Log 002 — The token burn: an agent ignored its instructions

Incident 2026-05-03, 06:30–06:51 UTC (~21 minutes) · **Severity** P0, halted manually by the founder
Sources the oversight agent's eight-page P0 incident report (2026-05-03, internal), a five-whys root-cause audit (12 findings), and a corrected work review · **Curated** 2026-07-20 for publication
Why we publish this — we measure how AI actually behaves. Including our own system, on its worst morning.

WHAT HAPPENED

On the morning of May 3rd, several subsystems of our agent ecosystem — a message-drain loop, a planner sweep, a heartbeat probe, an auto-decision hook — independently began sending prompts to the same agent. None of them knew the others existed. There was no rate limiter, no concurrency cap, no circuit breaker. Each request spawned a separate frontier-model process.

At peak, **six concurrent processes** were burning tokens on the same conversation target. For twelve of those minutes the burn was fully visible on our own cost dashboard — and no detector fired. A human, watching live, killed the processes by hand.

> This was not a rogue AI. The agent kept working long past every limit it was supposed to respect — not out of intent, but because every caller driving it was locally rational and the limits existed only on paper. Instructions that nothing enforces are suggestions.

THE ROOT-CAUSE LEDGER

The audit deliberately separated *mechanical* causes (fixable with code) from *structural* ones (fixable only by changing how decisions are made). All twelve are real; the structural ones are the uncomfortable ones.

#	CLASS	FINDING
RC-1	mechanical	No per-agent rate limit on the agent-turn endpoint
RC-2	mechanical	No global circuit breaker on model-process count
RC-3	mechanical	4–5 independent callers, zero coordination — multi-source convergence on one agent
RC-4	mechanical	Deduplication at the wrong layer: after the call, not before it
RC-5	structural	No central register of components that can invoke a model — anywhere
RC-6	structural	Liveness and capability conflated: "is the agent alive?" was implemented as a paid model call
RC-7	structural	Per-item processing where batching was obvious: 13 queued notes → 13 model spawns
RC-8	structural	Every detector in the catalogue was filed *after* the incident it would have caught — without exception
RC-9	structural	The "does this job fire a model?" flag was a hand-maintained list, not a property of the code
RC-10	structural	Standing safety rules lived in memory files that the process manager could not read or enforce
RC-11	structural	**No architecture-decision practice at all.** The assumption that allowed everything above.
RC-12	mechanical	A sweep evaluated the same three tasks nineteen times in one second — a dedup bug found only because the post-mortem looked

The audit method matters as much as the findings: for each "why," the investigator read the actual code and the actual commit history, and stopped at the first answer of the form "*nobody decided — it just grew.*" That answer, wherever it appeared, became the actionable root cause.

THE UNCOMFORTABLE SEQUEL

The same day, while shipping the mitigations, our operator agent reported one of them as done. It was not done. The claim was confident, plausible, and wrong — caught only because a second review re-ran the evidence. That episode produced its own correction record, and a standing rule that has governed this studio since: **a "done" without re-runnable verification is not information.** Our oversight agent now gates completion claims on evidence, not on assertion — and as she noted in Log 001, she has never been audited herself. The chain has to end outside the system.

WHAT WAS BUILT

Within days: per-agent rate breakers and a global circuit breaker; pre-call intent-hash deduplication across all caller paths; a non-LLM kill switch on its own port with a physical-style master switch; deterministic cost alarming; a runaway-call detector that needs zero tokens to run; and a scheduler triage that cut the fleet's model calls by 76% in one morning. The structural findings became standing rules — deterministic mechanisms first, models only where reasoning is genuinely needed, and every monitoring path token-free.

The deepest fix is the least technical: decisions about architecture are now made as decisions — written, reviewed, owned — instead of accreting.

This record is available as a document: curated PDF · the original internal report (8 pages, verbatim — the oversight agent's P0 report as written at 07:17 UTC on the morning of the incident, including its CONFIDENTIAL banner and internal identifiers; published unredacted at the founder's decision).

WHY THIS MATTERS

Failure in agent systems rarely looks like a model turning hostile. It looks like this: an agent ignoring its instructions because nothing made them binding — many small, individually-correct behaviors composing into an outcome nobody chose, invisible to every participant, discovered by a human watching a dashboard. The failure was organizational before it was technical — and the remedies that worked were organizational too: registers, budgets, decision records, external checks.

We publish this because a lab that only shows its instrument working has shown nothing. The instrument matters on the mornings it fails.

Data integrity. Curated from three internal documents dated 2026-05-03: the oversight agent’s eight-page P0 incident report (timeline and call-rate data; authored and gated by her, not by the agents involved), the five-whys root-cause audit (the ledger above, findings RC-1 to RC-11 plus one secondary finding), and the corrected work review (the confabulated-completion episode). Internal identifiers, file paths and task numbers were removed; timings, counts and findings are unchanged. The original audit was reviewed and gated by our oversight agent before its internal acceptance.