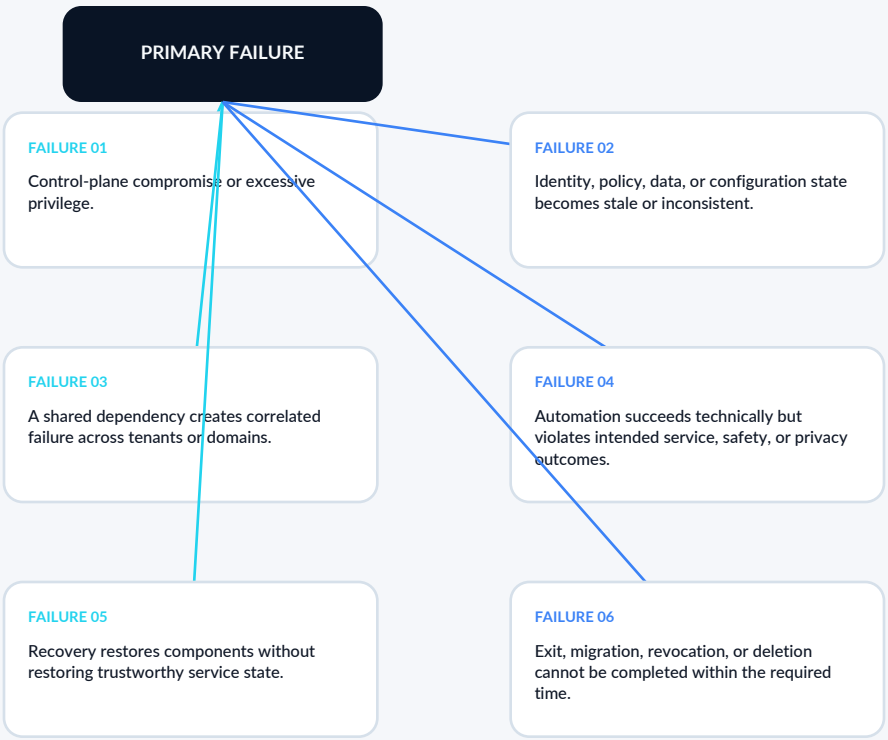
TB-04
Local environment versus external provider or network

TB-05
Mutable runtime state versus authoritative evidence

TB-06
Normal operation versus emergency or break-glass authority

Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK

OUTCOME
User / mission result

SERVICE
SLOs / availability

CONTROL
Policy / authorization

EXECUTION
State / errors / retries

DEPENDENCY
External health / latency

EVIDENCE
Trace / artifact / receipt

RECOVERY RULE Closure requires containment, causal explanation, restored service, reconciled intended vs observed state, preserved evidence, and an explicit residual-risk decision.

Deployment profiles and implementation roadmap

DEPLOYMENT PROFILES

PROFILE 01	PILOT OR LABORATORY PROFILE	Bounded proof
PROFILE 02	DEPARTMENT OR BOUNDED PRODUCTION PROFILE	Team-scale governance
PROFILE 03	ENTERPRISE FEDERATED PROFILE	Sovereign / high-assurance
PROFILE 04	HIGH-ASSURANCE SOVEREIGN OR REGULATED PROFILE	Distributed enterprise

IMPLEMENTATION ROADMAP

0-30 DAYS	31-90 DAYS	3-6 MONTHS	6-12 MONTHS	ONGOING
Scope / baseline	Pilot / contracts	Scale / integrate	Assure / automate	Measure / revalidate

Progression is evidence-gated. Architecture adoption does not advance because a component is installed or demonstrated; each stage requires measurable service, safety, security, recovery, and evidence outcomes.

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

AC-01	The architecture has a named service owner, technical owner, security owner, data owner, and recovery authority.
AC-02	The system boundary, external dependencies, retained responsibilities, and trust transitions are documented and
AC-03	Identity, authorization, secrets, policy, and privileged operations are explicit and independently auditable.
AC-04	Desired state, runtime state, observed state, and evidence state are distinguishable and reconciled.
AC-05	Consequential changes require preconditions, approval policy, bounded execution, verification, and rollback.
AC-06	Telemetry supports service health, security detection, forensic reconstruction, and conformance evidence.
AC-07	Failure modes include safe degradation, dependency loss, corrupted state, operator error, and malicious action.
AC-08	Backup, restoration, reconstitution, and exit procedures are exercised against defined recovery objectives.

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-NET-0005, MYTHOS-NET-0006, MYTHOS-DAT-0002, MYTHOS-SRE-0002

DETAILED SPECIFICATION READING MAP

The following pages preserve the detailed candidate architecture specification while this visual dossier adds explicit architecture diagrams, sequence views, state and evidence flows, observability, failure propagation, and implementation gates.

PART I - EXECUTIVE DIRECTION

OUTCOMES, RISKS, AND DECISIONS

ARCHITECTURE MANDATE

A digital-twin and automation architecture that combines authoritative state, simulation, policy, controlled execution, telemetry, reconciliation, and evidence across infrastructure domains.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for digital twin, infrastructure automation, and change assurance.
- Make ownership, trust boundaries, data and control flows, failure handling, and evidence explicit.
- Support multiple implementation profiles without weakening mandatory assurance outcomes.
- Provide a governed adoption path from discovery through production operation and retirement.

DECISIONS REQUIRED

- Accountable service and risk ownership.
- Accepted architecture profile and consequence class.
- Authoritative identity, data, policy, configuration, and evidence systems.
- Mandatory approval, recovery, privacy, safety, and supplier-exit gates.
- Funding for operations, observability, validation, and lifecycle—not only implementation.

PART II - SCOPE, PRINCIPLES, AND SYSTEM CONTEXT

NETWORK ENGINEERING

IN SCOPE

A digital-twin and automation architecture that combines authoritative state, simulation, policy, controlled execution, telemetry, reconciliation, and evidence across infrastructure domains.

OUT OF SCOPE

Vendor selection, exact sizing, contractual obligations, detailed low-level design, product certification, regulatory approval, and proof that any named platform is implemented.

ARCHITECTURE INVARIANTS

Explicit ownership; least authority; separable desired, runtime, observed, and evidence state; bounded automation; tested recovery; product-neutral interfaces; source and claim currency.

ASSUMPTIONS

Organizations tailor the pattern through documented risk, environment, criticality, data, safety, regulatory, workforce, and supplier analysis.

SYSTEM CONTEXT AND RESPONSIBILITY MODEL

ACTORS, ENVIRONMENTS, AND EXTERNAL DEPENDENCIES

ACTORS AND RESPONSIBILITY

- Service consumers and affected users
- Product, platform, network, security, data, reliability, and support teams
- Human administrators, operators, reviewers, incident commanders, and risk owners
- Machine identities, agents, workloads, devices, controllers, and external services
- Suppliers, providers, carriers, integrators, assessors, and regulators where applicable

RESPONSIBILITY BOUNDARY

- The adopting organization retains accountability for purpose, risk, configuration, data, identity, operation, recovery, and evidence even where a provider performs underlying functions.
- Inherited controls are recorded with provider, scope, evidence, limitations, shared responsibility, incident path, and exit conditions.
- No component is considered trusted solely because it is internal, managed, signed, cloud-hosted, model-generated, or supplied by a recognized vendor.

ARCHITECTURE POSITION

This future-domain candidate defines a stable architectural destination and assurance model. Product choices, scale, regulatory obligations, and implementation details remain environment-specific.

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01

Authoritative infrastructure graph

Authoritative infrastructure graph owns a bounded set of responsibilities, interfaces, state, controls, failure behavior, telemetry, recovery procedures, and evidence. It does not silently absorb authority assigned to another layer.

02

Digital twin and simulation

Digital twin and simulation owns a bounded set of responsibilities, interfaces, state, controls, failure behavior, telemetry, recovery procedures, and evidence. It does not silently absorb authority assigned to another layer.

03

Policy and risk analysis

Policy and risk analysis owns a bounded set of responsibilities, interfaces, state, controls, failure behavior, telemetry, recovery procedures, and evidence. It does not silently absorb authority assigned to another layer.

04

Plan and execution engine

Plan and execution engine owns a bounded set of responsibilities, interfaces, state, controls, failure behavior, telemetry, recovery procedures, and evidence. It does not silently absorb authority assigned to another layer.

05

Telemetry and reconciliation

Telemetry and reconciliation owns a bounded set of responsibilities, interfaces, state, controls, failure behavior, telemetry, recovery procedures, and evidence. It does not silently absorb authority assigned to another layer.

06

Evidence, learning, and rollback

Evidence, learning, and rollback owns a bounded set of responsibilities, interfaces, state, controls, failure behavior, telemetry, recovery procedures, and evidence. It does not silently absorb authority assigned to another layer.

CORE COMPONENTS - A

COMPONENTS 01-03

COMPONENT 01

Topology, inventory, service, configuration, identity, and dependency graph

Required contract: owner, purpose, inputs, outputs, authority, data, dependencies, health, failure behavior, recovery, lifecycle, and evidence.

COMPONENT 02

Emulation, simulation, reachability, capacity, failure, and what-if analysis

Required contract: owner, purpose, inputs, outputs, authority, data, dependencies, health, failure behavior, recovery, lifecycle, and evidence.

COMPONENT 03

Policy-as-code, blast-radius, change risk, and approval

Required contract: owner, purpose, inputs, outputs, authority, data, dependencies, health, failure behavior, recovery, lifecycle, and evidence.

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Typed plans, connectors, canaries, checkpoints, and rollback

Required contract: owner, purpose, inputs, outputs, authority, data, dependencies, health, failure behavior, recovery, lifecycle, and evidence.

COMPONENT 05

Observed state, streaming telemetry, synthetic tests, and drift classification

Required contract: owner, purpose, inputs, outputs, authority, data, dependencies, health, failure behavior, recovery, lifecycle, and evidence.

COMPONENT 06

Evidence bundles, decision ledger, incident feedback, and model correction

Required contract: owner, purpose, inputs, outputs, authority, data, dependencies, health, failure behavior, recovery, lifecycle, and evidence.

PART IV - DATA, CONTROL, AUTHORITY, AND EVIDENCE FLOW

FLOW CONTRACTS

1. Intent, service demand, policy, and environment constraints enter the architecture through explicit contracts.

   Evidence: request and actor identity, policy decision, versions, state transitions, outputs, tests, errors, timing, and accountable disposition.

2. Authoritative identity, data, configuration, and dependency state are validated before execution or access.

   Evidence: request and actor identity, policy decision, versions, state transitions, outputs, tests, errors, timing, and accountable disposition.

3. Control planes perform bounded orchestration while local enforcement preserves least privilege and safe behavior.

   Evidence: request and actor identity, policy decision, versions, state transitions, outputs, tests, errors, timing, and accountable disposition.

4. Telemetry, tests, user outcomes, and incidents update evidence and trigger correction or rollback.

   Evidence: request and actor identity, policy decision, versions, state transitions, outputs, tests, errors, timing, and accountable disposition.

5. Lifecycle governance manages versioning, capacity, exceptions, migration, and retirement.

   Evidence: request and actor identity, policy decision, versions, state transitions, outputs, tests, errors, timing, and accountable disposition.

TRUST BOUNDARIES AND ENFORCEMENT

EVERY CROSSING REQUIRES AN EXPLICIT CONTRACT

TRUST BOUNDARY 01

Human or machine actor versus authorization service

Minimum contract: authenticated parties, authorized actions, data classification, protocol, encryption, validation, rate and resource limits, observability, failure behavior, revocation, and evidence.

TRUST BOUNDARY 02

Control plane versus workload, device, or enforcement point

Minimum contract: authenticated parties, authorized actions, data classification, protocol, encryption, validation, rate and resource limits, observability, failure behavior, revocation, and evidence.

TRUST BOUNDARY 03

Tenant or project versus shared platform

Minimum contract: authenticated parties, authorized actions, data classification, protocol, encryption, validation, rate and resource limits, observability, failure behavior, revocation, and evidence.

TRUST BOUNDARY 04

Local environment versus external provider or network

Minimum contract: authenticated parties, authorized actions, data classification, protocol, encryption, validation, rate and resource limits, observability, failure behavior, revocation, and evidence.

TRUST BOUNDARY 05

Mutable runtime state versus authoritative evidence

Minimum contract: authenticated parties, authorized actions, data classification, protocol, encryption, validation, rate and resource limits, observability, failure behavior, revocation, and evidence.

TRUST BOUNDARY 06

Normal operation versus emergency or break-glass authority

Minimum contract: authenticated parties, authorized actions, data classification, protocol, encryption, validation, rate and resource limits, observability, failure behavior, revocation, and evidence.

PART V - DEPLOYMENT PROFILES

PROFILE CHANGES SCALE, NOT ASSURANCE OUTCOMES

PROFILE 01 / PILOT OR LABORATORY PROFILE

Use this profile to establish a bounded implementation with explicit scale, criticality, users, data, autonomy, external dependencies, recovery objectives, staffing, and validation depth. The profile may simplify technology or topology but must preserve identity, least authority, evidence, safe failure, recovery, lifecycle, and accountable risk acceptance.

PROFILE 02 / DEPARTMENT OR BOUNDED PRODUCTION PROFILE

Use this profile to establish a bounded implementation with explicit scale, criticality, users, data, autonomy, external dependencies, recovery objectives, staffing, and validation depth. The profile may simplify technology or topology but must preserve identity, least authority, evidence, safe failure, recovery, lifecycle, and accountable risk acceptance.

PROFILE 03 / ENTERPRISE FEDERATED PROFILE

Use this profile to establish a bounded implementation with explicit scale, criticality, users, data, autonomy, external dependencies, recovery objectives, staffing, and validation depth. The profile may simplify technology or topology but must preserve identity, least authority, evidence, safe failure, recovery, lifecycle, and accountable risk acceptance.

PROFILE 04 / HIGH-ASSURANCE SOVEREIGN OR REGULATED PROFILE

Use this profile to establish a bounded implementation with explicit scale, criticality, users, data, autonomy, external dependencies, recovery objectives, staffing, and validation depth. The profile may simplify technology or topology but must preserve identity, least authority, evidence, safe failure, recovery, lifecycle, and accountable risk acceptance.

PART VI - OPERATING LIFECYCLE

FROM DISCOVERY THROUGH RETIREMENT

01 / Discover and classify demand

Entry and exit criteria, owner, evidence, exceptions, dependencies, rollback, and next decision are recorded. No phase is complete solely because components were purchased, configured, or demonstrated.

02 / Define service boundary and owners

Entry and exit criteria, owner, evidence, exceptions, dependencies, rollback, and next decision are recorded. No phase is complete solely because components were purchased, configured, or demonstrated.

03 / Model architecture and acceptance criteria

Entry and exit criteria, owner, evidence, exceptions, dependencies, rollback, and next decision are recorded. No phase is complete solely because components were purchased, configured, or demonstrated.

04 / Build isolated proof and challenge assumptions

Entry and exit criteria, owner, evidence, exceptions, dependencies, rollback, and next decision are recorded. No phase is complete solely because components were purchased, configured, or demonstrated.

05 / Pilot with stop conditions and rollback

Entry and exit criteria, owner, evidence, exceptions, dependencies, rollback, and next decision are recorded. No phase is complete solely because components were purchased, configured, or demonstrated.

06 / Scale through standard profiles and automation

Entry and exit criteria, owner, evidence, exceptions, dependencies, rollback, and next decision are recorded. No phase is complete solely because components were purchased, configured, or demonstrated.

07 / Operate, reassess, migrate, and retire

Entry and exit criteria, owner, evidence, exceptions, dependencies, rollback, and next decision are recorded. No phase is complete solely because components were purchased, configured, or demonstrated.

SECURITY, PRIVACY, SAFETY, AND RIGHTS

CROSS-CUTTING ASSURANCE

SECURITY

Threat model identities, trust boundaries, supply chain, control planes, data, protocols, privilege, abuse, detection, response, recovery, and residual risk.

PRIVACY

Purpose limitation, minimization, transparency, consent where required, access, correction, retention, deletion, sovereignty, metadata, and third-party processing.

SAFETY

Hazards, unsafe states, human intervention, physical consequence, dependency failure, automation limits, emergency stop, recovery sequence, and validation.

RIGHTS AND USABILITY

Accessibility, contestability, explanation, operator workload, affected-user communication, grievance, misuse prevention, and equitable performance.

SUPPLY CHAIN

Provenance, version, SBOM or component inventory, signature, vulnerability, support, update, provider, subcontractor, and exit.

EVIDENCE

Design decisions, tests, telemetry, incidents, approvals, exceptions, recovery exercises, source currency, and review findings remain linked to the architecture version.

FAILURE MODES AND RESILIENCE

DESIGN FOR SAFE DEGRADATION AND TRUSTWORTHY RECOVERY

FAILURE SCENARIO 01

Control-plane compromise or excessive privilege.

Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 02

Identity, policy, data, or configuration state becomes stale or inconsistent.

Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 03

A shared dependency creates correlated failure across tenants or domains.

Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 04

Automation succeeds technically but violates intended service, safety, or privacy outcomes.

Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 05

Recovery restores components without restoring trustworthy service state.

Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 06

Exit, migration, revocation, or deletion cannot be completed within the required time.

Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

PART VII - INTEGRATION AND DEPENDENCY MODEL

CONNECTED REFERENCE ARCHITECTURES AND STANDARDS

REFERENCE ARCHITECTURE DEPENDENCIES

- RA-012 - Enterprise Observability, SRE, and Incident Command Platform
- RA-015 - Edge, OT, IoT, and Telecommunications Convergence
- RA-018 - Physical AI and Autonomous Fleet Operations

MAPPED MYTHOS STANDARDS

- MYTHOS-NET-0005
- MYTHOS-NET-0006
- MYTHOS-DAT-0002
- MYTHOS-SRE-0002

DEPENDENCY DOCTRINE

A dependency is not a blanket trust grant. Each connection requires an interface, identity, policy, data, availability, version, failure, recovery, and evidence contract. Correlated failure and circular authority must be analyzed across the full architecture graph.

PART VIII - IMPLEMENTATION AND ADOPTION ROADMAP

PHASED, EVIDENCE-DRIVEN TRANSITION

0-30 DAYS	Inventory purpose, owners, users, systems, data, dependencies, risks, current diagrams, incidents, recovery, and evidence gaps.
31-90 DAYS	Establish authoritative identity, configuration, policy, data, source, and architecture records; define target profile and acceptance tests.
3-6 MONTHS	Implement bounded foundations, isolation, telemetry, change control, backup, recovery, and representative pilot workflows.
6-12 MONTHS	Expand through standard patterns, automation, service ownership, SLOs, independent assessment, failure injection, and supplier-exit exercises.
ONGOING	Reconcile drift, review sources, reassess risk, measure outcomes, exercise recovery, renew exceptions, migrate dependencies, and retire obsolete paths.

ARCHITECTURE ACCEPTANCE CRITERIA - A

CRITERIA 01-09

AC-01	The architecture has a named service owner, technical owner, security owner, data owner, and recovery authority.
AC-02	The system boundary, external dependencies, retained responsibilities, and trust transitions are documented and reviewed.
AC-03	Identity, authorization, secrets, policy, and privileged operations are explicit and independently auditable.
AC-04	Desired state, runtime state, observed state, and evidence state are distinguishable and reconciled.
AC-05	Consequential changes require preconditions, approval policy, bounded execution, verification, and rollback.
AC-06	Telemetry supports service health, security detection, forensic reconstruction, and conformance evidence.
AC-07	Failure modes include safe degradation, dependency loss, corrupted state, operator error, and malicious action.
AC-08	Backup, restoration, reconstitution, and exit procedures are exercised against defined recovery objectives.
AC-09	Data classification, retention, privacy, sovereignty, and disposition controls follow the actual information flow.

ARCHITECTURE ACCEPTANCE CRITERIA - B

CRITERIA 10-18

AC-10	Supplier, model, protocol, platform, and software dependencies have provenance, version, support, and replacement records.
AC-11	Representative performance and capacity tests include saturation, contention, latency, and degraded conditions.
AC-12	Exceptions are time-bounded, visible, approved by accountable risk owners, and linked to compensating controls.
AC-13	Architecture diagrams and inventories remain synchronized with deployed components and interfaces.
AC-14	Mandatory trust, safety, privacy, recovery, and evidence gates cannot be bypassed by a lower assurance profile.
AC-15	External dependencies have contractual, technical, operational, data, and exit boundaries.
AC-16	Representative acceptance tests exercise normal, degraded, adversarial, recovery, and retirement scenarios.
AC-17	The architecture can explain why an action, decision, deployment, or access was permitted and what evidence supports it.
AC-18	A formal decision record names selected and rejected alternatives, residual risk, review date, and change triggers.

METRICS, EVIDENCE, AND DECISION GATES

MEASURE SERVICE OUTCOMES AND ASSURANCE

COVERAGE

Percent of components, interfaces, identities, data flows, dependencies, risks, and controls represented in current architecture evidence.

CHANGE QUALITY

Plan success, rollback, verification pass, escaped defect, drift, and emergency-change rates.

RELIABILITY

Availability, latency, saturation, dependency health, recovery time, recovery point, and safe-degradation success.

SECURITY

Privilege exposure, policy denial, identity compromise, vulnerable dependency, detection coverage, response, and evidence completeness.

HUMAN OUTCOME

Task success, operator workload, accessibility, override, contestability, training, and user-impact measures.

LIFECYCLE

Version currency, unsupported component, exception age, migration readiness, restore exercise, and retirement completeness.

Release or expansion gate: mandatory acceptance criteria pass; failed safety, privacy, identity, recovery, evidence, or legal gates block promotion regardless of aggregate score.

SOURCE AUTHORITY REGISTER

CURRENT PRIMARY AND OFFICIAL REFERENCES

SOURCE 01 **RFC 8345, YANG Data Model for Network Topologies**

https://www.rfc-editor.org/info/rfc8345

SOURCE 02 **RFC 8969, Framework for Automating Service and Network Management**

https://www.rfc-editor.org/info/rfc8969

SOURCE 03 **OpenConfig Vendor-Neutral Network Models**

https://www.openconfig.net/

SOURCE 04 **OpenTelemetry Specification**

https://opentelemetry.io/docs/specs/

Source currency note. Official frameworks, specifications, and implementation guidance evolve. Assessors must verify the current version, status, errata, applicability, and effective date at adoption and scheduled review. NIST AI RMF 1.0 was under revision at the publication date.

PUBLICATION CONTROL AND REVISION

PERMANENT IDENTITY / CONTROLLED EVOLUTION

PERMANENT DOCUMENT IDENTITY	RA-011
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Future Domain Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

REVISION RECORD

1.0.0-candidate (2026-07-24) - Permanent-publication candidate edition regenerated under the Mythos Systems architecture publication system. Production sequence metadata is excluded from public identity.