EPISODE 331

[INTRODUCTION]

[0:00:00.3] JM: Software Engineering Daily examines the world through the lens of software engineering. In most episodes, an expert in a particular topic joins the show as a guest, and we go into deep technical detail. Occasionally, we like to do episodes where we survey a collection of topics. In today's topic roundtable, Caleb Meredith and Courtland Allen join me for a discussion of several questions. Would it make sense for Facebook to build an operating system? Does online advertising work? How can you work productively on an engineering company with your brother as a cofounder? We also discuss many other questions.

Courtland is the founder of indiehackers.com, which was recently acquired by Stripe, and Caleb Meredith is the lead JavaScript correspondent of Software Engineering Daily. This was a blast talking to both of them, and we plan to do more roundtable episodes in the future. We have been hearing feedback from people who like these things, and we'd like to get your feedback, too. Please, fill out the Software Engineering Daily listener survey. It's available on softwareengineeringdaily.com/survey.

Also, Software Engineering Daily is having our third meet-up, which is Wednesday, May 3rd at Galvanized in San Francisco. The theme of this meet-up is fraud and risk in software. We're going to have great food, web-engaging speakers, a friendly intellectual atmosphere. You could find more at softwareengineeringdaily.com/meet-up. Please, if you're going to attend, please sign up now, so we know how much food to order.

Now, let's go on with this episode.

[SPONSOR MESSAGE]

[0:01:46.3] JM: For more than 30 years, DNS has been one of the fundamental protocols of the internet. Yet, despite its accepted importance, it has never quite gotten the due that it deserves. Today's dynamic applications, hybrid clouds and volatile internet, demand that you rethink the strategic value and importance of your DNS choices.

Oracle Dyn provides DNS that is as dynamic and intelligent as your applications. Dyn DNS gets your users to the right cloud service, the right CDN, or the right datacenter using intelligent response to steer traffic based on business policies as well as real time internet conditions, like the security and the performance of the network path.

Dyn maps all internet pathways every 24 seconds via more than 500 million traceroutes. This is the equivalent of seven light years of distance, or 1.7 billion times around the circumference of the earth. With over 10 years of experience supporting the likes of Netflix, Twitter, Zappos, Etsy, and Salesforce, Dyn can scale to meet the demand of the largest web applications.

Get started with a free 30-day trial for your application by going to dyn.com/sedaily. After the free trial, Dyn's developer plans start at just $7 a month for world-class DNS. Rethink DNS, go to dyn.com/sedaily to learn more and get your free trial of Dyn DNS.

[INTERVIEW]

[0:03:45.7] JM: Courtland Allen runs Indie Hackers, and Caleb Meredith is a lead JavaScript correspondent with Software Engineering Daily. Guys, welcome to Software Engineering Daily.

[0:03:54.9] CA: Thank you.

[0:03:55.7] CM: Thanks for having us.

[0:03:56.8] JM: Today, we're going to be doing a topic Roundtable. We're going to have a host of things that we're talking about. There'll be a bit of a free willing conversation, and we're going to touch on various themes in computer science, themes in the news today, themes in software engineering and business. I've got a bunch of stuff, but let's start with Facebook.

Caleb, you recently did a show about the Future of React Native. You also did a show recently about Web Assembly. There is a lot of exciting changes going on in terms of the web. When you have fundamental technological shifts like this, it creates opportunities for market disruption.

The market disruption that I think of Facebook as being most well-poised to attack right now is perhaps the mobile operating system market.

If Facebook wanted to build a mobile operating system today, what would the text stack for that mobile operating system be?

[0:05:01.3] CM: You were the first one to introduce this idea of a Facebook operating system to me. Of course, as a regular user of React and React Native, this idea is incredibly attractive. Facebook is working up and down the entire stack. At the lowest level, you'd probably have something like a C++ write where you could run or a V8, or a JavaScript engine to then run React Native on top.

You might sneak in some OCaml, or reasons thought that they have been working in there to get some more native performance in nicer APIs. On top of that, you'd have React Native, you'd have React, maybe, sprinkle in some GraphQL, then basically, merge all of Facebook's open-source technical properties to create an operating system, which would be incredibly fun to develop and especially for the community they've built.

[0:05:46.5] JM: Do you think it would be Linux-based?

[0:05:48.0] CM: I'm not sure about that. It definitely could be, but Facebook has enough engineers where if they wanted to do something that was custom and unique, they definitely could.

[0:05:56.0] JM: Which is, I think, what Google is doing. I think Google is building an operating system from scratch.

[0:06:00.6] CM: Mm-hmm, yeah.

[0:06:01.6] JM: I don't know if you guys have seen that. I can't remember what the name of it is. Courtland, from a business point of view, would the ROI of making a smart phone from Facebook's point of view, would that make sense today?

[0:06:14.9] CA: Yeah, that's a good question. I feel like, from my perspective, I have very little expertise to bring to this answer, so I'm just completely guessing. I think the mobile space is obviously extremely lucrative. You could be successful there. If you can't make a dent, then I don't think it's worth the resources.

For a company like Facebook who's got so much to spend, they might as well try. They probably aren't going to lose very much. Based on the text stack that Caleb just described, I know that I personally would love to develop for a JavaScript-based operating system.

[0:06:48.6] JM: That's part of the appeal to me, is you've got disenfranchised developers in the Apple ecosystem and in the android ecosystem. I feel the downside is really kept. You look at Amazon with the Fire Phone. Was it a really a waste? They spend a billion dollars. They get a better understanding of the hardware supply chain. There was probably some synergies with the Echo team. The Echo had not been released at that time, so it probably accelerated the development of Echo, which has obviously become a smash hit. It seems like there is almost no downside to this.

[0:07:25.4] CM: A billion dollars is a downside; and the brand, the brand impact, and the brand equity you lose.

[0:07:31.8] JM: Sure. Maybe, you lose a little brand equity, but it seems like people trust the Echo just fine. If we said, "Okay, Amazon spent a billion dollars to accelerate the development of Eco by a miniscule amount," I think it's pretty easy to see that the Eco ecosystem is multibillion dollar industry. If we think of that one billion dollars is a write-off towards development of the Echo, pretty good investment.

[0:07:31.8] CM: Yeah.

[0:08:05.1] JM: Anyway. Okay. Let's talk about alternatives. Facebook, obviously, is getting into hardware. I don't know if you guys saw the article recently that talked about building eight at Facebook, but it's their super-secret labs. It's like a Facebook equivalent of Google X, and they're working on hardware, they're working on ecommerce stuff.

It's clear that Facebook is going to get into hardware. Why wouldn't they? Of course, the lesson from Apple is there is something to be said for last mover advantage. Maybe, Facebook is waiting for somebody else to release the first augmented reality product, and then Facebook can wait a little bit and release something that's incrementally better. The first person who comes out with augmented reality looks like a fool, they look like an over-aggressive first mover, and Facebook gets to look like, in comparison, a shining example.

What are the pros and cons for Facebook? I feel like augmented reality is something that's there. Everybody's got their little augmented reality stuff in the lab somewhere. They could release it aggressively, or they could wait a bit. What are the pros and cons of being the first mover in the augmented reality space?

[0:09:15.2] CA: I think there're a lot of lessons to be learned from bigger companies hyping up a product and launching it to underwhelming response. Google Glass comes to mind, where there's a lot of hyper on Google Glass and to a degree that it's hard to be a bigger company and to invest a massive amount of resources into a product and not hype it up.

You either go all or nothing. You would believe in it, or you don't believe in it, which I don't think you have to take that route, but a lot of companies do. I think probably better to not be the first to be really augmented in reality space, because there hasn't ever been, to my knowledge, a popular augmented reality product.

[0:09:50.1] JM: Snap Spectacles?

[0:09:52.9] CA: How well are those doing?

[0:09:54.3] CM: Yeah, that's more like a hardware product, too. The real augmented reality Snapchat stuff, the filter is more so than Spectacles. I'd say getting the first move or advantage in AR would be super valuable for Facebook. How far off do you think the tech companies are from really releasing an AR product?

[0:10:10.9] JM: It's so hard to tell. There's some stuff going on and magically — I don't know. It's so hard to tell.

[0:10:17.7] CM: Out of all of them, I feel like Facebook and Google are the two with the biggest advantage, because Google has the Google Glass, and then Facebook has a lot of money and the social network. The data Facebook brings to the table from facebook.com, and you can walk around and pop up over everyone's sends either account. That's super valuable.

[0:10:38.0] JM: Certainly. I think this is the mistake people make in counting out Apple. People always say, "Apple's got all these cash. Where are they investing it in?" They're investing it in augmented reality. They're just not talking about it.

[0:10:49.6] CM: Yeah, but I can totally see Apple wanting to take the last move or step. Whatever they build, they're going to take forever to build.

[0:10:57.5] JM: It's what they did with the iPhone.

[0:10:58.8] CM: In the meantime, Facebook could come out, drop their AR product, capture market, and then maybe Apple comes along after that.

[0:11:06.9] CA: I'm not so sure that the first move or advantage will allow them to have any permanent market capture though if you have a product that's really not that good. There's a tricky balance there. If you're really trying to build a motor on your products, you need some proprietary tech advantage or some network effects, which obviously, Facebook is well-poised to have.

If the product isn't there, and people don't like it that much — Take a look at the Oculus, which is not augmented reality, but how many people are using an Oculus? How many people do you know who have an Oculus? What's to stop? The VR explosion hasn't happened yet to the degree where any company can get enough market share to say that they've got a foot hold that someone else can't later catch up and surpass them with a totally new product that's just better. I think it's similar to mobile phones.

[0:11:51.0] CM: Why do people stay on Facebook? Theoretically, it's possible to take React and all of the open-source Facebook's building and build a "Facebook clone". People stay on Facebook because all their data is there and all their friends are there. If Facebook can have a really tight integration with their data and their AR product, even if the AR product is sub-optimal, then people are still locked in to the data there and to the experience they have, thanks to that data.

Would you think that maybe Facebook would be super quick to build the AR app for the Apple AR product?

[0:12:24.0] CA: I think usually that makes a lot of sense. I think Facebook's primary advantage is their data and the social network, obviously, of people, that it's just hard to dismantle or to build some competing product, but the question remains to be seen whether or not they use this for augmented reality will center around your personal data, and whether or not it would be a software solution as much as it is a hardware solution.

It's quite possible that we'll see 5 or 10 years everybody building augmented reality solutions, and ultimately, none of them will win. It will be a software play that wins. I think Facebook is well-poised to win that, but that doesn't necessarily count on them having a first move or advantage.

[0:13:04.1] CM: For fun, what do you think the big app for AR is going to be? Off the top of my head, I think being able to see people's LinkedIn profiles in AR would be pretty cool.

[0:13:13.9] CA: That would be cool. I think there might be a little bit of public backlash though. You think about Google Glass. It's okay, "Why did people not like Google Glass?" It was a little bit sketchy. It was a little bit, you don't know exactly what this person is doing. Are they recording you? Are they taking photos of you, and once it gets you to a further step, can people see you LinkedIn profiles and you're personal data just by looking at you and identify on the street? I think it was a whole social aspect of that where we could look at movies and we could see what's cool, but I don't think people would really stand for that.

[0:13:41.4] CM: How interesting would it be? Just kind of imagine a world where if you see someone famous on the street, you know you got to go up and take a picture. How interesting would it be if you see someone you never seen before on the street, but their Twitter bubble pops up and they say they have 10,000 followers, what do you do at that point? That would be incredibly interesting how many Twitter followers does someone have over their head.

[0:14:03.2] JM: So I think the big utility of augmented reality — Caleb, you talked about what's killer app. I think the killer app is getting us to look up from our smartphone, because of all the time where — We're currently in this mode of operating where you're having a conversation with somebody and you're contact switching between reality and your phone. That sucks. It also makes sense to a point. You guys know I'm not allergic to technology, and I'm willing to be in an intimate conversation with somebody and then say, "Hey, I just remembered there's this this article that will be really relevant to what we're talking about right now. Hold on one second while I look it up."

That sucks. It's a contact switch. It's often worth to keep the conversation going. You guys are both intellectuals and you want a conversation that's rooted in the facts, and sometimes if you're having a conversation with somebody, you can't remember the facts about something. You pause to look it up. It would be — I think introducing more facts and information and integrating the raw well-structured information of the internet with the organic conversations that we're having day-to-day, I see that as wonderful, but it's going to take a mindset shift.

Courtland, to what you said about virtual reality, why don't people use virtual reality? I don't know if you guys have tried virtual reality. It makes me uncomfortable how engaging and addictive it is. It's just like I don't know if I'm ready for this. It's almost like you're in college and you take a hit of some drug and you're like, "Okay. That's the last time I'd try that for a while. I'm going to wait until we either have the pharmaceutical version of this, that I know is a little better tested." I'm just like, "I don't want —"

I've remember hearing about this with Google Glass, there was some dude who got admitted to a psychiatric institution, because he had overused Google Glass and he was like — Even when he wasn't wearing the Google Glass, he was swiping in front of his face. I don't want to develop those kind of tendencies.

[0:16:06.0] CM: Yeah. I can imagine a Google auto complete while you're talking. You're searching forward and Google tells what that is. How awkward would it be if you're in a conversation, we're sitting down, and it's like, "Oh! I read this really great article. Wait one second." You keep eye contact the entire time while you kind of like move your hand up and down trying to scroll or search through it with your mind or something. To me, that's even more awkward than pulling out your phone an maybe typing for a couple of seconds.

[0:16:31.0] CA: Did you guys see that move Her, where essentially everybody have a smartphone in their pocket with the camera facing out and just an earpiece end. It's not like some sort of thing obstructing their face and it could do what you were saying and get live information from the internet about any topic they're talking about. I thought that user interface is super cool, and that lends itself more to what you see with Google Home, or Amazon Echo, where it's how do you develop a voice controlled user interface that's totally hands off.

Yeah, I get the point as well, that it might be awkward to maintain eye contact while talking to your computer.

[0:17:02.9] CM: Yeah.

[0:17:03.3] JM: Well, Courtland, that's a great point, because that's probably going to be a product that comes out before augmented reality.

[0:17:10.2] CA: I would assume so.

[0:17:10.6] CM: Yeah. Do you think we have to worry about noise pollution in that world? In the Alexa world?

[SPONSOR MESSAGE]

[0:17:23.3] JM: Dice.com will help you accelerate your tech career. Whether you're actively looking for a job or need insights to grow in your current role, Dice has the resources that you need. Dice's mobile app is the fastest and easiest way to get ahead. Search thousands of jobs from top companies. Discover your market value based on your unique skill set. Uncover new opportunities with Dice's new career-pathing tool, which can give you insights about the best types of roles to transition to.

Dice will even suggest the new skills that you'll need to make the move. Manage your tech career and download the Dice Careers App on Android or iOS today. To check out the Dice website and support Software Engineering Daily, go to dice.com/sedaily. You can find information about the Dice Careers App on dice.com/sedaily and you'll support Software Engineering Daily.

Thanks to Dice for being a loyal sponsor of Software Engineering Daily. If you want to find out more about Dice Careers, go to dice.com/sedaily.

[INTERVIEW CONTINUED]

[0:18:42.8] JM: Oh, of course. That's like case in point for why nobody is going to be want to be first mover, because Apple — Let's say Apple is like, "Oh! We got our AirPods going. We got some users on AidPods. Now, let's do Siri version of her." If Apple says, "Hey, we've been secretly working on a different version of Siri, it's a million times better. That's why we didn't acquire Viv. We've got something brand new, and we're releasing it." Then, they release it. The UX, the actual UX is like, "Oh, I'm talking to somebody and all of a sudden Siri 2.0 interrupts me." I have to say to the person I'm having a conversation with, "Oh, hold on. My AirPod's interrupted me while you were saying that, and I couldn't hear either of you at that time." That's why Google and Alexa both do not do voice push notifications.

[0:19:33.8] CA: Right. That seems to me like a user interface issue that could potentially be solved with some cloud resolution that allows you to just mute the thing at the appropriate times. We're also heading into the territory where we're just augmenting communication abilities to such a degree. If you could walk around with a little earpiece in that could give you the information that you need at the right time with very little input from you, then you essentially — The internet is now part of your brain in a way that it's not when you've got a smartphone.

I got a Google Glass. I think it's the Explorer Edition. I got it before it was no longer cool. Just to your point, Jeff, about the guy who was addicted to it. Just the ease of being able to move my hand up to my face and get some information very quickly, versus pulling out my smartphone. It seems like it's not that different, but it's sort of around four or five times faster. I would just do things more often. I would send more text message. I would make more phone calls, because it was just five times easier.

If we had some sort of interface with people that actually just get information directly into their ear, I think they will probably be using it constantly and they would probably enable individual people to do things they normally couldn't do. You could climb under your car, for example, and just fix your car, and not be a mechanic. Just getting instructions live fed into your ear, and things like that. We'd all become kind of superhuman in a way.

[0:20:49.4] CM: Yeah, that would be interesting. You get to pick up your woodwork in hand.

[0:20:53.0] JM: Courtland, it sounds like for you, the Google Glass was too good, and that's what made it too early to market. We had these pictures of "glass holes" and you see people around, "Oh, I saw some asshole wearing his Google glass the other day." What makes him an asshole?

I think this is indicative of the growing divide between the tech people and the nontechnical people where there's a sense of resentment and any sort of a dividing line that is indicative of income ability and that lends itself to a further accrual of power, which is Google Glass. If it really was this massive piece of utility, I think that's why it didn't work. I don't think it was like, "This thing sucks." I never heard somebody who had a Google Glass say, "This thing was total garbage." I've heard more people say that about Apple Watch.

[0:21:45.1] CA: Yeah. It definitely did not suck. It was good in a lot of ways. It made things easier. I don't think that it was good enough to be worth the fashion cost. I think every society has a concept of fashion, or people will be — There will be moral outrage based on violating totally arbitrary fashion norms. Wearing a computer on your face violated fashion norms to a humungous degree, and it didn't help from the financial standpoint that you mentioned. It didn't help that Google made everybody who use an Explorer pay $1,500 with this experimental prototype.

I think they're focused on trying to demonstrate that they had business value and that could make money off of it. When in reality, they just made everybody who bought one seem like a rich-entitled asshole, because the first thing anyone ask you when you walk into a bar and you're wearing Google Glasses, "Where did you get it? How much did that cost you?"

[0:22:34.2] JM: All right. Let's talk about online advertising. Advertisers are pulling their ads from YouTube. We have articles trending on Hacker News where people are saying they're done with Google. They're not going to use Google AdSense to monetize their product anymore. Are we seeing the beginning of the end for the online advertising industry as it stands today?

[0:22:59.8] CM: No. I don't think so at all. I think advertising is too powerful a tool that it's never going to go away in any capacity on any platform. Even if there are some people pulling away from online advertising, that just makes online advertising cheaper, so that the people who wouldn't get online advertisement before, because it was so expensive, can now just come in and buy some more ads.

Maybe it's not write for some companies and for what they want to accomplish, but I don't necessarily think it means that online advertisement is going away. Do you think it is?

[0:23:30.3] JM: I think there are some massive factors that are not factored in to the market price of online advertising. You know I've done so many shows about advertising fraud, and I think the biggest problem is that you cannot tell if a user is a bot or not. If you're somebody who is buying advertising from a company that is selling you online advertising, whether they're some sort of middlemen distribution platform, like AppNexus, or you're buying from Facebook, or Google, you have a contact to that company and you can ask them, "Hey, how do you know that the ads that I'm paying you for are getting served to real users and not bots?" They will not be able to give you a straight answer.

The more advertisers start to ask that question, the more light will be shed on this problem. You guys both know, I've been reporting on this super aggressively. Basically, nobody cares.

[0:24:28.4] CM: Yeah. So then what are the alternatives? The two alternatives I can think of are regulation and perhaps monetary-driven ad results versus views or clicks.

[0:24:38.0] JM: I think it will be really tough to regulate. I think this is going to be something where the market responds and says, "We're not paying for this garbage," and they're going to look for other ways to shift their money such as to things like podcast advertising and stuff like —

[0:24:53.3] CM: That's got just as many problems.

[0:24:55.6] JM: it has some problems. Let me put it more directly, and I think this applies to you too Courtland. I think this is how you do advertising on Indie Hackers. Instead of selling based on clicks or some sort of metrics-driven campaign, you sell based on the amount of time that it runs against your content. If you say, "Hey, Nike. You get ads on my podcast." Every episode will have a Nike ad for the next two months. We'll look at the metrics and there will be some correlation between the cost and how many listeners there are, but we know there's a lot of bots out there, and we're not going to charge you based on what percentage of listens we get. We'll just charge you based on time.

Courtland, that's how you do advertising on Indie Hackers, right?

[0:25:42.4] CA: It varies. I think what you're talking about brings up a point that I've learned in dealing with advertising, because I've only been doing ad sales for the past four, five months, and I learned a lot about it. Really, you've got — Nike would be a brand advertiser. They might not care. They probably don't care about conversations, they just want to get their brand out there. Coca-Cola, McDonalds, any big company like that, Apple, Microsoft, versus maybe smaller advertisers who are trying to grow their business from a medium or small size and trying to find some channel that's going to get them customers at a cheaper rate than other channels. They really do care about conversions. They want to invest money in places where they're actually going to get clicks and they're actually going to get customers who pay them. Otherwise, it's not worth them spending the money, because they don't have the money to blow.

I think it really depends on what types of advertisers that we're talking about. Yeah, for brand advertisers, for sure. You could probably just sell them on time. You could just tell them how much reach your podcast has and they probably won't care about clicks at all.

[0:26:39.1] JM: To what degree do you think brand advertising is actually effective?

[0:26:44.0] CA: That's a good question, because I remember being a kid and always complaining about it on TV. I was like, "What do I have to watch another McDonalds commercial? Everybody knows what McDonalds is. It's just a waste of their money and a waste of my time." Then, the question arises, "Why do we all know what McDonalds is?"

Earlier, we were talking about modes and I was saying a couple of the modes that your company can have are network effects, or proprietary technology, but another big one is just economies of scale. When you can afford to do things that your smaller competitors simply can't. I think as long as there are giant companies with a lot of money to blow, whether or not brand advertising works, it's something that they can do that their competitors can't, and so they're going to invest in it.

[0:27:22.9] JM: You think about the brands that have been built over the past 10 years, none of them are built off of brand advertising, I don't think. You think about Dollar Shave Club, or Amazon, or Uber, or Chipotle, or Panera Breads, all of these companies, they deliver something that is qualitatively different than their competitors and they build awareness for that product through brand advertising, but the core product is differentiated.

[0:27:57.8] CA: Yeah, for sure. I'll be the first to say that you're not going to be able to build an inferior product that people don't find useful or that doesn't solve some problem, or satisfies some drive. Then, just make up for the advertising. You're just going to lose a lot of money if you go that route.

At the same time, I think that companies are as — There's the growth story. How do you get big? How do you take over? There's also the how do you stay big story. If you look at any graph, or any chart of the top 10, or 50, or 100 companies many time period, companies die. Most companies don't last for 30, or 40, or 50 years. Especially, public companies are big juggernauts. They're always worried. Facebook is probably always worried about what's going to kill them. What might kill them, and how do we eliminate those threats.

I think a lot of advertising is defensive as well. It's not so much how do we use brand advertising to become a juggernaut, but it's like how do we — What advantages do we have that we can use to maybe stave off our inevitable demise as a company? Brand advertising for bigger companies might be a tool in that toolbox.

[0:28:58.9] JM: When you guys look at the fundamental computer science problems faced by large advertising companies — Personally, I see two main problems. One is the YouTube problem that people are pulling their brands, because of which is we don't want our brands associated with controversial content. We told that to YouTube and they still matched us with controversial content. The indication there is that Google cannot detect controversial content on the fly. It's too hard. They can't put a human in the loop for every submission to YouTube, or if they try to do that, then the quality of the human in the loop goes down. The less frequently discussed issue is how can you identify bot traffic at scale. Do either of these problems seem solvable to you two?

[0:29:49.5] CM: Yeah, I definitely think the controversial content problem can be solved. I'm not confident that the bot traffic content can be solved, because with bot traffic, the actor on the other end is always fighting to undo whatever you did. On the controversial content side, that's — It's a little easier to even crowd source that. If someone reports a video, then all of a sudden that goes into a queue or something, or you start building up a database of what is controversial and what's not and you can train machine learning algorithms off of that. I think it's expensive. There's no clear answer, but it's solvable, because it's a fixed problem, unlike bot traffic.

[0:30:25.1] CA: Yeah, I agree. I think the first one is more of a technological solution. Like you said, you may be crowd source it, maybe someone will invent better technology as AI gets better. Maybe could automate it.

The second one seems to me more like a social problem, which is if everybody agreed to use the same software, maybe every single user uses Facebook login to go to any given website, or some sort of open-source tool, and the problem will be solved and you would be able to identify who's viewing what at any given time, but like, "How is that going to actually come to pass?" Probably it will never come to pass. It's more of a social than a technological problem, and I don't see a solution to it.

[0:31:01.0] CM: Even then, how hard would it be for a bot network to create fake Facebook accounts, infuse it with a bunch of data that's adjective shuffled from a normal user's account and then use that. At that point, it's like in order to verify the user you need to give your social security number in something like Facebook.

[0:31:22.7] CA: Technology, it's not hard to ask people for their social security number, but socially, are people going to do that? I don't think anyone's going to opt in any kind of system that would work.

[0:31:33.3] JM: Let's talk about developer tools. You guys are both programmers. Courtland, as the person running Indie Hackers, you see how people are building stuff constantly. Caleb, you're just a hacker. You've written a lot about open-source tools. You've obviously reported on stuff through Software Engineering Daily. What is the coolest paid tool — Let's start with Courtland. Courtland, what is the coolest paid tool that you saw for the first time recently?

[0:32:00.8] CA: Can I punt on this for one minute? I got to look at it from my inbox, because I don't remember very —

[0:32:05.1] JM: Okay. All right, Caleb, you go.

[0:32:05.9] CM: Yeah. Disclaimer, I am a little biased. I did a bit of contracting for Meteor and Apollo, but I really like their Apollo Optics tool. That's the perfect tool for GraphQL. If you're sold on GraphQL and if you like GraphQL and if you're going to use GraphQL, Apollo Optics is the perfect tool for you. It's almost required to create a GraphQL server. If you're not into GraphQL, then Optics is a great reason to start using GraphQL, because you get these great observability features. You can see the waterfall of your request as it goes through your system. It's a really great product for monitoring response time in a domain-specific application language.

[0:32:45.6] JM: Explain why you're bullish on GraphQL. One side point.

[0:32:50.5] CM: Yeah. To me, GraphQL represents the ideal way that myself as a UI engineer would interact and talk with data from the server. I think about my application as a tree. Whether it'd be an HTML DOM tree, or a view tree in a native application. REST does not exactly mirror that. REST IS more graph structured, or something else.

GraphQL allows me to think about my data and to fetch my data in the same way that I think about and build my user interfaces. GraphQL allows you to fetch your data as a tree and then throw that right into your component tree, your UI tree. Then, on top of that, it gives you a whole bunch of UI consistency features that are a pain to implement, and that require a lot of expertise and a lot of custom work if you're going to use a custom one of REST API. Whereas with GraphQL, because it's standardized, you can put a lot of these core fundamental real time and data-merging features, data consistency features into frameworks and abstract that across all applications requiring a different custom solution for each API. I hope that explains it well.

Sometimes, I feel like I'm so deep into GraphQL. It's hard to explain and especially hard to explain to people who might not be familiar, uncomfortable with using a REST API. I think that's the best explanation I've come to to-date.

[0:34:13.6] JM: No. No. Good explanation. All right, Courtland. First, the coolest paid tool you saw recently.

[0:34:18.8] CA: I'm probably not the best person to answer this question simply because I tend to go into these fonts whenever I'm working on a company or a product where I just stop paying attention to all of the new tech that comes out and I only focus on, "How do I grow my thing? How do I connect with customers or users?" Then, after I get to the end of it, it's like I wake up out of a stupor and I look at the world around me and everything is just new and different.

If I had to name a tool that I think is cool, I think Kite is cool. They describe it as the smart copilot for programmers, and it's basically this app that sits open net to your ID as you develop and pulls in useful contextual information from the internet to help you write the code that you're writing.

It's pretty new, and I don't know exactly how well it's doing, but I've taught a few people to code in my lifetime and I think it would be incredibly useful to have some sort of automatic tool that augments your programming ability for people who aren't learning the code. I think a lot of times people spend — You need just a really good Google-fu to learn to code basically, because you got to constantly run into errors. You need to Google. It simply aren't going to be in the textbook that you're learning whatever course you're learning it from.

There's actually a tool that could sit there and look at what you're trying to do, analyze it, come up with the most likely problems that you're experiencing and just service that to you instantaneously. I think we would see tremendous increases in the number of people who successfully learn to code, which would change the world on a lot of ways. I think Kite is —

[0:35:42.1] CM: Have you used it? Does it work?

[0:35:43.4] CA: I haven't actually used it. They just released a support for Python, which I think it's going to expand its reach a little bit. Like I said, I kind of just go into these holes where I don't use any of the new stuff. I just kind of read about it every now and then.

I'm not sure that it works, but I think we're going in the right direction. I can't imagine a world 5 or 10 years from now where there isn't something like this that is working and impactful.

[0:36:06.3] CM: Yeah, I know. I agree. It does sound super exciting. Just trying to imagine it working in the stacks that I work with daily, it's like there's so many different moving parts. I'd be fascinated to see it work.

[0:36:17.1] CA: Yeah. I think that's why I think its application is probably — Right now, I think they're targeting a lot of developers, but I think in the long run, I look at it, it would be for beginners, because if someone like you, you're using like how many different languages, and how many different — You're going [0:36:31.3] servers, like backend stuff, front-end stuff.

[0:36:34.7] CM: Still a lot of Stack Overflow on Google though. It would be great. It's just be fed Stack Overflow answers.

[SPONSOR MESSAGE]

[0:36:49.3] JM: Deep learning is at the forefront of evolving computing and promises to dramatically improve how our world works. In order to get us to that bright future, we need new kinds of hardware and new interfaces between this AI hardware and the higher level software. That's why Intel acquired Nervana Systems, a platform for deep learning.

Intel Nervana is hiring engineers to help develop this full stack for AI, from chip design to software frameworks. Go to softwareengineeringdaily.com/intel to apply for a job at Intel Nervana. If you don't know much about Intel Nervana, you can check out the interviews that I've conducted with engineers from Intel Nervana, and those are available at softwareengineering.com/intel as well.

Come build the future of AI and deep learning at Intel Nervana. Go to softwareengineering.com/intel and apply to work at Intel Nervana. Thanks to Intel Nervana for being a sponsor of Software Engineering Daily, and I really enjoyed the interviews I have done with the Intel Nervana stuff. I think you'll enjoy them too.

[INTERVIEW CONTINUED]

[0:38:12.3] JM: I've done some shows about static analysis tools. I think I did a show with a company called bitHound which does static analysis, and this is basically you do a git-push and somewhere in the push workflow it does analysis of your code and it can say, "Do you have a security flaw? Are you importing the right up-to-date version of your JavaScript package? Do you have some sort of no-pointer type of situation happening?"

Obviously, this is a really hard area of work, but there is constant work going. You can imagine it. Basically, it's the equivalent of the Google Glass thing that the manufacturing engineering wears, or the mechanic wears that is constantly giving them, "Hey, you might want to do this." It's got to be a little better than — If you could say, "Okay, Kite. Find me the answer on Stack Overflow," or something. That's so much better than contact switching to Stack Overflow and doing a search there, and then you're like, "Oh. This isn't the right answer."

[0:39:17.0] CA: Before you know it, you're browsing Facebook.

[0:39:18.5] CM: Yeah. The way I was thinking about it is it doesn't just statically look at your code, but it's somehow hooks into your terminal or something so that whenever an error Stack trace pops up, it takes the error message, or whatever, in the Stack trace and is able to narrow it down to the GitHub repository and to issues that look like this one.

[0:39:36.5] JM: Yeah, we're going to get to Stripe in a sec. I was using Stripe recently, and I had a question for support. What was cool about the support interface is as you're typing your support query, there's a little thing on the right where it's looking through the FAQ and the documentation of Stripe and it's looking for the answer that most closely matches your support query semantically. It's almost like you've got this real time interface of support information, "Oh! Maybe you don't have to actually file the support ticket. We've got this live updating search query on past support tickets and FAQ stuff." I was like, "This is really smart, and I haven't seen this in a whole lot of other places." We are literally entering a query and it's asynchronously doing a predictive thing. That's essentially what kite would do for your program.

[0:40:30.0] CM: Yeah. Stack Overflow, it does actually do that, where if you are opening a new question and you start typing the answer in the body of the question, it will popup similar questions. I've actually found that sometimes that can be better than Googling the air message or searching Stack Overflow is popping up a new question as you start to think about it more and think about how you'd word the problem. It actually pops up.

[0:40:53.1] JM: Okay. Open-source tools. Coolest new open-source tool you saw recently?

[0:40:58.1] CM: Yeah.

[0:40:59.4] JM: You could say GraphQL.

[0:41:01.2] CM: GraphQL. No. I think the obvious answer for me would probably be Expo. I was actually bearish on Expo before talking with the team on React Native. I thought, "Oh, you know. It's just overhead, that if you're a raw React engineer, you just pull up React Native and do your stuff."

What kind of realized when talking to them is this idea of the shared native runtime is the words they use which really caught my imagination, enables a whole bunch of interesting thing. You can basically do everything you could do when deploying to the web. Posting JavaScript to an endpoint and automatically reloads some better error tracking, et cetera.

You could do that on a Native app, and that's kind of insane. If you have a Windows machine, you can program for an iPhone, because there's a share native runtime. Probably, Expo and some of the stuff around React Native would be mine.

[0:41:53.0] CA: I'm in the same boat with this question as I was last time. I'm behind, but I'm a huge fan of Ember.js. A lot of frontend developers have their favorite framework, and I've been with Ember for years. They just released Glimmer.js, which is kind of their rendering engine. I think one of the problems of people adopting Ember.js in the past is it's a solution that go to when you want build your entire website from scratch and it's going to be single page app. It's a little bit unapproachable for some people, because compared to React, for example, you just jump and you build one component at a time, and that's all you have to learn about the React Stack. If you want to build a router, okay, then you want a new thing. Whereas Ember kind of forces you to learn the entire thing.

Glimmer, I think, is their attempt to decouple certain parts of Ember from other parts of Ember and allow people to bite off a smaller chunk. I think it's really cool, and I know I will be using it on future projects.

[0:42:44.4] JM: All right. Let's shift the topic again. Courtland, Indie Hackers was recently acquired by Stripe. Congratulations on that.

[0:42:52.0] CA: Thanks.

[0:42:52.5] JM: I wanted to talk to you about what you learned from the process of building something from Scratch, not raising any money. You boot-strapped it entirely and getting to a point wehre you could sell it to Stripe. Take me through that process, or at least — You don't have to take me through the entire process, because people have heard that — If people haven't heard, we did an episode with you. It was a very popular episode. What are the important lessons that you learned in building Indie Hackers from boot-strapping to acquisition? Looking back.

[0:43:31.8] CA: Yeah. Just kind of to get the context of that question. Boot-strapping from the very founding of Indie Hackers to the acquisition point is more of just an arbitrary period of time, unless I had a goal to get bot. In fact, when I got the email from Patrick Collision in Stripe, I was shocked that anyone would want to buy Indie Hackers, because it's basically a one-man blog, one-man media site. Not the kind of thing that build to get acquired, and I was focused for the last five months on really doing nothing but generating revenue, because the entire point of the project was to be self-sustaining, to be independent, and to be able to provide my own income without having to work a job elsewhere.

All the lessons that I had learned are more aligned with how to build a good product, how to build something that people liked, and how to make money from that project. I'm not really aligned with how to get acquired, because I think that depends entirely on the acquirer. What Stripe is looking for at any given moment is going to be totally different at what any other acquirer is looking for, and I think probably one of the worst things you could do is build a company with the sole intention of getting acquired, because it relies so much on just the arbitrary awareness of one or two human beings at that company.

Yeah, I think a lot of the lessons that I've learned in Indie Hackers, I learned in a sort of meta way, by doing the interview for Indie Hackers. So for those who don't know, Indie Hackers is a website where I talk to developers mostly and other entrepreneurs about their profitable side projects and business and I ask them, "How did you come with your idea? How did you grow your user base and find your first paying customers? How much money are you making?" Everybody shares their monthly revenue.

Inside these interviews, there are a lot of lessons as you can imagine, that people have learned the hard way, and they share it with everybody else that you don't have to learn it the hard way. I've taken a lot of those lessons to heart myself. I feel like eight months after starting Indie Hackers, I can look back on myself eight months ago and say, "Wow! That was a total new when I came to marketing and sales and building a product that people would actually like," just because the number of lessons that I've learned by doing the interviews. I would recommend anybody who's listening to go check out those interviews if that the goal that you have.

There are certain things that — For example, one thing that locks a lot of people is competition. I think, "Hey, I want to up with a new idea, or I want to start working on this project, but there's already someone in this space who's solving this problem, and therefore I can't do it."

When in reality, when you start to look at the examples, the fact that a space is crowded with lots of competitors almost always means the problem that's being solved is very really problem that solves the customer pain point that a lot of people are searching for. It's usually pretty lucrative. What you need to do is not differentiate and try to solve a different problem that your competitors necessarily, but you see a little bit of a unique solution.

If you're in a winner-take-all market, like social networking, yeah, you're probably not going to de-throne Facebook, but if you're in a different market, like analytics, different customers have different needs when it comes to analytics, or advertising, like we were talking about earlier. There are brand advertisers who have completely different needs than small time advertisers who are just trying to get .

You really need to differentiate your solution and not run away from problems, because there are competitors. In fact, you can use competitors to define how you're different. Nathan Barry, for example, from ConvertKit, basically, does email marketing, and he's got huge competitors like MailChimp, and he loves the fact that MailChimp exists because he can tell people when they ask about ConvertKit, "Oh, it's like MailChimp for professional bloggers." If you're a professional blogger and you hear that, which one are you going to use, MailChimp or MailChimp for professional blogger? You're going to use the second one.

I think that's a really big lesson that I've taken to heart. I'm in MicroConf right now talking to a lot of aspiring entrepreneurs, and that's one I find myself repeating to them over and over and over again, "Don't worry about the competition. Just make sure you're solving a problem that people actually needs solved, and then differentiate it on your solution."

[0:47:11.5] JM: The problem that you were solving with Indie Hackers was a problem of media, and I can speak firsthand that there is so much opportunity in the media industry, especially for engineers. If you're an engineer who doesn't know what business to start, first of all, there's a million good pieces of advice on indiehackers.com, advice on ideation and whatnot. If you still can't think of anything after looking at Indie Hackers, just go into media. Be a media person that reports on software, and it's so easy. The downside is totally capped, because worst case scenario, you learn a ton about software.

When I started Software Engineering Daily, I was like, "Okay. I've got enough money to last me for four months and hopefully I can to breakeven by then. If not, I'll just have worked really hard and learned a lot about software and my downside will be completely capped." It will be like I just — It's like people take out student loans to go learn with that sort of expected value, or worst expected value. You might as well start a media company and interview the smartest people in the industry for free.

[0:48:22.8] CA: I like that point a lot, because I feel the same way with Indie Hackers. The amount I've learned is totally worth the amount that I invested into the company. I did a similar thing, saving up money and then living off of my savings. If Indie Hackers never made a dime, never got acquired, just the amount of information that I've learned by talking to people who've done it would have made it worth it.

[0:48:40.3] JM: The big question I want to ask you about working on Indie Hackers, is around working productively with your brother, because — Recently, I started Adforprize, it's like another company I'm working on, and I hired my brother for some contract work in the process of getting him some equity. I love my brother. I admire him as a developer. I love his skill set. He is five years younger than me. There is a gulf in mindset, there a gulf in experience. There's not a gulf in talent, but he has different skills than I do. The risk is amplified by the fact that you really don't want to have a falling out with a family member in a business context. How do you protect against that sort of thing when you're working with your brother?

[0:49:35.8] CA: Yeah, it's funny, because I see it almost the exact opposite way. I think it's just a testament to the fact that every family is completely different. My brother, for those who don't know, it's my twin brother. We grew up basically fighting and arguing about everything under the sun for 18 years. It's pretty clear — Then, we lived together in San Francisco for another six years in our 20s. It's clear to me at this point that there doesn't exist something that could occur that would ruin our relationship, at least not in a business context.

That was, working with my brother is safer than working with any other random cofounder, because I know that we could disagree on — we can disagree on almost anything. Ultimate, our personal relationship won't be ruined, and we'll probably be able to work it out.

It really, I think, depends on your family relations. Also, I'm in kind of a similar situation to you where my brother — He majored in literature. He's written two novels. He didn't come from a tech background, whereas I was coding when I was in middle school. I actually taught him to code four or five years ago, and he kind of got into this industry. There's kind of a gulf in experience and a kind of a gulf in knowledge there. Also, I started Indie Hackers by myself and he only started working with me a couple of months ago.

Yeah, there's a lot of, I think, teaching and guidance that's to go on, but I talked to Laura Roeder, the founder of MeetEdgar the other. She had really good things to say about managing people and how she kind of grew as a person and grew as a manager by getting practice. The thing that she had to say was that, "You don't want to delegate. You don't want to send someone off and give them a task and have them come back and then say, "Okay. You did it all wrong. Let me make my fixes and change it."

What you really want to do is empower people and give them ownership and autonomy, so they just own something and they can do it to the best of their abilities. That's what I've been trying to do with my brother and onboarding him to Indie Hackers. I'm just like, "Okay. Edit this interview. It's yours. We're going to push it live."

Nowadays, we now work on improving a community form a lot, and that's something that I personally haven't touched in months. I imagine we'll divide up responsibilities in some way where he can have ownership of some part, I can have ownership over some part, and we can both take pride in trying to do a good job rather than one of us having to micromanage the other.

Those are all the tips I got now. I'm pretty new to this. This might go up in flames. Personally, I know that no matter what happens, we're always going to have a good relationship as brothers.

[0:51:54.5] JM: The acquisition, it makes so much sense for Stripe, because Stripe's goal is to increase the GDP of the internet. It makes sense to have a media outlet around Stripe, or within Stripe that is reporting on and encouraging this burgeoning Indie Hacker world, because Indie Hacker, like many other startups, is one of these things where it looks like it's a small industry. Who cares about these Indie Hackers that are just making these small side businesses? There's even this condescending term, lifestyle business. As you know, this is basically just a made up term for venture capitalists to make entrepreneurs feel worse so that they can lower the valuation of that company and get a better deal, and it's like — You've seen enough "lifestyle businesses" that are making a quarter million dollars every month to know that the term doesn't mean anything.

I love the fact that Stripe acquired it, and I am really looking forward to seeing support and investment that will come from Stripe affecting Indie Hackers, and I know we were talking some about like working together on some podcast stuff. In any case, those opportunities still remain if there's any kind of partnership we can do. I think that'd be great. I think Stripe is — Stripe is an amazing company.

[0:53:22.4] CA: Yeah, I would love to do a podcast episode on Software Engineering Daily. I found podcasting so difficult to get into. It's so hard for me to build up a regular rhythm of recording, and editing —

[0:53:32.4] CM: What's hard about it?

[0:53:34.4] CA: I think I end up juggling a lot of task in Indie Hackers, so I'm editing the tech interviews and sending the newsletters, strategizing for how to improve the community and coding there. Dealing with a lot of email, a lot of tweets, a lot of DMs, and I end up doing things kind of last minute. With podcast, if I'll record three episodes, and then I'll be, "Okay. Great. I'm set for three weeks," and then I don't necessarily edit them until right at the last minute where I have to edit them. It's kind of the only thing in my schedule that is scheduled.

For the interviews, I can do them kind of whenever I want. The podcast, I have to actually carve out a block of time in my calendar and meet with someone. That's something that I like — I think most people who had jobs were probably used to doing. I never had a full-time job in my life. I've never had to be anywhere at any given time. I think living on a calendar-based schedule, the podcast, and getting things out, it's been difficult to me.

[0:54:25.4] JM: Yeah. Maybe we could talk about Software Engineering Daily. Maybe we can be your backend, or help you with some of that if you want.

Caleb, you have been thinking about the podcast space. I know I've talked to you some about potential tooling that you could create. What are the ways that you see the podcast industry as being broken?

[0:54:46.3] CM: The number one way that I think it's broken currently — There are two ways, actually. The first on a technical perspective. There's no good way to get listener accounts, or to get good analytics for the people listening to your podcast. You host your podcast somewhere online. It's fed through an RSS feed and then someone listens to it, or 50% of people who listen to it through the Apple iTunes app, and that's about it. You don't get any more analytics. You get, perhaps, partial downloads, so three downloads from one device while it's streaming. You might get a bunch of downloads from someone trying to do aggregate analysis for all podcast.

It's very hard to discern if people listen. How long people listen? At what speed people listen, and all of these other data points that would greatly help both podcasters and advertisers get a better look at the industry.

The second problem is discovery. There's still currently no good way to find great podcasts other than word of mouth and viral growth in that direction. iTunes has — It's directly, or whatever. The top directory is the 500 podcasts that everyone knows. The top 500 everyone knows and talks about. It's really hard to find new interesting niche content that you might not be familiar with.

[0:55:57.0] JM: Great overview. Okay. I know we're wrapping upon time here. You guys have both been tremendously helpful to Software Engineering Daily, helping it grow, helping us increase quality and different voices. I want to close off by just asking you; what are the areas that you think Software Engineering Daily can improve and where would you like to see us expand towards?

[0:56:18.6] CM: I'd really love to see more just general topic, roundtable podcast like this with consistent guests and on a separate feed. Kind of like Exponent, if you listen to Exponent, but from a Software Engineering Daily perspective, because, Jeff, you know a lot of very interesting people with very interesting ideas. It would be great to sit some of the most popular guests down consistently every week or month or so, and to get an update on what they're thinking.

[0:56:47.3] CA: Yeah, I think that's a great suggestion. I also would like a higher percentage of the episodes to be relevant to me, which is a bit of selfish request. It's like, "Who am I? How do you know exactly what I want versus what a different listener wants?" It's easy for me to look at an episode and say, "Okay. This one, I can skip, because I don't know that much about [Lula], for example, are." I think the Roundtable format — This episode, it's been really fun to be on, because we're just kind of riffing on random topics, some of which I'm totally unqualified to speak towards, but I'm sharing my opinion anyway. Yeah, more Roundtables and more — What Caleb said, and using your expertise to find guests and partner then and paid them to see who would be good to get it.

[0:57:25.0] CM: Based on what Courtland said, sometimes it's really easy to tell by the title what episodes I'm interested in, but sometimes it's definitely not, because all you get is a title and they will have to listen to the first five to 10 minutes to hear, "Is this guest interesting? Is this an interesting topic?" Maybe that's a follow up of the podcast format, not giving me more information by an episode before listening to it. Yeah.

[0:57:45.6] JM: Okay. We'll close off there, because I know you guys both got stuff you got to get to, but we will do more of these. I really enjoyed talking to you both. As you guys know, we didn't get to half of the questions and it seemed like every discussion had more ground to cover us, probably a sign that we should do more of these.

I think to the point of what both of you said, there's a lot of stuff that we covered that is not relevant to everybody, and I think this format is more accessible to a broader audience. I will take that to heart.

[0:58:15.3] CA: Okay, great.

[0:58:15.4] CM: Cool.

[0:58:16.6] JM: Thank you both. Courtland, congratulations again. Caleb, congratulations on your excellent recent episodes. If people have not checked out the web assembly episode with Brendan Eich and the episode Caleb did with the Exponent members, then you should check it out. Thanks for coming on guys.

[0:58:34.6] CA: Thanks for having us.

[0:58:34.4] CM: Yeah, thanks so much for having us.

[END OF INTERVIEW]

[0:58:40.5] JM: Thanks to Symphono for sponsoring Software Engineering Daily. Symphono is a custom engineering shop where senior engineers tackle big tech challenges while learning from each other. Check it out at symphono.com/sedaily. Thanks again Symphono.

[END]