We've Rebuilt Buffer's Composer From the Inside Out

If you use Buffer, then you're familiar with our Composer. It's where every post begins. Where you draft, tweak, preview, and hit publish. It's the most-used surface in the entire product, and it's the core of our value.

For a while now, though, the composer has been showing its age. Not necessarily to users (it worked!), but under the hood, it was built on legacy code that made it increasingly difficult to maintain, improve, and extend. Every small change carried risk. Every new feature required navigating layers of complexity that slowed us down.

So we decided it was finally time to do something about it. Over the past few months, a small team at Buffer has been modernizing the composer's architecture and appearance. Not just to add new features, but to build the foundation for everything we want to do next.

Here's what we did, why we did it, and where it's all heading.

The appetite for change

This project didn't happen in a vacuum. Over the past year, we've been making meaningful changes across Buffer, from scheduling improvements to dark mode support to a visual refresh that's been rolling out across the product.

Each of these changes created momentum. They also created an expectation: The rest of the product should feel this good, too. The composer, being the centerpiece of Buffer, was the obvious next step.

There was a practical side to this as well. When I spoke with 30+ Buffer users last year, a consistent theme emerged: people want Buffer to feel like a cohesive, modern workspace, not just a scheduling tool.

Modernizing the composer is a direct response to that. If we want Buffer to be the place where you brainstorm, draft, collaborate, and publish, the Composer has to be up to the task.

What we did

Let me be upfront: this phase of the project was never about shipping something flashy. It was about doing the unsexy-but-essential work of rebuilding the foundation so we can move faster and build better going forward.

The composer relied on outdated data stores that made it unpredictable to work with. We introduced a modern state layer that’s leaner, easier to reason about, and gives us a much clearer picture of how data flows through the composer. Data now loads and moves more efficiently.

We also moved the composer to our current component system, which we call Popcorn. This brought the composer in line with the rest of Buffer’s interface and unblocked improvements we simply couldn’t make before, like new side panels, better responsiveness, and full support for dark mode and our visual refresh.

Speaking of side panels, we did a complete overhaul of how tools live and operate in the composer. Side panels are now more responsive and predictable, and we extended post previews

to all selected channels, even in the omni-composer mode.

Channel fields, media handling, and dialogs also got an update. These are the kinds of details users may not consciously notice, but they make the experience feel smoother and more consistent, contributing to a composer that feels more polished and reliable.

As mentioned earlier, this phase wasn't about flashy updates, even if the composer does look, feel, and operate better. This was about laying a better foundation for even greater changes coming soon.

How we made it happen

Things rarely happen in a vacuum at Buffer. We're a small, close-knit team, where collaboration and transparency are the default, and a lot of momentum for this project came from broader efforts aimed at making all of Buffer look and feel like the first-class product that it is. Engineering, design, and product all played a role. But, our customer Advocates deserve a special shout-out. The composer is a sensitive surface. Users rely on it every day, and even small changes can feel disorienting. Our Advocates did an incredible job navigating customers through the migration, gathering feedback, and helping us understand where things felt off.

That kind of cross-team collaboration is something I'm genuinely proud of at Buffer. No single team could have pulled this off alone.

Just the starting line

Here is the part that I'm most excited about. The entire point of this modernization was to unlock what comes next. We're already building on this foundation. Templates will soon be surfaceable directly in the composer, so you can start from

proven formats without leaving your workflow. We're working on deeper support for notes and content feeds, aiming to move Buffer closer to being the creative workspace our users have been asking for.

And, most importantly, we're exploring a more ambitious composer vision: one with post groups, improved drafts, the ability to plan out ideas and content on the calendar, and an overall smoother and more complete content creation workflow. None of this would be realistic on the old architecture. Now, it's not only realistic, it's also the plan.

If you've been following along with our thinking on making Buffer more than a scheduling tool, but a true social workspace where you can brainstorm, organize, publish, and analyze, then this project is a concrete step in that direction.

Building with momentum

One of the things I love about working at Buffer is that we build in the open. We share our thinking, our roadmap, and our successes (and failures), because we believe that transparency makes for a better product and a stronger relationship with the people who use it.

The composer modernization is a good example of what intentional product development looks like. Not every project results in a big feature announcement. Sometimes the most impactful work is the kind that makes everything after it possible.

We're building with momentum now, and every change builds on the last. If you're a Buffer user, you'll start feeling the difference, and there is a lot more on the way.

As always, if you have feedback on how the composer feels or ideas for what you'd love to see next, we'd love to hear from

you. These conversations are how we build the best version of Buffer — together.

https://buffer.com/resources/new-buffer-composer/