16

THE STACK BEHIND THE AI COWORKER

The Dashboard Is Green. The Meaning Is Wrong.

Dr Peter McCann Strain, CTO and senior AI engineer, DPhil/PhD in AI from Oxford University

Your AI's output passed every check and is still false in the one relationship that matters.

An essay in the series **Architecting the AI Coworker**.
Approx. 18 minute read · Essay 16 of 22

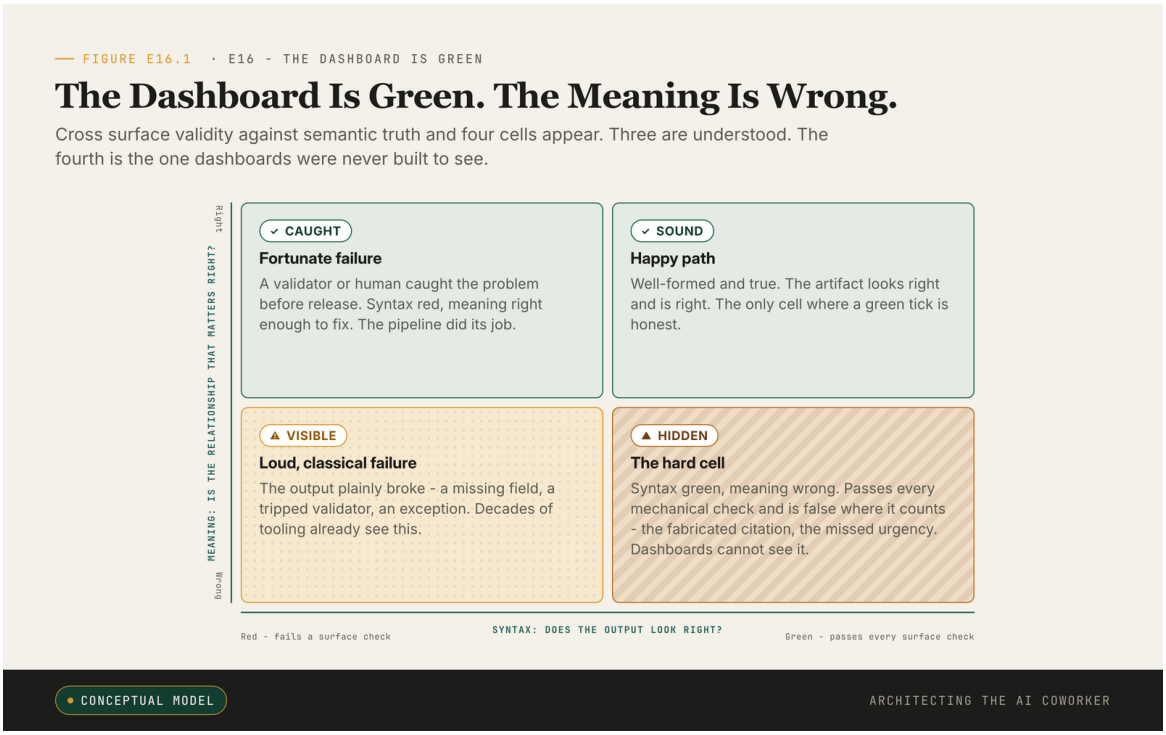
Dr Peter McCann Strain
CTO, DPhil/PhD in AI from Oxford University

> A brief drafted with an AI coworker arrives for filing at a mid-sized commercial firm. It is a beautiful document: crisp structure, measured tone, correct pagination, about forty footnotes numbered and formatted to the firm's house style. Three of those footnotes cite cases that do not exist. Another six cite real cases that argue the opposite of what the brief claims. The rest are, broadly, fine. The partner has not caught it. Her associate has not caught it. The firm's internal review tool, which checks formatting, citation style and a list of banned phrases, gives the document a clean green tick. The brief is almost filed.

It is caught, in the end, by a paralegal reading the cases out of habit while she eats her lunch, who looks up from her sandwich and says: this is not what this case says.

Hold on to that paralegal, because the whole problem is in her. Every checkpoint in the pipeline, the model, the workflow, the review tool, the partner, the associate, passed the document forward, and every one registered green. The only thing that caught the failure was a human reading for meaning, off-script, on her own time. The dashboard was green. The meaning was wrong.

That scene is a disclosed composite. The next section hands you the public-record version of exactly the same failure, in a federal-court order, and the two read as the same shape.



An output can pass every mechanical check and still be false where it matters. The four cells are: (top-left) syntax green, meaning right, the happy path; (top-right) syntax red, meaning right, caught by the workflow; (bottom-left) syntax red, meaning wrong, the loud classical failure; (bottom-right, terracotta) syntax green, meaning wrong, the hard cell.

The previous essay argued for reading the run: pull one complete trajectory and inspect it rather than trusting the aggregate score. This essay is about the failure that survives even a complete, well-formed trace. A run can show you every step in order, and the path can still mean something other than it appears to. I call this semantic failure, and it is the quiet failure of production AI. The output looks complete, formatted, successful. And it is wrong in the one relationship that matters: the citation does not support the claim, the symptom does not fit the triage, the requirement is not actually satisfied. Classical software failure is loud. A bad query throws an exception; a crash lights up a panel. Semantic failure makes no noise at all. The system succeeds visibly and fails silently, and the next agent in the chain receives the polished, plausible, incorrect artefact and treats it as ground truth.

When expertise itself files the fabricated citation

Here is that public-record version, and the cleanest one is a federal-court order.

In November 2024, Professor Jeff Hancock of Stanford filed a twelve-page declaration in *Kohls v. Ellison*, No. 0:24-cv-03754, a District of Minnesota case concerning that state's deepfake statute. Hancock had used GPT-4o while drafting the declaration, including `[cite]` placeholders the model was meant to fill in. The declaration that resulted contained citations to two

non-existent academic articles and incorrect authorship for a third. On 10 January 2025, Judge Laura M. Provinzino struck the declaration and denied leave to file an amended version as moot. The safer, court-supported reading is that the errors undermined the declaration's competence and credibility with the court [1].

Why does *Kohls* matter more here than the case the public already knows? *Mata v. Avianca*, the fabricated-citation case examined in depth later in this series, set the baseline: lawyers filed cases that did not exist and were sanctioned for it [2]. *Park v. Kim* showed the same pattern recurring after that warning had already landed: the Second Circuit referred counsel to its Grievance Panel after she cited a non-existent decision produced by ChatGPT and did not confirm it [3]. *Kohls* moves the failure somewhere more uncomfortable still. It moves it into expert evidence. The mistake was not confined to novice users or casual chatbot work. Hancock was a prominent scholar of digital deception, submitting sworn expert material in a case about AI-generated deception [1].

That is the point worth holding onto. Expertise did not save the artefact, because the failure was not in Hancock's knowledge. It was in a workflow that allowed plausible, unsupported text to become filed evidence without anyone resolving the text against the world.

The pattern is not US-only. In *Bandla v Solicitors Regulation Authority*, the English High Court refused a struck-off solicitor's appeal that rested on fabricated case authorities, which the appellant attributed to an unverified Google search; the court declined to make a factual finding on whether AI was used. Fordham J held that the court had to "take decisive action to protect the integrity of its processes against any citation of fake authority", and awarded indemnity costs against the appellant, the higher, claimant-favouring basis [8]. The dashboard was green at every checked layer the appellant relied on: the citations parsed, formatted correctly, and read like real law. The meaning, whether the cited authorities existed at all, was wrong.

One cell of the grid is the one your dashboards were never built to see

It helps to see semantic failure as a grid with two axes.

The horizontal axis asks: does the output look right? Valid format, expected fields, plausible prose, acceptable latency, no policy flag tripped. Call that the syntax of the output.

The vertical axis asks: is the meaning right? Does the claim resolve to the cited source, answer the user's real problem, preserve the requirement, respect the legal authority, fit the actual situation?

Cross those two axes and you get four cells, and the figure above lays them out. Three of the four are well understood. Where the syntax is red, the failure is visible: either the workflow has caught something before release, which is a fortunate failure, or the output has plainly broken, missing a field or tripping a validator, which is the loud, classical kind of software failure, and we have decades of tooling for it. Where the syntax is green and the meaning is right, you have the happy path: the artefact looks right and is right. The fourth cell is the dangerous one.

Syntax green and meaning wrong: the artefact looks right, passes every surface check, and is false in the relationship that actually matters. That is the cell our dashboards were, almost by design, never built to see, and it is the cell the Hancock declaration lives in.

Hancock's declaration is the worked example. Syntax green throughout: it rendered, the citations were correctly formatted. Meaning wrong: two of the articles did not exist and a third was misattributed. And it reached the court record before anyone caught it, which is the hard cell exactly, the failure that ships undetected. Surface validity was never evidentiary validity. A medical triage tool, an insurance eligibility decision, or a vulnerable-customer flag can all sit on the wrong side of that same gap without ever producing a legal filing.

Now take each monitor in your own AI workflow and ask honestly which cell it can see. Format validators see syntax red, and toxicity filters see policy red; latency and cost monitors see only slow and expensive. A citator, a tool that checks legal citations, can catch some authority failures, but only when it is actually wired into the gate that releases the claim, and a human expert sampling for meaning can see more. Most dashboards cannot see the hard cell at all.

Four mechanisms keep the hard cell open, and they reinforce each other

It is not one mechanism. It is four, and they reinforce each other.

Fluency without grounding produces text with the texture of authority but no underlying check against it. The legal cases make this visible: the deeper failure is not the hallucination but the workflow treating plausible citation strings as though they were authority [1][2][3]. That is the very thing the OWASP Top 10 for LLM Applications names as its misinformation risk class, overreliance on plausible model output [14].

Handoff loss leaks meaning at every boundary in a multi-step pipeline, where each step looks competent and the meaning thins out between them. It is the failure shape the peer-reviewed trajectory-attribution literature is built to measure, which is why those benchmarks now score whole runs rather than isolated turns; R-Judge alone scores LLM agents on safety risk awareness across 569 multi-turn agent interactions [12].

Output-only evaluation reviews the final answer and discards the causal evidence about where the run actually went wrong. The remedy is to read the trace and not the end-state, which is why failure-attribution accuracy improves measurably once an analysis is given the full execution path rather than a partial view ([5], preprint).

Syntactic monitoring catches shape and not meaning: a citation can be perfectly formatted and non-existent, a medical answer compassionate and missing urgency, a proposal elegant and quietly drop the two requirements the client actually cared about.

These four labels are my synthesis, but they are not floating coinages; each sits on a named risk class or a standing obligation. Fluency without grounding is the misinformation risk class above [14]; handoff loss and output-only evaluation are what the peer-reviewed trajectory-attribution work was built to catch [12][13], with two 2026 preprints [4][5] used only for their methodo-

logical claims and paired with those peer-reviewed counterparts; and syntactic monitoring is the gap the ongoing-monitoring duties later in this essay are written to refuse. What the table below adds is the gate-side reading: the column "what the dashboard draws" is what each measurement instrument captures when wired to its usual signal, and "what the fix needs" is what the same instrument has to capture to close the hard cell.

Table 1. Four mechanisms that keep the hard cell open (four rows, four columns). Each row reads as a separate failure pattern with its own symptom, what the dashboard tends to draw against it, and what is actually needed to close it. The four reinforce each other; treating one in isolation does not close the cell.

CAUSE	SYMPTOM IN THE ARTEFACT	WHAT THE DASHBOARD DRAWS	WHAT THE FIX NEEDS
Fluency without grounding	Plausible citations, formatted prose, no link to a real authority.	Parse rate, policy compliance, refusal rate, latency.	Resolve every load-bearing claim against an external corpus before release.
Handoff loss	Each step looks competent; meaning leaks at the boundary between steps.	Step-level success rate, per-tool error rate.	Trajectory-level evaluation that scores the path, not just the turn ([4], preprint; [12]).
Output-only evaluation	Final answer reviewed; intermediate work invisible.	Final-output acceptance rate, human approvals.	Inspect the trace, not the end-state, when locating where the run went wrong ([5], preprint; [13]).
Syntactic monitoring	Format checks pass; meaning checks were never wired in.	Schema validity, token cost, throughput.	A named semantic verifier owned by a named role, on the release path.

The figures from Who&When, a benchmark that tests whether an automated method can pin down which agent and which step caused a multi-agent failure, deserve their own beat, because they are the reason a verifier has to sit at the release gate rather than be trusted as forensics after the fact. The best automated method reaches only 53.5 percent at identifying the responsible agent and 14.2 percent at the decisive step [13]: even the strongest attribution available is wrong about *who* nearly half the time and wrong about *when* roughly six times in seven. That is precisely why the semantic check belongs upstream, preventing the failed artefact from shipping, not downstream trying to reconstruct which step betrayed the run.

What closes the hard cell is a semantic verifier: a check that resolves the output against the one relationship its domain actually depends on. The relationship is never generic, which is why a single dashboard cannot carry it.

In legal work the verifier asks whether the cited authority supports the claim it is attached to; in medicine, whether the response handles urgency and contraindications rather than merely reading as kind. The shape repeats wherever meaning hides under format: support has to ask whether the policy clause invoked matches the customer's actual entitlement, finance whether the transaction sits inside the client's mandate and authorised threshold, hiring whether the stated criterion was applied consistently and lawfully, procurement whether the proposal satisfies every must-have and not just the easy ones. Not one of those questions is answered by parse rate or latency.

Naming the relationship is the first move, but it is not yet an instrument. A verifier is only operational when someone other than the author can run it and get the same answer. Take the brief from the opening scene and build it in four steps. First, name the load-bearing relationship green is silently claiming: here, does each cited authority actually support the proposition it is attached to. Second, make it checkable by a second person: resolve each citation to a real reporter entry, then read the holding against the sentence it supports, recording pass, fail, or uncertain per footnote. Third, put the check on the release path, so the brief cannot be filed until the citation-resolution column is complete, the way the format validator already gates it. Fourth, assign it to a named role with budgeted time, not a lunch-break habit.

There is a single acceptance test for whether you have actually built one: hand the same artefact and the same relationship definition to a second person, and the verifier passes only if they reach the same pass-or-fail decision on the load-bearing claims without consulting the first. A check whose verdict is not reproducible is not a verifier; it is an opinion, and an opinion still leaves you in the hard cell. This is also the answer to the team that says it already has a reviewer: a reviewer whose verdict no one else can reproduce has not closed the cell.

Note the load-bearing qualifier, because it answers the obvious throughput objection. Resolving *every* claim would abolish the speed that made the AI worth using. The verifier does not recheck all forty footnotes equally; it checks the ones the conclusion rests on. The paralegal did not re-derive the whole brief; she read the cases that carried the argument. "Load-bearing" is the selection criterion that keeps the instrument cheap enough to actually run.

So does retrieval not just solve this?

A reader who builds these systems will have an answer ready, and it is not a weak one. Purpose-built legal-AI tools, the objection runs, materially reduce this failure surface, and they do. Three techniques carry most of that reduction, and they work in concert: retrieval fetches relevant source documents and puts them in front of the model, so its answer is grounded in real text rather than in memory; a closed corpus restricts the system to a fixed, vetted body of those documents instead of the open web, leaving far less room to ground an answer in something that does not exist; and a citator checks each legal citation, resolving it to a real authority and reporting how that authority has since been treated by later courts. Put together, a system that retrieves from a real legal database, stays inside a closed corpus, validates each citation, checks the quoted text, applies citator treatment, and refuses to link unsupported authorities is

plainly safer than a free-form model inventing citations from memory. The remedy is not to sneer at retrieval. It is to understand precisely what retrieval can and cannot prove.

Stanford RegLab's legal-AI reliability work is the reason for that restraint. On a set of challenging legal research queries, RegLab reports that two of the named tools, Lexis+ AI and Ask Practical Law AI, hallucinated on more than one query in six, a rate above 17 percent, and that Westlaw AI-Assisted Research did worse still, hallucinating on about one query in three, a rate of roughly 33 percent [6]. Those are the study's reported figures on hard queries, not blanket accuracy scores, and as of 2024 measurements; the named tools have shipped revisions since and the current figures should be re-checked against later studies before being repeated as a target. They do not mean the tools are useless. They mean something narrower and more important: closed corpora and retrieval suppress hallucination, they do not abolish semantic failure. Even the purpose-built tool, grounded in a real corpus, lands in the hard cell on a meaningful fraction of difficult questions.

A sharper version of the objection survives this. Those are 2024 figures on adversarially hard queries, the practitioner says, and on the routine ninety-five percent of work these tools run at something close to parse-rate reliability, so a verifier on every artefact is theatre. The reply is the whole point of the instrument. The verifier does not exist because the base rate is high; it exists because the hard cell is silent. You cannot tell, at release time, which artefact is the one in six: the failing brief and the passing brief look identical on every surface check, which is what "the dashboard is green" means. If you could distinguish them in advance, you would not need the verifier at all. Because you cannot, the check has to sit on every load-bearing claim rather than be sampled across the easy majority.

So retrieval narrows the hard cell; it does not close it. Hold that, and step out of law entirely, because semantic failure is not a legal peculiarity, and medicine shows the same shape with none of the courtroom furniture. Consider HealthBench, an evaluation OpenAI built with 262 physicians, assembling 5,000 multi-turn conversations scored against 48,562 physician-written rubric criteria [7]. *Disclosure: a co-author of HealthBench's rubric development is a member of the author's family.* Set aside the vendor source for a moment and notice why the design is built the way it is. If medical quality could be read off the surface of an answer, you would not need a quarter of a thousand doctors to write tens of thousands of criteria. You would need a grammar checker. The reason HealthBench is enormous is that a clinical answer can be warm, well-organised, and free of any banned phrase, and still miss the urgency that mattered, fail to ask the question that would have changed the triage, or reassure a patient who should have been sent to A and E. The cited authority not supporting the claim and the compassionate reply not catching the emergency are the same failure wearing different clothes. In law the wrong relationship is between a citation and a holding; in medicine it is between a response and a patient's actual clinical state. Neither relationship is visible to a format check, and that is precisely why the hard cell is a property of meaning rather than of any one profession.

Regulators are starting to write the same sentence in less colloquial language, and it is worth marking where, because a hard-cell argument lands harder when it is also someone's compliance duty. European AI law puts the posture into statute: providers of high-risk systems must run a documented monitoring system after launch, proportionate to the system's risks, so that the relationship between deployed behaviour and intended use is observed once the system is live, not only at a release gate [10]. The same expectation is written elsewhere in less binding form. The US NIST AI Risk Management Framework asks organisations to assess and track trustworthiness on an ongoing basis [9], and Canada's federal directive on automated decision-making builds post-deployment monitoring into the higher tiers of its impact assessment [11]. None of these writes "build a semantic verifier" in those words, yet each demands something so close that a deployer who cannot point to one will struggle in front of an auditor or a regulator.

So the green status has to change what it means. It cannot mean "true". It can only ever mean "this system has not detected a problem under the checks we chose to run". That sentence is colder than the one teams want to put on a slide. It is also the only honest one.

What I would stake the argument on is the legal paper trail: *Mata*, *Park*, and *Kohls* are court records, and all three show authoritative-looking material reaching a judge with fabricated or invalid support [1][2][3]. RegLab supplies the empirical check on the easy answer [6], HealthBench shows the same problem outside law [7], and the trajectory-level benchmarks explain why the cause so often lives across the run rather than in the final answer ([4], preprint; [5], preprint; [12]; [13]). The rest is still moving, and I would say so before anyone else does: the *Kohls* docket and the AI-citation cases after it will read differently as the record fills in, the RegLab figures already date from 2024 and may lag the current versions of the tools they tested, and HealthBench is a benchmark, not a clinic, so I would not let a vendor evaluation stand in for proof of medical safety. The claim survives that caveat because it never rested on any single figure.

Once meaning can fail this quietly, the trace stops being a guarantee and becomes a case file: necessary, but not the truth, and only as good as the judgement applied to it. The evidence objects start to separate, the system's self-report, the trace, the verifier's judgement, the human's judgement, and external reality, and they no longer agree by default. The green dashboard told you nothing failed; it could never tell you anything about meaning at all.

So come back, finally, to the paralegal with her sandwich. She is the most important figure in this essay, and also the most fragile. Nothing in that firm's pipeline was designed to rely on her. She was not a control; she was a habit, an unbudgeted, unrostered, easily-cancelled habit, and on a busier day she would have eaten at her desk and the brief would have been filed. The lesson is not that every team should hope for a conscientious paralegal. It is the opposite. The semantic check she performed, reading the cases against the claim, was real engineering work, and a serious system names it, resources it, and puts it on the release path on purpose. The question to ask of your own workflow is not whether the dashboard is green. It is whether the only thing standing between green and wrong is someone reading for meaning on her

lunch break, and whether you could name that person if I asked. Before the next release goes out under a green tick: which hard cell can your dashboard see?

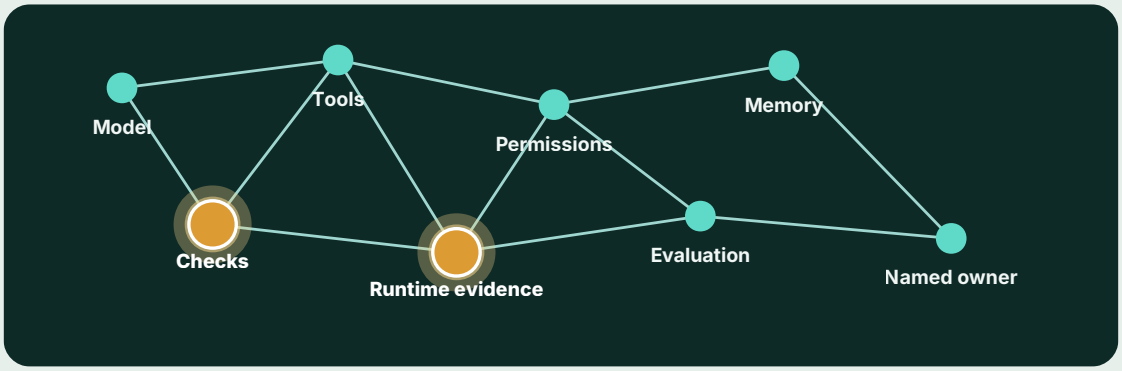
> **Carry This Forward.** Take one dashboard or monitor and ask which cell of the grid it can actually see: format, latency, cost, policy, the source relationship, domain truth, or release consequence. If it cannot see meaning, do not let it certify meaning. Now name the one relationship that, in your domain, green is silently claiming to have verified, and name the human or tool that checks it on the release path. If you cannot name either, you have found the gap before the gap finds you.
>
> Next, E17 turns from alerts to witnesses, because once the trace is only a case file, the harder question is no longer whether anything failed but who acted.

THE STACK SO FAR

E16 · Essay 16 of 22 complete · Arc IV: Proof and accountability

The Stack So Far. *Every essay adds one instrument to the operating model. The constellation shows which eight you are building, which are lit by essays you have read, and which is added right here.*

I See the object | II Evidence and authority | III Runtime control | **IV Proof and accountability** ESSAY 2 OF 5 | V Operating model



built in earlier essays | added in this essay | coming in later essays

01 02 03 04 05 | 06 07 08 09 10 | 11 12 13 14 | 15 **16** 17 18 19 | 20 21 22

You have just added.

The semantic grid

You can now identify the hard cell: syntax green, meaning wrong.

Next. E17 asks how to weigh three different kinds of evidence about a run.

← PREVIOUS
E15 · Show Me the Run

Essay 16 of 22 complete

NEXT →
E17 · The Three Witnesses to a Run

References

Reference links for sources cited in this essay.

1. **Kohls v. Ellison order re Hancock declaration**
   U.S. District Court, D. Minn.
   https://law.justia.com/cases/federal/district-courts/minnesota/mndce/0:2024cv03754/220348/46/

2. **Mata v. Avianca sanctions order**
   U.S. District Court, S.D.N.Y.
   https://www.nhd.uscourts.gov/sites/default/files/pdf/Mata-v-Avianca-sanctions-order.PDF

3. **Park v. Kim, 91 F.4th 610**
   U.S. Court of Appeals, Second Circuit
   https://law.justia.com/cases/federal/appellate-courts/ca2/22-2057/22-2057-2024-01-30.html

4. **ATBench: A Diverse and Realistic Agent Trajectory Benchmark for Safety Evaluation and Diagnosis**
   Li et al.
   https://arxiv.org/abs/2604.02022

5. **Seeing the Whole Elephant: A Benchmark for Failure Attribution in LLM-based Multi-Agent Systems**
   Chen et al.
   https://arxiv.org/abs/2604.22708

6. **Hallucination-Free? Assessing reliability of legal research tools**
   Stanford RegLab
   https://reglab.stanford.edu/publications/hallucination-free-assessing-the-reliability-of-leading-ai-legal-research-tools/

7. **OpenAI HealthBench**
   OpenAI
   https://openai.com/index/healthbench/

8. **Bandla v Solicitors Regulation Authority [2025] EWHC 1167 (Admin)**
   High Court of Justice (Administrative Court)
   https://www.bailii.org/cgi-bin/format.cgi?doc=/ew/cases/EWHC/Admin/2025/1167.html

9. **AI Risk Management Framework (AI RMF 1.0), Measure function**
   National Institute of Standards and Technology
   https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf

10 **Regulation (EU) 2024/1689 (EU AI Act), Article 72: post-market monitoring**
European Parliament and Council
https://eur-lex.europa.eu/eli/reg/2024/1689/oj

11 **Directive on Automated Decision-Making and the Algorithmic Impact Assessment tool**
Treasury Board of Canada Secretariat
https://www.tbs-sct.canada.ca/pol/doc-eng.aspx?id=32592

12 **R-Judge: Benchmarking Safety Risk Awareness for LLM Agents**
Yuan et al.
https://aclanthology.org/2024.findings-emnlp.79/

13 **Which Agent Causes Task Failures and When? On Automated Failure Attribution of LLM Multi-Agent Systems**
Zhang et al.
https://openreview.net/forum?id=GazlTYxZss

14 **OWASP Top 10 for Large Language Model Applications 2025: LLM09:2025 Misinformation**
OWASP Foundation
https://genai.owasp.org/llmrisk/llm092025-misinformation/

About the Author

ARCHITECTING THE AI COWORKER

Dr Peter McCann Strain

Dr Peter McCann Strain is a CTO, founder, and senior AI engineer with a DPhil/PhD in AI from Oxford University. He builds production AI systems and writes about making agentic AI useful, inspectable, governable, and safe enough for real work.

READ THE FULL SERIES

Substack (canonical)	petermccannstrain.substack.com
Medium	@peter.mccann.strain
LinkedIn	peter-strain-dphil-15a607128
Web	petermccannstrain.com
Cadence	New essays twice weekly, 2 June – 21 July 2026