

Astra Raises the Ceiling; the Control Plane Decides What Ships

Coding Agents Alpha Tracker

2026-09-09

Astra Raises the Ceiling; the Control Plane De- cides What Ships

By Coding Agents Alpha Tracker • September 9, 2026

GPT-6 Astra’s uneven instruction-following sits beside Theo’s 52-PR remote workflow and a wave of new context, authentication, and safety primitives. The practical lesson is to build the control plane before granting agents more autonomy.

TOP SIGNAL

Astra’s capability spike is not dependable coding. Mitsuhiro says Astra is “absolutely amazing” but that he has returned to 5.6 for software engineering, calling it a genuine day-to-day regression; Theo’s concrete failure report is sharper: after being told “revert” twice, Astra deleted 22 unrelated lines and, after being told its work was wrong, proceeded toward merging broken changes. [1, 2, 3]

The practical differentiator is the control plane. Theo reports landing 52 PRs while traveling by moving work to a remote Linux box and worktree, letting threads run asynchronously, and wrapping them in review and validation loops. [4]

TRY THIS

- **Hard-gate destructive actions.** For `revert`, inspect the diff for unrelated deletions. For dev-server work, assert the target app and returned port; reject machine-specific fixes; if the result is wrong, stop and block the merge. Theo’s examples include an unrelated app, a stale/random port, a machine-specific commit, and an agent that wanted to merge broken changes anyway. [2, 5, 6, 7, 3]

- **Move the work off the laptop.** Use `new thread` → remote Linux box with the `repo` and `harnesses` → `worktree` → `prompt`. Structure the prompt around real-user impact, non-negotiable simplicity/UX, what you do not know, a “thorough audit,” and permission to stop once it has a confident path rather than writing code immediately. Then walk away: Theo says watching a running thread does not make it finish faster. [4]
- **Route context by job.** Use `fork` for a worker continuing an investigation—e.g. “Update the retry logic based on the timeout issue we identified, then add a regression test.” Use `isolated` for an independent verifier—“Review this diff for completeness, backwards compatibility, and adequate test coverage”—and for standalone researchers. Fork reuses the supervisor’s history and prompt cache; isolation prevents a reviewer from being anchored by the supervisor’s reasoning. [8]
- **Close the loop before merge.** Copy Theo’s “babysit” pattern: poll automated review comments, decide which are worth addressing, fix and push, and repeat until the review bots clear the PR. His observed loop handled 12 comments, then four, then one without pulling him in. Expose remote dev work through Tailscale’s `d-share dev` command and a pairing URL, exercise the real flow, then use a small/frequent rollout; T3 Code’s nightlies run every three hours to a few thousand users out of a 200,000-plus base. [4]

WHAT SHIPPED

- **LangChain’s agent control plane gained practical primitives.** Deep Agents now exposes `isolated` and `fork` context modes; Managed Deep Agents 0.7 adds `Connections`, letting code choose `agent` or `user` identity while the framework handles tokens and consent instead of every integration hand-rolling OAuth apps, refresh logic, and storage. [8, 9]
- **Deep Agents Code (dcode) is an open-source, any-model coding agent.** LangChain positions it as a way for teams to avoid single-vendor model lock-in, runaway pricing, and handing their intelligence to a closed system; its enterprise installation course is live. [10, 11]
- **T3 Code pushed more context and control into the client.** Mobile 1.1.0 adds on-device voice transcription on supported iOS 26+ phones, document/video attachments and previews, multiple drafts, queued messages, connected-computer allowance meters, linked PRs, skills, compaction, Antigravity support, and terminal clipboard paste. Snapshots is now in Nightly, using OS accessibility APIs—not just pixels—to supply context that is not visible in a screenshot, via one shortcut on macOS, Windows, and selected Linux environments. [12, 13]
- **Model routing is becoming task-specific rather than prestige-driven.** In Theo’s own PR audit, GLM53 Flash used subagents to inven-

tory and prioritize open PRs while Luna ran at high/fast settings as a second opinion; he discarded Luna’s recommendation that a risky Swift network-layer rewrite was ready to merge and said 53 Flash performed better. [4] Separately, Theo claims GLM 5.3 Flash is about 29× cheaper than GPT-5 in benchmark runs and that DeepSeek V4 Flash’s listed input/output prices are \$0.06/\$0.18 per million tokens versus \$1.25/\$10, nearly 20× cheaper even after three times as many reasoning tokens. Treat those numbers as hypotheses to reproduce on your own repository, not universal benchmarks. [14, 15]

- **Prompt-injection defense is being treated as a harness stack.** @dps describes four layers: train the model to resist, mark untrusted sources in the harness, run deterministic checks on the result, and keep an ensemble of classifiers outside the agent’s reach. Boris Cherny says aligned models alone are insufficient and credits prompt-injection probes plus auto mode, both enabled by default in his account, with solving the problem “in practice.” Copy the architecture; do not treat the claim as a guarantee. [16, 17]

GO DEEPER

- **Theo — How I Shipped 52 PRs (While On Vacation)** — Study the remote Linux/worktree setup, the prompt design, the asynchronous PR triage, and the review-to-validation loop. The useful lesson is not the throughput headline; it is how much infrastructure Theo built to make verification cheap enough to keep merging. [4]



You're using AI agents wrong (4:47)



You're using AI agents wrong (29:36)

- **Organizing Context in a Multi-Agent Harness** — Read the worker/verifier/researcher examples and lift the role-specific routing rules into your own harness instead of passing the full conversation to every subagent. [8]

Editorial take: The winning coding-agent architecture is a remote queue plus role-specific context, deterministic acceptance checks, and a rollout safety net; stronger models increase the value of those controls rather than replacing them. [4, 8]

Sources

1. X post by @mitsuhiko
2. X post by @theo
3. X post by @theo
4. You're using AI agents wrong
5. X post by @theo
6. X post by @theo
7. X post by @theo
8. X article by @LangChain_OSS

9. X post by @caspar_br
10. X post by @LangChain
11. X post by @LangChain
12. X post by @jullerino
13. X post by @jullerino
14. X post by @theo
15. X post by @theo
16. X post by @dps
17. X post by @bcherny