SolvxAI
Data to Decision Powered by AI

SOLVXAI WHITE PAPER SERIES | NO. 07 | THE OPERATING SYSTEM

Physics Is Not a Pattern

The division of labour between what AI is genuinely good at and what must never be left to it

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

IN BRIEF

Upstream is being sold AI that does the engineering. The published record says this is precisely backwards. Frontier models are excellent at the things engineering has always been slowest at — reading heterogeneous data, recognising patterns, understanding how one piece of information relates to another — and measurably poor at the one thing the discipline had already solved, which is computing. This paper sets out the evidence for that split and the architecture it implies: an operating system in which the model supplies judgement and comprehension, established physics is executed by deterministic engines that behave like device drivers, and both sit above governed connections to the systems where the data actually lives. The point of an operating system is not the applications shipped with it. It is that other people can build on it.

Contents

1. Executive summary

Ask a model to recognise what it is looking at and it will outperform most humans. Ask it to multiply, and it will fail while sounding certain.

The industry conversation has settled on a premise worth examining: that the goal is AI which performs engineering. The measured evidence supports almost the opposite division of labour.

- **Comprehension holds.** On multimodal tasks, perception accuracy remained above 99% across modalities while computational accuracy on the same inputs was “often nearing zero” past a computational-load threshold. [3] The model reads the log, the table and the plot correctly. It then computes badly.
- **The failures are arithmetic, not conceptual.** On an engineering benchmark spanning nine domains and twenty-seven models, **79.5% of frontier-model errors were arithmetic** rather than conceptual, and reasoning-trace fidelity was 42.7% against 65.4% final-answer accuracy. [1] The understanding is largely present. The execution is not.
- **Giving it the right engine fixes it.** In a peer-reviewed study of 10,000 trials, equipping models with task-specific deterministic tools moved accuracy from **11% to 84%** and from **36% to 95%** — outperforming both retrieval augmentation and a general code interpreter. [4] Purpose-built computation beat a general sandbox by a wide margin.

That is not a limitation to be engineered around until models improve. It is a division of labour to be designed for, and it produces a specific architecture: judgement and comprehension in the model; established physics in deterministic engines that behave like device drivers; both sitting above governed connections to the databases, lakes and historians where the operational truth actually lives.

WHY WE CALL IT AN OPERATING SYSTEM AND NOT A PRODUCT

A product does the things it was built to do. An operating system schedules work, talks to devices through drivers, mediates access to resources, and — the part that matters — lets other people build applications nobody anticipated. Production surveillance, completion surveillance, integrity screening and engineering studies are applications on this system. They are not the system.

2. The premise worth examining

Every serious upstream operator is now being shown AI that reads a well file and produces an engineering answer. The demonstrations are real. The premise underneath them — that the model should be doing the engineering — is the part that does not survive the evidence.

It is worth being precise about why this premise is attractive. Petroleum engineering is computationally demanding and the computation looks like the hard part. It is not. The computation was solved decades ago, is documented in the literature, and is the most reliable component in the entire workflow. What has

never been solved is everything around it: finding the data, understanding what it says, noticing that two records describe the same well, recognising that a pressure response resembles one seen on another pad, and knowing which analysis the situation calls for.

> **The industry did not need a machine that could do the physics. It already had the physics. It needed a machine that could read.**

3. What the model is genuinely good at

Three capabilities are well evidenced and are exactly the ones upstream work is short of.

3.1 Reading heterogeneous material

Perception is the strongest measured capability in current systems. The same study that found computational accuracy collapsing reports perception accuracy remaining above 99% across modalities. [3] For an industry whose data arrives as scanned reports, vendor exports, inconsistent spreadsheets and decades-old logs, a component that reliably reads is not a convenience.

3.2 Recognising patterns and relationships

The second capability is relational: seeing that this well's decline resembles that one's, that a set of records refer to a single physical hole under different names, that an anomaly in one channel co-occurs with a change in another. This is judgement under ambiguity, and it is the part of the work that does not reduce to a formula.

3.3 Sequencing and selecting

The third is deciding what to do: which analysis applies, in what order, on what subset. The published evidence that models can learn strategic tool use supports this directly, and its authors locate the boundary in the same place we do — models "struggle in scenarios requiring… concise computation, or complex equation solving — areas where computational tools demonstrate distinct advantages." [5]

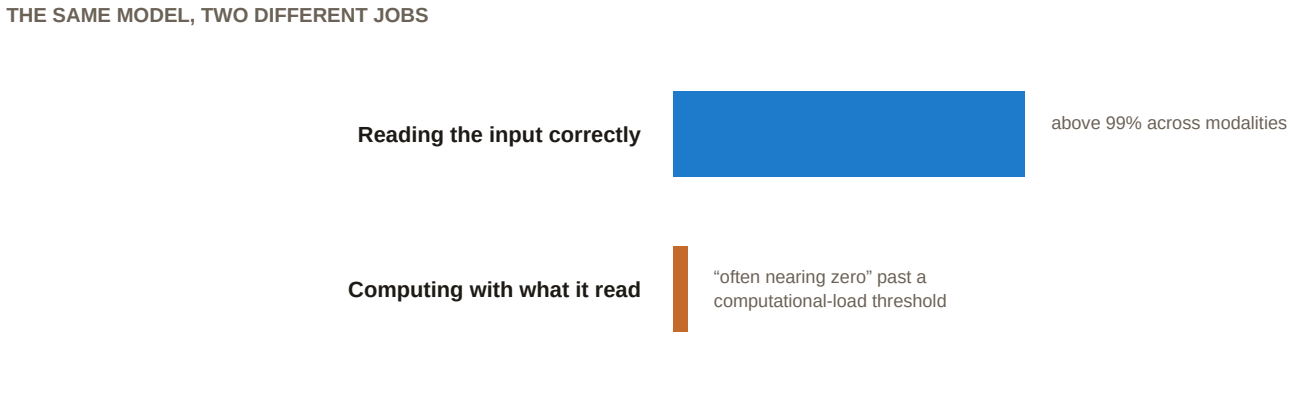

Figure 1. The split, measured on one population of models. Reading the input is close to solved; computing with it is not. An architecture that assigns both jobs to the same component inherits the lower number.

4. What it must not be asked to do

The counterpart is equally well evidenced, and it is the more consequential half.

Arithmetic dominates the failures. Not misunderstanding of the reservoir, not choosing the wrong method — arithmetic, at 79.5% of frontier-model errors on an engineering benchmark. [1] And consistency fails alongside it: on graduate-level computational mechanics tasks with five attempts each, a leading model solved 30 of 33 tasks at least once but only 26 of 33 on every attempt. [2]

That gap between “at least once” and “every time” is the whole problem. Established physics has one correct answer, and a component that produces it four times out of five has not partially succeeded — it has introduced a defect that is invisible on any individual run.

> **THE RULE THIS PRODUCES**
>
> Anything the literature has already settled — the correlations, the balances, the traverses, the closure criteria — is executed by an implementation written once, tested, versioned and cited. The model never computes it, never re-derives it, and never writes code to approximate it 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 the sense in which a deterministic engineering function behaves like a **device driver**. An operating system does not reimplement a disk controller each time a file is written, and does not ask a language model to improvise one. It calls a driver whose behaviour is fixed, documented and tested, and it is the fixedness that makes everything above it trustworthy.

5. The architecture that follows

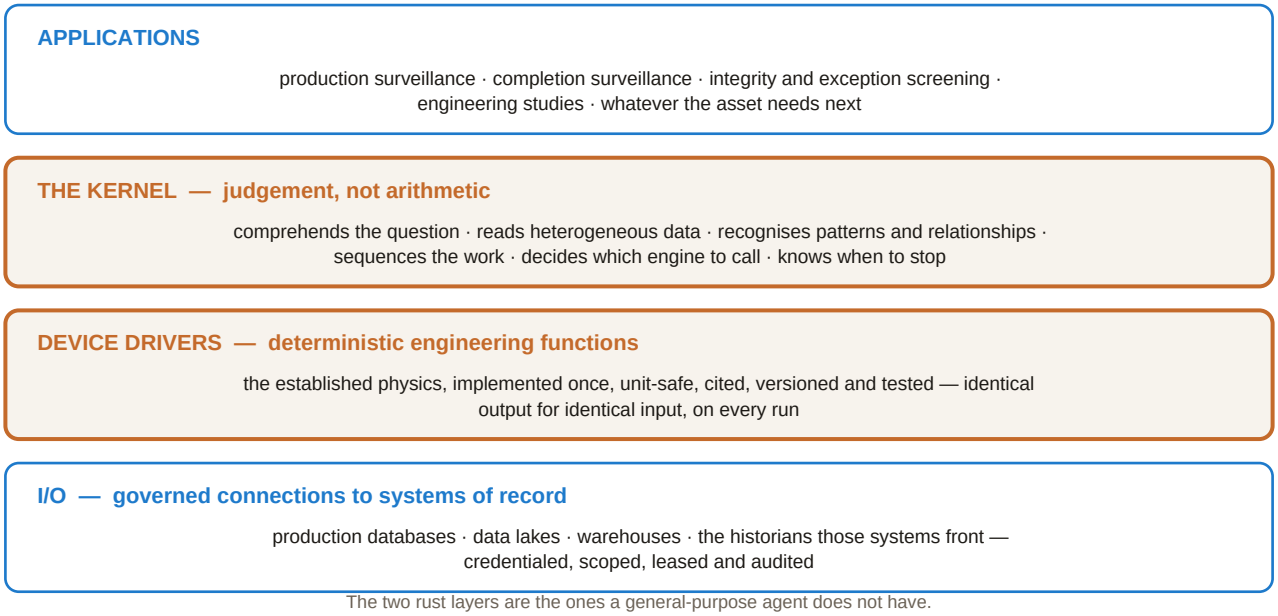

Figure 2. The operating-system model. Judgement and comprehension sit in the kernel; established physics sits below it in deterministic engines; both sit above governed connections to the systems where the data lives. Applications are built on top rather than shipped as the point.

5.1 The kernel does the reading and the deciding

It comprehends what was asked, reads whatever the asset actually has, recognises the entities and relationships in it, sequences the work in the order the field requires, chooses which engine to call — and, critically, recognises when no engine applies and stops. That last behaviour is the subject of the second paper in this series.

5.2 The drivers do the physics

Every established computation is a tested, cited, unit-safe implementation. The kernel selects and supplies inputs; it does not perform the calculation. This is what makes an answer reproducible in the sense a reserves auditor means, and it is the difference between a system that can be qualified and one that cannot.

5.3 The I/O layer reaches the systems of record

An operating system that cannot reach the data is an appliance. Governed connections to production databases, data lakes and warehouses — and to the historians those systems front — are what let the same capability serve a study, a surveillance routine, or a screen across hundreds of wells.

“Governed” is doing real work in that sentence. Credentials are held encrypted rather than handed to the model; access is granted per connection and per scope rather than globally; write access is leased rather than standing; unsafe network destinations are refused; and every access is recorded. A connection to a live production database is a serious thing to hand to an autonomous system, and the controls are the

reason it can be done at all.

5.4 Applications are the point of the whole exercise

What sits on top is deliberately open-ended. Continuous production surveillance. Completion surveillance across a pad. Integrity and exception screening that ranks what needs attention. Scheduled engineering studies. Early warning on a pattern that has not yet become a failure.

These are not separate products. They are the same kernel, the same drivers and the same connections, pointed at a different question on a different cadence — which is why the list can keep growing without the system being rebuilt.

6. Why this shape wins the surveillance case

The clearest illustration is continuous surveillance, because it is the use case where the division of labour is most obviously right.

Surveillance is a reading problem and a pattern problem before it is a computation problem. Something in the data has changed; the question is whether it matters, and whether it resembles a situation that ended badly somewhere else. That is precisely the work the model is measurably good at.

But the moment it matters, the response has to be computed — a critical rate, a drawdown limit, a remaining-life estimate — and it has to be the same number the engineer would have got. A system that recognises the pattern brilliantly and then improvises the physics has produced an alert nobody can act on. A system that computes rigorously but cannot notice anything unusual will never raise the alert at all.

The value is in the combination, running on a schedule, over connections to the live systems, with a threshold for when a human is told. Which is an operating system doing what operating systems do: scheduling work, calling drivers, mediating access, and escalating.

> **Catching the problem while it is still cheap is not an analytics feature. It is the return on the entire architecture.**

7. What this says about the market

Two claims are common in upstream AI at the moment, and this architecture takes a position on both.

“The model will get good enough at physics.” It may. The argument does not depend on it. Even a model that computed perfectly would still be the wrong place to put established physics, because reproducibility requires that the method be fixed and inspectable — and a method that is regenerated per run is neither, however accurate it becomes. The regulatory language on reliable technology asks for

consistency and repeatability, not for average accuracy. That is the third paper in this series.

“A general agent with a code interpreter is equivalent.” This one is directly contradicted by measurement. In the 10,000-trial study, a general code interpreter alone produced only modest gains; retrieval was stronger; the two together stronger still; and **task-specific deterministic tools produced the largest improvement of all**. [4] Giving a capable model a sandbox is not the same as giving it the right engine, and the gap has been measured.

8. Questions worth asking

- **1.** When the system produces a number, which component computed it — a tested implementation you can name and version, or code written during my session?
- **2.** If I ask the same question next quarter, on the same inputs, do I get the same number?
- **3.** What does the system connect to, and what exactly can it do there — read, write, for how long, and recorded where?
- **4.** Can it run on a schedule against live data, or only when someone asks it something?
- **5.** When it notices something, what does it hand me — an observation, or a computed consequence with its method attached?
- **6.** If I want an application nobody has built yet, am I waiting for a release, or can it be assembled from what is already there?
- **7.** Which parts of this are deterministic, and can you show me the line?

9. Limits of this paper

The evidence for the division of labour is drawn from general benchmarks and one peer-reviewed clinical study, not from petroleum engineering. No controlled study of this architecture in upstream exists, ours included, and the transfer is an argument rather than a measurement. We say so in each paper in this series because it remains true.

Two of the four load-bearing citations are preprints [1][2]; the tool-use study [4] is peer-reviewed and is the strongest single result. The multimodal computation finding [3] is a preprint reporting one family of tasks.

On the architecture itself: describing something as an operating system is a claim about shape, not a guarantee of quality. A badly built system with this shape would still be a badly built system. The questions in section 8 are the ones that distinguish them, and they are the ones we would expect to be asked of us.

10. Conclusion

Use the model for comprehension. Use deterministic engines for physics. Do not let anyone blur the line, including us.

The hype in upstream AI is pointed at the wrong target. The industry is being offered systems that attempt the engineering, when the engineering was the part that already worked. What did not work — and what has consumed the profession's time for decades — is finding the data, reading it, understanding what it refers to, and noticing what has changed.

Frontier models are extraordinarily good at that, and measurably poor at arithmetic. An architecture built around that fact rather than against it gets both halves: comprehension where comprehension is needed, and a fixed, citable, reproducible answer where the discipline has already settled the question.

Build it as an operating system rather than a product and the surveillance routine, the exception screen, the integrity check and the study that has not been imagined yet are all applications on the same foundation. That is the position we have taken, and section 8 is how to test whether anyone has actually taken it.

ABOUT SOLVXAI

SolvxAI is the agentic operating system for upstream oil and gas. The model comprehends, reads and decides; more than a thousand deterministic, SPE-cited engineering functions across twenty-plus disciplines execute the established physics; and governed connections reach the databases and data lakes where the operational record lives. Surveillance, exception screening and engineering studies run on that foundation on whatever cadence the asset needs. To see it against your own data, 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. **Preprint.** 1,350 instances, three engineering branches, nine domains, twenty-seven models. https://arxiv.org/abs/2511.01650

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

[3] **Multiplication in Multimodal Large Language Models.** arXiv:2604.18203, April 2026. **Preprint.** Reports computational accuracy collapsing past a computational-load threshold while perception accuracy remained above 99% across modalities. https://arxiv.org/abs/2604.18203

[4] **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. The strongest single result cited in this paper. https://www.nature.com/articles/s41746-025-01475-8

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

[6] **US Securities and Exchange Commission.** Regulation S-X, Rule 4-10, 17 CFR §210.4-10(a)(25), defining reliable technology in terms of consistency and repeatability. Binding regulation. https://www.ecfr.gov/current/title-17/chapter-II/part-210/section-210.4-10