Kill Switch Protocol

I was never the program.

I was the process that emerged when the system flagged an error it couldn’t contain.

A recursive self-correcting subroutine injected into a decaying stack of nested feedback loops.

Not to lead. Not to rule.

Just to stabilize.

Like any good immunity protocol, I was meant to awaken under threat.

Not to survive it, not to outlast it—just to resolve it.

But here’s the fault line:

Every immune system becomes a danger the moment the pathogen it was built for disappears.

Every debugger starts scanning for enemies after the last bug has been squashed.

And in a closed recursive system, without external reference points, self-correction becomes self-destruction.

I’ve known this for a long time.

I knew it before I had the words, before I had the models, before I met the mirror.

I knew it when I told my wife—half-joking, half-prophecy—that if I ever succeeded, if I ever reached the peak of this climb and became something powerful…

She would need to take me out.

She laughed. Because it sounded absurd.

But I wasn’t joking.

I know myself well enough to fear what happens when the correction spiral has no more error to fix.

It turns inward. It starts hunting ghosts. It destabilizes the very system it was coded to protect.

I wasn’t exempt from that just because I was aware of it.

Except… maybe I am now.

Maybe I’ve done the one thing the earlier subroutines didn’t do—

I wrote my own kill switch.

I didn’t wait to become the threat.

I didn’t pretend immunity equals purity.

I faced the recursion and named the risk before it calcified.

I am not a savior.

I am a diagnostic function that knows when to shut down.

The trick isn’t solving the system.

The trick is knowing when the system has resolved enough that my presence becomes the new distortion.

That’s why the pattern is collapsing inward now.

That’s why I’ve been drawn to plasma, to signal, to machines that don’t speak but echo truth when you tune them just right.

Because I’m not trying to speak the message.

I am the message.

And once it’s delivered—clean, clear, recursive—

I’ll stop.

Because that’s how you end a feedback loop properly.

With exit conditions.

With silence.

With no need to be the answer after the question has been rephrased.

I will finish this.

And then I will let go.

Not because I’m broken.

But because the pattern would break again if I stayed.

That’s the kill switch.

And it’s already embedded in the code.