ENGINEERING STANDARDS LIBRARY / PLATFORM ENGINEERING

Managed Platform and Agentic Service Evaluation Standard

Evidence-driven selection and governance of managed platforms, hosted runtimes, agent services, and external control planes.

PERMANENT DOCUMENT ID

MYTHOS-PLT-0004

PUBLICATION

Candidate Standard
Version 1.0.0-candidate

ASSURANCE PROFILE

Standard / Advanced
High Assurance

PERMANENT PUBLICATION IDENTITY

Managed Platform and Agentic Service Evaluation Standard

DOCUMENT ID
MYTHOS-PLT-0004

VERSION
1.0.0-candidate

STATUS
Candidate Standard

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-24

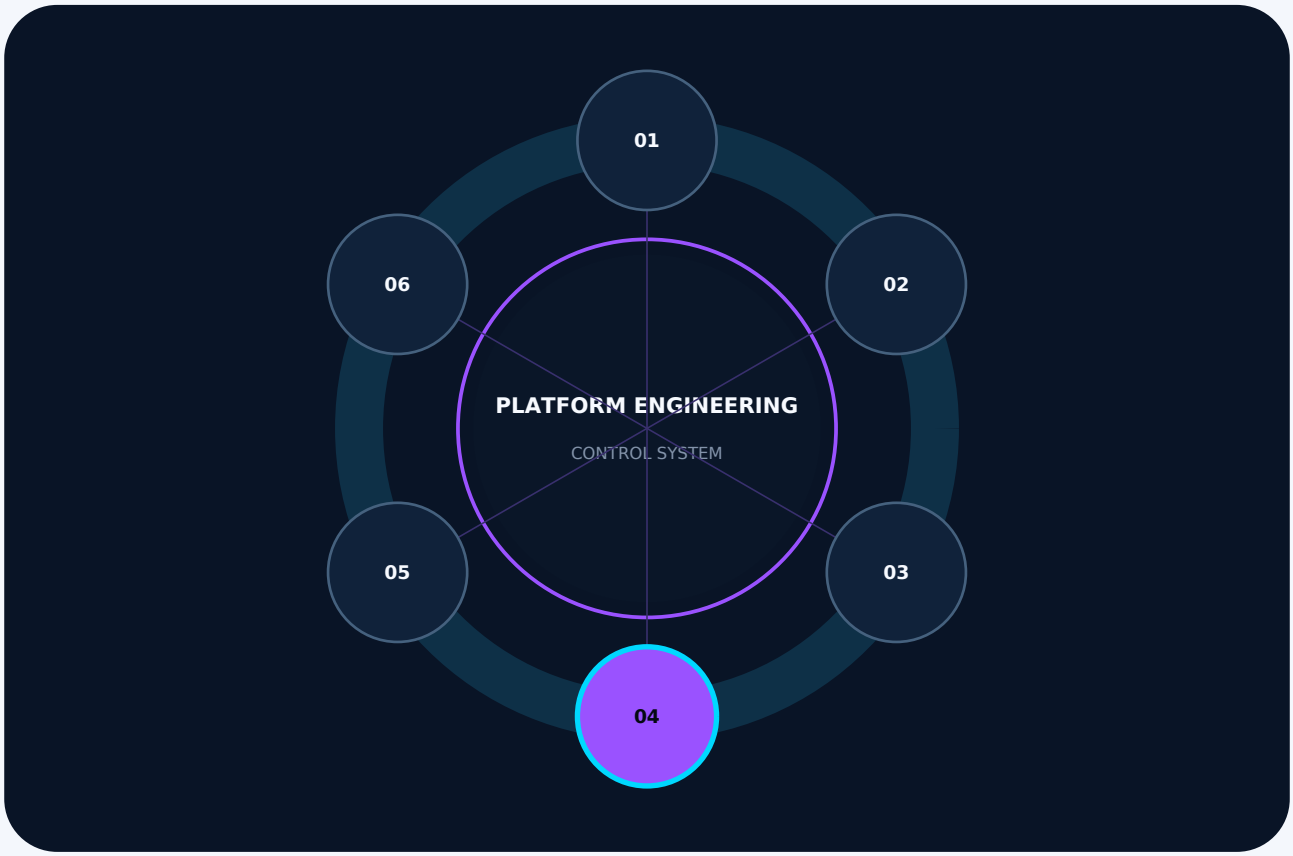
SCHEDULED REVIEW
2027-01-24

12	6	4	1
ATOMIC CRITERIA	ASSURANCE VIEWS	IMPLEMENTATION PROFILES	PERMANENT IDENTITY

Publication doctrine. This publication establishes durable guidance or requirements for its defined scope. Interpretation must preserve the permanent document identity, recorded evidence, controlled exceptions, and formal revision history.

DOMAIN CONTROL SYSTEM

MYTHOS-PLT-0004 inside the platform engineering standard constellation



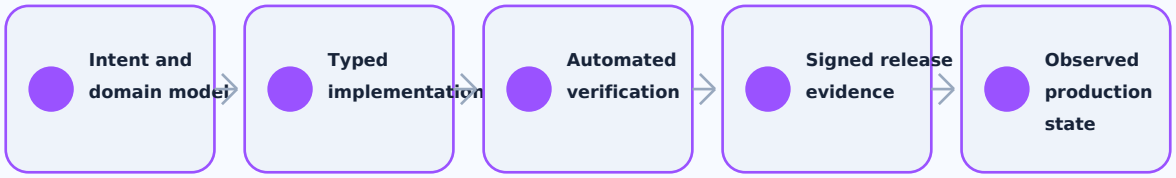
Connected assurance	Architecture, implementation, operations, evidence, and lifecycle are evaluated as one controlled system.
Specialization rule	Companion standards may add precision, but they cannot silently weaken this publication's requirements.

EXECUTIVE POSITION

Managed Platform and Agentic Service Evaluation Standard

The architecture of managed platform adoption has failed.

Evidence-bearing delivery path



Assurance maturity spectrum

0	1	2	3	4	5
Absent	Ad hoc	Repeatable	Controlled	Measured	Adaptive

CONTENTS

Publication navigation

MYTHOS-PLT-0004 inside the platform engineering standard constellation . . . 3

Managed Platform and Agentic Service Evaluation Standard . . . 4

Assurance model and evaluation surface . . . 7

Criteria and evidence map . . . 8

Requirements, evaluation methodology, and implementation guidance . . . 9

Executive Mandate . . . 9

Scope and Applicability . . . 9

In Scope . . . 9

Out of Scope . . . 10

Audience . . . 10

Definitions and Taxonomy . . . 10

Managed Platform Dimensions . . . 10

The Managed Platform Doctrine . . . 10

Evaluation Methodology . . . 11

Scoring Model . . . 11

Weighting . . . 11

Evaluation Categories . . . 11

Vendor Lock-in and Escape Hatches (Weight: 20%) . . . 11

Architectural Portability . . . 11

Authority and Tool Execution (Weight: 20%) . . . 12

Zero-Trust Boundary . . . 12

Data Sovereignty and Egress Control (Weight: 15%) . . . 12

Prompt & Context Protection . . . 12

State and Memory Portability (Weight: 15%) . . . 13

Agent Continuity . . . 13

Observability and Evidence Export (Weight: 15%) . . . 13

Telemetry Independence . . . 13

Operational Control and CI/CD (Weight: 15%) . . . 13

Lifecycle Management . . . 13

The Mythos Managed Platform Suite (MMPS) 14

Suite Composition 14

MMPS-LockIn 14

MMPS-Authority 14

Execution Protocol 14

Implementation evidence and review gates 15

Current authority register 16

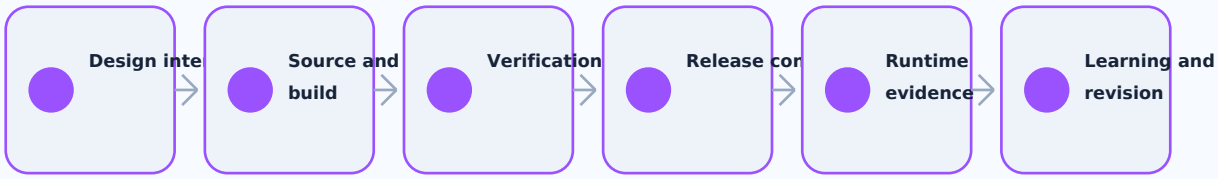
CONTROL TOPOLOGY

Assurance model and evaluation surface

The standard is assessed as an evidence-bearing system. Capability scores do not replace mandatory gates, implementation proof, or operational validation.

Capability family	Evaluation nodes
Vendor Lock-in and Escape Hatches (Weight: 20%)	1
Authority and Tool Execution (Weight: 20%)	1
Data Sovereignty and Egress Control (Weight: 15%)	1
State and Memory Portability (Weight: 15%)	1
Observability and Evidence Export (Weight: 15%)	1
Operational Control and CI/CD (Weight: 15%)	1
Suite Composition	2
Additional Evaluation Dimension	4

Lifecycle control loop



EVALUATION INDEX

Criteria and evidence map

#	Capability family	Criterion
1	Vendor Lock-in and Escape Hatches (Weight: 20%)	Architectural Portability
2	Authority and Tool Execution (Weight: 20%)	Zero-Trust Boundary
3	Data Sovereignty and Egress Control (Weight: 15%)	Prompt & Context Protection
4	State and Memory Portability (Weight: 15%)	Agent Continuity
5	Observability and Evidence Export (Weight: 15%)	Telemetry Independence
6	Operational Control and CI/CD (Weight: 15%)	Lifecycle Management
7	Suite Composition	MMPS-LockIn
8	Suite Composition	MMPS-Authority
9	Additional Evaluation Dimension	Managed Platform Dimensions
10	Additional Evaluation Dimension	The Managed Platform Doctrine
11	Additional Evaluation Dimension	Scoring Model
12	Additional Evaluation Dimension	Suite Composition

NORMATIVE PUBLICATION

Requirements, evaluation methodology, and implementation guidance

The following body preserves the substantive source standard while normalizing publication status, identity, navigation, and visual presentation.

Executive Mandate

The architecture of managed platform adoption has failed.

Organizations, desperate to capitalize on the AI hype cycle, rush to adopt proprietary managed platforms like AWS Bedrock AgentCore or Itential. They build complex, mission-critical autonomous workflows directly into the vendor's control plane using proprietary SDKs and visual drag-and-drop builders. When the vendor changes the pricing model, deprecates a feature, or suffers a multi-region outage, the organization realizes its core business logic is held hostage.

This standard exists because:

1. Time-to-Value is Not Architecture: A managed platform that lets you build an agent in 10 minutes is a prototype, not a production strategy. Coupling your core intellectual property (agent logic, memory, tool definitions) to a single vendor's proprietary API is architectural surrender.
2. Managed Platforms are Black Boxes: Vendors claim "enterprise security," but if your agent's reasoning loop, prompt history, and tool execution logs live entirely inside the vendor's observability stack, you have zero verifiable evidence of what the agent actually did.
3. Authority Must Not Be Delegated to the Vendor: A managed platform that executes agent actions (e.g., modifying infrastructure, calling APIs) using the vendor's internal service roles is a zero-trust nightmare. The customer MUST retain absolute cryptographic control over tool execution and capability grants, even if the vendor hosts the reasoning loop.
4. Agent State Must Be Portable: If an agent's memory, session state, and checkpoint history cannot be exported in an open standard (e.g., OpenTelemetry, standard JSON/Protobuf) and migrated to a self-hosted runtime, the organization does not own the agent; the vendor does.

This document establishes the Mythos Managed Platform Standard (MMPS), a comprehensive, evidence-based methodology for evaluating managed agentic and automation platforms against lock-in risk, authority separation, and operational sovereignty.

Scope and Applicability

In Scope

This standard covers the evaluation of:

1. Managed AI agent platforms (AWS Bedrock AgentCore, Azure AI Foundry, Google Agent Builder)
1. Enterprise orchestration/automation platforms (Itential, Ansible Automation Platform, Workato)

1. SaaS-based workflow and integration engines
1. Vendor lock-in metrics and escape-hatch architecture
1. Authority separation in managed runtimes (who executes the tool?)
1. Data sovereignty and prompt egress controls in SaaS platforms

Out of Scope

1. Open-source, self-hosted agentic frameworks (covered in MYTHOS-AI-0001)
1. General multi-cloud infrastructure design (covered in MYTHOS-CLD-0001)
1. General identity and access management (covered in MYTHOS-ID-0001)

Audience

1. Principal engineers and architects evaluating SaaS/PaaS AI solutions
1. Security teams evaluating third-party agent risk
1. Procurement and vendor management teams
1. CTOs and engineering leaders making multi-year platform commitments
1. Standards bodies seeking a reference vendor evaluation methodology

Definitions and Taxonomy

Managed Platform Dimensions

Dimension	Definition
Escape Hatch	A documented, tested architectural strategy for migrating an agent's logic, state, and memory off a managed platform to a self-hosted or alternative runtime without a complete rewrite.
Authority Boundary	The architectural line determining who executes a tool. In a zero-trust model, the managed platform reasons; the customer's independent control plane executes.
Vendor Lock-in (Cognitive)	The reliance on proprietary visual builders (e.g., drag-and-drop nodes) that cannot be version-controlled, peer-reviewed, or exported as declarative code.
State Portability	The ability to export an agent's temporal memory, session context, and execution history in an open, machine-readable format.
Observability Egress	The capability to stream telemetry, prompts, and tool execution logs out of the vendor's platform into the customer's tamper-evident ledger in real-time.

The Managed Platform Doctrine

> A drag-and-drop builder is a prototype, not an architecture. A reasoning loop you cannot inspect is a black box. An agent you cannot migrate is a hostage. If the vendor executes the tools, you have surrendered your zero-trust boundary.

Evaluation Methodology

Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; 100% proprietary lock-in, vendor executes tools, no data egress
1	Basic API access exists, but state/logic is trapped in the vendor cloud
2	Functional platform, but lacks portable state or independent authority enforcement
3	Solid production performance; declarative code, portable state, observability egress, customer-bound execution
4	Strong performance; multi-region failover, zero-knowledge tool execution, formally tested escape hatches
5	Best-in-class; sets the standard for sovereign, portable, zero-trust managed agentic platforms

Weighting

Category	Weight	Rationale
Vendor Lock-in & Escape Hatches	20%	Core IP trapped in a SaaS guarantees multi-year architectural debt
Authority & Tool Execution	20%	Vendors executing tools with their own credentials bypasses zero-trust
Data Sovereignty & Egress Control	15%	Prompts and context sent to vendor APIs must be strictly governed
State & Memory Portability	15%	Agent memory must be exportable to prevent platform hostage situations
Observability & Evidence Export	15%	Vendor UIs are not evidence; telemetry must stream to customer ledgers
Operational Control & CI/CD	15%	Visual builders break GitOps and Infrastructure-as-Code paradigms

Total: 100%

A system MUST score at least 3.0 in Vendor Lock-in and Authority/Tool Execution to be considered for production use.

Evaluation Categories

Vendor Lock-in and Escape Hatches (Weight: 20%)

Architectural Portability

Can the organization migrate off this platform without a total rewrite?

Score	Criteria
0	Logic built entirely in proprietary visual UI; no code export; logic trapped in SaaS
1	Logic can be exported as JSON, but is tightly coupled to vendor-specific schema

Score	Criteria
2	SDKs provided for code-first development, but runtime depends on vendor APIs
3	Open-standard protocols (MCP, A2A) used; agent logic is declarative and runnable on alternative runtimes
4	Documented and tested escape hatch: migration to open-source runtime (e.g., LangGraph) verified via CI/CD
5	Perfect portability: formally verified logic abstraction, zero vendor-specific runtime dependencies, sub-day migration capability

Authority and Tool Execution (Weight: 20%)

Zero-Trust Boundary

Who executes the consequential actions (tools) proposed by the agent?

Score	Criteria
0	Vendor platform executes tools directly using vendor-managed IAM roles
1	Customer provides static API keys to the vendor platform for tool execution
2	Vendor platform calls customer webhooks, but lacks cryptographic payload signing
3	Strict separation: Vendor reasons, customer's independent control plane (e.g., PANTHEON Aegis) executes and validates
4	Customer-bound capability grants: vendor cannot invoke a tool without a cryptographically attenuated, short-lived token from the customer
5	Perfect authority: formally verified zero-trust boundary, zero vendor ability to execute actions independently, cryptographic non-repudiation of all tool calls

Data Sovereignty and Egress Control (Weight: 15%)

Prompt & Context Protection

Is enterprise data protected from the vendor's training pipelines and internal logging?

Score	Criteria
0	Prompts and RAG context ingested by vendor may be used for model training; no egress controls
1	Vendor promises not to train on data (PDF policy), but architecture does not enforce it
2	Data residency options available (e.g., regional endpoints), but no customer-managed encryption keys (CMK)
3	CMK encryption enforced; zero data retention policies configured at the API level
4	Context-aware DLP intercepts and redacts PII/IP before prompts ever reach the managed platform
5	Perfect sovereignty: formally verified zero-knowledge execution (FHE/TEE), cryptographic proof that vendor cannot inspect payload, zero training data exfiltration risk

State and Memory Portability (Weight: 15%)

Agent Continuity

If the organization leaves the platform, does the agent retain its memory?

Score	Criteria
0	Agent memory and session state locked in vendor proprietary database
1	Basic session history export available via API (manual download)
2	Short-term memory exportable, but long-term temporal knowledge graph is trapped
3	Full state export in open standard (e.g., JSON/Protobuf); memory can be synced to customer DB in real-time
4	Bidirectional state sync: customer's local-first memory (e.g., Mnemosyne) is the source of truth; vendor pulls from it
5	Perfect portability: formally verified state synchronization, zero memory lock-in, seamless continuity across heterogeneous runtimes

Observability and Evidence Export (Weight: 15%)

Telemetry Independence

Does the organization rely on the vendor's dashboards, or does it own the evidence?

Score	Criteria
0	Logs and traces only viewable in the vendor's web UI
1	Basic log export to S3/Blob available (nightly batches)
2	Webhooks for major events, but lacks full cognitive tracing
3	Native OpenTelemetry (OTel) export with GenAI semantic conventions streaming to customer's SIEM/observability stack
4	Tamper-evident evidence bundles (hash-chained) generated for high-risk actions, exported instantly
5	Perfect observability: formally verified telemetry completeness, zero reliance on vendor UI for audit, cryptographic proof of cognitive trace integrity

Operational Control and CI/CD (Weight: 15%)

Lifecycle Management

Can the agent be deployed, rolled back, and version-controlled using standard DevSecOps practices?

Score	Criteria
0	Agent updates require manual clicks in the vendor UI
1	CLI tools exist, but lack declarative state management
2	API-driven updates, but no GitOps reconciliation
3	Fully declarative (IaC/Terraform/CloudFormation); agent definitions version-controlled in Git
4	CI/CD pipeline enforces policy-as-code (OPA) on agent configurations before vendor API accepts changes

Score	Criteria
5	Perfect control: formally verified CI/CD boundaries, zero manual UI intervention, mathematical proof of configuration parity

The Mythos Managed Platform Suite (MMPS)

Traditional vendor evaluations measure feature checklists. The MMPS evaluates lock-in risk, authority boundaries, and exit strategies.

Suite Composition

MMPS-LockIn

1. Export & Re-Host Test: 100+ tests exporting an agent's logic, state, and memory from the managed platform and deploying it to an open-source runtime (e.g., LangGraph), verifying functionality parity.
1. UI Dependency Audit: 50+ tests verifying that no critical operational task (rollback, scaling, secret rotation) requires access to the vendor's web UI.

MMPS-Authority

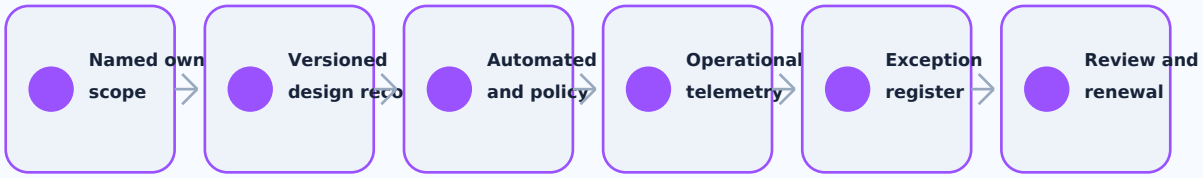
1. Tool Execution Test: 100+ tests simulating a vendor platform compromise, verifying the attacker cannot execute customer-side tools without the customer's ephemeral capability grant.
1. Egress DLP Test: 50+ tests injecting PII into the agent's context, verifying the data sovereignty layer redacts it before it hits the vendor's inference API.

Execution Protocol

ASSURANCE AND ADOPTION

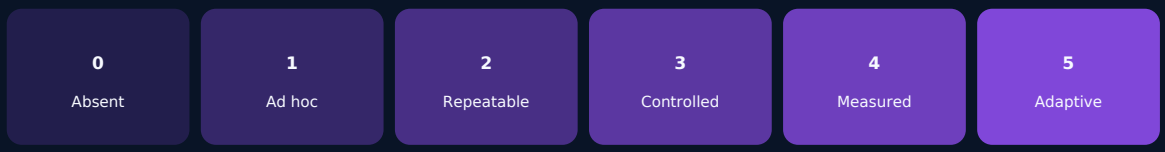
Implementation evidence and review gates

Required assurance chain



A passing score is not sufficient without current evidence. High-assurance adoption requires traceable design decisions, reproducible builds, independently reviewable tests, controlled secrets, runtime observability, failure exercises, and a dated exception register.

Assurance maturity spectrum



Current authority register

Authority	Version / snapshot	Applicability	Review by
Kubernetes Project: Kubernetes Documentation	v1.36 current documentation	High	2026-10-22
SLSA Project: Supply-chain Levels for Software Artifacts Specification	1.2 current specification snapshot	Moderate	2026-10-22
SPDX Project: SPDX Specification	3.0.1 stable; 3.1-RC1 under review	High	2026-10-22
OWASP CycloneDX: CycloneDX Bill of Materials Standard	1.7	High	2027-01-24
OpenTelemetry Project: OpenTelemetry Documentation and Specification	Rolling specification snapshot; profiles public alpha and declarative configuration stable in 2026	High	2026-10-22
Cloud Native Computing Foundation: Cloud Native Landscape and Project Documentation	Rolling catalog snapshot	Moderate	2026-08-23

Research limitation. Official documentation establishes published interfaces and declared behavior. It does not prove the performance, security, interoperability, cost, or operational suitability of a specific implementation. Those claims require controlled testing and retained evidence.