

Episode Transcript

112 - Creating Effective Design Systems for Small and Mid-Size Companies: Apiture's Journey

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 at [thedspod](https://twitter.com/thedspod). We'd love to hear from you.

Hey, everyone. Welcome to the Design Systems Podcast. I'm your host, Chris Strahl. Today I'm here with Ashley Willard and Skim. It's not often that I get a customer duo on the podcast, and so I'm excited to chat with you all. You know, you all have kind of a variety of different roles in the landscape of design systems at Apiture, and so I wanted to have you all just kind of talk about what you do for a second.

Ashley Willard (Apiture):

Sure. I will start. This is Ashley. I am the director of product design at Apiture. Apiture is a smallish fintech company. We're a SaaS company. We serve banks and credit unions, and we build banking software for them. We have, I think, around 350 employees, give or take.

So, yeah, on the smaller side, and I run the whole design team. So our UX designers, plus a couple of engineers that we now have working on the design system. But it's a very new initiative we've just been starting over these last few months, and so we're figuring it all out right now. And Skim is on my team.

Skim (Apiture):

Yeah, hey, I'm Skim. I'm a UX engineer on Ashley's team and focusing on the design system and kind of helping also just bridge any sort of tech UX gaps wherever I can.

Chris Strahl:

Well, the reason why I wanted to chat with you all and have you on the podcast is we oftentimes talk about really big enterprise problems on this show, and it's all about, like, what the biggest companies in the world are doing in terms of design systems, and a lot of them are knapsack customers. But at the same time, we don't talk that often about what we try to do with companies that are emerging or growing or tend to be on that smaller side that still have things that the design system helps them solve but aren't necessarily in the same place as a really big company is being able to talk about that, I think is the real focus for today. With that in mind, talk a little bit about where you all are at in terms of how you build products and the kind of

landscape as it exists today.

Ashley Willard (Apiture):

Sure. Yeah. And I think we are becoming more and more aware of that every day, that the solutions that might work for a giant enterprise company aren't necessarily going to be the same ones for us because we are, as skim put it, before we started recording, in many ways, we're a special snowflake. We have certain needs and requirements and things that probably are different than what a large organization that has a large design system team is facing. Really, our design system is a team of two engineers at the moment, one designer who's also dedicated to a number of other design things. So he's not dedicated just to this team. Similarly, we have a technical project manager on the team who's dedicated to running a few other teams, a QA engineer who is on a few other teams as well. So it is really a small and scrappy kind of design system team, but we're well integrated with other teams and other things going on in the company.

So where at a larger enterprise, I think you see a design system team working very much as an organism in and of itself. We aren't that way. We feel more cooperative and collaborative with the other engineers in the organization, the other designers in the organization. It's a much smaller and closer knit.

Chris Strahl:

Yeah. When you think about big enterprise customers, there's oftentimes this pretty distinct separation where you have the design systems team, which is building that hub, and then you have all those consumers that exist as sort of nodes that connect in some way to the hub. And very often in an enterprise setting, the people that are building that central hub have very little direct contact or connection with the consumers, and you all are mostly integrated with them.

And I think that that's actually fairly common across design systems landscapes, especially as mid market or smaller enterprise companies really get on board with these concepts. But I think that where you all do stand out as being unique is you have big enterprise ambitions here. I think that the idea of a lot of what is a design system for a smaller company is often like, hey, here's a doc site in a component library, and there's maybe some matching component library that exists in Figma or somewhere else. You all are really looking at trying to have a much more integrated take on how your design system powers the way you build product. So maybe talk about that vision a little bit, because I think it is interesting when you say, hey, look, we're a small team.

We don't even have a dedicated design resource. But this is what we're actually trying to do with a design system.

Ashley Willard (Apiture):

Yeah, we have a history of having a few different solutions within our suite of products that some of them are brand new, built in the last few years, others are years upon years upon years of

mergers and acquisitions and this and that. And that means that each of those things has its own completely independent code base and pretty independent team. We're trying to break down the silos across these teams now so that we can become more of a cohesive operation. And so having a design system in the mix, there is a big part of solving that problem, because no longer does left hand not know what right hand's doing, but rather left hand and right hand are using the same ingredients to build the things. So that's really important to us, to be able to unify both internally, obviously, for efficiency, to be able to build faster, to be able to build more consistently, but also externally so that we can deliver faster and create solutions more quickly, create more of a unified look and feel throughout our solutions.

Skim (Apiture):

And it's also worth mentioning that most of our engineering team is full stack engineers. We have a few like front end, like focused folks, but especially when you have a team that's full stack having a tool they can use for the UI stuff. Because I was a full stack engineer for a little bit, but I feel like you're not going to be equally good at both. Maybe you can be. I wasn't, unfortunately. But you know, I think like you lean one side or the other. So if you're someone who doesn't have as much of like a UI background as like a full stack engineer, having things built out for you in a UI toolkit is going to be a huge help.

Chris Strahl:

Yeah, I think it's funny, right? The full stack moniker in the land of design systems gets a lot of hate. And it's largely because like, you know, while it might be JavaScript all the way down, it's still a very distinct difference between those people that are building business logic systems in node versus the folks that are building UI and reactors or something like that.

Skim (Apiture):

Different ways of thinking for both of them, even though it's like, oh, it's similar technologies, but it's very different ways of thinking. So having to switch between the two is tough.

Chris Strahl:

Yeah, I mean, the back of the front end and the front of the front end. I mean, there's a lot of different ways to segment the user types, but I think that the point is to basically say, hey, this design system helps people that maybe don't have a deep expertise in interfaces and UI, be able to build the right thing and to.

Ashley Willard (Apiture):

Be able to not have to think so hard about it or dig for answers about what should the padding on this thing be and things like that. Right. The more consistency and automation that we bring to those decisions, the less that all these disparate developers who are in their home office many, many, many hundreds of thousands of miles away from each other. Because we are a fully distributed company that adds its own challenges for collaboration, the more we can just answer that for everybody, the less they have to worry about it and the more they can just focus on building the thing and making it work. Yeah.

Chris Strahl:

And skim talk about the technical makeup of the design system for a second, like, what is in there.

Skim (Apiture):

Yeah. First off, the less glamorous part of design systems is testing assumptions. So we kind of had an assumption and then talk to people more. We're kind of rethinking, like, okay, how can we make this integrate better with what people are doing? Rather than do something, we're going to do something a little bit nice and flashy. But right now we're looking into supporting angular. Throughout the 14 different platforms. There's some existing stuff that is written in angular that we are going to look at how it serves what we need. So we have to go through an audit there.

Not the most glamorous, but part of the stuff that I love that draws me to design systems where it's like, okay, cool. It's a lot of just how are we going to make this work for the engineers who are consuming this as well, and, like, talking to them a lot, having conversations and getting that feedback of like, does this work for what you need? Like, try using this and, like, give us feedback. What else does it need? And we're getting feedback, obviously, from the pilot group, but we're also doing some ux exploration that's mostly on gym side with the other engineering teams to figure out, okay, so, like, what are the things we actually need? Because, I mean, as folks who have worked on design systems before know, if you come in and work on a design system, you really need to just be listening constantly to know what needs to happen. I feel like my first few months was just kind of listening and taking notes and figuring out, okay, what really needs to happen for people. And I think we're on our way there now. We're getting a lot more buy in from people giving us feedback and hopefully going to increase that with our summit next week. So I'm excited, really. I just want engineers to shout to me if something isn't working, like, that is my happy place, where it's like, oh, hey, this thing needs this thing.

Why isn't it doing that? I'm like, yes, tell me so we can adjust. Because it's like, if you're building a thing that isn't the thing you were specifically going to use, you're not the one consuming it. It's very easy to build it in a way that lose track of how it's going to be consumed. So I've very much enjoyed having a lot of that feedback and a lot of that connection with the engineers across the company.

Chris Strahl:

You all have seemed to set up a pretty extensive feedback ecosystem. And again, diving into the human side of all this, you have a lot of different products, a lot of different teams, a lot of different technologies. Even though it's all unified and angular, there's lots of legacy stuff that's out there that sounds like it may need to be re examined in a structure where you have this pilot team that you're working with, and then ultimately the intention of having a small team serve a couple hundred people across 14 different parts of the organization. That feedback loop for you around the design choices you're making seems to be something you all have focused a lot on.

Ashley Willard (Apiture):

Definitely. I think we're applying ux practices to how we're building our design system. As Kim was saying, we're really not just trying to make our own assumptions about things, but actually trying to validate them by getting feedback and testing things. And we did just pivot from a web components and stencil approach to doing angular native components. We understood the pros and cons of that. We were a couple of months into building components, so we built five or six components in the web components and stencil approach, and then we pivoted because of feedback from engineers, because of things we learned from them. Right. And it's never great to have to pivot, and we realized, and we understand that, right, but it's much better to have to pivot when you only have five components built than if you have 50.

And so if we were going to do that, now is the time to do it. But Skim also mentioned Jim. He's our designer on the design system team, and part of his process, as he starts what we are calling discovery, is to interview the engineers about their understanding of what is a button or what is an input or whatever. Make sure we're using the same terminology. Make sure we're thinking about all of the things it needs to be able to do, you know, all of those things. Just like we might interview people as we're getting started with designing an interaction.

Chris Strahl:

So take me into the room of one of these feedback sessions because you took something that was a fairly codified technology choice and you basically said like, hey, this isn't the thing that's serving the needs of our users. And I want to understand from a really practical standpoint, what does that conversation look like? And when you're in that conversation where you've already started down a pathway to go serve the needs of these users, how does that ux principle of making sure you're talking to your users and getting their feedback really play into that conversation?

Skim (Apiture):

We had a really good session about the badge component that I'm working on, and I feel like, as an aside, when you're working with design systems, I feel like, I keep saying like, oh, I feel like this slack conversation or email should be a meeting rather than the other way around. Like, actually having everyone looking at the same thing in like a Zoom meeting, since we're all distributed, has been so incredibly helpful versus like, trying to talk about it over slack. It's like, let's just all get on a Zoom call, look at the thing, and then we can agree, like, okay, what is a badge? What isn't a badge? Which got, you know, fairly deep and philosophical, but now we have an answer. I don't think we could have done that as easily over slack. So it's nice to just, we're like, okay, this is our process now. We're all going to get as many people who can be available to talk through this and figure out where we use these things and how we should be using them and if they're even called the right things.

Chris Strahl:

So you're inviting people into the conversation. You're trying to make sure it's happening in a

medium that makes sense. You're talking about innately visual things. Like Slack has some visual capabilities, but for the most part, what you're trying to do is you're trying to understand how you name things, how you structure things, how you think about things, and you're trying to invite as many people as possible to contribute to that chat.

Skim (Apiture):

And I mean, also there's, people are working on a lot of teams, so we do have to do a lot of the slack interactions as well. I'm kind of hoping that after the front end summit, we can have a bit more in terms of conversations on the engineering side because I feel like we're doing a lot on the UX currently, and that's been amazing. But I would love to start doing more office hour things with our engineering team just so I know what questions they have. Knowing the questions and knowing what people are thinking about is very helpful for directing all of this.

Ashley Willard (Apiture):

It's important to making sure that they buy into it in six months or eight months or nine months. Whenever we start to say to them, okay, we have a pretty robust set of components ready for you to start consuming. We want them to want to do that. Of course, the expectation from a corporate level is that they all will do that, but we want them to be happy about it.

Chris Strahl:

It doesn't want to feel forced. And I think that, like, that's an interesting way of thinking about these feedback sessions. Right? It's like, yeah, of course, on the one side it's you all getting the information you need to build the best design system, but on the other side, it's securing Brian and generating some excitement about actually consuming it and using it. And this is actually one of the things that, you know, I rail about a lot is like, adoption as a metric sucks. And the reason why adoption as a metric sucks is because you can mandate adoption, but you can't mandate somebody that loves it and is bought into it and wants to contribute to it and is excited about it. I think that one of the things that is interesting about sessions like the one you conducted is you could have gone down the road of like, hey, look, we built these five components. This is our direction, this is what we're going to do. But instead, you had that feedback and you listened to those folks and you adapted.

And in that moment of listening and adapting, that's what probably is going to drive better metrics and more value to the people that are actually consuming it. And this is where adoption really falls down, right? Is you could have gone with an approach that mandated it, but it might not have been the right thing to actually create value for those teams. And adoption ignores the value side of that equation. And Dan Moles actually speaks about this a lot, too, in the frustrations he has with that metric. And we've actually talked about it together a couple of times. But in a lot of Dan's parlance, what he looks at is everybody makes a catalog of the 50 to 100 things that they do every quarter. And then when you look at that catalog of those things and you look at how a design system is going to change it, you want a bunch of those things to disappear and you want a bunch of new ones to show up. Instead of saying, I can do now 50 things in 90 days, I can do 65 things in 90 days, and ten of them are things I've never been able

to do before.

That's what that real core metric of value for a design system is all about.

Ashley Willard (Apiture):

If it's a dictatorship, where we're just saying, go do this thing. I think that's been our company's culture in the past, and that's exactly what we're trying to get away from. Is this just waterfall? Hey, engineers, you're at the end of all these other things that happened. By the time it gets to you, you just have to do the thing.

Chris Strahl:

Go into your code minds. Engineers.

Ashley Willard (Apiture):

Yeah, yeah. You know, and I can only imagine that that's not very, you know, compelling as your job to be told that kind of thing. And so them having been involved earlier and having influenced the decisions we've made and the results that have come from them is much more compelling for them, I think.

Chris Strahl:

Yeah, I think that feeling like you were a part of influencing those technology choices, those decisions, those structures that exist within the system, make you a lot more apt to want to use it. And I think that there's a lot of advantages you can talk about, too, that aren't necessarily about, please use this. Yeah. Like, if you use it, it will be kind of the thing that you're trying to solve. And, like, what are some of those things that you come up with in that kind of conversation, in that feedback session? Like, if you use it, x will happen. Things like this before and now it looks like this after. What kind of conversations do you have around that to help secure that buy in?

Ashley Willard (Apiture):

I always talk about it and think of it as, you know, it's all part of a whole. It's all building blocks that are going to get us to the future we want to get to as an organization, as a company. The infrastructure is in our way right now, in a way, the messy infrastructure, we have some legacy code in our system that we've just never been able to prioritize getting rid of. So it's always kind of like, oh, we can't do that because these old Perl pages don't support that, that kind of thing. Right. And we still have, you know, some of that in some of our applications, because, again, you know, we live in this world where we have newer things and older things. And as everyone who builds software knows, you often find yourself in a position of just frankensteining. Right? Like building something on top of something on top of something for years.

And that's how you end up with a messy code base. And I say all of these things as if I'm an engineer, but I will be the first to say I'm not really that technical. So I. Yeah, but the thing I.

Chris Strahl:

Like here is that you're painting a future vision. And that future vision is like, yeah, we're all moving towards this, I think. Skim on the more technical side. Do you talk about the practical ways that life is going to be better? Like, hey, you won't ever have to relitigate, like, your accessibility options for a button again type conversation. Is that the conversation you're having, or is it much more about like, hey, we're going to go fix all this technical debt that everybody hates? What does that look like?

Skim (Apiture):

Yeah, we're starting those conversations of things that are going to be a little better. Like, when I first started, I sat in on a lot of calls with designers and engineers and kind of got a sense of, like, what that process was like, of like, you know, doing the design, like, desk check, as we call it. And what I've come to learn, and I guess folks who've been there longer know, is that a lot of things change. Like, a lot of requirements change like they do in any company. There's always like that scope creep. Just sort of like, changing needs something, changes at the last minute and it's not necessarily reflected. Did all the things and that's been a big pain point for us and really everywhere that I've worked. So having an actual design system, having those components means that there's not going to be as much of.

Okay, why does your button not have rounded corners? It's already confirmed by ux. They've already thought through it. We have our design tokens, we have specific spacing, so we're not going to get the. I remember in my early days, the eleven pixels of padding is never right. Just, it's never. No designer wants that.

Chris Strahl:

All our padding is factors of prime numbers.

Ashley Willard (Apiture):

Yeah.

Skim (Apiture):

So, yeah, I'm going to get on that kind of soapbox, especially next week, to talk about, okay, like, yeah, this is frustrating for you to have to, like, kind of roll with these changes, but we're going to end up eliminating a lot of those changes that are happening to last minute because you have, like, this already made toolset and you kind of know you can rely on what's there already. And if something is amiss, that's when we have to really also push the feedback loop of, okay, if you're getting something that doesn't look like this reach out to us. But since we have such a strong partnership with the UX design team, that probably isn't going to happen from the UX design team. That might be, like, a business stakeholder being like, I don't know if they'll ever make the buttons some fluorescent pink, but we have a little more control over that. So if things are changing in the business needs, they can't really change as much in the UX without it almost forcing a conversation, which I think will make things a lot less frustrating for the engineers and probably also for the UX designers.

Chris Strahl:

Yeah, I mean, there's always all these, like, little skeletons in the closet, or I like to refer to it as empty whiskey bottles under people's desks. You know, little deviations that have been, in the moment, necessary, but have otherwise been things that you try to, like, hide a little bit. They creep in. Right. And it's almost inevitable as we're talking about requirements, changes, and everything else like that, where suddenly there's a time pressure on a thing, and the easiest way to get it done is to go outside of the way it should be done. And I think the great thing about a design system is it should be the path of least resistance. It should be the easy way to get it done. And in the cases where it's not like you can at least have that conversation, and there's a lot of that that happens in the product space that I think is kind of interesting.

That's even away from, like, design and engineering and more directly related to product owners. It's like, product owners have, like, one of the worst jobs ever. I mean, I say this. Being a former product owner where, like, your job of that harmonization and that decision making, when it's crunch time around which direction you go, that's an awful place to be in very, very frequently. I'm pretty sure the liquor store down the street for me went out of business when I moved. And, like, being able to solve that pain for those people and to really kind of ferret out those skeletons, I think that's an apt goal for one of these systems.

Skim (Apiture):

Yeah, I mean, working with our pilot team, when we released the first couple components for them to, like, kind of see how they felt about the new or, like, the updated UX version of it, I feel like there was a lot of pushback of, like, are we sure this is, like, okay, we have three different designs for this and what we've been looking at.

Chris Strahl:

People get anxious. Right.

Skim (Apiture):

It's only one is it okay that it's like all notifications are yellow and all errors are red. And use this thing, is that okay? When we showed them that it's like, that is indeed preferred by UX, like ran it through many of the UX designers and it just kind of felt like, I don't know if they fully felt the relief there yet. They're like, oh gosh, what if all of these need to be slightly different? And I'm trying to reassure like, no, they really shouldn't be. If you have a notification one place, it should function similarly to one somewhere else. Or especially like the notification alert banners as you get into discussions of like, oh, what's a toast and what's an alert? That whole can of worms.

Chris Strahl:

I think it's even amazing though, is even a relatively small organization. You all are having these conversations now, right? Because this does just represent good ux. It's about the idea of like, hey look, it's not something that is meta or Nike or something like that, but it is this interesting conversation about how do you spread good ux across a company? And that's just about

building better product. And I think that it's interesting to see that while it's a bit of a microcosm within the ecosystem you all are in, it's a lot of those similar principles playing out in those similar conversations.

Skim (Apiture):

Having worked on a lot of design systems in the past as an engineer, I am very excited about adapture. I work very, very closely with UX designers. We are full on partners. I am kind of helping them with some of their frustrations of things lost in translation or things just looking kind of off. So it's been awesome just being a part of the team that's really going to be the one that wants to drive this and has all this buy in and having this wonderful level of support from our team. Ashley, it's the perfect partnership.

Ashley Willard (Apiture):

It's been a huge pain point for us for a long time that we haven't had that connection and that close bridge between our designers and our developers. So skim was talking about, for example, you know, as things are getting built, depending on which engineer is working on it, they might be somebody who really cares about and is passionate about front end or they might be some full stack person that got thrown at front end and couldn't really give a damn about it. And you get different levels of, you know, successful execution with that as well. And so this will help with that a lot to remove those kinds of errors, let's say, and to support those people who have to do that work but don't care as much about it. But now having people like skip in the organization, I mean, frankly, that is the world we live in, right? There is an array of level of interest, I would say, in Ux among our development organization. And I don't say that to say that anyone's wrong or bad, but we just have not put a lot of focus, like I said earlier, on hiring people because they are interested in front end until very recently. And it's really making a difference for the designers, for sure. They really feel supported and feel like they have engineers who understand and can help to translate these things.

Chris Strahl:

It's funny, right? It's a candid take on something that's pretty universal. But not that many people talk about is that there's just a fundamental, like, you know, disconnect between the idea of what good design is and the practice of actually getting those people in the right places to make that good design happen.

Skim (Apiture):

I definitely had a lot of assumptions earlier in my career where I'm just like, oh, every team will have, like, a UX or front end engineer and, you know, working with more full stack engineers and understanding that, yeah, a lot of the times they don't really care about the UX and, like, that's fine for a lot of the stuff they work on, but, you know, sometimes that thing needs a button or, like, data displayed even though they're like, oh, it's not my world. Once you get them engaged just enough to tell you when something isn't working, that is very, very valuable feedback.

Chris Strahl:

Well, it's awesome to hear you all making the strides that you're making. I love the vision. I love the direction, and I also appreciate that you've chosen Knapsack to help you out along the way. I wanted to say I think that it's awesome to have a take like this and your ability to describe really candidly a lot of the intimate problems that you all face in the way you build product. It's cool because I think that companies large and small face these similar problems, and they're maybe brought into much more acute focus at the scale that you all are operating. But there's plenty of people in really big companies that feel the same way that just don't talk about it like this. And so I really appreciate you all being on, and I appreciate your transparency, your honesty, and the ability to talk about decision making inside of your organization. It's really cool.

Thank you very much.

Ashley Willard (Apiture):

Yeah, no, thank you. We're big fans of Knapsack. And one of the main reasons, I think, is that it helps to bridge that division between designers and engineers. Design can have a little more control over the design system. We can collaborate more closely. It's the first of its kind that I've seen out there, and I'm very excited for the future.

Chris Strahl:

Thank you for the kind words. Really stoked to see what you all are able to accomplish. And so thanks so much for taking the time to chat. This has been really awesome.

Ashley Willard (Apiture):

Yeah, thank you.

Skim (Apiture):

Great talking with you.

Chris Strahl:

This has been the Design System Podcast. I'm your host, Chris Strahl, signing off. Have a great day, everyone. 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 edspod. 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.