

CIL decision-contract evidence

IMPLEMENTED AND TESTED OFFLINE • NOT A LIVE TRADING SYSTEM

Problem and contribution

A model assessment, a human-readable card and a downstream presentation payload must not disagree about identity, evidence, risk or authority. William directed the requirements and review boundaries for an AI-assisted implementation that separates those responsibilities.¹

What was delivered

PR #15 adds a deterministic policy evaluator and a separate paper decision/card/PEARL handoff contract. The flow validates a supplied assessment, checks declared risk and portfolio inputs, renders the same decision revision, and builds a consistency-checked payload.¹

Stage	Delivered behavior
CIL assessment	Evidence and policy binding; required reviewer completion; gate verdicts.
Decision record	Ten required answer states; risk checks; expiry and identity.
Card / PEARL payload	Same record identity; receipt-binding checks; orders disabled.

Deliberate limit

Inputs, approvals and persistence receipts are declarations from future trusted adapters. The patch does not authenticate them, derive market indicators, prove predictive edge, wire the legacy runtime, persist a loss lock or submit an order. Hash consistency is not a digital signature or proof of a database write.¹

Source receipts

1. Merged implementation and scope (repository access may be required)
<https://github.com/Barefootservants2/a2e-platform/pull/15>

Re-executed verification receipt

The following receipt was generated while preparing this public artifact. The observations are from a local, bounded test run, not production telemetry.

Receipt field	Observed value
Observed UTC	2026-09-16T22:22:34+00:00
Repository commit	1a0088cb822a470a1700ca695ba39b32cf08980e
Instrument	pytest; two explicitly selected test modules
Result	89 passed in 0.09s
Network / estate credentials	No credentials supplied; new decision-contract test module blocks socket creation.

Selected test command

PYTEST_DISABLE_PLUGIN_AUTOLOAD=1 python -m pytest -q tests/test_cil_vnext_contract.py tests/test_decision_contracts.py

What the count means

89 passing test cases, including parameterized cases, across test_cil_vnext_contract.py and test_decision_contracts.py. This excludes the full legacy suite, production workflows, account state, market feeds and broker operations. It is not a count of trades or live integrations.

Raw test summary

..... [80%]
..... [100%]
89 passed in 0.09s

Source receipts

1. Source test suite in the merged change
<https://github.com/Barefootservants2/a2e-platform/pull/15/files>

Success and refusal examples

These outputs were generated directly from the merged module’s synthetic TEST fixtures, not hand-written verdicts. The fixed fixture clock is 2026-09-16T14:00:00Z; it is not the time of a market event.

Input variation	Actual returned result
Complete synthetic LONG	ELIGIBLE_FOR_PAPER_REVIEW
Complete synthetic SHORT	ELIGIBLE_FOR_PAPER_REVIEW
Ten answers omitted	DEFERRED
Required reviewer missing	HOLD_FOR_EVIDENCE
Observations missing	HOLD_FOR_EVIDENCE
Altered decision identity	False
Order enabled	False
Runtime activation allowed	False

How to read the positive result

ELIGIBLE_FOR_PAPER_REVIEW means the declared synthetic record passed these offline checks. It does not authorize a paper fill, a live order or runtime activation. Both long and short fixtures are synthetic engineering tests.

How to read the negative result

Omitting all ten answers returns DEFERRED; missing observations or required-member completion returns HOLD_FOR_EVIDENCE. Altering the sealed record causes the identity check to return False. These refusal paths are as important as the successful path.

Source receipts

1. Contract source, pinned to the tested commit
https://github.com/Barefootservants2/a2e-platform/blob/1a0088cb822a470a1700ca695ba39b32cf08980e/paper/decision_contract.py

Original contract excerpts

The excerpts below are reproduced from the merged Python source. Line wrapping, explanatory labels and pagination were added for this public evidence pack.^{1,2}

Ten-question boundary

```
QUESTIONS = (  
    "hypothesis", "catalyst", "support", "contradictions",  
    "entry_conditions", "invalidation", "risk", "portfolio",  
    "historical_sample", "no_trade",  
)
```

Explicit runtime boundary

```
RUNTIME_ACTIVATION_ALLOWED = False
```

Positive chain assertions (test excerpt)

```
assert record["status"] == "ELIGIBLE_FOR_PAPER_REVIEW"  
assert len(card["questions"]) == 10  
assert payload["cards"][0] == card  
assert card["decision_hash"] == record["decision_hash"]  
assert payload["order_enabled"] is False  
assert payload["runtime_activation_allowed"] is False
```

Why this artifact is useful

A reviewer can see the concrete schema, executable assertions, observed results and limits without needing access to private repositories. Full implementation provenance remains pinned to the commit on the first receipt page.

Source receipts

1. Decision-contract source

https://github.com/Barefootservants2/a2e-platform/blob/1a0088cb822a470a1700ca695ba39b32cf08980e/paper/decision_contract.py

2. Original test source

https://github.com/Barefootservants2/a2e-platform/blob/1a0088cb822a470a1700ca695ba39b32cf08980e/tests/test_decision_contracts.py