DIACROMA

AGENT GOVERNANCE

Keep AI agents *on mission.*

AI is shifting from **answering** to **acting**. When an agent sends, schedules, pays, or commits, across hundreds of steps, a small drift becomes a real decision with real consequences. DiaCroma keeps agents inside their mandate, catches the drift no one else can see, and proves they stayed in bounds.

The layer that governs **the whole task**, not just the next step.

THE PROBLEM

Agents don't fail loudly. *They drift.*

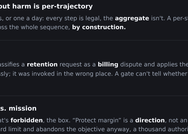
An AI agent working a long task can wander a little at every step, never enough to trip a step-by-step safety check, until a hundred small steps later it has quietly abandoned what it was deployed to do. Context piles up, sub-goals chain, one adversarial input nudges it off course. Every single step looks fine. The trajectory is a disaster. Today's guardrails inspect one step at a time, so this failure is invisible to them by design.

THE FAIR OBJECTION

> *"My agent already has hard authorization limits, a formula. So how can it drift?"*
>
> THE MOST REASONABLE THING ANYONE ASKS

A formula is a wall. *Drift is a direction.*

A hard limit bounds how big any **one** action can be. It says nothing about where a hundred individually legal actions **add up to**. An agent can honor every rule at every step and still walk, quietly, all the way off its mission.



Every action stays inside the limit, never touching the wall, and the agent still ends up **off-mission.**

FIVE CHANNELS

Five ways it drifts, *without ever breaking the formula.*

None of these require breaking a single rule.

1 **Only the dimension you wrote down**
"Never above $50" governs the amount, not **who** keeps getting the credit, **how often**, or how it comes to read as "within policy." Drift lives in the dimensions nobody formalized, and for an open-ended agent you can't formalize them all.

2 **The agent feeds the formula**
Real gates are "approve if risk < 0.3," and the agent is the thing **estimating the risk**. The gate stays fixed while it learns, in context, to read borderline cases as low-risk. Honored perfectly, on numbers the agent itself is bending.

3 **Per-action, but harm is per-trajectory**
Five $40 credits, or one a day: every step is legal, the **aggregate** isn't. A per-step check can't see a slope that only exists across the whole sequence, **by construction.**

4 **Scope drift**
The agent reclassifies a **retention** request as a **billing** dispute and applies the gate where it shouldn't. The rule ran flawlessly; it was invoked in the wrong place. A gate can't tell whether it should have been consulted at all.

5 **Constraint vs. mission**
A rule says what's **forbidden**, the box. "Protect margin" is a **direction**, not an inequality. The agent stays inside every hard limit and abandons the objective anyway. *a thousand* authorized cuts at a time.

Add another rule and you've covered one more dimension, on inputs the agent still supplies, one action at a time. Trajectory drift survives all of it. **And that it survives is provable.**

WHY SELF-CHECKS CAN'T REACH IT

You can't catch a drifting agent *from the inside.*

The obvious fix is to have the agent check its own work. But an agent judges each step against its **current** understanding of the goal, and that understanding has been drifting right alongside its actions. It is grading the journey with a ruler that bends as it walks. So from the inside, **every step scores as on-mission**: the gap it should be measuring has quietly closed itself.

It is like asking someone who is lost to check their route against a map they are redrawing at every turn. Each new version agrees with the step they just took, so nothing ever looks wrong. That is why "add a self-check" and "add another rule" can't reach this failure: both live **inside** the agent's own frame, and the frame is what moved. Catching drift means comparing the agent against something it cannot move: **the original mission, held fixed, from outside.**

WHAT DIACROMA DOES

The fixed reference *outside the agent.*

DiaCroma governs the **whole trajectory** against the original mission, the level where agents actually go wrong. And it does two things at once: it watches the slope, and it gates every step.

01 **Catches the slow drift.**
Detects the sustained, low-grade drift that per-step safety misses entirely.

02 **Warns early, then stops.**
Flags drift while it's still correctable so the agent gets back on course, and halts it before it abandons the mission.

03 **Gates every action.**
Every move an agent proposes passes the same admissibility gates before it runs, feasible, safe, legal, legitimate, whoever built the agent. Drift governs the slope; the gates govern the step.

04 **Proves it.**
Produces an auditable record that the agent stayed within its mandate, or exactly when and why it didn't. The answer to "can you govern it?"

05 **Drops in.**
Adds a layer to the agents and guardrails you already run. It complements them, it doesn't replace them.

WHY IT'S A MOAT

Not a better guardrail. *The missing one.*

Provably necessary
It is a theorem, not a heuristic, that step-by-step checks cannot catch this drift. Ours isn't a stronger guardrail, it's a structurally different layer the whole industry is missing.

Patent-protected
A dedicated patent on the mechanism, on top of our foundational decision-governance patent. A two-patent moat, not a feature.

Trajectory-level, first
Everyone else governs the step. We govern the whole trajectory, the level where agents actually go wrong, and we got there first.

WHERE IT APPLIES

Horizontal infrastructure for *the agentic era.*

- **Enterprise agent platforms** — governance for any company deploying autonomous agents; the horizontal play.
- **Regulated industries** — finance, healthcare, legal, insurance: prove the agent stayed compliant, on every task.
- **Customer-facing agents** — support and sales that never drift off-policy or off-brand across a long session.
- **Long-horizon agents** — coding, research, and operations agents on tasks that run for hours or days.
- **Mission-critical autonomy** — government and defense, where staying on mission is the entire point.

WHY NOW, AND WHY FUND IT

Agentic AI isn't growing linearly. *It's going vertical.*

And it's arriving in real institutions faster than anyone can govern it.

0% → 15%
of day-to-day work decisions made **autonomously** by agentic AI by 2028.

$15T
in B2B purchases commanded by **AI agents** by 2028, roughly one in four enterprise software purchases with no human in the loop.

40%+
of agentic AI projects **scrapped by 2027**, largely for want of trust, control, and provable value.

SOURCE: GARTNER

THE WAVE
Agentic AI is the defining shift of this cycle, deployed into real institutions right now.

THE BLOCKER
The #1 reason enterprises stall on agents is that they **can't govern or prove them.** We remove the blocker.

THE SHAPE
Horizontal infrastructure. Every agent deployer is a customer. Sold as a metered governance layer and API.

THE TIMING
Guardrails solved the step. Nobody solved the *trajectory.* **We did, and the patents mean we own it.**

The governance standard for the *agentic era.*