ENGINEERING STANDARDS LIBRARY / RELIABILITY ENGINEERING

FinOps, Token Economics, and Energy Efficiency Standard

Unit economics, token and compute cost, capacity efficiency, carbon-aware operation, and value-based engineering governance.

PERMANENT DOCUMENT ID

MYTHOS-SRE-0003

PUBLICATION

Candidate Standard
Version 1.0.0-candidate

ASSURANCE PROFILE

Standard / Advanced
High Assurance

PERMANENT PUBLICATION IDENTITY

FinOps, Token Economics, and Energy Efficiency Standard

DOCUMENT ID
MYTHOS-SRE-0003

VERSION
1.0.0-candidate

STATUS
Candidate Standard

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-24

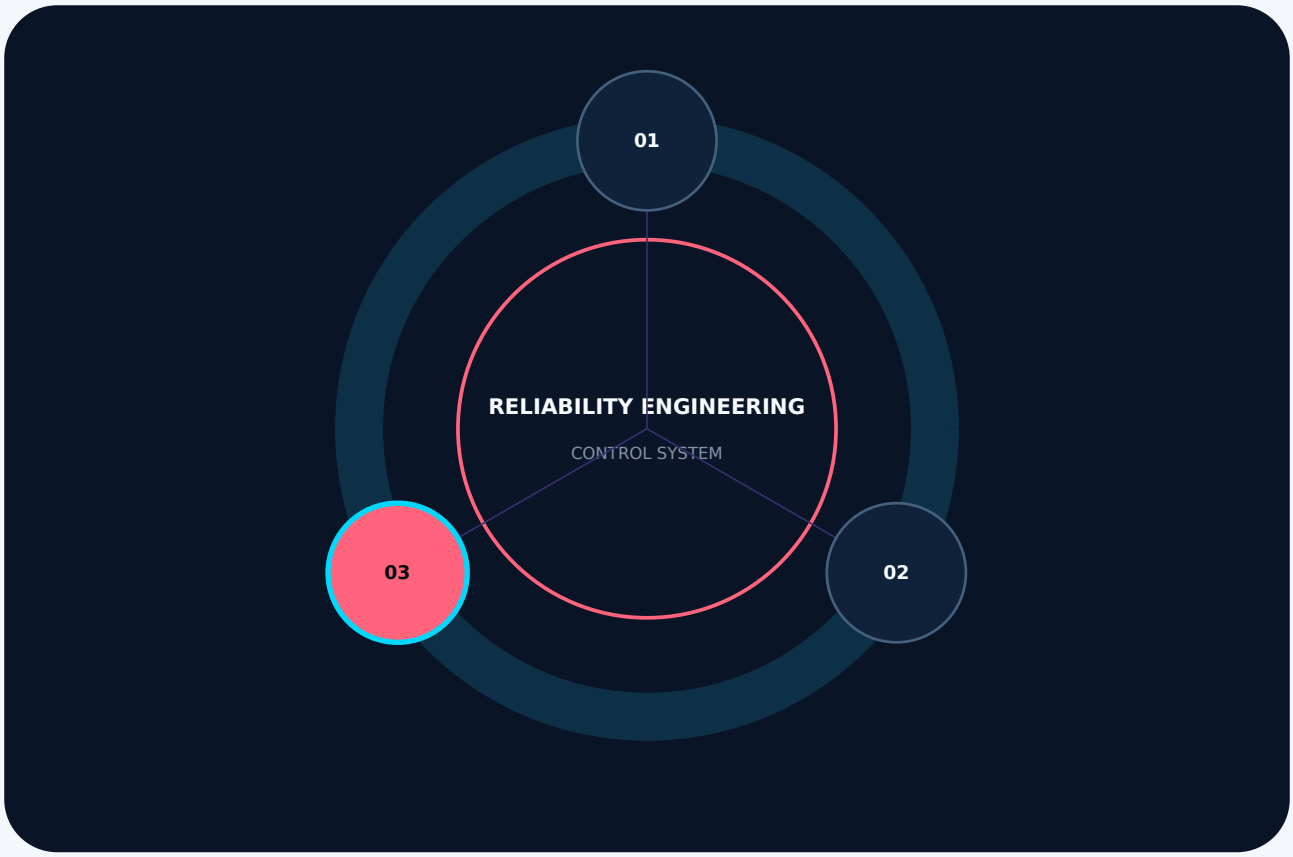
SCHEDULED REVIEW
2027-01-24

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-SRE-0003 inside the reliability engineering standard constellation



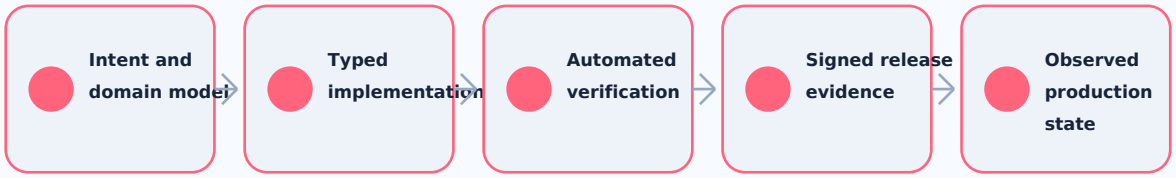
Connected assurance	Architecture, implementation, operations, evidence, and lifecycle are evaluated as one controlled system.
Specialization rule	Companion standards may add precision, but they cannot silently weaken this publication's requirements.

EXECUTIVE POSITION

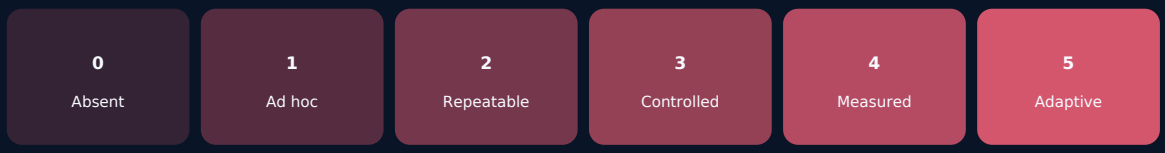
FinOps, Token Economics, and Energy Efficiency Standard

The architecture of cloud economics and energy management has failed.

Evidence-bearing delivery path



Assurance maturity spectrum



CONTENTS

Publication navigation

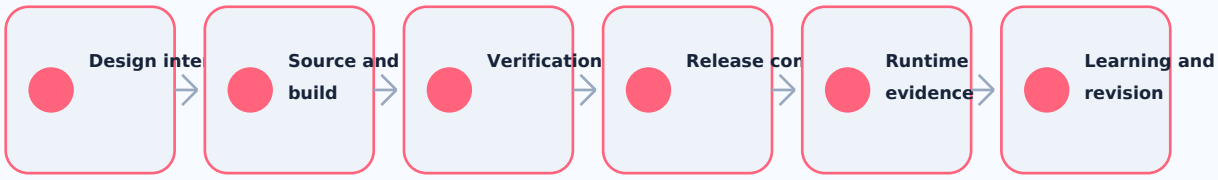
CONTROL TOPOLOGY

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
Cost Attribution and Tagging (Weight: 15%)	1
AI Token Budgets and Circuit Breakers (Weight: 20%)	1
Scale-to-Zero and Utilization (Weight: 20%)	1
Energy Efficiency (SCI/PUE) (Weight: 15%)	1
Carbon-Aware Workload Routing (Weight: 15%)	1
Automated Cost Controls (Policy-as-Code) (Weight: 15%)	1
Suite Composition	2
Additional Evaluation Dimension	4

Lifecycle control loop



EVALUATION INDEX

Criteria and evidence map

#	Capability family	Criterion
1	Cost Attribution and Tagging (Weight: 15%)	Financial Observability
2	AI Token Budgets and Circuit Breakers (Weight: 20%)	Agent Economic Control
3	Scale-to-Zero and Utilization (Weight: 20%)	Idle Compute Elimination
4	Energy Efficiency (SCI/PUE) (Weight: 15%)	Sustainability Metrics
5	Carbon-Aware Workload Routing (Weight: 15%)	Green Compute Placement
6	Automated Cost Controls (Policy-as-Code) (Weight: 15%)	Guardrail Enforcement
7	Suite Composition	MFEES-Token
8	Suite Composition	MFEES-Energy
9	Additional Evaluation Dimension	Economics & Energy Dimensions
10	Additional Evaluation Dimension	The FinOps & Energy 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 cloud economics and energy management has failed.

Organizations migrate to the cloud and deploy autonomous AI agents under the assumption that compute is infinite and cheap. They allow an agent to enter a recursive loop, burning $50,000 in LLM API tokens in a weekend. They leave hundreds of idle GPU instances running 24/7 because scaling policies are misconfigured. They measure success by application latency, completely ignoring the massive carbon footprint and wasted capital of inefficient code.

This standard exists because:

1. A Cloud Bill is a Lagging Indicator of Architectural Failure: If you discover you spent $100,000 on untagged, untracked EC2 instances at the end of the month, the damage is done. Cost tracking MUST be real-time, attributed to specific domains/agents, and actionable.
2. AI Tokens are a Finite Economic Resource: LLM inference is not a flat-rate utility; it is a highly variable, token-based economy. An autonomous agent without a hard token budget is a financial liability. Systems MUST enforce per-task and per-agent token ceilings that automatically suspend execution when exceeded.
3. Idle Compute is Environmental Malpractice: Running a datacenter at 20% utilization wastes 80% of the power and cooling. Systems MUST implement aggressive scale-to-zero architectures and serverless paradigms for non-continuous workloads.
4. Efficiency Must Be Measured in Carbon, Not Just Dollars: As AI compute scales, its energy footprint becomes a critical operational and regulatory constraint. Systems MUST calculate Software Carbon Intensity (SCI) and prioritize workload placement in low-carbon-intensity regions.

This document establishes the Mythos FinOps & Energy Standard (MFEES), a comprehensive, evidence-based methodology for evaluating financial operations against real-time cost enforcement, token economics, and green software engineering.

Scope and Applicability

In Scope

This standard covers the evaluation of:

1. FinOps platforms and cost attribution (CloudHealth, Apptio, native cloud billing)
1. AI token economics, budgeting, and circuit breakers

- Green Software Foundation metrics (Software Carbon Intensity - SCI)
- Datacenter PUE (Power Usage Effectiveness) and DVFS (Dynamic Voltage/Frequency Scaling)
- Scale-to-zero architectures and serverless cost optimization
- Carbon-aware workload routing and placement

Out of Scope

- General SLO and error budget tracking (covered in MYTHOS-SRE-0001)
- General GPU hardware topologies (covered in MYTHOS-HW-0001)
- General inference serving optimization (covered in MYTHOS-AI-0014)

Audience

- FinOps practitioners and cloud financial analysts
- Platform engineers building autoscaling and scale-to-zero pipelines
- AI engineers building token-budgeted agent loops
- SREs managing datacenter power and cooling envelopes
- Sustainability officers tracking carbon footprints
- Standards bodies seeking a reference FinOps methodology

Definitions and Taxonomy

Economics & Energy Dimensions

Dimension	Definition
FinOps	The operational framework and cultural practice of maximizing business value in the cloud by bringing financial accountability to variable spend.
Token Economics	The tracking, budgeting, and enforcement of LLM inference costs, calculated by multiplying input/output tokens by the provider's per-token pricing model.
Software Carbon Intensity (SCI)	A metric defined by the Green Software Foundation representing the carbon emissions per functional unit of software (e.g., gCO2e per user request).
Scale-to-Zero	An architectural paradigm where compute resources provision dynamically upon request and deallocate entirely to zero instances during periods of inactivity.
Carbon-Aware Routing	The practice of dynamically shifting compute workloads to datacenter regions where the local power grid is currently powered by renewable (low-carbon) energy.

The FinOps & Energy Doctrine

> A cloud bill is a lagging indicator of failure. An AI agent without a token budget is a financial liability. An idle server is environmental malpractice. If the architecture is not carbon-aware, it is unsustainable.

Evaluation Methodology

Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; untagged resources, monthly bill shock, idle compute, no token limits
1	Basic cost alerts, but manual tagging and no token budgets
2	Functional FinOps tracking, but lacks real-time enforcement or energy tracking
3	Solid production performance; Real-time domain attribution, hard token budgets, scale-to-zero, SCI tracking
4	Strong performance; Carbon-aware routing, AI-driven cost anomaly detection, formally verified budgets
5	Best-in-class; sets the standard for mathematically verified, zero-waste compute economics

Weighting

Category	Weight	Rationale
Cost Attribution & Tagging	15%	Untagged resources cannot be optimized or charged back
AI Token Budgets & Circuit Breakers	20%	Unbudgeted agents guarantee catastrophic financial surprises
Scale-to-Zero & Utilization	20%	Idle compute wastes capital and energy
Energy Efficiency (SCI/PUE)	15%	Datacenter power consumption is a hard physical limit
Carbon-Aware Workload Routing	15%	AI compute must follow renewable energy availability
Automated Cost Controls (Policy-as-Code)	15%	Manual cost optimization is reactionary and too slow

Total: 100%

A system MUST score at least 3.0 in AI Token Budgets and Scale-to-Zero to be considered for production use.

Evaluation Categories

Cost Attribution and Tagging (Weight: 15%)

Financial Observability

Can the organization attribute every dollar spent to a specific domain, team, or agent?

Score	Criteria
0	Single billing account; no tags; "shared infrastructure" black box

Score	Criteria
1	Manual tagging enforced via policy, but compliance is < 80%
2	Automated tagging on all resources, but cost allocation is delayed (monthly)
3	Real-time cost attribution mapped to DDD bounded contexts; per-domain chargeback models
4	Per-request/per-agent cost tracking (e.g., tracing the exact cloud cost of a single user transaction)
5	Perfect attribution: formally verified cost provenance, zero untagged resources, mathematical proof of domain-level financial parity

AI Token Budgets and Circuit Breakers (Weight: 20%)

Agent Economic Control

How is the cost of autonomous AI agents bounded?

Score	Criteria
0	No token tracking; agents run until they finish or crash, regardless of cost
1	Total token count logged per request, but no cost calculation
2	Cost calculated per request, but no hard budgets enforced
3	Hard token/cost budgets enforced per agent task; tasks are automatically suspended if budgets are exceeded
4	Predictive token budgeting: system estimates task cost before execution and asks for HITL approval if > $X
5	Perfect control: formally verified economic bounds, zero ability for runaway agents to cause bill shock, mathematical proof of cost efficiency

Scale-to-Zero and Utilization (Weight: 20%)

Idle Compute Elimination

How does the system handle periods of low or zero traffic?

Score	Criteria
0	Instances run 24/7/365 regardless of traffic; average utilization < 15%
1	Basic autoscaling, but minimum instance count is > 0
2	Scale-to-zero implemented for dev/staging, but production runs idle instances
3	Aggressive scale-to-zero for all non-critical paths; serverless (FaaS) utilized for bursty workloads
4	GPU scale-to-zero implemented with cold-start mitigation (e.g., keeping model weights in memory while suspending compute)
5	Perfect utilization: formally verified zero-idle waste, mathematical proof of optimal provision-to-request latency ratios

Energy Efficiency (SCI/PUE) (Weight: 15%)

Sustainability Metrics

Is the energy consumption and carbon footprint of software actively measured?

Score	Criteria
0	No energy or carbon tracking; PUE unknown
1	Datacenter PUE tracked, but no software-level metrics
2	Cloud provider carbon dashboards reviewed monthly
3	Software Carbon Intensity (SCI) score calculated per major workflow; DVFS enabled on hardware
4	Real-time SCI tracking integrated into observability dashboards alongside latency and cost
5	Perfect measurement: formally verified energy bounds, mathematical proof of minimal carbon per functional unit

Carbon-Aware Workload Routing (Weight: 15%)

Green Compute Placement

Does the system shift compute based on renewable energy availability?

Score	Criteria
0	Workloads run in a single region regardless of carbon intensity
1	Carbon intensity awareness exists, but requires manual routing
2	Batch jobs (e.g., AI training) scheduled to run during off-peak energy hours
3	Carbon-aware routing: batch workloads dynamically routed to regions with real-time surplus renewable energy (e.g., wind/solar peaks)
4	Real-time inference routing adjusted based on grid carbon intensity without violating SLOs
5	Perfect routing: formally verified carbon optimization, zero fossil-fuel-bound compute where renewables are available, mathematical proof of minimal carbon footprint

Automated Cost Controls (Policy-as-Code) (Weight: 15%)

Guardrail Enforcement

are financial guardrails enforced in the CI/CD pipeline?

Score	Criteria
0	No automated controls; engineers can provision $10,000/day GPU instances without approval
1	Basic cloud budget alerts sent to Slack
2	Hard spending limits configured in the cloud account, but blunt (blocks all provisioning)
3	Policy-as-Code (OPA/Sentinel) in CI/CD blocking the deployment of expensive instance types without peer review

Score	Criteria
4	Real-time anomaly detection: system automatically suspends workloads exhibiting cost spikes indicative of infinite loops
5	Perfect controls: formally verified financial guardrails, zero ability to bypass budget policies, mathematical proof of spend compliance

The Mythos FinOps & Energy Suite (MFEES)

Traditional FinOps reviews measure monthly bill reductions. The MFEES evaluates real-time token enforcement, scale-to-zero resilience, and carbon-aware routing.

Suite Composition

MFEES-Token

1. Runaway Agent Test: 100+ tests simulating an AI agent stuck in an infinite tool-calling loop, verifying the token circuit breaker suspends the task before exceeding the $10 budget.
1. Cost Attribution Test: 50+ tests verifying that the exact compute and token costs of a specific API request are tagged and attributable to the triggering user/domain.

MFEES-Energy

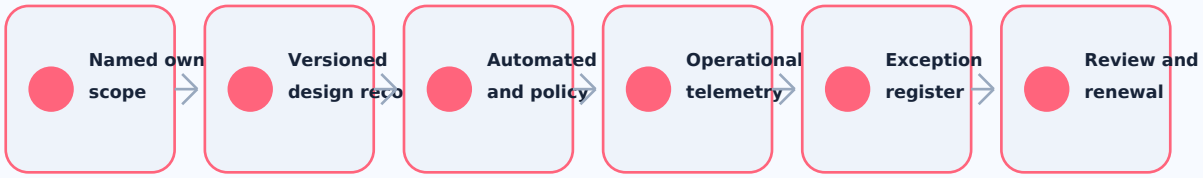
1. Scale-to-Zero Test: 100+ tests simulating 1 hour of zero traffic, verifying all non-critical compute instances deallocate to zero and stop accruing costs.
1. Carbon Routing Test: 50+ tests simulating a surge in renewable energy in Region B, verifying batch training workloads migrate from Region A to Region B automatically.

Execution Protocol

ASSURANCE AND ADOPTION

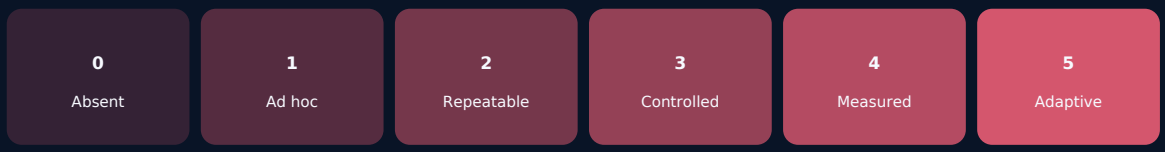
Implementation evidence and review gates

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
TLA+ Foundation: TLA+ Language and Tools Documentation	Rolling documentation snapshot	High	2027-01-24
Kubernetes Project: Kubernetes Documentation	v1.36 current documentation	High	2026-10-22
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
FinOps Foundation: FinOps Framework and FOCUS Specification	Rolling official documentation snapshot	High	2026-10-22

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.