

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.

Chris Strahl: Hey everybody. Welcome to the Design Systems podcast. I'm your host Chris Strahl. I'm here with Jay Smith, who's the design lead at REI works on the Cedar Design System. Jay, welcome to the program.

Jay Smith: Hey Chris, thanks so much for having me.

Chris Strahl: I'm excited for this conversation because we had a moment before the show to talk about a theme and when we talk about what are the core things that a design system represents in terms of value, you get lots of people talking about efficiency or they talk about time saved or they talk about building better product. But almost always people also mention the word consistency. And this is going to be a chat that is really focused on unpacking what consistency within a design system means. And so lemme start out with that. When you think about the application of consistency across REI, what does that mean to you?

Jay Smith: Yeah, so if I zoom out before thinking about it within a design system context, I think about the channels that our brand and experiences show up in. And so there's digital channels, there's physical channels, we have stores. Often those things interact with each other. And so consistency of expression for that ultimate end user or customer, how are they benefiting from our work to have consistency of experience across the various touchpoints that they have?

Chris Strahl: Yeah, I mean I think about my relationship with REI and the channels that exist there. You guys send me wonderful coupons via direct mail. I have the membership and that membership sort of experience. There's a little card that hangs on my key chain still. And then there's a store that I frequently go to either to buy stuff from or return. There's email. So how else would I found out about the garage sale? And then there's the shopping experience in rei.com and I'm just one person. And those are all the different ways that it kind of touches me.

Jay Smith: Yeah, absolutely. And it goes beyond just visual design as well. It's consistency of language, consistency of message. All these things are something that we think about here across those various touchpoints.

Chris Strahl: So one of the concepts you brought up was this idea of logical or computational consistency versus perceived consistency. And I think this is really interesting because most of the time when we think about statements like that, we put it in

terms of performance. How often perceived performance i.e having a progress bar spinner is actually more important than the performance itself. It doesn't necessarily matter if you have a subsecond load time on a webpage, as long as you have something that lets a user know that it's loading in a similar vein, it may not be necessarily as critical that you have something that's computationally consistent as long it presents as consistent to a user. Unpack that for me a little bit. How did you come to that conclusion?

Jay Smith: Yeah, I think consistency from a design system 101 standpoint, it's like, okay, how do these parts and pieces show up in a uniform manner across various contexts? I've started to see the system being used in ways which are more and more out of our direct purview or control and all the various dynamic experiences that could be created that you don't have the ability to oversee and think about how the parts and pieces assemble themselves in a consistent way. And so now it's become more how can you construct the various decisions that go into achieving consistency in a way that can show up in a way that is consistent for that end user, but it can't be predicted upfront or seen upfront. And I think that's become a very sort of interesting problem space for us right now.

Chris Strahl: So to take sort of an old example, you used to have some piece of your design system where you were able to wrap your head around the majority of use cases for that component or that pattern that existed. And now because of the broader adoption that you have and the complexity of things, I assume to some degree it's much more difficult to predict all the potential use cases for the patterns that exist in that design system. And because you can't always envision what the permutations of the end state will be, it's a lot more about defining the rules than it is about necessarily understanding all the states. Is that a fair summary?

Jay Smith: Yeah, it's like the problem now is not specific implementations and deciding what consistency of those implementations are, but can you define a model that can protect the concept of consistency for circumstances that you can't account for upfront?

Chris Strahl: So what does that look like in terms of a concrete example inside of C? When you think about that application of before, hey, it was kind of like we knew all our permutations of a button and where that button might get used, my favorite punching bag component or now we have to have a system for understanding how CTAs and buttons broadly work inside of the system. Is it something like that? Or give me an example that helps hammer that home?

Jay Smith: Yeah, so one of the example that we're working through right now is can we define a token language that has an understanding over a certain sort of background type and have it be aware of the context that those tokenized decisions are living in. And then what do they do when they're applied in a

different context? Can they be aware of where they're living? Can they be responsive to the various containers that they live in? And this opens up a whole new idea for enabling things like themes and palettes to exist. We don't necessarily know, we have a good guess around what our primary text color is going to be over a certain background type, but we don't necessarily know where that background color type is going to be used across various teams and implementations. So if we can manage that relationship, we can at least have a solid understanding of how it's going to show up in different contexts and be really sure about other considerations. Will that relationship be accessible in contexts that we weren't able to predict?

Chris Strahl: So how do you decide context right now? Is the context usage in a component? Is it in some sort of higher order component where you're basically defining some grouping of other components or some nesting? What does that look like in terms of the structure?

Jay Smith: Yeah, so right now it's sort of a two faceted approach. I am very interested in defining what is the sort of core expression for rei.com, for example, and what pallet sets are most appropriate for the majority of experiences that we have. And that becomes a set of decisions that the system can deliver. But adjacent to that, we can also deliver to teams the framework for those decisions for them to create their own pallets and themes. So if there's a different team who wants to focus on, for example, maybe like our membership R E I membership that right now might not be something that we want to deliver at scale, but they might find useful within their team to manage and maintain a membership expression. And so we want to give them the tools to do that knowing that someday it'll likely be that we can ingest that back into the system.

Chris Strahl: So when you're thinking about that expression itself, that is something that can represent a static set of assets. We have an understanding of this is your pallet and this is how that pallet applies in that particular use case. And that was generated by some higher order systems. Thinking at the core level of the design system, are you guys doing stuff in runtime as well where you have the idea of I'm determining a context on the fly and then changing things? Or is it all that static generation that is then used in different contexts that are more determined by the implementation?

Jay Smith: It's sort of interesting. We're just getting into this and what's been really exciting for me is developing, for example, a gray scale. Everybody's got a gray scale over a primary background color. And one of the bodies of work that I did was like, okay, let's look at that gray scale across three other background types. The first one is sort of secondary background, and then let's solve for the inverse of those to support something like dark mode. And with those pallets as defined, if you construct, say a given component across your primary background and the token relationship is there, if you want to construct that component across a different

background type, the relationship is maintained. And so the expression of that is very quick to determine. And I think that's extensible.

Chris Strahl: That's interesting. So the idea is that by designing for one set of contrasts, you're designing for a whole bunch of contrasts that are easy to extend into based on implementations.

Jay Smith: Yeah, exactly. And one of the things that's most interesting about it is it's a bit more upfront work in defining four different gray scales and who's defining multiple gray scales for a light mode. But what's interesting about it is you can maintain contrast values that way and maintain the same sort of expression of type across background types through managing that gray scale, which can be automated.

Chris Strahl: I see that as this interesting asset where you have this ability to say, I have a gray scale, that gray scale has all these different contexts that it applies to, and then I have two or three or four other ones that apply in this other set of contexts. And as a result, you're able to have that same visual language applied in lots of different places. So are you using the design system beyond? Do your print people be like, yo, give me some of that design system magic. What are those hex codes for your gray scale man?

Jay Smith: Yeah, I mean it might be a very long-term vision. I think what's important is that the model itself is scalable and where we want to apply it first is digital commerce. Let's gain value and traction there, demonstrate performance, and then open the gate to what's next. And there's digital environments within retail stores that we'd love to extend into in the future. But just to circle back on the idea of managing these gray scales across different background types, what else does this enable? It sort of plays into that idea of perceptual consistency and if you have a primary type color across white, that primary type color, if you maintain the hex value on a different, let's say a lighter gray, it changes the perception of that color. And so I think it's more interesting to try to manage that perception of that color through this token work as a way to maintain consistency or cohesiveness, whatever you want to call it, across those expressions.

Chris Strahl: So you said you're just getting into this. Is this something that's largely based in color right now or is there stuff like variable fonts or typefaces and spacing and stuff like that that you're doing as well?

Jay Smith: Color first because that feels most useful in terms of demonstrating performance and impact of the model. But there are immediate use cases identified here for extending into typography. And if you want to convey this as a theme or a palette set, you can quickly get into a standard type expression for rei.com and then apply that to a different context of maybe the type in this

situation needs to be a different weight or a different color. I would rather manage that through pallets, then have that work be done in a bespoke way.

Chris Strahl: I think that the interesting part of this is the very orchestrated way you think about the system, and I think that it's also interesting because the system being the focus versus the necessary results of like, Hey, it's got to be like grey on white is great because yeah, you talked about gray scale, but you could apply that very broadly to nearly everything.

Jay Smith: Yeah, absolutely. Our brand green put that over different background color types. The perception of that brand green is going to be different. And the digital space is very dynamic. The different situations that you might have a customer experiencing our brand or a wide range of different experiences. And so how do you design for that? And I think a more dynamic expression of brand is something that could be more tolerated. What do you let go of in terms of control instead of pointing to a hex value and say, this is brand green. Instead you point to a model that says this is responsible for carrying the idea of what green means in these different contexts.

Chris Strahl: And it gets at the root of one of the big challenges I've always had with palettes is that colors individually are kind of meaningless. It's one color on top of another that creates meaning. I mean they're not totally meaningless. Like gray conveys an emotion of feeling a purpose and brand, but if you put it on top of brown, now all of a sudden you have a contrast that you can understand the perception of that green and by thinking about it always in at least a set of two, I think that's a fascinating way to think about expression because now all of a sudden your brand primary color doesn't really mean as much because it could be any number of colors. It just matters if it tells the right story based on the color it's being compared against.

Jay Smith: Yeah, exactly. And we're realizing that with our gray scale now where, yeah, it's four different values of gray for primary type, but you put 'em together and they look the same, perceptually they're the same. And I think that is a more interesting design problem to solve.

Chris Strahl: So this is pretty heady. How is the implementation of this? How does it look? Is this, I have 10,000 tokens, I have all these different sets of things that we've effectively figured out how to group them by. What does that look like when you think about four different gray scales?

Jay Smith: I mean it's really messy on the system side. Well, I wouldn't say messy. It's a lot to manage on the system side and the sort of challenges. How can you make that as easy to use for the consumers of the design decisions, product designers and engineers and not pass that complexity onto them? So kudos to our engineers on the system team. It's a lot of work to manage and maintain, but ultimately I think the design value there will be worth it.

Chris Strahl: I was just thinking about a shirt that I bought from REI that is sort of burgundy in color and the logo representation of r e i on that shirt, it's an r e i branded shirt is I think orange or something like that. And I wonder, do you all harmonize between a brand team that is associated with applications of the brand like that versus the digital brand as well? Because I mean that's a substantial deviation from a core brand color. So this concept isn't only digital, it obviously applies elsewhere, at least in apparel. What does that collaboration look like?

Jay Smith: Yeah, that's really interesting and it sort of opens the door to maybe some edges of where this model might not scale to for things like hard goods or soft goods when we're making a tint or making a shirt, there are seasonal palettes developed for those things and the cycle seasonally of how those things change. It's very fast for this model in digital, I'm sort of thinking in fast and slow what needs to be changed at a slower rate. And I think that core expression certainly needs to be a slower rate of change. And can you design in through the system a way to protect that sort of core expression and then allow for other color types maybe to move at a faster rate on top of that,

Chris Strahl: An analog for you would be some sort of core brand that is slower iteration, slower moving, but then an application or an instance of that brand that could exist in a more rapid kind of environment where it could change seasonally or could change based on the specific application of that palette.

Jay Smith: And this sort of gets into our contradictory almost mandate in a way, or this conceptual idea of a mandate for the system, which is how do you enable teams to rapidly experiment but at the same time support consistency of expression? How do you do both of those things? Well, because you can over index on either if you have teams doing experimentation, eventually you'll see patterns emerge and if you can't capture them, you'll lose consistency. But at the same time, if you over index on consistency, you tend to block teams on experimentation. And so that's sort of a related problem space for us.

Chris Strahl: Yeah, it's interesting how that idea of constraints as the creation of consistency or the application of consistency and also that idea of rigid constraints inhibiting creativity. It's a theme sort of throughout the design systems concept, right? Because we need to have systems to help us scale. Like choosing 64 million different colors is not practical for every single design decision, but also only choosing between two maybe equally impractical.

Jay Smith: It reminds me of when we first started doing or working on our system, everybody does a color audit and you see, oh, here's 20 different values of gray, what are we doing? But it's so funny that it sort of circles back around now where if it's done intentionally, if you were to audit a site with this sort of dynamic palette in place, you might get back 20 different values of gray again. But it's intentional this time, so it's correct.

Chris Strahl: It wasn't created with the eyedropper tool, it was created with science, right?

Jay Smith: Yeah. Put a model in place that can scale. I think that's a more long-term solution. I think about that sort of dual mandate idea. It's like how do you accommodate faster rates of change? How do you support experimentation and curiosity? Have a system that encourages flexibility of use and then at the same time encourage slower rates of change, protect things like core brand identity, make it predictable, make it maintainable, and reinforce a brand language over time.

Chris Strahl: So I think it's also interesting because you've tried to inject a fair amount of the Re I culture into this as well, where you all are cooperative. There's a lot of your culture that is about contributions and trying to think about how that applies to your design system. I think that first of all, it's amazing that you consider this as a part of the values you're injecting into these decisions, but I'd also be curious to explore what does that look like when you think about Re i, the cooperative and how that applies to the systems that you're creating?

Jay Smith: Yeah, it's sort of an interesting idea. How can our business model at a macro level as a consumer cooperative, what can we sort borrow from that in terms of building a system that might function the same way? What does a design system builder cooperative look like? I think that's an interesting sort of space that doesn't have an easy answer, but we're exploring it right now. I think about the fact that we see the system as owned by the consuming teams and how do we provide the right sort of on-ramps for them to put their hand on the scale where and when they feel that there's value to do so.

Chris Strahl: Give me a practical idea of what that looks like. You view ownership as something that is much more last mile where you're in service to those than product, but what does that really mean in terms of the practical contribution to the system?

Jay Smith: It's a hard problem because it introduces risk, right? It's very easy for the design function to drop a bunch of assets into Figma and say, add this to the system. It's very hard on the engineering side to say, will happily accept the debt of maintaining this. There's a fine balance there, but we've had some successes and we hope to use those as examples for how to scale it out more broadly. But I think the important thing for us thus far has been to highlight the successes of what those initial contributions have been and then use that as a model for showing other teams of this is what good looks like when it comes to contributing back to the system and then reinforcing that.

Chris Strahl: And do folks between both disciplines really recognize that and hold that with that same degree of reverence. Do you find that that model is generally accepted as a part of the culture of contribution or governance inside of the organization?

Jay Smith: I think it's harder for me to speak to on the engineering side. Designers are generally very open to the idea of contribution, but everybody's busy. So how do you encourage contribution to this system? How can you show what a good contribution looks like as an example, and how do you provide the right incentives for someone to participate in the first place? There are plenty of incentives for consumers to adopt the system, but I see very limited discussion around what sort of incentives are in place to help people participate in the first place when teams have their own goals and their own work to do, and how do you drive the progress of the system from a consumer contribution angle? I think that's a hard problem.

Chris Strahl: Yeah, it's interesting. At Knapsack, we talk a lot about adoption because that's what everybody has decided is the correlation metric for design system success. But usually when we're actually doing an implementation of Knapsack, we talk about what does the contribution model look like for the system? Because that inherently does two things. It helps with the cross-disciplinary nature of the creation of systems. Systems that just serve design or just serve engineering or just serve product. They tend to be of less value than things that serve the entire process. And then as a second aspect to it, by having contributors, you innately have people that have adopted the system. And so that contribution allows for that democratization between disciplines and also allows pretty much anybody to leave it better than they found it. And sort of baking in that philosophical idea of cooperation and cooperativeness. I think that's a really awesome way to sort of foster that desire to collaborate.

Jay Smith: Yeah, I mean this idea of cooperative by design, I mean it's on the homepage of our documentation site and playing around with that concept has been pretty powerful and sort of interesting at times. I mentioned incentives for contribution and the system can't assign people to work on these problems. And so how do you pull from maybe altruism of contributors? We sort of experimented with this idea of, okay, well if you contribute to the system, the system will plant trees on your behalf. And so that's been an experiment that we've run and we have 5,000 trees that we've helped plant in various areas through contributions, and I think that's pretty cool. It's like a passive effect of the system growing. The system grows, we grow trees.

Chris Strahl: That's amazing. And talk about a crossover from a digital world to a physical one, having that tangible effect be really present through contribution. That's super cool.

Jay Smith: Yeah, it's really exciting. And it's very REI.

Chris Strahl: Yeah, I mean I love the living of the culture through your systems. I think that's special. It's one of those things where we all talk a lot about design principles and the purposes of design systems, but we often don't connect them deeply to

our values and our cultures as an organization. And I love how connected you all are to it.

Jay Smith: Yeah, I mean, I love that you brought that up, connected in principles. It sort of ties into our approach of our documentation site. We have this principle of adjacency and proximity. How can you put things together in a way that is more useful and meaningful? And that's sort of the next step for our documentation platform is as sort of a remote first company now, how can you start to put the system parts and pieces, the UI decisions next to things like brand decisions, brand language next to things like research and content strategy so that the entire package can be most useful for the people building digital experiences.

Chris Strahl: I love the way you think about it across lots of different disciplines, but also different facets of the discipline. I think that people don't often put UX research inside of their design system, and I think that probably they should much like how content and design systems is something that we've talked about on the podcast a couple of times. But I have seen very few enterprise design systems that have a strong content, learn anywhere inside of that. And I think it's great to have this idea of everything everywhere all at once, but also in a way that is organized and valuable more than the sum of its parts.

Jay Smith: Yeah, exactly. And ultimately, I want all the decisions associated with building quality digital experiences as close to a central source of truth as possible. And we're starting with UI because that's probably the hardest thing to do because there's designing code parody to manage and maintain, but alongside those decisions are research decisions, content decisions that also come into play. And putting those things together ultimately is going to be more useful to people trying to put those parts and pieces together to come up with the holistic experience.

Chris Strahl: It does come back to brands a lot as well, where what's the, not just expression of the brand, but where does the stewardship and maintenance of that digital brand live? And this is why, at least in my opinion, design systems are fundamentally linked to brand and that the most successful design systems that I've seen are the ones that are closest to the brand. All of those decision-making pieces that align to that central source of truth, that expression are fundamentally brand choices. And they're all about how are we going to express this via components or tokens or assembled prototypes in a design system, whatever that looks like. It's all about how do I lay out an intention and then how do I showcase the expression of that intention in a digital product? And it has limitations because most design systems aren't asset management solutions. Most design systems don't have a lot to do with physical branding and a physical space, but a lot of it is pretty fundamentally tied to the way that we express ourselves in digital.

Jay Smith: Yeah, exactly. And I think that relationship between brand and design system can be strengthened by putting those sort of decisions close together and the system has no intention of maintaining control of that information. We're more interested in being the distribution network of that information. The analogy that I've used in the past is it's really expensive to build a rail network, but it's really cheap to ship freight on rail. And it's in the same way it's expensive to build these systems, but your ability to deliver brand decisions efficiently and quickly to as many people as possible 24 hours a day through a documentation platform that's useful. And that's the other value that we're bringing here is the network effect of these brand decisions. So yeah, it's exciting.

Chris Strahl: So when you think about the work that you've done in brand right now, this expression of this centralized source of truth, what has been that evolution that you've seen? Because I know that you've changed a lot of your approaches and specifically things like how to be more dynamic, but what else has changed?

Jay Smith: Yeah, I think what we've sort of seen in this year in particular is largely a leadership organization that has said, Ooh, system has been successful to a degree, and we want more of that. And we think that there's opportunity for the system to have more of an opinion on UI expression. Thus far, we've been pretty neutral. Here's our parts and pieces, use 'em as you want. But we think that we can drive better brand consistency by having a more opinionated point of view on what does more complex UI look like. And we're sort of at the point where we're trying to think about how do you maintain separation of concerns between small parts and pieces and large ones? Where do those things live and how can you make them work with all the things that we've already talked about?

Chris Strahl: So you have a bunch of basic stuff that was pretty unopinionated, and the expression of those is relatively easy, very encapsulated. Those things sort of exist and you can remix them or do with them what you will. You're now in a place where maybe the extensibility matters a little bit more than the encapsulation where you're trying to think about, okay, these have an opinion, they have a set of constraints that aren't as freeform as the components in UI patterns you were creating earlier, but you want to be able to use them in a wide variety of places still and still apply the same strategies and other systems to them. Is that a fair representation?

Jay Smith: And the scope of some of the more complex UI isn't going to be as broad as those smaller parts and pieces, but we're sort of looking at it through a similar efficiency lens. Where is there different types of efficiency to be had by having more complex UI provided by the system? And maybe that means removing certain conversations from stakeholders and consuming teams. So if the system says, here's a lead example, top of page experience that we're going to provide that gives various stakeholders something to point to as reference and say, this is our current best practice, we should use this from an efficiency standpoint. And

if not, if there's a business justification for not doing that for your specific context, we'll make that business case.

Chris Strahl: You're almost used as a mechanism then to apply constraints about business decision making, unless about the technical decision-making of the componentry itself. It wouldn't be something like, Hey, we have this thing that has enforced technical limitations. It's like, Hey, we have a shopping cart experience, and that shopping cart experience should be the same unless you're buying something that fundamentally doesn't belong in a shopping cart. And that would represent sort of the process whereby you have that opinionated stance on what that experience should look like and how it should be used.

Jay Smith: Yeah, exactly. And one of the questions to answer for us is how far does designing code parity exist in a space like that? There are very real technical constraints that limit us from, here's a, you're ready to make code for that circumstance. And the range in which we can distribute that is more limited.

Chris Strahl: My personal belief is that we're going to look at design and code parody with the same sort of flippant attitude. We look towards pixel perfection today where we had this era where that really, really mattered to us. And I think that that era is slowly slipping away where we start to say that design is one expression that has its own limitations because design tools are innately an abstraction, and they're not going to function the same way as the medium that something's destined for, but they are really good at rapidly iterating and showcasing intent, whereas the actual implementation of something, if you're only doing what exists in a design file and Figma, you're limiting your expression in that medium. You're limiting the value of the web or of a native app or something like that because you can do things in that medium that you just can't do in a design abstraction. And so I definitely see that as on its way out.

Jay Smith: Yeah, it's sort of like accommodating those faster rates of change. How do you enable flexibility of use and while at the same time constrain the parts and pieces that you really want to lock down that should have that slower rate of change?

Chris Strahl: Well, Jay, I want to thank you for being on super informative, really interesting to get such a detailed and nuanced take on the idea of consistency and how it gets expressed across an organization. And we touched on a lot of things today, so I just want to say I really appreciate the breadth of knowledge and the thoughtfulness.

Jay Smith: Yeah, thanks so much. Super excited to talk through some of these concepts, and I hope to see them play out on our documentation site over time.

Chris Strahl: This has been The Design Systems podcast. I'm your host, Chris str. Thanks. Have a great day, everybody.

This transcript was exported on Oct 03, 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 with 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.