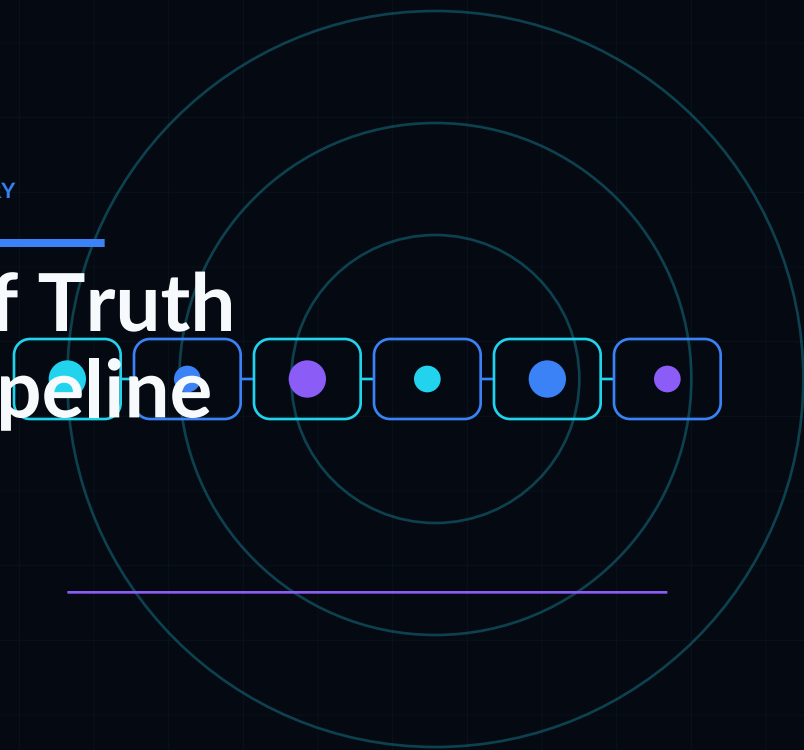




REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Network Source of Truth Implementation Pipeline



A closed-loop implementation pipeline that separates declarative intent, authoritative operational state, generated configuration, observed state, reconciliation, verification, and evidence.

PERMANENT DOCUMENT ID

RA-004

PUBLICATION

Candidate Reference Architecture

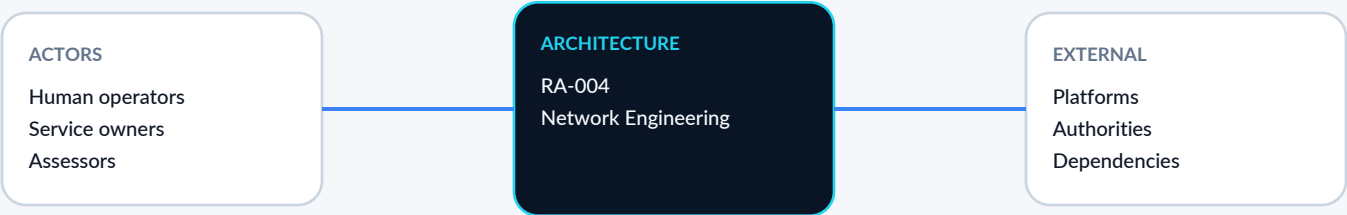
ARCHITECTURE SET

Canonical Set

Architecture identity and operating position

ARCHITECTURE SET Canonical Set	DOMAIN Network Engineering	LAYERS 6	COMPONENTS 6	ACCEPTANCE 18	DEPENDENCIES 0
--	-------------------------------	--------------------	------------------------	-------------------------	--------------------------

SYSTEM CONTEXT



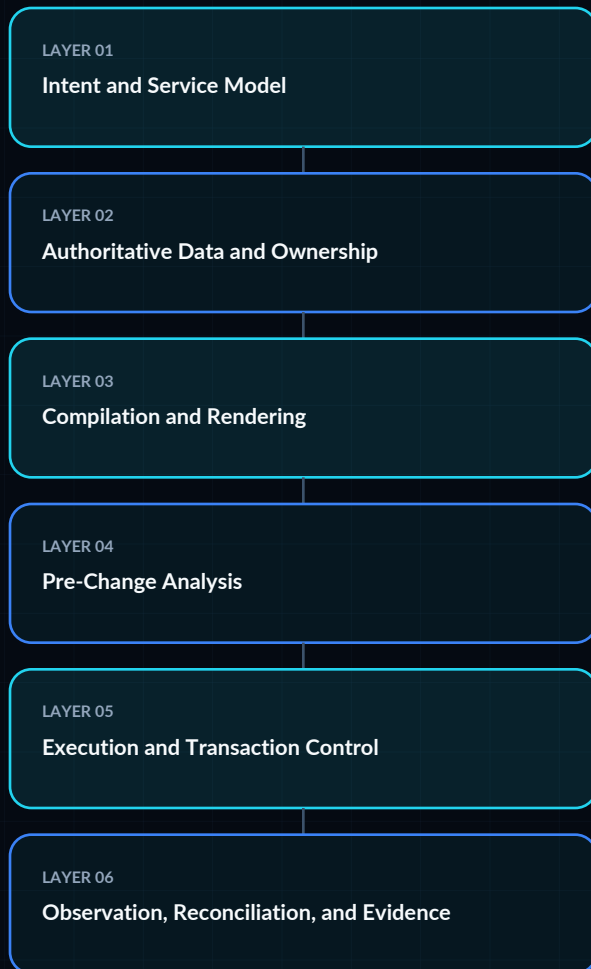
EXECUTIVE OUTCOMES

- 1 Represent service, topology, inventory, identity, addressing, policy, and lifecycle intent explicitly.
- 2 Generate platform-specific changes from validated models rather than hand-edited production state.
- 3 Verify proposed and deployed changes against safety, reachability, policy, and service outcomes.
- 4 Reconcile drift without erasing legitimate operational reality or emergency change.

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



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



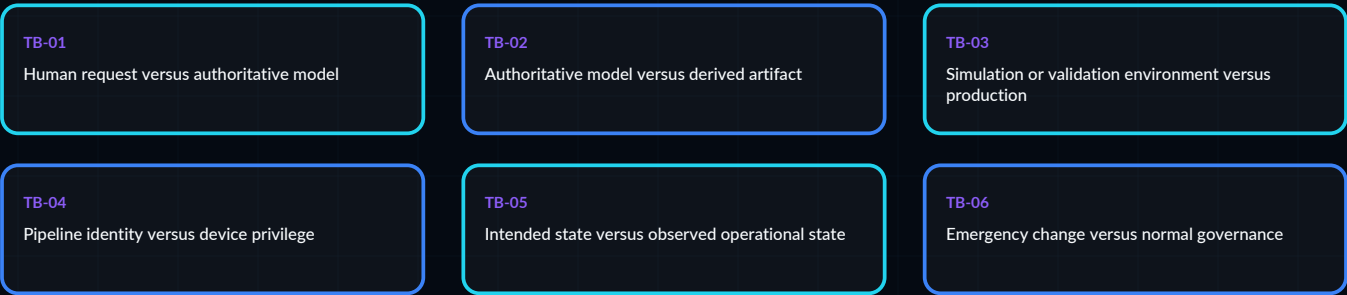
EVIDENCE THREAD

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

State, data, authority, and trust flow

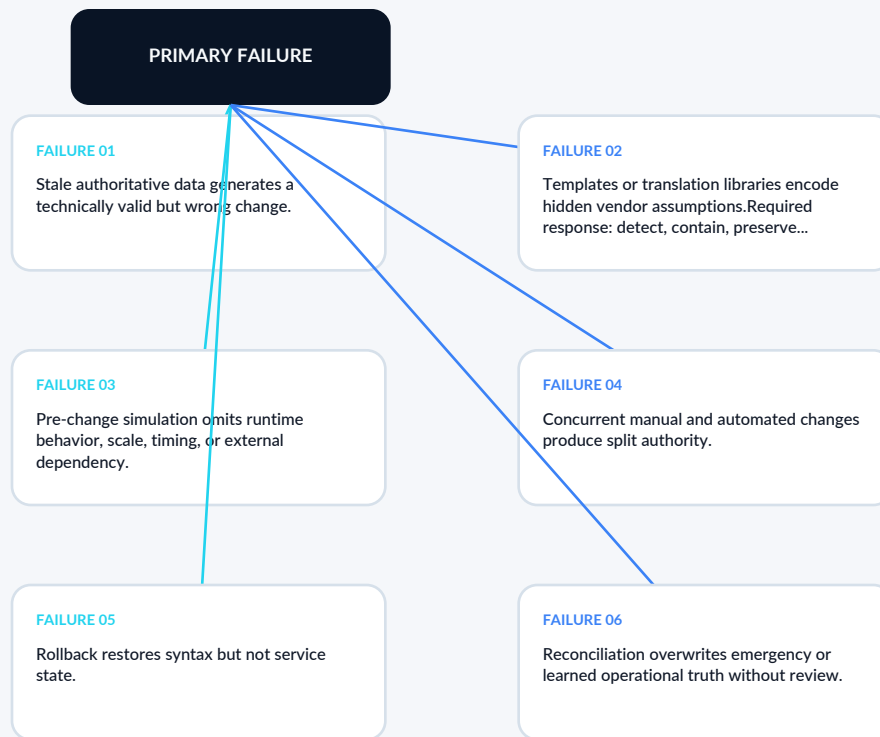


TRUST BOUNDARY REGISTER



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

READ-ONLY DISCOVERY AND INVENTORY

Bounded proof

PROFILE 02

GIT-BACKED SOURCE OF TRUTH WITH MANUAL DEPLOYMENT

Team-scale governance

PROFILE 03

CLOSED-LOOP NETWORK-AS-CODE PIPELINE

Sovereign / high-assurance

PROFILE 04

MULTI-DOMAIN ORCHESTRATOR WITH DIGITAL TWIN AND POLICY GATES

Distributed enterprise

IMPLEMENTATION ROADMAP

0-30 DAYS

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

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-NET-0008, MYTHOS-DAT-0001, MYTHOS-SWE-0004, MYTHOS-SRE-0002

DETAILED SPECIFICATION READING MAP

09	Executive direction	10	Scope and system context	12	Logical architecture
15	Flow contracts	16	Trust boundaries	17	Deployment profiles
20	Failure modes	21	Dependencies and standards	22	Implementation roadmap
23	Acceptance criteria	25	Metrics and decision gates	26	Source authority
27	Publication control				

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 closed-loop implementation pipeline that separates declarative intent, authoritative operational state, generated configuration, observed state, reconciliation, verification, and evidence.

EXECUTIVE OUTCOMES

- Represent service, topology, inventory, identity, addressing, policy, and lifecycle intent explicitly.
- Generate platform-specific changes from validated models rather than hand-edited production state.
- Verify proposed and deployed changes against safety, reachability, policy, and service outcomes.
- Reconcile drift without erasing legitimate operational reality or emergency change.

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 closed-loop implementation pipeline that separates declarative intent, authoritative operational state, generated configuration, observed state, reconciliation, verification, and evidence.

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

The source of truth is an authority model and lifecycle, not merely a database. No single store is presumed authoritative for every class of network state.

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01**Intent and Service Model**

Intent and Service Model 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**Authoritative Data and Ownership**

Authoritative Data and Ownership 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**Compilation and Rendering**

Compilation and Rendering 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**Pre-Change Analysis**

Pre-Change 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.

05**Execution and Transaction Control**

Execution and Transaction Control 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**Observation, Reconciliation, and Evidence**

Observation, Reconciliation, and Evidence 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

Service catalog, topology, inventory, IPAM, circuit, identity, and policy models

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

COMPONENT 02

Schema registry, ownership rules, validation, lineage, and lifecycle workflows

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

COMPONENT 03

Template, model, translation, and vendor-library compilation services

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Static analysis, simulation, topology validation, diff, policy tests, and risk scoring

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

COMPONENT 05

Approval, scheduling, locking, canary, deployment, checkpoint, and rollback engine

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

COMPONENT 06

Streaming telemetry, configuration retrieval, state comparison, incident feedback, and evidence storage

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

Requested service change becomes a reviewed intent change.

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

2

Models validate against schema, ownership, constraints, and dependency rules.

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

3

Compiler produces target-specific artifacts and a human-readable plan.

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

4

Analysis proves preconditions and expected effects within stated limits.

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

5

Executor performs bounded changes and captures platform responses.

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

6

Verification tests business and technical outcomes before commit.

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

7

Observed state reconciles into drift, learned reality, or approved model correction.

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 request versus authoritative model

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

TRUST BOUNDARY 02

Authoritative model versus derived artifact

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

TRUST BOUNDARY 03

Simulation or validation environment versus production

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

TRUST BOUNDARY 04

Pipeline identity versus device privilege

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

TRUST BOUNDARY 05

Intended state versus observed operational state

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

TRUST BOUNDARY 06

Emergency change versus normal governance

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 / READ-ONLY DISCOVERY AND INVENTORY

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 / GIT-BACKED SOURCE OF TRUTH WITH MANUAL DEPLOYMENT

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 / CLOSED-LOOP NETWORK-AS-CODE PIPELINE

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 / MULTI-DOMAIN ORCHESTRATOR WITH DIGITAL TWIN AND POLICY GATES

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 state authority

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 / Normalize models and ownership

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 / Establish read-only validation

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 / Generate and review plans

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 / Introduce bounded execution

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 / Add verification and reconciliation

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 / Expand domains and autonomous remediation cautiously

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

Stale authoritative data generates a technically valid but wrong change.
Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 02

Templates or translation libraries encode hidden vendor assumptions.
Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 03

Pre-change simulation omits runtime behavior, scale, timing, or external dependency.
Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 04

Concurrent manual and automated changes produce split authority.
Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 05

Rollback restores syntax but not service state.
Required response: detect, contain, preserve evidence, maintain or enter safe service, communicate, recover, verify, learn, and prevent recurrence.

FAILURE SCENARIO 06

Reconciliation overwrites emergency or learned operational truth without review.
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-011 - Digital Twin, Infrastructure Automation, and Change Assurance
- RA-014 - Enterprise Data, API, Event, and Integration Fabric

MAPPED MYTHOS STANDARDS

- MYTHOS-NET-0005
- MYTHOS-NET-0006
- MYTHOS-NET-0008
- MYTHOS-DAT-0001
- MYTHOS-SWE-0004
- 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

Every data class names its system of record, system of authority, owner, update path, validation rules, and conflict-resolution process.

AC-14

Generated artifacts are reproducible from versioned models, compiler versions, vendor libraries, and environment parameters.

AC-15

Plans show intended effects, untouched scope, dependencies, assumptions, uncertainty, and rollback conditions.

AC-16

Deployment uses least privilege, bounded scope, concurrency control, checkpoints, canaries, and explicit transaction semantics.

AC-17

Verification tests service outcomes and policy rather than accepting command success as proof.

AC-18

Reconciliation classifies differences as expected runtime state, tolerated drift, incident, emergency change, discovery, or model defect.

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 7950, YANG 1.1 Data Modeling Language

<https://www.rfc-editor.org/info/rfc7950>

SOURCE 02 RFC 8345, YANG Data Model for Network Topologies

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

SOURCE 03 RFC 8969, Framework for Automating Service and Network Management

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

SOURCE 04 OpenConfig Vendor-Neutral Network Models

<https://www.openconfig.net/>

SOURCE 05 NIST SP 800-53 Rev. 5

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

SOURCE 06 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-004
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Canonical 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.