SolvxAI
Data to Decision Powered by AI

SOLVXAI WHITE PAPER SERIES | NO. 01 | DOMAIN ENGINEERING

The Model Is Not the Product

Why general-purpose AI agents stop at the wellhead, and what domain harness engineering actually is

Published by the SolvxAI Research Team
July 2026 · Edition 1.0
Contact: sd@nexaconsultinc.com

IN BRIEF

Every evaluation of AI for upstream engineering now opens with the same question: we already have a general-purpose AI assistant, so why do we need a specialised one? This paper answers it without disputing the premise. The general agents are capable, they improve continuously, and SolvxAI runs the same frontier models beneath its own system. If the model were the product, there would be no argument to make. Drawing on published benchmarks of model behaviour on real engineering problems, a peer-reviewed study of what actually improves calculation accuracy, and the reserves-evaluation standards the industry already enforces, we set out why the decisive engineering happens outside the model, and what it costs to build.

Contents

1. Executive summary

The question is fair, and the honest answer starts with a concession.

General-purpose AI agents are genuinely capable. They read, plan, write code, call tools, and improve every few months on a schedule no single vendor controls. SolvxAI runs the same frontier models. We do not claim a better model, and any vendor who does is describing a lead measured in weeks.

What determines whether an engineering answer can be trusted, defended in an audit, and acted on with capital lives almost entirely outside the model. Whether the number came from a tested implementation of the published literature or from code the model improvised. Whether the analysis ran in the order the field requires. Whether the system recognises that the method does not apply, and says so. Whether every figure carries the method, the units, the validity band, and the source that produced it.

That surrounding system is the harness, and it is where the engineering actually happens. Three findings from the published record frame the case:

- On a contamination-resistant engineering benchmark spanning nine domains, the best of twenty-seven models reached **65.4% final-answer accuracy but only 42.7% reasoning-trace fidelity**, and **79.5% of frontier-model errors were arithmetic** rather than conceptual. A correct answer does not certify the derivation that produced it. [1]
- A published six-agent system built on the now-standard pattern improved reservoir history matching by **95% on a toy case, 69% on a moderate one, and 13% on a real field**. Performance collapses precisely as problems become real. [6]
- In a peer-reviewed study running 10,000 trials in a safety-critical discipline, equipping a model with **task-specific deterministic tools moved accuracy from 11% to 84% and from 36% to 95%**, and those purpose-built tools outperformed both retrieval augmentation and a general code interpreter. [7]

The third finding is the argument of this paper. The gains did not come from a better model. They came from what was built around it.

THE INDUSTRY ALREADY SETTLED THIS, IN 2019

The Society of Petroleum Engineers' auditing standards instruct evaluators to “exercise caution in accepting results produced through use of proprietary or commercial software without a full understanding of the internal mathematical algorithms and correlations,” and list among the required competencies of a qualified reserves evaluator an understanding of “the consequences of reliance on computer software without a full understanding of the internal calculation processes.” The objection to opaque computation is not a reaction to language models. It was codified years before they existed, and it applies to them exactly as written. [10]

2. The question every evaluation now opens with

Two years ago the question was whether AI could contribute to technical work at all. That question is settled. The current question is narrower and much harder to answer with a demonstration: *if a general-purpose agent can already read our documents, write code, and reason through a problem, what does a specialised system add?*

It deserves a direct answer rather than a feature list, because the people asking it are usually right about the premise. They have watched a general agent do something genuinely impressive with unfamiliar material. The reasonable inference is that the remaining gap is one of data access and prompting.

The inference is wrong, but not for the reason vendors usually give. The gap is not that the model lacks petroleum knowledge; frontier models have read more petroleum engineering literature than any individual engineer. The gap is structural, and it concerns *where the computation happens* and *what survives afterwards*.

3. What the published record shows

The evidence below is drawn from independent benchmarks and peer-reviewed studies, not from vendor testing. It is consistent across research groups and across model families.

3.1 Arithmetic is the dominant failure mode, and the derivation fails before the answer does

EngTrace, a symbolic benchmark designed to resist training-data contamination, evaluated twenty-seven models across 1,350 instances spanning three engineering branches and nine domains. The best model reached 65.4% final-answer accuracy but only 42.7% fidelity on the reasoning trace, and the authors report that models “arrive at correct answers through derivations that diverge substantially from expert solutions.” Of frontier-model errors, 79.5% were arithmetic and only 14.5% conceptual. [1]

For technical work generally this is uncomfortable. For reserves and reservoir engineering it is disqualifying, because **the audit is of the method, not the answer**. A figure that happens to be right, produced by a derivation an evaluator would reject, is not a defensible result. It is a coincidence with a decimal point.

3.2 Models fabricate computation rather than concede they cannot compute

LinAlg-Bench evaluated 6,600 model outputs against symbolically certified solutions and classified 1,156 failures. Beyond a modest problem scale, models exhibited what the authors term *computational abandonment*: rather than attempting the calculation, they produced “tool roleplay, constraint-consistent confabulation, and structured hallucination.” The behaviour was near-universal across model tiers. [2]

Tool roleplay deserves emphasis because it defeats the most common mitigation. A model that has been given a calculator may narrate using the calculator, present a plausible intermediate result, and never

invoke it. **Tool availability is not tool use.** Execution has to be recorded as a fact about the run, not inferred from the model's account of itself.

> **A correct answer does not certify the derivation. In reserves work, the derivation is the deliverable.**

3.3 Non-determinism is a compliance problem, not a philosophical one

FEM-Bench evaluated graduate-level computational mechanics tasks with five attempts per task. A leading model solved 30 of 33 tasks at least once, but only 26 of 33 on every attempt. [3] The distinction is not academic. United States securities regulation defines the reliable technology admissible for reserves estimation as methods demonstrated to give “reasonably certain results with consistency and repeatability.” [11]

A system whose answer depends on which attempt you observed is in direct tension with that definition, however impressive its average performance.

3.4 The failure is computation, not comprehension

A 2026 study of multimodal models found accuracy on multiplication tasks “often nearing zero” past a computational-load threshold while perception accuracy remained above 99% across modalities. [4] The model reads the log, the table, and the plot correctly. It then computes badly. This is precisely the profile that makes unaided model output dangerous in engineering: the inputs are ingested faithfully, so the output looks well-founded.

3.5 And the model does not know when it is out of its depth

Calibration studies across frontier models report root-mean-square calibration errors of 82–94% in low-accuracy regimes, with models “frequently provid[ing] incorrect answers with high confidence… failing to recognize when questions exceed their capabilities.” [5] An engineer who does not know a method is inapplicable is a hazard. A system with the same blind spot, operating faster and with more apparent authority, is a larger one.

3.6 The result: performance collapses exactly where the work is real

PetroGraph, published in 2026, is the most directly relevant study available. Its authors built a six-agent system for reservoir history matching, on the architecture the market now describes as agentic: an orchestrating planner, specialist agents, retrieval over simulator documentation, a non-model simulator in the loop, and human checkpoints. It is a competent, honest piece of work.

The results are the point. Weighted normalised error improved by 95% on a toy benchmark, 69% on a moderate one, and **13% on a real field**. [6]

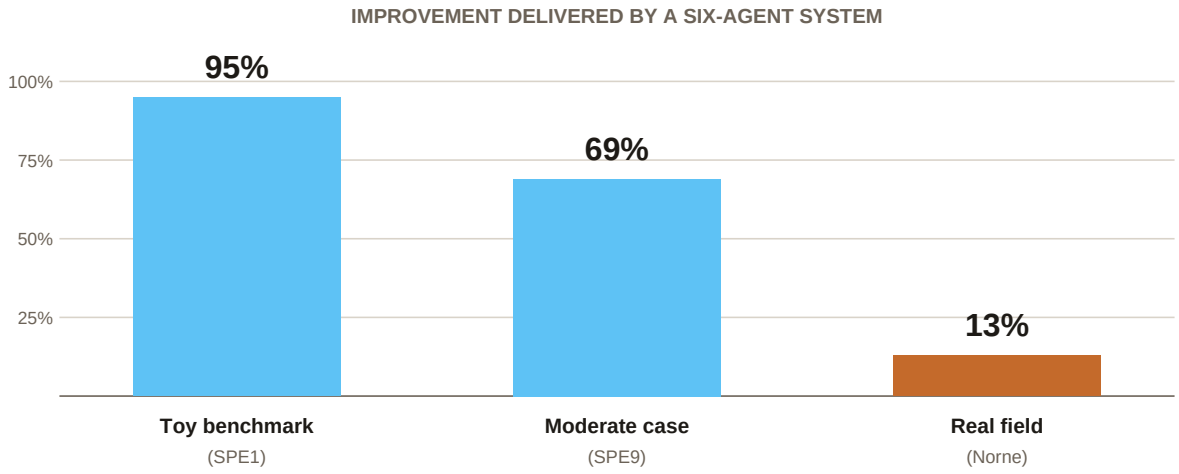

Figure 1. Reported improvement from a published six-agent reservoir history-matching system, by problem realism. Source: PetroGraph, 2026 [6]. The architecture is sound and publicly reproducible; the degradation is structural rather than a defect of that implementation.

This is not a criticism of the authors, who published the unflattering result rather than the flattering one. It is the clearest available evidence that wrapping capable models in a competent multi-agent structure does not, by itself, survive contact with a real asset.

4. The diagnosis: one structural mistake

In each of these systems, the model is being asked to do the engineering.

Ask a general-purpose agent for a material balance and it will improvise: write code, run it, and report a number. That code is unreviewed, unversioned, and quite possibly different next Tuesday. The reasoning and the computation are the same act, performed by the same probabilistic process, and neither is separable afterwards for inspection.

Petroleum engineering requires the opposite arrangement. The computation must be a fixed, tested, citable artefact that behaves identically every time it is invoked. The reasoning must be about *which* computation applies, whether its preconditions are satisfied, what the inputs mean, and whether the result is credible in context. Those are different jobs, and they call for different machinery.

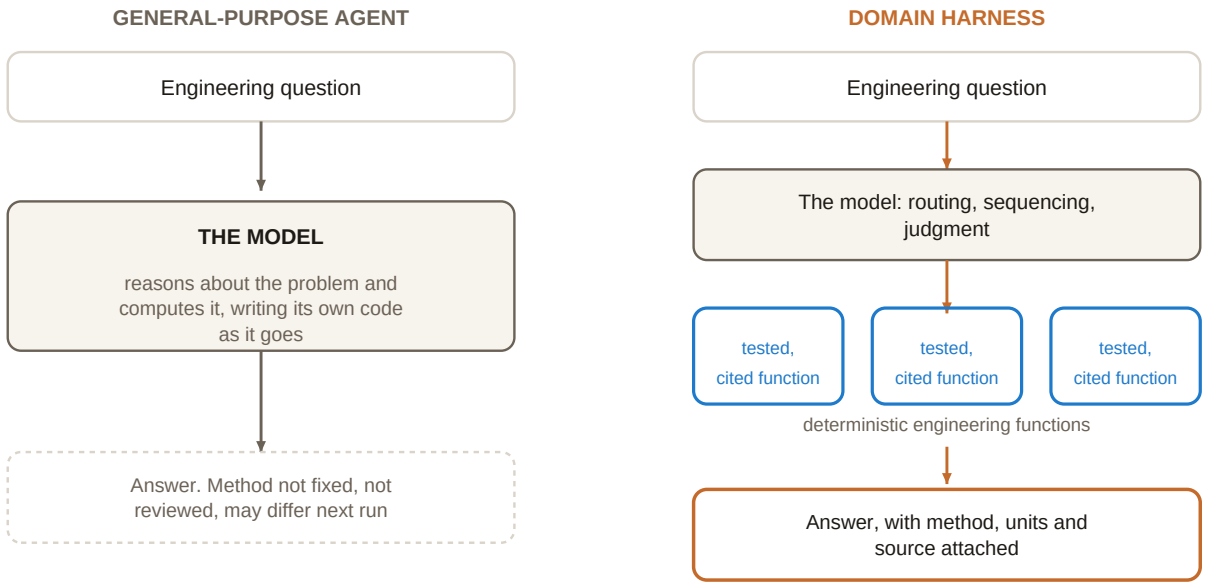

Figure 2. Where the computation happens. On the left, the model both reasons and computes, improvising its method. On the right, the model does what it is genuinely excellent at — understanding intent, selecting method, sequencing work, and judging results — while the numbers come from tested, cited engineering functions that behave identically on every run.

This is the whole distinction, and it explains every benchmark in Section 3. Arithmetic errors dominate because the model is doing arithmetic. Derivations diverge because the derivation is improvised. Answers vary between runs because the method varies between runs. Move the computation out of the model and that entire class of failure moves with it.

5. What a domain harness is

“Harness” is an unglamorous word for the part of the system that decides whether the output is worth anything. It has seven elements. None of them is a model, and none is obtainable by prompting.

5.1 Computation that is fixed, tested, and cited

Every number comes from an implementation of the published engineering literature that was written once, tested, versioned, and cited — not generated at the moment of asking. Identical inputs give an identical answer today, next quarter, and after the underlying model has been replaced twice. This is what makes a result reproducible in the sense a regulator means. [11]

5.2 The order the field actually works in

Engineering outputs are inputs to other engineering. Net pay must be counted before volumetrics mean anything. A fracture design needs the fluid efficiency that only a calibration injection test provides. A lift design serves an operating point that cannot be established without fluid characterisation. A domain harness carries that dependency structure explicitly, resolves each requirement in the cheapest correct way — reusing a value you already have before recomputing anything — and does not silently produce a downstream result from a missing upstream one.

5.3 Applicability limits, and the willingness to refuse

Every empirical method has a range in which it means something. Below a threshold of depletion, a material-balance solution is not merely uncertain, it is non-unique. A correlation fitted on one fluid system and pressure range does not transfer silently to another. Where several models match a pressure response equally well, naming one as the answer is a choice being presented as a finding.

A domain harness knows those boundaries and declines to cross them — returning the range, naming the competing interpretations, or refusing the estimate and explaining why. A general-purpose agent has no such notion. It will always answer, because answering is what it is for.

5.4 Units and basis, treated as a failure mode

The most dangerous error in petroleum computation is not a wrong formula but a right formula fed a value in the wrong unit or on the wrong basis. It propagates cleanly through correlations carrying baked-in constants and emerges as a plausible number. No amount of model capability catches this; it requires the system to carry units and basis with every value and reconcile them at every boundary.

5.5 Independent specialists, not one model in costumes

Asking a single model to “consider this as a completions engineer, then as an economist” produces one model's prior in several voices. Genuine cross-examination requires specialists that examine the evidence separately, reach positions independently, and are obliged to answer each other's challenges before a conclusion is allowed to close. What that produces is different in kind: surfaced disagreement instead of averaged confidence.

5.6 Provenance an evaluator can follow

Logging is not provenance. Recent work on the evidentiary value of runtime records makes the distinction precisely: a record answers a governance question only if it carries both a classification of what the recorded event was and the relationship — derivation, authority, temporal validity — on which the conclusion depends. [14] In practice, every figure must be attributable to the specific method that produced it and distinguishable from a figure the model merely asserted.

5.7 A boundary the architecture enforces

A general agent's design intent is to bring your material into the model's context. A system built for proprietary subsurface data takes the opposite stance: structured engineering data is referenced and computed upon, not shipped into a language model, and that is enforced in code rather than promised in a policy.

6. The evidence that the harness is what moves the number

The strongest available evidence that this distinction is decisive comes from medicine rather than energy, which is useful: it is peer-reviewed, high-volume, and free of any interest in this argument.

Goodell and colleagues, publishing in *npj Digital Medicine* in 2025, evaluated language-model performance on clinical calculations — safety-critical, numerically exacting work with a structure much like engineering computation. Unaided baseline performance was 66% across 212 trials, and only 39% on predictive tasks. The team then ran a focused analysis of 10,000 trials comparing mitigations. [7]

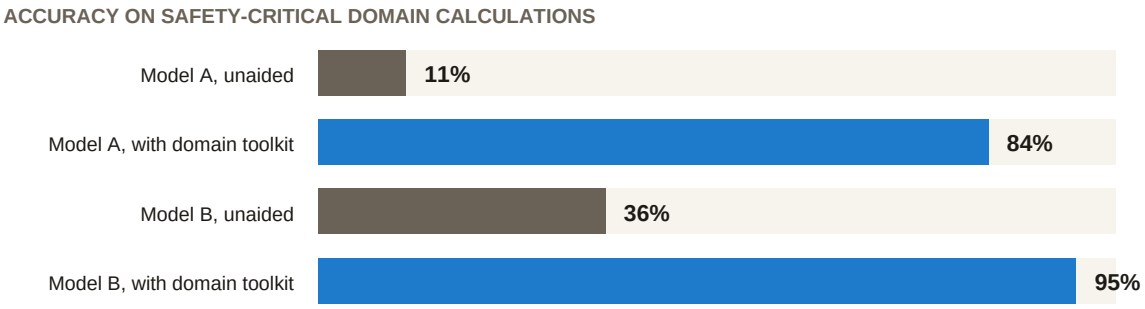

Figure 3. Accuracy before and after equipping models with task-specific deterministic tools, from a 10,000-trial peer-reviewed study in a safety-critical discipline. Source: Goodell et al., npj Digital Medicine, 2025 [7].

Two results matter. The magnitude: error fell by roughly five and a half times for one model and thirteen times for the other. And the ordering, which is counter-intuitive and rarely reported — a general code interpreter alone produced only modest gains; retrieval augmentation was stronger; the two together stronger still; and **task-specific deterministic tools produced the largest improvement of all**. [7] A separate 2026 study found a comparable pattern, noting that reasoning models “struggle in scenarios requiring… concise computation, or complex equation solving — areas where computational tools demonstrate distinct advantages.” [8]

This is the empirical case against the most common shortcut in the market: giving a capable model a sandbox and calling the result a domain platform. Purpose-built, validated computation for the specific task is not a marginal improvement over a general sandbox. In the only well-powered comparison

available, it was the difference between a system that fails a third of the time and one that fails one time in twenty.

7. “Then we will build the harness ourselves”

This is the right instinct, and it deserves a straight answer rather than a warning. Nothing described in Section 5 is unobtainable. There is no secret algorithm, and the architectural pattern is public — the academic work in Section 3.6 describes it in full, down to the agent roles. **The idea is not the asset.**

The asset is the accumulated, tested, domain-specific substance. In SolvxAI's case that is over a thousand engineering functions spanning more than twenty disciplines and domains, each unit-safe and traceable to the literature it implements; an audited map of which analyses consume the computed output of which others; explicit applicability criteria on the analyses where a wrong answer is worse than no answer; discipline-specific rigour rules that travel with the work; and roughly eight thousand six hundred automated tests that run on every change so that a figure produced last quarter is still reproducible this one.

It is buildable. It took a team of petroleum engineers and software engineers years, and the maintenance obligation does not end: every new model generation has to be revalidated against the same suite before it is allowed near a customer's result. The honest question for an operator is not whether this can be built, but whether building and maintaining it is the business you intend to be in.

8. Ten questions worth asking any vendor

These are diagnostic rather than rhetorical. They separate a domain system from a capable model with a domain-sounding interface, and they can be asked of us with the same force as of anyone else.

- **1.** Where does the number come from — a tested implementation you can name and version, or code the model wrote during my session?
- **2.** If I ask the same question twice, with the same inputs, do I get the same answer? Can you demonstrate that rather than assert it?
- **3.** What does the system refuse to do? Name three analyses it will decline, and the criteria that trigger the refusal.
- **4.** Show me a derivation, not an answer. Can an evaluator follow every figure back to its method and source?
- **5.** How do you know a tool was actually executed rather than described?
- **6.** What happens when two disciplines disagree — does the system average them, choose one, or surface the conflict?
- **7.** Does my structured engineering data enter a language model at any point?
- **8.** How are units and bases carried and reconciled between analyses?

- **9.** What was validated, against what benchmark, and when was that validation last re-run against the current model?
- **10.** Who is professionally accountable for the output, and what does the system give them to discharge that responsibility?

The last question is not rhetorical either. Professional engineering bodies have already answered it: a 2026 position statement holds that those who design, deploy, or oversee AI systems affecting public safety “should be held to the same standards as professional engineering licensure,” and names energy explicitly among the critical domains. [13] The industry's own responsible-AI guidance, published in 2026, frames the same obligations as a condition of maintaining the licence to operate. [12]

9. Conclusion

The model is a commodity input that improves on someone else's schedule. The harness is the durable asset.

This is not an argument that general-purpose agents are weak. They are remarkable, they will continue to improve, and any system worth buying should be able to adopt each new generation as it arrives — which is an argument for treating the model as replaceable rather than foundational.

It is an argument that the questions an operator actually cares about — can I defend this number, will it be the same tomorrow, does the system know when to stop, can a qualified evaluator follow it — are answered somewhere other than the model. They are answered by tested computation, an encoded understanding of how the discipline fits together, explicit limits, and provenance built to survive an audit.

Gartner estimates that of the thousands of vendors claiming agentic capability, only around one hundred and thirty offer anything substantially agentic, and forecasts that more than 40% of agentic AI projects will be cancelled by the end of 2027 on cost, unclear value, or inadequate risk controls. [9] In a market with that signal-to-noise ratio, the useful question is not which vendor has the better model. Everyone has approximately the same model. The useful question is what each has built around it, and whether they will show you.

How SolvxAI has already resolved this

The harness this paper describes is not a proposal. It is what SolvxAI is.

The computation is fixed: more than a thousand deterministic, SPE-cited engineering functions across twenty-plus disciplines, each unit-safe, versioned, and traceable to the literature it implements, regression-tested on every change by roughly eight thousand six hundred automated tests. The order is enforced: the harness understands the upstream asset lifecycle end to end and carries an audited map of which analyses consume the computed output of which others, so dependencies are resolved as the work is planned — net pay before volumetrics, fluid characterisation before lift design — rather than left

to chance per run.

And because the model is treated as a replaceable input rather than the foundation, each new model generation drops in and is revalidated against the same suite before it touches a customer's result. The ten questions in section 8 are ones we will answer on your data, not in the abstract.

ABOUT SOLVXAI

SolvxAI is the agentic operating system for upstream oil and gas. To see this working against your own asset, or to request the technical papers in this series, contact sd@nexaconsultinc.com.

References

Sources are listed with publication year; preliminary, self-reported, or vendor-authored work is flagged as such.

[1] **EngTrace: A Symbolic Benchmark for Verifiable Process Supervision of Engineering Reasoning.** arXiv:2511.01650 (November 2025; revised June 2026). 1,350 instances, three engineering branches, nine domains, twenty-seven models. https://arxiv.org/abs/2511.01650

[2] **LinAlg-Bench.** arXiv:2605.16675 (May 2026). 660 symbolically certified problems, 6,600 outputs, 1,156 classified failures. https://arxiv.org/abs/2605.16675

[3] **FEM-Bench.** arXiv:2512.20732 (December 2025; revised May 2026). Graduate-level computational mechanics; five attempts per task. https://arxiv.org/abs/2512.20732

[4] **Multiplication in Multimodal Large Language Models.** arXiv:2604.18203 (April 2026). https://arxiv.org/abs/2604.18203

[5] **Humanity's Last Exam.** arXiv:2501.14249 (January 2025). Cited here for the calibration pattern; per-model accuracy figures are superseded. https://arxiv.org/abs/2501.14249

[6] **PetroGraph: a multi-agent framework for reservoir history matching.** arXiv:2605.15028 (May 2026). Preprint; benchmarks SPE1, SPE9 and the Norne field. https://arxiv.org/abs/2605.15028

[7] **Goodell, A. J., Chu, S. N., Rouholiman, D., Chu, L. F.** "Large language model agents can use tools to perform clinical calculations." *npj Digital Medicine*, 17 March 2025. Peer-reviewed; 212 trials across 42 tasks plus a focused 10,000-trial analysis. https://www.nature.com/articles/s41746-025-01475-8

[8] **ReTool: reinforcement learning for strategic tool use.** arXiv:2504.11536 (2025; ICLR 2026). https://arxiv.org/abs/2504.11536

[9] **Gartner.** "Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027." Press release, 25 June 2025. Analyst forecast, not a measurement; the accompanying vendor estimate derives from Gartner's own market analysis. https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

[10] **Society of Petroleum Engineers.** *Standards Pertaining to the Estimating and Auditing of Oil and Gas Reserves Information*, approved 25 June 2019, sections 3.1 and 5.3.3. SPE standards are advisory; the Society imposes no sanction for non-use. https://www.spe.org/industry/docs/Reserves_Audit_Standards_June%202019_Final.pdf

[11] **US Securities and Exchange Commission.** Regulation S-X, Rule 4-10, 17 CFR §210.4-10(a)(24) and (a)(25), defining reasonable certainty and reliable technology. https://www.law.cornell.edu/cfr/text/17/210.4-10

[12] **International Association of Oil & Gas Producers (IOGP).** Report 817-1, *Principles for Responsible use of AI with guidelines for implementation*, April 2026. https://www.iogp.org/bookstore/product/iogps-principles-for-responsible-use-of-ai-with-guidelines-for-implementation/

[13] **National Society of Professional Engineers.** Position Statement No. 03-1774, *Artificial Intelligence*, adopted September 2023, revised February 2026. https://www.nspe.org/nspe-advocacy/explore-issues/professional-policies-and-position-statements/artificial-intelligence

[14] **From Runtime Records to Legal Findings.** arXiv:2607.00941 (July 2026). On the conditions under which runtime records can support governance determinations. https://arxiv.org/abs/2607.00941