EPISODE 945

[INTRODUCTION]

[00:00:00] JM: Frontend software development has become as complex as backend development. There was a time when frontend web development was simple. There were a small number of JavaScript frameworks and templating systems. Your CSS was simple configuration for the colors on your webpage. Today, there is a giant ecosystem of frontend tools, APIs and middleware for delivering data to the user.

Gatsby is a framework based on React that allows developers to build performant web applications. Gatsby is not easy to explain. In some ways, it's like a compiler for your website. Gatsby pulls in the data that you need to build your website, including CMS data, APIs, and markdown, and then it links that information into React components and CSS. This happens on the server, so your user gets served a website that does not require lots of roundtrips as your website renders for the user.

Kyle Matthews and Sam Bhagwat are the founders of Gatsby, and this company is based around the open source GatsbyJS project. They joined the show to describe their vision for the framework and their vision for the company.

[SPONSOR MESSAGE]

[00:01:29] JM: I remember the days when I went to an office. Every day, so much of my time was spent in commute. Once I was at the office, I had to spend time going to meeting rooms and walking to lunch and there were so many ways in which office work takes away your ability to be productive. That's why remote work is awesome. Remote work is more productive. It allows you to work anywhere. It allows you to be with your cats. I'm looking at my cats right now. But there's a reason why people still work full-time in offices. Remote work can be isolating. That's why remote workers join an organization like X-Team.

X-Team is a community for developers. When you join X-Team, you join a community that will support you while allowing you to remain independent, and X-Team will help you find work that

you love for some of the top companies in the world. X-Team is trusted by companies like Twitter, Coinbase and Riot Games.

Go to x-team.com/sedaily to find out about X-Team and apply to join the company. If you use that link, X-Team that you came from listening to Software Engineering Daily, and that would mean that you listen to a podcast about software engineering in your spare time, which is a great sign, or maybe you're in office listening to Software Engineering Daily. If that's the case, maybe you should check out x-team.com/sedaily and apply to work remotely for X-Team.

At X-Team, you can work from anywhere and experience a futuristic culture. Actually, I don't even know if I should be saying you work for X-Team. It might be more like you work with X-Team, because you become part of the community rather than working for X-Team, and you work for different companies. You work for Twitter, or Coinbase, or some other top company that has an interesting engineering stack, except that you work remotely.

X-Team is a great option for someone who wants to work anywhere with top companies maintaining your independence, not tying yourself to an extremely long work engagement, which is the norm with these in-person companies, and you can check it out by going to x-team.com/sedaily.

Thanks to X-Team for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[00:04:05] JM: Kyle and Sam, welcome to Software Engineering Daily.

[00:04:08] SB: Thanks.

[00:04:08] KM: Yeah, thanks for having us.

[00:04:10] JM: We did show a few years back about Gatsby. Explain what purpose Gatsby serves.

[00:04:17] KM: Gatsby is a simple way to build websites with React. React is of course an extremely popular JavaScript framework that's out of Facebook and it's been growing incredibly fast like the last –What is it now? 2013. So like five, years six years.

So back in 2015 I started using React pretty early and I loved it, and then I need to build a website for my startup at the time and out came Gatsby. It's kind of tied together all the different tools you need to use React for website to make it pretty straightforward.

[00:04:48] JM: How does Gatsby relate to GraphQL?

[00:04:51] KM: The initial version of Gatsby was pretty simple. You use like markdown files to provide kind of content to your site to create kind of like blog posts or whatever and then use like GSX to like build out the other pages and so forth. But what I realized, and I was watching what people are using with like the initial version of Gatsby is that they had much more ambitious plans. They are trying to hook it up to see MSs.

So they are doing like weird stuff like writing scripts to hit APIs, to pull down to write it down into files, which then Gatsby could use to build the site. I was like, "Okay. That's really cool and really interesting. What if we made that way, way easier so that –" Because it's actually a very compelling idea that you could like hookup React easily to external data sources. What if we made that really straightforward?

So the next version of Gatsby had this idea of like source plugins that you could like drop in to connect Gatsby to another third-party data source, and then also GraphQL as kind of a universal API so that any data sources that you plug in you could then query in like the exact same way across the whole thing, which is kind of like GraphQL's sweet spot for kind of wrapping a bunch of external APIs.

[00:06:01] JM: Give me a holistic overview of the typical Gatsby stack.

[00:06:10] KM: Typical, that's tough. Gatsby is intended to be very flexible. So people use it in remarkably different ways, but I guess like some constants is React and GraphQL, and the source plugins, the connect to anything that has an API basically, the community has now

created something like 350 different source plugins. Any sort of data you can imagine basically you can pull into a Gatsby.

Anything from like traditional CMSs, like Drupal and CMS, to kind of new age, like headless CMS, like Contentful, or Sanity, or Data.CMS, etc., two things that you don't even think of a sort of a CMS or data source or whatever you'd use for building a website. But if you kind of squint and shake your head around a little bit, it actually makes a lot of sense. Stuff like Airtable or Google Spreadsheets are great ways to build sites with.

We actually recently created like an event page on Gatsbyjs.org so that any time somebody is running a Gatsby event anywhere in the world, we want to put that on like kind of Gatsby events and so forth. So we just made an Airtable table, and now our marketing team can get submissions to put that there and then like the site rebuilds and pulls it from the Airtable API, and it's very straightforward for the marketing team to maintain as well as to like build out the page on our site.

[00:07:33] JM: We did a show while you about data connectors in the backend context. It was about ETL, basically. So a lot of people will have this kind of need where they've got Salesforce, or they've got Stripe and they want to import that data on a regular basis into their backend so they could do database joins with that data. It sounds like what you're talking about is a similar concept but for on-the-fly data loading for your frontend application.

[00:08:08] KM: Yeah. I mean the same trends that's making – Like combining all the services relevant to backends is also kind of heading website creation, because 20 years ago, everyone ran their own servers. There wasn't really this idea of like SaaS anything. To build a website, to build an application, you would download the software or write the software and install it on your server and run it and it was all kind of maintained within your own kind of platform.

But the world today is there's like all these like very mature cloud services that handle most things that you want. So the process of building any sort of software, including websites, is more about picking the different pieces that you want and then combining them together, like kind of integrating them together. That's what's Gatsby's goal is here, is to make it really easy to say, "Okay. Airtable is the bus place to store this data. Contentful is the best place to put this data."

Then we also have a Google Spreadsheet that we drop in to kind of drive this like kind of table view on this one page or whatever. You have to try like to like cram all the data into one CMS or some other form. They can live wherever they should.

Just like Stripe of course is the obvious place to have like financial data about like transactions and whatever. Salesforce holds a lot of like sales and marketing related data and so forth, but you don't really combine them, per se. They live there. That's the source of truth, but then you combine them on-the-fly to do whatever you need to do in your system.

[00:09:46] JM: Gatsby to me seems like a project that really resists analogy to anything. I mean, it's very hard to draw. I'm trying to think it's like kind of like a compiler, sort of, like you pull – Except just like pulling in all your data from your different data sources and like doing a build. It's kind of like an ETL tool, because it's like pulling in data from external sources and transmogrifying it so you have like a readable materialized view that happens to be a website. But all of these things are – It's just not good to describe it by analogy. I think that's one of the reasons why people who have not worked with Gatsby have trouble understanding what it is.

What's the most common misunderstanding about Gatsby?

[00:10:37] KM: Yeah, it's actually really funny that you say that, because a lot of people do try to like jam Gatsby in a box like, "This is the box that I want Gatsby to be." It's like kgh kgh kgh, kind of like bang it in and you're like, "It doesn't fit."

I think the most common misconception and it's kind of our mistake, because this old idea like static site generators have been around for a while, and when I initially released Gatsby, that was kind of the idea. I was like, "Oh! Gatsby is like a React static site generator." But the problem is that people have very strong preconceived notions of what a static site generator is. Largely that it's static and like that word static is our least favorite word now, because like people have – They just have very strong misconceptions about what that implies about what Gatsby does. So they think that like Gatsby can't do all-like stuff.

Anyway, so the intention of calling Gatsby a static site generator was just that it produces static files, and so you can deploy it on a CDN. You don't need a running server to do it, but it

produces a React app. It's just like create React app or any of the other kind of React frameworks out there. It's like no different from them. It's like they're static site generators too if you want go down that.

Anyways. Yes, that's something we're continually trying to kind of back out of, is like, "NO. Gatsby is an amazing way to create pretty much any application. It's not just for websites. It's not just from like static content that you can like – That's like kind of a full-fledged web framework.

[00:12:13] JM: I always thought, "Static site." I was like, "That is not the best marketing term." You guys are digging yourself into a hole there.

[00:12:22] SB: Yeah. What's funny is that I was on the – When V1 came out, I was on the edge of just like, "Nope. We're going away from this." I was like on Twitter was like, "Should we do that?" Then there's a bunch of people were like, "No. Keep it. It's simpler to understand. It's more approachable." Kind of like with marketing, you like want to kind of target – People have existing mental models and you're like, "Okay. It's like that, but a little bit different sort of thing."

That was kind of their argument. I was like, "Ahhhh." I was like, "Okay." I went with it. But then, yeah, in retrospect. It's like about a year ago we killed it off our website in GitHub, but it's like a bad disease that just keeps coming back. We can't quite rid it out of people's minds.

[00:13:03] JM: While we're on the subject of questionable terminology, what do you think of the term JAMstack.

[00:13:12] KM: I think it's tasty sending.

[00:13:15] SB: The interesting part of the term JAMstack is that is refers to a deployment paradigm. It refers to the idea of generating files that can then be served from a CDN and there are very real benefits from that security performance. Where JAMstack may be falls short is that it doesn't encapsulate the real modern development paradigms that have emerged over the last few years. So component-based such React and Vue.

JAMStack is kind of agnostic to that where, really, if we're thinking about the future of building for the web, the future of building for the web is frameworks like React and Vue that have a virtual DOM, that have a component model. I would say it's just not opinionated enough, because Matt Netlify who came up with it, it's like he had the same problem, dilemma, that we do, which is like people think static sites and they're like, "Oh! That's just his like very limited in what it can do." He was looking for a name that would like kind of shift people's thinking to like something much more expensive. Yeah, it's just like JavaScript APIs markup is a very generic sounding, because mostly when you say a stack, it's like more specific and like more opinionated. So that's what we're trying to do with Gatsby. It's very much like in line with the same idea, but more opinionated and like how you should build stuff, I guess.

[00:14:45] SB: People don't think of themselves as JAMStack developers. They think about themselves as React developers, or Vue developers, or Gatsby developers, or whatever. Ultimately – I mean, people's identity tends to center around the thing that they're building in.

[00:14:59] JM: When you say people think of themselves as Gatsby developers, what is a Gatsby developers do?

[00:15:05] SB: Use Gatsby mostly.

[00:15:07] JM: I mean, they still need to use Vue or React or something, right?

[00:15:11] SB: Right. It's just like it implies a further specialization within the React realm basically. Because you can do a lot of stuff with Gatsby that you can't do with like Vanilla React.

[00:15:24] JM: Give me an example.

[00:15:25] SB: I mean, just building a website CMS data is dramatically easier with Gatsby, because React is very – React's surface area is quite small. It's like the original marking was like it's just the view, and that literally is still just what it is. It just is like a toolkit for creating views within some sort of UI whatever. But everything around it like, how does the data get there? How do you build the site and deploy the site and whatever?

[00:15:55] KM: How do you make it really fast?

[00:15:56] KM: Yeah. How do you make it fast? React has zero opinions about it. What Gatsby does is it steps in and says like, “Hey! We’re going to take you from the great starting point that React’s left you at and take you all the way to the finish line so that you can just install Gatsby and you have like opinions, you have something that's like going to help you ship something that's really awesome.”

A lot of people that are like, “React is fun,” and they play around it, whatever, but then they start using Gatsby and then they get tremendously excited because Gatsby gets them to that finish line. They want to ship something and they want to get something. React is un-opinionated enough that it can be frustrating to get started with.

[00:16:39] JM: You’re the cofounders of the Gatsby company, describe the vision for the Gatsby business.

[00:16:47] SB: So we see like Gatsby business is really a way of sustaining the Gatsby open source community. There's been a lot of conversation around the right way of having sustainable open source communities, and there are several different models all which play some role. There's corporate sponsorships, there's like React, kind of the React model coming out of Facebook. There is individual sort of heroic developers like Tobias of Webpack and Henry Zhu of Babel that find a way of making the dollars and cents work for them to go fulltime or something close to it on their the project.

Open source is a tremendously valuable thing for us as developers, as engineers, because it creates great tools for us to use. The challenge with open source is the investment model of how you maintain it and make it kind of better overtime. The first thing is that Gatsby as a business is or mission is very much kind of make Gatsby the open source project better and make the community long-lasting and make Gatsby sort of an entering framework.

Yeah. Like 10, 15 years ago, the idea of like an open source company or commercial open source as silly maybe, like people are like, “What does that even mean?” But I think we’ve seen a lot of really great companies emerge over the last while that really kind of – We’re definitely

not following in their kind of footpaths, like Confluent and HashiCorp and Elastic, and MongerDB, etc., Mulesoft. These are all like really great open source projects but also have a very natural kind of business model tied to the open source project that kind of – It feels like done right, it can be a very virtuous cycle, that the open source helps improve the business and then the business helps improve the open source and that a lot of people can benefit from both.

[00:18:38] JM: Of those businesses you mentioned, the one that – I actually don't know much about Mulesoft, but I do's feel like Gatsby is probably most like Mulesoft, because it's kind of like middleware, like these other companies?

[00:18:53] SB: The founder of, actually, Mulesoft, Ross Mason, is one of our seed investors, because we got in touch with them specifically because we're like, "That's kind of similar." The first we talked to him, he's like, "You guys are doing the same thing that we did," type thing.

[00:19:06] KM: Mulesoft for websites I think was his turf. There're a lot of strong similarities there.

[00:19:10] JM: Right. Mulesoft, what was the term that they were able to like go to market with? Integration or something or like –

[00:19:21] SB: I don't remember the precise term, but they viewed themselves very much as like that sort of data middleware and kind of the connector layer between all the different – Sort of like an enterprise, if, this then that, or Zapier connecting all the different –

[00:19:36] KM: Yeah, because the problem of like having data in whole bunch of different places emerge first in the enterprise, because they created their own mess of having all these different systems. Now, stuff like Mulesoft, is necessary for everyone, because everyone has data kind of spread across different systems that needs to integrate it somehow.

[00:19:52] SB: There's HubSpot, and there's Saleforce and they're like Marketo, on and on down the list of like, to Kyle's point earlier, enterprise SaaS products that need to have kind of workflows without them moving from place to place.

[00:20:06] JM: What did Mulesoft sell?

[00:20:07] KM: There's two ways of answering that question. The first thing is they sold a product that let you move data and easily integrate any kind of two systems in your sort of enterprise organization, whether that's like HubSpot in Salesforce. Define ways of getting data from point A to point B and then sort of automatically getting it. The second way of answering that is saying that they sold digital sort of transformation and letting companies decompose themselves into kind of microservices and then create kind of new products out of there. Kind of create API-driven development products.

[00:20:43] JM: Do you know how they're – Did they end up as a services company mostly or did they –

[00:20:48] KM: I'd say mostly like enterprise sales. I mean like –

[00:20:50] SB: Yeah, they're on-prem for the most part. I mean, they sell to large companies that want to install Mulesoft in their own servers and stuff.

[00:20:58] JM: Okay. It is mostly open source stuff, right?

[00:21:03] KM: Yeah. Sort of the core of it is open source and all the connectors are – The connectors are kind of open source. I mean, Joseph Jacks has a great sort of series on open core and he kind of has like a hierarchy of like how much kind of closed commercial stuff is around the sort of open source core. You could see something like GitHub more or like, okay, gets at the core. There's a ton of like collaboration software that's in GitHub that is not open source. I think Mulesoft, like very similar, the core is like the static connector piece which is open source, but then like they've created a lot of these kind of workflow- hosted sort of tools around that.

I mean, every open source company sort of makes a decision of kind of how much is open source. How much is like a hosted kind of cloud version? How much is maybe an enterprise version? I mean, these are – Mulesoft certainly answered those in one way. I don't think we

would necessarily answer that question in the same way. I think there's a spectrum for open source businesses.

[SPONSOR MESSAGE]

[00:22:14] JM: Cruise is a San Francisco based company building a fully electric, self-driving car service. Building self-driving cars is complex involving problems up and down the stack from hardware to software, from navigation to computer vision. We are at the beginning of the self-driving car industry and Cruise is a leading company in the space.

Join the team at Cruise by going to getcruise.com/careers. That's G-E-T-C-R-U-I-S-E.com/careers. Cruise is a place where you can build on your existing skills while developing new skills and experiences that are pioneering the future of industry. There are opportunities for backend engineers, frontend developers, machine learning programmers and many more positions.

At Cruise you will be surrounded by talented, driven engineers all while helping make cities safer and cleaner. Apply to work at Cruise by going to getcruise.com/careers. That's getcruise.com/careers.

Thank you to Cruise for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:23:32] JM: These open source businesses that have found success in the past, they are historically –I mean, this is a very short history that we have obviously. It's tricky to derive based off of history, but they're mostly backend companies. They're like these huge enterprise endeavors like, "Okay. You want to get your Hadoop off the ground. We've got a ton of software to give you. We've got a ton of consultants to give you and it's going to cost you millions and millions of dollars.

[00:24:02] KM: Emphasis on the millions.

[00:24:04] JM: We haven't seen that on the frontend yet, right? Sort of.

[00:24:07] SB: Well, that's funny. I would summarize all open source companies, they make their model basically from infrastructure collaboration. GitHub is a great example. Git is open source, but what are you paying for? You're paying for the hosting of the git and you're paying for the collaboration tools around git, because like pull requests make git a heck of a lot nicer to use because it kind of drives a really pleasant workflow, which has real value in companies. Git is open-source, but GitHub is not, etc.

Yeah. The frontend, we're not going to make money. It is hard, like as you said, to make money from the frontend. But what are we offering to people commercially? We're offering infrastructure and collaboration.

[00:24:48] KM: At that, there is a healthy sort of tradition of businesses around website framework. Automatic around WordPress and [inaudible 00:24:56] around Drupal are two of the best known cases, although there are several. [inaudible 00:25:01] was recently acquired for a billion dollars. Automatic just raised more money at a $3 billion valuation.

[00:25:08] SB: I used to work at Pantheon, which is also Drupal WordPress hosting. Also, interestingly, they do really great hosting, but they also have collaboration tools, which was kind of their real and still is kind of one of their key distinguishing parts of their business, in their product.

[00:25:22] JM: I mean, those companies are so good because nobody ever churns. You never ever churn.

[00:25:26] SB: Yeah. I think the Squarespace basically.

[00:25:29] KM: Yeah. I think about like websites that folks are building, and this is kind of the other piece of what Gatsby is, is that the content management system of today, whether that's Drupal or WordPress or another one is typically a monolithic application. It glues your development toolkit with your content editing interface, with your database is actually storing that content with the whole sort of suite of other functionality, whether that's like forms or like AB testing, maybe personalization, maybe analytics, and then maybe you've got – Maybe it also

includes like your hosting. Maybe it's got your e-commerce functionality. CMS's of today are very kind of monolithic stacks. Once you're in on the stack, you're kind of in on that stack. Whereas the way that when we see the world is a much more sort of like flexible way.

[00:26:20] SB: It's kind of distributed modular CMS of sorts. Gatsby is like kind of the key orchestration bit to kind of make it all run. Kind of like we're like the operating system of the distributed CMS.

[00:26:33] KM: We call is the content mash because, ultimately, what people need is like, "Oh, hey! I've got my – My writers want their blog in WordPress. Maybe you've got this e-commerce thing, so I need Shopify as a backend. Then I want to do my UI in React, because I want a modern development toolkit so my developers can move quickly and, gosh, now I need some way of gluing these stuff altogether that works kind of out of the box."

[00:26:59] SB: Because like in the WordPress and Drupal, it's like youronly option is just to keep jamming more and more stuff into the CMS. WordPress, you're like, "Okay. WordPress," and then we'll toss [inaudible 00:27:09] and whatever. Then you have like 30 plug-ins doing all these different things of plugins of questionable quality and security and so forth.

With Gatsby, it's like, "No. Just pick the best of breed third-party manage service to provide whatever functionality you need." With Gatsby, there'll be a plugin that provides integration and it'd be very smooth to just kind of like slot it in and do exactly what you want.

[00:27:35] JM: I believe you in this world. This future world of – No. Seriously. As somebody who – Software Engineering Daily runs on WordPress and, look, I love WordPress, but it's not – I don't know how to really work it. I'm a software engineer, I'm like I don't want to touch this thing. I don't know how it works. It's literally how I felt when I worked on like Java model at a corporate job. I'm like, "I don't want to trust this thing."

[00:28:03] SB: I mean, WordPress is from a different world. If you're software engineering came up today, the LAMP is mysterious relic. It's technology. You're like, "Whoa! THs is what my ancestors did," sort of thing. I started engineering, like developing right around – It's kind of like the heyday. It was like kind of mid-2000s, and that was my thing. But even by like 2000's it was

like, “Now, APIs and JavaScript web aps are already starting to like clearly seem like the future in like early 2010s,” that’s when I switched to it and haven’t touched LAMP stuff since.

[00:28:38] JM: Yeah. I mean, there's this area of WordPress where like you couldn't like modify the code and it’s like warning. It warns you. Do not do this. I’m like, you convince me. I’m just not going to touch it. When we go find a plugin, this is going to cram even more PHP into my app. So that's why say I totally believe in the world that you’re building.

What I wonder is how do you make sure you capture enough value of that world? Because in the WordPress world, it's monolithic. Yeah, WordPress captures is all, whether your host is automatic or its WP engine, or whatever, Pantheon. Whoever your WordPress host of choice is. That’s who's capturing basically all the value in –

[00:29:28] KM: Well, I would actually push back a little bit against that, because if you compare WordPress or Drupal versus enterprise CMSs, the hosts are clearly capturing far less of the value than enterprise CMS does. Everything is proprietary, because something like Psycore EPiServer, etc.

[00:29:45] JM: Okay. I don’t know these things.

[00:29:47] KM: Yeah. Anyways, it's a whole different world. Anyways, they routinely sign tens of thousands, hundred thousand million dollar contracts. There's a lot more money –

[00:29:56] JM: So you’re potentially an enterprise software. You're not just for hipsters.

[00:30:01] KM: Yeah.

[00:30:02] SB: Certainly overtime, part of our journey as a commercial company that's kind of, again, sustaining the open source ecosystem is like starting with kind of common tools that like website teams need to use. Whether that's – Right now, that's Preview. The ability to see for marketers to see content before they go live.

Some of the things we have on the roadmap are diving much more into a content kind of collaboration piece. In many cases, folks using Gatsby are in some cases in kind of more modern chops. In some cases, starting to get into these larger orgs with these kind of like huge CMS contracts and ASCII. Looking for a kind of what's the future of our like websites stack.

[00:30:46] JM: Okay. The first product is Gatsby Preview. That, as I understand is, typical workflow is – I mean, this is kind of like keying into the burgeoning freelancing contracting software development world. Basically, there's this huge army of people who are out there just freelancing and maybe they're like entrepreneurs 40% of the time, freelancer 60% of time. In their freelancing work, they're building a site for somebody else and they need to show the site to another person or like give them read-only privilege. That's what Gatsby Preview is, correct?

[00:31:24] SB: Yeah, exactly. It's like you make a change and you're like, "Should this go live, maybe you want to see it yourself. What's it's going to look like before I publish it? Or you want to show it to other people and get their feedback. Yeah, the whole goal is that you get that feedback immediately. A lot of preview tools, where they fall down is like the feedback loop is pretty slow.

Imagine, if you're developing and you change some code and it takes like two or three minutes to see how it affects your actual running application, the feedback cycle is so slow that it takes you forever to get anything done. What we've done with Gatsby Preview is made like the feedback like instantaneous almost. It's like within a few seconds, when you update like content in the CMS, you then see the update on the actual real website.

So that allows you to have like very kind of experimental iterative kind of approach to modifying pages. You can be like, "Oh! What if I do this? What if I change the title here? What if I upload this photo instead of this other photo." What if I like reorder these things elements on the page and whatever? You can kind of go from this sort of abstracting out CMS content forms to like the real thing and kind of like feel out very quickly what's going to work and what's not going to work.

In a lot of ways, editing CMS content is kind of like coding in a way, because you have this sort of abstract form that then gets like kind of hydrated into like the real working website, and you

need that immediate connection between I change this thing in the abstract form and then see what it becomes for real in the actual website.

We've done like a lot of work to kind of make that work with almost any data source, and it's pretty cool. We have 60 customers today. We're in open beta still. Going live fairly soon, but, yeah, we have 60 customers to date and like people are using it a ton and really liking it.

[00:33:16] JM: These are mostly like freelancers or what?

[00:33:20] SB: It runs the gamut. There is like some small sites, whatever, that freelancers build. There are a lot of companies that have like rebuilt their own site in Gatsby and then need Preview to kind of help everyone maintaining the site work on it. We're starting to see like large enterprises like experimenting with Gatsby in a bunch of different ways. Some of them have signed up for Preview as well.

[00:33:42] JM: Of all the products you could've built first, why did you build Preview?

[00:33:46] SB: It was something that there was a very clear need for. Back when we or even before when we were just starting like, "Oh, maybe this could be a business," and just talking to Gatsby users. A very common thing people would say is like – We talk to a developer and they're like, "I love Gatsby so much. It's amazing." Then we're like, "Oh! That's awesome. You build a site." They're like, "Yeah, that was amazing. I love the development experience. It's so smooth and the site is so amazing."

Then they were like, "Oh, yeah. There's one weird thing though, is that I introduced it to the marketing team and they hate it, because all of a sudden they change stuff in the CMS and they can't see a preview. They click save and they have no idea when it's going to go live." Gatsby dramatically improves the developer experience and kind of that the end-user experience, but there is a clear regression in kind of the experience for kind of marketer content people. So that's something that was on our mind from the get go. We're like, "Okay. How would we fix that?" It was something that was clear that this would be a kind of ideally suited for a cloud service that we could offer and it would be really valuable for a lot of companies like as we're seeing.

[00:34:52] JM: This is like the marketing team is managing a blog, a HubSpot blog, and they want to be able to use the backend HubSpot blogging tool, but not whatever like jenky default frontend HubSpot blogging provides them. They would rather use custom, like trendy frontend theme that is composed via Gatsby. But they're not able to preview it or they were not able to preview it.

[00:35:27] KM: Yeah. I mean, typically, it was like the developer that was like pulling it in, because they would be tasked to build a website and they're like, "Well, I'm going to Gatsby, because that's the best way to build a website." Then they find [inaudible 00:35:37] and that it's like not meeting some of the needs of the marketing team, because you can think about Gatsby as like decoupling the code from the content. The developers can see both the code and the content, but marketers can only see the content.

Then the system that they're used to being the source of truth is you must. There's no longer the source of truth, because that is just data that's being pulled into the website, the development experience. So you need kind of Gatsby Preview to then serve as that source of truth about, "Hey, this is what it actually looks like. When you save, this is what's going to go live." Does that header overflow the bounce? Does –

[00:36:20] SB: Yeah. Our goal of Preview, and we're also working on like something to solve kind of the slow deploy time called incremental builds. The goal is that people working in the CMS, they won't even need to notice that it's been moved to Gatsby. The site has been moved to be using Gatsby, because it's kind of like you have that the frontend and the backend and our goal is just we have Gatsby be just like this perfects drop in replacement for existing frontends. It's like WordPress is an amazing CMS? People love – There's funny parts, but like the actual –

[00:36:54] JM: Oh! It's amazing. There's a reason I use it.

[00:36:57] SB: Yeah. The experience of editing, creating content, is like very smooth and very polished. They've spent absolutely enormous my time doing that and there's no reason that WordPress should not still be like amazing. But WordPress is much slower. It's much more – It's like the websites are slower. It's much more costly to kind of run, because you have to have like

running PHP in a database. There's constant security problems, etc., etc. Also, it's uses like PHP templates. It's hard to do kind of modern JavaScript app type stuff. That would be ideal.

Yeah. WordPress frontend is not nearly as nice as the WordPress backend. The same thing is true of a lot of other CMSs. We're talking to a lot of kind of traditional CMS vendors including all these enterprise CMS ones, and that's what they're all telling us, is that like, "Oh, yeah. We hear tons of demand to add support for like React and we're not really sure what to do." But then they see Gatsby and they're like, "Oh, okay. That's actually like a workable solution, because they can just – They can work with us to build a really nice integration and then development teams can just slip in Gatsby, but the experience using the CMS doesn't change for everybody."

The sites can get much faster. Hosting can get much cheaper, and the velocity of development teams can be much higher because they're using kind of a modern development stack. It's like all the benefits of the modern frontend without the downsides of breaking workflows for other people involved with the website.

[00:38:27] JM: What I wonder is – Again, I'm bought in to the future. I'm bought in to the future that you're selling me. I wonder how fast that future is coming and how big the market is today. The reason I bring that up is because you guys raised $15 million in a series A, which is awesome. Congrats. I don't blame you, but that's like a very large round. This is kind of – I mean, you guys know the market better than I do. It seems pretty nascent. It seems hard to build later like –

[00:39:08] KM: This is actually a fairly – The market is actually huge. Everybody needs a website.

[00:39:13] SB: The internet is a very, very large place.

[00:39:18] KM: If you look at the business analysts of the world, they'll tell you that CMS is a $6 billion a year market or $10 billion a year depending on how they cut it. My argument is that it's actually like larger, because people don't think about Shopify, for example, as a CMS. Shopify is a CMS. I mean, it's a very well and vertically integrated CMS. It does everything from hosting down to all the e-commerce plugins that you need to kind of provide and sell your goods and

handle the kind of tax implications. They've built a $40 billion business out of that by market cap.

Then when you start looking at like all the other things that go into kind of websites, if you think about like payments, AB testing, personalization. I mean, payments, again, like Stripe by itself. That's like payments on a website. That's like I won't say that's like all they do, because they do a lot of things, but that's the core of what they do, and Stripe is like a $35 billion company, because payments on websites is a huge problem to solve.

Again, each of these pieces of the website world are huge categories with like potentially large existing or potentially large businesses in them. We see Gatsby at kind of the center of all these kind of being the thing that like glues and stiches all these things together.

[00:40:43] SB: Yeah. I mean, there's already like roughly 60,000 Gatsby sites and we doubled in the last like five months or something, whatever. You extrapolate it out, because of course these sort of things continue forever until every grain of sand on the earth is a Gatsby [inaudible 00:41:06].

Anyways, but yeah, it's like we're solving really core problems on the web, like performance, developer experience and kind of making website teams more agile and flexible. Yeah, Gatsby grows to if a significant percentage of the web is using Gatsby and a significant percentage of those sites are using our commercial cloud kind of services, then it's not hard to justify the valuation.

[SPONSOR MESSAGE]

[00:41:48] JM: Feature flagging makes it easy for your team to quickly change the way that your product works, and CloudBees Rollout lets you manage feature flags easily. When you have a solution to manage feature flags at scale, you're empowered to continuously and intelligently rollout changes as soon as they are code complete on any platform, even mobile. You can decouple development from code release for a real-time change control. You can rollback only the changes that you don't want, or keep them around.

You can toggle features. You can use multi-varied flags for A-B testing and you can remove misbehaving features with a kill switch. All of these is part of the feature flag platform that is CloudBeess Rollout.

Visit softwareengineeringdaily.com/cloudbees and try a free 14-day trial and experience how CloudBees Rollout can help you with every release. Visit softwareengineeringdaily.com/cloudbees to get a free trial. Cloudbees Rollout is trusted by large users, such as Zekdesk and Jet.com. Try it out today at softwareengineeringdaily.com/cloudbees.

[INTERVIEW CONTINUED]

[00:43:12] JM: What I wonder is why you would raise so much so fast? Because there are projects that take off that much have trouble capturing enough value to really build a gigantic business around it or at least to build a gigantic business around it in the time frame that is necessitated by the funding cycle that they decide to buy into.

Again, I don't of the market super well, but like I assume you could have raised like 3 million, built out preview, get some revenue traction and then sort of like bide your time and then –

[00:43:55] SB: We just think everyone is clamoring for this. If you really like satisfaction with CMS's, it's plummeted in the last 10 years. It's just us a lot of smartphones. The rise of cloud computing. Yeah, CMS's are just like a relic of a former kind of technological age. People who are forced to build sites with them, which is a lot of people, don't like it.

I'm most familiar with the Drupal community. I used to do lot of stuff with Drupal, and like everyone there is basically like how do I get out, sort of their feeling. Yeah, the Mark Andresen's original product market fit blog post. It's like great products are pulled out of teams by the market, and that's kind of just what we're feeling. It's like everyone is clamoring for us to solve so many different problems and we have kind of the inside track to do that. Gatsby's architecture we think is like really proving itself out as the right way to solve all these kind of really long-standing, really painful problems. To solve them, it's really just a matter of scaling our team up to be able to tackle all of them. So the extra funding is necessary to do that.

[00:45:07] JM: You see the opportunity to build services that are high enough margin, highly differentiated enough to defend yourself from like Netlify from eating the same market, for example.

[00:45:21] SB: I mean, that like we're thinking about this as collaboration tools for markers and content teams. I think a lot of Gatsby sites run on Netlify, and that's great.

[00:45:32] KM: This is going to be a very large market. Again, there're millions of Gatsby sites. It's not going to be hard to have you know 50, 100,000 customers paying so much money or whatever, which is like –

[00:45:45] SB: Anyways, like doing the math to get to a hundred million revenue isn't particularly difficult.

[00:45:49] JM: Really?

[00:45:51] SB: Yeah.

[00:45:50] JM: That's great. Are the people coming to Gatsby from WordPress or are they coming to Gatsby as brand-new developers from all over the place?

[00:46:04] KM: The answer is yes, and that's kind of the exciting thing. Talking back to the earlier question as well, there are so many narratives that Gatsby hooks up into. There are folks in WordPress who are looking who feel like the WordPress frontend is outmoded and are looking for headless WordPress. There are folks in the Drupal community who are looking for like decouple Drupal and they don't like the sort of the new Drupal templating language and are looking for an alternative there.

There are folks in more of kind of the modern frontend development community, folks who are like more experienced with Babel and Webpack and these kinds of tools but are looking for a nice thing to kind of encapsulate it and pull in data from content and like just be like, "I do React. I want to build a website." Looks at WordPress, "Oh!" and then they like find Gatsby and like, "Okay. This is much nicer," and they roll with that.

Then there's like down the list of ecommerce CMSes is like Shopify enterprise focused CMS as like the ones that we're mentioning earlier. The web is such a diverse place and everyone in every like almost every corner of the web is looking to modernize their frontend stack and they're looking to React and then there asking, "Oh, how do I use React?" The answer for almost everyone that is like Gatsby is the solution for you.

[00:47:17] JM: How do you guys meet?

[00:47:19] SB: Kyle moved to Palo Alto. I was finishing up school and this is like 2011.

[00:47:25] KM: Yeah. Thereabouts. Yeah.

[00:47:27] SB: I can't believe it's – It feels such a recent time, but I guess it has been kind of eight years. We got to be really close friends and found ourselves like having lots of really like long, random conversations off the wall on abstract philosophical questions. What is life? Shat is a soap bubble. There we go. Sorry. I mean, like our lives kind of went along. I got married in 2014 and Kyle is my best man. Kyle got married later and his best man.

[00:48:03] KM: It's just when Gatsby was obviously doing well and I we're like real, "Hmm. Maybe –" Because, yeah, we're both very interested in startups. We both worked at startups. Tried our own stuff. We're like Gatsby is a very interesting compelling idea that kind of met both of our kind of personal goals and interests. Yeah, it kind of was a natural fit.

It was kind of interesting, we never like worked work together, but we're like – I don't know. We get along really well as friends. Maybe it will work, hopefully. It's worked out really, really well.

[00:48:35] SB: It worked incredibly well. The most challenging with founding a company with a cofounder in general is like kind of staying on the same mental wavelength and like people who have very different philosophical ideas of how to do things often struggle with like kind of agreeing on like the direction of a company and then like if the company tries to scale they'll be like, "Well, this –" The first cofounder says do this, but the other cofounder says to do that and it can be challenging for everybody.

We'll be like – Because Kyle and I are just been so close friends and like think about the world in very similar ways, we 95% of time will answer the same question the same way, which is just very like hopeful and trying to go build and scale something great.

[00:49:19] JM: You mentioned Airtable once already. How does Gatsby intersect with the low code or no code category of software?

[00:49:30] SB: Well, one is you write a lot less code with Gatsby, because the plugin ecosystem is massive already. We have like almost 1,400 plugins now for Gatsby and it's doubling every like six months or something I think like that at the current rate. Yeah, when the Drupal world, they call their extensions like modules. It's like, "There's a module for that," is sort of the little catchphrase in the Drupal community and like we're kind of getting to that same point, that almost anything you want to do it's like install a plugin and add an API key and then off you go. That saves an enormous amount of time in code. It's like Gatsby allows you to just focus on kind of the core parts of your site, which is the React components, the Vues, and then like how the data gets maintained and how it gets into the site is managed by another service.

I think that's like a huge product, Gatsby's success, is people. Because you can waste hours like reading API documentation and like figuring out how to do stuff and making it performant. You hit like the API rate limiting and you're like, "Oh crap!" and you have to add an extra code for that and on and on and on and on. With Gatsby it's like, "Well, somebody else already debugged that and figured that out," and you just install a plugin and off you go. That's like a huge time-saving and saves a lot of code.

Yeah. I mean, a lot of like the "no code, low code" folks, they're interested in building UIs through kind of visual programming type stuff, and we think there is an enormous amount of value in that as well. We've seen the success of like Wix and Squarespace. Wix went public. Squarespace is preparing to go public. Webflow is of course doing really well. That's like something that we're very interested in as well and we think that there is like a natural complement to Gatsby where you can say, "What if you could take your website and your React components, but build a page builder type experience on top of that for the non-technical folks who are involved in the website?"

We think that'd like a really compelling experience, because where all these like no code things break down is like what do you do when you do need code? What if you want to build a custom widget that isn't provided by the platform? What if you have like this theme system that maintains some other thing and you want to like merge the two together or whatever? You're going to need code at some point. But we think that the right approach is to have a code foundational layer and then build interfaces for non-technical folks on top of that to kind of manipulate the website in a way that's like natural for them.

[00:52:06] JM: What's been the hardest part of building the company so far?

[00:52:09] SB: Building the company. Yeah, I mean building the company is really – It's like finding great people, helping them find a place within the company and helping people communicate and work well together. Yeah, that's really difficult to create. It's a really hard challenge for any company. We've definitely had our struggles on the way. But yean, it's also really rewarding to part of it as well. Because when you do, you help a team like really start thriving, it's like a really unique experience to see them just like completely take over a big chunk of the work and own it and run with it and do things that we hadn't really ever thought about and so forth.

[00:52:47] JM: Can you guys give me some pieces of cutting-edge frontend technology, because I swear, every time I take my eye off the frontend and then pay attention to it again, there's like eight new projects that I like I'm not familiar. Like CSS Fire and I'm like, "I don't know what that is. What does that do?" What's with the frontend these days? Give me some like crazy cutting-edge related React stuff that's in the JAMStack world.

[00:53:21] KM: I would say my top three are – This isn't like new per se, but it's new news related to that, which like the React team for the last for years has been working on kind of making React a lot smarter about how it renders in the frontend like really complicated UIs, because when you have a lot of stuff going on, it's very easy for complex UI to start blocking inputs.

For example, you're like trying to type an input box, in the meantime, data updates are coming from the backend and like triggering re-renders and it can get like glitchy or something pretty easily. So they've developed a really just incredible kind of groundbreaking technology. They call it like concurrency mode or whatever, where nothing blocks each other. It's just sort of like everything is like – There's like prioritization so that the highest priority stuff can interrupt other work, and lower prioritization work will wait for the higher priority work to finish before resuming itself.

It's very much like an operating system, what operating system does, because they have the same sort of problems where it's like you have a CPU and memory, but there's all sorts of things that want that and like you could have some sort of little low-level program. Just grab all of it and like suck everything. Operating systems sort of like say, "Hey, no. We're prioritizing user-related stuff so that we have like – The user has a smooth experience," and then sort of coordinating between all these different stuff going on.

The modern frontend is really turning into the sort of complex, like application world. You have like all these components or almost little mini apps working together. React is turning into like the operating system for all the mini components doing stuff, which is like really, really fascinating to kind of see computer science like repeat itself. It's like Mark Andresen was like back in the 90s like, "Mozila is going to be the new operating system," and in a way, like the browser is sort of like – It's almost like the new hardware, and then like React is becoming like the new operating system sort of thing, which is like fascinating in a lot of ways. That's really interesting. I'm hugely excited about that and like we'll be able to leverage that for a lot of interesting stuff within Gatsby.

Another big thing is just like how we manage a state, because JavaScript applications is kind of client/server architecture, and syncing data back and forth between the two is a complicated problem. For a lot of team, it ends up sucking up a lot of time just trying to like optimize that. There is more and more sophisticated ways that are coming out to do that.

React work, they're working on something like low-level helpers to make that a lot nicer. Facebook's keeps working on like relay, which is their client library for GraphQL. Apollo is doing a lot of interesting work in kind of similar vein around GraphQL. So that's super fascinating.

Something we're investing really heavily in which we think has enormous potential is just the idea of like how do you theme your frontend and like kind of create design systems with constraints so that all of your UI ends up being predictably good, I guess, because a common problem is that designs kind of the diverge from each other in weird ways, because like frontend developers are like, "Hey, build this," and it's not really specked out too closely and they just like start building away. Then it looks strange compared to other parts of it, because they chose slightly different spacing values. It's like, "Well, this one is –" They did like eight pixels, and so the 10 pixels [inaudible 00:57:00] and you're like clicking around and you're like it just has this like weird inconsistency feeling to it.

This whole idea of like constraint-based design systems prevent that, because they kind of abstract a little bit away from like the raw pixel values and like give you a spacing of one or spacing of two or spacing of three, that sort of thing. A lot of systems are approaching this in different ways and a couple of members of our team, Jono Tender and Brent Jackson, headed by Brent Jackson anyways, is building something called Theme UI, which kind of bakes all this in into like a really kind of comprehensive package. Anyways, a lot of people are excited about. I'm super excited about that. It's kind of like a way to solve styling in a really comprehensive way for kind of websites and applications.

This is also something like very applicable to Gatsby too, because like one of the promises of Gatsby to large organizations is building lots of sites with Gatsby and then maintaining consistency across them, and something like Theme UI is really essential, because like what if you're IBM and you have very strict brand guidelines and whatever. Every team –

[00:58:17] JM: That's a huge problem.

[00:58:18] KM: Yeah, exactly. Every team is like trying to do that and like sometimes they succeed well and sometimes they don't and it's a very frustrating thing for a lot of companies to sort of like keep everyone on the same page. One example, you change from one flavor of blue to a slightly different flavor of blue. How do you update every website?

What these kind of constraint-based design systems allow you to do is kind of specify all the stuff in like one centralized team kind of maintains it all, but it's like constraint. So it's like it gives you flexibility like within certain constraints and how you build it. So it's like we're doing something custom. You're not like, "Well, too bad. You can't do that." It's like, "Okay, you still have the flexibility to do things." But it's constrained enough that it's like recognizably IBM. Still a bad point. Yeah, this is like a huge long-standing problem, and we're pretty hopeful that we're on to something that's like actually cracks in a way that none of other solution really has.

[00:59:13] SB: Generally, another kind of related way of thinking about these questions is that we really view HTML and CSS and JavaScript as in many ways becoming more of compiled targets than things to kind of write code in. This kind of follows the pattern of programming languages overtime. It's becoming more and more high-level.

[00:59:38] KM: We started with assembly, which is like very raw and then we're like, "Wait a second. We can build programming languages, which have infinite number of abstractions," and then just translate that down into the raw thing.

Once you have like compiler, you have the flexibility to create more like abstractions that are more aligned with what we're actually trying to do. Instead of being like you must write the same HTML or CSS value across everything around [inaudible 01:00:09]. It's just like, "Well, I don't know. Just say IBM blue, and then it works," which is a much more felicitous way to do things.

[01:00:17] JM: What you mentioned about React becoming the operating system, I think there's a lot of truth to that. You look at Apple – I mean, we're recording this on a MacBook right now, but Apple's kind of like the WordPress of the tech industry. It's like, "Ouch!" I mean, we're all using this big monolithic operating system and we just have a bunch of Chrome tabs open on it.

I've got my Chromebook and I'm like, "This keyboard is not great. It's like not as good, but to me it seems easier for Chrome to build hardware than for Apple to build like AI good cloud services. But at the same time, React is really building the best and the most modular frontend technology. Google doesn't really seem – I mean, they've got the Android and Chrome thing. Maybe that gets unified overtime or something. It'll be interesting to see how that operating

system war shakes out between Facebook and Google, because that it seems like that's the next –

[01:01:31] SB: Like React versus Angular?

[01:01:32] JM: It's not exactly React versus Angular. It's just Google versus Facebook. I mean, it's React versus all of Google's technology.

[01:01:40] SB: Yeah. I think, in general, historically we moved between more modular technologies and more integrated technologies during the 80s and 90s. We sort of transitioned from like these monolithic mainframes and minicomputers to Windows kind of driven world with like these like very defined interfaces for like, "Hey, I need a graphics card here, or I need like a motherboard there," and like computers are kind of assembled from that. Obviously, Apples kind of a shift back towards them, more integrated world.

I think that place out in a number of different areas and aspects. I think with React and the frontend and the kind of the browser world, there definitely is that shift back towards kind of modularity and it's kind of very kind of exciting. We kind of think that's a good thing.

[01:02:22] JM: Certainly for you, and I would say for the public in general. I mean, I feel that way. Guys, thanks for coming back on the show. It's been really great talking. Well, Sam, you're a first time. We didn't talk very much. You'll have to come back on at some point. We'll have to do a one-on-one show. You deferred too much to your cofounder, which is a good sign.

[01:02:39] SB: We're on the same page a lot of the time. It's great being here. Jeff, thanks for having us and we really enjoyed it.

[01:02:45] JM: All right, great.

[01:02:45] KM: Yeah, it's really fun.

[END OF INTERVIEW]

[01:02:56] JM: As a programmer, you think an object. With MongoDB, so does your database. MongoDB is the most popular document-based database built for modern application developers and the cloud area. Millions of developers use MongoDB to power the world's most innovative products and services, from crypto currency, to online gaming, IoT and more. Try Mongo DB today with Atlas, the global cloud database service that runs on AWS, Azure and Google Cloud. Configure, deploy and connect to your database in just a few minutes. Check it out at mongodb.com/atlas. That's mongodb.com/atlas.

Thank you to MongoDB for being a sponsor of Software Engineering Daily.

[END]