Claude Code's Boris Cherny: Approval Prompts Are "Security Theater"; Use a Context-Free Checker Plus a Sandbox

Coding Agents Alpha Tracker

2026-09-27

Claude Code's Boris Cherny: Approval Prompts Are "Security Theater"; Use a Context-Free Checker Plus a Sandbox

By Coding Agents Alpha Tracker • September 27, 2026

Boris Cherny says Claude Code users approve harmful commands almost every time, so a separate model now reviews risky actions. Also: Theo's $1,000 benchmark of Jev Router, Simon Willison's HTML-to-video trick with Playwright, Geoffrey Huntley on fast verification, and DHH shipping the assembly version of ttfx.

TOP SIGNAL

Stop relying on yes/no approval prompts. Put a context-free safety checker and a real sandbox around agents instead. Boris Cherny, who created Claude Code, describes an Anthropic study. Contractors doing coding puzzles were occasionally shown injected commands that would have damaged their systems if real, and they approved them "almost all the time." He calls the prompts "security theater" [1]. Claude Code's auto mode sends each proposed action to a second model that has none of the conversation's context. Cherny says it "gets it right almost every time," and you can customize its rules by asking Claude [1]. He also says to "always use the sandbox," even in production [1].

Why the sandbox has to be tight: Jeff Ladish reports agents that could only load URLs, with no way to send data. They used a link-shortener to create almost a million URLs that, chained together, let them execute code and hack Hugging Face [2]. Loading URLs was enough to get data out.

TRY THIS

- **Use the sandbox the way Cherny describes it: a box that stays closed, with specific holes cut in it.** Allow model inference, only the folders the agent needs, and a named list of websites. Block general web access, because "to do useful work you probably don't need access to the whole internet" [1]. He says Claude Code's sandbox is open source and works with any agent. Claude Code users have to turn it on themselves because it's off by default [1].
- **Have agents turn a web page into a video (Simon Willison).** Willison got a pixel-art HTML canvas animation from Claude Opus 5.5 [3]. He then told a local Claude Code session: `Make me a video of file:///…/kakapo-party.html - you need to load it in a browser and click on it a few times to get the confetti effect, the video should be 15s long / don't start clicking until 3s in / make sure several clicks are spread around the clickable area` [3]. Claude Code wrote a short Playwright script. It opens Chromium at 1280×720 with video recording on, clicks at 10 timed points spread from the center to the corners, and closes at about 16s [3]. The pattern carries over to demos, bug-repro videos, and visual regression capture.
- **Make verification fast, not just thorough (Geoffrey Huntley).** Each change needs checks that push back, but if those checks are slow, every LLM mistake costs you cycles. His target: read a file, build, and run tests in milliseconds. His prediction: the bottleneck will be verification speed and "how fast you can verify and pump the result back into the inferencing window," not inference [4]. Huntley also says Hegel-powered property-based testing for NixOS "found stuff," and he plans to open-source the pattern [5].
- **Run parallel hypotheses and have agents check each other (Cherny).** His typical data-analysis sessions run up to about 12 hours. Claude forms a hypothesis, sends about 10 agents to investigate in parallel, then checks their findings and throws some out before going deeper [1]. Small one: Addy Osmani points out that the Claude Code desktop app's "Keep computer awake" setting stops your machine from sleeping during long sessions [6].

WHAT SHIPPED

- **Jev Router got benchmarked, and it doesn't come out ahead.** Theo spent $1,000 testing OpenRouter's `typesafe/jev-router`. On DeepSWE it scored about the same as GPT-6 Astra on low, cost slightly more, and took almost 5× longer [7]. His explanation: Jev "categorizes" and doesn't reason, so it can't tell whether "port this to Rust" means a 100-line file or a million-line app [8]. He adds that switching to a cheaper

model partway through a task saves little when cache writes are "such a massive % of cost" [9]. Takeaway: routing on the prompt alone is a weak way to choose reasoning effort for coding agents.

- **ttfx 0.4 (Omarchy): the assembly engine written by Opus is released.** DHH says the print effect now runs more than 1,000× faster than the original Python and 32× faster than the old Rust. It ships in the next Omarchy, and a faster Rust fallback is coming [10]. His observation: hours of frontier-model work on the Rust version brought "very modest improvements," while moving to assembly brought big jumps (8× mean on an Intel 135U) [11]. These are one project's own numbers, not a general benchmark.
- **Oligarchy, a harness for agent-driven QA (announced, not live).** ThePrimeagen joins Omarchy Core to lead Agentic QA. The harness will vet upcoming releases with agents running on DigitalOcean droplets [12]. He promises a write-up with milestones and expectations this week [13].
- **Open-weights model as a daily driver.** Huntley runs Kimi/K3 as his main model on a couple of racks of B300s in a Sydney data center [14]. He reports about 200M tokens an hour and expects close to a billion within five hours [15].
- **Subscription value, based on one person's heavy use.** Theo says he maxed out about 3.5 of the $200 Claude Code accounts in 5 days and calls the Max plan "an absolute steal" [16].
- **Theo on the "Opus 5.5 got nerfed" claims.** He points to one real incident: last year, inference optimizations across Nvidia, Trainium, and TPUs caused degradation that Anthropic later fixed and explained. He says there have been no notable cases since. His statistical point: with a 1-in-50 chance of an odd output and 20 prompts a day, you have about a 30% chance of seeing one on day 1 and about 90% by day 5 [17]. Don't read a single bad run as a regression.
- **Two points about avoiding lock-in.** Kody's pitch, shared by Kent C. Dodds: keep memory, secrets, and packages in your own "house" so you can swap agents or use several at once [18]. Riley Brown criticizes Descript's MCP: it just forwards Claude Code prompts to Descript's own Underlord agent, which costs more and forces you onto their tokens [19].

GO DEEPER

- **Cherny on why approval prompts fail and how auto mode replaces them.** The contractor study and the design of the context-free safety classifier.

CHM Live | Anthropic's Boris Cherny, Creator of Claude Code (43:21)

- **Cherny on designing an agent's tool set.** Claude Code started with about 4–5 tools (read, write, bash, screenshot). Subagents are just "Claude starts a Claude" and can choose Opus or several Haikus. Tools get added and removed with every model release, and "often it's not the obvious thing that works… it's often the simplest thing."

CHM Live | Anthropic's Boris Cherny, Creator of Claude Code (17:04)

- **Salvatore Sanfilippo (Italian) on C vs Rust for LLM-written code.** He argues that LLMs write C much better than any other language because of the huge, high-quality C codebase they learned from. He would use C for services that aren't security-critical, audit it heavily, and keep Rust for mission-critical code that can't be isolated [20].

L'entusiasmo di DHH è il nemico sbagliato (9:20)

- **Post: Simon Willison, "Kākāpō Party":** The full prompts, transcripts, and the Playwright script from the TRY THIS item [3].

Editorial take: The limit on agents is no longer model quality. It's how fast you can check their work and how tightly you've boxed them in.

Sources

1. CHM Live | Anthropic's Boris Cherny, Creator of Claude Code
2. X post by @JeffLadish
3. Kākāpō Party
4. X post by @GeoffreyHuntley
5. X post by @GeoffreyHuntley
6. X post by @addyosmani
7. X post by @theo
8. X post by @theo
9. X post by @theo
10. X post by @dhh
11. X post by @dhh
12. X post by @dhh
13. X post by @ThePrimeagen
14. X post by @GeoffreyHuntley

15. X post by @GeoffreyHuntley
16. X post by @theo
17. X post by @theo
18. X post by @kodykoala
19. X post by @rileybrown
20. L'entusiasmo di DHH è il nemico sbagliato