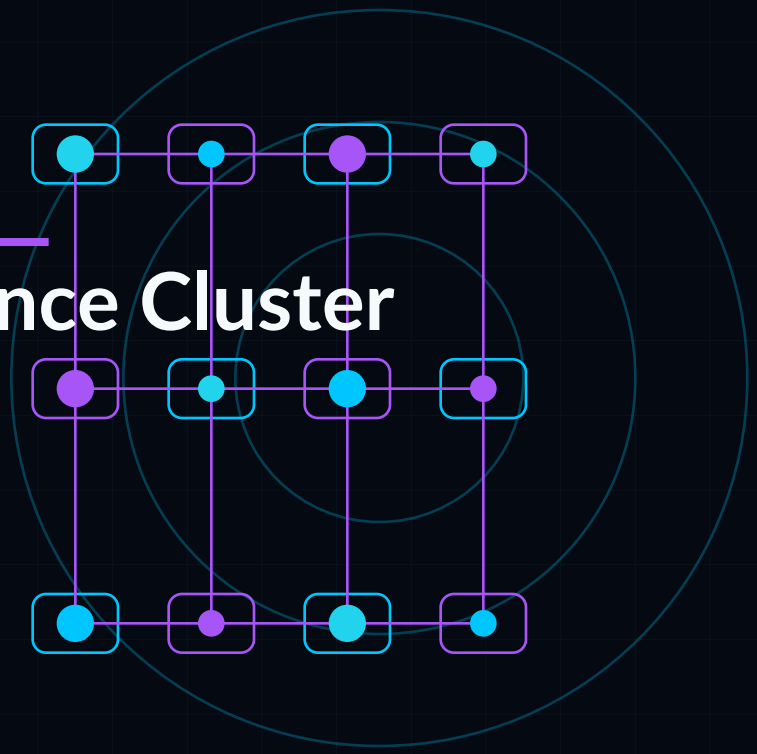
REFERENCE ARCHITECTURE / ENGINEERING STANDARDS LIBRARY

Local-First AI Inference Cluster and Edge Node



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

PERMANENT DOCUMENT ID
RA-006

PUBLICATION
Candidate Reference Architecture

ARCHITECTURE SET
Canonical Set

Architecture identity and operating position

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

SYSTEM CONTEXT

EXECUTIVE OUTCOMES

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

ARCHITECTURE DOCTRINE

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

Layered architecture and component model

LAYER 01
Model Registry and Supply Chain

LAYER 02
Scheduling and Model Routing

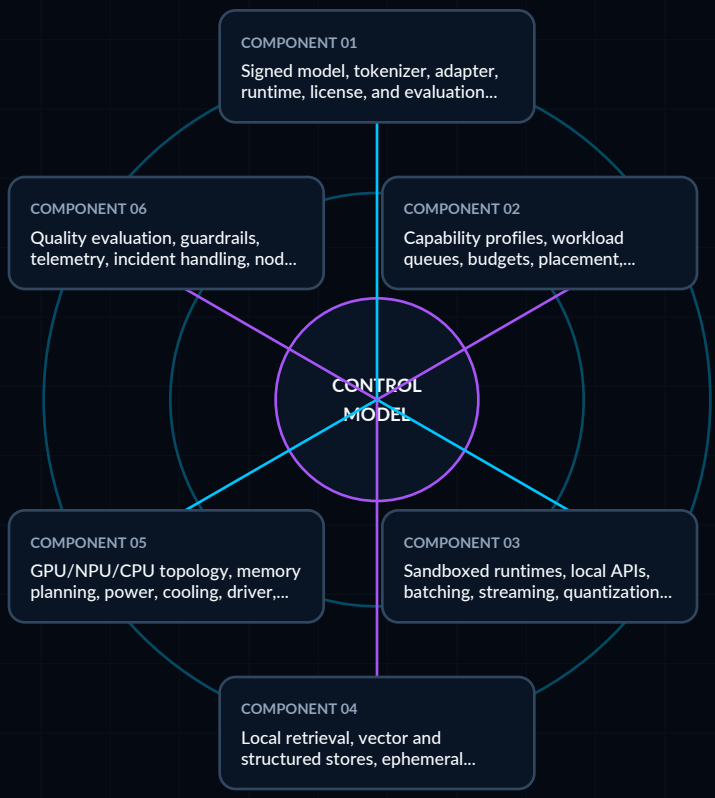
LAYER 03
Inference Runtime and Isolation

LAYER 04
Data, Retrieval, and Context

LAYER 05
Acceleration, Capacity, and Thermal Control

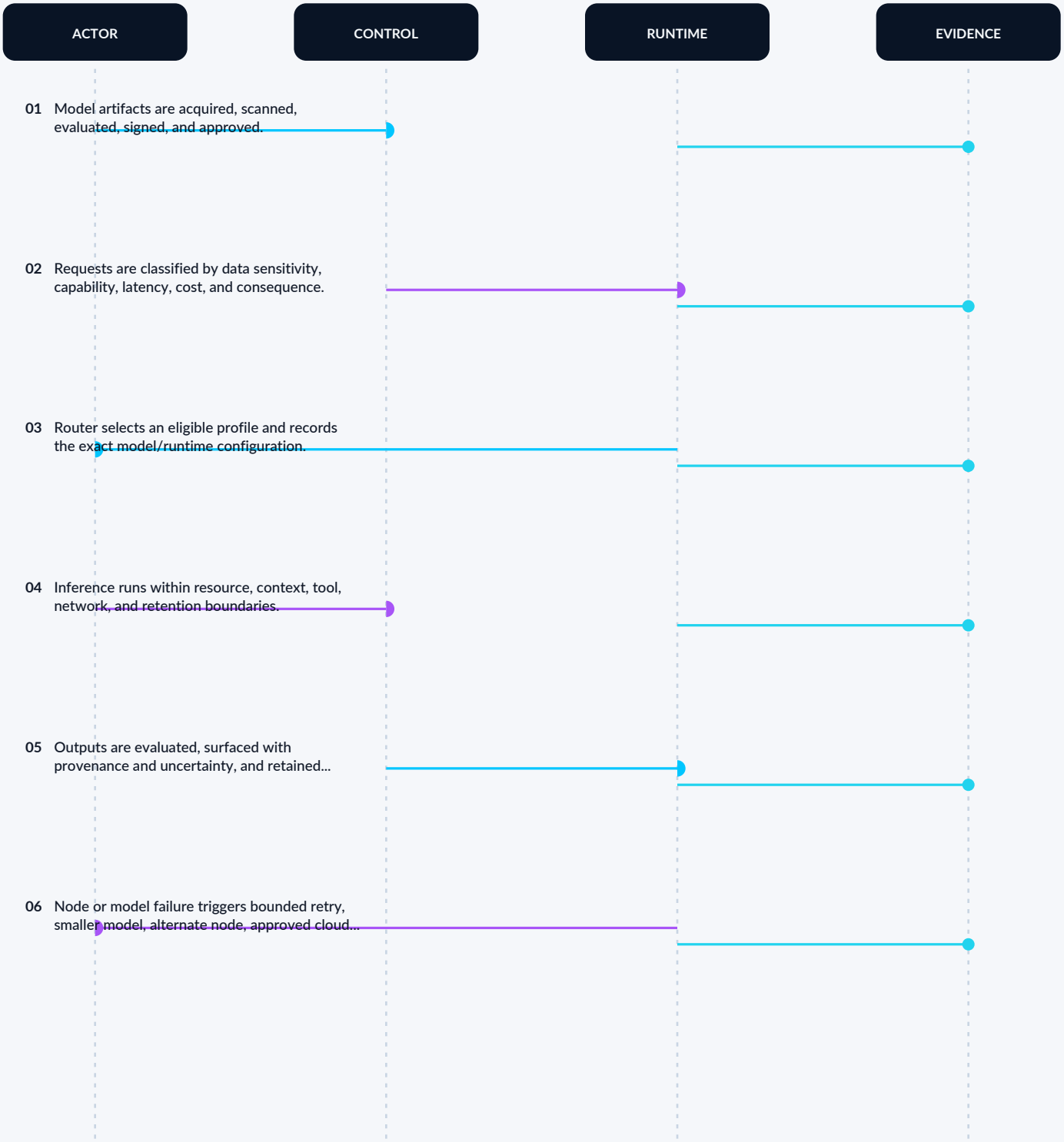
LAYER 06
Observation, Safety, and Recovery

CORE COMPONENTS



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

Primary sequence and control flow



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

State, data, authority, and trust flow

TRUST BOUNDARY REGISTER

TB-01
Untrusted model artifact versus trusted registry

TB-02
User data versus shared model service

TB-03
Inference runtime versus host operating system

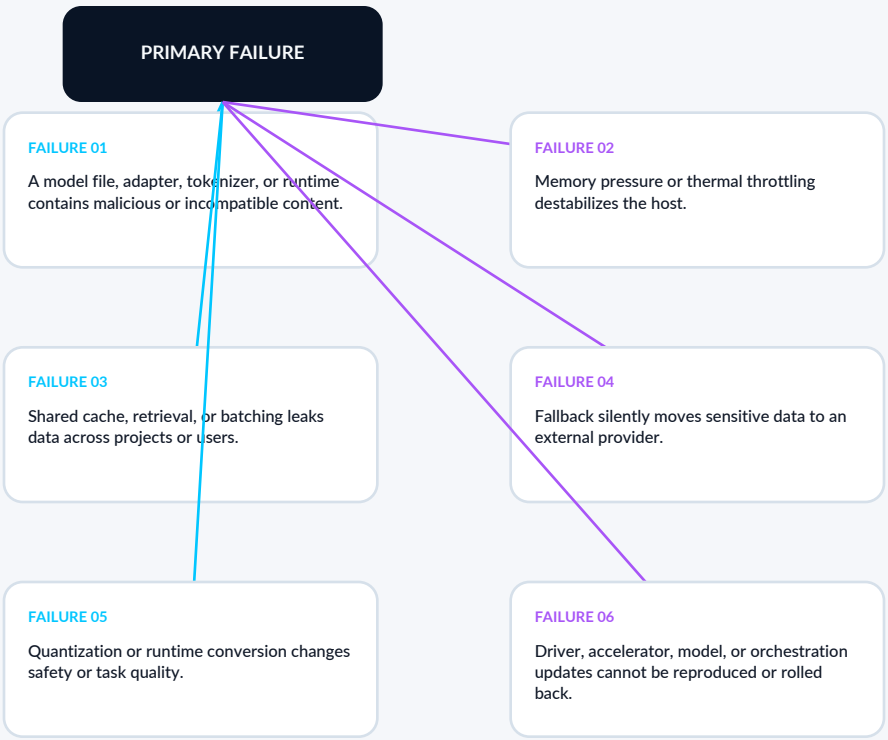
TB-04
Model process versus tools and network

TB-05
Local cluster versus approved external provider

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

Failure propagation and observability

FAILURE PATHS



OBSERVABILITY STACK

OUTCOME
User / mission result

SERVICE
SLOs / availability

CONTROL
Policy / authorization

EXECUTION
State / errors / retries

DEPENDENCY
External health / latency

EVIDENCE
Trace / artifact / receipt

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

Deployment profiles and implementation roadmap

DEPLOYMENT PROFILES

PROFILE 01 — SINGLE ACCELERATED WORKSTATION

Bounded proof

PROFILE 02 — MULTI-GPU LOCAL INFERENCE SERVER

Team-scale governance

PROFILE 03 — SMALL EDGE CLUSTER

Sovereign / high-assurance

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

Distributed enterprise

IMPLEMENTATION ROADMAP

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

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

Verification plan, dependencies, and reading map

ACCEPTANCE CRITERIA - REPRESENTATIVE SET

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

RELATED REFERENCE ARCHITECTURES

No mandatory RA dependencies declared

MAPPED MYTHOS STANDARDS

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

DETAILED SPECIFICATION READING MAP

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

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

PART I - EXECUTIVE DIRECTION

OUTCOMES, RISKS, AND DECISIONS

ARCHITECTURE MANDATE

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

EXECUTIVE OUTCOMES

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

DECISIONS REQUIRED

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

PART II - SCOPE, PRINCIPLES, AND SYSTEM CONTEXT

ARTIFICIAL INTELLIGENCE

IN SCOPE

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

OUT OF SCOPE

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

ARCHITECTURE INVARIANTS

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

ASSUMPTIONS

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

SYSTEM CONTEXT AND RESPONSIBILITY MODEL

ACTORS, ENVIRONMENTS, AND EXTERNAL DEPENDENCIES

ACTORS AND RESPONSIBILITY

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

RESPONSIBILITY BOUNDARY

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

ARCHITECTURE POSITION

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

PART III - LOGICAL ARCHITECTURE

LAYERS AND RESPONSIBILITIES

01

Model Registry and Supply Chain

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

02

Scheduling and Model Routing

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

03

Inference Runtime and Isolation

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

04

Data, Retrieval, and Context

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

05

Acceleration, Capacity, and Thermal Control

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

06

Observation, Safety, and Recovery

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

CORE COMPONENTS - A

COMPONENTS 01-03

COMPONENT 01

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

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

COMPONENT 02

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

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

COMPONENT 03

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

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

CORE COMPONENTS - B

COMPONENTS 04-06

COMPONENT 04

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

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

COMPONENT 05

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

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

COMPONENT 06

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

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

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

FLOW CONTRACTS

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

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

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

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

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

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

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

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

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

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

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

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

TRUST BOUNDARIES AND ENFORCEMENT

EVERY CROSSING REQUIRES AN EXPLICIT CONTRACT

TRUST BOUNDARY 01

Untrusted model artifact versus trusted registry

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

TRUST BOUNDARY 02

User data versus shared model service

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

TRUST BOUNDARY 03

Inference runtime versus host operating system

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

TRUST BOUNDARY 04

Model process versus tools and network

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

TRUST BOUNDARY 05

Local cluster versus approved external provider

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

TRUST BOUNDARY 06

Tenant or project context versus shared cache and memory

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

PART V - DEPLOYMENT PROFILES

PROFILE CHANGES SCALE, NOT ASSURANCE OUTCOMES

PROFILE 01 / SINGLE ACCELERATED WORKSTATION

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

PROFILE 02 / MULTI-GPU LOCAL INFERENCE SERVER

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

PROFILE 03 / SMALL EDGE CLUSTER

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

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

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

PART VI - OPERATING LIFECYCLE

FROM DISCOVERY THROUGH RETIREMENT

01 / Inventory workloads, data, and hardware

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

02 / Establish signed model registry

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

03 / Deploy isolated runtime and API

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

04 / Add routing and resource governance

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

05 / Integrate retrieval and tools cautiously

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

06 / Exercise failure and offline modes

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

07 / Optimize capacity with measured evidence

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

SECURITY, PRIVACY, SAFETY, AND RIGHTS

CROSS-CUTTING ASSURANCE

SECURITY

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

PRIVACY

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

SAFETY

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

RIGHTS AND USABILITY

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

SUPPLY CHAIN

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

EVIDENCE

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

FAILURE MODES AND RESILIENCE

DESIGN FOR SAFE DEGRADATION AND TRUSTWORTHY RECOVERY

FAILURE SCENARIO 01

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

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

FAILURE SCENARIO 02

Memory pressure or thermal throttling destabilizes the host.

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

FAILURE SCENARIO 03

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

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

FAILURE SCENARIO 04

Fallback silently moves sensitive data to an external provider.

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

FAILURE SCENARIO 05

Quantization or runtime conversion changes safety or task quality.

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

FAILURE SCENARIO 06

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

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

PART VII - INTEGRATION AND DEPENDENCY MODEL

CONNECTED REFERENCE ARCHITECTURES AND STANDARDS

REFERENCE ARCHITECTURE DEPENDENCIES

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

MAPPED MYTHOS STANDARDS

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

DEPENDENCY DOCTRINE

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

PART VIII - IMPLEMENTATION AND ADOPTION ROADMAP

PHASED, EVIDENCE-DRIVEN TRANSITION

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

ARCHITECTURE ACCEPTANCE CRITERIA - A

CRITERIA 01-09

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

ARCHITECTURE ACCEPTANCE CRITERIA - B

CRITERIA 10-18

AC-10	Supplier, model, protocol, platform, and software dependencies have provenance, version, support, and replacement records.
AC-11	Representative performance and capacity tests include saturation, contention, latency, and degraded conditions.
AC-12	Exceptions are time-bounded, visible, approved by accountable risk owners, and linked to compensating controls.
AC-13	Every served profile identifies model, tokenizer, adapter, runtime, quantization, hardware, driver, safety configuration, license, and evaluation state.
AC-14	Routing policy proves that sensitive requests cannot cross an unapproved provider, tenant, cache, or network boundary.
AC-15	Admission control prevents memory, power, thermal, storage, and queue saturation from destabilizing critical workloads.
AC-16	The platform exposes model cold-start, load, queue, token, latency, quality, failure, resource, and fallback telemetry.
AC-17	Offline operation, node loss, accelerator loss, and approved cloud overflow are exercised with representative tasks.
AC-18	Model retirement removes serving authority, cache, indexes, credentials, and retained data without deleting required evidence.

METRICS, EVIDENCE, AND DECISION GATES

MEASURE SERVICE OUTCOMES AND ASSURANCE

COVERAGE

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

CHANGE QUALITY

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

RELIABILITY

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

SECURITY

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

HUMAN OUTCOME

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

LIFECYCLE

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

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

SOURCE AUTHORITY REGISTER

CURRENT PRIMARY AND OFFICIAL REFERENCES

SOURCE 01 **NIST AI Risk Management Framework 1.0 (revision in progress as of publication date)**

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

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

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

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

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

SOURCE 04 **SLSA Specification 1.2**

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

SOURCE 05 **Kubernetes Documentation**

https://kubernetes.io/docs/

SOURCE 06 **OpenTelemetry Specification**

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

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

PUBLICATION CONTROL AND REVISION

PERMANENT IDENTITY / CONTROLLED EVOLUTION

PERMANENT DOCUMENT IDENTITY	RA-006
VERSION	1.0.0-candidate
PUBLICATION DATE	2026-07-24
SCHEDULED REVIEW	2027-01-24
STATUS	Candidate Reference Architecture
ARCHITECTURE SET	Canonical Set
CLASSIFICATION	Public
PUBLISHER	Mythos Systems

REVISION RECORD

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