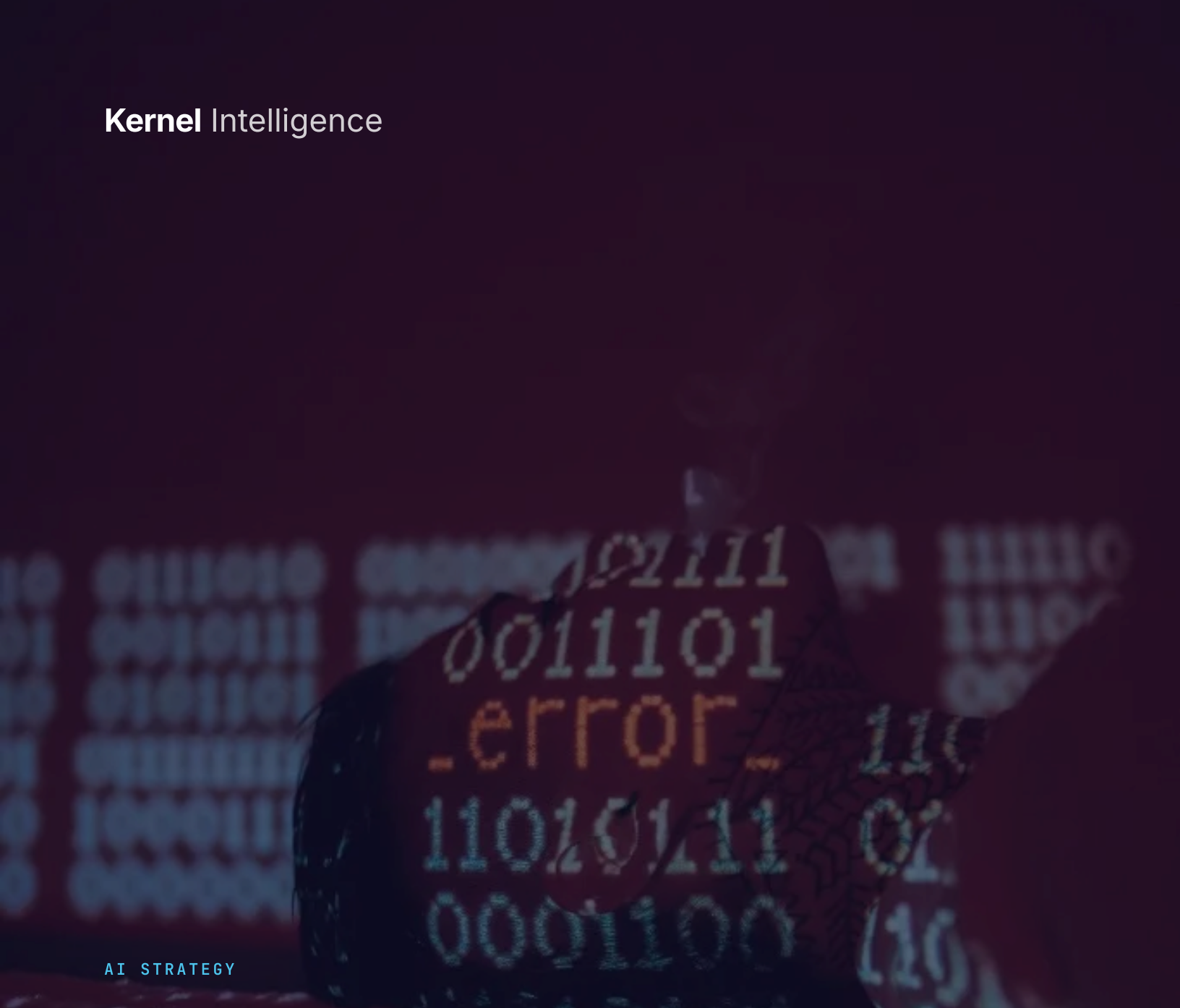
Kernel Intelligence



AI STRATEGY

The AI Battle Scars You Have Won't Save You From This

Chat fails loud. Agents fail quiet. The instincts you earned from your last AI project are real — and they're calibrated for a different fight.

Ross Goldenberg · CEO, Kernel Intelligence

A friend called me last week for a gut check on an AI project. It wasn't the first time this year I've gotten this call.

He'd shipped something last year that worked well. A chatbot built with a couple of prompts and not a lot else underneath it. Users liked it. It didn't take much to maintain.

Now he's building the next thing, and it's a different animal. A multi-step agent that reads certificates of insurance, checks coverage against what's required, and acts on what it finds. Flags a gap. Clears a vendor. Updates a compliance record. No human reading the output in between.

He's excited, and he should be. It's a real step up from where he was (and most of us were) last year. But he's reaching for the same playbook. Throw it at the model, see how it does, tweak the prompt when something's off, move on.

He's a smart guy. He has real scars from that first project. He knows to watch for AI that sounds too confident, that states something oddly, that drifts in tone when you least expect it. Those instincts are hard won and correct.

They're also for a different fight.

And the more sophisticated the agent, the further one bad read travels before anyone notices. Every added step is one more piece of orchestration built on top of the same shaky foundation, and impressive orchestration is exactly what makes a bad foundation hard to see. Whether that vendor gets cleared depends on whether the agent read the coverage limit right off page eleven. If it didn't, the flag, the clearance, the record, all inherit the mistake and act on it with total confidence.

Two kinds of wrong

> Chat fails loud. A reader feels an answer is off, rereads it, asks again. A human's sitting right there, evaluating in real time.

Agents fail quiet. When an agent reads a coverage limit, checks a certificate, clears a vendor, there's no one pausing to sanity check it. It's right or it's wrong, and nothing about the output says which. The thing that used to catch the mistake, a person reading in real time, isn't in the loop anymore.

Every wave of AI adoption leaves you with scars, and every set of scars trains you to spot one shape of failure. The mistake isn't having those scars. It's assuming they cover the next fight too, when the failure just flipped from something a person would catch to something no one's positioned to catch at all.

Sophistication hides the problem, it doesn't solve it

A five-step agent that reads a certificate, cross-references the policy, checks coverage against what's required, updates the record, and notifies the vendor looks like serious engineering. It is. The orchestration has gotten genuinely good.

But every step is downstream of the last one, and a mistake upstream affects everything that comes after. Misread the coverage limit and steps two through five never know it. They just do their job, correctly, on a bad number. The check passes because it's checking the wrong limit against the right threshold. The record updates. The vendor clears. Everything executes cleanly, and the whole chain is wrong.

> Sophistication has nothing to do with whether the data underneath is right. It just decides how many confident, correctly executed steps get stacked on the mistake before anyone has a reason to look.

Here, that reason shows up as an uncovered claim months later, not an error on a screen today.

None of this is specific to insurance. Swap in any multi-step agent doing real work: one that reads a lease and updates a rent schedule, reads a contract and files a renewal, reads an invoice and cuts a payment. Same shape every time. A chain of steps, each one competent, each one trusting the step before it, and one bad read at the bottom that nothing in the chain is built to notice.

We've watched this movie before

I heard some version of “won't this just get solved when the models get a little better” a lot about two years ago, when hallucination was the fear everyone could name. The real issue, that a model alone isn't a whole system, was invisible until people ran it at scale and watched it break. Nobody asks that question anymore, and nobody ships on a bare prompt anymore either. Not because someone explained it well. Because enough people hit the wall themselves.

“Why can't I just point the agent at it” is the same question, one cycle behind. If you haven't watched an agent quietly get something wrong for a week before anyone noticed, the warning sounds theoretical. You kind of have to get burned once.

The number that sounds better than it is

Point a model at certificates of insurance and, out of the box, you'll get maybe 60 to 70 percent of the coverage fields matching exactly. More with a week of tuning. Real number, decent place to start.

But think about what used to happen when an old compliance workflow broke, a spreadsheet, a rules engine, a checklist filled out by hand. You got nothing. A blank screen. A cell with an error. Everyone knew, instinctively, to go check.

AI at 60 to 70 percent doesn't fail that way. No blank field, no error. You get a coverage limit, confident and correctly formatted, sitting exactly where the right one should be. Nothing flags which bucket it's in.

> Most people are busy and won't double check a number that looks fine. So an unflagged wrong answer isn't neutral, it's worse than nothing.

Blank tells you something's missing. Wrong and confident tells you nothing, and the vendor's uncovered claim finds out first and potentially costs millions.

It's not chat versus agents, it's small versus scale

Chat isn't immune once it grows up either. Summarizing one document for one person is forgiving, the way any small project is forgiving. Summarizing thousands and letting people decide off those summaries hits the same wall structured tasks hit from day one.

> The real line isn't chat versus agents. It's small and low-stakes versus anything that has to hold up at scale or answer something objective. Small forgives you. Scale doesn't.

What actually fixes this

The instinct is “be more careful” or “add a review step.” That's the old playbook, a human reading in real time, and it doesn't scale to agents any better than it scaled to documents.

What helps is visibility into the middle of the chain, not just the ends. Most agents today give you the input, the final output, and a black box in between. You can't tell whether step three trusted a clean read or a guess, because by step four they look identical.

The fix is atomic units of data, each carrying its own confidence, so a low-confidence read shows up as low confidence instead of getting smoothed into a normal-looking answer. That's what lets you watch the pipeline instead of just the outcome.

And once an agent can see its own confidence, it can act like a careful person would. Someone unsure of a number rereads the paragraph, checks another page, flags it before moving on. An agent that knows it's uncertain can do the same: pull the corroborating page, check a second source, hold the step open instead of passing a guess downstream as fact.

Not a human reviewing everything at the end. Not blind trust in the model either. Confidence that travels with the data, at every step, so the system knows what it doesn't know before it acts on it.

The gut check

Right now, caution is calibrated to the failure that announces itself, the answer that sounds confident and wrong in a way you can point to in a demo.

The next set of scars comes from something quieter. The number read wrong. The gap that didn't get flagged. Correctly formatted, sitting where the right one should be, waiting to be trusted, until the day it matters and there's nothing behind it.

> **Don't ask how good the demo looks. Ask what it takes to get that same quality at scale, what breaks along the way, and how you'll know when it does.**

If you don't have an answer, you don't have a system. You have a very confident guess with a business card that says Vice President.

A NOTE ON KERNEL

Ground truth from your documents

This is the problem we built Kernel to solve. The data layer that sits between documents and agents: extraction with provenance, confidence scoring at the level of each individual fact, and the normalization that lets an agent trust a number instead of guessing at it.

That confidence travels with the data. An agent working with Kernel knows when it's looking at a clean, cited read versus something that needs a second source before it acts. Check the other page, hold the step open, ask before moving on, the way a careful person would.

If the pattern in this piece feels familiar, and your team is spending more time babysitting extraction than building the agent you actually set out to build, we'd like to have that conversation.

Ross Goldenberg
CEO, Kernel Intelligence
ross.goldenberg@heykernel.com
www.heykernel.com

Kernel Intelligence

Originally published at https://www.heykernel.com/blog/the-ai-battle-scars