SHADOW KIWI

AI is about to bypass your interface

PERSPECTIVE · ENGINEERING SIMULATION AND THE AI ERA

Engineering simulation has long been sold as a complete package. A validated solver sits at the core. Around it sit a graphical interface, scripting tools, templates and workflow automation. Much of the perceived value, and much of the switching cost, has lived in those outer layers.

That arrangement is changing. Frontier AI agents now read documentation, write scripts, generate input files and assemble working interfaces in hours. Some vendors find their product sitting beneath a layer they did not build and do not control.

Licence use depends on niche expertise

Engineering simulation software is complex and specialised, and proficiency takes years to build. When a trained user leaves a client organisation, the licence often sits unused and is not renewed.

Third-party AI models could change this, but only for products those models can work with.

The outer layers are reproducible

A graphical interface is a structured way to create and edit model data, and much of its value lies in knowing which menus, settings and click sequences produce a valid model. That knowledge usually lives with experienced users. An AI agent that understands the input format and the command set can drive the software directly, bypassing the interface and the tribal knowledge it demands.

This weakens a long-standing source of lock-in. Familiarity with an interface was once a reason to stay. When an agent handles the interface, familiarity matters less.

The solver is not reproducible

A validated solver is different. It embodies decades of numerical methods, verification against physical tests, certification evidence and accumulated edge cases. An agent can call a solver. It cannot credibly replace one. Engineering decisions with safety, regulatory or financial consequences still require results from a validated code.

The same holds for file formats and model data. Formats that carry geometry, materials, boundary conditions and results define how work moves between tools. Agents read and write these formats constantly, which makes them more important, not less.

Value therefore moves inward, toward the solver core and the model data.

Building a frontier model is not the answer

A common response is to build an in-house AI assistant to compete with general-purpose agents. This is rarely a sound use of capital. Frontier models improve faster than any single vendor can follow, and users will continue to bring their own agents.

We regard the stronger position as being the system that outside agents depend on. Four decisions follow from that.

1. Classify tool exposure

Not every part of a product carries the same exposure. We sort each tool, module and product layer into three groups, according to what it depends on.

- **Secure:** tools that depend on the validated solver and on certification. They are difficult to replace.
- **Defensible:** tools that depend on workflows, data and integrations. They can be protected with deliberate effort.
- **Exposed:** tools that depend mainly on the interface, templates and convenience features. An agent can reproduce these.

The classification shows where development effort protects the product and where it defends a position that is already weakening.

2. Protect the core

If value concentrates in the solver and the formats, protection should concentrate there as well. The solver remains behind a controlled boundary. File format policy is set deliberately: what is open, what is documented, and what remains proprietary.

Licence terms state how agents and third-party layers may use the solver. Seat-based terms assume a person at a screen; agents run batches and make calls at volumes no person would, and the terms address that access directly. Automation and API access are treated as part of AI strategy, through a defined interface that the vendor controls, rather than as a minor add-on to the desktop licence.

3. Make the product easy for agents to operate

Agents will reach the product either way. The choice is whether they run through the vendor or around it.

An agent onboarding layer makes the first path the easier one. It combines clean, stable APIs, connectors built on open standards, and machine-readable documentation that tells an agent how to use the product correctly. Clear error messages and validated examples matter more to an agent than to a person.

A product that is easy for agents to operate becomes the default inside automated workflows. A product that is difficult to operate is wrapped by someone else, and that party then holds the customer relationship.

4. Decide on an open-source posture

Open-source solvers, meshers and pre-processors are well represented in what frontier models have learned. Agents often reach for them first. Each product line needs a deliberate position.

One option is to couple with open tools: read and write their formats, interoperate cleanly, and position the commercial solver as the validated step in a mixed workflow. The other is to differentiate on accuracy, certification, support and physics coverage that open tools do not provide. Both are viable. Drifting between them is not.

Uncharted, but readable

AI integration in engineering simulation remains largely uncharted territory. The direction is clear. Agents are learning to drive simulation software directly, around the interface and the expertise it once demanded. The validated solver and the model data remain at the centre.

Vendors that prepare for this can come out ahead. They can open the product to the engineers who could not previously use it, make it the natural place for agents to do work, and protect what is difficult to copy. The aim is not to out-build frontier AI. It is to be the software that agents drive.

Few vendors have a clear picture of what current models can already do with their software. We demonstrate it within the vendor's own product, and identify where it is exposed.

With the product properly integrated, engineers with the technical background, but without experience in a specific package, would be able to operate it through third-party AI models and produce results immediately. More people would use the software, licence use would rise, and renewals would follow. Client teams would spend more time developing their own products and less time mastering the software.

Shadow Kiwi advises the simulation software industry on strategy for the AI era. shadowkiwi.com