

Verification is the bottleneck: Claude Code’s 90% self-authorship, Huntley’s Antithesis loops, and LangChain’s agent cost controls

Coding Agents Alpha Tracker

2026-10-18

Verification is the bottleneck: Claude Code’s 90% self-authorship, Huntley’s Antithesis loops, and LangChain’s agent cost controls

By Coding Agents Alpha Tracker • October 18, 2026

Anthropic credits verification for letting Claude write about 90% of Claude Code’s code, and Huntley is putting Overalls through two weeks of autonomous verification. Also covered: LangChain’s method for controlling coding-agent spend, Huntley’s composable inference proxies, credential hygiene, and smaller tool updates.

Verification, not review, is what lets agents write most of the code

Anthropic says Claude writes about 90% of Claude Code’s code. From June to September, merged PRs more than doubled while the share of PRs reverted fell by more than 40% [1]. **Addy Osmani** puts this down to one practice: give the agent a way to verify its work. When code is produced faster than people can review it, the bottleneck becomes proving that it works [2].

His recipe:

- **Replace the single reviewer with layers.** One person reading line by line can no longer be the gate. Use a multi-layered system with fast, deterministic feedback loops for agents: end-to-end tests, feature maps (which poteto advocates), property tests and multi-agent adversarial reviews [2].
- **Move humans to judgement.** People should focus on product intent, assessing risk and weighing evidence. Leave code checking to automated tests, scale approvals to risk, and turn test failures into living evaluations

[2].

Anthropic’s write-up is here [1]. One caution from the field: in a Reddit thread where a developer said Opus 5.5 cut their tickets to 1–2 hours a day, a commenter said manual review on their team still catches “significant issues, including data loss and security issues” [3].

Geoffrey Huntley is testing the same idea under stress. His starting point: just because an application compiles doesn’t mean it works. Overalls has gone into Antithesis’s “torture chamber”, where he will run autonomous discovery-and-rectification loops for two weeks until his New York talk. He wants to see what classes of problems turn up and whether agents with “the right level of back pressure” can fix failing properties on their own [4]. He also flagged a slide deck on formal verification of LLM-generated code [5].

He also proposed using thinking traces to check whether a skill or AGENTS.md is working: whether the agent considers the skill, and when it decides not to use it. He notes this isn’t possible with Claude [6].

Controlling coding-agent spend: LangChain’s three steps

Harrison Chase guesses LangChain spends roughly 30% of its engineering salaries on coding tools. That share has fallen over the last two months even though token use went up [7]. His steps:

1. **Visibility.** Log all coding-agent usage to LangSmith so you can see who spends what, and on what [7].
2. **Basic controls.** Set budgets with a way to escalate: the VP of engineering almost always raises the limit, but the budget guards against agents stuck in bad loops [7].
3. **Own the harness.** Their in-house cloud agent, which Chase starts from Slack, has prompt caching and model routing built in [7]. Providers now let you add tools mid-run, which used to break the cache [7].

The example that justifies tracking: a plugin sent a DOM snapshot back on every agent turn during frontend work, and that one loop cost about \$40k in a week [7].

Prompt caching is also behind **Theo**’s claim that Gemini is about 50% more expensive than it should be. His test showed a 27% cache-miss rate [8], using the same basic agentic-work test on all three APIs [9]. A respondent said their own miss rates are much lower [10]. Theo says the test is about 10 lines of code and published his methodology [11], so you can check your own workload.

Huntley’s composable inference proxies

Huntley has lost time to “model is at capacity” errors that start when Europe wakes up, so he released three pieces of software that can be used separately or together [12]:

- **Underclass** pools ChatGPT subscriptions behind a proxy, claims usage resets automatically and handles the prompt cache key so usage isn't wasted. He calls it stable, runs it with six ChatGPT 500 accounts, and it's on GitHub and available as a Nix Flake [12].
- **Preflight** is a local proxy that scans requests and attachments and redacts secrets before they reach the provider. Point Underclass at Preflight to use both [12].
- **geoclass** is an experiment. Tag VMs around the world on a Tailscale network, point Underclass at geoclass, and requests get routed across regions. "In theory" the rest just works [12].

Early results are anecdotal. Claude found no header in the Codex source or OpenAI docs that pins requests to a region [13]. After routing out of Germany via Australia, Huntley says OpenAI inference felt much better, but he hasn't measured it and it was a Saturday afternoon [14].

Keep secrets where agents can use them but not read them

A user blamed a \$90,000 Google bill on a leaked key, and said Opus 5.5 found that Codex had published their `.env` to Cloudflare Pages. They say Codex itself flagged the chat as a cybersecurity risk [15]. This is their account and hasn't been verified. **Kent C. Dodds** said he feels safer because he uses cloud agents and keeps secrets and OAuth tokens where agents can use them but not retrieve them [16]. He also restricts tokens to specific actions, for example Gmail drafts only [17]. Preflight, above, is another way to do this.

The opposite complaint also surfaced: auto mode in dots and Claude Projects refuses or misclassifies basic tasks, such as merging PRs into a repo set up for that purpose. This blocks work overnight or demands approvals you can't give remotely [18]. Riley Brown agreed.

Smaller items

- **Claude Code latency:** it now gzips the conversation it sends each turn, and cloud sessions start up as soon as you begin typing [19]. Theo says responses now start almost immediately, and the difference is most noticeable when steering, because you can watch the model correct itself [20, 21].
- **cc-stretch** is an MIT-licensed Claude Code plugin. Replies start compressed and you expand sections as needed (it claims 83% fewer words read), and it tells Claude which sections you expanded. Install with `/plugin install cc-stretch --marketplace pro-vi/cc-stretch` and run `/tui fullscreen` [3].
- **Consistent harness runs:** OpenHands looks stable mainly because it hardcodes temperature 0. Normalizing tool outputs fixes much of the variance in other harnesses [3].

- **Haiku 5.5** costs \$0.10/\$0.50 for prompts up to 100K tokens and is reported at 39.2% on Terminal-Bench 4.0 at max effort [3].
- **Linus Torvalds** used an LLM to refactor hard-to-follow associative-array code into smaller functions. Userspace tests confirmed the reported bug and found three more, all dating to 2013, according to an Italian video recap [22]. In the same video, Salvatore Sanfilippo says AI helped him write Redis patches that cut RAM use on large sorted sets by 50% [22].
- **Email as an agent channel:** Ben Tossell’s bot opens BB app threads for coding work, emails updates, and updates a shared task list and work log [23]. He says email threads work well for agent chats [24].
- **Riley Brown** edited a recorded video in one prompt with Opus 5.5, Blender and a skill he had worked on for a day. He rates it about 3% as good as it could be [25].
- **tsc-ts** from Vercel Labs ports TypeScript back from Go to TypeScript, compiles it with scriptc, and claims 2–4x faster than tsgo [26]. Theo says it beats his tsc-rs port in most things [27].

Sources

1. X post by @ClaudeDevs
2. X post by @addyosmani
3. [AINews] High Taste Testers Ep. 1
4. X post by @GeoffreyHuntley
5. X post by @GeoffreyHuntley
6. X post by @GeoffreyHuntley
7. The Future of Inference Market & Model Routing: Commodity or Increasing in Value? | LangChain CEO
8. X post by @theo
9. X post by @theo
10. X post by @_mohansolo
11. X post by @theo
12. pooling OpenAI subscriptions and bypassing regional inference capacity limits
13. X post by @GeoffreyHuntley
14. X post by @GeoffreyHuntley
15. X post by @JustLingonberry
16. X post by @kentedodds
17. X post by @kentedodds
18. X post by @tenobrus
19. X post by @ClaudeDevs
20. X post by @theo
21. X post by @theo
22. Linus Torvalds promuove l’AI sempre più
23. X post by @bentossell

24. X post by @bentossell
25. X post by @rileybrown
26. X post by @ctatedev
27. X post by @theo