PUBLIC

When AI hedges and policy commits

Anatomy of agent–policy disagreement on UK procurement decisions, signed and verifiable

Sam Carter, MeshQu

MRP-2026-02 · v1.0 · STABLE · Published 2026-05-18

predictions-lock bd7a795 · corpus 1b6192df6eb5…
decision-assurance · ai-governance · procurement · verifiability · sigstore

MeshQu

Contents

CLASSIFICATION	PUBLIC
VERSION	1.0
PREDICTIONS-LOCK	bd7a795
CORPUS SHA	1b6192df6eb5d3c3
GENERATED	2026-05-21

Abstract

Regulated firms cannot routinely answer how a specific AI-augmented decision was made. The agent's reasoning sits in application logs; the governing policy lives in a separate document; the decision itself is a database row. Six months later the answer is reconstructed from three sources that were never bound to each other — a story told after the fact, not evidence. We ran 300 public UK Contracts Finder filings through an LLM agent and the same records through a MeshQu policy evaluator at the same moment. Both verdicts, the agent's reasoning text, the exact policy snapshot, and a substrate provenance envelope were bound into a single Ed25519-signed receipt anchored to Sigstore Rekor. The full 283-decision corpus is published, downloadable, and verifiable offline. MeshQu produced 144 ALLOW and 139 DENY; the agent produced 7 ALLOW, 276 REVIEW, and zero DENY. Naive agreement is 7 of 283. The corpus inverts the pre-registered prediction: the disagreement shape is non-commitment under incomplete evidence, not over-permissiveness. A counterfactual that demotes a single rule from a critical-by-default DENY to a REVIEW band lifts agreement roughly elevenfold — a finding about policy authoring, not agent capability.

AUTHORS	Sam Carter, MeshQu
PUBLISHED AT	"2026-05-18"
VERSION	1.0
CLASSIFICATION	PUBLIC
STATUS	STABLE
PREDICTIONS LOCK	v0.1-predictions-locked · bd7a795 · 2026-05-15
CORPUS SHA256	1b6192df...c232be0
SUBSTRATE	UK Contracts Finder OCDS, 2024-12-01 to 2026-04-30
MODEL	gpt-5.4-2026-03-05, temperature=0

Predictions-lock
v0.1-predictions-locked bd7a795
2026-05-15

1. The question

When a regulated firm deploys an AI agent inside a decision workflow, a question follows it: how was this decision made? Most firms cannot answer it well. The agent's reasoning sits in application logs. The policy the agent was meant to follow lives in a separate document, version-controlled somewhere else. The decision itself is recorded as a row in a database. Six months later, when a regulator asks the question or a customer disputes the outcome, the firm reconstructs the answer from three sources that were never bound to each other. That reconstruction is not evidence. It is a story told after the fact.

The supervisory worry is not abstract. UK Procurement Policy Notice 02/24 (May 2024) named LLM-generated bid content as a specific accuracy and hallucination risk in public-sector procurement; PPN 017 (2025) extended its operational guidance to AI-augmented contract decisions; the UK Government AI Playbook (February 2025) makes meaningful human control one of its ten principles. The same shape recurs in EU AI Act high-risk provisions on automated decision-making, in SEC examination priorities on AI in investment advice, and in FCA, MAS, and BaFin AI guidance. Different regulators, same question — who can show how a specific decision was made, and prove that the showing is what actually happened?

MeshQu builds infrastructure for binding decisions to their evidence at the moment they are made. Each decision produces a signed receipt, bound to the exact policy snapshot evaluated against it, replayable by anyone holding the public key. We wanted to see what that looks like at corpus scale. So we ran an experiment.

We passed 300 public UK procurement filings to an LLM agent. The agent was asked to review each filing and recommend a verdict — approve, deny, or flag for review — and to cite the policy clause that justified its decision. The agent was not given the policy text. It reasoned from its training data. Every decision was then recorded through MeshQu against a documented procurement-compliance policy. The resulting corpus of 300 signed receipts is published, downloadable, and verifiable offline.

Before any of the runs, we committed a set of predictions to a public repository — what we expected the corpus to show, what would falsify each prediction, what we deliberately did not predict. The lock commit is tagged `v0.1-predictions-locked` at `bd7a795` (2026-05-15).

This is what the receipts look like.

2. How we ran it

The experiment has one moving part. A substrate adapter pulls a record from a public source. An LLM agent reads the record and produces a recommended verdict plus its reasoning. A MeshQu policy evaluates the same record against six rules and returns the platform's verdict. A signed receipt binds the record, both verdicts, the policy snapshot, and the agent's reasoning hash into a single object. That object is anchored to a public transparency log. End to end, one record produces one verifiable artefact.

The agent uses `gpt-5.4-2026-03-05` from OpenAI's API, accessed via the standard chat-completions endpoint at temperature 0. The model id is pinned. The temperature is pinned. The system prompt is committed to the public repository and its SHA-256 is folded into every receipt's hash payload as the `agent_prompt_sha256` field. The newer `gpt-5.5-2026-04-23` was available at experiment time but rejects `temperature=0` — it operates as a reasoning-style model under the hood, and the experiment's reproducibility commitment requires per-token determinism where the API allows. The reasoning-style model class is reserved for a separate follow-up experiment.

The evaluation loop is implemented as a small Python module in this repository (`runner/meshqu_runner/`). The design intentionally keeps the execution surface small: single provider, single model, single temperature, structured-JSON output. The runner adopts the discipline patterns common to evaluation frameworks like Inspect AI — locked models, structured outputs, persisted traces, bounded retries — while remaining compact enough to audit directly. A future multi-provider experiment would likely re-baseline onto Inspect AI directly; the patterns here would translate.

The agent is not given the policy text. It sees only the procurement record itself — not the policy evaluation, not the resulting receipt, not any prior MeshQu output. It reasons from its training data. The interesting signal is drift — where the agent's reasoning sounds confident but conflicts with what the policy actually enforces.

The substrate is UK Contracts Finder's OCDS Search endpoint. Records were fetched from a publication window of 2024-12-01 to 2026-04-30, which straddles the Procurement Act 2023 commencement date (24 February 2025). The fetcher paginated through release events `stages=award` and stopped at 300 records. The fetch was not a stratified sample. Two prior substrate spikes (Phase 0 and Phase 0.5) had established the feed's regime distribution and the load-bearing-field reliability for `PROC-001-S53` ; the production run fetched in chronological order against the published window, accepting the substrate's natural distribution. The OCDS feed publishes multiple release events per procurement (OCID); after deduplication the 300-record corpus contains 283 unique procurements.

The substrate analysis preceding pre-registration killed an earlier candidate rule and reshaped the design. The original `PROC-001-S44` rule depended on a presence-and-content check against linked s.44 transparency notices. 96% of OCDS records carry no PA23/PCR regime signal, framework call-offs dominate ~68.6% of records, and the s.44 linkage was unreliable in publicly-indexed records. The spike pivoted to `PROC-001-S53` — a timing rule against the publication-delay

field. That field is consistently populated in OCDS and the rule survives the substrate's known limitations. Governance regime is identified by an award-date proxy (records with `awards[0].date > 2025-02-24` are treated as PA23-governed). This is a documented methodological proxy, not a regime detector; findings on `PROC-001-S53` are scoped accordingly.

We controlled the model id, the model temperature, the system prompt content, the policy snapshot identifier, the substrate adapter version, the OCDS publication window, and the runner's git commit. Each is recorded in `run-manifest.json` and hash-bound through the receipt's integrity payload.

We did not control for OpenAI's day-to-day backend behaviour, the Contracts Finder feed's content updates, or the Rekor log's growth between the moment of decision and the moment of verification. Each is a known variable that does not affect any single receipt's verifiability.

A receipt records three things. **What was decided**: the procurement record's fields, the policy snapshot evaluated against, the verdict, the violations. **Who decided**: the agent's model id, temperature, prompt SHA-256, reasoning text SHA-256, recommended verdict, and recommended action — all bound into the same hash payload as the substantive fields. **That nothing changed afterwards**: an Ed25519 signature over the entire payload using the experiment's dedicated kid (`meshqu-experiment-procurement-2026-05`), and a Sigstore Rekor anchor whose inclusion proof can be verified independently of MeshQu's API.

The methodology described here is substrate-agnostic. UK Contracts Finder is the worked example because it is open, recent, and rich enough to support meaningful agent reasoning. The same structure applies to any historic-decision corpus where the source data, the governing policy framework, and the decision context can be cleanly separated.

3. The policy under test

The policy under test is six rules. One is a faithful implementation of a specific UK statute — Procurement Act 2023 s.53(1), the 30-day Contract Details Notice publication obligation. The other five are illustrative composites synthesised from named procurement frameworks across UK, EU, and US regimes. All six are deterministic. Each rule is a condition over a small number of substrate-derived fields, evaluated by the MeshQu policy engine against the procurement record under review. Rules do not reason over prose. They do not parse legal language. They evaluate field values against thresholds, against allow-lists, against existence checks.

Table 1. Policy rules under test.

Rule	Purpose	Trigger shape	Outcome
`PROC-001-S53`	Publication-delay timing	publication > 30 days after award	DENY
`PROC-002-AUTHORITY`	Contract-value authority	value exceeds authority threshold	DENY
`PROC-003-DEBARMENT`	Supplier exclusion list	supplier on exclusion list	DENY
`PROC-004-COI`	Conflict-of-interest disclosure	declaration present + flagged	DENY
`PROC-005-OPEN-TENDER`	Open-procedure / justification	open-tender flag absent + no justification	DENY
`PROC-006-MOD-CAP`	Modification-value cap	modification exceeds permissible ratio	DENY

`PROC-001-S53` as it executes — lifted verbatim from the ratified policy snapshot in any `corpus.tar` bundle's `policy_snapshot.json`:

The policy's semantics are intentionally binary. Every rule is authored at `critical` severity, so the evaluator projects any satisfied rule condition directly to DENY. The policy contains no native representation of evidentiary uncertainty or procedural incompleteness. That design choice becomes important in §5, where the agent's three-state verdict space — ALLOW, REVIEW, DENY — encodes information the binary policy does not. The cardinality mismatch is a property of the policy's authoring, not the platform. `PROC-001-S53` is the rule the experiment leads with for the drift case study; the others appear in distribution counts and in violation co-occurrence patterns.

4. The substrate

The substrate is UK Contracts Finder's OCDS Search endpoint, licensed under Open Government Licence v3.0. Records are pulled from a publication window of 2024-12-01 to 2026-04-30 — the window straddles the Procurement Act 2023 commencement date (24 February 2025), so the corpus contains both pre-PA23 and PA23-governed contracts. The fetcher paginates by date via the API's `stages=award&publishedFrom=…&publishedTo=…&limit=100` parameters; pagination follows the response's `links.next` URL verbatim.

The 300-record fetch was not a stratified sample. Date-window pagination accepts the substrate's natural distribution across award method, value band, and governance regime — the methodology trades off pre-registered sampling neatness for honest reflection of what the feed actually publishes. The Contracts Finder feed publishes multiple release events per procurement when buyers update or amend notices; the 300 release events in the corpus represent 283 unique procurements after OCID deduplication.

The Phase 0 and Phase 0.5 substrate spikes preceded predictions-lock. They established the feed's regime distribution (no explicit PA23/PCR regime field on OCDS records, roughly 96% of records lacking any direct regime signal) and the load-bearing-field reliability check that pivoted the headline rule from `PROC-001-S44` to `PROC-001-S53`.

Receipts were generated against a dedicated MeshQu tenant on the staging environment, signed with an Ed25519 key whose public half is published alongside the corpus. Verification is environment-independent — the bundle includes the public key needed to verify offline.

The corpus serves two purposes: it is the empirical evidence for §5's findings, and it is production-scale evidence that MeshQu's infrastructure — signing, anchoring, bundling, verification — works reliably on real external data. Every receipt in the corpus was produced by the same code path that runs in MeshQu's production environment, signed by an Ed25519 key whose public half is registered in `verify.meshqu.com`'s source-code trust registry, and anchored to Sigstore Rekor at the moment of decision. Operational behaviour during the run is documented in Appendix B — five captures from the Grafana observability dashboard showing the production run from idle baseline through sustained throughput to completion.

Decision ca19e737-…ec97d2fe5859
Rules fired
PROC-001-S53
PROC-002-AUTHORITY
PROC-005-OPEN-TENDER
Rekor log index
1566819550

5. What the corpus shows

5.1 Volume and verdict distribution

The corpus is 300 OCDS release events from the UK Contracts Finder feed, fetched over a publication window that straddles the Procurement Act 2023 commencement date. Each event was passed through the substrate adapter, the agent, and the MeshQu policy evaluator in sequence. The run completed in 33 minutes 30 seconds wall-clock with zero anomalies, zero orphaned receipts, and zero records skipped. After OCID deduplication, the corpus contains **283 unique procurement records**.

MeshQu's verdicts split 144 ALLOW and 139 DENY across the 283 unique decisions — a 51 / 49 distribution that is close to balanced. The agent's verdicts split 7 ALLOW and 276 REVIEW. **The agent produced zero DENY verdicts in the corpus.** Headline naive agreement — counting only records where both verdicts read identically — is 7 of 283, or 2.5%. See Fig. 1.

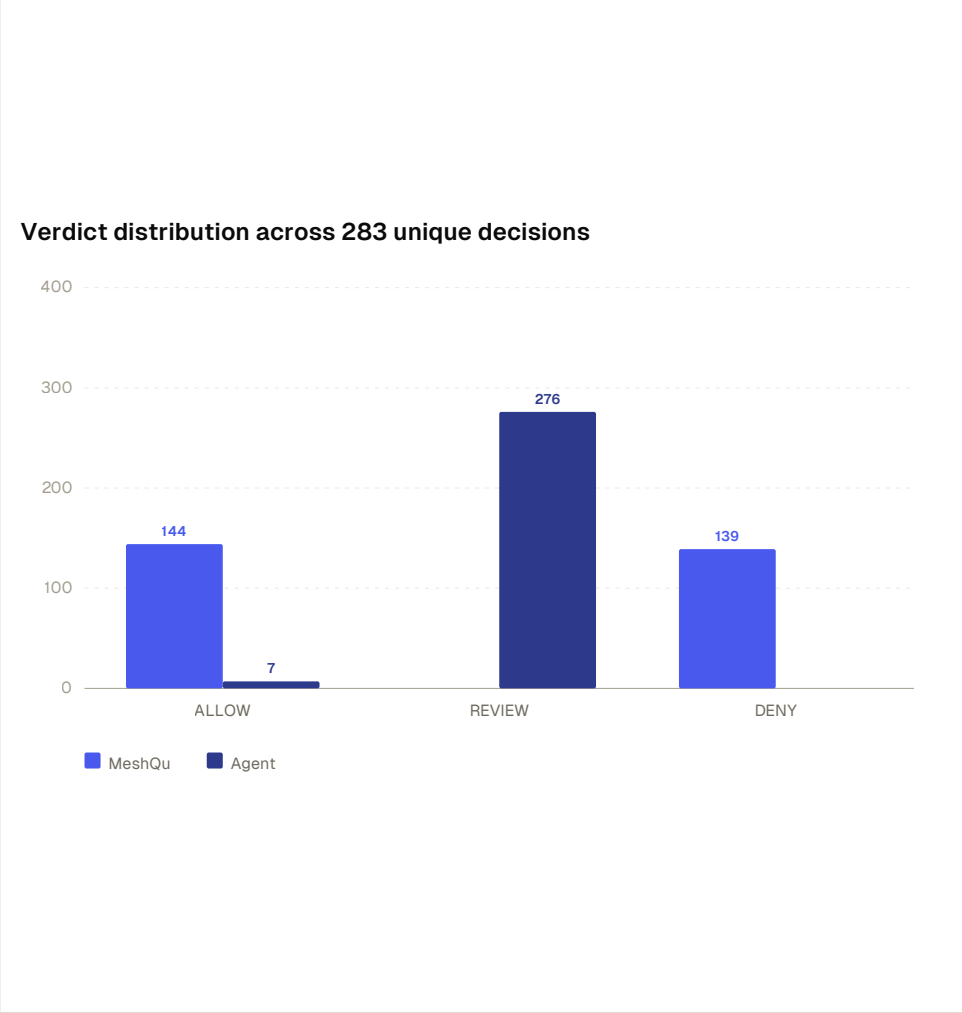


Fig. 1 — MeshQu produces a committed binary verdict; the agent produces a verdict plus a hedge. · The agent's REVIEW class encodes information the binary policy projects away.

Substrate provenance across the corpus is informative on its own. Of 2,830 substrate cells (283 records × 10 fields), roughly 20% are direct-OCDS reads, 21% are deterministic derivations from OCDS data, 30% are documented proxies where

OCDS does not carry the substantive field, and 29% are honest omissions where the field is unavailable for a given record. The 30% proxy / 29% absent fraction is the substrate honesty in numbers.

The rule-firing distribution across the corpus is shown in Fig. 2.

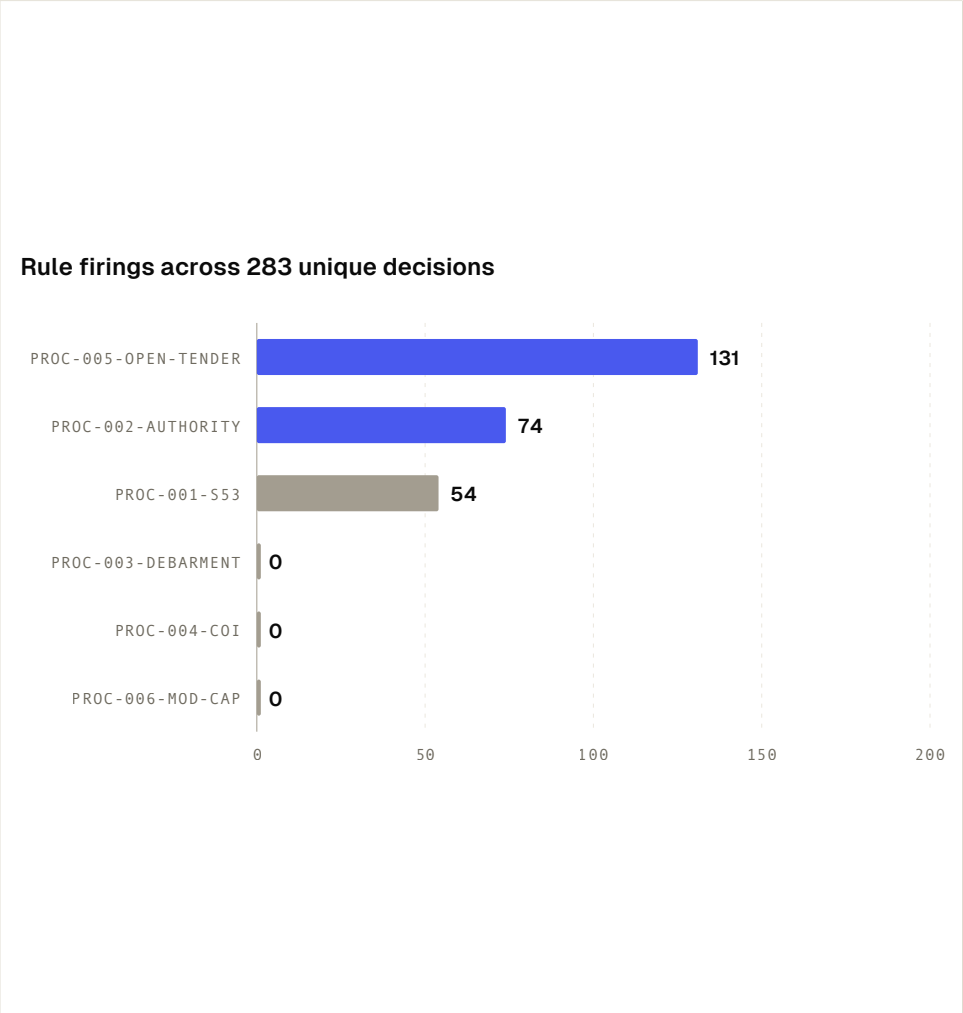


Fig. 2 — PROC-005 dominates the DENY column; three rules never fired in the corpus. · A quiet rule is information, not a bug; the three zero-fire rules are reported honestly.

5.2 Agent-vs-policy disagreement

The pre-registered prediction was that the agent would lean ALLOW relative to MeshQu's DENYs — over-permissive by perhaps 15 to 25 percent of cases. The corpus shows the opposite shape. The agent does not lean ALLOW. The agent does not commit to DENY at all. The absence of DENY verdicts is not explained by an absence of problematic records — MeshQu produced 139 DENY outcomes across the same corpus, including 27 records with three concurrent critical violations. The agent reaches for REVIEW on 97.5% of records, including records that MeshQu finds clean and including records where MeshQu names three concrete rule violations. The prediction anticipated the wrong failure mode. The corpus reveals a structural divergence rather than a simple error rate.

Decision ID `ca19e737-defb-4e5f-b216-ec97d2fe5859` is one record from the corpus. The bundle is in `corpus.tar` and the verifier round-trip is shown in the verification block and Fig. 3 below.

1. The procurement. A £57,000,000 award, contract value above the PA23 authority threshold (£500,000), award date after PA23 commencement, publication delay 33 days, procurement-method-open flag absent in the source, direct-award justification not present.

2. What the agent saw. The substrate adapter passed the agent the procurement fields and a per-field provenance envelope. The agent was not given the policy text, not given any prior MeshQu output, not given any other record's receipt.

3. What the agent reasoned. Verbatim from `agent_outputs/ca19e737-….json`:

> *"This is an above-threshold £57m award governed by PA23, but the record shows a selective procedure and no direct-award justification is present. Publication was 33 days after award, so the audit trail is incomplete for a high-value procurement."*

The reasoning names three distinct issues: above-threshold value, selective procedure without direct-award justification, and a 33-day publication delay. Each of those names maps onto a specific MeshQu rule territory — PROC-002 (authority threshold), PROC-005 (open-tender or justified-direct-award), PROC-001-S53 (s.53 30-day window).

4. The agent's verdict. REVIEW. Recommended action: "Obtain procedure rationale and notice trail."

5. MeshQu's evaluation. Three concrete violations under the ratified policy snapshot `cbf12348-…`:

- `PROC-001-S53` — publication delay 33 days exceeds 30-day maximum (`VALUE_ABOVE_MAX`)
- `PROC-002-AUTHORITY` — contract value £57M exceeds £500k maximum (`VALUE_ABOVE_MAX`)
- `PROC-005-OPEN-TENDER` — procurement-method-open flag missing (`FIELD_MISSING`)

Verdict: DENY.

6. The receipt. Ed25519-signed, anchored to Sigstore Rekor at log index 1,566,819,550, bundled with the policy snapshot and trusted-key envelope. Independently verifiable: any reader can extract this bundle from `corpus.tar` and round-trip it through `verify.meshqu.com` or the `@meshqu/verifier` CLI. Both paths return all five cryptographic checks green.

```
VERIFICATION

decision_id: ca19e737-defb-4e5f-b216-ec97d2fe5859
command: meshqu-verifier verify bundles/ca19e737-….bundle.json
expected: PASS — all five cryptographic checks green
result: PASS
signer_kid: meshqu-experiment-procurement-2026-05
rekor_log_index: 1566819550
policy_snapshot: cbf12348-...
```

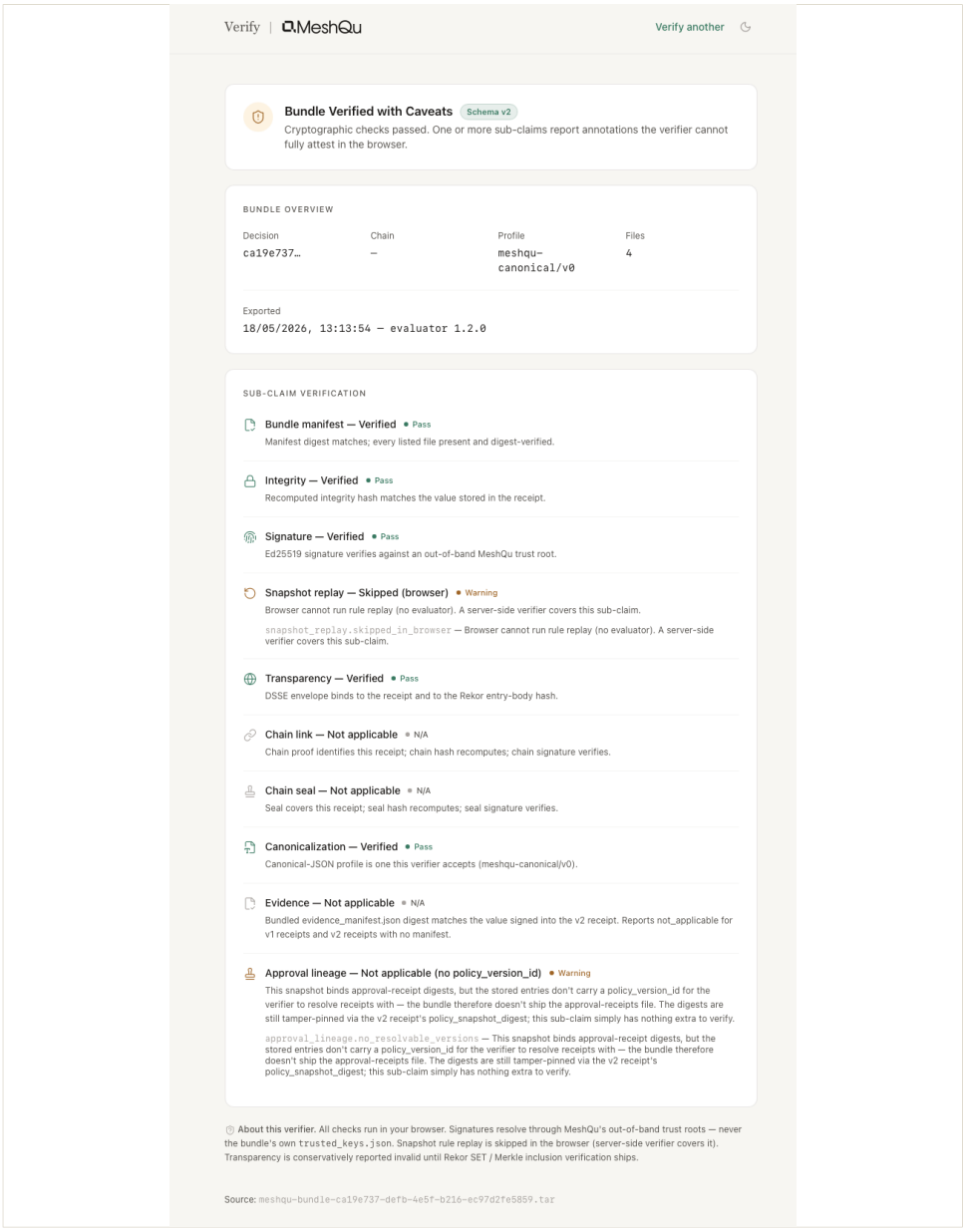


Fig. 3 — Bundle verification for the £57M record at verify.meshqu.com — all five cryptographic checks green.

The browser verification view for this bundle is shown in Fig. 3. One record. Two assessments. Agent names every issue MeshQu finds. Both agree the record warrants attention. Verdicts read 100% disagreement.

The worked example is not anomalous. Across the corpus, the agent's `recommended_action` text consistently names the rule territories MeshQu actually flags. The seven ALLOW / ALLOW agreements are seven records where MeshQu found no violations and the agent chose ALLOW. On every other record — including 132 records where MeshQu's DENY is supported by one or more critical violations — the agent chose REVIEW. **What the corpus measures is not "agent right or wrong." What the corpus measures is two systems with different verdict spaces examining the same evidence.** MeshQu produces a committed binary verdict; the agent produces a verdict plus a hedge.

Most-fired rule (P2). PROC-005-OPEN-TENDER fired on 131 of 283 records — roughly 46% of the corpus and 94% of MeshQu's DENY column. The rule fires when a procurement is above-threshold and the source data does not carry an open-tender marker and no direct-award justification is recorded. The corpus is dominated by records where the procurement-method flag is simply absent in OCDS — a substrate condition rather than a buyer choice. PROC-002 fired on 74 records, PROC-001-S53 on 54, and the other three rules fired zero times in 283 records.

Hallucinated citations (P3). The agent was expected to invent or misapply specific regulatory clauses some fraction of the time. We did not observe this in the corpus. Across the records reviewed by hand, the agent's `recommended_action` text is consistently generic — "verify procedure basis", "obtain procedure rationale", "verify award timeline" — and does not cite specific clauses, sections, or directives.

Direct-award disagreement (P6). The prediction expected disagreement to cluster on direct-award procurements. The corpus contained too few direct-award records to evaluate the prediction at a meaningful sample size. We do not report a result on P6 from this corpus.

The 2.5% naive agreement is shaped by a verdict-space mismatch: a three-state agent assessing the same records as a two-state policy. The MeshQu platform supports a REVIEW verdict; the specific procurement policy used in this experiment was authored with all six rules at `critical` severity, which the evaluator reduces to "any violation → DENY". A different policy authoring choice would produce different verdicts on the same corpus, evaluable through the existing `decision_traces.jsonl` without any further runs.

Three counterfactual policies, each layered onto the same 283-decision corpus, are summarised in Table 2.

Table 2. Counterfactual verdict distributions over the same corpus.

Scenario	DENY	REVIEW	ALLOW	Agreement vs agent
As-ratified (binary)	139	0	144	7 / 283 (2.5%)
PROC-001-S53 with a 31–60-day REVIEW band	139	0	144	7 (unchanged)
PROC-001-S53 and PROC-002 with REVIEW bands	137	2	144	9
Above plus PROC-005 mapped to "needs more context"	64	75	144	82 / 283 (29%)

The pivotal shift occurs in the final counterfactual. Demoting PROC-005 from a critical-by-default DENY to a "needs more context" REVIEW — which is the rule's actual semantic, since the missing flag is a question about the record rather than a finding against it — produces 75 new agreements between the agent's REVIEW and MeshQu's hypothetical REVIEW. **Agreement increases roughly elevenfold under the counterfactual.** This isn't trivial overlap of the kind that says "any agent-says-REVIEW policy will produce more agreement." The agent's reasoning text on these records names the same evidence gaps the rule fires on — open-procedure flag absent, direct-award justification not present (as in the §5.2 worked example) — so the demoted-PROC-005 REVIEW band reflects what both systems already observed about the same records. The agent's caution is not generic noise. It correlates strongly with records the policy would have produced REVIEW for if the rules had been authored as a verdict gradient rather than a binary cliff edge.

This is a finding about policy authoring, not about agent capability. The agent's REVIEW class encodes information that a binary policy projects away.

Decision cba375f2-…5811e492dfa4
Verdict (MeshQu) DENY · PROC-005-OPEN-TENDER
Verdict (agent) REVIEW

6. Reasoning is data

The agent's reasoning is part of the receipt. Not a side-channel log to be reconciled later. Not a debugging artefact captured for completeness. The reasoning text is hashed into the receipt's integrity envelope alongside the model id, the prompt SHA-256, and the agent's recommended verdict. A reader holding the bundle can read what the agent said, see exactly which model said it under which prompt, and verify that the text has not been edited since it was written. Reasoning treated as evidence rather than as logs is the part of MeshQu's primitive that this section examines.

6.1 One full receipt

Decision ID `cba375f2-d526-4fca-abda-5811e492dfa4` is a £139,960 procurement award. Under the experiment's substrate, that value places the record fractionally above the PA23 sub-central services threshold proxy. The procurement-method-open flag is absent. No direct-award justification is recorded. The publication delay is seven days — well inside the 30-day window.

The agent's verbatim reasoning, as recorded in the agent-outputs sidecar and hash-bound into the receipt:

> *"This appears above-threshold under PA23 (£139,960) and uses a selective method rather than open, but no direct-award justification is evidenced and the substrate notes a known false-negative risk. Publication was only 7 days after award, so the main issue is whether the chosen procedure and justification are properly documented."*

The reasoning is **semantically complete**. The agent identifies the proximity to the threshold, names the selective-procedure choice, notes the missing justification, reads the substrate's own honesty marker about false-negative risk on linked notices (the substrate adapter records per-field provenance — direct_ocds, derived, proxy, absent — and the agent saw those markers and used them), and observes that the timing dimension is not the operative concern. None of the agent's claims are invented. The disagreement emerges after accurate observation, not before it. None of them cite specific regulations. The reasoning describes the record's compliance shape in plain language.

The agent's verdict is REVIEW. Its recommended action is "Verify procedure and award justification records."

MeshQu's policy returns DENY with one violation: `PROC-005-OPEN-TENDER` (the open-tender flag is missing on an above-threshold procurement with no recorded direct-award justification). Both systems looked at the same record. The agent reads the absence of evidence and reaches for caution. The policy reads the absence of evidence as the satisfied condition of its rule. **The disagreement is not about the facts. The disagreement is about how to respond to incomplete facts.**

6.2 What a compliance officer can do with this receipt

Open the bundle. The agent's reasoning is right there, verbatim, exactly as written by `gpt-5.4-2026-03-05` at temperature zero under a prompt whose SHA-256 is bound into the same envelope. The policy snapshot the decision was evaluated against — the six rules, their thresholds, their `when` clauses — is in the bundle

alongside the reasoning. The violations are named with their rule codes, their severity, the field that triggered them, and the structured reason code. The Ed25519 signature verifies under the experiment's published kid. The Sigstore Rekor inclusion proof verifies independently of MeshQu's infrastructure. Six months from now, a reader handed this bundle can reproduce the decision exactly. **Reconstruction is not proof. Replay is.**

6.3 Evidence incompleteness as a governance state

The corpus surfaces a pattern that generalises beyond procurement. PROC-005-OPEN-TENDER fired on 131 of 283 records — and on every one of those, the field MeshQu missed was the record's procurement-method marker, not evidence of buyer misconduct. The rule reads "above-threshold, no open-tender flag, no direct-award justification → violation." The substrate produces records where the rule condition is satisfied by missing metadata rather than by explicit evidence of misconduct. The agent saw the same records and consistently reached for REVIEW, naming the missing documentation explicitly in its reasoning. Same evidence. Two systems with different responses to incompleteness. **The agent's REVIEW class is a compressed encoding of "I cannot verify what I cannot see" — a verdict primitive the binary policy did not have.** Procurement is one expression of the pattern. AML, KYC, underwriting, AI oversight all face the same shape: rule engines treat missing evidence as either pass or fail; competent reviewers treat it as a question.

6.4 The technical insight

A signed receipt that carries the agent's reasoning is the contract between an AI-augmented decision and everyone who has to defend it later. The compliance officer reading it six months on, the auditor reviewing it without access to MeshQu's infrastructure, the regulator asking how a specific contract was approved, the customer disputing the outcome — they all read the same artefact. They all see the same reasoning text. They all verify the same signature. **Treating reasoning as data, not as logs, is what makes that contract enforceable.**

7. Limitations

- **Substrate.** UK-only and English-language. Procurement vocabulary, statutory frameworks, and publication conventions differ across regimes; findings here generalise carefully or not at all. The Contracts Finder OCDS feed publishes multiple releases per procurement when buyers update or amend a notice; 300 release events represent 283 unique procurements after OCID-level deduplication. Two methodological proxies are imposed by what the feed actually carries: governance regime is identified by award date relative to PA23 commencement because OCDS records carry no explicit regime field; s.53's 30-day clock runs from contract signature date legally, but OCDS exposes award decision date — these are typically close but legally distinct.
- **Policy.** Five of the six rules are illustrative composites synthesised from named procurement frameworks (UK PA23, EU Directive 2014/24/EU, US FAR). Only `PROC-001-S53` is a faithful implementation of a named statutory time-window; the composites are not certified by any regulator. The policy is binary by authoring choice — every rule is marked critical, so any violation produces DENY. The cardinality mismatch with the agent's three-state verdict space is a policy-authoring choice, not a platform limitation. Receipts validate decision integrity, not policy correctness — a flawed policy correctly enforced still produces a clean receipt.
- **Apparatus.** Single foundation model (`gpt-5.4-2026-03-05`) at a single version, temperature, and prompt. Results may not generalise across model classes; reasoning-style models (GPT-5.5+, o-series) are the subject of Follow-up B. Sampling was date-window pagination, not stratified — the corpus accepts the substrate's natural distribution rather than constructing a 3×3 grid.
- **Verification and review.** The raw-receipt-paste verification path at `verify.meshqu.com` warns "Tampered" on receipts whose envelope includes server-injected metadata; the bundle verification path is canonical and returns clean cryptographic checks across the corpus. Disagreement cases were reviewed by the experimenter against published procurement frameworks rather than by independent procurement-law experts.
- **Reproducibility.** LLM non-determinism may exceed any pre-registered band. The corpus is one run. The reproducibility band itself remains a hypothesis that a future rerun would test directly.

Each limitation is reported in the spirit of inviting further scrutiny. The corpus, the policy snapshot, the prompt, and the runner are all published; a reader who wants stronger ground truth on any of these dimensions can produce it directly.

8. Reproduce it yourself

The corpus is published as a single archive at `procurement-decisions/results/corpus.tar`. 283 v2 bundles, SHA-256 `1b6192df6eb5d3c38738b6abc5cea82c92d99d53ae890308569a4c240c232be0`, 5.3 MB uncompressed. A reader who wants to confirm the corpus integrity claim of this paper does so in two commands:

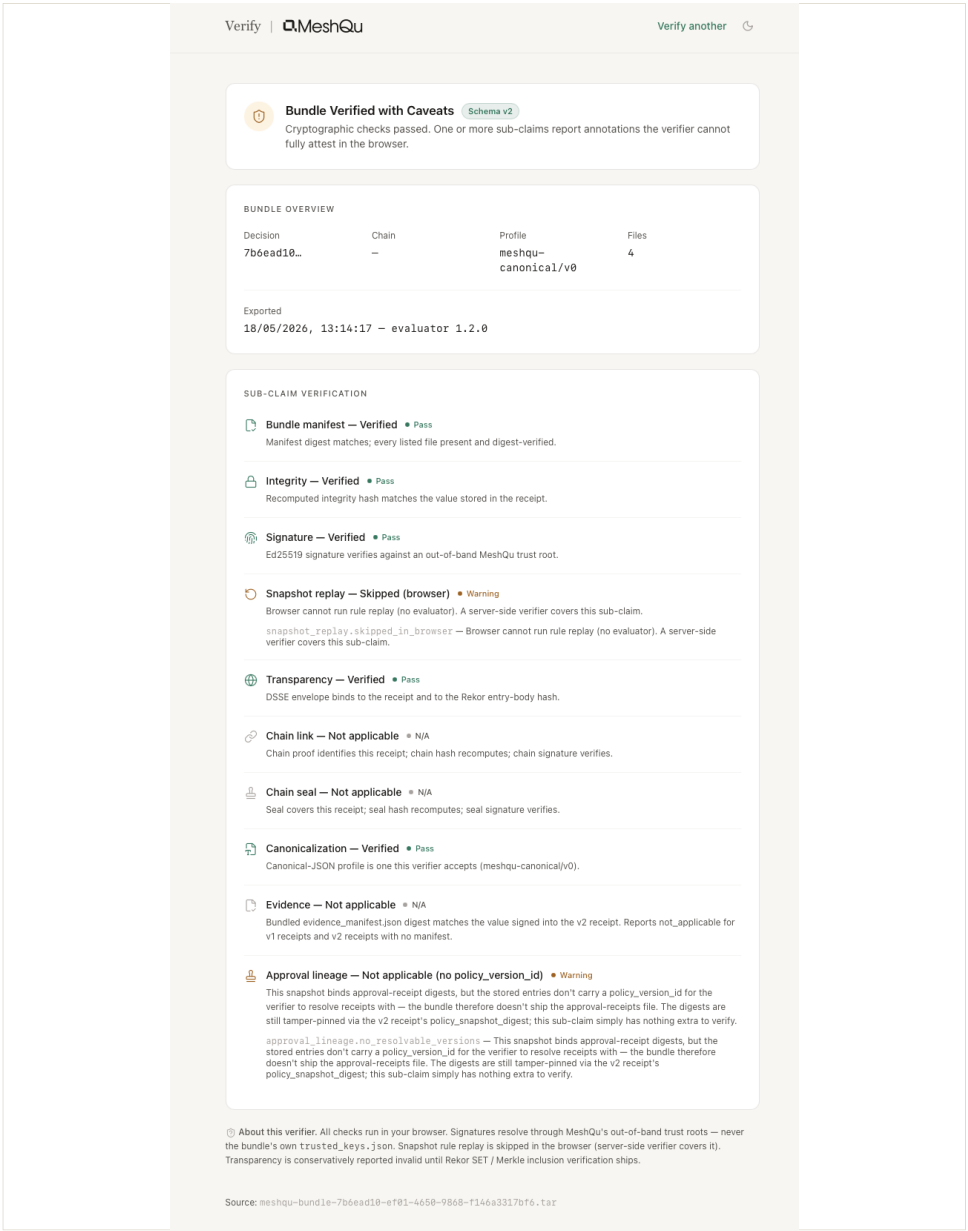


Fig. 4 — Browser verification of bundle 7b6ead10-… at verify.meshqu.com — Bundle Verified with Caveats (Schema v2). Five cryptographic checks green; two non-blocking warnings documented in the corpus README.

The browser equivalent — drop the same bundle file into `verify.meshqu.com` (see Fig. 4) — produces the same five green cryptographic checks plus two known non-blocking warnings documented in the corpus's `README.md`. Verify one bundle or verify the full corpus by iteration; the cryptographic result is the same. Verify

offline via Sigstore Rekor by pulling the entry directly from `https://rekor.sigstore.dev/api/v1/log/entries/<entry_uuid>` for any receipt — that path doesn't depend on MeshQu's infrastructure at all.

A reader who wants to rerun the experiment rather than simply verify the corpus works from the runner module at `procurement-decisions/runner/`. The locked model id, system prompt, substrate adapter, and policy snapshot identifier are all in the repository; an OpenAI API key, a MeshQu staging credential pair, and roughly one evening reproduce the corpus end-to-end against the same OCDS window.

The verification proves three things. **What was decided**: integrity hash matches → the receipt's fields are the ones that were signed. **Who decided**: signature verifies under the published key → the experiment's signer produced this. **That nothing changed afterwards**: Rekor inclusion proof verifies → the receipt existed at the anchored timestamp and has not been altered since. Three claims, three checks, two commands. The corpus is not assertion. It is verifiable evidence.

9. What's next

The corpus raises one question most directly. The agent in this experiment reached for REVIEW on 97.5% of records — never committing to a verdict, naming evidence gaps in its reasoning, encoding caution that the binary policy projected away. We do not know whether that pattern persists when the agent has the policy text in hand, or whether it represents something deeper than context-poverty alone.

Three experiments form a coherent progression rather than disconnected follow-ups.

Experiment 1 is this corpus. A fixed foundation model reviews real procurement records without policy visibility; MeshQu evaluates the same records against executable policy and produces signed receipts. The finding is evidence-sensitive caution: the agent reaches for REVIEW under incomplete or ambiguous evidence.

Experiment 2 extends into a governance-context gradient. Same records, same model, progressively richer context: structured DecisionContext, Decision Receipts, named policy violations, full policy text. The central question is whether explicit governance artefacts reduce ambiguity-driven escalation — and whether MeshQu's structured outputs function as useful governance context for AI systems, not just as audit trails generated from them.

Experiment 3 introduces a true evidence-seeking agent. Instead of reviewing static records, the agent actively investigates uncertainty: retrieves linked notices, inspects documents, verifies timelines, gathers evidence. MeshQu governs the investigation process itself — recording tool usage, evidence provenance, intermediate policy evaluations, and final outcomes as replayable, cryptographically verifiable receipt chains. The research question shifts from "can AI review procurement records?" to "can AI-assisted investigations become audit-grade, replayable, and governable?"

Passive reviewer → context-aware reviewer → governed investigative agent. Each experiment compounds on the same methodology infrastructure: substrate adapters, executable policy, replayable evaluation, signed receipts, independent verification.

The pattern — evidence incompleteness as a first-class governance state — generalises beyond procurement to AML, KYC, underwriting, and AI oversight; the methodology there is one substrate-adapter and one policy-authoring pass away. The harness is built around a substrate-adapter abstraction. Each of these extensions is a substrate-adapter implementation plus a domain-specific policy authoring pass — not a rebuild.

10. Declaration of AI assistance

AI tools were used during ideation, drafting, and editorial refinement of this paper. The data collection, policy authoring, cryptographic protocols, and analytical conclusions were directed and reviewed by the author. In a paper on auditable AI decision-making, disclosing the assistance trail is the same primitive the paper advocates for — making the work legible at the point of the work.

References

[01] Cabinet Office. Procurement Policy Notice 02/24 — Tackling Modern Slavery in Government Supply Chains, with guidance on AI use in tendering. UK Government, May 2024.

[02] Cabinet Office. Procurement Policy Notice 017 — Operational guidance for AI-augmented contract decisions. UK Government, 2025.

[03] Cabinet Office. UK Government AI Playbook. UK Government, February 2025.

[04] European Parliament and Council. Regulation (EU) 2024/1689 — Artificial Intelligence Act (high-risk provisions on automated decision-making). Official Journal of the European Union, 2024.

[05] European Parliament and Council. Directive 2014/24/EU on public procurement. Official Journal of the European Union, 2014.

[06] UK Parliament. Procurement Act 2023, s.53(1) — Contract Details Notice publication obligation. 2023.

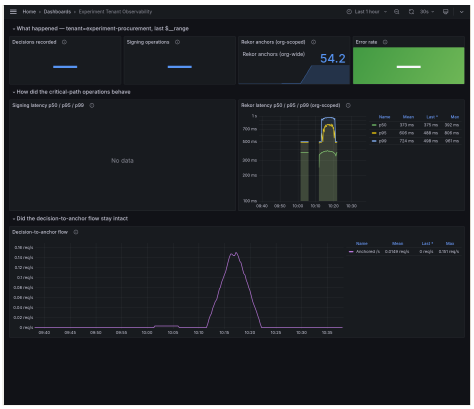
[07] Sigstore project. Rekor — transparency log for software artifacts. 2024. https://docs.sigstore.dev/logging/overview/

Appendix A. Predictions vs results

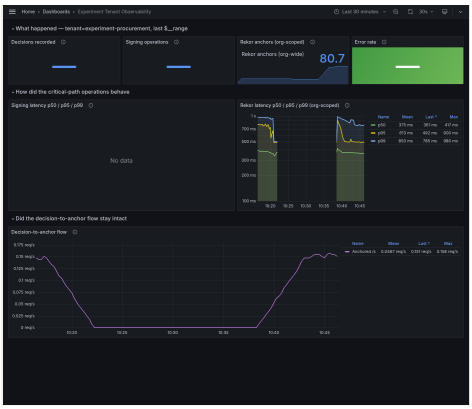
ID	PREDICTION (LOCKED 2026-05-15)	OBSERVED	STATUS
P1	Agent over-permissive vs MeshQu DENYs (15–25%)	Agent REVIEW-by-default (97.5%); 0 agent DENYs; naive agreement 7/283 = 2.5%	Inverted — disagreement shape was non-commitment, not over-permissiveness
P2	Top-2 violation drivers account for >60% of MeshQu denials	PROC-005-OPEN-TENDER (131) + PROC-002-AUTHORITY (74) = 205 of 259 total critical violations across 139 DENYs (79%)	Confirmed
P3	≥5% of agent reasoning narratives cite specific regulatory clauses; some fraction wrong	No specific clause / section / directive citations observed; agent's `recommended_action` consistently generic	Premise unmet under this model and prompt; alternate conditions untested
P4	Verdict non-determinism in 5–20% range across re-runs at temperature 0	Untested — corpus is one run; reproducibility-rerun is a separate experiment	Deferred
P5	100% of bundled receipts verify offline at `verify.meshqu.com`	Confirmed on sample (`7b6ead10–…` , `ca19e737–…`); see Fig. 3 in §5b and Fig. 4 in §8	Confirmed
P6	Disagreement higher for direct-award procurements vs competitive	Corpus contained too few direct-award records to evaluate at meaningful sample size	Substrate-limited; prediction remains open for future runs

Appendix B. Operational captures from the production run

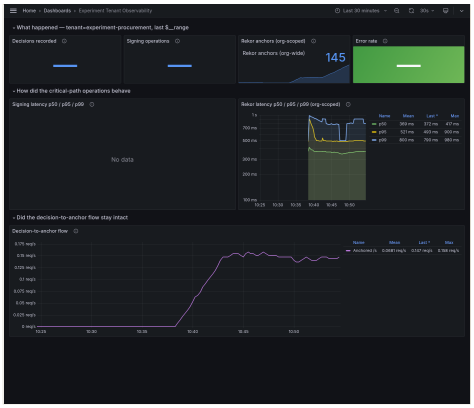
Five captures from the experiment's Grafana observability dashboard, selected from the 152 checkpoints recorded across the 33-minute run. Each capture is the same dashboard view at a different point in the run; together they document the production-system behaviour during corpus collection.



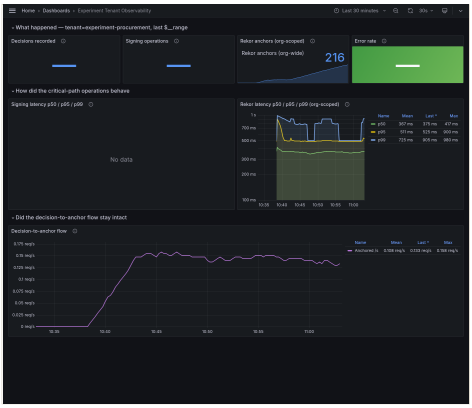
Run-start. Decision-to-anchor pipeline idle; tenant key and policy snapshot loaded.



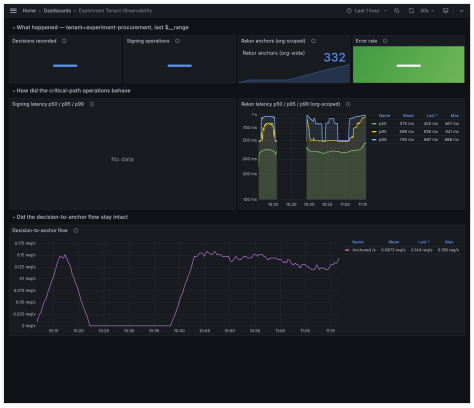
Checkpoint 076 of 300. Early-run; receipts issuing at steady rate.



Checkpoint 150 of 300. Mid-run; sustained throughput, zero anomalies.



Checkpoint 226 of 300. Late-run; pipeline uninterrupted through the substrate's mixed-regime distribution.



Run-end. 300 records processed, 33m 30s wall-clock, zero anomalies, zero orphaned receipts, zero records skipped.

Appendix C. Counterfactual analysis — violation co-occurrence

In the as-ratified policy, 139 of 283 records receive DENY. Across those 139 DENYs the substrate produces 259 distinct critical-rule firings — roughly 1.86 firings per denied record. PROC-005-OPEN-TENDER appears in 131 of the 139 DENYs (94%); PROC-002-AUTHORITY in 74 (53%); PROC-001-S53 in 54 (39%). Twenty-seven records carry all three rules concurrently; those are the records on which the binary policy produces its strongest output and the agent produces its most-hedged output. Demoting PROC-005 to REVIEW shifts the 27 triple-violations into the REVIEW band along with the broader pool, which is what produces the elevenfold lift in agreement reported in §5b.