Pilot Evidence Packet v1.0

SHADOW Mode External Log Ingestion

MathLedger Project

2025-12-17

Status: COMPLETE / FROZEN (SHADOW MODE)

1 Purpose

This document describes a **SHADOW ingest interface for external logs**. The pilot ingestion adapter receives external JSON/JSONL logs, attaches toolchain provenance, and routes data to an isolated namespace (`governance.external_pilot.*`).
Mode: SHADOW (observational only)
Action: LOGGED_ONLY
Enforcement: None

2 What This Demonstrates

Property	**Verification**
Namespace isolation	External data routed to `governance.external_pilot.*` only
Non-interference	Internal metrics, verifier state, CAL-EXP surfaces unchanged
Determinism	Repeated runs produce byte-identical outputs (timestamp-stripped)
Schema separation	Pilot schema does not extend v0.1 contract schemas

3 What This Does NOT Demonstrate

Non-Claim	**Explanation**
Validity	Pilot outputs do not constitute experimental evidence
Correctness	Execution does not imply correctness
Learning	No system learning or adaptation claims
Gating	Pilot outputs never block execution
Readiness	Does not demonstrate production readiness

4 Smoke Proof (Copy/Paste)

Run from repository root:

```
uv run python -c "
import json, tempfile
from pathlib import Path
from backend.health.pilot_external_ingest_adapter import (
    ingest_external_log, wrap_for_evidence_pack,
    attach_to_manifest, copy_to_evidence_pack,
)
with tempfile.TemporaryDirectory() as t:
    tmp = Path(t)
    log_json = tmp / 'sample_external.json'
    log_json.write_text(json.dumps({
        'log_type': 'pilot_demo',
        'entries': [{'event': 'smoke', 'value': 1}]
    }))
    pack = tmp / 'evidence_pack'
    pack.mkdir()
    r = ingest_external_log(log_json)
    e = wrap_for_evidence_pack(r, log_json)
    m = attach_to_manifest({'governance': {}}, [e])
    print(f'result={r[\"result\"]} mode={m[\"governance\"][\"external_pilot\"][\"mode\"]}')
"
```

Expected output:

```
result=SUCCESS mode=SHADOW
```

5 Evidence: Test Suite

5.1 Isolation Tests

File: `tests/integration/test_external_pilot_signal_isolation.py`

- 11 test classes, 38+ test methods
- Covers: valid, malformed, empty, adversarial, oversized logs
- Guarantees: namespace isolation, verifier invariance, input purity
- Meta-tests: registry tripwire (minimum 35 tests threshold)

5.2 Text Neutrality Tests

File: `tests/policy/test_pilot_text_neutrality.py`

- 3 test classes, 8 test methods
- Guarantees: no banned alarm words, no evaluative terms
- Structured tokens (PASS/WARN) explicitly allowed

5.3 Summary

Test Suite	**Classes**	**Methods**
Signal Isolation	11	38+
Text Neutrality	3	8
Total	**14**	**46+**

6 Language Posture

All pilot components use neutral language:

- Warning strings avoid alarm words
- Structured result codes (SUCCESS, SCHEMA_INVALID, etc.)
- Disclaimer included in all outputs

Pilot provenance disclaimer (required in all artifacts):

> *"PILOT PROVENANCE: Binds artifact to toolchain state at ingestion. NOT experiment provenance. Does not imply experimental validity, reproducibility, or parity with CAL-EXP runs."*

7 Contact / Next Step

Safe Pilot Handshake: External partners may provide JSON/JSONL logs for ingestion via the pilot adapter. All ingested data is isolated to `governance.external_pilot.*` and cannot affect internal metrics, verifier behavior, or CAL-EXP verification surfaces. To initiate a pilot handshake, contact the MathLedger governance council with sample log structure and expected schema.

Document: PILOT_EVIDENCE_PACKET v1.0
Mode: SHADOW (observational only)
Authority: PILOT_NON_INTERFERENCE_ATTESTATION.md
Generated: 2025-12-17