DAVID HEINEMEIER HANSSON

May 4, 2023

Even Amazon can't make sense of serverless or microservices

The Prime Video team at Amazon has published a rather remarkable case study on their decision to dump their serverless, microservices architecture and replace it with a monolith instead. This move saved them a staggering 90%(!!) on operating costs, and simplified the system too. What a win!

But beyond celebrating their good sense, I think there's a bigger point here that applies to our entire industry. Here's the telling bit:

> *"We designed our initial solution as a distributed system using serverless components... In theory, this would allow us to scale each service component independently. However, the way we used some components caused us to hit a hard scaling limit at around 5% of the expected load."*

That really sums up so much of the microservices craze that was tearing through the tech industry for a while: IN THEORY. Now the real-world results of all this theory are finally in, and it's clear that in practice, microservices pose perhaps the biggest siren song for needlessly complicating your system. And serverless only makes it worse.

What makes this story unique is that Amazon was the original poster child for service-oriented architectures. The far more reasonable prior to microservices. An organizational pattern for dealing with intra-company communication at crazy scale when API calls beat scheduling coordination meetings.

SOA makes perfect sense at the scale of Amazon. No single team could ever hope to know or understand everything needed to steer such a fleet of supertankers. Making teams coordinate via published APIs was a stroke of genius.

But, as with many good ideas, this pattern turned toxic as soon as it was adopted outside its original context, and wreaked havoc once it got pushed into the internals of single-application architectures. That's how we got microservices.

In many ways, microservices is a zombie architecture. Another strain of an intellectual contagion that just refuses to die. It's been eating brains since the dark days of J2EE (remote server beans, anyone??) through the WS-Deathstar nonsense, and now in the form of microservices and serverless.

But this third wave seems finally to have crested. I wrote an ode to The Majestic Monolith way back in 2016. Kelsey Hightower, one of the leading voices behind Kubernetes, put it beautifully in 2020:

> *"We're gonna break [the monolith] up and somehow find the engineering discipline we never had in the first place... Now you went from writing bad code to building bad infrastructure.*
>
> *Because it drives a lot of new spend, it drives a lot of new hiring... So a lot of people get addicted to all the flourishment of money, and marketing, and it's just a lot of buzz that people are attaching their assignment to, when honestly it's not gonna necessarily solve their problem."*

Bingo. Replacing method calls and module separations with network invocations and service partitioning within a single, coherent team and application is madness in almost all cases.

I'm happy that we beat back the zombie onslaught of that terrible idea for the third time in my living memory, but we still need to stay vigilant that we'll eventually have to do it again. Some bad ideas simply refuse to die no matter how many times you kill them. All you can do is recognize when they rise from the dead once more, and keep your retorical shotgun locked and loaded.

ABOUT DAVID HEINEMEIER HANSSON

Creator of Ruby on Rails, co-owner & CTO of 37signals (Basecamp & HEY), best-selling author (REWORK, It Doesn't Have to Be Crazy at Work, REMOTE), Le Mans class-winning racing driver, antitrust advocate, investor in Danish startups, frequent podcast guest, and family man.

Subscribe to get future posts via email (or grab the RSS feed)

Type your email... Subscribe

Sent to the world with HEY