

This transcript was exported on Aug 04, 2023 - view latest version [here](#).

Chris Strahl:

Hi, and welcome to the Design Systems Podcast. This podcast is about the place where design and development overlap. We talk with experts to get their point of view about trends in design, code, and how it relates to the world around us. As always, this podcast is brought to you by Knapsack. Check us out at [knapsack.cloud](https://knapsack.cloud). If you want to get in touch with the show, ask some questions, or generally tell us what you think, go ahead and tweet us, @TheDSPod. We'd love to hear from you.

Hey, everybody. Welcome to the Design Systems Podcast. I'm your host, Chris Strahl. Today, I'm here with Sriram. Sriram's over at Meta. Welcome to the program.

Sriram:

Thanks, Chris, for inviting me do this Design Systems Podcast.

Chris Strahl:

I'm really excited to have you because one of the things that caught my eye about our early back and forth was you're an engineer but you work with a lot of design every single day. You spend time in Figma making plugins, working with designers to try to really focus on this design-to-code process. Can you talk to me a little bit about that and what you do?

Sriram:

Yeah. I support a team called Design Systems Engineering. This consists of two design systems team, a team focused on design tooling which is the one that you are asking about in terms of Figma plugins and whatnot. And then also a platform which is trying to solve this interoperability problem across all design systems we have at Meta. Specifically about design tools, the main problem that we are trying to solve is how do we provide a platform for core system components that our team should use through better discoverability? And how do we solve the design engineering handoff problem? Because designers can spec out all these things, and when it reaches engineers, they need to know what is the right company to use.

Chris Strahl:

Gotcha. And then tell me a little bit more about the interoperability problem because I'm not as familiar with that side.

Sriram:

The interoperability problem is specific to Meta because of its scale. You could think of one design system per app, so Facebook, Messenger, Instagram, WhatsApp, and that's already four design systems. And we have a lot of different UI frameworks that we use in each of these apps. And when you look at the scalability problem here, we have a lot of different component libraries which are trying to do similar stuff. The button is a button is a button across all different design systems. The company mechanics are the same. What differs is the brand strategy with each of these apps. We are trying to split these two in terms of have the design decisions be represented as design tokens and the company mechanics to be agnostic of what app they are being used in.

Chris Strahl:

This transcript was exported on Aug 04, 2023 - view latest version [here](#).

I like this idea of this architecture where all of your components and all of your UI is essentially the same, and it's the brand application that varies per product implementation. Or it's not even really the product. It's the whole product group, Instagram or Facebook or whatever, Threads now. So with all that and this idea that you have common UI with dissimilar brand application all managed by tokens, that's a pretty daunting undertaking to take on at Meta scale. How do you conceptualize that work within that team? What does that work look like when you think about the strategy for planning out that system of systems approach?

Sriram:

Yeah, this is a complex piece of work because a lot of these design systems have been in existence for a lot of years, which means each one have their own copy of component libraries across multiple UI frameworks. So the way we are thinking about this problem is splitting it into three different layers. We need to have a platform-agnostic framework, agnostic source of truth. This is where we store our design tokens. And from there, we want to use consumable pieces or artifacts of each of our UI framework and also design tools. This is basically taking this platform-agnostic, framework-agnostic design tokens and representing the language of choice and then using them in existing component libraries. When we do this, we provide a way for graceful upgrade to using a common platform. And in the future, we could start consolidating these libraries because at the end of the day, the brand is supplied from outside and the core component mechanics are mostly going to look the same. Then it is an engineering consolidation problem.

Chris Strahl:

Yeah. Do you think about the brand as just like a JSON file that is the representation of that set of tokens, or do you think about it as something that's more akin to the Figma modes concept where you have a different application based on some context in the actual application of that brand?

Sriram:

We've been thinking about it more in terms of JSON files for now. Each of these applications will have, it's one set of JSON files or themes that we can use. Each of these applications will define these tokens of our foundational pieces like colors, type, pick ramp, color radii and all those things.

Chris Strahl:

I think that that's where most people landed on this right now because one of the really hard problems with the token space is the ability to influence tokens at runtime. But then as a second aspect to it, tokens aren't the same on the interoperability front. Tokens for web, tokens for native apps, tokens for living room devices, they're all fairly different in their output. And I think from the engineering perspective, trying to think about how that is, it's not just a CSS file and CSS vars. It's not just translators. There's a lot of complexity to making that work across a lot of different frameworks.

Sriram:

Yeah. You're making a really good point here. That's why we want to think about tokens in a platform-agnostic way to start with and pick things specific to mobile and have it represented as design tokens. And you can have a different set of tokens or other media. And then what sort of translators do we need to represent them in your framework or operating system of choice?

Chris Strahl:

How do you think about that in the sense of cross-discipline? So you have design tokens that are presumably originating in Figma or at least have to synchronize back to Figma once they're in a code medium. What does that specific interaction between design and engineering look like when it comes to a brand? And by the way, I love that you think about tokens as a brand, by the way. I think about it that same way. And everybody talks about tokens as like, "Oh yeah, it's just like Chrome or whatever or a theme." No, it's actually a brand expression. I love that.

Sriram:

I agree with that. It is the right way to think about how do we bring brand strategy to our code. To your original question of how do we bring designers and engineers aligned on these tokens, this is a more of a process problem in making them understand what are they trying to convey, what are they trying to represent from the design side, and how do we represent them on the engineering side?

So let's take JSON files for example, and the source of truth is going to be maintained by an engineer in collaboration with a designer. And this collaboration piece has to happen. It cannot be where a designer specs out everything in a Figma file and hits a magic button and then it automatically syncs, and then the engineer looks at this file and be like, "Nope, this cannot be translated back to code and this is not the format that I would want." So in our culture, we have quite a lot of meetings around this where we sit together, decide what are the core semantic tokens that we want, what are the component level tokens that we want, and how do we represent them, what capabilities do we need for different modes and whatnot.

Chris Strahl:

Yeah, I think that there's a lot of thinking in the industry that is like we make stuff in Figma and then it's engineer's responsibility to make that work for users. And I think that the hard part about that is it disconnects the fundamental design processes. There's a lot of design that actually happens in the engineering space, and that design that happens in the engineering space is transitions. It's states. It's all these other different things that are very difficult to represent in an abstraction and a design tool.

And ultimately the user experience is defined by what comes out of a JSON file at build time. And I think that it's, I don't know, shortsighted might not be the right word. It's myopic to think that the ability to say, "I can click a button, export stuff from Figma, and that's ultimately what flows into production," that's not actually really true. And I like what you're talking about in terms of solving that as a human problem. But presumably you do have tooling in technology that is helping bridge that gap. What does that process look like? Is it like okay, I designed or I made something in Figma. Is there a call that happens? Do you click a button? What does that look like that ultimately gets that then into JSON?

Sriram:

Right. The JSON files are separate from Figma. Our source of truth is on the JSON side and not starting from Figma, which means Figma needs to consume this source of truth to create its own set of variables, its own set of variables that can be applied to components and whatnot. We have built our own set of plugins to input these JSON files to make things available for us. We've been building plugins for Figma for quite some time, and this is one of those plugins where we'll input these and use them as variables within our Figma plugins.

This transcript was exported on Aug 04, 2023 - view latest version [here](#).

Chris Strahl:

That's awesome. And so does that go natively into variables in Figma, or is that using some sort of tokens plugin as well?

Sriram:

Since the variables are new, I'm little to not too tight to that work in terms of how that translation is happening.

Chris Strahl:

I mean super new, less than a month old as of this recording, so-

Sriram:

Yep.

Chris Strahl:

All right. Well, so that's interesting because I think that most people think about it the other way. They're like, "Hey, my tokens start in Figma and then I get them into code." And I happen to agree with you. I think that maybe initially at the very first time, they start in Figma and then they get into code. But then from there on, it's something that really is owned by the end user experience, not by the design side of things. And obviously it should be a collaboration. And I think that the emphasis of, look, there needs to be a human process, a human collaboration that is fundamentally multidisciplinary there is an important point to make.

Sriram:

Yeah. I'll try to shift the topic a bit into components and why we chose having the source of truth in JSON instead of having in Figma. Even before design tokens became a thing, when we are working on our component libraries, the critical problem that we had was designers would change things in Figma and then all our libraries will go out of sync. So we wanted to have some sort of contractual agreement of sorts to say that this particular component has these variants and these features and this is how it should be represented.

So think of it as more in terms of a spec document written as a text file that designers and engineers can agree on. To your point about a multidisciplinary collaboration, happens outside of your own area of expertise. And this ties them to say that, "Okay, this is where we are going to define what this thing is. And using this spec as the commonplace, now designers, go and update your Figma to reflect the new variant that you want. Engineers, go and update the components that you want." This is the same approach that we wanted to take with tokens where we don't want one unidirectional flow. We want people to come closer together and then work on this particular concept in a centralized place.

Chris Strahl:

Forgive my digression in tokens. It's been a hot topic lately, but I agree with you. And I think that it is interesting that you've made the spec the central collaboration point between design and engineering. And it's funny. Knapsack's actually built that way. We started when we were first envisioning the product thinking about using JSON schema and having that schema definition be really the thing that represents truth. And then the implementation of that truth in code or in design can be whatever that discipline

wants it to be. But fundamentally, that schema is what represents all of the expressions of that component. And that's basically a bunch of properties. It can be slots, it can be applications of we call them asset sets. It'd be like a mode and Figma, and all of that is the fundamental rallying point because that represents similar to a data contract with an API, a contract between how design and engineering should view the problem they're solving.

Sriram:

Yep. That's the whole point of taking things outside of their comfort zone and putting it in a third-party tool, in this case JSON schema, and have them debate about what they wanted to change and work together on that place.

Chris Strahl:

Yeah. It's hard because a lot of times when we talk about that solution, you get a bit of pushback from the design side about, "Well, is what I'm designing just bullet list? Is it just me telling you what props to use in a bullet list of components?" And my answer to that is always like, "Well, that's how the web works." Native apps in a similar sense work in a similar way. But if it's components all the way down, it is being like, "Here's a bullet list with a bunch of props or attributes that I'm specifying values for." And that's really the fundamental idea of what you're building. Do you ever get that sense from design, that it's somehow taking something away from that creative experience?

Sriram:

This is one of my favorite questions because the more and more I look at Figma, it is getting closer to how we build things on the engineering side, introduction of things like outer layout, introduction of things like variables and defining a component, defining a component with props. And we are bringing all these engineering concepts. And this is a new space for designers and they feel like, "Okay, I've been working on this in a completely different way and now I've been asked to change the way of how I think about components, how I think about product design in a way that engineer would think." So there is a big shift happening in Figma that I'm starting to recognize within my company.

Chris Strahl:

Yeah, it's interesting because I thought, I still think, for a while now that there's been two camps. There's the one camp that is like we should be able to draw boxes on a canvas and then something should make that into code for us, some automated process. So Figma to React is like there's a bunch of tools that sprung up about that. I think Figma even acquired a company that did something like that. And then there's the other side of it which is we should be doing more code-like processes in our design. And I am curious your thoughts on where this is headed because I think that design-to-code in an automated process is not likely because there's an abstraction and there's actually the CTO of, "Hey, 37 signals." Wrote an interesting blog about this the other day about how fundamentally every design tool is an abstraction of the medium that something is destined for.

And so there are inherent limitations in that abstraction that don't exist in the final medium, like states and transitions and stuff like that. And ultimately that makes it very, very difficult to automate the creation of code from a design file. But the idea that on the code side we're able to make that code flexible and easy to work with and easy to understand for people that don't actually write code, that's exciting to me because now all of a sudden you can basically say I'm actually designing in the medium it's destined for. What does that look like in your mind? Does that look like, "Hey, Figma just becomes

another way of coding"? Does it look like, "Hey, Figma ends up getting abandoned for these other tools that are able to make code easier for everybody else to understand and democratize that process"? Where in your mind is this head?

Sriram:

If you look at what we are trying to solve for in trying to do this design-to-code, we want to ship our products faster. We want to have a high fidelity of design to code and get these products out to our users faster. And if that is the problem that we want to solve, it is not how do we generate code from design files. It is about how do we have engineers looking to the right components that match what design has spec out in your product files.

So what we want to solve for is if you have, let's say, a React library and you need to have real connection points to Figma design files. I'm really excited about what the dev world is trying to do in Figma where you can connect these dots together by saying, "Here is a component that a system designer has used in this product-specific file. And if you are an engineer who wants to implement this, this is the standard component that you need to use."

Because we've tried some of these things like can you do design-to-code? The major problem comes down to performance. You could abstract things to an extent like okay, this is a rectangle with rounded corners and you can generate code. And it gets a lot more complex when you are trying to place things or optimize things. And I'm not saying the designers cannot do this in terms of finding the optimized layout that they want to use for composing all these components, but it requires UI framework knowledge. It requires OS-level knowledge to optimize things on the company library side or the product side. So for me, the main connection point is if a designer says, "Use a button," I want to know what button class I need to be using in code.

Chris Strahl:

Gotcha. Yeah, it's interesting you brought up performance because that's the sticky point that we always see as well where we've had tools like this for a while. There's Dreamweaver. We all use that. There was the, what, front page was Microsoft Solution for a while or something like that, right?

Sriram:

Yeah.

Chris Strahl:

And so all those were awful because what would happen is it would generate a website that would take you 12 minutes to load, and that was a terrible experience. And that is part of the design is how fast the page loads. So I think that the interesting thing there is there's also a lot of accessibility concerns. There's lots of things that exist in actual implementations at scale where milliseconds matter that you don't ever think about in things that don't represent scale. It's very easy to look at a comp and say, "That looks beautiful." But does that perform with, in your case, a billion people looking at it every single day?

Sriram:

Yep, that's right. So the thing that we are going for within Meta is you get down to the core set of components that we want to build the app with and design the app with. So the system designers are thinking about what are these components that would make up the core app? And the engineers on my

This transcript was exported on Aug 04, 2023 - view latest version [here](#).

team are looking at how do we build high-quality components with optimized performance that all of the product teams within Meta should be using? And how do we provide this connection so that a product engineer, when this person has given a design file, is this person able to pick up components from our code library and start building their products?

Chris Strahl:

That's great. How is it working? Is it the frictionless, wonderful utopia we all thought it would be?

Sriram:

I wish it was like that, but it is not. The challenges are in terms of discoverability. With new people coming into teams, with new people joining the company, it is always the onboarding experience that has been a problem for us. And I'm really excited about the dev mode because there is a clear way to say when a Figma file is handed over from design to engineering, and that is the right place for us to plug in and say, "Hey, this is the design that you are trying to build and here's the set of Lego pieces or components that you need to be using in your product."

Chris Strahl:

Gotcha. I think that it is interesting. One of the parts of this conversation that has sparked some fun thoughts with me is this idea that the spec is central and the implementation is up to whatever discipline is building the thing. And so we always have the conversation about do the products you create look the same in design as they look in user land in implementation? And very often, that's a big thing that we're trying to solve, is we're trying to make it so that thing in Figma looks a heck of a lot more like what a user actually sees. But how specific and how detailed you get into that is maybe irrelevant as long as everybody's rallied around that same idea of what a brand application is and the spec for a component.

Sriram:

Yep, that's right. So we are talking about two specific layers here. How do we apply the brand, which is coming from the design tokens. And if we are abstracting it away from designers and engineers and make them use from a common sense of truth, then the design tool of choice and the UI frameworks are consistent and they all stay up-to-date. Similarly, if you have a common spec for each component and something that designers and engineers agree upon, then any updates that we want to make is going to go through this update process which means all the components across all UI frameworks and the design tool will stay in sync. When you have the brand strategy and the components stay in sync, then the question or the thing where there is loss in fidelity is going to be in how these things are composed and how it is put together when it reaches the users.

Chris Strahl:

And so how do you think about that composition? Does that composition still largely happen in design or do you compose a lot directly in code? And I guess that's a weird term. Does it happen in Figma as a tool or does it happen in the actual implementation more?

Sriram:

We've looked at product development lifecycle. And in my opinion, the right place to look for whether the composition is right is looking at what the end solution that a designer has built and having a

This transcript was exported on Aug 04, 2023 - view latest version [here](#).

designer come in and say, "This thing looks right," or, "This thing doesn't look right and you need to fix these things." A couple of years ago, we ran an initiative to bring designers into reviewing the final output of a particular product implementation at diff time. So when engineers put a code for review, we would also have them attach mocks and videos and whatnot for designers to come and review this to see if it stays true to the design that the designer came up with. And that is the place for us to tweak these things. There isn't a magic way to compare shares, the design file, there's implementation in some automated way.

Chris Strahl:

Gotcha. And so the limitations that you're running into now are less about limitations about the harmonization of design and code and a lot more around the human processes of at what point do designers or engineers review or look at something? How do people discover the things, discover the right tools and the right pieces, the right Lego bricks to work with? And so that discovery and then review process is now the focal point of where innovation's going to happen inside of Meta.

Sriram:

Yeah, the main thing we are focusing on is our designers and engineers speaking the same language, and that is a human problem. And if you try to put more tools and try to make it automated, it hasn't worked. We've tried it. It never worked.

Chris Strahl:

Yeah. Is it a CTA or is it a button?

Sriram:

Yeah.

Chris Strahl:

Or is it a link? I don't know. Nobody's called it a link in a while, but the idea of how you harmonize names is the eternal problem in everything about the web is no matter what, we all have to figure out. Not only are we using the same words but do we actually see the same thing in our mind when we use that word?

Sriram:

This is where the connection between design and engineering comes in. I'll explain what happens with the spec and how we try to solve this problem. Spec is this contractual agreement that we talked about, and we create these components in Figma. We create the same components in UI frameworks as well. We have our own internal way to associate a Figma component to the actual component used in code. So when an engineer is looking for the right component to use, we also build internal tools around documentation where design and engineering code documentation coexist. So if you go and look for a button from a design side and you find the right button to use, then it is also linked to the framework component that needs to be used. You could call it as a link, you could call it as a CTA, you could call it as an action burden. That doesn't matter. But the engineers know that this is the actual class file. This is the actual component that we need to use in this particular framework. That design engineering connection is happening also in our source of truth.



This transcript was exported on Aug 04, 2023 - view latest version [here](#).

Chris Strahl:

Gotcha. So that documentation is where you do that meeting of the minds and everybody's looking at the same pattern even if it's called a different thing, even if the implementations are slightly different. Again, the core thing is that schema or that spec represents what that function is. It's really interesting. I like that point of view. When it comes to that discovery problem, how are you guys focused on solving the discovery issue?

Sriram:

A few years ago, we built this documentation tool that I was just talking about where we... To give a little bit more context, design had their one set of discoverability issues, and we built tools within the design tool of choice. Back then it was sketch where we would show all the components that a designer should use and relevant documentation around it. And then engineers had access to a different set of documentation around, think of it as like code browsers with API and whatnot with no indication of how a particular component would look like. We built a tool that brings design and engineering documentation together. The closest example I can point to what people can look at today is material design documentation. If you go to material design documentation, there is a design tab which talks about how this component should behave, how this component should be used, or what context you need to use it for versus not. And then there is an implementation tab where it says this is the actual component in material design library and this is the API for this particular component in this particular framework.

Chris Strahl:

Sriram, it's been so great to have you on the show. I think that in particular, your comment about the spec being the focal point of that connection to design and code, I don't think a lot of people think that way. But I think a lot of people should because what you're able to do is you're able to take that thing that is, like what you said, a contract, and you're able to manage it and put workflow behind it. And then that becomes the guiding connective tissue between these different disciplines. And that's super cool.

Sriram:

Yep. We went through a lot of iterations to get there, but we are happy that we are using it in our day-to-day process here at Meta.

Chris Strahl:

I so appreciate your time. Thank you for sharing and being so open about the work you've done. Just really appreciate you. Thank you so much for being on.

Sriram:

Thanks, Chris, for having me here in the Design System. It's so close to my heart so I was happy to come and share my thoughts about the design systems and what we do at Meta here.

Chris Strahl:

Awesome. Well, hey, this has been the Design System Podcast. Hope everybody has a great week. Take care, everyone.

This transcript was exported on Aug 04, 2023 - view latest version [here](#).

That's all for today. This has been another episode of the Design Systems Podcast. Thanks for listening. If you have any questions or a topic you'd like to know more about, find us on Twitter, @TheDSPod. We'd love to hear from you show ideas, recommendations, questions, or comments. As always, this pod is brought to you by Knapsack. You can check us out at [knapsack.cloud](https://knapsack.cloud). Have a great day.