

EPISODE 731

[INTRODUCTION]

[00:00:00] JM: A bank account is a platform for apps to be built on top of. If that sounds like a weird idea, think about the features of a bank account. Most users only have a single bank account, making it a great tool for identity and authentication. The series of transactions in a bank account provides a data set that can be used for analyzing payment history and issuing loans or insurance, but there are difficulties to building a platform on top of banking. There are thousands of different banks and if you want to build an application that integrates with a user's bank, you need to be able to integrate with any potential bank that the user might use. Whether it's Bank of America, Wells Fargo or Chase.

Plaid is a company that builds APIs for users to connect to banks. Applications such as Venmo, Betterment and Coinbase use Plaid to connect with the bank accounts of their users. Jean-Denis Greze joins the show to explain how applications use Plaid and how Plaid has scaled its infrastructure to handle a high-volume of requests. Jean-Denis also discusses the potential of banking as a platform and the strategy for expanding the APIs that Plaid can offer to developers.

I want to mention that Fintech Daily is a new podcast from Software Engineering Daily covering payments, and cryptocurrencies, and trading and the intersection of finance and technology. Fintech Daily is looking for new volunteer hosts for Fintech Daily. It's currently an un-monetized podcast. We're just building some steam with it. So if you're interested in working with us to conduct interviews, you can send an email to host@fintechdaily.co. We take care of much of the frustration of doing a podcast, of hosting podcast. We will teach you to podcast if you are technical enough to do a show on Fintech Daily and you are reliable enough to produce that show.

If you want to find out more, you can send us an email to host@fintechdaily.co and you can find the podcast on iTunes, Google, and everywhere else if you're interested in listening to it. Don't hesitate to reach out. We'd love to get some new hosts. With that, let's get on with this episode of Software Engineering Daily.

[SPONSOR MESSAGE]

[00:02:30] JM: We are running an experiment to find out if Software Engineering Daily listeners are above average engineers. At triplebyte.com/sedaily, you can take a quiz to help us gather data. I took the quiz and it covered a wide range of topics; general programming ability, a little security, a little system design. It was a nice short test to measure how my practical engineering skills have changed since I started this podcast. I will admit that though I've gotten better at talking about software engineering, I have definitely gotten worse at actually writing code and doing software engineering myself.

But if you want to check out that quiz yourself, you can help us gather data and take that quiz at triplebyte.com/sedaily. We have been running this experiment for a few weeks and I'm happy to report that Software Engineering Daily listeners are absolutely crushing it so far. Triplebyte has told me that everyone who has taken the test on average is three times more likely to be in their top bracket of quiz course.

If you're looking for a job, Triplebyte is a great place to start your search. It fast tracks you at hundreds of top tech companies. Triplebyte takes engineers seriously and does not waste their time, which is what I try to do with Software Engineering Daily myself, and I recommend checking out triplebyte.com/sedaily. That's T-R-I-P-L-E-B-Y-T-E.com/sedaily. Triplebyte, byte as in 8 bits.

Thanks to Triplebyte for being a sponsor of Software Engineering Daily. We appreciate it.

[INTERVIEW CONTINUED]

[00:04:29] JM: Jean-Denis, you are the head of engineering at Plaid. Welcome to Software Engineering Daily.

[00:04:33] JDG: Yeah, thanks a lot for having me, Jeff.

[00:04:34] JM: Plaid is a company that lets developers connect their users with banks and financial institutions. It's a way to connect the user's bank account to an app. What are some use cases when an app would need to connect to a bank account?

[00:04:50] JDG: Sure. Here's the overall picture. So firstly in the United States, the first thing to understand about why Plaid exists is there are 10,000 banks, right? It's not a very concentrated banking market where you just have a few institutions that are serving most of the population. So because of that, there's been just a lot less standardization around even things like how do you transfer money from bank accounts? How do you know that a bank account belongs to somebody? How do you build an app that's centered around understanding someone like spending habit, for example, to give them like financial advice with regards to saving for retirement or whatnot? So you have all these complexity that just stems from how like the lack of concentration in the market.

So the inside behind Plaid from four or five years ago was very simple, just like there's been a lot of innovation in tech, but it hasn't really touched like financials. The reason for that is if you want to build a tech company around like a fin tech company that's building a better product experience around anything that people do with a financial system every day, you need access to someone's spending history and you need to be able to transfer money back and forth.

I think a popular app today might be Robin Hood, that the provides zero fee like brokerage. But to do that, you first need to be able to get money into that app, and to get money into that app, you need to authenticate that a bank account actually belongs to somebody so that it's not a fraudulent transfer of funds. There was no really standard way to do that before Plaid, or if you want to create a better lending experience, like lending club, you need something better than a credit score to determine whether someone can pay back a loan. The way you generally determine that is by figuring out how much do they earn? How do they spend money? Are they like to conservative spender that you feel good making a loan to? But to do that, you need access to like their financial history, like where do they spend? How much do they spend? How much do they earn and so on and so forth. Traditionally, the only party that would have that is someone's primary bank.

The insight behind Plaid is if you make it easy for anybody to have access to that information, I want to say anybody, any app developer, if you make it easy for a consumer to kind of share that information with an app that they want to use, then you'll get all these product innovation on top of the financial system.

[00:06:55] JM: Let's say I'm a user. I'm registering for an app that uses Plaid, like Venmo, for example. What's the process? Describe the process of me signing up for that app and what am I giving the application and how is Plaid operating in the middle to connect my bank account to that app?

[00:07:15] JDG: You're a consumer signing up for Venmo. The whole thing that Venmo does is like allows consumers to send money to one another. You want to like – After you spend dinner, you want to like pay someone back, because s they fronted and paid with their credit card. How do you send them like \$50 to make up for their balance? You sign up for Venmo, first time user, and Venmo needs to know that you have a bank account that they can transfer money like to Venmo so that Venmo can then transfer money to the other user.

The problem for Venmo is they don't know who you are, or you're just a person on the internet. All they have is your phone number. So they need a way to go from that to like you're a real human being. You really have a bank account. The bank account has money and here's how you can transfer the money from the bank account to Venmo. All of those questions don't have easy answer.

Traditionally, in the U.S., you would use an account and routing number to say this is my bank account, but Venmo couldn't operate in a world where you could put in a random account and routing number and then they just take money out of that account. This's just not good enough. They want to make sure, really, you're the person with that account. This is where the Plaid flow plays into it. So the consumer basically is shown a screen where they log in to their bank. They log into their bank account via Plaid, we actually take care of connecting with the bank itself and we authenticate, "Hey, this person's username and password actually allow them to log in to the bank account." The bank account – – The bank gives us the account and routing number that we then pass back to Venmo.

So then Venmo doesn't just allow someone to answer a random account and routing number. They get it from the bank itself via us and that gives them a lot more certainty that it's a real human being. But then there's a lot more stuff that we do on this side. When you open a bank account, you go to a branch and they generally – They verify you're a real human being. They verify your driver's license, your mailing address, your phone number. So what we'll do with Venmo is we'll provide them, for example, this is the phone number on the bank account. Venmo also has your phone number, because when you signed up for the app, you have to do two factor auth where they send you a text message that you responded yes to.

Then they look, do the phone numbers match? That's another thing that helps determine whether the identity of the person is the real one. There is a number of things like that that Plaid enables that allow you to make sure, "Hey, not only is it like a real human being, but this is really their bank account."

Then the last step that we provide is how much money, like what the balance is in the bank account, because obviously you don't want someone to like try to transfer money that they don't have, considering it takes about 2 to 3 days for an ACH transfer to go through. That's kind of the last part of what Plaid does vis-à-vis transferring funds.

These are all very, very simple things. The product experience is extremely important. From the consumer's perspective, all they care about is they're like logging into their bank account and then magically they can use this app called Venmo. But it's a lot of very simple things that just don't exist in a standardized way. No bank makes an easy API available to do any of these. Plaid had to like build this from nothing and not just on top of like 1, 2, 3 or 4 banks, but literally on top of more than 10,000 institutions in the U.S.

[00:10:12] JM: Give me an overview on how that works, because if these banks don't offer an API for users to connect, you have to do some kind of magic behind the scenes in order to give what is essentially a unified API across all of those banks.

[00:10:29] JDG: I mean, that's the magic of the company. So I can only talk about how we do it to a limited extent. I'll tell you that the way it was done years ago was pretty different from how

it's done today. Inasmuch as like today, a lot of the banks actually like want to partner and work with us and will provide us their private APIs or ways to make these work faster.

So I think we announced a few weeks ago kind of a partnership that we have for Chase, for example, which is one of the big three banks in the United States. That makes life just a lot easier. The origins of the company, you have to start with screen scrapers or using banks, like web APIs, intended for consumers and build kind of like a B2B product on top of it. I think a lot of the early magic of Plaid was doing so was painful and it wasn't necessarily the – It's not the kind of work that when you're a CS students like in school that you dream of doing, but we did it at scale. We didn't build like 10,000 individual integrations. We found ways to integrate with banks in a scalable way. We have some integrations that worked with hundreds of banks at a time, for example.

This is probably as much as I can tell you about the details of how it works. The truth is like there are companies that try to do and actually did some of what Plaid did in the past and it took a very like one bank by one bank like manual approach to it. We've kind of taken a very scalable, like a hard engineering approach to it that leads a higher quality product that's been a lot easier to scale to get coverage in the U.S.

[00:11:57] JM: Interesting. I mean, I can imagine – I was just kind of thinking about how you could potentially solve this, and one way is, like you said, maybe you have basically a browser set up on a VM somewhere and then a user makes a request with their bank account credentials. You could literally open a webpage on the VM, go to bankofamerica.com, put in the banking credentials and then scrape the screen and then the user wouldn't really know the difference, except it would just be kind of high latency and it's just kind of a janky way of doing it, but it would get the job done. I wouldn't have any – As a user, I wouldn't have any opposition to that. Banking should be turned into a platform. So I would be happy with that if that got the job done.

But I can imagine that whatever your early approach was and whatever the kind of hacking approach was, as you built some traction and the bank started noticing, “This is great. This is useful for – This allows customers to do additional things with our bank.” I can imagine them being more eager to partner with you, especially because it would just expose them to a

technology – I mean, now you have banks and they're very eager to partner with fintech companies or explore fintech companies for potential acquisitions, because these banks are kind of – They're starting to notice the financial system is changing and they're wary of those changes.

How has your relationship with the banks changed as the business has evolved?

[00:13:31] JDG: So first of all, I think what you've just described is almost 90% accurate version of the early days of Plaid. Just like you said, I think, initially, I wouldn't say that all the banks were super excited that apps were being built on top of their data, because if you're a lender and if you believe in the hypothesis that to be a great lender, having as much information as you can about someone's spending habits gives you the kind of an advantage and how you can price that loan. Most people go to their primary bank to get a loan. That was true like 20 years ago. That's what you would do, because that person have the most information. They would give you the best loan. You could try to go to another lender and you'd have to print a bunch of PDFs and go there, but they don't have a relationship with you. It might be tougher to get the best rate.

So if you look at it from that perspective and then suddenly like anybody can now get access to the data that allows you to price a loan really well, that could be a little bit scary for a bank, because there's kind of – They used to have a more captive audience and now you're opening up the product experience and the learning experience much more broadly.

I think, in the initial days, I wouldn't say like every bank was like super enthusiastic to work with us. What's happened over time is, just as you said, like the consumers have like really reaped a lot of the benefits from the fintech ecosystem and they've demanded. I think for a lot of the banks, they realized, "Well, we're really good at things A, B, C and we're not good at solving problems may be like D, E, F," and there are a lot of startups that are like addressing those problems and providing real value to our consumers. So now let's empower those startups to work with us.

For a lot of apps like expensing apps, or business apps, like there's – For the bank, it just makes a ton of sense to work with those partners. As that has happened, what we've seen at

Plaid is the banks are like more and more willing to work with us in terms of like data access. I mean, I think we have like daily phone calls with like a large number of banks as part of our BD team to like strengthen the relationships and to find even better ways to work together to get around some of the latency issues, for example, that you described when you take a kind of screen scraping kind of browser approach, which is like frankly just doesn't scale super well.

The nature of the relationships have changed substantially, and today we definitely think of ourselves and I think most of the banks see us as a partner. There is only one little caveat to that, right? It's a goes back to the 10,000 bank issue, which is, "We're doing well." We're 170-person company. We don't have the ability to have relationships with all of the banks, and this actually, for us, something that's very important. Can we make sure that a small credit union in Omaha can provide their customers with all the apps that they're now expecting that you would naturally be able to connect to if you're working with Chase.

That's really important for us, because if we don't do our job well and connect as well with the smallest banks, then actually now, in today's world, that puts them at a competitive disadvantage, because their consumers will be like, "Wait, hold on a second. I can't use Venmo? I can't use Wealthfront to save for retirement or whatnot? The consumers will want to use these out.

That's what makes the work kind of complicated. I think what makes it like also very interesting, and there's a weird thing that happens, when any engineer joins Plaid, one of the first things that you do is you work on an integration. You work on like a connection with a bank and either trying to improve it or trying to take a bank that we don't have any support for and try to add support for it.

So when I joined a few years ago, even though I'm the VP of engineering, I wrote code at first like everybody to kind of understand the nature of the work, and you finish your integration and there's basically people who used to try to connect an app that they love and it just didn't work. Then you fix that and then it works for them and you stop getting tickets or you realize there's like 100 people every day who can now have a slightly better experience around their finances. I'm like – I don't know. That mission I think really, really resonates.

[00:17:19] JM: A bank account is this platform, and you can do all kinds of interesting things with bank account data. Some of those things are useful to the consumer. Some of those things could potentially be threatening to the consumer, because it's a very sensitive data. It's a bank account. Back in the day, people were afraid of putting their credit cards online originally. They overcame that fear. Now, I think people are slowly overcoming the fear of putting their bank account information online. But the most recent service that I used that I think used Plaid was Siftary – Siftary Connect I think is their product. Basically, it looks through your purchasing history for your business bank account and looks at all the SaaS tools that you've bought and helps you make buying decisions based off of your SaaS tools.

I saw this and I was like, "This is amazing. This is really useful to me," because I run a podcast, but even though it's just a podcast business, there's enough web services I buy that it's useful for me to see this diagnostic of where am I spending? Am I spending too much? But obviously I don't want my banking data to be abused. If I were to give an app permission, an app that I'm not familiar with, that I don't know about, access to my banking data using Plaid, what can that app do with my data? What are the restrictions?

[00:18:43] JDG: Okay. A couple of things that are really important. First of all, like the way Plaid is we can only get access of the data for consumer kind of – Like through our flow and empowers us to access it, and all we do with that data is we only provided to the app that you've connected. We're not reselling the data or anything. We're just a pipe between the bank and the app. But as you point out, there's a question like, "Can you trust the app in the first place?"

So you can self-serve into Plaid. You could, today, if you wanted, you could sign up for a Plaid account and you could use our SDK and you could start building app, but you couldn't put any meaningful volume through that app until the app went through like a pretty deep compliance check with us.

Basically, we have to be very, very careful like who is building a business on top of Plaid and to make sure that they themselves aren't doing anything like nefarious with the data. So that's like the first – That's the first answer.

The second answer is the nature of the data that they can access depends on the kind of app that they're building. So they either get access to account and routing number if they're an app that wants to transfer money. With that, generally, will come access to like your balance information, or they get access to your transaction history. This is for apps that either need to give you financial advice or that need something like credit scoring, like something that evaluates somebody's risk, or kind of – The third kind of big product that we have is something called kind of identity verification. That's generally used for businesses that are trying to see if somebody really is who they say they are to avoid kind of fraudulent scenarios.

Those of the three main types of data that people have access to, and it depends on the kind of app that's being built, like which part of the data we will connect to. When you connect to the app, it'll actually tell you what is happening. There's a pain in Plaid that tells you what data is being shared. So that like the 1st degree answer to.

The 2nd degree, is like next year, like we have a big effort internally like on providing users with more control over their data after they initially connect it. This would be the equivalent of seeing every app that you've connected. Deciding whether to un-connect them, and so on and so forth. Because a user controls like a big part of how we see our role in this system. Fintech has sometimes a history of data being shared with that kind of user's consent, and we just don't think that's. We view ourselves as like being an advocate for the consumer and giving them the control that they need and that they want. We want people to feel like if you're using – You're going to the Plaid flow to connect an app, there is like a baseline of like security and privacy that you can feel really comfortable in with that app.

[00:21:18] JM: Plaid has these multiple APIs. You've got authentication, identity, income. You've described some of these APIs broadly. Let's dig in to them a little bit. So there's this authentication API. What happens after the user has authenticated their bank account?

[00:21:36] JDG: After they authenticated their bank account, we give the app their account and routing number. At that point, the app can use any kind of other service to actually transfer the money using something called the ACH rails in the U.S., which is the main way that money can be transferred between bank accounts. For example, we have a partnership with Stripe where

we return to the app the account routing number and then they will use Stripe to transfer money to or from a certain bank account.

[00:22:03] JM: If I'm – Let's say I'm building my own payment service and let's say I call it Jeff Pay, do I want to use the authentication API? What are some of the other APIs I could use that will be useful just as if it's just a payment service?

[00:22:17] JDG: I mean, if you're using authentication, then you're going to have money go back and forth. You want to use balance to make sure that the money – There's enough money in the account for the transfer that happen. ACH rails allow you to transfer money, but they don't tell you if the money is there in the first place and there's a pretty steep penalty if you try to do a transfer and the funds are not available. So that's the first thing you do. You just balance.

The second thing you really want to do is you want to make sure that the person is who they say they are. So you want to find a way to get the person's identity, say, their name, first name, last name, or their phone number, or their email from some other service. Say they have a Gmail account. You ask them to authenticate with Gmail, so that you get their email and name. Then you want to make sure that it's the same one that they have on their bank account, like at the very least.

The third thing you might want to do if you really, really care about fraud, is look at the kind of transactions that have happened on the bank account. So this would require you to connect not just the off product, but the transactions product to see if the spending pattern is like way, way different than what they would expect. The reason why you want to do all these things is you want to get as far away from any like fraudulent money transfers as possible, because – Well, just for obvious reasons.

[SPONSOR MESSAGE]

[00:23:36] JM: Logi Analytics is an embedded business intelligence tool. It allows you to make dashboards and reports embedded in your application. Create, deploy and constantly improve your analytic applications that engage users and drive revenue.

You focus on building at the best applications for your users while Logi gets you there faster and keeps you competitive. Logi Analytics is used by over 1,800 teams, including Verizon, Cisco, GoDaddy and J.P. Morgan Chase. Check it out by going to logianalytics.com/datascience. That's logianalytics.com/datascience.

Logi can be used to maintain your brand while keeping a consistent familiar and branded user interface so that your users don't feel like they're out of place. It's an embedded analytics tool. You can extend your application with advanced APIs. You can create custom experiences for all your users and you can deliver a platform that's tailored to meet specific customer needs and you could do all that with Logi Analytics.

[Logianalytics.com/datascience](https://logianalytics.com/datascience) to find out more, and thank you to Logi Analytics.

[INTERVIEW CONTINUED]

[00:25:04] JM: The idea of building your own fraud analysis system on top of the transactions that a user has gone through, so this would be like an additional layer, because banks have their own fraud detection system. So if I'm building some kind of like a lending platform on top of my payment service, maybe I really do want to know how liable this person is to commit fraud. But why wouldn't I just trust the banking – Like the banking layer of a fraud detection system?

[00:25:37] JDG: Do you mean like the credit score system, for example?

[00:25:40] JM: No. I thought banks were pretty good at identifying fraudulent transactions. So if I make an API request the Plaid for this user's list of transactions, why would I want to do further analysis on that? Couldn't I just trust the bank's knowledge of what is a fraudulent transaction that they've rejected already?

[00:26:00] JDG: I mean, you look at their transaction history to see if the transfer they're trying to do via Jeff Pay is like out of character for that human being.

[00:26:08] JM: Okay. I see.

[00:26:10] JDG: For example, if the account connected is of somebody that's never used – You're literally the first fintech app ever that they've connected to the bank account. Maybe you want to ask them to do an extra verification step, such as you could use like a driver's license scanning service, or whatnot, to just really, really 100% make sure that it's this human being that's connecting through your app, because literally they've never done anything like this before.

[00:26:34] JM: Right. I understand. So we've gone through an overview of the APIs. What are some ways that people have used these APIs in methods that have surprised you?

[00:26:45] JDG: This is one of my favorite questions. So I always talk about this customer, because I think what they're doing is very cool. This company called Tally, they're not very big. They're a startup. What they do is they use us to look at how much interest rate your credit card company is charging you when you can't pay your credit card bill. What they determined from that is whether effectively your credit card company is overcharging you interest relative to your current kind of credit standing or ability to pay back your credit card.

Like a scenario might be you apply for a credit card after the financial crisis when you lost your job for whatever reason, right. So you had a pretty poor, let's say, credit score. So the credit card company came back to you with a really high APR card. You've kept it this whole time. Now you've got a good job and life is good and you're generally paying your bills on time, but not always. Sometimes you carry a balance on your card. When you carry balance, you're paying whatever, like a pretty high rate. So Tally will go in there and they'll be like, "Hold on a second. We'll pay your credit card bill for you and then we'll charge you less interest, because we think you're more trustworthy, or we think your credit standing has changed over time." So they're arbitrage between how risky someone really is and the balance on their credit card account.

I mean, when I heard about it – And they're going fast. They're doing well. I was like, "That's amazing." Literally, this app is helping. Like it's helping people like get a credit profile that's more in touch with their current ability to spend and not what it was whenever they applied for a credit card years ago. It's really powerful. There like innovation like that of happening that no one in the history of Plaid would have said like, "Hey, this is the thing that should happen." Or there's a company called Drop, it's a Canadian company, or at least launched initially in Canada. They

look how you're spending money on credit cards and they basically give you points with certain stores that you're spending a lot at.

For me, personally, I spend a lot at Whole Foods. So I get an extra, basically, like 20% cash back at Whole Foods from just connecting Drop. The idea is because they give me these coupons, it incentivizes me to keep going back to Whole Foods, because I want to keep kind of earning Drop points to work on there. It's a kind of like automated loyalty program where I don't have to worry about like scanning things or whatever. It literally just looks at how I'm spending money. It is very cool. For me, it's like a little bit of extra money that can go back to consumers.

So there's a ton of apps like that that are being built, and like we're totally neutral and like agnostic of any of these. Our goal is we want to make sure that things that are built on top of us are like trustworthy and like real businesses, but then we want to give the power to developers to be really innovative and build great things on top of Plaid. So I think every month – So we have internally like a dashboard of all the apps that are using us, and there's always – Every month, there's a couple of new apps in the top 20 or in the top 50 that I haven't heard of, and I do some research and half the time I'm like, “Well, that's very cool. That's something that should exist. I never would've thought about it myself, but it's just great to see that they found a way to build this business on top of us.”

[00:29:37] JM: Very cool. Well, let's get into the backend engineering. So we've talked in some detail about how Plaid is establishing a connection between the user and the bank. But there's obviously a whole lot more in terms of the supporting infrastructure. Give me an overview of the Plaid engineering stack.

[00:29:58] JDG: Sure. I mean, first, we're on AWS, which from of security and scalability standpoint, just makes a lot of sense and solves a ton of our problems for us. Language-wise, most of the backend is written in Go. Some of the bank integrations are mostly written in TypeScript today. Then probably the third part of the stack is we're really proud of not just our API, but the SDK that we give developers to build apps on top of Plaid. You can get started and built a very basic app in hours as far as the Plaid integration is.

Obviously, we have an android SDK, and we have an iOS SDK, and a web SDK, and those are written in Objective-C and JavaScript and Java for each platform. In terms of interesting technology – Personally, and I think most of the team at Plaid does not care that much about either hard problems or specific technologies, we're very, very pragmatic meaning. Meaning, we just want the easiest tool to use to solve the problem at-hand.

On AWS, we use Aurora and we use DynamoDB and we use Kinesis. As much as possible, we are using managed versions of Kafka, or like MySQL, or like a keyvalue store so that we don't have to worry about the operational aspects of it. The areas that are difficult for us generally stem from the fact that we don't control our database layer. Most companies, you generally control – There are some level of the stack that you control everything below. Well, at Plaid you control everything below until your database is literally 10,000 banks in the U.S. So that brings a lot of really pretty Plaid-specific interesting problems.

For example, we could – It's possible. We've tried very, very hard never to do it, but it's possible for us to like DOS a the bank. If you think about a small bank that doesn't have a ton of infrastructure where you have users that are you using apps that are required, like data to be updated very often, that's really dangerous. We don't control that. What we also don't control downtime of our banking partners. If they need to do a big version updates on a Saturday night for three hours, there's absolutely nothing we can do. We will not be able to connect accounts to that bank for that time window. Just because of that, we have a lot of challenges around monitoring this infrastructure that we don't control to make sure that we're being really good citizens with it. To make sure that we are correctly identifying when it's down, versus just slow, versus the APIs have changed and we weren't told that they were changing kind of scenarios. Then there's addition to that, there's a longtail problem, which is some of the smaller institutions, relatively speaking, don't get much traffic. How do you tell the difference between like it's down and just like it's so slow right now because of something going wrong along the way that it's not down? That's like one of the core challenges to Plaid.

Then it interestingly has – I don't know how you care about engineering culture, but it has a fascinating effect on engineering culture. So at most companies – So I used to work at Amazon. When you were there, you must've just been infatuated with the idea of like as many nines as

possible, 4, 5, 6 nines [inaudible 00:33:02]. What is it? I worked at Dropbox before, where for storage, we care about the nines quite a lot.

Plaid is a little bit different, because the truth is the banks themselves as a class. I'm not saying any one individual bank, but as a class, they're just not up the whole time. Some of them are built on old mainframes where you need to do software update, it could take like four hours on a Saturday. So if you're already losing – Say you're losing like four hours a month just from your bank partners on aggregate, like talking about four nines or five nines, it's makes no sense, because your customers, our customers are actually used to us sometimes just not being available through no fault of our own. But the problem with that culturally is it means that, now, then everything else at Plaid doesn't necessarily have to care about being more than three nines availability, because you've got this like baseline that you can't get better at.

Then if that's – Culturally, if that source is affecting your team, then you've got to ask yourself, “Does that mean people are not going to care as much about like great engineering practices?” Obviously, I think the nightmare scenario – So we care a lot about great engineering practices and we care a lot about having more than three nines for our own services, but that's something that we have to do like very consciously as a culture, because like the market isn't going to lead us in that direction naturally. That's one thing that's interesting.

Any fintech company, security and privacy is just like really high up on the list. I think there is some of the big companies out there, the ones that are used by hundreds of millions of people, most of them in their history have had a few bad security incidents. It might be making some user's photos available to other users. It might be – Like things have happened.

Generally, you get one or two bad incidents and then you wake up and you're like, “Okay. Well, we'd got to hire a big security team. We'd got to do this right. Can't have this go wrong.”

At plaid, there is a class of these things that if they happen are kind of company ending, and that's true about most fintechs where you transfer money. You can be down, but you can't mess other things up. There are some nightmare scenarios where money doesn't go from the right place to the right place, for example, because you make a core mistake.

Thinking about how do you build software while keeping those constraints intact earlier in the lifecycle of a company than you would normally have to is interesting. I have a Dropbox example, where like Dropbox was much larger before. I think it had to think about these things at the level that we have to at Plaid just because it could afford to wait a little bit longer. Like Dropbox security very, very seriously, I'm just making an observation in terms of Plaid had to worry about that way before I joined when it was just like a 20 or a 30-person company.

[00:35:36] JM: I think people – I'm not sure how – In the earlier days, people weren't maybe using Dropbox as aggressively in terms of mission-critical documents, or like storing – Not that I have one of these, but like a password file, like the keys to the kingdom kind of thing. But with banking information, it's kind of from day one you have this really high bar for how sensitive the data is.

You said about this idea that all the banks are basically the backends. So that is to say that if Plaid takes the data from the bank, like if I authenticate Plaid to give me all the transactions associated with the user, you're not storing those transactions yourself. You're just handing them off to the user and you're not storing them.

[00:36:28] JDG: It depends. We're effectively caching them, because if you've connected multiple apps and the multiple apps want updated transaction data, we can't hit the bank servers for every app every time they want an update. So there's like a caching aspect to what we do.

[00:36:42] JM: Can you tell me more about that caching layer? What's the conveyance of data look like between the bank and the user?

[00:36:48] JDG: On some interval, most banks we are forced to pull data at some – At a certain interval, we're like, "Hey, these apps need updated data for this user." We talk to the bank and get raw data from the bank. What we do is – And that's done through like – We use GRPC internally. So all that is synchronous, and then we get raw data from the bank. We actually store in – It's a permanent store, but it can be deleted, because we can always get the raw data back from the bank. Then we enhance the data. So we have a whole data pipeline that enhances the data.

One of the problems is if you use two different credit cards to buy the same thing at Home Depot, the transaction strings will look different with the two different credit cards and banks that are the issuers. So what we want to do is make the data look as standard as possible so that it's easy for developer to use.

So we have a bunch of machine learning and a data pipeline that it does everything, like it identifies like what's the location of the store. Is this a chain store? Normalizes the transaction to try to find like zip code and if it's recurring or not, categorizes it as this like a purchase at a grocery store, or is it at a gas station, or is it like a bill, whatnot?

So pretty complicated pipeline, that gets to eventually at the end of it like data that again is stored. Then what happens is we send web hooks to all the apps that are using us, just telling them, "Hey, there is new data available." Then each app, as they want or need it, will contact us just to like HTTP requests to our endpoint, authenticates and gets the updated data for that user. That's like the rough pipeline. There are like tweaks to it, but that's most of it.

For historical reasons, some of mostly micro-services based. You can think of a beautiful version of this, which is like every service is kind of stateless and does its thing. Unfortunately, like any fast-growing company, we have some technical debt from the old days that that's still in there. So a couple of these services rely on a shared MongoDB cluster to like share state, share metadata as the pipeline is going through and things like that don't scale and are harder to reason about than other parts. Please don't let any listeners think that everything is perfect. There's plenty of things that need to be fixed.

Then the other parts of it that are important is from a privacy standpoint and security standpoint, all sensitive data is basically encrypted as soon as we get it from the banks and not decrypted until it goes back out to customers. An example would be account and routing number. From most of our system, like 98%, it's just like some encrypted string.

[00:39:28] JM: One thing that you said that stood out was this idea that you have these 10,000 banks and their infrastructure can blip out at any time, or they can change maybe their API surface. So you want to have this proactive monitoring where you are, at some interval, pinging

these different banks and making sure like, “Okay, can we still authenticate with this bank? Can we still get transactions with this bank?”

With all these different APIs, you want to probably hit these banks at a regular interval to make sure that all the APIs still work for these given banks. What's the process of managing that infrastructure for doing that regular testing across all of those banks iterating through all of those banks and testing them all?

[00:40:15] JDG: Yeah. So when I joined, I actually came in and I was like, “Oh, we need just the service that you’ve just described,” but it turns out our volume is so huge that we don't really need to ping the banks to see if it's up. We find out with real traffic extremely quickly if something has changed.

Mostly it's about validation about some metric. A number of metrics that we try, like – That we track, in some metric will start to behave badly if the data is off. Number two, the second channel's support to tell us if something is off.

I'll give you an example, like very simple one is we get an account and routing number that's like the wrong length, or the name on an account changes, or like someone who use five accounts at a bank suddenly has zero. There is these like flag, these things that happen as soon as the data is and what it's supposed to be. We have a kind of interesting on-call structure internally. We have something called bank on-call, which every engineer is on regardless of what team they work on. Basically, if one of those things goes wrong, it's bank on-call's job to get it, to get it fixed like ASAP

[SPONSOR MESSAGE]

[00:41:30] JM: Managed cloud services save developers time and effort. Why would you build your own logging platform, or CMS, or authentication service yourself when a managed tool or API can solve the problem for you? But how do you find the right services to integrate? How do you learn to stitch them together? How do you manage credentials within your teams or your products?

Manifold makes your life easier by providing a single workflow to organize your services, connect your integrations and share them with your team. You can discover the best services for your projects in the manifold marketplace or bring your own and manage them all in one dashboard. With services covering authentication, messaging, monitoring, CMS and more, Manifold will keep you on the cutting-edge so you can focus on building your project rather than focusing on problems that have already been solved. I'm a fan of Manifold because it pushes the developer to a higher level of abstraction, which I think can be really productive for allowing you to build and leverage your creativity faster.

Once you have the services that you need, you can delivery your configuration to any environment, you can deploy on any cloud, and Manifold is completely free to use. If you head over to manifold.co/sedaily, you will get a coupon code for \$10, which you can use to try out any service on the Manifold marketplace.

Thanks to Manifold for being a sponsor of Software Engineering Daily, and check out manifold.co/sedaily. Get your \$10 credit, shop around, look for cool services that you can use in your next product, or project. There is a lot of stuff there, and \$10 can take you a long way to trying a lot of different services. Go to manifold.co/sedaily and shop around for tools to be creative.

Thanks again to Manifold.

[INTERVIEW CONTINUED]

[00:43:45] JM: That idea of on-call, maybe you could just talk about monitoring and devops more broadly. So like there are a number of companies that I've talk to where on-call rotations have become much, much easier over time as they've moved to more and more managed services, but there are some cases where there still is a lot of on-call due to certain infrastructure complexities or just the nature of the service. Can you tell me about monitoring and devops and maintaining uptime?

[00:44:15] JDG: Yeah. For Plaid as a whole, I don't think our devops is like super different from how other companies do it, and I think the observation that you just stated, which is likely with

managed services, we just have less worry there is mostly accurate. I think the deltas are, first of all, because we're a platform company, like sometimes we get spikes in like API usage of us that is totally unexpected. So one of our big customers launches a new use case where they need to call balance a lot more often and they decide they want balance calls like at 8 in the morning because that's when most of their users come online. Like, boom! We have a huge traffic spike that comes in. Things like that happen, and when they do like from an SRE perspective, it's like, "Well, auto scaling might break, or something," because it's just like so out of the ordinary, you have to deal with it.

The practices of a much bigger company, like Amazon, like the lightweight versions that we have internally take care of that. So for SRE team, we actually a platform, which I think is a little bit telling, because we don't call devops or SRE, we call it platform, because most of the work is actually in making internal developers like efficient the scaling aspects. We've kind of gone in a pretty good place. They have their own on-call rotation, and if something nightmarish like that happens, like off-hours, they're going to jump on it immediately.

The bank side is pretty different. So the truth is we have monitoring on things going wrong with banks. With 10,000 banks, things go wrong all the time. Literally, like every hour something happens. Some of the stuff that happens is like really weird. It's like bank A decided to merge these two saving account types into a new type, and we don't categorize that type as a saving account, because we've never seen it before. The metric that goes up is like, "Now, there's a bunch of accounts for this banks that are like they're categorized as miscellaneous, because we don't know if they're a credit card, or a checking account, or a saving account or whatnot." That happens all the time, and you actually don't – You definitely don't want to wake someone up like in the middle of the night to do that for a couple of reasons.

Reason number one is it happens all the time. So if you're going to do that, you'd have to have a really distributed team everywhere in the world if you want to like things good. At our size, it's just like way too much of a cost. Number two, users probably don't care that much at that point in time that that's correct as long as you fix it really, really quickly.

What we do, and the thing with monitoring is it's not just if something go wrong, but it's like what's the impact. How many users are likely going to be affected by this in the next 24 hours?

Obviously, if it's Chase, if there's a new account type at Chase, like all hands on deck. Figure it out. If Chase changes, like one of their internal APIs, I mean, they would always tell us in advance. But if that happen, like, "Okay, all hands on deck."

However, for a smaller institution, you might be like, "Well, we expect five or six people to be affected by this in the next couple of days." Then you just all create a ticket and then you have somebody take it on as part of like regular quality work, which we're pretty good at. There are exceptions to that. Anything that touches account and routing number going wrong, that's really high on the list obviously. But if like anything else, like our machine learning model is not able to categorize a larger percentage of transactions than usual, that's often not the end of the world. We find out about it. We figure out how much the impact is on our users from a data quality standpoint depending on the impact. We either work on it like that day, or that week, that month kind of mentality.

The final thing on this, which is I think important is a service is only as good – Only has to be as good as its competitors. Please, I hope no one is listening in a negative way about how I feel about the quality of Plaid, because I think your quality is super high. But the delta between the service we provide in most of our competitors is pretty big in terms of data quality, in terms of uptime.

So what's done is it actually gives us – Our customers are very understanding of the complexity of the problems that we're trying to deal with, and they have actually like – Frankly, they're really good partners and they have an understanding of how hard it is, because they themselves are building products on top of the same things. So they understand why it's hard for a machine learning model to always categorize things correctly, because they on their side are also trying to like categorize things as well and they know how tough it is.

That gives us like a nice level of understanding in terms of the domain-specific issues that happen in the banking world, and our bar is very high, because we think of how good we do all the things. That's the mode of the business. If there's going to be a competitor to Plaid, how many tens of millions of dollars and how many years are they going to have to invest to get to our quality level? So we take it very seriously, but you have to recognize that we can't do all of it perfectly all the time just given the complexity to the problem.

[00:48:59] JM: Definitely. Well, your engineering is somewhat – You're in a, like you said, this the spot where you have your backend is kind of controlled or bottlenecked by the banks themselves. It's an interesting constraint. I mean, like you said, you have to be just better than your next best competitor, and if you if he can do that, that's great. It's just the nature of the business that you can only do as good as the lower level of the bank kind of allows you to do, which makes me wonder.

We did a series of shows a while ago where we talked to some of these new banks. We talked to the Nubank, and Manzo, N26, and it was the thing to see how banking might change. But, I mean, more broadly, like I've talked to other fin tech companies, TransferWise, and Stripe and these Bitcoin companies, Coinbase, and it's like every fintech company I talk to, they have a gigantic vision for how things are going to change in their domain, because it's just like fintech is not going to be – Like this idea of kind of having to rely on banks and they're – All due respect to banks, it's just kind of old infrastructure and we kind of need better systems of financial movement in the world.

From where you sit at Plaid, what's your personal vision on how fintech changes in the next 5, 10 years? What are the things that you're most excited about?

[00:50:37] JDG: First I want to echo to kind of the last thing that you said, which is you said with all due respect to the banks, like from a technical perspective, they aren't up to the like latest standards or people's expectations about how systems can interoperate, or the quality of service that they provide.

What I'll tell you there, is one thing had been quite surprised by relative to before I joined Plaid to now, is how seriously like a large percentage of the banks take that and are trying to get much, much better at it. They're definitely not sitting on the sidelines of Manzo, or Simple and thinking that they don't need to kind of like keep stepping up their game in that dimension. They've got very good people who are working really hard at it.

I don't know if my endgame vision of the world is one where like banks are totally out innovated by startups. I do think what happens a lot of the banks are forced to step up and some are

successful at doing so and some are less successful at doing so because of these kind of competitive pressures from new fintechs.

That being said, our thesis at Plaid is pretty simple, which is there's going to be – We're like a very early stage as a fintech innovation. One way we think about it is a lot of the innovation should happen in terms of the product experience that's provided to consumers. Traditionally, there's this like infographics somewhere and it shows a picture of the Wells Fargo website and it shows all the drop-down menus in the website. For each drop-down, it has the name of like 5 to 6 companies that are trying to innovate for that product. It could be like transferring funds, or loans, or it could be mortgages, or it could be saving for retirement, or it could be 401(k), or it could be sending money to friends, literally. Literally, there's like 30, 50 boxes and each one has 3 to 5 companies.

When you talked to Robin Hood or some of the new banks, like they actually kind of want to replace Chase. They just want to be a better version of online Chase. Robin Hood doesn't just want to do the brokerage. They want to own the entire financial relationship for a consumer. I think the way we look at it at Plaid is a little bit different. We are not sure in the next wave they'll be a ton of consolidation. We kind of view, like there's going to be apps that are really good for saving for retirement if you're this kind of person, and other apps for saving for retirement if you have a different profile. Then there's going to be apps for like how you manage your finances if you're like really tech savvy and send some that do it for you.

But the point is there's going to be a kind of like a shopping mall of for every aspect of your financial life that you care about where there's going to be some category winners, but they're probably not – They're going to be too specialized to win horizontally. That's like kind of definitely personally the endgame that I see. That's what we're seeing with Plaid. We are seeing more and more innovation. Maybe at some point there is a consolidation phase that happens, but that that seems like more in the future than we are today. We're still seeing like the core things that banks do being heavily, heavily like iterated on top of and perfected from a product perspective.

That's even weird things. Like one example is – The name escapes me right now, but it's a company that they're helping with like loan repayments. All they do is the consumer experience

of like when you've got a loan and you need to make payments on it, how to make that better from the user's perspective. Then they're going and selling that to banks and to startups and to a bunch of people. That business is not going to expand to completely put the banks out of business, but it's like a multibillion-dollar. Like, "Damn!" That someone needs to go after.

What we want to do – And so what I'm really excited about is kind of going up the stack from where we are today to make it easier for the next generation of startups to create better product experiences, and I think the endgame of that is if you're – It should be as easy to develop in fintech as it is to just develop for anything. You shouldn't really have to worry about fraud, about how you move money around. You shouldn't have to worry about creating your own like lending model. You should just be able to use one that's just like best in class. Like all these things that are difficult today that you have to do in-house you shouldn't have to do.

If you look at like Coinbase, they have a big fraud team. They have an awesome fraud team that's trying to detect like fraudulent transfer money in and out of bank accounts like via Bitcoin as a way to anonymize it. They're awesome at it.

The next company in that space, they shouldn't have to build that functionality internally. We should be able to provide – Like a company like Plaid should be able to provide the scoring and confidence necessary that you don't need that the team internally.

Likewise, like for any lender in terms of like trying to determine like how creditworthy somebody is, like literally, every lender out there will have a data science team that's working on this on risk, and there's not a ton of competitive advantage. Two lenders that have access to the same input of data going in will make very similar decisions on the way out.

The delta has been what data are they looking coming in. There are some new lenders that are looking at signal from like social networks, or they're looking via Plaid at signal from transaction history. But once you have access to all the data, like you should just be able to provide a model that multiple people can use. Then the problem isn't one of like, "Hey, can I like issue a loan?" The problem is, one, is like, "How do I find places where people are looking for loans and can't get them? How do I get to those customers and make it easy for them to get like a loan when they're buying a television or whatnot?" That's the view of the future I'm really excited about, is

like making it trivial for developers to innovate on fintech, and fintech can touch everything. It should touch like a mattress company that needs offer financing to people. That shouldn't be difficult, but it is.

That's what I get excited about. The weird thing about Plaid, honestly, it's true about any platform company, is you see the innovation on top of you and you're enabling it, but you're not quite part of it. So one of the fun things about the job is like when I do talk to customers and learn about the problems that they have even building on top of us, because that just gives me a whole bunch of ideas of what we need to do next.

I think that's not very different from Amazon. Early Amazon, it's S3, it's EC2 and they're like, "Oh, people are using EC2 to run like MySQL all the time. Why don't we have a manage MySQL?" People are dealing with bigger and bigger datasets, and so they're running these giant Spark clusters or whatnot. Why don't we try build like a data warehouse for them? Or like, everyone is using Kafka and message queues. Okay, we'll just like have Kinesis.

I think there's a version of that for Plaid, which is going from raw data to like insights or to the financial building blocks. It's not, by the way, just tech innovation. It's also like compliance and like legal innovation. Like to issue a loan, you need a bunch of licenses and so on and so forth. What if we had an API for loan at issuance? That is like way in the future, but what if we could do that and you could start a company that issues loans, but you yourself don't have to like do any of the legal or compliance work? It's all taken care of for you. I think that would be an awesome future.

[00:57:19] JM: Well, why not. I mean, I log into QuickBooks and I know that there is some loan issuance system within QuickBooks and I believe that they outsource it to somebody. That is one of the markets, like loan issuance. That's one of the markets where, like you said, it's not going to be this some winner take all, like somebody dominates the API for lending. There's going to be a bunch of different APIs for being able to issue loans with different datasets and so on. What you've said about Plaid, it kind of reminds me of a show I did with Checkr recently, because they do the background check API, which sounds like this like another – this is kind of very similar thing, where it's like the way that they do it, the API for background check, it's not pretty. It's not clean. Because these background check process kind of sometimes requires

people going to look up paper documents, but the experience that Lyft gets is just they make an API request and they get the background check as a response and there's such high-volume that goes to the system that Checkr, like you said with Plaid, being able to leverage the data flows through the system can see what are the things that people want out of this system that we're not currently providing, and Checkr has already built a bunch of products that envision a future with a more futuristic version of a background check, and then eventually like what do people use background checks for? Applying to jobs. Maybe Checkr is like a job search platform or a job acquisition platform. It's like the room for expansion of these APIs that have a high-volume use cases, it doesn't seem to be limited. It's very rare that you hear about one of these API companies, like, "We're run out of ideas." You don't hear that.

[00:59:03] JDG: No. Your customers tell you what else they need, and generally it's pretty adjacent to the things you're already doing. Yeah, I mean Checkr is a great company. I think because we're much smaller than Stripe, I always try to look at like companies that are bigger that are kind of paving the way. I always feel like the way – You could think of Stripe as just a payments company, but if you think of them as like making really easy to start a business online, it just takes on like so much more scope. That's the mission, right? Because that's the mission, they can do things like Atlas, which just takes so much pain out of just starting a company and getting them online. I love that.

I mean, I think the view of Plaid's future is very much along those lines. The mistakes with startups, I think sometimes is too – If you haven't quite won your act one before you worry about act two. We're hell-bent on winning act one and we're starting to do some of these kinds of things next year, which I'm pretty excited about. But I wake up every day and it's like, "Are we still the best at what we do with the banks and are we setup to keeping the best for that for the next two to five years?" You can't ever lose the value of that price. As exciting as like other things that your customers want you to do for them and the pots of gold that that one day would represent.

[01:00:20] JM: Jean-Denis, thanks for your time. This has been really interesting and I'm excited about your future with Plaid.

[01:00:26] JDG: Yeah, thanks a lot. Thanks a lot for having me. Honestly, the questions are fantastic. I felt like you engaged much more with kind of what's different about fintech than a lot of people I talk to. I really appreciate your time on not just the interview, but the research.

[01:00:38] JM: Thank you.

[01:00:39] JDG: Yeah.

[01:00:39] JM: Thank you.

[END OF INTERVIEW]

[01:00:43] JM: OpenShift is a Kubernetes platform from Red Hat. OpenShift takes the Kubernetes container orchestration system and adds features that let you build software more quickly. OpenShift includes service discovery, CI/CD built-in monitoring and health management, and scalability. With OpenShift, you can avoid being locked into any of the particular large cloud providers. You can move your workloads easily between public and private cloud infrastructure as well as your own on-prem hardware.

OpenShift from Red Hat gives you Kubernetes without the complication. Security, log management, container networking, configuration management, you can focus on your application instead of complex Kubernetes issues.

OpenShift is open source technology built to enable everyone to launch their big ideas. Whether you're an engineer at a large enterprise, or a developer getting your startup off the ground, you can check out OpenShift from Red Hat by going to softwareengineeringdaily.com/redhat. That's softwareengineeringdaily.com/redhat.

I remember the earliest shows I did about Kubernetes and trying to understand its potential and what it was for, and I remember people saying that this is a platform for building platforms. So Kubernetes was not meant to be used from raw Kubernetes to have a platform as a service. It was meant as a lower level infrastructure piece to build platforms as a service on top of, which is why OpenShift came into manifestation.

So you could check it out by going to softwareengineeringdaily.com/redhat and find out about OpenShift.

[END]