

You Can't Predict Good AI, So You Must Be Able to Validate It

If an AI cannot be predicted, regulate on a record of what it actually did, one nobody can rewrite without the change showing.

John Stroh · 11 October 2026



The assumption inside most calls for control

Most assume we can know in advance how an AI will behave.

- Certified
- Tested to a standard
- Aligned before release

Stephen Wolfram's work says that assumption fails for many systems doing serious computation. Large language models are among them.

Computational irreducibility

For many systems there is **no shortcut** to knowing what they will do.

You run them, step by step, and watch.

- Nobody can be sure what an irreducible AI will do (Wolfram, 2023)
- A test only covers the cases you tried
- No set of laws can rule out unintended consequences, and that is a matter of theory (Wolfram, 2016)
- Unless a program has been run on a case, you cannot know what it will do there (Wolfram, July 2026)

His advice: find ways to **see inside** what a program is doing.

It does not rest on Wolfram alone

- **Turing, 1936:** no general method can decide whether an arbitrary program will ever stop (the halting problem).
- **Rice, 1953:** any non-trivial property of the function a program computes is undecidable in general. “Will this system ever take an unauthorised action?” is of that family, for an unbounded agent.
- **The caveat:** a model answering one prompt, with bounded context and output, is a finite function, so the problem there is intractability, not impossibility.
- **Agents change that.** A system that loops, calls tools and acts on what it reads is running an unbounded computation.

The behaviour of an open-ended agent cannot, in general, be certified beforehand. **What can be checked afterwards is the integrity of the record of what it did.**

Two kinds of AI, and a gap between them

The neural network: fluent, statistical, not certain to be right.

Precise symbolic computation: the same correct answer every time.

Wolfram has long argued the two should work together.

Some questions about what the AI did are not statistical at all. Did this happen before that approval? Was the instruction signed by someone entitled to give it? Has this record changed since it was sealed?

Exact questions. They need a faithful record to ask them of.

Controls that settle behaviour before it acts

- **Pre-deployment certification** — certifies the system as tested
- **Evaluations and red-teaming** — a clean run says nothing certain about the next input
- **Interpretability** — often no simple human story of why
- **Bans and moratoria** — cover only what can be named in advance
- **Principles and guidance** — no AI-specific record to check against
- **Developer constitutions** — still a constraint on an irreducible system
- **Content provenance** — says where content came from, not whether an action was authorised

Each is partial. None can carry accountability on its own.

Only logging looks at what the AI actually did

EU AI Act Article 12: high-risk systems record events automatically, so their functioning can be traced.

But it says logs must exist. It does not say **who holds the clock, who can edit the entries, or how an outsider proves nothing changed.**

A log on the operator's server, timed by the operator's clock, is testimony from an interested party.

SSST: sovereign, signed, sealed, timestamped

Ordinary, published cryptography. Four steps.

1. **Sealed** — each event is hashed and chained to the one before
2. **Signed** — Ed25519, against a published `did:web` identity; the key can sit in a separate signing service, or as three shares of which any two sign
3. **Timestamped** — an independent RFC 3161 authority; the authority sees only a fingerprint
4. **Sovereign** — held in New Zealand or the EU, in the tenant's own boundary; checked with an offline verifier that sends nothing back

What the independent timestamp does

Three separate jobs.

- **Direction** — this event provably came before that one
- **Binding** — a commitment made at one time can be shown to predate the events it governed
- **Resolution** — a receipt is fixed to the authority's clock; where it did not answer in time, the receipt says so on its face

Compliance questions become checkable facts

- **Was a person in the loop before this was sent?** The approval's signed time precedes the send in the same chain.
- **What did the policy decide, and when?** The decision is in the signed record, with its attested time.
- **Has the log been altered since?** Every hash still matches its sealed day root, and the day root matches the signed, stamped estate root.

Anyone can recompute each one and get the same answer.

Five products issue these records today

- **Governance API** — signed verdicts, deliberation records, attestations, mandates and an agent inventory
- **AI Visibility Auditor** — what AI assistants say about you, checked against your own site, each measurement sealed and independently timestamped
- **Attested Accounts** — signs every invoice, quote, expense and time entry as it is written; a bundle checks offline
- **Email Decision Record** — decides whether an agent's email goes out before it goes, holds it for a named approver, seals the decision, approval and send
- **MIO° Conduct Record** — what an organisation's agents are authorised to do, with the mandate in force at any time

What a hyperscaler cannot build

The methods are standard and published. What a hyperscaler cannot supply is **independence from itself.**

- **Worth comes from who doesn't control it.** One company that trains, runs, stores and times is self-attestation.
- **Jurisdiction follows the company, not the data centre.** A US-run region still answers to US law.
- **Scale works against it.** SSST has each community hold its own records and leave with them intact.

Where this leads: small, local, federated

- **Safeguards in the record, not the model.** Runtime controls can be reasoned around. Substrate controls do not depend on the model's cooperation.
- **Watch with the other kind of computation.** Guardian Agents check responses against the community's own corpus by mathematical similarity, not another generative model.
- **Keep the model situated.** One community's context, on hardware it controls. In the Blueprint, our specification of this direction, every instruction, action and approval lands in a sealed record the community holds.

The Blueprint requires exit to be rehearsed every year, with the result published whether it passes or fails.

What this does not do

- It does not make an AI good, or its output true. A confident falsehood, sealed, is a well-documented falsehood.
- It cannot see what the rules allow. A mandate too wide fires nothing.
- Recording is not recognising: three AI reviewers caught every blatant breach and 58 per cent of the well-disguised ones.
- [What we claim](#) gives the row-by-row status against the standard.

When this AI acts, who holds the record, and who holds the clock?

If the company that runs it: accountability rests on its word.

If the community it serves, timed by authorities nobody in the dispute controls: a regulator, a court or a member can check for themselves.



Start with one message you would have to defend

If your organisation sends email that matters, whether a person writes it or an AI agent does, the [Email Decision Record](#) decides before it goes, holds it for a named approver when your policy says so, and seals the decision, the approval and the send with an independent time.

Pick one message you would have to defend to a member, a regulator or a court, and [check its exported record in your browser](#).

Invoices, expenses and time entries: [Attested Accounts](#).

We cannot predict good AI. We can make sure that what it did, and who approved it, is on a record that neither the AI nor its vendor can quietly rewrite.