

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

Manuel Enrique Morales Santiago

LatinFlash Research | ORCID iD: [0009-0007-6921-7688](https://orcid.org/0009-0007-6921-7688) | Dialnet Code: [3123473](https://dialnet.org/3123473)

Document Date: August 29, 2026 (Audited Sep 2026) | Status: Independent Research Publication (v1.1 Bibliographically Reconciled Edition)

ABSTRACT

Automated assertion suites used to verify web telemetry delivery are themselves subject to verification defects. Ambiguous telemetry assertion failures were re-adjudicated using raw network transport evidence. In Phase I, 37 of 49 recorded assertion failures (75.5%) were adjudicated as oracle artifacts at the assertion level (91.7% at the event level). Re-adjudication utilized raw network transport as reference evidence. A post-hoc counterfactual reconstruction reached an 18/18 transport-pass result after correcting observation defects. Prospective validation of a corrected oracle produced zero oracle artifacts across 32 assertions (OAS = 0.0) on a fresh trace, demonstrating how measurement system artifacts can distort telemetry transport reliability, consistent with broader systemic flakiness literature (Parry et al., 2025).

1. Introduction

1.1 Problem Context

Modern web systems increasingly rely on third-party telemetry Software Development Kits (SDKs) to power critical business functions. The correctness of the data emitted by these SDKs is paramount. Engineering teams often employ automated end-to-end (E2E) assertions within testing pipelines to verify telemetry delivery. However, these assertions form a measurement system that is itself subject to defects.

1.2 Motivation

A failing telemetry assertion is inherently ambiguous: it may indicate a product defect or a verification artifact (e.g., timing races, decoder mismatches). This study addresses the "Oracle Problem" ([Barr et al., 2015](#)) in the telemetry domain by developing a methodology to re-adjudicate failures against transport-level evidence.

1.3 Contributions

1. **OAS Operationalization:** Quantifying measurement bias as the share of failures attributable to oracle artifacts.
2. **Dual-Source Adjudication:** Auditing E2E verdicts against Chrome DevTools Protocol (CDP) network captures (transport-level reference evidence).
3. **Failure Classification:** A telemetry-specific refinement informed by hierarchical flakiness classification schemes ([Li et al., 2026](#)).
4. **Prospective Validation:** Demonstrating the effectiveness of oracle repairs on unseen traces.
5. **Structural Analysis:** Alignment with systemic flakiness patterns ([Parry et al., 2025](#)) in telemetry pipelines.

2. Methods

2.1 Study Design

This study employs a two-phase design: Phase I retrospectively analyzed 82 historical assertion outcomes mapping to 21 telemetry events. A retrospective analysis plan was frozen before systematic re-adjudication following empirical software engineering case study guidelines ([Runeson & Höst, 2009](#)). Phase II prospectively evaluated a corrected oracle (32 assertions), governed by a protocol frozen before execution.

2.2 Transport-Level Reference Evidence

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

2.3 Oracle Corrections (Phase II)

The corrected oracle implemented: (1) recursive batch parsing, (2) GA4 POST-body decoding, (3) network-idle settlement, and (4) strict window binding, informed by test oracle improvement techniques ([Jahangirova et al., 2016](#)) and asynchronous wait settlement principles ([Pei et al., 2025](#)).

3. Results

3.1 Phase I: Recorded Outcomes

The Phase I dataset comprised 82 assertions mapping to 21 unique telemetry events, resulting in 49 recorded assertion failures.

Measurement Variable	Observed Retrospective Outcome
Phase I Assertions	82 assertions / 21 event units
Recorded Assertion Failures	49 failures
Adjudicated Oracle Artifacts	37 / 49 (75.5%)
Event-Level Artifact Result	11 / 12 events (91.7%)
Conservative Sensitivity Analysis	34 / 55 (61.8%)
Prospective Validation (Phase II)	0 oracle artifacts / 32 assertions (0.0%)

3.2 Phase I: Oracle-Artifact Share (OAS)

We define the primary retrospective metric as the **Oracle-Artifact Share (OAS)**, representing the conditional proportion of recorded assertion failures attributable to the verification mechanism:

$$\text{OAS} = \text{Oracle Artifacts} / \text{Total Recorded Failures}$$

Of the 49 recorded assertion failures in Phase I, 37 (75.5%) were adjudicated as oracle artifacts. Dependence mattered significantly: the dominant timing-race cluster generated 23 of the 37 assertion-level artifacts. At event level, 11 of 12 events with recorded assertion failures were associated with oracle artifacts (91.7%). A conservative sensitivity analysis expanding the denominator to 55 yielded an OAS of $34/55 = 0.6182$.

3.3 Phase II: Prospective Validation

An offline post-hoc reconstruction of original Run A evidence reached 18/18 transport-pass after correcting observation defects. The prospective validation run (32 assertions mapping to 7 events) produced no oracle artifacts (OAS = 0.0). One failure remained classified as a product-delivery-defect candidate.

4. Related Work

The study sits at the intersection of test-oracle quality ([Barr et al., 2015](#)), flaky-test research ([Luo et al., 2014](#)), runtime trace verification, and telemetry instrumentation. Prior work establishes the Oracle Problem and the importance of assessing imperfect oracles ([Jahangirova et al., 2016](#)) and implicit observation assumptions ([Staats et al., 2011](#)). Research on flaky tests shows that asynchronous timing and shared failure causes can generate non-independent failure patterns ([Luo et al., 2014](#)). Recent literature on systemic flakiness ([Parry et al., 2025](#)), hierarchy-aware classification ([Li et al., 2026](#)), and asynchronous wait web tests ([Pei et al., 2025](#)) provides valuable theoretical and taxonomy context. We note that our empirical study specifically evaluated telemetry transport assertion oracles (OAS), and we frame these recent publications as related literature and conceptual background rather than claiming our original experiments evaluated or confirmed their specific classification taxonomies.

5. Discussion & Limitations

5.1 Discussion

Most of the observed failure signal was measurement-system generated. Our findings on structural dependence align with Systemic Flakiness literature ([Parry et al., 2025](#)). We conclude that telemetry test harnesses require validation against transport-level evidence.

5.2 Limitations

- **Single-system empirical case study:** Conclusions support the observed telemetry verification process but do not establish a universal rate.
- **Structural dependence:** Assertion outcomes are structurally dependent and should not be interpreted as fully independent samples.
- **Transport-level reference evidence:** Confirms observed client transport, not downstream analytics ingestion or persistence.
- **Post-hoc counterfactual:** The 18/18 offline counterfactual reconstruction is post-hoc and non-confirmatory.
- **Researcher involvement in system studied:** Researchers were directly involved in the system under study; protocol freezing, artifact traceability, and adversarial review mitigate but do not eliminate this risk.

6. Conclusion

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 produced zero oracle artifacts across 32 assertions on a fresh trace, motivating systematic validation of telemetry test harnesses.

References

1. 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>
2. Luo, Q., Hariri, F., Eloussi, L., & Marinov, D. (2014). An empirical analysis of flaky tests. *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE)*, 643-658. <https://doi.org/10.1145/2635868.2635920>
3. Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), 131-164. <https://doi.org/10.1007/s10664-008-9102-8>
4. Jahangirova, G., Clark, D., Harman, M., & Tonella, P. (2016). Test oracle assessment and improvement. *Proceedings of the 25th International Symposium on Software Testing and Analysis (ISSTA)*, 247-258. <https://doi.org/10.1145/2931037.2931062>
5. Staats, M., Whalen, M. W., & Heimdahl, M. P. E. (2011). Programs, tests, and oracles: the foundations of testing revisited. *Proceedings of the 33rd International Conference on Software Engineering (ICSE)*, 391-400. <https://doi.org/10.1145/1985793.1985847>
6. Pei, Y., Sohn, J., Habchi, S., & Papadakis, M. (2025). Non-Flaky and Nearly Optimal Time-Based Treatment of Asynchronous Wait Web Tests. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 34(2), Article 45, 1-29. <https://doi.org/10.1145/3695989>
7. Parry, O., Kapfhammer, G. M., Hilton, M., & McMinn, P. (2025). Systemic Flakiness: An Empirical Analysis of Co-Occurring Flaky Test Failures. *Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE 2025)*, 1-10. <https://doi.org/10.1145/3756681.3756945>
8. Li, Z., Wu, Z., Lei, Y., Xie, H., Li, M., & Hu, J. (2026). HiFlaky: Hierarchy-aware flakiness classification. *Journal of Systems and Software*, 234, Article 112741. <https://doi.org/10.1016/j.jss.2025.112741>