EPISODE 1077

[INTRODUCTION]

[00:00:00] JM: Brex is a credit card company that provides credit to startups, mostly companies which have raised money. Brex processes millions of transactions and uses the data from those transactions to asses credit-worthiness, prevent fraud and service insights for the users of their cards. Brex is full of interesting engineering problems. The high-volume of transactions requires data infrastructure to support all those transactions coming through the platform. As a credit card company, Brex needs to integrate with credit card networks and banking systems. There are internal systems for applications such as dispute resolution.

Cos Nicolaescu is the CTO at Brex and he joins the show to discuss engineering at Brex, the dynamics of a credit card company and his strategies around management. It was an instructive look inside of a rapidly growing fintech company.

We are always looking for new show ideas. If you have a show idea, you can go to softwaredaily.com and create a quick posting about that show idea, whether it's a project that you're working on or the company that you work at. We are always looking for new stories to cover.

[SPONSOR MESSAGE]

[00:01:18] JM: Triplebyte is a platform for finding a great job faster. Triplebyte works with more than 400 tech companies, including Coinbase, Zooks, Dropbox and Facebook. Triplebyte is focused on matching high-quality engineers with great jobs. We know that the economic downturn has caused some significant upheaval for current computer science students and new grads and many students don't have the internships or the engineering roles that they'd expected, and you can learn more about these opportunities by going to triplebyte.com/sedaily to sign up for the webinar that Triplebyte is hosting May 28th at 5 PM. You can go to that webinar to find out more about how to prepare yourself for a job, prepare yourself for an interview. Triplebyte.com/sedaily, and during that webinar, Triplebyte will interview Tito Carriero, a current chief product development officer at Segment, as well as former early engineers of both

Facebook and Dropbox, and you'll get to learn more about how to get in the door at these opportunities for internships and jobs. You can also go to Triplebyte to find resources on engineering assessments, mentorship and other materials. You can sign up for all these at triplebyte.com/sedaily.

Thank you to Triplebyte for being a sponsor.

[INTERVIEW]

[00:02:53] JM: Cos Nicolaescu, welcome to the show.

[00:02:55] CN: Thanks. Nice to see you.

[00:02:56] JM: Nice to see you too. I'd like to start with an overview of the Brex business and then we can get into engineering. Brex is a credit card company. This is in contrast to Visa and Mastercard, which are credit card networks. Could you explain the difference between those two?

[00:03:11] CN: Yeah. We actually sit on top of Mastercard. We actually work with them. The easiest way to think about Brex from a credit card perspective is mostly from an issuer bank. Think about your bank, you have like a Chase card or a Bank of America card. This is the same thing where you have a Brex card. It's primarily for corporations and companies and startups. So our relationship with that is as you use that card, it goes through the Mastercard network and then everything around like the balance and approvals and statements is generated by us.

[00:03:41] JM: Why are there only two credit card networks? Visa and Mastercard?

[00:03:44] CN: I mean, there's a lot more. I think Visa and Mastercard are the two biggest global ones. You obviously have American Express as well, Discover, but you also have local networks like China UnionPay and obviously a bunch of local networks in Europe. What Visa and Mastercard have done really well is basically distribution. They act as a switch for all the merchants and banks to be able to work together.

I think if you look at how they started, they basically started by having a lot of these different networks of banks that were working together and they created a standard that was widely adapted and they made it very easy for merchants to be able to accept those with very little fees. In contrast, if you look at some of the other networks, they're primarily considered closed networks versus open networks. If you think about Visa and Mastercard, any issuer can work with them, whereas you think about American Express, Discover, they are the issuer and the network at the same time. So they don't have the same distribution because they're kind of their own bank and there're pros and cons to those models.

[00:04:43] JM: When somebody applies for a Brex card, what is the process? Can you describe from just the top-down?

[00:04:51] CN: Yup! You go to our website. It's super simple. You give us your email, your company name. You tell us what kind of company you are. You then give us a few information about your company. Who the major stakeholders are? Then you link your bank account. You basically can go log in to whatever bank account you're using. Then from there, it's basically done. You get a virtual card and a physical card if you want one, then you can start using the product.

The thing that Brex did uniquely relative to any other corporate bank before was that we actually use our information to underwrite you as a company and not you as an individual. Historically, if you are a founder, for example, and wanted to open a bank account and get a corporate card, you would primarily give your personal information. What's your credit score? What's your bank account? Then that's what the limit and that's what the underwriting was used upon. It didn't really matter that you had like $10 million in your bank account. It just mattered like are you as the founder, a person, that I as a bank would give you credit? If the answer was no, you wouldn't get it. If the answer was yes, you would get it, but you would get it using limits that you would get as a person and you would also be financially be liable for those. You're personally liable for the credit of your own company.

If I as an employee in your company use it and then something happens to your company, while don't have that separation between like individual and corporation, you would still be liable for that. What Brex did is we changed all that model in which we said, "No. That's kind of silly to do.

We should actually look at the company, model the risk for the company based on the industry investors. How they're spending money, etc. etc., and give them limit.

The other thing that was a huge differentiator from this perspective was it's relatively easy for me as someone who's been in the US for a while and has a good credit score to actually go to this. But if I go back to – I moved here from Romania. I remember my first year in college when I got here, like it was so hard to get a credit card, because I had no credit history. Now put yourself in the shoe of an international founder who wants to basically just get a corporate card for their bank. They went to YC, they raised a bunch of money, and now they can't get a simple credit card for their company because they have no credit history. That's just like something that is very painful for no good reason whatsoever. So we kind of changed all of that. But the process for us is basically like if I'm going to process, that's all online.

[00:07:14] JM: Got it. Okay. Now that we've given an overview for the business and the onboarding, let's start to talk about technology. You're the CTO of the company. You came in when it already had some significant momentum. Describe how you learned the infrastructure across the organization as you came into it. What was your personal onboarding trajectory to learning as much as you needed to learn as quickly as possible?

[00:07:41] CN: Yeah. I joined Brex in December 2018. The product had launched in June. We had about six months' worth of time that the product has existed. By then, the company was a little bit over 50 people. In June, it was about 20 people. My first two months in the company were spent I would say 50% as an individual contributor and 50% as a manager, and the 30% [inaudible 00:08:10] literally embedding in all of the different teams to understand the state of infrastructure, the state of the codebase and familiarize myself with like all the technical aspects of the product.

My job hypothesis is generally once you go into management, your technical bar is basically set there, and best it doesn't go down very drastically. But in practice, obviously there's some slowness. Similarly, the bigger the company, the harder it is to ramp up on technical aspects. I wanted to take the opportunity early on before I had a bunch of one-on-one's and organizational responsibility and recruiting to focus on understanding the technical stack to have a solid foundation for that.

[00:08:49] JM: Do you feel like most of your decisions are managerial at this point, or do you find yourself diving into technical discussions as well?

[00:08:59] CN: I dive into technical discussions from a pleasure perspective. I really enjoy technical discussions. I read every design doc in the company. We do a weekly design review session where people can present their design docs, either go to those or watch the recordings. I set up an engineering strategy meeting with all the senior engineering leaders to discuss kind of long-term technical decisions. In those, again, I try to attend those or at least watch the recordings. It's mostly from I really enjoy the ability to dive into technical things, but not as a decision-maker as much as possible. I don't think it's my job to actually have to decide every single technical detail. I think that's disempowering to every engineer.

Most of what I tend to think about are what are the right systems and forums and processes that we need to put in place to be able to make really good decisions? That includes organizational decisions for all the managers, in recruiting, in hiring, in performance management, in feedback, as well as product decisions, as well as technical decisions.

[00:10:02] JM: Okay. When a Brex card gets swiped, like let's say I raised $5 million for Software Engineering Daily and I open up a Brex account for me and all my employees and we all get credit cards. I go and take my credit card to Phil's and I swipe it for a cup of coffee. How is that transaction information making its way to your infrastructure?

[00:10:25] CN: You go to Phil's, you swipe your card. Phil's uses a payment processor. I have no idea who they actually use, but there's a payment processor. That payment processor works on top of all the different networks. They have an integration with Visa, an integration with MasterCard. They probably have one with American Express in the US, Discover.

From there, MasterCard, in our case, so the card would click MasterCard. MasterCard looks at the first six digits of your card. They're called a BIN number, bank identification number. Based on that, they determine that this is a Brex-issued card. So they route that particular call to us. So we get an authorization message, which includes the amount, the merchant, the card number etc. etc. routed to us. We then look up your account based on the card. You see what your

balance is. We see what the transaction is coming from. We put it through our fraud engine and then we either approve it or not. If we approve it, obviously we subtract it from your balance and showed on your statement, and then we'll return that to MasterCard, who then returns it to the payment processor and you get that little approval. Then at the same time, with Brex you would get a push notification or an SMS message about that transaction and then you could take a picture of your receipt and then send it over and then we would match you with that transaction so you don't have to worry about expense management later on.

[00:11:45] JM: From the network perspective, like who in the network is sending you that transaction so that you notice them the push notification? What's the integration point?

[00:11:54] CN: The integration point, like we're directly integrated with MasterCard. There's an actual technical integration with MasterCard. So we register our endpoint with them. There's certification process they go through where they run a bunch of different test cases to make sure they are handling different situations correctly. There's like a list of scenarios that you have to go through, and then they register. Basically, they have a mapping table. You can think about it where they have the BIN and then they have who to actually route the call to, and then they just forward the call over.

[00:12:24] JM: You have a high-volume of transactions that are being sent to you basically from MasterCard, from the MasterCard network. Describe the ingest process for this massive transactions. Are you buffering them? Are you sending them straight to some kind of service for processing? What's going on there at the ingest point?

[00:12:43] CN: Yeah. At the edge, we have a public endpoint, and then that public end point is load balanced. The call comes to that public endpoint. Behind that endpoint, we can have N-number of endpoints, like private endpoints for that service. So we can scale that service to like 10 instances, a hundred instances, a thousand instances, whatever we need to based on performance needs. That call gets routed basically round-robin to one of those endpoints. That particular box then loads up your data from a database. We have a database with all the cards. From the cards, we have a database with all the balances for each company, and then we run all the checks internally. Some of those are synchronous. Some of those are asynchronous.

The way it works is you have to return to MasterCard roughly within about 500 milliseconds. You can't really afford to do a lot of computation at the time of authorization. What you can do is you can do other checks later on to make sure that your transaction is correct to do processing with everything else. But the decision to actually improve or not has to be as cheap as possible and it has to be synchronous. So we basically do all that work in parallel. We do our fraud checks, while the same time we do our balance checks and a few other checks. We can then decide whether to approve it or not. We return the response and then we put a message on a queue. We use Kafka for messaging internally, and then a lot of the asynchronous processes will then handle that message to continue the work for that request.

I mentioned earlier, you get a push notifications or an SMS message that comes through an asynchronous flow, because we don't want to actually take time to do that synchronously because it's not necessary to actually decide the approval.

[00:14:26] JM: If I understood correctly, the transactions all come in, they get load balanced. You have some round-robin process that's handling the writes to your transaction database? You have a transaction database and then you're also sending it to Kafka, did you say?

[00:14:42] CN: Yeah. Comes to a public endpoint, load balanced, goes to a particular API endpoint. Does all the work to determine that, saves the data in the database, puts a message on Kafka.

[00:14:53] JM: Got it. Tell me about some of the systems that are reading the transactions off of Kafka.

[00:15:00] CN: Systems that we serve off of Kafka, like I said, notifications and all the communications that we do, the statement generation. You get a list of transactions on your statement. That gets processed asynchronously, because, again, we already have that transaction. We don't need to do a lot more writes while we're returning an answer to MasterCard. This get ingested further into more of our analytics systems. We have Snowflake and then we have an entire analytics systems for dashboard and metrics. Those get all processed of that. As well as a lot of the model building for fraud. Real-time, when the transaction comes in, we basically try to generate a score and we get information from

MasterCard as well about the riskiness of the transaction. Then a lot of the updates to our model are done asynchronous. So that feeds into that model that we run then afterwards. Those are kind of the major ones in think.

[00:15:51] JM: Do you dump all the transactions just into Snowflake or do you have a separate data lake system?

[00:15:57] CN: We use Snowflake for that and then we use Looker on top of that with Airflow. Basically, there's an Airflow job that pulls stuff into Snowflake, and then Looker dashboards that are powered by Snowflake is how most of the analytics system works.

[00:16:09] JM: Do you have any idea what the architecture is of that? Because I know you're not paying for like having all of the data in Snowflake to be accessible quickly all the time. I assume it's some kind of tiered storage system. Do you know anything about that?

[00:16:24] CN: Yeah. The way Snowflake works is – Snowflake works, rotating or your AWS infrastructure. From that perspective, you basically deploy it in your AWS systems and then you pay your AWS bill. So that kind of determines partly how much data you want to store there. How fast do you want to have it available? You then pull that data on a regular basis and then you do all your computation and transformations. There's two ways to do it. You can either do it with deltas sor you can do entire snapshots where you can basically say, "I'm going to recreate the entire world at whatever interval."

Typically, if you do smaller intervals, you like to do deltas. If you do larger intervals, you can just do snapshot. Snapshots are easier from that perspective, but they're obviously more expensive to run. So we do a combination of those where we have things that come in on a delta basis, on a much more regular basis, and then we also do a cleanup where we just recreate entire world on a less often basis.

[SPONSOR MESSAGE]

[00:17:27] JM: When you spend your spare time learning, you can accelerate your career. O'Reilly lets you learn through high-quality books, videos, courses and interactive experiences.

O'Reilly content has been built over decades. They're a trusted source of effective technology education. If you're an individual leveling up on your own, you can use O'Reilly to chart a course for your career goals. If you manage a team or a company, you can get access to O'Reilly's career development resources for your whole organization.

Go to softwareengineeringdaily.com/oreilly to explore O'Reilly's e-learning experiences. You can build the skills you need to future-proof your career. Check out softwareengineeringdaily.com/oreilly, and thank you to O'Reilly for being a partner with Software Engineering Daily for many years now.

[INTERVIEW CONTINUED]

[00:18:28] JM: Who are the consumers of Snowflake? Is it just the variety of dashboards or do you also have some data science workflows?

[00:18:35] CN: Yeah. All our data scientists work off that, all the dashboards that we have throughout the company. In general, as a company, were very data-driven. Our sales team is very metrics-oriented. Our finance team, our ops teams, everybody has a set of dashboards and we actually embed data scientists in each of those teams. Then we also use it for engineering purposes, basically like for anything that we do a rollout of our features, we can also look at the data to see how that's being used for new features, to see how – the distributions, etc. Engineers can also run queries on the data.

[00:19:13] JM: Do you have to do any data cleaning or do you have just complete control over the structure of like the transactions or are there any other pieces of data that need to be cleaned up?

[00:19:24] CN: Yeah. We have full control over it. We basically pull the data from the transactional store and put it into Snowflake during the ingestion time. This is where the Airflow jobs come in play. We actually scrub data for a certain [inaudible 00:19:38] information. You'll never actually see a card number into Snowflake. We'll actually scrub a lot of the customer

information so it's not personally identifiable. The company is still there, because you need to be able to find things.

In general, you're not going to have every information about every single card holder, what their email address is, what their phone number is, etc., to protect their identity. There's a different system for that which we use for support purposes where you have to have permissions to be able to kind of access that data.

[00:20:05] JM: One of the kinds of operations you have to perform throughout the lifecycle of a user is credit worthiness. Can you tell me about credit worthiness and just generally assessing the risk of a customer throughout their lifecycle?

[00:20:24] CN: Yeah. For the vast majority of our customers, they link their bank account, and depending on the type of customer. For example, if you're an ecommerce company, they might actually give us access to other information, such as like their Shopify data, for example, or data into the ERP systems. We ingest all that data on a daily basis and we complete the model to see how things have changed.

We try to stabilize things so that your limits don't vary up and down [inaudible 00:20:51] a pretty bad customer experience. But we do make sure that if things, like if you, for example, have a million-dollar in your bank account today and tomorrow that goes down to like $50,000. That's obviously going to have major effect and it's going to get flagged and someone's going to take a look to see what's going on there.

In general, what we try to do is we try to model basically when you sign-up, we try to model like what kind of company you are, the industries, etc., to make sure that like you're a legit business. Then on an ongoing basis, we predominantly try to model what your income flow is throughout the company. How much money are you spending? How much money is coming in? What's the rate that you're spending versus coming in? What are the different sources? Again, it gets more and more complicated as you're going to different verticals.

For e-commerce companies, inventory data is very useful, because typically you spend a lot of money to buy inventory, which you then get back through sales throughout a period of time.

Having this ability into that is very useful to be able to build the product that it's good for you. We do that literally on a daily basis to be able to be as accurate as possible, and that's helped us a lot in minimizing risk for us, because most traditional banks, again, going back to the flow that you get with a normal bank, is you go to a bank, you give up on some paperwork, they give you a credit limit and then that credit limit is static for the foreseeable and maybe you get updated once a month, once a year. But even as a consumer, your credit limit doesn't really get updated. If you want to update it, you call them and then you give them like your income and then they decide what your limit should be. We've decided to make it much more dynamic, which allows us to control our risk much tighter because we get a lot earlier signal if things are not going well.

[00:22:31] JM: Have you seen any surges in problematic credit worthiness since the COVID-19 recession?

[00:22:39] CN: We've had some customers who've reached out especially more on the e-commerce side about payment plans. The thing about companies is they don't lose all their money overnight, typically it's like they start worrying about it. They start seeing impact on their sales. For the most part, if you're a legit customer and a good business, you're trying to do the right thing. You're not out there to try to screw people over.

In the beginning, in March, around middle March when this started, we did have customers who started reaching out and said like, "Hey, I have this balance. Is there any flexibility that I have a payment plan that I can apply to this?" In general, like that process was entirely manual in the beginning, because it was so rare that we had to do that. One of the things that we did in preparation for this when we started seeing early signs is to automate that process. We make it as self-serve as possible so that customers can start scheduling those on their own and not have to have a lot of overhead to that. We're trying in general to make as many adjustments as possible. So we revamped the rewards program to make it more remote-friendly, because a lot of our rewards work towards bike restaurants and rideshares and hotels, which obviously are not very useful right now.

There's a package that we launched last month, which gives you rewards for Zoom and Slacken a lot of online collaboration tools and remote tools instead to kind of have good use for you rewards during this period of time on top of that.

[00:24:00] JM: What about fraud prevention? Can you tell me about your fraud prevention system?

[00:24:04] CN: Yeah. As an issuer, there's a process called [inaudible 00:24:07] process. Normally, if you think about as a consumer you use your credit card and if there's a transaction that you don't recognize, you call your bank. We're your bank in this case. When we think about fraud, we think about it from a few different angles. The first one is people who are trying to scam Brex. You're not a real legit business. You're just trying to get [inaudible 00:24:30] to be able to use it. So that's at the top of the funnel, which is why a lot more checks happen in that place.

Assuming you get through and you're a legit business, then fraud happens in one of two ways, either what's called transaction fraud or account takeover. Basically, either I steal your credit card or I somehow takeover your account and then start using it. So you are a legit business, but your transactions are not being legit transactions. Number two, it's called [inaudible 00:24:59] fraud. This is the case where we have people that might say like, "Oh, I never bought this," but in fact they did buy it, but they're just saying they didn't.

In the fraud process, the credit card networks use something called viability, the treasury chairman who's liable for that transaction. If you go with a stolen credit card and buy something online, there's a determination whether the merchant should have detected that and said, "Hey, you are actually using a stolen credit card," or whether the issuer should've done that. The bank should have said like, "Hey, I shouldn't accept this transaction because it's not a legit credit card."

For the most part for the way the industry is right now, in terms of physical transactions, when you use your EMV chip, the bank is liable. Basically, if I physically steal your credit card and go to a store and buy something with your card, Brex would be liable for that transaction. So it's up to us to either prevent that collection from happening or we have to cover the losses.

The second scenario is online. In the online case, predominantly, it is the merchant who is liable. In that case, typically, the merchant has more data to be able to determine whether a

transaction is fraudulent or not, because they can see whether this is a new account that you created and just added a bunch of credit card. They should be able to see your IP address that you're purchasing from. So you have a lot more signals that enable you to detect that. The network, say in that case, the merchant is liable for the. In the case where you use your card, throw my card and you go buy something online, we would do a chargeback process and the merchant would actually cover the cost of that transaction rather than Brex.

[00:26:33] JM: As far as deployment, if we talk about services throughout the company, what is your typical deployment medium? Are you deploying to EC2 instances or Fargate? What are these individual service teams using?

[00:26:48] CN: Yeah. We use EC2. We're all in AWS. We have multiple services in our architecture. So we have – I wouldn't call it a microservice architecture, and that I think maybe that' quite extreme to have a kind of small services. We have tens of services. I'd say that they're medium-size services. We deploy roughly around 50 times a day to production, various services. Our deployment process is fully automated. So you have a pull request, it gets reviewed. All the tests pass. We merge it to master. A bunch more tests pass. It's gets deployed to [inaudible 00:27:26] environment where it has smoke tests, and then it gets pushed to production in an automated fashion.

[00:27:31] JM: Tell me about some of the other AWS services that you're frequently using. Are you standardized on any database service or heavy uses of Lambda? Just tell me more about the AWS services.

[00:27:40] CN: Yeah. We use RDS with Postgres for that. That's our main store for all the data. Obviously, I mentioned Kafka, which we run in our AWS instance. We use Lambdas pretty rarely. There's a few in our website that we use for the frontend, but otherwise not a ton of Lambda usage. Biggest usage, obviously EC2. Then if I think about our AWS usage, there's the production environment where we have all these services and databases, but then we also have our developer environment. If you actually think about that deployment sites, that tends to actually be bigger, because first of all, our entire build process runs in AWS. We use build pipe. Obviously, that impacts het set of servers that they use. But also every developer can get a preview environment so they can run tests and tests things out if they want to in addition to the

local environments. Normally, in your local environment is where you would run like local test, etc. etc. But if you want to do something end-to-end, you typically create a preview environment. We dynamically generate one for you.

[00:28:42] JM: Are there any SaaS products or infrastructure projects you use that have surprised you in how useful they are?

[00:28:49] CN: I would say the build system is interesting. Throughout my career, I've gotten through so many of those. I would say [inaudible 00:28:55] has actually scaled really, really well for us. We used to use CircleCI before, and I've used Jenkins in the past as well. I think that one surprised just like how well it's worth. I think similarly for your earlier question around analytics and data lakes, we've had a really good experience with Snowflake. I've, again, used like Red Shift. I used Impala before, like a bunch of different technologies. Snowflake actually scaled also really, really well for us so far.

I'm trying to think if there're any other ones. Octa, I'm surprised how fast they became a de facto. We use Octa for multifactor authentication internally on our IT infrastructure across the board, and it's basically like de facto everywhere nowadays. Like every company I know that's not a huge corporation that has their own technology more or less uses Octa. For logging, I think it's the next thing. We use Datadog right now, and there're pros and cons with that. We're evaluating what our staff should look like for observability moving forward.

[00:29:54] JM: What are the cons?

[00:29:56] CN: The cons are the data is all in the cloud. So a lot of the issues are – The challenge with logs, is you don't want to have PII information in your logs that then get sent to a third-party and then that data lives with that third-party. [inaudible 00:30:08] again are really thoughtful about the customer privacy. If we have logging infrastructure on our own side, at least the data, like you have control of the data. I think that's the challenge number one. It's also against their terms of service, like they don't want to have PII by design. They don't want to have sensitive information. But obviously like there's times when you have a log entry that has that and so you have to then, first of all, catch it and then clean it up.

The second piece of ground that is just ease-of-use and how easy can you query things. I think Datadog has proven really well from a monitoring perspective. It's really good to have metrics and KPIs and set alerts on top of that. For just pure logs, there are other tools who are better. I've used Splunk in the past, and it's extremely powerful to be able to basically do like searching and parsing of logs and processing of logs in a much more powerful way.

We're trying to find the right balance between whether we operate it or it's in the cloud from a PI and sensitive data information. Second of all is how powerful it is to actually use and be able to troubleshoot stuff and filter things. Number three, obviously the costs, because the challenge with data, it's exponentially growing especially log data. So you want to like have the right thresholds and you want to have enough data to be able to debug any problem, but you don't want to have like every single piece of data, because that's just not economically viable or useful and it's too noisy.

[00:31:35] JM: On the frontend integrations, you've got a number of integrations you got to make. You got to integrate with the credit card networks. There're bank integrations obviously. How much out of the box SaaS is there for doing these kinds of integrations like integrating with credit card networks. I know you have Plaid for doing banks. I don't know if you guys use that. But I'm just wondering how tough it is to integrate with these external systems.

[00:32:04] CN: I think it's probably one of the hardest things to do so. If I think about fintech companies in general, the way I imagine it is you have a ton of wheels that are moving that need to be in sync and you're constantly like tweaking these other wheels and adding new wheels into the mix into your board and you constantly want to make them work. I think that's extremely challenging. We have obviously an integration with the networks to be able to process. That's extremely complex. There're certification costs, there's maintenance costs. Even then, like the thing that's really interesting with financial institutions as they've been around for a while, so they're very robust.

The flipside of that is despite having very robust kind of specifications and formats, the data quality isn't necessarily always – It's not going to be a hundred percent perfect because it moves through so many systems. For example, you might have a field that's like a date, and the date might be in a particular format.

Well, sometimes that date comes up as like 1-1-1900, or sometimes a date might come in a different format because it went through some other system that for some reason had a different format. You always have these kinds of exception cases. Another one that I've seen in the past is authorization code. MasterCard will give you a code in the message, but that code might be all zeros occasionally, because somewhere something got messed up. I think those are very challenging.

Then you have integrations, like Plaid, Finicity and other like bank integrations. Those are challenging because they are very customer-facing. I think the challenge there is, one, you wanted to have a very seamless product experience. So want it to look like your own product. Number two, they themselves integrate with a bunch of banks. If you think about Plaid as an example, all they've done is they build a bunch of integrations with different banks to be able to get your transaction data.

Initially, the way they used to do that was scrape a website. Obviously, that's not very reliable because it can break off and if you change the website. Then in the last few years, you've started having multifactor authentication. So that's been a challenge with some banks where you might have connected your bank account once with Plaid and then like after some period of time it gets disconnected because you need to refresh your token.

That industry is constantly trying to keep up and trying to maintain the data accuracy, and then you have a bunch of integrations on the backend side of things. So we integrate with a bunch of accounting systems to be able to have your data from Brex exported into that. Some of those are push where we push data from Brex into them. Some of them are pull, where they pull data from us through APIs. But you obviously, again, have to integrate all those and maintain those and make sure they're accurate because you don't want to mess up people's books.

[00:34:44] JM: Okay. Do you have like an ongoing integrations team that keeps up with these things?

[00:34:51] CN: We have different teams that own different systems. As part of her card team, we have a card processing team that actually owns the rails and integration with MasterCard.

Our identity team, which controls all the boarding flow, actually has integration to get access to your bank account data with Plaid, Finicity, etc.

We have a financial integrations team that does integrations with ERP systems like QuickBooks and Zero and NetSuite. It's not a single integrations team, because the shape of the problem tends to be different. They kind of own the business problem. In general, I tie to organized technical engineering teams around business problems. They own the systems that solve a particular business problem rather than just like owning integrations.

[SPONSOR MESSAGE]

[00:35:39] JM: When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

Go to softwareengineeringdaily.com/g2i to learn more about what G2i has to offer. We've also done several shows with the people who run G2i, Gabe Greenberg, and the rest of his team. These are engineers who know about the React ecosystem, about the mobile ecosystem, about GraphQL, React Native. They know their stuff and they run a great organization.

In my personal experience, G2i has linked me up with experienced engineers that can fit my budget, and the G2i staff are friendly and easy to work with. They know how product development works. They can help you find the perfect engineer for your stack, and you can go to softwareengineeringdaily.com/g2i to learn more about G2i.

Thank you to G2i for being a great supporter of Software Engineering Daily both as listeners and also as people who have contributed code that have helped me out in my projects. So if you want to get some additional help for your engineering projects, go to softwareengineeringdaily.com/g2i.

[INTERVIEW CONTINUED]

[00:37:28] JM: Talk about the management structure a bit. Can you describe the management structure of the overall company?

[00:37:33] CN: Yeah. At the company level, Pedro and Henrique are the founders. We then have engineering product, go-to-market, operations, finance, legal, and people. There are leaders for each of those organizations. Within engineering, I have a few different groups. There's a foundation team that focuses on all core infrastructure. Data team that focuses on all the underwriting and data analytics and data platform. Trust team that focuses on IT and security. A platform team that has all the common product experiences. Our boarding flow that I just mentioned, our accounting integrations that I just mentioned, and then we have teams focused on each business. We have our card business, our cash business and our ecommerce business. Each of those kind of optimize for their own business.

[00:38:22] JM: How does a foundation team compare to a platform team?

[00:38:26] CN: Foundations team has all the core infrastructure, so managing our AWS instance and our observability stack, developer productivity, like built systems, etc., all the common libraries. You can think of that as your general infrastructure team that exists in any company. You can take that team and you will see it replicated in almost any tech company.

Our platform team is very specific to Brex in terms of the product experience. If you think about, let's say card and cash, there are two different products, but we want people to onboard with Brex. They're signing up for a Brex account and then you'd get all these products. It doesn't make sense to have it in either of those teams. It's kind of shared.

Similarly, your accounting integration, if you have a card and you have a cash account, you want your data to flow into QuickBooks seamlessly. So you don't want to have two different integrations that the card team has built for the card and the cash team has built for the cash product. You want to have a single integration. Anything that's kind of like shared product experience is in the platform team.

[00:39:24] JM: Got it. What are the engineering principles that are true across the organization? Do you have any like 10 Commandments style things that it is true across the whole organization?

[00:39:38] CN: Yes. The company we have, company values. We have a value called Dream Big, generally, people at Brex are extremely ambitious and motivated. Seek truth, basically thinking very rigorously, going beyond the status quo. Owners mentality, basically making sure that we make decisions and optimize for the team and the long-term success of the company. Being a fixer, getting things done the right way, and then customer focus, empathizing with the customer.

Recently we rolled out engineering principles that are more specific to engineering. The first one that I have is accelerating the business. In general, the way I think about it is engineering is one of the only functions in a company that can actually exponentially change the trajectory. What I mean by that is if we think about most functions, in order to get more throughput, whatever that

means, you add more people and you get an incremental value. Engineering is the only function where basically I can have a PR that like literally automates work that hundreds of people are doing, for example, or can unlock like a tremendous amount of new functionality.

In order for that to actually work really well, you have to think about it very much from a business perspective. Otherwise, you're just doing engineering for the sake of engineering. The first thing that I have is basically like thinking about how you can accelerate the business. How is your work directly accelerating our business?

The second one is efficient thoroughness. We just talked about all the different systems that are in play. The payment space has like a lot of history, a lot of complexity and it's really important to go really deep and being very thorough about your work, because we're talking about people's money here. It's very different if like I click like on a Facebook page and it doesn't register. But it's very different if I have a transaction and it shows up twice or doesn't show up at all. Being efficiently thoroughness is extremely important.

The third one is pragmatic optimism. I think it's probably more about emulating me. Like in general, I tend to be a pretty optimistic person. But the reason why I created that principal was I want people to, in general, work well together. We're a very cross-functional company, and in order to solve problems, you need to have people with different experiences working really well together. What I wanted is to foster a culture in which debates are totally fine, and we want to have rigorous debates about any topics. But you want to move the conversation forward. You want to constantly like go towards a solution.

If someone comes to me like an engineer from one team, goes to another team and it's like, "Oh, I'm trying to do this thing." The other person [inaudible 00:42:08], "No. That's a bad idea. Don't do. Because that's not very helpful. I can understand you have some concerns," but the ideas I how do you like understand what the person is trying to solve and try to move things forward. In general, kind of having this like optimistic view. I kind of call it like usually analogy of improv. We are constantly saying yes and, and then similarly when you have debates, like you should figure out how to help your person move things forward.

The fourth one is preserving optionality. I think this is a very interesting one. I was afraid that this is going to get a weaponized, but it hasn't. I'm very happy to see. I think it applied from two perspectives. From a business perspective, we're in a world in which our business is growing way faster than we can keep up with. Teams are always understaffed. They always have way more problems to solve than they can. So it's really important to make sure that you prioritize the things that matter today and like things that you don't have to worry right now, just [inaudible 00:43:02] later. But in doing so, it's really important not to put yourself in a corner where future-self cannot actually get out of. That's one aspect of preserving optionality.

The step one is on technical decisions. We use a lot of [inaudible 00:43:15] services. We're on AWS, and I think it's fine to decide like how in the future we might want to own this piece of technology and build it our ourselves, but right now it's not worth it or we don't want to do it, but you should leave yourself the optionality to be able to have the flexibility of doing what you want in the future.

A concrete example, when we first launched the Brex product, we actually weren't directly integrated with MasterCard. We're using a service called Marketa, which was this third-party that does basically [inaudible 00:43:42] and processing. But the way we used it was in such a way that we could very easily move over to MasterCard when we have the rails done. It was pretty much a decision to accelerate the business and ship the product sooner while still being able to provide kind of similar experience.

Then about a year later, we then switched to MasterCard once we had that built. In order to do that well, we had to basically make sure we have control and decide whether the transaction is approved or not. We were the ones sharing the statements, which most companies didn't do. Most companies would let the third-party make all those decisions because it's the easy way to do it. But then if you actually want to move off, you can't. You're kind of stuck.

Again, preserving optionality has been quite important for us. Then lastly, it's being part of your craftsmanship. I really want people to be happy about the work that they do feel proud about the code that they write. I've always looked at code as art to some extent and like there's definitely times when you're really proud of it and there're times when you know you're just like hacking

something. We all know that both happen, but I want like 90% of the time people to be proud of the code that they werite.

[00:44:48] JM: You worked at Stripe for a reasonably long period of time before joining Brex. I think of both Stripe and Brex as very well-run fintech companies with a great entry point into a very broad fintech market. How do the two companies compare to one another?

[00:45:10] CN: Yeah. I mean, thank you for the compliment. [inaudible 00:45:12] companies. Stripe I think has an incredible job at basically making it extremely easy for companies to monetize their product. They've built incredible APIs, and as a platform company, they do a really good job being able to power all these businesses.

Brex is kind of on the opposite side of the technical spectrum where we are powering the company's finance departments and kind of financial services and we kind of like – If you think about Stripe, what I mentioned, there's like payment processing for a merchant. That's like the top of the stack. That's where Stripe is predominantly focused. Then we are on the issuing side where we are giving you the credit card, cash account, etc.

They're kind of like two ends of the business. In both cases, I think the companies have ties into their mission around helping growing businesses. In general, helping businesses focus on their core business rather than having to deal with the complexities of financial systems and services and the payment space. But they've just taken it from a very different angle. One is more like platform payment processing. The other one is financial services.

[00:46:22] JM: Are there any particular tactical elements or philosophies that the two companies differ on?

[00:46:29] CN: I mean, if I think about some of the differences that I see, go-to-market is very different. Brex kind of got really well-known at least in the Bay Area through advertising, and now basically it's quite popular based on that and a bunch of other companies have duplicated that, and it's because you kind of need a go-to-market muscle. You need a strong marketing muscle. You need a strong sales team to be able to actually sell financial services, because

otherwise people will just like they'll Google credit card. They'll probably go with like American Express [inaudible 00:47:01] bank. So you kind of need that muscle.

On the other hand, for Stripe, they're much more of a platform company. It's much more self-serve from that matter. You typically – If you think about the predominantly like new companies are getting started, you need payments. You're going to integrate with stripe and then you're going to collect payments and you kind of forget about it. That's a compliment, stripe work so well. Whereas with your credit card, you're constantly like using it. You have rewards. You're interacting with the product. You log into our portal. You submit receipts. The engagement level is also quite different there.

[00:47:36] JM: All right. Well, last question. From your vantage point after spending a lot of time with these two companies. How do you think fintech will change in the next five years?

[00:47:45] CN: I mean, you can obviously see based on the macroeconomic situation right now happening just like how fast things are changing. I think you're going to see, on the consumer side, I think a bunch of new banks that are going to come in and try to innovate that space and make it much easier and much more affordable for people to pay in very different ways. Like you've seen that with a firm. You're now seeing Monzo in the US, N26 and so on.

I think on the corporate side, what I think you'll notice is most industries is you have a lot of different tools and you want to unify them or make them work really well together. If I think about the corporate side of fintech, you're going to see a lot more integrations that are happening between companies directly or within a single product offering that they just provide a broader set of products.

Obviously, there's the crypto space, which I think is an entirely different direction. I think that's much more speculative. Either it will be where it is kind of today or you will be vastly different, because something unlocked and became huge and the applications of crypto will probably vary. But in general, the only direction that I've seen over – I guess I've been doing now payments and fintech for about 7, 8 years is, one, simplification. People want singular things. They don't want choices. They just want something that works. Two, unification. They just want things to work well together.

[00:49:13] JM: Okay. You say there's some particular problem with the current banking system that the newer challenger banks might tackle more effectively. Sorry. I know I said it was the last question.

[00:49:24] CN: No. No. No. It's all good. I think the challenge that you see on the consumer side with the current banks is the rate of innovation hasn't been very fast. If you think about it, you have a mobile app. That's probably the biggest innovation. You have your like debit card or credit card as a consumer. I think where you're going to start seeing is banks who are going to try to make it a more active service where you'll have I think more insight into your financials to help you make better financial decisions. I think it will offer you ways to make payments more flexibly, and you've seen that a little bit with Apple. I think security will be a big focus of how do you actually make payments more secure. Peer-to-peer payments is becoming more common, splitting bills. Think about that scenario today, right? Either throw a bunch of credit cards on the table or you do a bunch of math or use a third-party app. It's not very like native. For a consumer scenario, like you can easily see that the being tackled. I think, again, kind of like raising engagement levels from a purely transactional system where I give you my card and use my card and once a month I just pay it, to a higher level of engagement where you use multiple services to help you make payments.

[00:50:33] JM: Cool. Cos, thanks for coming on the show. It's been great talking to you.

[00:50:36] CN: Yeah, likewise. Thanks so much.

[END OF INTERVIEW]

[00:50:47] JM: Software Engineering Daily has over 1,000 episodes with lots of interviews with engineers from Google, Facebook, Uber and lots of other engineering companies. We also have interviews with investors. We have interviews about the philosophy of technology and culture and strategy around starting a software business. You can find all of our episodes in the Software Engineering Daily app for iOS and Android. These apps have all of our episodes sorted and searchable and easy to find in categories with related links and commenting

features. You can see our greatest hits, the most popular episodes that have stood the test of time.

If you don't want to hear advertisements, you can become a paid subscriber for $10 per month or $100 per year. Just go to softwareengineeringdaily.com/subscribe. We have put a ton of work into building the apps for Software Engineering Daily. We're creating the best listening experience for our users, and you can check it out today by downloading the Software Engineering Daily app from the iOS or Android app store, and I'd love to get any feedback you have on the apps or the show. You can always email me at jeff@softwareengineeringdaily.com.

[END]