

Issue #31



Helping devtools win over technical audiences

MARKUP MINDSET

www.hackmamba.io



TMM #31 – Benchmark Wars, AI Entry Points, and 17 Lessons from DevGTM Experts



Tahir, Senior Growth Marketer @ Hackmamba

Hey there,

Welcome back to another edition of The Markup Mindset.

I smiled when I saw [Prefect acquiring Dagster Labs](#) this week. If you'd asked me a year ago whether two of the strongest Airflow alternatives would end up under the same roof, I probably would've said no. Here's my bet: we'll see more acquisitions across developer infrastructure this year. Categories are getting crowded, and buying a competitor is sometimes easier than convincing developers to switch away from one.

This week we're covering:

- **How to challenge an incumbent the right way**

- **Are websites getting AI entry points?**
- **17 lessons from DevGTM experts on marketing to developers**

But first, let's hear from our sponsor for this week's edition.



Plan, write, review, and distribute marketing content at 10x speed with Boki.

Most content teams juggle 5+ tools just to ship one piece of content. Boki fixes that. One place to plan, write, review, and distribute, with an AI co-pilot that keeps things moving without the slop. Teams at Cloudinary, HCL Tech, Replit, CodeRabbit, and Mintlify already use it. So do we at Hackmamba, and we ship twice as fast.

Try it free

Let's get started!

1. Every benchmark invites scrutiny.

If you're building a developer tool, at some point you'll have to take on the incumbent. We've seen it with Databricks vs. Snowflake, MongoDB vs. PostgreSQL, Cloudflare vs. Fastly, Redis vs. Dragonfly, Redpanda vs. Kafka, and now Elastic vs. Qdrant. Benchmark reports have consistently been one of the strongest ways to challenge a market leader.



Andre Zayarni • Following

Co-founder & CEO @ Qdrant. Search AI for humans, agents, and robots.

4h • Edited •

Ok, **Elastic**. You wanted it that way. You created a benchmark to show that your paid premium feature is superior to the open-source one, right? You published a benchmark compared to **Qdrant**, right? Oh, Honey, I Shrunk the Search cluster .. here we go. 🍷

Elastic benchmarked Qdrant last week and found we're 7x slower than DiskBBQ. Impressive result. Even more impressive when you consider they achieved it by turning off our async disk scorer, skipping the two-stage retrieval pattern we document for exactly this workload, and then measuring how fast unbounded sequential disk reads are. (Spoiler: not fast. That's why we built the other stuff.) So we ran the same dataset, the same recall target, and the same loaded model — with Qdrant actually configured for disk.

Result: 2x their throughput, half their latency, on nodes with a third of the CPU and RAM. On our smallest config — 2 vCPU / 8 GB — we still beat their published numbers on 7 vCPU / 26 GB pods.

Oh, and DiskBBQ is an Enterprise-only feature that needs over 20 Gigabytes of RAM per pod to keep the JVM happy. Our memory-savvy alternative is Apache 2.0. Full methodology + reproduction kit in the post. Benchmark us however you like — just read the docs first

"The Silence of the RAMs" 🐘 <https://lnkd.in/ej6vQ9Xv> by **Alexis Musaelyan** 🧠

Oh, **Ashutosh Kulkarni** seems you are also in Paris today. Let's meet for a coffee? ☕



[Link](#)

But they also come with a tradeoff. The moment you publish a benchmark, you're inviting your competitor to review it publicly. If they find gaps in your methodology, benchmark configuration, hardware setup, or datasets, they'll publish a rebuttal. That's how almost every benchmark

war plays out. If you're going to make a performance claim, make sure another engineer can reproduce it. The attention from a benchmark lasts a few days. The credibility from a reproducible benchmark lasts much longer.

2. Websites are getting AI entry points

I recently noticed something interesting while using Claude Code with Railway. Instead of loading the homepage I normally see in my browser, it was served a different page built specifically for AI agents. That got me curious, so I went digging and found Railway's engineering post explaining how they approach agent traffic. They don't rely on llms.txt alone. They identify AI agents server-side and return content designed for agent-driven workflows such as onboarding, deployment, and debugging.



Mohammed Tahir ✓ • You

Helping devtools win trust with technical audiences | Full-stack Grow...

1w • 🌐

...

Is [Railway](#) the only company doing this?

I was using Claude Code to fetch something from Railway when I noticed something unusual.

Instead of rendering the same homepage I see in my browser, it opened a completely different page built specifically for AI agents (look at the image below)

That sent me down a rabbit hole.

I knew companies were publishing llms.txt and making their documentation easier for AI to consume. But this felt like a step further.

Railway is giving AI agents a different first impression of the company.

After digging deeper, I found they've openly written about this. They say they've reworked how their web properties respond to AI agents and don't rely on llms.txt alone.

That made me wonder if Railway is the only company doing this?

Not publishing AI-friendly content.
Not exposing Markdown docs.

I mean giving AI agents an entirely different landing experience from humans.

Interestingly, not everyone agrees with this direction. [Evil Martians](#) argues against serving different experiences to AI agents and instead recommends exposing AI-friendly resources without changing what visitors receive.

Either way, I think this is the next frontier.

Five years ago, we debated responsive websites.

Today, we're starting to debate whether websites should have two front doors.

One for humans.
One for AI agents.

Have you seen any other companies doing this? I'd love to know.

I think we'll see more developer first companies adopt this approach. Railway is one of the first I've seen to openly explain why they serve AI agents differently and how they handle it at the server level instead of relying solely on llms.txt. As more agents become the first touchpoint for evaluating products, that entry experience will matter just as much as the human one.

3. 17 Lessons on Marketing to Developers

Last week, I had the opportunity to speak with eight people who've spent years marketing to developers. Folks from PostHog, GitBook, Mozilla, Tinybird, and a few others. I asked them one simple question: "What's one lesson you've learned about marketing to developers that more marketers should know?"

Their answers were refreshingly honest. Some challenged conventional B2B advice, others reinforced things we've spoken about in this newsletter over the last few months. I put everything together, added my own take on each lesson, and turned it into a guide with 17 practical lessons on marketing to developers. If you're building for developers, I think you'll find a few ideas worth stealing.

[Read the blog.](#)

If you're new to The Markup Mindset, we've made our [entire newsletter archive public](#). Feel free to explore past editions if you missed any.

That's it for this week.

If you've been enjoying The Markup Mindset, I'd love to hear your thoughts. Tag me on LinkedIn and let me know what's resonating or what you'd like to see covered in future editions.

And if you think a friend or colleague would find this edition useful, feel free to share it with them.

See you in the next edition.

Tahir

More from us:

1. [10 developer tool homepages that get one thing right](#)
2. [6 best SEO agencies for dev tools \(2026\)](#)
3. [What is a developer advocate? The role, the work, and how to get in](#)

Helping devtools win over technical audiences

MARKUP MINDSET

Hackmamba Inc.

651 N Broad St Suite 201, Middletown
United States of America

You received this email because you signed up on our website or made a purchase from us.

[Unsubscribe](#)