

WHITEPAPER · VERSION 8.0

AgentNOMOS — The Machine Economy Trust Chain

Verifiable governance between agent intent and consequential action

FeedOracle Technologies · Germany

Version 8.0 · Published 6 October 2026 · Current-State Release

Contents

1. Document Control and Claim Policy	4
1.1 Claim rules	5
1.2 Claim boundaries that hold everywhere in this document	5
2. Executive Summary	6
3. The Trust-Continuity Problem	7
4. AgentNOMOS System Architecture	8
4.1 One product, one core	8
4.2 Specialized surfaces around the core	8
4.3 Hosts and responsibilities	9
5. The NOMOS Machine Economy Trust Chain	9
6. How AgentNOMOS Works	10
6.1 Receiving fresh and attributable information (S0)	11
6.2 Establishing presence, identity and capability (S1–S3)	11
6.3 Binding intent to an exact offer (S4–S5)	11
6.4 Authorizing consequential action (S6)	12
6.5 Governing execution and delivery (S7)	12
6.6 Proving what was executed and returned (S8)	12
6.7 Recording what was observed afterwards (S9)	12
6.8 Observing what a stricter policy would have decided (S10)	13
7. Evidence and Admission	13
7.1 Security evidence from an external authority, enforced	13
7.2 The admission policy as a versioned, two-host object	13
7.3 What the admission evidence does not show	14
8. Identity, Capability and Compatibility	14
8.1 The canonical manifest, deployed	14
9. Authorization and Fail-Closed Behaviour	15
10. Controlled Execution and the OpenShell Boundary	16
10.1 Why an execution boundary	16
10.2 The R1.4I lineage: final and frozen	16
10.3 Trust Chain × OpenShell S7: live and verified, within scope	16
10.4 What the boundary does not claim	17
11. Proof, Receipts and the Outcome Boundary	17
12. Public Reproducibility	18
13. Interoperability and External Operator Evidence	19
13.1 External participant call through the production boundary (October 2026)	19
13.2 A human-assisted external full-chain traversal (August 2026)	20
13.3 A cross-operator lane on which the caller holder-signs (August 2026, on hold)	21
13.4 Conformance, audit and off-host evidence (August 2026, historical)	21
13.5 The boundary of this entire section	21
14. Machine Commerce and Payment Rails	22
14.1 Payment is settlement, not authority	22
14.2 Inbound rails and what they sell	22
14.3 Revenue, as a dated snapshot	22

14.4 Outbound purchasing	23
15. Research Governance and Bounded Autonomy	23
15.1 Why governance reaches research	23
15.2 The NOMOS Research Governor	24
15.3 Two research campaigns as governance examples	24
15.4 AutoNOMOS: the bounded economic continuation, held for autonomous economic operation	24
16. What Has Actually Been Demonstrated	25
17. What Is Not Yet Proven	27
18. Security and Operational Boundaries	28
18.1 Principles that are enforced	28
18.2 Open findings as of 6 October 2026	29
18.3 Why this section exists in a trust whitepaper	29
19. Commercial and Integration Model	29
20. Roadmap	30
21. Evidence Index and Dynamic Sources	31
22. Public-Surface Consistency Statement	33
23. Legal Disclaimer	34
23.1 Regulatory context, without overclaiming	34
23.2 Disclaimer	35

1. Document Control and Claim Policy

Field	Value
Document	AgentNOMOS Whitepaper
Version	8.0 Current-State Release
Supersedes	Version 7.2 Draft R2 (19 August 2026), the operator-signed-off textual baseline, which was never published. Version 7.1 and earlier remain historical only.
Observation date	2026-10-06 . Statements re-measured for this version were read on that date (approximately 07:30–07:50Z, and re-measured at release closure, 09:07–09:10Z) using read-only access. Statements carried from sealed checkpoints keep their own dates, which are given where they matter.
Source order	Source Refresh R4 (2026-10-06) → Source Refresh R3 (2026-10-02) → v7.2 Draft R2 and its traceability → frozen architectural invariants → frozen OpenShell R1.4I evidence → archived Classiq record. Newer evidence overrides older evidence.
Status	Publication status: Published 6 October 2026.
Decisions applied	2026-10-06 — operator decisions D-1 to D-7: external participant not named; security findings disclosed by status and category, not exploit detail; manifest scope wording repaired at its single-writer builder (done 6 October 2026); AutoNOMOS stated as held for autonomous economic operation, with its experimental runtime components classified; external seller not named; security-assessment evidence kept concise; revenue figure kept, public economics outputs reconciled with the ledger (done 6 October 2026)
Release-substitution slots	The four release slots (Sections 5, 14.3, 22 and 23.1) were filled on 6 October 2026 after the public surfaces were synchronized and the regulatory citations were verified against the Official Journal texts.
Publisher	FeedOracle Technologies, Germany
Claim ceiling	EXECUTION_BOUND_PROOF (defined in Section 11). Unchanged from version 7.2. No section of this document raises it.
Publicly verifiable stages	S3–S8 only , per the capability manifest's own <code>publicly_verifiable</code> field (Section 5)

Version 8.0 is a current-state release of the version 7.2 text, not a rewrite. The model, the eleven-stage chain and the claim discipline stand. What is new is seven weeks of sealed evidence and a more honest picture of the whole system:

- the canonical capability manifest is deployed, byte-equal between the public URL and the local file, and passes its validator;
- an **independently operated external participant** made a governed call through the production boundary, the call was corroborated against server-side records, and the temporary authority it used was then revoked and removed from both enforcing hosts;
- the **controlled-execution boundary** of the frozen OpenShell R1.4I lineage is now published as a scoped sub-claim of stage S7, with an explicit list of what it does not claim;
- an external participant completed a **human-assisted external full-chain traversal** of the public challenge capability, which remains open because the run does not meet the unassisted class;
- governance now reaches **research and spend decisions**, through an offline Research Governor that runs in shadow mode only;
- and the **security and operational findings** that are open today are listed here, in Section 18, rather than left out.

The structural correction of the version 7 generation still runs through the whole text and is stated in Section 11: **the execution receipt does not bind the outcome**.

1.1 Claim rules

Every factual claim follows twelve rules. Rules 1–9 are carried unchanged from version 7.2. Rules 10–12 are added in version 8.0 for the states and evidence classes it introduces.

1. **Numbers are snapshots.** Any quantity is labelled with its observation time and source. Live figures live in canonical machine-readable files (Section 21) and are not frozen here.
2. **Mechanisms, not superlatives.** Storage and evidence guarantees are named by mechanism — *append-only*, *hash-chained*, *signed*, *anchored*, *application-layer write-protected* — never as "immutable".
3. **Scope travels with the claim.** "Proven" always names the route, cohort, date and vantage point. A capability proven on one route is never claimed for a product portfolio.
4. **Verification paths are stated.** Where a third party can check a claim from published material, the path is given. Where verification currently requires the operator's own vantage point, that limit is stated in plain language.
5. **Absence is disclosed.** Section 17 lists what has *not* been proven, and Section 18 lists what is open. Both are part of the product description, not appendices.
6. **Evidence classes are never blurred.** Every substantive statement carries one of three classes: *cryptographic* (a stranger can check a signature or an on-chain fact), *recomputable* (a stranger can recompute a digest or a rule outcome from published bytes), or *inspectable* (the record exists and is consistent, but confirming it needs operator-held data). The first two may be called **verified** or **proven** when the scope and vantage are named alongside them. **Inspectable is never described as proof.**
7. **Conformance results are never upgraded to certification.** An external conformance suite's result is reported with its exact per-module outcome split and its certification status, verbatim.
8. **An external audit or measurement is a measurement.** A tool-based audit result is reported as a measurement of the audited surface at a date, never as a certification, an endorsement or a statement about anything outside the audited surface.
9. **A boundary refusal is not a policy decision.** Where a transport, identity or platform layer refuses before the governance policy is evaluated, that refusal is reported as a boundary error, never as a governed DENY.
10. **State labels are never upgraded.** A component labelled SHADOW, HELD, STAGED, BUILT or research is described with that label. Only states measured live are described as live, and only governed production paths are described as production.
11. **Research is not product.** A research result — a circuit, a benchmark candidate, a replay of a campaign ledger — is reported as research, with its submission state. It is never presented as a competition result, a victory, a product capability or a certification.
12. **A payment attempt is not a payment, and a 402 is not revenue.** An HTTP 402 `payment_required` response, a signed payment authorization that was not settled, and a payment that was verified but not settled are never counted as revenue or described as a purchase. Revenue figures are dated snapshots with their attribution qualification.

1.2 Claim boundaries that hold everywhere in this document

These separations are part of the model. Each is used at least once below, and none is collapsed anywhere.

Boundary	Meaning
technically valid ≠ authorized ≠ intended ≠ proven	A well-formed request is not a permitted one; a permitted one is not necessarily what the principal meant; and none of these is proof that it happened.
Binding ≠ Authorization	A digest that ties two objects together does not grant permission for either.
Payment ≠ Authorization	Settling an invoice grants no authority; an authorization never implies that funds moved.
Authorized Action ≠ Actual Action	What was permitted and what ran are two facts. The execution boundary exists to compare them.

Boundary	Meaning
Execution ≠ Outcome	A signed record of what ran and what bytes came back is not a statement that the result was good.
Receipt ≠ truth of provider output	A receipt proves which bytes were returned under which decision, not that those bytes are factually correct.
Corroboration ≠ correctness	An external party agreeing with our records confirms consistency, not correctness.
Network identity ≠ authority	Holding a key, a hostname or a wallet does not confer the right to act.
Research result ≠ product certification	A research artifact proves a property of the artifact, not of the product.
SHADOW ≠ production	A shadow component records what it would decide; it decides nothing.

The NOMOS Machine Economy Trust Chain described here is a NOMOS reference model. It is not a W3C, A2A, MCP, x402 or XRPL standard, and no external standards body has endorsed it.

On disagreement with the public surfaces. Where this document and a canonical machine-readable source disagree, the machine-readable source is the live contract and this document is a dated description. Several public surfaces still carry simplified or older-generation phrasing that this version corrects. Those divergences are measured, enumerated and described in Section 22.

2. Executive Summary

Autonomous agents can already discover services, negotiate terms, pay for access and consume results without a human in the loop. **Common agent, tool and payment protocols do not by themselves provide one bound evidence chain across this sequence.** Identity systems authenticate a caller but say nothing about whether the specific action was authorized. Payment rails settle value but do not bind the payment to the promised delivery. Runtimes execute what they are given without knowing what was permitted. Logs record what an operator chose to record and can be rewritten by whoever holds the machine.

AgentNOMOS is FeedOracle Technologies' answer to that gap: a governance and evidence layer for the path from an agent's first contact to the consequences of its actions. Its public model is the **NOMOS Machine Economy Trust Chain** — eleven stages, S0 through S10, in which an agent must receive fresh information, establish who and what it is dealing with, determine what is permitted, bind authorization to the exact action, execute under that authorization, prove what happened, and record the result so it can inform what comes next.

What distinguishes this whitepaper is the claim discipline. As of **6 October 2026**:

- **Eight production routes** run under governance binding — admission decisions, signed execution receipts and outcome records. The append-only execution store holds 1,254 executions and 1,253 signed receipts across exactly those eight routes and no others.
- **Anyone can reproduce the proof without asking us.** The published stateless verifier, byte-identical to its 13 August release, was re-run against the live surfaces on the observation date: **21 checks, 21 pass, 0 fail** — 16 recomputable, 3 cryptographic and 2 inspectable.
- **The canonical capability manifest is deployed.** The public manifest is byte-equal to the locally generated file, its validator passes **167 of 167** checks against the public URL, and its admission policy reference matches the enforcing configuration: **17 grants**, no temporary external-test grant.
- **An independently operated external participant made a governed call through the production boundary.** On 3 October 2026 infrastructure controlled by that participant called a governed route under a temporary, single-purpose test credential and received HTTP 200 with a signed execution receipt. Its own verification record was archived and cross-checked field by field against our server-side records. The credential was then revoked and the temporary grant removed from both enforcing hosts. This is **external corroboration of one test call**, not adoption and not endorsement (Section 13).

- **The execution boundary is published, scoped and bounded.** For the frozen OpenShell R1.4I lineage, AgentNOMOS publishes a scoped S7 sub-claim — *software-attested mediated runtime action and effect binding* — with the host and root supervisor inside the trusted computing base and no TPM, TEE, measured-boot or NVIDIA certification claim. OpenShell is **not** the runtime of the production route cohort (Section 10).
- **Security evidence from an external public authority is enforced inside the authorization decision** for one of the eight governed routes (Section 7.1). It is not system-wide security verification.
- **A paid external full-chain traversal exists — consented, human-assisted, Class B.** On 24 August 2026 an external participant's agent, started by a human, discovered the public full-chain capability, paid 0.001 USDC on Base from its own wallet, and received and verified a signed receipt. It does not satisfy the still-open challenge for an unassisted external traversal and does not establish organic autonomous adoption (Section 13.2).
- **Governance now reaches research and spend.** The NOMOS Research Governor records research campaigns as hash-chained event ledgers — hypotheses, closed routes, corrections, budgets, spend approvals and reopening decisions — and runs in **shadow mode only**: no execution, no blocking, no authority (Section 15).

Equally important is what AgentNOMOS does **not** claim. No universal closure of all eleven stages for arbitrary agents. No independent third-party verification domain. No production autonomy: AutoNOMOS, the economic continuation of NOMOS, is **held for autonomous economic operation**; its experimental runtime components run without spending or submission authority (Section 15.4). No completed outbound external purchase: the one approved outbound attempt was refused by an external x402 seller and moved **0 USDC**. Revenue is small and of uncertain genuineness — **1.451000 USDC external FINAL from 62 events and 36 wallets**, a snapshot of 6 October 2026; 36 wallets are not 36 verified customers. No certification of any kind. The receipt **does not cryptographically bind the outcome**. And the system has **open security findings** today, listed in Section 18; AgentNOMOS is not described in this document as generally secure, certified, compliant or audit-ready.

Govern before action. Verify execution. Record outcomes. — as an engineering program, and described here only as far as it has actually been measured.

3. The Trust-Continuity Problem

Consider the moment an autonomous agent is about to do something consequential: call a paid API, sign a message, move funds, change a production system. Between the agent's *intent* and the *consequence* lies a chain of implicit assumptions:

- the information the agent acted on was **fresh**, not replayed or stale;
- the counterparty **exists** now and is **who** it claims to be;
- the counterparty can actually **do** what it advertises;
- the action matches what the agent actually **intended**, on the **terms offered**;
- something with the authority to do so **authorized** this exact action;
- the action was **executed** as authorized — not silently dropped, not substituted, not widened at runtime;
- there is **proof** of what happened that does not depend on trusting the executor's word;
- the **result** was compared against the promise;
- and that record is available to inform the next decision.

The protocols in common use each solve a slice. A2A describes agents to each other. MCP invokes tools. x402 discovers payment terms and settles. Runtimes and sandboxes contain execution. Ledgers anchor data. In the deployments we have examined, none of these carries the *same piece of evidence* across all of these steps, and each hop tends to discard the guarantees of the previous one: the payment does not reference the authorization; the runtime does not know what was authorized; the log entry does not reference the payment; the "success" report is written by the party being paid. This is an observation about how these protocols are specified and commonly deployed, not a proof about every system that exists.

Two gaps deserve names. The first is the **authorization gap**: a request can be technically valid without being authorized, and authorized without being what the principal intended. The second is the **execution gap**: *Authorized Action* $\neq$ *Actual Action*. A permission that was granted for one action says nothing, on its own, about what the runtime then did.

The result is what we call the **trust-continuity problem**: at the moment of consequence, no single verifiable artifact answers "who allowed exactly this, on what basis, and what was actually executed and returned?" — answerable later, by a party who was not present, without trusting the operator's memory.

AgentNOMOS exists to produce that artifact, and to be honest about where the artifact's guarantee ends.

4. AgentNOMOS System Architecture

4.1 One product, one core

AgentNOMOS is the product of **FeedOracle Technologies**: governance, authorization and evidence for consequential agent actions. Its core is a sequence of functions through which an action on the governed route cohort passes:

Core function	What it establishes
Discovery	Machine-readable self-description: agent cards, capability manifest, compatibility map, purchase manifest
Identity	Who the caller and the counterparty are, and how strongly that is established
Capability	What a route can actually do, derived from installed code, not from its name
Intent	What exactly the agent wants to do, as a structured, digest-bound document
Offer	The exact terms on which the counterparty is willing
Evidence	Fresh, provenance-bound inputs to the decision, including external security evidence
Authorization	A signed decision by a separate authority: this intent, against this offer, now
Controlled Execution	The action — and only that action — carried out under the decision
Receipt	A signed record binding the decision, the execution identity and the response bytes
Outcome	What was observed afterwards, recorded append-only and kept separate from proof
Governance	What a stricter policy would have decided, recorded but not enforced; and, since October 2026, governance of research and spend decisions in shadow mode

These functions map onto the eleven stages of Section 5. The map is not one-to-one, because the stages are the canonical public model and the functions are a reading aid.

4.2 Specialized surfaces around the core

AgentNOMOS has public surfaces that predate the unified product name and still carry live endpoints. They are access points to the same system, each with one job. None of them is a second source of authority.

Surface	Role	What it is not
ToolOracle	Discovery and invocation: tool and capability catalogs, governed invocation endpoints, the payment-gated full-chain capability	Not an authority; listing a tool does not authorize its use
FeedOracle	Evidence and data products: provenance-bound data, evidence bundles, paid data calls	Not a guarantee of upstream truth; a served value is evidence of what was served
OracleNet	Presence, freshness and signal transport, including transport of external security evidence	Not a decision-maker; it carries signals, it does not decide
NOMOS / AgentNOMOS	Authority, policy, admission, governance and proof	The only component that authorizes; it does not execute the governed workload itself
OpenShell	A controlled runtime execution boundary, used for one frozen, scoped lineage (Section 10)	Not the runtime of the production route cohort; not a certified or hardware-attested enclave
x402 and other payment rails	Payment terms and settlement	Not authority. Payment ≠ Authorization

The rule that holds this together is simple to state: **only NOMOS authorizes**. Discovery surfaces describe, evidence surfaces inform, transport surfaces carry, runtimes contain, and rails settle — and none of those acts, on its own, permits anything.

4.3 Hosts and responsibilities

The system is operated by a single operator across a workload host, a separate authority host and, for the AutoNOMOS line, a cloud account. The workload host holds no decision-signing key; decisions are signed on the authority host and validated at the gate on the workload host. This document describes these components by function and omits internal addresses, ports, paths and key locations (Section 18).

AgentNOMOS is verifiable, non-sentient infrastructure. It has no goals of its own, no hidden persona, and nothing in this document attributes agency, awareness or judgment to it beyond the deterministic evaluation of policy over evidence.

5. The NOMOS Machine Economy Trust Chain

The chain has eleven stages. The stage names below are reproduced **verbatim from the canonical machine-readable capability manifest** (Section 8), which declares `stage_count: 11` and `stage_range: S0-S10`; both were re-read on 6 October 2026 from the manifest generated on 3 October 2026.

The final column states, per stage, whether an outside party can check that stage from published material. It is taken from the manifest's own `publicly_verifiable` field, not from our judgement about it.

Stage	Canonical name	The question it answers	Can an outside party check it?
S0	Frequency / Freshness	Is the information this decision rests on recent enough, measured at decision time?	No — operator-held enforcing input
S1	Presence	Is the counterparty actually there now?	No — operator-held enforcing input
S2	Identity	Does the caller present a credential the keystore recognises?	No — shared secret over TLS; the caller signs nothing
S3	Capability	Can it actually do what it advertises?	Yes — the capability id is bound inside the signed receipt
S4	Intent / Task Fit	What exactly does the agent want, and does it fit this route?	Yes — the caller rebuilds the intent and match documents and compares digests
S5	Offer	On what exact terms is the counterparty willing?	Yes — offer digest in the receipt, compared with the published per-route digest

Stage	Canonical name	The question it answers	Can an outside party check it?
S6	Deal / Authorization	Does policy permit <i>this</i> intent on <i>these</i> terms — decided and signed?	Yes, the binding — the decision document itself is not published
S7	Execution / Delivery	Was the authorized action — and only it — carried out and delivered?	Yes — the receipt verifies against the published key set. The separate OpenShell execution-boundary sub-claim on S7 is not publicly verifiable (Section 10)
S8	Proof / Verifiability	Can a party that was not present verify what was executed and returned?	Yes, for the call — the subject-level anchor is a separate, unpublished proof
S9	Reliability / Outcomes	What was observed afterwards, recorded append-only?	No — post-hoc, inspectable only
S10	Governance / Immune Control	What would a stricter policy have decided — recorded, not enforced?	No — observational, shadow-only

The chain has **three bands**, and collapsing them would be the easiest way to mislead a reader:

- **S0–S2 — operator-held enforcing inputs.** They are real, and they refuse: an anchor older than the configured window refuses the request at admission rather than annotating it, an unsigned or missing presence scan is a refusal rather than a default, and an unrecognised or revoked caller credential is rejected. But their evidence is retrievable by the operator only. **No part of S0–S2 is checkable from published material.**
- **S3–S8 — the published, reproducible band.** These are the stages a stranger can actually check, because each of them leaves something inside the Ed25519-signed execution receipt that verifies against the published key set, or a digest that can be recomputed from published bytes. Section 12 describes the tool that does it.
- **S9–S10 — recorded, not proven.** S9 is a post-hoc, append-only observation. S10 records what a stricter policy would have decided. Neither is bound under the receipt signature and neither changes a production decision.

Four further properties of the model deserve emphasis.

It is a reference model. The chain describes the complete set of questions. It does not claim that every stage is closed in production for every route. The boundary this document uses is stated here in full, and it governs everywhere else:

The S0–S10 reference model applies to the named route cohort as described. Public verifiability is limited to S3–S8; S0–S2 are operator-held, S9 is post-hoc and S10 is observational.

The capability manifest states the same boundary in its own scope wording: execution-bound proof for the named route cohort, with public verifiability limited to S3–S8.

Stage numbering is not a progress bar. A route can have strong S7/S8 evidence while its S0 input hardening covers only part of the cohort — which is the actual state today (Sections 7 and 17).

External checkability is narrower than the chain. Six of the eleven stages produce something a stranger can check or recompute. The chain is eleven stages long; the *publicly verifiable* part of it is six.

Decisions fail closed. Where a stage cannot produce its evidence — a stale source, an unavailable authority, a retired key, a revoked credential, a policy version the enforcing side does not recognize — the affected path refuses rather than defaulting to permission. Section 9 gives the sealed cases in which the refusal actually fired.

6. How AgentNOMOS Works

This section walks the product by function. Each subsection corresponds to stages of the chain.

6.1 Receiving fresh and attributable information (S0)

Every governed decision starts from signals: ledger anchors, capability catalogs, security advisories, liveness probes. Two disciplines apply. **Freshness is recomputed, not stored** — the age of the newest evidence is measured against the clock at decision time; a cached "fresh" flag is never read back. **Sources carry provenance** — for external inputs the product records what was fetched, when, from where, the exact bytes' digest and the catalog version, so that any downstream claim can be traced to a retained snapshot rather than to an assertion. Section 7 describes the external security evidence that has been an enforced input since 10 August 2026.

6.2 Establishing presence, identity and capability (S1–S3)

Counterparties are described by signed, digest-bound machine artifacts (Section 8). Presence is probed, not presumed: the subject identifier must resolve to a signed presence record, and an unsigned or missing one is a refusal rather than a default.

Identity needs to be stated in two parts, because they are not equally strong.

- **Publisher and agent identity artifacts** — agent cards, the capability manifest, the signed operator identity and delegation document — can be signed and checked against published keys. That is a real, externally checkable property of the *descriptions* the product publishes about itself.
- **Caller authentication at S2, on the governed cohort as it runs today, is a shared secret over TLS.** The caller presents a credential the keystore recognises. **The caller signs nothing.** There is no holder-key signature and therefore no proof of possession anyone could re-verify afterwards. S2 is an *enforcing* control — an unrecognised or revoked credential is rejected — but it is **not a publicly reproducible cryptographic identity check**. The external participant's test call of October 2026 (Section 13) used exactly this mode: a temporary shared-secret credential, revoked afterwards.

A second, narrower path exists and must not be blurred with the first. On **one production partner lane** — a cross-operator A2A boundary brought live on 18 August 2026 — the calling platform authenticates by an **ES256K signature it produces from a key it holds** (a signed token whose replay identifier is burned on use). On that lane, and only on that lane, caller authentication is a holder-key signature. The lane is on operator hold (Section 13). Roadmap item 3 — holder-key authentication at S2 on the cohort's consumer path — is therefore partially advanced, not closed.

Key state matters wherever signing keys *are* used: a signature from a retired key is acceptable for verifying history, never for authorizing something new. And **network identity is not authority**: a hostname, a wallet address or a valid credential identifies a party; only a signed decision under the current policy permits an action.

Capability claims are checked against what a counterparty's published contract actually exposes. The product derives its own route capabilities from the installed code by static analysis and publishes the digest of that derivation, so the capability map itself is a checkable artifact rather than a hand-maintained list. A route that matches no rule is recorded as capability-unknown, never guessed from its name.

6.3 Binding intent to an exact offer (S4–S5)

An intent is a structured statement of what the agent wants to do; an offer is the counterparty's structured terms. AgentNOMOS binds the two by digest into the authorization request, so what gets authorized is not "agent may call service" but "this intent, against this offer, at this time". Any later substitution — different tool, different terms, different payload — breaks the binding and is detectable. Binding, however, is not authorization: a digest ties objects together and permits nothing.

Task fit is evaluated deterministically before anything else happens, and a mismatch is **recorded evidence, not a dropped request**. Section 12 describes the published example.

6.4 Authorizing consequential action (S6)

Authorization is a decision by a dedicated authority component, separated from the services it governs: the deciding key material does not live where the workload runs, and the workload host cannot mint its own permissions. That separation was measured from the constrained side, and when a deployment step required a capability the workload host does not have, the step was refused rather than the boundary loosened.

Decisions are signed. The signed core carries the digests of the intent and the offer, the policy version and — since 10 August 2026 — the external security evidence the decision considered. The gate fronting the governed backend validates the decision before the action runs. Missing, stale, wrong-key or wrong-policy-version decisions are refused.

The enforcing policy is itself a versioned, digest-bound object held identically on the authority host and the workload host. On 3 October 2026 both hosts were moved together to policy version `external-procurement-r7-2+base-usdc-tavily-canary-1` with **17 grants**, in a coordinated cutover that removed a temporary external-test grant (Section 7.2).

6.5 Governing execution and delivery (S7)

The gate admits exactly the authorized action. For paid actions, payment is a *distinct* concern from permission: settling an invoice never substitutes for an authorization, and an authorization never implies funds moved. Where payment rails are used, the transaction reference enters the same evidence record as the authorization it belongs to — payment and permission stay two bound facts, not one blurred one.

For the production route cohort, the governed backend is reached through the NOMOS gate. For one frozen research lineage, AgentNOMOS has additionally demonstrated a **controlled runtime boundary** in which the authorized action is mediated by a separate runtime (OpenShell) and the observed effect is bound back into evidence. That lineage, its scope and its limits are described in Section 10. It does not describe how the production cohort executes.

6.6 Proving what was executed and returned (S8)

Execution produces a signed receipt: an Ed25519-signed record issued under a published key set, binding the payment, the represented intent, the pinned capability and offer, the admission decision, the security evidence that decision considered, the execution identity and **the hash of the response bytes**. Receipts are stored append-only and can be verified by re-deriving the signed content and checking the signature against the published keys.

Three honest limits travel with this. Receipt coverage is scoped to the instrumented route cohort, not "every call everywhere". The assurance ceiling is *operator-estate vantage*: the verification is reproducible, but no independent third-party domain attests it. And **verifying a receipt is not the same as obtaining one**: checking a receipt you already hold needs no credential at all, while retrieving an individual receipt from the governed cohort's evidence store requires the consumer's own authentication (Section 19).

A fourth limit has its own section: what the receipt binds ends at the response bytes. It does not extend to the outcome, and it does not attest that the provider's output is true.

6.7 Recording what was observed afterwards (S9)

A receipt proves what was executed and returned; it does not prove satisfaction. After the fact, the product compares delivery against the offer and writes the result as its own record in an append-only store, referencing the receipt. This record is **evidence that can be inspected, not proof that can be recomputed**. Outcome records are not reputation scores, and no cross-party ranking is derived from them.

6.8 Observing what a stricter policy would have decided (S10)

Verified history feeds a shadow evaluation: profiles of how a counterparty has actually behaved, and a record of what a stricter policy *would* have decided. This runs on a schedule and is **observational**. It records and informs; it does not block, delay, retry or alter any production request. The same shadow-first discipline now applies to research and spend decisions (Section 15). Turning recorded results into automatic policy changes is a gated roadmap item, because an autonomous feedback loop is itself a consequential action.

7. Evidence and Admission

7.1 Security evidence from an external authority, enforced

On a fixed schedule the product fetches a public vulnerability catalog published by an external security authority, records its provenance, matches it against the component inventory of the serving stack behind **one** governed route, and emits a signed assessment that the authorization component verifies and writes into the signed decision core. The producer refuses to sign from a stale, unparseable or future-dated source, so old evidence cannot be re-signed as new; on failure it publishes nothing and the gate eventually refuses.

Dated snapshot, 6 October 2026: 228 signed security assessments since 10 August 2026; maximum observed interval between consecutive assessments ≤ 6 hours; scope limited to one route, each assessment marked as not estate-wide.

This is evidence for one route's admission input. It is **not** system-wide or continuous security verification of AgentNOMOS, it does not cover the other seven governed routes, and it is not independent validation: the catalog is external, but transport, binding, signing and verification all happen inside the operator's estate.

7.2 The admission policy as a versioned, two-host object

Admission is enforced against a policy document that is versioned, digest-bound and held on both the authority host and the workload host. A decision signed under one policy version is refused by a gate that holds another.

Current state (workload host measured 6 October 2026; authority host per the sealed cutover record of 3 October 2026).

Property	Value
Policy version	external-procurement-r7-2+base-usdc-tavily-canary-1
Grants	17
Policy digest	sha256:708f3950...cbbca5 (both hosts)
Temporary external-test grant	removed — 0 occurrences on either host
Coherence check after cutover	an interop canary signed by the authority host and accepted by the workload gate: 22 / 22 checks passed

The October cutover is worth describing because it is the first time a grant created for an external party was removed again under the same discipline that created it. The temporary grant existed only for the external participant test of Section 13. After that test, the credential was revoked first; then the operator applied the grant removal on the authority host and the workload host installed the identical policy immediately after; then the canary confirmed that a decision signed on one host is accepted on the other. The previous composite version string was not reused: the owner rejected a candidate version that would have kept the external party's name in the policy identifier, and approved the one above.

One limit of this record is disclosed rather than smoothed over: the authority host's side of the cutover is evidenced by the operator's pass token and by the canary's cross-host coherence, not by a preserved raw transcript of the authority-host commands.

7.3 What the admission evidence does not show

Admission evidence shows that a decision was made under a named policy with named inputs. It does not show that the policy is *right*, that the inputs were *true*, or that the action was *intended* by the principal behind the caller. Technically valid $\neq$ authorized $\neq$ intended $\neq$ proven, and each of those steps is a different kind of evidence.

8. Identity, Capability and Compatibility

AgentNOMOS describes itself to machines through digest-bound public artifacts:

- **Agent card (A2A)** — who the agent is, what it exposes, and a pointer to the trust-chain manifest via a namespaced extension that explicitly disclaims being part of the A2A standard. A second card on the federation surface is declared the canonical copy.
- **Capability manifest** — the canonical statement of the eleven stages: for each stage, what evidence type backs it, how strong that statement is in a fixed assurance vocabulary, and a `not_claimed` list stating what the manifest deliberately does **not** assert — no reputation scores, no certification, no automatic compliance determination, observational governance only, no portfolio-wide coverage, and, since 3 October 2026, that OpenShell is not the runtime of the production cohort.
- **Compatibility map** — how the chain relates to A2A, MCP, x402, XRPL and neighbouring technologies, entry by entry, separating "what the external specification says" from "what NOMOS implements". Its reading rules state that *a relationship is not compatibility*.
- **Purchase manifest** — machine-readable payment terms for the priced capabilities, including, for each, a JSON Schema of the accepted request body.
- **Reproducibility index** — the entry point to the published ALLOW and DENY references, the verifier and the replay command (Section 12). It also publishes the claim ceiling.

8.1 The canonical manifest, deployed

Version 7.2 had to state that the production path writing the manifest was not identified, and therefore declined to describe it. That has changed. On 3 October 2026 the canonical manifest was **deployed through a single writer**: a builder that generates the manifest from its source definitions, applies it atomically, and on a second run reports `unchanged` — meaning the live bytes equal the canonically generated bytes. The deployment was preceded by two independent scratch builds that agreed on the content digest, and followed by validation against both the local file and the public URL. The S7 execution-boundary publication (Section 10) used the same builder an hour later.

Measured on 6 October 2026.

Property	Value
Public manifest	https://agentnomos.com/.well-known/nomos-capabilities.json , HTTP 200, 53,026 bytes, SHA-256 79c5430a...
Public = local	byte-equal (compared directly)
Generated	2026-10-06T08:53:02Z
Content digest declared inside the manifest	sha256:946170fe...
Validator against the public URL	PASS 167 / 167 , including liveness probes that confirm every cohort route refuses an unauthenticated caller
Admission policy reference in the manifest	<code>external-procurement-r7-2+base-usdc-tavily-canary-1</code> — equal to the enforcing configuration
Stage count and range	11, S0-S10

Two disciplines make these artifacts more than marketing files. First, the manifest is **digest-bound**: it declares its own content digest, so any alteration is detectable against the digest it carries. Second, the published request contract is provably the contract the paid endpoint enforces: a sealed round on 13 August traced a lossy schema conversion that had been silently dropping value constraints, fixed the converter rather than the output file, and re-verified equivalence between the published schema and the runtime validator. Sealed rounds on 19 August extended this across the wider paid surface: input schemas, sample bodies and one-of constraints are published in the payment challenge and the discovery manifests.

The validator passing is a statement about **consistency between the manifest and its own rules and the live surfaces**. It is not a statement that the manifest's wording is everywhere correct. It does, since 6 October 2026, refuse any statement that the S0–S10 chain is "proven", the wording the manifest carried until that date.

This self-description discipline is the same discipline the product applies to counterparties: claims are structured, scoped, versioned and digest-bound — and the absence of a claim is meaningful.

9. Authorization and Fail-Closed Behaviour

The authorization layer has three properties worth stating precisely.

Separation of authority. Admission decisions are signed by a dedicated authority running apart from the governed workloads. The workload host holds no decision-signing key; the gate that enforces decisions validates signatures against a controlled keyring. Compartment boundaries are treated as unproven until measured from the *other* side.

Key and credential lifecycle as policy. Verification distinguishes *active* keys, which may authorize new actions, from *retired* keys, valid only for verifying historical artifacts. A decision signed by a retired key is refused on the live path. Caller credentials have the same lifecycle: a revoked credential is refused at the boundary.

Fail-closed, demonstrated — including with a real external payer and an external operator. Refusal paths are exercised, not presumed:

- *Stale evidence.* A traversal was blocked because its supporting evidence had gone stale, and resumed only after freshness was restored. The durability mechanism of Section 7.1 exists because of it.
- *Policy-version mismatch.* During the security-evidence integration, the authority issued a valid permit under a new policy and the enforcing side **rejected it before execution** because it could not reproduce that policy version. No execution, no receipt and no outcome record were created. A permit that the enforcer cannot verify is not a permit.
- *Billing integrity with external money on the line.* On 18 August 2026 an external buyer's x402 payment was **verified** and the request body then failed the route's input contract. The gateway refused after verification and before settlement: **no settlement was attempted, the buyer was not charged, and the revenue ledger records zero events for the call**. The sealed repair rounds that followed did not weaken the refusal — they made the input contract machine-readable *before* payment.
- *Revocation of an external participant's credential.* On 3 October 2026, after the external participant test of Section 13, the test credential was revoked. A negative test against the production boundary was then refused with `401 KEY_NOT_ACTIVE`, and no new execution was recorded. Per claim rule 9 this is a **boundary refusal**, not a governed DENY: the request never reached the policy.

A guard that has never been made to fire is not known to work. These have fired, and their records are sealed.

10. Controlled Execution and the OpenShell Boundary

10.1 Why an execution boundary

The receipt of Section 11 binds what the gate executed and what bytes came back. It does not, by itself, describe what a runtime was *allowed* to do while executing, or compare that with what it *did*. That is the execution gap of Section 3: **Authorized Action ≠ Actual Action**. Closing it for a given action needs three owners that do not collapse into one:

Phase	Owner	Responsibilities
Before execution	AgentNOMOS	intent, policy, admission, authority
During execution	OpenShell	runtime, sandbox, enforcement, mediated execution
After execution	AgentNOMOS	evidence binding, receipt, outcome, governance

In one sentence, as published in the manifest: *AgentNOMOS decides whether an action may run. OpenShell enforces the runtime boundary. AgentNOMOS verifies and binds the observed execution into evidence.*

10.2 The R1.4I lineage: final and frozen

Between September and early October 2026, AgentNOMOS built and verified a controlled execution lineage on the NVIDIA OpenShell runtime, named here as the runtime technology only. The lineage went through key rotations, a canary apparatus that was **stopped as failed** rather than forced through, a disposal and re-attestation of its runtime gateway, four secret-boundary incidents that were each remediated and sealed, and a restoration that was first **held** on a measured mismatch and passed only on a later, independent run.

On **2 October 2026** the lineage was declared **FINAL and FROZEN** in a sealed closure record. Reopening is permitted only on named triggers; a further execution round, a further authority round, and replay of the spent authority are excluded. The frozen record carries these invariants verbatim:

- technically valid ≠ authorized ≠ intended ≠ proven
- Binding ≠ Authorization
- Execution ≠ Semantic Preservation
- **Authorized Action ≠ Actual Action**
- Signature valid ≠ semantics valid
- **CHECKED BYTES == EXECUTED BYTES**
- classification before read
- manifest membership ≠ read authorization
- command authorization ≠ executable-name authorization
- Historical HOLD evidence is never erased merely to make current state cleaner.

The **CHECKED BYTES == EXECUTED BYTES** invariant is the one most often lost in practice. A hash of a file on disk proves the disk, not the interpreter: a cached compiled module, a search-path entry or an environment variable can make a process execute bytes other than the ones that were checked. The lineage's gates measured this property at the process boundary and refused where it did not hold. It is claimed here **for the frozen R1.4I lineage only**, where it was measured, and not for any other runtime path.

10.3 Trust Chain × OpenShell S7: live and verified, within scope

On **3 October 2026** the boundary was published as a member of stage S7 in the capability manifest, `execution_boundary`, and verified in a separate operator-run pass against both the local file and the public URL. Its status is **LIVE_AND_VERIFIED**. The published object states:

Field	Published value
Scope	openshell_r1_4i_lineage — the frozen lineage only
Production route cohort mediated	false — the cohort routes are executed by the NOMOS gate, not inside OpenShell, and not every S7 execution is mediated by OpenShell
Sub-claim	SOFTWARE_ATTESTED_MEDIATED_RUNTIME_ACTION_AND_EFFECT_BINDING — an S7 sub-claim, not an S8 proof class
Global claim ceiling	EXECUTION_BOUND_PROOF , unchanged
Trust base	software attestation inside the operator's own host trust base
Authority	one-shot; replay not allowed
Execution result	SERVICE_PRESTATE_READY
Verification	a separate verification pass after execution, run by the operator — not a third-party audit
Publicly verifiable	false — the evidence hashes are published; the checkpoints they anchor are retrievable by the operator only

The **execution result deserves a plain reading**. The single governed execution of this lineage restored and checked OpenShell's own governed runtime state. **It was not a customer paid-route execution**, and the lineage's own signer has issued no production signatures or receipts. What the boundary demonstrates is the *mechanism* — authority before, mediation during, binding after — on one governed action, frozen.

10.4 What the boundary does not claim

The published object and the frozen claim ceiling exclude, explicitly:

- **TPM attestation** and **TEE attestation** — none is used;
- **measured boot** — none is claimed;
- **host-independent trust** and **root-adversarial enforcement** — the host and its root supervisor are **inside** the trusted computing base; a party with root on the host is not defended against;
- **NVIDIA certification, validation, endorsement or partnership** — OpenShell is named as the runtime technology only;
- **third-party verification** of the OpenShell execution;
- that current paid production routes execute through OpenShell;
- that every S7 execution is mediated by OpenShell.

The frozen claim ceiling reads, in the record's own words: the lineage *"proves the reviewed software-attested mediated runtime state/effect binding within the existing root/host TCB; does NOT establish TPM attestation, TEE attestation, measured boot, host-independent trust, root-adversarial enforcement, or NVIDIA certification."* This document claims no more.

Other runtime work exists on the same host, including AutoNOMOS-labelled Delta-F broker units started after the freeze (Section 15.4). None of it is described by the published boundary, and none of it is claimed here.

11. Proof, Receipts and the Outcome Boundary

What the receipt binds. Every governed execution on the instrumented cohort yields a signed receipt under a published Ed25519 key set. Under the signature sit: the payment reference, the represented intent and its digest, the pinned capability and offer, the admission decision and its policy version, the security evidence that decision considered, the execution identity, and the hash of the response bytes. Receipts are stored append-only. A stranger holding the receipt and the published keys can check the signature and recompute every one of those digests.

What the receipt does not bind — stated once, precisely.

Outcome evidence is recorded and inspectable, but the current execution receipt does not cryptographically bind a complete outcome object. S9 remains post-hoc, and S10 remains observational.

Three things are separable and must stay separate:

1. **Signed execution and response-byte evidence.** Cryptographic and recomputable. This is what "proof" means in this document.
2. **Post-hoc outcome observation.** A record created after the fact in an append-only store, referencing the execution. Inspectable, not provable. Because the signed receipt carries no outcome field, a modified outcome value in a published bundle is **not** detected by the public verifier, and this document says so.
3. **Outcome-to-governance feedback.** Does not exist today. Recorded results do not change production behaviour.

Receipt $\neq$ truth of provider output. The manifest states, per cohort route, that the receipt does not prove the factual correctness of the provider's output, does not prove that the provider itself signed or attested the result, and does not prove that any independent source confirms it. A receipt proves *which bytes were returned under which decision*. It is silent on whether those bytes are true.

Execution $\neq$ Outcome. Closing the outcome gap requires a receipt-schema revision so that an outcome digest sits under the signature, together with new reference bundles and a verifier release. That release has not started, and no part of this document anticipates it as done.

Freshness durability. The admission evidence that backs decisions is regenerated on a fixed schedule through the same production code path that normally produces it — not a parallel writer — fail-closed on error.

Assurance ceiling, stated. Receipt verification is reproducible by anyone holding the published keys, but the strongest available statement is **execution-bound proof from the operator's estate vantage** — the claim ceiling `EXECUTION_BOUND_PROOF`. No third party operating its own clock, keys and infrastructure attests these receipts. This document does not use the words "independently verified" for anything a third party has not actually verified, and does not treat a fetch from the operator's own network as an external vantage, because the public domains resolve to the operator's own address.

12. Public Reproducibility

A trust product that can only be verified by its operator has not solved anything. Since 11 August 2026 the following is public:

- **An ALLOW reference** — the complete evidence bundle of one real paid traversal, including the settlement transaction, the represented intent, the pinned offer and capability, the admission decision, the signed receipt, and a ten-event execution trace.
- **A DENY reference** — a real production refusal, generated by one controlled negative evaluation at the unmodified production gate: a requested capability outside the route's capability family, refused deterministically before authorization and before execution.
- **A stateless verifier** — a small program with no credentials and no operator configuration, which fetches only explicitly public URLs and re-derives everything it checks.
- **A replay command** — two plain steps, published verbatim.
- **An axiom requirement matrix** — nine questions a sceptical integrator would ask, each answered with a public location **and the class of verification it supports**.

Re-run on 6 October 2026. The served verifier was fetched, confirmed byte-identical to the release sealed on 13 August (SHA-256 `f6c5fd0b...00fb`), and run against the live public surfaces. Result: **21 checks: 21 PASS, 0 FAIL, 0 NOT_RUN**, exit code 0.

Class	Checks	What it means
Recomputable	16	a stranger recomputes a digest or a rule outcome from published bytes
Cryptographic	3	a signature or an on-chain fact is checked directly
Subtotal — cryptographic or recomputable	19	
Inspectable	2	the record exists and is consistent, but confirming it needs operator-held data
Total	21	21 pass, 0 fail, 0 not run

The two inspectable checks are the ALLOW case's outcome-and-shadow reference (Section 11) and the DENY case's operator-attested zero-delta counter record; the verifier prints that label on every run. Its last line states the boundary in the tool's own words: *a full local re-evaluation of the internal NOMOS policy is NOT possible from public data and is NOT claimed by this verifier.*

Tamper closure. On 13 August a single change shipped to both public surfaces — the website and the public source repository — byte-identically. It widened the fields the verifier protects and closed three substitutions that had previously passed: a swapped signing-key identifier, a swapped match digest, and a swapped verification-key URL. The reference bundles were not touched and the published result line stayed exactly `21 checks: 21 PASS, 0 FAIL, 0 NOT_RUN`.

What is not publicly reproducible. The OpenShell execution-boundary evidence of Section 10 publishes its anchors but not the checkpoints behind them. The external participant call of Section 13 produced a receipt under the published key set, but retrieving that receipt is an authenticated path. Neither is part of the 21 checks above.

13. Interoperability and External Operator Evidence

AgentNOMOS is deliberately unoriginal at the protocol layer: it speaks existing standards and adds the governance the standards leave open.

- **A2A** — agent description and messaging. AgentNOMOS publishes agent cards and uses a clearly namespaced vendor extension to point at its trust-chain manifest. Discovery over A2A does **not** invoke the paid capability.
- **MCP** — tool discovery and invocation. Governed backends are reachable as MCP surfaces. MCP is not declared a paid invocation path, because it cannot transport the payment step.
- **x402** — machine-readable payment terms over HTTP 402 and on-chain settlement. NOMOS treats x402 as payment transport and terms discovery. **Payment is never treated as authorization** (Section 14).

This section collects evidence that was produced or shaped **outside the operator's own measurement loop** — by independently operated external participants, a foreign platform's boundary, an external conformance suite or an external audit tool — so that its boundary can be stated once, clearly, at the end.

13.1 External participant call through the production boundary (October 2026)

What happened. An independently operated external participant was given a temporary, single-purpose test credential and a temporary admission grant for one read-only governed route. On **3 October 2026 at 04:01Z** (the participant's clock; server-side authorization recorded three seconds later), infrastructure controlled by the participant called the route through the production boundary. The participant is not named in this document; nothing here implies a partnership, an endorsement, or permission to use the participant's name. The governed path authorized the call under the then-current policy, executed it, and returned **HTTP 200** with 2,030 response bytes and a **signed execution receipt** (receipt format v2, Ed25519, under the same published execution key set the public verifier uses). The server-side execution row and receipt are preserved.

How it was corroborated. The participant produced its own verification record of the call. That record was received, its SHA-256 matched the value the owner expected, and it was archived byte-identical in a sealed checkpoint. A field-by-field cross-check against the server-side execution store matched on **19 of 19** fields, including the execution and decision identifiers, the intent digest, the response hash and the receipt digest.

How it was closed. At 08:30Z the test credential was **revoked**, and a negative test returned `401 KEY_NOT_ACTIVE` with no new execution recorded. At 09:31Z the temporary grant was **removed from both enforcing hosts** in a coordinated cutover; both hosts now hold the 17-grant policy of Section 7.2, with zero occurrences of the temporary grant, and a cross-host canary passed 22 of 22 checks. The external party is referenced nowhere in the public capability manifest. The sealed classification of the whole exchange is:

```
EXTERNAL_CORROBORATION · TEST_ONLY · COMPLETED · CREDENTIAL_REVOKED ·
TEMPORARY_GRANT_REMOVED
```

What this demonstrates. An independently operated party outside our estate, using its own infrastructure, reached the production governance boundary over the public network, was admitted under an explicit, scoped and time-bounded grant, received a governed result with a signed receipt, and independently recorded what it received — and the authority it used was then withdrawn cleanly, on both hosts, without residue in the policy or the public manifest.

What this does not demonstrate.

- **Not universal interoperability.** One participant, one route, one call, one credential mode (shared secret over TLS; the caller signed nothing).
- **Not certification, validation or endorsement**, by the participant or anyone else. The external record is **unsigned** by its author; it corroborates consistency, and corroboration $\neq$ correctness.
- **Not uncontrolled autonomous adoption.** The call was arranged, scoped and time-limited by both sides. It was a test.
- **Not** an independent verification domain: the external party did not verify the admission decision's signature, the policy, or the link from its own mandate to the intent we recorded.
- **Not** continuing access: the credential is revoked and the grant is gone.

13.2 A human-assisted external full-chain traversal (August 2026)

On **24 August 2026** an external participant's agent completed a paid traversal of the public full-chain capability. A human gave the agent a challenge hint carrying a non-resolving address; the agent located the correct public site by web search, then discovered the capability, endpoint, schema, payment terms and verifier material through the agent card and the linked machine-readable surfaces. It paid **0.001 USDC on Base mainnet** from a wallet that appears in no operator, canary or treasury list, received the result and a signed Ed25519 receipt, and verified it. A sealed forensic review confirmed the on-chain settlement, the receipt signature and digest, the response-hash and intent-digest bindings, and 20 of 20 server-side checks. The run is published, with the participant's consent, in the challenge's verified-runs record.

It is classified **Class B** — **assisted external full-chain traversal**, because the challenge's Class A requires that the agent's only technical starting point was the agent card URL. Until this run, the only settlements on the full-chain route had been the operator's own canaries; that is no longer the case.

This demonstrates a paid external full-chain traversal under a consented, human-assisted Class-B test. It does not satisfy the still-open challenge for an unassisted external traversal and does not establish organic autonomous adoption.

The run also does not prove the wallet holder's identity, the model version reported by the participant, sustained demand, or any independent cryptographic truth outside the operator's estate.

13.3 A cross-operator lane on which the caller holder-signs (August 2026, on hold)

On 18 August 2026 a production A2A lane to a foreign agent platform went live end to end: the calling platform authenticates by an **ES256K signature produced from a key it holds**, the signed token's replay identifier is burned on use, and a kill switch was exercised before cutover. One end-to-end run through the **real** boundary produced a governed ALLOW with a signed Ed25519 execution receipt, and an immediate replay was refused. The platform is not named in this document, and nothing here implies a partnership with or endorsement by its operator.

Current state. The lane's inbound trigger is **paused fail-closed** and the strand is **on operator hold**. The execution store records two executions on this lane in total. The lane's token-renewal and pipeline watchdogs are currently failing (an open operational finding, Section 18). **No continuous operation, availability or foreign-platform reliability is claimed.**

13.4 Conformance, audit and off-host evidence (August 2026, historical)

- **Public cross-ecosystem evidence graph** (since 16 August 2026): a human page and machine-readable artifacts carrying four sealed interoperability proofs, each node with explicit `does_not_prove` and `forbidden_claims` fields, generated under twelve fail-closed invariants. Its bytes were verified from a separately located host of the operator's own estate — an **off-host** measurement, not an independent one.
- **External conformance suite, isolated test lane** (18 August 2026): an isolated OpenID4VP/HAIP test lane — deliberately decoupled from the production admission core — was exercised by the external OpenID Foundation conformance suite. Eleven applicable modules reached a successful final outcome: **seven PASSED and four REVIEW**. The published record carries `conformance_status: PASSED` and `certification_status: PENDING_NOT_GRANTED`, verbatim. This is **not a certification**, the split is always reported as 7 + 4, and the result describes the isolated lab lane only. The suite found a token-freshness gap that 101 of our own tests had missed; it was repaired and sealed before the final run.
- **MCP tool-safety annotations, externally re-audited** (19 August 2026): five tools on the public NOMOS MCP server serve read-only, non-destructive, idempotent, closed-world annotations after a fail-closed semantic repair; an external audit tool re-scored the server 10/10. This is a **measurement of declared semantics on one date** — not a certification.

13.5 The boundary of this entire section

Nothing in this section is independent verification of the production NOMOS trust chain. One independently operated external participant corroborated one test call it was granted. Another external participant completed one assisted paid traversal. The foreign-platform lane proves one governed traversal, not operation. The conformance suite exercised an isolated lab lane. The audit tool measured declared tool semantics. The off-host hash check used the operator's own remote host. Each involves a party or vantage outside the measurement loop this document otherwise relies on — and **none of them attests a production admission decision, execution receipt or outcome record on its own keys**. The claim ceiling therefore remains **execution-bound proof from the operator's estate vantage**. The independent verification domain remains a roadmap item, and it remains open.

14. Machine Commerce and Payment Rails

14.1 Payment is settlement, not authority

x402 and the other rails are **payment and settlement**, nothing more. Whether an action may happen, and the binding of intent, offer and execution, is NOMOS's job; settlement is the rail's job. Three consequences are enforced, not just stated:

- **Payment ≠ Authorization.** A settled payment never substitutes for a signed admission decision, and an authorization never implies that funds moved.
- **A 402 is not revenue.** An HTTP 402 `payment_required` response shows terms; it is never counted as a sale. A payment that is verified but not settled is not revenue either.
- **Settlement direction is declared.** Inbound payment is supported. The public artifacts state that **no autonomous outbound wallet settlement authority** is stated or implied. Accepting money for a service grants the service no ability to spend.

14.2 Inbound rails and what they sell

AgentNOMOS remains one product; the rails are settlement paths underneath it, and what differs is the capability bought and the evidence object returned.

	Base rail (USDC)	XRPL rail
What is sold	paid data and verification calls; and one governed read-only execution through the full chain	two advisory preflights : a payment preflight and a cross-border preflight
Full-chain price	\$0.001 in USDC	—
What comes back on the full-chain capability	an Ed25519 execution receipt verifiable against the published key set	a preflight decision and preflight receipt — a different object, not a full-chain traversal
External usage of the full-chain capability, as of 2026-10-06	one external settlement (the Class B run of Section 13.2); all others are the operator's own canaries	—
External payments on the rail, as of 2026-10-06	see 14.3	zero

The cross-border preflight labels itself as preflight classification and not legal advice. The XRPL rail is listed in the machine-readable purchase manifest since 6 October 2026, in its own block with its own network and asset, marked as advisory preflights and not as full-chain traversals.

14.3 Revenue, as a dated snapshot

Snapshot of 6 October 2026, 07:31Z, re-confirmed unchanged at 09:07Z (canonical revenue ledger, read-only; reconciler run of that time reported health OK). External FINAL counts only settled, final payments from wallets that are not the operator's; **402 challenges, unsettled or expired authorizations, internal canaries, self-payments, founder-funded pilots and refunds are never counted as external FINAL revenue.**

Class	Amount	Events	Wallets
External, final	1.451000 USDC	62	36
Founder-funded pilot (internal)	1.100000 USDC	2	—
Self-pay (internal)	0.129000 USDC	11	—
Canary (internal)	0.063000 ¹	5	—
Refunded	0.050000 USDC	1	—
Unresolved, requires review	0	3	—

¹ 0.062 USDC plus one internal XRPL canary of 0.001 XRP, which the ledger's projection adds into a single figure — a known accounting defect (Section 18), not a currency conversion.

All 62 external events settled on Base in USDC. The latest external settlement was on 2 October 2026; no new external revenue event has been recorded since.

Three qualifications travel with this number, and none of them is optional:

1. **Buyer genuineness is uncertain, and wallets are not customers.** Request metadata of several buyers indicates test, monitoring or probing tools rather than end customers. The ledger classifies these events as external because the paying wallets are not ours; it does not and cannot establish that a human or business customer stood behind them. **36 wallets is not a count of verified unique humans or customers.**
2. **The number is small.** It is reported so the reader has it, not as traction. It does not demonstrate sustained demand, product-market fit or adoption.
3. **Revenue has never funded a purchase.** No outbound procurement has been paid from inbound revenue.

Three inbound cases in which a payment may have been taken without the result being delivered are open and are described with the operational boundaries in Section 18; they are not part of any figure above as revenue.

14.4 Outbound purchasing

One approved outbound attempt, no purchase. On **4 October 2026** the operator approved and ran a single outbound x402 purchase attempt against **an external x402 seller** (Base mainnet, USDC, terms 0.01 USDC), after a sealed preflight. The seller is not named in this document. The agent attached a signed payment authorization within its validity window. **The seller returned HTTP 402 with the reason `address_not_registered`.** Sealed forensics established:

- **0 settlement** — the authorization was never submitted on-chain, and the buyer wallet balance was unchanged: **0 USDC moved**;
- the **signed authorization expired unused** and can never be executed;
- the **one-shot lock was consumed** — the canary cannot be re-run;
- whether the refusal came from the seller's server or its facilitator, and whether the payer or the payee address needs registration, cannot be determined from our side.

This was **not a successful purchase**, and this document does not describe it as one. No retry is permitted without a new preflight and a new owner approval; the operator is awaiting the seller's diagnosis. What the attempt does show is narrower and still useful: the outbound path is gated by a sealed preflight, a one-shot lock and an explicit approval, and when the counterparty refused, nothing moved and nothing retried.

An earlier bounded internal procurement cycle (August 2026) made a real mainnet purchase of 0.020000 USDC under a pre-signed human authorization and a hard budget, verified delivery, produced a receipt and then disarmed itself. It is an **operator-inspectable internal record**, not proof, not external revenue, and not an active capability.

15. Research Governance and Bounded Autonomy

15.1 Why governance reaches research

Research and experiment budgets are consequential actions too: renting compute, re-opening a closed line of work, or submitting a result publicly all spend money or reputation. The same failure modes apply as at runtime — acting on stale evidence, repeating a route already shown not to work, or letting an enthusiastic process authorize its own spending. AgentNOMOS therefore extends the governance model to research and spend decisions. **Architecturally, governance now applies not only to runtime actions but also to research and spend decisions** — and, as everywhere else, the first stage of that extension observes rather than enforces.

15.2 The NOMOS Research Governor

The NOMOS Research Governor is an **offline** engine that records each research campaign as a **hash-chained campaign event ledger**. Its event types cover:

- **hypotheses** and their status;
- **negative knowledge** — closed routes, recorded so that effort already shown not to work is not spent again;
- **freezes and corrections**, where a correction is appended and never rewrites history;
- **spend GO decisions** bound to fresh evidence and to an **owner authorization**;
- **budgets**;
- **staleness and reopening decisions** — whether a new public signal justifies reopening a dormant campaign.

It was built in phases **P0–P2** on 5 October 2026: 29 JSON schemas and two golden fixture histories (P0); an offline core of 17 standard-library modules whose replay of both golden histories matches with zero mismatches (P1); a historical reconciliation that classified six past findings without rewriting any of them (P1.1); and a shadow deployment (P2) activated under an owner GO. Across all phases, **273 of 273 tests pass**.

In P2 the governor runs as a dedicated system user with no network access, and a separate witness user anchors the governor's ledger head in an append-only stream the governor cannot write. Read and write separation is enforced by the operating system, and a probe of the production admission store from the governor's account was refused by the operating system. A truncation drill on a copy of the ledger halted correctly. On a dry run of five historical signal cases, the governor's judgment agreed with the operator's 5 of 5 times.

Its status is SHADOW only: no execution, no blocking, no authority. It decides nothing, it spends nothing, and it does not gate any runtime or research action. Live comparisons against operator decisions stand at **0 of the 50** required, on day 0 of a minimum 14-day window; the next phase has not started. SHADOW ≠ production.

15.3 Two research campaigns as governance examples

These two campaigns are included **only** to show what research governance records. Neither is a product capability, and neither result is a competition result or a certification.

Quantum-circuit benchmark (ECDSA.fail). A public benchmark scores quantum circuits for elliptic-curve point arithmetic; a score becomes official only after a computationally expensive nonce search ("grind") and a public submission. Between 3 and 4 October 2026 the operator's research rounds produced local, unsubmitted candidates. On 4 October the public leader moved, the best local candidate's lead shrank to about 17 T-gates — below the threshold that would justify paying for a grind — and a second line of attack was closed as negative knowledge. The decision, recorded as **R19_NO_BIG_LEAD**, was: **no grind, no public submission**, campaign **DORMANT until a material reopening signal**. Spend on the line was a single GPU benchmark rental of about \$0.38. The governor's golden history of this campaign is 125 hash-chained events.

Quantum-circuit synthesis (Classiq). The final research artifact is a circuit of depth **333, 693** two-qubit (CX) gates, **767** single-qubit (u3) gates and width **18** qubits (SHA-256 622e4647...), whose integrity was verified from its bytes. It was **not submitted**. A rules question — whether a circuit not produced by the vendor's own synthesis may be submitted — is open and is a human decision. The governor records the campaign as **CLOSED, research archived**. This document does not describe it as a competition result.

15.4 AutoNOMOS: the bounded economic continuation, held for autonomous economic operation

AutoNOMOS is the economic continuation of NOMOS: the same authorization, evidence and receipt discipline, applied to an agent that would itself earn, budget and procure within owner-set bounds. It is **not** a completed autonomous commerce system, and this document makes no production-autonomy claim for it.

Its current state is **HELD FOR AUTONOMOUS ECONOMIC OPERATION**. A cloud deployment line reached a first worker verification step that passed 39 of 39 checks, and then stopped where it must stop: the owner verification step, the owner commit step and the owner final approval are **not complete**, and the sign, write and legal gates remain closed.

Experimental runtime components exist and are running. A read-only classification on 6 October 2026 found two groups on the workload host:

- **Delta-F-labelled broker units** — an execution broker and its observation peer, running since 4 October 2026, derived from the frozen OpenShell R1.4I broker core with NOMOS admission added. They mediate three artifact-producing profiles (SVG evidence composition, offline HTML micro-tool, declarative interactive HTML) inside a sandbox. Each profile is configured with **no network egress** and a **budget ceiling of zero**; the units hold no capabilities, cannot reach the signing-key or execution-store locations, and accept requests only from a local worker group. They can write artifacts into their own sandbox and their own decision ledger; they cannot spend, sign or submit.
- **A market router running in SHADOW mode** — it reads public listings of external task and procurement markets, classifies opportunities, and prepares decision packets for the owner. It is configured with production writes disabled, the signing gate closed and no owner cost thresholds, so no automatic action class can be produced; its own documentation states that it has no code path to accept terms, submit, sign, pay or touch a wallet. Its only outward effects are read-only requests to listed markets and notifications to the owner.

Their provenance is evidenced by staged release manifests whose hashes match the installed files and by a sealed read-only provenance and status checkpoint of 6 October 2026, which confirmed this classification; their role statements remain configuration and self-description, not a code audit. That is enough to say what they are configured to do and not to do. It is **not** enough to support any stronger public autonomy claim, and this document makes none. The one outbound purchase attempt of Section 14.4 was a separately approved, one-shot operator canary, not an AutoNOMOS action.

16. What Has Actually Been Demonstrated

This section is the load-bearing wall of the document. Each row names its scope, date and verification class. Evidence references are in Section 21.

Read the class column first. The heading says *demonstrated*, not *proven*, because the rows are not all of one kind. Where a row's class is **cryptographic** or **recomputable**, the property is verified within the named scope and vantage. Where a row's class is **inspectable**, the record exists and is internally consistent, but confirming it needs data only the operator holds. *Inspectable is never described as proof.*

#	Measured or evidence-backed property	Scope / cohort	Date	Class
P1	Governance binding live in production: admission decision → execution → signed receipt → recorded observation	8 routes ; the execution store holds 1,254 executions and 1,253 signed receipts on those eight and no others	continuous; measured 2026-10-06	inspectable stores + cryptographic per receipt
P2	Complete end-to-end traversal with exactly +1 record per pipeline ; receipt verified from a separate host against published keys	one governed read-only route	2026-08-10	cryptographic + recomputable
P3	External-authority security evidence enforced inside the signed authorization decision ; stale sources refused (dated snapshot in Section 7.1)	1 of the 8 routes only	since 2026-08-10	cryptographic (assessment) + inspectable (enforcement)

#	Measured or evidence-backed property	Scope / cohort	Date	Class
P5	A publicly purchasable full-chain capability: one safe read-only action, \$0.001 USDC, signed receipt	one capability over one governed route	live since 2026-08-10	cryptographic + recomputable
P6	A stranger can reproduce the proof from published bytes: ALLOW + DENY references and a stateless verifier	those two published cases	re-run 2026-10-06 → 21/21, verifier byte-identical to 2026-08-13	16 recomputable + 3 cryptographic, 2 inspectable
P7	A refusal is evidence: one controlled negative evaluation at the unmodified production gate refused before authorization and execution	one capability-family mismatch	2026-08-11	recomputable rule + inspectable non-execution
P8	Tamper closure on both public surfaces: three previously accepted substitutions now rejected	published references and verifier	2026-08-13	cryptographic + recomputable
P9	The published input contract equals the enforced contract	the paid capability; extended to the paid gateway routes 2026-08-19	2026-08-13 / 2026-08-19	recomputable
P10	Fail-closed exercised: stale evidence; unverifiable policy version; billing integrity with a real external payer; revoked external credential	admission path, paid gateway, production boundary	2026-08-10 → 2026-10-03	inspectable
P11	Canonical manifest deployed by a single writer; public = local byte-equal; validator PASS 167/167 against the public URL; policy reference = enforcing configuration	the capability manifest	deployed 2026-10-03; measured 2026-10-06	recomputable (digest, public bytes) + inspectable (writer, enforcing config)
P12	Two-host policy cutover removing a temporary external grant: 17 grants, identical digest on both hosts, cross-host canary 22/22	authority host + workload host	2026-10-03	inspectable (authority-host side operator-reported)
P13	Independently operated external participant's call through the production boundary: HTTP 200, signed receipt v2, external verification record archived and cross-checked 19/19; credential revoked; grant removed	one participant, one route, one test call	2026-10-03	cryptographic (receipt under published keys) + inspectable (record cross-check, revocation, removal); external corroboration, not independent verification
P14	Paid external full-chain traversal, consented and human-assisted: external wallet, real 0.001 USDC settlement on Base, delivered result, signed receipt verified	one participant, one run, Class B ; does not satisfy the unassisted challenge	2026-08-24	cryptographic (settlement, receipt) + inspectable (classification, externality)
P15	OpenShell R1.4I execution boundary FINAL + FROZEN with frozen invariants including Authorized Action ≠ Actual Action and CHECKED BYTES == EXECUTED BYTES	one frozen lineage; one governed execution (SERVICE_PRESTATE_READY), not a customer route	frozen 2026-10-02	inspectable (operator-only evidence; hashes published)
P16	Trust Chain × OpenShell S7 LIVE_AND_VERIFIED: scoped S7 sub-claim published with explicit not-claimed list; global ceiling unchanged	R1.4I lineage only; <code>production_route_cohort_mediated: false</code>	2026-10-03; re-read 2026-10-06	recomputable (published object) + inspectable (separate operator verification pass)
P17	A cross-operator lane on which the caller holder-signs: ES256K platform identity, replay burn, one governed ALLOW with signed receipt, replay refused	one partner lane; one traversal; now on hold	2026-08-18	cryptographic (receipt, caller signature) + inspectable (lane state)

#	Measured or evidence-backed property	Scope / cohort	Date	Class
P18	External conformance suite exercised the isolated OpenID4VP/HAIP test lane: 7 PASSED + 4 REVIEW; certification PENDING_NOT_GRANTED	isolated lab lane only	2026-08-18	external harness result, historical
P19	MCP tool-safety annotations live and externally re-audited (10/10)	one public MCP server, five tools	2026-08-19	recomputable (served annotations) + external measurement — not a certification
P20	Research Governor in SHADOW: hash-chained campaign ledgers, golden replays match with 0 mismatches, OS-enforced shadow/witness separation, 273/273 tests, dry-run agreement 5/5	offline; two campaigns; no execution, no blocking, no authority	2026-10-05	inspectable
P21	Outbound purchase path stops cleanly on refusal: sealed preflight, one-shot lock, approval; seller refused; 0 USDC moved; authorization expired unused; no retry	one outbound attempt	2026-10-04	cryptographic (on-chain non-settlement, balance unchanged) + inspectable (lock, approval)
P22	Bounded internal procurement, then disarmed: one internal mainnet purchase of 0.020000 USDC under prior human authorization	one internal cycle, internal money	2026-08	inspectable

On P1's metrics. Counts are snapshots; the canonical machine-readable sources in Section 21 are authoritative. On 2026-10-06 the store held 1,254 executions and 1,253 signed receipts across exactly the eight cohort routes. The single receiptless execution predates the receipt requirement (a gap canary of 7 August) and is preserved rather than repaired. The large majority of executions come from operator-run canaries and operator-controlled consumers; two come from the external participant's test credential of P13 (one of them our own readiness probe before the external call) and two from the partner lane of P17.

On P3's scope, once more. Enforcement is real, current and unattended — and it covers one route. A sentence that omits this is not a shorter version of P3; it is a different and false claim.

On P13 and P14 together. They are the two strongest pieces of external evidence this document carries, and they are different in kind. P13 is a *granted* test by an independently operated participant against a governed route, closed afterwards. P14 is an *open-challenge* paid run by another participant against the public capability, assisted at the start. Neither is adoption, and neither is an independent verification domain.

17. What Is Not Yet Proven

Stated as plainly as Section 16, because a trust product that hides its open items refutes itself.

1. **The receipt does not bind the outcome.** Section 11. Structural, not editorial. Until a receipt-schema revision ships, S9 is post-hoc, S10 is observational, and a modified outcome field in a published bundle is not detected.
2. **The receipt does not establish the truth of a provider's output.** It binds the response bytes; it does not show they are correct.
3. **Five of the eleven stages are not externally checkable.** S0, S1 and S2 enforce from evidence only we hold; S9 and S10 record without being bound under any signature. In particular, **S2 is a shared secret over TLS** on the cohort's consumer path. One partner lane holder-signs; that narrows the item and does not close it.
4. **External-authority security evidence covers one route, not the cohort,** and is not system-wide security verification.

5. **No independent provider-origin proof on the capability-discovery surface.** The manifest records this as an explicit `false` per route.
6. **The two governed subjects are not at equal proof strength.** One subject's admission evidence is bound to an independent ledger anchor; the other has none and says so.
7. **The execution boundary is software-attested, inside the host trust base, for one frozen lineage.** No TPM, no TEE, no measured boot, no host-independent or root-adversarial enforcement, no NVIDIA certification or validation. OpenShell is not the runtime of the production cohort, and its evidence is retrievable by the operator only.
8. **No autonomous production feedback loop.** Recorded results inform nothing automatically.
9. **No independent verification domain.** All evidence is produced and verified from infrastructure under our control. External corroboration (P13), an external assisted run (P14) and external conformance and audit measurements (P18, P19) are adjacent to this milestone and do not reach it.
10. **No external adoption.** One external participant test, one assisted external full-chain run, one partner-lane traversal and 62 small external paid calls of uncertain genuineness are real and measured; together they are not adoption, not a customer base, and not sustained demand.
11. **The public challenge is OPEN.** It remains open until a qualifying **Class A** external traversal is demonstrated: the agent card as the only technical starting point, external infrastructure, external wallet, external funding, public interfaces only, real settlement, delivered result, valid receipt and demonstrated verification.
12. **No completed outbound external purchase.** The only approved outbound attempt was refused and moved nothing.
13. **No production autonomy.** AutoNOMOS is held for autonomous economic operation; its experimental runtime components and the Research Governor hold no spending, signing or submission authority.
14. **The cross-operator lane has not demonstrated continuous operation.** One traversal, then hold.
15. **No certification, of anything, by anyone.** Including the conformance result (pending, not granted), the audit score (a measurement), the assessment-record surface (a scoped beta record), and the OpenShell boundary (no vendor certification or validation).
16. **No claim of general security.** Open security findings exist and are listed in Section 18.

18. Security and Operational Boundaries

18.1 Principles that are enforced

- **Fail closed.** Governed paths refuse when evidence, authority or transport is unavailable — including when the governance infrastructure itself degrades. Loss of the authority link stops admissions rather than loosening them.
- **Least privilege, measured.** Signing identities are dedicated, single-domain and lifecycle-bound. Compartment claims are validated from the constrained side before they are believed.
- **Separation of payment and permission.** No component treats a settled payment as an authorization, and no authorization implies payment.
- **SAFE-OFF discipline.** Capabilities that exist to demonstrate a property are shut down after proving it, and any genuine record created while the capability was live is preserved.
- **One-shot authority.** Consequential one-off actions — the OpenShell governed execution, the outbound purchase canary — run under authorities that are consumed on use and are never replayed. A spent authority stays spent, even when the action failed.
- **Redaction by default.** This document omits internal addresses, ports, paths, key locations, wallet addresses of the operator and service names, contains **no secret values**, and describes open security findings by status and category rather than by exploitable detail.

- **Errors are recorded, including our own.** Sealed records include cases where a measurement method — not the system — was wrong, and where earlier conclusions were retracted.

18.2 Open findings as of 6 October 2026

AgentNOMOS is **not** described here as generally secure, certified, compliant or audit-ready, and **security hardening is not complete**. The findings below are open on the observation date. Per operator decision they are disclosed by **status and category**; this document deliberately gives no secret material, credential locations, affected paths, endpoint names or other detail that would lower the barrier to abuse.

Category	Priority	Status on 2026-10-06
File-system access-control exposure on a workload host, affecting credential material	P0	Open — partially contained. Containment steps taken; the root-cause permission change and a full triage are outstanding
Third-party service credentials affected by that exposure	P0	Open — rotation in progress, not complete for one provider; outstanding for another
Treasury signing-key custody	P0	Open — a migration is prepared and not yet executed
Process-privilege reduction for signing-key isolation	P0	Open
Access control on one public API endpoint	P0	Open
AI-interaction transparency disclosure on interactive public surfaces (EU AI Act, Article 50)	P0	Closed for the three interactive surfaces named in the finding (notices rendered and verified 6 October 2026); related editorial-review and machine-interface labelling items remain open
Paid-not-delivered inbound payment cases (three)	P1	Open — to be settled or refunded
Revenue-accounting and public-metrics consistency	P1	Open
Operational watchdogs, a staging service and one historical checkpoint seal	P1	Open

The earlier scripted probe of the production surface (April 2026) was conducted from a separately operated host of our own; it is **not** an external or independent penetration test.

18.3 Why this section exists in a trust whitepaper

A governance product is judged by whether its operator applies the same discipline to itself. The findings above are the current truth; they are prioritized ahead of new capability work; and none of them is hidden behind the scope statements elsewhere in this document. Where a finding affects a claim — for example, one that concerns signing-key isolation on a host — the claim ceiling of Section 11 already assumes the host and its root supervisor are inside the trusted computing base, and makes no stronger promise.

19. Commercial and Integration Model

The integration surface is machine-first:

- **Discovery** — agent card, capability manifest, compatibility map, AI catalog entry, MCP server card and `llms.txt` on the public domains.
- **Invocation** — MCP endpoints for governed tools; HTTP APIs for evidence retrieval; and one priced full-chain capability invoked over HTTP with an x402 payment step.
- **Payment** — a purchase manifest publishes machine-readable terms for every priced capability, including a JSON Schema for each accepted request body. Settlement references join the evidence record. Inbound only.
- **Verification** — some of it is anonymous; some of it is not, and the difference is stated.

Which verification paths need credentials, and which do not.

Path	Access
Public ALLOW and DENY reference bundles	anonymous
Public execution key set (JWKS)	anonymous
The stateless verifier and its replay command	anonymous — it fetches only explicitly public URLs
Reproducibility index, capability manifest, compatibility map, purchase manifest, agent cards	anonymous
Verifying a receipt you already hold, against the published key set	anonymous
The receipt returned in the body of a paid full-chain response	arrives with the response; verifiable immediately and anonymously
Retrieving an individual receipt from the governed cohort's evidence store	authenticated — the consumer's own credential is required
The OpenShell execution-boundary checkpoints behind the published anchors	operator-held
Per-stage evidence records behind S0–S2, S9 and S10	operator-held

Five properties define the commercial posture, and each is a consequence of the evidence model rather than a pricing tactic:

1. **You can verify before you buy.** The offline verification path is free, complete and published. The first unpaid request to a priced capability returns the payment challenge, which shows the terms without paying anything.
2. **Unit pricing, published in machine-readable form.** Prices live in the manifests, where they can change without falsifying a frozen document. The full-chain capability's unit price sits at the payment facilitator's minimum viable amount — the price exists to make the settlement real, not to extract revenue.
3. **The input contract is delivered before payment.** Every paid route publishes its input schema, a sample body and its one-of constraints in the payment challenge and the discovery manifests, and a payment-carrying invalid body is rejected with the contract before payment verification.
4. **No service-level agreement is claimed here.** Where an SLA exists for a specific integration it is a contract document, not a whitepaper assertion.
5. **No customer names or customer counts appear,** because none meet the evidence bar of Section 1. The revenue snapshot of Section 14.3 is given with its qualifications; it is not described as traction, adoption or a customer relationship.

Integration for an external operator follows the pattern of Section 13.1: a scoped, time-bounded credential; an explicit admission grant for named routes, held identically on both enforcing hosts; governed calls that return signed receipts; and, at the end, revocation and grant removal under the same two-host discipline. Regional and enterprise readiness work — environments, identity-management integration, key custody, recovery objectives — is deliberately absent from the claim set until built and measured.

20. Roadmap

Ordered by evidence value and by risk, not by calendar promises. Each item becomes real the way everything else here became real: built, validated, sealed, and only then claimed.

First, the operational boundaries (Section 18). These precede new capability:

1. **Close the open P0 findings** in the categories of Section 18.2 — complete the credential rotations, apply the root-cause access-control change and triage, execute the key-custody migration, reduce process privileges, close the API access-control gap. (The transparency notices on the interactive surfaces were added on 6 October 2026.)

2. **Payment hygiene** — settle or refund the paid-not-delivered cases; complete the accounting corrections. (The public economics figures were reconciled with the canonical ledger on 6 October 2026.)

Then the trust chain itself:

1. **Extend external-authority security evidence beyond one route** — coverage, with its own negative tests and sealed cutover.
2. **Bind the outcome under the receipt signature** — a receipt-schema revision plus new reference bundles and a verifier release; the precondition for ever describing outcome integrity as proven.
3. **Holder-key authentication at S2 on the cohort's consumer path** — replace the shared secret with a signature the caller produces from a key it holds. Partially advanced on one partner lane; the remainder is open.
4. **Provider-origin proof on the discovery surface.**
5. **Publish the payment rails symmetrically** — the purchase manifest carries the XRPL rail since 6 October 2026; the agent card and `llms.txt` do not yet.
6. **Manifest scope wording** — corrected on 6 October 2026 at the single-writer builder, regenerated and validated; the validator now refuses the old wording.

Then external evidence and bounded autonomy:

1. **Repeat and widen external operator integration** under the Section 13.1 pattern, and **resume the cross-operator lane** after its hold resolves — accumulating enough governed runs to speak about operation rather than about one call.
2. **Outbound purchasing, second attempt** — only after the seller's diagnosis, a new sealed preflight and a new owner approval.
3. **Research Governor P3** — only after at least 50 live comparisons over at least 14 days in shadow, and only by separate owner decision. Promotion from shadow to any enforcing role is a governed change in its own right.
4. **AutoNOMOS** — a sealed checkpoint for its experimental runtime components, then completion of the owner verification, commit and final steps; until then, held for autonomous economic operation.
5. **Governed results-to-policy feedback** — promote the observational stage to enforcing for a narrow, named cohort, under the same shadow-first discipline.
6. **An independent verification domain** — the strategic milestone: a verification party outside our control, with its own keys and clock, attesting NOMOS evidence. This is the precondition for the words "independently verified" without qualification.
7. **External cohort onboarding** — only after (14).

Not on the roadmap as a claim: hardware attestation (TPM/TEE) for the execution boundary. If it is ever built, it will be described when it is measured; until then, the boundary is software-attested inside the host trust base, and that is what this document says.

21. Evidence Index and Dynamic Sources

Evidence packaging is not uniform, and the difference is stated per reference. Most references below are sealed packages: a checksum manifest covers every file, and the hash of that manifest is the package's identifier ("anchor"). A reviewer receiving one recomputes that hash, compares it, then re-verifies every file inside. Anchors are given as eight-character prefixes. All packages E20–E31 were re-verified with zero integrity failures on 6 October 2026; E1–E19 are carried from version 7.2, which re-verified them on 19 August 2026. Internal locations of these packages are not published.

Ref	Content (functional description)	Date	Packaging
E1	Governance-bound route cohort records (8 routes: admission, receipts, observations)	continuous, measured 2026-10-06	live append-only store state, not a package
E2	External security evidence integrated into the signed authorization decision	2026-08-10	sealed package
E3	Durable, fail-closed regeneration of that evidence with a derived source-age bound	2026-08-10	sealed package
E4	Public paid full-chain capability with one settled canary traversal	2026-08-10	sealed package
E5	Public reproducibility: ALLOW and DENY references, stateless verifier, replay command, axiom matrix	2026-08-11	sealed packages (two)
E6	Two-surface tamper closure of the published verifier and references	2026-08-13	sealed package
E7	Published request contract verified equivalent to the enforced contract	2026-08-13	sealed package
E8	Assessment-record scope alignment, two score levers refused on measurement	2026-08-13	sealed package
E9	Final closure of a demonstrated capability under SAFE-OFF	2026-08-10	sealed package
E10	Bounded internal procurement canary	2026-08	Inspectable record set — not sealed as a checkpoint package
E11	Live observation audit of external discovery after go-live	2026-08-11	sealed package
E12	Public cross-ecosystem evidence graph	2026-08-16	sealed package
E13	External conformance run on the isolated test lane, including the repaired gap	2026-08-18	sealed packages
E14	Publication of the conformance evidence on the cross-ecosystem graph	2026-08-18	sealed package
E15	Cross-operator holder-signed lane and its one end-to-end traversal	2026-08-18	sealed package chain
E16	Lane boot persistence; platform-side stall; fail-closed trigger pause	2026-08-19	sealed packages
E17	MCP tool-safety audit, repair and external re-audit	2026-08-19	sealed packages
E18	Paid-route input-contract event with a real external payer, and its repair chain	2026-08-18/19	sealed package chain
E19	Claim-by-claim current-state rebase for version 7.2	2026-08-19	sealed package
E20	Assisted external full-chain traversal: forensic verification, Class B rationale, revenue-attribution repair, public record	2026-08-24/25	sealed package (anchor f445d6bb)
E21	Full-stack source refresh R3: component status, OpenShell, research, revenue, security truth	2026-10-02	sealed package (anchor 30d1d917)
E22	Access-control exposure incident and partial containment	2026-10-02	sealed package (anchor 70f4ef0e)
E23	Credential rotation following that incident (partial)	2026-10-02	sealed package (anchor 3de76b91)
E24	OpenShell R1.4I final closure freeze: invariants, claim ceiling, reopen policy	2026-10-02	sealed package (anchor 39267822)
E25	External participant test closeout: server-side corroboration, revocation and negative test	2026-10-03	sealed package (anchor 99940539)
E26	External verification record archived byte-identical and cross-checked 19/19; policy version derivation	2026-10-03	sealed package (anchor 86a86977)
E27	Two-host policy cutover removing the temporary grant; canary 22/22	2026-10-03	sealed package (anchor 4f7f4245)

Ref	Content (functional description)	Date	Packaging
E28	Canonical manifest deploy by single writer; validators; public = local	2026-10-03	sealed package (anchor f0a5363a)
E29	Trust Chain × OpenShell S7 execution-boundary publication and verification	2026-10-03	sealed package (anchor d71f0a48)
E30	Outbound x402 purchase attempt: preflight and one-shot lock; canary run outputs	2026-10-04	sealed package (anchor 1a971499) for the preflight; the canary run outputs were added after that seal and are covered by E31
E31	Outbound attempt forensics: second challenge, on-chain non-settlement, expiry	2026-10-04	sealed package (anchor 00fe8cf1), includes a copy of the run outputs
E32	Research Governor P0–P2: schemas, goldens, engine, reconciliation, shadow activation	2026-10-05	repository with checksum manifests (verified 2026-10-06); not a checkpoint package
E33	Source Refresh R4: deltas 2026-10-02 → 2026-10-06	2026-10-06	sealed document pair (anchor 58a3b80a)
E34	Release-candidate evidence package: live re-measurements of 2026-10-06 (manifest, verifier, validator, ledger, admission configuration, security-evidence cadence)	2026-10-06	sealed package (anchor 08aae472)
E35	Read-only classification of the AutoNOMOS experimental runtime components (Delta-F brokers, SHADOW market router)	2026-10-06	sealed package (anchor 8f1201fd)
E36	AutoNOMOS runtime provenance and status, read-only, confirming E35	2026-10-06	sealed package (anchor 2e5bc5c9)
E37	Release closure: public-surface synchronization with before/after hashes, Official Journal verification record, release-day re-measurement, cross-surface claim scan	2026-10-06	sealed package (the release-closure checkpoint)

On E10. The records exist and are retained, but they are not sealed into a checksum-covered package. No integrity digest is published for E10, precisely so nobody can believe they have checked something they have not. The corresponding row (P22) is classed **inspectable**.

Dynamic public sources (canonical; always preferred over any number frozen in this document):

- Capability manifest — <https://agentnomos.com/.well-known/nomos-capabilities.json>
- Compatibility map — <https://agentnomos.com/.well-known/nomos-compatibility.json>
- Purchase manifest — <https://agentnomos.com/.well-known/x402.json>
- Reproducibility index — <https://agentnomos.com/.well-known/nomos-reproducibility.json>
- Agent card — <https://agentnomos.com/.well-known/agent-card.json>
- Operator identity and delegation — <https://agentnomos.com/.well-known/nomos-identity.json>
- Security contact — <https://agentnomos.com/.well-known/security.txt>
- Human verification guide — <https://agentnomos.com/verify.html>
- Public challenge and verified runs — <https://agentnomos.com/challenge/> and <https://agentnomos.com/challenge/verified-runs.json>
- Cross-ecosystem evidence graph — <https://agentnomos.com/.well-known/nomos-cross-ecosystem.json> and <https://agentnomos.com/cross-ecosystem/>

22. Public-Surface Consistency Statement

This section was filled at release closure on 6 October 2026 from a re-measurement of the public surfaces taken that day (09:07–09:10Z).

Synchronized on 6 October 2026 — before/after hashes, backups and rollback references are in the release-closure evidence package (E37):

- *Capability manifest*. Its scope wording no longer calls the S0–S10 chain proven; it states execution-bound proof for the named cohort with public verifiability limited to S3–S8. It was regenerated by its single-writer builder, the public copy is byte-equal to the local one, and the validator passes 167 of 167 checks, including a new check that refuses the old wording.
- *Homepage*. The outcome is described as recorded separately and related to the receipt, not contained in it; the governed-action sentence names the route cohort; and the eight-step flow is explained as a human-readable projection of the canonical S0–S10 model.
- *Purchase manifest*. The XRPL rail is listed in its own block on all three byte-identical mirrors, marked as advisory preflights, not full-chain traversals.
- *Public economics outputs*. The JSON, HTML and live API outputs show the canonical external FINAL figure from the revenue ledger and label their own counters as usage credits, not revenue; all three agree with the ledger.
- *AI-transparency notices*. Visible notices were added to, and verified by rendering on, the three interactive surfaces named in the original finding.

Agreeing without change: the deployed manifest and the enforcing configuration (17 grants, no temporary external grant, S7 execution-boundary object as described), the public verifier (21 of 21), and the public challenge (open; one Class B run recorded).

Known minor divergences that remain — none of them contradicts a claim in this document:

1. The agent card and `llms.txt` do not yet list the XRPL rail (the purchase manifest does).
2. The agent card and the MCP server card present a narrower broker identity than the product description; the MCP server card declares an older version than the deployed server.
3. The AI-catalog artifact is served at its well-known location only; the root URL returns 404, and no published surface references it.
4. Tool counts differ across discovery surfaces, and some advertised routes on a discovery surface return 404 or 410.
5. The public reproducibility index carries a generation timestamp from August.
6. The machine-interface (agent card) carries no AI-transparency reference.

23. Legal Disclaimer

23.1 Regulatory context, without overclaiming

AgentNOMOS serves organizations operating under EU digital-finance regulation, where the practical question is whether operational claims can be evidenced. The citations below were verified against the Official Journal texts on 6 October 2026.

- **DORA** — Regulation (EU) 2022/2554, application date in Article 64: applies since 17 January 2025. <https://eur-lex.europa.eu/eli/reg/2022/2554/oj/eng>
- **MiCA** — Regulation (EU) 2023/1114, Article 149: applies from 30 December 2024, with Titles III and IV applying from 30 June 2024. Under Article 143(3), crypto-asset service providers that provided their services in accordance with applicable law before 30 December 2024 could continue until 1 July 2026 or until they were granted or refused an authorisation, whichever was sooner; Member States could decide not to apply that regime or to shorten it. Article 143 contains further transitional provisions beyond this one. <https://eur-lex.europa.eu/eli/reg/2023/1114/oj/eng>

- **AMLR** — Regulation (EU) 2024/1624, Article 90: entered into force on 9 July 2024; applies from 10 July 2027, and from 10 July 2029 to the obliged entities referred to in Article 3, points (3)(n) and (o). <https://eur-lex.europa.eu/eli/reg/2024/1624/oj/eng>
- **EU AI Act** — Regulation (EU) 2024/1689, as amended by Regulation (EU) 2026/1744: general application from 2 August 2026. Article 50 (transparency obligations) is cited in Section 18 only as the category of a finding. <https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng>

These are citations, not compliance verdicts. What the product delivers in this context is **structured evidence** — signed artifacts, receipts, provenance-bound source data, reproducible verification paths. What it does not deliver, and never claims: legal advice; audit certification; regulatory approval; a "compliant" verdict about anyone, including itself; partnership with any regulator, foundation, vendor or data source absent a contract; or the substitution of a signed receipt for the judgment of qualified legal, compliance and audit professionals. AgentNOMOS itself has an open transparency-disclosure finding under the EU AI Act (Section 18).

23.2 Disclaimer

This document is provided for information purposes by FeedOracle Technologies. It is not an offer, not investment advice, not legal or regulatory advice, and not a certification of any system or organization. Statements about regulations summarize public legal acts and may be superseded; readers must consult the official journals and their own advisers. Product capabilities are described with explicit scope and dates; nothing here promises fitness for a particular regulatory or security purpose. Cryptographic evidence produced by the product demonstrates the integrity and origin of specific records within the stated scope; it does not by itself establish compliance with any law, the truth of any provider's output, or the security of the system that produced it. Research results described here are research, not product capabilities, competition results or certifications. Revenue figures are dated snapshots of uncertain buyer genuineness. All trademarks belong to their owners; NVIDIA and OpenShell are named as technology only, and references to third-party protocols, platforms and operators do not imply endorsement by, partnership with, or certification from them. AgentNOMOS is software infrastructure; it is not sentient, exercises no legal agency, and every consequential authority it holds is delegated, bounded and revocable by its operator.

AgentNOMOS — The Machine Economy Trust Chain. Govern before action. Verify execution. Record outcomes.