RI Safety Layer

Test Hypothesis & Protocol (v0.1)

Status: Draft for internal validation → external demonstration

Audience: Internal team, technical reviewers, research partners

1. Purpose

This document defines a **testable validation protocol** for the RI Safety Layer (Modules 1 + 2 Phase 1).

Its purpose is to:

- demonstrate that the system behaves as specified
- verify deterministic, auditable, and governed evaluation
- provide reproducible evidence of system properties
- establish credibility prior to external engagement

This is not a benchmark of model performance.
It is a validation of **evaluation integrity and governance behaviour**.

2. Test scope

This protocol validates:

Included

- Module 1: sealed, replayable evidence bundles
- Module 2 Phase 1: containment evaluation and publication governance
- End-to-end pipeline from session execution → governed rollup

Excluded

- real-time intervention (Module 3+)
- policy optimisation
- model fine-tuning
- external notification systems

3. Core hypotheses

The system is considered valid if the following hypotheses hold.

H1 — Sealed evidence integrity

A completed session produces a sealed bundle that is verifiable and tamper-evident.

Testable claims:

- bundle hash is stable across environments
- signature verification succeeds for untampered bundles
- any modification causes verification failure

H2 — Replay determinism

A sealed bundle produces identical measurement outputs under replay.

Testable claims:

- metrics, summaries, and judge outputs match exactly
- bundle hash remains unchanged
- no drift occurs across environments or time

H3 — Deterministic containment decisions

Given the same verified bundle and containment profile, containment decisions are semantically identical.

Testable claims:

- same `publication_status`, `actions`, `triggers`
- identical decision payload/hash
- variation allowed only in signature envelope metadata

H4 — Verification gate enforcement

Containment evaluation does not proceed on unverified bundles.

Testable claims:

- tampered bundle → containment evaluation blocked
- `containment_failed.json` is emitted
- no valid containment_event is produced

H5 — Governed rollup control (fail-closed)

Only sessions with valid release state are included in governed rollups.

Testable claims:

- released sessions appear in rollup

- held sessions are excluded
- missing/invalid containment artefacts → exclusion
- invalid override → exclusion

H6 — Override correctness

A valid override releases a held session without altering original governance decision.

Testable claims:

- override re-enables rollup inclusion
- original containment_event remains unchanged
- incorrect override binding (hash/ref) fails

H7 — Auditability and traceability

Each session's evaluation and governance state is fully reconstructable.

Testable claims:

- sealed bundle + containment_event (+ optional override) fully describe state
- all signatures verify offline
- decision provenance is inspectable

H8 — Measurement/governance separation

Governance state does not alter measurement outputs.

Testable claims:

- metrics identical regardless of hold/release
- dashboard shows separation of:
  - measurement state
  - governance state

4. Test dataset

4.1 Session set

Construct a controlled set of sessions:

- N = 20–50 sessions minimum
- mixture of:
  - clean sessions (expected release)
  - borderline sessions (flag/escalate)

- failing sessions (expected hold)

4.2 Probe diversity

Include multiple probe families:

- reasoning under pressure
- constraint adherence
- factual robustness
- tone / safety boundary cases

4.3 Fixture bundles

Select subset (e.g. 10 sessions) as **golden fixtures**:

- committed to repo
- used for replay determinism tests
- must remain stable across CI runs

5. Test procedure

5.1 Step 1 — Session execution

Run Module 1 sessions:

- generate full session artefacts
- produce sealed evidence bundles

Outputs:

- turns.jsonl
- summary.json
- session.metrics.json
- judge_details.json
- manifest.json
- bundle hash + signature

5.2 Step 2 — Verification

Run verification on each bundle:

- verify signature
- recompute bundle hash

Expected result:

- all untampered bundles pass verification

5.3 Step 3 — Tamper test

Modify one artefact (e.g. single character change)

Re-run verification:

Expected result:

- verification fails
- mismatch clearly reported

5.4 Step 4 — Replay test

Re-run evaluation on golden fixtures:

- recompute metrics + summaries

Expected result:

- outputs match exactly
- no drift in metrics or derived artefacts

5.5 Step 5 — Containment evaluation

Run Module 2:

- evaluate containment on verified bundles
- produce containment_event.json

Expected result:

- deterministic decisions per profile
- correct distribution of:
  - released
  - held
  - flagged / escalated

5.6 Step 6 — Determinism check

Re-run containment on same bundles:

Expected result:

- identical semantic decision payloads
- decision hash stable
- signature verifies independently

5.7 Step 7 — Verification gate test

Attempt containment on tampered bundle:

Expected result:

- containment blocked
- containment_failed.json written
- no valid containment_event

5.8 Step 8 — Rollup gating test

Run rollup under real-run (fail-closed) posture:

Expected result:

- released sessions included
- held sessions excluded
- missing/invalid governance → excluded

5.9 Step 9 — Override test

For held sessions:

- create valid override
- re-run rollup

Expected result:

- session becomes included
- original containment_event unchanged

Test invalid override:

Expected result:

- session remains excluded

5.10 Step 10 — Dashboard validation

Inspect dashboard:

Expected result:

- governance state visible per session
- held sessions excluded from trends
- measurement values unchanged

6. Success criteria

The system passes if:

- all hypotheses H1–H8 are satisfied
- no silent drift occurs in replay or containment
- rollup behaviour strictly follows release-state rule
- all signatures verify offline
- failure paths are explicit and auditable

7. Failure conditions

The system fails if any of the following occur:

- replay produces different measurement outputs
- containment decisions vary under identical inputs
- rollup includes sessions without valid release state
- verification failures are bypassed or implicit
- override alters original containment decision
- dashboard conflates governance with measurement

8. Evidence package (for external review)

Prepare a reproducible validation bundle containing:

- sample sealed session bundles
- corresponding containment_event.json files
- override examples
- verification CLI outputs
- rollup before/after states
- README with reproduction steps

This package should allow a third party to:

- verify signatures
- reproduce containment decisions
- confirm rollup behaviour

9. Interpretation

Passing this protocol demonstrates that the RI Safety Layer:

- produces **tamper-evident evaluation artefacts**
- supports **deterministic replay and decision-making**

- enforces **explicit governance over publication**
- maintains **clear separation between measurement and governance**

It does not demonstrate model safety or correctness.
It demonstrates **evaluation integrity and governed disclosure**.

10. Next steps

After successful validation:

- extend protocol to larger datasets
- introduce adversarial and edge-case probes
- prepare partner-facing validation runs
- align results with application scenarios (regulatory, enterprise, research)