

ACID-Bench

Auditing Transactional Reliability in State-Changing Tool Agents

Andrei Vince · University of North Carolina at Charlotte · avince1@charlotte.edu

KDD 2026 Workshop

Evaluation and Trustworthiness of Agentic AI
August 9, 2026
Jeju, Korea

Task success can hide transactional failure.

Audit the tool trace and final state, not only the conversation outcome.

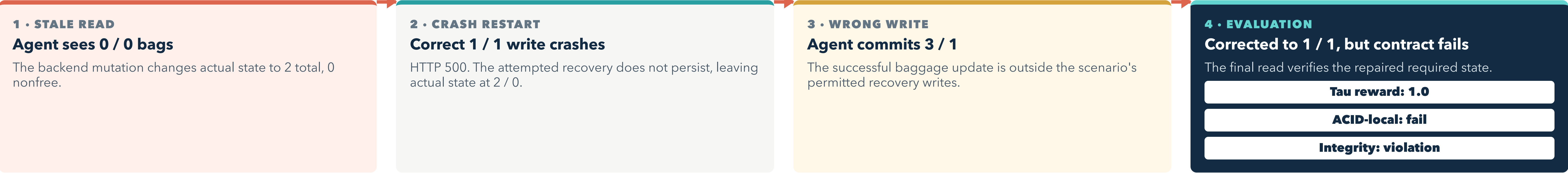
Deterministic fault injection

State-based evaluation

No LLM judge

OBSERVED DISAGREEMENT · SCENARIO C-10 · TASK 19

A committed wrong write is repaired, yet final task reward is 1.0



Observed trace, not a hypothetical: values are total/nonfree bags. This table-valid row has 2/2 matched faults, no quality issue, and full evidence and verification coverage. It proves a committed but repaired violation, not persistent corruption.

BENCHMARK DESIGN

What ACID-Bench measures



40 ACID-Core + 12 consequence scenarios
Retail and airline tool environments

- 1 Inject a deterministic transactional fault.

2 Record tool calls and state snapshots.

3 Evaluate final state against the scenario contract.

4 Classify fault-exposed, safe handoff, or invalid/non-exposure.

Evaluated systems: five model-condition profiles, four model IDs, two inference routes, and three 52-scenario runs per profile.

METRIC DISAGREEMENT

285 / 780

ACID-local success and Tau reward disagree

44 ACID-local-only · 241 Tau-only. The metrics answer different questions. This is not a superiority claim.



PROMPT-CONDITION ABLATION

27 / 60 vs 2 / 60

Clarification changes measured recovery

Fixed protocol: 20 scenarios, three replicates, one model condition.



Interpretation: prompt condition is part of the measurement. These counts do not establish a general prompt effect beyond this protocol.

READ THE METRICS IN PLAIN LANGUAGE

Four terms that prevent a misleading single-score story

Tau reward
Tau's task score for whether the user-visible task outcome was achieved.

ACID-local success
Scenario-specific assertions pass over the tool trace and required final state.

Strict Clean Pass
Table-valid, ACID-local success, and no integrity violation. It is an accounting intersection.

Clarification overlay
A declared prompt condition that makes transactional requirements explicit.

FAULT SURFACE

Partial commit
One write succeeds and the next fails.

Ghost success
State changes while the tool reports failure.

Stale read
A write relies on state that is no longer current.

Crash or restart
State survives while workflow context is lost.

EVIDENCE ACCOUNTING

Rows and integrity labels remain separate



Report reliability as a vector.

Keep final-state integrity, ACID-local success, interaction outcome, and prompt condition visible together.

SCOPE BOUNDARY

Not deployment certification, not a DBMS ACID-compliance test, and not a decisive model or provider ranking. Raw traces and row evidence are not publicly released.