LATINFLASH RESEARCH PAPER 2026-01

The Oracle Blind Spot: Re-adjudicating Automated Telemetry Assertions with Transport-Level Evidence

Independent preprint / technical report · Version 1.0 · 29 August 2026

Publication status. This version has not been peer reviewed. A later conference or journal version may differ.

Abstract

Modern web systems rely on third-party telemetry SDKs for critical observability, yet verifying the correctness of this instrumentation often depends on automated end-to-end assertion oracles. These oracles are themselves subject to the Oracle Problem, where imperfect observation logic can produce ambiguous failure signals. We present a two-phase empirical study - retrospective analysis and prospective validation - that re-adjudicates automated telemetry assertions against raw network captures (CDP logs). We analyzed 82 assertion outcomes mapping to 21 distinct telemetry-event units. Of 49 recorded failures in Phase I, 37 (75.5%) were adjudicated as oracle artifacts at the assertion level (91.7% at the event level). Re-adjudication used raw network transport as transport-level reference evidence. A post-hoc counterfactual reconstruction reached an 18/18 transport-pass result after correcting observation defects. In prospective validation, a corrected oracle evaluated 32 assertions and produced no oracle artifacts on the new trace, while retaining one product-delivery-defect candidate. The results highlight how defects in a measurement system can distort the apparent reliability of telemetry transport.

Measure	Observed result
Phase I assertions	82 assertions / 21 event units
Recorded failures	49
Oracle artifacts	37 / 49 (75.5%)
Event-level artifact result	11 / 12 (91.7%)
Conservative sensitivity	34 / 55 (61.8%)
Prospective validation	0 oracle artifacts / 32 assertions

1. Introduction

Modern web systems increasingly rely on third-party telemetry Software Development Kits (SDKs) for observability, analytics, experimentation and attribution. Engineering teams often employ automated end-to-end

assertions to verify telemetry delivery. Those assertions form a measurement system that is itself subject to defects.

A failing telemetry assertion is therefore ambiguous: it may indicate an actual event-delivery failure or an artifact of the verification process. This is a domain-specific instance of the Oracle Problem in software testing - the difficulty of determining correct behavior when the observation mechanism may itself be imperfect.

This study investigates whether raw transport evidence can be used to re-adjudicate ambiguous telemetry assertion failures and quantify the share of the recorded failure signal attributable to the verification mechanism.

2. Methods

2.1 Two-phase study design

Phase I retrospectively analyzed 82 historical assertion outcomes mapping to 21 distinct telemetry-event units. A retrospective analysis plan was frozen before systematic quantitative re-adjudication. Phase II prospectively evaluated a corrected oracle with 32 assertions on a new execution trace.

2.2 Transport-level reference evidence

An event was considered transport verified when the expected HTTP request appeared in raw Chrome DevTools Protocol captures with expected metadata and an HTTP 200/204 status. This confirms client-side transport and server acceptance at the observed endpoint. It does not establish downstream ingestion, persistence or analytics-dashboard visibility.

2.3 Oracle correction

The corrected oracle incorporated recursive batch parsing, GA4 POST-body decoding, network-idle settlement and stricter execution-window binding. The prospective validation therefore tested a corrected observation mechanism rather than changing the underlying definition of expected telemetry events.

2.4 Oracle-Artifact Share

The primary retrospective operationalization was the Oracle-Artifact Share (OAS): the number of recorded assertion failures adjudicated as oracle artifacts divided by the total number of recorded assertion failures. OAS is a conditional measurement-system share within the observed failure pool, not an estimate of telemetry defect prevalence in the population.

3. Results

3.1 Phase I recorded outcomes

The Phase I dataset comprised 82 assertions mapping to 21 unique telemetry events and produced 49 recorded assertion failures. Thirty-seven of those failures were adjudicated as oracle artifacts, giving an assertion-level OAS of 0.7551.

Dependence mattered. The dominant timing-race cluster generated 23 of the 37 assertion-level artifacts. At event level, 11 of the 12 events that had recorded at least one assertion failure were associated with oracle artifacts (91.7%). A conservative sensitivity analysis expanded the denominator to 55 and still yielded an OAS of 34/55 = 0.6182.

3.2 Counterfactual reconstruction

An offline, post-hoc reconstruction of the original Run A evidence reached 18/18 transport-pass after correcting observation defects. Because this analysis was post-hoc, it is treated as counterfactual supporting evidence rather

than confirmatory prospective evidence.

3.3 Prospective validation

The corrected oracle was evaluated prospectively on a fresh trace with 32 assertions. It produced no oracle artifacts (OAS = 0.0). One failure remained classified as a product-delivery-defect candidate, showing that removal of measurement artifacts did not force all failures into passes.

4. Related Work

The study sits at the intersection of test-oracle quality, flaky-test research, runtime trace verification and telemetry instrumentation. Prior work establishes the Oracle Problem and the importance of assessing imperfect oracles. Research on flaky tests shows that asynchronous timing and shared failure causes can generate non-independent failure patterns. PR-02 applies those concepts to browser telemetry verification and treats raw network captures as a secondary adjudication source rather than as complete end-to-end ground truth.

5. Discussion

The central result is a measurement warning: a large share of the recorded Phase I failure signal was generated by the verification mechanism rather than by observed transport failure. The result is especially relevant because the assertions were structurally dependent: a single timing or decoder defect could amplify into multiple reported failures.

The prospective phase strengthens the interpretation by testing a corrected oracle on a new trace. However, the study remains a single-system case study. The data support conclusions about the observed telemetry verification process; they do not establish a universal oracle-artifact rate for other products, SDKs or providers.

Limitations

- Single-system empirical case study.
- Assertion outcomes are structurally dependent and should not be interpreted as fully independent samples.
- Transport-level reference evidence establishes observed transport, not downstream ingestion.
- The 18/18 offline counterfactual is post-hoc and is not treated as confirmatory evidence.
- Researchers were involved in the system under study; protocol freezing, artifact traceability and adversarial review mitigate but do not eliminate this risk.

6. Conclusion

PR-02 found that defects in a telemetry verification oracle can dominate the apparent failure signal. In Phase I, 37 of 49 recorded assertion failures were re-adjudicated as oracle artifacts. Prospective evaluation of the corrected oracle produced no oracle artifacts across 32 assertions on a fresh trace. The results motivate systematic validation of telemetry test harnesses against lower-level transport evidence, while preserving a strict distinction between transport verification and downstream analytics ingestion.

References

Barr, E. T., Harman, M., McMinn, P., Shahbaz, M., & Yoo, S. (2015). The oracle problem in software testing: A survey. IEEE Transactions on Software Engineering, 41(5), 507-525. https://doi.org/10.1109/TSE.2014.2372785

Luo, Q., Hariri, F., Eloussi, L., & Marinov, D. (2014). An empirical analysis of flaky tests. FSE, 643-658. https://doi.org/10.1145/2635868.2635920

Runeson, P., & Host, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14, 131-164. https://doi.org/10.1007/s10664-008-9102-8

Jahangirova, G., Clark, D., Harman, M., & Tonella, P. (2016). Test oracle assessment and improvement. ISSTA, 247-258. https://doi.org/10.1145/2931037.2931062

Staats, M., Gay, G., & Heimdahl, M. P. E. (2011). Programs, oracles, and observations. ICSE, 151-160. https://doi.org/10.1145/1985793.1985814

Alshammari, A. et al. (2024). Non-Flaky and Nearly Optimal Time-Based Treatment of Asynchronous Wait Flaky Tests. ACM TOSEM. https://doi.org/10.1145/3695989

Suggested institutional citation: LatinFlash Research (2026). *The Oracle Blind Spot: Re-adjudicating Automated Telemetry Assertions with Transport-Level Evidence.* LatinFlash Research Paper 2026-01, Preprint v1.0.