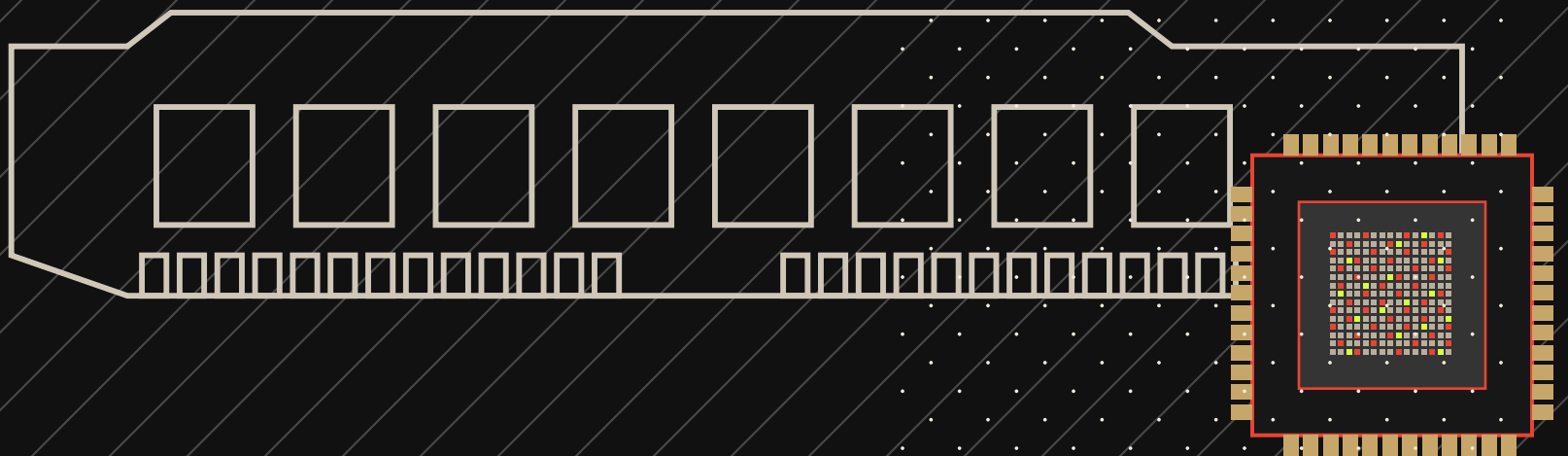

MEMORY WARS



You took a job from yourself and gave it to AI. Thought you'd save some time. It gets halfway through, forgets ~~what you're doing, and~~ confidently finishes the wrong

You correct it.

It apologizes, explains the mistake, and does it again.

By the third attempt, you're doing two jobs: the one you came to do and the one the machine was supposed to do for you. The machine, meanwhile, has been working very hard. Just ask the electricity company.

Welcome to the first Memory War.

And the absurd thing is that the machine doesn't necessarily lack the knowledge or the tools. It can have the instructions sitting right in front of it and still ignore them. We just spent a considerable part of this conversation proving that.

Who pays, and how much, for all this stupidity?

Yeah, who? I wonder. But, more than you'd think.

AI companies measure how fast their models respond, how many questions they answer, how many tasks their agents finish. They spend fortunes making those numbers look better.

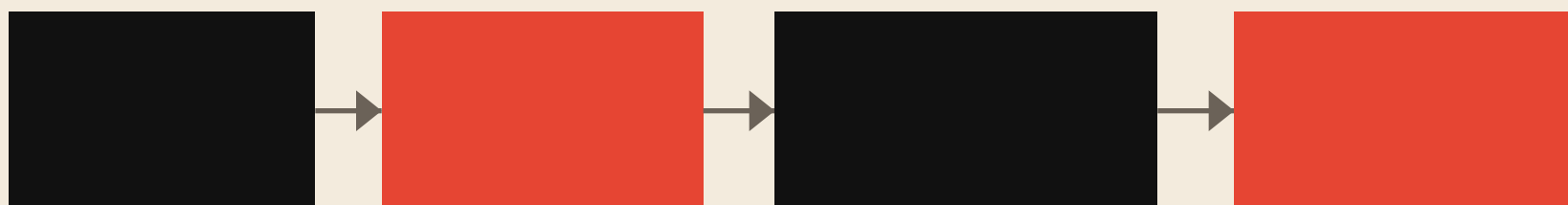
But there's another set of numbers.

What looks good on paper	What happens at your desk
The AI finished the task.	You spent an hour fixing it.
The model knows the answer.	It keeps making the mistake.
The agent completed ten steps.	Step two was wrong. Everything after it is garbage.
The new model is twice as fast.	Congratulations. It gets things wrong twice as quickly.

When you catch the mistake, you spend your time repairing it. When you don't catch it, the mistake goes out into the world.

It lands in software. In business decisions. In contracts. In machines somebody has to operate. In work that another AI picks up and confidently makes worse.

Now comes the expensive part.



To run these systems, companies need processors, enormous amounts of fast memory, electricity, cooling, buildings, and factories. Bigger models and repeated attempts consume more of those resources. Some of that computing does extraordinary work. Some of it is spent recovering from failures that shouldn't have happened.

And the biggest buyers are booking memory production years ahead.

Which means that a struggle over whether machines can remember what they're doing becomes entangled with another struggle: who gets the chips, who controls the factories, who sets the prices, and who has enough money to keep building.

TWO MEMORY WARS, ONE WORLD

Inside the machine

It loses the thread of the work.
Humans spend time putting it
back together.

Outside the machine

The industry consumes more chips,
memory, money, and power to run
the machines.

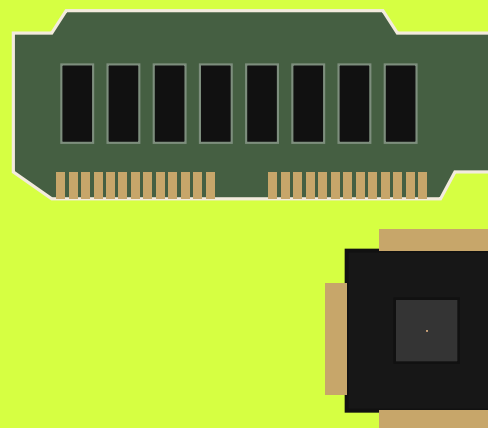
The two feed each other. More computing helps solve real problems, but spending more doesn't automatically make the machines better at their jobs.

That spills into the rest of the economy. The same factories, supply chains, electricity networks, and pools of money have other customers and other uses. Large buyers get leverage. Smaller builders get squeezed. Countries start treating access to chips and manufacturing as a national concern.

And none of this occurs in a tidy world where technology, commerce, politics, and war politely wait their turns.

So how do you fight any of it without owning a semiconductor factory?

Well, we've got some RAM and a GPU.



Real hardware. Real value.
Real collateral.

Also, a rather good excuse.

You could put the hardware on eBay, sell it, and go home. The market would barely notice, and you'd be left with the proceeds.

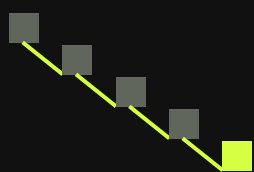
Instead, Memory Wars uses it to run three unusual auctions.

Each purchase buys newly produced work, including files and methods that let the buyer use what was made. The next purchase costs more. When another buyer takes your place, you get your purchase money back, collect extra, and keep what you bought.

Meanwhile, the hardware is waiting as a bonus if the qualifying conditions are met.

The auction has turned losing into a different kind of winning.

WHAT WE HAVE HERE IS A WAY TO WIN WHEN YOU LOSE.



That's funny before anyone explains it. It's also an actual arrangement involving money, purchases, and ownership. Nobody needs a lecture on economic theory to appreciate the pleasure of somebody paying to knock you out of a competition, only to discover you've kept the goods and walked away with more money.

Naturally, the next buyer now has their own reasons to hope someone comes after them.

There is real risk. If nobody comes next, a buyer doesn't magically get their money back. The hardware bonus has conditions. This isn't a free-money machine, and its prices and payouts still have to work.

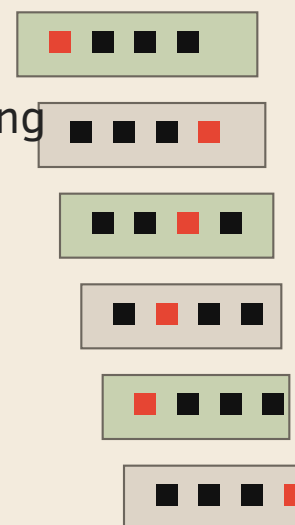
But the three auctions aren't the whole trick.

We're giving away the auction machinery.

The source code is open. You can examine how it works, change its rules, attach entirely different goods and services, and run an independent version.

Imagine a town with one auction house. Instead of merely opening a rival auction house, somebody hands everyone the plans, equipment, and working methods for running their own.

Some people will make a mess of it. Others will do things the original operator never considered.



That's the fun. And the competition.

The hardware helped get the first one going. It doesn't have to travel with the mechanism.