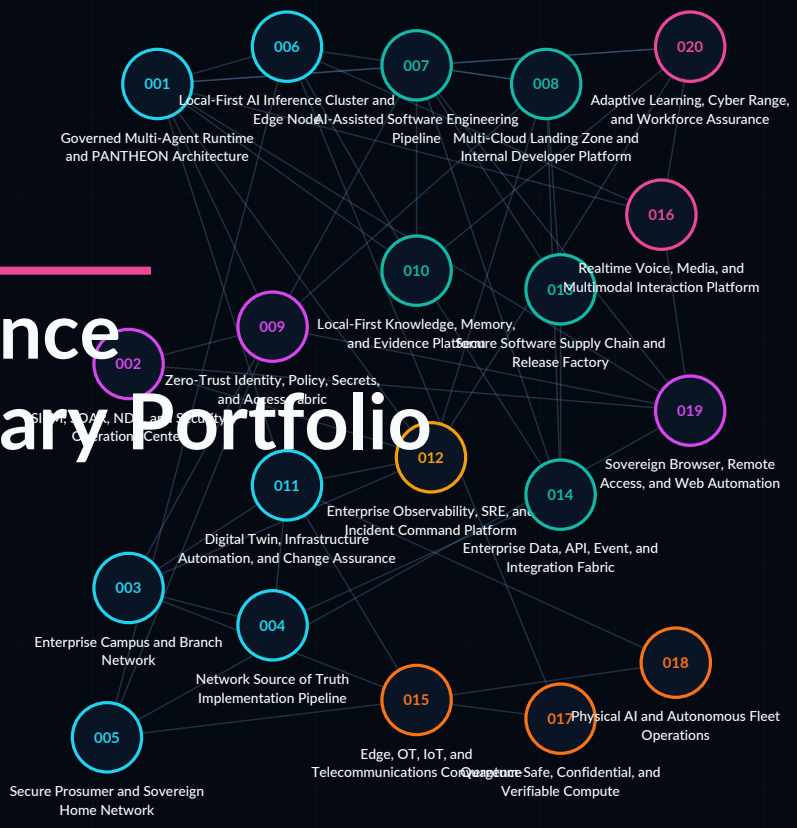
REFERENCE ARCHITECTURES / PORTFOLIO EDITION

Complete Reference Architecture Library Portfolio



20 permanent candidate reference architectures

PERMANENT DOCUMENT ID	ARCHITECTURES	STATUS
MYTHOS-RA-PORT-0001	20	Portfolio Edition

Portfolio register and page map

RA-001 3
Governed Multi-Agent Runtime and PANTHEON Architecture

RA-002 30
SIEM, SOAR, NDR, and Security Operations Center

RA-003 57
Enterprise Campus and Branch Network

RA-004 84
Network Source of Truth Implementation Pipeline

RA-005 111
Secure Prosumer and Sovereign Home Network

RA-006 138
Local-First AI Inference Cluster and Edge Node

RA-007 165
AI-Assisted Software Engineering Pipeline

RA-008 192
Multi-Cloud Landing Zone and Internal Developer Platform

RA-009 219
Zero-Trust Identity, Policy, Secrets, and Access Fabric

RA-010 246
Local-First Knowledge, Memory, and Evidence Platform

RA-011 273
Digital Twin, Infrastructure Automation, and Change Assurance

RA-012 300
Enterprise Observability, SRE, and Incident Command Platform

RA-013 327
Secure Software Supply Chain and Release Factory

RA-014 354
Enterprise Data, API, Event, and Integration Fabric

RA-015 381
Edge, OT, IoT, and Telecommunications Convergence

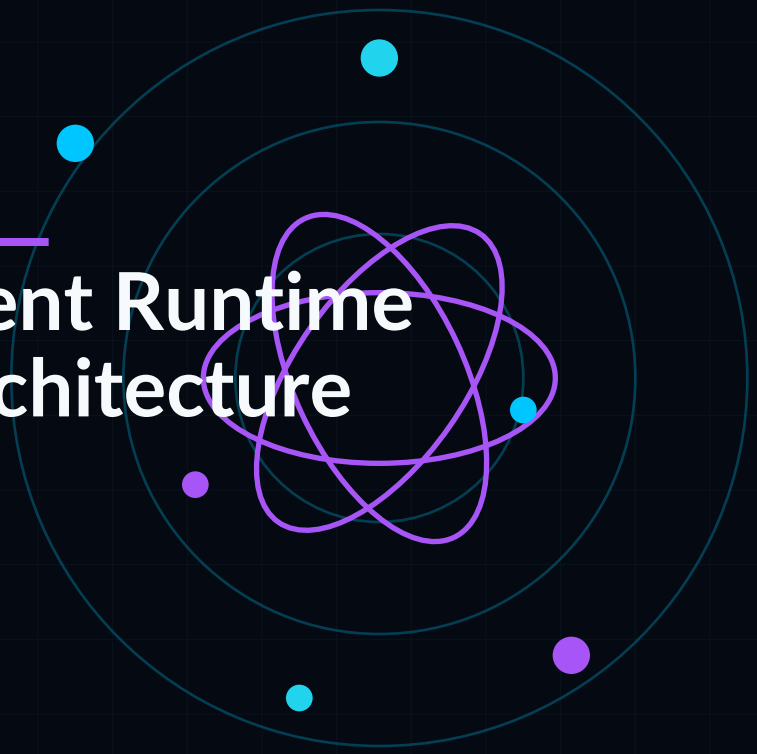
RA-016 408
Realtime Voice, Media, and Multimodal Interaction Platform

RA-017 435
Quantum-Safe, Confidential, and Verifiable Compute

RA-018 462
Physical AI and Autonomous Fleet Operations

RA-019 489
Sovereign Browser, Remote Access, and Web Automation

RA-020 516
Adaptive Learning, Cyber Range, and Workforce Assurance



REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Governed Multi-Agent Runtime and PANTHEON Architecture

A governed, durable multi-agent architecture that separates reasoning, authorization, deterministic execution, verification, memory, observability, and evidence.

PERMANENT DOCUMENT ID
RA-001

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Canonical Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Canonical Set	Artificial Intelligence	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Convert ambiguous human intent into explicit missions and typed plans.
2. Permit models and agents to propose actions without silently acquiring execution authority.
3. Execute consequential work through deterministic, policy-bound mechanisms.
4. Preserve evidence, checkpoints, approvals, memory boundaries, and verified 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
Experience and Intent

LAYER 02
Mission and Plan Compiler

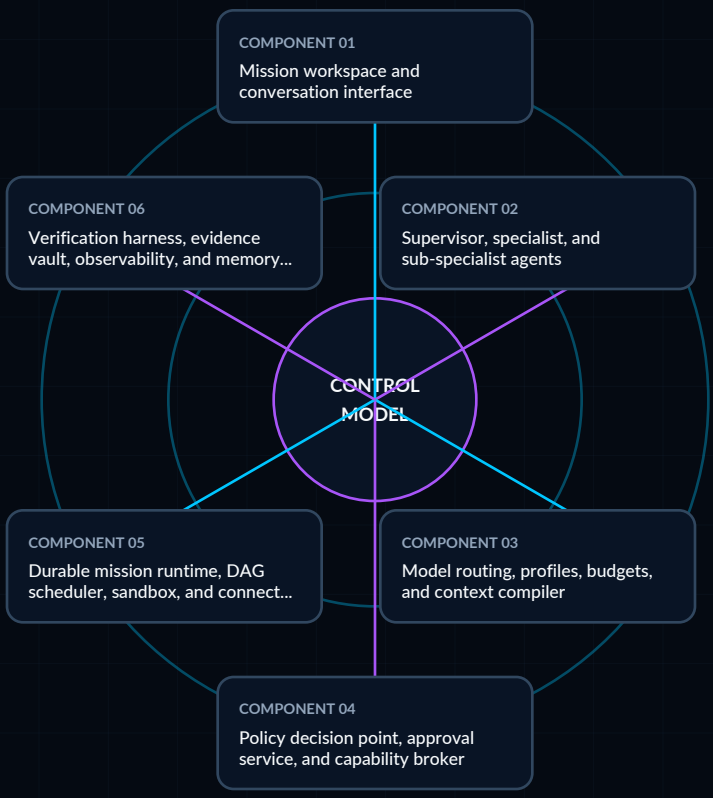
LAYER 03
Agent and Model Fabric

LAYER 04
Governance and Authorization

LAYER 05
Deterministic Execution

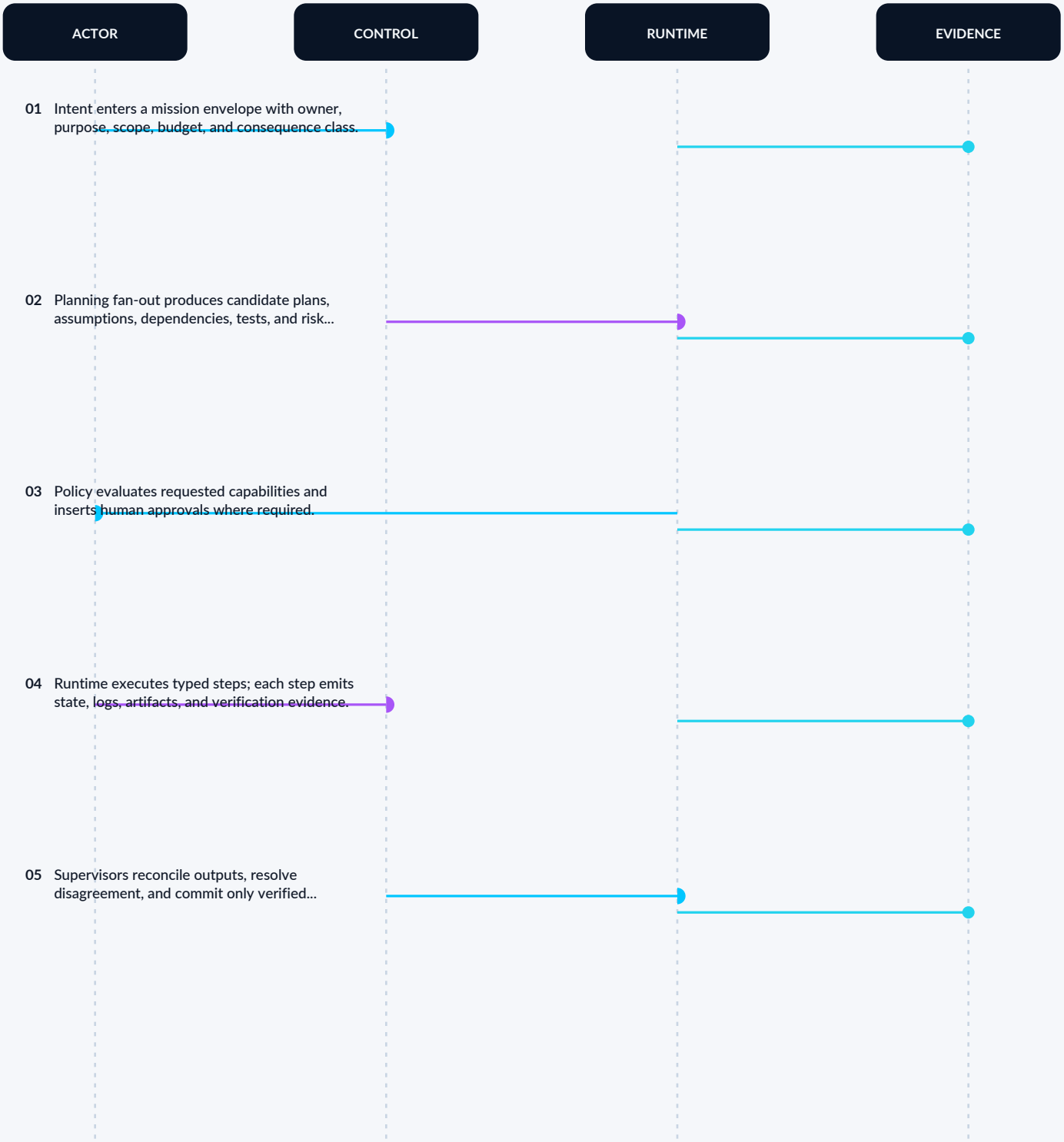
LAYER 06
Verification, Evidence, and Memory

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 intent versus model interpretation

TB-02
Model and agent proposal versus authorization

TB-03
Authorization versus deterministic execution

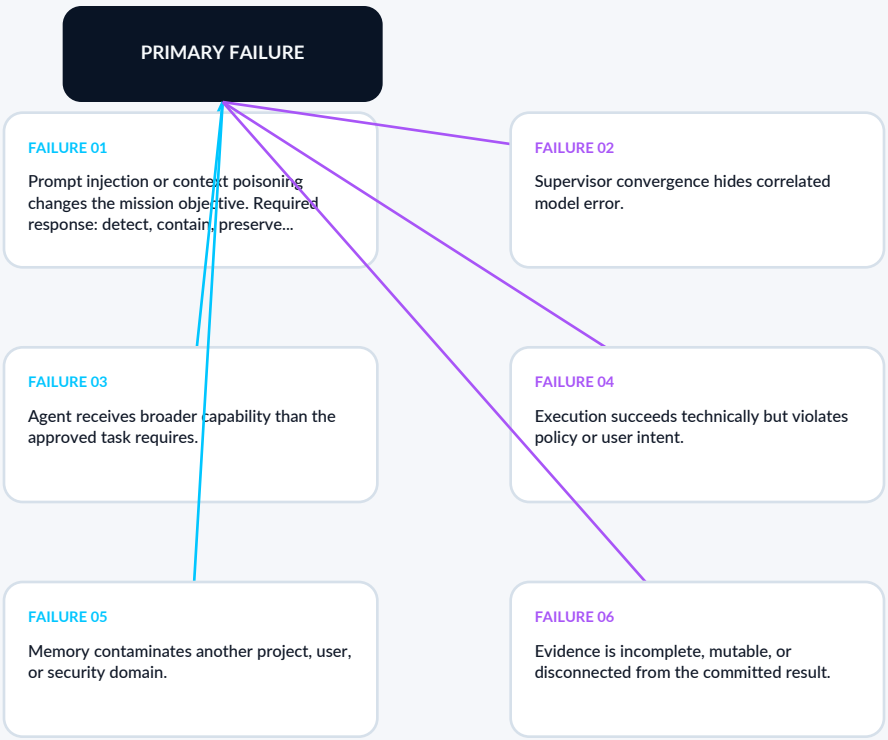
TB-04
Private project context versus shared organizational memory

TB-05
Tool connector versus external infrastructure

TB-06
Runtime evidence versus mutable conversational narrative

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 — **SINGLE-USER LOCAL WORKSTATION**

Bounded proof

PROFILE 02 — **GOVERNED ENGINEERING TEAM WORKSPACE**

Team-scale governance

PROFILE 03 — **AIR-GAPPED OR SOVEREIGN ENVIRONMENT**

Sovereign / high-assurance

PROFILE 04 — **ENTERPRISE CONTROL PLANE WITH DISTRIBUTED RUNNERS**

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-AI-0001, MYTHOS-AI-0002, MYTHOS-AI-0004, MYTHOS-AI-0007, MYTHOS-KNW-0001, MYTHOS-DAT-0005, MYTHOS-ID-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 governed, durable multi-agent architecture that separates reasoning, authorization, deterministic execution, verification, memory, observability, and evidence.

EXECUTIVE OUTCOMES

- Convert ambiguous human intent into explicit missions and typed plans.
- Permit models and agents to propose actions without silently acquiring execution authority.
- Execute consequential work through deterministic, policy-bound mechanisms.
- Preserve evidence, checkpoints, approvals, memory boundaries, and verified outcomes.

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

ARTIFICIAL INTELLIGENCE

IN SCOPE

A governed, durable multi-agent architecture that separates reasoning, authorization, deterministic execution, verification, memory, observability, 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

This candidate architecture describes the specified PANTHEON direction. It is not evidence that the platform is released, implemented in full, or production-certified.

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01

Experience and Intent

Experience and Intent 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

Mission and Plan Compiler

Mission and Plan Compiler 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

Agent and Model Fabric

Agent and Model Fabric 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

Governance and Authorization

Governance and Authorization 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

Deterministic Execution

Deterministic Execution 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

Verification, Evidence, and Memory

Verification, Evidence, and Memory 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

Mission workspace and conversation interface

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

COMPONENT 02

Supervisor, specialist, and sub-specialist agents

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

COMPONENT 03

Model routing, profiles, budgets, and context compiler

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Policy decision point, approval service, and capability broker

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

COMPONENT 05

Durable mission runtime, DAG scheduler, sandbox, and connector plane

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

COMPONENT 06

Verification harness, evidence vault, observability, and memory services

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 enters a mission envelope with owner, purpose, scope, budget, and consequence class.

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

2. Planning fan-out produces candidate plans, assumptions, dependencies, tests, and risk statements.

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

3. Policy evaluates requested capabilities and inserts human approvals where required.

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

4. Runtime executes typed steps; each step emits state, logs, artifacts, and verification evidence.

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

5. Supervisors reconcile outputs, resolve disagreement, and commit only verified outcomes.

   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 intent versus model interpretation

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

TRUST BOUNDARY 02

Model and agent proposal versus authorization

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

TRUST BOUNDARY 03

Authorization versus deterministic execution

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

TRUST BOUNDARY 04

Private project context versus shared organizational memory

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

TRUST BOUNDARY 05

Tool connector versus external infrastructure

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

TRUST BOUNDARY 06

Runtime evidence versus mutable conversational narrative

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 / SINGLE-USER LOCAL WORKSTATION

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 / GOVERNED ENGINEERING TEAM WORKSPACE

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 / AIR-GAPPED OR SOVEREIGN ENVIRONMENT

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 / ENTERPRISE CONTROL PLANE WITH DISTRIBUTED RUNNERS

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 / Register mission and consequence

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 / Compile plan and acceptance tests

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 / Authorize capabilities and budgets

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 / Execute checkpointed work

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 / Verify outcome and collect evidence

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 / Integrate or roll back

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 / Retain, summarize, and reassess memory

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

Prompt injection or context poisoning changes the mission objective.

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

FAILURE SCENARIO 02

Supervisor convergence hides correlated model error.

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

FAILURE SCENARIO 03

Agent receives broader capability than the approved task requires.

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

FAILURE SCENARIO 04

Execution succeeds technically but violates policy or user intent.

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

FAILURE SCENARIO 05

Memory contaminates another project, user, or security domain.

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

FAILURE SCENARIO 06

Evidence is incomplete, mutable, or disconnected from the committed result.

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-006 - Local-First AI Inference Cluster and Edge Node
- RA-007 - AI-Assisted Software Engineering Pipeline
- RA-009 - Zero-Trust Identity, Policy, Secrets, and Access Fabric
- RA-010 - Local-First Knowledge, Memory, and Evidence Platform
- RA-011 - Digital Twin, Infrastructure Automation, and Change Assurance
- RA-012 - Enterprise Observability, SRE, and Incident Command Platform
- RA-016 - Realtime Voice, Media, and Multimodal Interaction Platform
- RA-019 - Sovereign Browser, Remote Access, and Web Automation
- RA-020 - Adaptive Learning, Cyber Range, and Workforce Assurance

MAPPED MYTHOS STANDARDS

- MYTHOS-AI-0001
- MYTHOS-AI-0002
- MYTHOS-AI-0004
- MYTHOS-AI-0007
- MYTHOS-KNW-0001
- MYTHOS-DAT-0005
- MYTHOS-ID-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 autonomous or semi-autonomous action resolves to a mission, actor, policy decision, capability grant, execution trace, verification result, and accountable owner.
AC-14	Model output cannot directly bypass deterministic execution controls for consequential actions.
AC-15	Parallel agent and multi-model disagreement is retained and resolved through explicit arbitration criteria.
AC-16	Memory promotion, sharing, correction, expiration, and deletion are governed independently from chat history.
AC-17	PANTHEON-specific labels are treated as architecture assignments rather than proof of current implementation.
AC-18	A kill switch, pause, checkpoint recovery, budget stop, and safe rollback are demonstrated for representative missions.

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 **NIST AI Risk Management Framework 1.0 (revision in progress as of publication date)**

https://www.nist.gov/itl/ai-risk-management-framework

SOURCE 02 **NIST AI 600-1, Generative AI Profile**

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

SOURCE 03 **NIST SP 800-53 Rev. 5**

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

SOURCE 04 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

SOURCE 05 **NIST SP 800-218, Secure Software Development Framework**

https://csrc.nist.gov/pubs/sp/800/218/final

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-001
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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

SIEM, SOAR, NDR, and Security Operations Center

A federated, evidence-bearing security operations architecture integrating telemetry, SIEM, detection engineering, SOAR, NDR, case management, response, and incident learning.

PERMANENT DOCUMENT ID
RA-002

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Canonical Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Canonical Set	Cybersecurity and Network Security	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Normalize high-value security telemetry into queryable, governed schemas.
2. Treat detections, parsers, enrichments, and playbooks as versioned engineering artifacts.
3. Automate safe containment while preserving human authority for destructive or ambiguous response.
4. Produce complete incident evidence and continuous detection-quality feedback.

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
Telemetry and Sensing

LAYER 02
Collection and Normalization

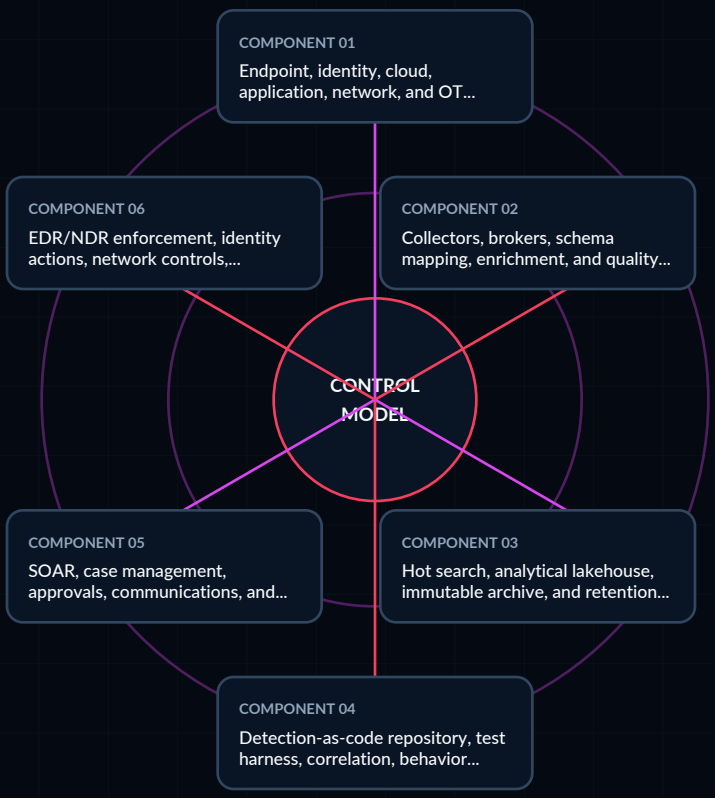
LAYER 03
Security Data and Search

LAYER 04
Detection and Analytics

LAYER 05
Orchestration and Case Management

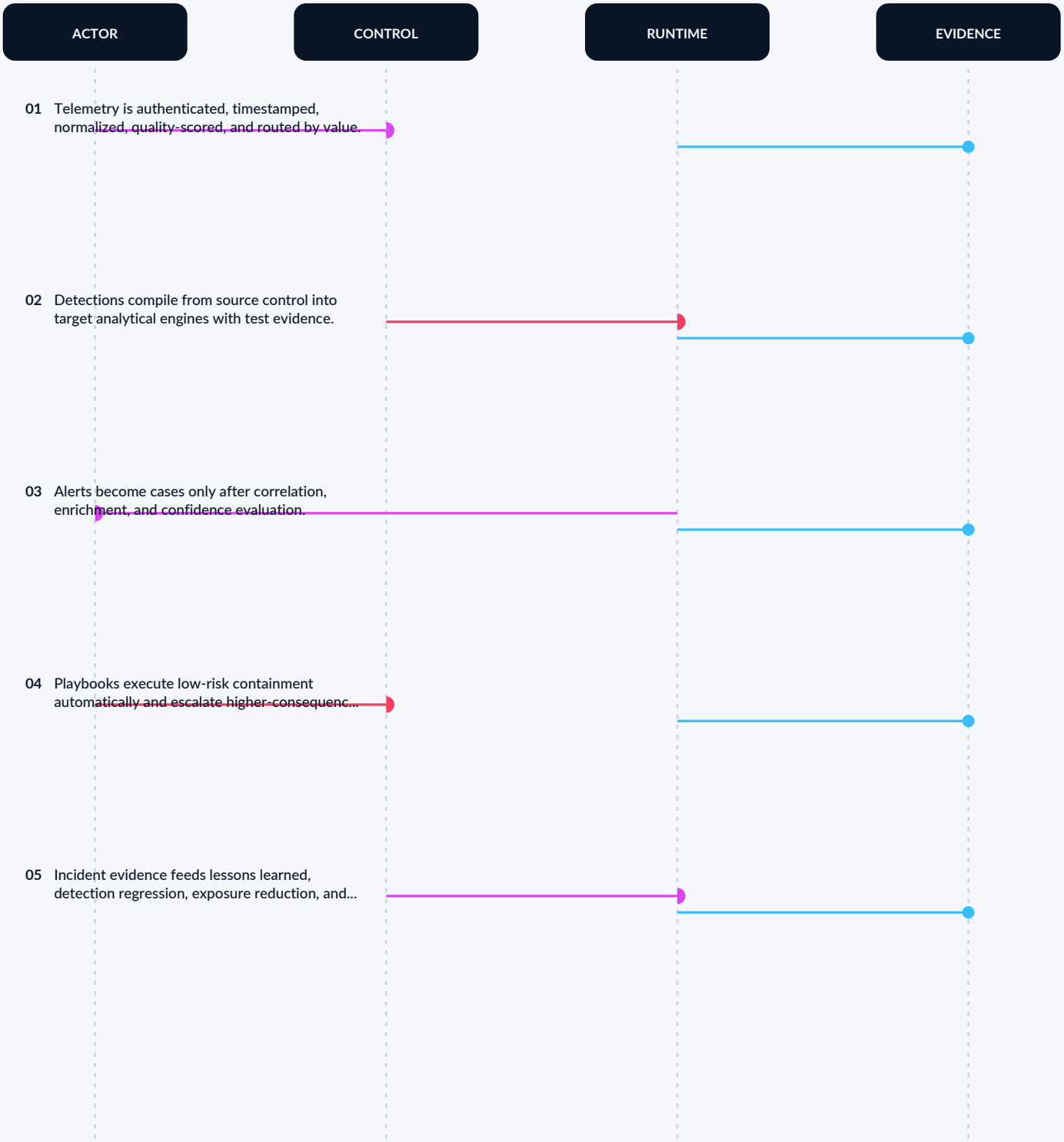
LAYER 06
Response and Evidence

CORE COMPONENTS



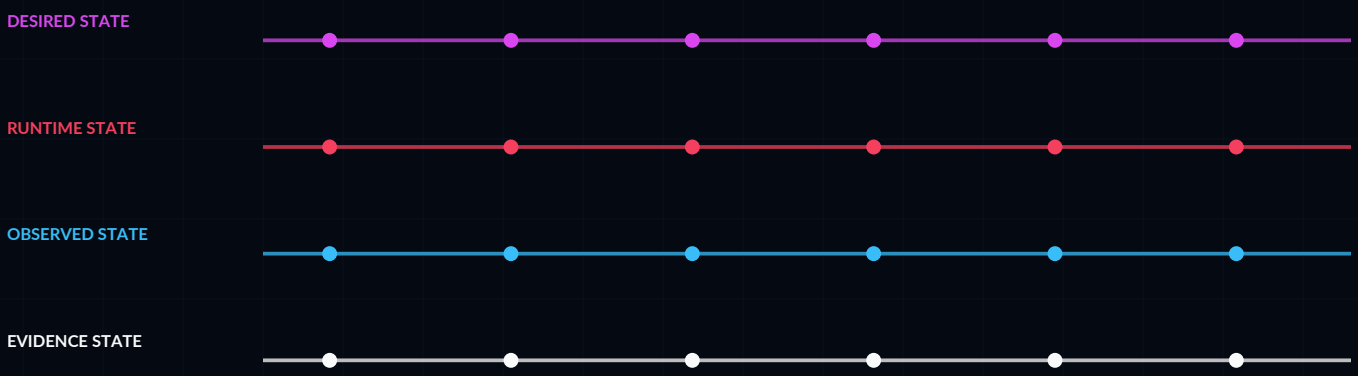
Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

Primary sequence and control flow



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

State, data, authority, and trust flow

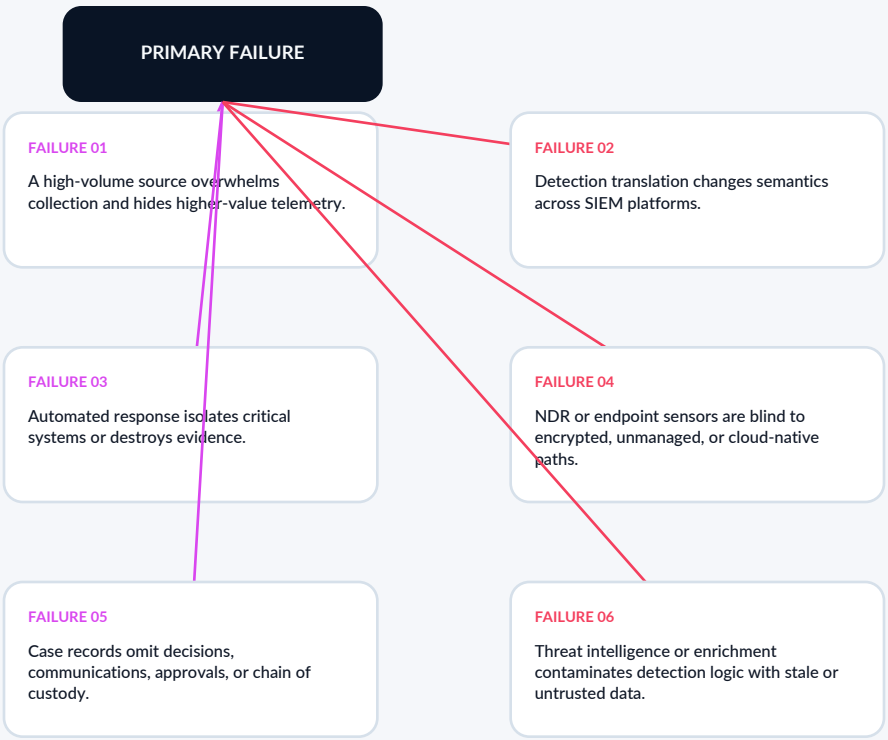


TRUST BOUNDARY REGISTER

TB-01 Sensor and collector trust	**TB-02** Security data plane and analytical tenant	**TB-03** Detection source repository and deployment path
TB-04 SOAR authority and target control planes	**TB-05** Case data and regulated evidence	**TB-06** Analyst workstation and privileged response session

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	SMALL ENTERPRISE MANAGED SOC	Bounded proof
PROFILE 02	FEDERATED ENTERPRISE SOC	Team-scale governance
PROFILE 03	HIGH-ASSURANCE REGULATED SOC	Sovereign / high-assurance
PROFILE 04	HYBRID IT/OT SECURITY OPERATIONS	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-SEC-0013, MYTHOS-SEC-0016, MYTHOS-DAT-0002, MYTHOS-DAT-0005, MYTHOS-PLT-0001, MYTHOS-SRE-0003

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 federated, evidence-bearing security operations architecture integrating telemetry, SIEM, detection engineering, SOAR, NDR, case management, response, and incident learning.

EXECUTIVE OUTCOMES

- Normalize high-value security telemetry into queryable, governed schemas.
- Treat detections, parsers, enrichments, and playbooks as versioned engineering artifacts.
- Automate safe containment while preserving human authority for destructive or ambiguous response.
- Produce complete incident evidence and continuous detection-quality feedback.

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

CYBERSECURITY AND NETWORK SECURITY

IN SCOPE

A federated, evidence-bearing security operations architecture integrating telemetry, SIEM, detection engineering, SOAR, NDR, case management, response, and incident learning.

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 architecture prioritizes governed detection and response over monolithic log accumulation, while preventing automation from outrunning evidence, authorization, or business safety.

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01

Telemetry and Sensing

Telemetry and Sensing 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

Collection and Normalization

Collection and Normalization 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

Security Data and Search

Security Data and Search 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

Detection and Analytics

Detection and Analytics 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

Orchestration and Case Management

Orchestration and Case Management 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

Response and Evidence

Response 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

Endpoint, identity, cloud, application, network, and OT sensors

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

COMPONENT 02

Collectors, brokers, schema mapping, enrichment, and quality gates

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

COMPONENT 03

Hot search, analytical lakehouse, immutable archive, and retention policy

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Detection-as-code repository, test harness, correlation, behavior analytics, and threat intelligence

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

COMPONENT 05

SOAR, case management, approvals, communications, and incident command

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

COMPONENT 06

EDR/NDR enforcement, identity actions, network controls, recovery workflows, and evidence vault

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. Telemetry is authenticated, timestamped, normalized, quality-scored, and routed by value.

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

2. Detections compile from source control into target analytical engines with test evidence.

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

3. Alerts become cases only after correlation, enrichment, and confidence evaluation.

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

4. Playbooks execute low-risk containment automatically and escalate higher-consequence steps.

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

5. Incident evidence feeds lessons learned, detection regression, exposure reduction, and recovery.

   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

Sensor and collector trust

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

TRUST BOUNDARY 02

Security data plane and analytical tenant

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

TRUST BOUNDARY 03

Detection source repository and deployment path

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

TRUST BOUNDARY 04

SOAR authority and target control planes

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

TRUST BOUNDARY 05

Case data and regulated 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

Analyst workstation and privileged response session

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 / SMALL ENTERPRISE MANAGED SOC

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 / FEDERATED ENTERPRISE SOC

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 / HIGH-ASSURANCE REGULATED SOC

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 / HYBRID IT/OT SECURITY OPERATIONS

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 / Onboard source and data contract

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 / Validate schema, clock, identity, and retention

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 / Engineer and test detections

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 / Triage and correlate

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 / Authorize and execute response

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 / Recover and preserve evidence

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 / Measure coverage and improve

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

A high-volume source overwhelms collection and hides higher-value telemetry.

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

FAILURE SCENARIO 02

Detection translation changes semantics across SIEM platforms.

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

FAILURE SCENARIO 03

Automated response isolates critical systems or destroys evidence.

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

FAILURE SCENARIO 04

NDR or endpoint sensors are blind to encrypted, unmanaged, or cloud-native paths.

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

FAILURE SCENARIO 05

Case records omit decisions, communications, approvals, or chain of custody.

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

FAILURE SCENARIO 06

Threat intelligence or enrichment contaminates detection logic with stale or untrusted data.

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-009 - Zero-Trust Identity, Policy, Secrets, and Access Fabric
- RA-012 - Enterprise Observability, SRE, and Incident Command Platform
- RA-019 - Sovereign Browser, Remote Access, and Web Automation

MAPPED MYTHOS STANDARDS

- MYTHOS-SEC-0013
- MYTHOS-SEC-0016
- MYTHOS-DAT-0002
- MYTHOS-DAT-0005
- MYTHOS-PLT-0001
- MYTHOS-SRE-0003

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 production detection is versioned, tested against positive and negative cases, peer-reviewed, deployed through controlled automation, and linked to coverage objectives.
AC-14	Telemetry quality measures include completeness, latency, parsing success, identity resolution, clock integrity, and dropped-event evidence.
AC-15	Response playbooks classify actions by reversibility, blast radius, evidence impact, and approval requirement.
AC-16	Security data retention separates rapid analysis, investigation, legal hold, and immutable archival needs.
AC-17	The SOC can operate through loss of one SIEM, identity provider, case platform, or response integration.
AC-18	Incident closure requires recovery validation, evidence completeness, detection regression, and accountable lessons learned.

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 **NIST Cybersecurity Framework 2.0**

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

SOURCE 02 **NIST SP 800-53 Rev. 5**

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

SOURCE 03 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

SOURCE 04 **Open Cybersecurity Schema Framework**

https://ocsf.io/

SOURCE 05 **Sigma Detection Format and Rule Specification**

https://sigmahq.io/sigma-specification/specification/sigma-rules-specification.html

SOURCE 06 **MITRE ATT&CK**

https://attack.mitre.org/

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

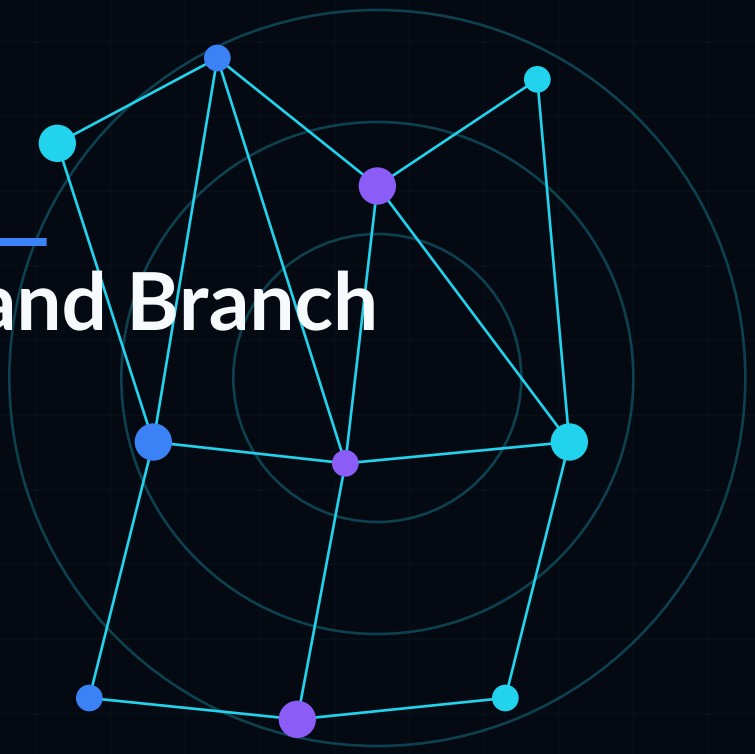
PERMANENT DOCUMENT IDENTITY	RA-002
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Canonical Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Enterprise Campus and Branch Network



An identity-driven, observable, programmable campus, branch, wireless, WAN, and security architecture with explicit source of truth and change assurance.

PERMANENT DOCUMENT ID
RA-003

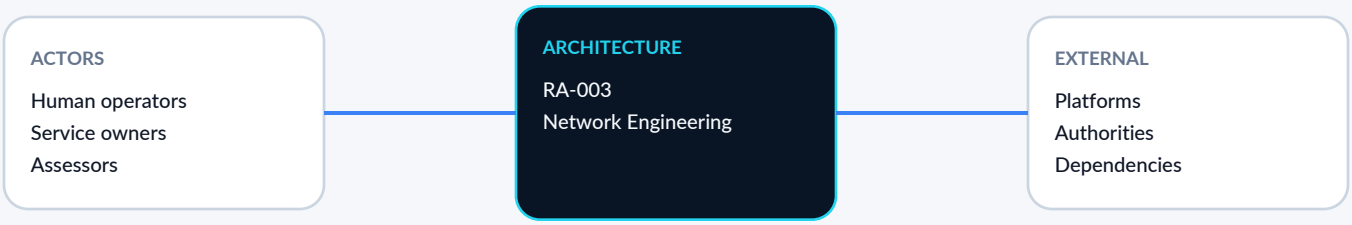
PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Canonical Set

Architecture identity and operating position

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

SYSTEM CONTEXT



EXECUTIVE OUTCOMES

1. Deliver predictable user, device, application, and site connectivity.
2. Bind access to identity, posture, role, and policy rather than physical location alone.
3. Reduce failure domains through routed design, resilient services, and explicit local survivability.
4. Operate the network from authoritative intent, telemetry, validation, and controlled 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

LAYER 01
Physical and Underlay Fabric

LAYER 02
Overlay and Segmentation

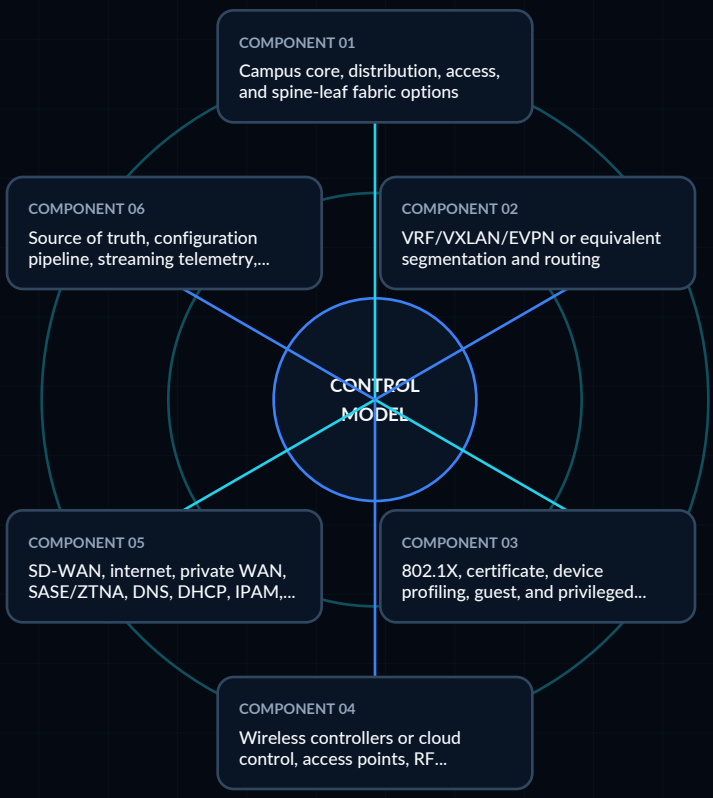
LAYER 03
Identity and Access

LAYER 04
Wireless and Edge Access

LAYER 05
Branch, WAN, Internet, and Cloud

LAYER 06
Automation, Observability, and Assurance

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.

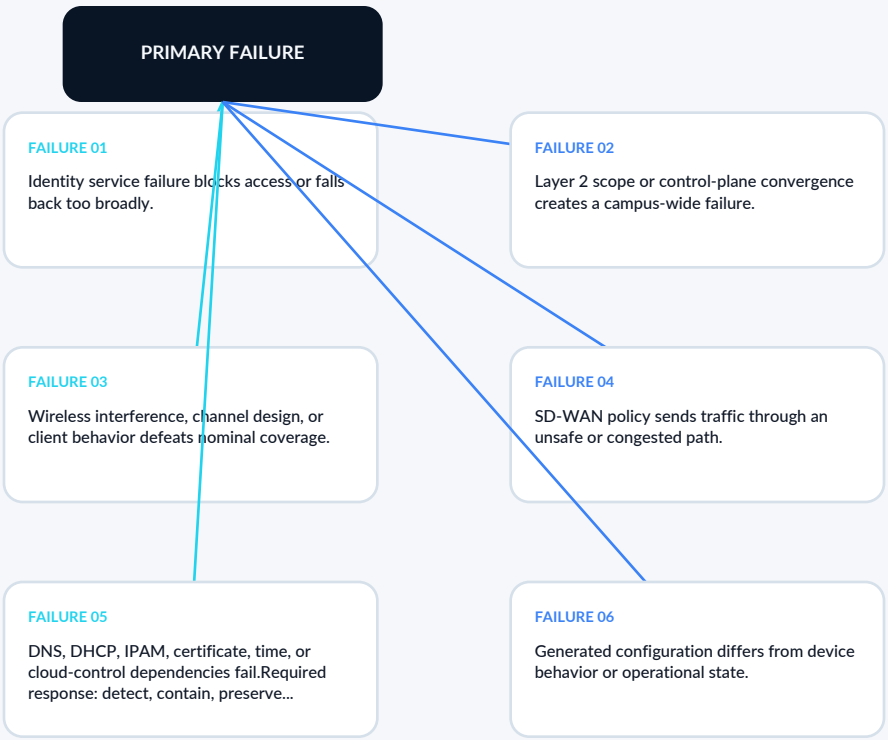
State, data, authority, and trust flow

TRUST BOUNDARY REGISTER

TB-01 Untrusted endpoint versus access edge	TB-02 User/device identity versus authorization policy	TB-03 Campus/branch segment versus shared services
TB-04 Enterprise network versus internet, SaaS, and cloud	TB-05 Management plane versus data plane	TB-06 Vendor controller or cloud service versus local survivability

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	TRADITIONAL ROUTED CAMPUS MODERNIZATION	Bounded proof
PROFILE 02	LARGE EVPN/VXLAN CAMPUS FABRIC	Team-scale governance
PROFILE 03	CLOUD-MANAGED BRANCH AND WIRELESS ESTATE	Sovereign / high-assurance
PROFILE 04	HIGH-ASSURANCE SEGMENTED CAMPUS	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-0001, MYTHOS-NET-0005, MYTHOS-NET-0010, MYTHOS-NET-0011, MYTHOS-SEC-0007, MYTHOS-TEL-0001

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

An identity-driven, observable, programmable campus, branch, wireless, WAN, and security architecture with explicit source of truth and change assurance.

EXECUTIVE OUTCOMES

- Deliver predictable user, device, application, and site connectivity.
- Bind access to identity, posture, role, and policy rather than physical location alone.
- Reduce failure domains through routed design, resilient services, and explicit local survivability.
- Operate the network from authoritative intent, telemetry, validation, and controlled 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

An identity-driven, observable, programmable campus, branch, wireless, WAN, and security architecture with explicit source of truth and change assurance.

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 architecture supports routed fabrics, wireless access, branch connectivity, segmentation, direct internet and cloud access, and policy enforcement without requiring a single vendor design.

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01

Physical and Underlay Fabric

Physical and Underlay Fabric 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

Overlay and Segmentation

Overlay and Segmentation 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

Identity and Access

Identity and Access 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

Wireless and Edge Access

Wireless and Edge Access 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

Branch, WAN, Internet, and Cloud

Branch, WAN, Internet, and Cloud 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

Automation, Observability, and Assurance

Automation, Observability, and Assurance 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

Campus core, distribution, access, and spine-leaf fabric options

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

COMPONENT 02

VRF/VXLAN/EVPN or equivalent segmentation and routing

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

COMPONENT 03

802.1X, certificate, device profiling, guest, and privileged access policy

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Wireless controllers or cloud control, access points, RF assurance, and client telemetry

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

COMPONENT 05

SD-WAN, internet, private WAN, SASE/ZTNA, DNS, DHCP, IPAM, and cloud connectivity

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

COMPONENT 06

Source of truth, configuration pipeline, streaming telemetry, synthetic tests, and change evidence

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. Device and service intent is modeled before configuration.

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

2. Identity and posture determine access and segment assignment.

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

3. Traffic follows policy through local, WAN, internet, cloud, and security paths.

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

4. Telemetry validates reachability, path, performance, experience, and policy.

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

5. Change evidence reconciles intended, generated, deployed, and observed state.

   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

Untrusted endpoint versus access edge

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

TRUST BOUNDARY 02

User/device identity versus authorization policy

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

TRUST BOUNDARY 03

Campus/branch segment versus shared services

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

TRUST BOUNDARY 04

Enterprise network versus internet, SaaS, and cloud

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

TRUST BOUNDARY 05

Management plane versus data plane

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

TRUST BOUNDARY 06

Vendor controller or cloud service versus local survivability

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 / TRADITIONAL ROUTED CAMPUS MODERNIZATION

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 / LARGE EVPN/VXLAN CAMPUS FABRIC

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 / CLOUD-MANAGED BRANCH AND WIRELESS ESTATE

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 SEGMENTED CAMPUS

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 / Inventory and baseline

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 / Model addressing, topology, identity, and policy

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 / Validate design and failure domains

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 / Deploy in bounded site waves

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 / Verify user and application journeys

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 / Operate with telemetry and experience SLOs

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 / Reconcile drift and retire legacy paths

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

Identity service failure blocks access or falls back too broadly.

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

FAILURE SCENARIO 02

Layer 2 scope or control-plane convergence creates a campus-wide failure.

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

FAILURE SCENARIO 03

Wireless interference, channel design, or client behavior defeats nominal coverage.

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

FAILURE SCENARIO 04

SD-WAN policy sends traffic through an unsafe or congested path.

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

FAILURE SCENARIO 05

DNS, DHCP, IPAM, certificate, time, or cloud-control dependencies fail.

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

FAILURE SCENARIO 06

Generated configuration differs from device behavior or operational state.

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-004 - Network Source of Truth Implementation Pipeline
- RA-009 - Zero-Trust Identity, Policy, Secrets, and Access Fabric
- RA-011 - Digital Twin, Infrastructure Automation, and Change Assurance
- RA-012 - Enterprise Observability, SRE, and Incident Command Platform
- RA-015 - Edge, OT, IoT, and Telecommunications Convergence

MAPPED MYTHOS STANDARDS

- MYTHOS-NET-0001
- MYTHOS-NET-0005
- MYTHOS-NET-0010
- MYTHOS-NET-0011
- MYTHOS-SEC-0007
- MYTHOS-TEL-0001

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 site has documented underlay, overlay, identity, service, management, security, and failure-domain diagrams.
AC-14	Access policy is tested for employees, administrators, contractors, guests, unmanaged devices, IoT, voice, and emergency operations.
AC-15	DNS, DHCP, IPAM, certificate, time, identity, telemetry, and management dependencies have local or tested degraded modes.
AC-16	Wireless assurance includes spectrum, channel, power, capacity, roaming, client compatibility, and application-experience evidence.
AC-17	WAN path policy is validated under carrier loss, brownout, asymmetric routing, SaaS failure, and security-service failure.
AC-18	A representative site can be rebuilt from authoritative data, signed artifacts, and documented bootstrap dependencies.

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 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

SOURCE 02 **NIST SP 1800-35, Implementing a Zero Trust Architecture**

https://www.nccoe.nist.gov/projects/implementing-zero-trust-architecture

SOURCE 03 **RFC 8345, YANG Data Model for Network Topologies**

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

SOURCE 04 **RFC 8969, Framework for Automating Service and Network Management**

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

SOURCE 05 **OpenConfig Vendor-Neutral Network Models**

https://www.openconfig.net/

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

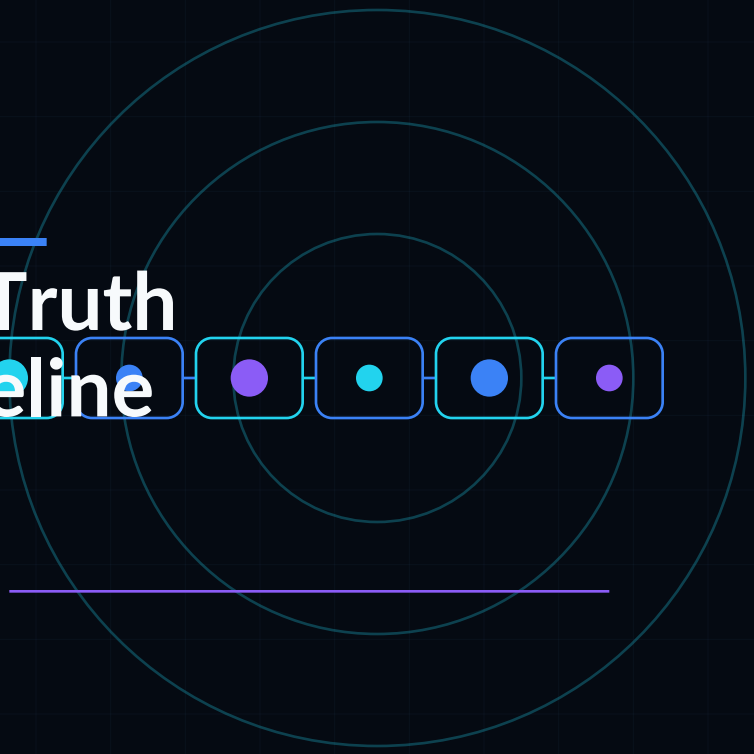
PERMANENT DOCUMENT IDENTITY	RA-003
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Canonical Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

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

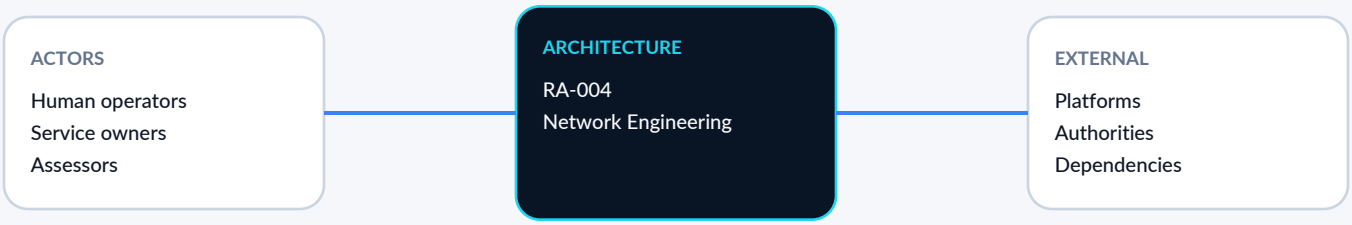
PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Canonical Set

Architecture identity and operating position

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

LAYER 01
Intent and Service Model

LAYER 02
Authoritative Data and Ownership

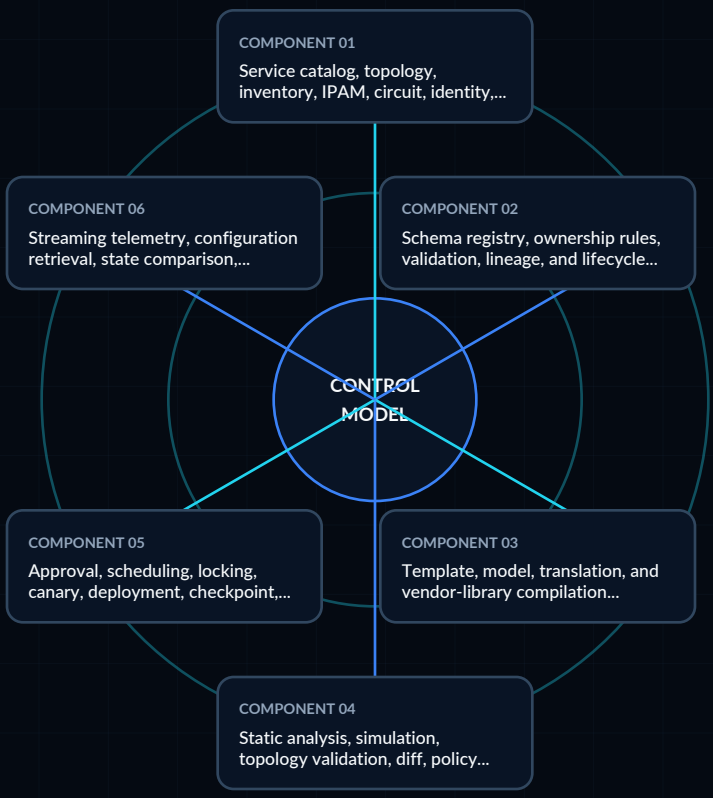
LAYER 03
Compilation and Rendering

LAYER 04
Pre-Change Analysis

LAYER 05
Execution and Transaction Control

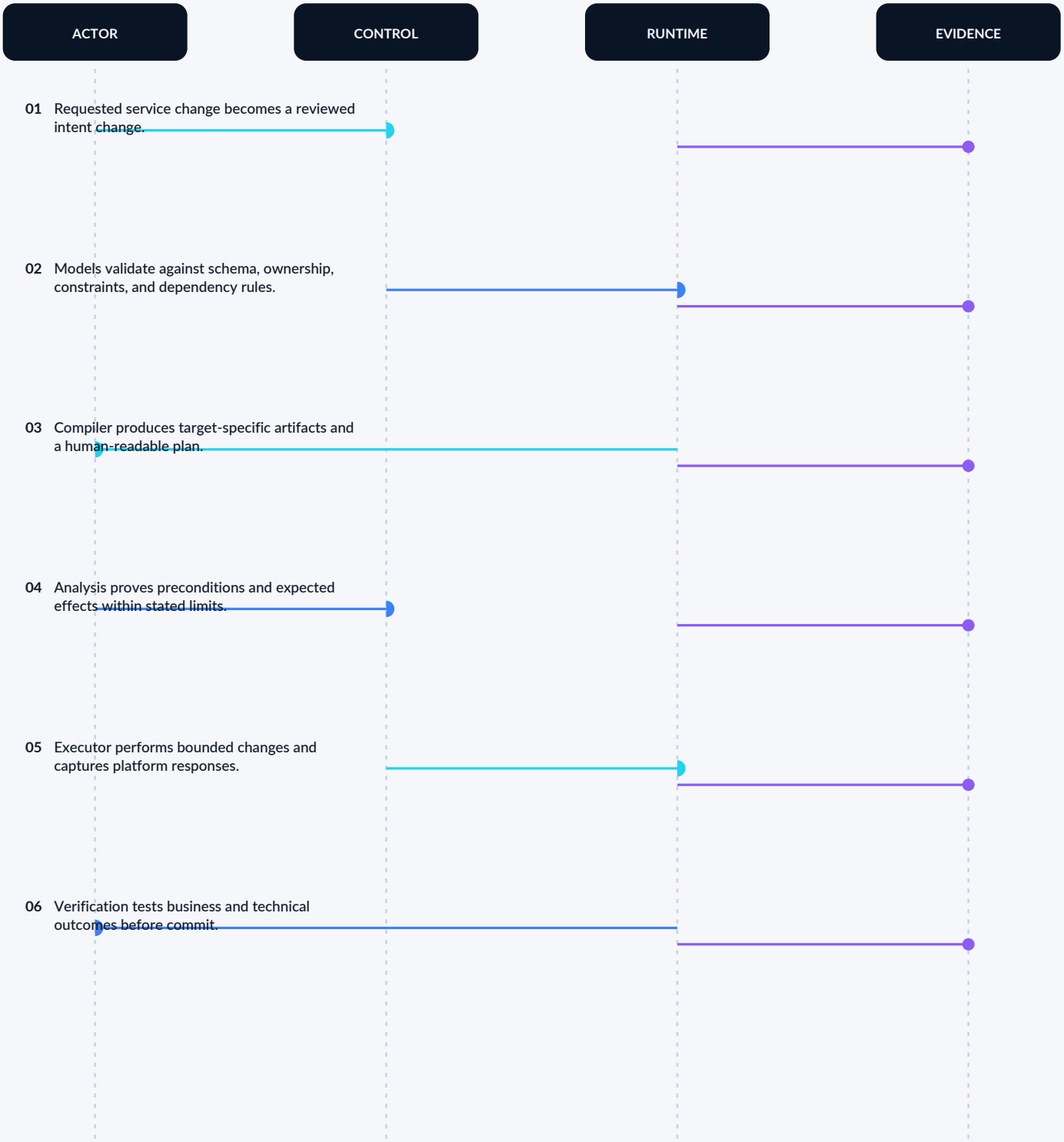
LAYER 06
Observation, Reconciliation, and Evidence

CORE COMPONENTS



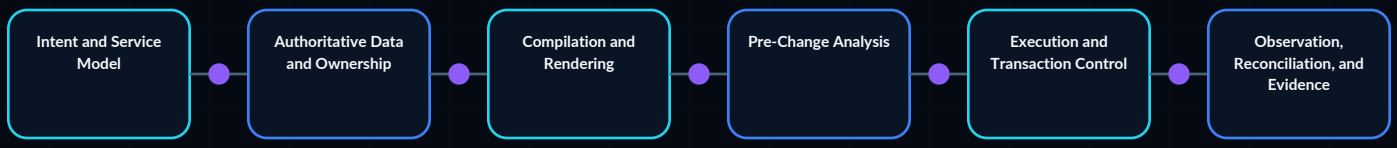
Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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 request versus authoritative model

TB-02
Authoritative model versus derived artifact

TB-03
Simulation or validation environment versus production

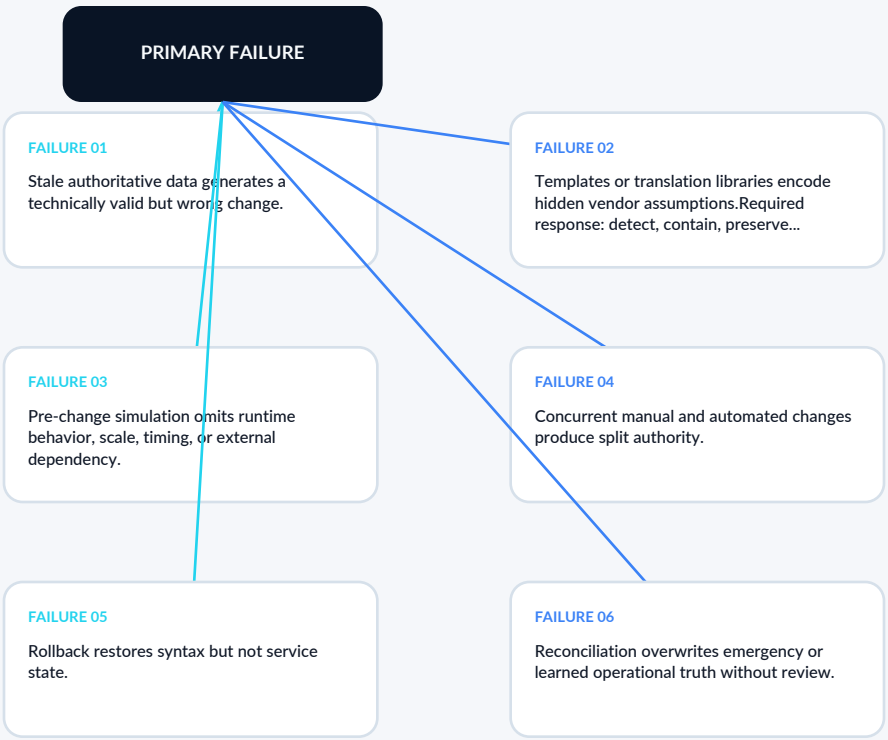
TB-04
Pipeline identity versus device privilege

TB-05
Intended state versus observed operational state

TB-06
Emergency change versus normal governance

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	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-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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Secure Prosumer and Sovereign Home Network

A privacy-preserving, resilient home and prosumer architecture for trusted administration, family safety, work, IoT, media, guests, laboratories, remote access, and local-first services.

PERMANENT DOCUMENT ID
RA-005

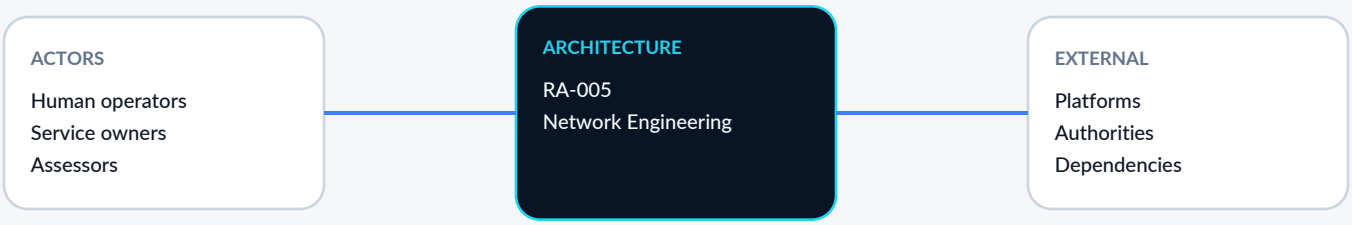
PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Canonical Set

Architecture identity and operating position

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

SYSTEM CONTEXT



EXECUTIVE OUTCOMES

1. Protect household members, identities, devices, media, work, laboratories, and local services.
2. Separate high-risk IoT, guests, work, administration, and sensitive local systems.
3. Preserve local operation during cloud, ISP, identity, DNS, or management outages.
4. Provide understandable administration, recovery, privacy, and evidence without excessive operational burden.

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
Physical and Power Foundation

LAYER 02
Internet Edge and Secure Routing

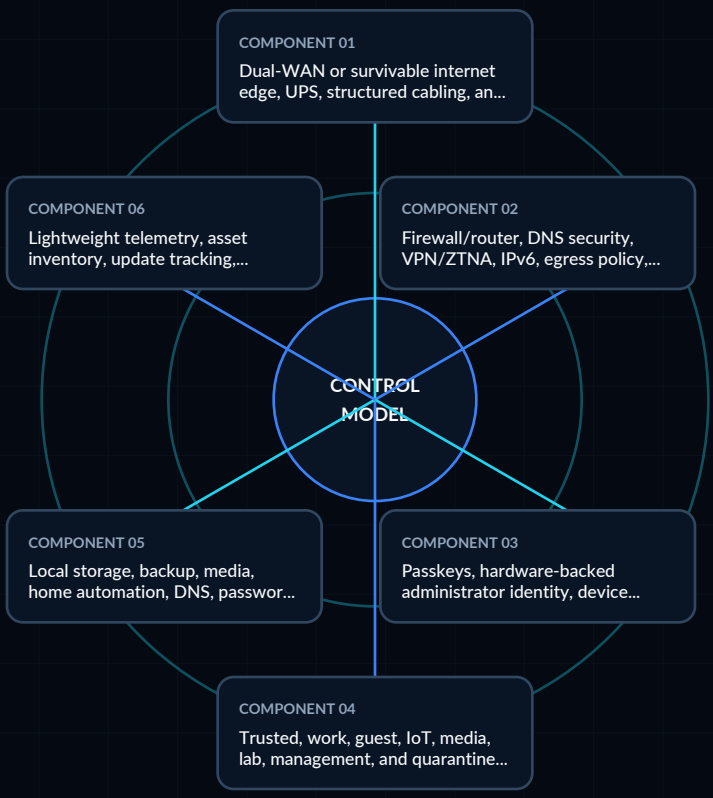
LAYER 03
Identity, Administration, and Remote Access

LAYER 04
Segmentation and Device Classes

LAYER 05
Local Services, Storage, and AI

LAYER 06
Monitoring, Recovery, and Privacy

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.

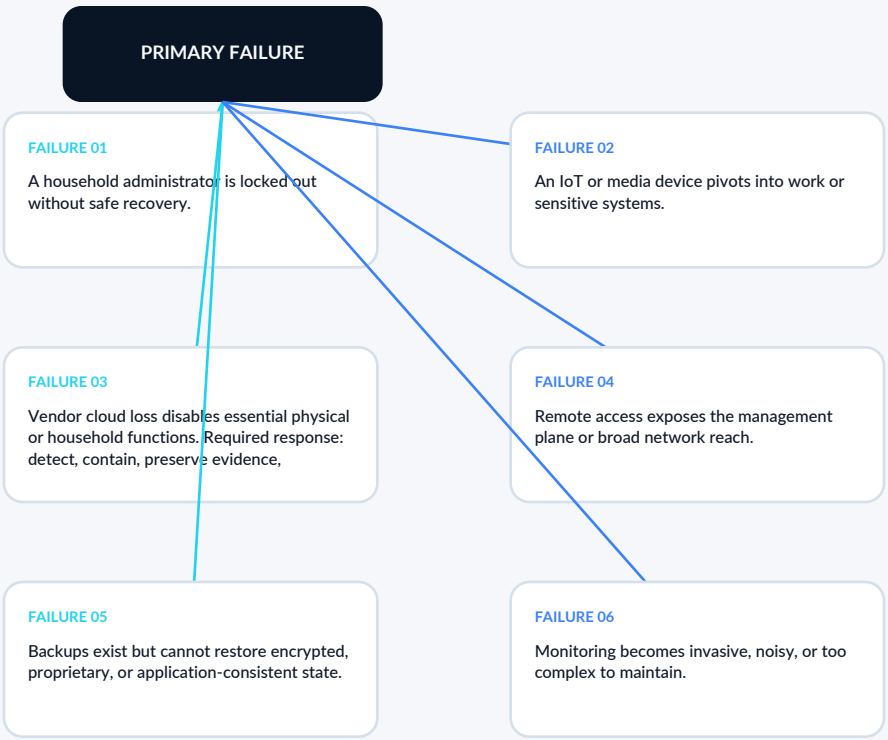
State, data, authority, and trust flow

TRUST BOUNDARY REGISTER

TB-01 Household member versus administrator	TB-02 Trusted personal device versus unmanaged or guest device	TB-03 IoT device versus sensitive local services
TB-04 Home network versus employer-managed resources	TB-05 Local service versus cloud relay or vendor control	TB-06 Remote-access session versus management plane

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	ESSENTIAL SECURE HOME	Bounded proof
PROFILE 02	PROSUMER AND REMOTE-WORK HOME	Team-scale governance
PROFILE 03	SOVEREIGN LOCAL-SERVICES HOME	Sovereign / high-assurance
PROFILE 04	ADVANCED LABORATORY AND CREATOR ENVIRONMENT	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-0001, MYTHOS-NET-0011, MYTHOS-SEC-0003, MYTHOS-SEC-0007, MYTHOS-IOT-0001, MYTHOS-WEB-0001

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 privacy-preserving, resilient home and prosumer architecture for trusted administration, family safety, work, IoT, media, guests, laboratories, remote access, and local-first services.

EXECUTIVE OUTCOMES

- Protect household members, identities, devices, media, work, laboratories, and local services.
- Separate high-risk IoT, guests, work, administration, and sensitive local systems.
- Preserve local operation during cloud, ISP, identity, DNS, or management outages.
- Provide understandable administration, recovery, privacy, and evidence without excessive operational burden.

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 privacy-preserving, resilient home and prosumer architecture for trusted administration, family safety, work, IoT, media, guests, laboratories, remote access, and local-first services.

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 architecture scales enterprise principles to a household without turning the home into an unmanageable miniature enterprise or sending unnecessary data to external services.

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01

Physical and Power Foundation

Physical and Power Foundation 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

Internet Edge and Secure Routing

Internet Edge and Secure Routing 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

Identity, Administration, and Remote Access

Identity, Administration, and Remote Access 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

Segmentation and Device Classes

Segmentation and Device Classes 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

Local Services, Storage, and AI

Local Services, Storage, and AI 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

Monitoring, Recovery, and Privacy

Monitoring, Recovery, and Privacy 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

Dual-WAN or survivable internet edge, UPS, structured cabling, and secure physical placement

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

COMPONENT 02

Firewall/router, DNS security, VPN/ZTNA, IPv6, egress policy, and service exposure

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

COMPONENT 03

Passkeys, hardware-backed administrator identity, device identity, family roles, and break-glass access

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Trusted, work, guest, IoT, media, lab, management, and quarantine segments

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

COMPONENT 05

Local storage, backup, media, home automation, DNS, password vault, inference, and private services

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

COMPONENT 06

Lightweight telemetry, asset inventory, update tracking, alerting, restoration, and privacy controls

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. New devices enter an onboarding and classification path before trusted access.

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

2. Identity and segment policy limit east-west and outbound communication.

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

3. Sensitive services remain local unless a declared external dependency is accepted.

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

4. Remote access terminates through strong identity and bounded application or network access.

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

5. Backups, recovery keys, configuration, and service data are periodically restored and verified.

   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

Household member versus administrator

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

TRUST BOUNDARY 02

Trusted personal device versus unmanaged or guest device

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

TRUST BOUNDARY 03

IoT device versus sensitive local services

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

TRUST BOUNDARY 04

Home network versus employer-managed resources

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

TRUST BOUNDARY 05

Local service versus cloud relay or vendor control

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

TRUST BOUNDARY 06

Remote-access session versus management plane

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 / ESSENTIAL SECURE HOME

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 / PROSUMER AND REMOTE-WORK HOME

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 / SOVEREIGN LOCAL-SERVICES HOME

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 / ADVANCED LABORATORY AND CREATOR ENVIRONMENT

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 / Inventory people, devices, services, and data

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 / Establish secure administration and recovery

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 / Segment device classes

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 / Harden DNS, internet edge, and remote access

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 / Move selected services local

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 monitoring and restore exercises

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 / Review privacy, usability, and operational burden

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

A household administrator is locked out without safe recovery.

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

FAILURE SCENARIO 02

An IoT or media device pivots into work or sensitive systems.

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

FAILURE SCENARIO 03

Vendor cloud loss disables essential physical or household functions.

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

FAILURE SCENARIO 04

Remote access exposes the management plane or broad network reach.

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

FAILURE SCENARIO 05

Backups exist but cannot restore encrypted, proprietary, or application-consistent state.

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

FAILURE SCENARIO 06

Monitoring becomes invasive, noisy, or too complex to maintain.

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-006 - Local-First AI Inference Cluster and Edge Node
- RA-009 - Zero-Trust Identity, Policy, Secrets, and Access Fabric
- RA-015 - Edge, OT, IoT, and Telecommunications Convergence
- RA-019 - Sovereign Browser, Remote Access, and Web Automation

MAPPED MYTHOS STANDARDS

- MYTHOS-NET-0001
- MYTHOS-NET-0011
- MYTHOS-SEC-0003
- MYTHOS-SEC-0007
- MYTHOS-IOT-0001
- MYTHOS-WEB-0001

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	The design distinguishes essential household functions from optional convenience and defines local failure behavior for each.
AC-14	Administrator recovery uses protected offline material and does not depend on the same identity or cloud service being recovered.
AC-15	Every device class has onboarding, update, network, data, privacy, and retirement expectations understandable to a capable household operator.
AC-16	Remote work and employer-managed devices are isolated from personal administration and local sensitive services.
AC-17	Local services publish backup, restore, migration, and vendor-exit procedures before becoming essential.
AC-18	Telemetry and filtering preserve household privacy, minimize content collection, and remain explainable to affected users.

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 **NIST Cybersecurity Framework 2.0**

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

SOURCE 02 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

SOURCE 03 **NIST SP 800-53 Rev. 5**

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

SOURCE 04 **W3C Web Authentication Level 3**

https://www.w3.org/TR/webauthn-3/

SOURCE 05 **OpenTelemetry Specification**

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

SOURCE 06 **CISA Secure by Design**

https://www.cisa.gov/resources-tools/resources/secure-by-design

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

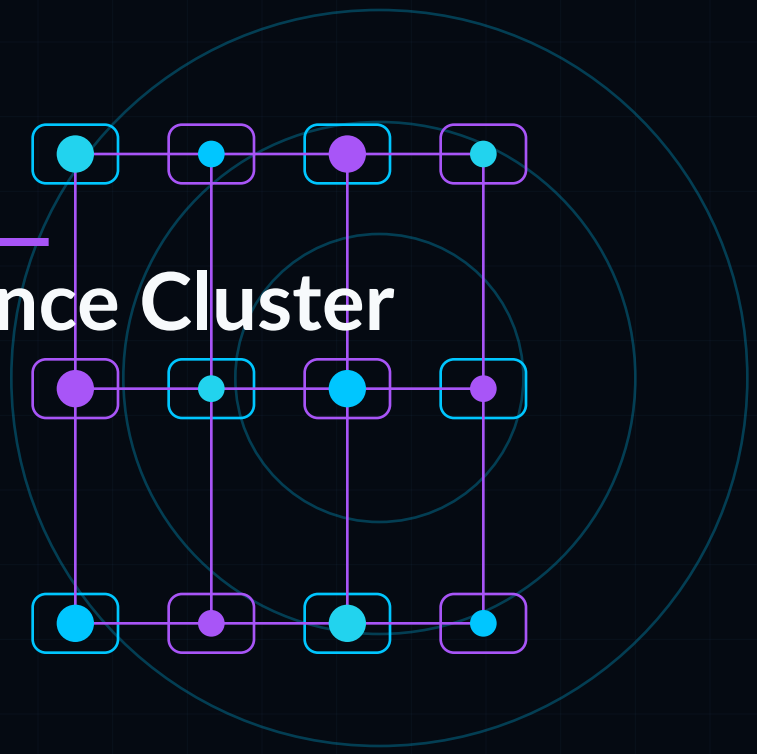
PERMANENT DOCUMENT IDENTITY	RA-005
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Canonical Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Local-First AI Inference Cluster and Edge Node



A local-first architecture for model acquisition, inference, routing, isolation, acceleration, retrieval, observability, privacy, and resilient edge operation.

PERMANENT DOCUMENT ID
RA-006

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Canonical Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Canonical Set	Artificial Intelligence	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Run approved models near the user or data when privacy, latency, resilience, or cost justify it.
2. Route workloads across CPU, GPU, NPU, accelerator, workstation, cluster, and approved cloud resources.
3. Protect model, data, prompt, context, cache, and tool boundaries.
4. Operate through resource pressure, model failure, offline periods, and node loss.

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
Model Registry and Supply Chain

LAYER 02
Scheduling and Model Routing

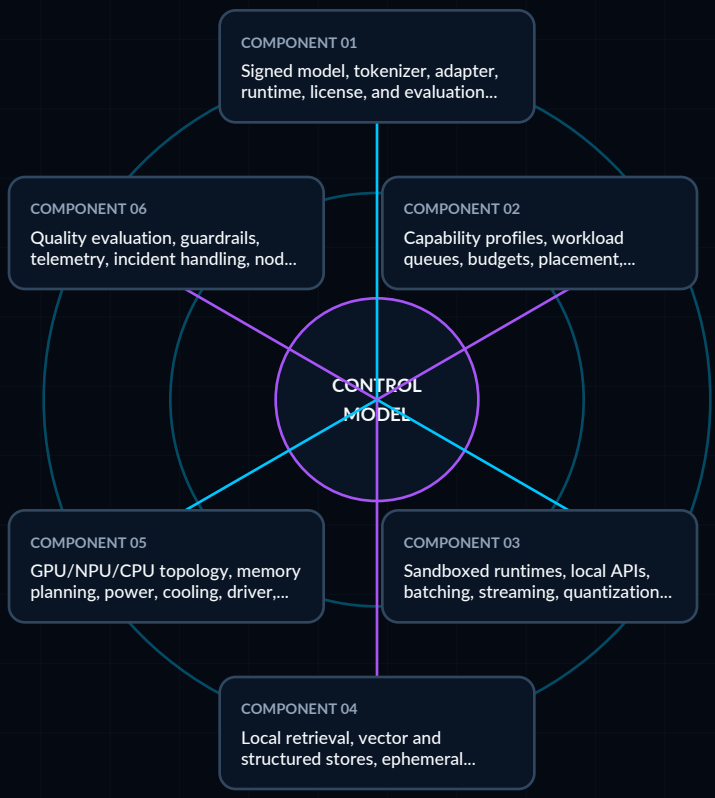
LAYER 03
Inference Runtime and Isolation

LAYER 04
Data, Retrieval, and Context

LAYER 05
Acceleration, Capacity, and Thermal Control

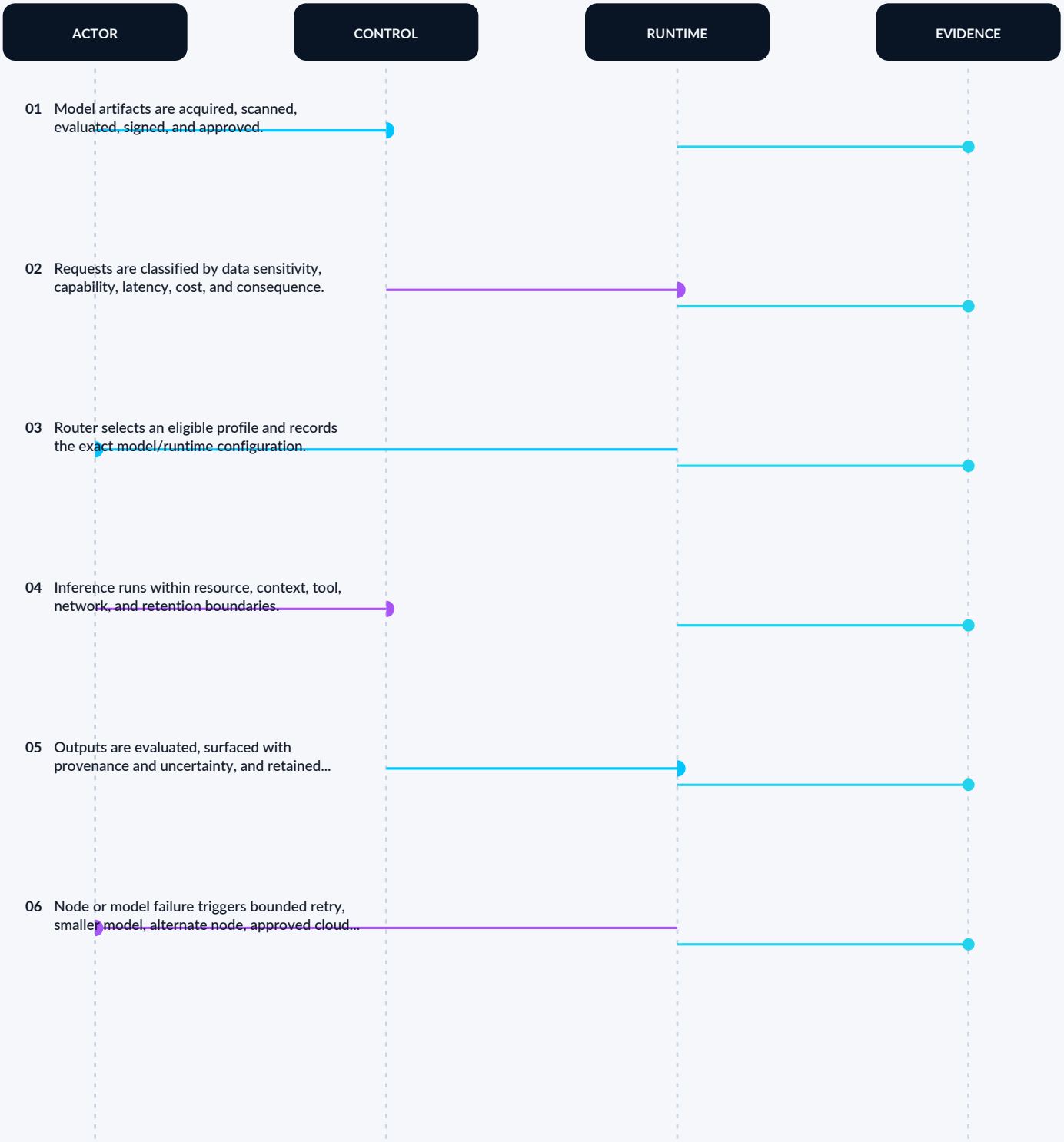
LAYER 06
Observation, Safety, and Recovery

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
Untrusted model artifact versus trusted registry

TB-02
User data versus shared model service

TB-03
Inference runtime versus host operating system

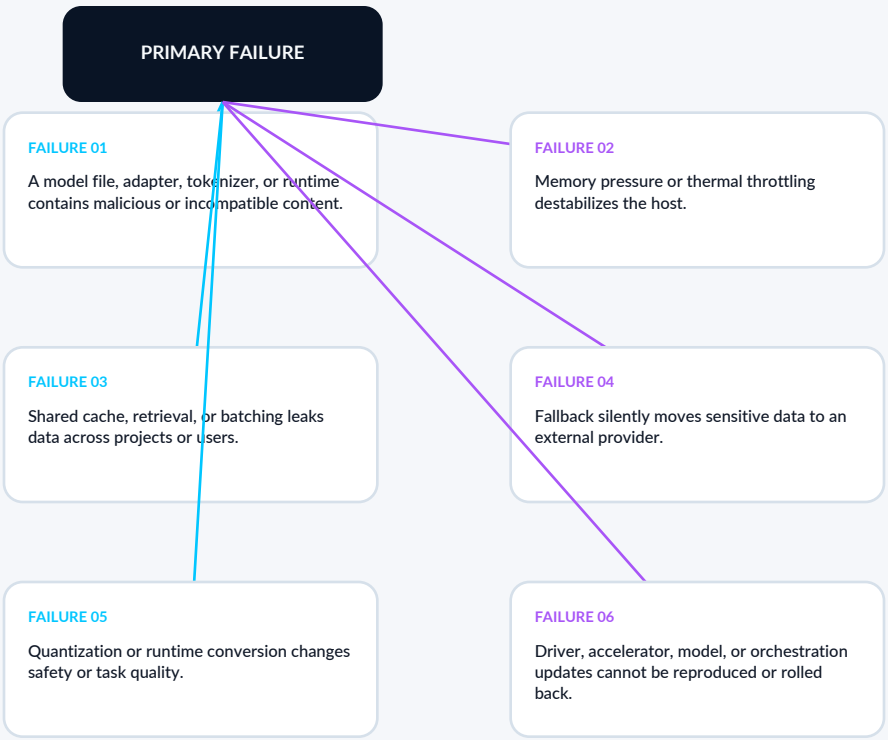
TB-04
Model process versus tools and network

TB-05
Local cluster versus approved external provider

TB-06
Tenant or project context versus shared cache and memory

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 — SINGLE ACCELERATED WORKSTATION

Bounded proof

PROFILE 02 — MULTI-GPU LOCAL INFERENCE SERVER

Team-scale governance

PROFILE 03 — SMALL EDGE CLUSTER

Sovereign / high-assurance

PROFILE 04 — FEDERATED LOCAL-FIRST FLEET WITH APPROVED CLOUD OVERFLOW

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-AI-0003, MYTHOS-AI-0005, MYTHOS-AI-0006, MYTHOS-DAT-0001, MYTHOS-HW-0001, MYTHOS-WEB-0001

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 local-first architecture for model acquisition, inference, routing, isolation, acceleration, retrieval, observability, privacy, and resilient edge operation.

EXECUTIVE OUTCOMES

- Run approved models near the user or data when privacy, latency, resilience, or cost justify it.
- Route workloads across CPU, GPU, NPU, accelerator, workstation, cluster, and approved cloud resources.
- Protect model, data, prompt, context, cache, and tool boundaries.
- Operate through resource pressure, model failure, offline periods, and node loss.

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

ARTIFICIAL INTELLIGENCE

IN SCOPE

A local-first architecture for model acquisition, inference, routing, isolation, acceleration, retrieval, observability, privacy, and resilient edge operation.

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

Local-first means local capability is a first-class operating mode; it does not imply that every model, dataset, or workload must run locally or that cloud services are prohibited.

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01

Model Registry and Supply Chain

Model Registry and Supply Chain 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

Scheduling and Model Routing

Scheduling and Model Routing 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

Inference Runtime and Isolation

Inference Runtime and Isolation 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

Data, Retrieval, and Context

Data, Retrieval, and Context 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

Acceleration, Capacity, and Thermal Control

Acceleration, Capacity, and Thermal 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, Safety, and Recovery

Observation, Safety, and Recovery 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

Signed model, tokenizer, adapter, runtime, license, and evaluation registry

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

COMPONENT 02

Capability profiles, workload queues, budgets, placement, admission, and fallback policy

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

COMPONENT 03

Sandboxed runtimes, local APIs, batching, streaming, quantization, and resource limits

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Local retrieval, vector and structured stores, ephemeral context, cache, and privacy policy

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

COMPONENT 05

GPU/NPU/CPU topology, memory planning, power, cooling, driver, and health management

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

COMPONENT 06

Quality evaluation, guardrails, telemetry, incident handling, node rebuild, and evidence

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. Model artifacts are acquired, scanned, evaluated, signed, and approved.

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

2. Requests are classified by data sensitivity, capability, latency, cost, and consequence.

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

3. Router selects an eligible profile and records the exact model/runtime configuration.

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

4. Inference runs within resource, context, tool, network, and retention boundaries.

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

5. Outputs are evaluated, surfaced with provenance and uncertainty, and retained according to policy.

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

6. Node or model failure triggers bounded retry, smaller model, alternate node, approved cloud, or safe refusal.

   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

Untrusted model artifact versus trusted registry

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

TRUST BOUNDARY 02

User data versus shared model service

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

TRUST BOUNDARY 03

Inference runtime versus host operating system

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

TRUST BOUNDARY 04

Model process versus tools and 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

Local cluster versus approved external provider

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

TRUST BOUNDARY 06

Tenant or project context versus shared cache and memory

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 / SINGLE ACCELERATED WORKSTATION

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 / MULTI-GPU LOCAL INFERENCE SERVER

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 / SMALL EDGE CLUSTER

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 / FEDERATED LOCAL-FIRST FLEET WITH APPROVED CLOUD OVERFLOW

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 / Inventory workloads, data, and hardware

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 / Establish signed model registry

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 / Deploy isolated runtime and API

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 / Add routing and resource governance

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 / Integrate retrieval and tools 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.

06 / Exercise failure and offline modes

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 / Optimize capacity with measured evidence

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

A model file, adapter, tokenizer, or runtime contains malicious or incompatible content.

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

FAILURE SCENARIO 02

Memory pressure or thermal throttling destabilizes the host.

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

FAILURE SCENARIO 03

Shared cache, retrieval, or batching leaks data across projects or users.

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

FAILURE SCENARIO 04

Fallback silently moves sensitive data to an external provider.

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

FAILURE SCENARIO 05

Quantization or runtime conversion changes safety or task quality.

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

FAILURE SCENARIO 06

Driver, accelerator, model, or orchestration updates cannot be reproduced or rolled back.

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-008 - Multi-Cloud Landing Zone and Internal Developer Platform
- RA-010 - Local-First Knowledge, Memory, and Evidence Platform
- RA-016 - Realtime Voice, Media, and Multimodal Interaction Platform
- RA-017 - Quantum-Safe, Confidential, and Verifiable Compute

MAPPED MYTHOS STANDARDS

- MYTHOS-AI-0003
- MYTHOS-AI-0005
- MYTHOS-AI-0006
- MYTHOS-DAT-0001
- MYTHOS-HW-0001
- MYTHOS-WEB-0001

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 served profile identifies model, tokenizer, adapter, runtime, quantization, hardware, driver, safety configuration, license, and evaluation state.
AC-14	Routing policy proves that sensitive requests cannot cross an unapproved provider, tenant, cache, or network boundary.
AC-15	Admission control prevents memory, power, thermal, storage, and queue saturation from destabilizing critical workloads.
AC-16	The platform exposes model cold-start, load, queue, token, latency, quality, failure, resource, and fallback telemetry.
AC-17	Offline operation, node loss, accelerator loss, and approved cloud overflow are exercised with representative tasks.
AC-18	Model retirement removes serving authority, cache, indexes, credentials, and retained data without deleting required evidence.

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 **NIST AI Risk Management Framework 1.0 (revision in progress as of publication date)**

https://www.nist.gov/itl/ai-risk-management-framework

SOURCE 02 **NIST AI 600-1, Generative AI Profile**

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

SOURCE 03 **NIST SP 800-53 Rev. 5**

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

SOURCE 04 **SLSA Specification 1.2**

https://slsa.dev/spec/v1.2/

SOURCE 05 **Kubernetes Documentation**

https://kubernetes.io/docs/

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-006
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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

AI-Assisted Software Engineering Pipeline

A governed engineering pipeline in which AI systems assist research, design, coding, testing, review, documentation, release, and operations without bypassing software assurance.

PERMANENT DOCUMENT ID
RA-007

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Canonical Set

Architecture identity and operating position

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

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Convert product intent into traceable requirements, architecture decisions, tasks, code, tests, evidence, and
2. release artifacts.
3. Use agents and models under repository, tool, secret, network, budget, and branch boundaries.
4. Verify generated work through compilers, tests, analyzers, review, runtime observation, and acceptance criteria.

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
Intent, Requirements, and Architecture

LAYER 02
Developer Workspace and Agent Harness

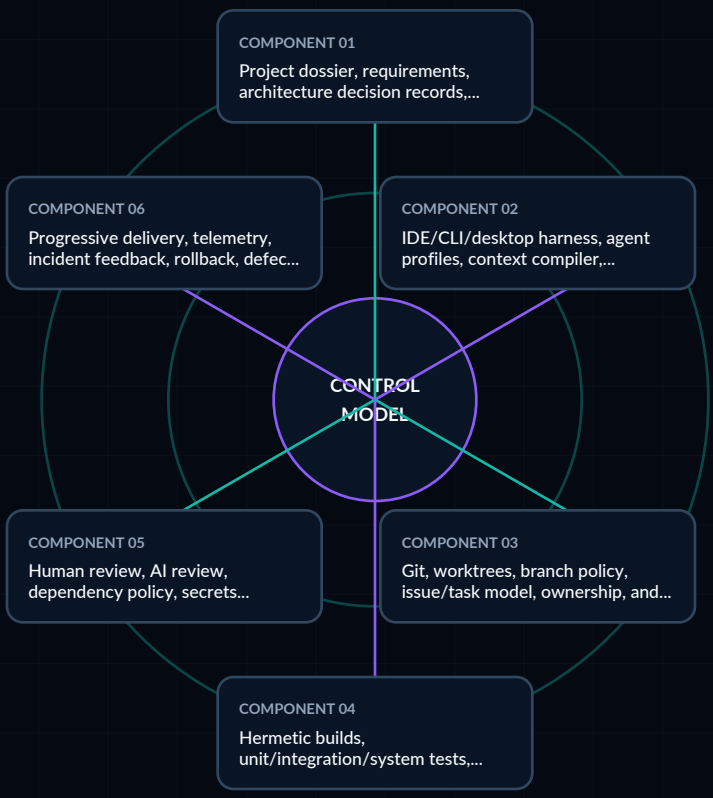
LAYER 03
Source, Branch, and Change Management

LAYER 04
Build, Test, and Analysis

LAYER 05
Review, Security, and Release Policy

LAYER 06
Deployment, Observation, and Learning

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
Model context versus confidential repository content

TB-02
Agent tool access versus developer workstation and network

TB-03
Generated change versus protected branch

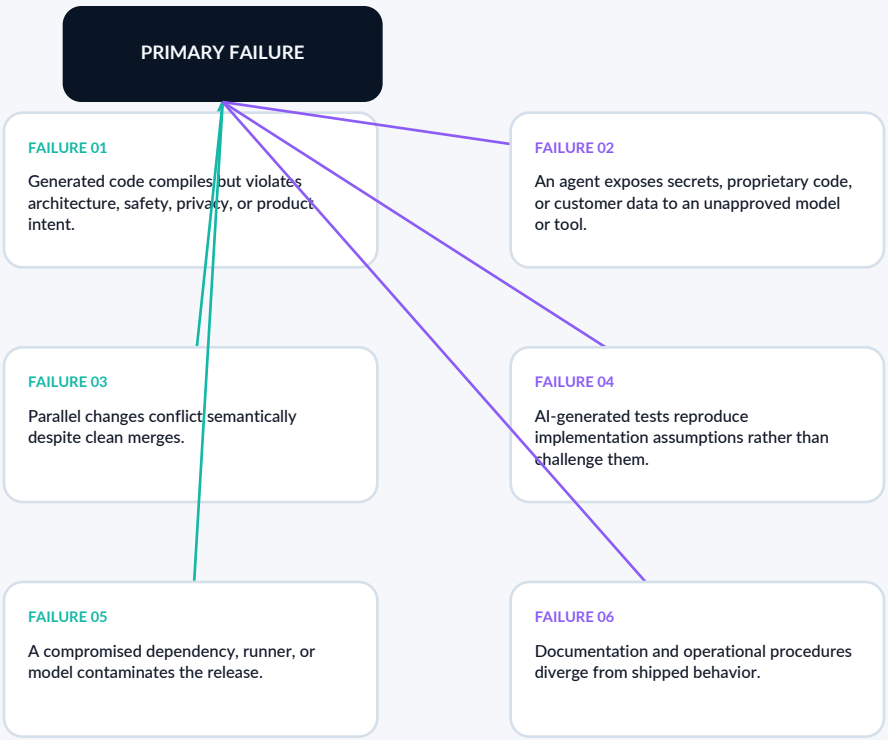
TB-04
Build environment versus release signing authority

TB-05
AI review versus accountable human approval

TB-06
Deployment automation versus production control plane

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 — **INDIVIDUAL DEVELOPER ASSISTANT**

Bounded proof

PROFILE 02 — **TEAM ENGINEERING PIPELINE**

Team-scale governance

PROFILE 03 — **REGULATED PRODUCT RELEASE FACTORY**

Sovereign / high-assurance

PROFILE 04 — **LARGE-SCALE MULTI-AGENT SOFTWARE DELIVERY PLATFORM**

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-SWE-0001, MYTHOS-SWE-0004, MYTHOS-SWE-0008, MYTHOS-PLT-0001, MYTHOS-PLT-0002, MYTHOS-AI-0007

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 governed engineering pipeline in which AI systems assist research, design, coding, testing, review, documentation, release, and operations without bypassing software assurance.

EXECUTIVE OUTCOMES

- Convert product intent into traceable requirements, architecture decisions, tasks, code, tests, evidence, and release artifacts.
- Use agents and models under repository, tool, secret, network, budget, and branch boundaries.
- Verify generated work through compilers, tests, analyzers, review, runtime observation, and acceptance criteria.
- Produce signed, reproducible, attributable releases with safe rollback and operational feedback.

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

PLATFORM ENGINEERING

IN SCOPE

A governed engineering pipeline in which AI systems assist research, design, coding, testing, review, documentation, release, and operations without bypassing software assurance.

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

AI accelerates engineering work but does not replace source control, reproducible builds, testing, security gates, review, ownership, or release authority.

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01

Intent, Requirements, and Architecture

Intent, Requirements, and Architecture 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

Developer Workspace and Agent Harness

Developer Workspace and Agent Harness 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

Source, Branch, and Change Management

Source, Branch, and Change Management 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

Build, Test, and Analysis

Build, Test, and 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

Review, Security, and Release Policy

Review, Security, and Release Policy 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

Deployment, Observation, and Learning

Deployment, Observation, and Learning 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

Project dossier, requirements, architecture decision records, threat model, and acceptance tests

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

COMPONENT 02

IDE/CLI/desktop harness, agent profiles, context compiler, sandbox, and tool broker

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

COMPONENT 03

Git, worktrees, branch policy, issue/task model, ownership, and change provenance

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Hermetic builds, unit/integration/system tests, static/dynamic analysis, fuzzing, and artifact generation

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

COMPONENT 05

Human review, AI review, dependency policy, secrets scanning, SBOM, provenance, signing, and release gates

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

COMPONENT 06

Progressive delivery, telemetry, incident feedback, rollback, defect learning, and documentation maintenance

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 is decomposed into traceable work and testable outcomes.

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

2. Agents receive bounded repository views, tools, credentials, network access, time, and token budgets.

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

3. Changes occur in isolated branches or worktrees with continuous compile and test feedback.

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

4. Independent verification challenges implementation, security, UX, performance, and documentation.

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

5. Release factory produces signed artifacts, attestations, SBOMs, deployment plans, and rollback evidence.

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

6. Production signals create defects, tests, risk updates, and controlled future work.

   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

Model context versus confidential repository content

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

TRUST BOUNDARY 02

Agent tool access versus developer workstation and network

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

TRUST BOUNDARY 03

Generated change versus protected branch

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

TRUST BOUNDARY 04

Build environment versus release signing authority

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

TRUST BOUNDARY 05

AI review versus accountable human approval

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

TRUST BOUNDARY 06

Deployment automation versus production control plane

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 / INDIVIDUAL DEVELOPER ASSISTANT

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 / TEAM ENGINEERING 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 03 / REGULATED PRODUCT RELEASE FACTORY

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 / LARGE-SCALE MULTI-AGENT SOFTWARE DELIVERY PLATFORM

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 / Establish project dossier and repository policy

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 / Add isolated assistant and read-only analysis

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 / Enable bounded change generation

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 / Strengthen automated verification

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 provenance and signed releases

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 / Integrate progressive delivery and telemetry

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 parallel agents under measured governance

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

Generated code compiles but violates architecture, safety, privacy, or product intent.

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

FAILURE SCENARIO 02

An agent exposes secrets, proprietary code, or customer data to an unapproved model or tool.

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

FAILURE SCENARIO 03

Parallel changes conflict semantically despite clean merges.

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

FAILURE SCENARIO 04

AI-generated tests reproduce implementation assumptions rather than challenge them.

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

FAILURE SCENARIO 05

A compromised dependency, runner, or model contaminates the release.

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

FAILURE SCENARIO 06

Documentation and operational procedures diverge from shipped behavior.

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-008 - Multi-Cloud Landing Zone and Internal Developer Platform
- RA-009 - Zero-Trust Identity, Policy, Secrets, and Access Fabric
- RA-010 - Local-First Knowledge, Memory, and Evidence Platform
- RA-013 - Secure Software Supply Chain and Release Factory
- RA-014 - Enterprise Data, API, Event, and Integration Fabric
- RA-019 - Sovereign Browser, Remote Access, and Web Automation
- RA-020 - Adaptive Learning, Cyber Range, and Workforce Assurance

MAPPED MYTHOS STANDARDS

- MYTHOS-SWE-0001
- MYTHOS-SWE-0004
- MYTHOS-SWE-0008
- MYTHOS-PLT-0001
- MYTHOS-PLT-0002
- MYTHOS-AI-0007

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 generated change is attributable to a mission, model profile, context set, tool grant, branch, authorizing user, and verification record.
AC-14	Agents cannot access secrets, protected branches, production, or unrestricted networks by default.
AC-15	Acceptance tests originate from requirements and risk analysis, not solely from generated implementation.
AC-16	Builds are reproducible or explain why they are not; release artifacts include provenance, SBOM, signatures, and policy results.
AC-17	Independent review can reject, amend, or request evidence without being silently overridden by the generating agent.
AC-18	Production incidents and defects create regression tests and architecture updates before the same class of change is retried.

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 **NIST SP 800-218, Secure Software Development Framework**

https://csrc.nist.gov/pubs/sp/800/218/final

SOURCE 02 **CISA Secure by Design**

https://www.cisa.gov/resources-tools/resources/secure-by-design

SOURCE 03 **SLSA Specification 1.2**

https://slsa.dev/spec/v1.2/

SOURCE 04 **NIST AI Risk Management Framework 1.0 (revision in progress as of publication date)**

https://www.nist.gov/itl/ai-risk-management-framework

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

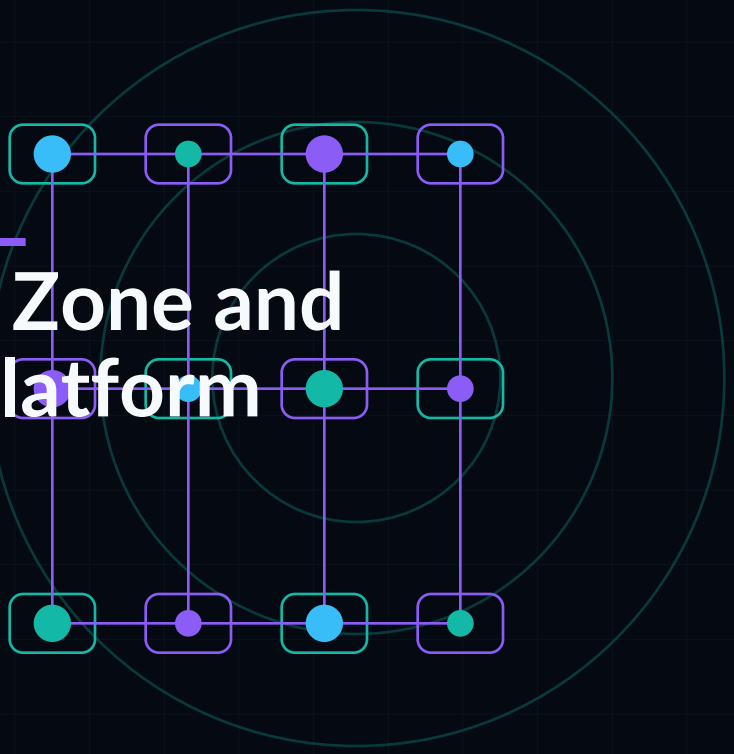
PERMANENT DOCUMENT IDENTITY	RA-007
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Canonical Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Multi-Cloud Landing Zone and Internal Developer Platform



A governed multi-cloud foundation and product-oriented developer platform that standardizes identity, policy, networking, data, delivery, observability, cost, and service ownership.

PERMANENT DOCUMENT ID
RA-008

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

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

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for multi-cloud landing zone and internal developer
2. platform.
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
Organization and tenancy

LAYER 02
Landing zones and shared services

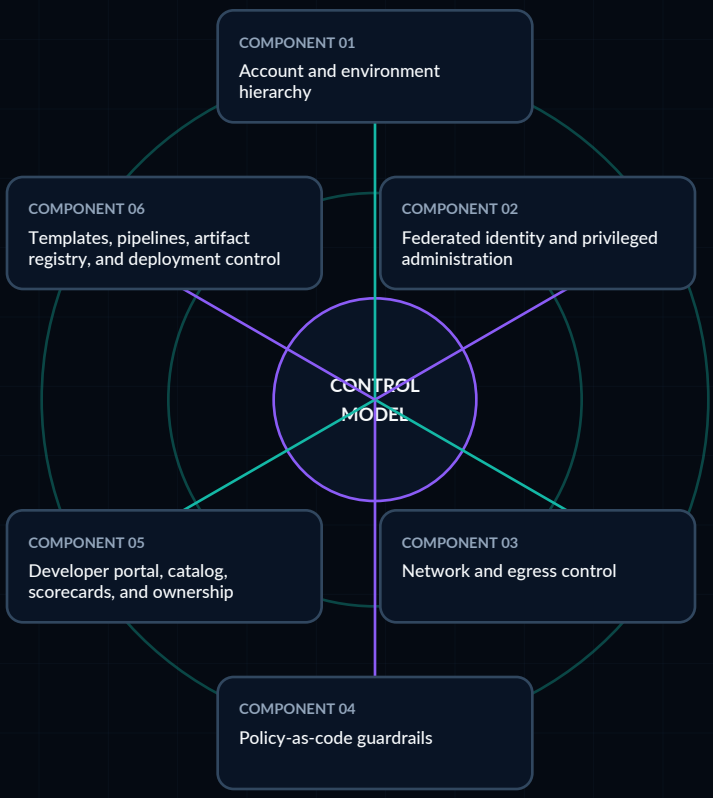
LAYER 03
Developer portal and service catalog

LAYER 04
Golden paths and templates

LAYER 05
Workload identity and policy

LAYER 06
Delivery, observability, FinOps, and exit

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

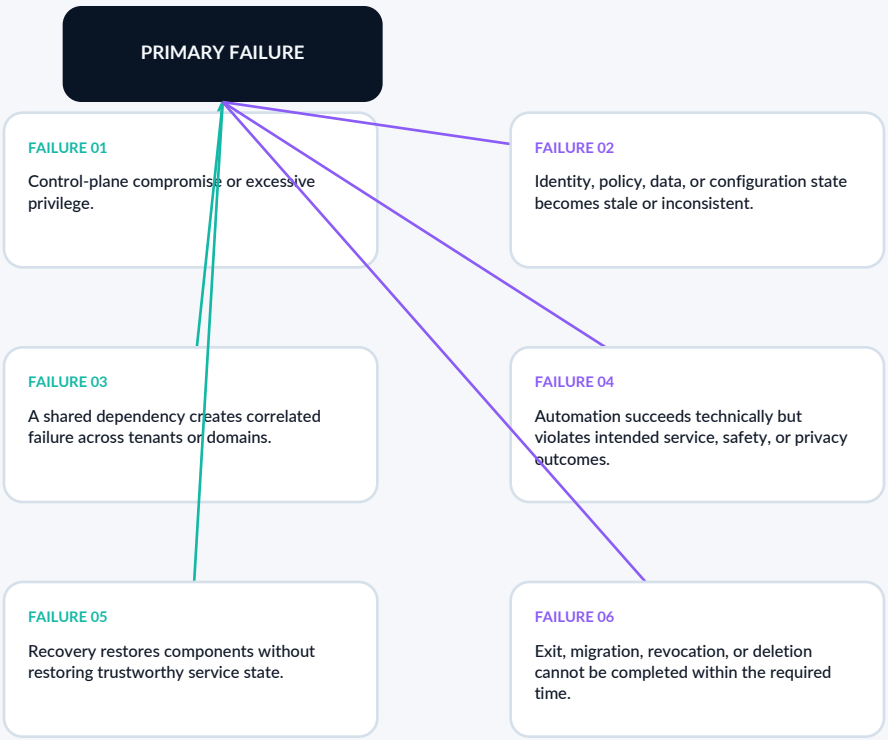
TB-04
Local environment versus external provider or network

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-CLD-0001, MYTHOS-CLD-0007, MYTHOS-PLT-0002, MYTHOS-SRE-0001

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 governed multi-cloud foundation and product-oriented developer platform that standardizes identity, policy, networking, data, delivery, observability, cost, and service ownership.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for multi-cloud landing zone and internal developer platform.
- 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

PLATFORM ENGINEERING

IN SCOPE

A governed multi-cloud foundation and product-oriented developer platform that standardizes identity, policy, networking, data, delivery, observability, cost, and service ownership.

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

Organization and tenancy

Organization and tenancy 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

Landing zones and shared services

Landing zones and shared services 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

Developer portal and service catalog

Developer portal and service catalog 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

Golden paths and templates

Golden paths and templates 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

Workload identity and policy

Workload identity and policy 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

Delivery, observability, FinOps, and exit

Delivery, observability, FinOps, and exit 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 01-04

COMPONENT 01

Account and environment hierarchy

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

COMPONENT 02

Federated identity and privileged administration

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

COMPONENT 03

Network and egress control

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

COMPONENT 04

Policy-as-code guardrails

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

CORE COMPONENTS - B

COMPONENTS 05-07

COMPONENT 05

Developer portal, catalog, scorecards, and ownership

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

COMPONENT 06

Templates, pipelines, artifact registry, and deployment control

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

COMPONENT 07

Telemetry, SLO, cost, quota, recovery, and provider exit

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-009 - Zero-Trust Identity, Policy, Secrets, and Access Fabric
- RA-012 - Enterprise Observability, SRE, and Incident Command Platform
- RA-013 - Secure Software Supply Chain and Release Factory
- RA-014 - Enterprise Data, API, Event, and Integration Fabric

MAPPED MYTHOS STANDARDS

- MYTHOS-CLD-0001
- MYTHOS-CLD-0007
- MYTHOS-PLT-0002
- MYTHOS-SRE-0001

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 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

SOURCE 02 **NIST SP 800-218, Secure Software Development Framework**

https://csrc.nist.gov/pubs/sp/800/218/final

SOURCE 03 **SLSA Specification 1.2**

https://slsa.dev/spec/v1.2/

SOURCE 04 **Kubernetes Documentation**

https://kubernetes.io/docs/

SOURCE 05 **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-008
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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Zero-Trust Identity, Policy, Secrets, and Access Fabric

A unified identity and authorization architecture for humans, workloads, agents, devices, services, privileged sessions, policy, credentials, secrets, and continuous trust decisions.

PERMANENT DOCUMENT ID
RA-009

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Cybersecurity and Network Security	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for zero-trust identity, policy, secrets, and access
2. fabric.
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
Identity sources and proofing

LAYER 02
Credential and authenticator lifecycle

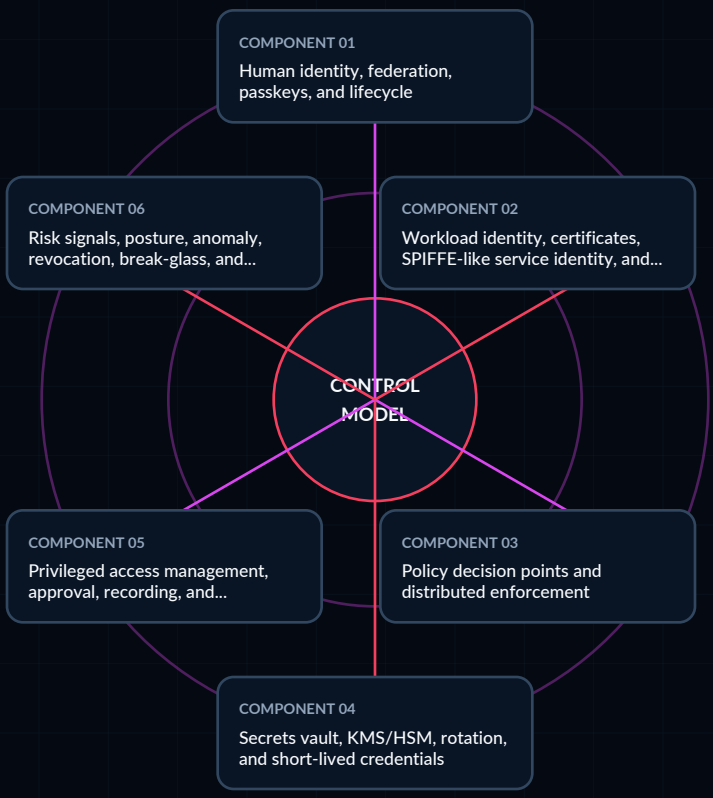
LAYER 03
Workload and device identity

LAYER 04
Policy decision and enforcement

LAYER 05
Secrets and cryptographic services

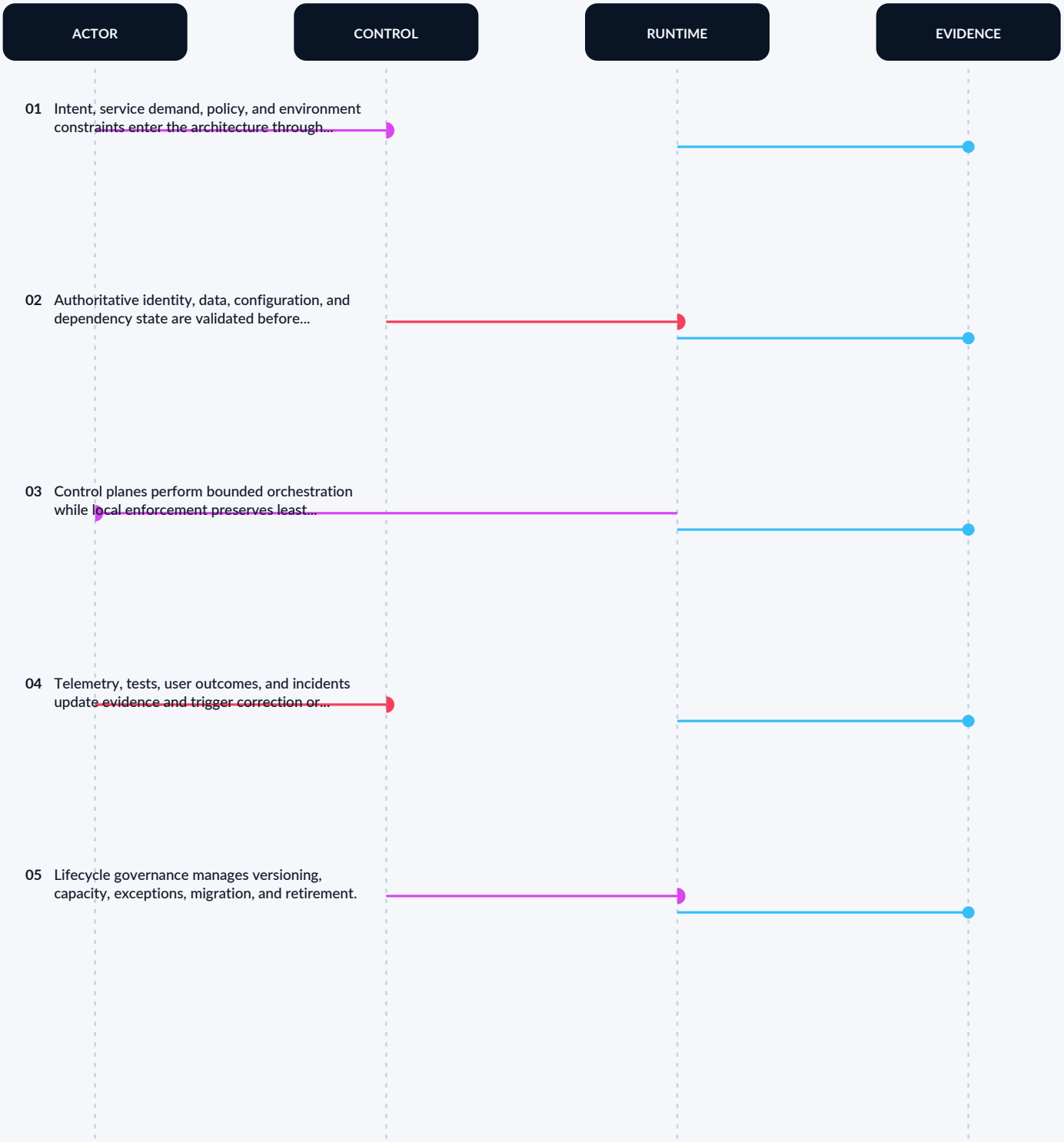
LAYER 06
Privileged access, session evidence, and recovery

CORE COMPONENTS



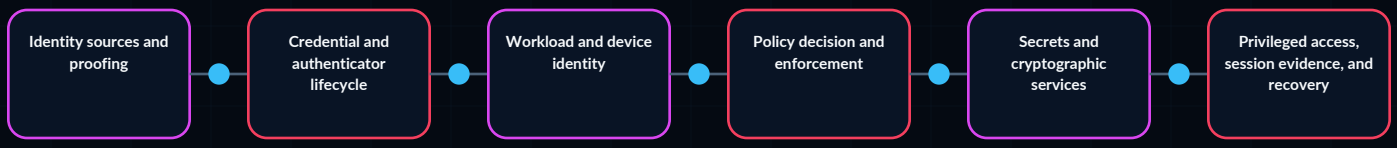
Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

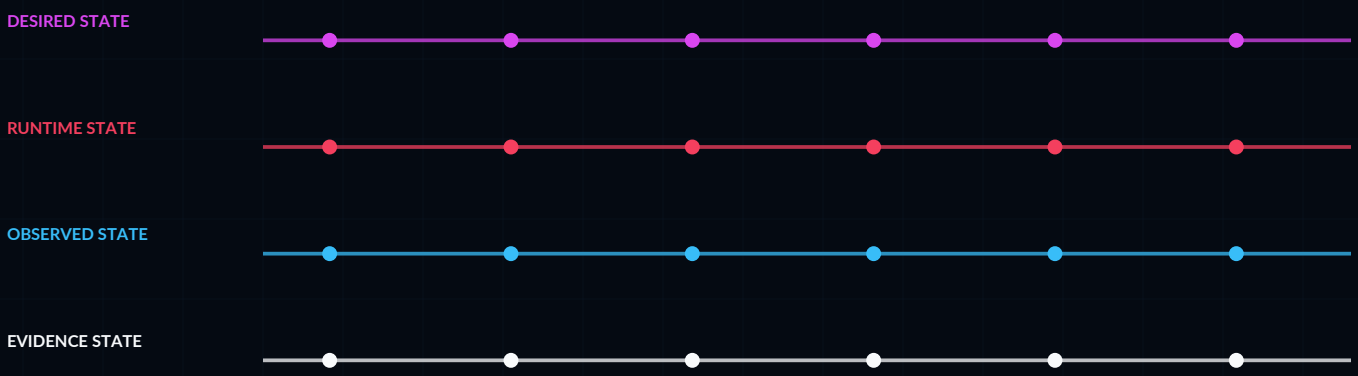
Primary sequence and control flow



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

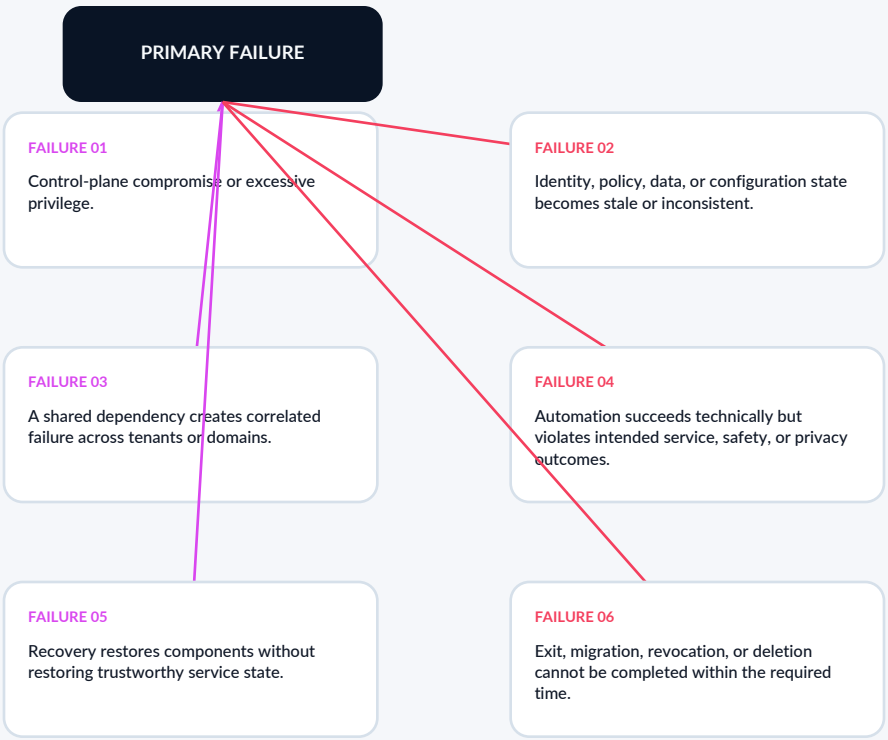
TB-04
Local environment versus external provider or network

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

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-ID-0001, MYTHOS-ID-0002, MYTHOS-ID-0003, MYTHOS-SEC-0003

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 unified identity and authorization architecture for humans, workloads, agents, devices, services, privileged sessions, policy, credentials, secrets, and continuous trust decisions.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for zero-trust identity, policy, secrets, and access fabric.
- 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

CYBERSECURITY AND NETWORK SECURITY

IN SCOPE

A unified identity and authorization architecture for humans, workloads, agents, devices, services, privileged sessions, policy, credentials, secrets, and continuous trust decisions.

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

Identity sources and proofing

Identity sources and proofing 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

Credential and authenticator lifecycle

Credential and authenticator lifecycle 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

Workload and device identity

Workload and device identity 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

Policy decision and enforcement

Policy decision and enforcement 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

Secrets and cryptographic services

Secrets and cryptographic services 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

Privileged access, session evidence, and recovery

Privileged access, session evidence, and recovery 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

Human identity, federation, passkeys, and lifecycle

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

COMPONENT 02

Workload identity, certificates, SPIFFE-like service identity, and attestation

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

COMPONENT 03

Policy decision points and distributed enforcement

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Secrets vault, KMS/HSM, rotation, and short-lived credentials

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

COMPONENT 05

Privileged access management, approval, recording, and just-in-time elevation

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

COMPONENT 06

Risk signals, posture, anomaly, revocation, break-glass, and evidence

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-019 - Sovereign Browser, Remote Access, and Web Automation

MAPPED MYTHOS STANDARDS

- MYTHOS-ID-0001
- MYTHOS-ID-0002
- MYTHOS-ID-0003
- MYTHOS-SEC-0003

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 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

SOURCE 02 **NIST SP 1800-35, Implementing a Zero Trust Architecture**

https://www.nccoe.nist.gov/projects/implementing-zero-trust-architecture

SOURCE 03 **NIST SP 800-53 Rev. 5**

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

SOURCE 04 **W3C Web Authentication Level 3**

https://www.w3.org/TR/webauthn-3/

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

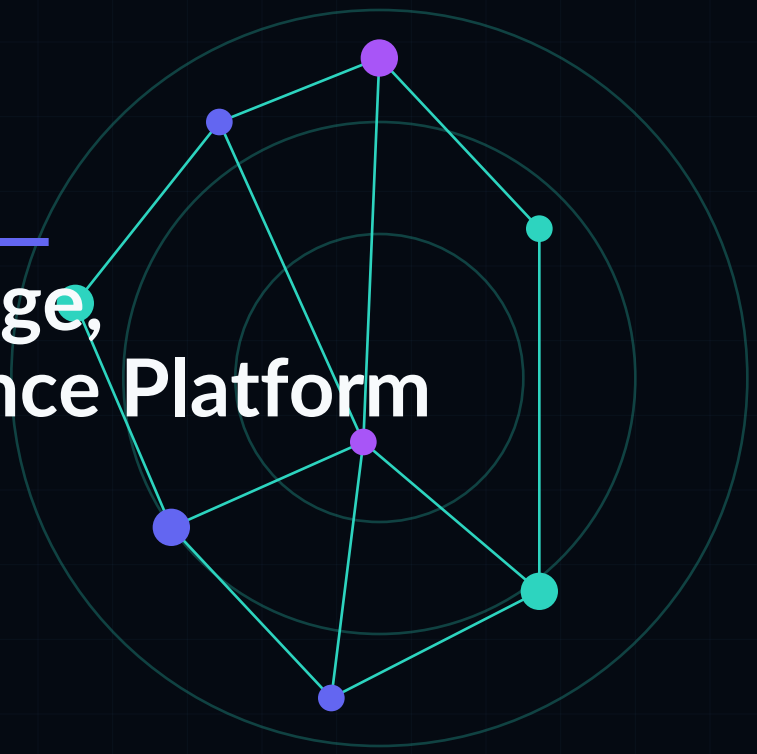
PERMANENT DOCUMENT IDENTITY	RA-009
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Future Domain Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Local-First Knowledge, Memory, and Evidence Platform



A local-first platform for authoritative knowledge, retrieval, context, memory, provenance, evidence, retention, correction, and controlled synchronization.

PERMANENT DOCUMENT ID
RA-010

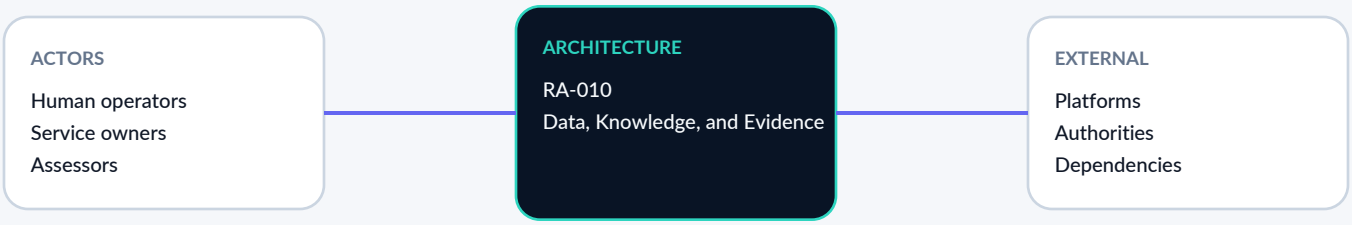
PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Data, Knowledge, and Evidence	6	6	18	0

SYSTEM CONTEXT



EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for local-first knowledge, memory, and evidence
2. platform.
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
Source and authority registry

LAYER 02
Ingestion and transformation

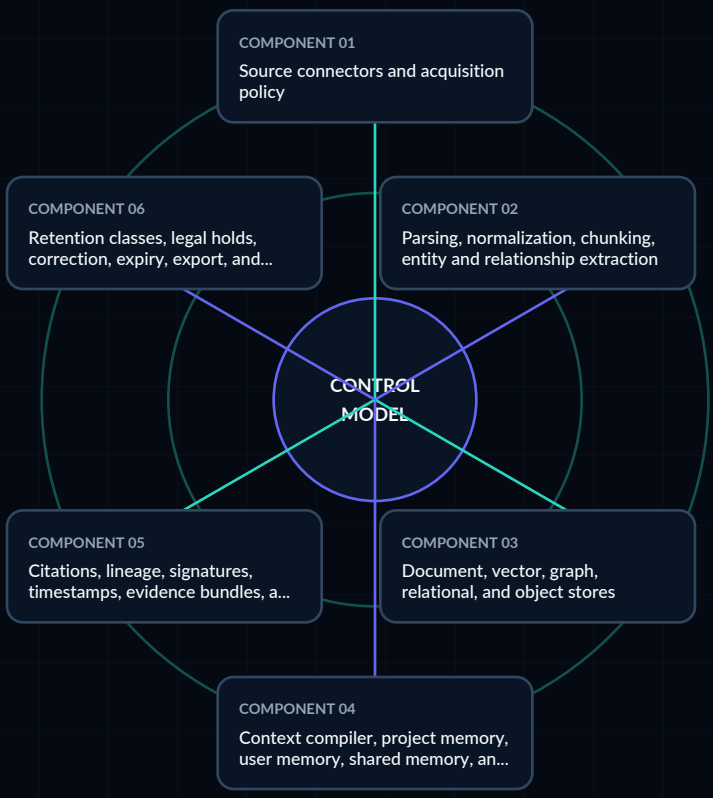
LAYER 03
Knowledge and retrieval stores

LAYER 04
Context and memory services

LAYER 05
Evidence and provenance ledger

LAYER 06
Retention, correction, synchronization, and deletion

CORE COMPONENTS

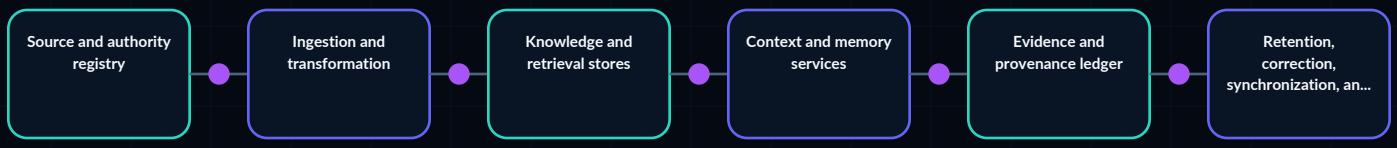


Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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

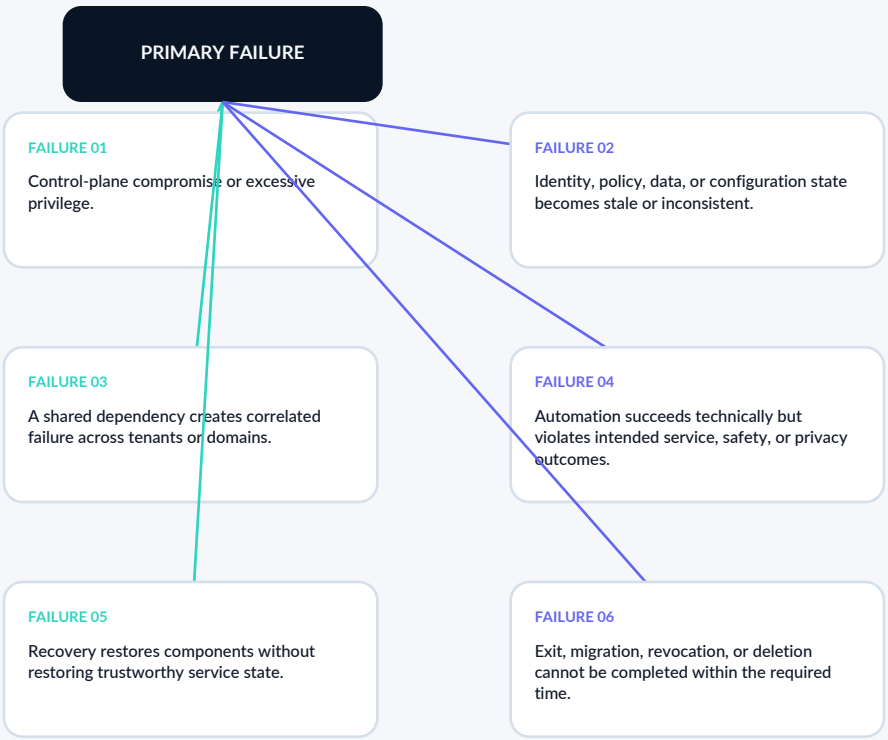
TB-04
Local environment versus external provider or network

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-KNW-0001, MYTHOS-KNW-0002, MYTHOS-DAT-0001, MYTHOS-DAT-0005

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 local-first platform for authoritative knowledge, retrieval, context, memory, provenance, evidence, retention, correction, and controlled synchronization.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for local-first knowledge, memory, and evidence platform.
- 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

DATA, KNOWLEDGE, AND EVIDENCE

IN SCOPE

A local-first platform for authoritative knowledge, retrieval, context, memory, provenance, evidence, retention, correction, and controlled synchronization.

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

Source and authority registry

Source and authority registry 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

Ingestion and transformation

Ingestion and transformation 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

Knowledge and retrieval stores

Knowledge and retrieval stores 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

Context and memory services

Context and memory services 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

Evidence and provenance ledger

Evidence and provenance ledger 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

Retention, correction, synchronization, and deletion

Retention, correction, synchronization, and deletion 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

Source connectors and acquisition policy

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

COMPONENT 02

Parsing, normalization, chunking, entity and relationship extraction

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

COMPONENT 03

Document, vector, graph, relational, and object stores

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Context compiler, project memory, user memory, shared memory, and promotion gates

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

COMPONENT 05

Citations, lineage, signatures, timestamps, evidence bundles, and uncertainty

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

COMPONENT 06

Retention classes, legal holds, correction, expiry, export, and secure deletion

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-014 - Enterprise Data, API, Event, and Integration Fabric
- RA-020 - Adaptive Learning, Cyber Range, and Workforce Assurance

MAPPED MYTHOS STANDARDS

- MYTHOS-KNW-0001
- MYTHOS-KNW-0002
- MYTHOS-DAT-0001
- MYTHOS-DAT-0005

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 **NIST AI Risk Management Framework 1.0 (revision in progress as of publication date)**

https://www.nist.gov/itl/ai-risk-management-framework

SOURCE 02 **NIST SP 800-53 Rev. 5**

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

SOURCE 03 **OpenTelemetry Specification**

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

SOURCE 04 **C2PA Technical Specification**

https://c2pa.org/specifications/

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

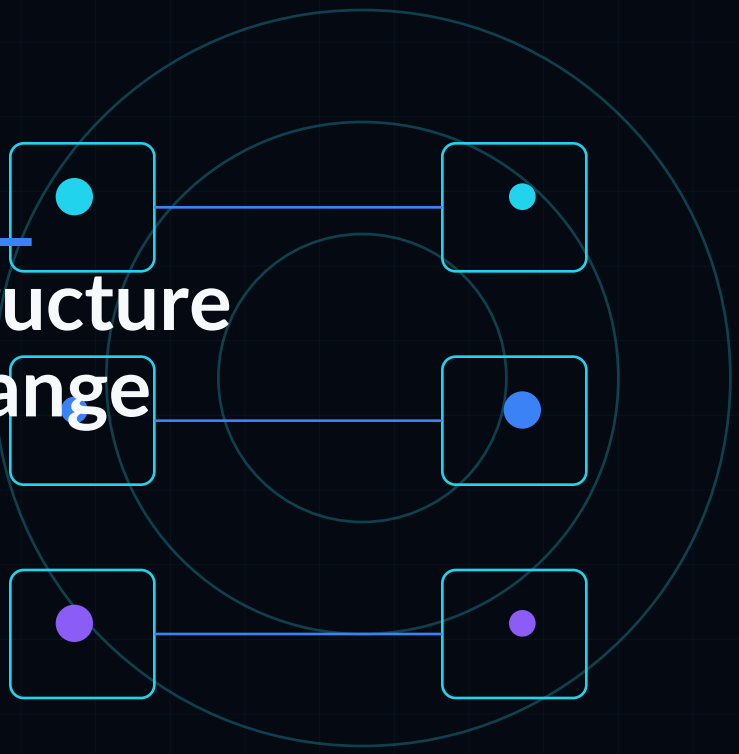
PERMANENT DOCUMENT IDENTITY	RA-010
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Future Domain Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

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

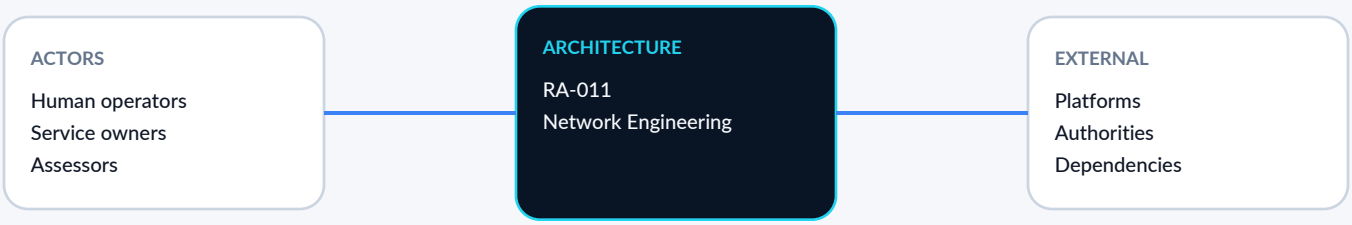
PUBLICATION
Candidate Reference Architecture

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

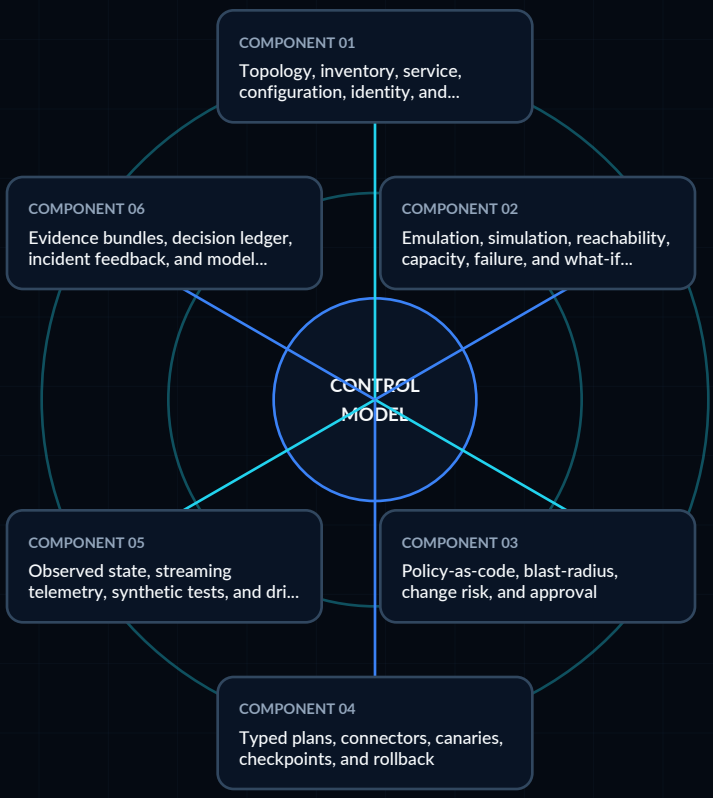
LAYER 03
Policy and risk analysis

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

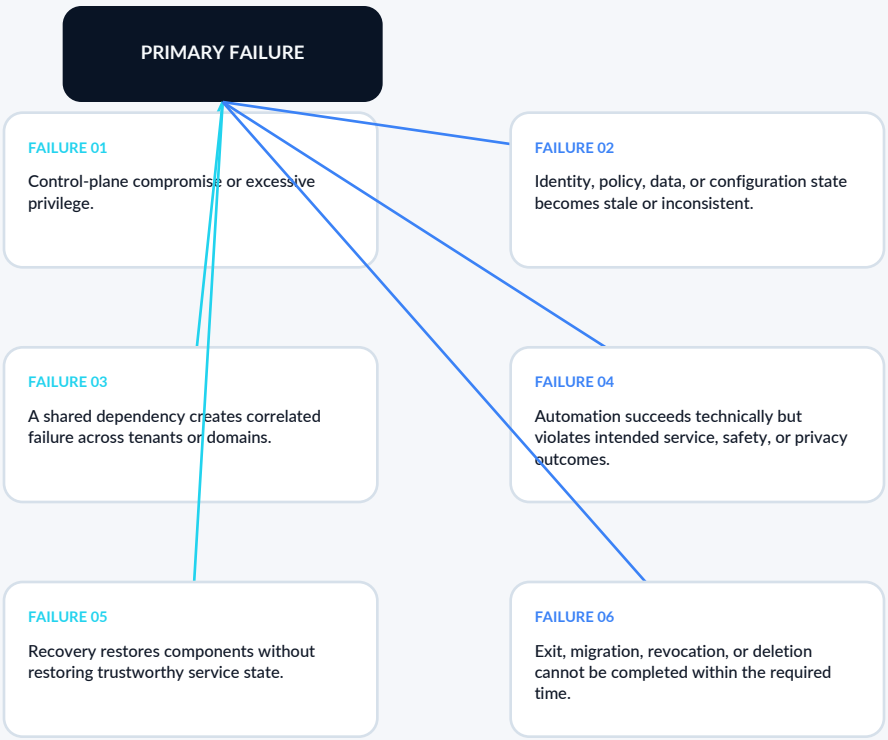
TB-04
Local environment versus external provider or network

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

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 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.

MYTHOS SYSTEMS

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Enterprise Observability, SRE, and Incident Command Platform

An enterprise observability and reliability architecture joining telemetry, SLOs, service ownership, incident command, automation, evidence, and continuous improvement.

PERMANENT DOCUMENT ID
RA-012

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Reliability and Operations	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for enterprise observability, sre, and incident
2. command platform.
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
Telemetry pipeline

LAYER 02
Service catalog and ownership

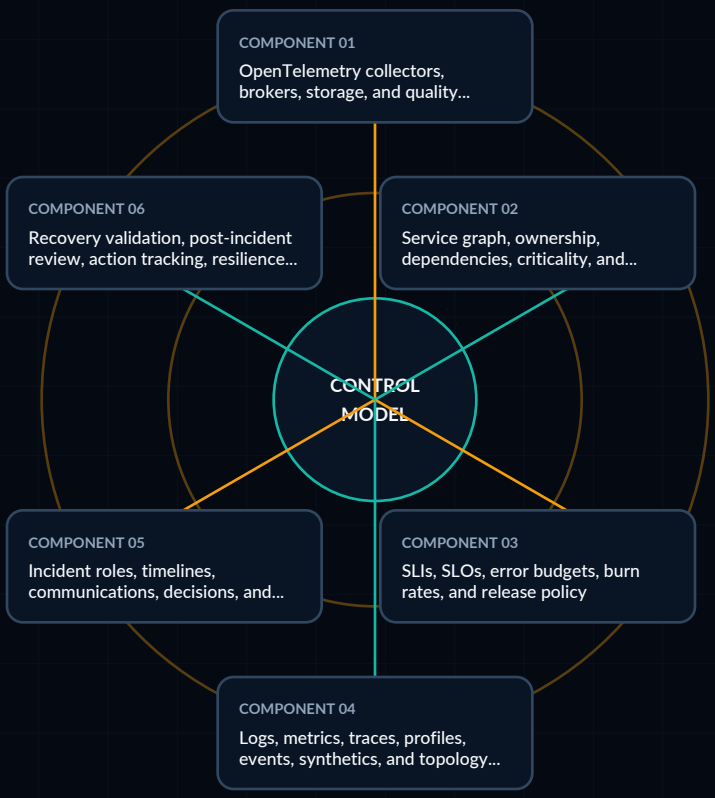
LAYER 03
SLO and error-budget system

LAYER 04
Detection and diagnosis

LAYER 05
Incident command and communications

LAYER 06
Recovery, learning, and reliability governance

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

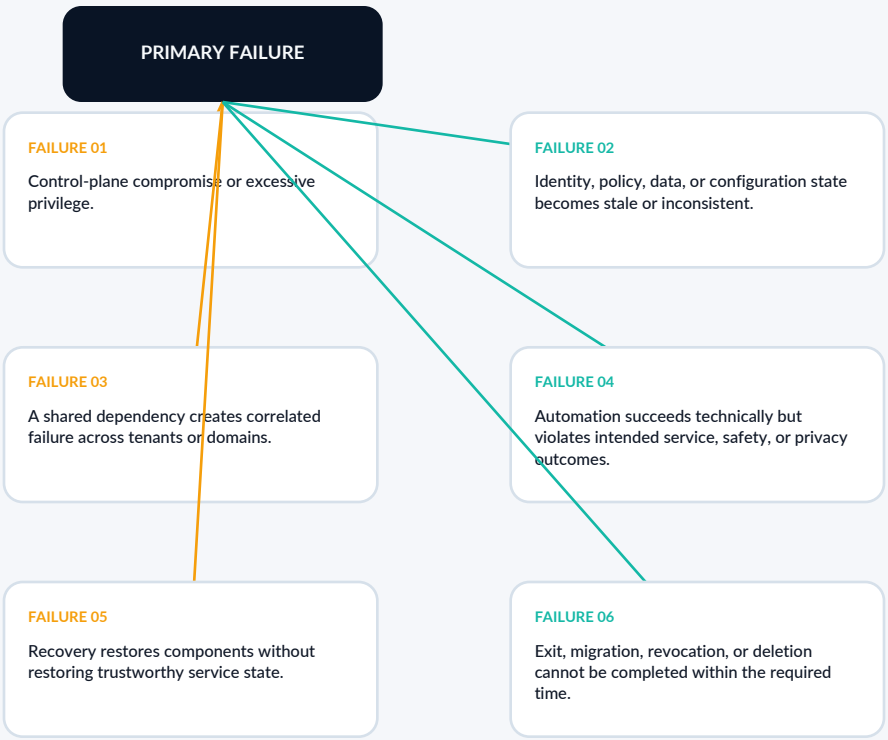
TB-04
Local environment versus external provider or network

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

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-SRE-0001, MYTHOS-SRE-0002, MYTHOS-SRE-0003, MYTHOS-DAT-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

An enterprise observability and reliability architecture joining telemetry, SLOs, service ownership, incident command, automation, evidence, and continuous improvement.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for enterprise observability, sre, and incident command platform.
- 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

RELIABILITY AND OPERATIONS

IN SCOPE

An enterprise observability and reliability architecture joining telemetry, SLOs, service ownership, incident command, automation, evidence, and continuous improvement.

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

Telemetry pipeline

Telemetry pipeline 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

Service catalog and ownership

Service catalog 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

SLO and error-budget system

SLO and error-budget system 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

Detection and diagnosis

Detection and diagnosis 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

Incident command and communications

Incident command and communications 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

Recovery, learning, and reliability governance

Recovery, learning, and reliability governance 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

OpenTelemetry collectors, brokers, storage, and quality controls

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

COMPONENT 02

Service graph, ownership, dependencies, criticality, and runbooks

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

COMPONENT 03

SLIs, SLOs, error budgets, burn rates, and release policy

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Logs, metrics, traces, profiles, events, synthetics, and topology analytics

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

COMPONENT 05

Incident roles, timelines, communications, decisions, and automation

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

COMPONENT 06

Recovery validation, post-incident review, action tracking, resilience testing, and evidence

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-020 - Adaptive Learning, Cyber Range, and Workforce Assurance

MAPPED MYTHOS STANDARDS

- MYTHOS-SRE-0001
- MYTHOS-SRE-0002
- MYTHOS-SRE-0003
- MYTHOS-DAT-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 **OpenTelemetry Specification**

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

SOURCE 02 **NIST Cybersecurity Framework 2.0**

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

SOURCE 03 **NIST SP 800-53 Rev. 5**

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

SOURCE 04 **CISA Secure by Design**

https://www.cisa.gov/resources-tools/resources/secure-by-design

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-012
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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Secure Software Supply Chain and Release Factory

A hardened release factory for source, dependencies, builds, tests, SBOMs, provenance, signatures, policy, promotion, deployment, revocation, and recovery.

PERMANENT DOCUMENT ID
RA-013

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

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

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for secure software supply chain and release factory.
2. Make ownership, trust boundaries, data and control flows, failure handling, and evidence explicit.
3. Support multiple implementation profiles without weakening mandatory assurance outcomes.
4. Provide a governed adoption path from discovery through production operation and retirement.

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
Source and contribution trust

LAYER 02
Dependency and package governance

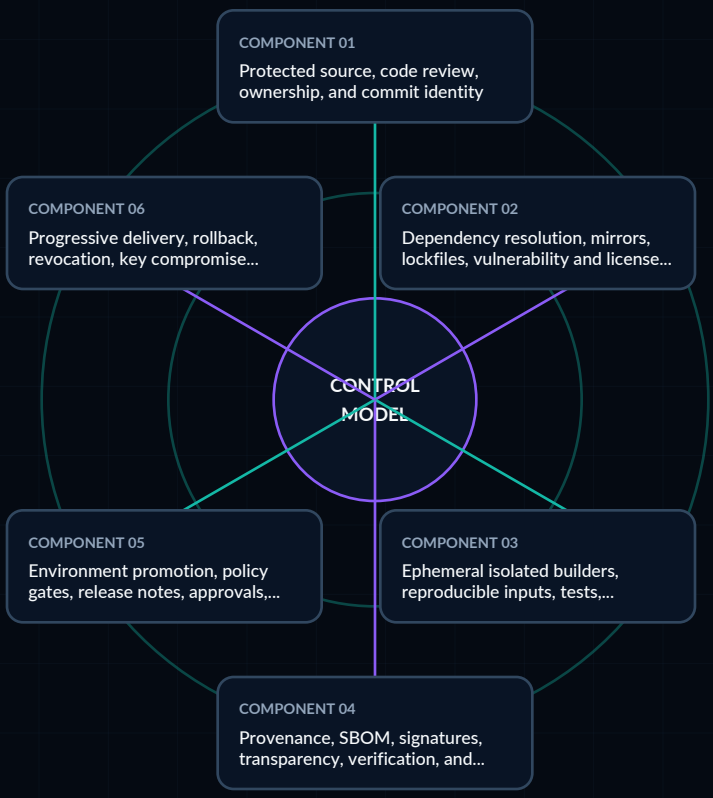
LAYER 03
Hermetic build and test

LAYER 04
Attestation, SBOM, and signing

LAYER 05
Promotion and release policy

LAYER 06
Deployment, revocation, recovery, and evidence

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

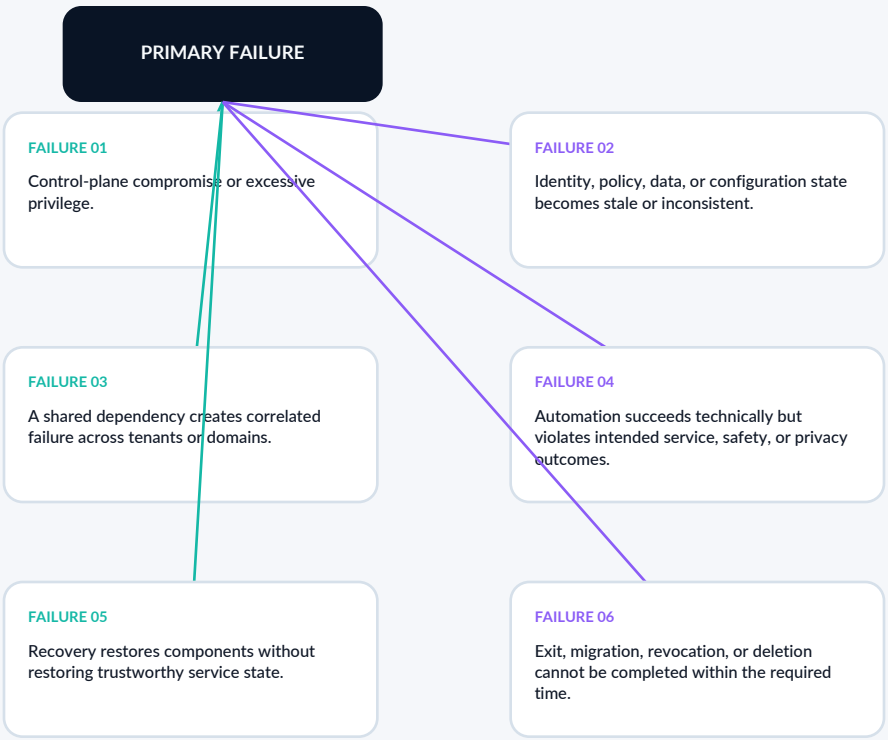
TB-04
Local environment versus external provider or network

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

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-PLT-0001, MYTHOS-SWE-0004, MYTHOS-SWE-0008, MYTHOS-SEC-0015

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 hardened release factory for source, dependencies, builds, tests, SBOMs, provenance, signatures, policy, promotion, deployment, revocation, and recovery.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for secure software supply chain and release factory.
- 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

PLATFORM ENGINEERING

IN SCOPE

A hardened release factory for source, dependencies, builds, tests, SBOMs, provenance, signatures, policy, promotion, deployment, revocation, and recovery.

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

Source and contribution trust

Source and contribution trust 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

Dependency and package governance

Dependency and package governance 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

Hermetic build and test

Hermetic build and test 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

Attestation, SBOM, and signing

Attestation, SBOM, and signing 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

Promotion and release policy

Promotion and release policy 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

Deployment, revocation, recovery, and evidence

Deployment, revocation, recovery, 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

Protected source, code review, ownership, and commit identity

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

COMPONENT 02

Dependency resolution, mirrors, lockfiles, vulnerability and license policy

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

COMPONENT 03

Ephemeral isolated builders, reproducible inputs, tests, scanning, and fuzzing

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Provenance, SBOM, signatures, transparency, verification, and trusted time

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

COMPONENT 05

Environment promotion, policy gates, release notes, approvals, and artifact immutability

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

COMPONENT 06

Progressive delivery, rollback, revocation, key compromise response, and release evidence

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-014 - Enterprise Data, API, Event, and Integration Fabric

MAPPED MYTHOS STANDARDS

- MYTHOS-PLT-0001
- MYTHOS-SWE-0004
- MYTHOS-SWE-0008
- MYTHOS-SEC-0015

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 **NIST SP 800-218, Secure Software Development Framework**

https://csrc.nist.gov/pubs/sp/800/218/final

SOURCE 02 **CISA Secure by Design**

https://www.cisa.gov/resources-tools/resources/secure-by-design

SOURCE 03 **SLSA Specification 1.2**

https://slsa.dev/spec/v1.2/

SOURCE 04 **OCI Distribution Specification**

https://github.com/opencontainers/distribution-spec

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

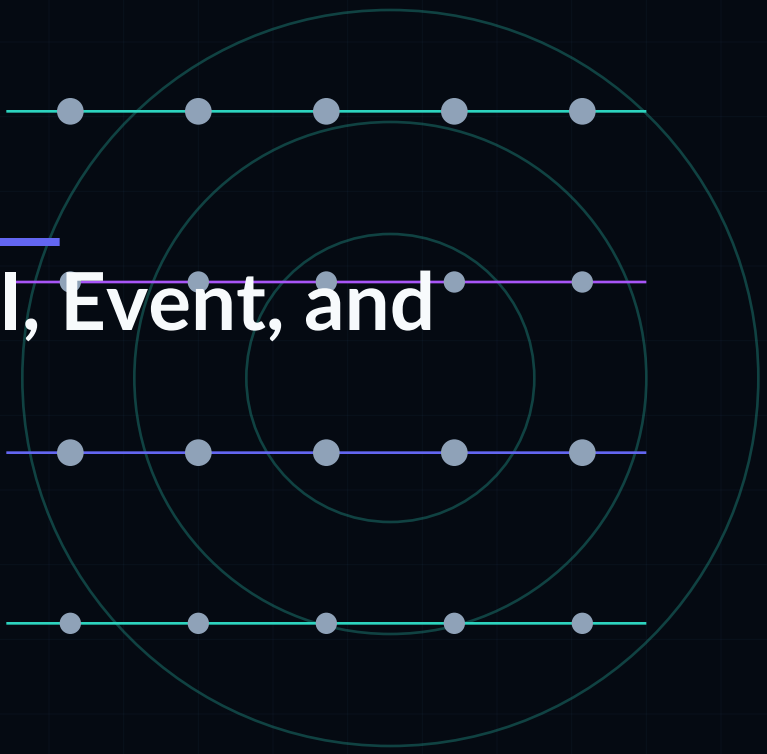
PERMANENT DOCUMENT IDENTITY	RA-013
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Future Domain Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Enterprise Data, API, Event, and Integration Fabric



A governed integration fabric for data products, APIs, events, schemas, workflows, lineage, quality, privacy, observability, and resilient cross-domain exchange.

PERMANENT DOCUMENT ID
RA-014

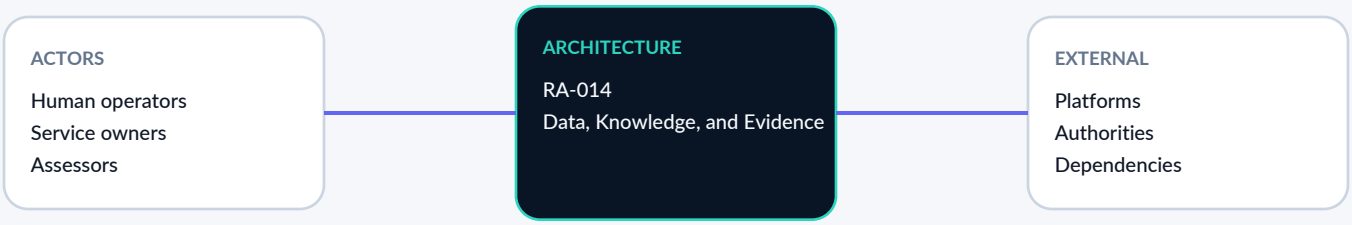
PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Data, Knowledge, and Evidence	6	6	18	0

SYSTEM CONTEXT



EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for enterprise data, api, event, and integration fabric.
2. Make ownership, trust boundaries, data and control flows, failure handling, and evidence explicit.
3. Support multiple implementation profiles without weakening mandatory assurance outcomes.
4. Provide a governed adoption path from discovery through production operation and retirement.

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
Data products and ownership

LAYER 02
API and service interface fabric

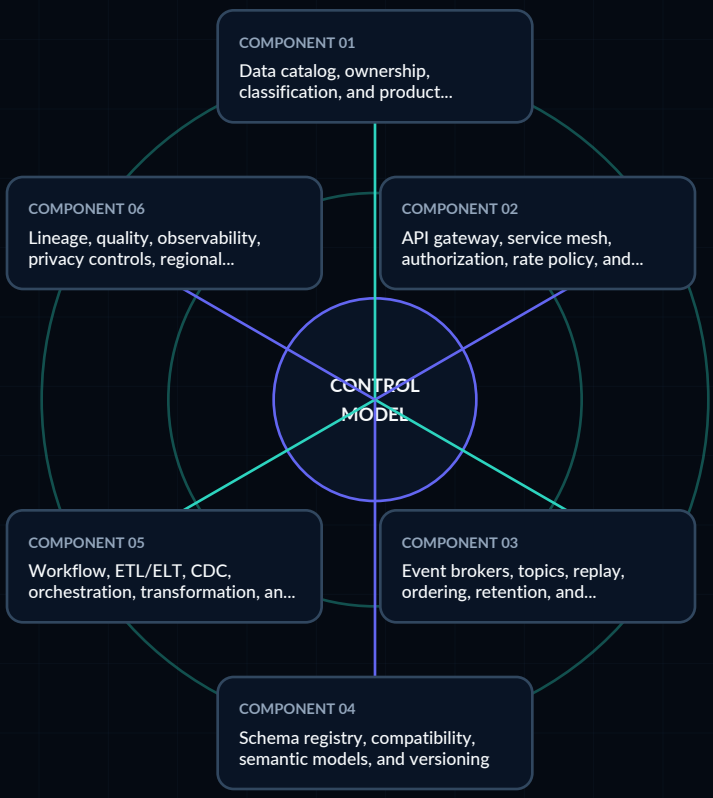
LAYER 03
Event and streaming backbone

LAYER 04
Schema and contract registry

LAYER 05
Integration workflows and transformation

LAYER 06
Lineage, quality, privacy, resilience, and evidence

CORE COMPONENTS

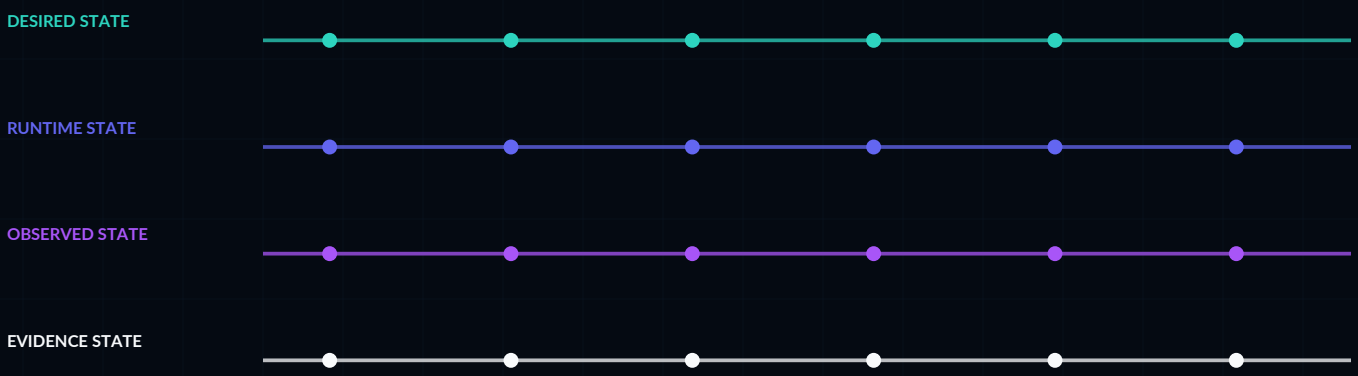


Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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

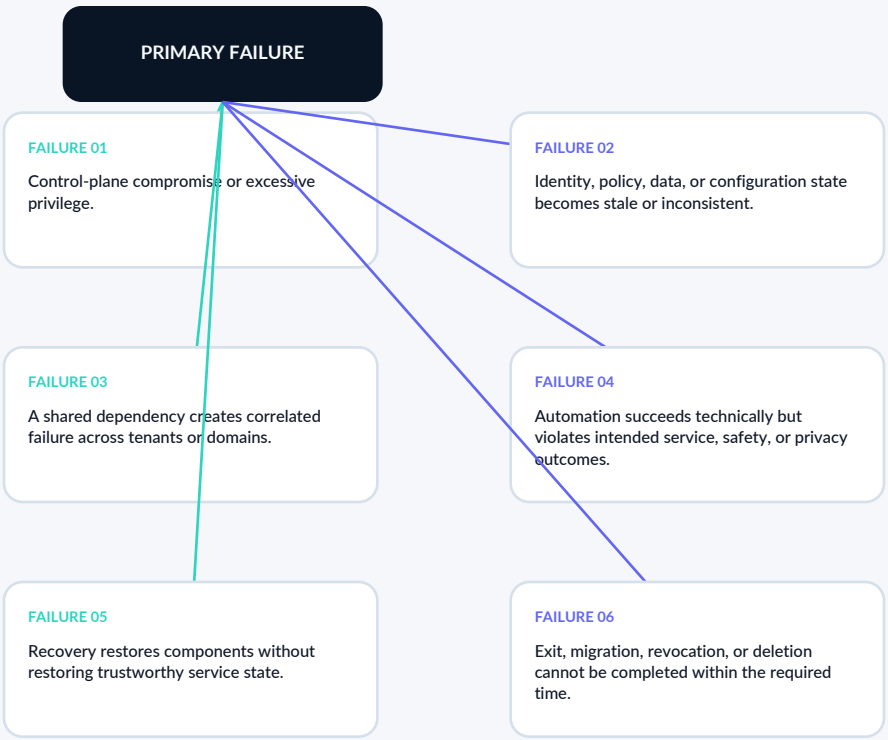
TB-04
Local environment versus external provider or network

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-DAT-0001, MYTHOS-DAT-0003, MYTHOS-DST-0001, MYTHOS-SWE-0005

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 governed integration fabric for data products, APIs, events, schemas, workflows, lineage, quality, privacy, observability, and resilient cross-domain exchange.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for enterprise data, api, event, and integration fabric.
- 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

DATA, KNOWLEDGE, AND EVIDENCE

IN SCOPE

A governed integration fabric for data products, APIs, events, schemas, workflows, lineage, quality, privacy, observability, and resilient cross-domain exchange.

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

Data products and ownership

Data products 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.

02

API and service interface fabric

API and service interface fabric 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

Event and streaming backbone

Event and streaming backbone 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

Schema and contract registry

Schema and contract registry 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

Integration workflows and transformation

Integration workflows and transformation 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

Lineage, quality, privacy, resilience, and evidence

Lineage, quality, privacy, resilience, 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

Data catalog, ownership, classification, and product contracts

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

COMPONENT 02

API gateway, service mesh, authorization, rate policy, and lifecycle

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

COMPONENT 03

Event brokers, topics, replay, ordering, retention, and dead-letter handling

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Schema registry, compatibility, semantic models, and versioning

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

COMPONENT 05

Workflow, ETL/ELT, CDC, orchestration, transformation, and compensation

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

COMPONENT 06

Lineage, quality, observability, privacy controls, regional boundaries, and recovery

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

- No mandatory architecture dependency. Interfaces to adjacent domains remain governed through explicit contracts.

MAPPED MYTHOS STANDARDS

- MYTHOS-DAT-0001
- MYTHOS-DAT-0003
- MYTHOS-DST-0001
- MYTHOS-SWE-0005

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 **NIST SP 800-53 Rev. 5**

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

SOURCE 02 **OpenTelemetry Specification**

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

SOURCE 03 **CISA Secure by Design**

https://www.cisa.gov/resources-tools/resources/secure-by-design

SOURCE 04 **SLSA Specification 1.2**

https://slsa.dev/spec/v1.2/

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-014
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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Edge, OT, IoT, and Telecommunications Convergence

A converged cyber-physical architecture for field devices, industrial control, edge compute, private wireless, public telecom, IoT, safety, disconnected operation, and fleet lifecycle.

PERMANENT DOCUMENT ID
RA-015

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Edge, OT, and Future Compute	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for edge, ot, iot, and telecommunications
2. convergence.
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
Physical process and safety

LAYER 02
Device and controller layer

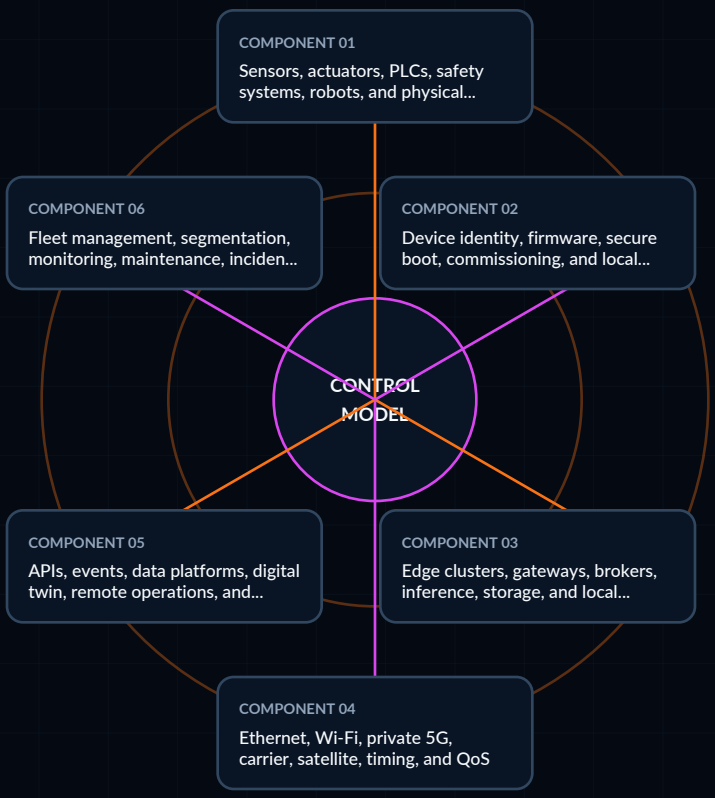
LAYER 03
Edge compute and gateway

LAYER 04
Private/public wireless and transport

LAYER 05
Cloud and enterprise integration

LAYER 06
Fleet, security, observability, and recovery

CORE COMPONENTS

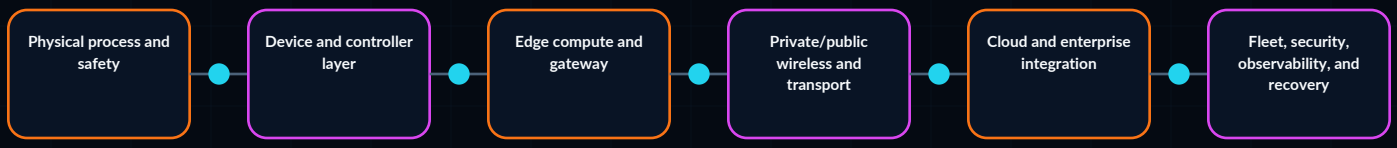


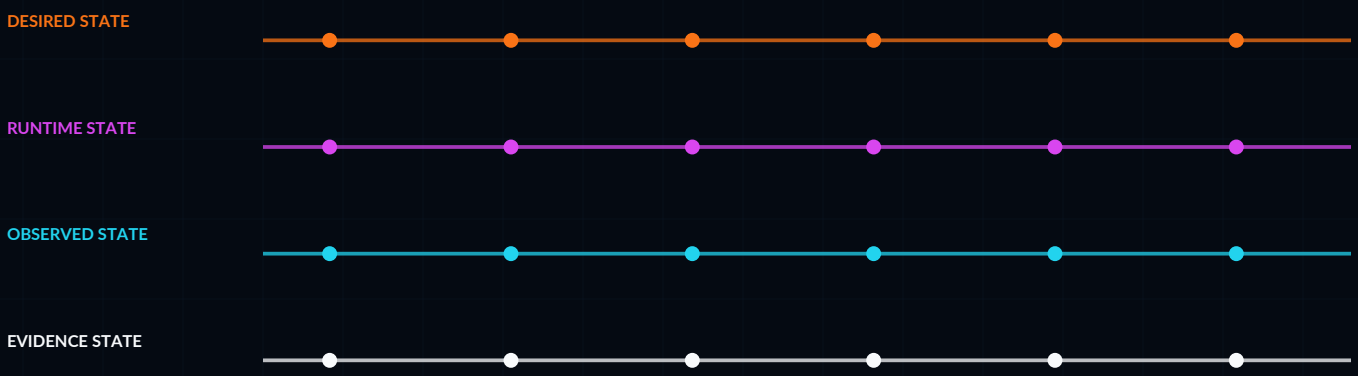
Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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

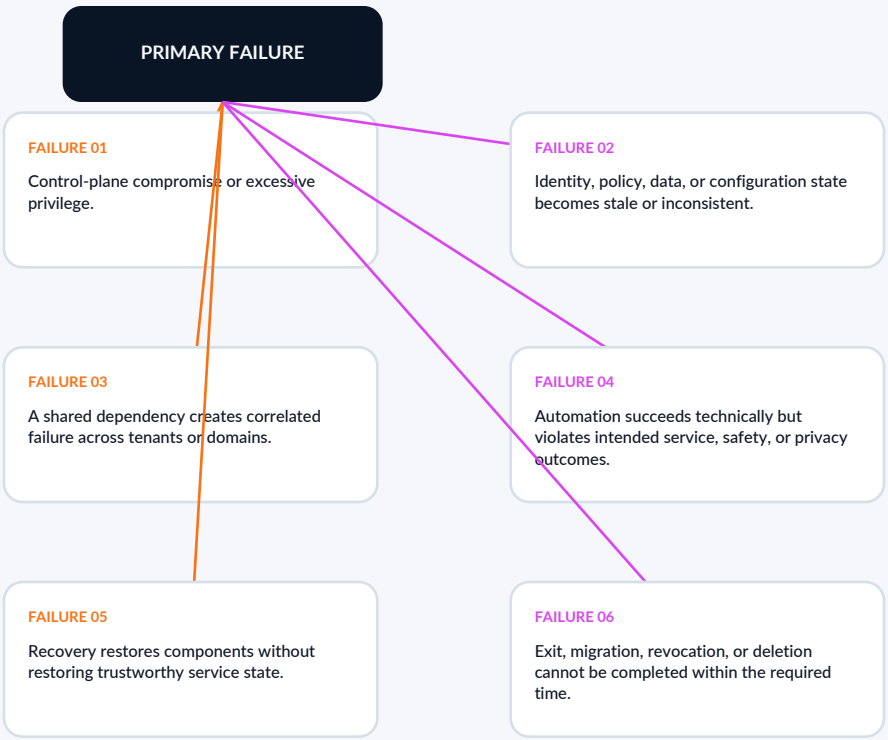
TB-04
Local environment versus external provider or network

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

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-OT-0001, MYTHOS-IOT-0001, MYTHOS-TEL-0001, MYTHOS-CLD-0010

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 converged cyber-physical architecture for field devices, industrial control, edge compute, private wireless, public telecom, IoT, safety, disconnected operation, and fleet lifecycle.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for edge, ot, iot, and telecommunications convergence.
- 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

EDGE, OT, AND FUTURE COMPUTE

IN SCOPE

A converged cyber-physical architecture for field devices, industrial control, edge compute, private wireless, public telecom, IoT, safety, disconnected operation, and fleet lifecycle.

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

Physical process and safety

Physical process and safety 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

Device and controller layer

Device and controller layer 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

Edge compute and gateway

Edge compute and gateway 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

Private/public wireless and transport

Private/public wireless and transport 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

Cloud and enterprise integration

Cloud and enterprise integration 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

Fleet, security, observability, and recovery

Fleet, security, observability, and recovery 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

Sensors, actuators, PLCs, safety systems, robots, and physical process

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

COMPONENT 02

Device identity, firmware, secure boot, commissioning, and local control

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

COMPONENT 03

Edge clusters, gateways, brokers, inference, storage, and local autonomy

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Ethernet, Wi-Fi, private 5G, carrier, satellite, timing, and QoS

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

COMPONENT 05

APIs, events, data platforms, digital twin, remote operations, and enterprise services

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

COMPONENT 06

Fleet management, segmentation, monitoring, maintenance, incident and disaster recovery

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-017 - Quantum-Safe, Confidential, and Verifiable Compute
- RA-018 - Physical AI and Autonomous Fleet Operations

MAPPED MYTHOS STANDARDS

- MYTHOS-OT-0001
- MYTHOS-IOT-0001
- MYTHOS-TEL-0001
- MYTHOS-CLD-0010

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 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

SOURCE 02 **NIST SP 800-53 Rev. 5**

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

SOURCE 03 **OpenTelemetry Specification**

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

SOURCE 04 **CISA Secure by Design**

https://www.cisa.gov/resources-tools/resources/secure-by-design

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

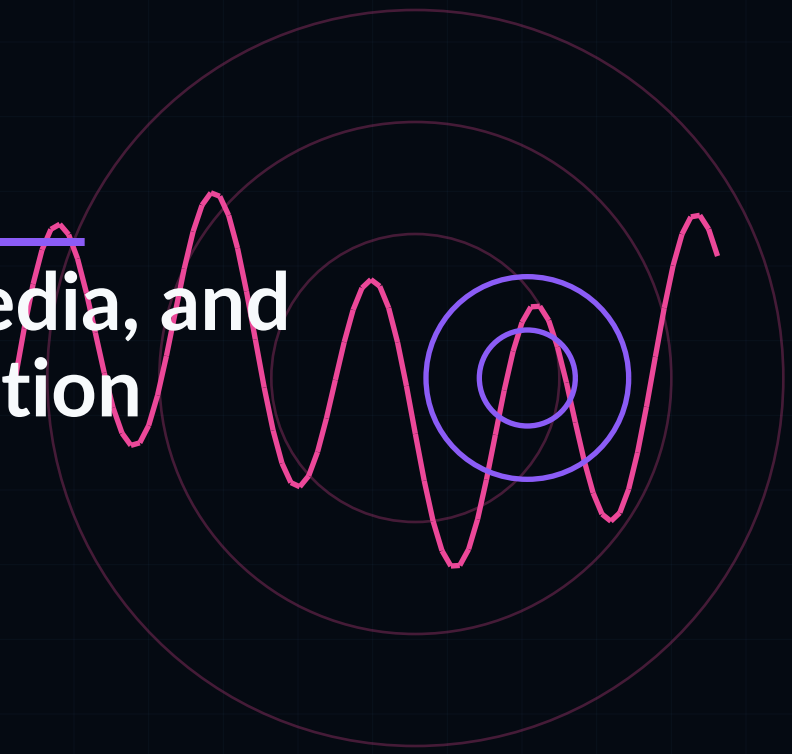
PERMANENT DOCUMENT IDENTITY	RA-015
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Future Domain Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Realtime Voice, Media, and Multimodal Interaction Platform



A realtime interaction architecture for speech, audio, video, avatars or presence, transcription, translation, multimodal models, accessibility, consent, provenance, and low-latency orchestration.

PERMANENT DOCUMENT ID
RA-016

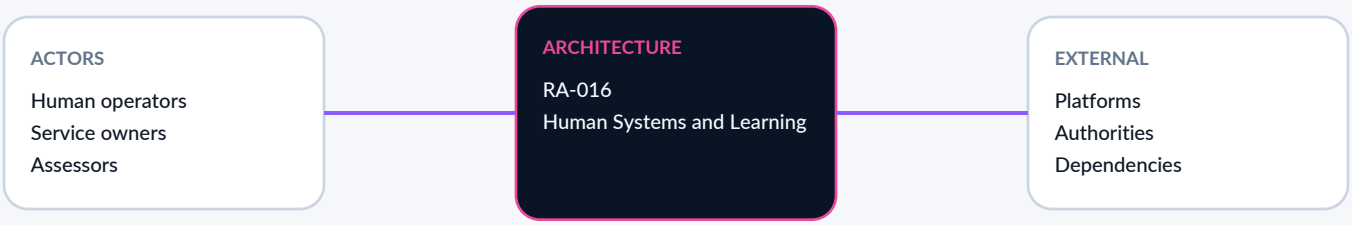
PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Human Systems and Learning	6	6	18	0

SYSTEM CONTEXT



EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for realtime voice, media, and multimodal interaction
2. platform.
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
Capture and device interaction

LAYER 02
Realtime media transport

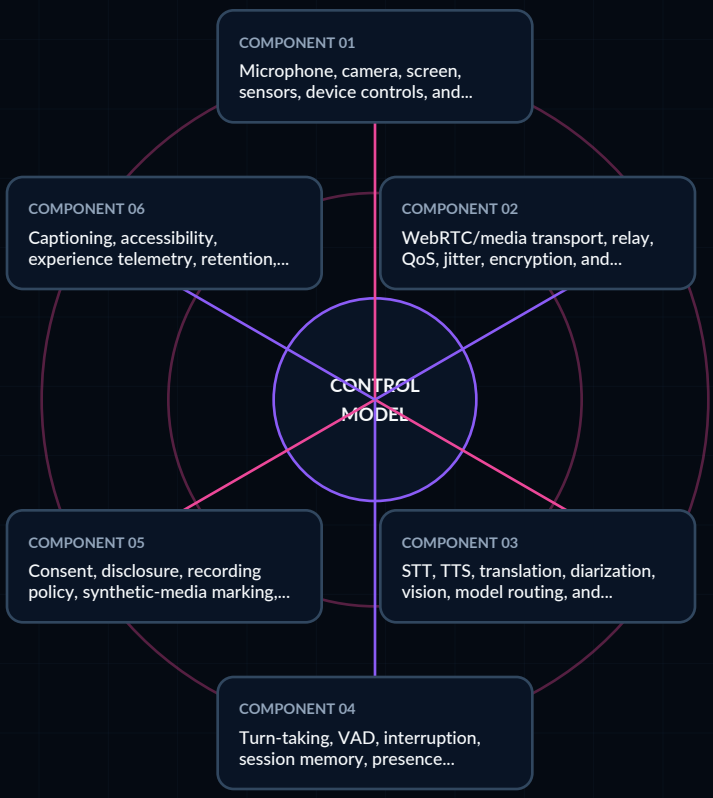
LAYER 03
Speech and multimodal intelligence

LAYER 04
Conversation and presence runtime

LAYER 05
Consent, privacy, safety, and provenance

LAYER 06
Quality, accessibility, storage, and recovery

CORE COMPONENTS

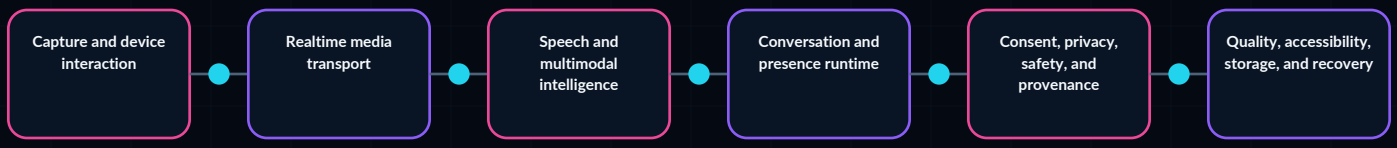


Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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

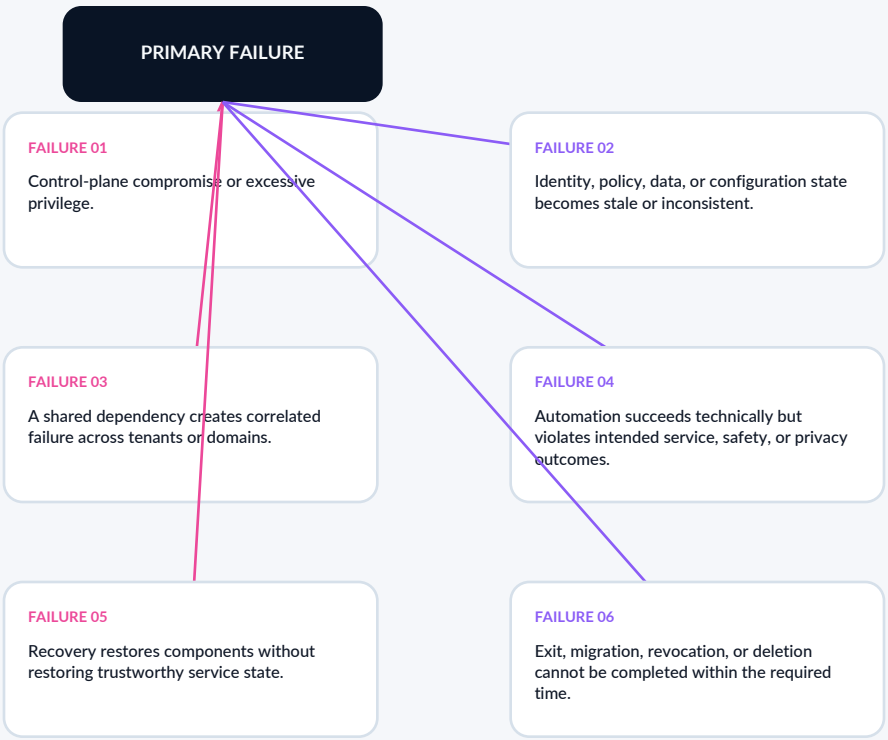
TB-04
Local environment versus external provider or network

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

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-MED-0001, MYTHOS-AI-0008, MYTHOS-ID-0001, MYTHOS-WEB-0001

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 realtime interaction architecture for speech, audio, video, avatars or presence, transcription, translation, multimodal models, accessibility, consent, provenance, and low-latency orchestration.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for realtime voice, media, and multimodal interaction platform.
- 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

HUMAN SYSTEMS AND LEARNING

IN SCOPE

A realtime interaction architecture for speech, audio, video, avatars or presence, transcription, translation, multimodal models, accessibility, consent, provenance, and low-latency orchestration.

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

Capture and device interaction

Capture and device interaction 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

Realtime media transport

Realtime media transport 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

Speech and multimodal intelligence

Speech and multimodal intelligence 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

Conversation and presence runtime

Conversation and presence runtime 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

Consent, privacy, safety, and provenance

Consent, privacy, safety, and provenance 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

Quality, accessibility, storage, and recovery

Quality, accessibility, storage, and recovery 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

Microphone, camera, screen, sensors, device controls, and permissions

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

COMPONENT 02

WebRTC/media transport, relay, QoS, jitter, encryption, and regional routing

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

COMPONENT 03

STT, TTS, translation, diarization, vision, model routing, and grounding

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Turn-taking, VAD, interruption, session memory, presence behavior, and tool invocation

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

COMPONENT 05

Consent, disclosure, recording policy, synthetic-media marking, identity, moderation, and audit

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

COMPONENT 06

Captioning, accessibility, experience telemetry, retention, export, incident and fallback

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-019 - Sovereign Browser, Remote Access, and Web Automation
- RA-020 - Adaptive Learning, Cyber Range, and Workforce Assurance

MAPPED MYTHOS STANDARDS

- MYTHOS-MED-0001
- MYTHOS-AI-0008
- MYTHOS-ID-0001
- MYTHOS-WEB-0001

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 **C2PA Technical Specification**

https://c2pa.org/specifications/

SOURCE 02 **NIST AI Risk Management Framework 1.0 (revision in progress as of publication date)**

https://www.nist.gov/itl/ai-risk-management-framework

SOURCE 03 **NIST SP 800-53 Rev. 5**

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

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-016
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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Quantum-Safe, Confidential, and Verifiable Compute

A compute trust architecture combining crypto-agility, post-quantum migration, confidential computing, attestation, measured boot, workload identity, verifiable artifacts, and controlled key release.

PERMANENT DOCUMENT ID
RA-017

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Edge, OT, and Future Compute	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for quantum-safe, confidential, and verifiable
2. compute.
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
Cryptographic inventory and agility

LAYER 02
Hardware root of trust

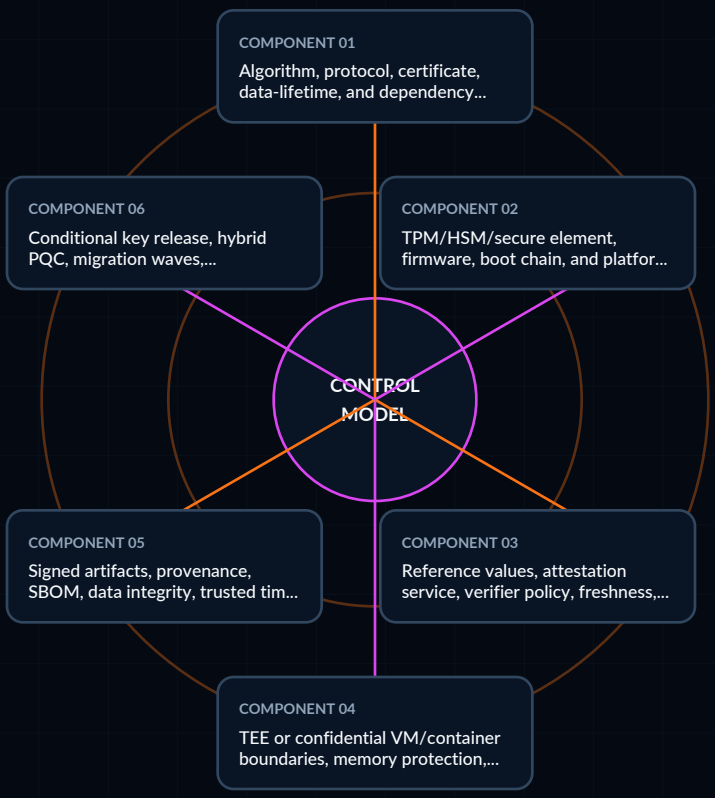
LAYER 03
Measured boot and attestation

LAYER 04
Confidential workload execution

LAYER 05
Verifiable software and data

LAYER 06
Key release, migration, recovery, and evidence

CORE COMPONENTS

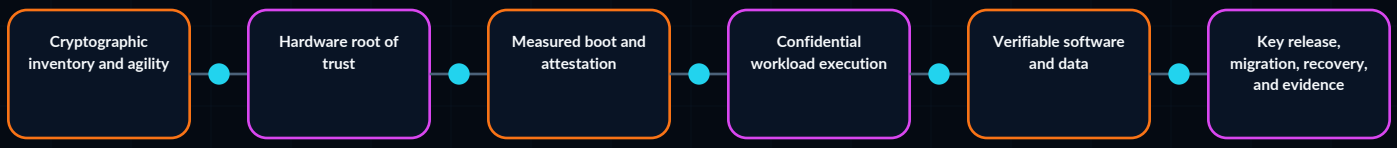


Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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

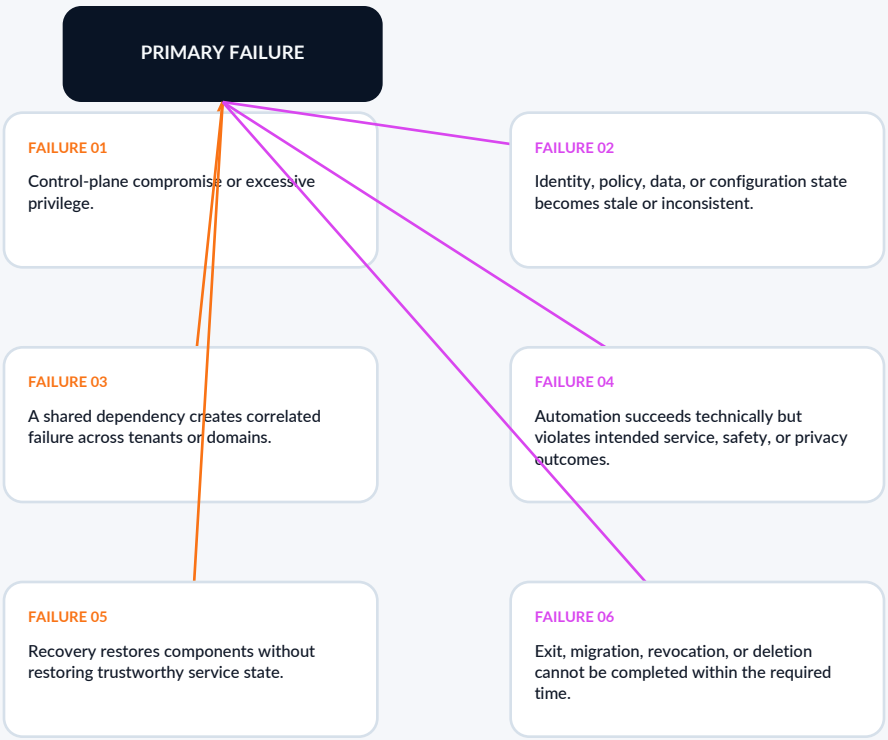
TB-04
Local environment versus external provider or network

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

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-EMT-0002, MYTHOS-QTC-0004, MYTHOS-HW-0002, MYTHOS-ID-0003

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 compute trust architecture combining crypto-agility, post-quantum migration, confidential computing, attestation, measured boot, workload identity, verifiable artifacts, and controlled key release.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for quantum-safe, confidential, and verifiable compute.
- 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

EDGE, OT, AND FUTURE COMPUTE

IN SCOPE

A compute trust architecture combining crypto-agility, post-quantum migration, confidential computing, attestation, measured boot, workload identity, verifiable artifacts, and controlled key release.

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

Cryptographic inventory and agility

Cryptographic inventory and agility 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

Hardware root of trust

Hardware root of trust 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

Measured boot and attestation

Measured boot and attestation 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

Confidential workload execution

Confidential workload execution 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

Verifiable software and data

Verifiable software and data 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

Key release, migration, recovery, and evidence

Key release, migration, recovery, 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

Algorithm, protocol, certificate, data-lifetime, and dependency inventory

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

COMPONENT 02

TPM/HSM/secure element, firmware, boot chain, and platform identity

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

COMPONENT 03

Reference values, attestation service, verifier policy, freshness, and revocation

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

TEE or confidential VM/container boundaries, memory protection, and side-channel controls

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

COMPONENT 05

Signed artifacts, provenance, SBOM, data integrity, trusted time, and transparent evidence

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

COMPONENT 06

Conditional key release, hybrid PQC, migration waves, break-glass, backup, and platform replacement

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

- No mandatory architecture dependency. Interfaces to adjacent domains remain governed through explicit contracts.

MAPPED MYTHOS STANDARDS

- MYTHOS-EMT-0002
- MYTHOS-QTC-0004
- MYTHOS-HW-0002
- MYTHOS-ID-0003

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 **NIST Post-Quantum Cryptography Project**

https://csrc.nist.gov/projects/post-quantum-cryptography

SOURCE 02 **NIST SP 800-53 Rev. 5**

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

SOURCE 03 **SLSA Specification 1.2**

https://slsa.dev/spec/v1.2/

SOURCE 04 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

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

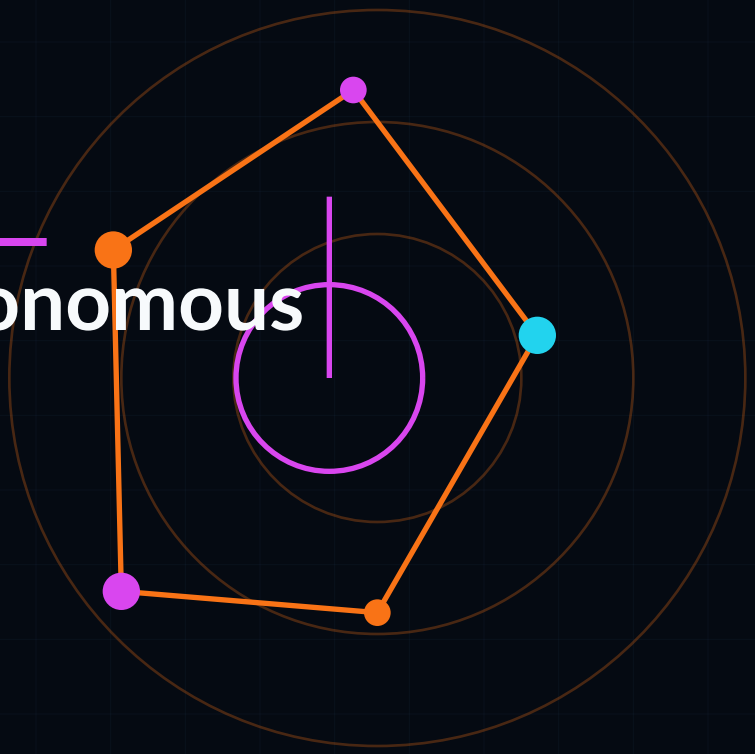
PERMANENT DOCUMENT IDENTITY	RA-017
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Future Domain Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Physical AI and Autonomous Fleet Operations



A safety-governed architecture for robots, autonomous machines, perception, planning, local control, fleet orchestration, human intervention, updates, maintenance, and incident evidence.

PERMANENT DOCUMENT ID
RA-018

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Edge, OT, and Future Compute	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for physical ai and autonomous fleet operations.
2. Make ownership, trust boundaries, data and control flows, failure handling, and evidence explicit.
3. Support multiple implementation profiles without weakening mandatory assurance outcomes.
4. Provide a governed adoption path from discovery through production operation and retirement.

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
Physical platform and safety

LAYER 02
Perception and world model

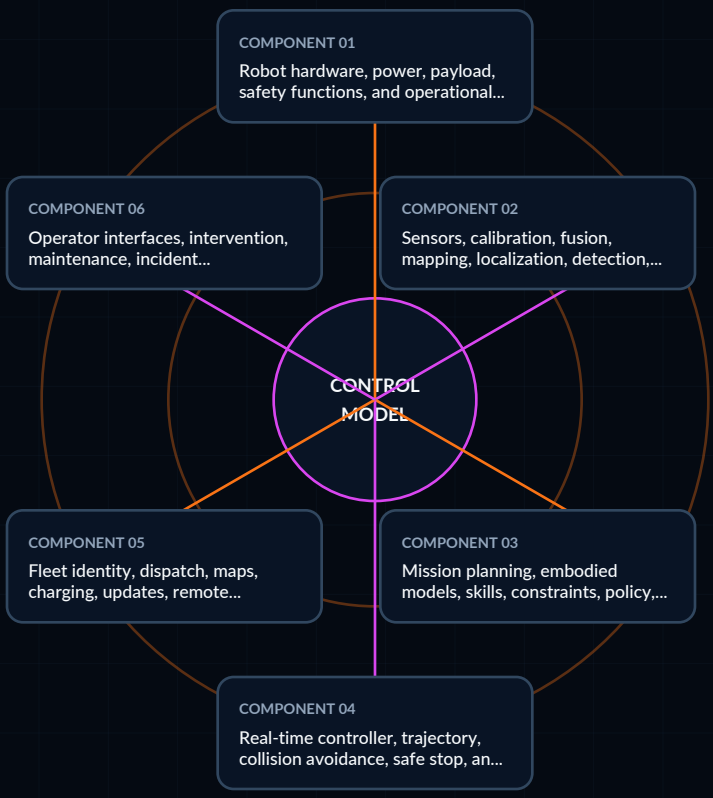
LAYER 03
Planning and task intelligence

LAYER 04
Motion and local control

LAYER 05
Fleet orchestration and communications

LAYER 06
Human oversight, maintenance, evidence, and recovery

CORE COMPONENTS

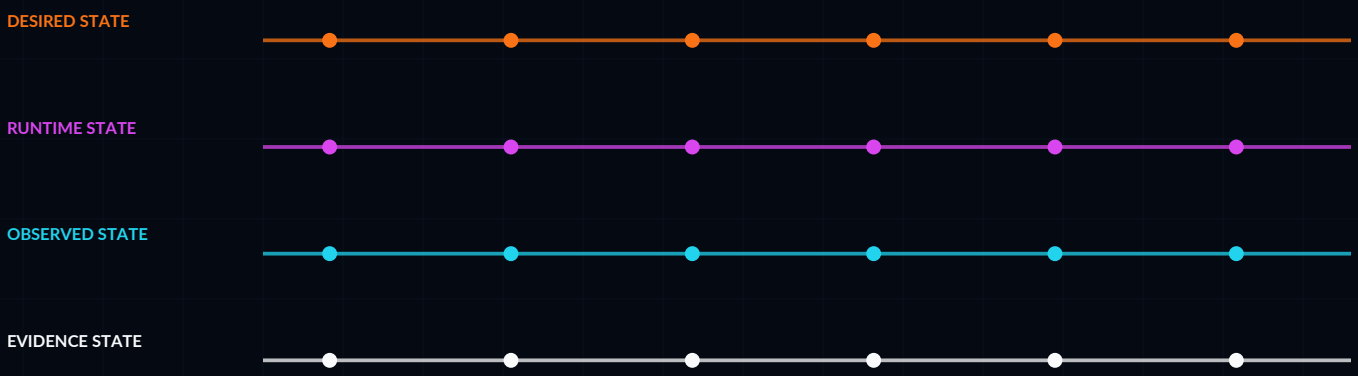


Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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

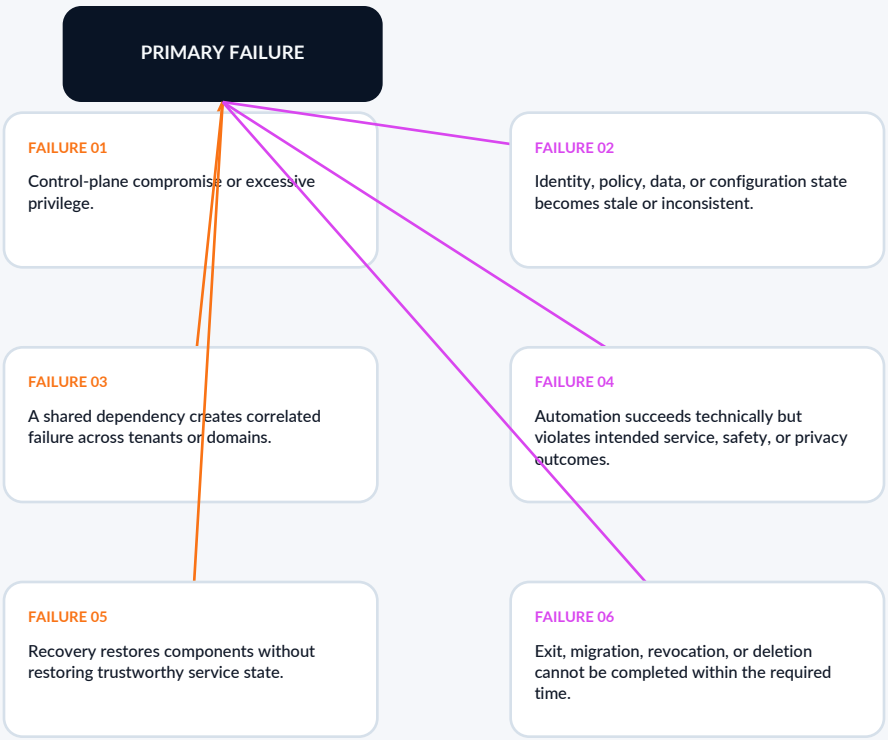
TB-04
Local environment versus external provider or network

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

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-ROB-0001, MYTHOS-ROB-0002, MYTHOS-ROB-0004, MYTHOS-ROB-0005

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 safety-governed architecture for robots, autonomous machines, perception, planning, local control, fleet orchestration, human intervention, updates, maintenance, and incident evidence.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for physical ai and autonomous fleet operations.
- 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

EDGE, OT, AND FUTURE COMPUTE

IN SCOPE

A safety-governed architecture for robots, autonomous machines, perception, planning, local control, fleet orchestration, human intervention, updates, maintenance, and incident 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

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

Physical platform and safety

Physical platform and safety 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

Perception and world model

Perception and world 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.

03

Planning and task intelligence

Planning and task intelligence 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

Motion and local control

Motion and local 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.

05

Fleet orchestration and communications

Fleet orchestration and communications 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

Human oversight, maintenance, evidence, and recovery

Human oversight, maintenance, evidence, and recovery 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

Robot hardware, power, payload, safety functions, and operational design domain

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

COMPONENT 02

Sensors, calibration, fusion, mapping, localization, detection, and uncertainty

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

COMPONENT 03

Mission planning, embodied models, skills, constraints, policy, and simulation

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Real-time controller, trajectory, collision avoidance, safe stop, and local autonomy

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

COMPONENT 05

Fleet identity, dispatch, maps, charging, updates, remote operations, and lost-link behavior

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

COMPONENT 06

Operator interfaces, intervention, maintenance, incident reconstruction, validation, and retirement

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

- No mandatory architecture dependency. Interfaces to adjacent domains remain governed through explicit contracts.

MAPPED MYTHOS STANDARDS

- MYTHOS-ROB-0001
- MYTHOS-ROB-0002
- MYTHOS-ROB-0004
- MYTHOS-ROB-0005

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 **NIST AI Risk Management Framework 1.0 (revision in progress as of publication date)**

https://www.nist.gov/itl/ai-risk-management-framework

SOURCE 02 **NIST SP 800-53 Rev. 5**

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

SOURCE 03 **OpenTelemetry Specification**

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

SOURCE 04 **CISA Secure by Design**

https://www.cisa.gov/resources-tools/resources/secure-by-design

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

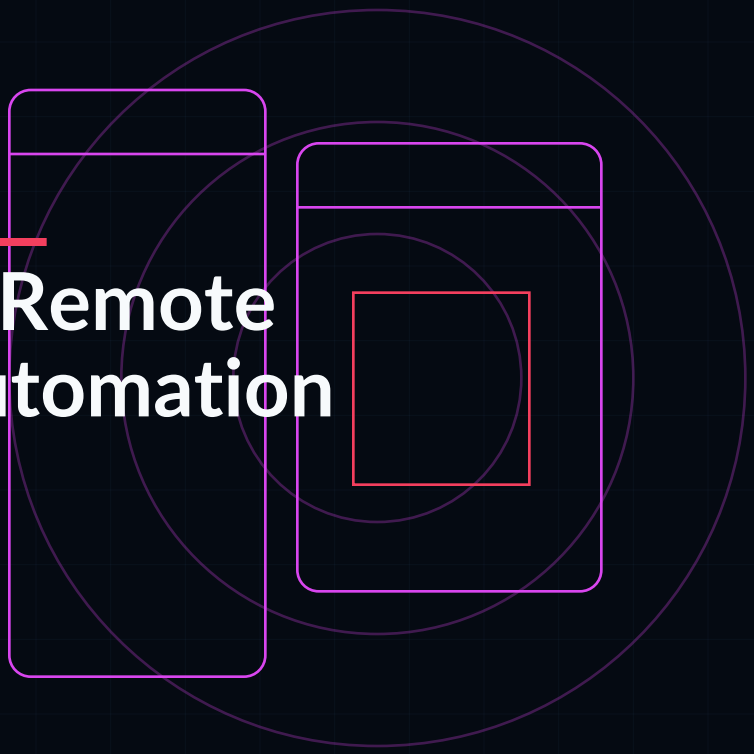
PERMANENT DOCUMENT IDENTITY	RA-018
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Future Domain Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Sovereign Browser, Remote Access, and Web Automation



A sovereign browser and remote-interaction architecture for isolated browsing, credential protection, remote access, web automation, session evidence, data boundaries, and controlled tool execution.

PERMANENT DOCUMENT ID
RA-019

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Cybersecurity and Network Security	6	6	18	0

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for sovereign browser, remote access, and web
2. automation.
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
Browser and workspace isolation

LAYER 02
Identity, credentials, and session authority

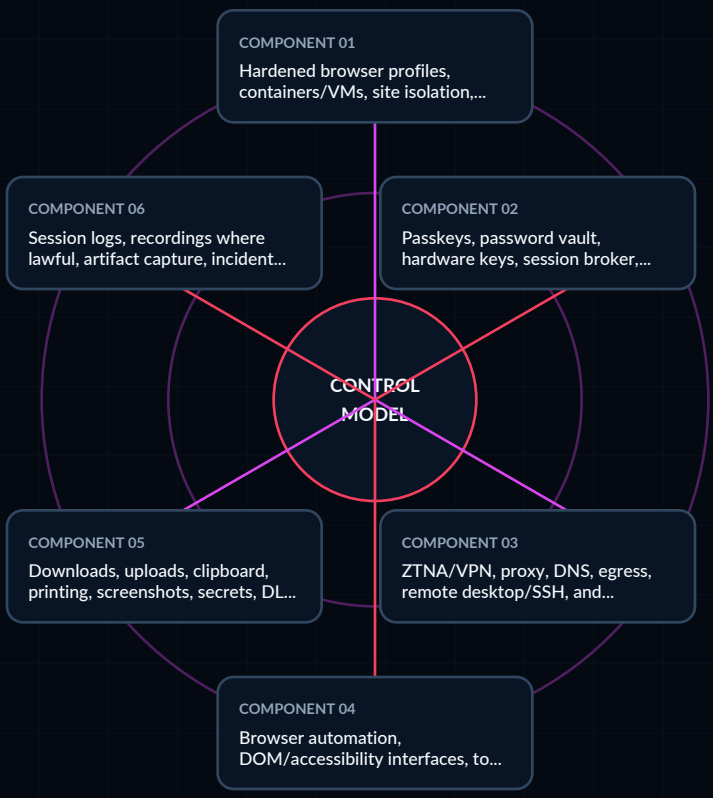
LAYER 03
Network and remote-access plane

LAYER 04
Automation and agent tool plane

LAYER 05
Data, download, upload, and clipboard policy

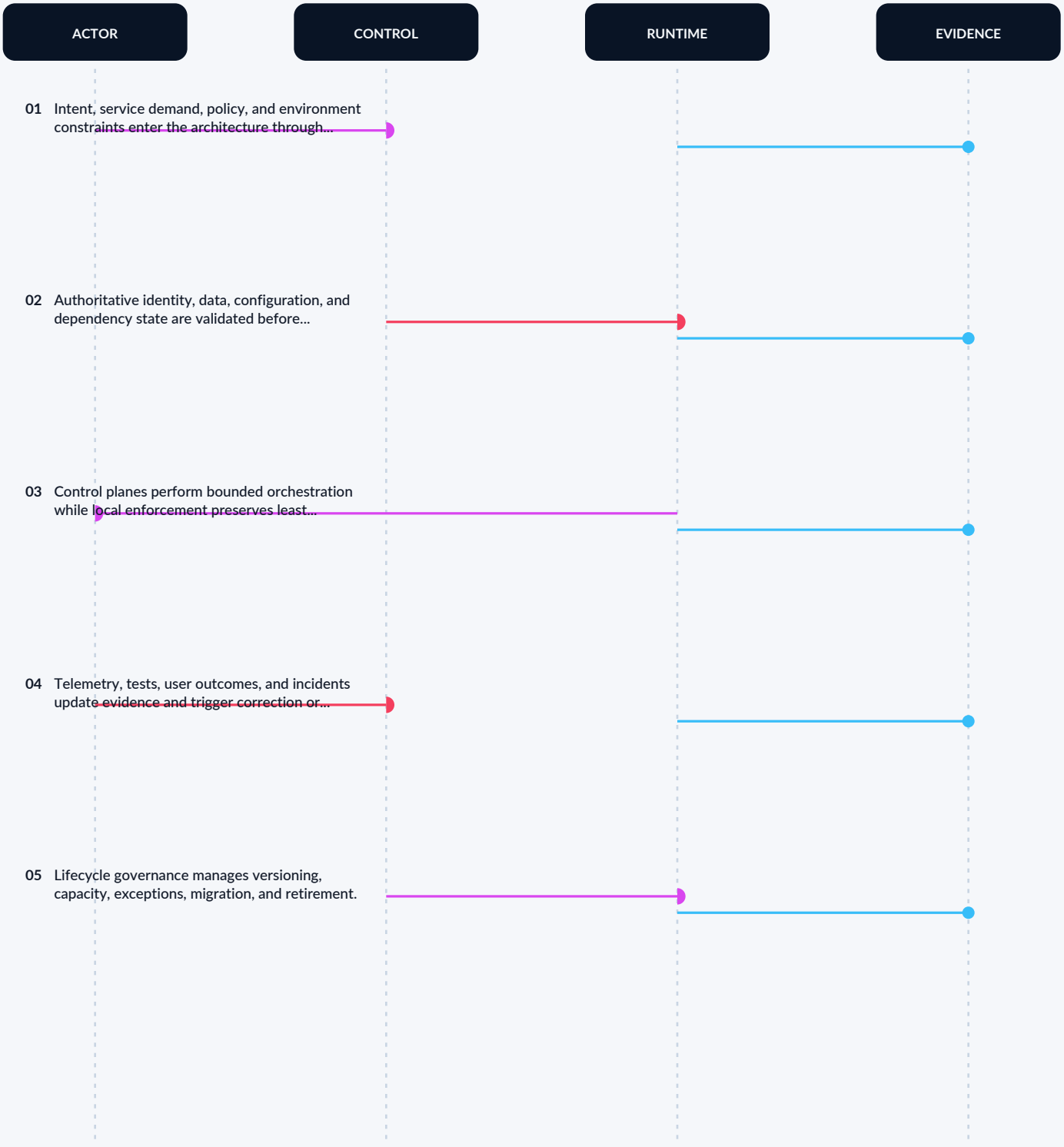
LAYER 06
Observation, evidence, recovery, and privacy

CORE COMPONENTS



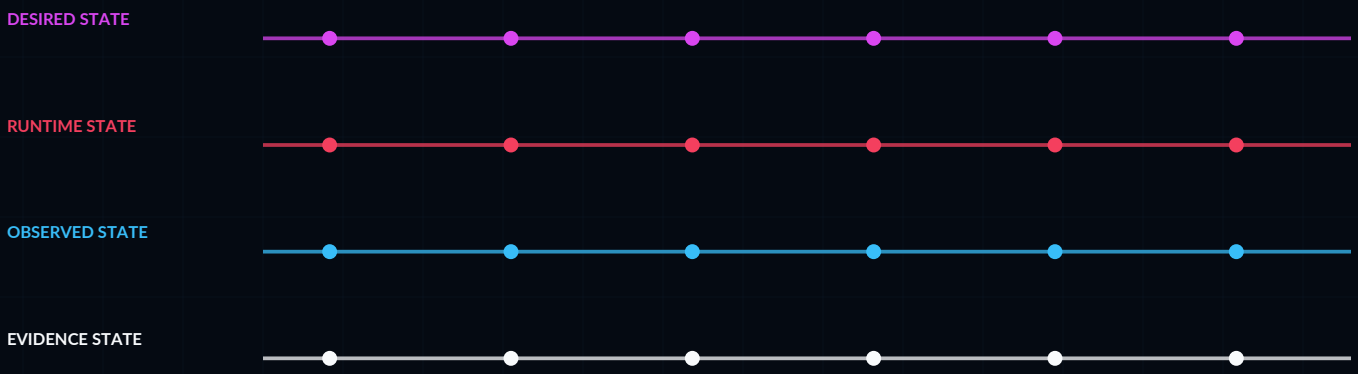
Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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

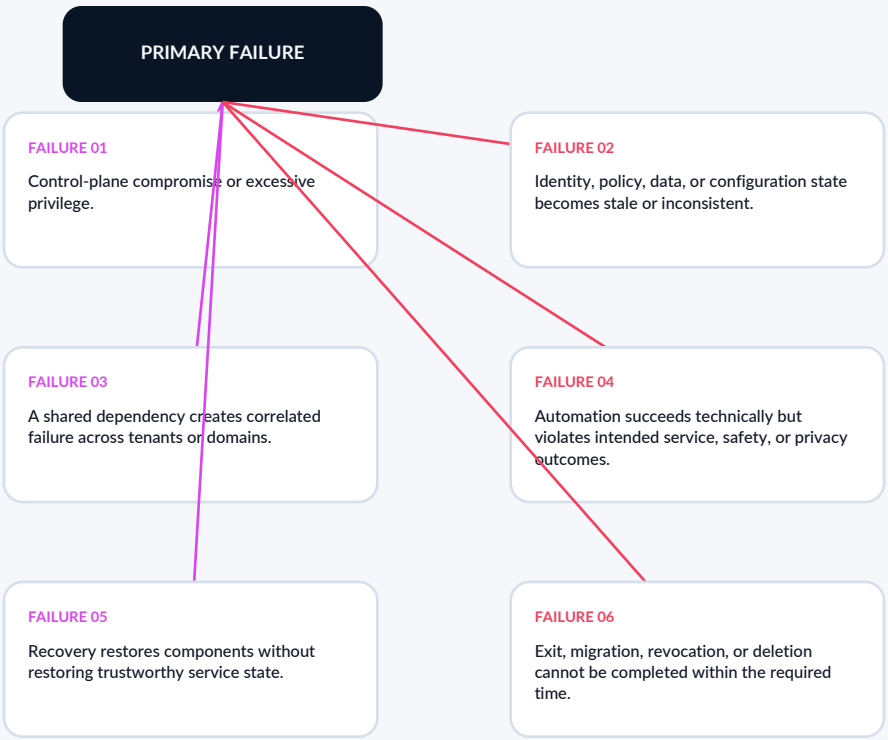
TB-04
Local environment versus external provider or network

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-WEB-0001, MYTHOS-SEC-0003, MYTHOS-ID-0001, MYTHOS-END-0001

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 sovereign browser and remote-interaction architecture for isolated browsing, credential protection, remote access, web automation, session evidence, data boundaries, and controlled tool execution.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for sovereign browser, remote access, and web automation.
- 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

CYBERSECURITY AND NETWORK SECURITY

IN SCOPE

A sovereign browser and remote-interaction architecture for isolated browsing, credential protection, remote access, web automation, session evidence, data boundaries, and controlled tool execution.

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

Browser and workspace isolation

Browser and workspace isolation 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

Identity, credentials, and session authority

Identity, credentials, and session authority 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

Network and remote-access plane

Network and remote-access plane 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

Automation and agent tool plane

Automation and agent tool plane 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

Data, download, upload, and clipboard policy

Data, download, upload, and clipboard policy 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, evidence, recovery, and privacy

Observation, evidence, recovery, and privacy 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

Hardened browser profiles, containers/VMs, site isolation, extension policy, and updates

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

COMPONENT 02

Passkeys, password vault, hardware keys, session broker, privilege, and step-up

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

COMPONENT 03

ZTNA/VPN, proxy, DNS, egress, remote desktop/SSH, and destination policy

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Browser automation, DOM/accessibility interfaces, tool broker, approvals, and anti-injection controls

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

COMPONENT 05

Downloads, uploads, clipboard, printing, screenshots, secrets, DLP, and local storage

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

COMPONENT 06

Session logs, recordings where lawful, artifact capture, incident response, reset, and secure disposal

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

- No mandatory architecture dependency. Interfaces to adjacent domains remain governed through explicit contracts.

MAPPED MYTHOS STANDARDS

- MYTHOS-WEB-0001
- MYTHOS-SEC-0003
- MYTHOS-ID-0001
- MYTHOS-END-0001

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 **NIST SP 800-207, Zero Trust Architecture**

https://csrc.nist.gov/pubs/sp/800/207/final

SOURCE 02 **NIST SP 1800-35, Implementing a Zero Trust Architecture**

https://www.nccoe.nist.gov/projects/implementing-zero-trust-architecture

SOURCE 03 **W3C Web Authentication Level 3**

https://www.w3.org/TR/webauthn-3/

SOURCE 04 **CISA Secure by Design**

https://www.cisa.gov/resources-tools/resources/secure-by-design

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-019
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.

REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Adaptive Learning, Cyber Range, and Workforce Assurance

An adaptive learning and cyber-range architecture connecting skills, role profiles, personalized instruction, laboratories, scenarios, assessment, evidence, safety, and workforce readiness.

PERMANENT DOCUMENT ID
RA-020

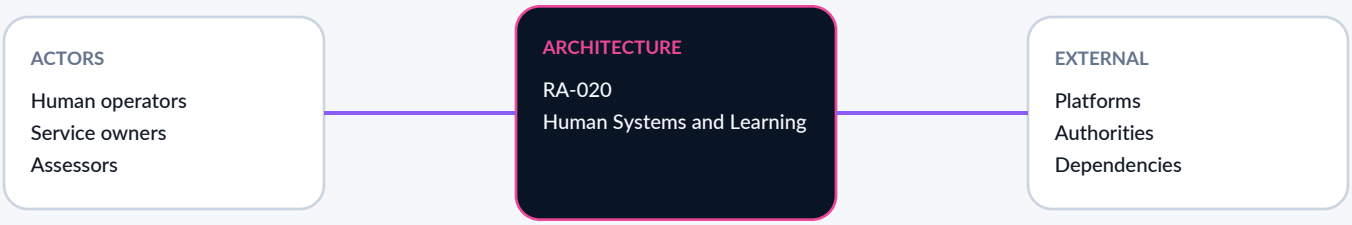
PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Future Domain Set

Architecture identity and operating position

ARCHITECTURE SET	DOMAIN	LAYERS	COMPONENTS	ACCEPTANCE	DEPENDENCIES
Future Domain Set	Human Systems and Learning	6	6	18	0

SYSTEM CONTEXT



EXECUTIVE OUTCOMES

1. Establish a deployable, product-neutral target architecture for adaptive learning, cyber range, and workforce
2. 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
Role and competency model

LAYER 02
Learner identity and profile

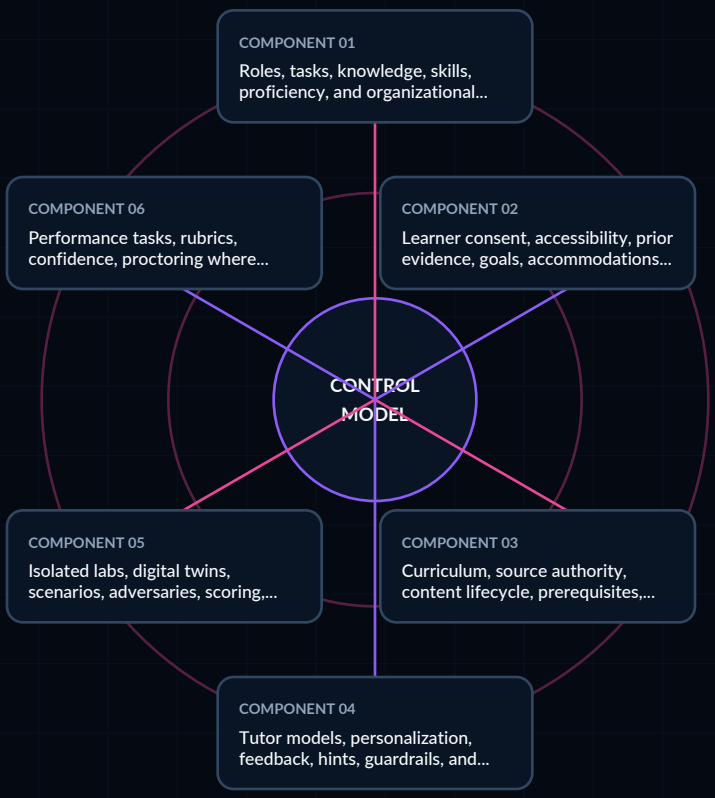
LAYER 03
Content and knowledge system

LAYER 04
Adaptive tutor and agent support

LAYER 05
Lab, simulation, and cyber range

LAYER 06
Assessment, evidence, credentialing, and improvement

CORE COMPONENTS

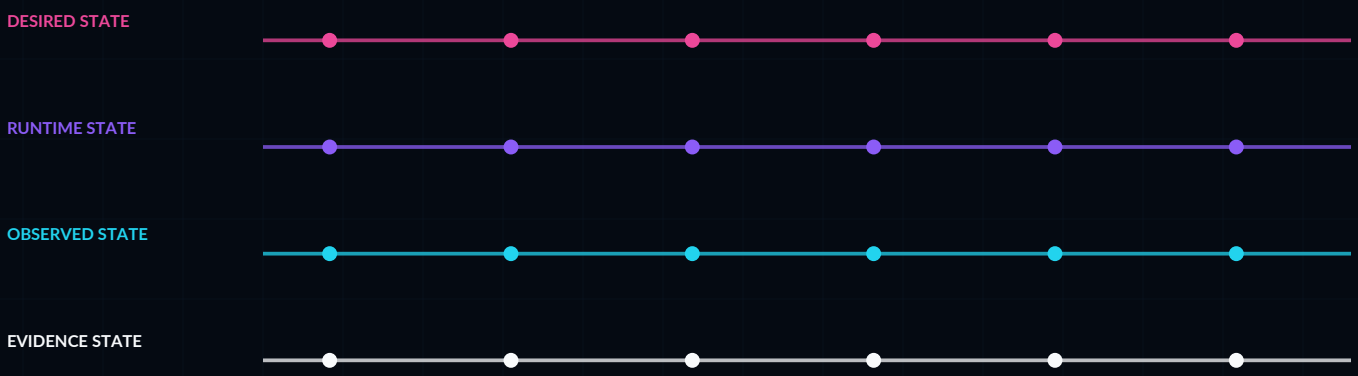


Each layer and component owns an explicit interface, state model, authority boundary, failure behavior, telemetry contract, recovery path, and evidence obligation.

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

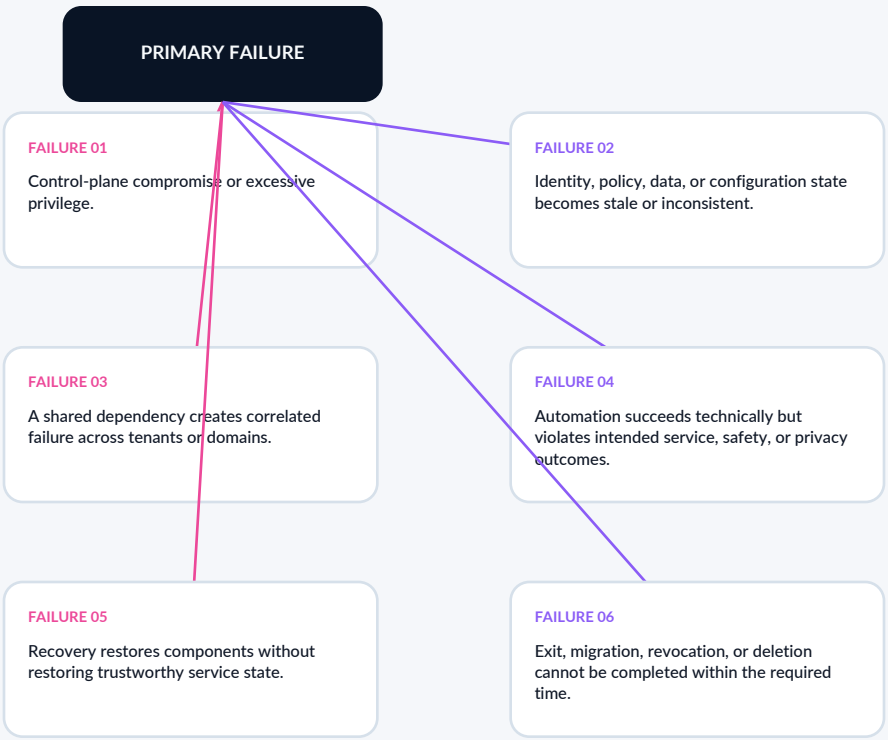
TB-04
Local environment versus external provider or network

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

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-EDU-0001, MYTHOS-EDU-0002, MYTHOS-AI-0008, MYTHOS-DAT-0005

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

An adaptive learning and cyber-range architecture connecting skills, role profiles, personalized instruction, laboratories, scenarios, assessment, evidence, safety, and workforce readiness.

EXECUTIVE OUTCOMES

- Establish a deployable, product-neutral target architecture for adaptive learning, cyber range, and workforce 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

HUMAN SYSTEMS AND LEARNING

IN SCOPE

An adaptive learning and cyber-range architecture connecting skills, role profiles, personalized instruction, laboratories, scenarios, assessment, evidence, safety, and workforce readiness.

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

Role and competency model

Role and competency 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

Learner identity and profile

Learner identity and profile 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

Content and knowledge system

Content and knowledge system 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

Adaptive tutor and agent support

Adaptive tutor and agent support 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

Lab, simulation, and cyber range

Lab, simulation, and cyber range 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

Assessment, evidence, credentialing, and improvement

Assessment, evidence, credentialing, and improvement 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

Roles, tasks, knowledge, skills, proficiency, and organizational demand

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

COMPONENT 02

Learner consent, accessibility, prior evidence, goals, accommodations, and privacy

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

COMPONENT 03

Curriculum, source authority, content lifecycle, prerequisites, and retrieval

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

Tutor models, personalization, feedback, hints, guardrails, and human escalation

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

COMPONENT 05

Isolated labs, digital twins, scenarios, adversaries, scoring, reset, and safe infrastructure

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

COMPONENT 06

Performance tasks, rubrics, confidence, proctoring where appropriate, credentials, analytics, and program review

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

- No mandatory architecture dependency. Interfaces to adjacent domains remain governed through explicit contracts.

MAPPED MYTHOS STANDARDS

- MYTHOS-EDU-0001
- MYTHOS-EDU-0002
- MYTHOS-AI-0008
- MYTHOS-DAT-0005

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 **NIST NICE Workforce Framework**

https://www.nist.gov/itl/applied-cybersecurity/nice/nice-framework-resource-center

SOURCE 02 **NIST AI Risk Management Framework 1.0 (revision in progress as of publication date)**

https://www.nist.gov/itl/ai-risk-management-framework

SOURCE 03 **NIST SP 800-53 Rev. 5**

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

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-020
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

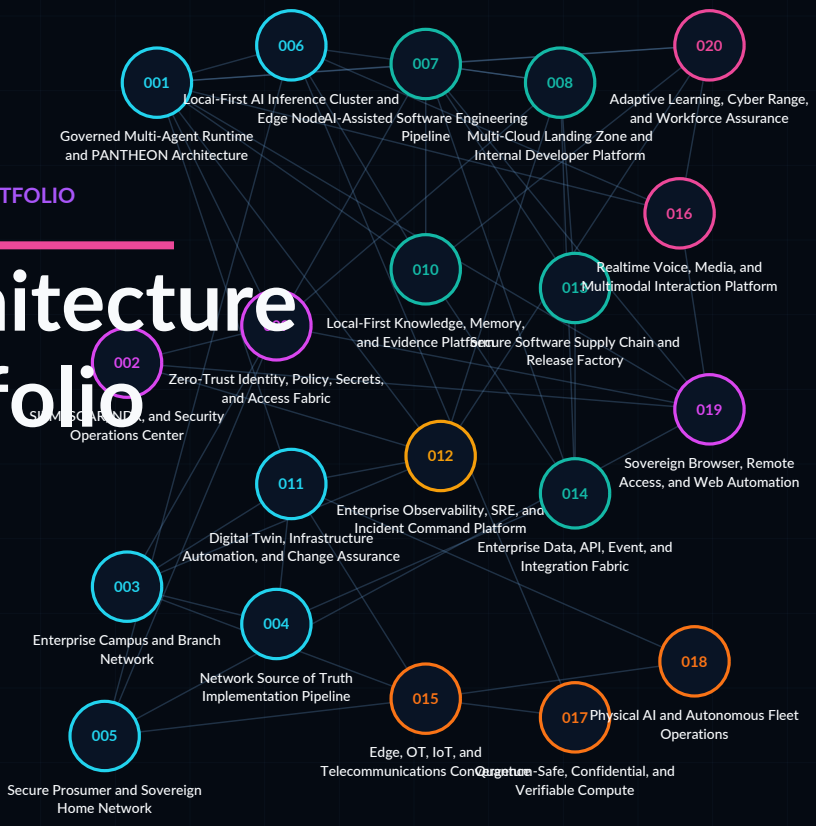
REVISION RECORD

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.

MYTHOS
SYSTEMS

REFERENCE ARCHITECTURE LIBRARY / EXECUTIVE PORTFOLIO

Reference Architecture Executive Portfolio



Executive map of the permanent Mythos reference architecture system: canonical architectures, approved future-domain architectures, dependency relationships, adoption sequence, assurance doctrine, and decision gates.

PERMANENT DOCUMENT ID	ARCHITECTURES	SETS
MYTHOS-RA-EXEC-0001	20	Canonical + Future Domain

Portfolio position and operating doctrine

20	7	13	50	360
PERMANENT ARCHITECTURES	CANONICAL SET	FUTURE DOMAIN SET	DEPENDENCY EDGES	ACCEPTANCE CRITERIA

PERMANENT IDENTITY Architecture IDs survive revisions and portfolio assembly.

CONTEXT BEFORE COMPONENTS Business outcome, actor, trust, state, and failure context comes before technology selection.

EVIDENCE-BEARING DESIGN Every consequential crossing and lifecycle decision carries observable evidence.

SAFE FAILURE Architectures define degraded behavior, containment, recovery, and rollback - not only happy paths.

VERIFIABLE ADOPTION Implementation advances through acceptance criteria and observable outcomes, not component presence.

Permanent architecture register

RA-001 CANONICAL
Governed Multi-Agent Runtime and PANTHEON Architecture

RA-002 CANONICAL
SIEM, SOAR, NDR, and Security Operations Center

RA-003 CANONICAL
Enterprise Campus and Branch Network

RA-004 CANONICAL
Network Source of Truth Implementation Pipeline

RA-005 CANONICAL
Secure Prosumer and Sovereign Home Network

RA-006 CANONICAL
Local-First AI Inference Cluster and Edge Node

RA-007 CANONICAL
AI-Assisted Software Engineering Pipeline

RA-008 FUTURE
Multi-Cloud Landing Zone and Internal Developer Platform

RA-009 FUTURE
Zero-Trust Identity, Policy, Secrets, and Access Fabric

RA-010 FUTURE
Local-First Knowledge, Memory, and Evidence Platform

RA-011 FUTURE
Digital Twin, Infrastructure Automation, and Change Assurance

RA-012 FUTURE
Enterprise Observability, SRE, and Incident Command Platform

RA-013 FUTURE
Secure Software Supply Chain and Release Factory

RA-014 FUTURE
Enterprise Data, API, Event, and Integration Fabric

RA-015 FUTURE
Edge, OT, IoT, and Telecommunications Convergence

RA-016 FUTURE
Realtime Voice, Media, and Multimodal Interaction Platform

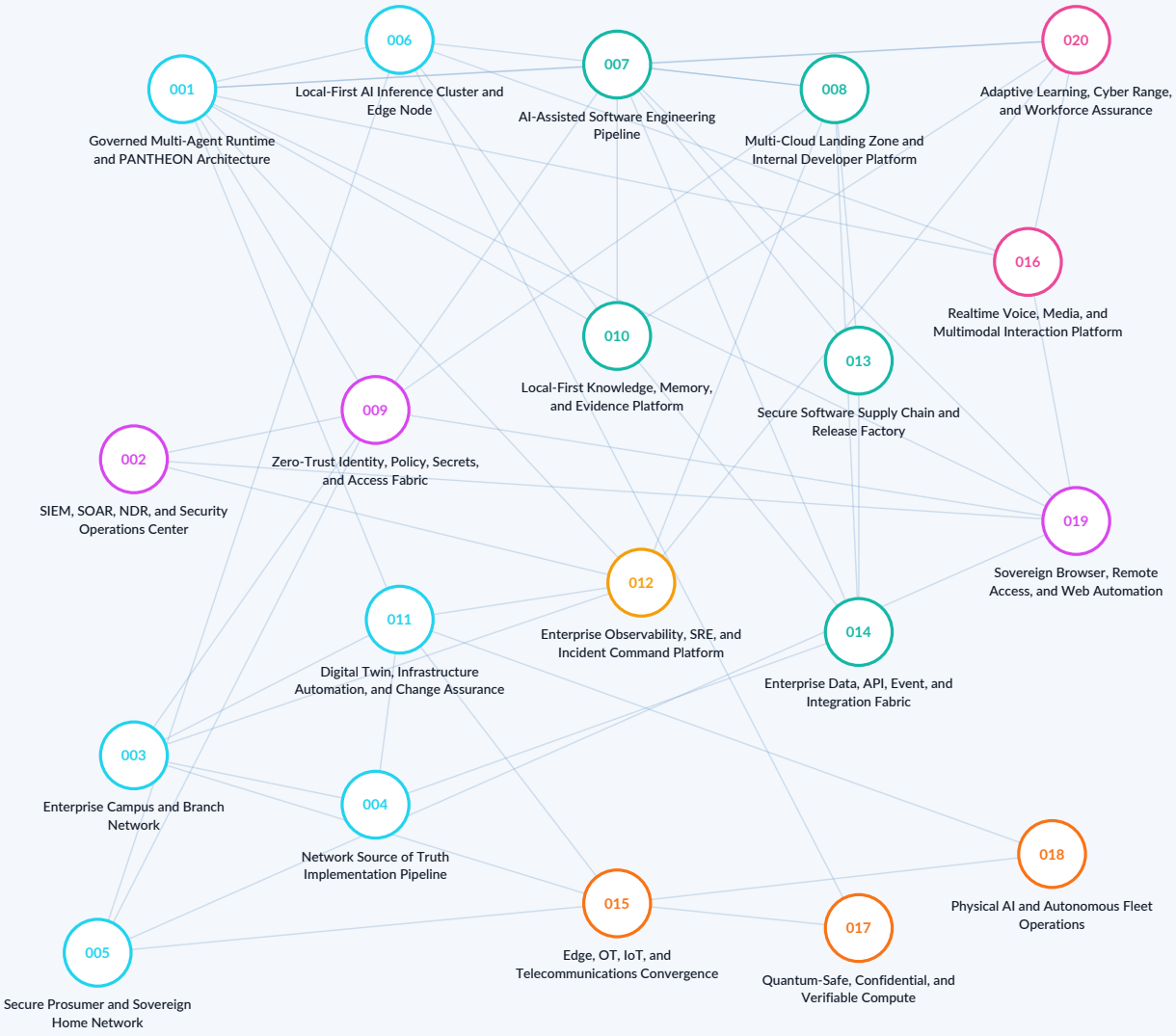
RA-017 FUTURE
Quantum-Safe, Confidential, and Verifiable Compute

RA-018 FUTURE
Physical AI and Autonomous Fleet Operations

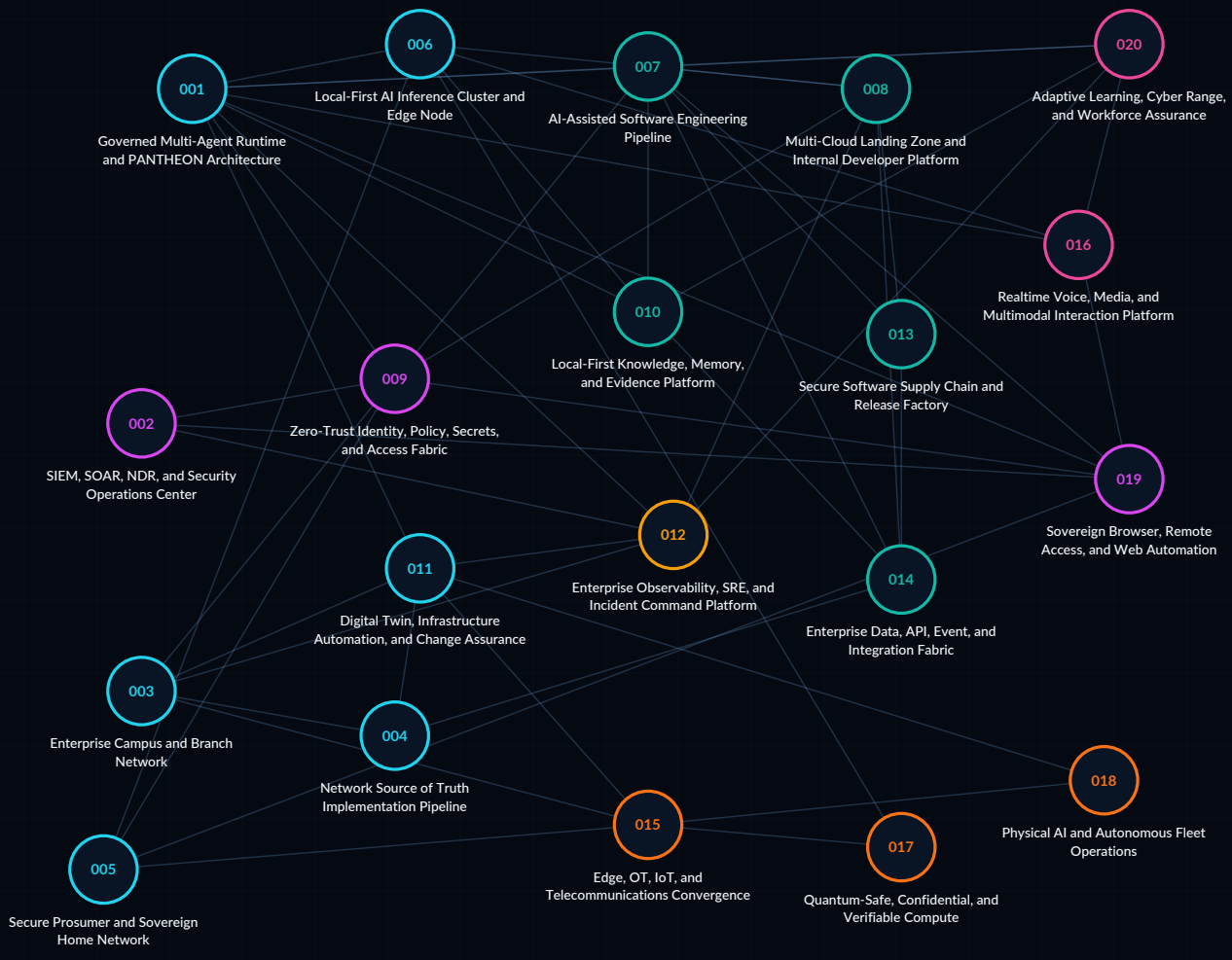
RA-019 FUTURE
Sovereign Browser, Remote Access, and Web Automation

RA-020 FUTURE
Adaptive Learning, Cyber Range, and Workforce Assurance

Architecture landscape by domain



Dependency landscape



DEPENDENCY DOCTRINE

Relationships are directional architecture contracts - not blanket trust. Every integration requires identity, policy, interface, data, version, availability, failure, recovery, observability, and evidence terms.

Adoption sequence across the architecture system

1

FOUNDATION

FND standards + scope + ownership

2

CANONICAL SET

RA-001 through RA-007 establish core operating patterns

3

DOMAIN EXTENSIONS

RA-008 through RA-020 add cloud, identity, data, observability, edge, media, quantum, physical AI, browser, and learning patterns

4

INTEGRATION

Resolve cross-architecture contracts, data authority, failure propagation, and lifecycle dependencies

5

ASSURANCE

Test failure, recovery, observability, rollback, evidence, and acceptance criteria

6

CONTINUOUS REVIEW

Revalidate authorities, versions, operational evidence, assumptions, and residual risk

Decision gates for architecture adoption

OUTCOME	Is the business or mission outcome explicit and measurable?
AUTHORITY	Are human, machine, model, policy, and execution authorities separated?
STATE	Can desired, approved, runtime, observed, and evidence states be reconciled?
FAILURE	Are degraded modes, containment, recovery objectives, and rollback exercised?
OBSERVABILITY	Can operators detect health, policy, security, dependency, and outcome divergence?
EVIDENCE	Can claims be supported by current, reproducible, scope-bound evidence?

A failed mandatory gate cannot be averaged away by maturity scoring, feature breadth, vendor position, or implementation progress.

Reading path

01 / CANONICAL ARCHITECTURE SET

Start with RA-001 through RA-007 to understand the core agent, security, network, local AI, and software-engineering patterns.

02 / FUTURE-DOMAIN ARCHITECTURE SET

Use RA-008 through RA-020 when the target system introduces cloud platforms, identity fabric, knowledge state, digital twins, SRE, supply chain, integration, edge/OT, realtime media, quantum-safe compute, physical AI, sovereign browser automation, or adaptive learning.

03 / ARCHITECTURE LANDSCAPE COMPANION

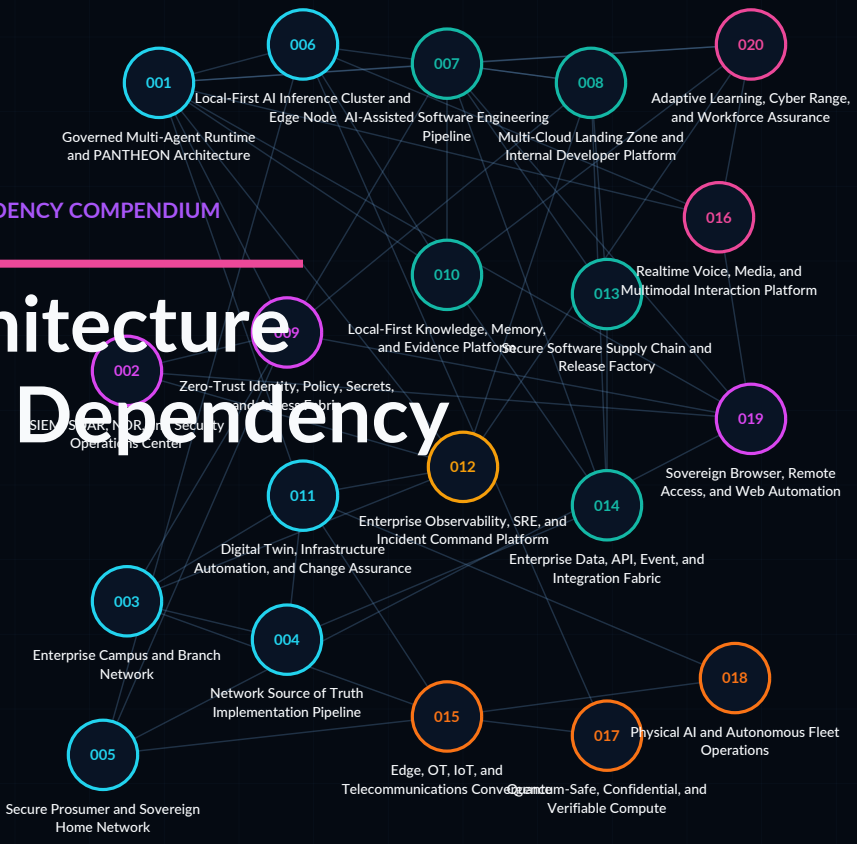
Use MYTHOS-RA-COMP-0001 for cross-architecture dependencies, shared boundaries, integration risks, and portfolio-level relationships.

04 / INDIVIDUAL ARCHITECTURE VERIFICATION

Use each architecture acceptance criteria, metrics, source authority register, and revision record before adoption or implementation claims.

REFERENCE ARCHITECTURE LANDSCAPE / DEPENDENCY COMPENDIUM

Reference Architecture Landscape and Dependency Compendium



Cross-architecture relationships, shared trust boundaries, dependency contracts, integration failure modes, deployment interactions, observability surfaces, and portfolio-level verification.

PERMANENT DOCUMENT ID	ARCHITECTURES	RELATIONSHIPS
MYTHOS-RA-COMP-0001	20	50

Landscape summary

ARTIFICIAL INTELLIGENCE

2 RA-001, RA-006

CYBERSECURITY AND NETWORK SECURITY

3 RA-002, RA-009, RA-019

NETWORK ENGINEERING

4 RA-003, RA-004, RA-005, RA-011

PLATFORM ENGINEERING

3 RA-007, RA-008, RA-013

DATA, KNOWLEDGE, AND EVIDENCE

2 RA-010, RA-014

RELIABILITY AND OPERATIONS

1 RA-012

EDGE, OT, AND FUTURE COMPUTE

3 RA-015, RA-017, RA-018

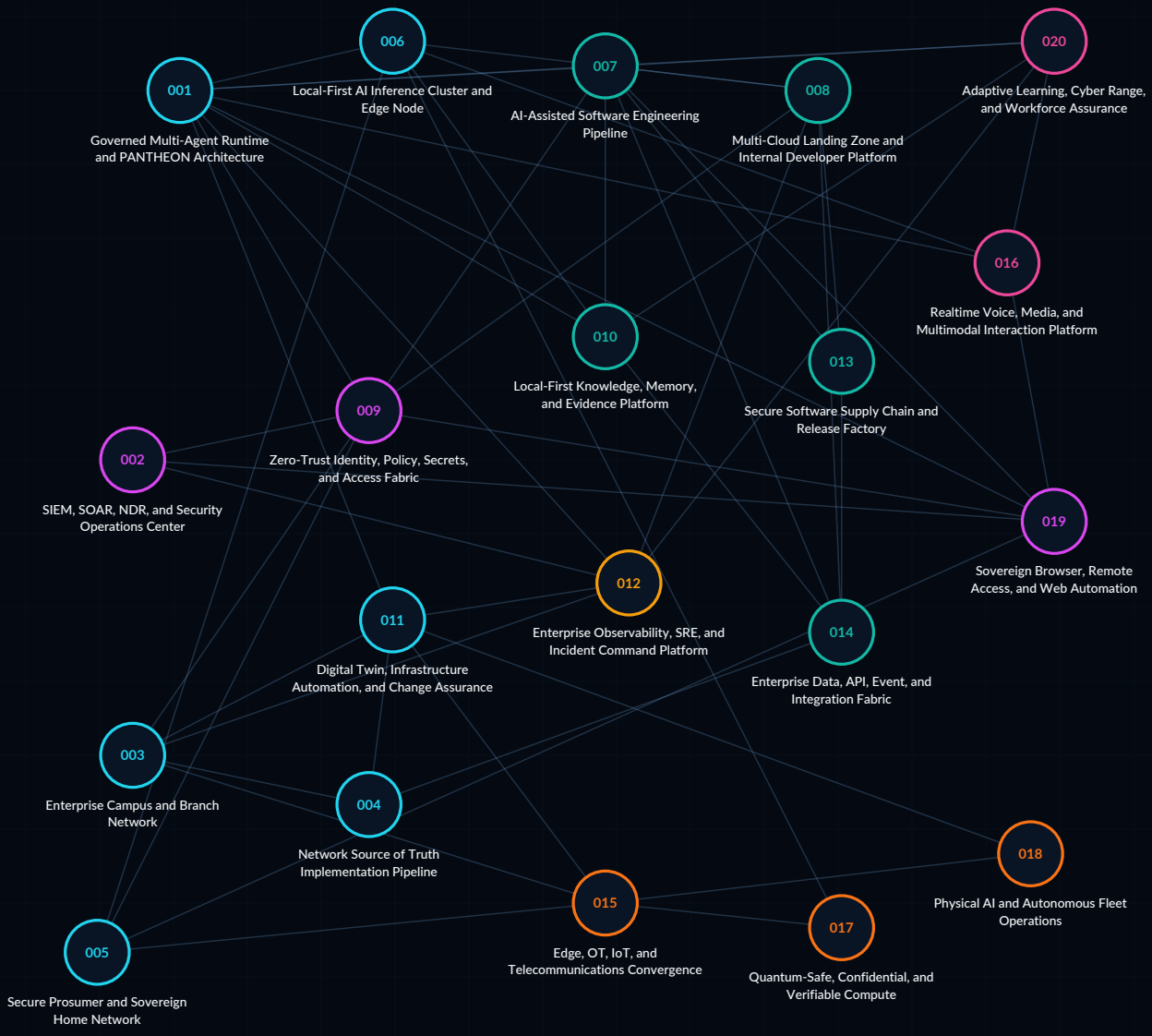
HUMAN SYSTEMS AND LEARNING

2 RA-016, RA-020

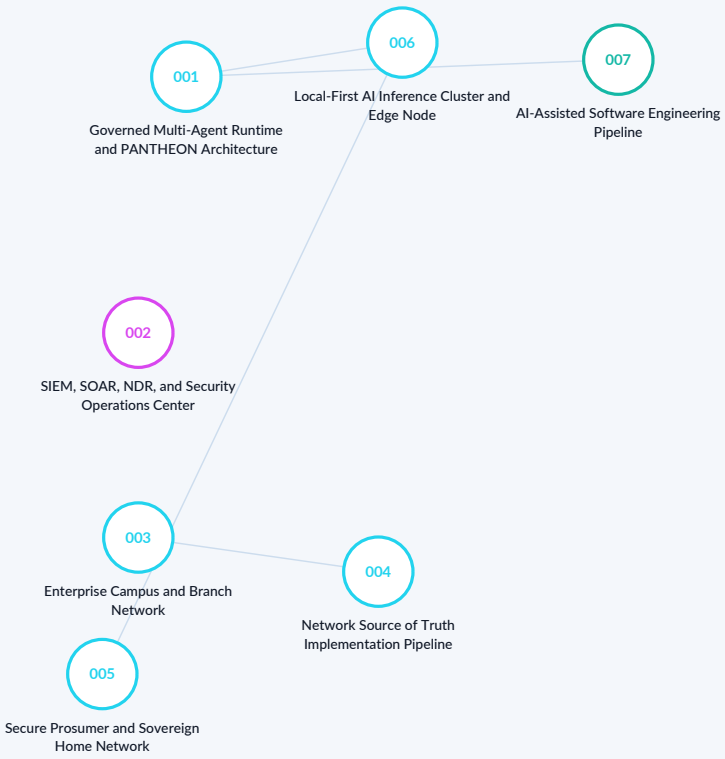
LANDSCAPE POSITION

The architecture system is intentionally compositional. Individual architectures remain independently citable, while shared dependencies are explicit and must not create hidden authority, circular trust, or unbounded failure propagation.

Complete architecture map

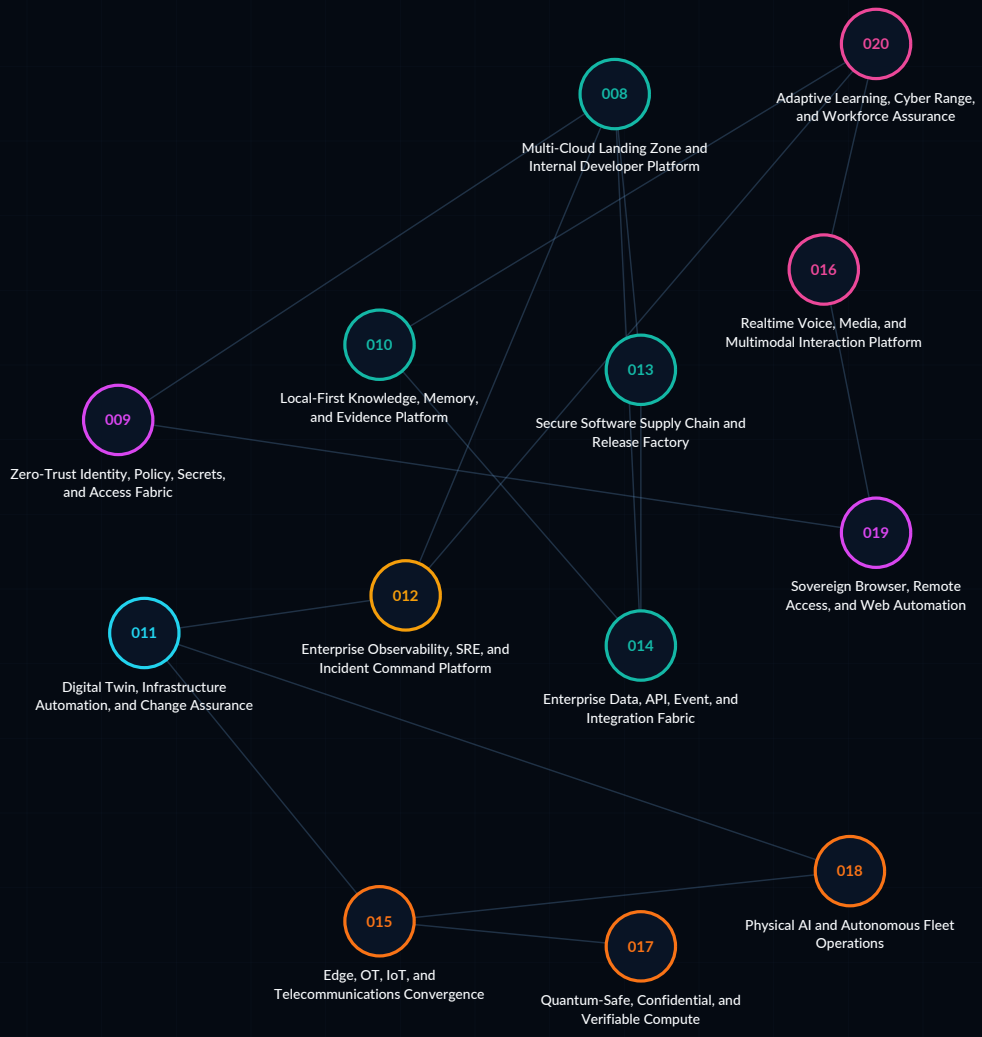


Canonical architecture set

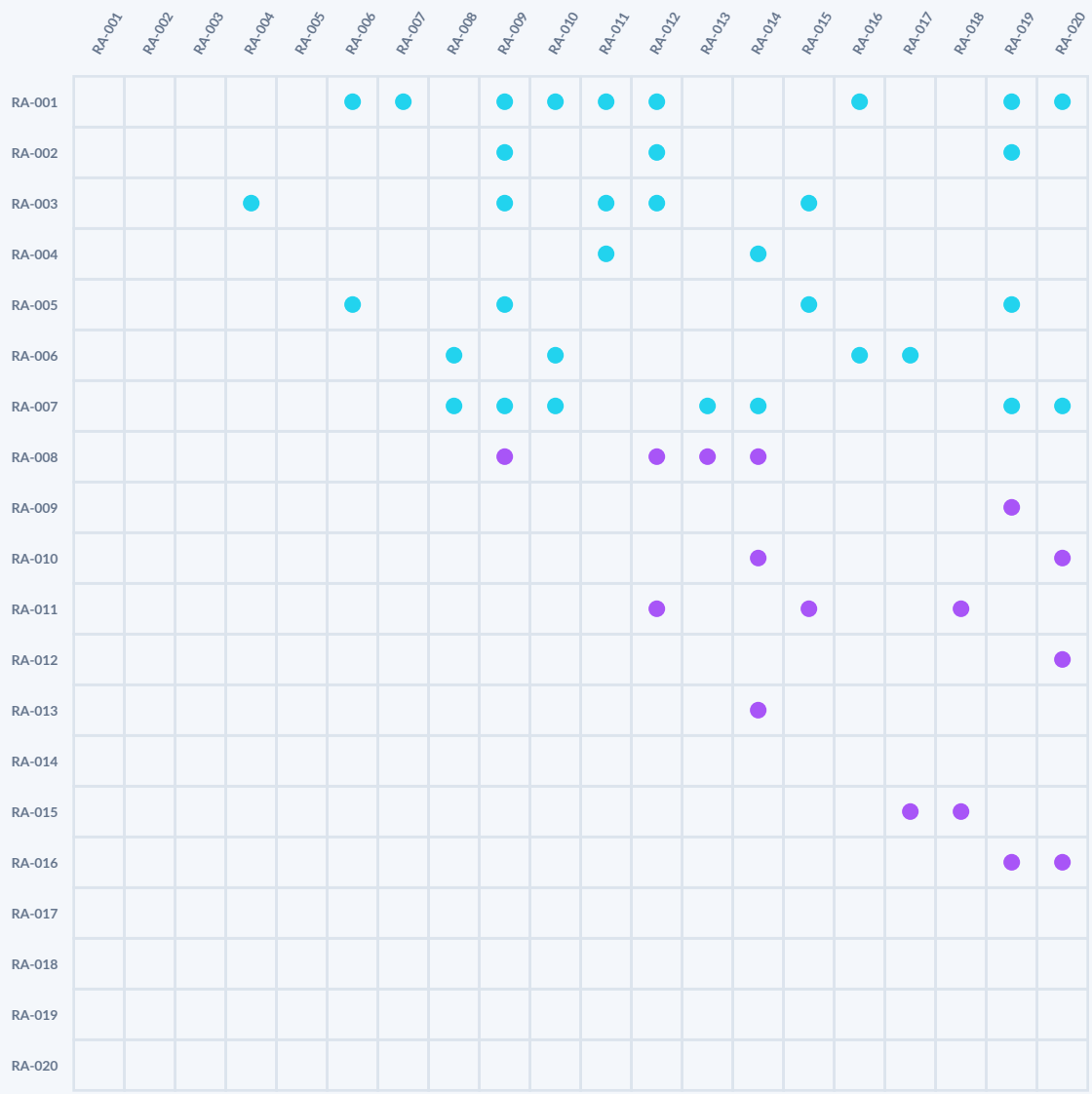


The canonical set establishes the base operating patterns for governed agent runtime, security operations, enterprise networking, authoritative network state, sovereign edge networking, local AI inference, and AI-assisted software engineering.

Approved future-domain architecture set



Directional dependency matrix



A marked cell indicates an explicit integration/dependency relationship from row architecture to column architecture. Relationships are directional and require interface, authority, data, failure, recovery, and evidence contracts.

Shared trust-boundary doctrine

IDENTITY	Authenticate and bind human, service, workload, device, agent, and external identities.
AUTHORITY	Authorize the exact action, scope, duration, resource, and consequence class.
DATA	Classify data, purpose, provenance, retention, residency, privacy, and egress.
PROTOCOL	Pin interface, schema, version, encryption, validation, retry, timeout, and compatibility.
FAILURE	Define loss, degradation, backpressure, isolation, containment, rollback, and recovery.
EVIDENCE	Record decision, versions, state changes, output, validation, timing, owner, and disposition.

Cross-architecture failure propagation

IDENTITY / POLICY

RA-009 impairment can block or weaken downstream execution, remote access, software delivery, and browser automation.

OBSERVABILITY

RA-012 impairment can hide degradation across RA-003, RA-008, RA-011, RA-015, RA-018, and RA-020.

DATA / INTEGRATION

RA-014 impairment can stall knowledge, supply chain, automation, telemetry, and cross-domain workflows.

EDGE / CONNECTIVITY

RA-015 impairment can isolate physical systems, autonomous fleets, local compute, or field telemetry.

SOFTWARE SUPPLY CHAIN

RA-013 compromise can propagate trusted-looking artifacts into RA-007, RA-008, RA-019, and dependent runtime environments.

AGENT / MODEL RUNTIME

RA-001 or RA-006 faults can corrupt decisions or automation unless authority, deterministic execution, verification, and evidence remain independently enforceable.

Portfolio observability and verification

OUTCOME

Mission and user outcomes

SERVICE

SLOs, capacity, performance

CONTROL

Identity, policy, approvals

STATE

Desired, runtime, observed

SECURITY

Threats, trust, abuse

DEPENDENCY

External service and supply chain

RECOVERY

Restore, failover, rollback

EVIDENCE

Trace, artifacts, receipts, decisions

Verification is architecture-specific, but acceptance always requires defined scope, testable criteria, relevant negative/degraded/recovery cases, current evidence, accountable ownership, and explicit residual risk.

Lifecycle, revision, and limitations

PERMANENT IDENTITY	Architecture IDs remain stable across ordinary revision.
CANDIDATE STATUS	Architecture publications are not certification, legal compliance, or proof of implementation.
VERSIONED CHANGE	Material architecture changes require rationale, affected relationship analysis, source refresh, and regression verification.
FRESHNESS	External authorities and implementation assumptions must be revalidated at adoption and scheduled review.
SCOPE LIMITS	Reference architectures are patterns. Product, sector, jurisdiction, workload, and environment constraints still require engineering decisions.