MYTHOS-UX-0004

Spatial Computing and WebGPU/wgpu Integration Standard

Defines architecture, rendering, latency, persistence, input, safety, accessibility, data sovereignty, and assurance requirements for spatial computing, XR, volumetric engineering views, WebGPU, and native wgpu applications.

6	5	UX
EVALUATION CRITERIA	CURRENT AUTHORITY RECORDS	CANONICAL SERIES

CANDIDATE STANDARD

Mythos Systems | Engineering Standards Program | 2026-07-22

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

Contents

Use the interactive entries below or the PDF bookmark outline to navigate the publication.

Document Control

Field	Value
Document ID	MYTHOS-UX-0004
Version	1.1.0 - Enterprise Candidate Edition
Status	Candidate Standard
Classification	Public
Publisher	Mythos Systems
Owner	Aaron Stovall
Domain	User Experience and Human-System Interaction
Initial source date	2026-07-16
Enterprise reconstruction	2026-07-22
Review cadence	Quarterly for dynamic authorities; at least annually for stable controls
Supersedes	Source draft retained in Appendix A
Publication lineage	Permanent canonical series

CANDIDATE STATUS

The source document identified itself as a definitive published standard. This enterprise edition corrects that status. It remains a candidate until current-source validation, qualified domain review, assessment engineering, accessibility review, and formal approval are recorded.

Part I - Executive Direction

1. Executive Overview

Defines architecture, rendering, latency, persistence, input, safety, accessibility, data sovereignty, and assurance requirements for spatial computing, XR, volumetric engineering views, WebGPU, and native wgpu applications.

2. Principal Findings

- Spatial interfaces are safety-sensitive real-time systems, not flat screens positioned in three dimensions.
- Rendering fidelity, motion-to-photon latency, tracking quality, and frame stability must be measured on target hardware.
- Physical-space mapping and anchors are sensitive data requiring explicit retention, consent, and egress controls.
- WebGPU, wgpu, WebXR, and OpenXR capabilities must be selected through compatibility and evidence, not novelty.

3. Required Leadership Decisions

- Define supported devices, runtimes, frame-rate targets, latency budgets, and fallback modes.
- Require target-hardware performance, comfort, accessibility, and safety testing.
- Classify spatial maps, gaze, hand, voice, and environmental sensor data.
- Adopt a typed spatial scene and interaction model with deterministic state ownership.

4. Enterprise Target State

TARGET OPERATING MODEL

Organizations applying MYTHOS-UX-0004 operate the subject as a governed lifecycle: inventory and classify; establish authoritative policy and architecture; enforce controls technically; observe actual state; verify outcomes independently; retain evidence; manage exceptions; and revalidate when authorities, platforms, risks, or operating conditions change.

5. Business, Safety, and Operational Impact

- Reduces reliance on manual evidence and undocumented expert knowledge.
- Makes risk acceptance, exceptions, and unresolved uncertainty visible to accountable leaders.
- Creates measurable implementation and assurance outcomes rather than a one-time score.
- Supports procurement, architecture review, operational readiness, and independent assessment from one source-controlled publication.

Part II - Standard Foundation

6. Purpose and Scope

Defines architecture, rendering, latency, persistence, input, safety, accessibility, data sovereignty, and assurance requirements for spatial computing, XR, volumetric engineering views, WebGPU, and native wgpu applications.

6.1 Applicability

Apply this standard according to system purpose, deployment model, data classification, user population, safety impact, regulatory obligations, and organizational risk tolerance. Tailoring must be documented. A lower implementation profile cannot silently waive a mandatory legal, safety, accessibility, contractual, or security obligation.

6.2 Out of Scope

This publication does not replace legal advice, contracting interpretation, regulator guidance, platform-specific engineering, human-factors testing, safety certification, or implementation evidence. Companion Mythos standards remain applicable where their domains intersect.

7. Requirements Language and Conformance

The terms MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY are normative only when written in uppercase. Conformance requires applicable mandatory criteria to be implemented and supported by current evidence. A weighted score is informative and cannot compensate for a failed mandatory gate.

State	Meaning
Conformant	All applicable mandatory requirements are satisfied with sufficient current evidence.
Partially Conformant	Some applicable requirements are implemented, but material gaps remain.
Not Conformant	One or more mandatory requirements are not satisfied.
Compensating Control Accepted	A formally approved alternative achieves the required outcome.
Exception Active	A time-bounded, accountable risk acceptance exists.
Not Assessed	Evidence is insufficient to reach a conclusion.

8. Assurance and Maturity

Level	Description
M0 Unmanaged	The outcome is not intentionally controlled.
M1 Documented	Responsibilities and procedures exist but are inconsistent or manual.
M2 Implemented	The control operates in the applicable environment.
M3 Measured	Performance, exceptions, and effectiveness are measured.
M4 Assured	Independent testing and continuous evidence support the control.
M5 Adaptive	The control responds safely to changes in risk, environment, and evidence.

Part III - Risk and Architecture

9. Risk Model

- Authority drift: external requirements, specifications, programs, and platform behavior change faster than the publication is reviewed.
- Control illusion: documentation or visual state indicates compliance while runtime behavior differs.
- Automation overreach: a generated plan or score is treated as authorization or proof.
- Evidence gaps: conclusions cannot be reconstructed from source, configuration, event, test, and approval records.
- Human factors: users cannot understand, interrupt, challenge, or safely recover from consequential behavior.
- Dependency concentration: a provider, runtime, device, framework, or external authority becomes a hidden single point of failure.

10. Reference Control Architecture

ARCHITECTURE SEPARATION

Source authority defines applicable obligations and constraints. Human and organizational governance establishes accountability. Typed policy and architecture encode the target state. Runtime enforcement controls behavior. Independent testing verifies outcomes. Evidence systems preserve what was proposed, approved, executed, observed, and verified.

11. Roles and Accountability

Role	Accountability
Executive sponsor	Approves risk posture, funding, and unresolved material exceptions.
Domain authority	Owns interpretation, architecture, and control applicability.
Engineering owner	Implements controls and operational lifecycle.
Security/privacy/accessibility/safety reviewer	Challenges design and verifies domain-specific obligations.
Independent assessor	Evaluates evidence and tests without relying only on implementer assertions.
Publication owner	Maintains version, sources, traceability, expiration, and change record.

Part IV - Normative Evaluation and Control System

12. Evaluation Doctrine

The preserved source uses a weighted 0-5 evaluation model. This edition retains that material but corrects its interpretation. Score 5 means optimized, strongly evidenced, and independently assured within the assessed scope. It never means perfect, mathematically infallible, universally compliant, or permanently secure.

Score	Enterprise interpretation
0	Absent, unsafe, or unable to perform the required outcome.
1	Initial or ad hoc capability with substantial manual dependence.
2	Repeatable partial implementation with material gaps.
3	Production baseline: implemented, governed, and evidenced.
4	Advanced: automated, measured, resilient, and independently tested.
5	Assured: optimized for the defined profile, continuously evidenced, and subject to recurring independent validation.

MANDATORY GATE

An organization cannot average away a failed mandatory criterion. Applicability, safety, legal requirements, accessibility, security boundaries, authorization, and evidence sufficiency must be evaluated before weighted scoring is used.

13. Criterion Index

ID	Criterion	Weight	Primary outcome
MYTHOS-UX-0004-C01	WebGPU/wgpu Pipeline and Compute	20%	Use a typed, resource-bounded rendering architecture with explicit device capability detection, shader validation, memory accounting, and non-WebGPU fallback where required.
MYTHOS-UX-0004-C02	Motion-to-Photon Latency	20%	Define and measure latency and frame-stability targets per device, interaction, and comfort profile rather than rely on a universal marketing threshold.
MYTHOS-UX-0004-C03	Spatial Persistence and Anchoring	15%	Represent anchors, coordinate systems, ownership, drift, sharing, retention, privacy, and revocation as typed state with source and confidence.
MYTHOS-UX-0004-C04	Foveated and Edge Rendering	15%	Use foveated, split, or edge rendering only with measured quality, latency, privacy, network, and failure tradeoffs and a safe local fallback.
MYTHOS-UX-0004-C05	Volumetric Data Semantics	15%	Define typed spatial entities, scale, uncertainty, occlusion, selection, labels, provenance, and accessible two-dimensional alternatives.
MYTHOS-UX-0004-C06	Sensor Fusion and Input Tracking	15%	Fuse head, hand, eye, voice, controller, environmental, and external sensor inputs through calibrated confidence, time synchronization, privacy, and fault handling.

MYTHOS-UX-0004-C01 - WebGPU/wgpu Pipeline and Compute

NORMATIVE REQUIREMENT

The organization MUST use a typed, resource-bounded rendering architecture with explicit device capability detection, shader validation, memory accounting, and non-WebGPU fallback where required.

Objective and Risk Addressed

Use a typed, resource-bounded rendering architecture with explicit device capability detection, shader validation, memory accounting, and non-WebGPU fallback where required.

Applicability

Apply this criterion wherever the evaluated capability, data, workflow, interface, user, environment, provider, or authority is material. Document exclusions and any compensating control.

Minimum Acceptable Implementation

A production baseline corresponds to source score 3: the capability is deliberately implemented, technically enforced where practical, owned, monitored, and supported by current evidence. Source scores 4 and 5 represent increasing automation, resilience, continuous assurance, and independent validation.

Implementation Direction

- Define authoritative state, owner, interfaces, dependencies, lifecycle, and failure behavior.
- Encode enforceable policy and typed contracts instead of relying only on prose or visual convention.
- Provide safe fallback, interruption, exception, recovery, and decommissioning behavior.
- Collect evidence at the point of action and verify the observed result independently.
- Revalidate after material source, platform, model, device, provider, architecture, or risk change.

Required Evidence

- render architecture
- feature matrix
- shader validation
- memory budgets
- fallback tests
- dependency versions.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Run target scenes across supported adapters and browsers/runtimes; validate device loss, shader failure, memory pressure, and fallback.
- Inspect failures, exceptions, and negative tests rather than reviewing only successful examples.
- Rate evidence confidence and record untested conditions, unsupported platforms, and unresolved uncertainty.

Metrics and Assurance Indicators

Metric	Expected use
frame time	Track trend, threshold, exception, and accountable response.
GPU memory	Track trend, threshold, exception, and accountable response.
device-loss recovery	Track trend, threshold, exception, and accountable response.
unsupported-device completion	Track trend, threshold, exception, and accountable response.

Preserved Source Scoring Anchors

The following anchors are retained from the source draft. Absolute words such as “perfect” are historical wording and are normalized by the enterprise interpretation above.

Score	Preserved source criterion
0	WebGL 1.0 / 2.0 (state-machine, high overhead)
1	WebGPU via JavaScript, but heavily bound by GC pauses

Score	Preserved source criterion
2	Native WebGPU implementation, but single-threaded command buffer generation
3	wgpu (Rust) or native WebGPU with multi-threaded command buffer encoding
4	Advanced compute shaders for particle systems, spatial hashing, and culling
5	Perfect pipeline: formally verified memory safety, zero-allocation render loops, mathematically proven frame budget adherence

Exceptions and Compensating Controls

Any exception must identify the unmet outcome, affected scope, owner, risk, compensating control, evidence, expiration, and remediation plan. Exceptions do not change the underlying requirement.

MYTHOS-UX-0004-C02 - Motion-to-Photon Latency

NORMATIVE REQUIREMENT

The organization MUST define and measure latency and frame-stability targets per device, interaction, and comfort profile rather than rely on a universal marketing threshold.

Objective and Risk Addressed

Define and measure latency and frame-stability targets per device, interaction, and comfort profile rather than rely on a universal marketing threshold.

Applicability

Apply this criterion wherever the evaluated capability, data, workflow, interface, user, environment, provider, or authority is material. Document exclusions and any compensating control.

Minimum Acceptable Implementation

A production baseline corresponds to source score 3: the capability is deliberately implemented, technically enforced where practical, owned, monitored, and supported by current evidence. Source scores 4 and 5 represent increasing automation, resilience, continuous assurance, and independent validation.

Implementation Direction

- Define authoritative state, owner, interfaces, dependencies, lifecycle, and failure behavior.
- Encode enforceable policy and typed contracts instead of relying only on prose or visual convention.
- Provide safe fallback, interruption, exception, recovery, and decommissioning behavior.
- Collect evidence at the point of action and verify the observed result independently.
- Revalidate after material source, platform, model, device, provider, architecture, or risk change.

Required Evidence

- latency budget
- measurement method
- frame traces
- target-device results
- comfort study
- degraded-mode policy.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Measure input-to-display latency under representative load, network variation, thermal throttling, and tracking conditions.
- Inspect failures, exceptions, and negative tests rather than reviewing only successful examples.
- Rate evidence confidence and record untested conditions, unsupported platforms, and unresolved uncertainty.

Metrics and Assurance Indicators

Metric	Expected use
motion-to-photon p95/p99	Track trend, threshold, exception, and accountable response.
missed frames	Track trend, threshold, exception, and accountable response.
reprojection frequency	Track trend, threshold, exception, and accountable response.
comfort-related exit rate	Track trend, threshold, exception, and accountable response.

Preserved Source Scoring Anchors

The following anchors are retained from the source draft. Absolute words such as “perfect” are historical wording and are normalized by the enterprise interpretation above.

Score	Preserved source criterion
0	Untracked latency (>50ms); causes severe motion sickness
1	Basic IMU tracking, but frame drops cause stutter (30ms+)
2	Stable 72Hz framerate, but latency peaks during high compute (25ms)
3	Strict <20ms motion-to-photon latency enforced via hardware-level timewarp/spacewarp
4	Sub-15ms latency with predictive tracking and late-latching
5	Perfect safety: formally verified latency bounds, zero frame drops under any load, biologically guaranteed comfort

Exceptions and Compensating Controls

Any exception must identify the unmet outcome, affected scope, owner, risk, compensating control, evidence, expiration, and remediation plan. Exceptions do not change the underlying requirement.

MYTHOS-UX-0004-C03 - Spatial Persistence and Anchoring

NORMATIVE REQUIREMENT

The organization MUST represent anchors, coordinate systems, ownership, drift, sharing, retention, privacy, and revocation as typed state with source and confidence.

Objective and Risk Addressed

Represent anchors, coordinate systems, ownership, drift, sharing, retention, privacy, and revocation as typed state with source and confidence.

Applicability

Apply this criterion wherever the evaluated capability, data, workflow, interface, user, environment, provider, or authority is material. Document exclusions and any compensating control.

Minimum Acceptable Implementation

A production baseline corresponds to source score 3: the capability is deliberately implemented, technically enforced where practical, owned, monitored, and supported by current evidence. Source scores 4 and 5 represent increasing automation, resilience, continuous assurance, and independent validation.

Implementation Direction

- Define authoritative state, owner, interfaces, dependencies, lifecycle, and failure behavior.
- Encode enforceable policy and typed contracts instead of relying only on prose or visual convention.
- Provide safe fallback, interruption, exception, recovery, and decommissioning behavior.
- Collect evidence at the point of action and verify the observed result independently.
- Revalidate after material source, platform, model, device, provider, architecture, or risk change.

Required Evidence

- spatial schema
- anchor store
- retention and consent
- drift measurements
- conflict and deletion tests.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Create, relocate, share, revoke, and recover anchors across sessions, devices, environmental change, and offline conditions.
- Inspect failures, exceptions, and negative tests rather than reviewing only successful examples.
- Rate evidence confidence and record untested conditions, unsupported platforms, and unresolved uncertainty.

Metrics and Assurance Indicators

Metric	Expected use
anchor drift	Track trend, threshold, exception, and accountable response.
resolution failure	Track trend, threshold, exception, and accountable response.
stale anchors	Track trend, threshold, exception, and accountable response.
deletion completion	Track trend, threshold, exception, and accountable response.
unauthorized sharing	Track trend, threshold, exception, and accountable response.

Preserved Source Scoring Anchors

The following anchors are retained from the source draft. Absolute words such as “perfect” are historical wording and are normalized by the enterprise interpretation above.

Score	Preserved source criterion
0	Objects exist only in app-space; reset on reload
1	Local-only persistence; drifts over time
2	Basic cloud anchors (e.g., ARCore/ARKit cloud anchors)
3	Semantic spatial graph: objects bind to recognized physical surfaces (tables, walls)
4	Multi-user shared spatial state with sub-centimeter accuracy
5	Perfect persistence: formally verified cross-device synchronization, zero spatial drift, cryptographic binding of digital state to physical coordinates

Exceptions and Compensating Controls

Any exception must identify the unmet outcome, affected scope, owner, risk, compensating control, evidence, expiration, and remediation plan. Exceptions do not change the underlying requirement.

MYTHOS-UX-0004-C04 - Foveated and Edge Rendering

NORMATIVE REQUIREMENT

The organization MUST use foveated, split, or edge rendering only with measured quality, latency, privacy, network, and failure tradeoffs and a safe local fallback.

Objective and Risk Addressed

Use foveated, split, or edge rendering only with measured quality, latency, privacy, network, and failure tradeoffs and a safe local fallback.

Applicability

Apply this criterion wherever the evaluated capability, data, workflow, interface, user, environment, provider, or authority is material. Document exclusions and any compensating control.

Minimum Acceptable Implementation

A production baseline corresponds to source score 3: the capability is deliberately implemented, technically enforced where practical, owned, monitored, and supported by current evidence. Source scores 4 and 5 represent increasing automation, resilience, continuous assurance, and independent validation.

Implementation Direction

- Define authoritative state, owner, interfaces, dependencies, lifecycle, and failure behavior.
- Encode enforceable policy and typed contracts instead of relying only on prose or visual convention.
- Provide safe fallback, interruption, exception, recovery, and decommissioning behavior.
- Collect evidence at the point of action and verify the observed result independently.
- Revalidate after material source, platform, model, device, provider, architecture, or risk change.

Required Evidence

- rendering profile
- gaze-data policy
- network budget
- quality metrics
- fallback
- provider and region records.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Test gaze loss, edge outage, bandwidth reduction, privacy restrictions, and scene complexity; verify graceful degradation.
- Inspect failures, exceptions, and negative tests rather than reviewing only successful examples.
- Rate evidence confidence and record untested conditions, unsupported platforms, and unresolved uncertainty.

Metrics and Assurance Indicators

Metric	Expected use
quality score	Track trend, threshold, exception, and accountable response.
edge round-trip latency	Track trend, threshold, exception, and accountable response.
fallback time	Track trend, threshold, exception, and accountable response.
gaze-data egress	Track trend, threshold, exception, and accountable response.
visual artifact rate	Track trend, threshold, exception, and accountable response.

Preserved Source Scoring Anchors

The following anchors are retained from the source draft. Absolute words such as “perfect” are historical wording and are normalized by the enterprise interpretation above.

Score	Preserved source criterion
0	Renders full resolution everywhere (exhausts GPU)
1	Fixed foveated rendering (static center high-res)
2	Dynamic foveated rendering driven by eye-tracking
3	Edge-split rendering: heavy compute offloaded to edge cloud, lightweight render locally
4	AI-predictive foveation: anticipates saccades to pre-render high-res areas
5	Perfect optimization: formally verified compute bounds, zero perceptible quality loss, seamless edge failover

Exceptions and Compensating Controls

Any exception must identify the unmet outcome, affected scope, owner, risk, compensating control, evidence, expiration, and remediation plan. Exceptions do not change the underlying requirement.

MYTHOS-UX-0004-C05 - Volumetric Data Semantics

NORMATIVE REQUIREMENT

The organization MUST define typed spatial entities, scale, uncertainty, occlusion, selection, labels, provenance, and accessible two-dimensional alternatives.

Objective and Risk Addressed

Define typed spatial entities, scale, uncertainty, occlusion, selection, labels, provenance, and accessible two-dimensional alternatives.

Applicability

Apply this criterion wherever the evaluated capability, data, workflow, interface, user, environment, provider, or authority is material. Document exclusions and any compensating control.

Minimum Acceptable Implementation

A production baseline corresponds to source score 3: the capability is deliberately implemented, technically enforced where practical, owned, monitored, and supported by current evidence. Source scores 4 and 5 represent increasing automation, resilience, continuous assurance, and independent validation.

Implementation Direction

- Define authoritative state, owner, interfaces, dependencies, lifecycle, and failure behavior.
- Encode enforceable policy and typed contracts instead of relying only on prose or visual convention.
- Provide safe fallback, interruption, exception, recovery, and decommissioning behavior.

- Collect evidence at the point of action and verify the observed result independently.
- Revalidate after material source, platform, model, device, provider, architecture, or risk change.

Required Evidence

- scene schema
- semantic layers
- provenance
- accessibility alternative
- selection tests
- level-of-detail policy.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Render representative graphs, point clouds, twins, and uncertainty; test selection accuracy, scale comprehension, and fallback.
- Inspect failures, exceptions, and negative tests rather than reviewing only successful examples.
- Rate evidence confidence and record untested conditions, unsupported platforms, and unresolved uncertainty.

Metrics and Assurance Indicators

Metric	Expected use
selection error	Track trend, threshold, exception, and accountable response.
occlusion defects	Track trend, threshold, exception, and accountable response.
label collisions	Track trend, threshold, exception, and accountable response.
accessible-equivalent coverage	Track trend, threshold, exception, and accountable response.

Preserved Source Scoring Anchors

The following anchors are retained from the source draft. Absolute words such as “perfect” are historical wording and are normalized by the enterprise interpretation above.

Score	Preserved source criterion
0	2D React panels rendered as billboards in 3D space
1	Basic 3D bar charts floating in space
2	Volumetric point clouds (e.g., network topology, agent state) with depth occlusion
3	Spatial UI bound to physical surfaces (e.g., data pouring out of a physical server rack)
4	Interactive volumetric shaders representing real-time multi-dimensional state
5	Perfect semantics: formally verified spatial cognitive load, zero 2D bias, mathematically optimal use of depth and parallax for data parsing

Exceptions and Compensating Controls

Any exception must identify the unmet outcome, affected scope, owner, risk, compensating control, evidence, expiration, and remediation plan. Exceptions do not change the underlying requirement.

MYTHOS-UX-0004-C06 - Sensor Fusion and Input Tracking

NORMATIVE REQUIREMENT

The organization MUST fuse head, hand, eye, voice, controller, environmental, and external sensor inputs through calibrated confidence, time synchronization, privacy, and fault handling.

Objective and Risk Addressed

Fuse head, hand, eye, voice, controller, environmental, and external sensor inputs through calibrated confidence, time synchronization, privacy, and fault handling.

Applicability

Apply this criterion wherever the evaluated capability, data, workflow, interface, user, environment, provider, or authority is material. Document exclusions and any compensating control.

Minimum Acceptable Implementation

A production baseline corresponds to source score 3: the capability is deliberately implemented, technically enforced where practical, owned, monitored, and supported by current evidence. Source scores 4 and 5 represent increasing automation, resilience, continuous assurance, and independent validation.

Implementation Direction

- Define authoritative state, owner, interfaces, dependencies, lifecycle, and failure behavior.
- Encode enforceable policy and typed contracts instead of relying only on prose or visual convention.
- Provide safe fallback, interruption, exception, recovery, and decommissioning behavior.
- Collect evidence at the point of action and verify the observed result independently.
- Revalidate after material source, platform, model, device, provider, architecture, or risk change.

Required Evidence

- sensor inventory
- calibration
- time sync
- confidence model
- privacy policy
- failure tests.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Introduce sensor loss, conflict, drift, spoofing, latency, and permission revocation; verify safe interaction behavior.
- Inspect failures, exceptions, and negative tests rather than reviewing only successful examples.
- Rate evidence confidence and record untested conditions, unsupported platforms, and unresolved uncertainty.

Metrics and Assurance Indicators

Metric	Expected use
tracking loss	Track trend, threshold, exception, and accountable response.
fusion disagreement	Track trend, threshold, exception, and accountable response.
calibration age	Track trend, threshold, exception, and accountable response.
spoof detection	Track trend, threshold, exception, and accountable response.
privacy exceptions	Track trend, threshold, exception, and accountable response.

Preserved Source Scoring Anchors

The following anchors are retained from the source draft. Absolute words such as “perfect” are historical wording and are normalized by the enterprise interpretation above.

Score	Preserved source criterion
0	Standard Xbox controller mapped to 3D space
1	6-DoF controller raycasting (e.g., Meta Touch)
2	Basic hand-tracking (pinch to click)
3	High-fidelity skeletal hand-tracking with physical collision
4	Multimodal fusion: eye-tracking for focus, hands for manipulation, voice for command
5	Perfect interaction: formally verified zero-latency intent prediction, sub-millimeter tracking accuracy, zero false positives

Exceptions and Compensating Controls

Any exception must identify the unmet outcome, affected scope, owner, risk, compensating control, evidence, expiration, and remediation plan. Exceptions do not change the underlying requirement.

Part V - Implementation and Operations

14. Implementation Profiles

Profile	Required posture
Baseline	Documented ownership, minimum production controls, evidence, and exception process.
Enterprise	Central policy, automation, integration, continuous monitoring, and independent review.
High Assurance	Strong isolation, high-assurance identity, formal or adversarial verification, short evidence windows, and two-person governance where material.
Sovereign or Air-Gapped	Local control planes, approved dependencies, offline update and evidence workflows, restricted egress, and explicit supply-chain custody.

15. Operational Lifecycle

Stage	Required activities
Discover and classify	Inventory scope, authority, data, users, dependencies, risks, and current evidence.
Design	Define target state, architecture, policies, tests, metrics, and ownership.
Implement	Deploy controls through reviewed changes with rollback and evidence.
Operate	Monitor state, exceptions, incidents, drift, performance, and provider changes.
Verify	Perform recurring independent assessment and negative testing.
Change	Reassess applicability and invalidate stale evidence after material changes.
Retire	Revoke authority, remove data and credentials, preserve required evidence, and update the registry.

16. Anti-Patterns and Prohibited Practices

- Declaring conformance from a document review without implementation evidence.
- Using one weighted score to override a failed mandatory requirement.
- Treating model output, interface state, scanner output, or tool success as independent verification.
- Using stale or unversioned external authorities for current decisions.
- Hiding exceptions, unsupported platforms, incomplete testing, or low-confidence findings.
- Hard-coding a single provider, device, framework, or vendor as a universal requirement unless the publication is explicitly vendor-scoped.

Part VI - Assurance, Evidence, and Traceability

17. Evidence Model

Evidence must identify subject, scope, source, version, time, actor, method, result, integrity, retention, and relationship to the assessed criterion. Screenshots and generated summaries are supporting artifacts, not sufficient evidence by themselves.

18. Assessment Confidence

Confidence	Basis
Low	Self-assertion, incomplete samples, stale evidence, or untested material conditions.
Moderate	Current documentation and representative evidence with limited independent testing.
High	Current direct evidence and repeatable independent testing across material conditions.
Very High	Sustained evidence, broad negative testing, independent review, and verified operation over time.

19. Current Authority Register

Authority	Status and scope	Valid until
WebGPU	Candidate Recommendation Draft - Browser graphics and compute API; implementation status remains dynamic	2026-10-20
WebXR Device API	Candidate Recommendation Draft - Immersive web session and device integration	2026-10-20
OpenXR 1.1.61	Current Khronos specification release - Cross-platform native XR application interface	2027-01-18
WCAG 2.2	W3C Recommendation - Accessible alternatives, input, motion, and interface behavior	2027-01-18
ISO/IEC 40500:2025	Published international standard - Accessibility baseline for web-delivered spatial controls	2027-01-18

AUTHORITY LIMITATION

The register identifies current technical and governance authorities relevant to this candidate standard. It does not establish legal applicability, contractual compliance, certification, or clause-level equivalence. Dynamic sources must be refreshed before material decisions.

20. Publication Gate

- Qualified domain technical review completed.
- Current authorities and versions independently confirmed.
- Criterion requirements, evidence, and tests reviewed for measurability.
- Legal, contracting, regulatory, safety, privacy, and accessibility applicability reviewed where relevant.
- Machine-readable criterion catalog validated.

- PDF accessibility and visual quality reviewed.
- Formal approval, effective date, and review triggers recorded.

Appendix A - Complete Preserved Source Draft

The following source draft is preserved in full for completeness and traceability. Conversational framing, “definitive” status, universal claims, obsolete program assumptions, and “perfect” score language are historical source content and are not controlling statements in this enterprise candidate edition.

Document ID: MYTHOS-UX-0004 Version: 1.0.0 (Definitive Registry Edition) Status: Published Standard Classification: Public Organization: Mythos Systems (MYTHOS AI) Owner: Aaron Stovall Effective Date: 2026-07-16 Review Cadence: Quarterly, and immediately after a material shift in WebGPU/WGSL standards, XR runtime APIs (OpenXR), or spatial anchoring paradigms Applies To: All organizations evaluating, selecting, or operating spatial computing platforms, XR runtimes, 3D rendering engines, WebGPU/wgpu applications, and digital twin visualizers for production, agentic, or enterprise use Companion Standards: MYTHOS-UX-0001 (AI-Native Interfaces), MYTHOS-UX-0003 (Visual State Semantics), MYTHOS-SWE-0006 (Desktop Architecture), MYTHOS-DAT-0003 (Data Sovereignty)

0. Executive Mandate

The architecture of spatial computing has failed.

Organizations attempt to build immersive data environments and XR applications using legacy WebGL pipelines and JavaScript garbage-collected game loops. They treat Augmented Reality (AR) as simply rendering a 2D React component floating in mid-air. They ignore the physical safety of the user, allowing network latency to induce motion sickness, and they fail to persist digital objects in physical space across sessions.

This standard exists because:

1. WebGL is a Legacy Bottleneck: WebGL is a state machine wrapped around a 20-year-old API. It cannot handle the complex compute shaders, instanced rendering, and multi-threaded command buffer generation required for high-density spatial data. Spatial applications MUST utilize WebGPU (or its native Rust counterpart, `wgpu`) for modern, low-overhead rendering.
2. 2D-in-3D is Not Spatial: A spatial interface is not a virtual monitor. It is an environment where data has physical presence, depth, and parallax. UI components MUST be mapped to spatial semantics, utilizing volumetric rendering and world-locked anchors.
3. Latency is a Biological Threat: If the motion-to-photon latency (the time between a user moving their head and the screen updating) exceeds 20 milliseconds, the vestibular system detects the mismatch, causing nausea and disorientation. Latency in XR is not a performance issue; it is a health and safety requirement.
4. Space is State, Not a Backdrop: In AR, a digital object placed on a desk must remain there tomorrow. Physical environments MUST be mapped into persistent, semantic spatial graphs (cloud anchors), allowing digital state to be bound to physical geometry.

This document establishes the Mythos Spatial Computing Standard (MSCS), a comprehensive, evidence-based methodology for evaluating 3D engines and XR runtimes against rendering performance, biological safety, and spatial persistence.

Part I. Scope and Applicability

1.1 In Scope

This standard covers the evaluation of:

- WebGPU and `wgpu` (Rust) rendering pipeline architectures
- XR runtimes and APIs (OpenXR, WebXR, Apple visionOS)
- Motion-to-photon latency and frame rate stability (90Hz-144Hz+)
- Foveated rendering and eye-tracking integration

- Spatial anchors and persistent digital-physical mapping
- Volumetric data visualization (3D graphs, point clouds, digital twins)
- Edge-rendered split rendering (cloud rendering + local display)

1.2 Out of Scope

- General 2D UI semantics and accessibility (covered in MYTHOS-UX-0001)
- Visual aesthetic color tokens (covered in MYTHOS-UX-0003)
- General network transport latency (covered in MYTHOS-NET-0001)

1.3 Audience

- 3D graphics engineers and spatial computing architects
- UI/UX designers building volumetric data interfaces
- Platform engineers evaluating WebGPU vs. native engines
- AI engineers visualizing high-dimensional agent state spaces
- Standards bodies seeking a reference spatial computing methodology

Part II. Definitions and Taxonomy

2.1 Spatial Dimensions

Dimension	Definition
WebGPU / wgpu	The modern, low-overhead graphics API for the web (WebGPU) and its native, memory-safe Rust implementation (`wgpu`), replacing WebGL.
Motion-to-Photon Latency	The total elapsed time between a physical user movement (e.g., head turn) and the corresponding photon being emitted from the display. Must be <20ms to prevent nausea.
Foveated Rendering	A technique that uses eye-tracking to render high resolution only at the user's focal point and lower resolution at the periphery, drastically saving GPU compute.
Spatial Anchor	A digital coordinate persisted in a cloud graph, mapped to physical world geometry, allowing digital objects to remain fixed in space across sessions.
Volumetric UI	Interface components that possess depth and occupy 3D space, interacting with lighting and occlusion, rather than flat 2D billboards.

2.2 The Spatial Architecture Doctrine

SOURCE STATEMENT

WebGL is a legacy bottleneck. A floating 2D panel is not spatial. Latency above 20ms is a biological attack. If the digital state does not persist in physical space, it is a temporary illusion.

Part III. Evaluation Methodology

3.1 Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; WebGL, 2D billboards, unbound latency, no persistence
1	Basic WebXR, but relies on garbage-collected JS game loops
2	Functional 3D, but lacks foveated rendering or spatial persistence
3	Solid production performance; `wgpu`/WebGPU, <20ms latency, spatial anchors
4	Strong performance; foveated rendering, volumetric data, edge-split rendering
5	Best-in-class; sets the standard for mathematically verified, biologically safe spatial computing

Weighting

Category	Weight	Rationale
WebGPU/wgpu Pipeline & Compute	20%	WebGL cannot handle high-density spatial data
Motion-to-Photon Latency	20%	>20ms causes physical nausea; biological safety
Spatial Persistence & Anchoring	15%	Space must be treated as durable state
Foveated & Edge Rendering	15%	High resolutions require optical and network optimization
Volumetric Data Semantics	15%	2D UI in 3D space fails to leverage depth
Sensor Fusion & Input Tracking	15%	Hand/eye tracking must be precise and low-latency

Total: 100%

A system MUST score at least 3.0 in WebGPU Pipeline and Motion-to-Photon Latency to be considered for production use.

Part IV. Evaluation Categories

4.1 WebGPU/wgpu Pipeline and Compute (Weight: 20%)

4.1.1 Rendering Architecture

Does the system use a modern, low-overhead graphics pipeline?

Score	Criteria
0	WebGL 1.0 / 2.0 (state-machine, high overhead)
1	WebGPU via JavaScript, but heavily bound by GC pauses
2	Native WebGPU implementation, but single-threaded command buffer generation
3	`wgpu` (Rust) or native WebGPU with multi-threaded command buffer encoding
4	Advanced compute shaders for particle systems, spatial hashing, and culling
5	Perfect pipeline: formally verified memory safety, zero-allocation render loops, mathematically proven frame budget adherence

4.2 Motion-to-Photon Latency (Weight: 20%)

4.2.1 Biological Safety

Is the system mathematically guaranteed to prevent vestibular mismatch?

Score	Criteria
0	Untracked latency (>50ms); causes severe motion sickness
1	Basic IMU tracking, but frame drops cause stutter (30ms+)
2	Stable 72Hz framerate, but latency peaks during high compute (25ms)

Score	Criteria
3	Strict <20ms motion-to-photon latency enforced via hardware-level timewarp/spacewarp
4	Sub-15ms latency with predictive tracking and late-latching
5	Perfect safety: formally verified latency bounds, zero frame drops under any load, biologically guaranteed comfort

4.3 Spatial Persistence and Anchoring (Weight: 15%)

4.3.1 State Permanence

Do digital objects remain fixed in physical space across sessions and devices?

Score	Criteria
0	Objects exist only in app-space; reset on reload
1	Local-only persistence; drifts over time
2	Basic cloud anchors (e.g., ARCore/ARKit cloud anchors)
3	Semantic spatial graph: objects bind to recognized physical surfaces (tables, walls)
4	Multi-user shared spatial state with sub-centimeter accuracy
5	Perfect persistence: formally verified cross-device synchronization, zero spatial drift, cryptographic binding of digital state to physical coordinates

4.4 Foveated and Edge Rendering (Weight: 15%)

4.4.1 Optimization

How does the system handle the massive compute requirements of high-resolution XR displays?

Score	Criteria
0	Renders full resolution everywhere (exhausts GPU)
1	Fixed foveated rendering (static center high-res)
2	Dynamic foveated rendering driven by eye-tracking
3	Edge-split rendering: heavy compute offloaded to edge cloud, lightweight render locally
4	AI-predictive foveation: anticipates saccades to pre-render high-res areas
5	Perfect optimization: formally verified compute bounds, zero perceptible quality loss, seamless edge failover

4.5 Volumetric Data Semantics (Weight: 15%)

4.5.1 3D Interface Design

Is the data visualized using true volumetric depth and spatial semantics?

Score	Criteria
0	2D React panels rendered as billboards in 3D space
1	Basic 3D bar charts floating in space
2	Volumetric point clouds (e.g., network topology, agent state) with depth occlusion
3	Spatial UI bound to physical surfaces (e.g., data pouring out of a physical server rack)
4	Interactive volumetric shaders representing real-time multi-dimensional state
5	Perfect semantics: formally verified spatial cognitive load, zero 2D bias, mathematically optimal use of depth and parallax for data parsing

4.6 Sensor Fusion and Input Tracking (Weight: 15%)

4.6.1 Interaction Precision

How does the user interact with the spatial environment?

Score	Criteria
0	Standard Xbox controller mapped to 3D space
1	6-DoF controller raycasting (e.g., Meta Touch)
2	Basic hand-tracking (pinch to click)
3	High-fidelity skeletal hand-tracking with physical collision
4	Multimodal fusion: eye-tracking for focus, hands for manipulation, voice for command
5	Perfect interaction: formally verified zero-latency intent prediction, sub-millimeter tracking accuracy, zero false positives

Part V. The Mythos Spatial Computing Suite (MSCS)

Traditional 3D benchmarks measure polygon count. The MSCS evaluates biological safety, render pipeline memory safety, and spatial persistence.

5.1 Suite Composition

5.1.1 MSCS-Latency

- Compute Spike Injection: 100+ tests simulating heavy agent telemetry loads to verify the render loop maintains <20ms motion-to-photon latency.
- Drift Test: 50+ tests running for 1 hour to verify spatial anchors do not drift relative to physical objects.

5.1.2 MSCS-Pipeline

- Memory Safety Scan: 100+ tests running `wgpu` validation layers to verify zero memory leaks or buffer overruns during complex compute shaders.
- Allocation Tracking: 50+ tests verifying the render loop performs zero heap allocations during steady-state operation.

5.2 Execution Protocol

text

1. Deploy spatial application on target XR hardware
2. Run MSCS suite (automated load injection + memory profiling)
3. Score each category 0-5
4. Check for WebGL usage or >20ms latency spikes (instant disqualification)
5. If all thresholds met: approve for production

Part VI. Security Threat Model for Spatial Computing

6.1 Threat Catalog

6.1.1 Vestibular DoS (Latency Attack)

Severity: Critical Description: An attacker (or a runaway agent) consumes GPU resources, dropping the XR framerate below 20ms motion-to-photon latency, causing immediate physical nausea and disorientation to the user.

Required Controls:

1. Hardware-level timewarp that operates independently of application frame rate.
2. Hard compute budgets on spatial applications; runaway processes are instantly killed.

6.1.2 Spatial Anchor Spoofing

Severity: High Description: An attacker compromises the spatial graph and shifts a digital "Danger: High Voltage" warning sign three feet to the left, causing a user to walk into a physical hazard.

Required Controls:

1. Cryptographic signing of critical spatial anchors.
2. Semantic verification of anchor placement against physical LiDAR maps.

6.1.3 Biometric Data Exfiltration

Severity: High Description: Eye-tracking and pupil dilation data, which can indicate user cognitive load, stress, or sexual interest, is exfiltrated to a third-party analytics cloud.

Required Controls:

1. On-device processing of all biometric tracking data.
2. Zero egress of raw eye/hand tracking coordinates; only high-level "intent" events leave the device.

Part VII. The Mythos Spatial Computing Standard

7.1 The Mythos Spatial Doctrine

text WebGL is a legacy bottleneck. A floating 2D panel is not spatial.

Latency above 20ms is a biological attack. If the digital state does not persist in physical space, it is a temporary illusion.

Graphics may be virtual. Spatial physics must be absolute.

7.2 Selection Rules

1. No spatial architecture enters production without passing MSCS.
2. No architecture is approved if it uses WebGL for core rendering.
3. No architecture is approved if motion-to-photon latency exceeds 20ms.
4. No architecture is approved without spatial persistence for enterprise state.
5. No architecture is approved if biometric tracking data egresses the local device.

7.3 The Prime Directive for Spatial Architecture

SOURCE STATEMENT

Encode the command buffer, do not mutate the state machine. Bound the latency, do not hope for the framerate. Anchor the state, do not just render the frame. Fuse the sensors, do not just raycast the controller.

text Academic benchmarks measure polygon throughput. Applied benchmarks measure memory-safe pipelines. Security benchmarks measure biometric egress. Operational benchmarks measure motion-to-photon bounds.

SOURCE STATEMENT

Reality may be augmented. Biological safety must be absolute.

End of Standard MYTHOS-UX-0004

Appendix B - Source Disposition

Source	Disposition
MYTHOS-UX-0004_Spatial_Computing_3D_Engine_WebGPU_wgpu_Standard.md	Preserved in full; normative status corrected; evaluation anchors retained; enterprise architecture, implementation, evidence, assessment, authority, and publication controls added.
Original “Published Standard” label	Superseded by Candidate Standard status pending formal review and approval.
Original score-5 “perfect” wording	Preserved as source lineage but normalized to optimized and independently assured within defined scope.

Current authority validation

Validated 2026-09-17 / source-bound freshness record

This appendix records authority state verified for the permanent canonical edition. It does not replace the preserved normative body or imply certification, legal compliance, or implementation effectiveness.

W3C - Web Content Accessibility Guidelines (WCAG) 2.2

2.2 / W3C Recommendation

Current stable W3C Recommendation used as the accessibility baseline for public web experiences.

https://www.w3.org/TR/WCAG22/

W3C - Accessible Rich Internet Applications (WAI-ARIA) 1.2

1.2 / W3C Recommendation

Stable semantic accessibility specification; ARIA 1.3 remains under development and is not substituted here.

https://www.w3.org/TR/wai-aria-1.2/

W3C - WebGPU

Candidate Recommendation Draft 20 August 2026 / Candidate Recommendation Draft

WebGPU remains a Candidate Recommendation Draft; the latest Candidate Recommendation Snapshot is dated 2024-12-19. Revalidate before production baselines.

https://www.w3.org/TR/webgpu/