Claude's Agent SDK uses up the 5-hour limit faster than Claude Code, and Theo's cross-model "council" for reviews

Coding Agents Alpha Tracker

2026-11-03

Claude's Agent SDK uses up the 5-hour limit faster than Claude Code, and Theo's cross-model "council" for reviews

By Coding Agents Alpha Tracker • November 3, 2026

How Anthropic's Ultra plan change interacts with apps built on the Agent SDK, Theo's blind cross-model review data, LangChain's trace-to-eval improvement loop and Managed Deep Agents beta, plus OpenAI's Pando Git backend.

Claude subscriptions in third-party harnesses: the 5-hour limit catch

Anthropic renamed Max 20x to **Ultra** (still $200/month) and Max 5x to **Max** (still $100). It is also removing the 5-hour limit for *verified* Ultra accounts, starting with most existing subscribers. Newer accounts keep a temporary limit until they are verified, and weekly limits are unchanged [1]. Theo says Anthropic is vague about what "verified" means. His tip: under settings → usage, if you don't see a current-session bar, your account is verified [2].

This matters if you use a Claude subscription through anything other than the interactive Claude Code CLI. According to Theo, a user measured T3 Code using about **1.8x more of the 5-hour window per token** than the CLI. `-p` mode has the same problem, and Theo confirmed the numbers himself [2]. Any app using the Agent SDK takes the hit, and he says OpenCode's Claude support uses the Agent SDK too. That leaves the $20 and $100 tiers worse off [2]. The cause, he says, is a `cloud code entry point` field that is set to `SDK-TS`. Changing it removes the penalty. He got "semi-solid" permission from Anthropic to report T3 Code as the entry point, so once that release ships, T3 Code should get the same 5-hour and weekly usage as Claude Code directly [2].

He also says he routes his own inference through a CLI proxy, but he won't vouch for the ban risk [2].

Theo's "council": one review task, several model families

Theo's `council` skill sends a proposal, plan or change to **3–5 model families** for feedback. Because every model gets the exact same "review this change" task, it doubles as a like-for-like eval [3]. He then had Opus and Sol (transcribed as "Soul") blind-judge each other's reviews. Opus picked Opus 66 of 83 times, and Sol picked Opus 54 of 83 times [3]. Median cost per review was $1.65 for Opus versus about $0.36 for Sol. Against what each subscription gives you, he estimates about 205 Opus reviews on Claude's plan versus about 290 Sol reviews on ChatGPT Pro, so the usage gap is much narrower than the price gap [3]. His current split: Claude subscriptions for day-to-day coding, OpenAI subscriptions for computer use and verification, and Grok for Grokbot [3].

Improving an agent: traces → eval → change

At Runtime 2026, Harrison Chase described a loop you can apply to any harness, including your own coding-agent setup. Run the agent, collect feedback, read the traces, add the failure as an eval data point, propose a change, and check whether it fixes the eval [4]. Harness changes (code, tools, base prompt) start from traces and stay human-reviewed. LangSmith Engine automates the analysis: it runs a sandboxed agent that pulls and clusters traces and suggests fixes, dataset additions and evals [4]. Traces need exact inputs, outputs and tool-call order to be useful for debugging agents or post-training [4]. Evals should cover realistic inputs and sometimes the agent's path, not just the final answer, and they need a simulated environment when the agent is wired to production systems [4]. On fine-tuning, he says SFT on filtered good traces works best for simpler pieces like guardrails or subagents, while RL is better for the core loop but needs an environment and scoring [4].

LangChain also moved **Managed Deep Agents** to public beta on LangSmith Cloud [5]. An agent is a folder: `agent.py` plus `instructions.md`, `skills/` and `tools/` [5]. Commands: `uvx --from managed-deepagents mda init <name>`, `uv run mda dev` (local server plus Studio traces), `uv run mda deploy`. There is also a `managed-deep-agents` skill so a coding agent can do the setup [5]. Each deployment exposes `/mcp`, so Claude Code can call it as a tool [5]. Coding patterns LangChain reports from beta teams: an issue-tracker webhook that triggers a build-and-verify loop, browser tests in preview environments before human review, and alert triage that opens a fix PR [5].

Reworking an existing repo instead of starting fresh

Riley Brown gave Codex (GPT-6.1 Sol, extra high) his existing Mac and iOS app repo and the API docs. He told it to read the repo first and reuse the current architecture, then described the new purpose in detail: a screenshot every 30

seconds, judged on-task, off-task or unsure with a confidence score, and nudge rules with thresholds and a cooldown [6]. The build ran for 18 minutes [6]. Next came targeted UI feedback, then a 4x-speed developer mode so the time-based behavior could be tested quickly [6].

Short notes

- **OpenAI Pando:** a new Git backend for OpenAI's monorepo that cut push latency by about 80% and serves nearly a petabyte of Git data a day, with no change to how engineers or agents use Git [7]. Greg Brockman says OpenAI is considering offering it to others [8].
- **Weight-space editing:** in an MIT study, GPT-6 Astra traced Qwen3.5-0.8B's reasoning loops to two layers and edited 0.016% of its weights. With sampling held fixed, MATH-500 went from 7.9% to 48.1% [9].
- **GitHub Copilot benchmarks:** GitHub published per-model performance data for its harness at benchmarks.github.com [10]. Theo had hoped for "more than two existing open source benchmarks" [11].
- **Grounding agents in a point of view:** Riley points Claude Code or Codex at specific podcasts through Exa's transcript search before tasks that need cultural context, such as landing pages or scripts [12].
- **Test fan-out small first:** Legora's tabular review runs the first 10 rows so you can iterate before scaling to thousands of documents [13]. Its lists primitive triggers agent runs the way you might coordinate coding agents through Linear [13].
- **Automation for unglamorous work:** Hex's automated system made 110 fixes in a week, mostly cross-platform compatibility issues the team says it would never have done by hand [14].
- **Evals by side agent:** @badlogicgames had a spare agent build a tool plus an eval across models. The eval runs in codemode and spawns subagents in the same durable pi session [15].
- **Open problems in context engineering:** LangChain lists async subagents notifying a still-running main agent, deciding what memory should keep, and goal-gaming (e.g., "get rid of all bugs" → delete the codebase) [16].
- **dots model routing:** OpenAI's Tibo says dots will route among models including GPT-6 Astra and GPT-6.1 Terra. This is in progress, not launched [17, 18].

Sources

1. X post by @claudeai
2. Anthropic can't stop winning
3. OpenAI is getting desperate
4. Building agents that improve over time | Harrison Chase CEO at LangChain, Runtime 2026

5. Managed Deep Agents: the fastest way to ship a production deep agent
6. I Turned My Phone Into a Focus Machine (That Watches Me Work)
7. X post by @OpenAIDevs
8. X post by @gdb
9. X post by @latentspacepod
10. X post by @pierceboggan
11. X post by @theo
12. X post by @rileybrown
13. Building Legal Agents When You Can't See the Data — Jacob Lauritzen, CTO of Legora
14. X post by @thdxr
15. X post by @badlogicgames
16. New context engineering challenges in late 2026
17. X post by @thsottiaux
18. X post by @thsottiaux