



Claude is an Electron App because we've lost native

In [“Why is Claude an Electron App?”](#) Drew Breunig wonders:

Claude spent \$20k on an agent swarm implementing (kinda) a C-compiler in Rust, but desktop Claude is an Electron app.

If code is free, why aren't all apps native?

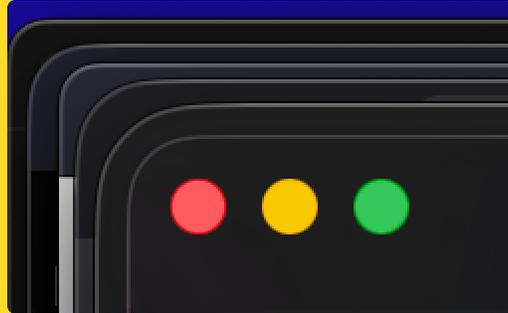
And then argues that the answer is that LLMs are not good enough yet. They can do 90% of the work, so there's still a substantial amount of manual polish, and thus, increased costs.

But I think that's not the real reason. The real reason is: native has nothing to offer.

API-wise, native apps lost to web apps a long time ago. Native APIs are terrible to use, and OS vendors use everything in their power to make you not want to develop native apps for their platform. That explains the rise of Electron before LLM times, but it's also a problem that LLMs solve now: if that was a real barrier to developing native apps, it doesn't exist anymore.

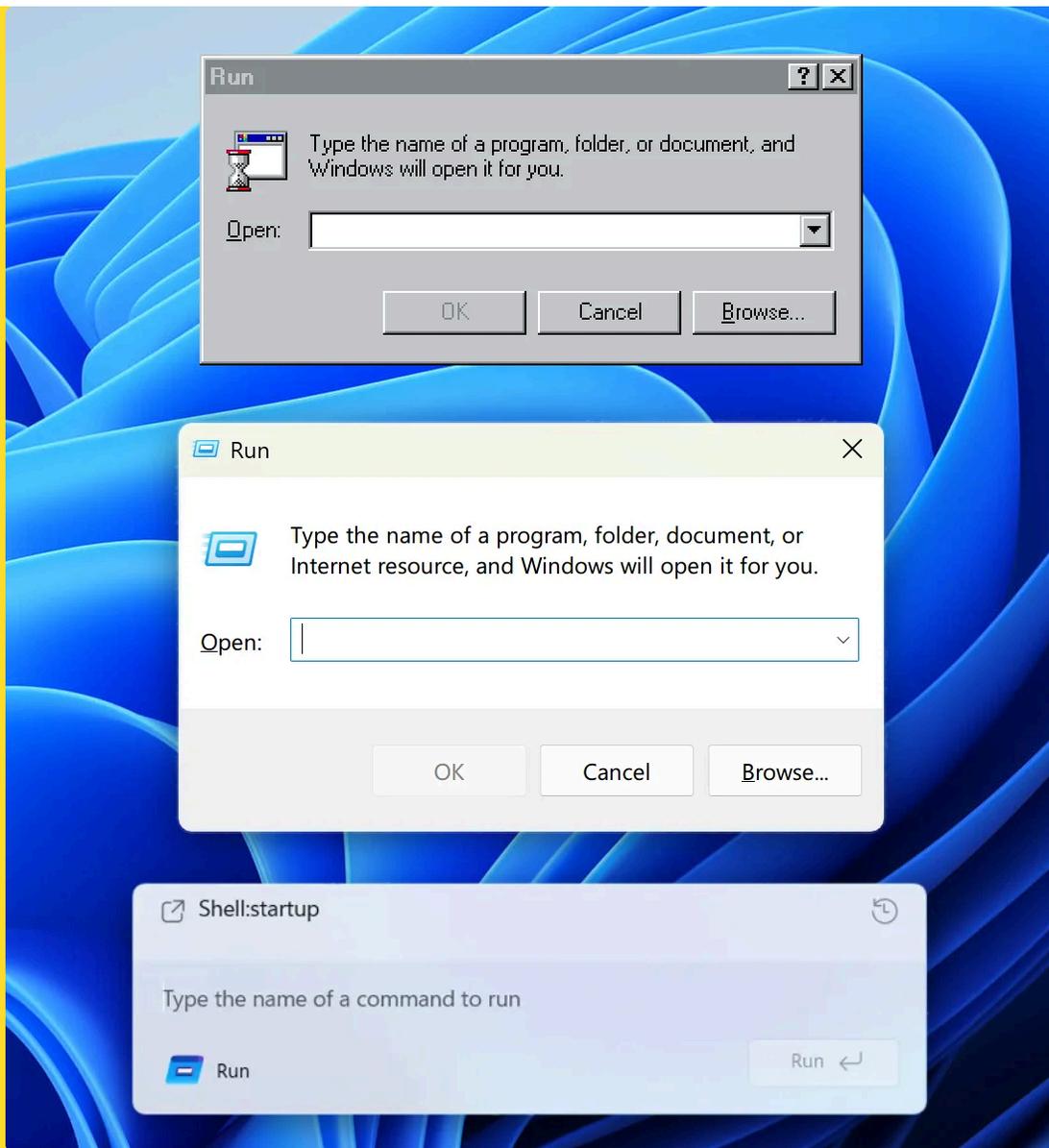
Then there're looks and consistency. Some time ago, maybe in the late 90s and 2000s, native was ahead. It used to look good, it was consistent, and it all actually worked: the more apps used native look and feel, the better user experience was across apps (which we used to call programs).

These days, though, native is as bad as the web, if not worse. Consistency is basically out the window. Anything can look like anything, buttons have no borders, contrast doesn't exist, and neither do conventions. Apple, for example, seems to place traffic lights and corner radius by vibes rather than by any measurable guidelines.



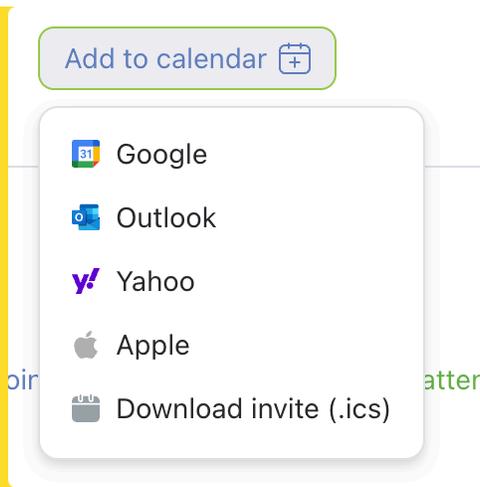
Maybe the server should round the corners?

Looks could be good, but they also can be bad, and then you are stuck with platform-consistent, but generally bad UI (Liquid Glass ahem). It changes too often, too: the app you made today will look out of place next year, when Apple decides to change look and feel yet again. There's no native look anymore.



Computer UIs also degrade over time

Theoretically, native apps can integrate with OS on a deeper level. This sounds nice, but what does that mean in practice? There are almost no good interoperable file formats; everything is locked inside individual apps, most services moved to the web, and OSes dropped the ball for making a good shared baseline. You can integrate with OS-provided calendar, but you can't do it with web calendar. Well, you can, of course, but it's easier on the web; native doesn't help with it at all.



Web pages only lead to more web pages

Finally, the last hope of people longing for native is performance. They feel that native apps will be faster. Well, they can, but it doesn't mean they will. Web apps can be faster, too, but in practice, nobody cares. There's no technical reason why Slack needs to load 80 MiB just to show 10 channel names and 3 messages on a screen. The web is not the problem here! It's a choice to be bad. What makes you think it'll be different once the company decides to move to native?

Don't get me wrong: writing this brings me no joy. I don't think web is a solution either. I just remember good times when native did a better-than-average job, and we were all better for using it, and it saddens me that these times have passed.

I just don't think that kidding ourselves that the only problem with software is Electron and it all will be butterflies and unicorns once we rewrite Slack in SwiftUI is not productive. The real problem is a lack of care. And the slop; you can build it with any stack.

March 3, 2026 · Discuss on [Hacker News](#) [Lobsters](#)

Hi!

I'm Niki. Here I write about programming and UI design [Subscribe](#)

I consult on all things Clojure: web, backend, Datomic, DataScript, performance, etc.
Check out my [Github](#) and get in touch niki@tonsky.me