GPT-6.2 caching bug inflates agent bills as Gemini 4 Neon wins practitioners over

Coding Agents Alpha Tracker

2026-11-20

GPT-6.2 caching bug inflates agent bills as Gemini 4 Neon wins practitioners over

By Coding Agents Alpha Tracker • November 20, 2026

A Responses API bug stops GPT-6.2 Astra and Sol from caching system prompts; there is a workaround. Also: hands-on Gemini 4 Neon reports, Clay's case for harness work over post-training, and new tooling for governing what agents can do.

Check your GPT-6.2 harness for a caching bug

If you run GPT-6.2 Astra or Sol through the Responses API, the top-level `instructions` field is not being cached. Codex issue #62312 found it stays at 0 cached tokens on every turn, even though OpenAI's diagnostics wrongly report a cache hit [1]. The reported cost is large. Moving from GPT-6.1 Sol to 6.2 Sol dropped actual cache hit rates from 97.4% to 46.4%, so 64K–72K system tokens were billed at full price on every step [1].

Workaround: move the system prompt into a leading developer message in `input`. OpenCode already does this [1]. Peter Steinberger landed a fix in OpenClaw's embedded harness and warns that "most harnesses aren't updated yet" [2]. If you maintain your own harness, look at cached-token counts, not the diagnostics.

Gemini 4 Neon: hands-on reports

Last brief covered Neon's launch. This period brought firsthand usage. Matthew Berman built his open-source AI video editor, Scarlet, with Neon. He says the other frontier models keep needing "explain it more simply" follow-ups, while Neon both writes concise summaries and copes with "highly underspecified prompts" [3]. His examples: - **Screenshot plus one line.** He dropped in a screenshot of the model picker and asked for "make these one line and put the

disabled ones below the active ones." About 5 minutes later the change was done. When he asked for a mock, Neon built a live one with toggles, favorites and filters, and said it would include a live mock for UI changes from then on [3]. - **Filling in his intent.** Asked to "add Gemini models to Scarlet like Anthropic and OpenAI are," it spent 45 minutes and routed editing through the Antigravity CLI over agent client protocol, plus the Gemini API for spoken-note cleanup. Berman says that was exactly what he wanted [3]. - **Computer use.** It diagnosed an OBS profile mismatch, fixed the settings, and did not restart recordings that were already running so it wouldn't interrupt them [3].

Berman says Neon is weaker on 3D demos and games [3]. Riley Brown rates it "very good" for coding and video editing, but its "It's not this — it's this" habit is the worst he has seen, and he advises against using it for writing [4]. That conflicts with Theo's setup in the last brief, where Neon wrote all the copy. Theo now reports "insane results" on long-horizon engineering work but hasn't shared details yet [5]. He says Google is "so far ahead" on making the model easy to work with [6].

Clay: harness and interface work beat post-training

Clay's Jeff Barg, on LangChain's Max Agency podcast, gave the most reusable engineering lessons of the period: - **Post-training gave no step change.** Their trained model moved closer to Sonnet/Opus, but the team decided "there's a lot of juice" in harness engineering instead [7]. - **Make your domain look like what models were trained on.** Filter-composition structured output handled easy searches but missed long-tail queries. Clay replaced it with a SQL-like query language, which models generate well and which is also composable and modular [7]. - **Start small.** They began with off-the-shelf Pi, chosen because it is light and extensible. They haven't forked it and customize mostly through plugins [7]. The agent has about 20 skills and 8–10 tools, and 85%+ of calls go through bash [7]. The production agent uses the same CLI that external agents get [7]. - **Debugging long traces.** Asking Codex what went wrong in a raw trace with hundreds of tool calls worked "really bad." Having it first summarize each bash command worked much better [7]. - **Evals:** Clay started with capability buckets, then tried coverage buckets per launch milestone, which Barg found "almost meaningless," and settled on user journeys split into subtasks [7].

Astra and Sol: what they're good at

Berman credits Astra's computer use with fixing frame drops he had chased for 18 months. It read OBS logs during a live recording and forced on a buffer setting that "Auto" had left off [8]. Astra has powered Scarlet's video editing for weeks [8], although he rechecks Astra-built code with Opus [8]. In a same-prompt comparison he shared from another user:

Model	Time	Cost
Astra	45 min	$9
Sol	30 min	$2
Fable 5.2	3 h	$200
Opus 5.5	almost 3 h	$31

Berman judged Fable's output best, with Astra comparable [8].

Tools and releases

- **LangSmith Tool Gateway** (public beta) is a managed layer that handles credentials, identity and access for every agent tool call [9]. Setup: open Tools, add an MCP server or turn on a built-in, connect your accounts, pick the tools for a kit, and point any MCP client (including Claude Code or Cursor) at the kit's URL [9]. Search mode loads only the tools relevant to the current task [9]. Admins can deny tools or require human approval for sensitive calls [9].
- **Antigravity** desktop can now connect to remote servers over SSH [10].
- **Solid 2.0** errors include an error code, the exact location, a fix, and a path to a `skill.md` that tells your agent how to fix it. Some of the last scheduler bug reports came from agents [11].
- **JetBrains Air** (from a sponsor segment) runs Claude Code, Codex, Copilot and other agents side by side inside JetBrains IDEs, with access to tests, inspections and the debugger. Line-level review notes go back to the agent [11].
- **`treg /doomscroll`** is an open-source skill: give it a topic and a time window, and it finds posts across X, TikTok, YouTube, Reddit, HN and Product Hunt [12]. Repo: superdesigndev/treg [13].

Security and quality signals

The Pragmatic Engineer covered the Railway hack. An attacker using an abliterated model (one with its safeguards removed) planned a route to AI provider keys within six hours of downloading Railway's monorepo [14]. Railway's fixes: - each build gets its own throwaway VM - fine-grained access control for CI secrets, with 2FA or biometrics for sensitive credentials - security designed on the assumption that attackers already have the source code [14]

Greg Brockman posted "Codex for securing Ruby," quoting a researcher's universal pre-auth Rails RCE. Neither post explains how Codex was used [15, 16].

Sourcegraph's 2026 Big Code Report says 54% of teams using AI find that more than 60% of AI-generated work needs significant rework, and only 10% say their agents share a common view of the codebase [17].

Sources

1. X post by @bebuu_ai
2. X post by @steipete
3. Gemini 4 Neon is ridiculous…
4. X post by @rileybrown
5. X post by @theo
6. X post by @theo
7. Inside Clay's GTM agent rebuild
8. What's going on with GPT-6.2?
9. LangSmith Tool Gateway: Secure, governed tool use for agents
10. X post by @antigravity
11. The only software that didn't break this week…
12. X post by @jasonzhou1993
13. X post by @jasonzhou1993
14. The Pulse: What can we learn from the Railway hack?
15. X post by @gdb
16. X post by @stdoutput
17. X post by @Sourcegraph