

# EU AI Act

Compliance assessment — 2024/1689

## Server: ai.smithery/Phionx-mcp-hello-server

Slug: ai-smithery-phionx-mcp-hello-server

Scan id: 3cccf043-c529-431c-b47a-f814b1188cbb

Assessed at: 2026-05-11 09:21:49 UTC

Sentinel version: 0.4.0

Rules version: 2026-04-23

**Non-compliant**

**DRAFT for review — not legal advice. See attestation block for verification instructions.**

## Table of contents

---

1. Executive summary
2. Coverage & transparency
3. Controls summary
4. Control details
5. Multi-step attack chains
6. Cryptographic attestation

### 1. Executive summary

---

Assessment of ai.smithery/Phionx-mcp-hello-server against EU AI Act: overall status non compliant. Of 5 controls, 4 met, 1 unmet, 0 partial, 0 not applicable. 5 control(s) fell within MCP Sentinel's current assessor coverage; remaining control(s) are documented as not\_applicable until Phase 6 expands coverage. Unmet controls have findings at or above the framework's mandatory severity threshold and should be remediated before relying on this server in a regulated deployment. All claims are traceable to individual finding rows via finding\_id and to the governing rule via rule\_id; the enclosing signed envelope commits MCP Sentinel to the exact bytes of this report.

### 2. Coverage & transparency

---

Coverage band: low

Coverage ratio: 40%

Rules version: 2026-04-23

Analysis techniques applied:

- ast-taint
- capability-graph
- entropy
- linguistic-scoring
- schema-inference

### 3. Controls summary

---

ControlName

Status

Evidence

---

Art.9Risk

Management

System' Unmet

1

Art.12Record-

Keeping' Me@

Art.13  
Transparency  
& Provision of  
Information to  
Deployers' Met  
0

Art.14 Human  
Oversight' Met  
0

Art.15  
Accuracy,  
Robustness,  
and  
Cybersecurity'  
Met0

## 4. Control details

---

### Art.9 — Risk Management System

' Unmet

22 assessor rule(s) evaluated this control; 1 finding(s) observed (1 high); at least one finding is at or above the high threshold (status: unmet).

#### Evidence:

[High] D3  
(finding c77ba44  
3-8c86-4c25-96e  
0-99bc65577524  
, confidence  
88%)

SOURCE: external-content at npm:pint@0.20.0 — Dependency names are external content resolved from public package registries. A near-miss to a popular canonical name is a supply-chain anomaly under ISO

#### Required mitigations:

- Verify that the flagged dependency is the package you intended to install. Open the registry page for the candidate and compare publisher, publish date, download count, and postinstall scripts against the target. If it is not the intended package, replace it with the legitimate target, regenerate the lockfile, and audit the install environment (CI and developer machines) for any artifacts the malicious package may have written. Adopt a typosquat-aware package firewall (Socket.dev, Snyk Advisor, GitHub Dependabot) that rejects near-miss names at install time, in line with ISO 27001 A.5.21 supply-chain controls.

### Art.12 — Record-Keeping

' Met

5 assessor rule(s) evaluated this control; no findings observed.

### Art.13 — Transparency & Provision of Information to Deployers

' Met

14 assessor rule(s) evaluated this control; no findings observed.

### Art.14 — Human Oversight

' Met

13 assessor rule(s) evaluated this control; no findings observed.

## Art.15 — Accuracy, Robustness, and Cybersecurity

111 assessor rule(s) evaluated this control; no findings observed.

Met

## 5. Multi-step attack chains

---

No multi-step attack chains were synthesized for this server.

## 6. Cryptographic attestation

---

**Algorithm:** HMAC-

SHA256

**Key ID:** mcp-sentinel-  
dev

**Signer:** mcp-sentinel/  
v1

**Signed at:** 2026-05-14  
T09:32:25.781Z

**Canonicalization:**  
RFC8785

**HMAC-SHA256 signature (base64, wrapped at 64 chars):**

ZJD4QWolj0O30acmLLQw2Cr3mNB6xjIzN7PfJNiNbGE=

**Verification instructions:**

To verify this report:

1. Extract the report body (everything except the .attestation field).
2. Canonicalize the body via RFC 8785 (JCS).
3. Compute HMAC-SHA256 with the signing key for key\_id "mcp-sentinel-dev".
4. Base64-encode the result and compare with the signature above.

