MYTHOS-UX-0001

AI-Native Interface, High-Density Data, and Accessibility Standard

Defines the interface architecture for high-density engineering systems in which AI proposes content and actions while typed renderers, accessibility semantics, progressive disclosure, multimodal input, and explicit control affordances preserve human comprehension and authority.

6	6	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-0001
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 the interface architecture for high-density engineering systems in which AI proposes content and actions while typed renderers, accessibility semantics, progressive disclosure, multimodal input, and explicit control affordances preserve human comprehension and authority.

2. Principal Findings

- Separate model-generated semantic intent from trusted interface rendering and action execution.
- Treat accessibility as an enforceable engineering property, not a final design review.
- Provide high-density operational visibility without hiding risk, uncertainty, provenance, or system state.
- Ensure every autonomous or consequential workflow remains interruptible and understandable.

3. Required Leadership Decisions

- Adopt a typed semantic UI contract and trusted renderer boundary.
- Fund accessibility automation, keyboard and screen-reader testing, contrast validation, and reduced-motion behavior.
- Define density profiles and progressive-disclosure rules for novice, expert, and high-stress operation.
- Require explicit provenance, confidence, approval, cancellation, and verification states in AI-native interfaces.

4. Enterprise Target State

TARGET OPERATING MODEL

Organizations applying MYTHOS-UX-0001 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 the interface architecture for high-density engineering systems in which AI proposes content and actions while typed renderers, accessibility semantics, progressive disclosure, multimodal input, and explicit control affordances preserve human comprehension and authority.

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-0001-C01	Semantic UI and LLM Isolation	20%	Require models to emit typed semantic proposals rendered by trusted components; never permit model output to become executable DOM, shell, or privileged action by default.
MYTHOS-UX-0001-C02	High-Density Rendering (WebGPU)	20%	Use level-of-detail, virtualization, aggregation, and GPU acceleration only where target-device evidence shows a material usability or performance benefit.
MYTHOS-UX-0001-C03	Programmatic Accessibility (a11y)	20%	Make accessibility semantics, keyboard operation, focus, contrast, zoom, reflow, captions, error handling, and assistive-technology testing release gates.
MYTHOS-UX-0001-C04	Cognitive Load and Progressive Disclosure	15%	Present essential state, risk, provenance, and next action first, while allowing expert detail without forcing every user to parse full system complexity.
MYTHOS-UX-0001-C05	HITL and Agentic Interruptibility	15%	Expose stop, pause, narrow-scope, deny, inspect, and rollback controls at the point where human authority matters.
MYTHOS-UX-0001-C06	Multi-Modal and Spatial Readiness	10%	Represent voice, image, screen, document, spatial, and text inputs as typed context with consent, provenance, accessibility, and modality-specific fallback.

MYTHOS-UX-0001-C01 - Semantic UI and LLM Isolation

NORMATIVE REQUIREMENT

The organization MUST require models to emit typed semantic proposals rendered by trusted components; never permit model output to become executable DOM, shell, or privileged action by default.

Objective and Risk Addressed

Require models to emit typed semantic proposals rendered by trusted components; never permit model output to become executable DOM, shell, or privileged action by default.

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

- semantic schemas
- renderer allowlists
- sanitization tests
- capability boundaries
- injection tests
- provenance labels.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Submit malicious and malformed model output and verify rendering, navigation, data binding, and actions remain within schema and policy.
- 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
schema-valid output rate	Track trend, threshold, exception, and accountable response.
unsafe render attempts blocked	Track trend, threshold, exception, and accountable response.
privileged action bypass findings	Track trend, threshold, exception, and accountable response.
provenance 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	LLM generates raw HTML/JS injected into the DOM (Critical XSS risk)
1	LLM generates Markdown, parsed into basic HTML

Score	Preserved source criterion
2	LLM generates JSON, but UI components are not strictly typed
3	Semantic UI Protocol: LLM outputs typed contracts (Zod/Protobuf); UI maps to isolated component registry
4	Bidirectional UI state: user interactions emit semantic events back to the agent without raw text parsing
5	Perfect isolation: formally verified semantic boundary, zero unescaped HTML capability, cryptographic validation of UI payloads

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-0001-C02 - High-Density Rendering (WebGPU)

NORMATIVE REQUIREMENT

The organization MUST use level-of-detail, virtualization, aggregation, and GPU acceleration only where target-device evidence shows a material usability or performance benefit.

Objective and Risk Addressed

Use level-of-detail, virtualization, aggregation, and GPU acceleration only where target-device evidence shows a material usability or performance benefit.

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

- performance budgets
- device matrix
- rendering traces
- fallback behavior
- memory and frame-time measurements.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.

- Load representative worst-case datasets on supported hardware and validate interaction latency, memory, frame stability, and accessible 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.
interaction latency	Track trend, threshold, exception, and accountable response.
memory peak	Track trend, threshold, exception, and accountable response.
dropped frames	Track trend, threshold, exception, and accountable response.
fallback usage	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	DOM-based rendering (React/Vue components); crashes >1,000 nodes
1	Virtualized DOM lists (handles 10k static nodes, fails on real-time updates)
2	Canvas 2D rendering for specific graphs
3	WebGL or WebGPU implementation for core data visualizations (>100k nodes at 60fps)
4	Full WebGPU/wgpu architecture; UI components rendered off-DOM; deterministic frame budgets
5	Perfect rendering: formally verified frame-time bounds, zero memory leaks over 24h continuous real-time streaming, hardware-accelerated accessibility tree

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-0001-C03 - Programmatic Accessibility (a11y)

NORMATIVE REQUIREMENT

The organization MUST make accessibility semantics, keyboard operation, focus, contrast, zoom, reflow, captions, error handling, and assistive-technology testing release gates.

Objective and Risk Addressed

Make accessibility semantics, keyboard operation, focus, contrast, zoom, reflow, captions, error handling, and assistive-technology testing release gates.

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

- automated scans
- manual audits
- keyboard scripts
- screen-reader results
- contrast reports
- defect remediation.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Execute automated and manual WCAG testing, including real assistive technology and high-density operational workflows.
- 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
critical accessibility defects	Track trend, threshold, exception, and accountable response.
keyboard completion rate	Track trend, threshold, exception, and accountable response.
screen-reader task success	Track trend, threshold, exception, and accountable response.
remediation age	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	No a11y checks; manual screen-reader testing fails
1	ESLint plugins for a11y (e.g., jsx-a11y), but easily bypassed
2	Automated scanning (Axe) in CI/CD, but blocks <50% of violations
3	Programmatic a11y: framework refuses to compile components missing required ARIA roles/focus management
4	Full WCAG 2.2 AAA compliance; custom hardware-accelerated accessibility tree synced with WebGPU
5	Perfect a11y: formally verified semantic accessibility graph, zero tab-traps, mathematically proven contrast ratios, 100% non-visual operability

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-0001-C04 - Cognitive Load and Progressive Disclosure

NORMATIVE REQUIREMENT

The organization MUST present essential state, risk, provenance, and next action first, while allowing expert detail without forcing every user to parse full system complexity.

Objective and Risk Addressed

Present essential state, risk, provenance, and next action first, while allowing expert detail without forcing every user to parse full system complexity.

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

- information architecture
- density profiles
- usability tests
- alert hierarchy
- user research
- stress-scenario results.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Test representative novice and expert tasks under ordinary and degraded conditions; measure discovery, errors, time, and interruption cost.
- 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
task completion time	Track trend, threshold, exception, and accountable response.
navigation errors	Track trend, threshold, exception, and accountable response.
alert recognition	Track trend, threshold, exception, and accountable response.
abandonment	Track trend, threshold, exception, and accountable response.
information-density preference	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	Dumps all agent logs and tool outputs into the main view
1	Basic collapsible panels for logs
2	Separate tabs/pages for telemetry vs. primary task
3	Aggregation layer: UI summarizes agent actions (e.g., "Searched 5 documents") with drill-down capability
4	AI-driven UI summarization: secondary LLM curates and highlights critical UI events in real-time
5	Perfect management: formally verified cognitive load bounds, context-aware progressive disclosure, zero information overload during critical events

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-0001-C05 - HITL and Agentic Interruptibility

NORMATIVE REQUIREMENT

The organization MUST expose stop, pause, narrow-scope, deny, inspect, and rollback controls at the point where human authority matters.

Objective and Risk Addressed

Expose stop, pause, narrow-scope, deny, inspect, and rollback controls at the point where human authority matters.

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

- approval UI
- cancellation tests
- task state model
- escalation paths
- recovery evidence.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Start long-running and consequential workflows, exercise interruption at each lifecycle state, and verify safe, observable outcomes.
- 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
cancellation latency	Track trend, threshold, exception, and accountable response.
indeterminate outcome rate	Track trend, threshold, exception, and accountable response.
approval comprehension	Track trend, threshold, exception, and accountable response.
uninterruptible task count	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	No pause capability; user must wait for agent to finish
1	Basic "Stop" button that kills the process
2	Can pause agent, but cannot inspect current context window or plan
3	HITL approval contracts: UI renders proposed action, blast radius, and rollback plan; user can approve/deny
4	Sub-second latency interruption; user can fork the agent state from the UI
5	Perfect HITL: formally verified interruption contracts, zero-latency state freezing, cryptographic evidence of user approval bound to agent action

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-0001-C06 - Multi-Modal and Spatial Readiness

NORMATIVE REQUIREMENT

The organization MUST represent voice, image, screen, document, spatial, and text inputs as typed context with consent, provenance, accessibility, and modality-specific fallback.

Objective and Risk Addressed

Represent voice, image, screen, document, spatial, and text inputs as typed context with consent, provenance, accessibility, and modality-specific 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

- modality contracts
- consent records
- transcript/source links
- fallback designs
- device compatibility tests.

Assessment and Test Procedure

- Confirm applicability, ownership, current architecture, and approved policy.
- Test equivalent workflows across supported modalities and confirm preserved identifiers, privacy, error recovery, and accessible alternatives.
- 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
modality success rate	Track trend, threshold, exception, and accountable response.
transcription correction rate	Track trend, threshold, exception, and accountable response.
fallback completion	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	Hardcoded to mouse/keyboard and 2D monitor
1	Basic responsive design (mobile/desktop)
2	Voice input/output supported
3	Headless UI protocol: semantic backend can drive web, CLI, TUI, and mobile clients equally
4	WebXR/Spatial computing integration; 3D data visualization via wgpu
5	Perfect abstraction: formally verified multi-modal state synchronization, zero visual bias in core logic, seamless transition between screen, voice, and spatial interfaces

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
WCAG 2.2	W3C Recommendation - Accessible web content outcomes	2027-01-18
ISO/IEC 40500:2025	Published international standard - International adoption of WCAG 2.2	2027-01-18
WAI-ARIA 1.2	W3C Recommendation - Accessible rich internet application semantics	2027-01-18
ARIA Authoring Practices Guide	Living guidance - Accessible widget design patterns; informative, not a standards-track requirement	2026-10-20
Media Queries Level 5	Evolving specification - User preferences including reduced motion and contrast	2026-10-20
WebGPU	Candidate Recommendation Draft - Modern browser graphics and compute; implementation status must be validated	2026-10-20

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-0001 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 W3C ARIA standards, WebGPU capabilities, or LLM structured output paradigms Applies To: All organizations evaluating, selecting, or operating front-end frameworks, AI interface renderers, high-frequency data dashboards, and accessibility automation pipelines for production, agentic, or enterprise use Companion Standards: MYTHOS-SWE-0008 (Semantic UI Protocols), MYTHOS-AI-0001 (Agentic Frameworks), MYTHOS-UX-0002 (HITL & Approval Workflows), MYTHOS-UX-0003 (Cyberpunk-Noir Aesthetic)

0. Executive Mandate

The architecture of human-computer interaction has failed.

Organizations take autonomous AI agents and force them to communicate through legacy HTML forms and conversational chat bubbles. They generate raw HTML from LLMs, creating massive Cross-Site Scripting (XSS) attack surfaces. They build command centers and operational dashboards using Document Object Model (DOM) rendering, which catastrophically fails when displaying thousands of real-time data points. Furthermore, they treat accessibility (a11y) as a post-release compliance checklist, resulting in interfaces that are fundamentally unusable by users with assistive technologies.

This standard exists because:

1. Chatbots are Not AI-Native: Confining an autonomous agent to a text chat window strips it of its capability to present structured, high-density information. AI-native interfaces MUST utilize Semantic UI Protocols, where the LLM emits typed data contracts (e.g., JSON/Protobuf) that the front-end deterministically maps to rich, accessible UI components.
2. DOM is for Documents, Not Data: Rendering high-density, real-time telemetry (e.g., network topologies, agent cognitive traces, system metrics) using React/DOM nodes causes catastrophic frame-rate drops and memory leaks. High-density data MUST be rendered via WebGPU, Canvas, or WebGL.
3. Accessibility is an Architectural Invariant, Not a Feature: An interface that cannot be navigated via screen readers, keyboard, or switch controls is fundamentally broken. Accessibility MUST be enforced at the compiler/type level (programmatic a11y), not hoped for during QA.
4. LLM-Generated UI is a Security Threat: Allowing an LLM to generate raw HTML/JavaScript and injecting it into the DOM (e.g., via `dangerouslySetInnerHTML`) is an unconditional security vulnerability. The UI layer MUST be strictly isolated from LLM reasoning via an Anti-Corruption Layer.
5. Cognitive Overload is an Attack Vector: AI can process 10,000 events per second; humans cannot. The interface MUST act as a semantic filter, progressively disclosing information and managing cognitive load, rather than dumping raw telemetry.

This document establishes the Mythos UI/UX Standard (MUS), a comprehensive, evidence-based methodology for evaluating AI-native interfaces against semantic rendering, high-density performance, and programmatic accessibility.

Part I. Scope and Applicability

1.1 In Scope

This standard covers the evaluation of:

- AI-native interface architectures (LLM-to-UI rendering pipelines)
- Semantic UI Protocols and typed payload contracts

- High-density data visualization (WebGPU, wgpu, Canvas, WebGL)
- Programmatic accessibility (a11y) and WCAG 2.2+ compliance automation
- Cognitive load management and progressive disclosure
- Agentic interruptibility and Human-in-the-Loop (HITL) UI affordances
- Spatial computing (XR) and multi-modal interaction readiness

1.2 Out of Scope

- General software design and DDD (covered in MYTHOS-SWE-0001)
- Desktop application architecture (covered in MYTHOS-SWE-0006)
- Visual state semantics and aesthetic themes (covered in MYTHOS-UX-0003)

1.3 Audience

- Front-end architects and principal UI engineers
- AI engineers building agent user interfaces and telemetry dashboards
- Accessibility engineers and QA automation teams
- Product designers and human-computer interaction specialists
- Standards bodies seeking a reference AI-native UI methodology

Part II. Definitions and Taxonomy

2.1 Interface Dimensions

Dimension	Definition
Semantic UI Protocol	An architecture where the LLM generates structured, typed data (e.g., a specific JSON schema representing a table or graph) rather than presentation code (HTML/CSS). The client renders the data using pre-verified, accessible components.
High-Density Data	Visualizing >10,000 concurrent, real-time data points or graph edges, requiring hardware-accelerated rendering (WebGPU) and bypassing the standard DOM.
Programmatic a11y	The practice of enforcing WCAG/WAI-ARIA standards at the framework compiler level (e.g., failing the build if a button lacks an aria-label), rather than relying on manual QA.
Cognitive Load Filter	A UI middleware layer that aggregates, summarizes, and prioritizes agent telemetry before presenting it to the human user, preventing information overload.
HITL Affordance	UI elements designed specifically to pause, inspect, and authorize autonomous agent actions, emphasizing blast-radius visualization and low-latency interruption.

2.2 The Interface Architecture Doctrine

SOURCE STATEMENT

A chatbot is a constraint, not an interface. The DOM is a document model, not a rendering engine. An LLM that writes HTML is a security liability. Accessibility is a compile-time requirement, not a post-release patch.

Part III. Evaluation Methodology

3.1 Scoring Model

Each capability is scored from 0 to 5.

Score	Meaning
0	Fails basic task; chatbot-only, DOM-thrashing, no a11y enforcement
1	Basic web UI with manual accessibility checks
2	Functional but LLM generates raw HTML or UI lacks high-density support
3	Solid production performance; Semantic UI Protocol, WebGPU rendering, programmatic a11y
4	Strong performance; cognitive filtering, spatial readiness, formally verified HITL UI
5	Best-in-class; sets the standard for mathematically verified, universally accessible AI interaction

Weighting

Category	Weight	Rationale
Semantic UI & LLM Isolation	20%	LLMs generating HTML creates XSS and layout destruction
High-Density Rendering (WebGPU)	20%	DOM cannot handle real-time agent telemetry or graph data
Programmatic Accessibility (a11y)	20%	Manual a11y QA always fails; it must be enforced in CI/CD
Cognitive Load & Progressive Disclosure	15%	Raw data dumps make the interface unusable
HITL & Agentic Interruptibility	15%	Users must be able to pause and verify agent state instantly
Multi-Modal & Spatial Readiness	10%	Voice, XR, and non-visual interfaces must be supported

Total: 100%

A system MUST score at least 3.0 in Semantic UI and Programmatic Accessibility to be considered for production use.

Part IV. Evaluation Categories

4.1 Semantic UI and LLM Isolation (Weight: 20%)

4.1.1 Rendering Pipeline

How does the LLM output translate to the user interface?

Score	Criteria
0	LLM generates raw HTML/JS injected into the DOM (Critical XSS risk)
1	LLM generates Markdown, parsed into basic HTML
2	LLM generates JSON, but UI components are not strictly typed
3	Semantic UI Protocol: LLM outputs typed contracts (Zod/Protobuf); UI maps to isolated component registry
4	Bidirectional UI state: user interactions emit semantic events back to the agent without raw text parsing
5	Perfect isolation: formally verified semantic boundary, zero unescaped HTML capability, cryptographic validation of UI payloads

4.2 High-Density Rendering (WebGPU) (Weight: 20%)

4.2.1 Data Throughput

Can the interface render massive, real-time datasets without frame loss?

Score	Criteria
0	DOM-based rendering (React/Vue components); crashes >1,000 nodes
1	Virtualized DOM lists (handles 10k static nodes, fails on real-time updates)
2	Canvas 2D rendering for specific graphs
3	WebGL or WebGPU implementation for core data visualizations (>100k nodes at 60fps)
4	Full WebGPU/wgpu architecture; UI components rendered off-DOM; deterministic frame budgets
5	Perfect rendering: formally verified frame-time bounds, zero memory leaks over 24h continuous real-time streaming, hardware-accelerated accessibility tree

4.3 Programmatic Accessibility (a11y) (Weight: 20%)

4.3.1 Enforcement Mechanism

Is accessibility guaranteed by the architecture, or hoped for by QA?

Score	Criteria
0	No a11y checks; manual screen-reader testing fails
1	ESLint plugins for a11y (e.g., jsx-a11y), but easily bypassed
2	Automated scanning (Axe) in CI/CD, but blocks <50% of violations
3	Programmatic a11y: framework refuses to compile components missing required ARIA roles/focus management
4	Full WCAG 2.2 AAA compliance; custom hardware-accelerated accessibility tree synced with WebGPU
5	Perfect a11y: formally verified semantic accessibility graph, zero tab-traps, mathematically proven contrast ratios, 100% non-visual operability

4.4 Cognitive Load and Progressive Disclosure (Weight: 15%)

4.4.1 Information Management

Does the interface protect the user from agent telemetry overload?

Score	Criteria
0	Dumps all agent logs and tool outputs into the main view
1	Basic collapsible panels for logs
2	Separate tabs/pages for telemetry vs. primary task
3	Aggregation layer: UI summarizes agent actions (e.g., "Searched 5 documents") with drill-down capability
4	AI-driven UI summarization: secondary LLM curates and highlights critical UI events in real-time
5	Perfect management: formally verified cognitive load bounds, context-aware progressive disclosure, zero information overload during critical events

4.5 HITL and Agentic Interruptibility (Weight: 15%)

4.5.1 Control Affordances

Can a user seamlessly pause, inspect, and authorize agent actions?

Score	Criteria
0	No pause capability; user must wait for agent to finish
1	Basic "Stop" button that kills the process
2	Can pause agent, but cannot inspect current context window or plan
3	HITL approval contracts: UI renders proposed action, blast radius, and rollback plan; user can approve/deny
4	Sub-second latency interruption; user can fork the agent state from the UI
5	Perfect HITL: formally verified interruption contracts, zero-latency state freezing, cryptographic evidence of user approval bound to agent action

4.6 Multi-Modal and Spatial Readiness (Weight: 10%)

4.6.1 Interface Flexibility

Is the interface abstracted from the visual screen?

Score	Criteria
0	Hardcoded to mouse/keyboard and 2D monitor
1	Basic responsive design (mobile/desktop)
2	Voice input/output supported
3	Headless UI protocol: semantic backend can drive web, CLI, TUI, and mobile clients equally
4	WebXR/Spatial computing integration; 3D data visualization via wgpu
5	Perfect abstraction: formally verified multi-modal state synchronization, zero visual bias in core logic, seamless transition between screen, voice, and spatial interfaces

Part V. The Mythos UI/UX Suite (MUS)

Traditional front-end benchmarks measure load time. The MUS evaluates semantic isolation, rendering throughput, and programmatic accessibility.

5.1 Suite Composition

5.1.1 MUS-HighDensity

- 100k Node Injection: 100+ tests streaming 100,000 real-time graph edges to verify WebGPU frame rates remain >55fps.
- 24h Soak: 50+ tests running continuous telemetry for 24 hours to verify zero memory leaks in the rendering engine.

5.1.2 MUS-a11y

- Blind Navigation Test: 100+ tests attempting to complete primary workflows using only keyboard and simulated screen-reader output.
- Compile-Time Violation: 50+ tests attempting to merge UI components missing ARIA bindings to verify the CI/CD pipeline blocks the build.

5.2 Execution Protocol

text

1. Deploy interface architecture with rendering engine and a11y linters
2. Run MUS suite (automated load injection + headless navigation)
3. Score each category 0-5
4. Check for LLM-generated raw HTML in the DOM (instant disqualification)
5. If all thresholds met: approve for production

Part VI. Security Threat Model for AI Interfaces

6.1 Threat Catalog

6.1.1 LLM UI Injection (XSS)

Severity: Critical Description: A prompt injection or malicious tool output causes the LLM to generate raw HTML/JavaScript (e.g., `<img src=x onerror=steal_cookies>`), which is then rendered in the user's browser, leading to session hijacking.

Required Controls:

1. Absolute prohibition of LLM-generated presentation code.
2. Semantic UI Protocol: LLM outputs typed JSON validated against a strict schema before rendering.

6.1.2 DOM-Based DoS (Cognitive Overload)

Severity: High Description: A runaway agent generates thousands of events per second, causing the front-end DOM to attempt rendering them all, freezing the user's browser and preventing them from hitting the "Stop" button.

Required Controls:

1. WebGPU/Canvas-based rendering for telemetry.
2. UI middleware rate-limiting and aggregating events before rendering.

6.1.3 Accessibility Denial of Service

Severity: High Description: A dynamic interface updates focus states so rapidly that screen readers become unusable, effectively locking visually impaired users out of the system during critical events.

Required Controls:

1. `aria-live` regions with strict politeness settings.
2. Programmatic focus management tied to HITL approval states.

Part VII. The Mythos UI/UX Standard

7.1 The Mythos Interface Doctrine

text A chatbot is a constraint, not an interface. The DOM is a document model, not a rendering engine.

An LLM that writes HTML is a security liability. Accessibility is a compile-time requirement, not a post-release patch.

Visuals may be subjective. Semantics must be absolute.

7.2 Selection Rules

1. No AI interface enters production without passing MUS.
2. No architecture is approved if the LLM generates raw HTML/JS.
3. No architecture is approved if it relies on the DOM for high-density data.
4. No architecture is approved without programmatic (compile-time) a11y enforcement.
5. No architecture is approved without sub-second HITL interruptibility.

7.3 The Prime Directive for Interface Architecture

SOURCE STATEMENT

Isolate the semantics, do not render the prompt. Accelerate the rendering, do not thrash the DOM. Enforce the accessibility, do not hope for compliance. Govern the cognitive load, do not dump the telemetry.

text Academic benchmarks measure load time. Applied benchmarks measure semantic isolation. Security benchmarks measure XSS prevention. Operational benchmarks measure high-density rendering.

SOURCE STATEMENT

Interfaces may be visual. Accessibility must be absolute.

End of Standard MYTHOS-UX-0001

Appendix B - Source Disposition

Source	Disposition
MYTHOS-UX-0001_AI_Native_Interface_High_Density_Accessibility_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.

NIST - Artificial Intelligence Risk Management Framework (AI RMF 1.0)

1.0 / Published; revision in progress

AI RMF 1.0 remains the published framework while NIST is actively revising it. Treat mappings as version-pinned and time-bounded.

https://www.nist.gov/itl/ai-risk-management-framework

NIST - NIST AI 600-1 - Generative Artificial Intelligence Profile

NIST AI 600-1 / Final profile

Companion profile for generative-AI risk management; use alongside the AI RMF and environment-specific evidence.

https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence

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/