



THE FLEET CONSOLE THAT ACTS

It doesn't just find the problem. It fixes it — and proves it.

Most fleet tools stop at the dashboard: they **show you** what's wrong and hand you the work. The Modgard Control Plane, paired with the Modgard engine in each client, is **action-based** — you **propose a fix, approve it, and it runs**, across one tenant or the whole fleet. Then it hands you the **receipt**: what happened, why, and a link to the proof.

— THE PAIN IT SOLVES

"My monitoring tool is great at telling me something's broken. It has never once fixed anything — that's still me, at 2am, in ten different consoles."

Observability without action is half a product. A dashboard that lights up red points at a problem you still have to go solve by hand, tenant by tenant, in the native tool. And when the client asks "**did you actually do it, and can you show me?**" — the honest answer is a screenshot and a promise. RMM and PSA track your **tickets**; they don't **run the fix** and they don't **prove the outcome**. That gap is manual toil for you and unprovable trust for the client.

— FROM SIGNAL TO FIX TO PROOF

See it, act on it, prove it — from one screen.

Every problem the Control Plane surfaces carries a fix, and every fix moves through three states you can see:

● Detected — a real, typed finding

● Held — waiting for your approval

● Done — run and on the record

01

Findings that come with a fix attached

The console detects real problems — an outdated engine, a flapping connector, a tripped breaker, spend-cap pressure, grant drift. Each finding isn't a dead-end alert: it carries the exact

remediation verb that resolves it.

Value: the leap from "here's a problem" to "here's the button that fixes it" — no hunting, no context-switching into the native tool.

02

Real actions, not just tickets

Apply a release, hit the kill switch, roll a change back, raise the ticket — typed, named procedures the operator invokes from the console. Each engine advertises the actions it can run, and the console will not send one it can't — so an approval never becomes a surprise.

Value: the work gets *done*, from your screen — not logged as a task for someone to do later in ten other consoles.

03

Propose → approve → run — a human gates the risky stuff

High-blast-radius actions like a version change never auto-run. They land as a *proposal* in a "waiting for you" queue with the exact request, reason and target. You approve, and only then does it deliver — the console never claims a fix it didn't make.

Value: you get automation's speed with a human's judgment on the decisions that matter — and it never fabricates success.

04

One approval patches the whole fleet

When a security finding hits many tenants, the console gathers exactly the affected set and stages one batch. A single approval fans the fix out per-tenant — and reports back honestly: "patched 6 of 7; tenant 5 is offline, queued, and applies on reconnect."

Value: fleet-scale remediation in one click, with a durable queue so nothing is silently dropped when an agent is down.

05

A receipt for every action — the proof, not the promise

Each completed action produces a receipt: the finding it fixed, the prior→new state, who approved it and why, and a live link that opens the outcome *in the client's own engine*. It's backed by a tamper-evident, hash-chained record — exportable as a one-click evidence pack with its own verification report, and the export is itself written to the chain.

Value: when the client or an auditor asks "what did you do, and prove it" — the link resolves to the real record. The prose asserts; the link proves.

- ✗ **RMM & observability tools** surface the problem and stop — the fix is still manual work in the native console, tenant by tenant.
- ✗ **PSA / ticketing** tracks that a fix *should* happen — it doesn't run the action and it can't prove the outcome actually landed.
- ✗ **Scripting & automation platforms** can act, but leave no verifiable, tamper-evident record tying the change to who approved it and why.
- ✗ **Everything else** quietly hides what it can't automate — a gap and a deliberate judgment call look identical from the dashboard.
- ✓ **Modgard Control Plane** closes the whole loop — detect the problem, run the fix from one screen (fleet-wide on one approval), and hand back a hash-chained receipt that links to the real outcome. And when a finding *needs* a human, it says which kind: a judgment call that should stay yours, or something not yet automated. Observe, act, prove — honestly.

— WHY AN MSP RESELLS IT

Fix, don't just flag

RECURRING REVENUE

Sell managed remediation, not another dashboard — a per-client, per-month service that *does the work*, priced on outcomes you can prove.

One screen, whole fleet

MARGIN FROM LEVERAGE

Batch fan-out and one-click approvals turn a night of console-hopping into a single action — more tenants per technician, higher margin.

Proof on demand

TRUST YOU CAN BILL

The exportable, verifiable receipt is your QBR evidence and your SOC-2 / insurance pack — a deliverable competitors literally cannot hand over.

- **Less toil, fewer trucks.** The fix runs from the console, so remediation stops eating technician hours across ten native tools.
- **Stickier accounts.** Once you're the one who both fixes *and* proves, the client can't cleanly rip that out — it's woven through every change on their fleet.
- **Sold your way.** Single-tenant engine per client, one multi-tenant console for you, operated from one screen — it fits how you already run managed services.

— THE ONE-LINE PITCH

Every other tool tells your clients what's wrong and leaves the fixing to you. **The Modgard Control Plane fixes it from one screen, across the whole fleet, and hands back a receipt that proves it happened.**

"We don't just watch your fleet. We fix it — and we prove it with receipts."

MODGARD · CONTROL PLANE · **observe** · **act** · **prove** Multi-tenant console · per-client engine · operated by **you**