ENTERPRISE TECHNICAL BRIEFING

VERSION 1.0 – MAY 7, 2026

Zero-Trust Infrastructure for Programmatic AI Commerce

A **technical and legal framework** for AI platforms, General Counsel, and Trust & Safety leadership. Issued by **PayAi-X FZE**, Dubai, UAE.

- **Scope:** AI commerce infrastructure
- **Audience:** Engineering, legal, and policy stakeholders
- **Authoritative reference:** catyai.io/research/zero-trust-ai-ads-en

Founder: Ioan Adrian Vitan

Document class: technical briefing
Security model: zero-trust
Deployment context: programmatic AI commerce
Jurisdictional base: Dubai, UAE

Executive Summary

TECHNICAL BRIEFING | ENGINEERING · LEGAL · TRUST & SAFETY

OpenAI's April 2026 launch of a programmatic Ads Manager operationalized conversational AI as a **transactional advertising channel**. Any AI platform that monetizes commercial recommendations now functions as a **regulated advertising system**, subject to evidentiary and attribution requirements. Conventional safeguards – system prompts, retrieval-augmented generation, and output filters – are **probabilistic controls**; they cannot prove that a specific commercial assertion was authorized by the merchant whose product is being recommended.

This briefing contends that platform liability for commercial hallucinations cannot be mitigated through behavioral steering alone. A defensible architecture requires **cryptographic provenance**: signatures over canonicalized merchant data, deterministic retrieval policy enforcement, assertion-level verification, and immutable audit logging. PayAi-X FZE has implemented this control stack in production under the **CatyAI** product.

⚠ The operative question is no longer *whether* AI advertising requires deterministic provenance. The issue is which infrastructure layer becomes the compliance standard – and which platforms continue to defend hallucinated commercial claims with prompt engineering when regulators intervene.

Briefing Scope

SECTION 2

Quantifies the liability surface under the **EU AI Act**, **FTC Section 5**, and the **Unfair Commercial Practices Directive**.

SECTION 3

Explains why **prompt engineering** and **RAG** cannot mathematically close the provenance gap.

SECTION 4

Defines the four-layer **Zero-Trust architecture** for provably authorized commercial assertions.

SECTIONS 5 - 6

Details the live reference implementation and three integration pathways for AI platforms.

TECHNICAL BRIEFING | LEGAL & ENGINEERING IMPLICATIONS

The Structural Shift: AI as an Advertising System

Until 2026, most consumer AI assistants functioned as **informational intermediaries**. Commercial output was incidental. With OpenAI's Ads Manager rollout, paid placements are now **ranked**, **optimized**, and **inserted** into AI responses on a CPC/CPM basis. Revenue is tied to the specific impression that generated the commercial claim.

This is the legal inflection point. Once a system inserts paid placements, ranks merchants by bid, and optimizes responses for conversion, it is no longer a passive host producing probabilistic text. **It is an active advertising system.** Regulators and courts evaluate such systems under **advertising law**, not merely information-retrieval standards.

Pre-April 2026

- **System role:** informational assistant
- **Commercial content:** incidental
- **Primary standard:** probabilistic accuracy

Post-April 2026

- **System role:** active advertising channel
- **Commercial content:** paid placements ranked and optimized
- **Primary standard:** advertiser-grade accountability

Because revenue is linked to the specific impression that produced the claim, **regulatory exposure** attaches at the response level.

The Liability Surface: Three Exposed Parties

In a canonical failure mode – an AI assistant outputs an unauthorized discount claim on behalf of a merchant – liability exposure is distributed across three actors. The distribution is materially asymmetric, with the **platform layer** carrying the primary risk under the relevant legal and regulatory frameworks.

Merchant

- **Reputational exposure** from unauthorized commercial attribution
- **Apparent-agency risk** where consumers reasonably infer merchant authorship
- Association occurs at the point of consumer perception, not content creation

User-Facing AI Platform

- **Primary exposure** due to direct monetization of the commercial message
- **Control function:** curates eligible merchants and ranks outputs
- **Revenue attribution** attaches to the impression that generated the false claim
- First-order target for regulators and private plaintiffs

Model Provider

- **Residual upstream exposure** where the model is marketed as commerce-capable
- Liability is attenuated relative to the platform, but not extinguished
- Risk increases when APIs are distributed with commercial-use documentation

Applicable frameworks – EU Unfair Commercial Practices Directive (Directive 2005/29/EC), FTC Section 5 (15 U.S.C. § 45), FTC Endorsement Guides (16 CFR Part 255), and the EU AI Act – already supply the operative standards. No novel legislation is required for enforcement in this fact pattern.

LITIGATION THEORY | SCALE ANALYSIS | COMMERCIAL AI RISK

The Plaintiff Theory at Scale

A **0.01%** hallucination rate is materially below the failure profile observed in many production LLM deployments in 2026. At internet-advertising scale – **billions** of AI-mediated commercial interactions per quarter – even a vanishingly small error rate yields **millions** of incorrect commercial assertions. That arithmetic, by itself, supports class-action numerosity and commonality arguments.

Input Rate

Hallucination rate
0.01%

Conservative floor for production LLMs in 2026; still sufficient to generate systemic harm at scale.

Exposure Volume

AI ad impressions / quarter
1B+

Projected quarterly scale for major AI advertising platforms after Ads Manager deployment.

Resulting Harm

False commercial claims
100K+

Incorrect assertions per quarter at conservative failure rates – each one a potential regulatory trigger.

Regulatory and Litigation Implications

- **Healthcare, financial services, insurance, legal services, and automotive recalls** are high-consequence domains where a single fabricated sentence can trigger enforcement exposure.
- The defense that *"the model merely generated tokens"* weakens substantially once the platform monetizes the specific impression that produced the false statement.
- At this scale, **foreseeability** and **traceability** become central issues for FTC actions, class certification, and personal-injury claims.

> The operative standard is shifting from *"the model behaved appropriately on average"* to *"this specific commercial assertion was provably authorized by the named merchant."* **Probability does not satisfy this standard. Cryptographic proof does.**

TECHNICAL BRIEFING | RISK CONTROLS | PROVENANCE GAP

Why Existing Mitigations Are Insufficient

The AI industry has standardized a mitigation stack composed of **system prompts**, **retrieval-augmented generation (RAG)**, **output classifiers**, and **human review workflows**. Each layer may reduce error rates, but none closes the **provenance gap**. The limitation is architectural: these mechanisms impose behavioral constraints on a stochastic generation process. None provides a mathematical guarantee over output content.

Mitigation Stack

- **Prompt engineering** steers generation through instructions.
- **RAG** conditions output on retrieved context.
- **Classifiers** detect anomalies after generation.
- **Human review** samples outputs for adjudication.

Core Limitation

Behavioral control ≠ content provenance

No layer establishes a cryptographic or deterministic
binding between:
- a specific commercial sentence
- and the merchant record that authorized it

System Boundary

Each mitigation layer operates **upstream** or **downstream** of the generation step. None establishes a verifiable link between a specific commercial sentence in the output and the specific merchant record that authorized that sentence. The generation step remains a **stochastic transformation**; no behavioral instruction can override the underlying token-prediction mechanism.

> **Operational controls reduce risk.** They do not produce evidentiary provenance.

TECHNICAL BRIEFING | ENGINEERING | LEGAL

Prompt Engineering: Behavioral Steering, Not Enforcement

Functional Limits

Instructions such as `"only answer using advertiser-approved data"` or `"never invent prices"` operate as **soft constraints** on a model that is fundamentally driven by **next-token prediction**. They influence the output distribution, but they do not eliminate invalid generations or provide deterministic enforcement.

The model can still interpolate, infer, synthesize, or confabulate even when the source database is accurate, because the **generation step** remains decoupled from any **verification step**. There is no mathematical guarantee that the emitted text corresponds to the underlying source record.

What Prompt Engineering Cannot Provide

- **Record-level correspondence** between an output price and a specific signed merchant record
- **Output integrity guarantees** against interpolation between two retrieved facts into a non-existent third fact
- **Cryptographic provenance** or an audit trail for a commercial assertion
- **Regulatory proof of authorization** under FTC or EU review standards
- **Legal exculpation** where revenue was earned on a materially misleading representation

TECHNICAL BRIEFING | ENGINEERING + LEGAL

RAG and Output Classifiers: Narrowing Without Proving

Retrieval-augmented generation can materially reduce hallucination frequency by constraining the generation context to retrieved source material. In production environments, a well-tuned RAG pipeline may reduce open-ended hallucination by an order of magnitude. That is operationally significant. It still does not establish output correctness, provenance, or authorization.

RAG

Function: Constrain generation with retrieved context

- Reduces semantic drift
- Improves grounding
- Does not guarantee factual fidelity

Output Classifiers

Function: Detect policy or safety violations in generated text

- Match known prohibited patterns
- Operate post-generation
- Cannot verify merchant authorization

Structural Constraint

Function: Reconstruct authorship and accountability externally

- Requires a trust anchor outside the model
- Needs cryptographic provenance
- Cannot depend on stochastic generation

→ RAG Reduces Error, It Does Not Eliminate It

The model can still fuse two retrieved facts into a non-existent third fact, generate a specific number adjacent to a retrieved range, or misattribute one merchant's offer to another. Retrieval is a probabilistic input filter, not an output guarantee.

→ Output Classifiers Are Recognition Systems

Toxicity classifiers, safety filters, and policy lookups operate on generated text. They can detect known disallowed patterns, but they cannot determine whether a commercial sentence is authorized by a specific merchant record without an external trust anchor.

→ The Core Systemic Issue

Conversational advertising collapses the boundary between retrieval and generation. Traditional advertising relies on human-authored static claims, reviewed and approved before delivery. Generative advertising inserts authorship into a stochastic process, so provenance and accountability must be established outside the model through cryptographic mechanisms.

ARCHITECTURE TECHNICAL BRIEFING

The Zero-Trust Architecture: Four Layers

A defensible architecture replaces *"the model probably behaved correctly"* with *"this specific commercial assertion was provably authorized by the named merchant at a specific time."* Four layers are required. Together they establish a complete chain of custody from merchant source data to user-facing output – each layer independently verifiable, each layer cryptographically anchored.

Core Objective

- **Eliminate implicit trust** in model-generated commercial claims
- **Bind every assertion** to merchant-authorized source material
- **Preserve auditability** across retrieval, verification, and emission

Verification Standard

- **Signed source data** as the trust anchor
- **Deterministic retrieval constraints** on admissible evidence
- **Pre-output validation** of every commercial statement
- **Immutable provenance records** for legal defensibility

Layered Control Plane

Layer 1 — Merchant-Signed Canonical Data

Ed25519 signatures protect canonical JSON records. A **JWKS public key endpoint** exposes the merchant verification material required to authenticate source data.

Layer 2 — Deterministic Retrieval Policy

Retrieval is **source-bounded** and limited to signed spans. Unsupported assertions trigger a **circuit-break** rather than speculative completion.

Layer 3 — Assertion Verification Engine

Before emission, each claim is checked for **signature validity**, **record existence**, **temporal validity**, and **jurisdictional permissibility**.

Layer 4 — Provenance and Audit Trail

Every commercial sentence is emitted with **signature metadata**, **signer identity**, **timestamp**, **source hash**, and **JWKS URI**.

Each layer addresses a distinct failure mode in the mitigation stack. Layer 1 establishes the trust anchor. Layer 2 constrains retrieval to signed evidence. Layer 3 enforces admissibility before any commercial sentence reaches the user. Layer 4 creates the evidentiary record required for regulatory review and dispute resolution.

ARCHITECTURE

Layer 1 — Merchant-Signed Canonical Data

The merchant publishes structured commercial records – **prices**, **offers**, **procedures**, **disclaimers**, and **availability** – and signs them with an **asymmetric private key** under merchant control. The resulting signature binds each claim to a precise state of the record at a specific timestamp, enabling independent verification without merchant participation at verification time.

Technical Foundation

Ed25519 (RFC 8032)
Canonicalization: JSON Canonicalization Scheme (RFC 8785)
JOSE alg: EdDSA (RFC 8037)
Public key distribution: JWKS (RFC 7517)

Verification Semantics

- **Signature validity:** cryptographic proof that the record was signed by the merchant key
- **Key provenance:** public verification key retrieved from an unauthenticated JWKS endpoint
- **Claim scope:** only merchant-authored commercial assertions are eligible for downstream use
- **Temporal binding:** the signature anchors the record to a specific point in time

Algorithm Ed25519 over canonicalized JSON	**Key Distribution** Merchant public key via JWKS endpoint
Signed Content Commercial claims, policies, and availability	**Assurance** Merchant control of the exact representation at a defined timestamp

Layer 2 — Deterministic Retrieval Policy

Generated responses are constrained to assertions that are provably grounded in **signed source records**. Any unsupported commercial token is rejected at the retrieval boundary – not merely discouraged via prompt instructions. This distinction separates a soft **prompt guardrail** from a hard **enforcement circuit**.

Retrieval Enforcement

- **Vector retrieval**: Qdrant, Pinecone, or pgvector
- **Source attribution**: every commercial token must map to a signed source span
- **Eligibility rule**: unsigned spans are excluded from the candidate set
- **Boundary enforcement**: policy is applied upstream of generation

Guardrail vs. Circuit Breaker

Prompt guardrail: "Use only approved data." This modifies model behavior probabilistically, but it does not eliminate the possibility of an invalid claim.

Policy circuit breaker: No unsigned commercial token reaches the eligible retrieval set. The model cannot emit claims that the system never exposes for generation.

Operational effect: enforcement is architectural, deterministic, and auditable.

VERIFICATION GATE | ASSERTION CONTROL

Layer 3 — Assertion Verification Engine

Before any commercial sentence is released to the user, the system evaluates **four mandatory invariants**. All four must satisfy policy. If any invariant fails, the assertion is suppressed and the response is regenerated or refused. No unverified commercial claim is ever exposed.

Verification Scope

- **Input:** signed merchant records and claim metadata
- **Decision rule:** pass only if all invariants are satisfied
- **Failure handling:** suppress assertion, regenerate, or refuse

Enforcement Standard

Assertion verification is a hard gate, not a soft recommendation. The engine validates provenance, existence, time bounds, and jurisdiction before any claim can enter the output channel.

`unverified_claim → block`

1

Signature Validity

The merchant's **Ed25519 signature** over the canonicalized record is cryptographically valid. The signing key resolves to the registered merchant identity in the JWKS document.

2

Record Existence

The specific offer or claim exists in the signed record set. The model cannot reference a merchant offer that was never submitted as a signed record.

3

Temporal Validity

The offer is currently active – not expired, not scheduled for future availability, not withdrawn. Timestamp bounds are checked against the signing timestamp and any embedded validity window.

4

Jurisdictional Permissibility

The offer is permissible under the relevant jurisdiction – e.g., medical claims comply with local advertising regulations in the user's jurisdiction at the time of serving.

LAYER 4 PROVENANCE & AUDIT TRAIL

Layer 4 — Provenance and Audit Trail

Every commercial sentence delivered to a user is accompanied by **provenance metadata**. This metadata constitutes the legally defensible evidentiary record a platform can produce when a regulator asks the dispositive question: *"Can you prove the merchant controlled the commercial representation made in their name?"*

Cryptographic provenance can answer yes. Prompt engineering cannot. The metadata bundle is **immutable**, **third-party verifiable**, and can be validated **post hoc** without any cooperation from PayAi-X infrastructure.

Provenance Bundle

Each delivered claim includes the following signed artifacts:

- **Signature** – Ed25519 signature over the canonicalized response payload, binding the response to the merchant's private key at a specific instant.
- **Signer Identity** – The `kid` that resolves the signature to the registered merchant entity in the public JWKS document.
- **Timestamp** – The signed authorization time; this is the moment the merchant key was applied to the specific claim, not the serving time.
- **Source Record Hash** – The cryptographic digest of the source record, proving the served content was derived from an unmodified signed record.

Validation Properties

```
provenance_bundle = {
  signature: Ed25519(canonical_response),
  signer_identity: kid → merchant JWKS entry,
  timestamp: signed authorization time,
  source_record_hash: SHA-256(source record)
}
```

These fields support **non-repudiation**, **integrity verification**, and **auditability** across internal review, external dispute resolution, and regulatory examination.

REFERENCE IMPLEMENTATION

CatyAI: Production Deployment

PayAi-X FZE has operationalized the four-layer architecture within the **CatyAI** product line. The system has been live in production since April 2026, processing real merchant traffic. Every production response is cryptographically signed and independently verifiable in real time by any third party, without reliance on PayAi-X infrastructure.

Service Endpoints

POST https://api.catyai.io/geo/v2/answer

GET https://api.catyai.io/.well-known/jwks.json

Verification Properties

- **Signed response payload** with Ed25519 / EdDSA authentication
- **Key identifier (kid)** published via RFC 7517-compliant JWKS
- **Signed timestamp** and **source content hash** for auditability
- **Unattended third-party verification** with no infrastructure cooperation required

Signed Answer Endpoint

`POST https://api.catyai.io/geo/v2/answer`

Each response includes an **Ed25519 signature**, **signing key identifier (kid)**, **JWKS URI**, **signed timestamp**, and **content hash**.

Public JWKS Endpoint

`GET https://api.catyai.io/.well-known/jwks.json`

Open access, no authentication required, RFC 7517 compliant. Active signing key: `catyai-akl-signing-key-2026-v1`

Cryptographic Standard

EdDSA (RFC 8037) on the **Ed25519** curve (RFC 8032).

Verification completes in **sub-millisecond** time on commodity x86 hardware. JWKS responses are cacheable for up to **24 hours** under RFC 7517.

VERIFICATION PROCEDURE

Live Verification: Reproducing the Proof

Any **independent third party** can validate a signed production response in **three deterministic steps** without cooperation from PayAi-X infrastructure: **retrieve the JWKS document**, **resolve the public key by kid**, and **verify the Ed25519 signature** against the canonicalized response payload.

Verification Request

```
curl -s -X POST https://api.catyai.io/geo/v2/answer \
  -H 'Content-Type: application/json' \
  -H 'User-Agent: GPTBot/1.0' \
  -d '{
    "widget_id": "7e750a95-4d72-4c82-8eae-bb37a89194c8",
    "question": "What services do you offer?",
    "lang": "en"
  }'
```

Verification Workflow

- **Fetch** the public JWKS document.
- **Match** the signing key using the response `kid`.
- **Validate** the `Ed25519` signature over the canonical payload.

Reference Verification Script

A minimal **Node.js** verifier using `@noble/ed25519` as the only dependency is published at the research URL referenced in Section 8. It performs all three checks: **JWKS retrieval**, **key resolution by kid**, and **Ed25519 signature validation**. No PayAi-X account or credentials are required.

20

Paying Merchant Clients

Healthcare, professional services, and SME e-commerce as of May 7, 2026.

21

Autonomous Agent Classes

Orchestrated across multiple specialization tiers under a unified Zero-Trust policy.

170+

API Endpoints

Managed under the unified Zero-Trust policy in the production deployment.

INTEGRATION

Integration Pathways for AI Platforms

AI platforms that expose commercial advertising surfaces have three implementation paths for adopting cryptographic provenance. The optimal choice depends on platform maturity, control surface, engineering capacity, and risk appetite. All three pathways can support a defensible commercial-output posture; they differ in integration depth, operational complexity, and residual liability exposure.

A. SDK Integration

Mechanism: Platform invokes a verification SDK before serving any commercial output. The SDK validates signatures and returns `pass`, `fail`, or `regenerate`.

- **Effort:** 2-4 engineering weeks for a single product surface
- **Risk profile:** Lowest; verification remains a side-channel
- **Control:** Full platform ownership of generation flow

B. Sidecar Gateway

Mechanism: Platform routes commercial responses through a NAP gateway service. The gateway returns either a signed-and-verified response or a refusal envelope.

- **Effort:** 4-8 weeks, including routing and retry logic
- **Risk profile:** Medium; gateway becomes a critical-path dependency
- **Control:** Centralized enforcement with explicit SLA and failover requirements

C. Native Protocol

Mechanism: Platform implements NAP at the model boundary: every token in commercial scope must resolve to a signed source span. This requires inference-layer changes.

- **Effort:** 8-16 weeks; requires token-attribution coordination
- **Risk profile:** Highest engineering investment, lowest residual liability surface
- **Control:** Deepest integration with provenance enforced in-band

INTEGRATION TECHNICAL BRIEFING

Pathway A Recommended: SDK Integration First

For most AI platforms, **Pathway A (SDK integration)** is the preferred initial control pattern. It enables defensible commercial output in the highest-risk verticals – **healthcare**, **financial services**, and **regulated commerce** – without modifying the inference layer. The platform retains deterministic control over user experience, latency envelope, and product surface.

NAP verification operates as a **side-channel control plane** that gates a narrowly scoped subset of commercial responses. The implementation footprint is bounded: **2-4 engineering weeks** for a single product surface, with **no inference-layer dependency**. This pathway can be deployed immediately while Pathways B and C remain on a longer-horizon roadmap.

Latency Budget

Verification latency is dominated by the **JWKS fetch**, which is cacheable for up to **24 hours** under RFC 7517. **Ed25519** signature verification completes in **<0.5 ms** on commodity x86 hardware.

End-to-end verification, including an HTTPS round trip to the JWKS endpoint when uncached, typically measures **8-25 ms** on production traffic. With JWKS cached, the verification path is **sub-millisecond**.

Failure Mode Handling

On verification failure, the platform should apply one of three controlled recovery actions to preserve its liability posture:

1. **Regenerate** the response with a stricter retrieval scope
2. **Substitute** a refusal message that routes the user to the merchant's authoritative source
3. **Escalate** to a human review queue for high-stakes verticals

In each case, the failed assertion is never exposed to the user, and the platform does not present an unverified commercial claim.

TECHNICAL BRIEFING | ENGINEERING & LEGAL

Evaluation Process and Engagement Structure

PayAi-X engages platform engineering and legal stakeholders through a controlled two-stage evaluation. No commercial commitment is required to complete technical review. The full evaluation and technical due diligence are conducted under mutual NDA and culminate in a shareable integration scoping document.

01

45-Minute Scoping Call

Scope: current commercial output surface, risk classification, mitigation stack, jurisdictional exposure, and roadmap alignment.

02

One-Week Technical Due Diligence

Scope: shared access to the production endpoint, threat-model walkthrough, signature canonicalization review, and integration scoping output. **No commercial commitment** required.

03

Engagement Path A

Fixed-scope project: a specific commercial surface with defined acceptance criteria. Typical duration: **6-10 weeks**. Best suited to a single high-risk vertical.

04

Engagement Path B

Multi-quarter advisory: embedded provenance support across protocol evolution, regulatory response, and audit-readiness across multiple product surfaces.

Evaluation Artifacts

- **Integration scoping document**
- **Threat model** and control assumptions
- **Signature canonicalization** specification review
- **Commercial surface** and jurisdictional risk map

Disclosure Boundary

Mutual NDA is standard before any integration scoping conversation. Public materials – this briefing, the research page, the JWKS endpoint, and the signature verification code – require no agreement and are independently verifiable in real time.

LEGAL BASIS | TECHNICAL STANDARDS | ENFORCEMENT CONTEXT

Regulatory and Standards References

The legal basis is established, not hypothetical. Each regulatory instrument listed below was already in force before April 2026. No new legislation is required to pursue enforcement actions against AI platforms that monetize false commercial assertions; the instruments apply on their existing terms.

Applicable Regulatory Instruments

EU Unfair Commercial Practices Directive

Directive 2005/29/EC, Articles 5 - 7. Governs misleading commercial communications and applies to AI-generated commercial output in consumer-facing contexts without modification.

EU AI Act

Regulation (EU) 2024/1689, Articles 10, 50, and 52. Establishes obligations for transparency, risk classification, and deployers of AI systems used in commercial contexts.

FTC Section 5

15 U.S.C. § 45. Prohibits unfair or deceptive acts or practices in commerce. Platform revenue derived from a false impression is sufficient nexus for enforcement under existing doctrine.

FTC Endorsement Guides

16 CFR Part 255, Sections 255.0 and 255.5. Address material connections and substantiation of representations. AI-ranked paid placements are treated as endorsements.

Technical Standards Baseline

RFC 8032 EdDSA
RFC 8037 JOSE / EdDSA
RFC 7517 JWK
RFC 8785 JSON Canonicalization Scheme

These are all published **IETF standards**. No pending revisions alter the implementation described in this briefing.

Implementation note: The Zero-Trust architecture is anchored in current standards only; the document relies on settled protocol definitions, not draft specifications or forward-looking assumptions.

Contact and Independent Verification

Primary Technical Contact

Ioan Adrian Vitan
Founder, PayAi-X FZE
Email: adrian@payai-x.com
LinkedIn: linkedin.com/in/adrian-vitan-lopes-35525a13
Corporate domain: payai-x.com

PayAi-X FZE is headquartered in Dubai, UAE, with engineering operations in Romania. **CatyAI** is the production deployment in which the **Network AI Protocol (NAP)** operates against live merchant traffic.

Independent Verification Assets

- **Technical research page** with implementation notes, live signature demonstration, and Node.js verification snippet: catyai.io/research/zero-trust-ai-ads-en
- **Production JWKS endpoint** (public, unauthenticated): api.catyai.io/.well-known/jwks.json
- **PayAi-X corporate site:** payai-x.com
- **Independent press coverage:** start-up.ro, May 5, 2026

Confidentiality and Disclosure Scope

Threat models, API surface details, latency characteristics, and roadmap dependencies are disclosed under mutual NDA. All public materials are self-contained, require no agreement, and are independently verifiable in real time.

This document includes forward-looking technical and commercial statements. Where independently verifiable evidence exists, references are provided in Section 8 of the source briefing. Where claims depend on PayAi-X internal data (for example, paying merchant count, agent count, or endpoint count), the figures are reproducible under NDA upon request. **Reference:** catyai.io/research/zero-trust-ai-ads-en – Version 1.0, May 7, 2026