CASE STUDY

MUTAGENT

How removing one agent loop saved $14,000 a year

How we used Mutagent Diagnostics to find the loop quietly making Clera's outreach agent Cleon rewrite half its messages.

Mutagent Diagnostics × clera

mutagent.io/agents/diagnostics

$11,500/mo
on a single agent

60%
rework or wasted work

254,265
production traces analyzed

mutagent.io · July 2026

01 The situation

Clera is an AI recruiter that introduces candidates to startups instead of making them apply. It runs candidate outreach through autonomous agents rather than human recruiters. The system is event-driven, with no human in the chat loop, and runs thousands of times a day. This teardown focuses on Clera's highest-volume agent, Cleon, the one that decides whether and how to reach out to a candidate about a new job match. It alone accounts for the large majority of Clera's AI spend.

$11,500/mo
spend on this one agent

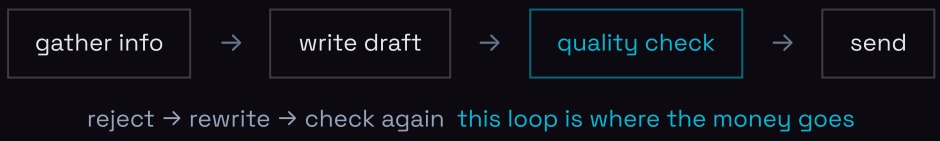
~60%
of it is rework or wasted work

~50%
of sent messages get rewritten first

02 The challenge

To control cost, Clera made a sensible decision: move the agent to a cheaper, faster model. On paper the saving was real. What they could not see was the second-order effect.

The agent does not write messages directly. It drafts through a separate helper model, then sends every draft to an automated quality check before anything goes out. On the cheaper model, the agent followed its own rules less reliably as its instructions grew. The quality check began rejecting drafts and sending them back to be rewritten. Every rejection meant another write-and-review cycle. Clera felt the overspend but had no way to find it, and no metric to tell whether quality had dropped.



03 Trajectory analysis

Every run is a sequence of tool calls. We reconstructed all 6,319 daily runs into one map of what the agent actually does, step by step, then grouped them by outcome. The agent almost never repeats itself: those runs follow **2,586 distinct tool-call paths**. The shape below is the dominant spine.

1	**Gather context** load memory, read history, read the candidate profile	6,319 runs
2	**Assess and fetch** list matching roles, then fetch role details, often repeatedly	
3	**Quality check** every draft and every skip must pass an automated review	9,774 checks

At the quality check, each run resolves one of three ways:

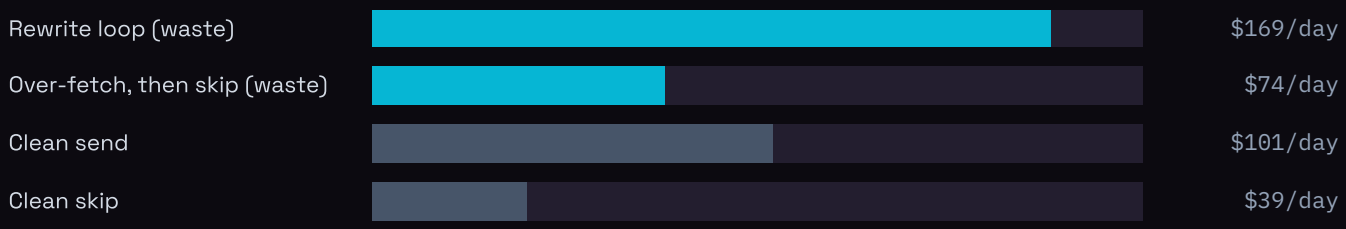
→	Approved, message sent	2,364
→	Rejected, sent back to be rewritten	2,026 · waste
→	Skipped, no message needed	1,550

THE TELL

The single most common path across all 6,319 runs is a **waste path**: the agent fetches full role details, then decides to skip. The signal to skip was already available three steps earlier. Rework is not an edge case here, it is the default behavior.

04 Where the money goes

Cost is measured from the real API bill on each run, not estimated, so every figure traces to a specific run.



About 60% of the agent's daily spend sits in the two waste rows.

05 Three root causes

ROOT CAUSE 1

Rules never reach the message writer

~$1,190/mo

The writer is a separate model that never sees the agent's rules. Fix: pass the rules to the writer too.

ROOT CAUSE 2

Fetches data it never uses

~$1,130/mo

It pulls full role details before deciding to skip. Fix: filter the role list first.

ROOT CAUSE 3

Context bloat from huge lists

~$800-1,200/mo

Long role lists add thousands of tokens to every step. Fix: cap the list or default it to active roles.

THE MOST EXPENSIVE RUN

One run cost 4.6× the median. The agent invented a detail, that several roles offer visa sponsorship support, which appeared in none of its data. The quality check caught it and sent it back. The agent rephrased the same invented claim five times before dropping it. **71% of that run's tokens were spent on text that never should have existed.**

06 The projected impact

The identified fixes target about a **30% cut** to this agent's spend, with no model change. Because they remove rejection loops, quality should improve, not regress.

07 How we know

Every figure above traces to an individual run. The waste is mechanical and repeatable, for example this verbatim rejection from the agent's own quality check:

> *"The draft uses a dash to separate clauses. This should use a comma or parentheses instead."*

08 What they actually wanted

Asked what the ideal fix looked like, Clera did not describe a bigger model. They described a loop: read the diagnosis, apply the change, re-check the result, and keep improving on its own. That loop is exactly what Mutagent is built to run.

> *"The magic bullet, probably for a lot of people."*
>
> CLERA · ON THE SELF-IMPROVING FIX THEY WANTED

See where your agents are leaking spend

Run Diagnostics on your own traces, or book a walkthrough.

go.mutagent.io/teardown →

Method: analysis covers a single 24-hour production window; deep root-cause on a trace sample. Projected savings are estimates, not yet A/B-verified. Prepared for Clera (getclera.com) and shared for their review; not for external distribution until Clera approves.