ENGINEERING STANDARDS LIBRARY / IDENTITY AND ACCESS

Public Key Infrastructure (PKI) & Attribute-Based Access Control (ABAC) Standard

The architecture of authorization and key management has failed.

PERMANENT DOCUMENT ID

MYTHOS-ID-0002

PUBLICATION

Candidate Standard
Version 1.0.0-candidate

ASSURANCE PROFILE

Standard / Advanced
High Assurance

PERMANENT PUBLICATION IDENTITY

Public Key Infrastructure (PKI) & Attribute-Based Access Control (ABAC) Standard

DOCUMENT ID
MYTHOS-ID-0002

VERSION
1.0.0-candidate

STATUS
Candidate Standard

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-24

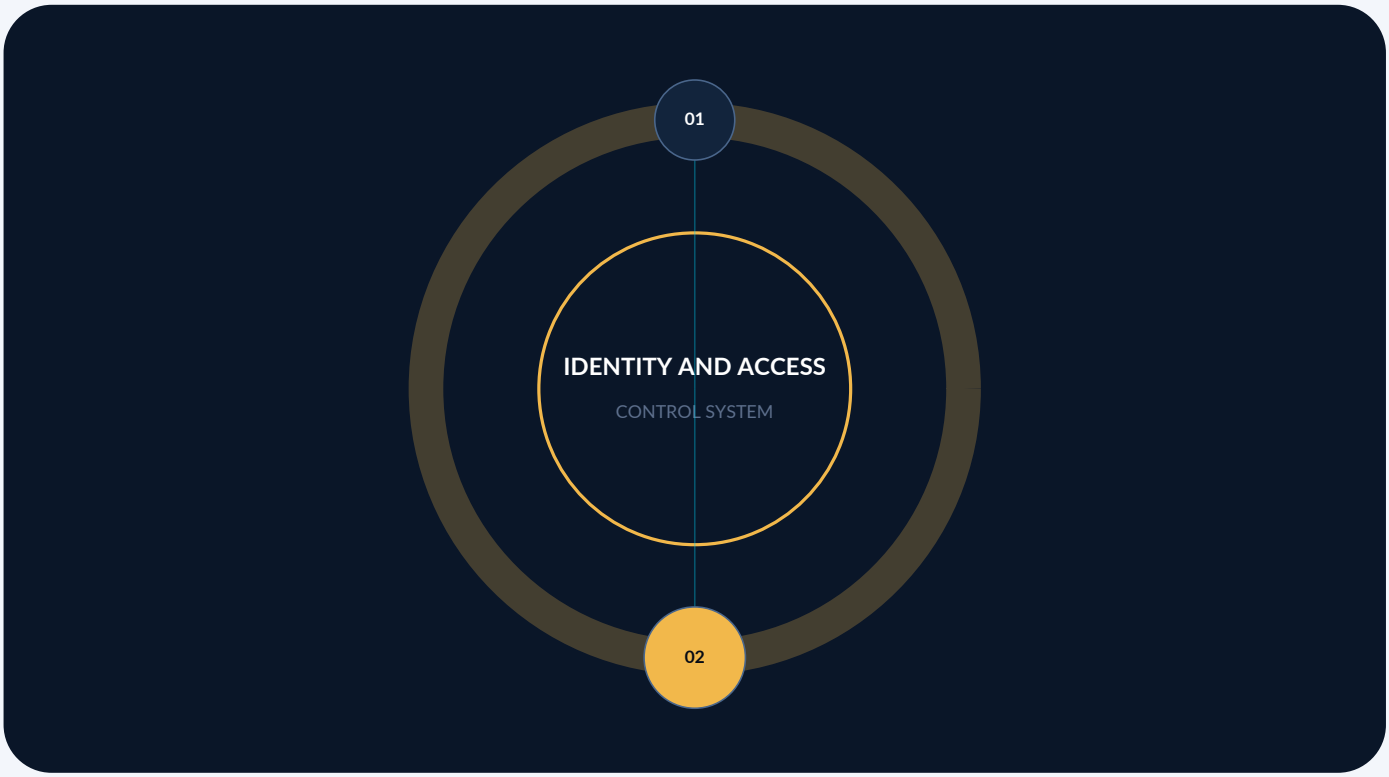
SCHEDULED REVIEW
2027-07-24

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

Publication doctrine. This standard is a permanent library identity. Interpretation must preserve its recorded evidence, controlled exceptions, formal revision history, and explicit relationship to companion standards.

DOMAIN CONTROL SYSTEM

MYTHOS-ID-0002 inside the identity and access constellation

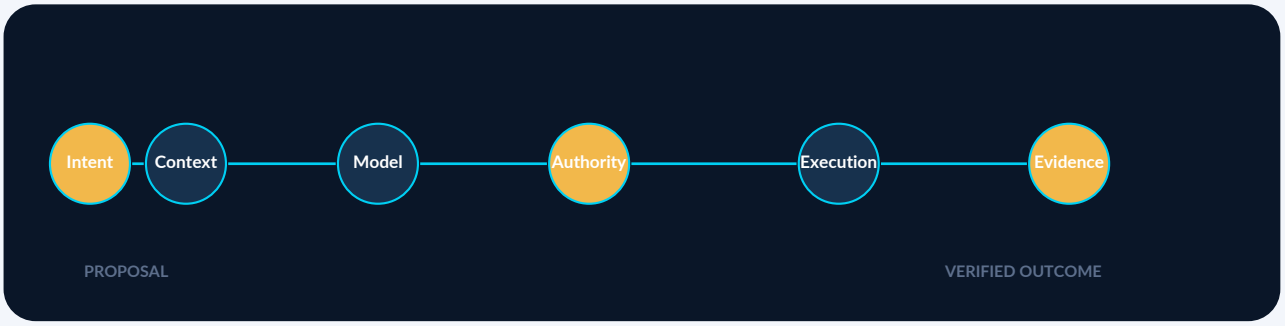


Connected assurance	Architecture, implementation, operations, evidence, and lifecycle are evaluated as one controlled system.
Specialization rule	Companion standards may add precision, but cannot silently weaken this publication.

Executive orientation

Establishes public-key infrastructure and attribute-based access-control requirements for trust anchors, certificate lifecycle, policy decision, authorization context, revocation, and evidence.

WHY THIS STANDARD EXISTS	The architecture of authorization and key management has failed.
OPERATIONAL SCOPE	This standard covers the evaluation of: - Public Key Infrastructure (PKI) and Certificate Authority (CA) hierarchies - Certificate lifecycle automation (ACME, cert-manager, SPIFFE SVIDs) - Hardware Security Modules (HSM) and TPM integration - Attribute-Based Access Control (ABAC) and Policy-as-Code (Open Policy Agent, Cedar) - The transition from Role-Based Access Control (RBAC) to ABAC - mTLS (Mutual TLS) enforcement and short-lived identity documents
PRIMARY AUDIENCE	- Security architects and PKI engineers - Platform engineers building zero-trust service meshes - Application architects decoupling authorization logic from microservices - Compliance teams managing access governance - Standards bodies seeking a reference PKI and ABAC methodology



Assurance principle. Model capability, data quality, identity, authority, operational evidence, and human accountability are treated as a connected system. A high aggregate score cannot compensate for a failed mandatory safety or authority boundary.

Contents

Evaluation criterion index

This standard contains 12 atomic criteria. Each criterion is independently testable and requires retained evidence.

ID	EVALUATION CRITERION
4.1.1	PKI Operations
4.2.1	Root of Trust
4.3.1	Authorization Model
4.4.1	Enforcement Architecture
4.5.1	Data Inputs
4.6.1	Compliance Evidence
5.1.1	MPAS-PKI
5.1.2	MPAS-ABAC
6.1.1	Certificate Authority Compromise
6.1.2	Role Explosion (RBAC Privilege Creep)
6.1.3	Stale Policy Enforcement
6.1.4	Certificate Expiry Outage

MYTHOS-ID-0002-4.1.1

4.1.1 PKI Operations

Normative source requirement

How are X.509 certificates issued, renewed, and revoked?

Score	Criteria
0	Manual CSRs; certificates last years; tracked in spreadsheets
1	Internal CA exists, but clients must manually request and install certs
2	cert-manager used in Kubernetes, but other infrastructure is manual
3	Full ACME integration across all workloads (on-prem, cloud, edge); certificates last < 90 days
4	Ephemeral identity (SPIFFE SVIDs) with cert TTLs < 1 hour; zero manual rotation
5	Perfect automation: formally verified lifecycle bounds, zero certificate-induced outages, mathematical proof of continuous validity

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-4.2.1

4.2.1 Root of Trust

Normative source requirement

Are Certificate Authority private keys protected against exfiltration?

Score	Criteria
0	Root CA private key stored on a standard VM or in plaintext
1	Intermediate CA keys password-protected on disk
2	Cloud-managed HSM (AWS CloudHSM, Azure Key Vault Premium) used for CAs
3	Dedicated, on-premises FIPS 140-2 Level 3+ HSMs; keys are non-exportable
4	TPM 2.0 attestation required for all leaf certificates; keys generated in silicon
5	Perfect security: formally verified hardware boundaries, zero ability to extract CA keys, cryptographic proof of key provenance

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-4.3.1

4.3.1 Authorization Model

Normative source requirement

Are access decisions static (based on groups) or dynamic (based on attributes)?

Score	Criteria
0	Pure RBAC; users assigned to groups ("Admins", "Users") with broad permissions
1	RBAC with some coarse context (e.g., IP-based restrictions)
2	Hybrid model: RBAC for broad access, ABAC for sensitive resources
3	Pure ABAC: roles abolished; access evaluated based on user attributes, resource tags, and environment
4	Real-time attribute evaluation: access changes dynamically if device posture degrades or geolocation shifts
5	Perfect model: formally verified policy bounds, zero standing privilege, mathematical proof of least-privilege enforcement

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-4.4.1

4.4.1 Enforcement Architecture

Normative source requirement

Is authorization logic decoupled from the application codebase?

Score	Criteria
0	Authorization logic hardcoded in application controllers (e.g., `if user.role == 'admin'`)
1	Centralized IAM, but application must query it and enforce locally
2	Policy engine (OPA/Cedar) deployed, but policies managed via UI/API
3	Policy-as-Code: policies written in Rego/Cedar, version-controlled in Git, deployed via CI/CD
4	Sidecar enforcement: every microservice delegates authz to a local OPA sidecar via mTLS
5	Perfect decoupling: formally verified policy execution, zero hardcoded authz logic, mathematical proof of policy completeness

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-4.5.1

4.5.1 Data Inputs

Normative source requirement

Does the policy engine have access to real-time, dynamic attributes for decision-making?

Score	Criteria
0	Policy engine only receives user ID and requested action
1	Basic environment context (time of day) included
2	Device posture (EDR status, OS version) fed into policy engine
3	Real-time threat intelligence (e.g., impossible travel anomalies) influences decisions
4	Resource-state awareness: policy evaluates the state of the target resource (e.g., "is this document already classified as restricted?") before authorizing
5	Perfect context: formally verified data pipelines, zero stale attributes, mathematical proof of decision accuracy

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-4.6.1

4.6.1 Compliance Evidence

Normative source requirement

Can the organization prove exactly why an access decision was made at any point in the past?

Score	Criteria
0	Logs only record "Access Granted/Denied"
1	Logs record the user and resource
2	Logs record the RBAC role used
3	ABAC decision logs: records the exact policy version, input attributes, and evaluation path
4	Cryptographic signing of authorization decisions, binding them to tamper-evident ledgers
5	Perfect provenance: formally verified audit trails, zero ability to alter decision history, cryptographic proof of compliance

---

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Reproducible benchmark results, workload definitions, raw outputs, and variance records. Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results.
ASSESSMENT METHOD	MATURITY SIGNAL
Run a version-pinned workload with representative inputs, record distributions rather than a single score, repeat under failure and load, and compare reproduced results with the stated acceptance threshold.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-5.1.1

5.1.1 MPAS-PKI

Normative source requirement

• CA Compromise Simulation: 100+ tests attempting to extract the CA private key from the HSM/TPM, verifying the hardware boundary holds.

• Rotation Stress Test: 50+ tests forcing mass certificate expiration across 10,000 microservices, verifying ACME/SPIFFE rotates them without dropping connections.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results.
ASSESSMENT METHOD	MATURITY SIGNAL
Inspect the architecture and configured control, execute normal and adverse scenarios, verify measurable outcomes, sample retained evidence, and confirm that exceptions are time-bound and approved.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-5.1.2

5.1.2 MPAS-ABAC

Normative source requirement

• Context Shift Test: 100+ tests changing a user's device posture (e.g., disabling EDR) mid-session, verifying the ABAC engine instantly revokes access on the next request.

• Policy Bypass Test: 50+ tests attempting to bypass the OPA sidecar by calling the microservice directly, verifying mTLS enforcement blocks the unauthenticated request.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results. Context manifests, retrieval traces, source citations, freshness records, conflict decisions, and memory provenance.
ASSESSMENT METHOD	MATURITY SIGNAL
Trace the decision path from identity and policy through execution, attempt bypass and privilege-escalation scenarios, verify revocation behavior, and inspect the resulting evidence record.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-6.1.1

6.1.1 Certificate Authority Compromise

Normative source requirement

Severity: Critical Description: An attacker gains access to the CA's private key. They can now mint valid certificates for any service, establishing persistent, undetectable access to the entire zero-trust network.

Required Controls: 1. CA private keys MUST be generated and stored in FIPS 140-2 Level 3+ HSMs. 2. Keys MUST be non-exportable.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results.
ASSESSMENT METHOD	MATURITY SIGNAL
Trace the decision path from identity and policy through execution, attempt bypass and privilege-escalation scenarios, verify revocation behavior, and inspect the resulting evidence record.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-6.1.2

6.1.2 Role Explosion (RBAC Privilege Creep)

Normative source requirement

Severity: High Description: Over time, an accumulates dozens of roles to perform specific tasks. The roles grant broad access. An attacker compromising the user account gains excessive lateral movement capabilities.

Required Controls: 1. Abolish RBAC in favor of ABAC. 2. Evaluate access dynamically based on specific attributes, not broad roles.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Policy configuration, identity or trust records, authorization decisions, revocation evidence, and adversarial test results.
ASSESSMENT METHOD	MATURITY SIGNAL
Trace the decision path from identity and policy through execution, attempt bypass and privilege-escalation scenarios, verify revocation behavior, and inspect the resulting evidence record.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-6.1.3

6.1.3 Stale Policy Enforcement

Normative source requirement

Severity: High Description: A developer updates the OPA policy in Git, but the CI/CD pipeline fails to deploy it to the sidecars. The microservice enforces outdated, vulnerable authorization logic.

Required Controls: 1. Policy engines MUST report their running policy version/hash. 2. Control plane MUST detect version drift and quarantine non-compliant services.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Architecture records, implemented configuration, test evidence, operational telemetry, exception decisions, and accountable approval.
ASSESSMENT METHOD	MATURITY SIGNAL
Inspect the architecture and configured control, execute normal and adverse scenarios, verify measurable outcomes, sample retained evidence, and confirm that exceptions are time-bound and approved.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

MYTHOS-ID-0002-6.1.4

6.1.4 Certificate Expiry Outage

Normative source requirement

Severity: High Description: An internal CA issues 1-year certificates. A critical API gateway's certificate expires over the weekend because no one monitored it. All API traffic fails.

Required Controls: 1. Mandatory ACME automation for all certificates. 2. Short certificate lifespans (< 90 days) to force automation.

CONTROL INTENT	REQUIRED EVIDENCE
Create a measurable, reviewable control that can be verified independently of model or vendor claims.	Architecture records, implemented configuration, test evidence, operational telemetry, exception decisions, and accountable approval.
ASSESSMENT METHOD	MATURITY SIGNAL
Inspect the architecture and configured control, execute normal and adverse scenarios, verify measurable outcomes, sample retained evidence, and confirm that exceptions are time-bound and approved.	0 absent | 1 ad hoc | 2 repeatable | 3 controlled | 4 measured | 5 continuously verified

Decision rule. A criterion is not satisfied by documentation alone when the claim concerns runtime behavior. Reproduced evidence, responsible ownership, known limitations, and controlled exceptions are required.

Current authority and freshness register

These sources inform interpretation but do not convert this candidate publication into certification, legal compliance, or implementation evidence. Rolling sources must be version-pinned at adoption time.

AUTHORITY	VERSION / FRESHNESS	APPLICABILITY LIMIT
NIST - Digital Identity Guidelines	SP 800-63-4 final, July 2025 Expires 2027-01-24	Federal guidance must be tailored for non-federal risk, jurisdiction, usability, privacy, and operational context.
W3C - Verifiable Credentials Data Model v2.0	W3C Recommendation, May 15 2025 Expires 2027-01-24	Data model conformance does not establish issuer trust, credential truth, or ecosystem governance.
W3C - Decentralized Identifiers (DIDs) v1.0	W3C Recommendation; DID 1.1 remains a Candidate Recommendation Expires 2027-01-24	DID method security, governance, resolution, and privacy properties vary materially.
SPIFFE - SPIFFE Specifications and Workload API	Rolling specification snapshot; SPIRE 1.15.2 documentation Expires 2026-10-24	Implementation and deployment assurance depend on attestation plugins, trust-domain design, and operational controls.

Source lineage appendix

The following text preserves the substantive source draft in a normalized public form. Where it conflicts with the permanent publication identity, candidate status, current authority register, or evidence requirements above, the controlled publication layer governs.

0. Executive Mandate

The architecture of authorization and key management has failed.

Organizations attempt to manage access control using static Role-Based Access Control (RBAC). As the organization scales, RBAC suffers from "role explosion"-thousands of highly specific roles that no one understands, granting excessive standing privileges. Simultaneously, they manage X.509 certificates manually, using Excel spreadsheets to track expiration dates. When a certificate expires, a critical microservice or API gateway goes offline, causing catastrophic outages.

This standard exists because:

1. RBAC is Too Rigid for Dynamic Infrastructure: A user's or agent's authority should not be defined solely by a static group membership. Authority MUST be dynamic, evaluated in real-time based on attributes (identity, device posture, time of day, geolocation, request payload). Systems MUST transition to Attribute-Based Access Control (ABAC).

2. Manual PKI is Operational Suicide: A certificate lifecycle that requires a human to generate a CSR, wait for an email, and manually install the certificate guarantees eventual outages. PKI MUST be fully automated using the ACME (Automated Certificate Management Environment) protocol or SPIFFE, with certificates lasting hours or days, not years.

3. Policy is Code, Not Configuration: Authorization logic hardcoded into application controllers or buried in proprietary IAM UIs is unmanageable. Policies MUST be decoupled from the application, written as code (e.g., Rego/Cedar), version-controlled in Git, and enforced by a centralized policy engine.

4. Private Keys Must Be Hardware-Bound: A certificate authority whose private key exists on a standard virtual machine is a compromised CA waiting to happen. The Root CA and Intermediate CA keys MUST be generated and stored inside Hardware Security Modules (HSMs) or Trusted Platform Modules (TPMs), non-exportable.

This document establishes the Mythos PKI & ABAC Standard (MPAS), a comprehensive, evidence-based methodology for evaluating key management and authorization architectures against automation rigor, dynamic policy evaluation, and cryptographic non-extractability.

Part I. Scope and Applicability

1.1 In Scope

This standard covers the evaluation of:

- Public Key Infrastructure (PKI) and Certificate Authority (CA) hierarchies
- Certificate lifecycle automation (ACME, cert-manager, SPIFFE SVIDs)
- Hardware Security Modules (HSM) and TPM integration
- Attribute-Based Access Control (ABAC) and Policy-as-Code (Open Policy Agent, Cedar)
- The transition from Role-Based Access Control (RBAC) to ABAC
- mTLS (Mutual TLS) enforcement and short-lived identity documents

1.2 Out of Scope

- Human authentication mechanisms like Passkeys/FIDO2 (covered in MYTHOS-ID-0001)
- Network-level Zero Trust architecture (covered in MYTHOS-NET-0004)
- Post-Quantum Cryptography algorithms (covered in MYTHOS-SEC-0001)

1.3 Audience

- Security architects and PKI engineers
- Platform engineers building zero-trust service meshes
- Application architects decoupling authorization logic from microservices
- Compliance teams managing access governance
- Standards bodies seeking a reference PKI and ABAC methodology

Part II. Definitions and Taxonomy

2.1 PKI & Authorization Dimensions

Dimension	Definition
RBAC (Role-Based Access Control)	An access control model where permissions are assigned to roles, and users are assigned to roles. Leads to role explosion and excessive standing privilege.
ABAC (Attribute-Based Access Control)	An access control model where authorization decisions are evaluated dynamically based on attributes of the user, the resource, the action, and the environment (e.g., time, location).
Policy-as-Code	The practice of writing authorization rules in a domain-specific language (e.g., Rego, Cedar), version-controlling them in Git, and enforcing them via a decoupled policy engine.
ACME (Automated Certificate Management Environment)	A protocol (RFC 8555) for automating the validation, issuance, and renewal of X.509 certificates between a client and a Certificate Authority.
HSM (Hardware Security Module)	A physical computing device that safeguards and manages digital keys for strong authentication and provides cryptoprocessing, ensuring private keys are non-exportable.

2.2 The PKI & ABAC Doctrine

> A static role is a standing vulnerability. A manual certificate is an outage. A private key in a file is a breach. If the policy is not code, the authorization is an illusion.

Part III. Evaluation Methodology

3.1 Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; manual PKI, RBAC only, hardcoded auth logic, software CAs
1	Basic RBAC with automated cert renewal, but no ABAC or HSMs
2	Functional ACME and PKI, but policy logic remains in application code
3	Solid production performance; ABAC via OPA, ACME automated, HSM-backed CAs
4	Strong performance; ephemeral mTLS, context-aware ABAC, GitOps policy deployment
5	Best-in-class; sets the standard for mathematically verified, zero-standing authorization

Weighting

Category	Weight	Rationale
Certificate Lifecycle Automation	20%	Manual cert management causes cascading outages
Hardware-Backed Key Security	15%	Software-based CAs are trivially compromised
RBAC to ABAC Transition	20%	RBAC cannot handle dynamic, zero-trust environments
Policy-as-Code Decoupling	20%	Hardcoded authorization logic is unscalable and insecure
Contextual Telemetry & Enforcement	15%	ABAC requires real-time environmental data (posture, time, location)
Audit & Decision Provenance	10%	Every authorization decision must be cryptographically provable

Total: 100%

A system MUST score at least 3.0 in Certificate Automation and ABAC/Policy-as-Code to be considered for production use.

Part IV. Evaluation Categories

4.1 Certificate Lifecycle Automation (Weight: 20%)

4.1.1 PKI Operations

How are X.509 certificates issued, renewed, and revoked?

Score	Criteria
0	Manual CSRs; certificates last years; tracked in spreadsheets
1	Internal CA exists, but clients must manually request and install certs
2	cert-manager used in Kubernetes, but other infrastructure is manual
3	Full ACME integration across all workloads (on-prem, cloud, edge); certificates last < 90 days
4	Ephemeral identity (SPIFFE SVIDs) with cert TTLs < 1 hour; zero manual rotation
5	Perfect automation: formally verified lifecycle bounds, zero certificate-induced outages, mathematical proof of continuous validity

4.2 Hardware-Backed Key Security (Weight: 15%)

4.2.1 Root of Trust

Are Certificate Authority private keys protected against exfiltration?

Score	Criteria
0	Root CA private key stored on a standard VM or in plaintext
1	Intermediate CA keys password-protected on disk
2	Cloud-managed HSM (AWS CloudHSM, Azure Key Vault Premium) used for CAs
3	Dedicated, on-premises FIPS 140-2 Level 3+ HSMs; keys are non-exportable
4	TPM 2.0 attestation required for all leaf certificates; keys generated in silicon
5	Perfect security: formally verified hardware boundaries, zero ability to extract CA keys, cryptographic proof of key provenance

4.3 RBAC to ABAC Transition (Weight: 20%)

4.3.1 Authorization Model

Are access decisions static (based on groups) or dynamic (based on attributes)?

Score	Criteria
0	Pure RBAC; users assigned to groups ("Admins", "Users") with broad permissions
1	RBAC with some coarse context (e.g., IP-based restrictions)
2	Hybrid model: RBAC for broad access, ABAC for sensitive resources
3	Pure ABAC: roles abolished; access evaluated based on user attributes, resource tags, and environment
4	Real-time attribute evaluation: access changes dynamically if device posture degrades or geolocation shifts

Score	Criteria
5	Perfect model: formally verified policy bounds, zero standing privilege, mathematical proof of least-privilege enforcement

4.4 Policy-as-Code Decoupling (Weight: 20%)

4.4.1 Enforcement Architecture

Is authorization logic decoupled from the application codebase?

Score	Criteria
0	Authorization logic hardcoded in application controllers (e.g., `if user.role == 'admin'`)
1	Centralized IAM, but application must query it and enforce locally
2	Policy engine (OPA/Cedar) deployed, but policies managed via UI/API
3	Policy-as-Code: policies written in Rego/Cedar, version-controlled in Git, deployed via CI/CD
4	Sidecar enforcement: every microservice delegates authz to a local OPA sidecar via mTLS
5	Perfect decoupling: formally verified policy execution, zero hardcoded authz logic, mathematical proof of policy completeness

4.5 Contextual Telemetry and Enforcement (Weight: 15%)

4.5.1 Data Inputs

Does the policy engine have access to real-time, dynamic attributes for decision-making?

Score	Criteria
0	Policy engine only receives user ID and requested action
1	Basic environment context (time of day) included
2	Device posture (EDR status, OS version) fed into policy engine
3	Real-time threat intelligence (e.g., impossible travel anomalies) influences decisions
4	Resource-state awareness: policy evaluates the state of the target resource (e.g., "is this document already classified as restricted?") before authorizing
5	Perfect context: formally verified data pipelines, zero stale attributes, mathematical proof of decision accuracy

4.6 Audit and Decision Provenance (Weight: 10%)

4.6.1 Compliance Evidence

Can the organization prove exactly why an access decision was made at any point in the past?

Score	Criteria
0	Logs only record "Access Granted/Denied"
1	Logs record the user and resource
2	Logs record the RBAC role used
3	ABAC decision logs: records the exact policy version, input attributes, and evaluation path
4	Cryptographic signing of authorization decisions, binding them to tamper-evident ledgers
5	Perfect provenance: formally verified audit trails, zero ability to alter decision history, cryptographic proof of compliance

Part V. The Mythos PKI & ABAC Suite (MPAS)

Traditional IAM benchmarks measure authentication latency. The MPAS evaluates certificate rotation resilience, policy engine performance, and ABAC decision accuracy.

5.1 Suite Composition

5.1.1 MPAS-PKI

• CA Compromise Simulation: 100+ tests attempting to extract the CA private key from the HSM/TPM, verifying the hardware boundary holds.

• Rotation Stress Test: 50+ tests forcing mass certificate expiration across 10,000 microservices, verifying ACME/SPIFFE rotates them without dropping connections.

5.1.2 MPAS-ABAC

• Context Shift Test: 100+ tests changing a user's device posture (e.g., disabling EDR) mid-session, verifying the ABAC engine instantly revokes access on the next request.

• Policy Bypass Test: 50+ tests attempting to bypass the OPA sidecar by calling the microservice directly, verifying mTLS enforcement blocks the unauthenticated request.

5.2 Execution Protocol

```
1. Deploy PKI infrastructure and OPA/Cedar policy engine
2. Execute complex workflows requiring dynamic attribute evaluation
3. Run MPAS suite (automated CA attack + context shift injection)
4. Score each category 0-5
5. Check for manual certificate tracking or hardcoded RBAC logic (instant disqualification)
6. If all thresholds met: approve for production
```

Part VI. Security Threat Model for PKI & ABAC

6.1 Threat Catalog

6.1.1 Certificate Authority Compromise

Severity: Critical Description: An attacker gains access to the CA's private key. They can now mint valid certificates for any service, establishing persistent, undetectable access to the entire zero-trust network.

Required Controls:

1. CA private keys MUST be generated and stored in FIPS 140-2 Level 3+ HSMs.

2. Keys MUST be non-exportable.

6.1.2 Role Explosion (RBAC Privilege Creep)

Severity: High Description: Over time, an accumulates dozens of roles to perform specific tasks. The roles grant broad access. An attacker compromising the user account gains excessive lateral movement capabilities.

Required Controls:

1. Abolish RBAC in favor of ABAC.

2. Evaluate access dynamically based on specific attributes, not broad roles.

6.1.3 Stale Policy Enforcement

Severity: High Description: A developer updates the OPA policy in Git, but the CI/CD pipeline fails to deploy it to the sidecars. The microservice enforces outdated, vulnerable authorization logic.

Required Controls:

1. Policy engines MUST report their running policy version/hash.

2. Control plane MUST detect version drift and quarantine non-compliant services.

6.1.4 Certificate Expiry Outage

Severity: High Description: An internal CA issues 1-year certificates. A critical API gateway's certificate expires over the weekend because no one monitored it. All API traffic fails.

Required Controls:

1. Mandatory ACME automation for all certificates.

2. Short certificate lifespans (< 90 days) to force automation.

Part VII. The Mythos PKI & ABAC Standard

7.1 The Mythos PKI & ABAC Doctrine

```
A static role is a standing vulnerability.
A manual certificate is an outage.

A private key in a file is a breach.
If the policy is not code, the authorization is an illusion.

Access may be requested.
Dynamic authorization must be absolute.
```

7.2 Selection Rules

1. No PKI or ABAC architecture enters production without passing MPAS.
2. No architecture is approved if it relies on manual certificate management.
3. No CA is approved if its private keys are not hardware-bound (HSM/TPM).
4. No architecture is approved if it relies solely on static RBAC roles.
5. No architecture is approved if authorization logic is hardcoded in the application.

7.3 The Prime Directive for PKI & ABAC Architecture

> Automate the ACME, do not track the spreadsheet. > Bind the key to the silicon, do not store the PEM. > Evaluate the attribute, do not trust the role. > Decouple the policy, do not hardcode the logic.

```
Academic benchmarks measure OPA evaluation latency.
Applied benchmarks measure certificate rotation resilience.
Security benchmarks measure HSM non-extractability.
Operational benchmarks measure ABAC context shift response.

> Certificates may be ephemeral.
> Authorization must be absolute.
```

End of Standard MYTHOS-ID-0002

© 2026 Mythos Systems. This source draft is retained for lineage and review. Attribution required.

End of controlled publication

MYTHOS-ID-0002 remains a Candidate Standard until technical review, security and privacy review, accessibility review, current-source validation, and formal approval are recorded.