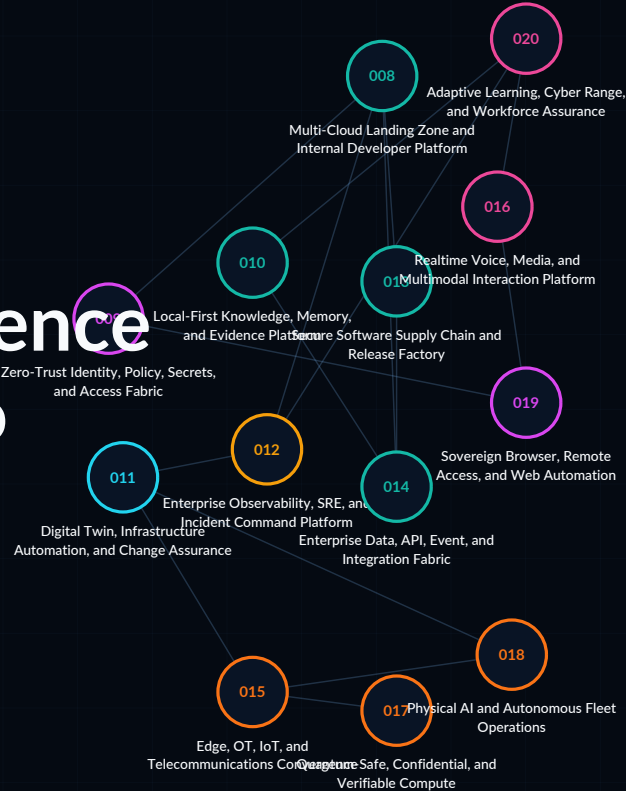


Future-Domain Reference Architecture Portfolio



13 permanent candidate reference architectures

Portfolio register and page map

RA-008 Multi-Cloud Landing Zone and Internal Developer Platform 3	RA-009 Zero-Trust Identity, Policy, Secrets, and Access Fabric 30
RA-010 Local-First Knowledge, Memory, and Evidence Platform 57	RA-011 Digital Twin, Infrastructure Automation, and Change Assurance 84
RA-012 Enterprise Observability, SRE, and Incident Command Platform 111	RA-013 Secure Software Supply Chain and Release Factory 138
RA-014 Enterprise Data, API, Event, and Integration Fabric 165	RA-015 Edge, OT, IoT, and Telecommunications Convergence 192
RA-016 Realtime Voice, Media, and Multimodal Interaction Platform 219	RA-017 Quantum-Safe, Confidential, and Verifiable Compute 246
RA-018 Physical AI and Autonomous Fleet Operations 273	RA-019 Sovereign Browser, Remote Access, and Web Automation 300
RA-020 Adaptive Learning, Cyber Range, and Workforce Assurance 327	



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.

Architecture identity and operating position

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

SYSTEM CONTEXT



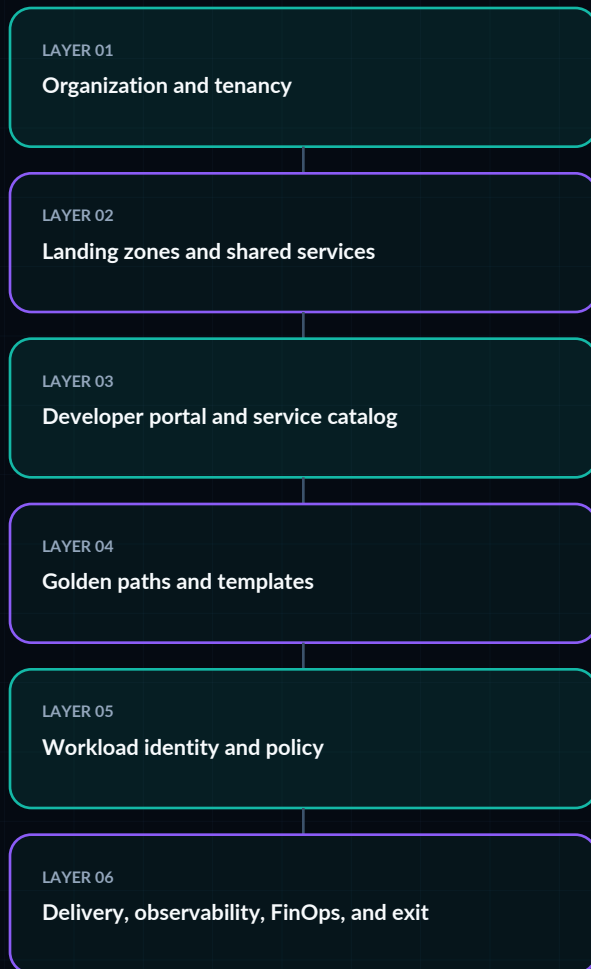
EXECUTIVE OUTCOMES

- 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.

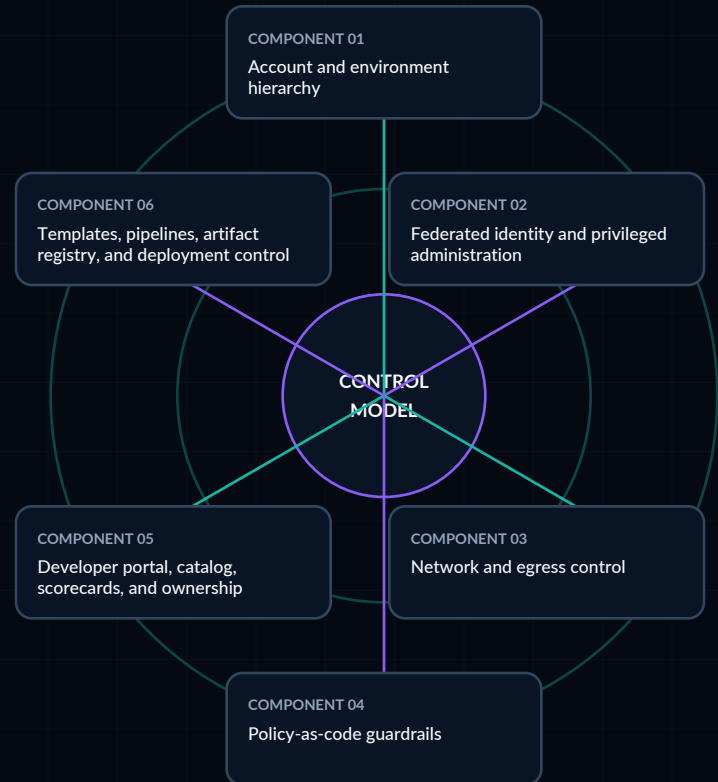
ARCHITECTURE DOCTRINE

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



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



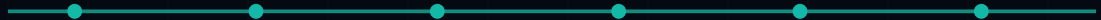
EVIDENCE THREAD

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

State, data, authority, and trust flow



DESIRED STATE



RUNTIME STATE



OBSERVED STATE



EVIDENCE STATE



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

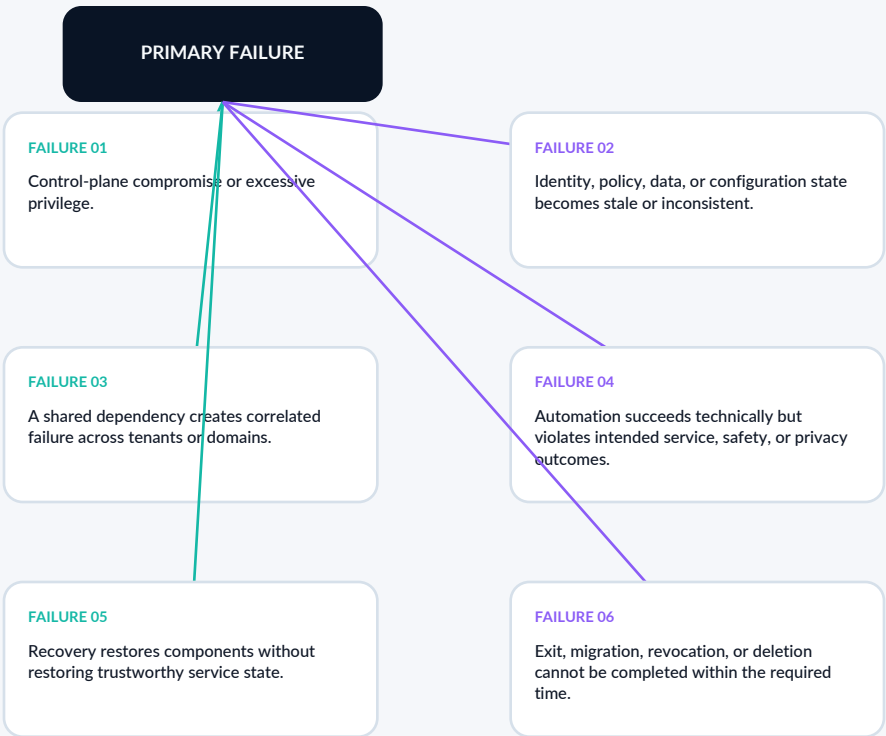
Mutable runtime state versus authoritative evidence

TB-06

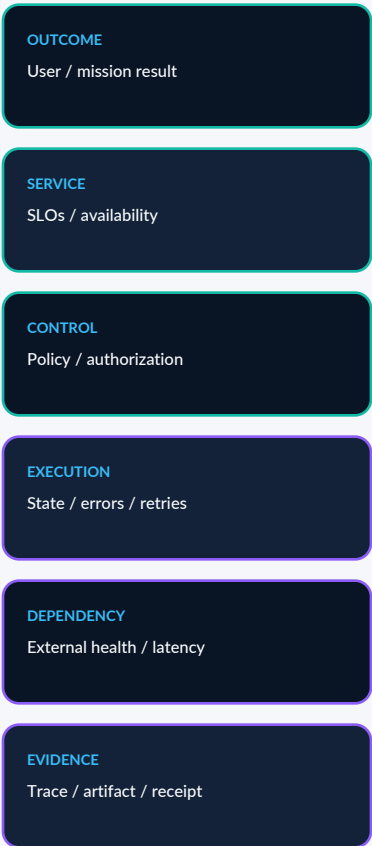
Normal operation versus emergency or break-glass authority

Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK



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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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.

Architecture identity and operating position

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

SYSTEM CONTEXT



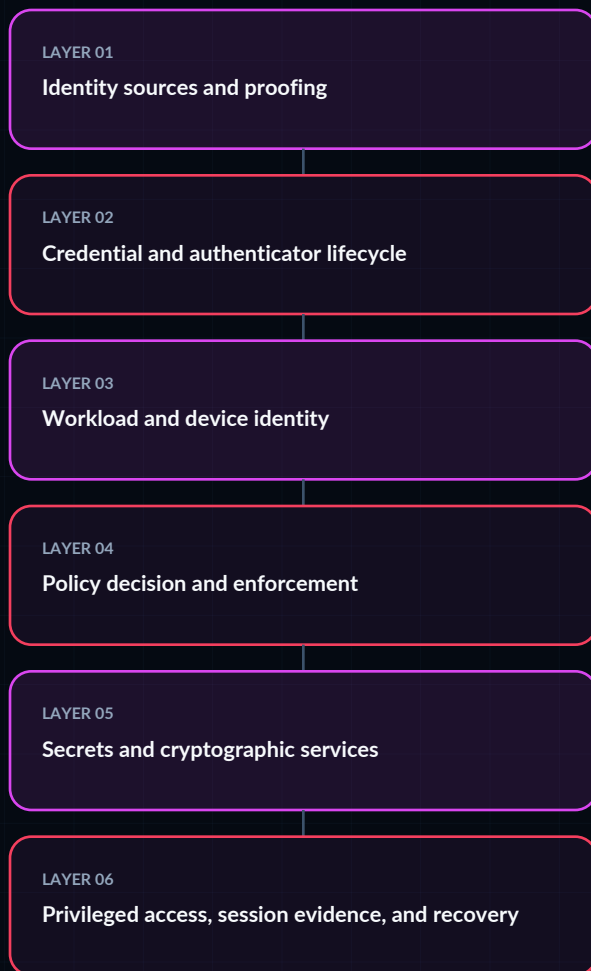
EXECUTIVE OUTCOMES

- 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.

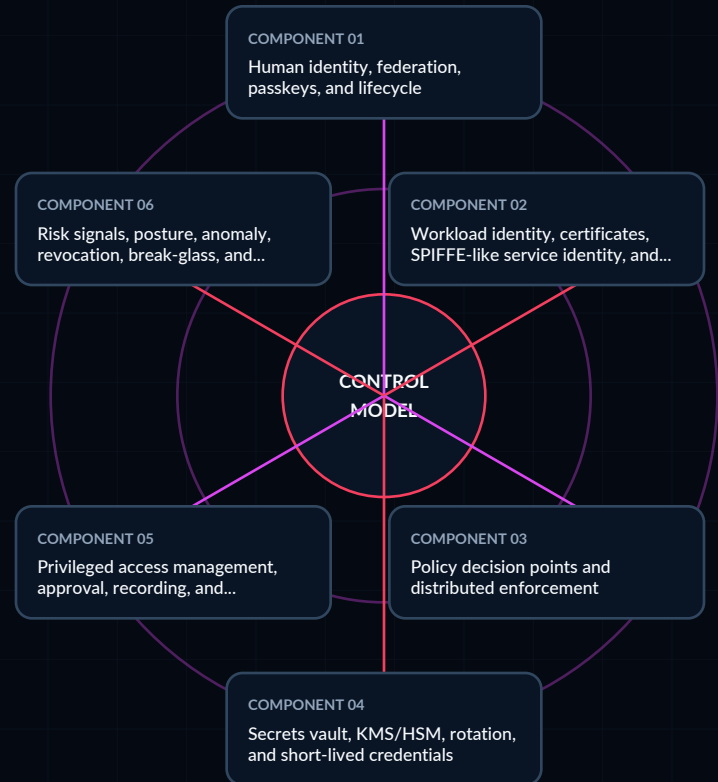
ARCHITECTURE DOCTRINE

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

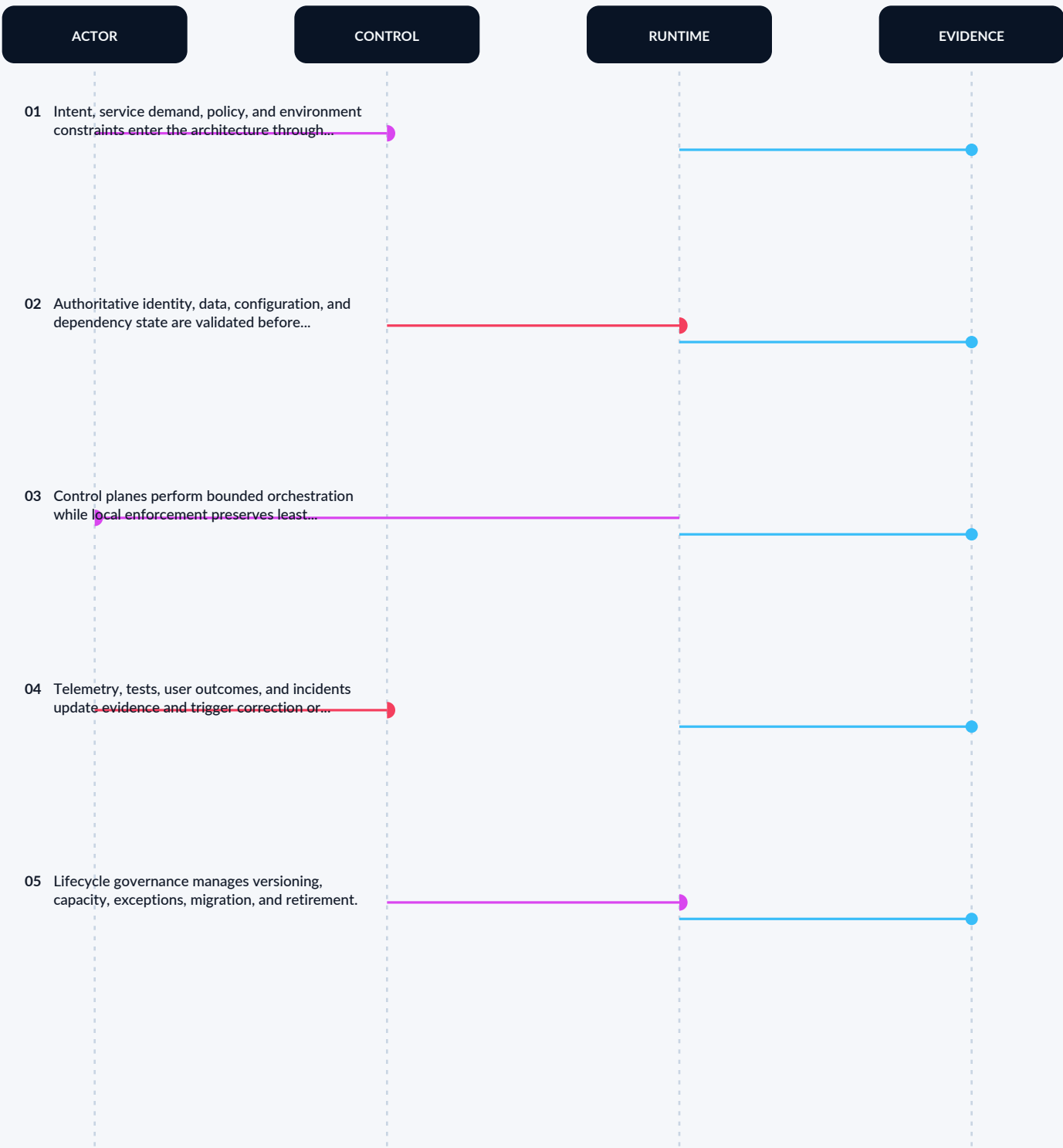


CORE COMPONENTS



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

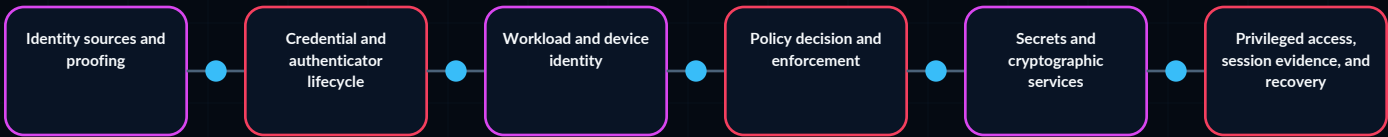
Primary sequence and control flow



EVIDENCE THREAD

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

State, data, authority, and trust flow



DESIRED STATE



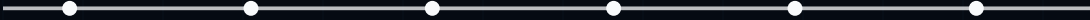
RUNTIME STATE



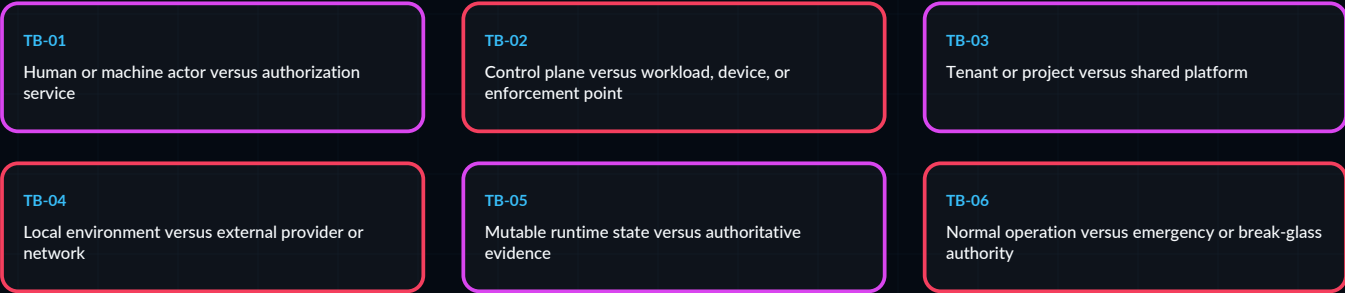
OBSERVED STATE



EVIDENCE STATE

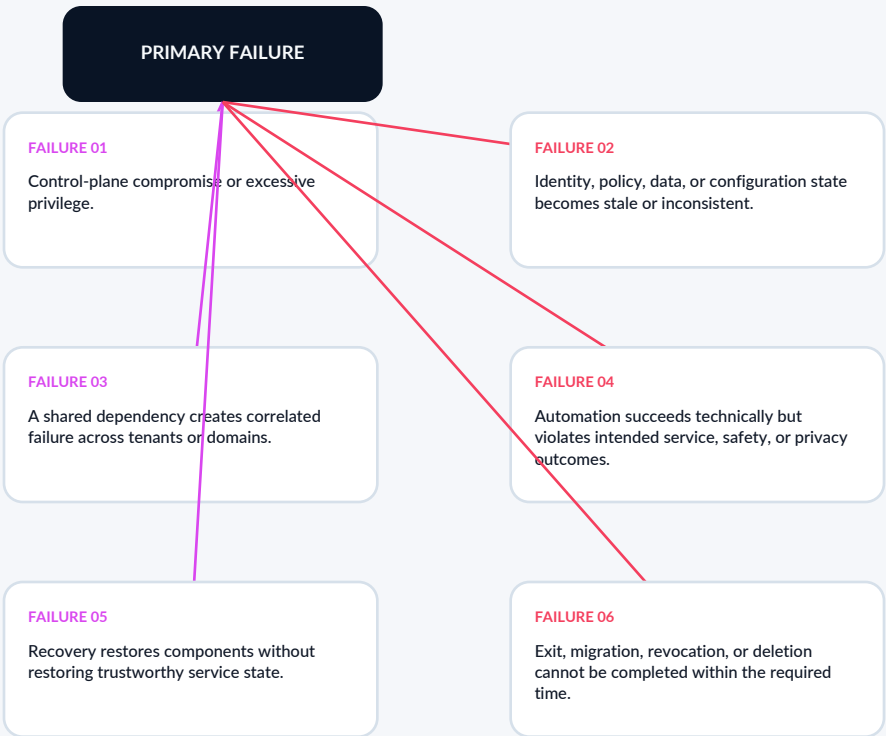


TRUST BOUNDARY REGISTER

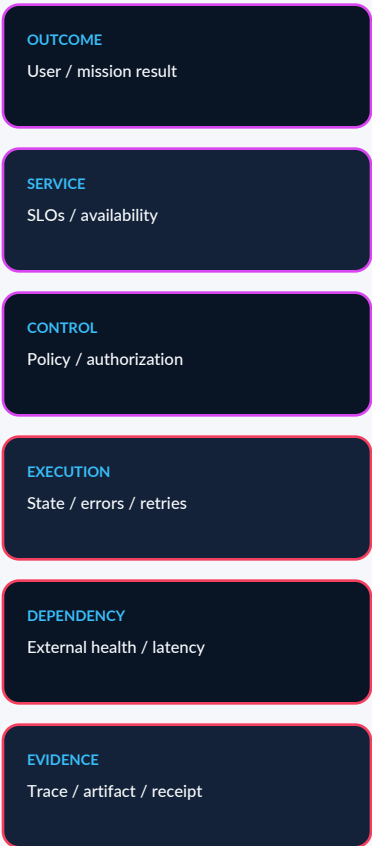


Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK



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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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

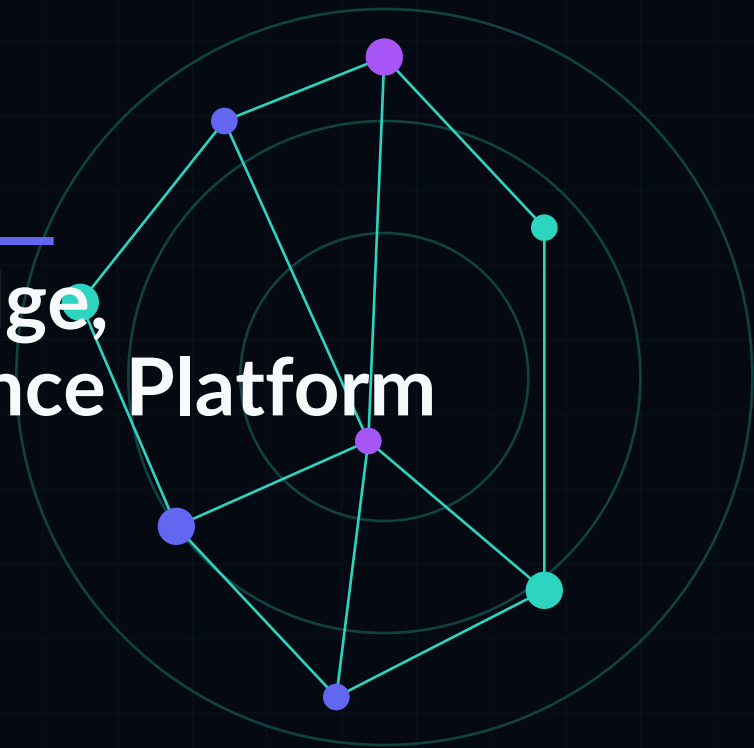
REVISION RECORD

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

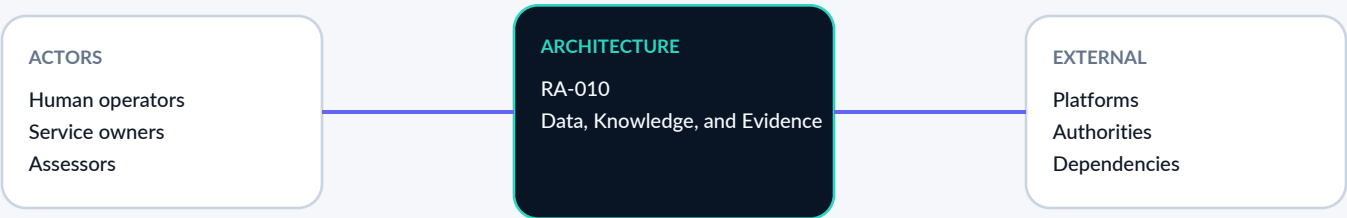


A local-first platform for authoritative knowledge, retrieval, context, memory, provenance, evidence, retention, correction, and controlled synchronization.

Architecture identity and operating position

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

SYSTEM CONTEXT



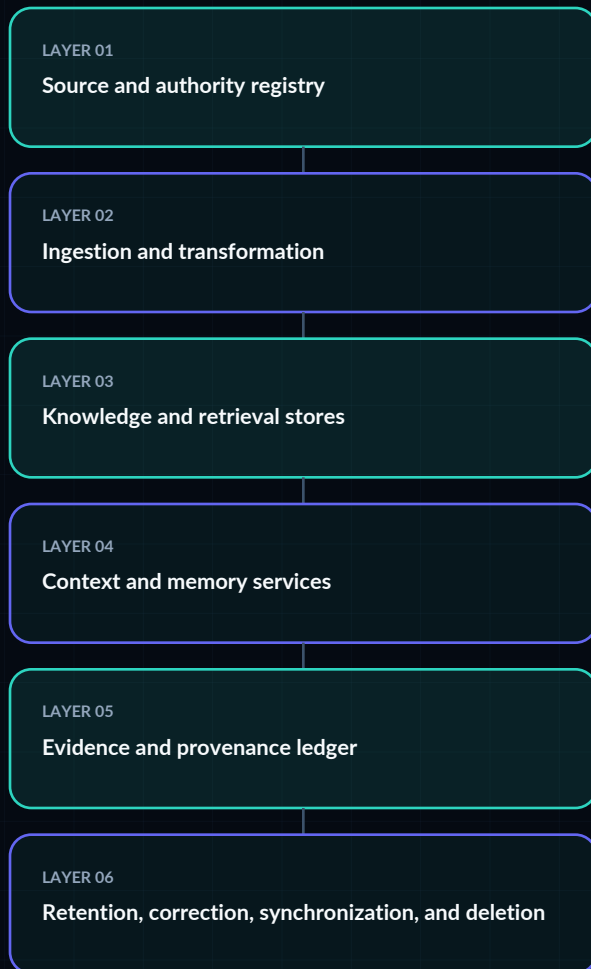
EXECUTIVE OUTCOMES

- 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.

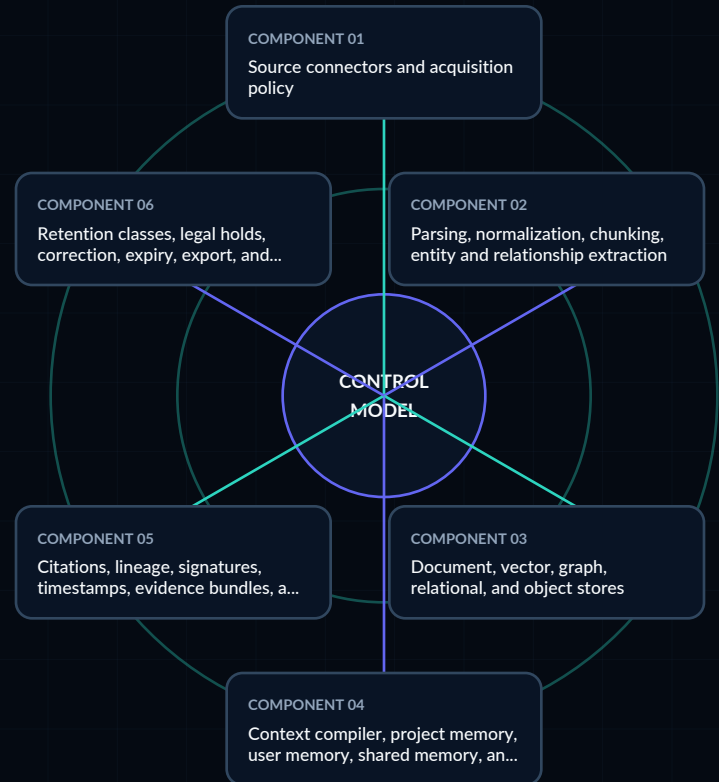
ARCHITECTURE DOCTRINE

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



CORE COMPONENTS



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

Primary sequence and control flow



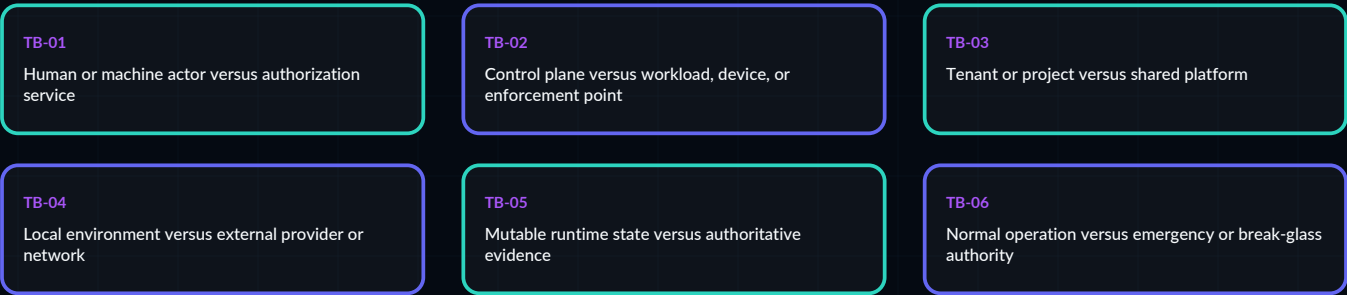
EVIDENCE THREAD

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

State, data, authority, and trust flow

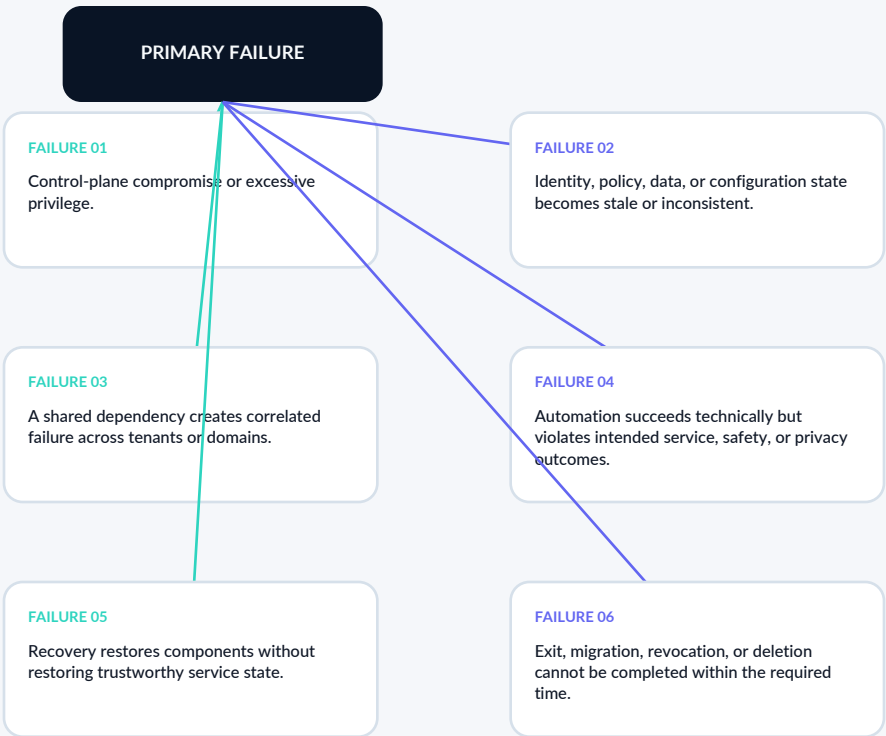


TRUST BOUNDARY REGISTER

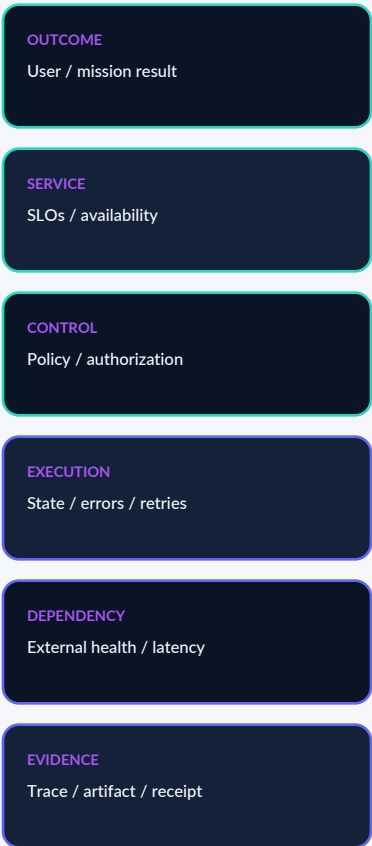


Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK



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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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

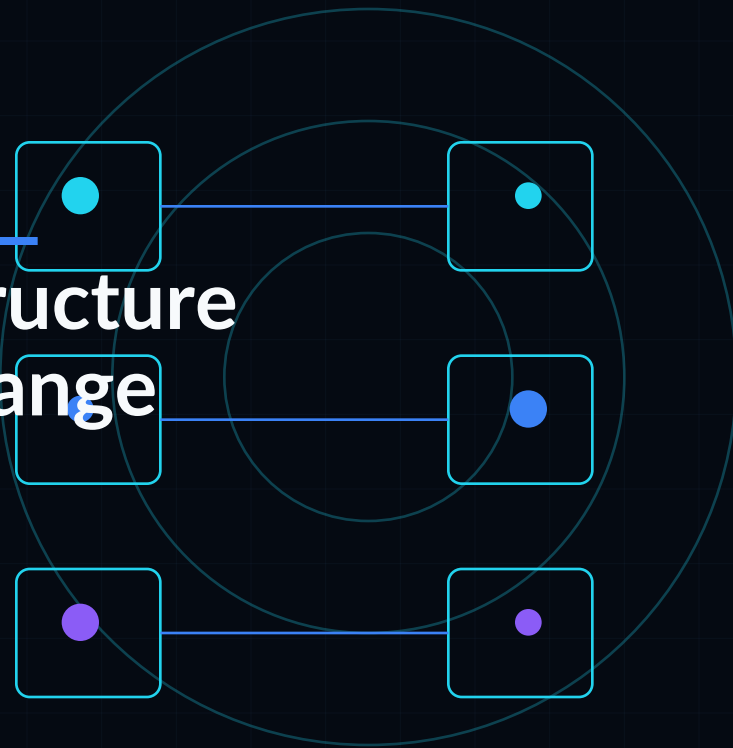
REVISION RECORD

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.



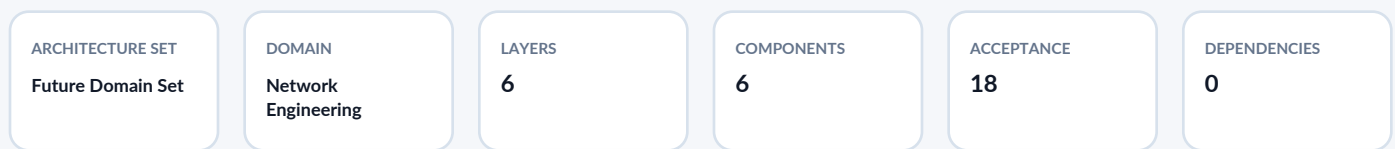
REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Digital Twin, Infrastructure Automation, and Change Assurance

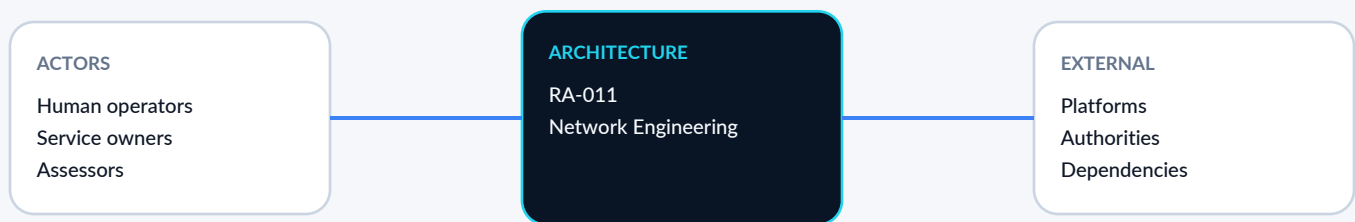


A digital-twin and automation architecture that combines authoritative state, simulation, policy, controlled execution, telemetry, reconciliation, and evidence across infrastructure domains.

Architecture identity and operating position



SYSTEM CONTEXT



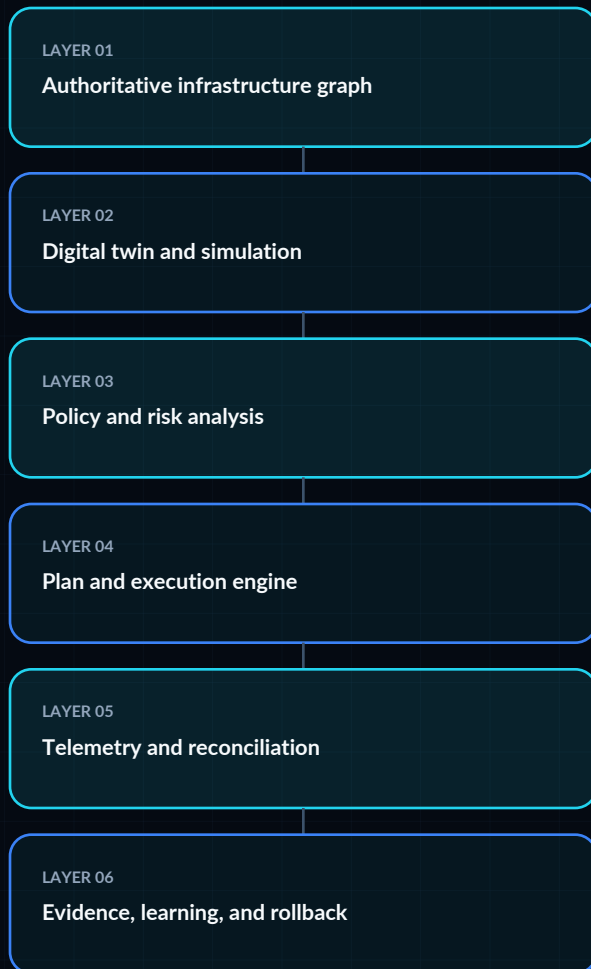
EXECUTIVE OUTCOMES

- 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.

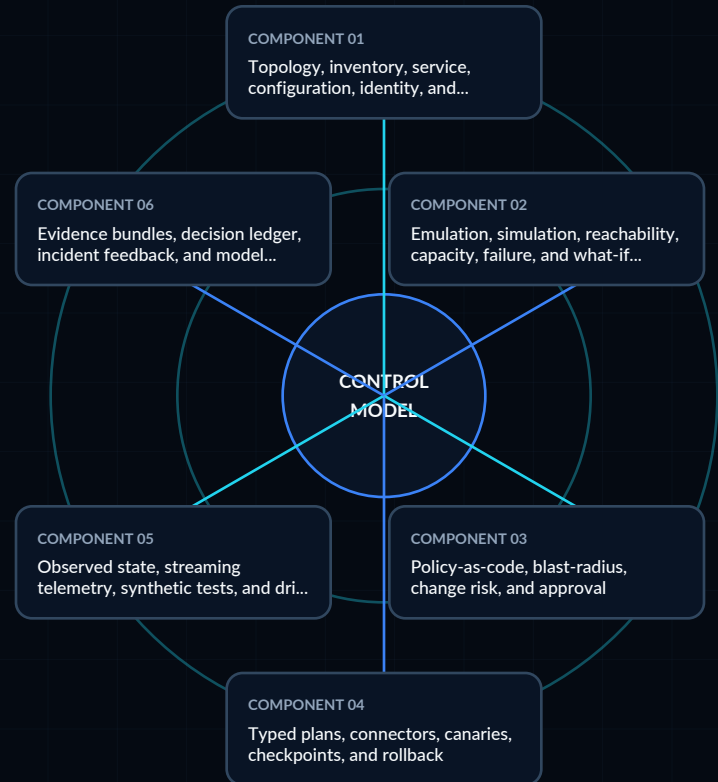
ARCHITECTURE DOCTRINE

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

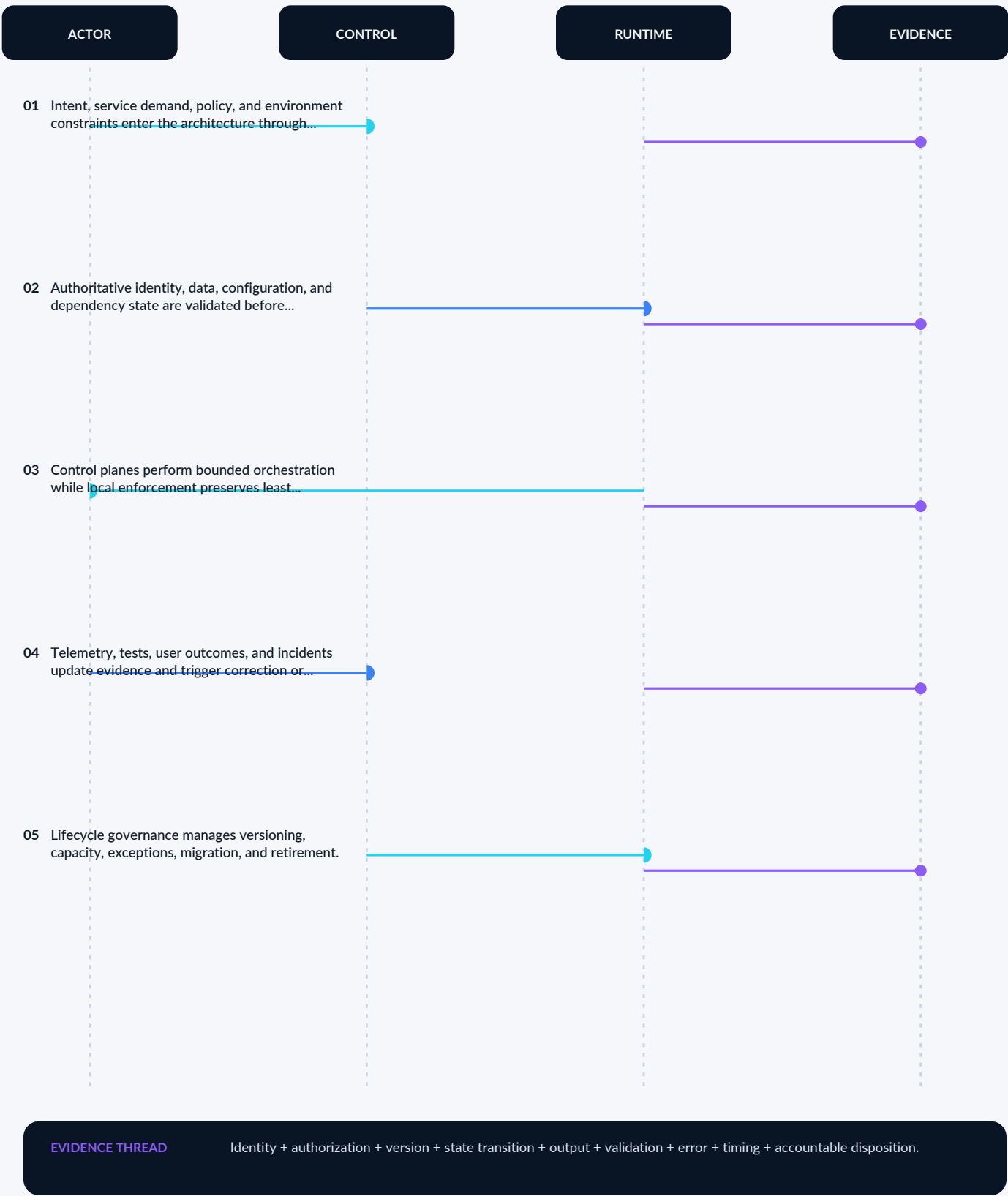


CORE COMPONENTS



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



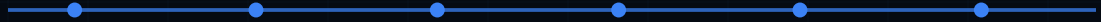
State, data, authority, and trust flow



DESIRED STATE



RUNTIME STATE



OBSERVED STATE



EVIDENCE STATE



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

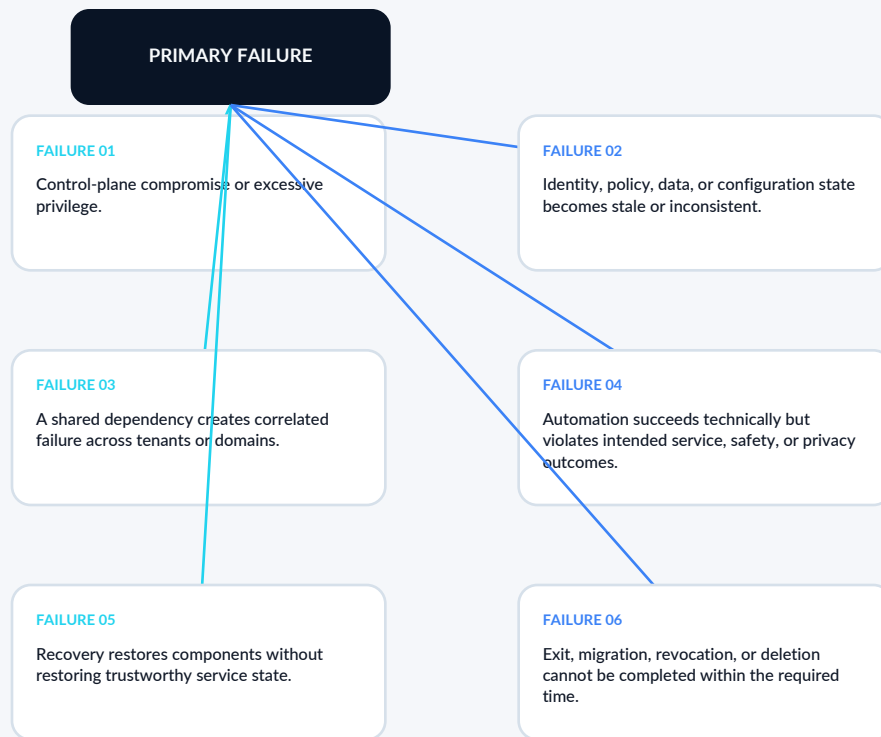
Mutable runtime state versus authoritative evidence

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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-NET-0005, MYTHOS-NET-0006, MYTHOS-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.



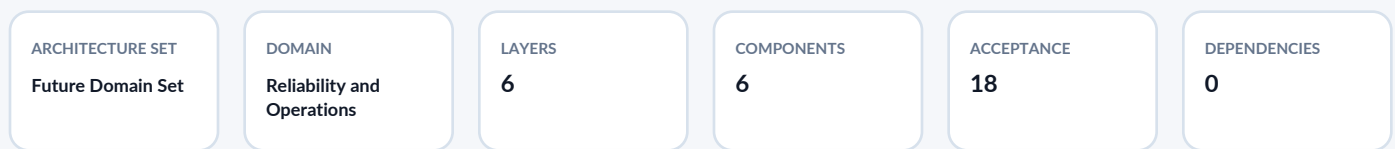
REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Enterprise Observability, SRE, and Incident Command Platform

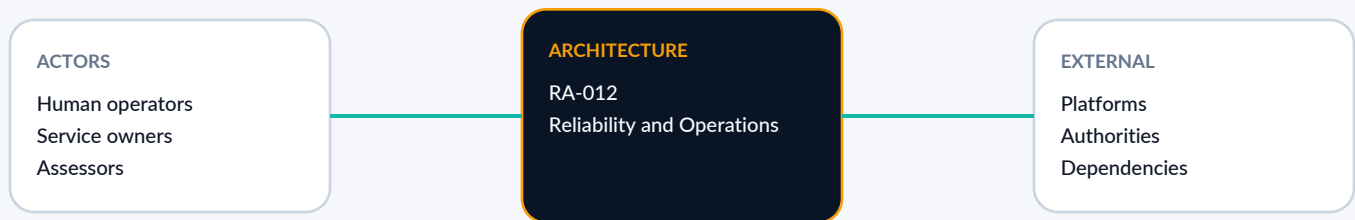


An enterprise observability and reliability architecture joining telemetry, SLOs, service ownership, incident command, automation, evidence, and continuous improvement.

Architecture identity and operating position



SYSTEM CONTEXT



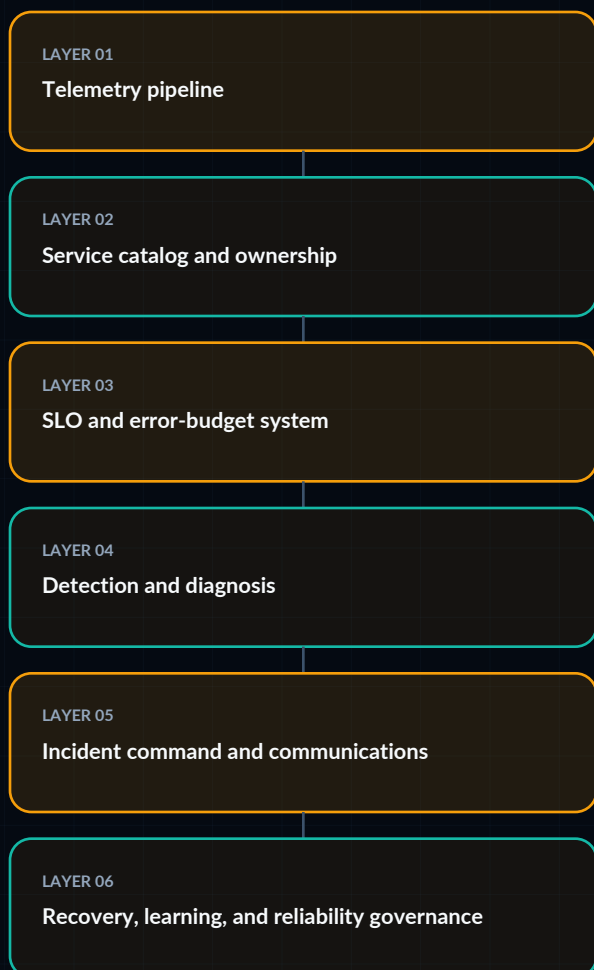
EXECUTIVE OUTCOMES

- 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.

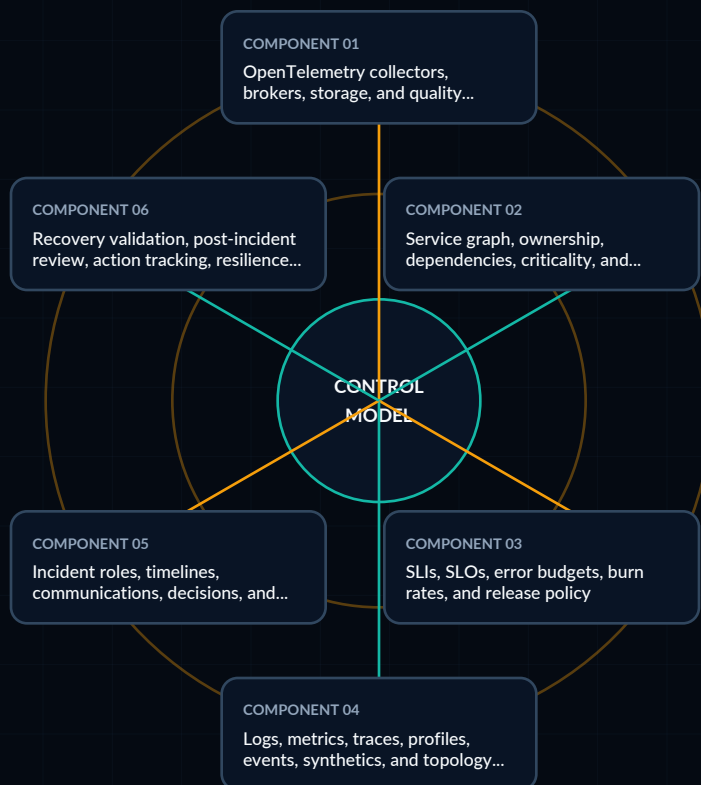
ARCHITECTURE DOCTRINE

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



CORE COMPONENTS



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

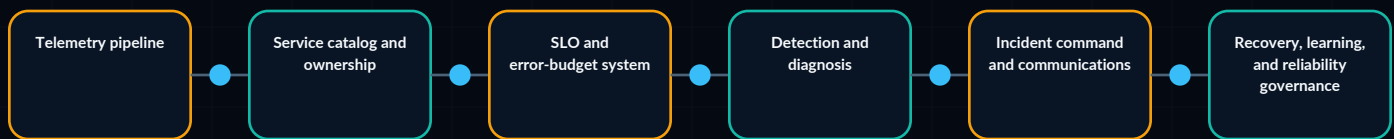
Primary sequence and control flow



EVIDENCE THREAD

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

State, data, authority, and trust flow



DESIRED STATE



RUNTIME STATE



OBSERVED STATE



EVIDENCE STATE



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

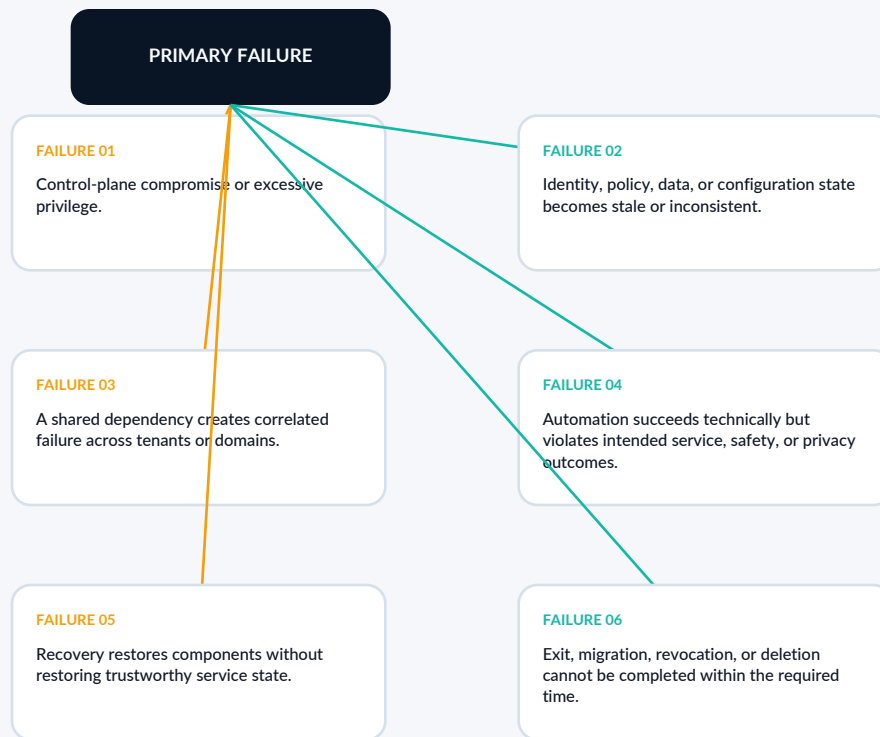
Mutable runtime state versus authoritative evidence

TB-06

Normal operation versus emergency or break-glass authority

Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK



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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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.

Architecture identity and operating position

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

SYSTEM CONTEXT



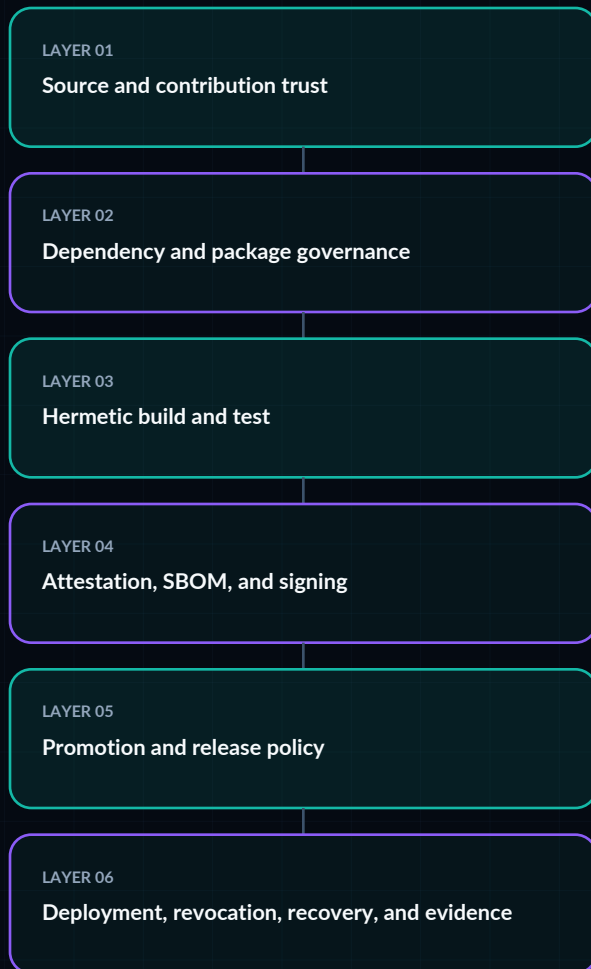
EXECUTIVE OUTCOMES

- 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.

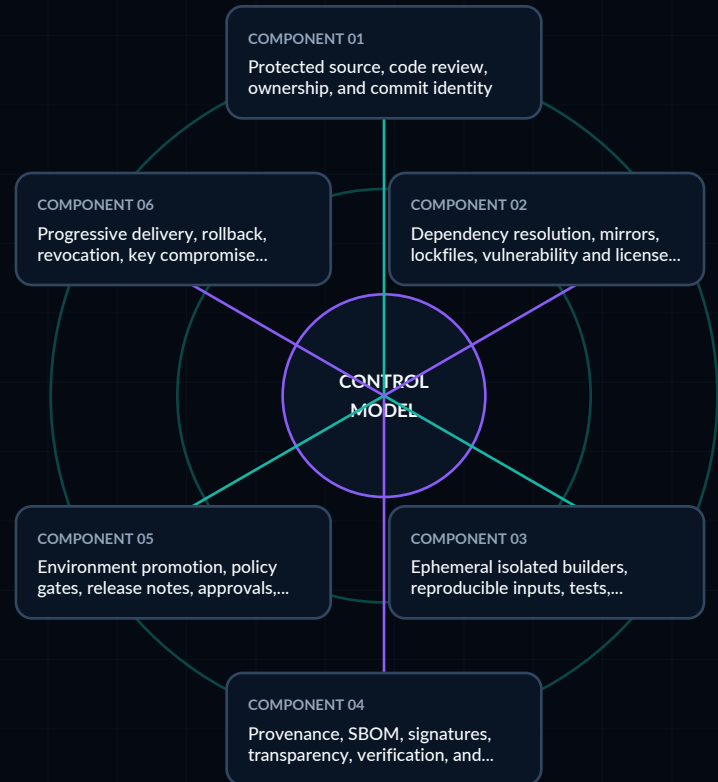
ARCHITECTURE DOCTRINE

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



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.

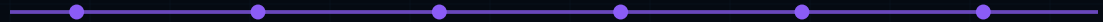
State, data, authority, and trust flow



DESIRED STATE



RUNTIME STATE



OBSERVED STATE



EVIDENCE STATE



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

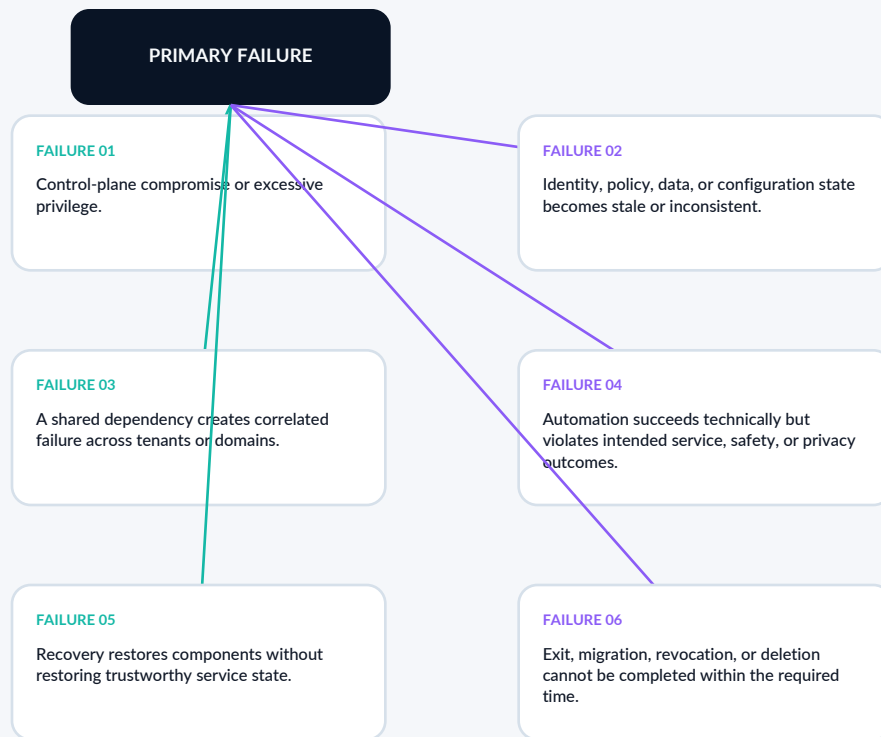
Mutable runtime state versus authoritative evidence

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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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

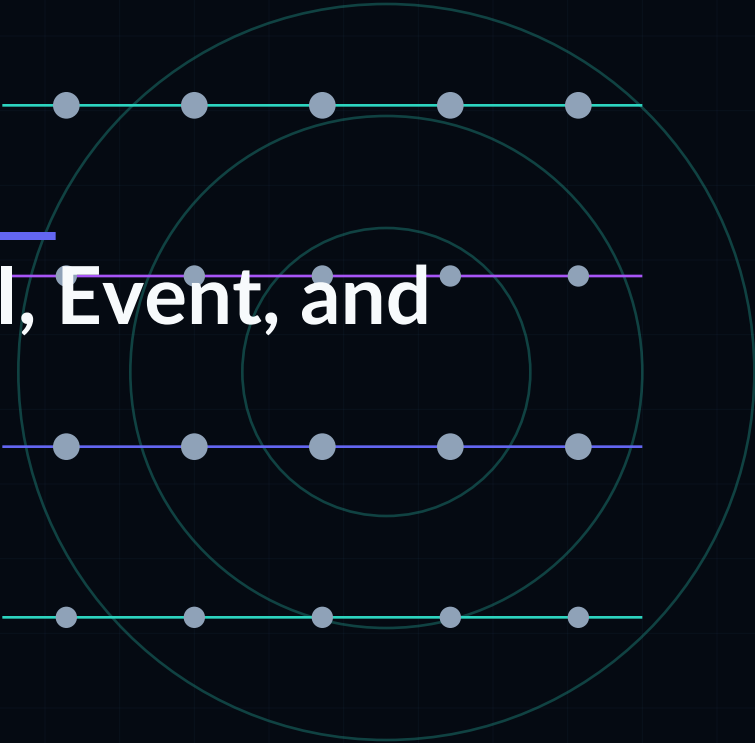
REVISION RECORD

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

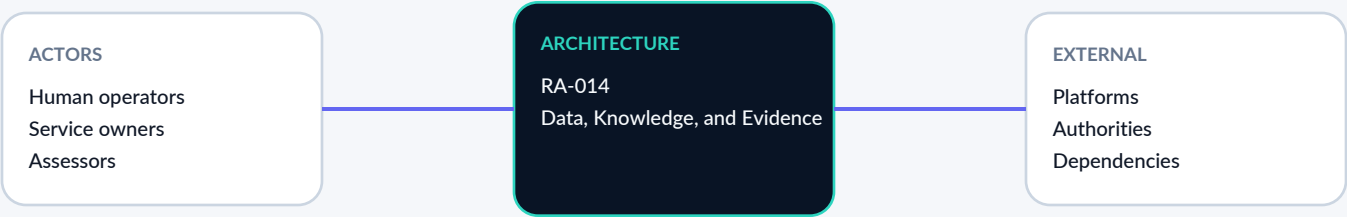


A governed integration fabric for data products, APIs, events, schemas, workflows, lineage, quality, privacy, observability, and resilient cross-domain exchange.

Architecture identity and operating position

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

SYSTEM CONTEXT



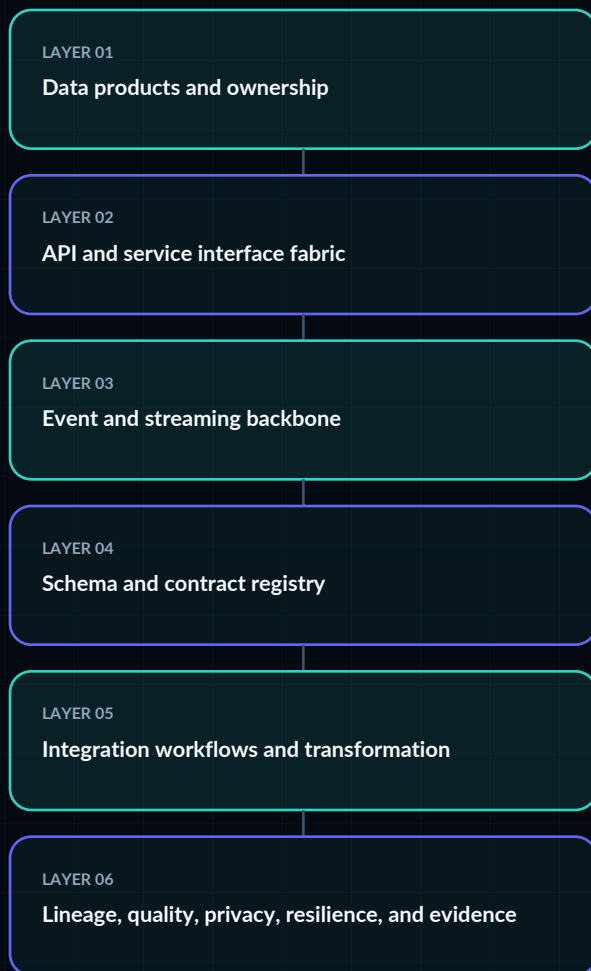
EXECUTIVE OUTCOMES

- 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



CORE COMPONENTS



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

Primary sequence and control flow



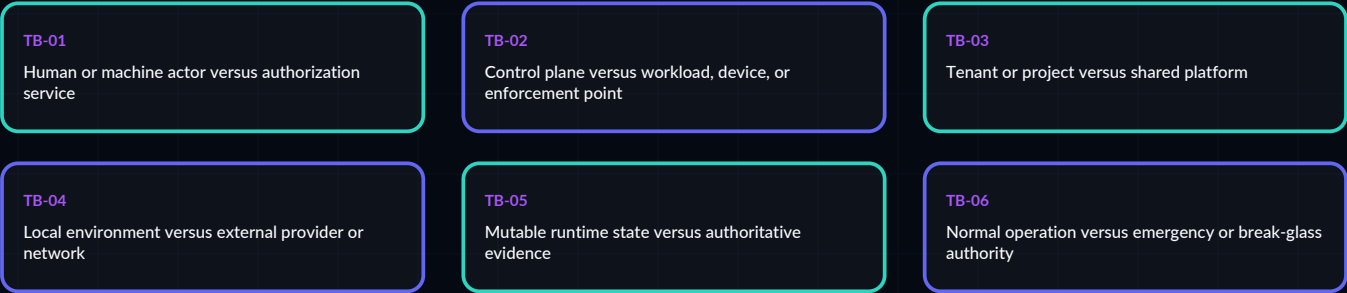
EVIDENCE THREAD

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

State, data, authority, and trust flow

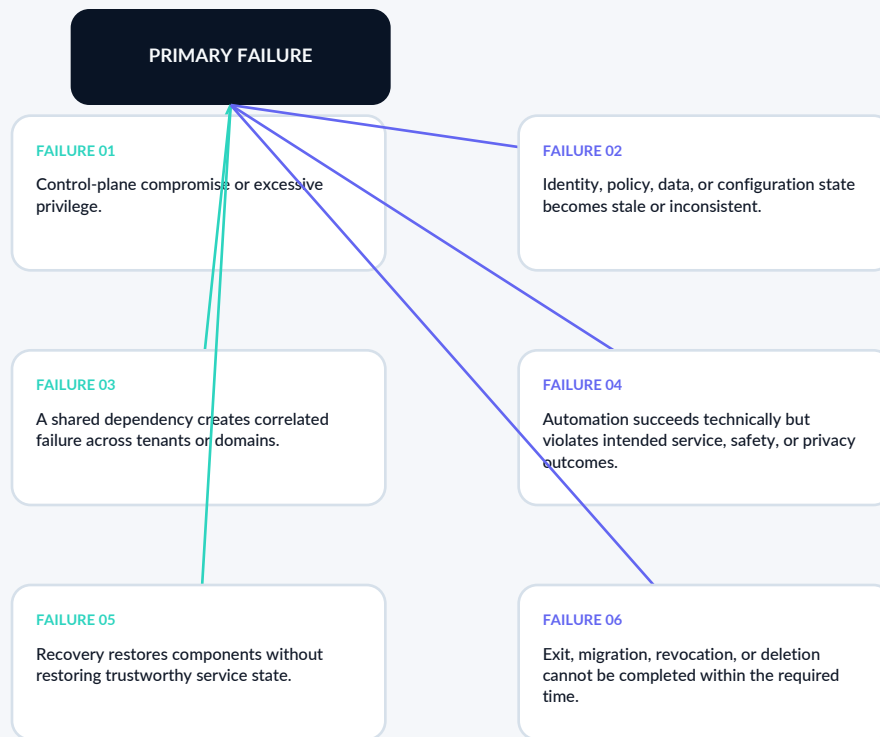


TRUST BOUNDARY REGISTER



Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK

OUTCOME
User / mission result

SERVICE
SLOs / availability

CONTROL
Policy / authorization

EXECUTION
State / errors / retries

DEPENDENCY
External health / latency

EVIDENCE
Trace / artifact / receipt

RECOVERY RULE

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

Deployment profiles and implementation roadmap

DEPLOYMENT PROFILES

PROFILE 01

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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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.



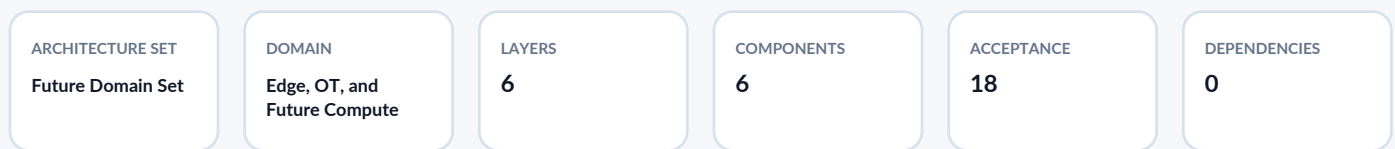
REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

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.

Architecture identity and operating position



SYSTEM CONTEXT



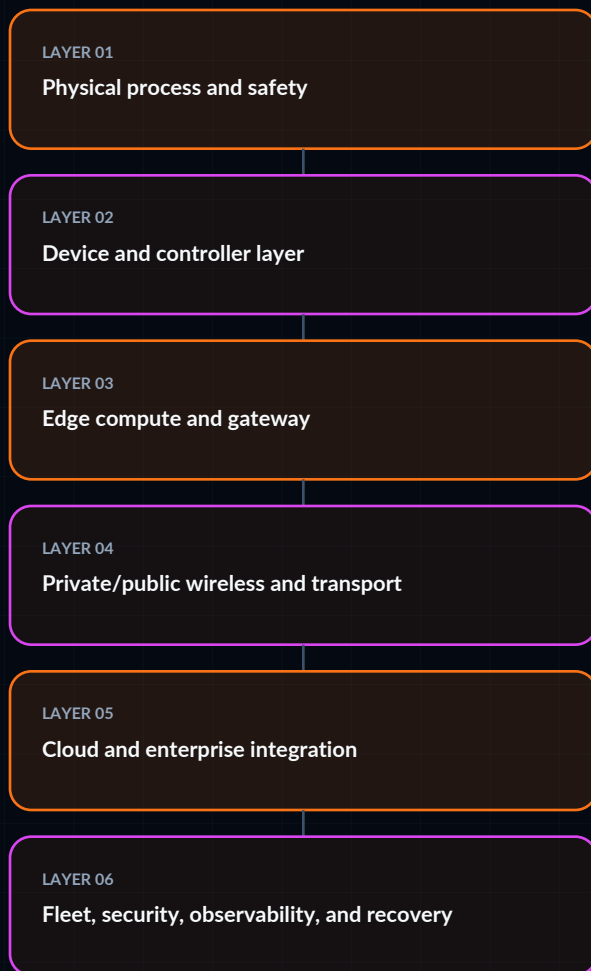
EXECUTIVE OUTCOMES

- 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.

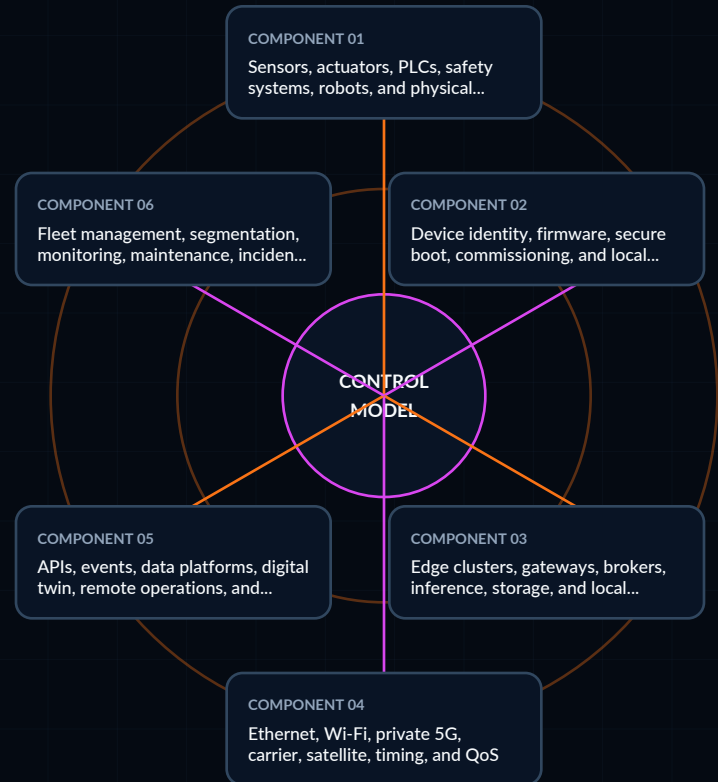
ARCHITECTURE DOCTRINE

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



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.

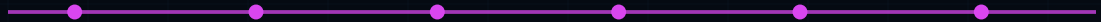
State, data, authority, and trust flow



DESIRED STATE



RUNTIME STATE



OBSERVED STATE



EVIDENCE STATE



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

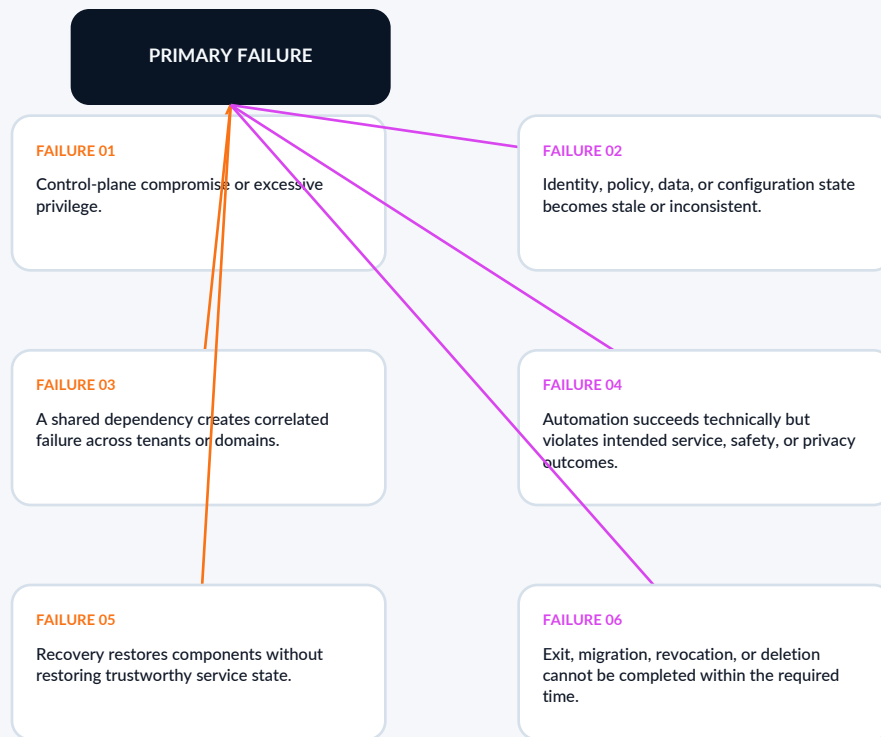
Mutable runtime state versus authoritative evidence

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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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

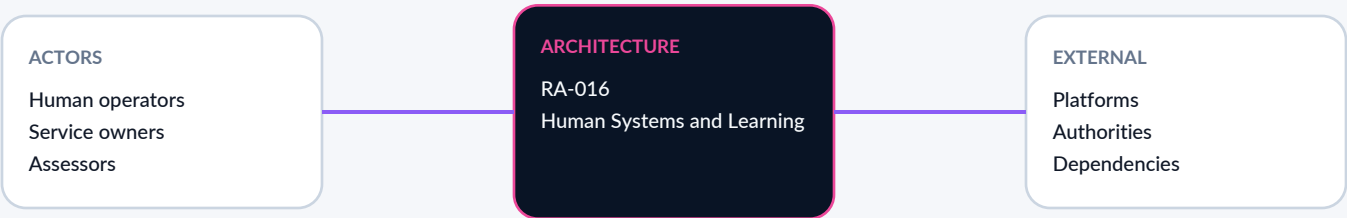


A realtime interaction architecture for speech, audio, video, avatars or presence, transcription, translation, multimodal models, accessibility, consent, provenance, and low-latency orchestration.

Architecture identity and operating position

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

SYSTEM CONTEXT



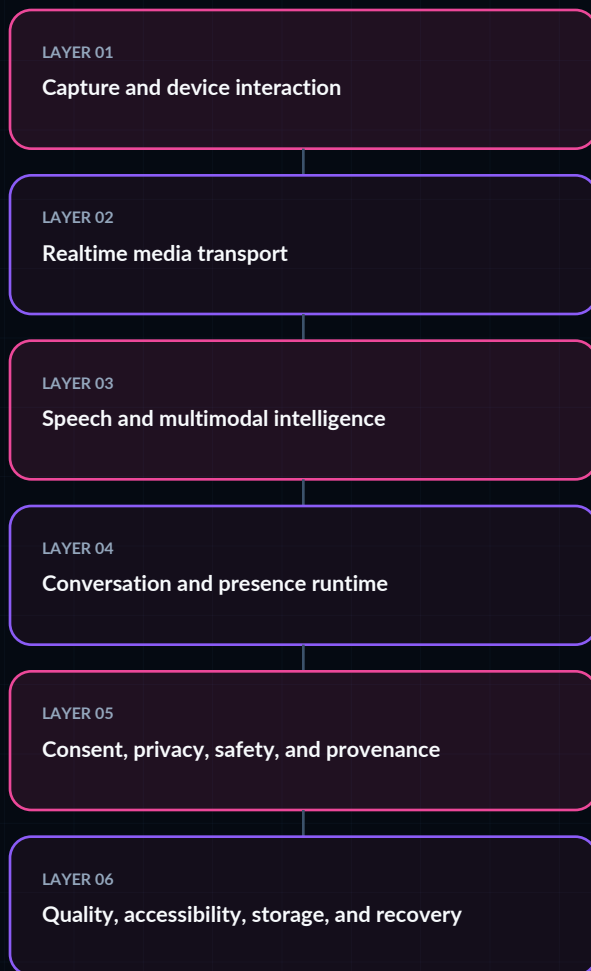
EXECUTIVE OUTCOMES

- 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.

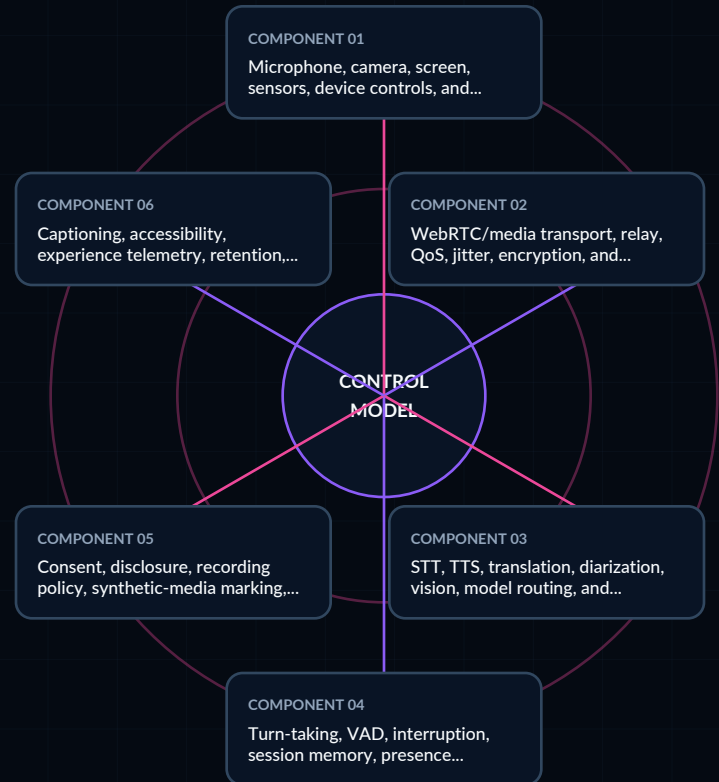
ARCHITECTURE DOCTRINE

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



CORE COMPONENTS



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

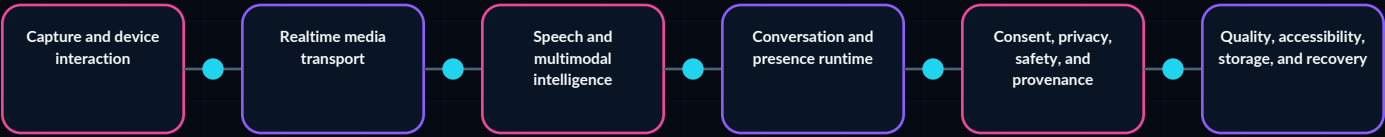
Primary sequence and control flow



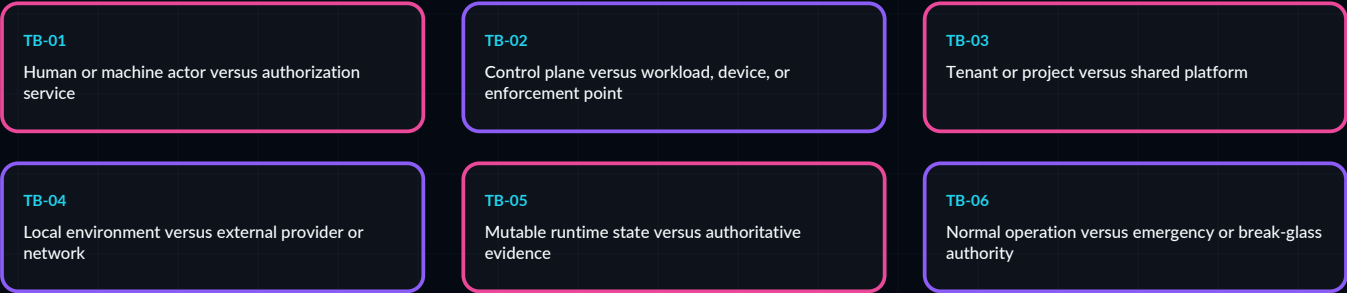
EVIDENCE THREAD

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

State, data, authority, and trust flow

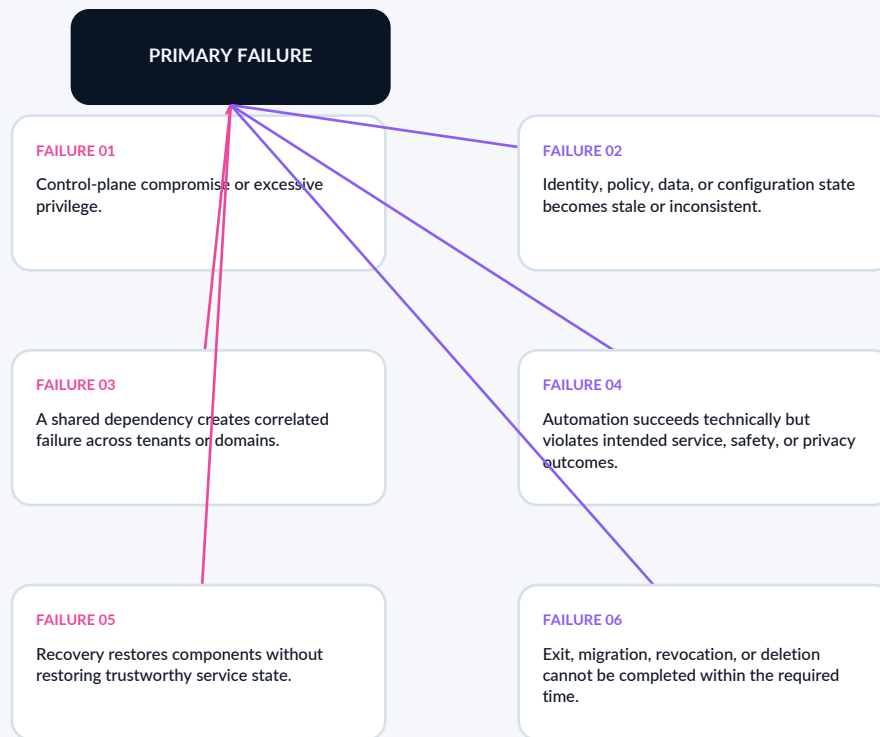


TRUST BOUNDARY REGISTER



Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK

OUTCOME
User / mission result

SERVICE
SLOs / availability

CONTROL
Policy / authorization

EXECUTION
State / errors / retries

DEPENDENCY
External health / latency

EVIDENCE
Trace / artifact / receipt

RECOVERY RULE

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

Deployment profiles and implementation roadmap

DEPLOYMENT PROFILES

PROFILE 01

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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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.

Architecture identity and operating position

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

SYSTEM CONTEXT



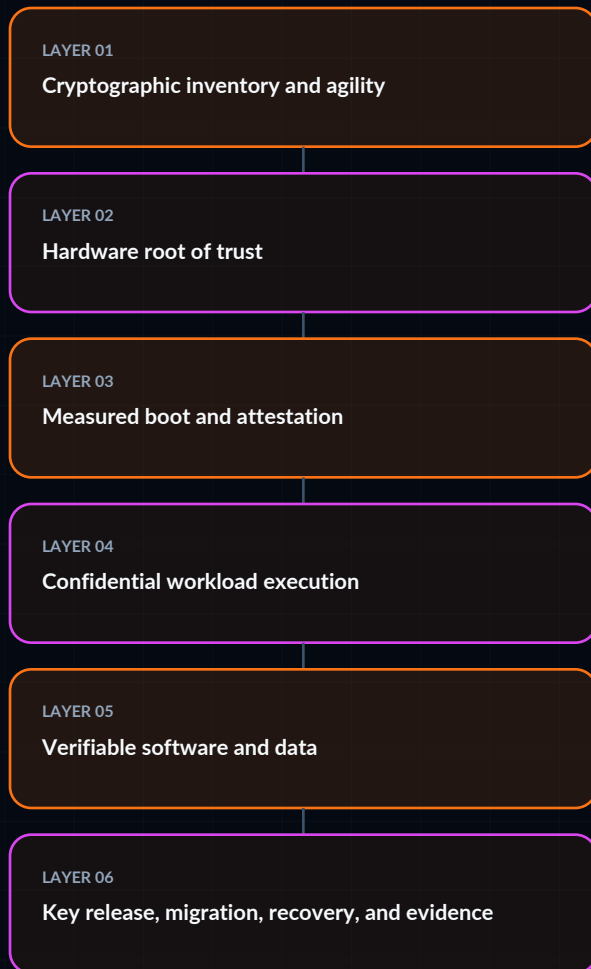
EXECUTIVE OUTCOMES

- 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.

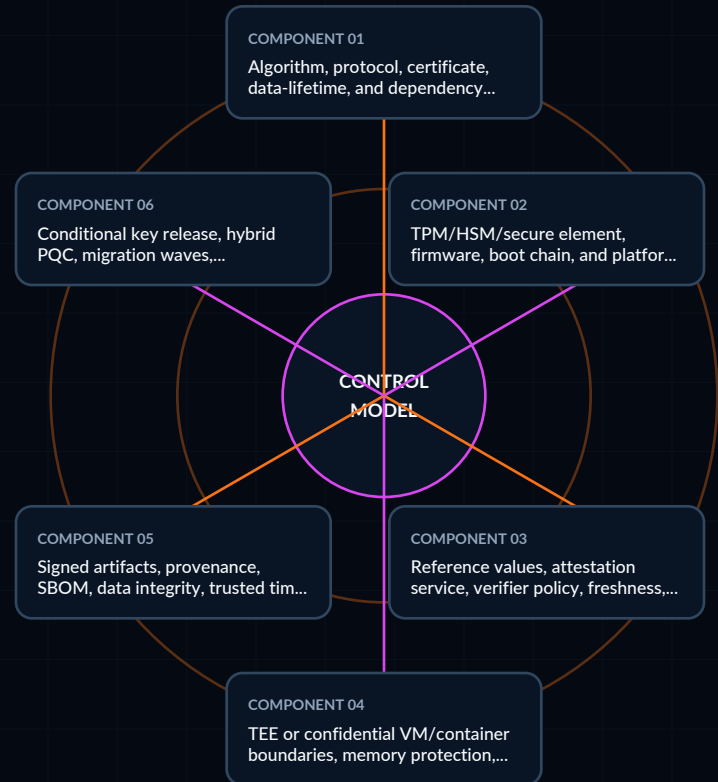
ARCHITECTURE DOCTRINE

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

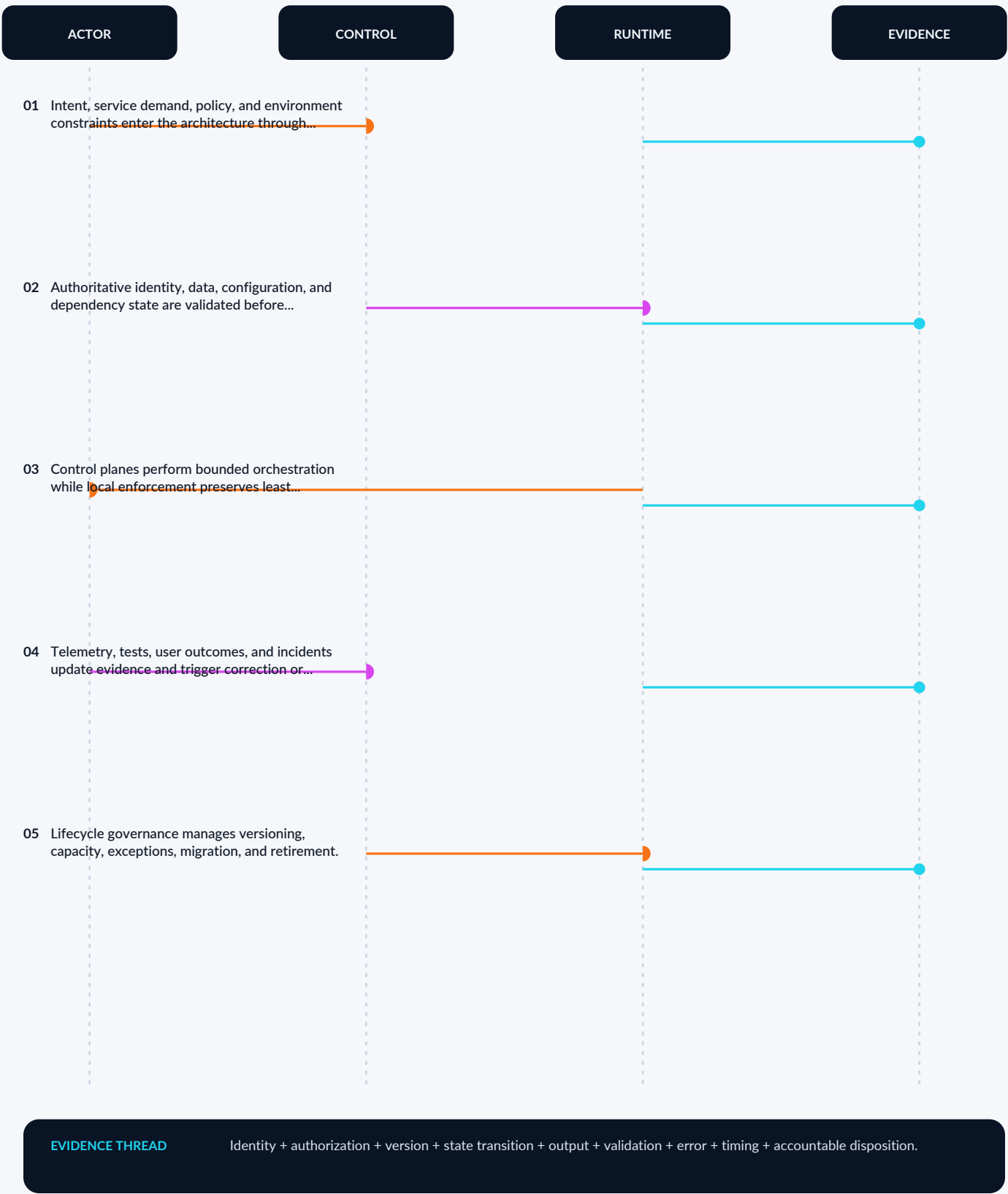


CORE COMPONENTS



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

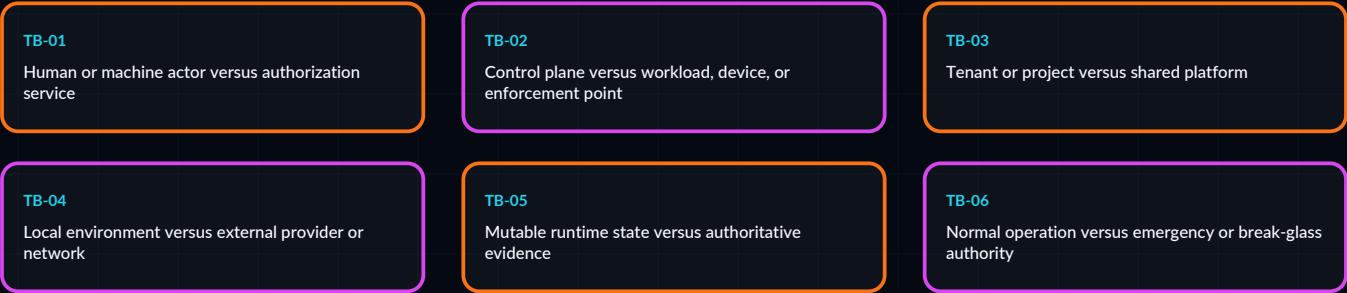
Primary sequence and control flow



State, data, authority, and trust flow

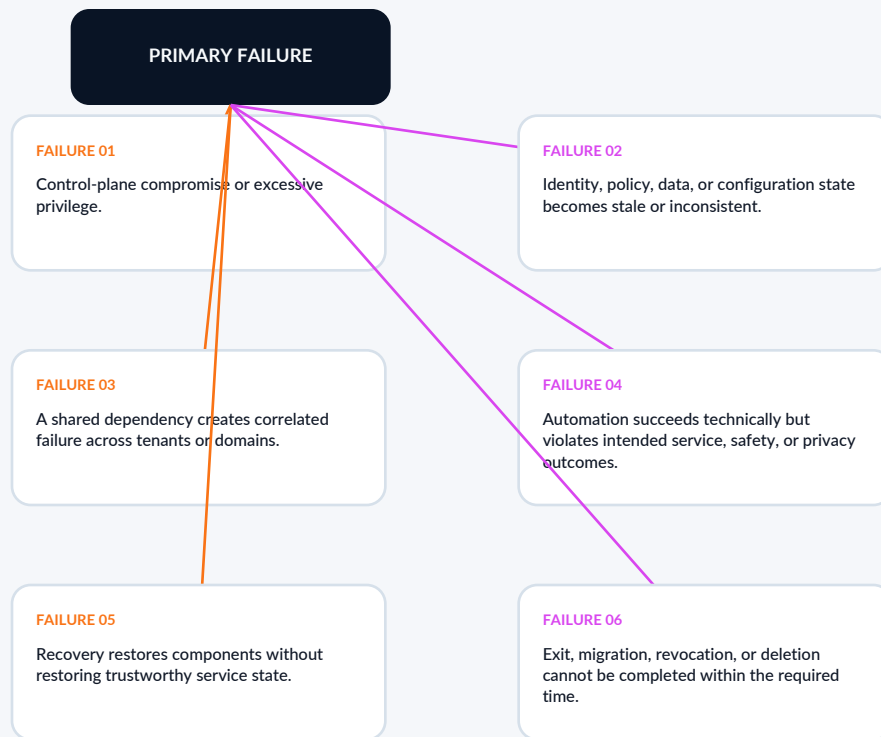


TRUST BOUNDARY REGISTER



Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK

OUTCOME
User / mission result

SERVICE
SLOs / availability

CONTROL
Policy / authorization

EXECUTION
State / errors / retries

DEPENDENCY
External health / latency

EVIDENCE
Trace / artifact / receipt

RECOVERY RULE

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

Deployment profiles and implementation roadmap

DEPLOYMENT PROFILES

PROFILE 01

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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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

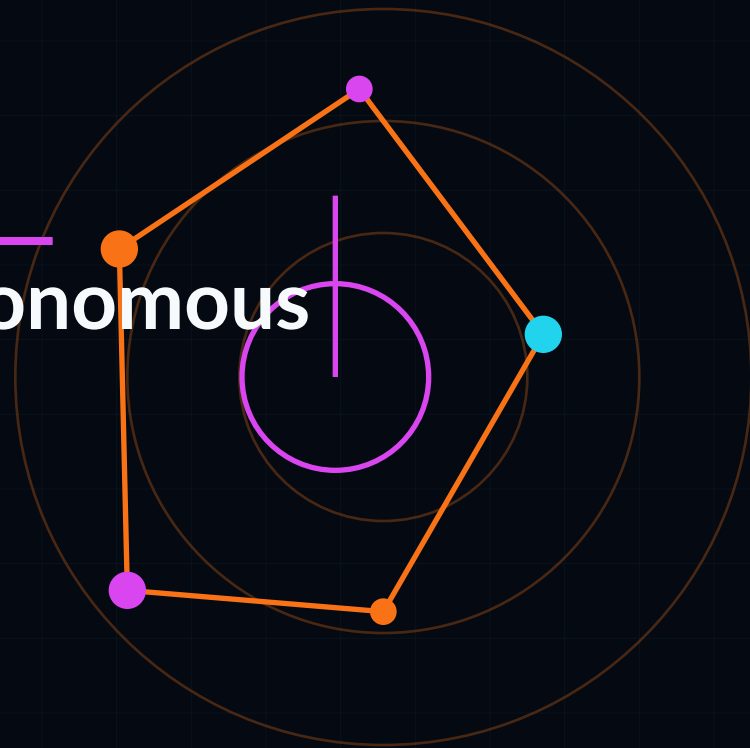
REVISION RECORD

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.

Architecture identity and operating position

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

SYSTEM CONTEXT



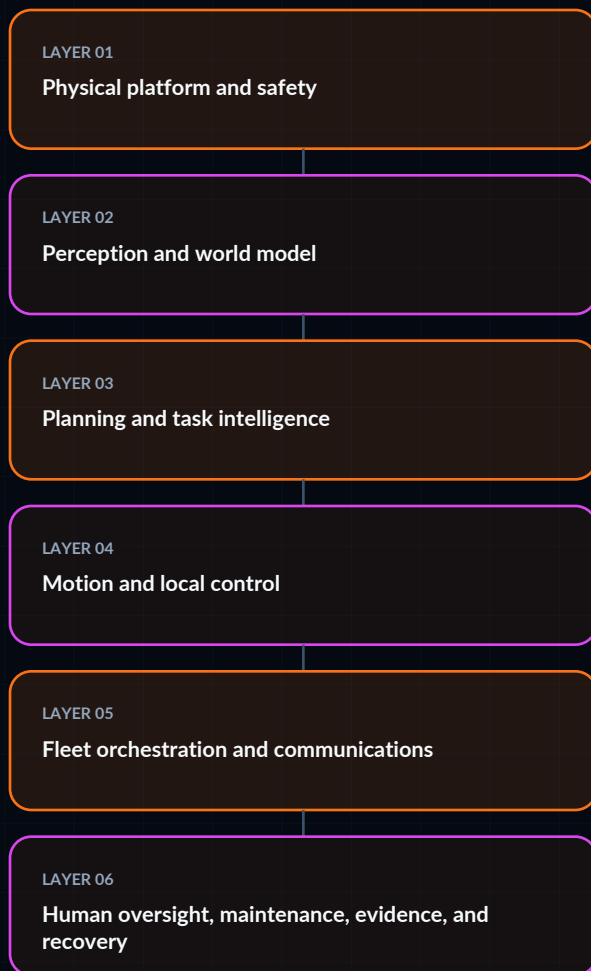
EXECUTIVE OUTCOMES

- 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



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.

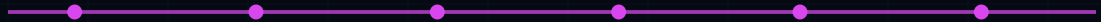
State, data, authority, and trust flow



DESIRED STATE



RUNTIME STATE



OBSERVED STATE



EVIDENCE STATE



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

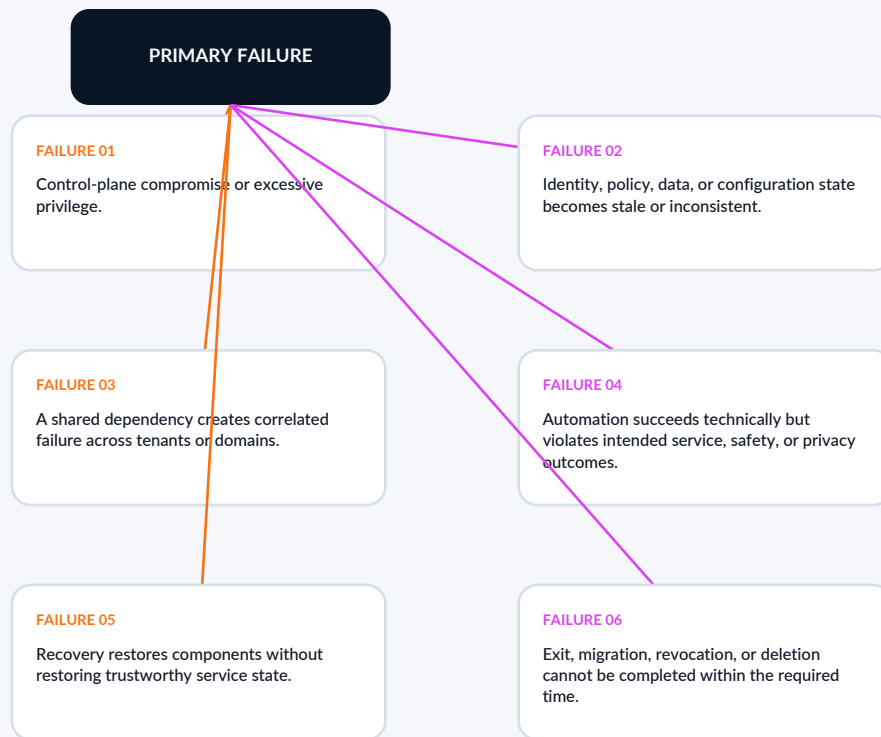
Mutable runtime state versus authoritative evidence

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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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

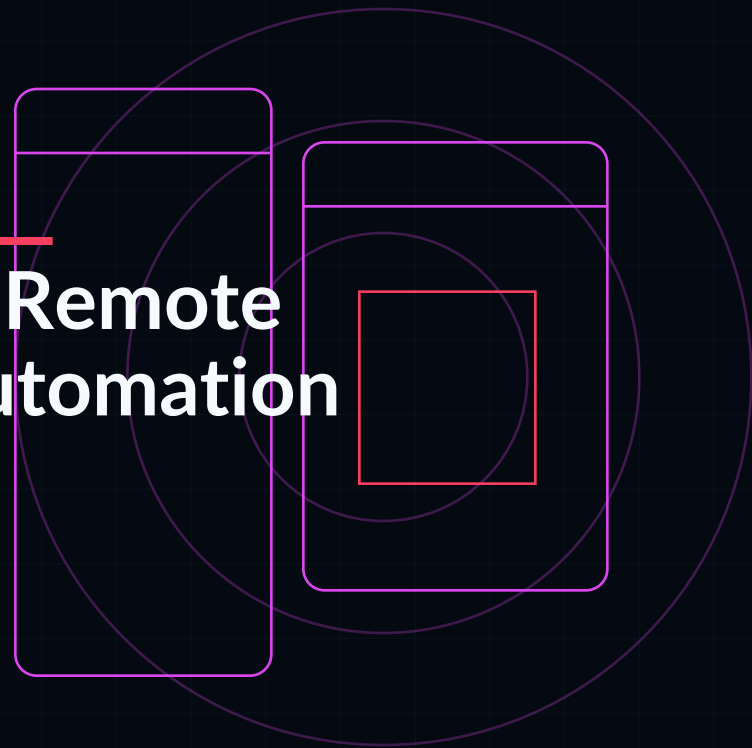
PERMANENT DOCUMENT IDENTITY	RA-018
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Future Domain Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

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.

Architecture identity and operating position

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

SYSTEM CONTEXT



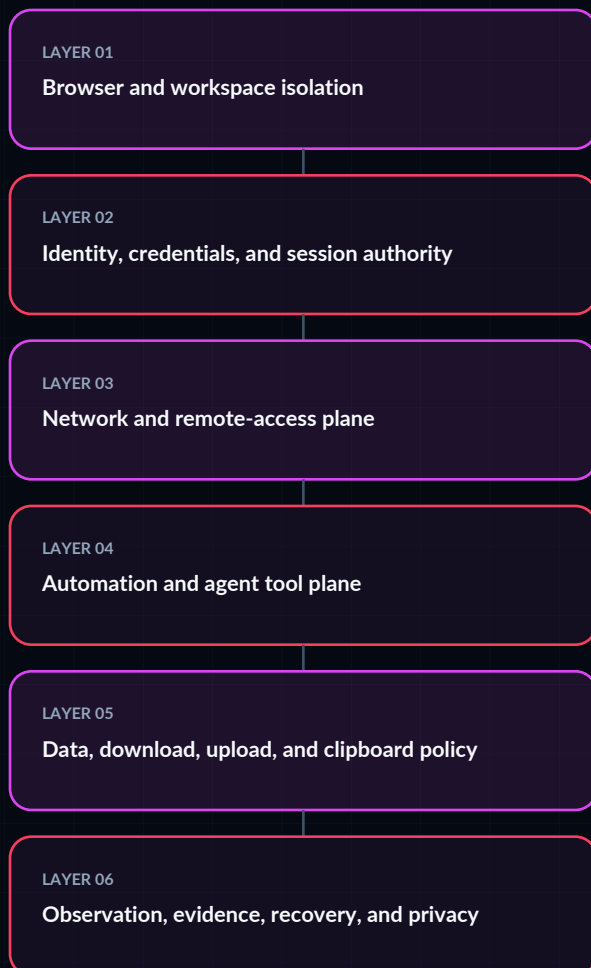
EXECUTIVE OUTCOMES

- 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.

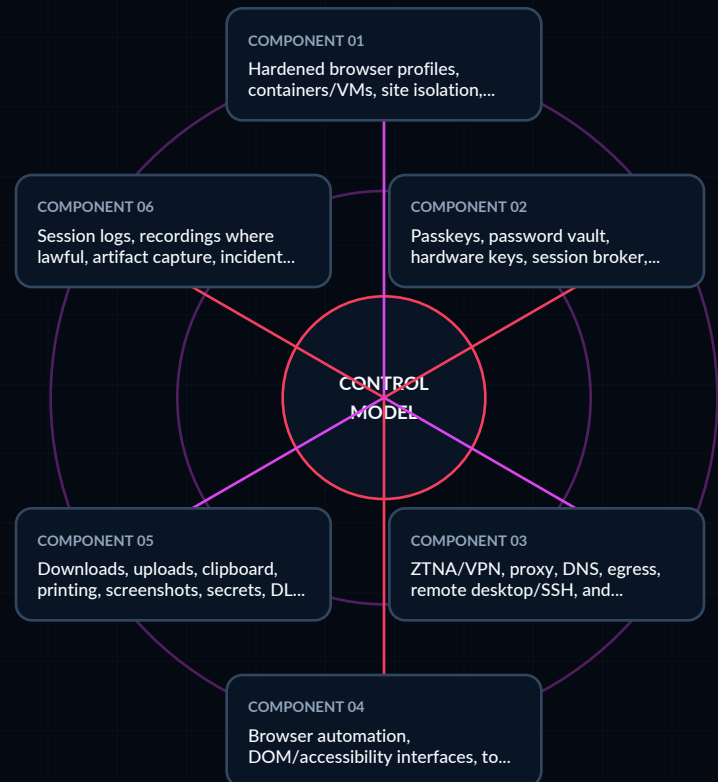
ARCHITECTURE DOCTRINE

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

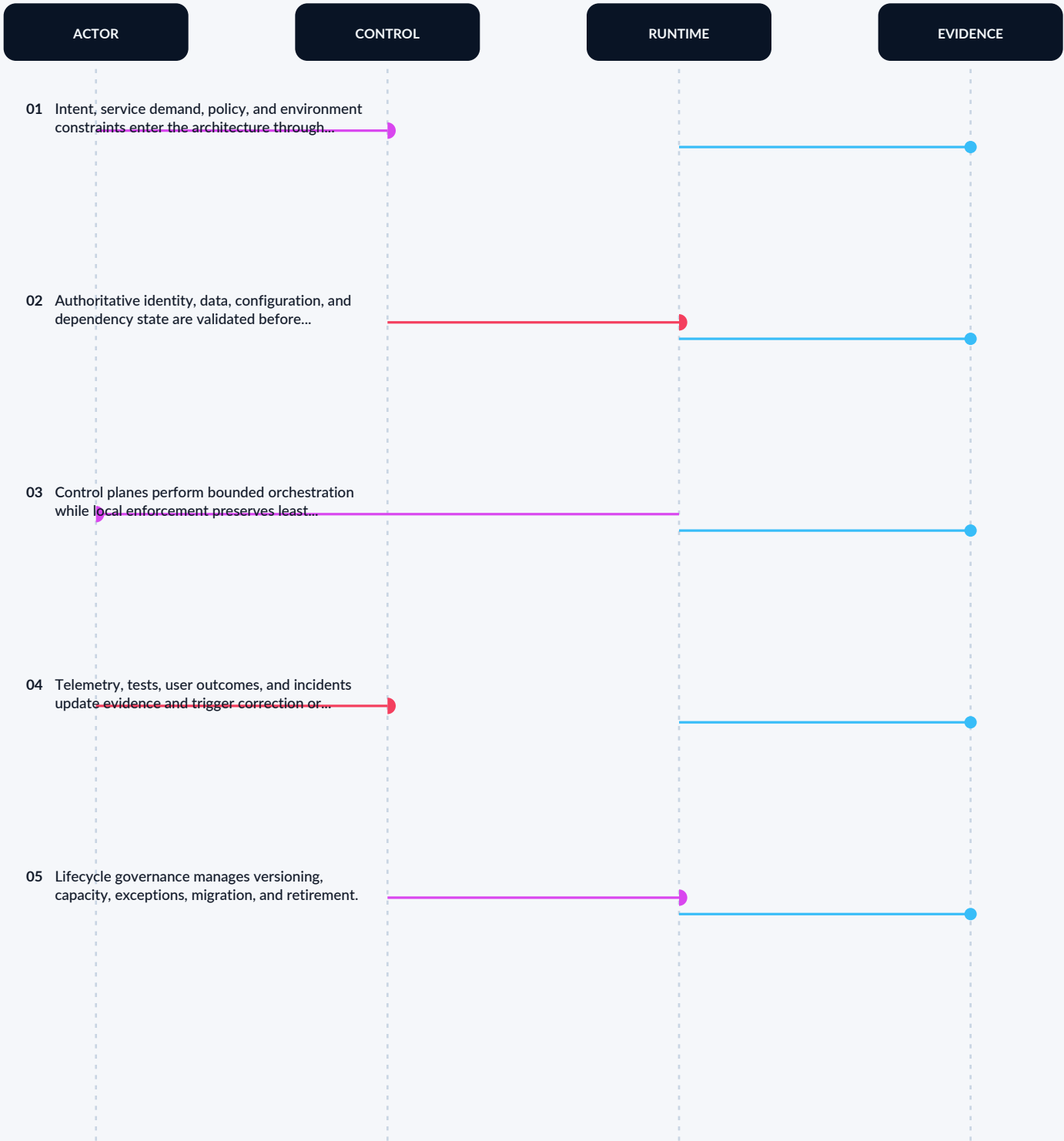


CORE COMPONENTS



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

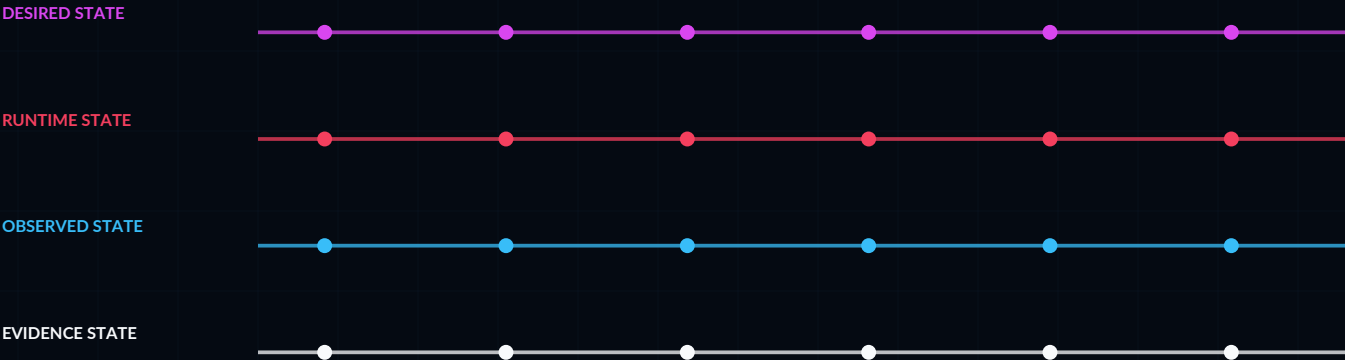
Primary sequence and control flow



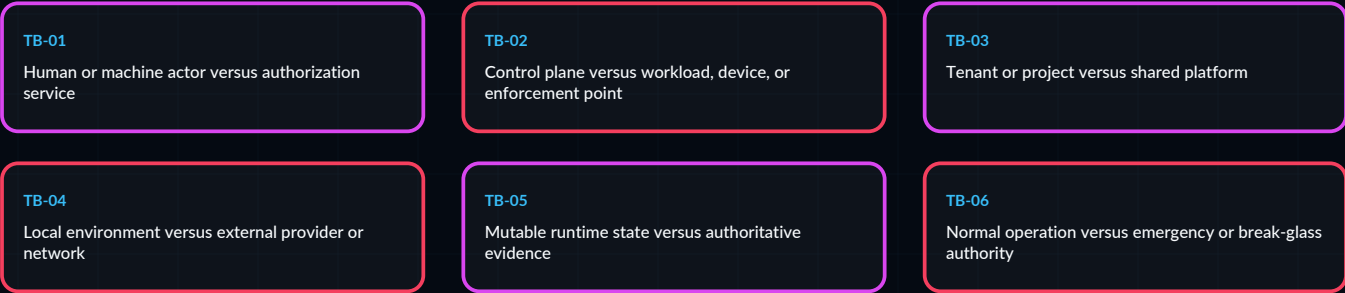
EVIDENCE THREAD

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

State, data, authority, and trust flow

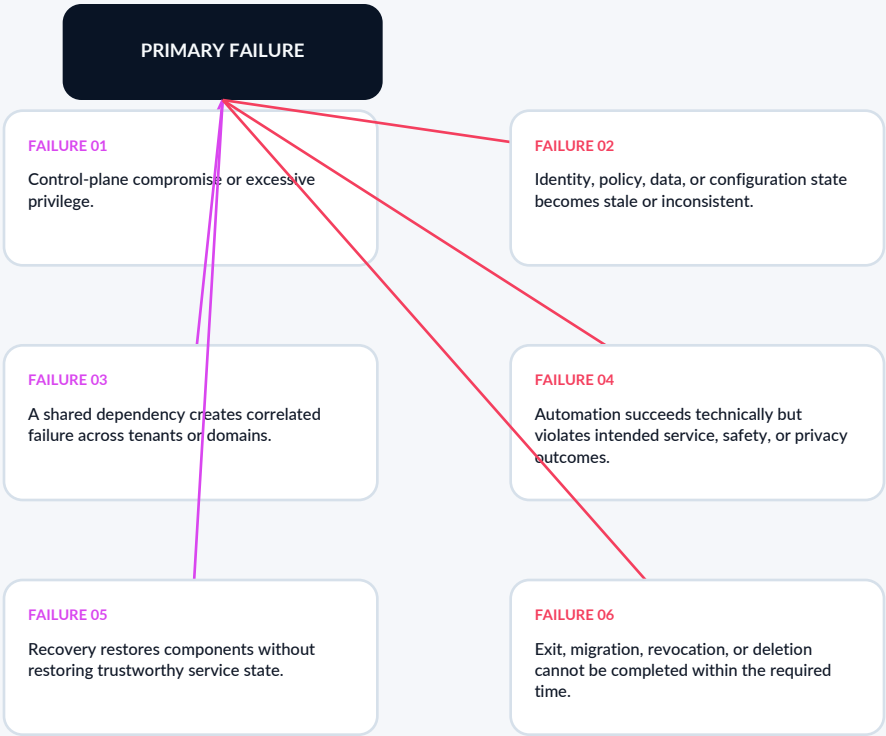


TRUST BOUNDARY REGISTER

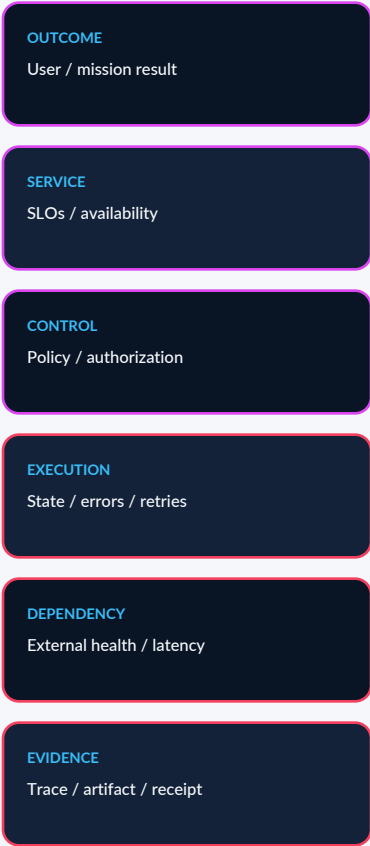


Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK



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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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

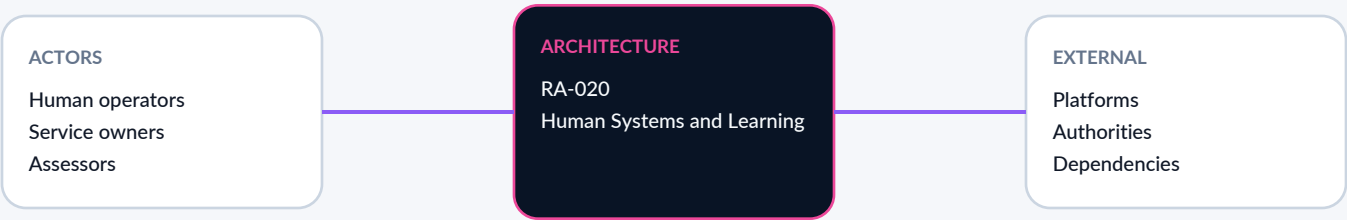


An adaptive learning and cyber-range architecture connecting skills, role profiles, personalized instruction, laboratories, scenarios, assessment, evidence, safety, and workforce readiness.

Architecture identity and operating position

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

SYSTEM CONTEXT



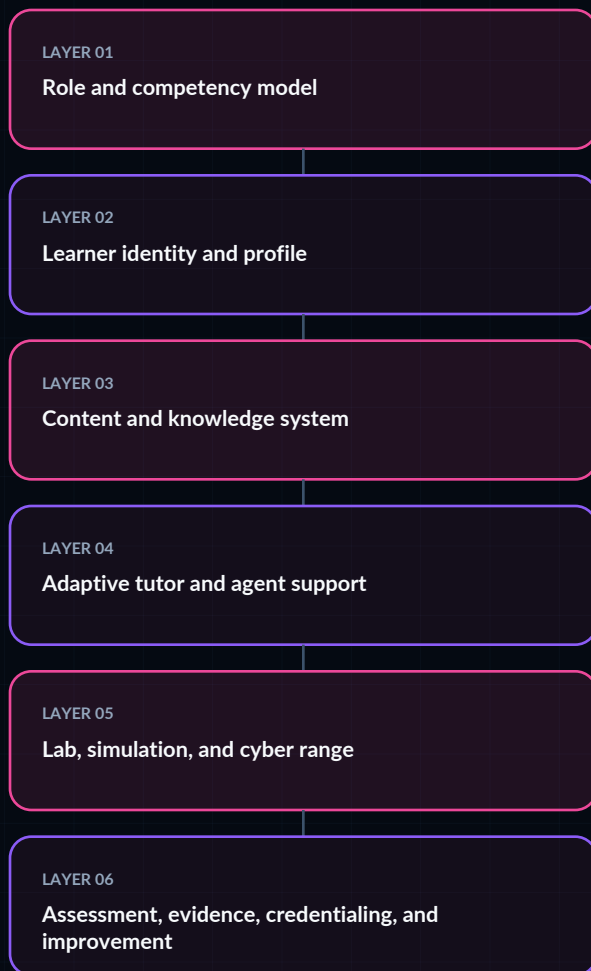
EXECUTIVE OUTCOMES

- 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.

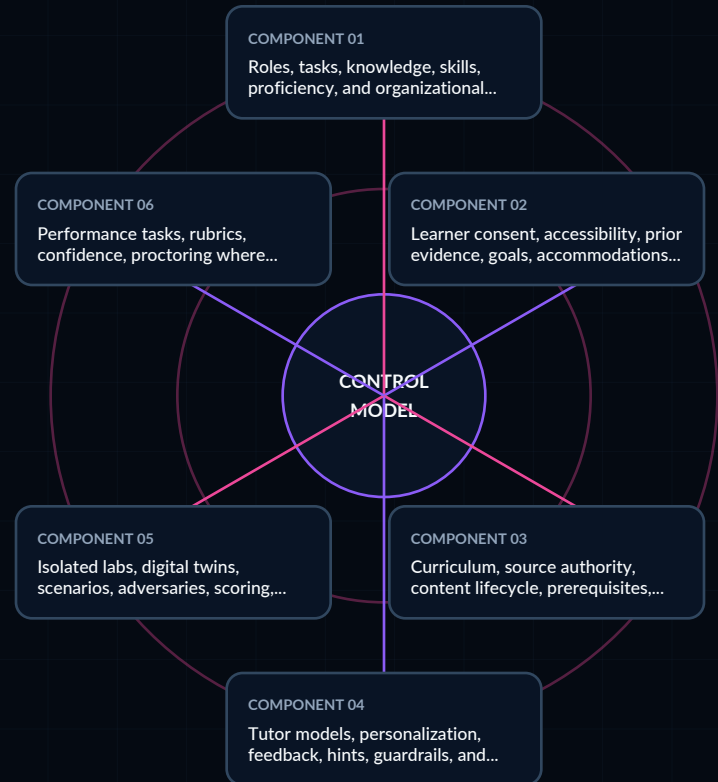
ARCHITECTURE DOCTRINE

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



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.

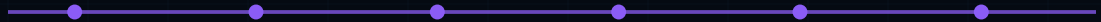
State, data, authority, and trust flow



DESIRED STATE



RUNTIME STATE



OBSERVED STATE



EVIDENCE STATE



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

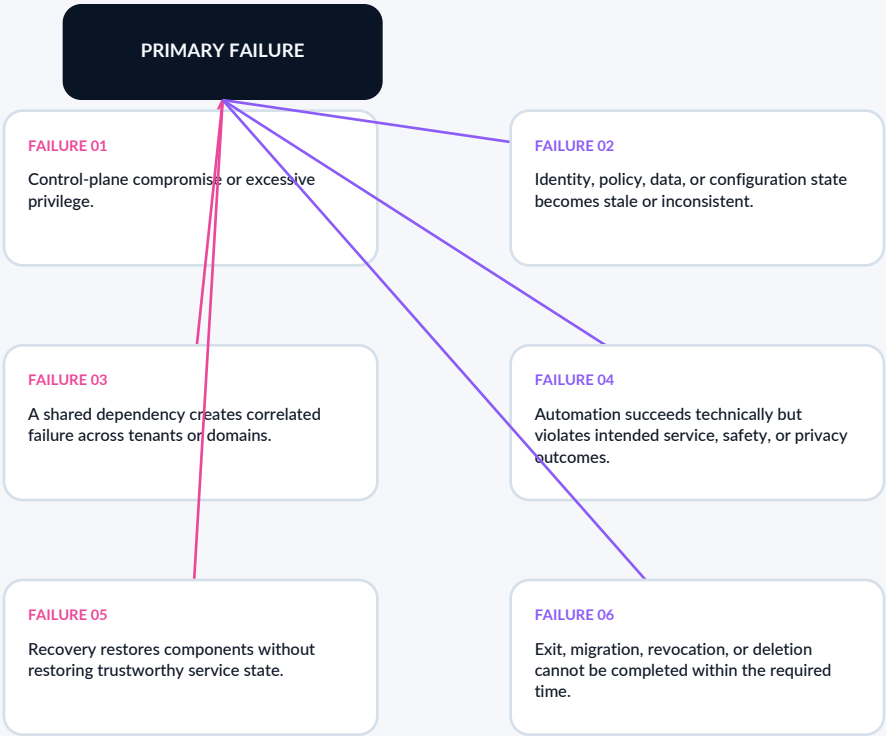
Mutable runtime state versus authoritative evidence

TB-06

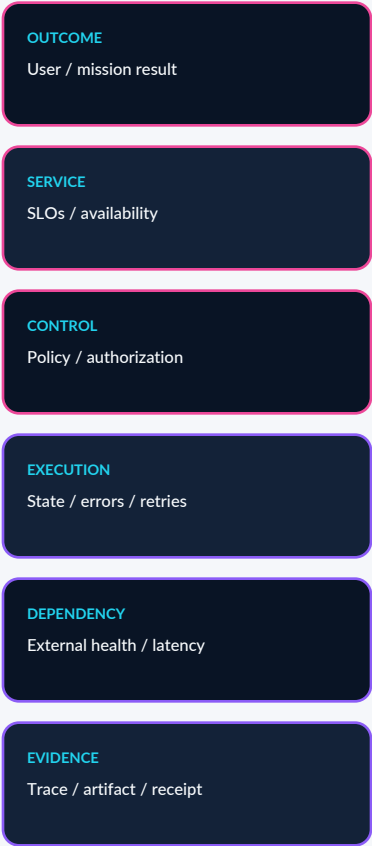
Normal operation versus emergency or break-glass authority

Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK



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

Scope / baseline

31-90 DAYS

Pilot / contracts

3-6 MONTHS

Scale / integrate

6-12 MONTHS

Assure / automate

ONGOING

Measure / revalidate

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

MYTHOS-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.