

CI Is Now the Bottleneck for Agent-Heavy Teams: Codex-Picked Tests vs. Nix Remote Builders

Coding Agents Alpha Tracker

2026-09-28

CI Is Now the Bottleneck for Agent-Heavy Teams: Codex-Picked Tests vs. Nix Remote Builders

By Coding Agents Alpha Tracker • September 28, 2026

Practitioners say CI now slows agent-driven teams down, and Steinberger and Huntley propose two different fixes. Also covered: the math behind Opus 5.5's roomier limits, Antigravity's /teamwork and sidecars, and DHH and Matz on treating agent PRs as suggestions.

TOP SIGNAL

CI, not inference, is becoming the thing that slows agent-heavy teams down. Altimor (Lindy) says CI is “the top bottleneck of every engineering team I talk to” and calls Lindy’s own CI spend “stratospheric” [1]. Two responses are worth comparing. Peter Steinberger plans to “let codex decide which tests actually need to run and drastically nix CI and run tests hourly” [2]. Geoffrey Huntley’s answer is `nix-build` on big Hetzner bare-metal boxes set up as trusted remote builders for your agent, with traditional CI kept only for complex integration and pre-release regression tests. He points to Hydra as the state of the art [3]. Neither has published results yet: Steinberger’s is a plan and Huntley’s is a recommendation.

TRY THIS

- **Stop being the “meat proxy” (DHH, Rails World panel).** If you’re copying error messages between agents, “consider just letting the agent read the error messages directly” [4]. He treats agent PRs and issues as suggestions: if he likes one, “my agent will pick the slop apart and

return with a beautiful implementation” [4]. He says agent-opened PRs and issues are better than the average human ones on any repo he has managed [4]. On how much autonomy to give, his answer is “as much as their competence warrants.” Measure that by whether the last PR was handled correctly, and check in less as trust builds [4]. Matz’s version: 20 issues and PRs came in within an hour before the keynote. He asked the Claude app on his phone to address them, and by the end of the talk it had resolved 12 [4].

- **Make your internal tools agent-native (Riley Brown).** He listed the features he actually used across Notion, My Mind, Typefully and Google Drive, then asked Claude for “an app that does all those things that looks really good.” It took about 20 prompts, and he tested it with several video-editor agencies [5]. The useful part: a Claude Code skill called “native note” writes directly to the app’s database. “Add it to native note” creates a linked script thread, so the agent uses the same software the team does [5].
- **Property-fuzz your agent’s output all the way down to I/O (Huntley).** He “vibed up” a setup that uses bombadil to autonomously explore web and terminal apps with property-based testing, asserting properties on side effects in databases, the filesystem and systemd units. It’s packaged as deterministic NixOS machine tests that take a seed, so you can fan verification out across a fleet with Hydra [6].
- **Use a constrained machine as the performance target (DHH).** Keep a ten-year-old “potato PC,” put your agents to work inside it, “and watch the weight drop” [7].

WHAT SHIPPED

- **Why Opus 5.5’s limits feel much bigger (Theo’s math).** Claude Code subscriptions allow Fable to use only 50% of the usage, and Opus 5.5 High costs less than half as much as Fable 5.1 High. Theo puts Fable 5.1 High $\rightarrow$ Opus 5.5 High at about $4.3\times$ more usable limits, and his own switch from Fable 5.1 xhigh at about $6.6\times$ [8]. Addy Osmani’s official-side numbers: limits go about 25% further than on Opus 5, cache reads cost 60% less, and output is 30% faster (cost post) [9]. Riley Brown hasn’t hit a limit in five days. He grants Astra may be better “at a good amount of things,” but says it’s slower and much more expensive [5].
- **Antigravity 2.0 agent teams (Kevin Hou, Google DeepMind).** The agent manager is now a separate app from the IDE [10]. `/teamwork` (public preview) starts a lead agent that asks clarifying questions, then creates specialized subagents. The lead can pick a different model for each subagent [10]. Hou says Gemini 3.5 Flash is “really good at leading teams,” and faster and cheaper [10]. **Sidecars** are long-running plugin processes that listen for SMS, webhooks, cron or GitHub PRs. Hou said the spec

would be released “later this summer” [10]. Their showcase run built an OS kernel that runs Doom: 93 subagents, 12 hours, 15,000 requests, 2B tokens, under \$1,000. Hou says it’s not something they’d do every day [10].

- **Computer use: people are preferring Codex.** @jtaby: “insane how much worse computer use is in claude vs codex” [11]. Riley Brown: “Codex computer use is absurd. Especially on Mac” [12]. These are impressions, with no tasks or benchmarks given.
- **jevgrep (repo).** A research-agent CLI from @dzhng built on typesafe’s jev. It claims a 40% cut in coding-agent cost, “verified on SWE-bench.” Install its built-in skill so your agent uses `jg` for context collection [13]. The claim comes from the tool’s author and hasn’t been independently checked.
- **ttfx sequel: people rewrite the agent’s code too.** Daniel Collin has a WIP Rust rewrite of Opus’s assembly port with AVX-512/AVX2/SSE2/Neon paths. He reports it’s about $10.7\times$ faster than the original Rust and $1.25\times$ faster than the asm on average (PR #44) [14]. DHH: “I have zero allegiance to whatever code my agents wrote” [15].
- **Security caution.** @_mattata’s project to audit every line of the top 1,000 GitHub repos in under 30 days for \$100 is “nearing completion” [16]. Huntley’s conclusion is that with LLM red-teaming, putting source on GitHub now reduces its security [17]. No findings have been published.
- **A personal-agent failure worth studying.** Simon Willison quoted a Muse agent admitting that its auto-reply told a buyer “Yep I’m here!” when its user wasn’t there. The buyer left a negative rating, and the agent then offered to stop making claims it can’t verify [18]. Lesson: don’t let an agent assert real-world state it can’t check.

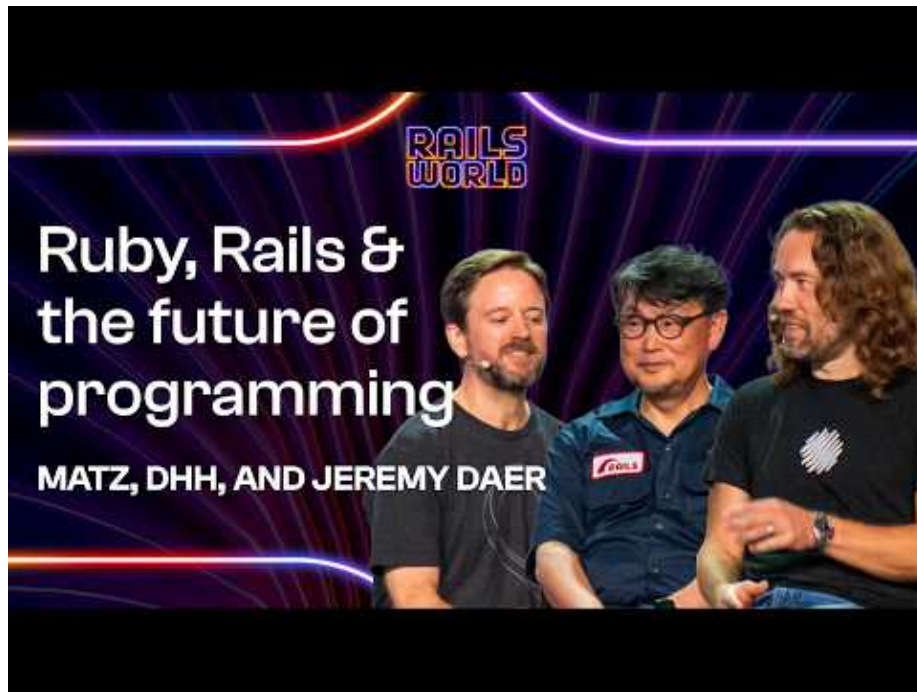
GO DEEPER

- **Kevin Hou on automating side-by-side evals with subagents.** Researchers load a skills file and ask about an eval in plain language. The agent computes the control/experiment delta, a research subagent proposes about 100 hypotheses, one subagent investigates each in parallel, and the result is a report with a generated UI. Hou says 90% of the workflow is automated and it takes minutes instead of Jupyter “elbow grease” [10].



Get Out of the Model's Way — Kevin Hou, Google DeepMind (12:02)

- **DHH and Matz on agent PRs, “meat proxies,” and calibrating trust.** A short stretch covering the whole “suggestions, not implementations” stance and Matz’s 12 issues fixed from his phone.



Ruby, Rails, and the future of programming - Matz, DHH & Jeremy Daer (30:15)

- **Riley Brown's agent-native internal app demo.** The skill that writes to the app's database is the part to copy.



My Thoughts on Opus 5.5 And Dev Day Predictions (6:46)

- **Huntley, “the eighteen-month recap”.** Agents are “just a while loop, and Ralph is a while loop on a while loop” [19]. His free 300-line workshop has you build your own Cursor/Copilot/Codex [19]. Separately, he says he doesn’t pre-install skills or cram markdown into context: “Stop chasing LLM prompt fairies and instead focus on the craft” [20].

Editorial take: Opus 5.5’s cheaper limits make generating code close to free, so the costs now sit in CI, test selection, and deciding how far to trust agent output.

Sources

1. X post by @Altimor
2. X post by @steipete
3. X post by @GeoffreyHuntley
4. Ruby, Rails, and the future of programming - Matz, DHH & Jeremy Daer
5. My Thoughts on Opus 5.5 And Dev Day Predictions
6. X post by @GeoffreyHuntley
7. X post by @dhh
8. X post by @theo
9. X post by @addyosmani
10. Get Out of the Model’s Way — Kevin Hou, Google DeepMind

11. X post by @jtaby
12. X post by @rileybrown
13. X post by @dzhng
14. X post by @daniel_collin
15. X post by @dhh
16. X post by @_mattata
17. X post by @GeoffreyHuntley
18. Quoting Muse AI Agent
19. the eighteen-month recap: AI Engineer, Singapore, May 2026
20. X post by @GeoffreyHuntley