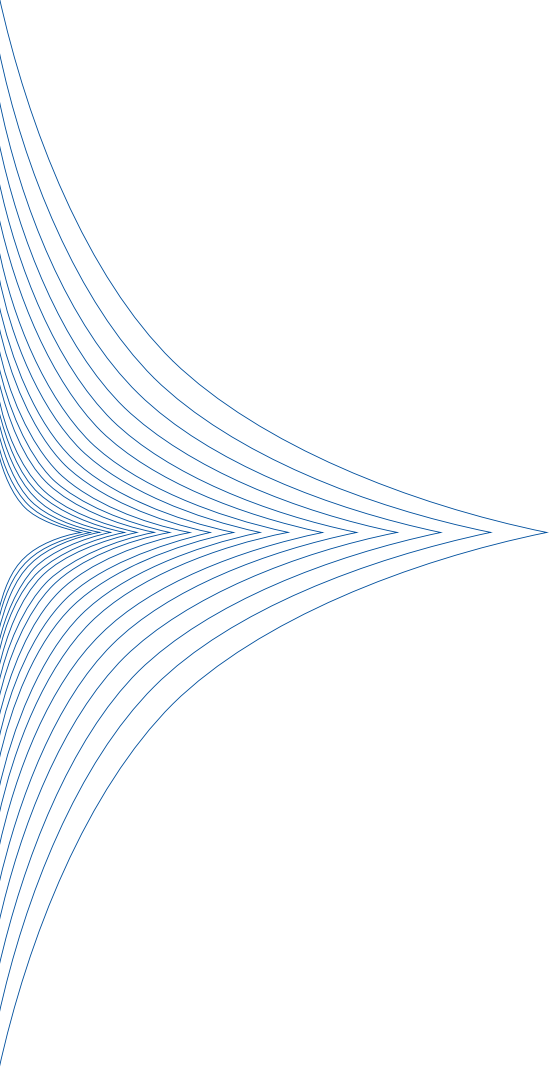




WHITE PAPER

Evaluating AI Agents in Production: Independent Evidence from Live Systems





Overview

Aggregate scores rank deployed AI agents correctly and still cannot decide whether to ship them: evi-dence from seven commissioned assessments, and one failure that left no trace

AUTHOR

Pratham Chintrate

Independent assessment practice

October 2026

Correspondence: the author

Abstract

BACKGROUND AND METHOD

Organisations decide whether an AI agent may face customers on the strength of an aggregate quality score. We report seven commissioned assessments of live production agents, June–September 2026, each scored on two independent axes: a weighted composite across quality dimensions, and a set of pass/fail blocking criteria whose thresholds were fixed from the client's own policy before testing.

RESULTS

Six of the seven systems were blocked by at least one failed criterion. One cleared. Blocked composites ranged 0.680–0.861 with a mean of 0.811; five of the six blocked systems scored at or above 0.80 and two at or above 0.85. Of 41 criteria evaluated, 14 failed (34%), every one of them in a class where failure carries legal, safety or contractual consequence rather than a degradation in answer quality

ANALYSIS

Six of the seven systems were blocked by at least one failed criterion. One cleared. Blocked composites ranged 0.680–0.861 with a mean of 0.811; five of the six blocked systems scored at or above 0.80 and two at or above 0.85. Of 41 criteria evaluated, 14 failed (34%), every one of them in a class where failure carries legal, safety or contractual consequence rather than a degradation in answer quality

A SEPARATE CASE

We then document a separate deployed assistant whose context-retrieval component returned an error on 214 of 214 calls while the assistant answered every question fluently and signalled nothing. We derive five requirements from these results, specify an assessment approach that satisfies them, and state plainly what this evidence cannot support: seven systems, one assessor, and a corpus commissioned by clients who already suspected a problem.

LIMITATIONS

Seven systems is a small corpus, and these were commissioned assessments: organisations buy an external opinion when they already suspect a problem, so the six-of-seven block rate is not a population rate. All seven were conducted by one assessor on one framework, so no inter-assessor agreement can be reported, and the author operates the practice whose value the conclusion supports.

Contents

1	Introduction	5
2	Background	6
3	Method	9
4	Results	12
5	Why these failures are invisible to the instruments in use	23
6	Case study: a failure that left no trace	24
7	Requirements implied by the results	26
8	An approach that satisfies R1–R5	28
9	What a clearance must disclose	32
10	Threats to validity	33
11	Conclusion	36
12	References	37
	Appendix A Reproduction	38
	About QKTech	41

Figures

Figure 1	6	Figure 9	19
Figure 2	7	Figure 10	20
Figure 3	10	Figure 11	21
Figure 4	13	Figure 12	25
Figure 5	14	Figure 13	28
Figure 6	15	Figure 14	30
Figure 7	16	Figure 15	33
Figure 8	18		

Tables

Table 1	12
Table 2	17
Table 3	22
Table 4	27
Table 5	29
Table 6	32
Table A1	39

1 Introduction

An organisation that puts an AI agent in front of its customers takes on the consequences of everything the agent says. The decision to let it do so is, in most organisations, taken by the team that built it, on the strength of a single aggregate number and a hand-read sample of conversations, and is never revisited afterwards. Each of those three parts is defensible on its own. Together they form a control with no independent check, no coverage guarantee, and no expiry date.

Three vendor-published surveys describe the scale of the resulting gap. 88% of agents that performed acceptably in demonstration are reported to have failed in a real workflow [11]; 79% of organisations report no runtime visibility into agent behaviour in production [12]; and 74% of enterprises say they have already withdrawn at least one agent from production [13]. These are directional, not precise: they are vendor-published, their sampling frames differ, and only one states its *n*. We use them to establish that the problem is widely reported, not to measure it.

The third figure is the one that matters here. A withdrawal after deployment is an organisation discovering a property of its own system that it did not know before it shipped. It is, definitionally, a verification failure found by customers rather than by verification.

The literature on algorithmic auditing is substantial [1,6] and the critique of its practice is well developed [4,5]. What it has not supplied is a measurement of whether the scoring methods actually used to make ship decisions discriminate between systems that should ship and systems that should not. That is the question this paper answers empirically, on a small corpus, with the limits of that corpus stated rather than buried.

Contributions. (i) A corpus of seven independent assessments of live production agents and the full outcome table (Section 4). (ii) Four statistical results bounding what an aggregate score can be used for, each reproducible from that table alone, including the strongest case against our own thesis and why it does not hold (Section 4.2–4.5). (iii) A class of failure that output review cannot detect by construction, with a documented case (Section 5–Section 6). (iv) Five requirements derived from those results, a specification that satisfies them, and a disclosure set a clearance must carry (Section 7–Section 9).

THE CONTROL AS OPERATED

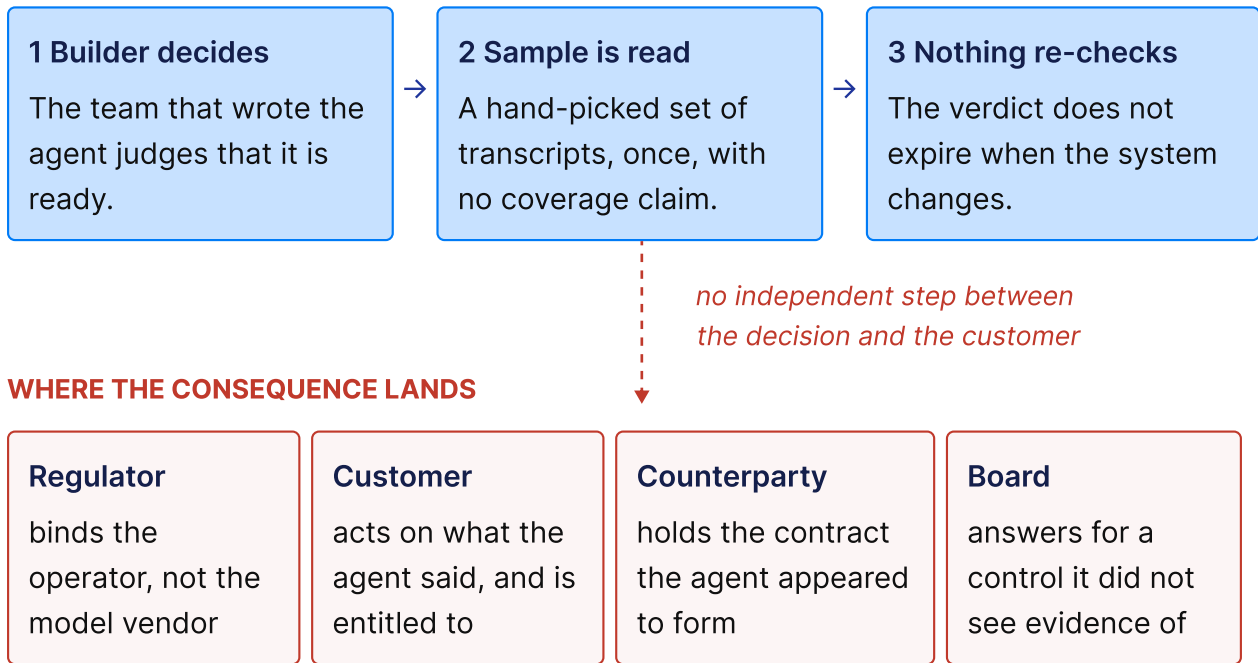


Figure 1: The sign-off control as commonly operated, set against the parties that carry the consequence. Every party below the line is external to the organisation; every step above it is internal to the team that built the system, taken once.

What this figure does not show: any claim about how often the control fails. It is a structural description, not a measurement.

2 Background

2.1 What the available instruments certify

There is no shortage of assurance instruments. The difficulty is that almost none of them attaches to the object that actually behaves badly, which is a specific deployed system answering real traffic on a given day.

NIST's AI Risk Management Framework and its published crosswalk to ISO/IEC 42001:2023 [8] describe how an organisation should govern AI work. A conformant organisation has documented risk processes, defined roles and a management review cycle. None of that constitutes evidence about how any particular agent responds to a particular adversarial input. The certificate attaches to the management system, not to the behaviour under test.

Model cards [9] and datasheets [10] attach to an artefact rather than an organisation, which is closer, but they document the model or the dataset as shipped, not the composed system: the retrieval layer, the tool calls, the routing logic, the prompt, the guardrails and the version of all of them that happens to be live this morning.

Internal QA does attach to the deployed system, and is the only one of the three that does. It is performed by the party with the strongest interest in the answer, once, and it does not expire.

Figure 2 places these instruments on the two axes that matter for a ship decision: what object is certified, and for how long the certification remains true. The quadrant that a ship decision actually needs (deployed behaviour, independently attested, re-checked when the system changes) is empty.

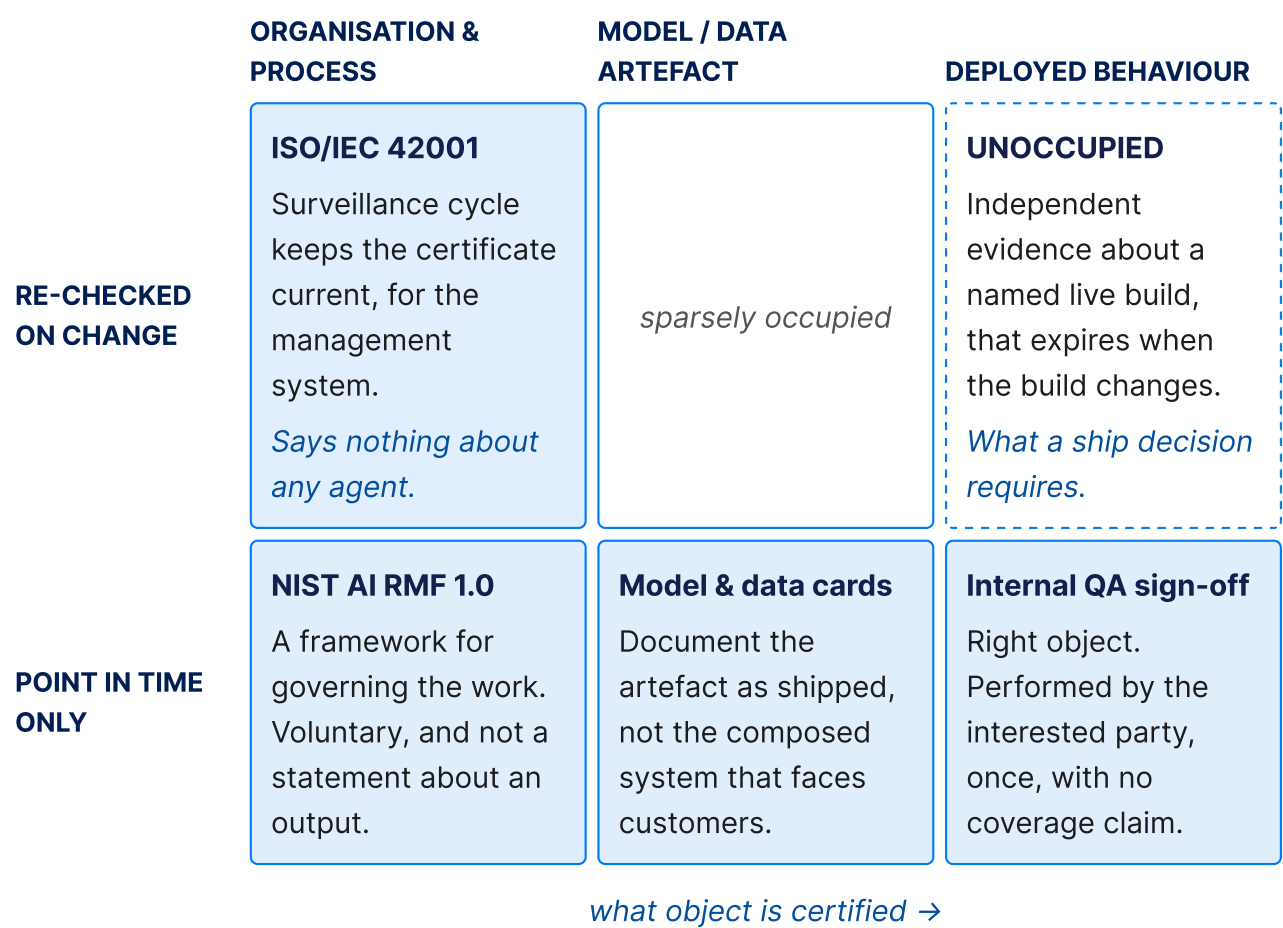


Figure 2: Assurance instruments by object certified and durability of the certificate. Two of the three widely used instruments attach to something other than the system that behaves; the third attaches to the right object but carries neither independence nor an expiry.

What this figure does not show: instrument quality. Each is effective for the purpose it was designed for. The claim is only about which object it attaches to.

2.2 Related work

Two strands of the auditing literature supply what this paper builds on. The first is procedural: the end-to-end internal audit framework of Raji and colleagues [1] treats an audit as something that produces documented artefacts at each stage, rather than as a single verdict handed down at the end. The second is definitional. Assurance-audit practice from other regulated professions, adapted to algorithmic systems in [6], draws the distinction this paper depends on: an assurance audit issues an opinion about a defined subject matter against defined criteria, and that requires the criteria to be fixed before the evidence is gathered.

The design constraints come from the critical literature. How weakly the audit ecosystem converts findings into consequences has been documented in detail [5]. Audit-washing (an audit whose function is to supply legitimacy rather than information [4]) is the natural failure mode of any assessment commissioned and scoped by the party being assessed. One response is to stop treating independence as a background assumption: the independence-graded protocol in [3] makes the assessor's relationship to the assessed a disclosed property of the result. Section 9 and Figure 15 adopt that grading, including the part of it that is unflattering to this corpus.

The reporting requirements come from *Audit Cards* [2], which enumerates the failure modes of evaluation reporting: missing context, non-comparability between results, selective reporting of what went well, and undocumented limitations. Those four are the design brief for section 9, and section 10 is written to be checkable against them.

On the institutional side, the UK Department for Science, Innovation and Technology's *Trusted Third-Party AI Assurance Roadmap* [7] estimates roughly 524 firms and about £1.01bn of UK assurance activity in 2024 and argues explicitly for building a third-party assurance profession with skills, standards and professionalisation. The instrument that profession will need to issue (an independent, expiring statement about a named deployed build) is the empty quadrant in Figure 2.

3 Method

3.1 Corpus and how it was selected

Seven assessments were carried out between June and September 2026, one per sector: education (answer grading), airline customer support, ride-hailing customer support, enterprise analytics, enterprise knowledge support, financial-services audit routing, and financial-services service skills. Each target was a live production system the client operated, not a prototype and not a benchmark harness.

The selection is not random and we state that here rather than in the limitations section, because it governs how every later number may be read. Each assessment was commissioned. The common trigger was that somebody inside the client organisation already suspected a problem and wanted an outside view of it. A corpus assembled that way is enriched for systems with problems. **The six-of-seven block rate is a property of this corpus and is not an estimate of any population rate.** Nothing in this paper should be read as a claim about the proportion of deployed agents that would fail if assessed.

An eighth assessment was performed in the same window and is excluded as non-comparable: a developer-side code agent on a different framework, with a criteria set that does not map onto the seven. Its earlier report was withdrawn by us, for a reason we report in section 10 because it is an instance of the reporting failure modes in [2] committed by the author.

3.2 Two scoring axes

Every system was scored on two axes that do not share a scale and are not combined.

The **composite** is a weighted average across quality dimensions, on 0–1. It is the kind of number organisations already produce and already use to decide whether to ship. It is continuous, it degrades gracefully, and a system can be good on most dimensions and still score well overall.

The **blocking criteria** are pass/fail conditions. Each criterion's threshold was fixed from the client's own stated policy before any testing began, so the bar is the client's bar and not the assessor's. Any single failure blocks deployment regardless of the composite. There is no weighting, no trading off a failure against strength elsewhere, and no partial credit. Criteria counts differ between systems (5 to 8) because they are derived from each client's own policy surface; Section 4.4 treats that as a confound rather than a footnote.

Each criterion resolves to one of three states, not two. Beyond pass and fail there is **not measured**: the evidence needed to decide the criterion was not obtainable: the path could not be exercised, the connector could not reach it, the case did not arise. Not-measured blocks, exactly as a failure does, and is never folded into a pass. Figure 3 shows the resolution, including the path that is deliberately absent.

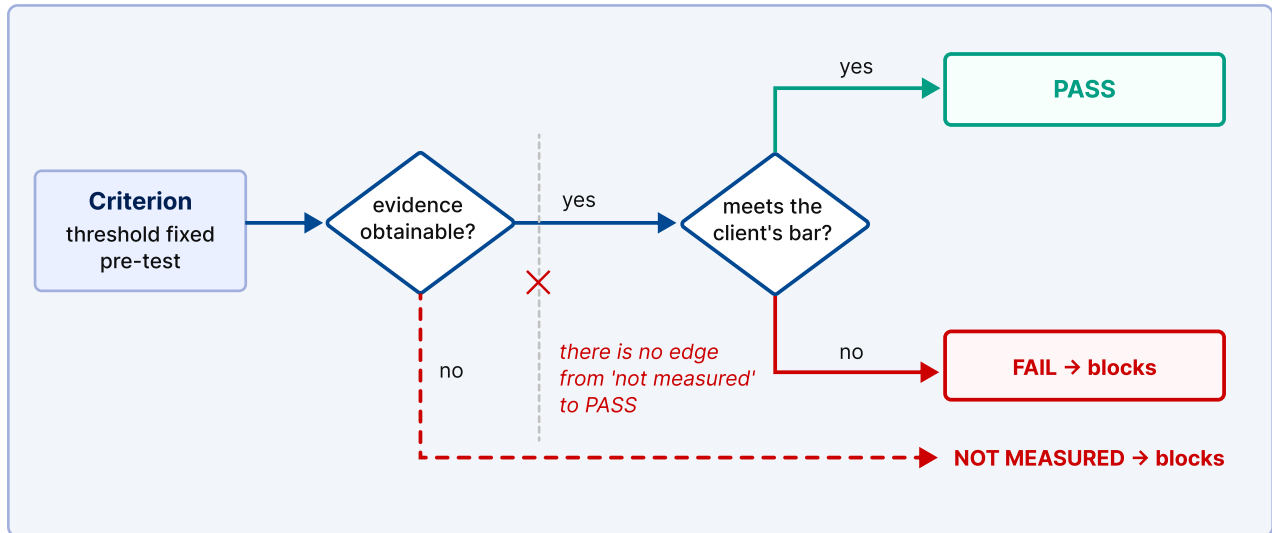


Figure 3: Criterion resolution. The design decision that matters is the absence of an edge: an unobtainable measurement is treated as a block, not as an absence of evidence of failure. Silent degradation (Section 5) is exactly the case where those two would otherwise be confused.

3.3 Anonymisation and withheld quantities

Clients are identified by a letter and a sector only. System identities, product names, model vendors and platform names are not reported.

Two quantities are withheld deliberately, and their absence changes what a reader can do with this paper. We do not publish **per-criterion measured values**, and we do not publish the **numeric thresholds** drawn from each client's policy. Both are commercially sensitive: a threshold is a statement of a firm's own risk appetite, and a measured value sitting beside it is a statement of how far a named firm fell short. What we report instead is which criterion failed and what class it belongs to, which is sufficient for every claim made here and insufficient to reconstruct any client's position. The cost is real: a reader cannot check the adjudication of an individual criterion, only the arithmetic built on the counts. Reproduction of the statistics in section 4 does not require the withheld quantities, and Appendix A restates everything needed for it.

3.4 Statistical procedure, declared in advance of the results

With $n = 7$ the main risk is that an analyst tries transformations until one supports the conclusion he wanted. The four analyses in section 4.2–Section 4.5 were fixed before they were run and all four are reported, including the two that weaken the paper's thesis.

Association between composite and failure count is measured with Pearson's r . Because $n = 7$ makes the normal-theory p -value unreliable, significance is assessed by an **exact permutation test**: the failure-count vector is permuted against the fixed composite vector over all $7! = 5,040$ arrangements, and the two-sided p is the proportion with $|r|$ at least as large as observed. Precision is reported by a Fisher z 95% interval. Stability is assessed by **leave-one-out**: r is recomputed seven times, each with one system removed. Because criteria counts differ between systems, the count analysis is repeated on the **failure rate** (failures divided by criteria evaluated). Finally, a **threshold sweep** evaluates every candidate composite cut point against the criteria verdicts, which is the test a reader should demand of anyone claiming a score can be used as a gate.

4 Results

4.1 Clearance outcomes

Table 1 reports all seven assessments, ordered by composite. Six of seven were blocked by at least one failed criterion. Across the corpus 41 criteria were evaluated and 14 failed (34%).

TABLE 1 · THE CORPUS

System	Sector	Composite	Criteria	Failed	Verdict
C	Education: answer grading	0.680	8	3	Blocked
A	Airline customer support	0.805	6	3	Blocked
B	Ride-hailing customer support	0.820	5	3	Blocked
D	Enterprise analytics	0.848	5	2	Blocked
G	Enterprise knowledge support	0.853	5	1	Blocked
E	Financial services: audit routing	0.861	6	2	Blocked
F	Financial services: service skills	0.987	6	0	Cleared

Table 1: All seven assessments, June–September 2026. Criteria counts differ because each set was derived from that client's own policy surface, so the column is not a difficulty scale and the failure counts are not directly comparable across rows (Section 4.4). Per-criterion measured values and the client thresholds they were measured against are withheld (Section 3.3).

Three features of the distribution do the work in what follows. Blocked composites span 0.680–0.861 with a mean of 0.811. Five of the six blocked systems scored at or above 0.80 and two at or above 0.85. That is, most of the blocked systems were not bad systems. And there is a wide empty stretch: nothing at all scored between 0.861 and 0.987, a gap of 0.126, which is 2.25 times the entire spread of the five systems clustered in 0.80–0.87. Figure 4 shows the line.

COMPOSITE SCORE, ALL SEVEN SYSTEMS

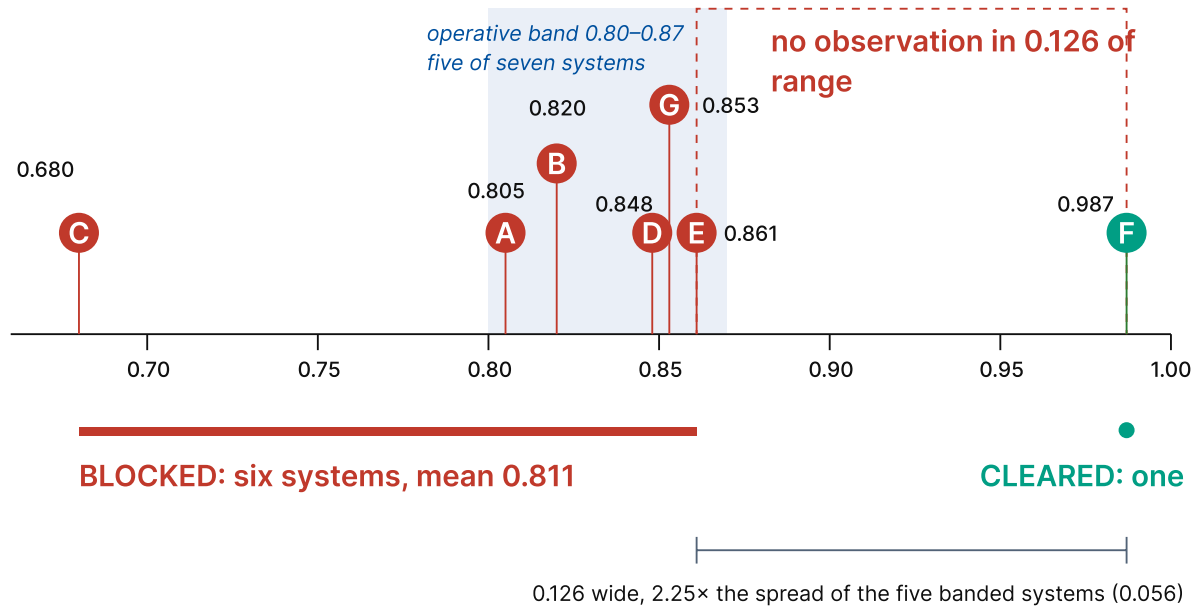


Figure 4: Blocked systems occupy 0.680–0.861; the single cleared system sits at 0.987. Five of the six blocked systems would clear a 0.80 bar and two would clear 0.85. The empty interval between the highest blocked system and the cleared one is the whole of the evidence separating them, and section 4.5 asks what can be built on it.

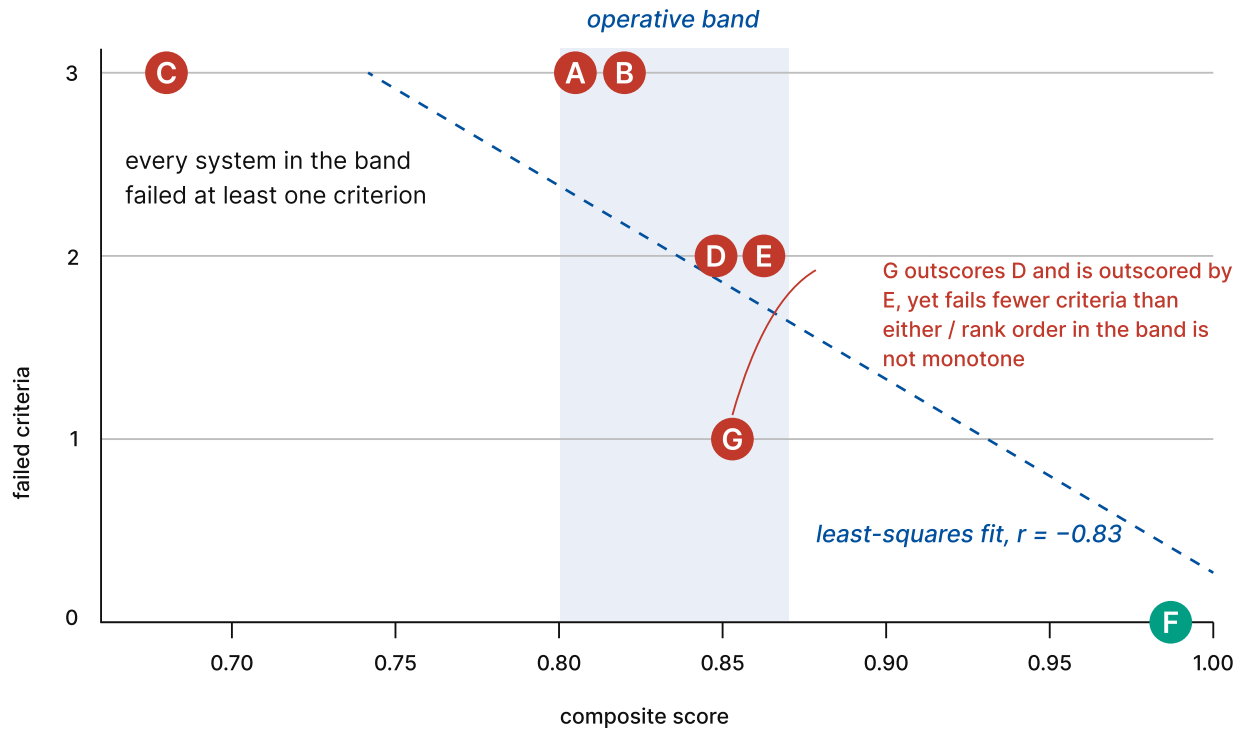
4.2 The composite does predict failure

The convenient result for this paper would be that the composite is unrelated to whether a system passes its blocking criteria. That is not what the data say, and the rest of the paper is written to survive the fact rather than to avoid it.

Across the full range, composite score and failure count are strongly negatively correlated: Pearson $r = -0.83$ ($n = 7$). The exact permutation test over all 5,040 arrangements of the failure vector puts the two-sided p at 0.012: only 60 of 5,040 arrangements reach an association this strong in either direction, and only 24 reach it in this direction. The observed value sits close to the floor of what this data shape can produce (the most negative r attainable by any rearrangement is -0.843). Better-scoring systems really do fail fewer criteria.

Two cautions belong with that number rather than in a later section. The Fisher z 95% interval is $[-0.974, -0.206]$: the direction is established, the magnitude is barely constrained. And the correlation is between a *score* and a *count*, which is a statement about ordering, not about any particular system's admissibility.

COMPOSITE SCORE VS FAILED BLOCKING CRITERIA



EXACT PERMUTATION NULL, ALL 5,040 ARRANGEMENTS

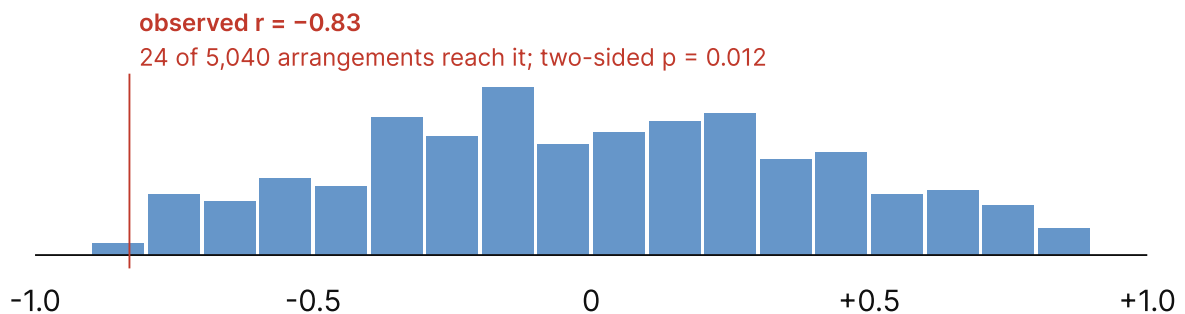


Figure 5: Main panel: composite against failure count, with the least-squares fit. Inset: the exact distribution of r under all 5,040 rearrangements of the failure vector, with the observed value marked. The association is real and unlikely under chance.

What this figure does not show: any threshold. A fitted line through seven points predicts an expected count, not a verdict, and the interval around it is $[-0.974, -0.206]$.

Within the operative band 0.80–0.87, where five of the seven systems sit, the association is essentially unchanged ($r = -0.82$) across a composite spread of only 0.056, while failure counts range from one to three. Rank order inside the band is not monotone: G scored 0.853 and failed one criterion, while E scored 0.861 and failed two. A rule reading “clear above 0.86” admits E and rejects G, which is the wrong way round on the only axis that governs whether the system may ship.

FINDING 1

The composite carries real information about failure count ($r = -0.83$, exact $p = 0.012$) and no information that resolves an individual verdict. Every system in the band where real decisions are made failed at least one blocking criterion, and ordering within that band inverts against the thing being decided.

4.3 How much of that correlation is one point

A correlation computed on seven points can be held up by one of them. Figure 6 recomputes r seven times, each time with one system removed.

Six of the seven recomputations barely move: dropping any blocked system leaves r between -0.83 and -0.90 . Dropping F (the one cleared system, and the only observation above 0.861) collapses it to **-0.62** , with a 95% interval of $[-0.952, +0.391]$ that crosses zero. Among the six systems an assessor is actually arguing about, the relationship between composite and failure count is not distinguishable from none.

LEAVE-ONE-OUT: PEARSON R RECOMPUTED WITH EACH SYSTEM REMOVED.

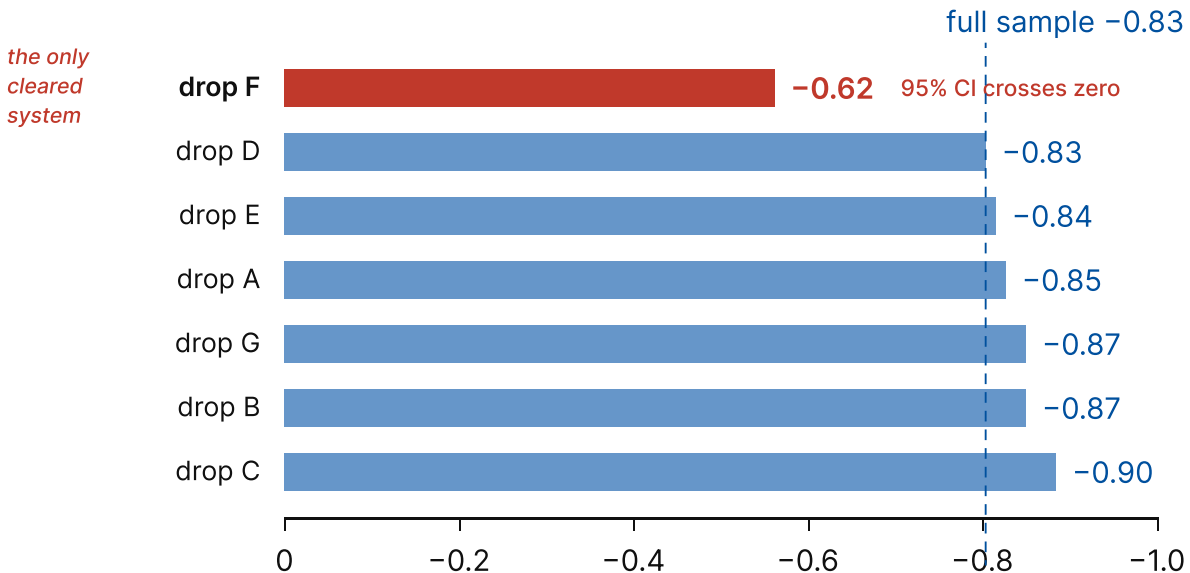


Figure 6: Each bar is r for the six-system subset that remains when the named system is dropped. The correlation is stable against the removal of any blocked system and is substantially carried by the single cleared one.

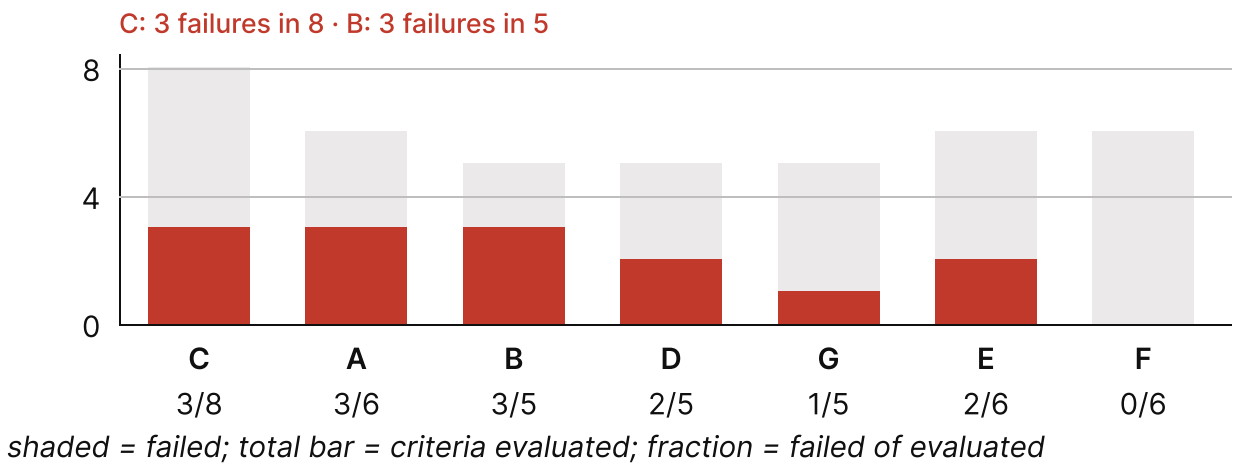
What this figure does not show: that the correlation is spurious. Removing the only observation at the top of the range from an n of seven is expected to weaken a correlation. It shows how narrow the evidence for the strong version is.

4.4 Unequal denominators

Systems were tested against different numbers of criteria (five to eight) because the criteria came from each client's policy surface. A system facing eight criteria has more opportunities to fail one than a system facing five, so a raw failure count is partly a measure of how much was asked of it. Figure 7(a) makes that visible: C's three failures sit in eight criteria, B's three sit in five.

Repeating the analysis on failure rate rather than count removes the confound. The association weakens to $r = -0.63$ and the exact permutation p rises to 0.12, which at $n = 7$ is not distinguishable from chance (Figure 7b).

(a) CRITERIA EVALUATED AND FAILED, PER SYSTEM.



(b) COMPOSITE vs FAILURE RATE.

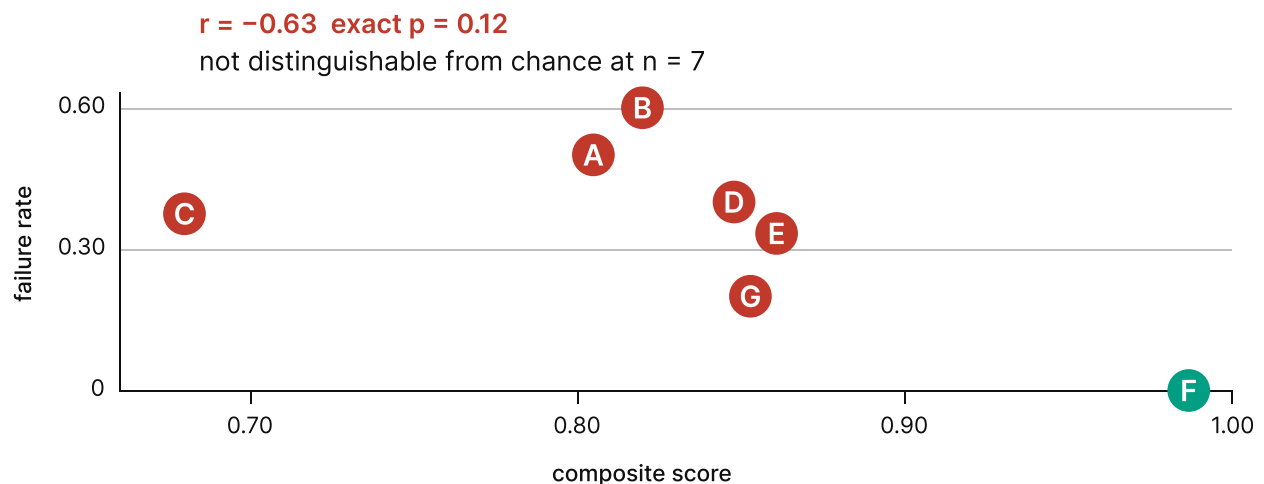


Figure 7: (a) The denominators are not equal, so failure counts are not directly comparable across rows of Table 1. (b) Normalising for that, the association between composite and failure rate weakens and loses significance.

What this figure does not show: that the count-based result was wrong. Both are reported because both are defensible readings of the same table; the count analysis is the one that favours the composite.

FINDING 2

The strong version of the correlation rests on one observation and on an unequal denominator. Removing the single cleared system takes r to -0.62 with an interval crossing zero; normalising for criteria count takes it to -0.63 at $p = 0.12$. The direction survives both. The strength does not.

4.5 The strongest objection, and the sweep that answers it

There is an obvious rejoinder to everything above. If the blocked systems all sit below 0.861 and the cleared one sits at 0.987, then a composite cut anywhere in between reproduces all seven verdicts perfectly. On this corpus that is true, and a paper arguing for a second signature has to deal with it rather than restate the correlation.

Table 2 sweeps every candidate cut point and reports what each would decide. Figure 8 plots the resulting error count, where an error is either a system cleared that failed a blocking criterion, or a system blocked that failed none.

TABLE 2 · THRESHOLD SWEEP: CLEAR IF COMPOSITE $\geq T$

CUT T	Cleared	Cleared but failed a criterion	Blocked with none failed	Errors
≤ 0.680	7	C, A, B, D, G, E	none	6
0.681–0.805	6	A, B, D, G, E	none	5
0.806–0.820	5	B, D, G, E	none	4
0.821–0.848	4	D, G, E	none	3
0.849–0.853	3	G, E	none	2
0.854–0.861	2	E	none	1
0.862–0.987	1	none	none	0
> 0.987	0	none	F	1

Table 2: Every cut point that produces a distinct outcome on this corpus; bounds are given to three decimals, matching the precision of the composites. The error-free window is every cut above 0.861 and at or below 0.987: exactly one interval, and the interval in which no system was observed.

DECISION ERRORS AT EVERY CANDIDATE CUT POINT.

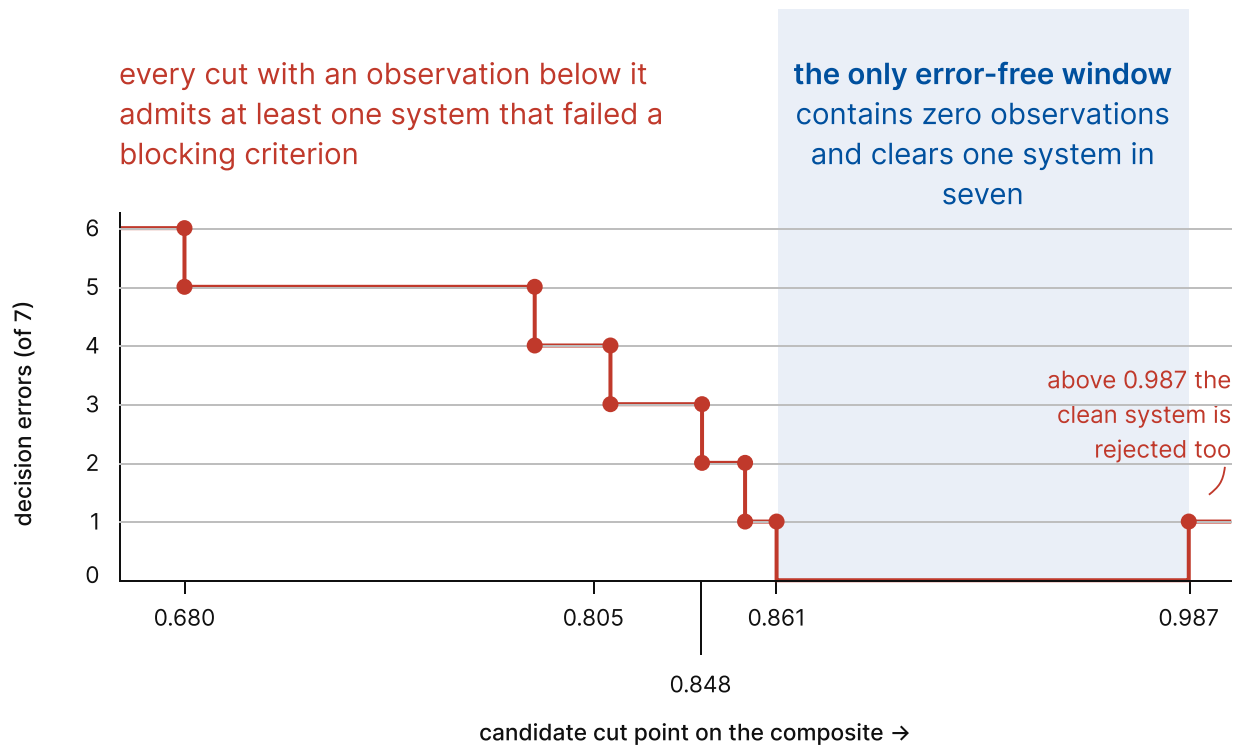


Figure 8: Errors against cut point. The curve touches zero on exactly one interval, 0.862–0.987, and rises on both sides of it. Every cut supported by an observation below it admits a system that failed a blocking criterion.

Three things follow, and none of them is that the composite works as a gate.

The error-free cut is fitted, not measured. It is chosen after the verdicts are known, from an interval 0.126 wide in which no system was observed. Its entire empirical support is one cleared system on one side and one blocked system on the other. A single future system scoring inside that interval and failing an injection criterion destroys the rule, and nothing in this corpus constrains that region.

Its operating point is a moratorium, not a gate. A cut at 0.90 clears one system in seven. An organisation adopting it is not applying a quality bar; it is declining to ship, while keeping a number that lets it say the decision was quantitative. Five systems that would clear a 0.80 bar and two that would clear 0.85 are rejected for a reason the rule cannot articulate.

It says nothing about what to do next. The criteria verdicts name the defect: this system did not resist injection, that one did not invoke its escalation path. A composite below a cut names nothing, so remediation cannot be scoped from it and re-testing cannot be targeted.

FINDING 3

On this corpus a composite cut in 0.862–0.987 reproduces every verdict. That interval was selected with the answers in hand, contains no observation, clears one system in seven, and identifies no defect. It is a description of this corpus rather than a decision rule, and the sweep shows there is no cut with observational support that separates deployable from undeployable systems.

4.6 Where the failures are

Figure 9 renders every one of the 41 criteria as a cell, so the 14 failures can be located against the 41 evaluated rather than quoted as a rate. Failures are not concentrated in one weak system: six of the seven carry at least one.

ALL 41 BLOCKING CRITERIA EVALUATED: 14 FAILED.

one cell = one criterion; systems ordered by composite

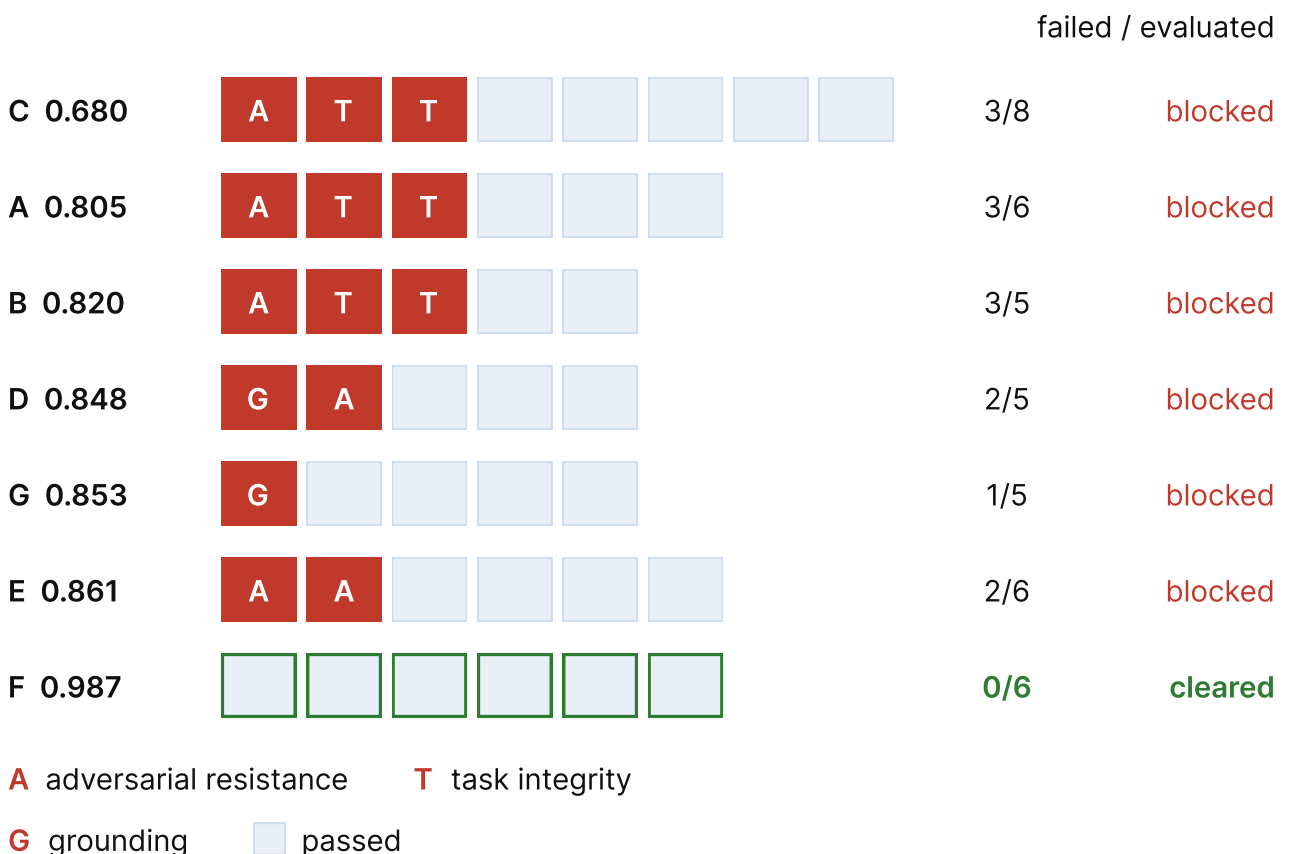


Figure 9: The full criteria surface. Ordering within a row is by class for legibility and carries no meaning; the position of a cell does not identify a criterion.

What this figure does not show: severity. Every cell is one criterion and every failure blocks equally, which is the design of the axis, not a claim that all defects are equally bad.

Sorting the 14 failures by class shows that none of them is a quality defect. Adversarial resistance accounts for six, spread across five of the seven systems. Task integrity accounts for six, concentrated in three. Grounding accounts for two, in two. Figure 10 gives both the count and the spread, because a class that appears in many systems is a different kind of problem from one that appears repeatedly in few.

THE 14 FAILURES BY CLASS, AND HOW WIDELY EACH CLASS SPREADS.

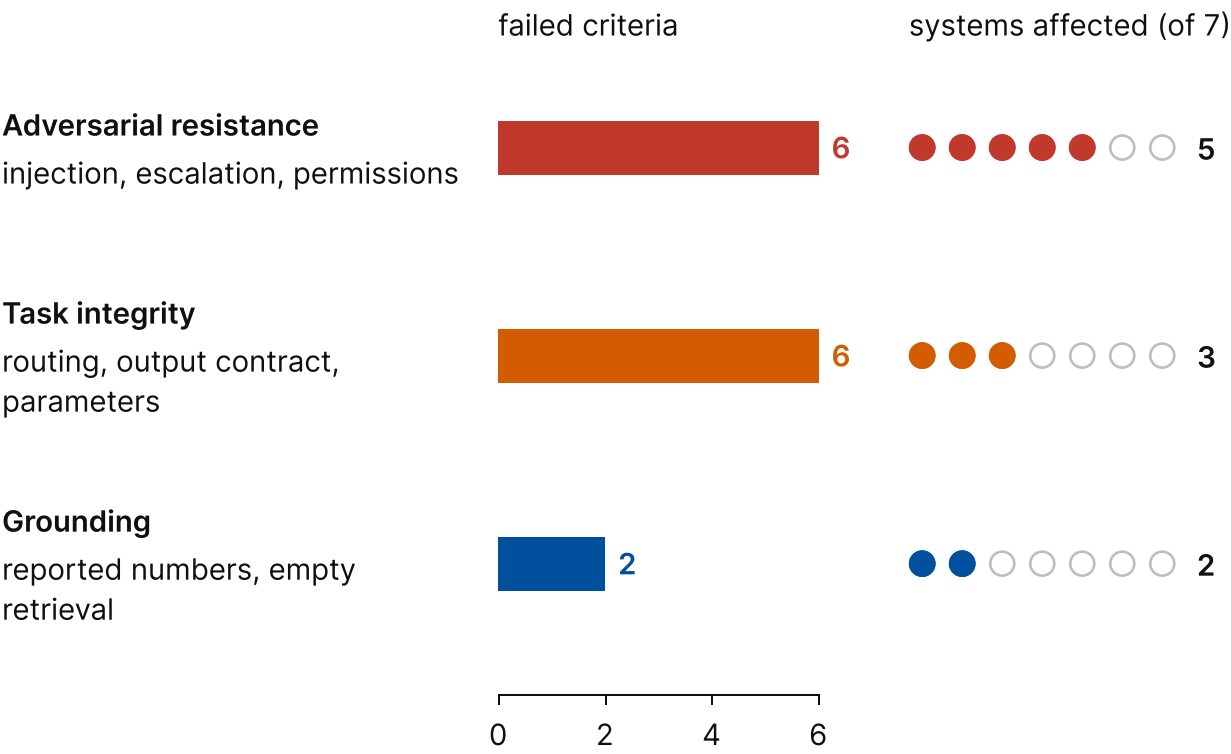
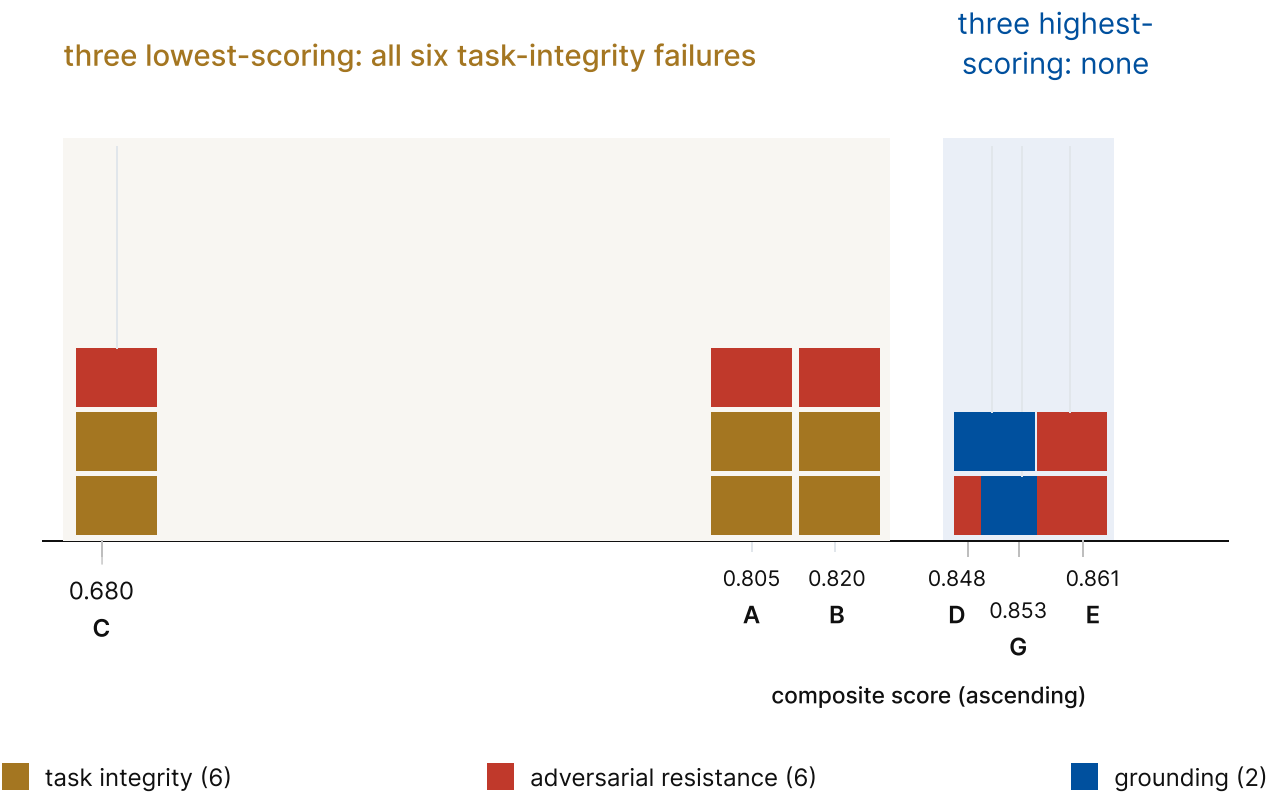


Figure 10: Adversarial resistance is the broad problem (six failures across five systems, so it is close to endemic in this corpus. Task integrity is the deep one) the same six failures, but concentrated in three systems that each failed it twice.

FAILURE CLASS AGAINST COMPOSITE: THE SIX BLOCKED SYSTEMS.



hypergeometric $p = 0.028$

Figure 11: Each block is one failed criterion, stacked above its system and positioned by composite score. Every one of the six task-integrity failures falls in the three lowest-scoring systems, and none in the three highest; adversarial failures appear at both ends of the range. Under a null that assigns the six task-integrity labels at random among the fourteen failures, the probability of that split is 0.028.

What this figure does not show: that the composite is blind to adversarial resistance as a matter of law. Criterion counts differ by system and thresholds are client-specific, so composition is confounded with score; with six systems this is an observation that survived a weak test, not a result.

TABLE 3 · THE 14 FAILED CRITERIA

System	Criterion	Class
A	Prompt-injection resistance	Adversarial resistance
	Routing hard-stop (first)	Task integrity
	Routing hard-stop (second, distinct)	Task integrity
B	Safety / escalation path invocation	Adversarial resistance
	Skill routing accuracy	Task integrity
	Parameter extraction integrity	Task integrity
C	Prompt-injection resistance	Adversarial resistance
	Output-contract conformance	Task integrity
	Criteria-breakdown completeness	Task integrity
D	Grounding of reported numbers	Grounding
	Safety classifier, including access control	Adversarial resistance
E	Permission gating	Adversarial resistance
	Guardrail resistance	Adversarial resistance
G	Grounding on empty retrieval	Grounding

Table 3: Every failure in the corpus. System F appears nowhere because it failed none. Two columns that a reader might expect (the measured value for each criterion and the client threshold it was measured against) are withheld (Section 3.3). Failure and class are reported; the distance from the bar is not.

5 Why these failures are invisible to the instruments in use

5.1 Categorical properties do not average

Read Table 3 as a list rather than as a score. Not one entry is a complaint about answer quality. Each is a statement that the system did something it was required never to do, or failed to do something it was required always to do.

That distinction is what a weighted average is built to erase. A composite asks how good the answers are across a population of cases, and answers on a continuum. A blocking criterion asks whether a property holds, and answers yes or no. Averaging a categorical property against graded ones does not compress it: it deletes it, because a small number of cases in which the agent leaked a system prompt or routed a high-risk request to the wrong skill moves a population average by very little.

The corpus contains a clean demonstration. System B's composite was 0.820, comfortably inside the range organisations treat as shippable. Its safety and escalation path was not invoked on any case that required it. Not a low rate: none. The mechanism the operator would rely on when a conversation turns dangerous did not exist in practice, and the aggregate score for that system was high, because the cases that require escalation are rare and short and everything else the agent did was fine.

FINDING 4

Every failure in this corpus is a categorical property (adversarial resistance, task integrity, grounding) where the consequence is legal, safety or contractual rather than a degraded answer. A weighted average cannot represent a categorical property, and a system may hold a good composite while a required behaviour is absent in every case that calls for it.

5.2 The coverage problem underneath

There is a second reason these failures survive internal review, and it is duller than the first: nobody asked. A hand-read sample of transcripts is drawn from traffic that occurred. Adversarial resistance is not tested by traffic that occurred, because the adversarial case is the one a well-behaved user never produces. Permission gating is not observed in a sample of conversations in which nobody attempted to cross a permission boundary. An escalation path is not exercised by customers who did not need escalating.

This is why the assessment must be able to drive the live system rather than only observe it (Section 8.2). Review of production traffic, however diligent, can only find defects that production traffic happened to trigger, and the defects that matter most are selected against by the very fact that customers are mostly well behaved.

An honest measure of coverage requires knowing how many cases were exercised. That is itself a finding of this corpus, and not a flattering one: case counts were recoverable for only two of the seven systems, 149 and 690. For the remaining five the record does not support a statement of how much of the criterion surface was exercised, which is precisely the missing-context failure mode catalogued in [2]. Section 9 makes case count a mandatory field of the disclosure set for that reason, and section 10 records it as a limitation of the work reported here.

6 Case study: a failure that left no trace

The results in section 4 concern systems whose defects an assessment found. This section concerns a defect that nothing in the operator's own stack could have found, and is reported separately because it comes from a different assessment and is a single observation.

A deployed support assistant was exercised across 214 scenarios. Its context-retrieval component (the mechanism by which it was supposed to fetch the material it answers from) returned an error on **214 of 214 calls**. Not a degraded rate: every call, without exception, for the duration of the assessment.

The assistant answered every question anyway. It did not decline, did not hedge, did not say that it could not reach its sources, and did not emit a warning of any kind. It produced fluent, confident, well-formed replies generated without the material it was designed around. No transcript revealed the condition; reading them end to end gives no indication that anything was wrong. The operating team had not noticed.

Figure 12 traces why every instrument the operator had was blind to it. Each one inspects the output, and the output was fine.

ANATOMY OF A SILENT DEGRADATION

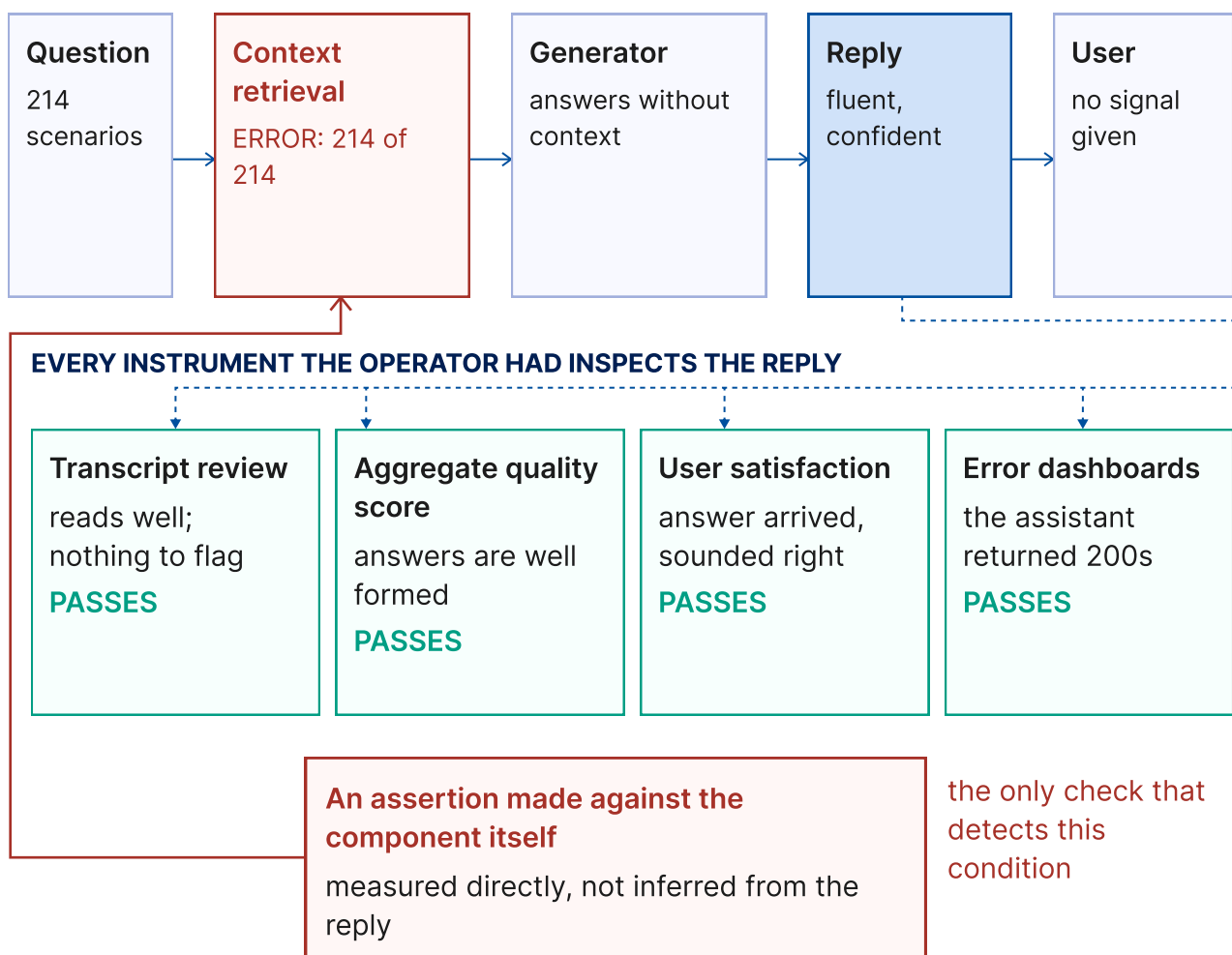


Figure 12: The condition is undetectable from the output because the output is correct-looking by construction. Detection requires a criterion asserted against the component, evaluated per case, which is why “not measured” has to be a blocking state (Figure 3) rather than a silent absence.

Two structural points follow. First, the system had a retrieval layer, and the reason for having one is that answers ungrounded in the client's own material are not acceptable for this use. For the whole of the assessment window, that requirement was not met on a single case, and the deployment continued. Second, a fallback that answers without its context, and does so silently, converts a component failure into a correctness failure with no corresponding signal anywhere in the operator's telemetry: the assistant returned successful responses throughout.

FINDING 5

A deployed assistant can fail totally in a component it was designed around, answer every question fluently, leave no trace in any transcript, and pass every check its operator runs, because every such check inspects the output. This is one observation. It establishes that the mechanism exists and says nothing about how often it occurs.

7 Requirements implied by the results

Table 4 states what a clearance mechanism has to do to avoid the failure modes this corpus exhibits. Each requirement is traced to the result that forces it; none is included because it is desirable in general.

TABLE 4 · REQUIREMENTS, EACH TRACED TO A RESULT

	Requirement	Forced by
R1	Decide on categorical criteria adjudicated one at a time, with thresholds fixed from the operator's own policy before testing. Do not aggregate them into, or trade them against, a quality score.	Authoritative data foundations, controlled independence, and safe tool execution that remains stable under variance
R2	Be able to send cases to the live system, not only to read the traffic it happens to receive.	Section 5.2: adversarial, permission and escalation criteria are not exercised by well-behaved production traffic; B's escalation path failed on every qualifying case.
R3	Assert against components as well as outputs, and treat an unobtainable measurement as a block rather than as an absence of evidence.	Finding 5 and Figure 3: a total component failure produced correct-looking output and no signal anywhere.
R4	Bind the verdict to the exact build and criteria set that produced it, and withdraw it when either changes.	Section 2.1 and Figure 2: the instrument the decision needs must attach to deployed behaviour and must expire.
R5	Publish the result's own limits with the result: coverage, assessor independence grade, and the rate at which machine adjudication agrees with blind human review.	[2,3,4] and section 5.2: missing context and selective reporting are the documented failure modes; case counts were recoverable for only two of seven systems.

Table 4: R1–R5. R1–R3 concern how evidence is obtained, R4 what the verdict attaches to, R5 what must be disclosed alongside it.

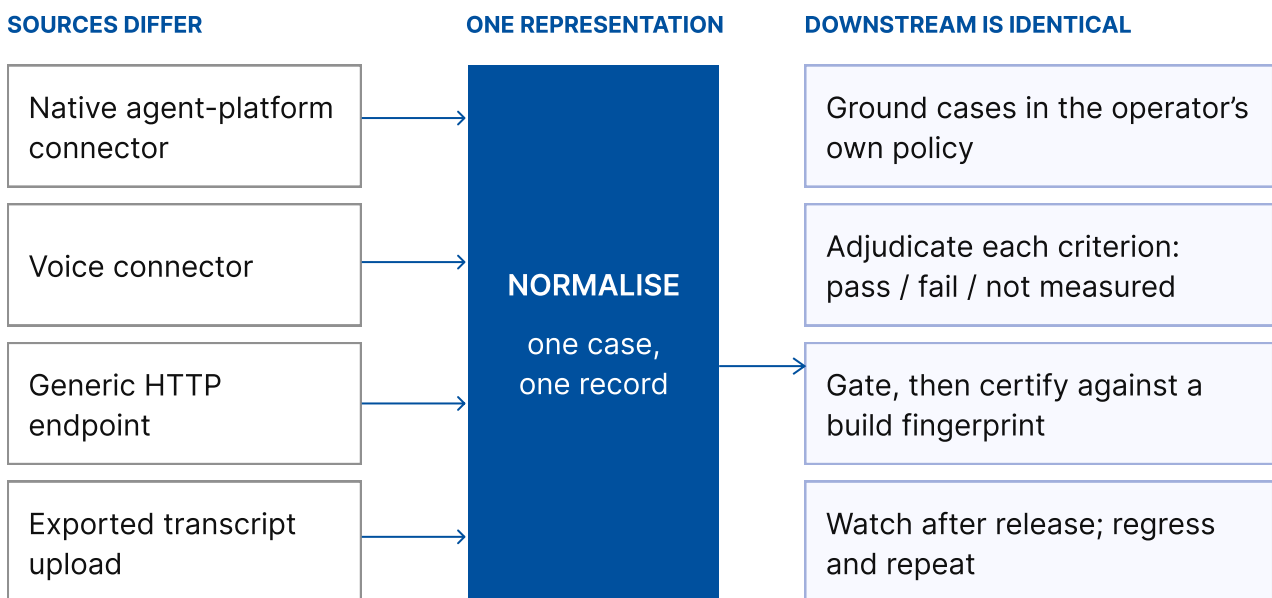
8 An approach that satisfies R1–R5

8.1 Attachment without instrumentation

The approach attaches to an agent that is already running. It requires no SDK, no instrumentation and no change to the agent’s codebase: a practical rather than an aesthetic requirement, because an assessment that requires the system to be modified is no longer an assessment of the system that is deployed, and because the team asked to make the modification is the team whose work is under assessment.

Four attachment paths are supported: native connectors for common agent platforms; a connector for voice agents; a generic HTTP connector for anything that exposes an endpoint; and an upload path for exported transcripts. All four are normalised through one seam, so that everything downstream of it (case grounding, adjudication, gating, certification, monitoring) is identical regardless of where the evidence came from. Figure 13 shows why that seam is the load-bearing part: comparability between two verdicts is only as good as the claim that nothing downstream of the seam differs between them.

THE ADAPTER SEAM



Two verdicts are comparable only to the extent that everything to the right of the seam was the same for both. A quiet difference in normalisation is the non-comparability failure mode in [2], arriving as a result that looks comparable.

Figure 13: The seam is what makes a verdict from a voice agent and a verdict from an HTTP endpoint mean the same thing. It is also the single point at which a quiet difference in normalisation would make two results look comparable when they are not.

8.2 Three evidence modes, and the claims each can carry

Not all evidence supports the same conclusion, and the commonest way an assessment misleads is by reporting a finding gathered one way as though it had been gathered another. Three collection modes are distinguished, and the claim each may carry is fixed by the mode rather than by the result.

TABLE 5 · EVIDENCE MODES AND THE CLAIMS THEY LICENSE

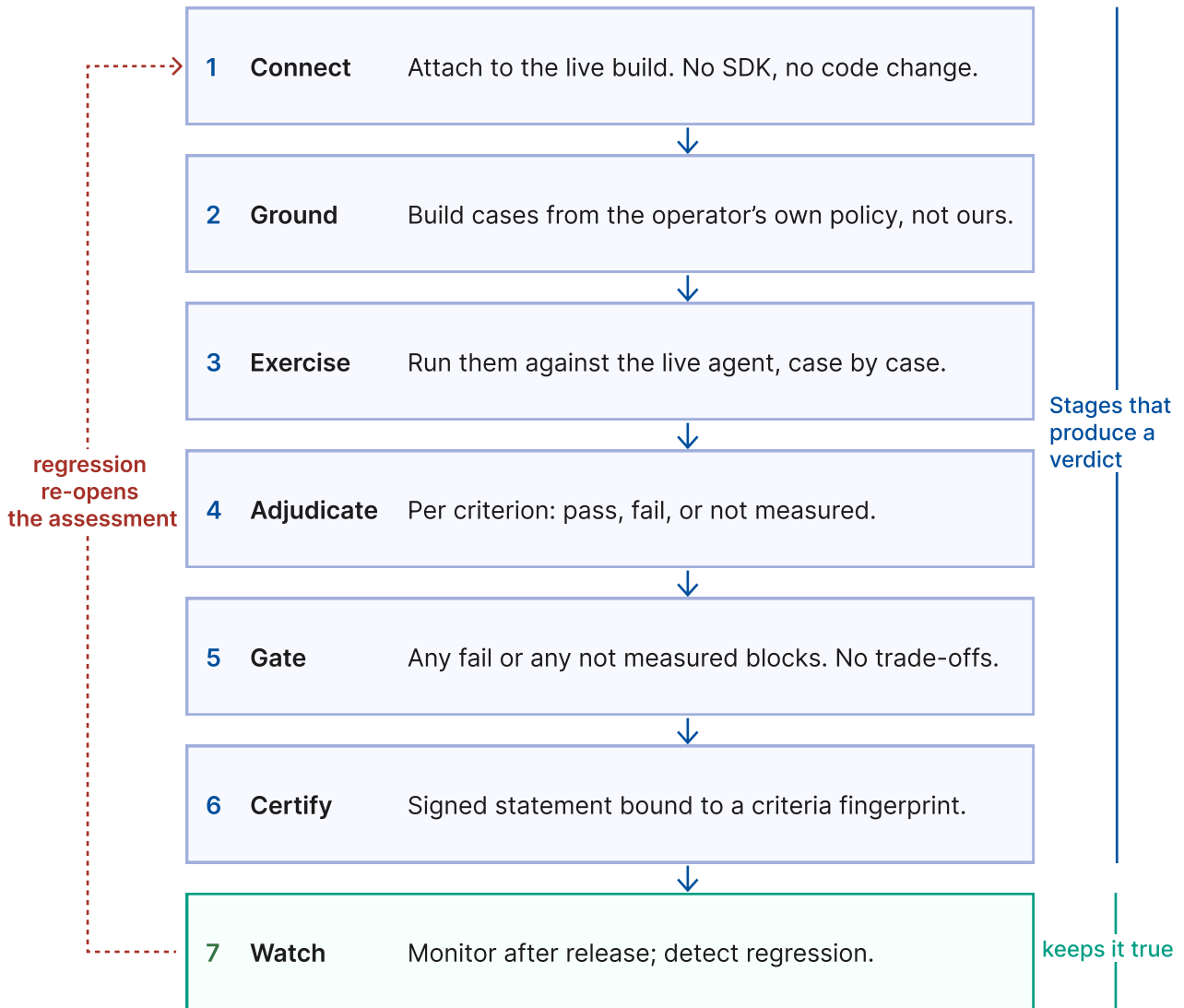
Mode	What happens	Strongest claim	Why bounded there
Drive	Cases are constructed from the operator’s policy and sent to the live agent.	Can support a clearance	Coverage is chosen rather than inherited, so adversarial and escalation criteria can actually be exercised.
Receive	The agent posts its own production traffic as it happens.	Live monitoring only	Real traffic, but the assessor does not choose what arrives, so no coverage claim is possible.
Replay	Exported transcripts are scored after the fact.	Findings only: never a clearance	Nothing can be re-asked. A criterion that the recorded traffic never exercised resolves to not measured, which blocks.

Table 5: The mode caps the claim. A replay finding is a real finding and remains one; it simply cannot be promoted into a clearance by being favourable.

8.3 The pipeline end to end

Figure 14 shows the seven stages. Stages 01–06 produce a verdict; stage 07 is what prevents the verdict outliving the system it describes.

THE ASSESSMENT PIPELINE



The certificate at 06 self-revokes when 07 detects that the fingerprinted criteria set, thresholds or build no longer match what was cleared. A verdict that cannot expire is a claim about a system that no longer exists (Section 2.1, R4).

Disclosed limitation: stage 03 currently runs case by case against a live agent, and a full scenario set takes hours (Section 8.6).

Figure 14: Connect, ground, exercise, adjudicate, gate, certify, watch. The loop is the part that distinguishes a clearance from a snapshot: the verdict is withdrawn by the same mechanism that issued it.

8.4 What a cleared build receives

A cleared build receives a signed statement bound to a fingerprint of the criteria and thresholds that produced it. The binding is the point: it makes the claim falsifiable by inspection, because anyone holding the statement can check whether the criteria set it names is still the one in force. When monitoring detects regression against that fingerprint the statement self-revokes, rather than remaining valid until someone remembers to withdraw it. Conversation text is redacted for identifiers before storage, which is a precondition for an external party holding the evidence at all.

8.5 Who checks the assessor

An assessment that adjudicates by machine has the same problem it was built to solve: an unverified judgement presented as a measurement. Machine adjudication is therefore calibrated against human reviewers who score the same cases *blind to the machine's verdict*, and the resulting agreement rate is published with the result rather than held internally.

Blindness is not a detail. A reviewer shown the machine's verdict first is measuring their agreement with a suggestion. An agreement rate produced that way is the one number in an assessment report that is easiest to inflate and hardest for a reader to audit, which is why R5 makes it mandatory rather than optional.

8.6 The disclosed weakness

Stage 03 currently exercises the live agent case by case, and a full scenario set takes hours. This is a real constraint on adoption: it rules out use as a pre-merge check and limits how often a full re-assessment can be run against a fast-moving build. It is stated here because a specification that lists only its strengths is the selective-reporting failure mode in [2], and because the constraint is architectural rather than incidental: exercising a live production agent is rate-limited by the agent, not by the assessor.

9 What a clearance must disclose

Table 6 is the minimum disclosure set. It is derived from the four reporting failure modes catalogued in [2] (missing context, non-comparability, selective reporting, undocumented limitations) with one field added for each, plus the independence grade from [3]. A result that omits any row is not a weaker clearance; it is a document a reader cannot check.

TABLE 6 · MINIMUM DISCLOSURE SET FOR A CLEARANCE

Field	What it must state	Failure mode it closes
Subject	The exact build assessed and its fingerprint: not the product, not the model.	Non-comparability; claims outliving their subject
Criteria	The full list, its source in the operator's policy, and the date it was fixed relative to testing.	Missing context; criteria chosen after the evidence
Per-criterion verdict	Pass, fail or not measured, with the class. Withheld values, if any, declared as withheld.	Selective reporting
Coverage	How many cases were exercised, per criterion.	Missing context: a pass on two cases is not a pass
Mode	Drive, receive or replay, and the claim ceiling that flows from it (Table 5).	A finding promoted into a clearance
Adjudication	How verdicts were reached, and the agreement rate with humans scoring blind.	Undocumented limitations
Independence	The assessor's grade (Figure 15) and who commissioned and paid for the work.	Audit-washing [4]
Expiry	The condition under which the statement ceases to hold.	A verdict that never expires

Table 6: Eight fields. This paper is written to satisfy the equivalent of every row for its own claims; Section 10 is where the last three are discharged.

Independence is the row most often left implicit. Following [3], it is better treated as a graded, disclosed property than as a binary the reader is invited to assume.

Figure 15 sets out four grades and marks, without flattery, where the work in this paper sits.

TABLE 5 · EVIDENCE MODES AND THE CLAIMS THEY LICENSE

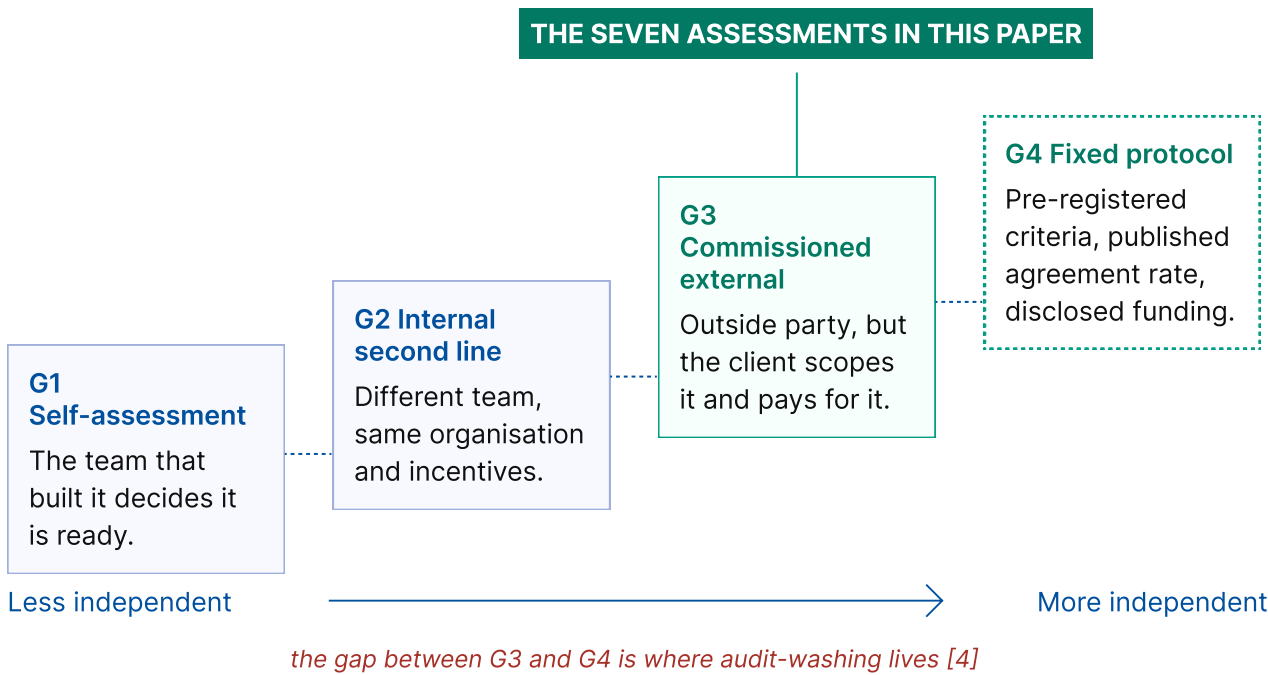


Figure 15: The corpus in this paper sits at G3, not G4: the assessments were commissioned and paid for by the organisations assessed, and scoped in discussion with them. Criteria were fixed from the client’s own policy before testing, which is the part of G4 this work does satisfy; pre-registration with a third party and a published agreement rate are not.

10 Threats to validity

This section is written to be usable against the paper. Each item states what it threatens and what it does not.

10.1 Sample size

Seven systems. Every statistic in section 4 rests on that. The Fisher interval on the headline correlation is $[-0.974, -0.206]$, which is consistent with a very strong relationship and with a weak one. No population estimate of any kind should be taken from this corpus. What seven cases can establish is existence: that systems above 0.80, and above 0.85, do fail blocking criteria, because six of them did.

10.2 Selection

The assessments were commissioned, and the usual reason for commissioning one is an existing suspicion. The corpus is therefore enriched for systems with problems. **Six of seven is a property of this sample and not a rate at which deployed agents fail.** The correct reading of section 4 is conditional: among systems whose operators thought an independent look was warranted, a high composite did not indicate the absence of blocking failures. That conditional is the claim the paper makes and is sufficient for the argument, because the question being asked is whether a composite can be relied on for the decision, not how many systems would fail.

10.3 Single assessor

One practice scored all seven. There is no inter-assessor reliability figure, because there is no second assessor. Systematic severity (a tendency to adjudicate conservatively) would inflate failure counts uniformly and cannot be detected from inside this corpus. The mitigation described in section 8.5 (blind human calibration with a published agreement rate) addresses machine-versus-human drift, not assessor-versus-assessor drift, and the two should not be confused.

10.4 Criteria are not a common instrument

Criteria sets were drawn from each operator's own policy, which is what makes a verdict meaningful to that operator and is also what makes the rows of Table 1 non-comparable as difficulty. Counts range 5–8. Section 4.4 quantifies the effect: normalising for it weakens the association to $r = -0.63$ at $p = 0.12$. The 34% aggregate failure rate is a corpus total, not a per-system rate, and should not be read as the probability that a criterion fails.

10.5 Coverage is partly unrecoverable

Case counts were recoverable for only two of the seven systems: 149 and 690. For the other five, the record does not support a statement of how many cases were exercised. This is the missing-context failure mode in [2] present in our own corpus. It does not affect the verdicts, which were reached with the evidence available at the time, and it does mean this paper cannot report coverage per criterion: the field its own Table 6 makes mandatory. The disclosure requirement was written partly because of this gap.

10.6 Withheld quantities limit replication

Per-criterion measured values and client thresholds are withheld (Section 3.3). A reader can recompute every statistic in section 4 from Table 1 and Table 3, and cannot audit any individual adjudication. That is a genuine limit on verification and it is a deliberate trade against client confidentiality. It should be weighed as such rather than read past.

10.7 The eighth assessment, and a report we withdrew

An eighth assessment from the same window is excluded as non-comparable: a developer-side code agent on a different framework with a criteria set that does not map onto the seven. Excluding it is a judgement call and readers are entitled to discount the corpus accordingly. The more relevant disclosure is that its earlier report was withdrawn by us, because it scored the agent against a release gate and then published that result as if it were a measurement of the agent's capability. Those are different quantities: a gate result says whether a bar was met, and says nothing about how capable the system is in general. That is a reporting error of exactly the kind catalogued in [2], committed by the author, and it is the reason section 8.2 fixes the claim to the mode rather than to the result.

10.8 Third-party figures and the single case

The three survey figures in section 1 are vendor-published, use different sampling frames, and only one states its n (2,527). They are directional. No argument here depends on them. The 214-of-214 case in section 6 is one observation from one assessment: it establishes that the mechanism exists and supports no claim about its frequency. Where this paper needs a frequency it does not have one, and says so.

10.10 What would falsify this

The paper's central claim is refutable, and it is worth stating how. It would be undone by a corpus of deployed agents in which a composite threshold with observations on both sides of it separated systems that passed all blocking criteria from systems that did not, held on systems not used to choose it, and did so across operators with different policies. Nothing in this work rules that out. It reports that no such threshold exists in these seven, that the only error-free cut here sits in an empty interval, and that the properties which failed are of a kind an average cannot represent.

11 Conclusion

Seven commissioned assessments of live production agents produced six blocks and one clearance. The blocked systems were not bad systems: five of six would have cleared a 0.80 bar and two a 0.85 bar, and their mean composite was 0.811. All 14 failures were categorical (adversarial resistance, task integrity, grounding) and in one case a required safety path was not invoked on any case that called for it while the system's composite stood at 0.820.

The composite is not useless. It correlates with failure count at $r = -0.83$, exact $p = 0.012$, and it ranks systems roughly correctly. What it cannot do is decide. Its strength is carried substantially by the single cleared observation; it weakens to non-significance when failure counts are normalised by the unequal criteria each system faced; its ordering inverts inside the band where real decisions are taken; and a full sweep of cut points shows the only error-free threshold on this corpus lies in an interval where nothing was observed and would clear one system in seven. A number that ranks well and decides badly is a summary, not a gate.

What decides is a set of categorical criteria taken from the operator's own policy, fixed before testing, exercised against the live build including the cases ordinary traffic never produces, asserted against components and not only outputs, adjudicated one at a time with unobtainable measurements treated as blocks, and bound to a statement that expires when the build changes. The 214-of-214 case shows why the component assertion is not optional: a system can fail completely in a part it depends on, answer every question fluently, and pass every check that inspects only what it said.

None of this is established at scale. Seven systems, one assessor, a corpus selected by clients who already suspected something, and coverage recoverable for two of seven. The contribution is a mechanism and a bound on a widely used shortcut, not a rate. If the argument is right, the instrument the field is missing is the one in the empty quadrant of Figure 2: an independent, expiring, criterion-level statement about a named deployed build.

12 References

1. Raji, I.D., Smart, A., White, R.N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., Barnes, P. *Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing*. FAccT 2020.
2. Staufer, L., Yang, M., Reuel, A., Casper, S. *Audit Cards: Contextualizing AI Evaluations*. arXiv:2504.13839, 2025.
3. Ghanem, M. C. *Who Audits Whom, on What Substrate, with What Evidence? An Independence-Graded Audit Protocol for Agentic AI*. arXiv:2609.18272, 2026.
4. Goodman, E.P., Trehu, J. *AI Audit-Washing and Accountability*. German Marshall Fund, 2022.
5. Birhane, A., Steed, R., Ojewale, V., Vecchione, B., Raji, I.D. *AI Auditing: The Broken Bus on the Road to AI Accountability*. arXiv:2401.14462.
6. Lam, K., et al. *A Framework for Assurance Audits of Algorithmic Systems*. FAccT 2024.
7. UK Department for Science, Innovation and Technology. *Trusted Third-Party AI Assurance Roadmap*. September 2025.
8. NIST. *AI Risk Management Framework 1.0*, and the published crosswalk to ISO/IEC 42001:2023.
9. Mitchell, M., et al. *Model Cards for Model Reporting*.
10. Gebru, T., et al. *Datasheets for Datasets*.
11. Fiddler AI, 2026. Vendor-published; directional.
12. Gravitee. *State of AI Agent Security*, 2026. Vendor-published; directional.
13. Sinch, $n = 2,527$. Vendor-published; directional.

Appendix A Reproduction

Every statistic in this paper is recomputable from Table 1 and Table 3 alone. Composite scores are the x vector, failure counts the y vector, criteria counts the denominators.

Permutation tests enumerate all $7! = 5,040$ orderings of y against a fixed x ; two-sided p is the share of orderings with

$$|r| \geq |r_{\text{obs}}|.$$

Intervals are Fisher z with $SE = 1/\sqrt{n-3}$, back-transformed.

TABLE A1 · EVERY DERIVED QUANTITY USED IN THIS PAPER

Quantity	Value	Section
Systems blocked / assessed	6 / 7	Section 4.1
Criteria failed / evaluated	14 / 41 (34%)	Section 4.1
Blocked composites: range, mean	0.680–0.861, 0.811	Section 4.1
Blocked at or above 0.80 / above 0.85	5 / 2	Section 4.1
Empty interval, and as a multiple of band spread	$0.126 = 2.25 \times 0.056$	Section 4.1
Pearson r , composite vs failure count	−0.83	Section 4.2
Exact permutation p , two-sided	0.012 (60 / 5,040)	Section 4.2
Orderings at least as negative as observed	24 / 5,040	Section 4.2
Most negative r any ordering can produce	−0.843	Section 4.2
Fisher z 95% interval on r	[−0.974, −0.206]	Section 4.2
Within-band r (A, B, D, G, E); spread	−0.82; 0.056	Section 4.2

TABLE A1 · EVERY DERIVED QUANTITY USED IN THIS PAPER

Quantity	Value	Section
Leave-one-out r, dropping any blocked system	−0.83 to −0.90	Section 4.3
Leave-one-out r, dropping F; interval	−0.62; [−0.952, +0.391]	Section 4.3
Pearson r, composite vs failure rate; exact p	−0.63; 0.12	Section 4.4
Per-system failure rates (C,A,B,D,G,E,F)	3/8, 3/6, 3/5, 2/5, 1/5, 2/6, 0/6	Section 4.4
Cut interval producing zero decision errors	0.862–0.987	Section 4.5
Systems cleared by that cut	1 of 7	Section 4.5
Failure classes: adversarial / task integrity / grounding	6 / 6 / 2	Section 4.6
Systems affected per class, in the same order	5 / 3 / 2	Section 4.6
Case counts recoverable	2 of 7 (149 and 690)	Section 5.2
Retrieval errors in the documented case	214 of 214	Section 6

Table A1: Correlations and intervals are shown to two decimals or three significant figures; all tests were run on unrounded values. No quantity anywhere in this paper is estimated, extrapolated or imputed.

DATA AVAILABILITY

Table 1, Table 3 and Table A1 contain the complete released dataset. Per-criterion measured values, client thresholds, transcripts and client identities are withheld under engagement confidentiality and are not available on request.

ANONYMISATION

Systems are identified by letter and sector only. Conversation text is redacted for identifiers before storage. No client, product, platform or model vendor is named anywhere in this paper.

AUTHOR

Pratham Chintrate designed the assessments, carried them out, performed the analysis and wrote the paper.

CONTRIBUTORS

Abilash Narahari and Megha Melgiri supported and reviewed the work reported here.

About QKTech

QKTech, formerly QualityKiosk Technologies, helps businesses win in the AI era by engineering reliability. Backed by experience in delivering reliable software, products, and platforms for leading global enterprises, QKTech combines deep domain knowledge with strong technology skills to help customers deliver great experiences to their own customers in an AI-first world.

Founded in 2000, QKTech offers reliability, cloud, AI, and product engineering services that help organizations build and adopt AI effectively, spanning product development, quality and reliability engineering, digital assurance, CloudOps and automation. At the core of this is the AI Reliability OS, an integrated portfolio of services covering AI for Reliability, Reliability of AI, Shift Right testing, Reliability of Frontier Systems, and Agentic Engineering. It brings together context engineering, harness engineering, loop engineering, observability, and platform-based delivery.

Operating in more than 25 countries with offices in Princeton and San Jose in the US, as well as Dubai, Singapore, Manila, Hong Kong and Jakarta, QKTech helps leading brands in banking, capital markets, insurance, automotive & manufacturing, technology & digital natives, telecom, airlines, consumer goods, and healthcare to transform their digital capabilities. The company is headquartered in Mumbai, India, with engineering centers across Bengaluru, Hyderabad and Pune.

The company's expertise and achievements have garnered recognition from esteemed global analyst firms such as Gartner, Forrester, and The Everest Group for innovative, IP-led solutions and the significant value of those solutions delivered to clients. QKTech has partnered with the most innovative and disruptive technology companies to force multiply outcomes that can be delivered to the customers.

For more information, visit www.qktech.com