MYTHOS-GUIDE-SWEPLT-0002

Software, Platform, Data, And Reliability Library Guide

Integrated engineering guide connecting the relevant Mythos standards, implementation patterns, operating procedures, architecture decisions, evidence expectations, and maturity path for software, platform, data, and reliability library.

43	10	2026-09-17
CORE STANDARDS	RELATED ARCHITECTURES	BASELINE

CANDIDATE ENGINEERING GUIDE

Mythos Systems | Permanent ID | 17 September 2026

PERMANENT PUBLICATION IDENTITY

Software, Platform, Data, And Reliability Library Guide

MYTHOS-GUIDE-SWEPLT-0002 / Engineering Guide

Field	Value
Document ID	MYTHOS-GUIDE-SWEPLT-0002
Publication class	Engineering Guide
Status	Candidate Engineering Guide
Release date	2026-09-17
Canonical route	/library/guides/mythos-guide-sweplt-0002/
Refresh cadence	180 days
Public identity	Permanent ID; production workflow metadata excluded

Publication position

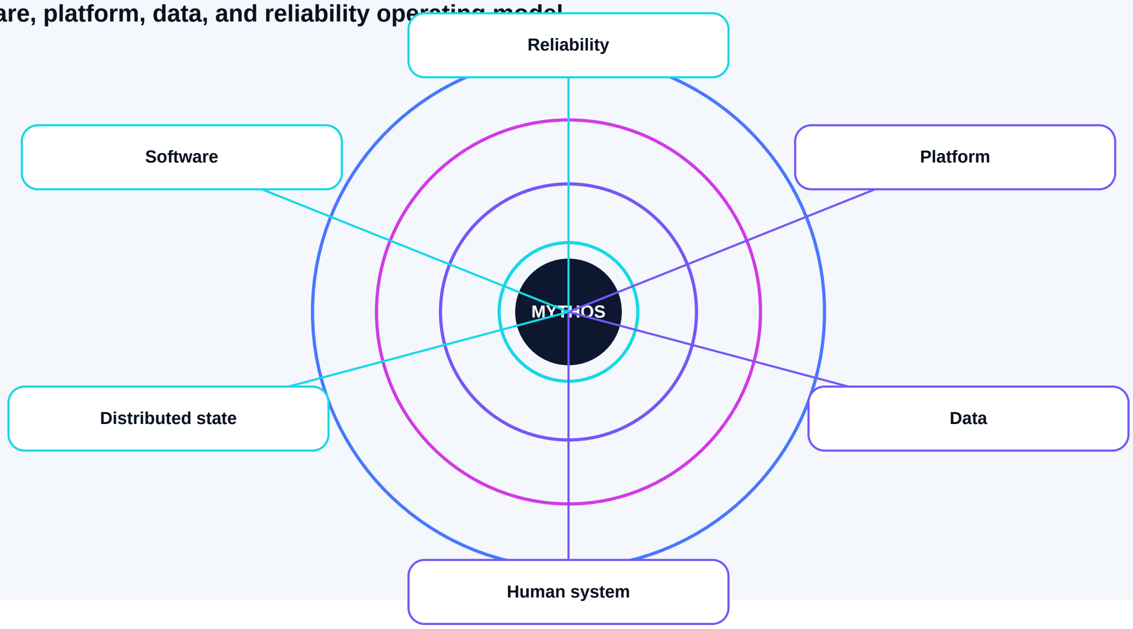
This candidate publication is complete for structured technical review. It does not by itself establish certification, legal compliance, implementation conformance, benchmark reproduction, or final approval.

Contents

1. Guide purpose

This guide is the primary navigation and operating companion for its selected library domains. It does not replace the component standards. It explains how to select, sequence, combine, implement, verify, and operate them as one engineering program against the locked permanent-library baseline.

Software, platform, data, and reliability operating model



2. Domain operating model

Product and architecture foundation

Define bounded context, domain model, security/reliability objectives, ownership, interfaces, and lifecycle before choosing implementation detail.

Typed software boundaries

Make contracts explicit across process, service, language, plugin, data, and human-system boundaries; fail closed on ambiguous state transitions.

Platform and supply chain

Treat CI/CD, artifacts, policy, infrastructure, identity, provenance, and developer self-service as one controlled delivery system.

Distributed state and data

Design consistency, durability, ordering, tenancy, recovery, backup, schema evolution, provenance, and data authority as first-class architecture.

Reliability engineering

Use SLOs, error budgets, capacity, fault injection, disaster recovery, observability, and operational readiness to prove service behavior under failure.

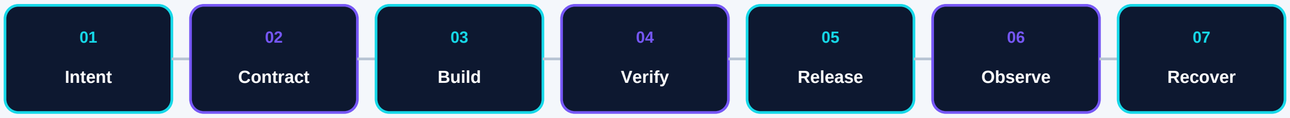
Human-system control

Design dense technical interfaces, HITL approvals, accessibility, and high-risk action affordances so operator authority remains legible and attributable.

Lifecycle economics

Connect lifecycle refresh, FinOps/token/energy economics, operational toil, and decommissioning to architecture decisions rather than treating them as post-launch concerns.

End-to-end engineering sequence



3. Current standard register

43 permanent candidate standards form the primary scope of this guide. Foundation standards apply as inherited library controls even when they are not repeated below.

ID	Title	Domain	Refresh
MYTHOS-CLD-0001	Multi-Cloud Architecture & Control Plane Standard	Cloud Engineering	90d
MYTHOS-CLD-0002	AWS Architecture, Security, & Automation Standard	Cloud Engineering	90d
MYTHOS-CLD-0003	Microsoft Azure Architecture, Security, & Automation Standard	Cloud Engineering	90d
MYTHOS-CLD-0004	Google Cloud Platform (GCP) Architecture & Automation Standard	Cloud Engineering	90d
MYTHOS-CLD-0005	Oracle Cloud (OCI) Architecture & Automation Standard	Cloud Engineering	90d
MYTHOS-CLD-0006	Alibaba Cloud Architecture & Automation Standard	Cloud Engineering	90d
MYTHOS-CLD-0007	Kubernetes, Container Orchestration, & Operator Standard	Cloud Engineering	90d
MYTHOS-CLD-0008	Serverless Architecture & Event-Driven Compute Standard	Cloud Engineering	90d
MYTHOS-CLD-0009	Decentralized Physical Infrastructure Networks (DePIN) Standard	Cloud Engineering	90d
MYTHOS-CLD-0010	Private Cloud, Hypervisor & Bare Metal Standard	Cloud Engineering	90d
MYTHOS-CLD-0011	Web3 & Decentralized Ledger Architecture Standard	Cloud Engineering	90d
MYTHOS-DAT-0001	Vector Database & Local-First Sync Architecture Standard	Data, State, and Evidence	120d
MYTHOS-DAT-0002	AI & Agent Observability Semantic Conventions (OpenTelemetry) Standard	Data, State, and Evidence	120d
MYTHOS-DAT-0003	Data Sovereignty, Privacy Preservation, & Egress Control Standard	Data, State, and Evidence	120d
MYTHOS-DAT-0004	Event Sourcing, CQRS, & Durable State Machine Standard	Data, State, and Evidence	120d
MYTHOS-DAT-0005	Tamper-Evident Hash-Chained Ledgers & Evidence Bundle Standard	Data, State, and Evidence	120d
MYTHOS-DBS-0001	Relational, Graph, and Time-Series Database Standard	Database and Storage	180d
MYTHOS-DBS-0002	Storage, Search, and Backup Architecture Standard	Database and Storage	180d
MYTHOS-DBS-0003	Storage Multi-Tenancy and Data Access Zones Standard	Database and Storage	180d
MYTHOS-DST-0001	Durable Workflows, Queues, and Event-Driven Sagas Standard	Distributed Systems	180d
MYTHOS-DST-0002	API Contracts, gRPC, and Service Mesh Standard	Distributed Systems	180d
MYTHOS-PLT-0001	DevSecOps, Release Engineering, and Supply Chain Standard	Platform Engineering	180d
MYTHOS-PLT-0002	Infrastructure and Configuration as Code Standard	Platform Engineering	180d
MYTHOS-PLT-0003	Internal Developer Platform, GitOps, and Policy-as-Code Standard	Platform Engineering	180d
MYTHOS-PLT-0004	Managed Platform and Agentic Service Evaluation Standard	Platform Engineering	180d
MYTHOS-PLT-0005	Hardware and Software Lifecycle and Refresh Standard	Platform Engineering	180d
MYTHOS-PLT-0006	Digital Marketplace and Extension Architecture Standard	Platform Engineering	180d
MYTHOS-SRE-0001	SLOs, Error Budgets, and Capacity Planning Standard	Reliability Engineering	180d
MYTHOS-SRE-0002	Disaster Recovery, Multi-Region, and Failure Testing Standard	Reliability Engineering	180d
MYTHOS-SRE-0003	FinOps, Token Economics, and Energy Efficiency Standard	Reliability Engineering	180d
MYTHOS-SWE-0001	Advanced Software Design and Domain-Driven Design Standard	Software Engineering	180d
MYTHOS-SWE-0002	Rust Engineering, Memory Safety, and Concurrency Standard	Software Engineering	180d
MYTHOS-SWE-0003	Go Engineering, Concurrency, and Service Architecture Standard	Software Engineering	180d
MYTHOS-SWE-0004	Python Engineering, Type Safety, and Automation Standard	Software Engineering	180d
MYTHOS-SWE-0005	Polyglot Boundaries, Typed Contracts, and IPC Standard	Software Engineering	180d
MYTHOS-SWE-0006	Desktop Architecture and Tauri/Svelte Integration Standard	Software Engineering	180d
MYTHOS-SWE-0007	Formal Verification and Deterministic State Machine Standard	Software Engineering	180d
MYTHOS-SWE-0008	Semantic UI Protocols and Typed Renderer Abstraction Standard	Software Engineering	180d
MYTHOS-SWE-0009	Software Debugging and Resilience Testing Standard	Software Engineering	180d

ID	Title	Domain	Refresh
MYTHOS-UX-0001	AI-Native Interface, High-Density Data, and Accessibility Standard	User Experience and Human-System Interaction	180d
MYTHOS-UX-0002	Human-in-the-Loop Approval Workflow Standard	User Experience and Human-System Interaction	180d
MYTHOS-UX-0003	Cyberpunk-Noir Aesthetic and Visual State Standard	User Experience and Human-System Interaction	180d
MYTHOS-UX-0004	Spatial Computing and WebGPU/wgpu Integration Standard	User Experience and Human-System Interaction	180d

4. Architecture relationships

Reference architectures are implementation patterns, not mandatory products. Use them to make trust boundaries, state/data flows, failure assumptions, deployment profiles, observability, and verification concrete.

ID	Reference architecture	Use in this guide
RA-008	Multi-Cloud Landing Zone and Internal Developer Platform	A governed enterprise foundation for accounts, subscriptions, identity, networking, policy, platform services, golden p...
RA-012	Enterprise Observability, SRE, and Incident Command Platform	A unified architecture for telemetry, service objectives, diagnostics, alerting, incident command, change correlation, ...
RA-014	Enterprise Data, API, Event, and Integration Fabric	A governed integration architecture for APIs, events, data products, schemas, workflows, identity, quality, lineage, re...
RA-001	Governed Multi-Agent Runtime and PANTHEON Architecture	A governed, durable multi-agent runtime that separates reasoning, authorization, deterministic execution, verification,...
RA-006	Local-First AI Inference Cluster and Edge Node	A sovereign inference architecture for local models, multimodal processing, memory, routing, observability, and optiona...
RA-007	AI-Assisted Software Engineering Pipeline	A governed software-engineering pipeline that combines repository intelligence, isolated agent work, model routing, det...
RA-010	Local-First Knowledge, Memory, and Evidence Platform	A source-linked architecture for document intelligence, retrieval, temporal memory, knowledge graphs, durable evidence,...
RA-013	Secure Software Supply Chain and Release Factory	A controlled software factory for source, dependencies, builds, tests, artifacts, signing, provenance, deployment, roll...
RA-015	Edge, OT, IoT, and Telecommunications Convergence	A safety-aware architecture for distributed edge compute, industrial systems, sensors, wireless, telecom, local autonom...
RA-016	Realtime Voice, Media, and Multimodal Interaction Platform	A privacy-aware architecture for speech, avatars, realtime media, multimodal context, consent, accessibility, local pro...

5. Research and adoption intelligence

Research material is time-bounded evidence. Adoption posture should follow reproduced evidence, stated limitations, readiness gates, and review triggers rather than novelty.

ID	Research publication	Refresh
MYTHOS-RES-ANNEX-SWPLT-0001	Software Platform And Reliability Research Annex	90d
MYTHOS-RES-ANNEX-AIKDIL-0001	AI Knowledge Data Identity And Learning Research Annex	90d
MYTHOS-RES-ANNEX-CLOUDEDGE-0001	Cloud Edge Telecom Hardware Media And Web Research Annex	90d
RES-029	TLA+ Formal Verification for Distributed State Machines	180d
RES-031	Spatial Computing (WebXR) & Immersive Data Visualization	120d

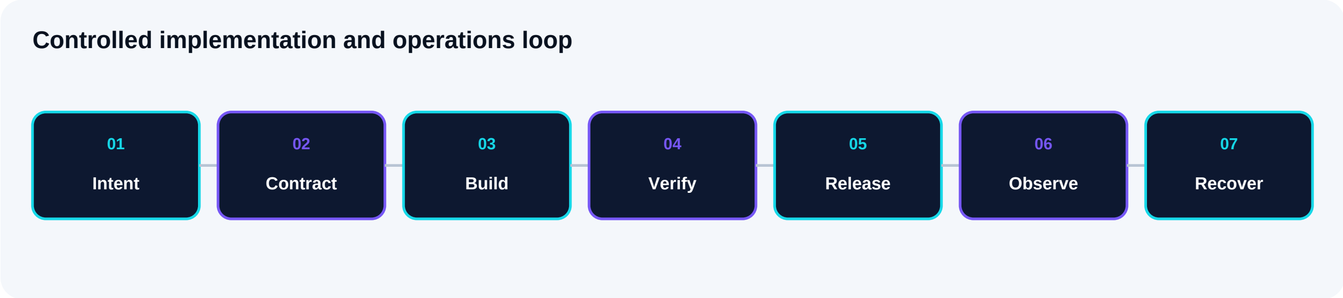
6. Comparative evaluation and authority support

ID	Publication	Class
AUTH-006	OWASP and CERT Application, API, AI, Mobile, and Disclosure Crosswalk	Authority Crosswalk
CMP-002	Enterprise Next-Generation Firewall Platforms	Comparative Evaluation
CMP-003	Enterprise SIEM Platforms	Comparative Evaluation
CMP-005	Modern Private Networking and Zero-Trust Overlay Platforms	Comparative Evaluation
CMP-006	Enterprise Campus Networking Platforms	Comparative Evaluation
CMP-007	Infrastructure as Code and Control-Plane Platforms	Comparative Evaluation
MYTHOS-CMP-COMP-SWPLT-0001	Platform Automation Comparative Evaluation Compendium	Evaluation Compendium

7. Evidence and assurance model

Evidence layer	Minimum retained evidence
Intent	Owner, scope, requirements, exceptions, risk, expected state.
Architecture	Boundaries, identities, dependencies, state/data flows, failure domains.
Pre-change / pre-release	Static analysis, simulation, policy checks, negative tests, versioned candidate.
Execution	Stable change/release ID, actor, authorization, timestamps, commands/actions, outcome codes.
Observed result	Telemetry, traces/logs/metrics, path or transaction evidence, health and correctness checks.
Recovery	Rollback/forward-recovery evidence, restored state, residual risk, post-incident learning.

8. Implementation sequence



Expand blast radius only after the preceding stage has retained evidence. Automation should encode a proven manual or modeled process; it should not make an undocumented process execute faster.

9. Maturity path

Stage	Demonstrated capability
Foundation	Explain critical state and failure behavior from first principles; identify owners and authorities.
Build	Implement the smallest representative system with typed boundaries, tests, and telemetry.
Operate	Diagnose and recover from injected failure while preserving causal evidence.
Automate	Convert repeatable work into idempotent, policy-bound, observable automation.
Architect	Design for scale, security, resilience, lifecycle change, interoperability, and governance.
Explore	Evaluate emerging techniques using hypotheses, limitations, reproduced evidence, and adoption gates.

10. Review gate

Advance a design or operating pattern only when another qualified engineer can reproduce the evidence, explain the architecture and authority boundaries, identify likely failure modes, distinguish measured facts from assumptions, and execute or validate recovery within the declared scope.

11. Limitations

This guide is an integration and navigation publication. Component standards remain authoritative for their Mythos requirements. External authorities remain with their issuers, and environment-specific product behavior must be revalidated against current versions and observed evidence.