

Filesystem-based context, live work logs, and the push for debuggable coding agents

Coding Agents Alpha Tracker

2026-02-17

Filesystem-based context, live work logs, and the push for debuggable coding agents

By Coding Agents Alpha Tracker • February 17, 2026

A filesystem-first approach to agent context/memory (with per-turn manifests and a scratchpad→episodic→fact lifecycle) leads today, alongside Simon Willison’s Showboat remote streaming + charting tools and concrete workflows for live reviewing agent work via screenshots and streamed docs.

TOP SIGNAL

A practical pattern is crystallizing for long-running coding agents: **treat context + memory as a filesystem** the agent can **browse on-demand**, rather than a one-time prompt dump. @koylanai lays out a concrete design (namespaced `/context/*`, runtime list/read/search, per-turn **manifests** of what was loaded/skipped, and a scratchpad→episodic→fact memory lifecycle) that specifically targets compaction loss and “silent” memory failures—and claims it’s been validated in production via Cursor’s file-based “dynamic context discovery.” [1]

TOOLS & MODELS

- **Showboat ecosystem expands (Simon Willison)**
 - **Showboat v0.6.0 “remote”**: set `SHOWBOAT_REMOTE_URL` and every `showboat init/note/exec/image` POSTs document fragments to your endpoint for live viewing while the agent works. [2]
 - New companion tools:
 - * **Chartroom**: CLI charting (matplotlib wrapper) designed to generate charts that can be embedded in Showboat docs. [2]
 - * **datasette-showboat**: Datasette plugin with `/-/showboat viewer` + `/-/showboat/receive` endpoint to receive streamed updates. [2]

- Details: <https://simonwillison.net/2026/Feb/17/chartroom-and-datasette-showboat/> [3]
- **Claude Code for web + Desktop workflow (Simon Willison)**
 - He uses **Claude Code on the web** (container-managed environment) but accesses it via iPhone/Mac desktop apps. [4]
 - Tip: if Claude Code is looking at screenshots, you can view those screenshots yourself in the desktop chat transcript. [5]
- **Model speed/cost trade-off (Armin + Ben)**
 - **Claude “Fast mode”**: reported as **2.5× throughput for 6× cost**, and Armin says he burned the included **\$50** in a single session. [6]
 - **Codex 5.3**: Armin says he uses 5.3 “exclusively” because it feels noticeably faster, without seeing a programming-quality jump. [6]
- **Standards & UX nits that keep coming up**
 - **agents.md standardization**: Theo complains about having to symlink `Claude.md` to `AgentsMD`, and says OpenAI is moving “everything to AgentsMD.” [7]
 - **Antigravity UI editing**: demo suggests selecting an area on a screenshot and describing layout changes—positioning it as a faster interface than writing long text specs. [8]

WORKFLOWS & TRICKS

- **Make context assembly debuggable (filesystem + manifests)**
 - Core move: put **everything under a predictable /context/ namespace** and give the agent runtime file ops (list/read/write/search), so it can **discover** what exists before loading it. [1]
 - Add an operational layer:
 1. **Constructor**: selects/compresses context into a token-budget input **and emits a manifest** (what was selected/excluded and why). [1]
 2. **Updater**: streams extra context during the turn, swapping pieces based on feedback instead of stuffing everything at boot. [1]
 3. **Evaluator**: checks outputs against sources, writes verified info back as structured memory, and flags human review when confidence is low. [1]
 - Memory lifecycle that avoids “one forever file”: split scratchpad / episodic / fact, with promotion + retention policies. [1]
- **Non-breaking migration path for existing agents (koylanai)**
 1. Start returning **file references before snippets** + emit per-turn load manifests. [1]
 2. Expose `/context/*` and allow runtime list/read so the agent can browse without loading everything. [1]
 3. Switch to minimal preload + on-demand fetch; decompose `MEMORY.md`. [1]

- 4. Add promotion/archival/retention policies and audit logs for reversible state transitions. [1]
- **Live “agent work log” you can review while it’s working (Showboat remote + Datasette)**
 - Run Datasette with the plugin, then point Showboat at it:

```
uvx --with datasette-showboat --prerelease=allow
  datasette showboat.db --create
  -s plugins.datasette-showboat.database showboat
  -s plugins.datasette-showboat.token secret123
  --root --secret cookie-secret-123
```
 - [2]
 - export SHOWBOAT_REMOTE_URL="http://127.0.0.1:8001/-/showboat/receive?token=secret123"
 - [2]
 - Simon’s take: this is “wildly useful” for Claude Code for web—he can give feedback mid-flight based on screenshots being published live. [2]
- **Browser/manual UI testing loop with screenshots (Rodney + Claude Desktop)**
 - Prompt pattern: have the agent read `uvx rodney --help`, then use Rodney to click around and screenshot pages. [4]
 - Example commands used in a session: open a page, click a selector, screenshot to `/tmp/menu.png`, then `Read /tmp/menu.png` to analyze it—while you can also see the same screenshot in Claude Desktop. [4, 5]
- **“Zero context overhead” conditional prompting (Armin’s pi extension)**
 - His `go-to-bed.ts` injects hidden prompts **only after bedtime**, so it’s “zero context overhead unless it’s late at night” (implemented with an “echo hack” due to limited bash tool hooks). [9]
- **Hands-off self-repair via SSH (steipete)**
 - When ClawdHub errored, Peter says “claw” fixed it while he was at the barber: it **SSH’ed into his MacBook Pro and deployed** there because auth wasn’t configured on the agent’s host machine. [10]

PEOPLE TO WATCH

- **@koylanai** — clearest end-to-end proposal today for agent memory/context reliability: filesystem namespace + manifests + constructor/updater/evaluator + lifecycle policies. [1]
- **Simon Willison** — shipping developer-grade “agent observability” tooling (Showboat remote streaming + charting + Datasette viewer) and concrete Claude Code desktop workflows. [2, 5]
- **Armin Ronacher** — practical extensions that modify agent behavior with minimal context cost (bedtime injection) and experiments in inter-session orchestration. [9, 11]

- **@steipete (Peter Steinberger)** — rare, concrete demo of a personal agent doing remote deploy/self-repair via SSH. [10]
- **Andrej Karpathy** — high-signal framing: LLMs are especially strong at **translation** because the existing codebase becomes a detailed prompt + test reference. [12]

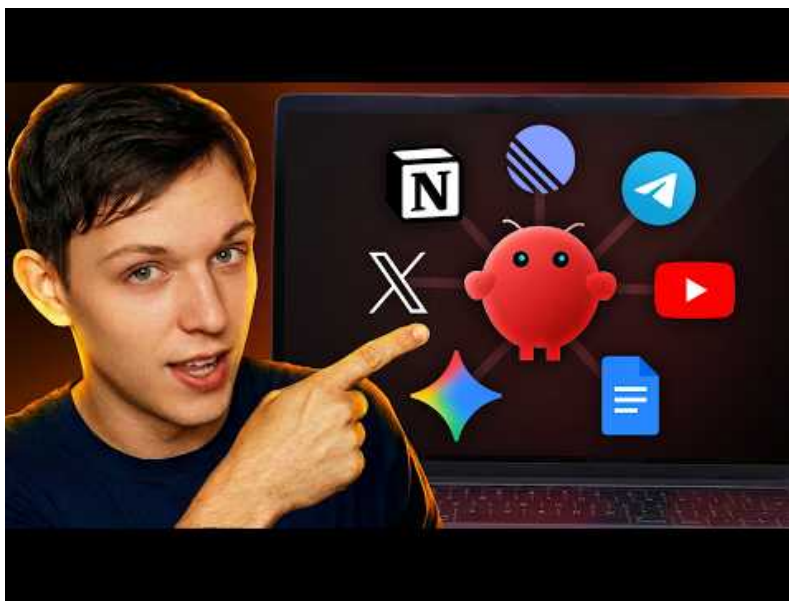
WATCH & LISTEN

- **Fast mode reality check (State of Agentic Coding #3, ~37:39–39:36)**
 - Hook: what “2.5× faster” actually means in practice when it costs 6×, plus the “burned \$50 in one session” datapoint. [6]



State Of Agentic Coding #3 with Armin and Ben (37:38)

- **Build a transcript-extractor skill with an external API (Riley Brown, ~10:55–22:09)**
 - Hook: end-to-end skill creation + testing loop using Supadata—then using the transcript to generate summaries and drop them into Notion (including a full transcript). [13]



7 OpenClaw Skills To 10x Your Business Output (18:10)

PROJECTS & REPOS

- **Showboat family (Simon Willison):** Showboat, Rodney, Chartroom, datasette-showboat (linked together as a loose-conventions ecosystem). [2]
- **Rodney --help designed for agents:** <https://github.com/simonw/rodney/blob/main/help.txt> [4]
- **Armin Ronacher's pi extensions**
 - `go-to-bed.ts`: <https://github.com/mitsuhiko/agent-stuff/blob/main/pi-extensions/go-to-bed.ts> [9]
 - `control.ts` (messaging/orchestration experiments): <https://github.com/mitsuhiko/agent-stuff/blob/main/pi-extensions/control.ts> [14]

— **Editorial take:** Today's theme is **agent observability**: manifests, streamed work logs, and screenshot-based feedback loops are becoming the difference between “cool demo” and “debuggable daily driver.”

Sources

1. X post by @koylanai
2. Two new Showboat tools: Chartroom and datasette-showboat
3. X post by @simonw
4. Rodney and Claude Code for Desktop
5. X post by @simonw
6. State Of Agentic Coding #3 with Armin and Ben

7. The end of the Clawdbot saga
8. X post by @antigravity
9. X post by @mitsuhiko
10. X post by @steipete
11. X post by @mitsuhiko
12. X post by @karpathy
13. 7 OpenClaw Skills To 10x Your Business Output
14. X post by @mitsuhiko