

```
$ operators_map --trends=3
```

The software economy is being repriced.

Three trends in AI: what intelligence costs, how software is shaped, and how work gets handed over.

01 THE INPUT

02 THE STRUCTURE

03 THE RELATIONSHIP



LARRY GALAN · ESSAY · 12 MIN · JULY 2026

Every map of AI you have seen is one of two kinds. The first is a stack: infrastructure at the bottom, models in the middle, applications on top, with logos pinned to each layer like specimens. The second is a list: eleven trends, thirteen trends, ranked by survey response and annotated with adoption percentages. Both are useful, and both are static. They tell you where everything sits. **Neither tells you what is moving.**

This view comes from inside an AI-native company — not from polling the phenomenon, but from standing in it. From inside, what is happening does not look like thirteen trends. It looks like three, and they are not parallel observations. They are a sequence, each one forcing the next.

One sentence of scope, because it matters: this is an argument about the software economy — not the semiconductor supply chain, not geopolitics, not the labor market at large. Within that scope, nearly every development you can name — every price cut, protocol, agent framework, eval startup, and outcome-based contract — is a claim about one of three things: what intelligence costs, how software is shaped, or how work gets handed over. An economy has no fourth position: supply, production, demand.

Here is the chain.

THE CHAIN

TREND 01 – THE INPUT

What intelligence costs

// supply



TREND 02 – THE STRUCTURE

How software is shaped

// production



TREND 03 – THE RELATIONSHIP

How work gets handed over

// demand

each trend ends by creating the necessity of the next.

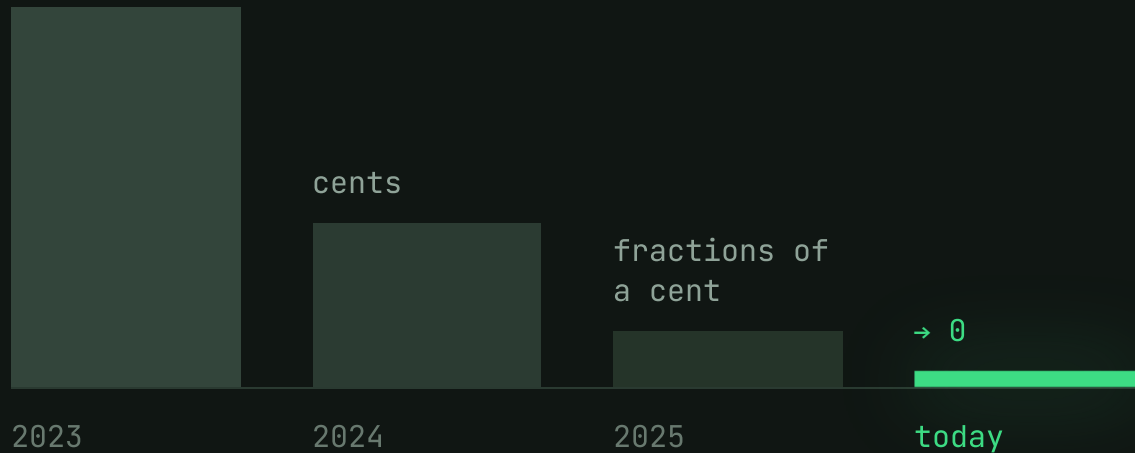
TREND 01 / THE INPUT

Intelligence is commoditizing and diffusing.

Start with the price. The cost of a unit of cognition — a competent answer, a working function, a clean summary — has collapsed for three straight years. Capability that cost dollars per task in 2023 costs fractions of a cent today, and each model generation delivers last year's frontier performance at a fraction of last year's price.

COST PER UNIT OF COGNITION

dollars /
task



last year's frontier, at a fraction of last year's price – every generation.

The gap at the top is compressing too. Open-weight models — DeepSeek, Llama, Qwen — now trail the frontier by months, not years. Distilled small models match what flagships did a release cycle ago. Benchmark scores across labs are clustering, not separating. Whatever any single lab can do, the field can do shortly after, cheaper.

The clearest evidence that this is commoditization, not just competition, is the infrastructure around it. Routing layers like OpenRouter shop a request across dozens of models and providers, filling it wherever quality clears the bar at the lowest price.

■ **You cannot arbitrage a differentiated good. The existence of an arbitrage layer is the market telling you the underlying product is interchangeable.**

And because the input is cheap, it is everywhere. That is the second face of the same trend: diffusion. Expert-grade output now arrives at the same marginal cost to a teenager in a bedroom and a Fortune 500 procurement office. Reasoning that once required hiring — an analyst, a paralegal, a junior developer — now comes through a text box, priced like a utility.

Be precise about what is diffusing, though, because this is where the trend gets overstated. Expertise is decoupling from experts. The knowledge is diffusing; the judgment is not. A model gives you an oncologist's knowledge. It does not give you an oncologist — the calibrated sense of when the confident answer is the wrong one, the accountability, the license on the wall. **The price of an expert answer is heading toward zero. The price of knowing whether it is right is not.** Hold that gap; it returns at the end of the chain and sets its speed limit.

WHAT DIFFUSES — AND WHAT DOESN'T

expert answers



→ 0

judgment



holds

the knowledge diffuses; the judgment does not.

One honest complication: while intelligence commoditizes at the point of consumption, its production is concentrating — compute, power, and datacenters accumulating in fewer, larger hands. That is how commodities usually behave. Oil production concentrated into a handful of giants precisely while oil became the world's most interchangeable product. Concentration upstream, commoditization downstream: a signature of the trend, not an exception.

The consequence: there is little durable advantage left in access to intelligence alone. Everyone has it, everyone will have more of it, and it keeps getting cheaper. Whatever value was going to accrue to simply *having* the input migrates up the stack.

Which forces the question the next trend answers: **up the stack to what?**

TREND 02 / THE STRUCTURE

The application is dissolving into the system.

For forty years, the unit of software value has been the application. You bought apps, opened apps, were trained on apps; entire org charts and budget lines are shaped like application portfolios.

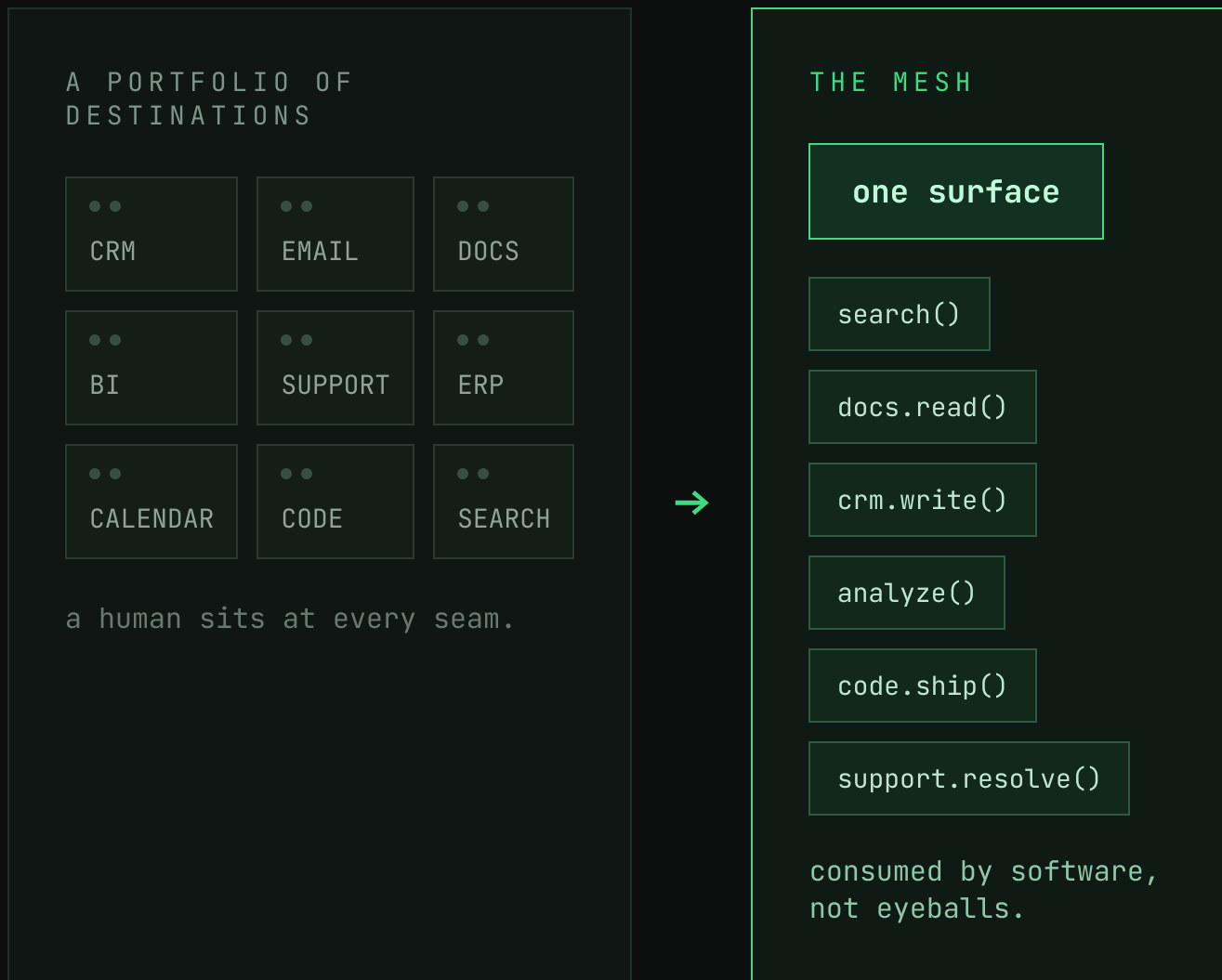
But the application was never really the unit of value. It was the unit of *packaging* — an interface, some logic, and accumulated data, bundled and sold as a destination. That bundle made sense when integration was expensive and the interface was how a human reached the logic. Cheap, universal intelligence breaks all three seams at once.

The interface is collapsing into universal surfaces. More work now starts in a single conversational window that reaches into whatever capability it needs — search, documents, analysis, code — without the human visiting the tools that provide them. Microsoft is threading Copilot across the entire Office suite; enterprise search products wire CRM, email, and chat into one queryable layer. The destination model of software weakens when one surface can operate all the destinations.

The logic is becoming a callable capability. Interoperability protocols — MCP is the clearest example — turn applications into tools that other software invokes. An app that spent a decade building a workflow finds that workflow addressable as a function call inside someone else's system. This is why the first wave of thin wrappers — the chat-with-your-PDF products, the single-feature writing tools — was absorbed so quickly: not because they were bad products, but because “product” stopped being the shape of what they built.

A feature reachable by protocol does not need its own front door.

And the systems doing the invoking are converging on a familiar architecture. Inside any serious agent harness you find the pattern language of distributed systems — orchestration, queues, retries, state management, observability — the same machinery that runs payment infrastructure, now applied to cognition. When cognition becomes a cheap callable service, coordinating it becomes an infrastructure problem — and the industry already owns that playbook.



Put those together and the direction is hard to miss: software is converging from a portfolio of destinations into a mesh — **one large product, consumed increasingly by other software rather than by eyeballs.** The UI is not disappearing because designers failed. It is receding because the primary consumer of software is, more and more often, not a person.

The consequence sets up the final move: a mesh can span an entire workflow end to end. A portfolio of apps never could — every seam between apps was a place a human had to sit, translating, copying,

clicking. You cannot delegate work to a fragment of it. **The seams are going.**

TREND 03 / THE RELATIONSHIP

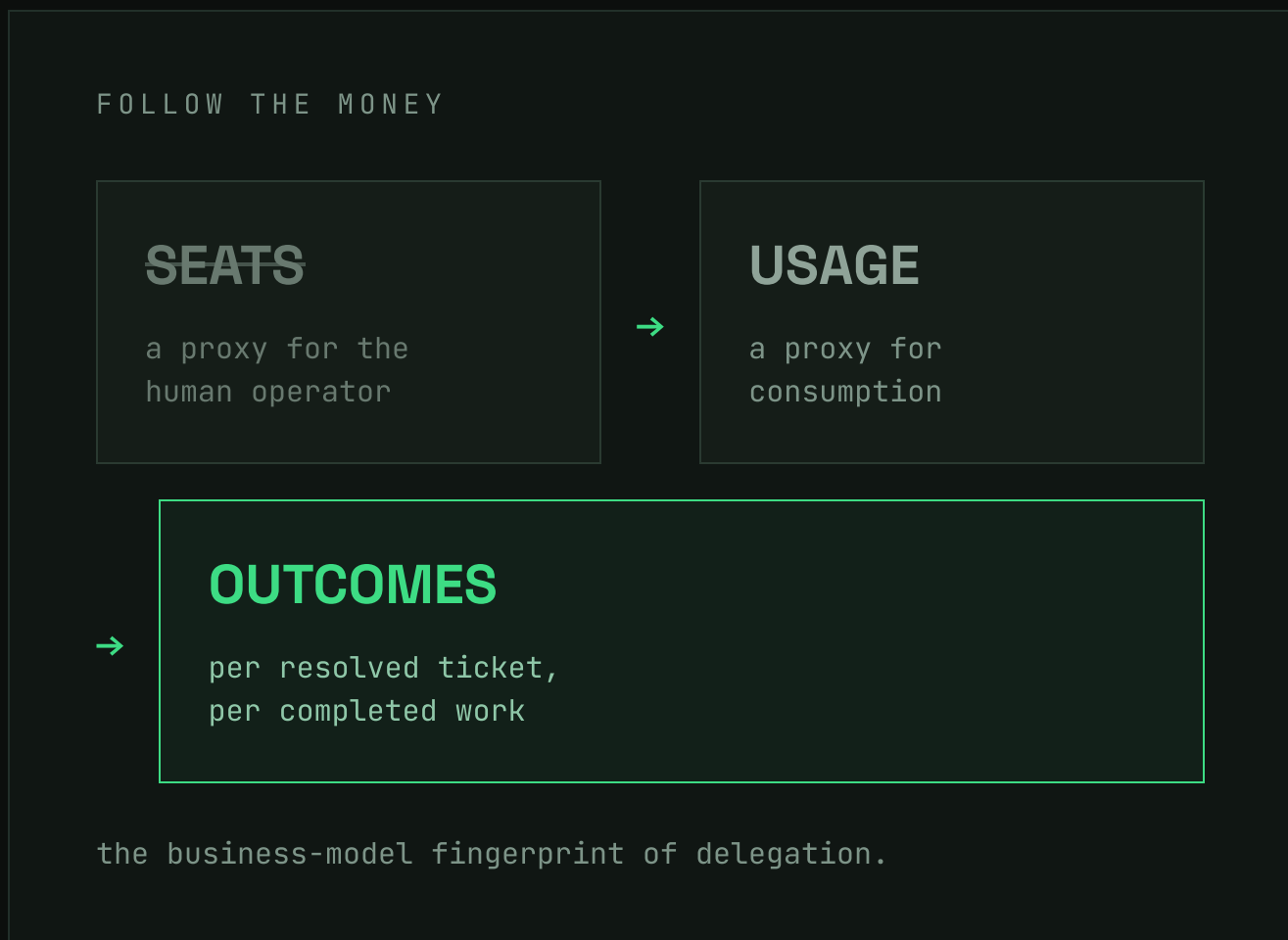
Operation is becoming delegation.

For the entire history of software, the relationship was operation. Software was a tool; a human ran it. Better software meant a better-run tool. Every pricing model, training program, and productivity metric in the industry assumes that arrangement.

That arrangement is beginning to invert. People are starting to stop *using* software and start *assigning work to it*. Coding agents take a ticket and return a pull request. Support agents take a queue and return resolutions. Research agents take a question and return a brief. The human's contribution shifts from operating the tool to specifying the work and judging the result — **from operator to supervisor.**

If you want to know whether this is real, do not watch the demos. **Watch the money.** Pricing is migrating from seats to usage to outcomes. Intercom prices its support agent per resolved conversation, not per

seat; coding agents are sold by work performed, not by license. That migration is the business-model fingerprint of delegation, because the seat was always a proxy for a human operator — you do not pay an employee per login. Once software does the work rather than assisting it, the seat has nothing left to measure. Procurement follows: companies are starting to buy completed work where they used to buy licenses, which puts software in price competition not with other software, but with labor.



And here the gap from trend one returns, exactly on schedule.

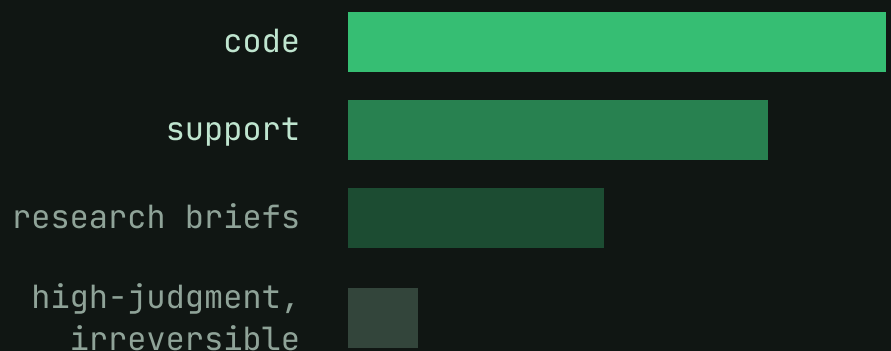
Delegation is gated by verification. You only hand over what you can check. The expert answers are everywhere; the judgment to evaluate

them is still scarce, still human, still expensive. That gap is not a footnote to the delegation trend. It is the governor on its speed.

It explains the order of advance: work is handed over not by how impressive it is but by how checkable it is. Code goes early — not because it is easy, but because tests, compilers, and review provide cheap verification, and version control makes errors reversible. Support goes early because resolution is measurable. High-judgment, irreversible decisions go last — not because models cannot attempt them, but because no one can affordably confirm the attempt succeeded.

■ **The delegation frontier is the verification frontier, moving one checkable job at a time.**

THE VERIFICATION FRONTIER



bar length = how cheaply the work can be checked. checkable goes first.

It also explains an industry that would otherwise be hard to account for: eval platforms, agent observability, guardrails, audit tooling, insurance against agent error. A verification economy is materializing around the gap between what machines can produce and what humans can confirm. Those companies are not selling caution. **They are selling delegation capacity** — every improvement in checking expands the amount of work that can be handed over.

One more thing, addressed rather than dodged. The big executive surveys — careful, well-run, worth reading — report that most leaders expect AI to remain a support tool in the near term. I do not think they are wrong about their intentions. But surveys measure expectations, and this is an argument about incentives: when completed work costs less than the tools-plus-labor it replaces, and verification keeps expanding what counts as checkable, **expectations rarely beat unit economics. They just lose on a lag.**

THE CHAIN, CLOSED

The inversion of scarcity.

Each trend ends by creating the necessity of the next. Intelligence commoditized, so value migrated up the stack. It migrated into structure: applications dissolved into systems that span whole workflows. And a

system that spans a whole workflow is, for the first time, something you can delegate to — at whatever speed verification allows. What intelligence costs, how software is shaped, how work gets handed over.

Input, structure, relationship. One chain.

The chain ends in an inversion worth sitting with, because most strategy conversations still point at the wrong end of it.

The things being fought over hardest — access to frontier models, AI features on the roadmap, the size of the application portfolio — are the things the chain makes abundant. Intelligence is a commodity. Features built on a commodity input converge. Applications are dissolving as a category. **Competing to accumulate the abundant things is running up an escalator that is heading down.**

What the chain makes scarce sits at the other end. **Proprietary context:** the data and workflow understanding that turn cheap general intelligence into something specifically useful, which no competitor can download.

Distribution: the relationships and channels that put intelligence in front of real work — worth more under commoditization, not less, because when the product is interchangeable, the shelf decides. And **verification:** the capacity to check delegated work, the binding constraint on the entire third trend and the closest thing this economy has to a new bottleneck resource.

FOUGHT OVER — AND
ABUNDANT

↓ access to frontier
models

↓ AI features on the
roadmap

↓ the application
portfolio

ACTUALLY SCARCE

↑ proprietary context

↑ distribution

↑ verification capacity

So the strategic question changes shape. It is no longer “what is our AI strategy?” — a question about acquiring the abundant input, which everyone will answer the same way at the same price. The better question is the one the chain actually poses: *what work do we understand deeply enough to verify?* That boundary — not model access, not headcount, not the app portfolio — is the boundary of what you can delegate. And delegation is where the value is going.

Keep the familiar maps; they tell you where everything sits. This one tells you where it is all going — and the chain ends at the same door every time.


```
$ exit --one-line
```

**In an economy
where every answer
is cheap, the most
expensive thing left
is knowing which
ones are right.**



LARRY GALAN

building an AI-native firm

JULY 2026

