Drive your agents from the web, not from the CLI

Most people stay in a local terminal because a terminal is where serious work happens and a browser tab is where it doesn't. That is a feeling, not an argument, and the three things you get for giving it up are not small ones.

16 September 2026 · 6 min read

Every serious agent now has a web client, and the web client is not really a client. It is a machine: it checks your repositories out into a VM somewhere, works in them, and hands you a pull request when it is done. Claude, Codex, Cursor — the shape is the same. The alternative, which is where most people start and where a great many stay, is to run the same agent on your own laptop and watch it think in real time.

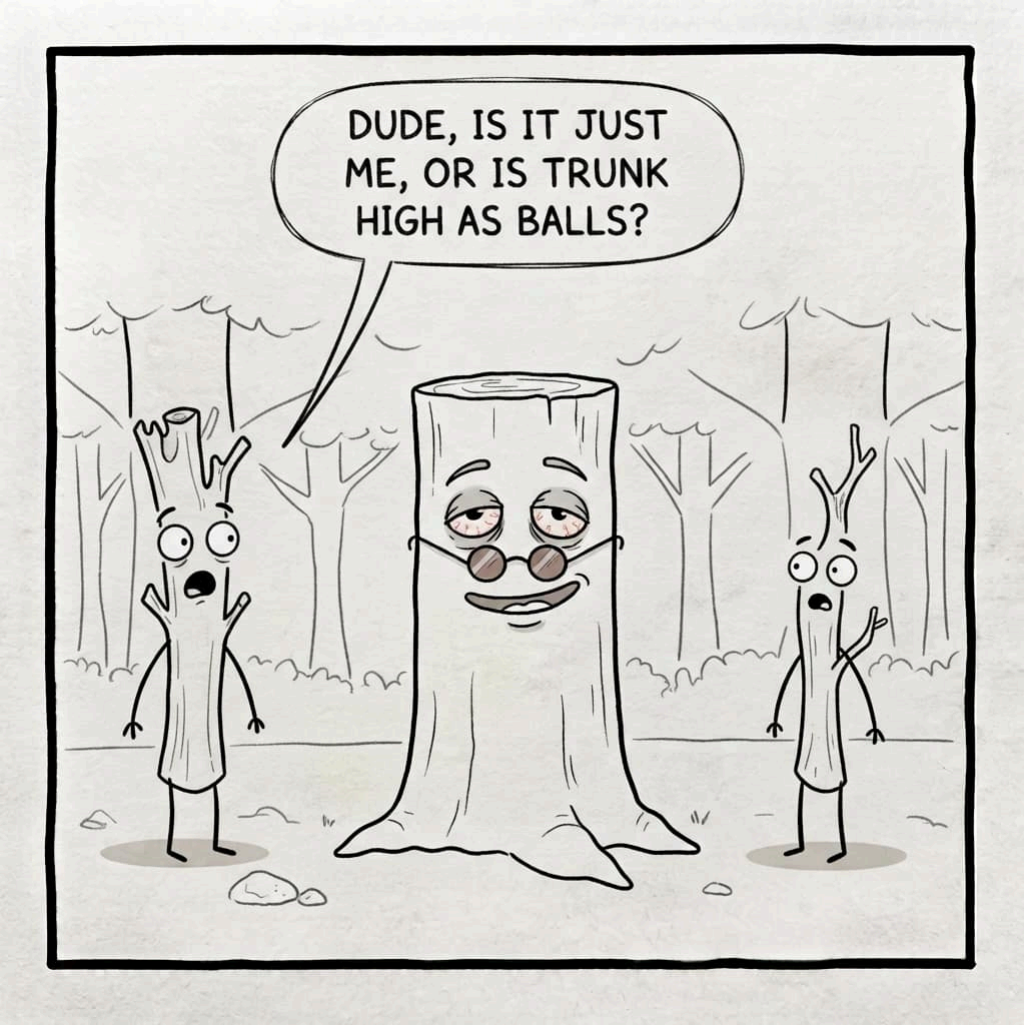
Stop doing that.

Mostly people stay for a reason nobody says out loud: a terminal is where serious work happens and a browser tab is where a person waits for a delivery. I will come back to that at the end, because it is truer than the arguments and it is not an argument. The argument I actually had was about merges — so that is where I will start, three reasons, in the order of how badly I had them weighted before I tried.

The branches do not fight

Locally, you work in one repository. Whatever your agents are up to, they are up to it in the same working tree, so they see each other coming; at any moment your checkout is the current state of everything you have going, and merging is somebody else's problem. Move to the cloud and every session gets a branch of its own, and you immediately start doing arithmetic about how ten of those are ever going to come back together.

I put the move off for months on the strength of that arithmetic. It was wrong. Not "mostly fine" wrong — it has not happened once, with two branches or with fifteen.



Agents turn out to be very good at working out what `main` did while they were away. What changed since they started; which of their own decisions that invalidates; which side of a conflict to keep, which to take from `main`, and where the honest answer is that both sides are now obsolete and the thing wants rewriting. A conflict across ten or fifteen files is not an event. Bigger ones are a slower conversation rather than a crisis, and they go better still if you give the agent a skill that says how merging is done in your repository — which is a subject for its own article.

(The one real exception is database migrations: two branches will both name theirs `0080`, and if they are pointed at the same development database they will trip over each other running them. That is a narrow problem with answers of its own, and it is not a reason to keep fifteen sessions on your lap.)

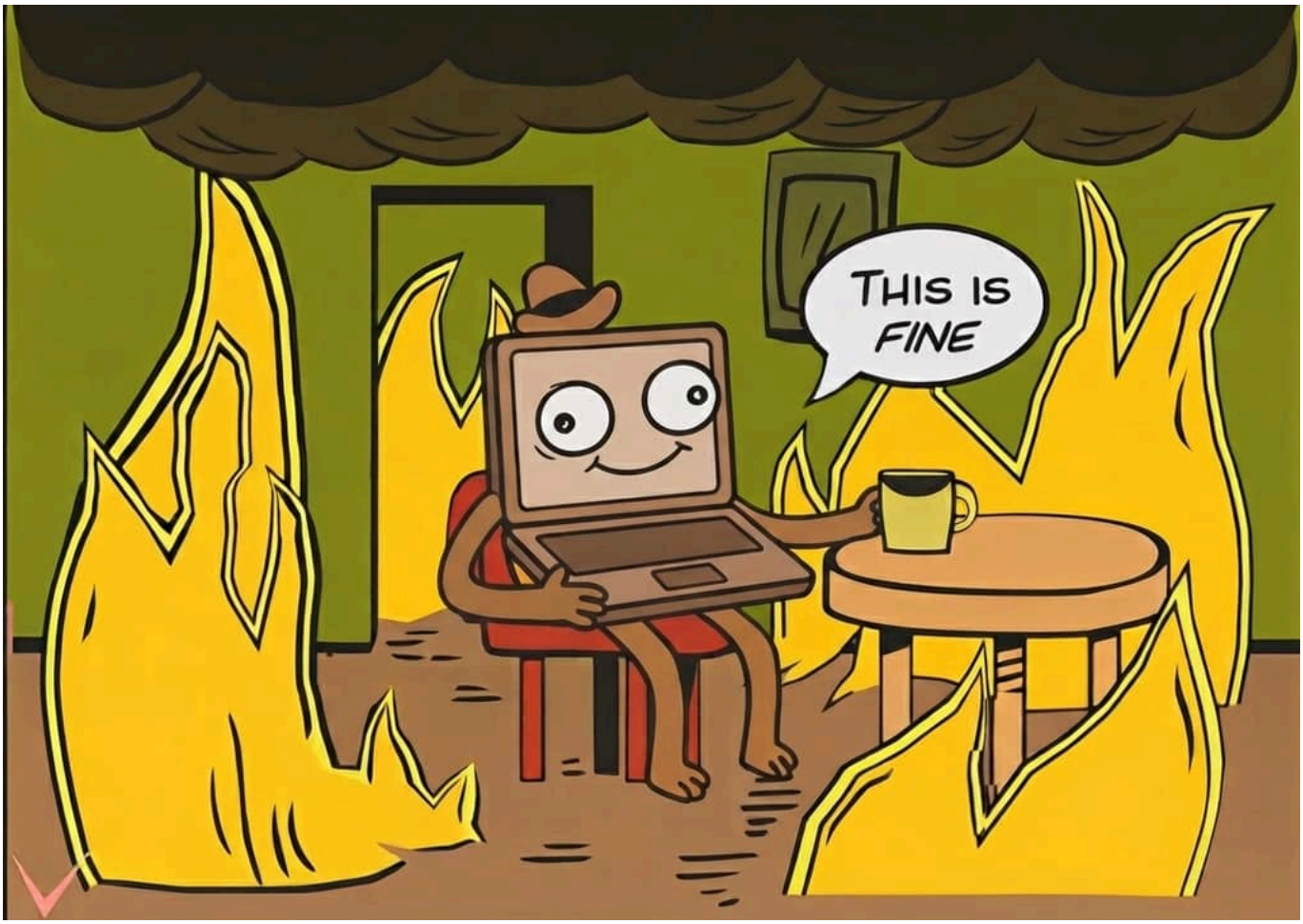
So the fear was a genuine fear of an imaginary thing, which is the most expensive kind, because nothing ever disproves it while you are avoiding it.

Your laptop stops melting

The second reason is, and I mean this fairly literally, thermodynamic.

Running locally I could hold three to five sessions, six at a push, and they beat on the processor hard enough that the fan went to takeoff power and still lost. The machine ran like a stove. And you cannot

close the lid, because the agents stop when you do.



In the web, none of it happens on your computer. Somebody else's processor gets hot, and unlike your own it does so for free: your subscription buys you as many VMs as you care to start, and nobody has yet come to have a word with me about it. You can also put a few other things in that VM — your CI, for one, which does wonders for the GitHub Actions bill — but that too is another article.

And you can close the lid.

The day becomes a pipeline

Which gets us to the third reason, and the one that actually changed how the work feels. You stop waiting.

Ten sessions, fifteen, however many the work divides into. You hand out the tasks and they all start thinking at once. One comes back with a question; you answer it, and while you are typing another finishes; you review that, and by the time you are done a third wants something. The day is a steady loop of handing out work, answering questions, and accepting results, and at no point in it are you watching tokens crawl up a screen with the expression of a man following them closely.

And, let us be honest about the local alternative: sooner or later you will drift. You start an agent, you read along, there is nothing for you to do, so after a while you get bored and open Netflix. You come back to find

the agent finished forty minutes ago, and you are two and a half episodes into the second season of Severance and have shipped nothing.

You do lose something real: engagement with each individual session. Reading an agent's reasoning is interesting and genuinely instructive, and now and then you catch a bad decision while it is still a thought rather than a diff. But weighed against the hours it takes, it is not much leverage. Choose a task size the agent can actually finish — also its own subject — and nearly everything you would have said mid-flight you can say to finished code instead, where it is cheaper to say and cheaper to act on, because there is something concrete under the comment.

My own loop, once a task comes back: the agent opens a pull request with a proposed merge message on it, so three or four paragraphs tell me what it did and why. I read the diff in the review tab and leave comments in it — "not like that", "pull this out into a function", "let us rethink the big picture here". Then I go back to the session, compact it rather than dragging the whole accumulated context along, and say: there are comments waiting, go handle them. It re-reads the PR, works through them, and comes back. Two or three rounds, usually, and it lands.

The part I did not see coming is that the switching is not a tax. It is where a good share of the insight comes from. A thought that belongs to the parser branch turns up while you are reading the deploy branch, because you have been in both of them in the last twenty minutes, and the two have no business having anything to do with each other, which is exactly why it works. Fifteen shallow contexts cross-pollinate in a way one deep one does not — and that is the human half of what another article here is about: the insight arrives from next door, and there is no way to schedule it.

What you give up

Status, mostly. In a terminal you are a hacker: three iTerm tabs, an agent in each, green text moving. In a browser you are a person with tabs, which is what everybody with a SaaS subscription is, and the work is happening somewhere you cannot see. The feeling of being hands-on is real and you do lose it.

It is worth losing. Every client has pinning, so the tasks you are actively holding sit at the top of the list and the day is a matter of clicking between the ones that have come back for you. It is also remarkably good for ADHD, diagnosed or self-awarded: you will never be too bored to work, because there are fifteen unrelated things in flight and you can simply switch to a different one. It presses on your context a bit, holding that many threads — but I was through the adjustment inside a week and enjoying myself.

Each branch gets to `main` in its own time, and mostly the order does not matter; you just keep methodically adding to the tower. Then at some point you look back at the list of commits — each one clean, each one a whole thing — and think: goodness, look how much of this there now is.

So

Open claude.ai/code, or whatever the equivalent address is for the agent you use, and work there.