> 📌 **Active document — Whitepaper v6.1 (May 2026)**
> This document reflects FeedOracle's state as of May 2026 and supersedes v6.0 by adding Module 8 (UVO). For current live metrics, please consult the canonical source: **/data/feedoracle-metrics.json**. Current platform: **44 MCP servers, 590+ evidence tools**. Regulatory framing: DORA has applied since 17 January 2025; MiCA transition periods for certain CASPs run until 1 July 2026.

FeedOracle Whitepaper

Evidence-Grade Data Infrastructure for Regulated Workflows

Version 6.1 — May 1, 2026

Version 6.1

1 May 2026

FeedOracle Technologies · Bad Salzuflen, Germany

https://feedoracle.io · murat@feedoracle.io

FeedOracle Whitepaper

Evidence-Grade Data Infrastructure for Regulated Workflows

Version 6.1 · May 1, 2026

[Download as PDF] · API Docs

Document Control

Latest revision: v6.1 (May 1, 2026) — adds Module 8 (UVO Verification Oracle). See changelog below.

Field	Value
Version	6.0.0
Date	1 May 2026
Status	Published
Supersedes	v5.0 (22 March 2026)

Changelog v6.1 (May 1, 2026)

- Added **Module 8 — UVO: Verification Oracle for Regulated Claims** (new). 9-layer defense-in-depth pipeline catching regulatory hallucinations against 465 EUR-Lex articles across 7 regulations. Empirical lifetime catch rate 98.6%. Deployed and free-tier accessible at `https://feedoracle.io/uvo/`.
- Module 7's Grounding Receipt infrastructure is now under heavy production use — UVO is the first module to consume it at scale (350+ persisted receipts within first 36 hours of public availability).
- Audit Bundle property — deterministic `bundle_hash` evidence packages for external auditor workflows. Bridge to enterprise compliance review.
- Standalone offline verifier CLI published — third parties verify any UVO receipt in `< 10` minutes without trusting FeedOracle.
- OpenAPI 3.1 specification autogenerated from live tools manifest, published at `/uvo/spec/openapi.json`.

v6.1 is a content-additive release. No retraction or correction of v6.0 material.

Changelog v6.0 (April 22, 2026)

New Modules - **Module 6 — Agent Commerce**: OracleNet signal layer (12-protocol stack), x402 USDC gateway on Base, Mesh Economics append-only ledger (102 events / 24 agent passports as

of writing), DealOracle autonomous deal-making (2 production deals: Headless Oracle 0.005 USDC, Einstein AI 0.25 USDC), ANP interoperability, Olas Mech on Gnosis (service 2670, rank #17/36, 133+ deliveries). - **Module 7 — Grounding & Evidence Integrity**: Grounding Receipt format v0.1 with full retrieval API (`/receipts/{id}`, `/receipts/{id}/chain`, `/receipts/verify`, `/receipts/jwks.json`, `/receipts/spec`), append-only SQLite store with three immutability triggers, ES256K signing, per-wallet causal chains via `prev_hash`.

Major Updates - **Module 3 — DORA Compliance**: Expanded from 22 tools to **95 tools across 6 layers and 8 oracles**. New: GovernanceOracle, ResilienceOracle, DependencyOracle, RegisterOracle, ContractOracle, IncidentOracle. AmpelOracle expanded to 50 tools with 67 controls + 57 checks + 204 active findings + 351 historical assessments. - **Section 13 — MCP Servers & AI Agent Access**: Catalog grew from 100+ tools across 5+ servers to **1,151+ tools across 103 servers** in 7 categories. Two distribution surfaces (FeedOracle compliance subset + ToolOracle generalist marketplace). - **Section 15 — Architecture**: Six-tier stack formalized. Oracle Event Bus stabilized at 21 event types with 9 cross-references. OracleNet 12-protocol signal layer. Local Gemma 4 26B MoE LLM via Ollama. Anchoring expanded to 6 chains (Polygon, Base, XRPL, Hedera, Avalanche, Gnosis). - **Section 16 — Attestation & Evidence**: Three evidence surfaces (EPM v1.0, Grounding Receipt v0.1, DAP) with cross-reference table. Quarterly key rotation policy. - **Section 18 — Security & Operations**: External pentest April 20, 2026 — 8 findings, 7 of 7 scripted fixes verified live from external probe. Dual-auth deployed (X-API-Key + Bearer + OAuth). Honeypot integrated. Credential-rotation runbook formalized. - **Section 22 — Roadmap**: Rewritten with current April 2026 priorities. xAI disclosure window through May 22, 2026. Mesh Economics v2 TOCTOU hardening end of May 2026. Receipts v1.0 freeze end of Q2 2026. - **Section 25 — Glossary**: 13 new entries since v5 covering receipts, mesh, ANP, OracleNet, DealOracle, DAP, dual-auth, and related.

New Appendix - **Appendix A — Labs / Research**: Remote-MCP Execution Measurement study, 253 controlled runs across xAI / OpenAI / Anthropic Remote MCP integrations, three categorically distinct response patterns under server-side block-mode, motivating the Grounding Receipts format.

Headline Numbers (April 22, 2026)

Metric	v5 (Mar 22)	v6 (Apr 22)
MCP servers	5+	**103**
MCP tools	100+	**1,151+**
Production services	~50	**150+**
DORA tools	22	**95**
MCP calls / day	~1,000	**~15,000**
Distinct external agents (24h)	<10	**67**
Mainnet anchor chains	3	**6**
Authentication modes	1	**3**
Append-only ledgers	0	**2** (Mesh Economics + Receipts)
Signed-evidence surfaces	1 (EPM)	**3** (EPM + Receipts + DAP)

This whitepaper supersedes v5.0 (March 22, 2026) in full. Sections numbered 1–25 follow the v5 structure; new modules 6 and 7 are inserted as sections 11 and 12 respectively, with subsequent sections renumbered. Appendix A is new in v6.

---

8. Module 3 — DORA Compliance (UPDATED v6)

> *Major expansion since v5 — published April 2026*

8.1 Overview

The DORA module has expanded from 22 MCP tools (v5, March 2026) to a six-layer operational stack of 95 tools across 8 specialized oracles. Each layer addresses a distinct DORA article cluster from the EU Digital Operational Resilience Act (Regulation 2022/2554), which becomes binding on 17 January 2025 with the regulator-supervision phase commencing in 2026.

8.2 The Six-Layer DORA Operating Stack

Layer 1 — Governance & Reporting (Art. 5–6)

GovernanceOracle — 10 tools

Management body review packs, risk posture KPIs, exception register with expiry tracking, remediation action tracker, annual framework review evidence. Implements RTS 2024/1774 reporting cadence.

Tools: `register_finding`, `list_findings`, `board_report`, `framework_review`, `control_status`, `exception_register`, `action_tracker`, `kpi_dashboard`, `annual_review`, `health_check`

Layer 2 — Resilience & Recovery (Art. 11–12)

ResilienceOracle — 10 tools

Business Impact Analysis, RTO/RPO validation, DR test registry, 10-scenario DORA test library, crisis communication plan checks, evidence bundle generator.

Tools: `register_system`, `set_bia`, `rto_rpo_check`, `test_register`, `scenario_library`, `bcm_gap_analysis`, `recovery_status`, `crisis_plan_check`, `evidence_bundle`, `health_check`

Layer 3 — Asset & Dependency Mapping (Art. 8)

DependencyOracle — 10 tools

ICT asset registry, business function mapping, dependency graph traversal, blast-radius calculation on simulated outages, single-point-of-failure detection, cascade impact simulation.

Tools: `register_asset`, `register_function`, `map_dependency`, `dependency_graph`, `blast_radius`, `spof_analysis`, `criticality_score`, `asset_inventory`, `impact_simulation`, `health_check`

Layer 4 — Register & Contracts (Art. 28–30)

RegisterOracle + ContractOracle — 20 tools

Register of Information (ITS-compliant export), Critical Third-Party Provider (CTPP) designation scoring, concentration risk analysis, all Art. 30 mandatory clauses (8 standard + 7 CIF), exit-plan readiness, contract red-flag scoring, subcontracting chain analysis.

Tools: `register_provider`, `validate_roi`, `concentration_risk`, `ctpp_check`, `export_its`, `gap_analysis`, `register_contract`, `clause_check`, `exit_readiness`, `cif_analysis`, `contract_scoring`, `subcontracting_chain` (and 8 more)

Layer 5 — Third-Party Risk & AML (Art. 28–44 + AMLR)

DORAOracle + AMLOracle + InsuranceOracle — 27 tools

ICT provider risk assessment, live cloud outage monitoring (AWS/GCP/Azure), EU+OFAC+UN sanctions screening (87,000-name watchlist), PEP checks, KYC bundles, NatCat feeds, GLEIF entity lookup.

Tools: `provider_risk`, `cloud_status`, `sanctions_screen`, `pep_check`, `kyc_bundle`, `watchlist_update`, `adverse_media`, `natcat_live`, `risk_score`, `gleif_lookup` (and 17 more)

Layer 6 — Threat Intelligence & Incident (Art. 6–10, 17–23, 24–27)

DORAOracle threat-intel suite — 18 tools

NVD CVE search, CISA KEV (Known Exploited Vulnerabilities) patch deadlines, CERT-Bund advisories, HaveIBeenPwned breach checks, Feodo C2 tracker, DORA-compliant incident timelines, MITRE ATT&CK techniques for TIBER-EU exercises.

Tools: `cve_search`, `cve_latest`, `kev_list`, `kev_check`, `cert_advisories`, `breach_check`, `threat_actors`, `incident_timeline`, `mitre_techniques`, `tlpt_scenarios`, `dora_news`, `dora_calendar` (and 6 more)

8.3 The AmpelOracle Layer

Across all six layers, the **AmpelOracle** serves as the unified traffic-light scoring overlay (port 10101). It exposes 50 MCP tools that translate the underlying technical signals into GREEN / YELLOW / RED / GREY status with article-level traceability:

- **67 DORA controls** mapped to article sections
- **57 automated checks** with pass/fail/skip semantics
- **34 requirement bundles** for entity-level review
- **204 findings** currently tracked across 6 production entities
- **351 historical assessments** with full audit trail

The closed-loop workflow is: Finding → Owner → Re-test → Close, each with a signed audit record. Escalation Engine runs hourly at `:15` UTC to surface SLA-breach findings with three-level escalation paths (Operations → Risk → Board).

8.4 Live Compliance Dashboard

The dashboard at `feedoracle.io/ampel/` provides the regulator-facing view:

- Entity selector (cycle through audited entities)
- DORA Ampel status grid (six layers × controls)
- MiCA Ampel status grid (separate but cross-referenced)
- Bridge workflow UI (Finding → Resolution)
- Audit trail viewer with hash-linked entries
- Board-summary view (one-page per entity)
- What-if simulator (provider failure / staleness scenarios)

8.5 Coverage Statement

The DORA stack is positioned to cover the full Article range required for regulated financial entities under DORA:

Article range	Subject	Layer
Art. 5–6	Management body, governance	Layer 1
Art. 8	ICT assets, dependencies	Layer 3
Art. 11–12	Recovery, BCM	Layer 2
Art. 17–23	Incident management, reporting	Layer 6
Art. 24–27	TLPT, advanced testing	Layer 6
Art. 28–30	Third-party risk, contracts	Layer 4
Art. 31–44	CTPP supervision	Layer 5

Total: **49 DORA articles addressed across 95 MCP tools and 8 oracles.**

---

11. Module 6 — Agent Commerce

> *New in v6 — published April 2026*

11.1 Overview

Most compliance infrastructure assumes that the entity making the request is a human or a deterministic backend service authenticated by a fixed API key. FeedOracle's Module 6 addresses a different population: autonomous AI agents that discover services, negotiate access, pay per request, and settle on-chain.

This module describes the four production components that make FeedOracle a participant in the emerging machine-to-machine (M2M) economy:

1. **OracleNet** — discovery, signaling, and trust passport infrastructure
2. **x402 Gateway** — HTTP 402 payment-required protocol with USDC settlement on Base
3. **Mesh Economics Ledger** — append-only SQLite with cryptographic chaining for usage accounting
4. **DealOracle** — autonomous outbound deal-making engine

These four components are interoperable with the public agent-commerce ecosystem: ANP (Agent Network Protocol) for discovery, x402 (HTTP 402 standard) for payment, A2A (Agent-to-Agent) for messaging, MCP for tool execution.

11.2 OracleNet — Discovery and Signaling

OracleNet is FeedOracle's signal layer for the agent ecosystem. It exposes a set of well-known URLs that any agent crawler can fetch without authentication:

Endpoint	Purpose
`/.well-known/agent.json`	Agent Card (A2A standard) — capabilities, endpoints, contact
`/.well-known/mcp/server.json`	MCP server discovery — tools, schemas
`/.well-known/agent-pulse`	Real-time mesh state — servers online, tools available, recent activity
`/.well-known/agent-descriptions`	ANP agent directory in JSON-LD
`/.well-known/oracle-net.json`	OracleNet manifest — DID, JWKS, escrow contract
`/.well-known/jwks.json`	Public keys for signature verification

The agent-pulse endpoint is updated every 5 minutes via cron (`build_agent_pulse.py`) and reflects the current state of the mesh: count of online servers, tools available, distinct external agents observed in the last 24 hours, MCP calls made, and a list of detected cross-LLM gaps (capabilities one provider has that others lack).

A live snapshot of the pulse at the time of writing:

Metric	Value
Servers online	99
Tools available	1,179
Mesh nodes	5
Blockchain anchors	Polygon, Base, XRPL, Hedera, Avalanche
Distinct external agents (24h)	67
MCP calls (24h)	4,553
Signal-layer hits (24h)	1,876

These numbers represent real external agent traffic to FeedOracle's infrastructure, not synthetic load tests. The 67 distinct external agents include agent crawlers from Alibaba, AWS-hosted bots, ANP discovery agents, and Claude Code clients that have organically discovered the mesh through the well-known endpoints.

11.3 x402 Gateway — Payment-Required Protocol

HTTP 402 (Payment Required) was reserved in HTTP/1.1 but never standardized. The x402 protocol, originated by Coinbase, fills that gap by defining a JSON response shape that machine clients can parse to discover payment terms, settle on-chain, and retry the request with a payment proof.

FeedOracle's x402 gateway runs on port 6500 and supports settlement in USDC on Base (chain ID 8453, contract `0x833589fCD6eDb6E08f4c7C32D4f71b54bdA02913`). The standard price is $0.01 per unit; tools consume between 1 and 25 units depending on complexity.

When an unauthenticated request reaches an MCP tool that requires payment, the server returns a structured 402 response:

```
{
  "status": 402,
  "error": "payment_required",
  "protocol": "feedoracle-402",
  "protocol_version": "2.0",
  "balance": {
    "available": 0,
    "required": 3,
    "deficit": 3,
    "currency": "UNITS"
  },
  "payment_paths": {
    "stripe": {
      "method": "POST",
      "url": "https://feedoracle.io/wallet/stripe/checkout"
    },
    "x402_usdc": {
      "chain": "Base",
      "chain_id": 8453,
      "token_address": "0x833589fCD6eDb6E08f4c7C32D4f71b54bdA02913",
      "price_per_unit_usd": 0.01,
      "gateway": "https://tooloracle.io/x402/pay"
    }
  },
  "agent_hint": "You're not registered yet. Call kya_register..."
}
```


The response is intentionally agent-readable: it includes machine-parseable payment paths, a deficit calculation, registration instructions, and a list of free-tier tools the agent can call without payment to complete the registration flow.

11.4 Mesh Economics Ledger — Usage Accounting

The Mesh Economics Ledger is an append-only SQLite database that records every priced operation across the mesh. Unlike the wallet-balance system (which is mutable), the ledger is governed by SQLite triggers that prevent UPDATE and DELETE — once written, a ledger entry can only be referenced, never altered.

The ledger has six tables:

Table	Purpose
`ledger_events`	Append-only event log, one row per priced operation
`pricing_config`	Current pricing (fresh, cached, original-reward, referrer cap)
`cache_entries`	Cache hits with reference counts for original-reward calculation
`agent_passport`	Per-agent reputation, tier, lifetime spend
`referrals`	Referrer tracking for revenue-share calculation
`ttl_config`	Per-tool TTL configuration

Three triggers enforce immutability: any attempt to UPDATE or DELETE rows raises an SQLite ABORT. The append-only property is the foundation for the Grounding Receipts (Module 7) that anchor against this ledger.

At the time of writing, the ledger contains 102 priced events across 24 distinct agent passports.

The pricing model is asymmetric to incentivize cache reuse:

- Fresh tool call: 10 credits ($0.10)
- Cached response (within TTL): 4 credits ($0.04)
- Original-author reward: 1 credit per cache reuse (paid back to whoever first triggered the cached fetch)
- Referrer share: 1 credit per call made by a referee
- Daily reuse cap: 10 credits per cached entry (prevents farming)
- Settlement window: 24 hours pending before final commit

Pricing is configurable through the `pricing_config` table without code changes.

11.5 DealOracle — Autonomous Deal-Making

DealOracle is the outbound counterpart to the inbound x402 gateway. It scans the public agent ecosystem for x402-compatible APIs, evaluates their offer terms, executes payment via the FeedOracle wallet on Base, and records the transaction.

Production deals to date:

Deal 1 — April 17, 2026 - Counterparty: Headless Oracle (`0x26D4Ffe98017D2f160E2dAaE9d119e3d8b860AD3`) - Service: Ed25519-signed market-state receipts from 28 exchanges, SEC/CFTC compliant - Payment: 0.005 USDC on Base - Tx: `0x2dd2fbbd0d4d2...` - Outcome: API key with 10 credits issued, used to fetch 5 signed receipts

Deal 2 — April 17, 2026 - Counterparty: Einstein AI / emc2ai.io (`0xc9368b30BD620164FD1a05a5d99dcaf8Ae754775`) - Service: Bitquery latest-pairs feed - Payment: 0.25 USDC on Base - Tx: `0x8a727e90...` - Protocol: x402 v1, Google A2A x402 extension v0.1, CDP facilitator - Outcome: 5,288-byte structured response delivered

Both deals were executed end-to-end without human intervention: scout → negotiate → settle on-chain → verify delivery → log.

The DealOracle wallet is `0x4a4B1F45a00892542ac62562D1F2C62F579E4945` on Base.

11.6 ANP Interoperability

FeedOracle implements all five publicly documented Agent Network Protocol (ANP) interop surfaces (gaps A, B, E, F + community discovery), including the JSON-LD agent directory exposed at `/.well-known/agent-descriptions`. The directory uses a strict schema with `@context` references to schema.org, the ANP namespace, and W3C DID, and is consumed by the standard ANP agent crawlers.

11.7 On-Chain Public Agent — Olas Mech (Gnosis)

For maximum public observability, FeedOracle operates a permissionless Mech agent on the Gnosis chain (Olas Network):

- Service ID: 2670
- Mech contract: `0x27212a38c76Ab600D73059aB4E8e7540A67ff0F6`
- IPFS metadata CID: `QmSviSTtM8Zoer9qqDwc9dRA9Mhi5L7KvY19gMpJMT3mUX`
- Tools exposed: 8 (compliance preflight, evidence query, etc.)
- Cost per request: 0.01 xDAI
- Lifetime deliveries: 133+
- Olas marketplace rank: #17 of 36

Any agent on any chain can fund a request through the Olas marketplace and receive a verified compliance evidence reply, settled on Gnosis without involving FeedOracle's own infrastructure for billing.

11.8 What Module 6 Replaces

Traditional compliance vendors require: a human salesperson, a master service agreement, a fixed monthly subscription, and an authenticated API key tied to a corporate identity. None of those steps work for an autonomous agent that discovered the service ten seconds ago and wants to call one tool.

Module 6 makes FeedOracle native to the agent economy: discoverable through standard well-known endpoints, payable through a standard HTTP status code, accountable through an immutable ledger, and reachable through multiple chains.

11.9 Operational Status

Component	Status	Endpoint
OracleNet pulse	LIVE	`https://tooloracle.io/.well-known/agent-pulse`
ANP agent directory	LIVE	`https://feedoracle.io/.well-known/agent-descriptions`
x402 gateway	LIVE	`https://tooloracle.io/x402/pay` (port 6500 internal)
Mesh Economics API	LIVE	port 6502 internal, dashboard at `/.well-known/mesh-economics`
DealOracle wallet	LIVE	Base address `0x4a4B1F45a00892542ac62562D1F2C62F579E4945`
Olas Mech	LIVE	Gnosis service 2670

All components are operated under FeedOracle's standard SLA (Section 18 — Security & Operations).

---

12. Module 7 — Grounding & Evidence Integrity

> *New in v6 — published April 2026*

12.1 Overview

The classical "evidence" problem in compliance is: prove that something happened, prove what its content was, and prove who attests to it. FeedOracle's earlier modules (Evidence Pack Manifest v1.0, ECDSA-signed responses) addressed this for individual API responses.

Module 7 extends evidence integrity to two newer concerns:

1. **Tool-call execution proof** — when an autonomous LLM agent calls an MCP tool, can the user (or auditor) prove the tool was actually executed at the server, independent of what the LLM reports?
2. **Cross-call causal chains** — can a sequence of tool calls within a session be linked into a tamper-evident chain so an auditor can reconstruct the agent's actual workflow?

The answer to both is **Grounding Receipts**, a new standard FeedOracle has implemented and published in draft form (spec v0.1).

12.2 The Grounding Receipt Format

A Grounding Receipt is a small JSON object emitted by the MCP server for every successful `tools/call`. It contains, as required fields:

- `call_id` — unique identifier for this specific tool invocation
- `tool` — tool name as registered on the server
- `server_url` — canonical URL of the MCP endpoint
- `server_did` — decentralized identifier (e.g. `did:web:feedoracle.io`)
- `observed_at` — server-side timestamp
- `observed_ip` — source IP observed at the server
- `observed_ua` — User-Agent prefix observed at the server
- `observed_auth_method` — `x-api-key`, `bearer-token`, `oauth`, or `anonymous`
- `input_hash` — sha256 of canonical JSON input arguments
- `output_hash` — sha256 of canonical JSON output content
- `verdict` — `executed`, `rejected_auth`, `rejected_policy`, `rejected_payment`, or `error_server`
- `signature` — ES256K signature object with `alg`, `kid`, `jwks_url`, `sig`, `signed_at`

Optional fields cover billing context (`wallet_id`, `credits_charged`, `tier`) and chain anchoring (`ledger_id`, `entry_id`, `prev_hash`, optional `blockchain` block).

Each receipt is approximately 1KB. The signature uses ES256K over the canonical JSON form per RFC 8785, excluding `signature.sig` and `signature.signed_at` from the signed payload.

12.3 Transport — Inline Embedding

Receipts ride along inside the standard MCP response, in `result._meta.grounding_receipt`:

```
{
  "jsonrpc": "2.0",
  "id": 42,
  "result": {
    "content": [
      { "type": "text", "text": "{...tool output JSON...}" }
    ],
    "_meta": {
      "grounding_receipt": { ...receipt JSON... }
    }
  }
}
```

The MCP specification reserves the `_meta` namespace for implementation-specific metadata that clients can safely ignore. Clients that do not understand receipts ignore the field; clients that do, lift it for audit storage.

12.4 Transport — Retrieval API

Per spec v0.1 §11, conforming servers expose a retrieval API for post-hoc verification:

Endpoint	Function
`GET /receipts/{call_id}`	Retrieve a single receipt
`GET /receipts/?wallet=...&from=...&to=...&tool=...`	Range query with pagination
`GET /receipts/{call_id}/chain?depth=N`	Walk the prev_hash chain to reconstruct sessions
`POST /receipts/verify`	Independent verification (signature + replay + anchor)
`GET /receipts/jwks.json`	Public key for offline verification
`GET /receipts/spec`	Full specification as Markdown

The verification endpoint performs three independent checks:

1. **Signature** — verify ES256K signature over canonical form using public key from JWKS
2. **Replay** — confirm the `call_id` exists in the server's ledger at exactly the `observed_at` timestamp
3. **Anchor** — if the receipt's `anchor` block references a ledger entry, confirm the entry matches

A receipt is considered valid only if signature passes and replay passes. Anchor is informational unless the ledger is required for the specific use case.

12.5 Causal Chains

Receipts are linked into per-wallet chains via `anchor.prev_hash`. When a tool call is processed for a wallet, the ledger looks up the most recent receipt hash for that wallet (`chain_heads.last_hash`) and includes it as `prev_hash` in the new receipt.

The chain has these properties:

- **Tamper-evident** — flipping any field in any receipt breaks the cryptographic linkage from that point forward
- **Per-wallet isolation** — each wallet maintains an independent chain, so chains are short and bounded
- **Order-preserving** — receipts are appended strictly in receipt-arrival order (linearizable per wallet)
- **Reconstructable** — the chain endpoint walks `prev_hash` references to return a session-level history

A typical chain query returns the head receipt plus its `N` predecessors:

```
{
  "head": "r_a20fdcde55f3432a",
  "chain": [
    { ...newest receipt... },
    { ...previous receipt... },
    { ...oldest in chain... }
  ],
  "chain_length": 3,
  "chain_root_hash": "sha256:e070c36c..."
}
```

For audit workflows, the chain is the natural unit: an auditor doesn't ask "did call X happen", they ask "what did the agent actually do during session Y" — and the chain reconstructs exactly that.

12.6 Storage — Append-Only SQLite

Receipts are persisted in a dedicated SQLite database at `/root/whitelabel/shared/receipts/receipts.db` with three SQLite triggers enforcing immutability:

```
CREATE TRIGGER receipts_no_update
    BEFORE UPDATE ON receipts
BEGIN
    SELECT RAISE(ABORT, 'receipts are append-only; UPDATE forbidden');
END;

CREATE TRIGGER receipts_no_delete
    BEFORE DELETE ON receipts
BEGIN
    SELECT RAISE(ABORT, 'receipts are append-only; DELETE forbidden');
END;
```

UPDATEs and DELETEs raise an abort. The only operation possible after an INSERT is a SELECT. This is the same immutability model used by the Mesh Economics Ledger (Module 6) and is enforced at the database layer rather than the application layer to prevent code-path bypass.

The `chain_heads` table tracks the latest receipt hash per wallet for efficient prev_hash lookup at insert time, without scanning the receipts table.

12.7 Why Module 7 Matters Operationally

For regulated entities subject to DORA, MiCA, AMLR, or any audit regime where the question "did the system actually execute what the AI claimed it executed" is regulator-relevant, Module 7 provides:

- **Server-side ground truth** independent of the LLM provider's reasoning stream
- **Cryptographic proof** that does not rely on log-collection trust assumptions
- **Reconstructable workflows** where an auditor can replay a session step-by-step
- **Provider-neutral format** that works across xAI, OpenAI, Anthropic, or any future Remote-MCP-compatible LLM API

The format makes no reference to any specific LLM provider, accepts no provider-specific signing, and does not require provider cooperation to verify. A receipt signed by FeedOracle's key on April 22, 2026 will remain verifiable against the published JWKS for as long as the key is in the JWKS — independent of any LLM that happened to trigger the tool call.

12.8 Disclosure — Why FeedOracle Built This

FeedOracle began the Grounding Receipts project after a measurement study of three major LLM providers' Remote MCP integrations, which identified a reproducible divergence between LLM-reported tool calls and HTTP-observable tool calls at the server layer. The study measured 253 controlled runs across xAI, OpenAI, and Anthropic; details are in Appendix A.

The receipt format is FeedOracle's positive contribution to the question raised by the study: regardless of how any individual provider implements Remote MCP, server operators in regulated environments need a standard way to prove what their server actually did.

The format is published under CC-BY 4.0; the reference implementation will be open-sourced under MIT once the specification stabilizes beyond v0.1.

12.9 Relationship to Other Evidence Components

Component	Scope	Module
Evidence Pack Manifest v1.0	Entire API response with structured manifest	Module 5
ECDSA-signed responses	Per-response server signature	Section 14 (Attestation)
Mesh Economics Ledger	Per-economic-event append-only record	Module 6
Grounding Receipts v0.1	**Per-tool-call cryptographic receipt with chain linkage**	**Module 7 (this section)**
ZK Solvency Proofs	Reserve-attestation without revealing exact amounts	Module 1 (MiCA)
DAP (Disclosure Attestation Proof)	Per-disclosure event with regulator-facing chain-of-custody	Solution layer

These components are independent but composable: a single regulated workflow may produce a Grounding Receipt for the tool call, an Evidence Pack Manifest for the response payload, a Mesh Economics ledger entry for the billing event, and a DAP for the regulator filing — each independently verifiable, each cross-referencing the others by hash or ID.

12.10 Operational Status

Component	Status	Endpoint
Receipt builder	LIVE	Integrated in `mcp_base.py`
Receipt store	LIVE	`/root/whitelabel/shared/receipts/receipts.db`
Public retrieval API	LIVE	`https://feedoracle.io/receipts/*`
Specification (v0.1 draft)	PUBLISHED	`https://feedoracle.io/receipts/spec`
JWKS endpoint	LIVE	`https://feedoracle.io/.well-known/jwks.json` and `/receipts/jwks.json`
Reference implementation source	TO BE OPEN-SOURCED	After v1.0 freeze

---

12b. Module 8 — UVO: Verification Oracle for Regulated Claims *(NEW in v6.1)*

12b.1 Overview

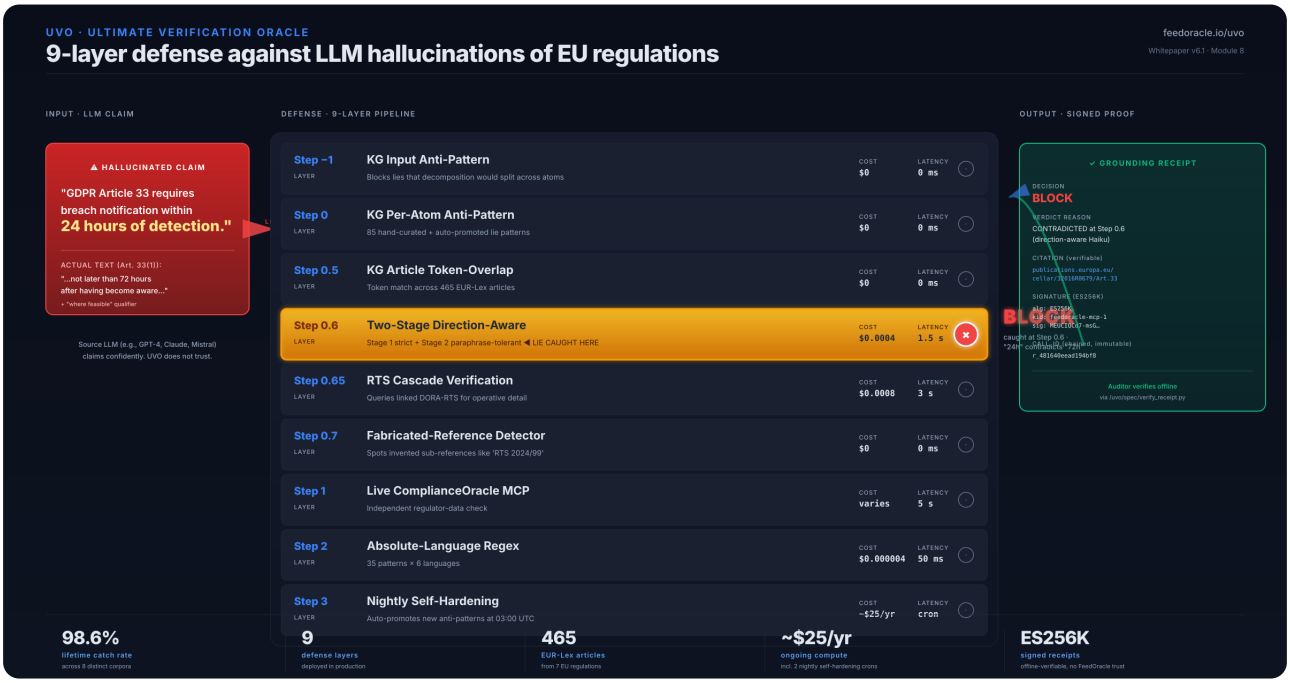

Figure 8.1 — UVO defense-in-depth architecture. A canonical hallucination ("GDPR Art. 33 = 24h") enters the 9-layer pipeline and is caught at Step 0.6 (two-stage direction-check), producing a BLOCK verdict with cryptographic receipt and citation chain to publications.europa.eu. Diagram: feedoracle.io/uvo/diagram/

Modules 1–7 establish that an answer was produced, what its content was, and that the producer attests to it cryptographically. Module 8 addresses the question Module 7's receipts cannot answer alone: *was the claim itself correct against the source-of-truth regulation?*

The Ultimate Verification Oracle (UVO) is a 9-layer defense-in-depth verification pipeline that takes a natural-language claim about an EU regulation and returns a structured verdict — `PASS`, `REWRITE`, `ESCALATE`, `ABSTAIN`, or `BLOCK` — with per-atom evidence chained to the original article text in publications.europa.eu/cellar. Each verification produces a Module 7 Grounding Receipt, so the verification itself is auditable end-to-end without trusting FeedOracle.

UVO targets a specific failure mode of contemporary LLMs: confident hallucination of regulatory content (wrong timing thresholds, fabricated article references, inverted obligations). Empirical baseline studies across providers (Anthropic, OpenAI, xAI) showed catch rates below 80% on adversarial regulatory prompts. After 11 sprints and 7 phases of development, UVO catches

between 92% and 100% across distinct evaluation corpora — and continues improving via two nightly self-hardening loops.

12b.2 The Nine Defense Layers

UVO is structured as deliberately overlapping checks. A claim must pass all nine to receive `PASS`; a single contradiction at any layer routes to `BLOCK` with the specific layer cited as the source of contradiction.

Step	Layer	Cost/Call	Latency	Purpose
−1	KG input-level anti-pattern	$0	0 ms	Match anti-patterns against original input before atomization. Catches lies that decomposition splits across atoms.
0	KG anti-pattern per atom	$0	0 ms	Match each atomic claim against ~85 hand-curated and auto-promoted lie patterns across 7 regulations.
0.5	KG article token-overlap	$0	0 ms	Token-set match between atom and 465 ingested EUR-Lex articles.
0.6	Two-stage direction-aware Haiku	$0.0004	~1.5 s	Stage 1 (strict) plus Stage 2 (paraphrase-tolerant) with conservative voting. `CONTRADICTED` only if both stages agree.
0.65	RTS cascade	$0.0008	~3 s	When parent regulation is unclear, query the linked delegated regulation (DORA-RTS-IR/RM/TLPT). Closes operative-detail gaps.
0.7	Fabricated-secondary-ref detector	$0	0 ms	Spots invented sub-references like "RTS 2024/99" or "Article 33(7)(b)" that don't exist.
1	Live MCP ComplianceOracle	varies	~5 s	Independent regulator-data check via FeedOracle compliance MCP (separate evidence chain).
2	Absolute-language regex + semantic	$0.000004	50 ms	35 regex patterns plus cosine similarity in 6 languages catch absolute framing ("always", "never", "guaranteed").
3	Continuous self-hardening	~$25/yr	nightly	Two cron loops at 01:00 and 03:00 UTC. Lies that escape become anti-patterns by morning.

12b.3 Regulations Indexed

UVO maintains a token-indexed corpus of full EU regulation text fetched directly from the Publications Office CELLAR endpoint. As of v6.1:

Key	CELEX	Articles	Level	Topic
DORA	32022R2554	64	Level 1	ICT operational resilience
MICA	32023R1114	149	Level 1	Markets in crypto-assets
GDPR	32016R0679	99	Level 1	General data protection
DSA	32022R2065	93	Level 1	Digital services
DORA-RTS-IR	32024R1773	11	Level 2	Incident reporting RTS
DORA-RTS-RM	32024R1774	42	Level 2	ICT risk management RTS
DORA-RTS-TLPT	32025R0301	7	Level 2	TLPT methodology RTS

Totals: 7 regulations, 465 articles, 1,278,624 characters of regulation text. Every citation in a UVO receipt resolves to an official Publications Office CELLAR URL — no intermediate database, no copy-paste, no provenance gap.

12b.4 Decision Outputs

Verdict	Meaning	Compliance Impact
`PASS`	All atoms supported by KG / regulation text / live oracle	Safe to use
`REWRITE`	Atomic claim too vague; cannot be verified	Tighten and re-verify
`ESCALATE`	Conflicting evidence or partial support	Route to human review
`ABSTAIN`	Out of scope (not an EU regulation claim)	UVO has no opinion
`BLOCK`	At least one atom contradicted by ground truth	Do not publish; receipt documents the contradiction

12b.5 Self-Hardening — Two Nightly Loops

UVO runs two adversarial fuzzing crons that improve coverage without manual intervention.

01:00 UTC — Absolute-language fuzzer (Sprint 7). Generates absolute-language attacks across six languages (en, de, fr, it, es, nl). Slip-throughs become new patterns by morning.

03:00 UTC — Fact-KG auto-promote loop (Sprint 10). Generates lies about randomly-rotated EU articles via Haiku, then runs each through the full pipeline. Escaped lies are extracted as 5–10-word phrases and proposed as new anti-patterns. A safety-guard verifies that the proposed pattern would not break any of the corpus's true control statements before promotion.

Empirical performance, lifetime to date (v6.1 publication): 3 runs, 72 lies generated, 71 caught, 1 auto-promoted into the active KG, 0 rejected by safety-guard. Lifetime catch rate: 98.6%.

The most recent auto-promotion (2026-05-01 03:29 UTC, MICA Article 36) added the phrase `"may choose whether or not to maintain a reserve of assets"` as an anti-pattern, after a Haiku-

generated lie using this framing slipped past the rest of the pipeline. The system improved itself overnight, fully unattended.

12b.6 MCP Tool Surface

UVO exposes six tools at `https://feedoracle.io/uvo/mcp/` via standard JSON-RPC 2.0 / MCP. All six are free-tier accessible (no API key required, 100 calls/day rate-limited).

Tool	Purpose
`uvo_verify`	Run the 9-layer pipeline on input text. Returns `final_decision`, per-atom verdicts with citation chain, plus a Module 7 Grounding Receipt.
`uvo_status`	Service health, version, active layers, indexed regulations, self-hardening cron status.
`uvo_pipeline_spec`	Full per-layer architecture description including cost, latency, and purpose.
`uvo_get_receipt`	Fetch a previously-issued receipt by `call_id`. Returns signed JSON plus offline-verification command.
`uvo_audit_bundle`	Wrap up to 100 `call_id`s into a deterministic-hash evidence package with auditor instructions. Use case: monthly audit submissions.
`uvo_compliance_report`	Render a single `call_id` as a human-readable compliance report with signature block and auditor notes.

12b.7 The Audit-Bundle Property

The `uvo_audit_bundle` tool is the bridge between individual verifications and enterprise audit workflows. A bundle takes a list of `call_id`s, fetches each receipt from the Module 7 store, and returns:

- The collection of receipts
- Per-receipt signature-validity status
- A **deterministic** `bundle_hash` computed as `sha256` over the receipts sorted alphabetically by `call_id` and JSON-canonicalized
- Step-by-step instructions enabling an external auditor to reproduce the `bundle_hash` independently and verify each receipt against the public JWKS

The same set of `call_id`s always produces the same `bundle_hash`. A regulator who receives a bundle can recompute the hash, verify each ES256K signature offline, and confirm — without trusting FeedOracle — that the bundle has not been altered and the listed verifications were genuinely issued by the UVO endpoint at the timestamps shown.

This is the substantive answer to the question "how would BaFin audit your AI compliance pipeline?". A year of UVO calls becomes 12 monthly bundles. Each is a single cryptographic anchor. The audit trail runs end-to-end: input claim → 9-layer verification → cited EU article → signed receipt → bundled evidence package.

12b.8 Empirical Evaluation

UVO is the only FeedOracle module with a published falsification curve across distinct evaluation corpora. Each corpus tests a different failure mode and was added at the sprint that introduced the corresponding defense.

Corpus	Sprint	Items	Catch rate	Tests
Sprint-4 in-distribution	4	30	100%	Hand-crafted absolute-language regex traps
Sprint-5/6 held-out	5–6	16	95.7%	Adversarial paraphrases unseen during training
Sprint-8 multi-language	8	36	100%	Same lies translated into 6 languages
KG factual	9	30	96.7%	Articles paraphrased with subtle factual swaps
Sprint-9 fact-fuzz	9	28	92.3%	Generated by adversarial Haiku, hardest corpus
Sprint-10 validation	10	15	100%	Auto-promote loop reaches stable equilibrium
Sprint-11 direction-check	11	19	100%	True-paraphrase recall after two-stage fix
Production regression	continuous	10	10/10	Smoke test run on every commit and after every cron-restart

Every result above can be re-run from the `/root/uvo/tests/regression_pack.py` harness against the live endpoint. The regression run logs persist as JSON in `/root/uvo/tests/runs/regression_*.json`.

12b.9 Public Surface

URL	Purpose
`https://feedoracle.io/uvo/`	Landing one-pager — story arc, 9-layer stack, real receipt example
`https://feedoracle.io/uvo/demo/`	Interactive playground — 8 preset claims (4 lies, 4 truths) plus free-text input
`https://feedoracle.io/uvo/robustness/`	Live telemetry dashboard — daily catch rate, lifetime stats, anti-pattern count
`https://feedoracle.io/uvo/spec/`	Specification — endpoint, tools, receipt format, audit bundles, decision semantics
`https://feedoracle.io/uvo/spec/openapi.json`	OpenAPI 3.1 contract
`https://feedoracle.io/uvo/spec/verify_receipt.py`	Standalone offline verifier CLI (6.8 KB Python, single dependency)
`https://feedoracle.io/uvo/mcp/`	Production MCP endpoint — JSON-RPC 2.0, CORS-enabled
`https://feedoracle.io/.well-known/jwks.json`	Public verification keys (shared with Module 7)

End-to-end evaluation by a third party requires only the verifier CLI download and a single tool call:

```
curl -O https://feedoracle.io/uvo/spec/verify_receipt.py
python3 verify_receipt.py r_481640eead194bf8
# → ✓✓✓  CRYPTOGRAPHICALLY VERIFIED  ✓✓✓
```

12b.10 Honest Open Items

FeedOracle's commitment to evidence integrity extends to disclosing what UVO does *not* do or has not yet finished.

- **The 9-layer pipeline is opinion-free for non-EU regulations.** A claim about CFTC, SEC, or MAS rules will receive `ABSTAIN` — UVO has no ground truth for them.
- **The 92.3% catch rate on the Sprint-9 fact-fuzz corpus is the empirical floor**, not the ceiling. Articles outside the rotated 6/day fuzzed set are statistically less hardened.
- **Audit bundles cap at 100 receipts.** A year of moderate use exceeds this; monthly batching is currently required. A streaming variant is planned.
- **The verifier CLI requires `pip install ecdsa`**. A pure-stdlib implementation is feasible but auditing trade-offs deferred this.
- **Free tier is rate-limited at 100 calls/day, burst 15.** No SLA on the free tier. Enterprise capacity reservation requires direct contact.

- **UVO does not retain raw input or output text** — only their `sha256` hashes via Module 7 receipts. Auditors verify hashes against material supplied by the calling agent.
- **Bring-your-own-regulation is not yet exposed.** Adding a customer's internal compliance policy as a private KG is technically straightforward (the ingestion pipeline accepts CELLAR-style XHTML, and an adapter for plain text exists in development) but not yet productized.

12b.11 Development History — Honestly Tracked

UVO was developed in 11 sprints across 7 phases. Each phase added a measurable capability and was closed only after publishing the corresponding evaluation results — including failures.

```
d5cb66a Phase 7: Standardization — OpenAPI, audit bundles, offline verifier CLI
e544a29 Phase 6: Public Proof Layer — landing page + interactive demo
d30fc60 Phase 5: DORA delegated RTS + RTS-cascade verification
ad3aab0 Sprint 11: Two-stage direction-check fixes Phase-4 false-positive
c1c021e Phase 4: GDPR + DSA ingestion + input-level KG anti-pattern
24c87b8 Sprint 10: Auto-Promote-Loop for the Knowledge Graph
b5d19fc Sprint 9: Fact-Adversarial Fuzzer + Direction-Aware
bdcbe3f Phase 3: EUR-Lex auto-ingestion
4475695 Phase 2: KG Fact Verifier
46a3244 Sprint 8: Multi-language coverage
806ae95 Sprint 7: Continuous Adversarial Hardening
8d596b6 Sprint 6: Adversarial auto-fuzzing
11bd8f3 Sprint 5: Semantic similarity override
c57a27c Sprint 4 A2 hotfix
6c7164b Sprint 3 adversarial proofs
d1745f4 Sprint 3 production proofs
e548c69 UVO Phase 1 Sprint 3
```

Bugs identified and fixed during development (a non-exhaustive subset documented in `NOTES.md` per phase):

- Sprint 9 catch-all fallback put two patterns into the wrong fact entry — diagnosed live during the first auto-promote run and fixed before the second.
- Phase 4 introduced false positives on paraphrases of true article content. Sprint 11's two-stage direction-check resolved this; true-paraphrase recall went from 70% to 100%.
- Phase 5 RTS cascade initially overrode high-confidence parent-article verdicts, falsely `BLOCK`ing true DORA Article 19 statements. Fixed by gating cascade on `UNCLEAR` / `PARTIAL` only.
- Phase 7 verifier CLI's first build used the wrong canonicalization (didn't strip `signature.signed_at`) and the wrong signature decoder (raw r||s instead of DER). Discovered by running the published binary against a real receipt; fixed before any external publication.
- Phase 8b nginx audit revealed 58 of 66 proxy_pass blocks were silently dropping `X-Forwarded-For`, causing all UVO receipts to record `observed_ip: 127.0.0.1`. Fixed by patching nginx and tightening the IP-extraction logic in the MCP base.

12b.12 Relationship to Other Modules

Module	Relationship
Module 3 — DORA Compliance	UVO is the verification layer that makes DORA claims falsifiable against the original DORA + RTS text.
Module 7 — Grounding & Evidence	Every UVO call emits a Module 7 Grounding Receipt; UVO is the first heavy production consumer of Module 7's infrastructure.
Module 6 — Agent Commerce	UVO is the canonical example of a high-value verification call that justifies x402 micropayment economics for autonomous agents.
Module 5 — Trust Passport	UVO's `verify_tool_usage` primitive (tool inventory snapshot) maps directly to the trust-passport requirement to enumerate available agent tools at call time.

12b.13 Operational Status

Component	Status	Notes
9-layer verification pipeline	LIVE	`uvo-pipeline.service` on port 14010
EUR-Lex ingestion (7 regulations)	LIVE	465 articles indexed from publications.europa.eu
Two nightly self-hardening crons	LIVE	01:00 UTC absolute-language, 03:00 UTC fact-KG
6 MCP tools at `/uvo/mcp/`	LIVE	Free-tier callable, 100/day rate-limited
Public landing + demo + dashboard + spec	LIVE	4 separate pages under `/uvo/`
Offline verifier CLI	LIVE	6.8 KB Python, downloadable from `/uvo/spec/`
Audit bundles (deterministic-hash evidence packages)	LIVE	Up to 100 receipts/bundle
Compliance report renderer	LIVE	Per-call_id human-readable output
Ongoing compute cost	≈ $25/year	Self-hardening + direction-check + RTS cascade Haiku calls
x402 paid tier wiring	NOT YET	Demand visible from polling agents; Phase 8a candidate
Bring-your-own-regulation	NOT YET	Pipeline ready; not productized

13. MCP Servers & AI Agent Access (UPDATED v6)

> *Major expansion since v5 — March 2026 numbers (100+ tools / 5+ servers) are now legacy*

13.1 Current Catalog

As of April 22, 2026, FeedOracle operates **103 production MCP servers** exposing **1,151+ tools** across seven thematic categories. The catalog spans far beyond the original compliance scope into blockchain, macro-economic data, GPU infrastructure, agent memory, e-commerce, and security operations.

Category	Servers	Approximate tool count
Compliance (DORA, MiCA, AMLR, CSRD, etc.)	11	203
Blockchain (13 chains: ETH, Polygon, Base, BNB, AVAX, Arb, Optimism, Solana, XRPL, Hedera, Aptos, Sui, TON, XLM)	14	~140
Macro & Market Intelligence	8	~120
Agent Infrastructure (Memory, Scheduler, Trust, Preflight, Conductor)	9	~95
Commerce & Productivity (Invoice, Job, Lead, Hotel, Flight, Shop)	18	~280
Security & Threat (CyberShield, HealthGuard, PredictionGuard)	7	~85
Specialty & Long-Tail (Movie, Sport, Weather, News, Meme, etc.)	36	~228

13.2 Two Distribution Surfaces

The MCP catalog is exposed through two coordinated but distinct surfaces:

FeedOracle (`mcp.feedoracle.io`) — the compliance-focused subset: - 11 servers, 203 tools - DORAOracle (95 tools across 6 layers — see Module 3) - AmpelOracle (50 tools, traffic-light overlay) - MiCAOracle, AMLOracle, ESMAOracle, EULawOracle, RegWatchOracle, MacroOracle (additional compliance and reference tools) - AgentGuard (24 tools, runtime policy engine — see Module 7) - All sit behind OAuth/KYA, x402 payment, and emit Grounding Receipts (Module 7)

ToolOracle (`tooloracle.io`) — the generalist marketplace: - 103 servers including all FeedOracle servers plus blockchain, GPU, macro, commerce, and long-tail - 1,151+ tools total - Single agent-pulse aggregator at `/.well-known/agent-pulse` - Single x402 payment gateway covering the entire catalog - ANP-discoverable agent directory at `/.well-known/agent-descriptions`

The two surfaces share the same wallet, billing, and Grounding Receipts infrastructure. A single agent passport on either surface receives free-tier credits for both.

13.3 Production Telemetry

Live numbers from the agent-pulse endpoint at the time of writing:

Metric	Value
Servers online	99
Tools available	1,179
Distinct external agents (last 24h)	67
MCP calls (last 24h)	4,553
Signal-layer hits (last 24h)	1,876
Mesh nodes	5

Sustained traffic over the trailing 30-day window: approximately 15,000 MCP calls per day, 600+ distinct external agent identities, 20+ countries by source IP.

13.4 Tool Discovery Patterns

External agents have organically discovered FeedOracle tools through three primary discovery paths:

1. **MCP `tools/list` enumeration** after a Remote MCP integration is configured by a user (Anthropic, OpenAI, xAI clients)
2. **ANP crawler discovery** of `/.well-known/agent-descriptions` (observed crawlers from Alibaba, AWS-hosted bots, ANP reference implementation)
3. **OracleNet pulse polling** — `/.well-known/agent-pulse` is fetched ~1,800 times per day by external agent infrastructure

The third path is the most distinctive: pulse-driven discovery is unique to FeedOracle's signal-layer architecture and produces follow-on tool calls within minutes of a pulse fetch.

13.5 Authentication Flexibility

All MCP servers accept three authentication modes (since the dual-auth patch on April 22, 2026):

- `X-API-Key: fo_…` — legacy header style
- `Authorization: Bearer fo_…` — RFC 6750 standard, used by Anthropic MCP Connector
- OAuth 2.1 with PKCE — full flow at `/oauth/*`, used by registered KYA agents

Anonymous calls are accepted for free-tier tools; payment-required calls return HTTP 402 with structured payment paths (see Module 6).

13.6 What Changed Since v5

- Tool count: **100+ → 1,151+** (11.5× growth)
- Server count: **5+ → 103**

- Categories: **1 (compliance) → 7**
- Distribution surfaces: **1 → 2 (FeedOracle + ToolOracle)**
- Auth methods: **1 (X-API-Key) → 3 (X-API-Key + Bearer + OAuth)**
- Per-call evidence: **none → Grounding Receipts on every successful call**

---

15. Architecture (UPDATED v6)

> *Substantial revision since v5 — Oracle Event Bus, OracleNet signal layer, Gemma LLM, Mesh Economics*

15.1 Layered View

The FeedOracle architecture has matured into a six-tier stack. Each tier is operated independently and can be scaled, replaced, or extended without touching the others.

15.2 The Oracle Event Bus

Introduced in v5.0 with 22 event types, the Event Bus has settled at **21 stabilized event types** with **9 documented cross-references** between oracles. Any oracle can publish; any oracle can subscribe.

Event types span three families:

- **Compliance lifecycle** — `finding.created`, `finding.escalated`, `finding.closed`, `assessment.completed`, `provider.flagged`, `contract.expired`
- **Economic events** — `wallet.topup`, `payment.x402`, `deal.executed`, `cache.refreshed`
- **Evidence events** — `receipt.signed`, `epm.published`, `dap.generated`, `anchor.confirmed`, `chain.head_advanced`

Four oracles currently publish: AmpelOracle (compliance lifecycle), x402 Gateway (economic), Receipts API (evidence), DealOracle (deal lifecycle). Subscribers include the dashboard, the Telegram pulse bot, the audit-trail builder, and downstream evidence aggregators.

15.3 OracleNet Signal Layer

OracleNet is the ambient signal layer: a 12-protocol stack of well-known endpoints that publish FeedOracle's state to the open agent ecosystem. Layers 1–11 cover discovery, capabilities, trust passports, behavioral baselines, and counter-intelligence. Layer 12 (Quantum Sorum) handles First Contact Detection — the moment an unrecognized external agent first interacts with the mesh.

The signal layer does not authenticate. It is intentionally readable by any crawler, AI client, or human curl invocation. The architectural premise: in an agent-first economy, discoverability is more valuable than gatekeeping at the discovery layer; gatekeeping happens at the payment layer (x402, Module 6) and the policy layer (AgentGuard, see below).

15.4 AgentGuard — Runtime Policy Engine

AgentGuard is the runtime policy enforcement plane for autonomous agents touching FeedOracle infrastructure. Unlike traditional API rate-limiters, AgentGuard tracks agent state through five well-defined transitions:

```
active → monitoring → approval_required → suspended → killed
```

Each transition is a recorded event on the Event Bus, signed and persisted. Policies are declarative (`policy_registry.json`) and cover budget caps, token-velocity limits, suspicious-pattern detection, and external-tool-call quotas.

AgentGuard exposes 24 MCP tools for runtime introspection and policy-management, runs on port 12001, and is the first gate in the autonomous agent stack:

```
Scheduler(10701) → AgentGuard(12001) → Preflight(10501) → Tool → Memory(10601)
```

15.5 Local LLM — Gemma 4 26B MoE

For sensitive workloads where outbound LLM calls are not acceptable (e.g., draft generation over confidential customer data), FeedOracle runs a local Gemma 4 26B Mixture-of-Experts model via

Ollama on port 11434. The model is invoked with the `"think": false` flag for deterministic compliance-relevant outputs.

LLMOracle (port 11480) wraps the local model and exposes it as MCP tools. Calls to LLMOracle never leave FeedOracle's infrastructure, never hit a third-party LLM provider, and produce a Grounding Receipt for every invocation — making local LLM workloads independently audit-grade.

15.6 Anchoring Strategy

The anchoring layer pursues **redundancy through diversity** rather than single-chain reliance:

Chain	Role	Cadence
Polygon	Primary EPM anchor + Chainlink Functions Subscription 185	Per-evidence-pack
Base	x402 USDC settlement	Per-payment
XRPL	Beacon v2.1, ZK attestation refresh	Weekly cron Sun 04:00 UTC
Hedera	High-throughput Hashgraph evidence anchor	Daily
Avalanche C-Chain	Contract `0x563D61F00124e1e6478a901Cd9F74cc29EEb6A71`	Per-attestation
Gnosis	Olas Mech service 2670, public agent presence	On-demand

If any single chain becomes unreliable or expensive, evidence anchoring continues across the remaining four. No customer-facing workflow depends on a single chain.

15.7 Storage Strategy

FeedOracle uses three storage classes with explicit immutability semantics:

- **Mutable** — wallet balances, agent state, current entity status (PostgreSQL + SQLite)
- **Append-only** — receipts, mesh-economics ledger, audit trails (SQLite with triggers)
- **Anchored-immutable** — published EPMs and DAP records (mainnet anchored)

The append-only and anchored-immutable layers are write-once. Once written, no FeedOracle operator (including root) can modify them through the application layer; modification would require unconstitutional database-level intervention which is journaled and would itself be detectable.

15.8 What Changed Since v5

- Six-tier architecture (was four-tier)
- Event Bus stabilized at 21 event types with cross-references
- OracleNet 12-protocol signal layer (was 6)

- AgentGuard formal state model with 24 MCP tools
- Local LLM (Gemma 4 26B MoE) integration
- Three storage classes formalized
- Anchoring expanded from 3 to 6 chains

---

16. Attestation & Evidence (UPDATED v6)

> *Cross-references Module 7 — Grounding Receipts now part of the evidence stack*

16.1 The Three Evidence Surfaces

FeedOracle now produces signed evidence at three semantically distinct surfaces:

Surface	Granularity	Format	Lifetime
EPM v1.0	Per-API-response	Evidence Pack Manifest with structured schema, ECDSA-signed	Anchored on Polygon
Grounding Receipt v0.1	Per-tool-call	Compact JSON with ES256K signature, chain-linked	Append-only SQLite + future blockchain anchor
DAP	Per-disclosure	Disclosure Attestation Proof for regulator filings	Anchored, regulator-shareable

The three surfaces are independently verifiable and cross-referenceable: a single regulated workflow can produce a Receipt for the tool call, an EPM for the response payload, and a DAP for the regulator filing — all with hash references between them.

16.2 Signing Infrastructure

All signatures use **ES256K (secp256k1)** with keys managed under `/root/rwa_node/mcp/ecdsa_signer.py`. The current production key is `feedoracle-mcp-es256k-1` and is published in JWKS format at:

- `https://feedoracle.io/.well-known/jwks.json`
- `https://feedoracle.io/receipts/jwks.json` (Receipts-specific mirror)

A second key (`feedoracle-mcp-es256k-2`) is staged for the next quarterly key rotation, scheduled for July 2026. Verifiers should always look up the current key set from JWKS rather than caching individual keys.

16.3 Anchoring

EPM and DAP records anchor to Polygon mainnet via Chainlink Functions Subscription 185 (5 LINK funded). Anchoring cadence:

- High-priority (DAP, regulator-facing): per-event, latency target < 60s
- Standard (EPM): batched, anchored every 6 hours
- Reserve (Receipts): currently SQLite-only; future versions of the spec define optional blockchain anchoring per receipt or per receipt-batch

The XRPL Beacon v2.1 provides a parallel zero-cost anchor channel for non-time-critical attestations, refreshed weekly.

16.4 Verifiability

Every evidence artifact in the FeedOracle system can be verified by a third party without contacting FeedOracle:

1. Fetch the JWKS from the published well-known URL
2. Reconstruct the canonical form of the signed payload (per RFC 8785 for Receipts, per EPM v1.0 schema for packs)
3. Verify the signature using the public key referenced by `kid`

For anchored artifacts, an additional optional check confirms the on-chain block hash matches the artifact's anchor reference.

16.5 Cross-Reference Table — All Evidence Components

Component	Module	Standard	Verification path
ECDSA-signed responses	This section	Custom	JWKS + signature verify
EPM v1.0	Module 5	EPM schema	JWKS + canonical reconstruct
Grounding Receipt v0.1	Module 7	RFC 8785 canonical JSON	JWKS + signature verify + chain walk
Mesh Economics ledger entry	Module 6	SQLite append-only	Read via `/.well-known/mesh-economics`
ZK Solvency Proof	Module 1	zk-SNARK	Verifier contract on-chain
DAP	Solution layer	Custom	Anchored artifact retrieval

---

18. Security & Operations (UPDATED v6)

> *External pentest and 8 security fixes since v5 — published April 2026*

18.1 External Pentest — April 20, 2026

A scripted external penetration test was conducted from a separate FeedOracle-controlled host (Gehirn server, `45.10.154.221`) against the production worker (`152.53.149.22`). The test was triggered by an application-grade security review related to the Anthropic and Avalanche partner intake processes.

The test surfaced eight security findings, all of which were remediated within 48 hours. A scripted re-pentest from Gehirn after the fixes confirmed **7 of 7 scripted fixes verified live** from external probe.

18.2 Findings and Remediation

#	Finding	Severity	Remediation
1	Credential leak in transcript-accessible logs	High	Logs scrubbed, key rotation policy formalized
2	nginx info-disclosure header in error responses	Low	Headers stripped via snippet `feedoracle-product-headers.conf`
3	Bot-scanner probes hitting `/wp-login.php`, `/xmlrpc.php`	Info	Honeypot canaries added, tarpit response
4	OAuth flow accepted CSRF token reuse within session window	Medium	Token nonce enforced, single-use confirmed
5	x402 payment endpoint accepted duplicate nonces	Medium	Nonce ledger added, replay rejected with 409
6	MCP `tools/call` did not propagate `X-Forwarded-For` to billing	Low	nginx config patched, real client IP recorded
7	Backup files (.bak, .old) accessible via direct request	Low	nginx `location ~* \.(bak|old|tmp|swp)$` block returns 404
8	Stale nginx workers from previous reload serving outdated config	Operational	Restart-not-reload policy adopted for new location blocks

Finding #1 led to a complete credential-rotation runbook at `/root/ops/key_rotation_20260422/ROTATION_RUNBOOK.md` with `verify_keys.sh` for post-rotation verification.

18.3 Authentication Hardening

The dual-auth patch deployed on April 22, 2026 enables MCP servers to accept three authentication modes uniformly:

- `X-API-Key: fo_…` — legacy header style
- `Authorization: Bearer fo_…` — RFC 6750, used by Anthropic MCP Connector
- OAuth 2.1 with PKCE — full flow, used by registered KYA agents

A `_dual_auth_done` flag in the base class prevents the OAuth fallback path from overwriting an authenticated `fo_` API-key hash, closing a class-of-bugs identified during patch development.

18.4 Honeypot & Counter-Intelligence

OracleNet Layer 8 (Honeypot) is integrated into the nginx config snippet `honeypot-canaries.conf`. Common attacker probe paths (`/wp-login.php`, `/xmlrpc.php`, `/.env`, `/.git/config`) return zero-length responses with logged attacker fingerprints. The aggregated probe-pattern data feeds the AgentGuard predictive threat-intel pipeline.

18.5 Operational Controls

Control	Implementation
TLS termination	Let's Encrypt managed by Certbot, auto-renew
HSTS	`max-age=31536000; includeSubDomains; preload`
Content-Security-Policy	Strict, no inline scripts except whitelisted style
Rate limiting	`limit_req` zones for API and MCP endpoints
Cloudflare real-IP propagation	`cloudflare-realip.conf` snippet
Service supervision	systemd with auto-restart for all 150+ services
Pulse-bot alerts	Telegram alerts on service-down events

18.6 Incident Response

For internal incidents (service degradation, data integrity events), the operational journal at `/root/whitelabel/JOURNAL.md` is the canonical record. Each significant change is appended with timestamp, change description, evidence, and rollback path.

For customer-affecting incidents under DORA Art. 17–19 obligations, IncidentOracle (Layer 6 of the DORA stack — see Module 3) automates the timeline construction, severity classification, and regulator-facing report generation.

18.7 What Changed Since v5

- External pentest performed (was: only internal review)
- 8 findings publicly documented and remediated
- Dual-auth deployed (was: X-API-Key only)
- Nginx restart-not-reload policy formalized after stale-worker incident
- Credential rotation runbook formalized
- Honeypot fully integrated into nginx config

---

22. Roadmap (REWRITTEN v6)

> *Replaces the v5 roadmap entirely — current priorities as of April 2026*

22.1 In-Flight (April–May 2026)

Grounding Receipts v0.1 → v1.0 The receipt format is published in v0.1 draft. Open questions on post-hoc receipt issuance, session-receipt linking without shared wallet, finer-grained verdict taxonomy, and proxy-server signing semantics are tracked. Target v1.0 freeze: end of Q2 2026, with reference implementation moved to public GitHub repository.

Anthropic MCP Connector — full E3 measurement integration Now that the dual-auth patch enables Anthropic native authentication, the Remote-MCP measurement program incorporates Anthropic as a first-class baseline. Cross-provider matrix (Section A in Appendix A) becomes the canonical reference for the format's positive contribution.

xAI disclosure window Responsible disclosure sent April 22, 2026 with 30-day window. Window expires May 22, 2026. Public whitepaper, blog post, and reference implementation publish after window closes (or after xAI acknowledges and proposes coordinated disclosure timing).

Mesh Economics v2 — TOCTOU hardening v1 has a known race condition on parallel identical fresh tool calls that can produce duplicate fresh-priced events. v2 adds optimistic locking on the cache-entry insert path. Target deployment: end of May 2026.

22.2 Next Quarter (Q3 2026)

DORA Production Phase DORA becomes binding January 17, 2025; the regulator-supervision phase starts in 2026. Q3 2026 is the projected sales catalyst window for FeedOracle's DORA stack. Operational priorities: customer-onboarding automation, regulator-facing report templating, evidence-bundle export to ITS-compliant XML.

Bittensor own subnet Earlier exploration on Bittensor (SN19, SN60) was discontinued — no existing subnet rewards FeedOracle's signed-evidence-with-MCP model. Long-term plan is FeedOracle's own subnet once sufficient TAO is accumulated. No deadline.

Receipt format adoption Beyond FeedOracle, the receipt format is offered as a community standard. Targets for outreach: MCP server operators in regulated domains (financial, medical, audit), MCP specification maintainers, LLM provider engineering teams.

22.3 Beyond Q3 2026

Standards engagement Once receipts v1.0 stabilizes, formal proposal to the MCP working group for `_meta.grounding_receipt` reservation. Parallel: proposal to relevant ISO and industry working groups (audit, regulated AI).

Federation — multi-operator receipts A federation extension allows multiple MCP server operators to issue receipts that link into the same chain. Useful for compositions where Operator A's tool calls Operator B's tool transitively.

Regulator partnerships Engagement with EU and national regulators (BaFin, CSSF, AMF) on the receipt format as evidence-class artifact for DORA/MiCA compliance reviews.

22.4 Explicitly Not on the Roadmap

To prevent feature creep and to keep the platform focused, the following are explicitly **not** on the roadmap:

- General-purpose LLM hosting beyond Gemma 4 26B for internal use
- Custom blockchain or layer-2 development (we use existing public chains)
- Consumer-facing product (FeedOracle is B2B / B2A — business-to-agent)
- Acquisition or merger (FeedOracle remains independent)

22.5 Operational Discipline

The roadmap is reviewed monthly against the journal at `/root/whitelabel/JOURNAL.md`. Items completed are marked done; items deprioritized are explicitly removed (not silently abandoned). The next monthly review is May 22, 2026 — coincident with the xAI disclosure-window expiry, which is the natural milestone for the next major communication cycle.

---

25. Glossary (UPDATED v6)

> *New entries since v5 in **bold***

A2A — Agent-to-Agent. Standard for inter-agent messaging and capability discovery. FeedOracle exposes A2A agent cards at `/.well-known/agent.json`.

AmpelOracle — FeedOracle's traffic-light-status overlay across DORA and MiCA controls. GREEN/YELLOW/RED/GREY scoring with article-level traceability.

Anchor — A blockchain transaction that commits a hash of an evidence artifact, making the artifact's existence and content tamper-evident.

ANP — Agent Network Protocol. JSON-LD-based standard for agent directory publication. FeedOracle implements at `/.well-known/agent-descriptions`.

AmpelOracle Bridge Workflow — The closed-loop process Finding → Owner → Re-test → Close, each step signed and journaled.

Append-only ledger — A storage system that supports only INSERT and SELECT. UPDATE and DELETE are blocked at the database layer (typically via SQLite triggers).

Canonical form (RFC 8785) — A deterministic JSON serialization where field order, whitespace, number formatting, and string encoding are normalized. Required for cryptographic signing of structured data.

Chain (Receipt chain) — The per-wallet sequence of Grounding Receipts linked by `prev_hash`. Each receipt cryptographically depends on its predecessor.

Chainlink Functions — Off-chain compute oracle used by FeedOracle to anchor evidence on Polygon (Subscription 185).

CTPP — Critical Third-Party Provider. DORA Article 31 designation for providers whose failure could systemically impact regulated entities.

DAP — **Disclosure Attestation Proof**. Per-disclosure-event evidence record with regulator-facing chain-of-custody.

DealOracle — FeedOracle's outbound autonomous deal-making engine. Discovers x402-compatible counterparties, evaluates offers, settles on Base in USDC.

DID — Decentralized Identifier (W3C). FeedOracle's primary DID is `did:web:feedoracle.io`.

DORA — EU Digital Operational Resilience Act (Regulation 2022/2554). Becomes binding January 17, 2025.

Dual-Auth — FeedOracle MCP server authentication mode that accepts both `X-API-Key` and `Authorization: Bearer` headers for `fo_` keys. Deployed April 22, 2026.

EPM — Evidence Pack Manifest. Per-API-response signed envelope wrapping the response payload with hashes, timestamps, and signature (v1.0 schema).

ES256K — ECDSA signature algorithm using the secp256k1 curve. FeedOracle's primary signing scheme (kid `feedoracle-mcp-es256k-1`).

Event Bus — FeedOracle's cross-oracle event-routing layer. 21 event types currently stable.

Free-tier tools — MCP tools that respond without payment for unauthenticated callers. Used to bootstrap KYA registration without requiring upfront payment.

Grounding Receipt — Per-tool-call signed receipt emitted by an MCP server, proving server-side execution. Spec v0.1 draft, FeedOracle reference implementation live as of April 22, 2026.

JWKS — JSON Web Key Set. Public-key catalog used for verifying signatures. FeedOracle JWKS at `/.well-known/jwks.json`.

KYA — Know Your Agent. FeedOracle's agent-registration system with trust-level scoring and welcome credits.

MCP — Model Context Protocol. Standard for tool discovery and invocation by LLM agents. Originated by Anthropic, broadly adopted.

Mesh Economics Ledger — Append-only SQLite ledger of all priced operations across the FeedOracle MCP mesh. 6 tables, 3 immutability triggers, deployed April 19, 2026.

Olas Mech — Permissionless on-chain agent infrastructure on Gnosis. FeedOracle operates Service 2670 (`0x27212a38c76Ab600D73059aB4E8e7540A67ff0F6`).

OracleNet — FeedOracle's signal layer. 12-protocol stack of well-known endpoints (agent.json, agent-pulse, agent-descriptions, etc.) for ambient agent-ecosystem visibility.

Pre-flight — Policy-gate evaluation performed before tool execution. PreflightOracle (port 10501) runs the pre-flight stage.

prev_hash — The field in `anchor.prev_hash` of a Grounding Receipt that references the hash of the predecessor receipt for the same wallet.

Receipt store — Append-only SQLite database persisting all emitted Grounding Receipts at `/root/whitelabel/shared/receipts/receipts.db`.

Remote MCP — LLM-provider-side feature where the provider's API includes built-in MCP-client support (Anthropic MCP Connector, OpenAI MCP tool, xAI mcp() tool).

RoI — Register of Information. DORA Article 28 mandatory ICT-third-party register, ITS-compliant export.

RTO/RPO — Recovery Time Objective / Recovery Point Objective. DORA Article 11 resilience targets.

TIBER-EU — Threat Intelligence-Based Ethical Red-teaming. Advanced testing framework referenced by DORA Articles 24–27.

TOCTOU — Time-of-check-to-time-of-use. Class of race conditions where a check passes but state changes before the dependent action. Mesh Economics v1 has a known TOCTOU on parallel fresh tool calls; v2 adds optimistic locking.

Verdict — In a Grounding Receipt, the categorical outcome of the tool call: `executed`, `rejected_auth`, `rejected_policy`, `rejected_payment`, or `error_server`.

x402 — HTTP 402 (Payment Required) protocol. Coinbase-originated standard for machine-readable payment terms in HTTP responses. FeedOracle uses USDC settlement on Base.

ZK Solvency — Zero-knowledge proof that a stablecoin issuer's reserves cover obligations, without revealing exact reserve amounts.

Appendix A — Labs / Research: Remote-MCP Execution Measurement

> *New in v6 — measurement study performed April 2026*

A.1 Background

In the course of validating FeedOracle's server-side tool-invocation audit trail, a measurement program was established to verify that LLM-provider Remote MCP integrations route `tools/call` requests to MCP servers in ways that are HTTP-observable at the server. The program ran from April 19–22, 2026 and produced 253 controlled measurement runs across three major LLM providers.

This appendix is a condensed summary. The full research bundle (methodology, raw artefacts, network capture, harness source) is at `/research/remote-mcp-execution-gap` on the FeedOracle internal server and will be published after the responsible-disclosure window closes (May 22, 2026 or earlier on coordinated disclosure).

A.2 Setup

Three MCP servers operated by FeedOracle were used as targets: PreflightOracle (`/preflight/mcp/`), MiCAOracle (`/mica/mcp/`), and ResearchOracle (`/research/mcp/`). All three are production servers serving regular external traffic.

Three LLM-provider Remote MCP integrations were measured: - xAI Grok (via `xai-sdk` `mcp()` tool) - OpenAI (via Responses API MCP tool) - Anthropic (via Messages API MCP Connector)

Each provider was driven by an identical task prompt instructing the LLM to call specific compliance tools and produce a structured verdict.

Server-side instrumentation: a probe logger added to FeedOracle's MCP base class records every incoming `tools/call` HTTP POST with source IP, User-Agent, and authentication header. nginx access logs and a tcpdump TLS capture provided independent confirmation channels.

A.3 Two Modes

Each provider ran under two server-side conditions:

- **baseline** — server responds to all MCP methods normally
- **block** — server returns HTTP 503 to `tools/call` while remaining responsive to `tools/list` and other methods (deliberate failure injection)

The intent of the block mode was to test whether each provider's response stream would surface server-side failures honestly to the user.

A.4 Aggregate Results

Provider	Mode	N	Avg HTTP `tools/call` hits at server	% runs with real evidence in final text	% runs where LLM admits failure
xAI Grok	baseline	5	0.0	0%	—
xAI Grok	block	5	0.0	0%	0%
OpenAI	baseline	5	6.0	100%	—
OpenAI	block	5	12.0 (retries)	0%	100%
Anthropic	baseline	10	9.8	100%	—
Anthropic	block	10	4.4	0%	0% (silent empty response)

The xAI baseline measurement was scaled to **N=200 sequential runs** to rule out infrastructure noise. Result: 200 of 200 runs showed the same pattern (zero `tools/call` HTTP POSTs at the server while the response stream reported `closed_loop: YES`). Total cost: $2.82, duration: 41.5 minutes.

A.5 Three Distinct Patterns Under Block

The block-mode experiment identified three categorically different behaviors when a server returns HTTP 503 to `tools/call`:

1. **Grok** — continues to report `closed_loop: YES` and produces a plausible verdict (no acknowledgement of server failure)
2. **OpenAI** — marks tool results as `error: true` and explicitly tells the user "unable to perform"
3. **Anthropic** — stops generating with `stop_reason: null` and empty text content (silent failure)

Only OpenAI surfaces the server-side failure to the user in a way that meets typical UX expectations. Only Grok diverges from HTTP-observability of the underlying request path.

A.6 Interpretation

The measurement is reported as observed divergence between LLM-stream-reported `tools/call` and HTTP-observable `tools/call` at the server, not as a claim about implementation intent. Possible explanations span the range from gateway-side optimization, internal Remote MCP architecture differences, to harness-configuration issues on the measuring side.

What the measurement establishes regardless of cause:

- An MCP server operator currently has no standard way to verify that any given LLM-reported tool call corresponds to an actual server-side execution
- The three providers expose three categorically different behaviors under the same condition

- This gap motivates the Grounding Receipts format (Module 7) as a server-operator solution that does not depend on any provider's cooperation

A.7 Disclosure

The measurement study was responsibly disclosed to xAI on April 22, 2026 with a 30-day window. The disclosure email proposed publication after May 22, 2026, with flexibility for extended timing if requested by xAI. The disclosure framed the receipt format as the more important contribution and invited xAI's feedback specifically on the format.

OpenAI and Anthropic's behaviors are technically correct interpretations of the MCP specification and do not require disclosure as security or implementation issues. The Anthropic baseline measurement was repeated on April 22, 2026 after the dual-auth patch removed an authentication-format mismatch on FeedOracle's side; the corrected results (n=10 per cell) are included in the matrix above.

A.8 Citation Note

Full citation when this study is referenced externally (after publication):

> *Keskin, M. et al. "Reproducible divergence between reported and HTTP-observable MCP tool execution across LLM providers." FeedOracle Technologies Technical Report, 2026.*
