

EPISODE 632

[INTRODUCTION]

[0:00:00.3] JM: Banking has been a part of the economy for 600 years, and banking is always evolving. The most recent evolution, the financial industry has been going digital. Newer fintech companies have created innovative ways of doing everything related to money; from friendly peer-to-peer payments, to budgeting; from business transactions, to insurance. However, the traditional banks themselves have been relatively slow at adjusting to these digital changes creating the opportunity for native digital banks, often referred to as challenger banks.

N26 is a digital first bank established in Berlin and it's active in 17 European countries with a million users. Pat Kua is the CTO at N26. Digital banks are hosted on the cloud without a physical branch that the user must go to to perform an operation. The user accesses their account through a mobile application and can complete everything online from opening an account, to performing transactions, and this system has numerous advantage; simplicity for the user, higher scalability in terms of users from the bank's perspective and several other advantages that we'll discuss.

We talk about the advantages of digital banks in this episode and we also talk about the fintech industry more broadly, of course, N26, as a financial company and a technology company itself. We also explore product development and how Pat manages his time as the CTO, and this is a useful discussion for anyone who is learning to be a technical leader or a manager. I certainly found it useful, and Pat is great at explaining how he balances his time between teams, such as product and design and engineering.

[SPONSOR MESSAGE]

[0:01:58.2] JM: Cloud computing can get expensive. If you're spending too much money on your cloud infrastructure, check out Dolt International. Dolt International helps startups optimize the cost of their workloads across Google Cloud and AWS so that the startups can spend more time building their new software and less time reducing their cost.

Dolt international helps clients optimize their costs, and if your cloud bill is over \$10,000 per month, you can get a free cost optimization assessment by going to D-O-I-T-I-N-T-L.com/sedaily. That's a D-O-I-T-I-N-T-L.com/sedaily. This assessment will show you how you can save money on your cloud, and Dolt International is offering it to our listeners for free. They normally charge \$5,000 for this assessment, but Dolt International is offering it free to listeners of the show with more than \$10,000 in monthly spend. If you don't know whether or not you're spending \$10,000, if your company is that big, there's a good chance you're spending \$10,000. So maybe go ask somebody else in the finance department.

Dolt International is a company that's made up of experts in cloud engineering and optimization. They can help you run your infrastructure more efficiently by helping you use commitments, spot instances, rightsizing and unique purchasing techniques. This to me sounds extremely domain specific. So it makes sense to me from that perspective to hire a team of people who can help you figure out how to implement these techniques.

Dolt International can help you write more efficient code. They can help you build more efficient infrastructure. They also have their own custom software that they've written, which is a complete cost optimization platform for Google cloud, and that's available at reoptimize.io as a free service if you want check out what DoIT International is capable of building.

Dolt International are experts in cloud cost optimization, and if you're spending more than \$10,000, you can get a free assessment by going to D-O-I-T-I-N-T-L.com/sedaily and see how much money you can save on your cloud deployment.

[INTERVIEW]

[0:04:21.1] JM: Pat Kua, welcome to Software Engineering Daily.

[0:04:23.2] PK: Hi, thanks for having me.

[0:04:25.5] JM: You are the CTO at N26, which is a bank, and I'm looking forward to discussing your perspective on how modern banks work, how a new bank works. I think we all have traditional experience with the interface of banking software and people who have used banks

other than newer banks like N26. They probably have some fairly low expectations for the software that they interact with.

If I think about the technology that has evolved in terms of the intersection of banking and software, I think about depositing a check from my phone. That's about as novel as an experience as we've gotten, despite all these innovation in software and other areas of the world.

Why has the banking experience lagged behind other technological updates, like movie watching for Netflix, or the taxi experience with Uber?

[0:05:24.1] PK: Yeah, I think it's really great observation that you have there, and that banking historically around the world has been very sort of under invested in terms of technology. I think part of the – There are probably two constraints why we don't see things that we see with Netflix or, say, Uber in banking, and I think one of them is definitely something about regulation.

So I think you probably see some of this as well in other heavily regulated industries where people see regulations as a big barrier to entry, and I think this is one of the perceptions that I think maybe we'll get to a little bit later about actually. I don't think it is actually a big barrier to entry if you really understand what it is that you need to do and what controls or risks that you are trying to mitigate and you have sensible practices in there, and I think that's one of the reasons why we have gone, N26, at sort of trying to tackle this because we're also trying to take these regulations and look at the minimum, more modern way and instead of being fearful of them and see them as what is typically seen as excess overhead in terms of paperwork or administration, we're trying to look at ways at how do we build that into common every day technical practices that are very common in other parts of the industry.

I think one of the other sides to kind of the banking stuff is kind of – Well, I mean, regulations keep the entry to market very high, and I think this means that existing banks haven't really had a reason to innovate and they haven't had a competition to sort of stimulate them. So I think banks have been around for quite a while, or little banks have been around for quite a while, and it's also a kind of consequence of legacy and that a lot of bank are sort of may be run by mainframe sort of systems built up in maybe the 70s or the 80s. So all the processes and layers

of people in terms of how they actually build software or how they run their processes, and it's very hard to maybe shift all of the organization towards more modern practices, which is what the benefits of newer companies, like Netflix or Uber have and what I feel that we also have because we get to challenge everything from the ground up.

So one of our values in our company is this idea about simplicity. So I'm pretty sure that you've probably had a very similar experience signing up for your own bank when you sign up for an account. There's a very, very heavy document terms and conditions that you sort of leave through, and we really try to apply this idea or simplicity in all aspects. So even in our sort of sign-up experience, so people can go from nothing to being a customer in eight minutes on their mobile phone.

Terms and conditions, even our lawyers have really gone into looking at how do we clarify normally a very, very complex terms and conditions set and really try to simplify it down into a very easy to use understandable experience, particularly in the day and age of being mobile? So I think there are a couple of the big factors, which I think make a big difference as to why I think traditional banks haven't really innovated, and this is where we're trying to make a bit of a difference.

[0:08:27.0] JM: The business opportunity of a bank, even at just the core level of how a bank makes money is quite tremendous. So I think the typical way that a bank makes money is you have people who deposit their money in the bank. It's a safe place to keep it and use it as a way to transact with other people and would interface with their credit card system, and then the bank can loan out that money and can make some money loaning it out. It's a very good business. Banks have made good money doing that throughout the years.

But from a technology perspective, there're a whole lot of other opportunities that you can build on top of that system. Once you have a lot of transactions flowing through a technology stack, there's lots of opportunities for leverage and expansion. I think these other kinds of businesses are more often categorized in the fintech space of categorization. Do you see a delineation between banks, like new banks, like N26 and fintech companies?

[0:09:32.7] PK: Not necessarily. I think I struggle a little bit with the idea of fintech as a whole, because it's so broad. So I think it's quite interesting because I've met quite a lot of other people from sort of fintech companies, and anything that has to do with sort of money or anything that's maybe technology that's related to money ends up being classified as fintech.

So everything from looking at insurance, everything from managing stocks, everything from even sort of budgeting, personal budgeting gets classified as fintech. Then you have the banking sector around where you manage your own money and what you do with that money, and then you have other sort of fintech companies sort of specializing in sort of payments or payment methodologies. So you have a lot of companies trying to offer more intuitive ways of making transactions. So QR codes or others sort of payment things, like Square.

I think because finance or money ends up being kind of being quite broad and applicable in lots of different areas, I think the fintech space is actually really, really broad. So I actually think the delineation is very difficult between, say, fintech and a bank like N26, because it does start to blur a little bit. So there are companies where maybe a traditional payment company that's setting to move into providing international exchange accounts, are they now a bank or are they originally sort of a payments sort of company. I think is getting blurrier and blurrier.

I think maybe going back to your sort of first question around why banks don't really innovate, is that now, I think everyone has to innovate, because there is so much of this blurring between sort of boundaries of what does it mean to actually deal with money and transact there in a more digital era.

[0:11:20.5] JM: And from the business point of view, is N26 focused on building services and infrastructure just around that core classical banking business of serving customers that are storing their money there and transacting with that money, and then is N26 just doing the classic sort of make money off of the float of that storage of money or are you already thinking about expanding into more of the fintech services?

[0:11:50.1] PK: Yeah, definitely. So I think there're a couple of different interesting things to how we look at monetization. So I think one of the interesting things is we have a much lower cost

space than a lot of traditional banks. So we don't have physical outlets like a normal branch, and this means that we can actually scale up our organization much, much more effectively.

So when we onboard a new customer, we use the same banking platform that we've built to add them on without having to establish new branch and then sort of manage more people from that perspective. So I think the fact that we're a purely digital first kind of bank actually allows us to scale our cost basis a lot greater.

I think also the fact that we've built more modern banking sort of platform means that we're not tied to legacy mainframe systems or legacy systems, which means that once again the overall cost of making change is a lot lower because we can actually innovate a lot more. What that means is that we can actually pass those savings on to our customers much more effectively.

So in Europe, it's very common to pay for a normal standard sort of savings or your base sort of cash account, and because of our very low cost basis, we can actually offer that sort of for free and we can sell offering those services. What we really want to focus on is very much this idea of similar to more modern companies as well, is that we want to provide services that people are really willing to pay for.

As an example, one of our more premium products is our sort of black card, and with that black card you get sort of free travel insurance and you get better sort of international rates of sort of you get the exchange sort of rate when traveling, as well as all the normal sort of basic account, things that are made an offer through our normal sort of savings account.

So what we are really trying to focus on is really provide differing sort of value added products that then people are then willing to sort of pay for. So, of course, like a normal bank, there are also sort of loan type products of which you make some money off. But what we're really focused on is really providing an excellent customer experience in a simple way and then we hope the customers value that enough that they actually want to pay for the extra value add products on top of that.

I think that's surprised me about this kind of digital product is being a sort of more digital native person myself. We still are amazed at the number of people that also want something physical.

So we actually launched the first metal card that's contactless here in Europe, and that's our sort of premium flagship sort of product and it is quite amazing, because I think people like it not only just because of the packaging of services that we offer with that, but also because it's a hardware thing, which is something that I find really interesting in today's age being very, very digital.

[0:14:32.3] JM: I wanted to start with this high level discussion of the business, because I think that informs the structure of the engineering organization as well as some of the decisions in the engineering culture. Can you give me a high level view of the engineering stack at N26?

[0:14:52.5] PK: Yeah. So we're a little bit unusual for our European bank, because we're actually in the cloud, and I think that has pros and cons and that we'll definitely probably observe a lot closer by our regulators in auditors, because it's a bit of an unknown quantity. At the same time, it gives us a lot more flexibility as all those people who've migrated to the cloud understand the benefits of more sort of flexible infrastructure without having to manage the actual physical nature of that.

So I think understanding that we're in the cloud is actually a big thing. That allows us to maybe use different approaches to how we actually build. So we're mostly JVM sort of stack. So we've been mostly based on Java. We're actually moving toward [inaudible 0:15:35.9]. We did some exploration last year around migrating to this that we think will actually help us move towards more flexible and sort of productive kind of teams.

We're sort of a mix of – We have a couple of smaller monoliths may be for people sort of tech stack, but we've also got a collection of micro services. So a lot about micro services leverage things like, Spring Boot and we build them using sort of Maven and Jenkins. So these are very typical set of tools that I think everyone is expected to have.

We have infrastructure as code as well. So because we're in the cloud and we want repeatability and reliability, we manage our service like cattle and not pets. So we use a lot of infrastructure as code to make sure that we can replicate all of our different sort of areas. So we are big fans of HashiCorp. So we use Nomad and SaltStack to provision and run infrastructure.

We use console full service discovery and configuration management, and Vault for managing secrets.

Now, like all sorts of errors of today's deployments and micro services, we are also a big fan of Docker, and we use that for our deployments. So we have a couple of internal applications that are actually built on sort of JavaScript environment. So these are more of some of our internal tooling. Now you also get packaged up in terms of Docker containers and deployed and then we use a lot of sort of centralized sort of logging through ELK, and there are a lot of the sort of tools that we use for building the services and the sort of banking platform that we have.

What I'm really excited by about the team that we've built, is that we have a really strong dev secops culture embedded into the team. So the idea of like continuous delivery is pretty much a given of the way that we actually work, and I think that's a really interesting thing in a bank, because having consulted for banks in the UK, a lot of people have this perception that things move slowly in a bank. So people are lucky if they can deploy once every three months, maybe once every six months into a normal bank because of all the sort of controls.

I think one of the things that we were very careful to sort of put in very early on was this rigorous repeatable process that's automated and strictly controlled, and this actually allows us to do a lot of different sort of changes to our sort of environment really rapidly.

So, as an example, I think last week we did over 900 bills of various different services and we hit 177 live deployments into production. It kind of gives you a shape of the pace of how quickly we move our services, and I think it's a really exciting tech stack to actually work on and to be able to evolve the sort of product from that side.

[0:18:12.6] JM: The point you lead with that the cloud provider, you could maintain compliance despite using the cloud provider. If I understand correctly, that's kind of a new thing. I think some of the cloud providers have been working steadily to get compliant, HIPAA compliance, whatever other flavors of compliance you need to have a bank. Was that one of the barriers perhaps to – When we try to think about, “Why wasn't a bank started earlier?” Why did we get these companies in the sort of web 2.0 phase? We got the Ubers and the Airbnbs. It seems like

a little bit earlier, then we started to get the digital banks. Was it about this compliance layer in the cloud?

[0:18:58.7] PK: I can imagine it would be. So I think actually a lot of the major cloud providers all have all types of compliance certification. So I think this is one of the things where if you talk about R, AWS, or Google Cloud or Azure PCI Compliance, they pretty much all are asking for ISO 27001 certificate. I'm pretty sure most of the cloud vendors would probably have all of those standard ones.

Well, I'm not so familiar with HIPAA. I think it's an American standard for health, but I'm not so sure.

[0:19:28.7] JM: Yeah. That's right.

[0:19:30.7] PK: But I think the cloud vendors probably early on also realized, actually, for them to give safety to other regulators, they also needed to be able to guarantee and demonstrate all of these things and controls to a regulator. So maybe that was a natural thing of when they first started. It was more unregulated kind of places where they did that and realized, actually, the things, the good practices that you have to do at scale, you have to have strong repeatable processes, right? And all the things about regulation and controls are about making sure that you have these documented and repeatable processes that are very sensible.

I think the only way you can scale is actually by having very good sort of processing controls, and that's all that sort of a lot of regulators ask and then it's just about making sure that you understand which controls are mapping to which risk factors that they're trying to mitigate according to which standards and in which way.

Yeah. I guess the short answer is, yeah, I think a lot of the maybe more regulated industries didn't go on to the cloud because those providers didn't have certificates to demonstrate all of these – Meeting these standards. But I think that's a very different story today, because, I mean, cloud technology has been around for a long time now and I think the practices of managing the internals to that are very mature now and they've passed through all of that and that big

providers can provide a lot of these documentations for the things that you need to demonstrate to regulators and auditors.

[SPONSOR MESSAGE]

[0:21:04.8] JM: At Software Engineering Daily, we have a web, we have an iOS app, an android app and a backend that serves all of these frontends. Our code has a lot of surface area and we need visibility into problems that occur across all of these different surfaces. When a user's mobile app crashes while playing a podcast or reading an article, Airbrake alerts us in real time and gives us the diagnostics that let us identify and fix the problem in minutes instead of hours.

Check out airbrake.io/sedaily to start monitoring your apps free for 30 days. Set up takes only a few minutes. There's no complicated configuration needed, and Airbrake integrates with all of your communication tools from Slack, to GitHub, to JIRA, and it enhances your current workflow rather than disrupting it.

You can try out Airbrake today airbrake.io/sedaily. If you want to monitor and get visibility into the problems that may be occurring across your application, check out Airbrake at airbrake.io/sedaily. Thank you to Airbrake.

[INTERVIEW CONTINUED]

[0:22:24.8] JM: You're a CTO. You are the CTO at N26, and you've already described a sequence of products that are being worked on. There're multiple products. You've also defined a well-crafted engineering culture where you can release frequently. It sounds like you've got a platform, kind of a consistent platform across the company or you're making your way towards having a consistent platform across the company so that deploying services in one area of the company isn't remarkably different than in another area of the company.

A CTO could work on many different things within a company in a given day. You could work on this platform layer. You could work on figuring out what blockers any of these individual products have, whether we're talking about the metallic credit card, the contactless credit card, or the loan services or anything. You could be working on hiring.

How do you structure the time allocation of the CTO role? What are your personal KPIs and how do you determine what to focus on in a given day?

[0:23:38.8] PK: Yeah. It's a really good question, and I think maybe to answer it properly, it probably makes sense to explain a little bit more of the context of how our organization is sort of set up. So we have a CEO who's one of the founders, and he helps drive the direction of where we go as a company strategically, and that naturally aligns to sort of product and market launch type of initiatives.

We also have a chief product officer, and both the chief product officer and myself report in to the CEO, and we kind of work very closely together. So we're not a very old company by banking standards, but one of the things that I was very conscious of doing when I came into the role was avoiding what a lot of other companies have, which is a very clear – Or not a very clear, but a very wide gap between products thinking and technical thinking.

So when I came in, the teams that we have are actually sort of cross functional in nature of how they work. So we've really gone down the path of trying to have product or domain driven teams and organization, and then we have these kind of cross functional skills to optimize for flow so that we can make sure that we get fast feedback loops between products and user experience and what's technically difficult to what's easy to operate. We can optimize around that.

So the CPO and myself work really closely at making sure that we can deliver the product roadmap about where we want to go. So we work very closely on both product idea, but also than implementing and making those things happen in reality.

So, as I said, we're not very old company. So we've been around now for about three years as a bank, and we are still a sort of startup. So like everything, we're growing quite rapidly. So I joined in September last year. We had 450,000 users. We're now a million users in Europe. Looking to expand into the UK and into the US at some point in the future, and there's lots of things that we would like to do, not only iteration over existing products, but also new product initiatives.

So, for example, we recently – And maybe this isn't a surprise for people in the US, but in Germany we haven't been able to release Google Pay, because it's one of those constraints where Google Pay as a system wasn't enabled yet. But we were the first bank in Europe to go live with them, because it's one of those features we know that a lot of users want to actually have.

So a lot of my time is actually spent obviously hiring. But one of the things that I really want to make sure is that we just don't hire for the sake of hiring, but we understand how teams are most effectively going to work, and it's quite different depending on which scale of an organization that you're at. So when I first came into the role late last year, it was looking at where were we now and what are the other things that we wanted to do from a product roadmap and what are the shape of our organizations that we actually want to have as we go forward?

As you can imagine, having a 30 person development team is quite different from saying having a 150 person development team, and you need to start to work at how you split down things into maximizing autonomy, but also making sure that you still have good alignment. So I always say everyone can have complete autonomy, but then it gets like a bit messy when somebody's autonomy steps on to somebody else's autonomy. This is where it's quite important where I've been looking at the structures of the organization of how we work and have been sort of bringing in new capabilities.

So part of that is actually a process of hiring and interviewing and sort of attracting people. Another side of my role is really looking at sort of the technical landscape and sort of trying to understand where we're going and try to make sure that everyone on the team understands where we go.

So what you were talking about, a sort of consistent banking platform is one of those things where we want to sort of make it as user-friendly as possible to engineers and to teams. So they want to continue basically building on top of it, not something that they are sort of forced to do because they're told to do, but because they see the value in actually doing that and they can focus on building the product on top of that.

So a lot of that is sort of looking at this overall architecture, looking at sort of feedback loops and sort of supporting different projects at different points on where they are. Sometimes my days do not look very regular. Sometimes there's a bit of a deep dive into a particular project or a product. Sometimes it's really about sort of working across sort of the roles that I've brought in and making sure that everyone has a clear enough context about what they're doing and also trying to work at how do we unblock some of the things in the environment that are preventing people from doing the best things that they can.

I think one of the things that I'm particularly passionate about, and I've written a book around this, it is called *Talking with Tech Leads*, is really about building sort of technical leadership skills in engineers as they grow up. So anybody who's been on sort of a tech lead track, some terms called it lead developer, and some organizations you might call them sort of an architect. They'll will start to realize, actually, to be successful, it's quite different to the skills and capabilities that you need from being a successful engineer.

So resolving conflicts between engineers, trying to understand what are the system quality attribute to trying to build in and design and make explicit the trade-offs around the, and that's also one of the things that I spent a lot of time in my organization doing. We've actually kicked off sort of a tech lead development program for sort of what I say is the next generation of people who are on that cusp where they're really great sort of engineers that can sort of build small services, but then as we build a sort of team around them, we need to support them in that transition rather than what a lot of other companies do and sort of throw them in the deep end and say, "Good luck. Have fun leading people," without setting expectations around what is and also giving people the tools around that.

On other days, it's really sort of working with the product side and working with the bank as well. So I try to make sure that we get the right, I guess, documentation in place that we need to demonstrate to auditors. Yeah, it's quite a blur sometimes, but it's an exciting growth stage and it's an exciting time to join a company that's growing so rapidly.

[0:29:44.4] JM: I want to dive deeper into this idea of the tech lead and how that is manifesting at N26 for you. So what I heard you describe was a situation where you are frequently interacting with the CEO and this, I think, CPO you said. Is that chief product officer?

[0:30:02.1] PK: Yup! Chief product officer. Yup.

[0:30:04.2] JM: Right. So you and the CPO and the CEO can sit around a table and talk about what are the products that we're trying to build? What are the challenges we're trying to overcome? Then you can use that information and that integration with the product officer and the executive officer and use it to craft how you're spending your time and the energy that you're putting into different parts of the company and perhaps the KPIs, or the OKRs that you are determining for different areas of the company, in different products within the company. That way, the information can trickle down and can empower people to understand how they are affecting the overall moving of the ball forward for the entire company.

What I also here there is that you have the ability to move between the product thinking, the business thinking, and the engineering thinking, and that sounds like a flavor of the tech leadership. It's almost like – It's like with the whole devops thing. I think the idea of devops is like, "Let's break down the silos between development and operations."

But if you take that a step further, you actually need to break down silos between development, and operations, and accounting, and design, and platform engineering, and basically different organizations, different members of the organization need to be comfortable interacting with each other. I mean, you don't need to be an expert in everything, but you need to have some familiarity and ability to interface with the rest of the company. Is that how you see tech leadership? Is it kind of like the erosion of this highly defined bounded context for what an engineer within the organization is doing?

[0:31:52.0] PK: Yeah, absolutely. I mean, I think what you talked about there, of this blurring or sort of boundaries, is exactly sort of the culture that we've sort of cultivated. So we have sort of cross functional teams. We have sort of a core team. That's kind of our infrastructure, and they have security and they work with security champions and teams. The idea is really about that sort of spreading of knowledge and holistic ownership.

So one of the key messages that I often sort of repeat is this idea that anyone can be a technical leader. As you say, it's like one of those things about how do you blur the boundaries

of – You're saying that your role is not just about the code and the software, but actually understanding how does that software solve a person's problem from a user experience perspective? How does that meet some need from the business around either helping more customers come aboard or how does that help sort of the product direction? All the way to the back office and the operations side of the bank of how do we design something so that it's not good to be a problem for us sort of back-office people, or it will be easier where won't create sort of failure demand on the system.

Then also the sort of technical operations side of making sure the we have good alerting in place, and I think that's exactly the type of technical leadership of people sort of thinking beyond their own task and actually thinking more holistically in terms of the value chain. I think what you've kind of observed or heard from my description is I try to do that at a higher level, which is looking at sort of student between departments or between sort of functional areas to really try to optimize the organization for sort of lower value around different areas.

Now, obviously, our company is too big for me to be able to lead specific architecture products, and I see my role as a leader explicitly about also developing people and growing people to make sure that they take care of that. For me, that's a really key thing about good leader, is that they act as multipliers.

So one of the big challenges, and I gave a talk actually at Lead Dev Austin earlier this year, which is also on, I think, Vimeo or YouTube. I talk about moving this mindset away from being a maker to a multiplier. So this is a big shift of the things is an engineer. Often you get rewarded and you get this personal satisfaction. Sometimes I miss that as an engineer of the – You have this technical problem. You type some things into a computer and then you sort of get is green or red kind of state, it works or it doesn't work.

At a higher level, at a sort of leadership level, anything you do to multiply the sort of output of a team or to multiply other people to be more effective is a successful leadership, and I think that's a really key message I really encourage everyone to sort of take on board and every one of their roles is really thinking about anything that you do to multiply the output or yourself or with other people is an act of technical leadership, and it's that mindsets about how you reward yourself moving away from what you personally do to what have you enabled other people to do even

broader than what they could've done before. That's a big essential part to this sort of mindset shift to being a successful leader, at least in my opinion.

[0:34:52.7] JM: This is very important, this maker versus multiplier idea. It's very important in modern software engineering, I think in part because the number of managed services and products that you can buy off-the-shelf that will solve your problem, it has just exploded. If your job as an engineer in many cases is actually to shop around, it's not like sit down and build the thing that needs to be implemented. It's like shop around and find out who already has built this thing, and like can you take it off-the-shelf and pay 999 a month for it? As supposed to spending two quarters building out a service.

[0:35:33.8] PK: Absolutely, and I think that's a big shift, which is a lot of engineers, they see that and they see this personal satisfaction. These systems sort of get rewarded as well as a lot of places, reward people for the individual contribution, and it's super important. I think as people move into the idea of this leadership mindset. It's just the behavior differences, things that will make you successful in that sort of role.

[0:35:56.8] JM: So the platform itself, I'm curious how well-defined your platform is, or if there are areas of the company where it's more specialized or it's different or it's less homogenous than you would like it to be. How standardized is the process of service creation and service deployment and service monitoring and logging, et cetera, within N26?

[0:36:21.3] PK: I would actually say it's quite standardized. I think it's really quite interesting when I came in, because I think one of the interesting things I observed is that we have a hugely collaborative culture. So I think what's an interesting thing that I haven't seen in a lot of other organizations is people just make decisions for themselves and they go, "That's exactly it," and I think the collaborative culture that we have is one of those things where everyone's really focused on get stuff done, and this is kind of a mentality that we sort of talked about at N26 quite a lot.

I think one of the things that people realize is that there are sudden decisions if we standardize, we going to have these hypothetical arguments about which thing is actually 2% better than this

other solution. It gets rid of a whole bunch of problems and we can focus on building products and customer value, which is exactly what drives a lot of us of sort of engineers.

So, actually, the standardization came through a lot of sort of, I guess, organic agreements between a lot of the sort of technical leads and the technical team at the time, because who really wants to go and decide how they're going to do monitoring and then set it up and then do that if it's easy to just plug in a couple of things that you know you want to be able to monitor and get alerting on. That's exactly how the teams kind of had that approach.

So I think it's a very pragmatic attitude I would say to how we approach technology. Is this something helping us build something for our customers? If not, then maybe we should look at standard solutions that allow us to get rid of a whole host of problems and we don't need to spend time, as you said, spending half a year building this ultimate solution, which will only get out of date. That doesn't really add a lot of value to our customers at the end of the day.

[0:37:56.2] JM: Do you have platform engineering team, or is it just service teams?

[0:38:01.6] PK: Yeah. So we have a couple of funny names, I guess, in our sort of teams. So we have a sort of cross functional teams that own a sort of product area. So as an example, payment space is quite complex, and so we have a couple of teams in that sort of group. They're responsible for that, because the service characteristics are so high on the backend that it makes sense to have sort of dedicated teams around that.

We do have what we call an application platform team. So they're much focused on sort of platform features that would be not unique to a particular product. So as an example, we have a thing that we refer to as certification. So one of the interesting things about mobile is that you can actually give an interesting to user experience, which is your push notification in order to confirm a particular action.

So we use this for instance in implementation of Mastercard 3D Secure. So for those of you who don't know Mastercard 3D Secure, it's often this extra fraud mechanism or fraud protection mechanism that merchants online can enable, and that when you use your Mastercard card and Mastercard thinks that there's a potentially fraudulent transaction, they pop up an extra

dialogue. The most common implementation, which I experience a lot when I lived in the UK before joining N26, is this horrible – It feels like early 90s designed screen. It asks you to create a password. If you have a password, you have to remember your 4th, or 8th or, 12th character and put them in. Nobody ever remembers it. As a result, a lot of people cancel their transaction and don't go through. So a lot of merchants don't do that.

But for us, we basically send a notification to your phone saying, “Okay. We think there's a fraudulent transaction. Please confirm or certify that this is you,” and we make the most of simplifying that experience just through the mobile experience and pushing a notification.

So that is that sort of certifying kind of feature is triggered through a couple of different mechanisms, which isn't sort of a product specific thing, and that belongs to our kind of platform team. So they own I guess what you would call generic application platform features. Then we actually have what we call a core team, which in other places might be called the platform team. This is where we have a lot of the sort of, I guess, more of the infrastructure automation skills and capabilities working there to sort of help build out some of our infrastructure as coded and server management so that people can operate at a different level.

So they're kind of responsible for managing things like our continuous integration service, continuous deployment processes. Then they work with teams very closely to build out new, perhaps, deployment pipelines for new services, or to speed up basically new service creation and to also iterate over existing tools to make sure that everyone keeps up to date with our sort of moving infrastructure as we sort of grow.

[0:40:54.3] JM: Do you have a lot of data science going on within N26?

[0:40:59.6] PK: Yeah. So we actually have a fairly good data group. So we have like a whole bunch of sort of data analysts. They help us build out a lot of the reporting that we have. So we have a lot of sort of product metrics to understand how our application gets used. It's like one of those really nice things that you can do if you actually think about it early on and you start to understand what is it that you'd like to measure in terms of product success.

Then they also sit with sort of data engineers who are people who are helping build out some of the sort of data models and sort of put them into our sort of services. Then we do have a number of data scientists who help us work with different elements of our sort of application.

So, for instance, one of our data science products is actually sort of machine learning algorithm that actually sits behind our implementation of Mastercard 3D Secure as well. So what we've sort of built there is some intelligence that allows the model to understand patterns about you and your card and how it gets used. To work how if your card is being consistently challenged in weird places, when to actually popup this Mastercard 3D Secure, or when to say, "Actually, we think this is you and actually we won't even bother triggering that process." So that's exactly one of the areas where we have some of our data scientists at work.

[SPONSOR MESSAGE]

[0:42:21.2] JM: Citus Data can scale your PostgreS database horizontally. For many of you, your PostgreS database is the heart of your application. You chose PostgreS because you trust it. After all, PostgreS is battle tested, trustworthy database software, but are you spending more and more time dealing with scalability issues? Citus distributes your data and your queries across multiple nodes. Are your queries getting slow? Citus can parallelize your SQL queries across multiple nodes dramatically speeding them up and giving you much lower latency.

Are you worried about hitting the limits of single node PostgreS and not being able to grow your app or having to spend your time on database infrastructure instead of creating new features for you application? Available as open source as a database as a service and as enterprise software, Citus makes it simple to shard PostgreS. Go to citusdata.com/sedaily to learn more about how Citus transforms PostgreS into a distributed database. That's citusdata.com/sedaily, citusdata.com/sedaily.

Get back the time that you're spending on database operations. Companies like Algolia, Prosperworks and Cisco are all using Citus so they no longer have to worry about scaling their database. Try it yourself at citusdata.com/sedaily. That's citusdata.com/sedaily. Thank you to Citus Data for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:44:06.2] JM: Have you felt a need to aggressively hire machine learning developers or more data scientists? How are you thinking about the development of your data teams?

[0:44:19.3] PK: So I think proportionally wise, we actually have a pretty good number or sort of data scientists. So for our sort of group, we've got sort of three to four 4 different data scientists, and what happens is they kind of end up sitting down and working closely with a product team. So if we have a product where we think there needs to be some sort of machine learning or some sort of data science kind of project, this sort of collaborative dynamic teaming is kind of a characteristic of how we kind of work.

So they'll sit down with the product team and then agree on exactly what is it that they'd like to maybe do? What maybe approach or training model would actually make sense? Then iterate to actually over productionizing that with a different development team.

So it works quite well. For us, it doesn't really make sense to have like data science everyone. Not yet. But it's really very focused on where we think the most practical uses, but we have very good sort of collaboration. That means that we now – Well, people are made aware very early on as to whether this thing would be a good place to have some data scientist work a bit closer to help us improve the experience.

[0:45:19.7] JM: Yeah. So that's exactly what I was thinking about, is you've still got probably a lot of greenfield, just like feature development application, basic core application development, product iteration to go before the machine learning side of things can really be higher leveraged in that, because the machine learning side of things is more of like optimization over products that have already been proven and developed significant datasets.

[0:45:45.9] PK: Yeah. I mean, at the same time, we have a very large user base of more than a million people and we've processed more than sort of billion euros a month. So there's actually quite a lot of data in there. I mean, we do use it in some simple examples. So, for example, when you make a transaction, sort of transaction list, when it appeared, we have some classifiers that try to classify and give a hash tag as to what it is.

This ends up being used in terms of statistics that we demonstrate as to your monthly spend, and for instance, one of the interesting things that we've used data science for, is that often what happens with, say, if you make a transaction or a payment somewhere and then it gets refunded, that refund actually appears maybe two, three, depends on the merchant a few days later. We also try to connect these different transactions in the application.

So there's some interesting intelligent there about saying about the sort of merchant, the amounts and connecting them. So that in the application, when you open up your transaction list, you can say, "Oh, yeah! Why did I pay that? Oh, no that got refunded," and those two things are actually connected rather than a sort of boring transaction list where you see it much, much further and then you have to make that connection yourself.

So we actually have – Where we have large data models, where it makes sense. Once again, it goes back to our sort of philosophy of a simple user experience and really trying to understand how do we use that? Where it makes sense given the data that we have to make that a really good user experience.

[0:47:08.5] JM: Yeah. That's labeling of the transactions going through a system. That seems like an unsolved problem. There is not yet an API that you can call on a user transaction to get it canonically classified. The reason I know this is because I use it – Not to knock on QuickBooks, but I use QuickBooks, and it does not classify things correctly. I'm like looking at it, it's like X-dollars spent on hospitality, like [inaudible 0:47:36.7] hospitality, and it's like Amazon Web Services, like, "Oh! I don't really call that hospitality," but, yeah. I don't know. It will be nice when that is a solved problem.

[0:47:46.8] PK: Yeah, and I think maybe it's not so known from a lot of sort of users. But with our transactions, you can actually sort of reclassify and retag them. We also have machine learning there. So if it learns for you personally that this thing isn't actually sort of a maybe food expense, but this thing is actually like a household expense, then it'll reclassify it for you.

So there's also not just general learning at the sort of application level, but also the user level. Once again, focused on the whole idea of trying to make the user experience as good as possible.

[0:48:18.1] JM: So as far as the payment system, the modern payment system, as I have interfaced with it, it's typically – It moves pretty slow and I'm not sure how much of this is because of transactions need to be somewhat reversible, or because accounts could potentially be hijacked. Is the slow speed of payments that move through the banking system, is that a feature or a bug?

[0:48:43.0] PK: I think it's a bug to be honest. I think the payments, it's actually quite interesting, because I think a lot of this is actually probably driven by banks and that the sort of history and the legacy. So if anyone is listening, has worked for a bank, you probably all know about batches, and batches are the way that banks work. Well, typical banks kind of work.

So, for us, it's actually kind of interesting, because we're actually real-time banks. We move money pretty much instantaneously. But if you imagine back in the sort of 80s and 90s, is that people would shut down their branches. They would like do the calculations and then there would be this sort of overnight sort of, "Quick, we need to do books," and then sort of then ship it off. I think, historically, that's just the legacy problem of how the banking industry worked.

We know this in the US, where it was sort of exploring, that it's still this pretty much like batch base system. In the UK, it's a little bit different, because they actually have two, say, payment schemes. One which is called BACS, and the other one which is called FPS, so faster payment scheme. Actually, it was an initiative because of this problem from a user experience, and I am not sure when it came in, but it's been around for a while, is that banks had to – Towards their industry regulation change, and that banks had to adhere to the faster payment scheme to offer instantaneous transfers. I think it was exactly the same sort of experience for you.

In the UK, it would take three days for magic to magically move from one number account to another number account because of legacy reasons. In the UK, at least, they sell that through FPS to avoid this kind of batching. So it's one of those things where, once again, we benefited from really thinking about how do we make sure we avoid batching and we really do real time

per transaction, because that's exactly the experience that customers expect and that's the experience we want to provide them.

[0:50:27.2] JM: What's the most surprising thing that you've learned about banking technology?

[0:50:31.9] PK: I think the thing that surprised about banking technology is I think people are a lot more scared about compliance and regulation than they should be. So I think it's quite interesting, because our head of sort of systems engineering, he's helped us sort of rollout infrastructures code in this kind of sort of process, and I think what I've learned is that any time that you go through an audit, is that these things I consider as sort of like a QA for your entire organization around processes.

So just like how QA's in sort of software development have moved away from being sort of the end of the line, and we really try to think about pushing quality all the way at the front. Quality sort of assurance today is very much about sort of [inaudible 0:51:12.1] testing, finding the edge cases and finding where your weaknesses are. I think that's exactly one thing where we don't want to ever be afraid of what our auditors have to say, because we should already be knowing that actually we meet all of these things and we have all of these sort of process in place.

So I think that's the thing, is that I think most people, when they expect what for a digital bank, they think, "Oh, no. These things can be really heavy weight and get in the way." But one of the things that I worked very closely with with the rest of the organization is really think about how do we make sure we keep modern approaches, agile software development, continuous delivery, infrastructure as code. All these sort of site reliability engineering practices alive, because all of these are actually controls that help us with meeting compliance and regulation, and people just have to be brave enough to talk about, "Okay. What you've asked for, here's how we do it in today's world." Actually, this is repeatable and will automated as well and we shouldn't be afraid of what compliance and regulators have and they shouldn't also necessarily slow us down. It's just we have to sort of bridge that understanding between what they want to mitigate risk. We have different ways of mitigating risk today that don't necessarily involve meetings, and signoff, and checklist and documentation.

[0:52:26.8] JM: So a lot of people listen to this show because they like to get ideas about technology that they could potentially build. I think there's a lot of people that listen that are potentially thinking about maybe starting a company at some point in the future, and I've been doing a lot of shows about fintech recently, and one thing I find interesting about fintech is in contrast to an area like the sharing economy.

So if you look at the sharing economy, Uber came out and then a bunch of people were like, "Okay. I'm going to build Uber for X," whether it's Uber for dog care or Uber for dishwashing or something like that. There really weren't a whole lot of Uber for X opportunities that were actually successful big opportunities. There were a bunch of them, but probably not as many as people initially thought.

But with fintech, when I look at fintech as a sector, it seems like if you're an entrepreneur, if you are just kind of looking for a space to get into and where you can just study it and then eventually find an opportunity, fintech seems pretty ripe, that there's just like an abundance of opportunity. Am I mistaken there or is there really just like a complete abundance of opportunities, and if you just start looking into the industry you see tons of opportunities?

[0:53:39.1] PK: I'm pretty sure there is. I mean, I think, as you said, it's like one of those things where maybe traditional finance environments, there hasn't been a lot of innovation particularly thinking about how do you apply modern approaches or modern things to innovation. As an example, I have an ex-colleague of mine who started up a fintech insurance company in London called Risk, and this is really about if you think about insurance, that's also a really sort of industry, which there hasn't been a lot of innovation.

I think that's – Like in each of these different areas, I think there's definitely sort of time to understand what it means to sort of innovate around that. I think there's definitely going to be lots of opportunities for people. Yeah, it's an exciting time to sort of be in fintech in general. I know that from sort of a funding perspective, there's lots of sort of money, because I think investors are also – They're seeing that there's a lot more attention and there's going to be some really good ideas about some extra value that can be unlocked, because it's never been approached in this way using sort of modern interfaces or modern technology.

[0:54:40.5] JM: This is your first CTO job, as far as I know. What has been the biggest challenge of being a CTO?

[0:54:47.9] PK: Yeah. I've been working in the industry for about 16 years, 14 of those years I was a consultant at ThoughtWorks, and I think probably one of the biggest challenges was moving away from consulting. So I think there are sort of pros and cons to consulting. One of the interesting things about it is that you get to see lots of different industries over a sort of fairly short amount of time compared to other people in sort of career.

So I think one of the things that I found challenging is kind of missing a little bit about seeing and hearing what other people are doing in different industries, and I think that's actually where a lot of innovation comes from as you can see different ideas from one place into another and there's a lot of value in that sort of consulting.

On the flipside, I think, for me, I also get to sort of see and I get a lot of reward from seeing some of the things that I start to actually pay off as well, and I think that's the really rewarding side to moving away from a consulting role into a more permanent position and an initial management position, is that you I don't necessarily need to convince anyone or to influence anyone and to think these ideas are good. I need to get my team onboard, but once they're onboard and if they're completely onboard, then we just make it happen. So we can actually move really rapidly, because we have people who want to move in that direction as well.

So in terms of the sort of challenging, I think it's a little bit of a different experience of being in a single place for a lot longer. Also, trying not to maybe intermingle this difference between sort of your role, where you have authority and your opinions. I think this is something that, as a consultant, really rely heavily on your influencing to try to get people onboard with your sort of ideas. In my management role, I'm at the top of the food chain from a sort of technology perspective. I have to be really careful about maybe saying things that I would've said as my opinion, and I don't want people to simply say yes because it's my role in the company, but I want them to say yes because they think it's a good idea.

So there's a natural tension where I'm maybe a little bit more held back in my ideas where I try to ask for everyone else's opinions before I present mine. To make sure that just the fact that I'm

a CTO doesn't influence and sort of close out opinions or valuable ideas that would actually be really useful, and I think that's a really interesting balance too to find.

[0:57:00.7] JM: Yeah. Well, I mean, I can relate to the appeal of the consultant, because [inaudible 0:57:05.2] somewhat similar to how I operate. Just kind of like interviewing people from different companies and never going deeper than an hours' worth of surveying of the technology, which is fun, but it's also what am I really developing a deep understanding of? I mean, podcasting, that's for sure. Software in general? Maybe. But I think probably similar to what you're discovering at N26, is there is a depth of understanding a certain domain and how software works in that domain that probably is very fulfilling.

[0:57:39.4] PK: Yeah, absolutely. I think that's a real key thing, which is, once again, being able to learn the domain and to apply my sort of experiences from before and then combining that and also being there to sort of see those ideas actually materialize. So I think one of the things that you learn in any sort of leadership role is that you kind of like initiate things and it takes a while for those things to happen. As a consultant, you're not normally around to see all of them come up. Some of them to pay out and some of them do. I've been in my role already long enough to actually see a lot of these sorts of things come to fruition. I have to say, it's really, really fulfilling as well.

[0:58:15.5] JM: Pat Kua, thanks for coming on Software Engineering Daily . It's been a real pleasure talking to you.

[0:58:19.7] PK: Thanks very much for having me. It's been a pleasure talking to you as well.

[END OF INTERVIEW]

[0:58:25.2] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use, and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plug-ins. Use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on-the-fly. GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on GoCD with the new Kubernetes integrations. You can check it out for yourself at gocd.org/sedaily.

Thank you so much to ThoughtWorks for being a longtime sponsor of Software Engineering Daily. We are proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]