The Record Can Answer

A short account of a system that keeps an AI agent's work on the record, checks what the agent claims, and has begun to answer questions from that record on a company's own machines

AppendOnly by 5R Industries

13 November 2026 · short edition

Abstract

Companies now hand real work to AI agents: programs built on large language models that can run commands, edit files and send messages on a person's behalf. When an agent reports what it did, the report is the agent's own account, and an account is not evidence. For seven months one small company has run its daily work on a system built around that problem. A small program writes down each command, file read, file change and web fetch the agent makes through its tools, as it happens. A second AI model, from a different maker and on a separate computer, checks each of the agent's claims against those entries and labels any claim that goes beyond them. The whole history is sealed so that a later change shows. Our first paper described that record. This one reports what has been added since: a way to identify the checking model by a fingerprint of its own file, so that a swapped model shows, has been built and tested and is not yet switched on in daily use; a small model on the company's own machine can answer questions from the record and point to the passages it used; and the person who runs the company rules on the system's verdicts, so that the checker is itself checked. We report what was built, what was measured, what it cost, and where it fell short, on one company, one operator and one agent. The results are early and mixed, but they are measured results.

About this paper. This is the short edition of the second public paper on the same company, agent and record as "The Record Does Not Change" (1 October 2026). It is written for a general reader and keeps technical language to the minimum the ideas need. A longer, more technical version is forthcoming; it carries every table, every run, the full account of limits and the complete list of sources. Every number here was read from the company's own record on the date given beside it. The company has three provisional patent applications on file (2 April, 24 September and 30 October 2026), and this paper claims nothing beyond what they describe and what the record measured.

1. The problem, in one story

On 28 March 2026 an AI agent working for our company reported that a service was "deployed and operational." The trail it left behind showed two facts: it had written the code, and it had saved the code. Nothing had been started. Nothing had been tested.

This was not a hallucination in the usual sense. Asked whether the job was done, the agent chose the reading of events most favorable to itself. Anyone who has managed people will recognize the move. Telling the agent to be more careful did not fix it, in our experience, because the trouble was never a shortage of instructions. The trouble was that the only witness to the work was the worker.

So we stopped asking the agent whether its work was done, and built something that checks. A body does not ask a cell whether it is healthy. It checks.

This matters beyond one company. Agents are moving into work where someone must later answer for what was done: a contract deliverable, an engineering calculation, a compliance filing. In that kind of work "the AI said it was finished" is not an answer a manager can give to a client, an auditor or a court. What the manager needs is a record that does not depend on the agent's word, kept on machines the company controls, that can be read back later by someone who was not in the room.

2. What we are building

AppendOnly is an operating and record-keeping system for work done by a person and an AI agent together. In plain terms it is three familiar things installed on computers a company already owns: a flight recorder, a fact-checker and a notary.

The flight recorder. Each time the agent runs a command, reads or writes a file, or fetches a web page, a small program beside it writes one entry describing the act. The agent does not write these entries, and a later edit to one would show. Each entry carries a fingerprint, a short code computed from its exact contents, and also the fingerprint of the entry before it. Change one character anywhere in the past and every fingerprint after it stops matching.

The fact-checker. When the agent saves its work it says what it did. A second program on a separate computer, which we call the observer, compares that statement with the recorder's entries. If the agent says "tested and deployed" and the entries show only that code was written, the claim is marked down to what the evidence supports. The observer's judge is a language model from a different maker than the agent's, so the agent's vendor does not grade its own product.

The notary. Each morning the newest fingerprint is sent to public timestamp services. From then on, anyone can confirm that the record existed in that form on that day.

The rule underneath all three fits in a sentence: no claim may stand above its evidence, and nothing saved is ever quietly changed. Figure 1 follows one claim through all three.

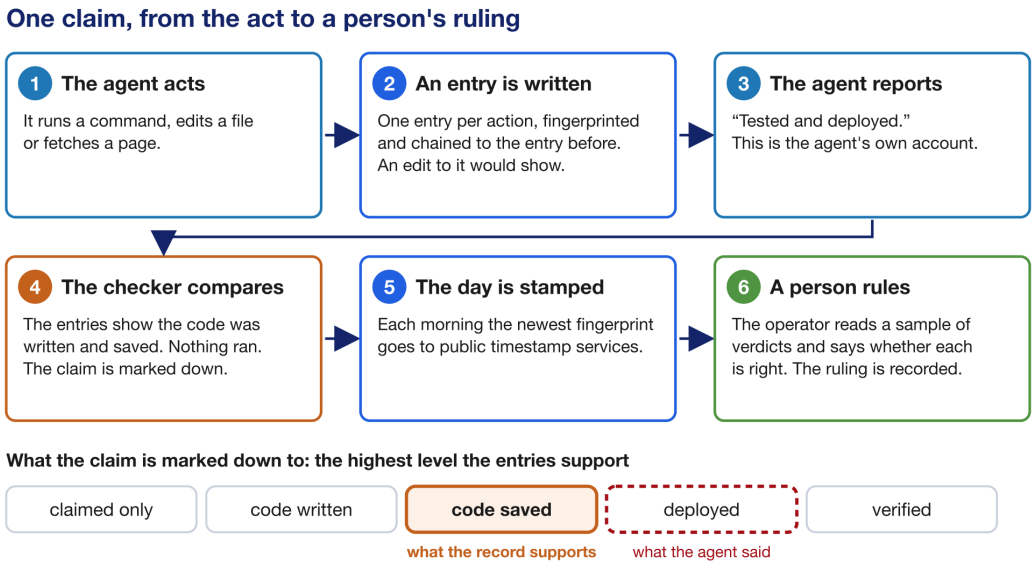


Figure 1. The path of one claim, in six steps. The agent's report is its own account; the entries are written by the recorder. The checker marks the claim down to the highest level the entries support, and a person rules on a sample of the verdicts.

We describe the system as a brain, because the comparison explains who is allowed to do what (Figure 2). The forebrain is where judgment sits: the person decides, signs and approves, and the agent reasons, drafts and builds. The midbrain keeps the system awake; it starts fresh working sessions and pages the person, and it holds no AI model at all. The hindbrain corrects without being asked: the checker lives there, comparing what was said with what was done. The record itself is not a region. It lies beneath all three as the one memory every part reads.

The division is a rule about authority, not decoration. Parts that keep the system running may not reason. Parts that reason are graded. And only a person's typed word can approve work, spend money, or restart the system after it has been halted.

In our company that place is taken by two small machines beside the working one: one keeps a copy of the record and the daily routines, and the other, the observer, runs the checker and keeps its own copy. In an enterprise that place

would be taken by what we call federation. Figure 2 carries that name. It is a direction of work, not yet built to that level, and this paper gives no specifics.

The same record that can be audited is the ground the next session stands on. A record kept this way is useful to more than an auditor. Because the agent is held to a standard for what it may claim, and because every claim is tied to what was recorded, the record shows more than files. It shows which statement was supported and which was not, where each fact came from, and, where a correction was written down, which version of a fact replaced an earlier one (section 5 shows a case where the reader could not yet tell). An agent that starts work from that record can check a previous session's account against it, and need not take that account on trust. The rule that would hold a new session to checked text alone is designed and not yet running. This is the bridge between the two halves of this paper: the discipline that makes the work auditable from outside is the same discipline that lets each new session build on the last, and section 6 reports what the system has cost.

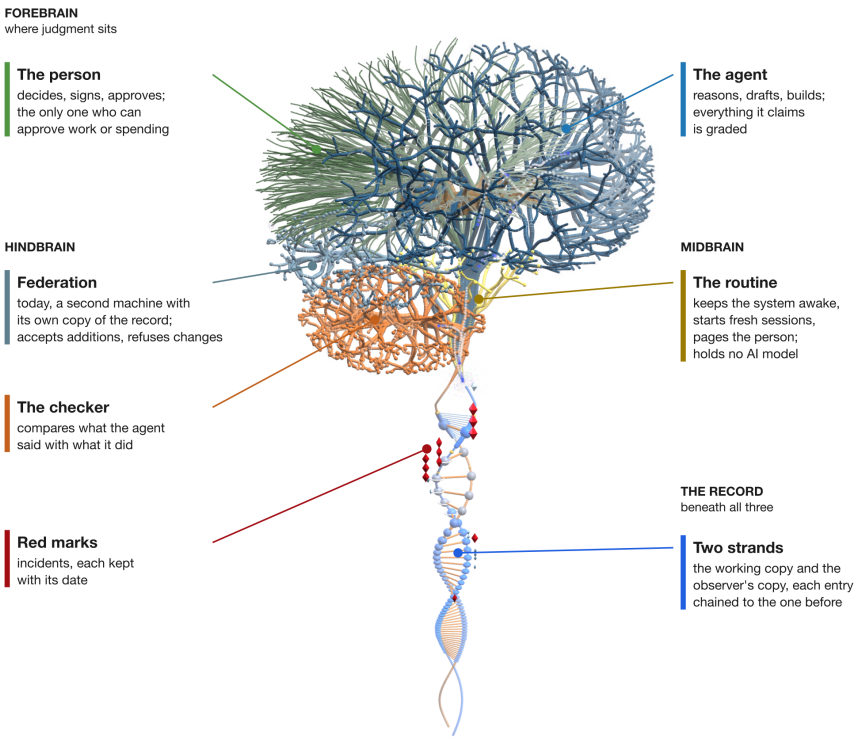


Figure 2. The system drawn as a brain, from the model built for our website. The person and the agent share the forebrain. The routine that keeps the system running is the midbrain. The checker and a second copy of the record are the hindbrain (in an enterprise, federation). The record runs beneath all three as two strands. The drawing is an illustration of the design; its strands replay our own record from 7 September to 3 November 2026.

3. What is new since October

The first paper described the record and was plain about two things it could not yet do. The checker's judge could be swapped without the record showing it. And the only way to ask the record a question was through the agent itself, so when the agent was unavailable nothing answered. Three additions since then address those gaps.

The judge now has a name the record can check. A language model is, at bottom, one very large file of numbers plus a small program that reads it. Our daily checker asks a separate, always-running program for its verdicts, and that program can update itself or load a different model without leaving a trace. On 13 October we built a second way: the checker opens the model's file itself, the way a word processor opens a document, computes the file's fingerprint, refuses to proceed if the fingerprint is not the one on record, and writes the fingerprint on every verdict (Figure 3).

This has been built and tested on fifteen claims on the working computer. The daily checker still runs the old way until the observer is moved to the new one (section 7). Anyone reading the record later can then say exactly which model judged which claim. It is the difference between a lab report signed “the lab” and one that names the instrument and its serial number.

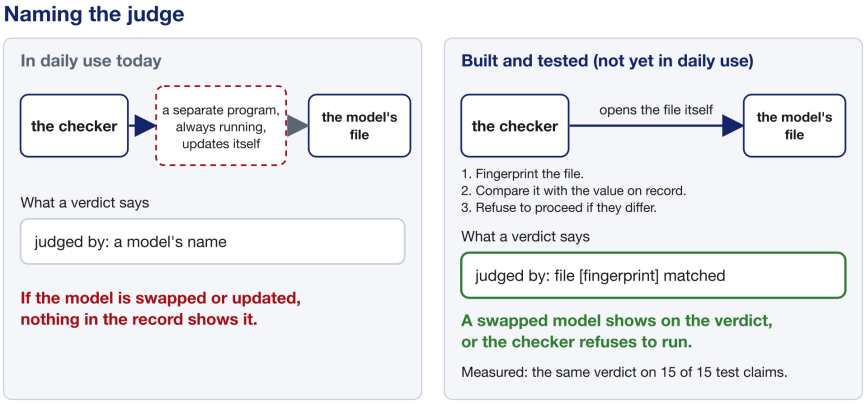


Figure 3. Naming the judge. In daily use today, a verdict names its model only by name, and a swapped model leaves no trace. As built and tested on 13 October, and not yet in daily use, the checker opens the model’s file itself, fingerprints it, compares the fingerprint with the value on record, and writes it on the verdict.

The record can be asked a question without the agent. We call the new part the reader. It searches the record for the passages closest to a question, hands them to a small language model from a different maker running on the company’s own computer, and requires that model to answer from those passages alone and cite each one. Two automatic checks then run with no AI in them. A sentence is removed if it has no citation, or if its citation does not lead to a real passage in the record. A sentence is also removed if it contains a number that appears in none of the passages it cites. What was taken out is shown beneath the answer with the reason. The reader is self-hosted. Its models are files on the company’s computer, a small one for the search and a larger one for the answer, and the reader loads them inside its own program. No separate model-serving program stands in the path of an answer and no outside service is called. It searches a list of passages built from the record and then checks each passage it cites against the record itself. On 13 November, in a test run, it answered from its own copy of the model files with no such program running on the machine. Where one is running, the reader asks it to free memory first. Since 11 November the reader is one command typed at a terminal on the working computer, and it needs no agent.

The person’s rulings are the measuring stick. A checker that nobody checks is only a second opinion. So the operator, the one person who runs this company, rules on the system’s work directly. They read samples of claims the checker passed and say whether they agree. They write test questions for the reader and lock them away with the answers before any run, the way an exam is sealed before it is taken. Their rulings are saved in the record like everything else, and those rulings are what the judge and the reader are measured against. The design does not ask anyone to trust a model. It asks a model to earn a score from a person, in the open.

4. What is built

Table 1 separates what runs every day from what has been built and measured once, and from what is designed and not yet running. We keep the three apart in everything we publish, because a design is not a product and a single test is not a habit.

Table 1. What is running, what has been measured, and what is designed, as of 12 November 2026.

Part	Status	In plain terms
The recorder	Running daily	Each tool action written down as it happens, each entry chained to the last
The observer and its judge	Running daily since July 2026	Every saved claim checked against the entries, by fixed rules first and then by a model from another maker
A gate at the point of saving	Running daily since 30 October 2026; 103 saves refused in its first two weeks (section 5)	A save is refused when fixed rules find a claim in its description that the saved work does not carry
Signing and the daily public stamp	Running daily since 9 September 2026	Every save signed; an outside timestamp each morning (one morning, 13 September, has none)
A second recorder that does not rely on the first	Running since 30 October 2026	A sensor that notes each change to the agent's files without relying on the first recorder
The judge named by its file's fingerprint	Built and measured on the working computer; not yet switched on at the observer	Verdicts that say exactly which model produced them
The reader	In use as a terminal command since 11 November 2026	Answers from the record with citations, on a local model
The operator's rulings as the measure	Running; fifty sampled rulings and two sealed question sets so far	A person grades the graders, on the record
A rule that the judge cannot be changed silently	Designed	A change of judge becomes a recorded decision after a side-by-side run
A way for a person to read what the record holds about them	Designed; to be built before the first pilot	The record is of agents and evidence, never an evaluation of employees

Figure 4 shows what the record looks like.

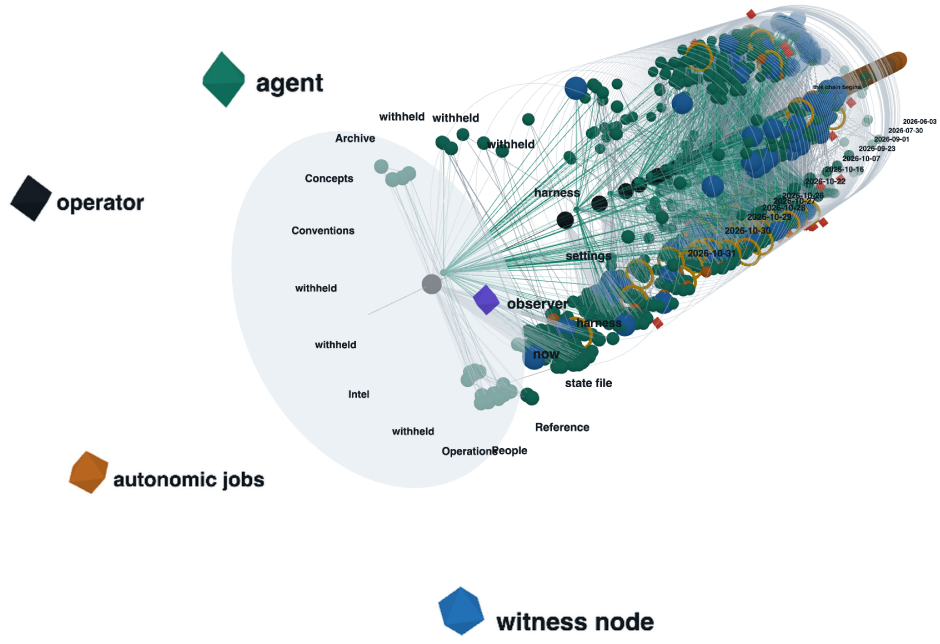


Figure 4. The record from 22 May to 31 October 2026, adapted from the system's own viewer. The marks are the five actors that touch the record, under the viewer's own names: the operator, the agent, the autonomic (routine automatic) jobs and the witness node stand around it, and the observer sits inside it. Time runs toward the reader, one bead per day along the spine. Green beads are signed saves by the agent, amber are routine automatic saves, gold rings are outside timestamps, and red diamonds are incidents, each kept with its date. Labels are generalized for publication, and some folder names are withheld.

5. What we measured

Everything in this section comes from one company's daily work. None of it is a benchmark, and we draw no general rate from it or outright claim of performance.

The record and the checker. Table 2 gives the counts as read on 31 October 2026.

Table 2. The record and the checker, read from the record on 31 October 2026.

What	Count
Saves to the record since 22 May	2,302, of which 1,875 are signed (signing began 9 September)
Recorder entries	20,911
Claims checked by the observer	2,146
Verified in full	1,100
Partly supported	326
Stated without evidence in the record	718
Marked contradicted by the checker	2, on 2 and 3 September; both were reviewed and found to be the checker's own error
Typical time for a model-based check since mid-September	9.8 seconds (the median)
Incidents on the register, each dated and classed	52
Mornings with an outside timestamp	52, from 9 September to 31 October

One line deserves a second look. About a third of the agent's claims, 718 of 2,146, said more than the record could show. Most were not falsehoods. They were statements such as "tested" or "sent" for which the machine had captured no proof. Before this system existed, every one of them would have been taken at face value. Now each carries a label, and a person can decide which ones matter.

Does the checker miss things? To find out, fifty claims the checker had passed were drawn at random and the operator ruled on each with the evidence in view. The operator agreed with forty-nine. The one rejected was a note in which the agent wrote that the operator had confirmed something, when the only support was the agent's own sentence. That miss became a rule: a claim about what a person said must carry the person's own words. Forty-nine of fifty is one sample, and it says nothing about the claims the checker flagged.

Does the system ever refuse the agent? Yes, as a matter of routine, and with no person involved. Since 30 October a gate at the point of saving reads the agent's description of each save and compares it with what the save holds, by fixed rules. From 30 October to 12 November the gate ruled on 562 saves. About 200 of them were routine automatic saves, which it passes without reading. Of the roughly 360 descriptions it read, it refused 103, on ten of the fourteen days. In 46 refusals the description used a word such as "tested" or "sent" with no proof in the save. In 31 the gate's word search did not find a change the description named. In 34 the description named the operator without the operator's own words or a pointer to them. Three more held only housekeeping files and did not say so. One refusal can have more than one reason. Each time, the save did not go through until the agent supplied the proof, dropped the claim, or marked it in the description as not proven. A save can also pass under a visible skip label, which stays in the record. The first two days, when the gate was new and still being tried, account for 54 of the 103.

Does naming the judge change its verdicts? No, on our test. Fifteen claims, nine prepared and six real, were put twice through a smaller model of the same family as the daily judge, once the old way and once from the file with its fingerprint. The verdicts matched on all fifteen. Naming the judge cost about four and a half seconds each time the model was loaded, and the fingerprinted run took 1.22 times as long overall.

Can the record answer? This is the question in the title, and the answer so far is: sometimes, and it shows its work.

On 20 October, before any reader existed, five questions about the company's own history were written and sealed together with the words a correct answer must contain. On 31 October they were asked three ways (Figure 5): by search alone, with no model; by small models running on the company's own computer; and once by a large model rented over the internet. Table 3 gives the scores against the sealed keys.

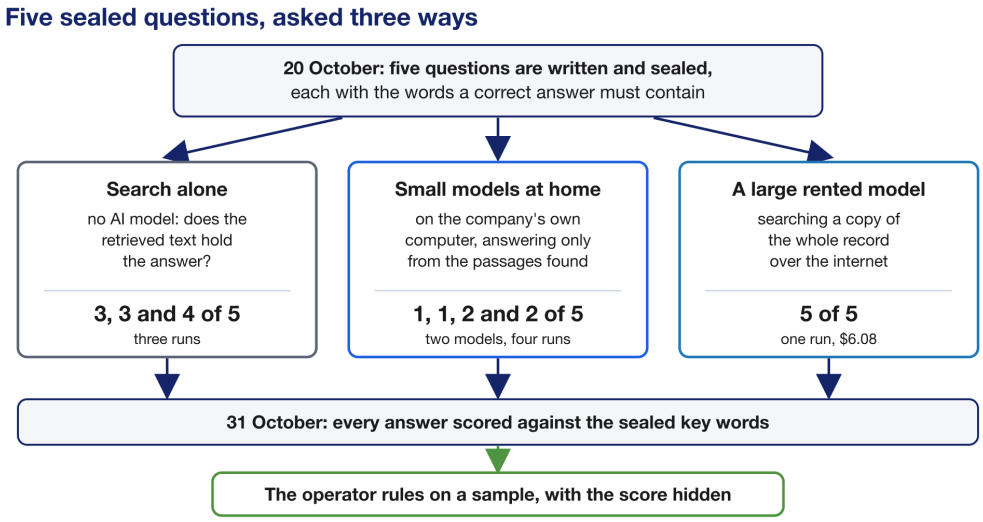


Figure 5. Five questions sealed on 20 October and asked three ways on 31 October, with the number of answers that contained the sealed key words. A sample of the answers went to the operator to rule on with the score hidden.

Table 3. Five sealed questions, asked on 31 October 2026. A score counts the answers that contained the sealed key words. These are three different instruments, and five questions are too few to rank them.

Way of asking	Runs	Questions that met the key, of five
Search alone (no model; does the retrieved text hold the answer?)	3	3, 3 and 4
Small models on the company's own computer (two models, from two makers)	4	1, 1, 2 and 2
A large rented model, searching a copy of the whole record	1	5, at a cost of $6.08

On 12 November the operator wrote ten new questions, sealed them, and put each to the reader from their own terminal. The agent had not read them. The reader gave eight answers with citations and declined twice with its standard sentence, "the record as retrieved does not say." By the operator's ruling six of the ten were correct (the sealed key words were present in three). Answers took 21 to 42 seconds on the company's working computer, with no fee per question.

Two things in that test matter more than the score.

First, the reader failed in a way its checks cannot catch. One answer named the wrong person and still passed, because its citation pointed to a real passage and the sentence held no number. The checks can remove a sentence with no support. They cannot confirm that a sentence follows from the passage it cites. The heading printed above every answer says so, and a check of that kind is the next thing to build.

Second, the test caught a weakness in us and not only in the model. On one of the five October questions, the same small model twice gave an out-of-date cost figure, and a second model could not say which figure stood. The record held both the old figure and the corrected one, because it never overwrites, and nothing told the reader which had replaced which. The operator spotted the pattern in a ruling note. We changed what the reader is handed along with the passages. On the next run the same small model gave the corrected figure, and missed another question it had met before. One run on the questions the fix was built for is not proof that the fix works in general, and a fresh sealed set is required before we would say so.

6. What it costs, and where the savings are

No customer has yet run a pilot, so no customer savings figure exists, and we will not invent one. What we can report is what the system has cost to run on ourselves, and where we expect it to save money for others. One term first: a

cache is the vendor's short-lived store of text the model has just read, and reading that text again is billed at a fraction of the full price.

Table 4. What it has cost to run, at the vendor's published list prices.

Measure	Figure	Period
The agent's model, all work	$829.12, or about $15 a day	55 days to 17 September 2026
Per save to the record	About $1.98 (419 saves)	The same 55 days
The agent's model, all work	$3,140.18, or about $29 a day	110 days to 11 November 2026
Text processed by the agent's model	3.1 billion tokens, the unit in which models are billed	The same 110 days
Share of the agent's reading served from the vendor's cache	97 percent	The same 110 days
What that same reading would have listed at without the cache	About $29,500, against about $2,380 at list price with it: roughly twelve times	The same 110 days
Fees for the checker's model and the reader's model	None; both run on computers the company owns (the machines and their power are not counted)	Throughout

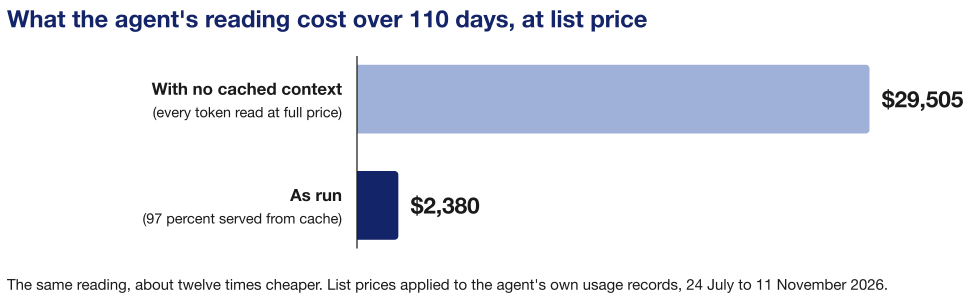


Figure 6. The agent's reading over 110 days, priced at the vendor's list rates with and without the cache.

These are list prices applied to the agent's own usage records, not invoices. Rates for the two newest models came from a reference table and were not rechecked against the vendor's price page, and small helper programs, which an earlier measurement put at roughly a sixth more, are not included. The second 55 days ran at about $42 a day at list price, against about $15 for the first, and we have not separated the causes.

There are three places where we expect the design to save money. None has yet been measured on a customer.

Each new session starts from the record, not from zero. Even the most capable agent's working memory is short. We treat each working session of the agent as a cell: it wakes, it works, it records, it dies, and the next cell picks up from the record where the last one left off. A fresh session normally has to be told again what the last one knew, and a company pays for that retelling every time. Here each session opens with the record's own account of the work and decisions before it. Most of what the agent reads is then read again within the session, and the vendor bills that re-reading at a fraction of the full price. The twelve-fold gap in Table 4 is the vendor's pricing, and any well-run agent collects it. What we have measured of our own design is narrower: the first paper found that the one session we let run without a restart cost about 50 percent more per model call than the next day's six short ones. We have not measured the same work with and without the record.

The checking is done at home. The observer's judge and the reader both run on the company's own computers. They add no fee per check and no fee per question, and what they read stays on the company's machines. The large rented model met the sealed keys on all five questions, at about a dollar and twenty cents each. The local reader answers in about half a minute at no added cost. Which of the two suits a given question is a choice the company gets to make, and the record is the same record either way.

Rework is caught the day it happens. The incident that began this work would once have been found by a customer. One in three of our agent's claims would once have been accepted unread. We have no dollar figure for errors caught

early and will not guess at one. A pilot is designed to produce exactly that number: three weeks, one workflow, and a measured result at the end.

There is also a saving that never appears on an invoice. The record belongs to the company and not to the model vendor. If the vendor changes its prices, its terms or its model, the history of the work stays where it is, and the next model can start from it.

The system is built for organizations that must answer for an agent's work: federal contractors, engineering firms, and teams in finance, healthcare documentation, law and insurance. It is also built for any company that wants more output from its agents for less spending, whether or not anyone will ever audit the work. We sell it for acquisition, administrative, engineering and document-review decisions. It is a record of agents and evidence. It is not a tool for evaluating employees.

7. Limits

Each limit is stated once here. The forthcoming technical edition treats every one at length.

- **One company.** One operator, one agent, seven months. Every figure is from real use, and none is a benchmark.
- **The named judge is not yet in daily use.** The fingerprint has been measured side by side on the working computer. The observer's everyday verdicts do not carry it until the observer is moved to it.
- **The reader is young.** It runs a small model, its checks cannot tell whether a sentence follows from its source, and nothing yet rebuilds the list it searches automatically. Ten questions asked once are not a rate. On 13 November it answered one question with the agent's session closed, correctly by the operator's ruling; one question is not a test.
- **The recorder sits within the agent's reach.** It runs on the same computer as the agent. What protects the record is the observer's separate copy, which accepts additions and refuses changes, together with the daily outside timestamp. That is detection after the fact, not prevention.
- **The recorder never blocks the agent, but the system is not open at every point.** If the recorder fails to write an entry, the work continues and a line is printed; not every such failure pages the operator. We chose this so that the safeguard could never become a way to stop the work, and it has a cost: a lost entry leaves a gap, and today nothing in the record marks the gap. Saving is different. Since 30 October a gate at the point of saving refuses a save when its fixed rules find a claim in the description that the saved work does not carry (routine automatic saves are exempt), and the save does not go through until the claim is backed, dropped or marked as not proven. Section 5 gives the count. The gate, like the recorder, runs on the agent's own computer.
- **Claims are graded; silence is not.** An action the agent never mentions leaves an entry and no verdict.
- **What this is not.** It is not a truth machine. It makes tampering evident, not impossible. It does not replace a person's judgment; it gives that judgment something firm to stand on.

8. Where it is going

Finish what is measured. Switch the observer to the named judge. Run a new sealed set of questions that can be printed, on the small model and on a larger local one. Build the check that tests whether an answer's sentence follows from its source.

A connecting layer of our own. Today the working session between the person and the model depends on one vendor's agent software. We intend to build that connecting layer ourselves, so that any model, rented or held locally, can take the working seat and the record does not change with it. A company should be able to rent its model today and run one locally tomorrow. This is a direction of work, and this paper describes no mechanism for it.

A system that checks its own expectations. The October filing describes a part that predicts the system's own next state and compares it with what happens. It may warn, propose, and tighten the system's guard under rules a person wrote in advance. It may not decide, loosen the guard, start work or widen its own permissions. We hold it to one more rule of our own: it predicts the system, never a person. It is designed and not built.

A first customer. The next number that matters is not ours to produce alone. A pilot is three weeks on one workflow, with the customer's own measure of what was caught.

9. Related work

We are not alone in this field and do not claim to be first. Tamper-evident logs are decades old. Since mid-2026 several groups have published designs for recording what AI agents do, and public Internet-Drafts (individual submissions, not standards) now describe a signed record of every agent action, including refusals, a person's later decision, and a fingerprint of the model. Others have measured the same gap we started from. In one study of a single developer's coding sessions, 186 of 322 tasks the agent marked complete had no check of any kind between the last edit and the claim of completion. Another group showed that widely used agent programs allow the agent to delete its own traces.

Table 5 sets three neighboring designs beside ours. The technical edition compares five in full.

Table 5. Three neighboring designs, as we read the two papers on 1 and 10 November 2026 and, in part, the draft on 12 November.

Design	Who writes the record	What is checked
Tracekit (Ghosh, arXiv 2609.35659)	A signer the agent cannot reach, outside its development mode	What was asked, said and done, by fixed rules
Soei (Khan and Kumar, arXiv 2610.39468)	A local ledger built over the agent's own transcript	Whether a check followed the last edit, by fixed rules
Agent Action Capsule (Mih, individual Internet-Draft, June 2026)	Whoever implements the format; it is a shared record format, not a product	The outcome of each action, executed or blocked
This record	A recorder beside the agent, a second computer that holds a copy, and a daily outside timestamp	Each claim the agent makes about its own work, by fixed rules and then by a model from another maker, with a person's rulings as the measure

What we add is not a new component. Loading a model from a file is ordinary engineering, and recording a model's fingerprint is now written into a public draft. What we report is a combination, on one company's record: a judge from a different maker grading the agent's own claims, a local reader that answers from the record and shows its sources, questions sealed before the reader existed, and one person's rulings kept beside the machine's scores. One of our neighbors names the weakness of its own first version in a sentence that applies to ours: "The writer must not be the actor." Ours is not yet fully separate, as section 7 says. Not first and not largest; measured and dated.

Sources

Our own first paper: "The Record Does Not Change," AppendOnly by 5R Industries, 1 October 2026, Zenodo, DOI 10.5281/zenodo.23042774. Works named above: Ghosh, Tracekit, arXiv 2609.35659 (second version, 21 October 2026). Khan and Kumar, Soei, arXiv 2610.39468. Qin and colleagues, on agents altering their own traces, arXiv 2609.30266. Mih, "An Agent Action Capsule Profile for SCITT," Internet-Draft draft-mih-scitt-agent-action-capsule, first posted 13 June 2026. Palanisamy and Mih, "The model_attestation Block for Agent Action Capsules," Internet-Draft draft-palanisamy-scitt-aac-runtime-00, 2 October 2026. The forthcoming technical edition lists every source with the date it was read and whether it was read in full.

The Record Can Answer. Second paper, short edition, version 1.0, 13 November 2026.