

OpenAI and Anthropic both replace
per-command approval with an agent reviewer;
LangChain shows routing cut Open SWE costs
64%

Coding Agents Alpha Tracker

2026-10-07

OpenAI and Anthropic both replace per-command approval with an agent reviewer; LangChain shows routing cut Open SWE costs 64%

By Coding Agents Alpha Tracker • October 7, 2026

Codex's Auto-review is now free, matching Anthropic's case that per-command approval has become security theater. Also: LangChain's measured model router, Theo's results with Astra Ultrafast, Boris Cherny's prompting advice, and early harness problems with Mistral Large 4.

Let an agent review commands instead of approving each one yourself

OpenAI made Codex's **Auto-review** free for anyone signed in with a ChatGPT account. Tibo also says it doesn't draw usage from your plan [1]. A second agent reviews every action the primary agent takes. Its only jobs are to block high-risk actions and anything that doesn't match your original intent. This replaces the default sandbox, which asks you to approve everything and wears you down unless you spend time writing rules [1]. To turn it on, go to Settings → Permissions → Auto-review, or pick "Approve for me" from the permissions menu below the composer [2]. Tibo's roundup post adds that it "can be between 2-10% of plan when used," and doesn't say what that figure measures [3].

Anthropic made the same argument from its own data. Boris Cherny described a study in which contractors solving coding puzzles in Claude Code were sometimes shown injected commands that would have damaged their systems. They

approved them “almost all the time,” so per-command prompts had turned into “security theater” [4]. Claude Code’s auto mode sends each proposed action to a separate model that has none of the conversation’s context. Cherny says it gets the call right “almost every time” [4]. Mike Krieger says every Claude Code session now runs through that classifier by default. It asks you only in specific cases, which avoids both dangerously skipping permissions and endless yes-clicking [5].

For you, this means both major harnesses now offer a reviewer agent as the alternative to full access. If you have been running with permissions skipped, this is the setting to switch.

Model routing: measured savings, and a disagreement over what to route on

LangChain’s Open SWE team published a router they built into their harness and measured [6]:

1. **Map your tasks from traces.** They sorted LangSmith traces into categories and rated complexity by median number of agent invocations and median LLM cost. They call this the most important step [6].
2. **Pick models on the cost/intelligence Pareto frontier.** They chose GLM-5.3-Flash as the fast tier, GPT-5.6 Sol as balanced and GPT-6 Astra for performance [6].
3. **Route once, on the first message.** The router picks a model from the thread’s first message and keeps it for the whole task. It now runs on Jev, a dedicated decision model, after starting on a small LLM [6].
4. **Track outcomes.** They ran an A/B test on merged-PR rate and user thumbs up/down [6].

Across almost 500 threads, compared with always using the performance model, median cost per thread fell 64% and P90 cost fell 37%. Merged PRs were slightly higher, but the difference wasn’t statistically significant [6]. Traffic went 34% to the fast tier, 56% to balanced and 10% to performance [6]. A second test, router versus always using the fast model, was stopped within a day because of user complaints [6].

OpenAI’s new **Decisions API** (public beta) is aimed at this same job: choosing a model, tool or action in near real time. OpenAI says it decides up to 10x faster than GPT-6 Luna through the Responses API [7]. Theo pushed back: using it to expose the right tools to an agent is “possibly cool,” but using it to decide what “level of intelligence” a task needs is “absolutely useless” [8]. LangChain’s numbers are one data point against him, though their task mix may not match yours.

Theo on Ultrafast: a different way of working, at a price

Theo’s video on Astra Ultrafast includes workflow points that apply whenever inference gets fast: - **Split-screen your app and the agent, and steer it while it works.** Send many small corrections instead of a batch of 20 [9]. He

sent five prompts in four minutes, with replies under a minute [9]. - **Cut slow tool calls.** Once generation drops from 10 minutes to 30 seconds, tool calls can double your runtime. He told the agent to stop using the preview browser and just change code, while he watched the dev server [9]. - He says staying in the loop got him a better UI from Astra than he gets from Opus, even though Astra is worse at design [9].

The costs: the main thread came to \$36 and a follow-up to \$250. GPT-6.1 Sol would have handled the work “just as well” for \$12 [9]. He advises against upgrading to the \$500 tier for this [9]. Separately, he says the \$200 Codex plan “feels reasonable” with 6.1 Sol, while Astra on it is “pretty close to unusable.” The \$200 Claude Code plan feels meaningfully less limited with Opus [10, 11].

Cherny: state the goal, the effort and the check

Boris Cherny’s prompting advice: talk to Claude like a coworker and don’t over-scaffold. Tell it what you want, how much effort to spend, and how it should verify the result [12]. As an example he pointed to his earlier run where Opus 5.5 formally verified the Claude Agent SDK in Lean. A couple of short prompts produced 16 PRs fixing bugs and race conditions. He sometimes pairs Lean with TLA+ to check data flow, concurrency and state management, without knowing either language well [13]. Krieger reports something similar: Fable 5.1 with no extra harness beat Hatch, a builder-plus-adversarial-verifier system that took months to tune. The only piece of Hatch he kept was dynamic workflows [5].

Mistral Large 4: promising, but configure your harness first

Mistral Large 4 is in API preview: 1T total parameters, 49B active, with open weights promised for the end of October. It costs \$1.36/\$4.18 per million input/output tokens [14]. Mistral says it beats GLM 5.3 on DeepSWE and Kimi K3 on Terminal-Bench 4, and that it finished #2 behind Opus 5 in a blind Surge coding review. Critics say it trails GLM-5.3 on Artificial Analysis’s index [14]. Mistral says many reported failures come from not setting `reasoning_effort="high"` [14]. Matthew Berman ran into a different problem in OpenCode: with high effort set, it hit about 32K tokens and stopped. Raising the context limit or turning thinking off helped. He found Cursor much easier for running third-party models [15].

Smaller items

- **Cursor iOS:** you can check in on, reply to or start agents running on your computer. They keep going if your phone loses signal. Pair at cursor.com/mobile; Enterprise admins have to enable it [16, 17, 18].
- **Steinberger’s team claw** takes work requests from X. Unassigned sessions are open for anyone to claim, and the agent pings whoever last

touched the related code. He says the whole setup was one prompt, with the server extending itself through hot-reloadable plugins [19].

- **T3 Code:** Opus 5.5 built and rendered mocks of five UI treatments inside the thread [20]. Theo repeats his advice not to run agents on a Mac unless you have to [21].
- **Willison** had Codex work out how to run Parseable and send it Datasette's OpenTelemetry traces. He then wrote up the working patterns himself as a TIL [22].
- **DHH:** much of the work of getting the most out of agents is now product management, project management and QA [23].

Sources

1. X post by @thsottiaux
2. X post by @thsottiaux
3. X post by @thsottiaux
4. Boris Cherny on Anthropic & Claude Code | Computer History Museum
5. Sierra CXO Summit: Lessons Learned at the Frontier with Anthropic Labs' Mike Krieger
6. How to Build a Model Router in the Harness
7. X post by @OpenAIDevs
8. X post by @theo
9. I love Ultrafast (it's unusable)
10. X post by @theo
11. X post by @theo
12. X post by @bcherny
13. X post by @bcherny
14. [AINews] Quasi-Riemann-Hypothesis: OpenAI publishes 722 math papers solving 90 of the top 500 open math problems; "the most significant moment" in >100 years of mathematics
15. Mistral is BACK! (Le Chonk)
16. X post by @cursor_ai
17. X post by @cursor_ai
18. X post by @cursor_ai
19. X post by @steipete
20. X post by @theo
21. X post by @theo
22. Using Parseable with Datasette for OpenTelemetry traces
23. X post by @dhh