Harness Tuning, Hybrid Routing, and Safer Sandboxes Move Coding Agents Forward

Coding Agents Alpha Tracker

2026-04-14

Harness Tuning, Hybrid Routing, and Safer Sandboxes Move Coding Agents Forward

By Coding Agents Alpha Tracker • April 14, 2026

Harness quality emerged as today's real edge: Theo unpacked the agent loop, Cursor confirmed live harness A/B testing, and Cloudflare shipped new primitives for safer, stateful agent sandboxes. Also inside: Cursor 3.1 upgrades, a practical local-vs-cloud routing playbook, and reproducible repo experiments from Simon Willison.

TOP SIGNAL

Today's clearest signal: **harness engineering is becoming a first-class performance lever**, not a footnote to model choice. Theo's breakdown defines the harness as the tool/runtime loop around the model, cites an independent benchmark where Opus went from **77% in Claude Code to 93% in Cursor**, and Cursor CEO Michael Truell separately says Cursor A/B tests the harness itself on live traffic [1, 2].

Practical takeaway: stop evaluating models in isolation—the **tool descriptions, permissions, context bootstrap, and retry loop** are part of the product, and Theo shows even small description changes can materially alter tool behavior [1].

TOOLS & MODELS

- **Cursor 3.1:** split agents for multitasking, pick the branch for a cloud agent, better voice input with `Ctrl-M` hold-to-talk, jump from a diff to the exact file line, workspace search include/exclude filters, and an **87% reduction in dropped frames** for large file edits. Full changelog: http://cursor.com/changelog/3-1 [3, 4, 5, 6, 7, 8]

- **Cursor's team is tuning more than the model:** Truell says Cursor A/B tests **model checkpoints, UX, and the agent harness**, including sending **<1% of traffic** to compare how Claude behaves under the Claude Code harness versus Cursor's default harness [2]
- **Cloudflare Durable Object Facets:** sandboxed Dynamic Workers can now access **SQLite** through standard Durable Object implementations with fast synchronous reads/writes; a supervisor Durable Object can create attached databases and pass specific ones into workers. Kent C. Dodds says he is integrating this into Kody immediately and expects a significant capability boost. Blog: https://blog.cloudflare.com/durable-object-facets-dynamic-workers/ [9, 10, 11]
- **Cloudflare outbound Workers for Sandboxes:** credential injection, egress logging, and zero-trust policies at the **network layer** for agent sandboxes. Dodds notes Kody previously had to solve the same basic secret-injection problem earlier at the **template layer** because this feature did not exist yet. Announcement: https://cfl.re/4tfSt1G [12, 13, 14]
- **Practical model routing from OpenClaw:** Berman keeps **Opus 4.6 / GPT 5.4** for coding, planning, and orchestration, then offloads embeddings, transcription, voice, PDF extraction, classification, and some chat to local models like **Qwen 3.5**, **Nemotron**, and **Gemma** via LM Studio. His hardware heuristic: ~**30B** models are the sweet spot for many consumer GPUs [15]

WORKFLOWS & TRICKS

- **DIY harness in a weekend:** Theo's minimal version is small enough to build yourself. Core loop: define a few tools like `read_file`, `list_files`, and `edit_file` (or just `bash`), list them in the system prompt, let the model emit `tool: name {json}`, execute the tool, append the output to history, repeat [1]
- **Tune tool descriptions per model, not once:** Theo demos that changing only a tool description can change which tool the model reaches for. His broader point: models only see the descriptions/context you give them, and different models react differently to the same wording [1]
- **Keep upfront context short; let tools do the exploration:** use `.claude.md` or `.agent.md` for the highest-value bootstrap context, then let the model search/read its way to the rest. Theo's case against repo stuffing is blunt: large contexts make models worse, tool-based exploration beat Repomix-style packing, and staying in one thread preserves useful history [1]
- **Three-stage local-model rollout:** Berman's pattern is clean: **(1)** experiment with frontier models only, **(2)** productionize and identify sub-tasks already working on weaker models, **(3)** move repeated, lower-complexity work local after edge-case testing. His examples: notification classification, company-news relevance, CRM context extraction, and

knowledge-base summarization [15]

- **Concrete way to wire a local model into an agent stack:** run LM Studio on the target GPU machine, load a model like **Qwen 3.5 35B**, ask Cursor to SSH in and add it to OpenClaw's routing config, then smoke-test it in Telegram with `/status` and a quick prompt. Berman reports about **65 tok/sec** on DGX Spark and faster simple chat round trips than Sonnet in his setup [15]
- **Rule-first prompting is emerging as a sane default:** ThePrimeagen says he is codifying his own programming rules, applying them through several stages, and keeping the scope to small changes while staying accountable for every line instead of letting agents dump code over the wall [16]

PEOPLE TO WATCH

- **Theo** — Best demystifier today. He turns harnesses from buzzword into a concrete loop, then shows why tool descriptions, prompts, and context loading materially change outcomes [17, 1]
- **Michael Truell** — Rare firsthand confirmation that Cursor is testing the harness itself on real traffic, not just swapping models behind the scenes [2]
- **Addy Osmani** — Strong firsthand signal from inside Google: **40K+ SWEs** use agentic coding weekly, with internal custom CLIs, MCPs, orchestrators, agent loops, and virtual SWE teams in daily use [18]
- **Matthew Berman** — Shared the clearest frontier-to-local routing playbook of the day: use the best cloud models for code and planning, then offload repeatable sub-tasks locally once you've validated the workflow [15]
- **Simon Willison** — Still the best source for bounded, reproducible agent experiments: this time he had Claude Code explore the new `servo` crate, build a working screenshot CLI, and publish both the repo and the task PR [19]

WATCH & LISTEN

- **Theo — 15:30-19:17:** Best short explainer on why stuffing an entire repo into context is the wrong instinct. He walks through why tool-driven context building beats Repomix-style packing, and why bigger context can

make models worse [1] *How does Claude Code* actually* work? (15:30)*

- **Theo — 20:37-23:05:** The minimal harness primer. Three tools, a system prompt, and a loop. Watch this before you over-engineer your own

agent runtime [1] *How does Claude Code* actually* work? (20:37)*

- **Latent Space — 42:54-46:30:** Sharp management clip on the new failure mode: engineers juggling many agents all day get fatigued, then still have to review critical PRs. The takeaway is simple—AI increases the need for serious human review, not less [20]

The best engineers don't write the most code. They delete the most code. — Stay Sassy (42:54)

PROJECTS & REPOS

- **`servo-shot` / Simon's `servo` exploration repo:** Claude Code explored `servo` v0.1.0, identified the API surface, and built a headless CLI that renders URLs or HTML to PNG. The replication steps, repo, and task PR are all public: https://github.com/simonw/research/tree/main/servo-crate-exploration#readme · https://github.com/simonw/research/pull/108 [19]
- **`html5ever` / `markup5ever_rcdom` WASM playground:** compiling Servo itself to WebAssembly was not feasible, but Claude still produced a narrower useful artifact: a browser playground for turning HTML fragments into a parse tree [19]
- **`pi-tutorial`:** clever repo pattern from Armin Ronacher—package onboarding as an interactive agent experience instead of docs. Repo: http://github.com/earendil-works/pi-tutorial [21]

Editorial take: the real edge right now is not one magic model—it's better harnesses, tighter context, and safer orchestration around the model [1, 2, 9]

Sources

1. How does Claude Code *actually* work?
2. X post by @mntruell
3. X post by @cursor_ai
4. X post by @cursor_ai
5. X post by @cursor_ai
6. X post by @cursor_ai
7. X post by @cursor_ai
8. X post by @cursor_ai
9. X post by @KentonVarda
10. X post by @kentcdodds
11. X post by @kentcdodds
12. X post by @Cloudflare
13. X post by @kentcdodds
14. X post by @kentcdodds
15. "But OpenClaw is expensive…"
16. X post by @ThePrimeagen
17. X post by @theo
18. X post by @addyosmani
19. Exploring the new `servo` crate
20. The best engineers don't write the most code. They delete the most code. — Stay Sassy
21. X post by @mitsuhiko