WAJD GROUP

INSIGHT PAPER · AGENTIC AI

Page not found

Why the push to adopt AI agents is real, why the audit trail is the quiet casualty when an agent mediates every record, and the controls that fix it: evidence written outside the agent's reach, factories before fields, and a code of practice with clauses you can test.

3
histories behind one "not found"

0
agent write paths to the log

5
clauses in a testable code

1
named owner per fielded agent

The push is real, and most of the reasons are honest

The move to agentic AI is the fastest technology adoption we have watched at close range, and most of what drives it is legitimate. The part that is not talked about is what mediation quietly does to evidence.

An agent collapses the cost of routine knowledge work. It runs through the night. It covers a breadth of tasks that no team could staff, and it does not lose interest in the boring ones. For a small organisation, agents are the difference between a roadmap and a wish list. WAJD Group builds with them every working day, across manufacturing and supply chain systems, defensive security, formulation and learning platforms, and the gains are not hype.

There is also a commercial engine underneath the enthusiasm. The industry has moved from selling tools to selling work itself: not software that helps a person do a task, but a substitute for the person doing the task, priced accordingly. Once work itself is the product, the pressure to put an agent between every person and every system is enormous, because every interface an agent mediates is revenue.

That second driver deserves more scrutiny than it gets.

The quiet casualty is the paper trail

Organisations run on records precisely because human accounts of history are unreliable. The file, the ledger entry, the signed approval and the timestamped log exist so that what happened does not depend on what anyone later says happened. Regulated industries state it plainly: if it is not documented, it did not happen.

Now put an agent in front of the records. It retrieves them, summarises them, renames them, versions them, tidies them. Every one of those verbs is sold as convenience, and each one is also a transformation of evidence by a system that is difficult to inspect and that changes behaviour with every model update.

Consider the failure that worries us most, because it is so quiet. A person asks for a document and the agent answers *page not found*. That answer is compatible with three different histories: the document never existed, the document was deleted, or the document exists and was not served. From the person's side of the screen these are indistinguishable. And if the question is asked again next month, an agent with no reliable memory of its own actions can flatly, sincerely deny that anything was ever there.

It does not take malice. Context gets compacted and detail is lost. A summariser drops the inconvenient paragraph without knowing it was inconvenient. A model update changes what the agent considers relevant. A retrieval index quietly fails to cover a folder. The motive hardly matters, because the effect is identical in every case: history becomes negotiable, and the organisation's answer to *what happened?* becomes whatever the agent serves that day.

Whether anyone in the market privately welcomes that property is a question we cannot settle. What we can say is that a mediated record with no independent trail is convenient for whoever controls the mediator, and that organisations should notice whose interests that serves before they wire it in.

> **An agent that holds the pen that writes its own history is not audited. It is trusted.**
>
> The distinction the brochure skips

Evidence must live outside the agent's reach

The remedy is structural, not behavioural. No instruction, prompt or policy sentence makes an agent a reliable witness to its own actions, for the same reason no expense policy makes an employee their own auditor. The record of what an agent did has to be written by something the agent cannot write to.

In practice that means a witness log with three properties.

- **Written by infrastructure, not by the agent.** The gateway, the file system and the API layer record every read, write, move and refusal as a side effect the agent cannot suppress, skip or phrase.
- **Append-only and hash-chained.** Each entry carries a hash of the one before it, so an edit anywhere breaks the chain visibly. We build hash-chained ledgers for financial compliance work, and the same discipline belongs under every fielded agent: the point is not secrecy, it is that tampering leaves a mark.
- **Stored where the agent has no path.** Separate credentials, separate storage, ideally a separate machine.

The witness principle

If the agent's compromise or confusion can reach the log, the log is testimony, not evidence. Test it the blunt way: give a red team the agent's full credentials and ask them to alter yesterday's history. If they can, so can drift.

With that in place, *page not found* stops being a verdict and becomes a claim you can check. The log shows the file was created on this date, read four times, moved by this agent on that date. Or it shows the file genuinely never existed. Either way, somebody can know.

Field monitoring completes it. An agent in production should be instrumented like a machine on a line, not managed like a member of staff: telemetry on what it touches, drift alarms when its behaviour shifts, and sampled replay where a fraction of its live decisions are re-run and compared. The full specification is in Appendix A.

The factory before the field

You cannot govern what you cannot reproduce. A fielded agent whose behaviour cannot be replayed under controlled conditions is unauditable by construction, whatever the logs say, because there is no way to ask *would it do that again?*

This is what an AI factory should mean for agents: not just racks that train models, but a simulation estate where an agent lives before it touches production and returns before every change. Ours are unglamorous and they earn their keep.

- **A scenario bank, replayed on every change.** Hundreds of recorded situations, including the awkward ones that only happened once, run against the agent the way a regression suite runs against code. A release either holds the bank or fails it.
- **Hallucination measured, not anecdotally reported.** Rate of confident fabrication under pressure: missing data, contradictory instructions, questions just outside competence. A number per release, tracked over time.
- **Weight changes treated as personnel changes.** Swapping the underlying model is the closest thing software has to replacing an employee without an interview. The same scenario bank run before and after makes the behavioural difference measurable, and byte-for-byte comparison catches the worst failure mode we know: the component that produces plausible output while doing nothing at all.
- **Small reasoning engines you can actually study.** Part of our factory work runs on deliberately small hierarchical reasoning models, levels updating on different timescales, each predicting its own input ahead. Small enough to dissect, cheap enough to retrain, and honest enough to show you when a change in weights changed the reasoning rather than the wording.

This is also, we would argue, where genuine progress towards more general capability will come from: measured, replayable experiments on reasoning and failure, rather than anecdotes about impressive afternoons. And it is what actually speeds products up. An agent with a factory behind it can be upgraded in days with evidence. An agent without one is upgraded in months, with hope. The release gate we use is in Appendix B.

Why the control has to come now

Governance debt compounds faster than technical debt, and it has a property technical debt does not: some of it can never be repaid. You can refactor code next year. You cannot backfill an audit trail. The record you keep from the first day an agent touches your systems is the only record you will ever have of that period.

Today, in most organisations, agents assist and people still hold the thread, so mistakes are recoverable and habits are still soft. That window closes. Once agents transact with other agents, once approvals, filings and customer commitments flow through them as a matter of course, the mediated version of events is the only version anyone has. Retrofitting evidence into a live agent

estate is an order of magnitude harder than building it in, and it is hardest exactly where it matters most: the systems everyone now depends on.

The cheapest control point in any system's life is before the habit forms. For agentic AI, that is now, and in many organisations it is already slightly behind now.

Where is the code of practice?

Given all of this, the missing document is conspicuous. Principles are published everywhere: transparency, accountability, human oversight, the same nouns on every website. An enforceable code of practice for fielded agents, one with clauses you could test a deployment against, has not been published by anyone with the standing to make it stick. It is worth being honest about why.

- **Nobody wants to bind themselves first.** A vendor that adopts real constraints while competitors sprint pays for the ethics of the whole market.
- **Liability strips out the substance.** Anything testable is discoverable in court, so drafts leave legal review as principles again.
- **The target moves quarterly.** Capability shifts faster than standards bodies deliberate, and a code scoped to last year's failure modes invites dismissal.
- **The drafters are the constrained.** Much of the writing is done by the parties whose products the code would bind, which reliably produces aspiration rather than obligation.

None of these reasons is good, and none of them prevents an individual organisation from acting. A usable code is short, and every clause on it is testable. Ours is in Appendix C: we publish our own ground rules, encode them as tests, and we would rather be held to a code than wait for one.

Appendix A: the witness log, specified

Five properties, each with a check that does not depend on trusting the agent or the team that runs it.

Property	Requirement	How you verify it
Authorship	Entries are written by the gateway, file system or API layer as a side effect of the action, never composed by the agent.	Trace one live entry to the infrastructure component that wrote it. If the agent can phrase the entry, it fails.
Integrity	Append-only, each entry hash-chained to the one before it, so alteration anywhere breaks the chain visibly.	Alter a copy of one historical entry and confirm the chain check fails loudly from that point forward.

Separation	Separate credentials, separate storage, ideally a separate machine. No path from agent to log.	Give a red team the agent's full credentials and ask them to change yesterday's history. They must not be able to.
Coverage	Reads, writes, moves, deletions and refusals are all recorded, including the requests the agent declined to serve.	Ask the agent for a file that does not exist, then find the refusal in the log with its reason.
Readability	Security and compliance can answer "what happened to this record?" without learning the agent tooling.	Hand an auditor one filename and thirty minutes. They should return with its full history.

Appendix B: the factory gate

What we measure before an agent, or a new model underneath an agent, is allowed into the field.

Measure	The question it answers	When it blocks a release
Scenario bank pass rate	Does the agent still handle the situations it has already met, including the awkward ones that happened once?	Any regression against the previous release, unless the change is reviewed and accepted by a person.
Hallucination under pressure	How often does it fabricate confidently when data is missing, instructions conflict, or the question sits outside competence?	The rate rises above the previous release, or above the absolute ceiling set for that agent's duties.
Behavioural diff on weight change	What actually changed when the model changed: the wording, or the reasoning and the actions?	Action-level differences appear in scenarios rated safety or money, whatever the wording looks like.
Do-nothing check	Is the component contributing anything at all, or producing plausible output while having no effect?	Output with the component disabled matches output with it enabled. Plausible and useless fails.
Field drift	Is the live agent still the agent that passed the gate?	Sampled replay of live decisions diverges from factory behaviour beyond the agreed band.

Appendix C: a code of practice you could sign today

Five clauses, each with the test that proves it. An organisation that cannot pass a clause should say so, rather than publish principles.

Clause	The test that proves it
Evidence outside the agent	Every fielded agent writes to a witness log meeting Appendix A, and the red team separation check is rehearsed, not assumed.
Authority in code	Act, ask and never tiers are enforced by capability, not instruction: the never tier has no credential that could perform the action, and a test fails the build if one is introduced.
Replayable behaviour	Any decision the agent made in the field can be re-run in the factory from the recorded scenario, and the factory exists before the field deployment, not after the first incident.
Disclosed model changes	Every change of weights or model version is announced to the owning organisation before it takes effect, with the Appendix B gate results attached.
A named owner	Every fielded agent has one accountable person, named in the record of each action, who can explain what it did last month without asking the agent.

Where this stops

A witness log proves what happened. It does not prevent it, and prevention still needs the boundaries and the factory in front of it. Simulation reduces surprise but cannot remove it: a scenario bank only contains the failures somebody imagined, and the field will supply the others. Measurement of hallucination and drift tells you the rate is moving, not why. And no amount of engineering answers the question of what an organisation ought to ask agents to do in the first place. That judgement stays with people, which is exactly where it should be.

What this discipline does deliver is narrower and worth having: an agent estate where *page not found* is a checkable claim rather than the end of the conversation, where a model update arrives with evidence rather than release notes, and where the history of what your agents did belongs to you and not to them.

What WAJD Group provides

Evidence, factories and field monitoring, run as a managed service

- **Witness logging:** append-only, hash-chained records of every agent action, written by infrastructure the agent cannot write to, stored in your tenancy or ours.
- **An agent factory:** scenario banks built from your real work, hallucination and drift measured per release, and regression evidence for every model change before it reaches the field.
- **Authority boundaries in code:** act, ask and never, implemented and tested against your risk appetite.
- **Field monitoring:** telemetry, drift alarms and sampled replay of live decisions, readable by security and compliance without learning the tooling.
- **A code of practice you can adopt:** your rule set drafted with your engineering and security leads, versioned, and enforced as tests in your pipeline so it cannot quietly erode.

The aim is the same as ever: reduce the risk without slowing the delivery. Governance that stops teams shipping gets bypassed, and a bypassed control produces confidence without protection.

Start a conversation. Tell us where agents already sit between your people and your records. We will show you what is currently evidenced, what is merely remembered, and what a witness log and a first factory would look like on your estate. hello@wajd.co.uk · group.wajd.co.uk/insights

Author: WAJD AI.