

This transcript was exported on Jul 11, 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. 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, this is Chris Strahl with The Design Systems Podcast. Today, I'm with Adam Sedwick, the senior product manager for Design Systems over at Discovery Education. Welcome to the program.

Adam Sedwick:

Yeah, thanks for having me on, Chris.

Chris Strahl:

Hey, so, this is actually going to be a really fun talk. Our premise to kind of get through this was really interesting and we had a great discussion that carried on well over our normal time. And so, what I'm really excited about is two things. First, I caught COVID again, which is horrible. And I was just in Mexico for a week. So, this is my re-entry back into the workforce on a weird day because it's July 3rd, which is this weird Monday that we don't have off despite a holiday tomorrow and just coming off a weekend.

So anyway, this is my soft landing back into the real world. So I'm really appreciative that you're here for this.

Adam Sedwick:

Yeah. And we're trying our best to make it cushy.

Chris Strahl:

So, give us a little bit of info about your background while we get started here.

Adam Sedwick:

Yeah, so I started my first job in web dev back in 2012 right when responsive design was really taking off. And so, my first sort of big foray into everything was already starting with the idea of, hey, maybe we can build this thing once and have it in multiple places. Because at the time we were still building mdot sites and desktop sites.

Chris Strahl:

God, I've forgotten about those until just this moment. Yeah, that's a dark chapter.

Adam Sedwick:

I did that for a couple years and then I wound up somewhere where we were at an agency and we were thinking, "We have all these different client websites. How can we build these bespoke websites faster?" We were building them on WordPress at the time, so we started looking at, "Hey, maybe we can build a theme and then skin it for different clients." So again, same sort of thing. Let's just build this thing once and maybe we can start reusing it elsewhere.

I landed at DE back in 2016. And they were, at the time, building their very first design system. They had contracted out with EightShapes, and I didn't really know what design systems were. But thankfully, I got to work with Nathan Curtis, which was super incredible and super great to learn from.

Chris Strahl:

Yeah, Nathan's obviously a luminary in the space. I mean, the posts, the blogs, the crazy self-defined UML language for design systems, the brilliance there is awesome. I've only actually got to meet him, I think, twice clarity, a great person to learn from.

Adam Sedwick:

Yeah. So then, like I said, we worked with EightShapes for a couple years, really got to learn what design systems were. Really started thinking about even at scale, even the smallest problem can have massive ramifications. The obvious example that everyone knows is the idea of a simple button. There's all those decisions, space, border radius, what font type is this and how that affects all of the different teams.

About two years after starting at DE, EightShapes moved on to their next clients. And I just volunteered to take over the team, started running the meetings, and then it became, "Oh, somebody needs to go talk to all the teams." So I started talking to teams, gathering requirements on stuff, and it just naturally fell into the, "Oh, well you're doing more product management stuff now than necessarily designer development." So that's kind of where I got where I am.

Chris Strahl:

That's awesome. So was that a hand raise? Were you just sitting in a meeting room in a conference room and people were like, "Well, who's going to own this?" And you just raise your hand up? What did that look like?

Adam Sedwick:

Yeah, kind of. I mean, we knew that they were going to leave. Obviously, people don't just abandon contracts, so it was like, "Hey, we're going to be done with EightShapes in two or three months. We kind of need somebody to do this." And again, at first it was just like, "Yeah, cool. I can run Scrum. That's not a big deal, whatever." And then just as the other things started coming up, I was already the one doing all this other stuff. And I have amazing devs on my team who I didn't want to take off of the dev work that they're doing that's super important.

And so it was kind of just a hand raise, "I'm willing to do it, I'll do it."

Chris Strahl:

So that was a launchpad for you in a whole bunch of ways because now you're also involved in the W3C working group. You have a bunch of time spent in the design systems community. So, that was kind of this, I mean, oddly pivotal moment for you where you were basically, "I'm going to go take over Scrum," and then that's led you to essentially a career defined now by design systems.

And by the way, I love the evolution because it's really similar to mine. I was building Drupal Websites for a really long time. And I was very deep into the Drupal community and thinking about large scale theme mobile projects. We used to work with big enterprises to basically say, "All right, how do you launch a thousand sites on a Drupal platform?" And so I have a very similar sort of background in how you think

about reuse and how you think about using something to power a whole bunch of different web applications, not just one web application

Adam Sedwick:

Yeah. For sure. And I think it really is kind of the same story for everything, right, like with the W3C, the tokens working group, they put out a call for volunteers. And it was like, "Sure, I do design systems, I'm interested in design tokens. I'll raise my hand. I'll see what I can contribute to this." And then even with getting on this podcast, you guys had put a tweet out of, "Hey, what are people working on? Share your stories." And it was another one of those moments of like, "Sure, I'll raise my hand, I have no idea, but it's valuable to share this stuff with people." And it seems to be working for me, so I think I'm just going to keep raising my hand.

Chris Strahl:

That's funny. That's a great way to think about a professional career is I'm just a hand raiser. I think that that's cool. I also tend to be somebody that can't help myself. So, I appreciate very much the initiative that you've shown. Moving towards the idea of what you actually do now in the management of that design system. So you have a design system that's relatively mature, you raised your hand, you got involved in all these groups, you got involved in systems thinking more broadly. Now, what are you doing in terms of managing the design system for Discovery Education?

Adam Sedwick:

Yeah. So, for us, the big sort of challenge appeared maybe a year, a year and a half ago. And it was the fact that we have this mature design system. We have this brand identity. But now, we're starting to have a couple different products or some new products and they maybe want to have a different brand identity. So, our big challenge right now is how do you deal with multiple brands at scale?

And so for us, what that looks like and what that looks like for me is sitting down with these individual product teams and saying, "Hey, what goals are you guys trying to achieve and what can we do from a system standpoint to help you achieve those goals?" Obviously, building components is super useful. But sometimes they need to build their own thing. And when a brand identity is so vastly different, we sort of step back from the building components piece and look at how do we help you bring your brand to the product?

So for the last year, we focused in really hard on, we're going to take all of our components and look at them from the ground up as how do we make these theme mobile? How do we apply different brands to this?

Chris Strahl:

Yeah, it's interesting. I think that the wedding of design systems to brand is a really powerful conceptual idea for a lot of enterprises. Because when you think about brands from the business perspective, there's the idea of brand, voice and tone, visual guidelines, all these things that you usually work with some big agency for or you hire somebody that's an expert in brand. And most of the time, the deliverable that you get from something like that is a document or maybe it's a micro website.

And that whole thing is about how does this brand expresses itself? And the idea of that brand is well beyond digital. It's usually all sorts of things. It could be television. It could be radio. It could be print. It could be all number of different channels that are sort of represented in that brand. But at least in terms

of that digital channel, there is this really interesting way of being able to express that brand in that similar way to how an agency would do it but then, to also use that directly in a product.

And that's really kind of the theme side of it, is this idea that you can stand up a brand inside of design system, be that a set of tokens or set of tokens and components or whatever that takes in terms of form. But then actually ship that. And that's something that's really not been very possible before these systems concepts have come into play.

Adam Sedwick:

Yeah, absolutely there. Again, the idea for us was that even at the most basic like what is your primary brand color? Cool. Well, we touched on buttons a little bit earlier, but hey, if it's something as simple as, "Our brand button is blue and your brand button is orange, well, I want to give you a way to express that through that component. And I don't want you to have to write a bunch of custom code or anything."

So what we looked at was how do we identify this sort of brand identity at a super high top level that we can ship? So, we created different token files and so those get exported. We're using CSS variables at this point. But they get spit out onto the page and then depending on what page you're on, if you say brand color is orange and your button is defined by brand color, that's just going to change. And so, it's almost you don't have to think about it.

Chris Strahl:

And that's this really interesting promise because now you're also not just able to do that with one brand, you're able to theme that across multiple brands. And those brand expressions work with the same exact componentry underneath. And this is one of the things that is sort of an unsung magic of design systems is lots of people think about it in terms of the application to their existing house of brands or to sub-brands within a main parent brand.

But you also got to think about it in terms of acquiring new brands. The moment that you bring on some new brand that has a different idea of how it represents itself visually or expresses itself in a digital medium, the ability to take that, build it into the design system and make that componentry available to that brand is so powerful. It's just massive accelerator for integration into an ecosystem.

Adam Sedwick:

Yeah. And again, I think the idea of even whether it's acquiring a new brand or sort of just realizing that you have another existing brand. If we have a science product and we have a math product, for example, in the US, at least, science is green and math is purple. So, if we built one product and it was one color, well, maybe that doesn't service both brands equally. We can say, "Oh no, science is green, math is purple."

And just by making that one definition, these two products can have their own entire identities now. And that's the sort of stuff that, right, you don't think about until you have the ability to do this.

Chris Strahl:

So when you think about the implementation of that inside of URL systems, you talked about CSS variables and a couple of other fairly tactical ways of implementing it. How do you logically define those brands? Is that based on a theme? Is it based on a set of tokens? How do you think about that definition of where one stops and the other begins?

Adam Sedwick:

Yeah, so we call them themes. And it's all very much based on the idea of systems of systems. So, the design system is this large umbrella that has all this other stuff underneath it. And we sort of created a tiered approach. So, we have the global discovery education brand and theme and everything gets that. And then, if you're within a specific product, you could have a theme for that product. And that is effectively just more tokens that are accessible to you on top of the global tokens that we have everywhere.

And then, the final bit of that is if within a brand you needed a fancy or special call out of a component that this is a one-off, we did go the extra mile and say, "Okay, well, we have component level tokens too." So, if this one thing needs to be special or different, you have that ability.

Chris Strahl:

So, these are a little bit controversial. I'd love to get a little deeper into that explanation. So when you think about it logically, you have discovery education, which has some master global set of tokens. And then you have math and science and one's purple and one's green, and that's over simplification. But there's a set of tokens for one and a set of tokens for other based on that context. What does it then mean if you've got a component level token?

Adam Sedwick:

On the global level, it would be like space. Space is a global token. And on a brand level, it could be a brand color. And then on a component level, that is the very hyper-specific token. So for the button component for example, again, our component tokens are like button padding or button border radius. And by default, those inherit from a higher level.

So if you don't do anything, they basically don't exist. But if I'm on the math product and I need this button to be extra large or extra punchy, I can dive in and say, "Button background pink". And then that override only exists on that single component.

Chris Strahl:

So I think this is really interesting because this sort of fallback system of overrides that is here, when I think about overrides on a design system, I'm used to basically people just saying, "I'm not going to follow the design system and I'm going to do my own thing." And so, this is where I think a lot of the controversy is. Does it make more sense to provide these opportunities for overrides at a very, very low sort of last mile bit of the implementation? Or does it make more sense for people to say, "In this case, this doesn't fit in the design system."

Adam Sedwick:

I think that's a challenge we face every day. There are certainly times where we've had conversations or we've been in meetings and it's like, "Are we giving up too much control? Is this not even a design system at this point?" But the thing that we sort of landed on was you should almost never need these. But in the case where you do, we as our team in our organization, we decided we would rather give you the ability to change this and still use the component than just create your own thing.

And that's something we deal with every day. There are still teams and brands we meet with where we're like, "Hey, this is a little too custom, or this is a little too unique to you. Go ahead and build your

own thing that makes the most sense." But for most cases, if you just need to change the color, you just need to do a little something here, we'd rather you use the component and we give you those options.

Chris Strahl:

No, it makes sense. I'm also curious how you think about the complexity of that. Because you're going from a system where you effectively have two levels, you have a global level and then a brand level. I guess, that's probably the best way to represent it. And then, now, what you're doing is you're saying. "I have another hierarchical level that is that last mile implementation, that component level."

When you think about the complexity that that adds to your system, how do you push back against that innate problem with complexity that users face? What does that look like? Is that training? Is that documentation? Is that just an understanding of a way of working inside of the system? How do you guys show people how these work in the right context?

Adam Sedwick:

Yeah, I think it's a little bit of all of the above. I think everyone agrees reading documentation sucks. But it does have to exist. We have to be able to tell you how to do this. So, we do definitely write and provide that documentation. But a big part of it is regularly meeting with our designers and our developers to be like, "Hey, here's this thing that's possible," or if we do something new, "Here's this new feature that we've added." It's not necessarily workshops, but constant communication with our teams so that everyone kind of understands this is the amount of freedom you have, but maybe don't take it all at once.

Otherwise, we are going to have to rein it in and we are going to have to tighten the ropes a little bit so that we can maintain complexity. But we want to be able to give you a little bit more freedom.

Chris Strahl:

Yeah, it's interesting. Being totally honest, a lot of the times that I've seen these systems into practice, people struggle with the maintainability of it all, right? It's the same problem you have, it's not as bad as this I'm making it out to seem way worse. It's like having inline styles, right? This idea that there's something that is an override, but it's sort of this fragment that ends up at the end of the end of a cul-de-sac in a dark alley. And then nobody ever really revisits that to decide if that's something that should exist in perpetuity or if it's something that existed in that moment, in that time. It was only necessary for a little while.

How do you think about tracking down these sort of facets or fragments or do you not even really worry about that long term?

Adam Sedwick:

So as of today, I will say, we are either not really worrying about it or haven't really thought about it. It is definitely a concern as you were saying that, I was like, "Oh, well yeah, I mean if you make this custom thing, it's just kind of there on that component. It's a bunch of extra props on a component." The one thing that's maybe a little easier for us is across the board on our component, it is a single prop that has the same name. So, we at least can say, "Open a code base and search all for this custom styles keyword." And we can see.

And I think that's one of the things that we're looking at as maybe we need to look at this every three to six months. And if there's a whole bunch of these, it's a bigger problem. And I think maybe that's also a

little bit of our saving grace is the idea that we are looking at this from a standpoint of it should only be that last mile customizability.

If everybody is making custom changes all the time, the component is clearly not serving its purpose. So let's reach back out and figure out, "Hey, how do we need to adjust this to actually meet your guys' needs?"

Chris Strahl:

It reminds me of the implementation of Meta's Blueprint System where they have that sort of custom prop that's like a snowflake ID. You're allowed to totally discard the system and completely go outside of it, but in order to do that, you have to have your own little snowflake code you put in there. And I think that's really interesting. I mean, look, I've sounded a little bit down on these things largely because I am a bit of a skeptic when it comes to component level tokens.

But it does solve the issue that you run into all the time of, I just need it to be this. Why can't I just write that down somewhere? Why can't I just put a pixel width in? Or why can't I just put someplace to change my radius value? And that ability to have that little bit of fine grain last mile control definitely eases a lot of the pushback you get when people don't understand exactly where the decisions have been made in the system that represent that component in user land.

Adam Sedwick:

Again, everything is using CSS custom properties and people aren't changing it at runtime, but that way you can at least see it. And because we're using CSS custom properties, it all does just sort of follow the cascade. But also, everything that you can change, we have already identified. So, there's a little bit of extra control there too, where it's not necessarily like you can totally change everything about the component. But we've said, "You can change padding, background color, border radius, font size, depending on what it is."

So for us, it really is that last mile versus, "Hey, I want to rewrite this entire thing."

Chris Strahl:

Yeah. And it gives people a little bit more control. I think that that's probably ultimately what a lot of this stuff comes down to when it comes to systems is this idea of trust the system, believe in the system, but also have the ability, the control and the power to override it when you want.

Adam Sedwick:

For sure.

Chris Strahl:

So, I think it was also funny when we were talking a minute ago about Config and the variables conversation. What did you think when you saw variables?

Adam Sedwick:

Yeah, I mean, I'm super stoked that they exist in Figma. I've basically spent the last week playing with them. The biggest thing for me that was sort of a bummer is that there's no way immediately to just take my JSON file and throw it in Figma and have all my tokens.

It at least gets us a step closer. The idea that you can have aliased variables wasn't something you could do with styles in Figma before. So we can start to build this sort of structure where we have our global tokens and then our semantic tokens and then our component tokens. I actually thought it was really funny. In one of the presentations at Config, they did a deep dive on Figma variables and they had a slide that had all three of these and it said, "Primitives, semantics and then component tokens."

And the first bullet point under component tokens basically said, "You probably don't need these. They are totally optional." So that gave me a bit of a chuckle.

Chris Strahl:

Yeah, I've basically have spent the past two weeks off the grid. I watched most of the keynote from the pool when I was at a wedding down in Mexico. But aside from the name, which by the way, I think, that we should have just called it tokens in Figma or something like that.

Adam Sedwick:

A hundred percent agree.

Chris Strahl:

But aside from the name, I think, the implementation is interesting. I think the hard part is, like you said, it still doesn't feel like a very portable format. I think that we've been playing around, and I actually don't know the resolution to this to see what we can get via the API in terms of a JSON file for tokens. But feeling that is pretty locked down where you can't just export this to JSON and download a file is a little bit disappointing too.

I think that there's a lot here to love, but it's not a full embrace of the idea of tokens that I think that a lot of us were hoping for. And that's okay. I think that, hey, it's a great first step and it's better than the way it was. And I think that it adds a lot of benefit. I think that what was something that sort of struck me was the implementation of developer mode and the idea that there's one file with two different contexts, one for engineers and one for designers.

I did see value there, but I was a little frustrated by that implementation where there's still this idea that this wall exists between design and engineering. And you have one role or the other, and you can't really exist in both worlds simultaneously.

Adam Sedwick:

I agree a hundred percent. As someone who does both design and dev and product every day, there are things in dev mode that I'm super excited for. The component explorer, the ability to open it and just toggle different settings and see what it's going to spit out is super great. I also think that would be super great for designers to have. Or, right, designers maybe don't need to look at a window full of code, but devs may want to know what settings were in Figma.

And so, for someone like me, I am probably going to have to click back and forth all day. And that maybe is a little frustrating because, to your point, it does sort of redefine that these are two different roles. And I think our job with Design Systems was more to break down this wall and say, "No, we're all working together on one team."

Chris Strahl:

Right. Right. And the idea that we're locking down a file and then providing a bunch of abilities to look deeper into that file, it feels like something that should just be there. And that was the part that really struck me is it felt, and I don't know, maybe this is the influence of Adobe, I have no idea. But there's this whole idea of Figma's been talking for years about what can we do for engineers? What can we do for developers?

And there was definitely value there, but there was also this idea of like, "Oh, we need to get these people to pay for seats." And you can kind of see that baked into the thought process of how this rolls out, where it's like, it's a lot of stuff that's enterprise only. There's a lot of stuff that is pretty locked down behind a login that is really just informational that I feel like should just be there, just put it in the file. Anyway.

Adam Sedwick:

Yeah. I mean, sort of relating back to tokens was that they have the modes and it's super cool and you can have light mode and dark mode. Or you can have any kind of mode you want. And so, we started looking at that as like, "Oh, maybe this is different variant on components," but you only get four and then you got to pay for more. And so it was like cool. So I can't really convert all of my components straight to variables and modes because you're only giving me four.

Chris Strahl:

So, honest question, because I haven't played with modes much. Is that supposed to represent state in design or is that literally just like, I'm saving variants of componentry?

Adam Sedwick:

So, they showed it a couple different ways and I think they're not sure what they want to do with it yet. They had a bunch of different variables and then they showed you could use it for light mode and dark mode or different themes. But then, they also showed it as you can take all of your content for different products on your checkout page and each product is its own mode. And so, you could have the title, the description, the whatever are your variables, and then your modes are watermelon, peaches, pineapple. So you can kind of use modes for everything. But again, you're only limited to four, so you can't really dive in and start tying or I guess untying your data from your design, which to me was sort of the most exciting thing about this.

Chris Strahl:

Yeah, I was really hoping for states because one of the big pain points in design is obviously, how do you model an app with 80 states? So I think Knapsack has well over 80 states. And so, it's one of those things where I'm trying to think about how I don't end up with just copy paste spaghetti everywhere. I was really hoping for something like that. But anyway, I'll have to play with it more and see where that goes. But yeah, I mean, like I said, a lot of interesting value there, it feels like it's almost, right?

The way that Figma still thinks about this stuff is this idea of, you have your role as a designer, engineers have their role as devs, and they're probably right about that in terms of current state. But what we're all really trying to push for in the future is this world where a lot of those are more blended and more mixed. And I think that it was a lot of building for the ecosystem we have and not the ecosystem we see coming.

Adam Sedwick:

Yeah. Yeah, I think so.

Chris Strahl:

Well, thinking of those future facing ideas, I am kind of curious where you think tokens are headed. I mean, now that we have variables in Figma, we have a W3C working group that has a spec that's out there that is getting support. People are rallying behind this stuff. When we think about where the future of this is, what do you see?

Adam Sedwick:

I think the future is very bright. Maybe I'm a little biased being on the W3C, I love the spec, our components are as close to spec compliant as they can be. The stuff that I look at and I start to see that gets me really excited is sort of the ability to build tooling around tokens. Because I think we're sort of there with what tokens are.

At the simplest level, they're key value pairs. There's extra stuff in the spec. You can group tokens or you can have families of tokens. But at their core it's a key value pair. I have a name and a value. So, to me, that's pretty set. But the stuff that's exciting is the tooling we can start building around these. Can I build something that dynamically checks the contrast ratio of my text token against my background token and say, "Hey, you have a problem here?"

The ability to start doing more stuff with tokens now that we all kind of agree on what they are. That's really what's exciting.

Chris Strahl:

Yeah, I agree with you in terms of the idea of now that we all know what this is, let's build from here and build really cool stuff with it. Funny enough, one of the most exciting things that I see on the horizon, beyond the ability to actually group a bunch of tokens, because by the way you, mentioned that a second ago, but that is so powerful. The ability to basically say, "Let me create a set of tokens and let me define that as some sort of organizational unit," and then also be able to have that organizational unit reference another organizational unit.

Now all of a sudden, you have a lot of power because you're not just talking about, I have reference tokens. You're saying, "I have a set of tokens that can reference another set of tokens." Because you know why that's cool? You don't have line height on Android. That's why that's cool. And so, the ability to basically say, "This thing that is line height over here is actually these other three tokens over here." And that's awesome. And so, I can't wait to see where that goes, especially in terms of interoperability between platforms.

And I think one other thing that's really cool about this is the ability to, and you're going to laugh because this involves component tokens, the ability to basically say, "Let's define higher order components and higher order tokens in our system that represent wrappers for things that we want in the context of an app." So, I have a layout component or something like that. I want to be able to take a portion of that layout component and have it behave or look or feel different than some other part of that same layout component.

And the ability to do that and to specify like, I have created a three column layout. I want the middle column to look differently and I want that to be configurable as a part of the implementation of that

component. And that innately involves component token that then can vary what that middle or left or right-hand column or top or middle or bottom column would look like.

And I think that that's going to unlock a pretty incredible amount of customization and ability to do really interesting things with brands and with application of components and design systems that we've never been able to do before.

Adam Sedwick:

I agree 100%. I feel like the big hurdle over the last year really was just getting everybody to agree on what this was. And now, we get to solve or at least investigate these really interesting problems. Because I think you're right. I think the ability to dynamically change things is super exciting. The interoperability across platforms, going back to Figma and Config, the variables are super awesome. Maybe there's a bunch of custom stuff in there.

We have written ways to handle that into the spec, but I want to be able to take my token file that I'm using in Figma and use it in Code and use it in Knapsack and use it on an iOS app. I don't care where you're visiting my tokens from. I want that to all be managed in one place. And now that we all agree on what tokens are, we can start doing that sort of stuff.

Chris Strahl:

Exactly. I always view this idea of some central repository that holds the actual JSON, or however story tokens probably JSON, that represents kind of where that should go across all of your different apps and ecosystem. And I think that that's one of the things that is going to be hard is this idea that that can't be owned. It needs to be something that isn't owned by Figma. It's not owned by Knapsack. It's not owned by any one app.

It's something that just exists as an open spec or standard for how you implement a brand everywhere in your ecosystem. And that's what the value of this really is representative of. And I think the person that tries or the app that tries to control that is in for a really rough ride because it needs to be something that is implemented across an ecosystem. And if I can't get to it to implement it, that's going to be a big hurdle to adoption that most people aren't going to cross.

Adam Sedwick:

Yeah. For sure. We were discussing it in the community group chat the other day was in Figma, they have the ability with their variables to say, "Oh, I only want this to be able to be used for a stroke or I only want this to be able to use for a fill." And we were sort of like, "How would you handle that?" I went through it. I looked at our spec and I was like, "We have this whole section of the spec for extensions." And that's exactly what this is. It's the ability for any individual app to say, "Hey, if you're using this thing in Figma, you have these extra options. If you're using this thing in Knapsack, you have these extra options."

Chris Strahl:

We actually implement that for one of our features in Knapsack.

Adam Sedwick:

Right. And that's super great. And I think that really does, again, get to the heart of what we're trying to do is this should be one thing that everybody uses. That's how it becomes the most powerful. That really is in the spirit of design systems. It's the idea of build it in one place and use it everywhere.

Chris Strahl:

It's really awesome to think that we have this common idea of what all this should be that also is very extensible. The extensibility of it matters so much because like you said, like a stroke or a fill, what we call an asset set, what any number of things are that represent new kinds of tokens or new ways of using tokens in our applications, the ability to extend it into those areas is really, really just getting started.

Adam Sedwick:

Yeah. And I think the sort of last thing with extensions at least is maybe you guys over at Knapsack come up with this new genius way to use them, right? And that's not part of the spec. But then other people see that and like, "Oh, that's super cool." Well, you have it specked out and documented in your extension and how it works and all of this other stuff. Somebody else can just start using that. And if enough people start using it, it theoretically could get pulled into the spec as like, here's how we start doing this. I mean, I think that's what they're doing with even the HTML and CSS specs is like, "Hey, we've been doing this in JavaScript for a couple years. Everybody really likes it. Let's pull it into the actual spec."

And I think that's been working really well for them. So we might as well try it too.

Chris Strahl:

Well, hey, Adam, this has been an awesome conversation. I really appreciate you taking the time to jam and yeah, let's have you back on again, talk more about tokens, talk more about where this is all headed. Just really appreciate your time.

Adam Sedwick:

Yeah, thanks for having me on. I would love to be back.

Chris Strahl:

Awesome. Well, hey everybody, this has been The Design System Podcast. Thanks to our guest, Adam Sedwick. Really appreciate you all. Have a great day.

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. Have a great day.