EPISODE 1491

[00:00:01] ANNOUNCER: This episode is hosted by Lee Atchison. Lee Atchison is a software architect, author and thought leader on cloud computing and application modernization. His most recent book, *Architecting for Scale*, is an essential resource for technical teams looking to maintain high availability and managed risk in their cloud environments. Lee is the host of his podcast, Modern Digital Business, an engaging and informative podcast produced for people looking to build and grow their digital business with the help of modern applications and processes developed for today's fast-moving business environment. Subscribe at mdb.fm and follow Lee at leeatchison.com.

[INTERVIEW]

[00:00:45] LA: Enterprise-grade authentication is often an essential ingredient to virtually all applications in today's world. However, companies often have a hard time understanding the value of that authentication especially during the early stages of product development. And hardening of an application is often left as an afterthought. Add enterprise-level requirements such as single sign-on and two-factor authentication and other requirements that the once afterthought becomes are now major issues. Delaying launch schedules and new feature launches.

WorkOS is dramatically shortening this development time by providing tools to allow in integrating complex enterprise standards with just a few lines of code, allowing developers to focus on core products and features. Michael Grinich is the CEO of WorkOS and is my guest today.

Hello, Michael, and welcome to Software Engineering Daily.

[00:01:42] MG: Thanks so much for having me, Lee.

[00:01:44] LA: It seems that the bread and butter of your company, the customer, the main customer of your company is the SaaS company that starts out small and grows to the point where they now need to have enterprise capabilities in order to attract larger customers to their

product offerings. If that's the case, talk to me about what it means for a web application to add enterprise-ready features to their application.

[00:02:10] MG: Yeah, absolutely. We typically work with companies who are kind of post-product market fit. Meaning, they've launched their initial product. They've brought it to market. They've started getting some initial users and started growing. But they're ready to really move up market and sell the bigger organizations. And that could be selling a team of, say, a hundred people.

And when you do that, when you actually look to move up market, typically there's an it department or some procurement process you go through. And that's the first time a lot of these companies start experiencing the need to have some kind of enterprise features or becoming enterprise-ready. Sometimes you call this the enterprise chasm. You have to cross this in order to sell into these organizations.

And so, that's the moment where they begin to start having this need. But we've seen companies you know even much later down the line that still need to keep adding these features to their products and it gets more and more complicated and kind of more sophisticated as you grow. That's just the starting point. But we end up working with companies that are much larger as well.

[00:03:06] LA: Yeah, it's an interesting side effect that doing authentication is the first thing. And building a product is easy to do or relatively easy to do. But it's never thought of first. It's always thought of later. And especially adding enterprise capabilities like SSO, etc., it can be really daunting for companies that are selling, have customers want to make a living, but now need to focus on these enterprise features.

Now, SSO is a classic example of an enterprise feature. But what are some other examples of capabilities that companies often have to add that aren't directly SSO, but something that's considered enterprise-ready?

[00:03:48] MG: Yeah. SSO is definitely the first one. If you think about what every app needs, it first of all needs to know who you are in order to use it. And we're talking web apps through the

browser that you're using. The second thing that's really common actually is that all apps today are really multiplayer. If you think about products like Figma, or Google Docs, or Dropbox, or just kind of modern SaaS tools, they're all about teams of people working together in groups.

And so, a very common second feature that companies will need is some way to integrate with the directory systems of those companies and plug into the kind of group management system. And we have a feature around that called directory sync. There's a protocol called SCIM, which is kind of like SAML. Getting the weeds a little bit there. But essentially what it lets you do is provision users and deprovision users and really have your application be aware of the organization and group structure of that company. And that lets you build these like rich multiplayer experiences.

Beyond that, there's a lot of other stuff. There's stuff around auto logging, and access control, and compliance features, and really quite a long list of different capabilities. But the first ones that people really get started with are those pieces around enterprise authentication and related pieces around user management.

[00:04:59] LA: And these take time for companies to build on their own. And that's really where your value comes in.

[00:05:05] MG: Yeah, that's right. And the shape of the work, you can almost think about it as integration work. People often compare WorkOS to something like Twilio or even Plaid, where those platforms allow you to plug in once and integrate with all the different telephony systems out there. Or in the case of Plaid, all the different financial institutions.

WorkOS, you plug into it once. And it's not just giving you single sign-on as a feature. It's also plugging your app into all the different enterprise identity systems like Octa, and OneLogin, and Active Directory. And in the case of directory sync, it's plugging it into Workday, and BambooHR, and Rippling, and all these HR systems.

WorkOS acts as this sort of aggregation layer across all these different fragmented services. And the effect that it has for a developer is you just plug it in once and you're kind of done. You don't have to think about it anymore. And it just works out of the box. And developers can go

build the features that they probably are more excited to build, which are the unique features for actually their product or their app that they're creating.

[00:06:02] LA: You don't see yourself as a better Octa or a better OneLogin. You really see yourself as an integrator of those tools. And so, customers still want Octa, or they still want Auth0, or whatever capability they're looking for, or BambooHR for their directory. They don't have to change to use you. They continue to use those existing tools. But you provide that isolation between the application and those tools.

[00:06:29] MG: We're like the other piece of the puzzle. We don't replace Octa, or replace Workday, or BambooHR, or these systems. What we really do is help developers plug into all of those different environments and ecosystems. And WorkOS, we don't really sell the IT departments. IT departments are typically the ones buying Octa. We sell to the developers building products having to plug into them. It's kind of like saying is Stripe competitive with Visa? It's like, "Well, not really actually." That's one of their tightest integrations. It's just a different side of the puzzle piece.

[00:07:01] LA: You wouldn't call yourself an authentication company. You'd call yourself a tool set company?

[00:07:05] MG: Yeah, yeah, exactly.

[00:07:05] LA: How does someone actually integrate with WorkOS? What are the requirements to use WorkOS? You said, first of all, web app versus standalone app. I get that. But what frameworks do you support? What languages do you support? What's involved in any integration without getting into too much of the weeds anyway?

[00:07:24] MG: Yeah, we support a ton of different stuff. Pretty much however you've built your app, whether it's a legacy application, or a brand-new modern serverless node.js app or something on the bleeding edge, we support it all. We make it really, really easy to plug in. Essentially, the interface that a developer has we make really really simple with just like a modern connection.

In the case of SSO and studying needing to learn SAML 2.0 and OpenID Connect and all these legacy protocols, it's just OAuth. If you've ever built like a Facebook login, where it's just a few lines of code, that's the integration with WorkOS.

We've taken essentially all this legacy complex interfaces and complex old really archaic protocols and put a layer on top that matches the way modern developers think about building products. A modern developer who's been like building a web app, if you just go check out the WorkOS docs, it's just going to feel like you're right at home. And we have SDKs for everything under the sun, from like Python, to Ruby, to Node, to PHP, Elixir, it's got it all.

[00:08:27] LA: All the modern frameworks, all the modern languages. Your history, you started out as a Rails app. Is that correct? Is that what your early history was, was in Rails? I know you use Heroku. And there was a tight integration with Rails back in the olden days anyway, yeah.

[00:08:41] MG: WorkOS is actually not a Rails app. We do run a bunch of stuff on Heroku or historically have. We are actually a node.js app and we use TypeScript across the whole company, across the whole stack. And so, early on, three, four years ago, decided that we wanted to have one language for the whole company. And so, engineers are able actually really to work full stack with one tool set in one system. We run a node on the backend. And then we're running in the browser in the frontend.

[00:09:07] LA: Cool. Cool. Let's talk a little bit about multi-factor authentication. And there's been concerns in recent years about the security of using SMS, getting pages on your phone as a platform for multi-factor authentication. You log in, it says, "Enter the code that's sent to your phone." You type that code in and you're in.

Early on, that was considered a really good valuable valid way of doing multi-factor authentication. But it's kind of starting to fall out of favor when you start hearing about phone spoofing and other techniques that really have made it less secure. Certainly, things like one-time passcodes, OTPs, they seem like a decent replacement for using SMS messages. But yet you still see all these applications that focus on SMS as the way of doing multi-factor. First of all, do you know why that is? Or where do you think the industry actually is heading in this respect?

[00:10:08] MG: You know, it's a great question. SMS multi-factor authentication has been really successful. If you go back years, just in terms of preventing phishing attacks and spoofing and other issues, it's been a huge win for just like consumer security globally and consumer products, whether that's logging into Gmail or buying something on eBay. Being able to have a second factor of authentication has really helped secure the web.

I don't want to like kind of talk too down on something like that, because it's done really well. Multifactor authentication, really, if you break it down into its pieces, really what it means is there's two pieces to it at least. Something you know, which is in most cases your password. And then something you have, which is a device.

And SMS is really that thing that you have. And so, you enroll a device, typically a cellphone. You get that SMS that gives you a code and you can prove that you have that device, right? And you combine those two things together. Those are the two factors. Boom! You got multifactor auth.

SMS as a transport mechanism is not super secure. There's a lot of different ways that phone numbers get transferred. The actual SMS protocol and sending actual text messages is like not the most secure thing in the world. It started off as a hack years ago on top of this like uptime telemetry thing that the telecoms have. It never really was built to be a secure transport mechanism.

And there's unfortunately been a lot of hacks where people have taken over SMS transport and hacked into people's Bitcoin wallets and things like that. And what that's really trying to do at that point is prove that you have something. We're using SMS to contact a device that you have your phone to prove that you have that.

Today, there's many other options for that. People have seen the TOTP, time-based one-time passcode, auth flows. Most people have seen that with that QR code that you scan and then you have an app that gives you a six-digit code. That way, there's no transport ever being sent. There's no SMS being sent. Your device just generates codes. Every 60 seconds, it'll generate a new one. And so, you can prove that you have that physical device itself.

And there's software versions of this. Like, 1Password has a virtual TOTP device. And there's Google Authenticator and all these other products. And so, I think what we're seeing is like a change from needing to use SMS as this transport mechanism to verify the device. To now having these hardware or virtual-based devices that prove that you have something.

And the last place that we're going with this, which is really exciting, is the world of WebAuthn. WebAuthn is essentially a browser protocol that allows you to communicate with hardware security devices like a YubiKey, or really any device that's connects over to FIDO, kind of alliance of different products. This is like touch ID or like face ID on a phone. Apple's been pushing forward a bunch of stuff around this as well.

And so, it's just really encouraging to see like a lot of new types of devices available to do multifactor auth. And I think with these things, especially as it becomes easier for consumers to set it up, we're going to see people moving to it for more secure scenarios. I think even today, you can set up like your Facebook login with a YubiKey, or your Twitter login with a YubiKey.

But at the end of the day, having multifactor auth is still so powerful, that if all you can add is an SMS passcode, it's really better than nothing. I always encourage people to set it up. WorkOS, we have a product around multifactor auth. We support both SMS and also TOTP and other protocols. It's moving in the right direction, but there's still a lot more to do.

[00:13:29] LA: Yeah. And by the way, I want to reiterate what you said. SMS, even if it isn't perfectly secure, it's a hundred thousand times better than not using anything. I 100% agree with you. I have lots of websites that I am connected to that I use SMS for my third-party authentication. I moved them all to the time-based, the TOTP, when I'm able to. But not every site supports it yet. That's why you still see a lot of SMS authentication.

I'm just wondering from a company standpoint, do you see companies converting as well or do you see them – Every consumer has a cellphone. Not every consumer has the Google Authentication app, or want to get another app, or want to deal with that. Do you see that as part of the reason why especially consumer-based companies are not moving to the OTP model?

[00:14:21] MG: I think it's part of it. And part of it is just the threat model and what's subject to this access. Every consumer doesn't have like a YubiKey, a hardware security key that can generate cryptographic codes. That's pretty unusual. I mean, probably your friends and my friends have them. But most people in the world don't have these.

And so, when you're just trying to authenticate, say, I don't know, your Amazon account, where the worst case is someone could log in and buy some shoes and it gets flipped for fraud. That's one thing. And so, probably SMS, two factor auth in that scenario is okay.

But in a scenario where, for example, you're authenticating access to a production database, you definitely want to put that behind a different type of authentication. And I think that's why you see companies adopting hardware security systems, YubiKeys, other types of like FICO compliant keys in order to have that really increased level of authentication where it can't be spoofed.

The downside of that stuff is it's a little bit harder to recover. If you lose that thing or it's destroyed, there's no real way to have account recovery. But also, in the case of teams and companies, you usually have some type of administrator who can reset these things. And so, that's what you'll see is the identity systems, whether it's Octa, or Azure AD, or other identity providers for enterprise organizations, there's a lot of sophisticated tools that give IT admins the ability to reset or push new keys into these products. It's more of kind of like a managed identity service. Whereas on the consumer side for individuals, you got to kind of keep your own security. You got to take care of it yourself. And that can be a little bit more challenging for people.

[00:15:51] LA: Valid point. I always tell people that I live and die by 1Password. That is my favorite tool of all time.

[00:15:58] MG: I love 1Password too. Huge fan. Yeah, they've done a great job.

[00:16:01] LA: Yeah. It generates my passwords for me. I do OTP built-in. And being able to cloud sync that across multiple devices so I don't have to worry about if my cellphone changes, I

have to go through and reset all my OTP passwords everywhere. It just all works. And I really love that.

[00:16:17] MG: I mean, it's such an incredible achievement that they took something so complex and have made it so simple and easy to use. I think that's – I totally agree. 1Password is a great feat of product design and engineering. Definitely recommend people use it. We use it too.

[00:16:28] LA: Absolutely. Great. Great. Now, you mentioned WebAuthn. Now, is that related to this new trend you hear nowadays about passwordless access and systems that have not a matter of tools like 1Password that store the password for you? But systems that remove the password and use other authentication mechanisms usually in browser cookies or other things like that to do that. Is that part of WebAuthn? Or are those two different things?

[00:16:56] MG: I think it's sort of part of it. That's a really good question. The word passwordless kind of doesn't – It means everything and nothing at the same time. When I've seen most people using it, they're often using it to describe the authentication flow of going through and putting in an email address and getting like a magic link sign-in. Or sometimes getting a code that gets texted to them for their SMS and they sign it with that.

And so, I think it was like Snapchat originally didn't have passwords. You would just put in your phone number. It would text you a code. And then you would sign in. And so, it worked really well for this like consumer login experience.

Similarly, things like Magic Link, which is this experience where you put in your email address, it emails you a link. You click it. That logs you in. That creates the session. And you're into the web app. That's really good for things like e-commerce. Or if you think of – I don't know. Like, you bought a book from a publisher years ago. You can't remember your account. You don't want to have to create a new password for that. Just email me the link.

It's a lot less popular in the enterprise and within business context. And it sort of makes sense if you think about it. If you think about what's the thing that all the IT admins tell you about links

and emails, they're like don't click the link. Don't click the weird link that's come through an email. Log in through another mechanism.

And so, I think Magic Link and this idea of passwordless login through enterprise systems actually goes against what IT admins are looking for in terms of managed security. In those scenarios, they want to have it centralized behind a single identity. It really is a single password. And that's why people adopt products like Octa, OneLogin, Ping Identity. There's a lot of different identity solutions. Even Google login. Think about like the sign in with Google button. In a way, that's sort of like a password login experience, passwordless, because you only have one password you're putting in. I think that term is kind of inflated. But it seems to be more popular for like consumer type of products, b2c products, versus more like b2b products that you might use in the workplace.

[00:18:50] LA: Right, right, right. Yeah, I think I've seen it recently used in some techniques involving long-term cookie storage. And essentially, it's much like the logged in session. But this is for logging credentials as well. And it's a private key mechanism from what I understand. But I don't remember all the details. But I wasn't sure if that was related to WebAuthn or not.

[00:19:11] MG: I think they're separate. You could do it through WebAuthn with a hardware security key. And there's also like browser certificates you can do authentication through, like Kerberos and other systems. But most people are talking about Magic Link, I think, when they say passwordless.

[00:19:25] LA: Probably. Yeah, yeah. We've talked about authentication. And a lot of people I talk to confuse authentication and authorization. And to our listeners, when I talk about authentication, that's the ability to prove to me that you are who you say you are. That's what authentication is. And authorization is do you – You person who are now authenticated, do you have permission to do this particular task in this app or not?

WorkOS provides really solid authentication capabilities through your partnerships and the security requirements around them. And that's where things like OTP and SMS fit into all of that. But what about authorization? Do you have any systems that help you with the authorization

side of the puzzle? Or is that a direction you're headed? Or do you already have capabilities there?

[00:20:14] MG: This is such a great question. I love talking about this stuff. Yeah, authentication versus authorization. People confuse it all the time when I talk to them as well. You'll sometimes see this abbreviated as people writing authn to being authentication, versus authz to be about authorization.

And authorization is really like do you have access to something? It gets even more confusing, because the way that most people experience this from a consumer side is it happens at the same time. If you think about like, say, logging into an app with Facebook connect, your Facebook identity. Say, you're going to log in to like Airbnb with your Facebook account. First of all, it does authentication, where Airbnb can say you are who you say you are. You are Lee. Okay, this is your Airbnb account.

But then also in that same flow, Facebook will say, "Do you want Airbnb to have access to your profile photo? Your friends list? Yow your email address?" And in that scenario, you're actually doing an authorization grant. You're giving them an entitlement to access this type of data. And it's very common that both happen at the same time in consumer products. But it's worth saying that there are two different things. You can authenticate someone but not authorize access to different things.

Today, WorkOS mostly focuses on authentication, enterprise SSO authentication, where you're connecting through something like Octa. You're signing in. Well, really, what we're allowing developers do is to add enterprise auth for their customers to sign in. But then who gets access to what? That's sort of left as an exercise to the reader. Because we don't know – Say, in the case of a company like Webflow, one of our customers, a user might sign in. And we don't know if they should have editing permissions or not. That group is actually – That permission set is actually set inside of Webflow.

One thing people do actually do with WorkOS is they will pull membership from different groups and then assign permissions to those groups and assign roles to those groups. And so, you might say like, "Hey, if you're inside –" Like, Vercel is another one of our customers. Maybe if

you're using Vercel and you're connected in this right way with groups and permissions, people in the engineering group can have access to push code and change stuff. But maybe everyone else can only view stuff. You can only view preview deploys.

And so, we give people that kind of primitives for actually building authorization into their app and the data connectors. But today, we don't provide authorization layer on top. And there's a bunch of different open source projects and different ways people have been exploring different ways to do authorization that are out there. Kind of every week on Hacker News, there's almost another one. You can essentially use WorkOS with any of these things. Y ou can plug it in and it works pretty well.

[00:22:46] LA: Cool, cool. Yeah. Yeah, interesting that you – If you want to get into a philosophical discussion, which we really shouldn't be here, but group capabilities authentication or authorization. But we don't need to spend any time on that.

[00:22:59] MG: You kind of need to do both together. Yeah, it's like peanut butter and jelly. They taste pretty good together.

[00:23:04] LA: Yeah. Yeah, exactly. Exactly the case. Where is the world of security, and authentication, and authorization, and all the things that go with that, where is that headed in the industry?

[00:23:17] MG: That's a great question. Well, I think for the companies we work with, these startups growing up market, we're trying to help them become enterprise-ready and grow. We're really trying to help them accelerate their roadmap. And so, a lot of what we provide is not like novel technology in the sense of the bleeding edge of where the world is going. We're helping them just get the features into their app that all the people in IT need. And these are the same features that like Slack built inside of Slack Enterprise, and Dropbox builds in Dropbox Enterprise. And Asana has an Asana Enterprise. We're helping companies like Webflow, and Vercel, and Airbase and companies like that do it. I would say we build like infrastructure that's somewhat already needed versus being on the bleeding edge. I'm personally not working on this stuff.

I think probably what we're going to see is there's really two dimensions of this. One, in enterprise authentication, companies are just using more and more apps constantly. If you look at the rise of number of SaaS products that organization uses, it has just exploded in the last two or three years, especially due to Covid. The tools that people use in order to like collaborate, communicate in a fully remote or hybrid remote setting, you just need more tools. And this has put additional burden on IT teams and coordination around not just buying these, but setting them up to be secure.

And so, that's why Octa has done really well, as a bunch of other identity products to help secure it. But we're seeing the increase in provisioning and deep provisioning and other capabilities needed in different compliance features. And I think that's going to keep growing to be the case in big enterprise and kind of medium enterprise companies.

On the consumer side, it's a little bit different. People are spending more time online, buying more stuff online. Our online identities are more and more important to keep secure. And there's more threats out there in terms of, "I know, if there's more stuff about you online, you can do more stuff about you online." You need to secure that.

And so, that's why we're seeing companies like Google and Apple like collaborate on new ways to provide authentication fundamentals and primitives. Like, face ID was a big thing for this. And Apple literally just announced in the last few weeks some new browser APIs that are coming out in iOS that seem pretty powerful. And some new passwordless authentication stuff. But like I said, I'm sort of less on the consumer side and more in the infrastructure for growing company side.

[00:25:32] LA: B2b side. Yeah, that makes sense.

[00:25:34] MG: There's probably a bunch of stuff in crypto world too. Honestly, I don't pay any attention to that. You have to have a different guest to talk about that.

[00:25:42] LA: It's a smart man there. Yeah.

[00:25:42] MG: Yeah. Someone asked me the other day, "How do you think about Web3 identity?" And I was like, "I don't even know where to start."

[00:25:50] LA: That's actually a great conversation. I've had that conversation with a couple people. And I'd love to have your thoughts on that. But that's a whole different podcast. Next time. Yeah.

[00:25:59] MG: I mean, I'm dealing with like SAML, which is like a protocol designed like 20 years ago. It's like XML, XML canonicalization. It's probably the most archaic thing you can touch. It's a total opposite of blockchain.

[00:26:11] LA: I would have guessed more like 30 years ago. I'm not sure exactly when. But, yeah. It's an ancient protocol. But it's what's in use today.

Just looking at your company name, WorkOS. I'll be honest, when I first saw that and I didn't know anything about you, and I looked you up, it said authentication company. It took me a while to make that correlation WorkOS is an authentication company. And calling you an authentication company is not fair to what you're currently doing. I get that.

But when I'm first looking at you and trying to figure out what you do, I saw that connection. But to me, it's really clear that you have a lot more in mind that you want to accomplish than just integrating with authentication providers. You've been on this podcast before, the Software Engineering Daily podcast, and you talked about wanting to be the modern age equivalent of Microsoft Windows was in the 90s. And that sounds like you really want to be an OS for web apps. Is that an accurate depiction? And are you still working towards that goal?

[00:27:15] MG: Yeah, I think you need to unpack the OS word a little bit, because there's a little nuance there. And I've had people say like, "You guys aren't building an OS." Like, expecting that we're building a new Linux kernel or something like that. And that's true. We're not building an OS in that sense.

But in the sense that we're building components that are needed for every application and are kind of undifferentiated, that is the role of the operating system. It's these underlying pieces that every app connects to that power the core experience of every product.

And today, if you're building an app using web technology, using node and deploying it to the browser using AWS, every app you pretty much have to build from scratch. You can get open source things. But it's just this huge labor to create new experiences. And it's way harder than it used to be.

Back when people could build on Windows, there was such a rich development environment. You could create these products really fast. It's super hard to do that on the web today. And I think we are working towards it. It's just kind of brick by brick. How do we build this? And how do we build these pieces?

A few months ago we acquired a company called Modules, which joined WorkOS. And one of the products that Modules built, this open source project, it's called Radix. And radix is a really powerful headless UI design system builder, essentially. It's a component library for building apps.

If you're building an app – And this is used by a ton of companies today. If you're building an app and you need buttons, and drop-downs, and menu items, navigation, the previous world is you kind of had to build this stuff from scratch or cobble together a few open source pieces. Whereas Radix gives you this all out of the box. It's this whole really high-level, expressive, powerful component system that allows any company to have a full design system really quickly.

And without something like this, companies are building design systems teams. Like, Stripe has a whole design systems team. Airbnb has one. Asana has one. And so, I think we are trying to move closer to the idea of just giving developers these building blocks. Some of these will commercialize like in the sense of our SSO authentication product. That makes a lot of sense to commercialize. It's a really kind of slam dunk in terms of the value it has and people paying us.

But for other pieces like Radix, which really we want people to be able to use when they get started, it's something we probably aren't going to commercialize. It's just a component. But it's

really all under that same umbrella of just helping developers build things faster and take their ideas and turn them into products and experiences really, really quickly. And that's really our whole mission at WorkOS. And that's going to be a thread that carries us through, future products that we build, and future services that we give to developers.

[00:29:43] LA: So, more great things to come. You are focused on more than just authentication. You've got a lot of tools available for developers.

[00:29:51] MG: Yeah, exactly. And I kind of joke sometimes that the stuff we've built so far around SSO, and directory sync, and these underlying pieces, in the Microsoft analogy, it's kind of like DOS. If you remember Microsoft DOS, if you looked at it and you're like, "Oh, I don't know what's really here." But DOS laid the foundation for them building Windows 3.1, and then Windows 95, which was extremely powerful. And a lot of that stuff ushered in things like desktop publishing. It just takes a while to get going. And so, we're plotting along and staying focused on these components and really just trying to help developers grow their companies and grow their products at the same time.

[00:30:28] LA: I love that analogy with DOS to Windows 95. That's exactly the way this feels like to me. That's a great analogy.

Thank you, Michael. Is there anything else you want to tell the listeners about WorkOS?

[00:30:43] MG: I guess if you're looking for these enterprise features, come sign up. I mean, you don't really need to talk to anyone to WorkOS to use this stuff. We're like a company built for developers. You can just sign up and put in your credit card and plug in and get started.

We spend a lot of time on our documentation and getting started materials. It's really, really easy to do it. Most people figure it out in a day or two. And then I would be remiss if I didn't plug that we're also hiring. We are growing our team. We have a ton of stuff to build. We're like an engineering-led and engineering-focused company.

So, anyone listening that loves developer products, and developer tools, and just obsessing over developer experience, please DM me on Twitter or reach out to us. We are many, many kindred spirits just like that.

[00:31:21] LA: Are you looking in any particular geographic area? Or remote is fine?

[00:31:25] MG: We are a fully remote team, actually since even before the pandemic. We have people in North America, South America, Europe, all over. Just please reach out. If you have Internet and you got a laptop.

[00:31:38] LA: Well, great. Thank you, Michael. I really appreciate your time today talking to us. This is Michael Grinich, CEO of WorkOS. And thank you for joining me in Software Engineering Daily.

[00:31:48] MG: My pleasure. Thanks for having me, Lee.

[END]