Precise Records, Unstable Meanings: Measurement Validity and Unsupported Claims Derived from AI Agent Telemetry

Rolando Bosch
Hermes Labs — San Francisco, California, USA
roli@hermes-labs.ai

July 30, 2026

Abstract

Agent systems are increasingly evaluated and modified through operational telemetry. Yet precise records do not necessarily identify the sessions, tasks, outcomes, or behaviors named by later metrics. When those interpretations guide routing, memory, autonomy, evaluation, or redesign, measurement validity becomes an architectural concern.

We report a naturalistic measurement-validity audit of one coding-agent orchestration-harness deployment observed over twelve weeks. The frozen corpus comprised 41,495 transcript files and six auxiliary telemetry streams. Each candidate claim was examined for producer provenance, analytical unit, construct, population, observation window, denominator, validation evidence, and sensitivity to alternative treatments of inter-event gaps.

Across the audited mappings, valid record-level quantities did not support several broader construct-level interpretations. Transcript-file event spans produced a file-level distribution but no validated measure of logical-session or task duration; 46 of 87 files spanning more than eight hours fell below that threshold after removal of their largest internal gap. Of 28,055 claims-ledger rows, 27,938 were automatic retrieval-reflex events; the remaining 117 heterogeneous entries did not establish completed tasks or correct outcomes. An anomaly instrument flagged approximately 65% of scored turns, but without independent behavioral labels this was an instrument-positive rate, not an estimate of failure prevalence.

The study provides an empirical account of how record-level measures change at the claim layer and proposes a Telemetry-to-Claim Gate for documenting producer, unit, construct, population, window, denominator, validation, and sensitivity before telemetry is used in evaluation, governance, or system redesign.

Keywords: AI agent telemetry; measurement validity; construct validity; agent evaluation; operational telemetry; telemetry provenance; agent observability; coding agents; Telemetry-to-Claim Gate

Author's Note on Method, Authorship, and AI-Mediated Research

This preprint emerged from the Hermes Autonomous Lab (HAL), an agentic research and software environment that I designed and operate at Hermes Labs. I tasked the Lab with identifying and advancing a small number of high-leverage, high-autonomy investigations capable of producing publishable evidence. This audit was one of the resulting projects.

The manuscript was not produced by asking an AI system to draft a predetermined argument. Its central argument emerged through a maieutic process: repeated questioning, conceptual pressure, counterargument, evidence checking, and adjudication across Claude Code and OpenAI Codex sessions. I selected the problem, established the evidence boundary, challenged proposed interpretations, accepted or rejected revisions, controlled scope, and made the final publication decisions. The systems performed substantial corpus inspection, analysis programming, aggregation, visualization, drafting, criticism, and revision. After the first empirical-paper draft, the manuscript underwent six documented substantive revision passes. Fresh-context and adversarial model reviews were used at several stages as internal challenge mechanisms, not as independent peer review.

Most of the prose in this manuscript was written and revised through AI systems. I wrote this note myself, then used AI to help clarify it. I am listed as the sole author not because I manually wrote every sentence, but because authorship here denotes accountable judgment. I directed the work, determined what claims the evidence could support, resolved interpretive disagreements, and accept responsibility for the final artifact and its errors. The AI systems involved cannot assume that responsibility.

This mode of production creates a real tradeoff. A fundamental thesis of Hermes Labs' work is that language is not simply a form of communication; it is also infrastructure, environment, and interface. A writer's contribution lies partly in propositions and partly in linguistic texture: what is foregrounded, resisted, repeated, or left implicit, and how questions are formed. When AI systems compose most of the prose, some of that texture is inevitably flattened or transformed. The resulting paper may become easier for some readers to enter while losing part of the horizon from which its ideas arose. I accept that tradeoff here in the interest of making the evidence available while it remains relevant, and I disclose it rather than presenting the prose as conventionally authored.

As answers become abundant, inexpensive, and immediate, generating genuinely new insight increasingly depends on forming better questions, recognizing when a fluent answer has outrun its evidence, and adjudicating among plausible interpretations. That premise shaped both the process that produced this paper and the paper's argument: operational telemetry must earn the meanings assigned to it before those meanings are used to evaluate, govern, or redesign agent systems.

This case documents one application of a maieutic, AI-mediated research process. It does not establish that process as generally effective; evaluating it as a research method would require separate study. Related processes informed earlier Hermes Labs publications, including *The Asymmetric Burden of Proof* [1] and *A Taxonomy of Epistemic Failure Modes in Large Language Models* [2]. Those works demonstrate continuity of method, not validation of it.

1 Introduction

Persistent agent systems are increasingly surrounded by records of their own operation: transcript files, tool traces, evaluator verdicts, detector flags, receipts, summaries, and derived metrics. These records begin as traces of what happened. They do not necessarily remain traces. They can become inputs to evaluation, routing, memory, autonomy limits, retry policies, release decisions, and the redesign of the harness itself.

That transition gives operational telemetry an unusual role. It describes a system while also supplying evidence for changing that system. Before a number enters that feedback loop, an operator has to know not only whether it was calculated correctly, but what it is a number of.

The distinction is easy to miss because the records often look precise. A transcript file has timestamps and can be measured to the second, but it may contain an interactive exchange, a probe, a spawned run, a resumed fragment, or orchestration traffic. A receipt may be durably stored while establishing neither that a task was completed nor that the result was correct. A detector may apply one rule consistently across thousands of turns while its positive rate describes the detector threshold rather than the prevalence of behavioral failure. Precision at the trace layer is not validity at the claim layer.

If those distinctions are ignored, accidental properties of storage and instrumentation can acquire operational authority. A transcript file is treated as a session; session duration becomes an autonomy metric; the harness is then tuned to increase it. A receipt is treated as completion; receipt production becomes a success proxy. A detector flag is treated as failure; policy is optimized around the detector's idiosyncrasies. These are general failure paths, not downstream effects demonstrated in this study. They explain why the prior interpretive step matters: once a measurement is promoted into a control signal, its error can become architectural.

This is a hermeneutic problem in a precise and practical sense. The issue is not only whether the record was produced faithfully. It is whether the passage from record to meaning is warranted. Agent telemetry can therefore be arithmetically correct yet hermeneutically unfit to govern the system it describes: the value is real, but it has not earned the interpretation required for the role assigned to it.

A related conceptual preprint, *The Generative Horizon: Applied Hermeneutics, Linguistic Attractors, and the Limits of Model Self-Report* [3], develops a broader account of how interpretations can acquire operational authority. The empirical argument here addresses a distinct measurement-validity problem at the record-to-claim boundary; it does not depend on, test, or validate the framework of recursive interpretive conditioning.

The present study asks what must be established before operational records can support claims about sessions, tasks, failures, accomplishments, or extended event spans. We examine one coding-agent orchestration-harness deployment operated over twelve weeks. The harness launched, resumed, and coordinated heterogeneous coding-agent work while accumulating 41,495 transcript files and six auxiliary telemetry streams. Harness-level operation continued across the observation window even as the harness and its instrumentation evolved. No equivalent continuity was assumed for an individual model process, operating-system process, logical session, task, or transcript file.

The deployment is not offered as representative of every agent fleet or as an ideal telemetry architecture. It functions as a naturalistic stress case. Its instruments evolved for operational use, began at different times, changed schemas, and recorded a mixture of agent and machine activity. Those conditions made a broader measurement problem unusually visible: a semantically unstable operation can still produce abundant, orderly, internally consistent telemetry, and that apparent precision can lend false stability to the entities and constructs named in later analysis.

Better instrumentation can remove mixed producers, stabilize schemas, and provide explicit links among processes, files, sessions, and tasks. It cannot retroactively establish that a stored file was a logical session, that a logical session was a task, that activity was useful work, or that an instrument output was behavioral ground truth. Those are interpretive and evidentiary claims. They require more than collection.

The audit therefore proceeded from proposed interpretations rather than from attractive aggregates. For each candidate claim, we asked which producer generated the record, what analytical unit was being counted, which construct the unit was supposed to represent, which population and observation window were in scope, what denominator was available, and what

validation evidence connected the record to the proposed meaning. Data could be cleaned or partitioned without granting the resulting metric a stronger interpretation.

That process changed several apparently straightforward readings. A large claims-ledger count became a partition between 27,938 automatic retrieval events and 117 heterogeneous residual ledger entries; neither side identified completed tasks. A tail of apparently long-running "sessions" became a qualified distribution of transcript-file event spans, many dominated by silent gaps. A high anomaly rate became a finding about an instrument awaiting behavioral validation. Short-lived telemetry streams became dated event counts rather than fleet-historical rates. The important result was not simply that some numbers changed. The set of statements those numbers were entitled to support changed with them.

This distinction also appears in recent work beyond this deployment. Studies of coding-agent interactions separate transcript-judged success from harder within-session evidence and from durable repository outcomes; benchmark audits show that precise scores can be dominated by defects in prompts, tests, or task definitions. These populations and rates are not directly comparable to the present corpus. Their relevance is conceptual: traces, apparent success, durable state, and verified outcome are different evidentiary objects.

The paper makes three bounded contributions. First, it documents how multiple plausible operational interpretations changed when one deployment's telemetry was audited at the level of producer, unit, construct, coverage, denominator, and validation. Second, it preserves those changes in a claims ledger that distinguishes transformations of the data from dispositions of the claims. Third, it derives a proposed Telemetry-to-Claim Gate for structuring and recording the evidentiary basis on which an operator may decide whether and how to use a telemetry-based claim. The study does not estimate productivity, useful work, task success, orchestration reliability, downstream harm, or the prevalence of these problems across other fleets.

2 Background and related work

Observability standards give spans, metrics, logs, and events shared names [4]. Session conventions can provide identifiers and links between episodes [5]. These facilities improve collection and correlation. They do not, by themselves, establish that an emitted identifier corresponds to the analytical session, task, or population named in a later claim.

Designed evaluations can begin with tasks that have defined outcomes and human-duration estimates [6]. Naturalistic operational telemetry often cannot. It inherits storage units, partial histories, and instruments created for purposes other than research. Measurement modeling supplies the relevant distinction: an observable indicator is not identical to the construct it is intended to represent [7]. Empirical software-engineering guidance makes the same demand when repository records stand in for a process of interest: the unit, collection procedure, and threat model must be explicit [8-10].

Recent coding-agent studies make the distinction concrete. Anthropic's analysis of approximately 400,000 interactive Claude Code sessions separates transcript-judged success, "verified success" with a hard within-session signal, and real-world outcomes it could not observe [11]. SWE-chat links interaction traces and agent-authored code to whether that code survives into user commits, separating trace production from durable repository state [12]. OpenAI's audit of SWE-Bench Pro shows how defects in prompts, tests, and task definitions can dominate an apparently precise capability result [13].

Those studies use different populations, units, and denominators from the present corpus, and their rates are not compared here. Together they sharpen the evidentiary question: what permits

a trace to stand for success, persistence, failure, or outcome? This study applies established validity, provenance, missing-data, denominator, and sensitivity principles to operational claims proposed over naturally evolved agent telemetry.

3 Study setting and data

3.1 Study setting

The observed system is one coding-agent orchestration-harness deployment operated by one person from late April through July 23, 2026. Harness-level operation continued throughout that window, but the harness and its instrumentation evolved while launching, resuming, and coordinating coding-agent work and writing several telemetry streams. No equivalent continuity was assumed for an individual model process, operating-system process, logical session, task, or transcript file.

Transcript files were the primary stored unit available for analysis, but a file could represent a probe, a spawned run, an interactive exchange, or a resumed fragment. Resume and bridge linkage existed operationally but was not validated as an analytical session boundary, so files were not merged into inferred sessions or tasks.

The corpus is naturalistic. The harness and its instruments evolved for operational use rather than for this study. Streams began on different dates, changed schemas, and included both agent-authored and machine-generated traffic. These conditions are part of the evidence boundary, not defects silently repaired away.

3.2 Included data

The frozen analytical snapshot contains 41,495 transcript files and six auxiliary streams recording turn-level text shape, anomaly scores, claims-ledger activity, drift summaries, Hermeneutic gate verdicts, and stop-too-soon flags. The streams were analyzed for different questions; they were not joined into a synthetic account of task performance.

Of the transcript files, 36,502 entered the primary stratum, 4,992 were usable only with declared provenance or parsing caveats, and one contained no eligible event from which to calculate an event span. The evidence dossier preserves the exact stream inventory, schema history, coverage windows, exclusions, and aggregate bindings.

3.3 Privacy boundary

The extraction pipeline used timestamps, event classes, tool-activity indicators, categorical fields, and numeric text-shape measurements. Message content was not read into the analytical dataset. Public outputs contain aggregate tables, day-level or bucketed time, neutral source identifiers, and no session identifiers. Raw transcripts, working rows, salts, local paths, detailed control-plane implementation, and internal review dialogues remain private.

4 Audit methodology

4.1 From records to candidate claims

The audit began by stating the operational claim that a metric was being considered to support, for example that transcript-file event span represented session duration, that a receipt represented completed work, or that a detector-positive rate represented behavioral-failure prevalence. Each

candidate claim was then decomposed into its metric, analytical unit, intended construct, population, observation window, denominator, and proposed interpretation. The claim was challenged against producer provenance, schema history, coverage and missingness, contamination, linkage, detector validation, sensitivity to plausible definitions, and available outcome evidence.

Candidate claims were selected from the author-directed research brief and protocol, from names and fields in the available telemetry, and from plausible interpretations proposed by research agents during the audit; they were not all claims previously used in operation.

This order matters. A data treatment can repair a population without validating the construct attached to it. Partitioning rows on a native class field can identify recorded instrumentation classes while leaving unanswered who produced the remaining rows or whether any row represents a completed task.

4.2 Record treatment and sensitivity

Records entered a primary stratum only when they satisfied a declared source-specific rule. Records with known provenance, identity, schema, parsing, or backfill limitations were partitioned into a usable-with-caveats stratum. Exact duplicates, excluded producers, and records without a valid observation were quarantined. Unobserved time inside a stream's span was reported as missing rather than imputed as zero. The exact deterministic rules and their seeded development check are documented in the method appendix.

Transcript-file event span was the wall-clock interval between the first and last eligible recorded events in a file. It did not establish continuous process execution, preserved state, active engagement, useful work, task duration, or session persistence. A gap-capped inter-event measure was also computed by summing adjacent event intervals after limiting each interval to 5, 15, or 30 minutes. The largest internal gap in each file was subtracted in a separate sensitivity test of the apparent long-span tail. Neither transformation was interpreted as an estimate of work or attention.

The claims-ledger partition was derived from the instrumentation's recorded `claim_class` field, not from manual or model-based semantic classification. Of 28,055 entries, 27,938 carried `claim_class=retrieved`. The originating implementation appended one such row when a UserPromptSubmit entity-retrieval hook matched a configured entity, queried the local memory service, and injected returned context. The field therefore recorded a retrieval-reflex event, not an agent-authored claim or a judgment that retrieved material was relevant or correct.

The remaining 117 entries carried heterogeneous class and status values. Ninety-eight had a non-empty `verification` field, which explains their earlier shorthand treatment as receipts, but the schema did not define the residual population as one semantic class. Across the ledger, timestamp and session fields could establish cadence and limited linkage. The available fields did not establish a task identifier, completion, success, correctness, or explicit producer identity for each residual row. The partition therefore established instrumentation class, not content meaning or task outcome. No exact duplicate claims-ledger row was found in the frozen snapshot.

Detector outputs were treated first as observations about instruments. A behavioral interpretation required independent evidence that the instrument operationalized the named behavior. Short-lived streams were analyzed only inside their observed windows; silence before first observation or during an unknown outage was not converted into a zero rate.

One such stream contained verdicts from Hermeneutic v0.1.7, open-source software developed and maintained by Hermes Labs. Its `PASS` and `RISK` verdicts were instrument outputs, not verified overclaim labels. The study did not evaluate Hermeneutic's correction mining, retrieval, integrations, or downstream effectiveness.

4.3 Claim disposition and analytical scope

Data treatments were recorded separately from claim dispositions. A proposed interpretation could be retained, narrowed to a smaller population or window, relabeled as an instrument or storage finding, withheld for lack of validation, or declared not identifiable from the available telemetry. This separation prevented successful cleaning from being mistaken for successful measurement.

The analysis is descriptive. It reports counts, proportions, quantiles, gap-capped inter-event measures, and declared sensitivity conditions. It tests no causal hypothesis or population effect beyond this deployment. Parser-development evidence, complete classification rules, survival calculations, stream-level coverage, and bound aggregate tables remain available in the evidence dossier rather than carrying the main narrative.

5 Results

5.1 Stored files did not define the sessions or tasks they appeared to count

The corpus contained 41,495 transcript files. That count was exact, but its apparent unit was unstable. Files were created by interactive exchanges, probes, spawned runs, orchestration work, and resumed activity. Cross-file linkage was not validated, and the available sidechain field did not identify spawned activity known to exist operationally.

Among the 36,502 files in the primary stratum, 35,652, or 97.7%, spanned less than five minutes. That finding describes the storage population: most trusted transcript files were brief. It does not establish that most logical sessions or tasks were brief, because the study could neither join fragments into validated episodes nor prove that every file represented an episode of the same kind.

This is more than a naming preference. If transcript-file event span is promoted into a session or autonomy metric, the surrounding system may begin optimizing a property of file creation and closure rather than persistence of work. The audit therefore retained the file-level distribution and declared task- and session-duration distributions not identifiable from the available linkage.

5.2 Apparent long event spans depended on how silence was interpreted

Transcript-file event span treated the interval between a file's first and last eligible recorded event as one span. That is a valid property of the file, but not necessarily a period of continuous execution, preserved state, or active engagement. The sensitivity analysis asked how much of the result survived when long internal gaps were capped or removed.

The central file-level picture was stable: the median and 90th percentile were unchanged under 5-, 15-, and 30-minute gap caps. The summed gap-capped inter-event measure was not: it ranged from 777.6 to 1,166.0 hours. A single total therefore could not be presented as a definition-independent measure of work or engagement.

The tail changed more visibly. Of 87 files with eligible-event spans above an exploratory eight-hour threshold, 59 satisfied the gap-dominance criterion: one internal gap exceeded half of the file's event span. Subtracting each file's largest internal gap caused 46 to fall below eight hours, leaving 41 above the threshold. These predicates overlap but are not complementary partitions. The supported result is a gap-sensitive tail of stored files. It is not evidence of 87 sustained sessions, and the remaining 41 are not thereby validated as continuous execution, preserved state, active engagement, or completed tasks.

5.3 A cleaner claims ledger still did not measure completed work

The claims ledger contained 28,055 rows. Its filename, claim field, verification field, and aggregate count made agent claims or completed-work records plausible candidate interpretations for testing. The count was displayed operationally as an entry count, but the forensic trace did not show that all 28,055 rows had previously been used as a measure of completed work.

The native field exposed a different instrumentation class. The 27,938 entries carrying `claim_class=retrieved` were automatic records of the entity-retrieval reflex firing. They were well-formed records of that event and were quarantined only from analyses of agent claims or completion. The 117 entries not carrying that value had heterogeneous classes and schemas. Ninety-eight contained non-empty verification objects, but neither that field nor the residual partition established one producer, a task denominator, successful completion, or result correctness.

Resolving this interpretation required applying the same record-to-claim discipline used elsewhere in the audit. A literal native-field partition replaced an unsupported semantic grouping. It did not turn either the retrieval events or the residual entries into task outcomes.

That distinction matters wherever record production is used as a completion proxy. A system rewarded for producing a receipt can satisfy the proxy without satisfying the task. This study does not show that such a policy governed the deployment; it shows that the ledger could not justify one.

5.4 A detector-positive rate described an instrument before it described behavior

After one exact duplicate was quarantined, the anomaly instrument flagged 20,860 of 31,904 scored turns, approximately 65%. The numerator and denominator were well defined. The behavioral interpretation was not.

No independent labels established that the detector's positive class corresponded to behavioral anomaly or failure. The audit therefore retained the rate as a property of the instrument under its observed threshold and withheld the prevalence claim. The positive rate alone did not show that the detector was accurate, inaccurate, or miscalibrated; it showed that calibration and behavioral validation were prerequisites for using the value that way.

Two young streams made the temporal version of the same problem visible. The Hermeneutic gate recorded 128 `RISK` verdicts among 394 events across July 22-23, and the stop-too-soon stream recorded 13 flags on July 22. Those were valid counts inside their observed windows. They were not twelve-week overclaim or failure rates, because the instruments did not exist across that denominator and their verdicts had not been independently labeled as outcomes.

5.5 What survived the audit

The audit did not merely lower counts. It changed the permissible relationship between each record and the meaning proposed for it.

Proposed reading	What the audit established	Disposition
Transcript-file event span measures task or session duration	File-level event spans were observable; task and session linkage was not validated	**NOT IDENTIFIABLE**
87 files above eight hours were sustained sessions	87 files crossed an exploratory event-span threshold; 46 fell below it after largest-gap subtraction, leaving 41	**NARROW**
28,055 claims-ledger rows represented agent claims or completed work	27,938 rows recorded automatic retrieval-reflex events; 117 heterogeneous ledger entries did not carry that class	**WITHHOLD**

Proposed reading	What the audit established	Disposition
The residual 117 entries represented completed tasks	They established residual-entry cadence and, for 98 rows, the presence of a verification object, without a validated task-outcome construct	**NOT IDENTIFIABLE**
Approximately 65% of scored behavior was anomalous	The detector flagged 20,860 of 31,904 scored turns under its observed threshold	**RELABEL**
Hermeneutic's two-day verdict ratio described twelve-week overclaiming	The stream supported only a dated count of instrument outputs	**NARROW**

Some calculations survived while their constructs did not. Other statements survived only after their unit, population, or window was named more narrowly. In every case, the decisive question was not whether a number existed, but whether the record-to-claim connection had been established.

6 Synthesis: from telemetry to defensible claims

The practical question is not simply whether an operator should audit telemetry more carefully. It is what must happen before a particular metric is allowed to support a particular decision.

The transformations in this study followed one recurring pattern. A record offered an apparently obvious meaning: file as session, receipt as completion, flag as failure. The audit reconstructed the conditions surrounding that meaning. It identified the producer and stored unit, named the construct and population, bounded the observation window and denominator, and asked what validation or outcome evidence connected the record to the interpretation. Only then was a claim disposition recorded: retained, narrowed, relabeled, withheld, or declared unidentifiable.

The proposed **Telemetry-to-Claim Gate** is an operational framework for structuring and recording the evidentiary basis on which an operator may decide whether and how to use a telemetry-based claim. Before a metric is promoted into evaluation, routing, governance, or redesign, the operator writes the proposed claim as a tuple:

> metric · analytical unit · construct · population · window · denominator · interpretation

The operator then tests provenance, schema, coverage, contamination, linkage, sensitivity to plausible definitions, detector validity, and outcome evidence. Any necessary data treatment—such as partitioning producers, deduplicating rows, excluding invalid records, or recomputing a sensitivity condition—is recorded separately from the disposition of the claim.

That separation is the method's central operational move. Separating entries marked `claim_class=retrieved` clarified one instrumentation class; it did not turn the heterogeneous residual entries into completed tasks. Removing a duplicate repaired the anomaly denominator; it did not validate failure. Data can become cleaner while the desired interpretation remains unsupported.

A `RETAIN` disposition records that the framework's stated checks were satisfied under declared assumptions; it does not certify truth. A narrowed or relabeled claim is considered only under its revised meaning. A withheld or unidentifiable claim can still describe the existence of records, but it should not be used as evidence of the missing construct. The framework structures and records the basis for the operator's claim-use decision; it does not make that decision.

This is an agent-telemetry-specific operational synthesis and field application of established measurement principles, not a new theory of validity or an automated truth classifier. Its complete

worksheet and worked examples are included with the supplementary materials. The framework was derived from this case; the consistency of its application across analysts and deployments and its effects on decisions were not evaluated.

7 Discussion

7.1 Telemetry as operational scaffolding

Agent observability is often discussed as though better collection will progressively reveal the operation as it really is. The present case shows why collection and interpretation have to remain separate. Better producer, schema, episode, continuation, and outcome fields would have prevented several ambiguities found here. They would not, by themselves, establish that a file was a task, that activity was useful work, or that an instrument output was behavioral ground truth.

The distinction becomes consequential when telemetry enters a feedback loop. A metric used only for inspection can be renamed or discarded. A metric used to allocate retries, summarize memory, alter autonomy, rank systems, or trigger redesign begins to shape the operation according to the interpretation embedded within it. The system then acts on an account of itself produced by its own instruments.

This study did not measure those downstream effects. It located the prior failure boundary: the point where an arithmetically correct record is assigned a construct it has not been shown to measure. The proposed framework makes that passage from trace to claim inspectable. It does not decide what an organization should value, how much uncertainty it should tolerate, or which intervention is worth its cost.

7.2 Self-measurement and external validation

Because the telemetry was generated and audited within the same institution, internal knowledge made producer and schema history recoverable while also creating a risk of favorable interpretation and procedural self-confirmation.

The study therefore reports claim dispositions, evidence bindings, exclusions, and sensitivity conditions as inspectable features of this case. These materials support auditability; they do not provide independent replication.

The next evidence should come from application of the Gate to claims from unrelated operators and schemas, prospective instrumentation with explicit episode and outcome fields, and intervention studies testing whether claim dispositions change consequential decisions.

8 Threats to validity

Setting and analytical units. The empirical results describe one deployment operated by one person. They establish neither industry prevalence nor the Gate's effectiveness across fleets. Transcript files were observable, while logical sessions and tasks were not reliably linked; file-level findings cannot be promoted to task- or session-level findings.

Classification and private evidence. Privacy restrictions prevent independent reproduction from raw transcripts. The deterministic classification rules and public aggregates are inspectable, but the 400-file seeded parser check had no independent labels, second analyst, or error estimate. The interpretation of native producer fields also depends on the emitting instrumentation.

Coverage and sensitivity. Streams began at different times, schemas changed, and outages could not always be distinguished from quiet periods. Event-span tiers and gap dominance were

exploratory sensitivity devices defined after the distribution was inspected. Central quantiles survived the tested gap caps, while the summed gap-capped inter-event measure did not.

Constructs and outcomes. Detector and gate streams lacked independent behavioral labels. The corpus contains no validated measure of task success, productivity, usefulness, orchestration reliability, downstream harm, or the decisions that these telemetry interpretations may have influenced.

9 Reproducibility and disclosure

Data and code availability. The public evidence dossier contains the frozen snapshot definition, aggregate tables and figures, claims ledger, classification and sensitivity appendix, field-level provenance and exclusions, file hashes, and a deterministic verifier for package integrity and selected manuscript-to-aggregate bindings. The verifier establishes package consistency, not source-corpus truth, construct validity, or cross-fleet replication. Raw transcripts, working rows, identifiers, salts, and sensitive orchestration details remain private. The public paper, reports, prose, templates, aggregate evidence, and figures are licensed CC BY 4.0; verification and rendering code are licensed MIT.

AI contribution and accountability. The initial audit concept emerged during a Claude Code session and was selected, bounded, and developed into the present study by Rolando Bosch. Bosch determined the evidence boundary, adjudicated interpretations, controlled scope, made the final publication decisions, and retains responsibility for the work. The Hermes Autonomous Lab, operating through Claude Code and OpenAI Codex sessions, performed substantial corpus inventory, analysis programming, aggregation, visualization, validation, criticism, and drafting under human-set constraints. Model-generated criticism was an internal challenge mechanism, not independent peer review or scientific validation. No AI system is listed as an accountable author.

Competing interest. Bosch leads Hermes Labs, which develops and maintains Hermeneutic. The study treats Hermeneutic verdicts as unvalidated instrument outputs and does not evaluate the product.

10 Conclusion

Operational telemetry does not remain outside the agent systems it describes. It can become evidence for deciding what succeeded, what failed, what should be remembered, where autonomy should expand, and how the surrounding harness should change. That gives the interpretation of telemetry architectural consequences even when the underlying arithmetic is correct.

In this twelve-week deployment, the audit supported fewer and narrower claims than the records initially appeared to permit. Transcript files were not validated sessions or tasks. Long event spans were sensitive to silence. The claims ledger contained 27,938 automatic retrieval-reflex events and 117 heterogeneous residual entries; neither partition established completed tasks or correct outcomes. A high detector-positive rate described an instrument before it described behavioral failure.

The result is not that agent telemetry is unusable. It is that a metric must earn the meaning required for the role assigned to it. The proposed Telemetry-to-Claim Gate structures and records the evidentiary basis for that judgment by binding an interpretation to its producer, unit, construct, population, window, denominator, validation evidence, and sensitivity to alternative definitions. This case demonstrates why the distinction matters; future work must determine how consistently the framework can be applied and whether it improves decisions.

11 References

1. Bosch Rodriguez, R. The Asymmetric Burden of Proof: LLMs Show a Null-Result Asymmetry in a Matched-Vignette Benchmark. Zenodo, Working paper, March 4, 2026.
2. Bosch Rodriguez, R. A Taxonomy of Epistemic Failure Modes in Large Language Models. Zenodo, Preprint, March 15, 2026.
3. Bosch, R. The Generative Horizon: Applied Hermeneutics, Linguistic Attractors, and the Limits of Model Self-Report. Hermes Labs, Preprint, July 2026.
4. OpenTelemetry. Semantic Conventions 1.43.0.
5. OpenTelemetry. Semantic conventions for session. Status: Development.
6. METR. Time Horizon 1.1. January 29, 2026.
7. Jacobs, A. Z., and Wallach, H. Measurement and Fairness. ACM FAccT, 2021.
8. Runeson, P., and Höst, M. Guidelines for Conducting and Reporting Case Study Research in Software Engineering. Empirical Software Engineering, 2009.
9. Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., and Damian, D. The Promises and Perils of Mining GitHub. Proceedings of the 11th Working Conference on Mining Software Repositories, pp. 92-101, 2014.
10. Menzies, T., and Shepperd, M. "Bad Smells" in Software Analytics Papers. Information and Software Technology 112:35-47, 2019.
11. Hitzig, Z., Massenkoff, M., Lyubich, E., Zhang, S., Heller, R., and McCrory, P. Agentic coding and persistent returns to expertise. Anthropic Economic Research, June 16, 2026.
12. Baumann, J., Padmakumar, V., Li, X., Yang, J., Yang, D., and Koyejo, S. SWE-chat: Coding Agent Interactions From Real Users in the Wild. arXiv:2604.20779, 2026.
13. OpenAI. Separating signal from noise in coding evaluations. OpenAI Research Publication, July 8, 2026.