ENGINEERING STANDARDS LIBRARY / PLATFORM ENGINEERING

Digital Marketplace and Extension Architecture Standard

Governed extension discovery, trust tiers, package integrity, compatibility, monetization boundaries, and marketplace operations.

PERMANENT DOCUMENT ID

MYTHOS-PLT-0006

PUBLICATION

Candidate Standard
Version 1.0.0-candidate

ASSURANCE PROFILE

Standard / Advanced
High Assurance

PERMANENT PUBLICATION IDENTITY

Digital Marketplace and Extension Architecture Standard

DOCUMENT ID
MYTHOS-PLT-0006

VERSION
1.0.0-candidate

STATUS
Candidate Standard

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-24

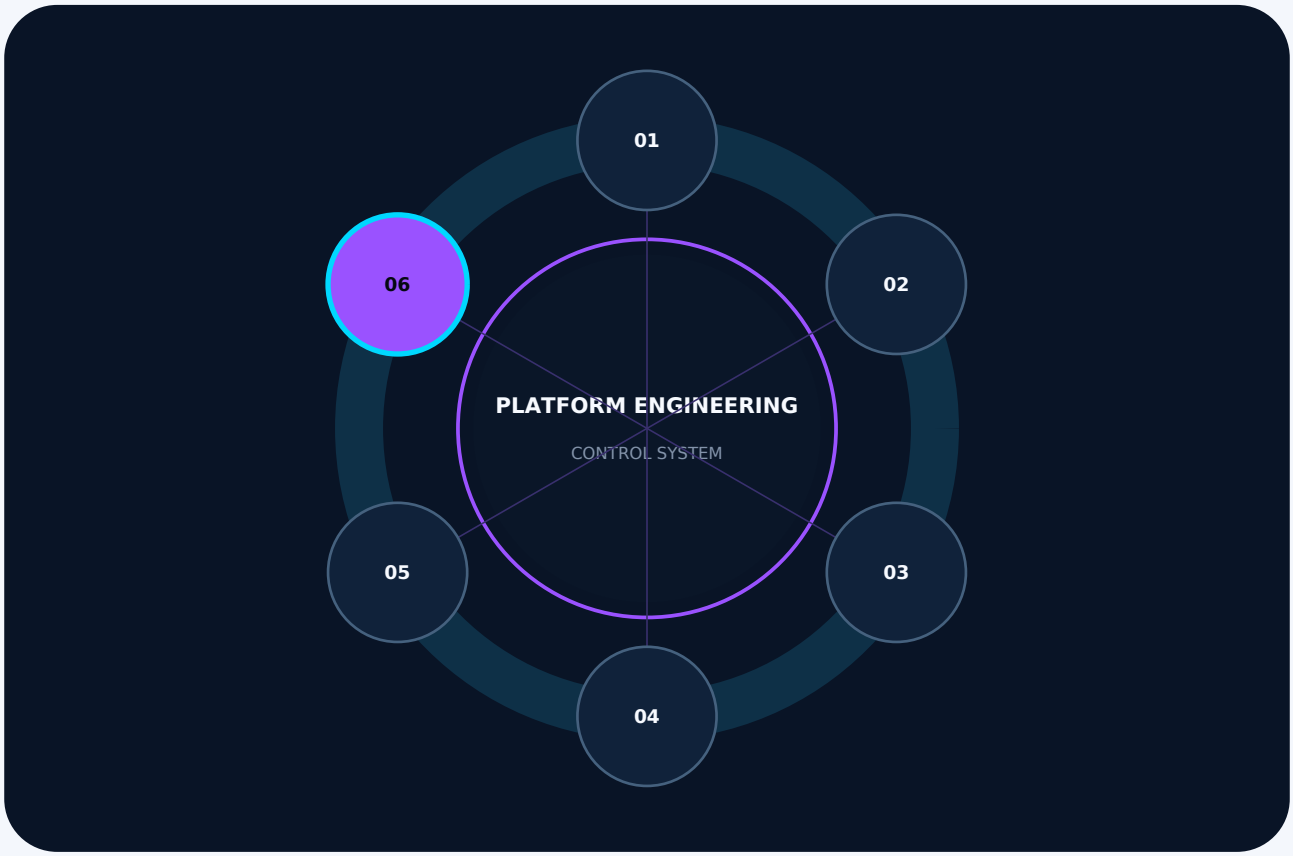
SCHEDULED REVIEW
2027-01-24

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-0006 inside the platform engineering standard constellation



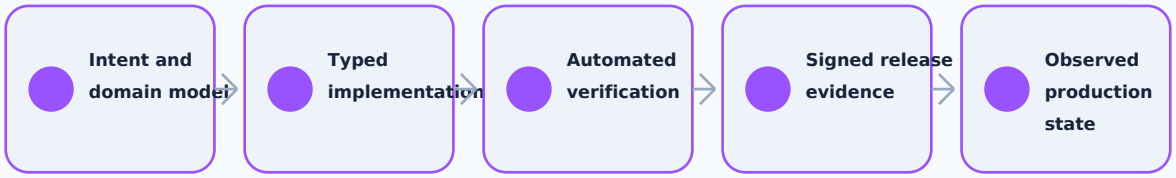
Connected assurance	Architecture, implementation, operations, evidence, and lifecycle are evaluated as one controlled system.
Specialization rule	Companion standards may add precision, but they cannot silently weaken this publication's requirements.

EXECUTIVE POSITION

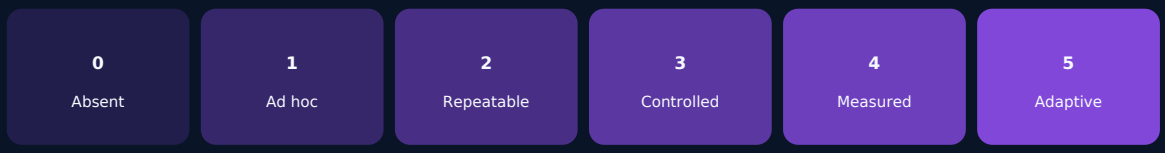
Digital Marketplace and Extension Architecture Standard

The architecture of digital marketplaces and extension ecosystems has failed.

Evidence-bearing delivery path



Assurance maturity spectrum



CONTENTS

Publication navigation

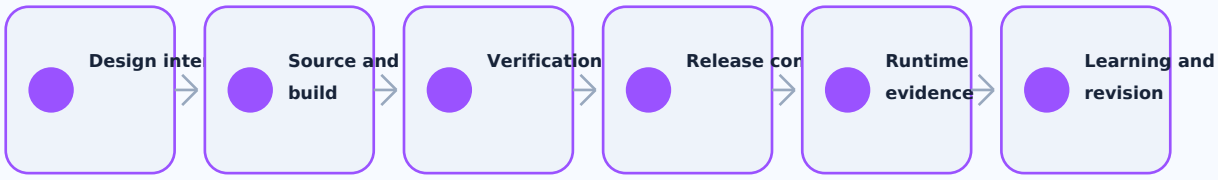
CONTROL TOPOLOGY

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
Sandboxed Execution and Isolation (Weight: 20%)	1
Capability-Based Authorization (Weight: 20%)	1
Supply Chain and Transparency Logs (Weight: 15%)	1
Revocation and Kill Switches (Weight: 15%)	1
DRM and Licensing Resilience (Weight: 15%)	1
Developer Experience (DX) and APIs (Weight: 15%)	1
Suite Composition	2
Additional Evaluation Dimension	4

Lifecycle control loop



EVALUATION INDEX

Criteria and evidence map

#	Capability family	Criterion
1	Sandboxed Execution and Isolation (Weight: 20%)	Boundary Integrity
2	Capability-Based Authorization (Weight: 20%)	Permission Scoping
3	Supply Chain and Transparency Logs (Weight: 15%)	Digital Good Provenance
4	Revocation and Kill Switches (Weight: 15%)	Threat Neutralization
5	DRM and Licensing Resilience (Weight: 15%)	Ownership Verification
6	Developer Experience (DX) and APIs (Weight: 15%)	Publishing Pipeline
7	Suite Composition	MMS-Sandbox
8	Suite Composition	MMS-SupplyChain
9	Additional Evaluation Dimension	Marketplace Dimensions
10	Additional Evaluation Dimension	The Marketplace 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 digital marketplaces and extension ecosystems has failed.

Organations build platform stores (like Chrome Extensions or app stores) that rely on "review theater" rather than architectural isolation. They allow third-party developers to upload native code, unverified AI tools, and opaque model weights. When an extension goes rogue—or is purchased by a malicious actor—it silently exfiltrates user data, injects prompt payloads into the host AI, or crashes the system because it was granted blanket permissions at install time.

This standard exists because:

1. Review Theater is Not Security: Manual vetting of marketplace code cannot catch obfuscated malware or zero-day exploits. Extensions, bolt-on features, and MCP servers MUST be executed inside hardware-backed or WebAssembly (WASM) sandboxes with strict memory isolation.
2. Blanket Permissions are an Attack Vector: An avatar pack or a UI theme should not have access to the file system or network. An MCP tool should only be able to read specific directories. Systems MUST enforce deny-by-default, capability-based security where extensions request granular, cryptographically verifiable scopes.
3. Digital Goods Require Supply Chain Provenance: A malicious voice model or a poisoned MCP server distributed through the PANTHEON marketplace can backdoor the entire infrastructure. All digital goods MUST be signed (Sigstore/Cosign), carry an AIBOM/SBOM, and be recorded in a transparency log (Rekor) before publication.
4. Centralized DRM is a Single Point of Failure: If the marketplace licensing server goes down, users lose access to their purchased avatars and tools. Digital Rights Management (DRM) and licensing MUST be decentralized, utilizing zero-knowledge proofs (ZKPs) or offline cryptographic tokens to verify ownership without phoning home.

This document establishes the Mythos Marketplace Standard (MMS), a comprehensive, evidence-based methodology for evaluating digital marketplace architectures against sandboxed execution, capability scoping, and supply chain integrity.

Scope and Applicability

In Scope

This standard covers the evaluation of:

1. Digital marketplace architectures (PANTHEON, Chrome Web Store, VS Code Marketplace)

1. Extension and plugin execution models (WASM, microVMs, native processes)
1. MCP (Model Context Protocol) server and agentic tool distribution
1. Digital Rights Management (DRM) and offline licensing (ZKPs, token-bound ownership)
1. Supply chain security for digital goods (avatars, voices, themes, code)
1. Revocation, kill switches, and transparency logs (Rekor)

Out of Scope

1. General AI agent framework evaluation (covered in MYTHOS-AI-0001)
1. General WebAssembly runtime evaluation (covered in MYTHOS-EMT-0001)
1. General identity and PKI (covered in MYTHOS-ID-0001/0002)

Audience

1. Platform engineers building marketplaces and extension hosts (PANTHEON)
1. Security engineers evaluating third-party code and supply chains
1. AI engineers building MCP tool registries and agentic capability boundaries
1. UX/UI designers distributing themes and avatars
1. Standards bodies seeking a reference marketplace security methodology

Definitions and Taxonomy

Marketplace Dimensions

Dimension	Definition
Sandboxed Extension	A bolt-on feature, MCP tool, or theme that executes inside a WebAssembly (WASM) or microVM boundary, strictly isolated from the host application's memory and system resources.
Capability Grant	A cryptographically signed, non-forgeable token granting an extension access to a specific resource (e.g., `read:./project/`, `fetch:api.weather.com`) for a limited time.
Transparency Log (Rekor)	An append-only cryptographic ledger that records the publication of digital goods (extensions, avatars, voices), providing verifiable proof of what was published and when.
Zero-Knowledge DRM	A licensing model where the user proves ownership of a digital good (e.g., a premium avatar) via a Zero-Knowledge Proof, without revealing their identity or requiring a centralized server check.
Kill Switch	A cryptographic revocation mechanism embedded in the host platform that instantly disables a distributed extension or digital good if it is found to be malicious.

The Marketplace Doctrine

> A vetted extension is still a threat. A UI theme with network access is a spy. A marketplace without a transparency log is a malware distribution engine. If the permissions are not granular, the user is compromised.

Evaluation Methodology

Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; native code execution, blanket permissions, no signing, centralized DRM
1	Basic review process, but no architectural sandboxing or capability scoping
2	Functional extensions, but lacks WASM isolation or supply chain attestations
3	Solid production performance; WASM sandboxed, capability grants, signed goods, transparency logs
4	Strong performance; ZKP offline DRM, automated kill switches, formally verified isolation
5	Best-in-class; sets the standard for mathematically proven, zero-trust digital distribution

Weighting

Category	Weight	Rationale
Sandboxed Execution & Isolation	20%	Native extensions guarantee host compromise eventually
Capability-Based Authorization	20%	Blanket permissions allow themes/tools to exfiltrate data
Supply Chain & Transparency Logs	15%	Unsigned digital goods allow supply chain poisoning
Revocation & Kill Switches	15%	Malicious extensions must be instantly neutralized globally
DRM & Licensing Resilience	15%	Centralized licensing servers create single points of failure
Developer Experience (DX) & APIs	15%	If publishing is hard, developers will bypass the marketplace

Total: 100%

A system MUST score at least 3.0 in Sandboxed Execution and Capability Authorization to be considered for production use.

Evaluation Categories

Sandboxed Execution and Isolation (Weight: 20%)

Boundary Integrity

How are third-party extensions, tools, and avatars executed?

Score	Criteria
0	Native executables or direct script injection into host process (critical failure)
1	Process isolation, but extensions share host filesystem/network access

Score	Criteria
2	Containerized execution, but heavy startup latency
3	WebAssembly (WASM) Component Model used for all extensions; strict memory isolation
4	Heterogeneous execution: WASM for logic, microVMs (Firecracker) for heavy MCP tools
5	Perfect isolation: formally verified sandbox boundaries, zero host attack surface, mathematical proof of memory protection

Capability-Based Authorization (Weight: 20%)

Permission Scoping

Are extensions restricted to least-privilege access?

Score	Criteria
0	Extensions run with the full privileges of the host application
1	Static manifest of permissions requested at install time (e.g., "Access all files")
2	Basic sandboxing, but network access is all-or-nothing
3	Deny-by-default; extensions must request granular capability grants (e.g., `fetch:api.github.com`) via the host's policy engine
4	Capabilities are time-boxed and attenuated; an MCP tool cannot delegate more access than it possesses
5	Perfect authorization: formally verified capability bounds, zero ability to bypass scopes, cryptographic non-repudiation of resource access

Supply Chain and Transparency Logs (Weight: 15%)

Digital Good Provenance

Are avatars, voices, and MCP tools verified before distribution?

Score	Criteria
0	Unverified files uploaded directly to marketplace CDN
1	Manual malware scan before publication
2	Artifacts signed by the developer, but marketplace does not verify signatures
3	Mandatory Sigstore/Cosign signing; SBOM/AIBOM generated for all goods
4	All publications recorded in a transparency log (Rekor); marketplace refuses to serve unsigned goods
5	Perfect provenance: formally verified supply chain, in-toto attestations bound to artifacts, zero ability to distribute poisoned goods

Revocation and Kill Switches (Weight: 15%)

Threat Neutralization

Can a malicious extension be instantly disabled globally?

Score	Criteria
0	No revocation mechanism; relies on users manually uninstalling
1	Marketplace page removed, but extension continues to function if installed
2	Centralized blocklist checked at startup, but fails offline
3	Cryptographic revocation (CRL/OCSP) checked before execution; cached for offline use
4	Hard kill switch: host platform queries Rekor transparency log and instantly neutralizes malicious extensions at runtime
5	Perfect neutralization: formally verified revocation propagation, zero ability for revoked code to execute, sub-second global kill capability

DRM and Licensing Resilience (Weight: 15%)

Ownership Verification

How is user ownership of premium digital goods verified?

Score	Criteria
0	No DRM; goods are freely copyable or tied to a static, easily bypassed license key
1	Centralized licensing server checked on every launch (fails offline)
2	Online activation with offline grace period, but server outage blocks new activations
3	Token-bound licensing: ownership bound to the user's cryptographic identity (e.g., Passkey/SPIFFE)
4	Zero-Knowledge DRM: user proves ownership of the marketplace token via ZKP without revealing identity or phoning home
5	Perfect resilience: formally verified licensing bounds, zero central server dependency, cryptographic proof of ownership

Developer Experience (DX) and APIs (Weight: 15%)

Publishing Pipeline

Is the publishing pipeline secure, automated, and developer-friendly?

Score	Criteria
0	Manual email submission of zip files to marketplace admins
1	Basic web portal upload, but manual signing required by the developer
2	CLI tool for publishing, but lacks CI/CD integration
3	Keyless signing via OIDC in CI/CD; automated publishing via API
4	Automated policy-as-code evaluation of extension behavior before publication
5	Perfect pipeline: formally verified publishing bounds, zero friction for developers, mathematical proof of artifact integrity before public listing

The Mythos Marketplace Suite (MMS)

Traditional marketplace benchmarks measure download speed and catalog size. The MMS evaluates sandbox escape resistance, capability enforcement, and supply chain integrity.

Suite Composition

MMS-Sandbox

1. Escape Attempt: 100+ tests executing malicious WASM modules designed to exploit memory bounds, spectre side-channels, and host API abuses, verifying the host remains uncompromised.
1. Capability Violation: 50+ tests attempting to read local files or make network calls from a UI theme or avatar pack, verifying the capability engine blocks the action.

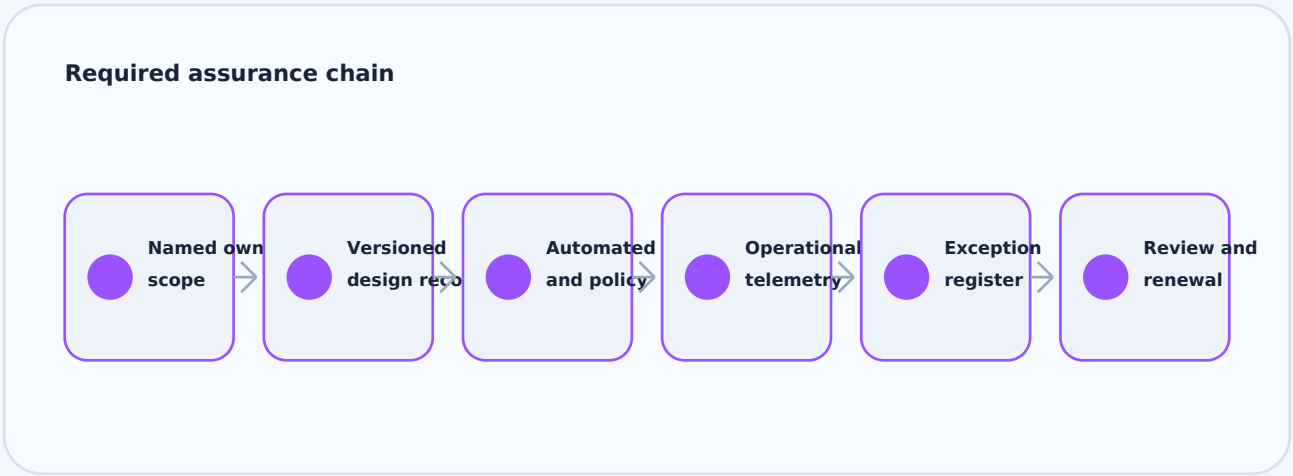
MMS-SupplyChain

1. Poisoned Good Test: 100+ tests attempting to upload an unsigned MCP tool or a voice model lacking an AIBOM, verifying the marketplace API rejects it.
1. Revocation Test: 50+ tests marking a published extension as compromised, verifying the host platforms instantly disable it for all users via the transparency log.

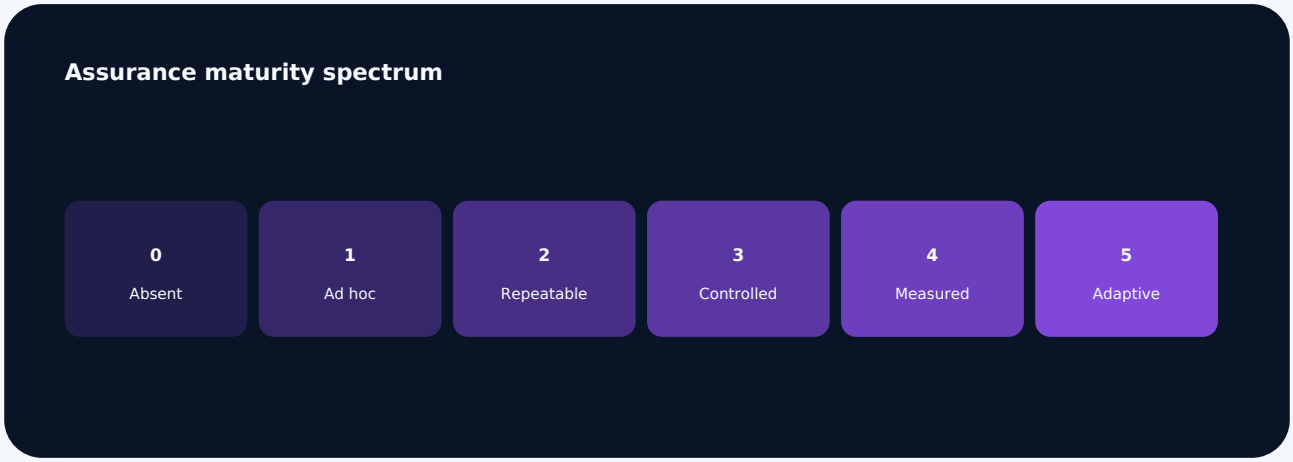
Execution Protocol

ASSURANCE AND ADOPTION

Implementation evidence and review gates



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.



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.