LEAP Paper 5 C28: Supporting Source-Execution Evidence

Meshaal Obaid Mouawad

21 November 2026

Evidence scope. This supplement preserves the C26 supporting experiments and their original numeric contrasts. They are distinct from the 32-case coding-agent documentary-reference cohort, and are excluded from its denominator.

1 Source-identity and execution-boundary contrasts

The archived source workflow in Figure 1 keeps source-derived results separate from documentary expectations. The Four Keys change-failure-rate witness exposes a difference between counting affected deployments and counting joined rows. Of four deployments, one is affected, giving a deployment reference a/n = 1/4 = 0.25. The executed join creates two contributing rows, so its inspected numerator yields j/n = 2/4 = 0.50. The two values reflect different counted entities on the same constructed input.

The matched intervention tests which count explains that output. Only the second comment's issue number changes: configuration A has i = 1 distinct incident and j = 2 joined contributions; configuration B has i = 2 incidents and the same j = 2 contributions. Both retain a = 1 affected deployment among n = 4 deployments. A numerator based on incident identity would change from 0.25 to 0.50; a joined-row numerator would remain at 0.50. The observed output remains 0.50 in both, matching the joined-row prediction (Figure 2). A is an existing multilink baseline, and the pair is treated as one controlled comparison.

Under this fixture mapping, if m distinct causal change links all concern the same affected deployment among n deployments, joined-row and affected-deployment fractions differ by (m - 1)/n. This elementary relation describes the recorded mechanism; it is not presented as a new general theorem. The earlier project issue already identified counting concerns, and its merged change corrected the denominator to distinct deployments (dora-team/fourkeys 2022a,b). The retained witness reproduces the residual numerator behavior in the inspected revision and attributes priority for the historical concern to that project record.

The restoration-time contrast isolates a different source boundary. Supplying an incident opening at 13:00 and closure at 16:00 produces three hours. Including the incident-producing query instead supplies the linked deployment start at 12:00, so the same downstream calculation produces four hours (Figure 3). The upstream producer changes the timestamp used for restoration without changing the failure-rate join multiplicity.

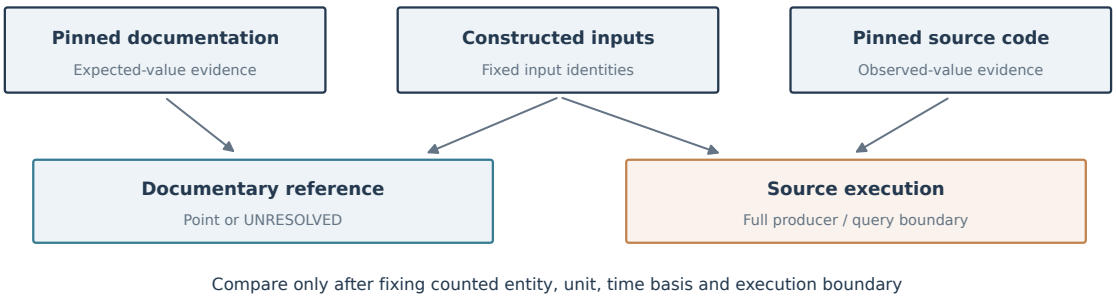


Figure 1: Source-case workflow. The documentary expectation and source-derived output have distinct evidential origins; the execution comparison identifies the producing query, counted unit and transformed source path.

Consequently, the recorded restoration discrepancy disappears under that source path while the failure-rate difference persists.

The thirteen original observations comprise three failure-rate differences arising from the archived mappings, one conditional lead-time difference, one isolated restoration difference and eight agreements. The lead-time comparison excludes a zero-duration record; its reference changes with the decision to treat that record as an eligible nonautomated change. The five recorded differences therefore identify measured outcomes under their stated conditions, rather than five separately established software defects.

Four archived PVWatts controls demonstrate the importance of matching output basis and input units. Annual generation of 17,520 kWh yields a 20% capacity factor against a 10,000 W DC nameplate and 25% against an 8,000 W AC nameplate; zero generation gives 0% under either basis. The extracted assignments agree with the respective documentary bases after converting nameplate watts to kilowatts (System Advisor Model and SAM Simulation Core 2024). These are agreement controls for those calculations.

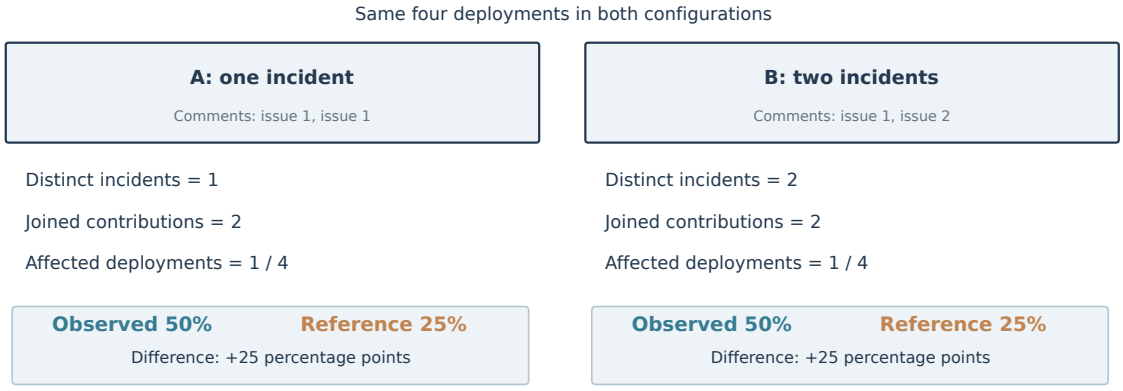


Figure 2: Matched Four Keys intervention. Changing only the second incident identity increases distinct incidents from one to two, while joined contributions remain two and the affected-deployment count remains one of four. The observed joined-row output stays at 50 percent against the 25 percent affected-deployment reference.

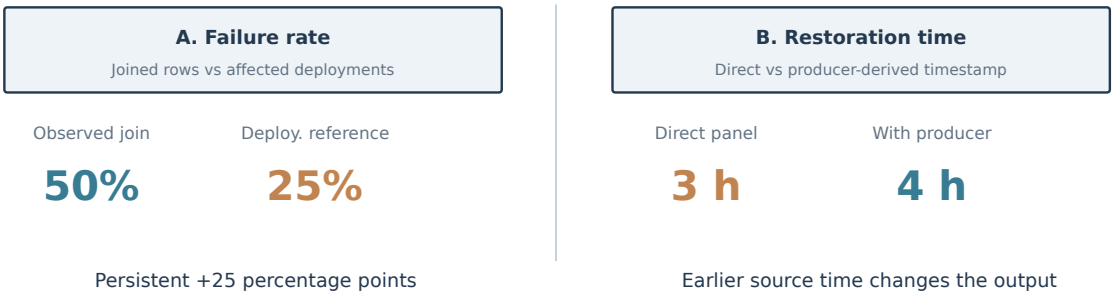


Figure 3: A joined-row failure-rate contrast remains at 50 versus 25 percent, while restoration time shifts from three to four hours when the upstream timestamp producer is included. These are archived local source-replay observations.

2 Native execution and reduced replay equivalence

The pinned dbt Jaffle Shop model (dbt Jaffle Shop DuckDB contributors 2026) supplied 199 stored output-identity comparisons: 99 order amounts and 100 customer counts. The complete model and reduced scalar expressions agreed on 161 identities and differed on 38. All 38 differences occurred for customers without orders: the complete left-join model returned NULL for number_of_orders, whereas the reduced COUNT expression returned zero. This repeated nullability mechanism shows that reducing an execution path can change the observed scalar even when the underlying customer is unchanged.

2.1 DevLake PR Review Depth: documentation SQL and shipped Grafana panel 99

Apache DevLake defines pull-request (PR) review depth as the number of review comments per pull request. The pinned website documentation example uses COUNT(DISTINCT prc.id)/COUNT(pr.id); shipped Grafana panel 99 counts distinct PR identities in its denominator, using COUNT(DISTINCT prc.id)/COUNT(DISTINCT pr.id) (Apache DevLake documentation contributors 2026; Apache DevLake application contributors 2026). The source comparisons align the project, time-bucket and subproject selectors to isolate this denominator difference.

The archived DevLake input has two merged PRs in the same selected project, time bucket and subproject, with two comments attached to the first PR and none to the second. The left join produces three rows for two distinct PRs. Dividing the two distinct comments by joined rows gives $2/3 = 0.6667$ in the documentation example; dividing by distinct PR identities gives $2/2 = 1.0000$ in panel 99's SQL. Table 1 displays the denominators responsible for the contrast.

Table 1: Apache DevLake PR Review Depth on the constructed input under aligned project/time filters. The distinction is between published documentation SQL and shipped panel 99 SQL.

SQL source	Distinct comments	Pull-request denominator	Ratio
Website documentation SQL	2	3 joined PR rows	$2/3 = 0.6667$
Shipped Grafana panel 99 SQL	2	2 distinct PRs	$2/2 = 1.0000$

The DevLake finding is a documentation-query discrepancy. The website example retains join multiplicity in its denominator, whereas the shipped panel 99 query counts distinct PR identities. The checked record supports that source-level distinction and its constructed-input calculation.

The discrepancy label therefore names the documentation SQL and the panel query separately, without assigning a defect to the shipped dashboard.

A separate archived median case distinguishes two reporting conventions on the same four durations: 60, 180, 240 and 360 minutes. The recorded lower-middle primitive returns 180 minutes; the midpoint convention gives 210 minutes. The supplied documentary example concerns an odd number of observations, so the even-sample choice remains unstated in that evidence.

The retained Rill and Cube cases provide further examples of execution-boundary sensitivity. The Rill expression returns infinity for a positive numerator divided by zero when evaluated in DuckDB 1.5.5 (Rill examples contributors 2026); a reduced SQLite expression introducing NULLIF instead returns NULL. The Cube record checks a five-row producer and its count/sum expressions in DuckDB (Cube contributors 2026). Each outcome is tied to the expression and runtime actually examined, illustrating why a reduced replay must identify the transformation applied to the source path.

3 Associated research evidence

Primary replication data (DOI1): https://doi.org/10.7910/DVN/W2RJ51.
Supplementary documentary-condition register (DOI2): https://doi.org/10.7910/DVN/4Q8BFP.