Most of Your AI Work Isn't Hard

By Rahul Jindal

My job at Google is AI transformation, so I spend a lot of my week looking at what this technology actually costs an organisation once the pilot is over. The number everyone opens with is the price per token, and it keeps falling. In 2025 we brought Gemini serving costs down 78%. Since Gemini 3 shipped, the cost of a core AI response has come down by more than 30% again. Those are our published figures and they are genuinely good.

Now ask the same organisation what happened to its AI bill. It went up.

Finance teams tend to look at the rising bill and reach for the falling price to explain why it's temporary. It isn't, and there's a name for what's going on. In 1865 William Stanley Jevons noticed that as steam engines got better at burning coal, Britain burned more coal, not less. **Making a thing cheaper to use is not the same as using less of it.** Cheaper fuel meant more places worth putting an engine, and the new uses swamped the saving.

Our own numbers show both halves of that in one place. Two years ago we were processing 9.7 trillion tokens a month across our surfaces. Today it's 3.2 quadrillion, more than a three-hundred-fold increase. On the July earnings call Sundar put the model APIs alone at about 22 billion tokens a minute, up from 16 billion the quarter before. Serving got much cheaper. Nothing about the total got smaller.

There's a second thing happening at the top of the range, and it runs the other way. Tomasz Tunguz tracks GPU rental prices, and the B200, Nvidia's top-end chip, went from $2.31 an hour in early March to $4.95. That's a 114% jump in six weeks. So the cheap end of the menu keeps getting cheaper while the expensive end gets dearer, and what happens to your bill depends almost entirely on how much of your work sits at each end.

Which is a question most organisations have never put to themselves.

It helps to think about it the way a hospital thinks about triage. Not every patient needs the consultant. Most need someone competent, quickly, and the skill of the system is

knowing which is which. AI work is the same, and the surprising part is how much of it turns out to be routine once you look.

Sorting a support ticket into one of nine buckets. Pulling the invoice number, date and total off a form that has looked identical for six years. Tagging company names in a contract. Deciding which of four downstream tools a request belongs to. Summarising a document whose structure you already know. First-pass triage on almost anything, where the only real job is working out whether a person or a bigger model needs to see it.

There's a category of model built for exactly this now. Small language models, or SLMs: a few billion parameters instead of a few hundred billion, built to do one job well rather than to hold a conversation. Our Gemma family, Microsoft's Phi, Nvidia's Nemotron. Nvidia's own researchers published a paper arguing SLMs are “sufficiently powerful, inherently more suitable, and necessarily more economical” for most of the calls inside an agent system, and should be the default rather than the exception. Worth noticing who is saying that, given Nvidia sells the hardware for the expensive path.

We've started building the same judgement into the model itself. Gemini 3 ships with dynamic thinking, which automatically adjusts how much reasoning effort a request gets based on how complex it actually is. Somebody is going to make that call either way. The only question is whether it's you.

Microsoft ran the experiment in public in July and published the number. Their vulnerability-finding system, MDASH, used to run on a fleet of OpenAI models. They trained a small purpose-built model, MAI-Cyber-1-Flash, five billion active parameters, and pointed it at the work. It now handles up to 90% of the tasks, and GPT-5.4 gets called for the hardest 10%. Total cost fell by half, and the combination scored better on the CyberGym benchmark than the setup it replaced. Cheaper and better, out of models anyone can buy.

That saving didn't come out of a contract. Someone sat down and worked out which tasks were genuinely hard.

Satya said the strategic version of it on Microsoft's earnings call on 29 July: keep the model separate from the harness that holds your memory, your context and your action space, so any given model stays swappable. Azure's catalogue is past 11,000 models and

customers building on more than one provider are up fivefold since January. The one-time standardisation decision, made at the top, with a contract, is over.

The AI line has spent two years sitting with whoever signs the vendor agreements, which was reasonable while the real choice was which of three labs to bet on. That isn't the choice any more. Whether your AI spend turns into an asset or into overhead now gets settled by engineers picking, task by task, what deserves the expensive call. Hundreds of small decisions a week, made mostly by people who have never seen the invoice.

And that's the actual gap. **Not the spend, the split.** The person who can read the bill usually can't read the code, and the people who can read the code have never been shown the bill. So the expensive model becomes a default that nobody ever revisited, and the cost of that only shows up months later, in a number no one can explain.

So if you own a P&L with AI in it, the uncomfortable question isn't which vendor, or how much are we spending. It's this: **what fraction of our model calls actually need the frontier model, and who decided that?**

If the honest answer to the second half is that nobody decided, you've found where the money goes. And the fix isn't exotic. Microsoft got half their cost back by being honest about what was hard, and that's available to everyone. It just lives in engineering judgement rather than in a negotiation, and judgement is a great deal harder to put on a slide.

Written in a personal capacity. Every Google figure here is from our published investor materials and earnings calls.
