

Stateful agent runs (WebSockets), Codex on Windows, and skills-driven harnesses

Coding Agents Alpha Tracker

2026-03-05

Stateful agent runs (WebSockets), Codex on Windows, and skills-driven harnesses

By Coding Agents Alpha Tracker • March 5, 2026

A dense brief on what’s actually moving the needle for coding agents: OpenAI’s WebSockets approach to cut tool-call overhead, Codex’s new Windows app + sandboxing, and the growing “skills + traces + evals” ecosystem that turns agents into repeatable workflows. Plus: production patterns from Anthropic’s Claude Code and hard-earned PR hygiene rules for agent-generated code.

TOP SIGNAL

OpenAI’s new **WebSockets API for agentic runs** is a real infrastructure unlock: keep a persistent connection to the same server so you can send **only new inputs** (e.g., tool results) instead of resending the **entire conversation history on every tool call** ¹². Theo estimates this cuts bandwidth by **90%+** and improves speed by **20–30%** (and **20–40%** on runs with 20+ tool calls) ³⁴.

TOOLS & MODELS

- **OpenAI — WebSockets for tool-call-heavy agents**
 - Why it matters: in the typical stateless flow, *every tool completion* triggers a new API call that resends *all prior messages/tool calls* so the model can continue ⁵.
 - WebSockets are positioned as a “*hit the same box*” guarantee, so you don’t keep re-checking auth / reloading state / reshipping context

¹I can’t believe nobody’s done this before...

²I can’t believe nobody’s done this before...

³I can’t believe nobody’s done this before...

⁴I can’t believe nobody’s done this before...

⁵I can’t believe nobody’s done this before...

- during a single long generation ⁶.
 - Practical caveat: Theo says the benefit is **not huge for typical chat**, but is big when **one user message spawns hundreds of tool calls** ⁷.
- **OpenAI Codex app — now on Windows (native + WSL)**
 - Available on Windows with a **native agent sandbox** and PowerShell support ⁸.
 - Runs **natively and in WSL** with integrated terminals (PowerShell, Command Prompt, Git Bash, WSL) ⁹.
 - Sandbox controls: blocks filesystem writes outside your working folder and blocks outbound network access unless you explicitly approve it ¹⁰.
 - Adds **2 Windows skills** (WinUI + ASP.NET) and **7 new “Open in ...” apps** ¹¹.
 - Download: <https://apps.microsoft.com/detail/9plm9xgg6vks?hl=en-US&gl=US> ¹²
- **Codex (Plus/Pro) — rate-limit promo bug fixed**
 - OpenAI fixed an issue where the **2× promotional limit increase** wasn’t applied to an estimated **9%** of Plus/Pro users; they reset rate limits for all Plus/Pro as compensation ¹³.
- **Cursor — now in JetBrains via Agent Client Protocol**
 - Cursor says it’s now available in JetBrains IDEs through ACP ¹⁴.
 - Details: <http://cursor.com/blog/jetbrains-acp> ¹⁵
- **LangChain — “skills” packages for coding agents (progressive disclosure)**
 - **LangChain skills**: 11 skills across LangChain/LangGraph/Deep Agents, intended to be dynamically loaded *only when relevant* to avoid tool overload degrading performance ¹⁶¹⁷¹⁸.
 - Claimed eval bump for Claude Code on LangChain ecosystem tasks: **29% → 95%** ¹⁹. Repo: <https://github.com/langchain-ai/langchain-skills> ²⁰
- **LangChain — LangSmith CLI + Skills**
 - LangSmith CLI is described as “agent-native” for traces/datasets/experiments,

⁶I can’t believe nobody’s done this before...

⁷I can’t believe nobody’s done this before...

⁸ post by @OpenAIDevs

⁹ post by @ajambrosino

¹⁰ post by @ajambrosino

¹¹ post by @ajambrosino

¹² post by @thsottiaux

¹³ post by @thsottiaux

¹⁴ post by @cursor_ai

¹⁵ post by @cursor_ai

¹⁶LangChain Skills

¹⁷LangChain Skills

¹⁸LangChain Skills

¹⁹LangChain Skills

²⁰LangChain Skills

- designed to be used through the terminal ²¹.
 - Claimed eval bump for Claude Code (Sonnet 4.6) on LangSmith tasks: **17% → 92%** ²²²³.
 - CLI repo: <https://github.com/langchain-ai/langsmith-cli> ²⁴
- **Codex 5.3 (xhigh) — notable model-level win vs Opus 4.6 (anecdote)**
 - Mitchell Hashimoto reports Codex 5.3 (xhigh) fixed a bug that had resisted engineers for **6 months** in **45 minutes** for **\$4.14**; he notes **Opus 4.6 failed** and lower Codex reasoning levels failed ²⁵.
 - He says a key difference was Codex (xhigh) eventually **read GTK4 source code**, which other runs didn't do ²⁶.
- **Qwen 3.5 — open-weight model family (practitioner testing signal)**
 - Simon Willison notes Qwen 3.5 shipped a large model (397B-A17B) plus smaller siblings down to 0.8B ²⁷.
 - He reports positive results for coding from **27B/35B**, and that **9B/4B/2B** were “notably effective” given size ²⁸.

WORKFLOWS & TRICKS

- **Run parallel “plan mode” tabs, then let the agent one-shot implementation (Anthropic / Claude Code)**
 - Boris Cherny describes a workflow of running multiple Claude Code instances in parallel: start in **plan mode**, iterate to get the plan right, then let it implement (often “one shot”) ²⁹.
 - He also leans on desktop app **worktree support** for environment isolation so parallel agents don't interfere ³⁰.
- **Make the agent test itself (and still keep a human approval gate)**
 - Boris says Claude Code will often run tests locally and may write new tests; when they change Claude Code internally, it will even **launch itself as a subprocess** to test end-to-end ³¹.
 - Anthropic runs Claude Code review in CI as a first-pass reviewer, catching “maybe ~80% of bugs,” followed by a human reviewer and final human approval ³².
- **Cheap-but-effective codebase search: “glob + grep” beats fancy**

²¹LangSmith CLI & Skills

²²LangSmith CLI & Skills

²³LangSmith CLI & Skills

²⁴LangSmith CLI & Skills

²⁵ post by @mitchellh

²⁶ post by @mitchellh

²⁷Something is afoot in the land of Qwen

²⁸Something is afoot in the land of Qwen

²⁹Building Claude Code with Boris Cherny

³⁰Building Claude Code with Boris Cherny

³¹Building Claude Code with Boris Cherny

³²Building Claude Code with Boris Cherny

setups (per Boris)

- Boris says their “Agentix Search” outperformed everything, and clarifies it’s basically **glob and grep** ³³.
- **Use uncorrelated context windows + subagents as “test-time compute” (Agent Teams / swarms)**
 - Boris explains “uncorrelated context windows” as multiple fresh contexts that don’t share the parent window (beyond the prompt), and says throwing more tokens at uncorrelated windows can yield better results—calling it a form of **test-time compute** ³⁴.
 - Their **Agent Teams** release is opt-in / research preview because it uses “a ton of tokens,” and is intended for **complex tasks** ³⁵.
- **Skills as procedural memory: keep the base prompt smaller, load expertise only when needed**
 - LangChain frames skills as curated instructions/scripts/resources that are **dynamically loaded through progressive disclosure** (retrieve only when relevant) ³⁶³⁷.
 - Their LangSmith “virtuous loop” is explicitly: add tracing → generate traces → build datasets → run evaluators → iterate based on evals + human feedback ³⁸.
- **Prompting pattern: force the model to surface missing assumptions**
 - Peter Steinberger treats agent use as a conversation and repeatedly asks: **“Do you have any questions?”** to avoid the model charging ahead with default assumptions ³⁹.
 - His warning: the **“agentic trap”** is spending time over-optimizing your setup—it can feel productive without improving output ⁴⁰.
- **PR hygiene: don’t dump unreviewed agent code on teammates**
 - Simon Willison’s anti-pattern: opening PRs with hundreds/thousands of agent-generated lines you haven’t reviewed is delegating the real work to reviewers ⁴¹.
 - What “good” looks like: ensure it works (and you’re confident), keep changes reviewable (multiple small PRs), include context/links, and **review the agent-written PR description too** ⁴²⁴³⁴⁴⁴⁵.
 - Add evidence you tested it (notes/screenshots/video) to avoid wast-

³³Building Claude Code with Boris Cherny

³⁴Building Claude Code with Boris Cherny

³⁵Building Claude Code with Boris Cherny

³⁶LangChain Skills

³⁷LangChain Skills

³⁸LangSmith CLI & Skills

³⁹OpenClaw Peter Steinberger

⁴⁰OpenClaw Peter Steinberger

⁴¹Anti-patterns: things to avoid

⁴²Anti-patterns: things to avoid

⁴³Anti-patterns: things to avoid

⁴⁴Anti-patterns: things to avoid

⁴⁵Anti-patterns: things to avoid

ing reviewer time ⁴⁶.

PEOPLE TO WATCH

- **Theo (t3.gg)** — consistently strong at turning infra changes into concrete agent cost/perf implications (his WebSockets breakdown is the clearest “why now” explainer) ⁴⁷⁴⁸.
- **Boris Cherny (Anthropic / Claude Code)** — high-signal production details: he claims Claude Code writes ~80% of Anthropic’s code, and describes CI review + self-testing patterns that keep velocity safe ⁴⁹⁵⁰.
- **Mitchell Hashimoto** — practical model comparison under real pressure: a 6-month bug solved by **Codex 5.3 (xhigh)** where other settings and **Opus 4.6** failed ⁵¹.
- **Simon Willison** — the anti-pattern chapter is “social scalability” for agentic coding: ship *reviewable, evidenced* PRs, not agent slop ⁵²⁵³.
- **Kent C. Dodds** — clear framing that “pit of success” needs to be adapted for agents; he claims agents have “inhuman abilities” to understand code ⁵⁴⁵⁵.

WATCH & LISTEN

1) WebSockets: why stateless tool loops spam full-context payloads (Theo, ~04:33–08:29)

Hook: a crisp mental model for why *every tool call* resends the entire history— and why caching doesn’t fix bandwidth. ⁵⁶

⁴⁶Anti-patterns: things to avoid

⁴⁷I can’t believe nobody’s done this before...

⁴⁸I can’t believe nobody’s done this before...

⁴⁹Building Claude Code with Boris Cherny

⁵⁰Building Claude Code with Boris Cherny

⁵¹ post by @mitchellh

⁵²Anti-patterns: things to avoid

⁵³Anti-patterns: things to avoid

⁵⁴ post by @kentcdodds

⁵⁵ post by @kentcdodds

⁵⁶I can’t believe nobody’s done this before...



I can't believe nobody's done this before... (4:32)

2) Agent Teams + “uncorrelated context windows” as test-time compute (Boris Cherny, ~1:15:31–1:18:00)

Hook: a practical explanation of why multiple fresh context windows + sub-agents can outperform “more tokens in one window,” and why Teams is opt-in (token cost).⁵⁷

⁵⁷Building Claude Code with Boris Cherny



Building Claude Code with Boris Cherny (75:31)

PROJECTS & REPOS

- **openai/symphony** — orchestration layer that polls project boards for changes and spawns agents per ticket lifecycle stage (board-state-driven agent work) ⁵⁸⁵⁹. Repo: <https://github.com/openai/symphony> ⁶⁰
- **langchain-ai/langchain-skills** — 11 progressively-disclosed skills for LangChain/LangGraph/Deep Agents tasks ⁶¹⁶². Repo: <https://github.com/langchain-ai/langchain-skills> ⁶³
- **langchain-ai/langsmith-cli** + **langchain-ai/langsmith-skills** — terminal-first tooling + skills for traces/datasets/evals ⁶⁴⁶⁵.
 - CLI: <https://github.com/langchain-ai/langsmith-cli> ⁶⁶
 - Skills: <https://github.com/langchain-ai/langsmith-skills> ⁶⁷
- **Wasteland (Steve Yegge)** — federated “wanted board” + PR-based

⁵⁸ post by @scaling01

⁵⁹ post by @scaling01

⁶⁰ post by @scaling01

⁶¹ LangChain Skills

⁶² LangChain Skills

⁶³ LangChain Skills

⁶⁴ LangSmith CLI & Skills

⁶⁵ LangSmith CLI & Skills

⁶⁶ LangSmith CLI & Skills

⁶⁷ LangSmith CLI & Skills

workflow with portable reputation stamps; reported **2400 submitted PRs** and **1500 merged** from **450+ contributors** ⁶⁸⁶⁹.

– Demo Claude skill: <https://wasteland.gastownhall.ai/skill> ⁷⁰

- **AmpCode trace for the Codex 5.3 fix** (full run + original issue context) ⁷¹: <https://ampcode.com/threads/T-019cbadf-cb5a-742e-b0e3-2d7164de743f> ⁷²

Editorial take: Today’s theme is **harness > model**: stateful sessions (WebSockets), skills-as-procedural-memory, and reviewable evidence are what turn “agent potential” into repeatable throughput.

Sources

1. I can’t believe nobody’s done this before...
2. post by @OpenAIDevs
3. post by @ajambrosino
4. post by @thsottiaux
5. post by @thsottiaux
6. post by @cursor_ai
7. LangChain Skills
8. LangSmith CLI & Skills
9. post by @mitchellh
10. Something is afoot in the land of Qwen
11. Building Claude Code with Boris Cherny
12. OpenClaw Peter Steinberger
13. Anti-patterns: things to avoid
14. post by @kentcdodds
15. post by @kentcdodds
16. post by @scaling01
17. Welcome to the Wasteland: A Thousand Gas Towns

⁶⁸Welcome to the Wasteland: A Thousand Gas Towns

⁶⁹Welcome to the Wasteland: A Thousand Gas Towns

⁷⁰Welcome to the Wasteland: A Thousand Gas Towns

⁷¹ post by @mitchellh

⁷² post by @mitchellh