EPISODE 16

[INTERVIEW]

[0:00:00.3] JM: Charity Majors, welcome back to Software Engineering Daily.

[0:00:03.6] CM: Thank you so much for having me. I'm looking forward to talking to you about Parse and Facebook and Honeycomb as well. I was looking at your background and you worked at Linden Lab for five years.

[0:00:16.9] CM: Yes, I did. Yeah.

[0:00:19.8] JM: This was starting back in 2004, for those who don't know, Linden Lab makes Second Life, which is a popular online world game. It's virtual reality before virtual reality. Tell me about the software architecture of Second Life.

[0:00:36.5] CM: Oh, boy. It was distributed systems before distributed systems were cool. So many of the things that we did, like database sharding before sharding was a thing – we also had to invent our own version of Chef, or Puppet or whatever. We literally are synced the entire operating system over every night for thousands of servers.

Seemed like a good idea at the time. The tools just did not exist at the time. It was such a great place to work. That was college for me. I was too young to know any better and we were growing really fast. This is before Facebook too, so it was – it was a great Wild West. Everybody knew that something was going to be huge.

For a while there, it looked like it might be Second Life. Fun fact, Second Life, Linden Lab is actually still around, still very profitable. They make a 100 million dollars a year. It's the greatest failure of Silicon Valley.

[0:01:35.2] JM: What's hard about scaling a virtual world?

[0:01:39.9] CM: What's hard. What was hard then was that we had – it was like, we were on a desert island. We had to build our own forks and spoons out of – things didn't exist. Right now, there are so many components, you have your pick of them. It's like, well I want this library. I want this automation framework. I want this stuff. Back then, we were really pulling ourselves up by our bootstraps.

Beyond that, what's hard about scaling a virtual world is the users. The fact that everybody's so emotionally connected to – and they make up these conspiracy theories about how the Linden's are trying to manipulate for their personal gain, or because they have this very conspiratorial worldview. People care, they get invested and you hate to break their hearts.

[0:02:30.8] JM: You joined in 2004. That was before the cloud became popular. A lot of your writing and your speaking in a modern sense is about modern operations with the cloud. How was operations different in the pre-cloud days?

[0:02:50.8] CM: Yeah, how was it different? God, I remember driving to the colo to flip the switch in the MySQL primary so many times. Virtual hands didn't exist, let alone there being an API that we could SSH in and just destroy it and bring up a new one. Those are the battle days. I don't really like to think about that. I swore an oath five, six, seven years ago that I was never going to their Colo facility again. As far as I'm concerned, Harvard does not exist. That's the level of abstraction that I'm willing to work at nowadays. It's better.

[0:03:24.9] JM: You worked at several other companies between Linden Lab and eventually joining Parse in 2012. We'll skip over some of your career, although I'm sure there's plenty interesting stuff in there. Tell me how you found out about Parse.

[0:03:43.1] CM: It's so funny. I basically have two types of jobs in my past. There's the ones that I've done for a year and a day, because I felt that sense of obligation to stick it out long enough. There's the ones where I have fallen in love, right?

With the company, with the mission. It's so funny, as an infrastructure person, I fed myself this lie for years that I didn't care about the product. That I was just an infra-nerd, that I just didn't

care what was going on up above me. I've toiled out in the salt mines of servers and racks and databases. That was a dirty lie. I have always cared about the mission. I think that all of us do.

Nothing is so wonderful about seeing people use what you make and seeing people get excited and seeing people build their own things on top of it. I mean, the thing that I love about being an infrastructure is that you're a force amplifier, right? You're building the infrastructure for everyone else to do their creative stuff on. I guess, that's the conclusion that I came to after being at a couple of places where I genuinely didn't care about the mission, right? What brought me to Parse was funnily enough, a recruiter.

The only time in my life that this has ever worked on me, a recruiter just happened to cold e-mail me while I was sitting there, just not engaged in my work at all. I realized that I missed it. I was living the dream, some of my friends thought, because I was working an hour to a day and getting great reviews. Everybody thought I was amazing. I was bored out of my skull and I needed something interesting and challenging and hard.

This recruiter from Parse happened to e-mail me and I was like, “Platforms. I think I like platforms.” I responded. I was her first infrastructure hire, actually. It was pretty beta. It was just five, six, seven software engineers in a tiny little office. That's a niche that I've enjoyed in my career is being at first or very early hire who comes in when bunch of software engineers have a crazy idea that they think might just be a real thing and then I help them grow up.

[0:05:52.0] JM: Tell me about the product of Parse at the time you joined and how that product was implemented.

[0:05:58.8] CM: Yeah. At the time I joined, it was just a few mobile APIs on top of MongoDB. I got to become a MongoDB expert real fast. MongoDB at the time was a very immature product. It had a single global lock per replica set, which for a multi-tenant system was interesting to run. Over time, it grew up into being a very robust ecosystem of basically the best way to build mobile apps. I belong with Parse. I will never forgive Facebook for killing Parse, man.

[0:06:35.9] JM: We’ll get there.

[0:06:37.5] CM: As an app developer, you could just – with a single pasted line, you could create an app and just start focusing on what made your app different and cool and fun, without having to do all of the boilerplate, the stuff on the backend that was just the same for app after after app after app. We just took care of it. Push notification, which is a simple API. Storage is a simple API. Login is a simple API. It was groundbreaking in that way

[0:07:05.3] JM: Parse is a back-end as a service, a mobile back-end as a service, or is or was / both. There were several different storage systems eventually. It started off with as you said, Mongo. Over time, it had Mongo, it had Cassandra, it had Redis, it had MySQL. Why were there so many different database types? How did the data model advance?

[0:07:31.0] CM: Yeah. There's two different things that distinguish here. There is the databases that we needed on the back-end to make the platform work, which is MySQL was always our source of truth for customer data, like passwords, login stuff. We knew better than to trust that to Mongo. Relational database, it was boring, it was safe, right? That's where all of our – the parse data went. User data and went to Mongo, which was honestly, I don't think Parse could have been built without MongoDB, because it was so flexible.

We had great heavy apps. We had read-heavy apps. Apps are doing 3D modeling. All of these things were possible in MongoDB. I didn't require you to define a schema upfront. The schema changes alone would have made it almost impossible to build Parse on MySQL or PostgreSQL. PostgreSQL later added JSON support. At the time, it was really MongoDB or nothing.

Cassandra was the backing store for our analytics product. Your mobile analytics was like integrating tech toy with everything else. Redis was – Redis is not a database, or this is a cache, right? It was the cache that backed our entire – we ran push notifications millions a day, off of a single core on Redis. It was incredibly fast and efficient at that. Never put any data that you actually care about into Redis itself. Just don't do that. We needed all of those in the back-end to make to Parse work.

Now later on, we actually implemented an open parse type strategy, where you could hook up other databases to using your – for your APPA data too. You can be backed by Heroku, you can

be backed by PostgreSQL, you can be backed by anything. Because here's the thing about data, when you're getting started, when it's small, it's easy. Everything's easy.

If you're lucky and you're getting traction, you cross an event horizon beyond which that is not true and you will have to know your data. You'll have to know your data model. You have to know things about the query planner. You have to know things about how the data store underneath it works. There's just no way around that. This is something that we and our customers ran into like a brick wall around two years after I joined, because we suddenly had all these big customers who had gotten started really easily.

Now they got big enough that they need to understand things about their data model that we didn't surface. We were on the phone with them explaining to them what made for an efficient query and how that translated into our SDKs, which is not the situation you want to be in ever.

[0:10:11.3] JM: Parse grew very quickly. Being in a startup like Parse sounds both nerve-wracking and exhilarating. It's the thing, you catch lightning in a bottle only so many times in a career, if ever. Do you have any particular memories of the adrenaline during that period of time that – particular memories of outages, or difficult problems, or milestones. Things that got your adrenaline going?

[0:10:45.6] CM: Oh, so many. Back then, it was me. I was on call for about five years straight, four years straight. I had hired a couple people. At the time, we were not making software engineers be on call for reasons that are no longer true. It was just us and ops, right? Ops was such a key pillar of Parse. We were outsourced ops for a million mobile apps. I have so many memories of getting woken up, tending overnight, rebuild index builds.

I love fire Friday. I'm not ashamed to admit it. I have always gotten this serious high off that shit, and it was really fun. However, it did start to wear on one. There came a point when I was realizing that. Parse was built in Ruby, right? We had a pool of Ruby apps all talking to MongodDB back-ends. The number of MongoDB database has started to balloon. Now you had a single pool of Ruby apps all talking to say, 20 databases.

Now the Ruby on Rails model is such that if a thread is being held open to a back-end that's not responding, it just hangs and there's no real way to interrupt it without doing some seriously hacky garbage, which all which we did. The teal of the art is that when any one of these data storage sources got slow, the entire unicorn pool would fill up within seconds before anyone could react. All of parse would go down for everyone on all of the shards.

This was exciting and fun the first couple times it happened. When it got to a point that it was happening a few times a day and there was nothing – we couldn't react fast enough, we couldn't catch it fast enough, we couldn't kill those fuckers, we couldn't – so we'd have to go and manually blacklist of replicas set, or figure out which app it was and blacklist that, and rinse and repeat hour after hour, day after day. That got old real fast. That's when we started to grapple with the fact that this was one of those rare cases where a real, actual rewrite was called for.

We could have rewritten in JRuby, which would have meant rewriting all the gens, none of the [inaudible 0:13:11.0]. We ended up rewriting it in Go. This was the right solution. It took as you can predict, longer than we thought it would. It was harder than we thought it would be, but ultimately, yield a – an app here that was written in the like with threads. The threads could just timeout and we could bring up as many back-ends as we needed, etc. I don't know if you want to get into that little saga, but yes. Adrenalin, absolutely. It is a magical time to be in a startup.

I will point out that it was rocket ship growth for free apps. This is a lesson that Christine and I, my co-founder took to heart. We took three years to introduce a free tier to Honeycomb, because we saw how much blood, sweat and toil went into apps and we're effectively disposable and nobody was – free users are the worst users. They are so demanding and they do not give a shit and they're giving you nothing.

[0:14:16.6] JM: That's true. This is actually something I heard Ilya discuss on another podcast, where he was talking about some of the lessons from Parse. I think one of them was that well, if you are one of these systems with the free tier, the way that your business model works is often, you have a very small percentage of users, like we're talking 1% to 5% maybe who are paying customers.

If you think about Heroku, of course. Most of the people are on hobby dinos. They're doing everything they can to keep it free. They're setting up scripts to ping their server just to keep it alive. It's like, they'd rather pay for AWS lambda than for paying for –

[0:15:00.1] CM: It makes no sense. People get attached to free and it makes no sense. They do crazy things.

[0:15:07.2] JM: That's actually a really good point, because once you start with free, it's like, I don't know. There's some loss aversion, or something like that.

[0:15:15.2] CM: Yeah. People feel entitled to it.

[0:15:18.0] JM: Right. Yeah. Anyway, so we can maybe get into a business model lessons, or something later on. Let's continue down this storyline. Eventually, Parse was acquired by Facebook. My understanding of the process is that there were different people who are interested, it was a very desirable acquisition. What are your memories of the acquisition process? Was that totally shielded from you, or was it in your purview?

[0:15:50.6] CM: Yeah. We did not know anything. This is a thing that was reflecting on it. Christine and I have chosen very different model of transparency. We had no idea that they were fundraising, or that they were and talks for acquisitions or anything. I was at a conference giving a talk and we got called back for an all-hands. We had never had an all-hands before, so that was a little weird. We had this all-hands and announced that we've got bought by Facebook. A few people cried. The entire room just stunned and shocked. Just like, “Oh.” Nobody was happy that I can recall.

[0:16:29.8] JM: Really?

[0:16:30.7] CM: Yeah. No. Maybe one or two people. I don't know. We just moved into this great office and we were all – we were all so stoked to be working for parse. We believed in our mission, right? We hadn't really been prepared for this. None of us wanted to be commuting down to the South Bay every day and we were very dubious about what our overlords would do

to the product to the team, all which was completely warranted in more. They gave us our payouts and most of us decided to stick around for a little while to see how it went.

I ended up sticking around for a pretty long time. Bought my house, so I can't exactly complain. It's real first-world problems here. No, we were not stoked as a team. No. Now, Christine and I, we are transparent with our team about this, because the worst part was the shock, that we just had no idea that this might be in the cards. The worst part was just showing up, “Oh, all-hands, cool.” Not even thinking this.

We tell our team when we are considering – we tell our team up to the point where we don't want it to be distracting to people, but we don't want them to feel just blown up with shock. It's their company too, right? They joined the startup, because they want to have this experience of ownership, this experience of figuring things out. We feel that they deserve to be informed of what we're doing, because of that experience, I think.

[0:18:06.9] JM: Although, it's not a democracy though.

[0:18:08.9] CM: No, of course not.

[0:18:10.1] JM: You’re just clear about it. You're like, look, we're going to make it –

[0:18:12.7] CM: We’re clear about it and we care about what they think and feel. I mean, we are doing our best to steer it to a good outcome for our people. That matters way more than our own personal outcomes, but it's not a democracy because you can't operate that way.

[0:18:28.4] JM: It's a pragmatic way of looking at things. Yeah, anyway, so did your – how did your work change after you were acquired? I know it did change.

[0:18:41.9] CM: Yeah, it changed a lot. Probably went and screeched to halt for a couple few months as everything got thrown up in the air and resettled. I became a manager mostly because I felt that I didn't have the ability to tap into the information that I needed without becoming a manager, which was also a pretty impactful realization for me.

I don't feel anyone should become a manager because they feel they can't get access to the information that they want, or because they feel they're not in the loop, unless they're a manager. That's a terrible reason to become a manager. It's how I felt there. I became a manager.

We suddenly have more resources. I got to stop worrying about our AWS bill. That was fun. Started commuting two, three hours a day on a good day down to the South Bay. I started working a lot less. I mean, it was just a huge culture shock. We're all startup kids. We got to work with some amazing engineers there who I still love and keep in touch with. Wasn't used to having my work questioned so much. Wasn't used to showing up in a room and just being looked up and down and had people go, “Are you technical?” I'd be like, “Fuck you. I'm here representing a technical product, a technical organization. I'm here. Just talk to me. I'm a human.”

It was a real culture shock, but I'm glad that we’ve had the experience. I learned a lot mostly about how bureaucratic organizations run, which is a good set of lessons to have. When I was coming out of Facebook, feces were coming to me. For the first time in my life, I had a pedigree.

For the first time ever, I was not a better engineer coming out of Facebook. It was the first time I had to stamp that made people come to me and go, “Do you like a couple million dollars to do something?” Which is why I felt I had a responsibility to the universe to take it. I'm never going to have that happen again. Take it and do something.

[0:20:51.7] JM: Actually, no. I think you probably have that ticket for the rest of your life.

[0:20:56.2] CM: Well, maybe. At the time, that was my thinking was like, “This is never going to happen again. I should do this now.”

[0:21:03.7] JM: I'm glad you've taken it. Coming back to Facebook and Parse. In our last episode together, you talked about how there was not direct synergy between Facebook and Parse. At least not in the way that Facebook and Instagram make sense. There were some synergies, like back in 2013 it was very easy to imagine Facebook becoming some infrastructure, or platform for identity, or something.

[0:21:30.5] CM: Yeah, there are real possibilities for that. Yeah, it was just a mismanaged acquisition. There are two acquisitions that stand at my mind as models of good acquisitions, right? Facebook and Instagram and Salesforce and Heroku. I think both of those were – they're very different, but they were both done very well. Salesforce and Heroku, I think is a – Parse could have been done that way, right?

Salesforce gave Heroku a long leash for a couple of years, at least. They were out on their own. The original crew got to gracefully transition out. There wasn't this abrupt, like, well now you're shipping down to somewhere else. They have their own culture. They still do. A lot of independence.

Facebook for us, I learned afterwards and things that it was basically Zuck and our internal champ – Here's one lesson I learned; acquisitions need to have a C-level champion. Parse did not. It was like a product manager over on the platform side who championed this, who got reassigned or fired shortly after. He and Zuck cooked this up. When it was announced, the C-levels and VPS were just as shocked as anyone else. We never really had a sponsor. We bounced around internally a bunch.

[0:22:48.0] JM: Real quick, it's worth noting. I think this was what? It was what? 85 million dollar acquisition.

[0:22:53.1] CM: I know. I know. Yeah, exactly. We never really had executive sponsor. We had three different orgs within three – within a year and a half. Nobody felt vested in our success, right? That's probably why we never succeeded. Facebook platform had some huge problems that we could definitely have helped with, like ways to make developers stop hating you. Well, stop issuing non-breaking changes, right? Basically, we rolled up and they were like, “How could we make developers stop hating us?” Our API developer said that, stop making breaking changes.

They're like, “Ah, no. Well, we can't do that. What else you got?” We were just like, “Uh.” They wanted to basically – just we showed up and the team, the goddess platform, who again didn't

really have a vision for us, just wanted to envelop us, or merge us into the Facebook platform without any idea of how that would actually work. I don't know.

A lot of this, I may be wrong in some of the details, because I'm getting this for people after the fact. I was not privy to these things that were happening at the time. This is a story that makes sense to me, because it jives with what I experienced, which was just everybody is like, “Uh, we don't know where to put you. I don't what to do with you.”

Certainly, their platform did not take any suggestions from us whatsoever. It's not clear what our utility was to them after a while, which is why they moved us to a different org after a couple of quarters. It was just a collective – Facebook clearly did not have any commitment to its own platform. Once became clear that they would have to do things that conflicted in any small way with their own bottom line.

[0:24:31.3] JM: There are some days where I wake up, maybe I have a little too much coffee, I get a little over-caffeinated and I start a new project and I'm really excited about the project. Then by the end of the day, I'm like, “Why did I spend my whole day on this terrible project?”

[0:24:46.3] CM: That was like Zuck and developers.

[0:24:49.3] JM: Right. He had a few too many espressos. I’ll spend 85 million on Parse. Why not? Seems like a good idea.

[0:24:57.9] CM: Sink or swim. That fun kids.

[0:25:01.5] JM: I mean, maybe you've already answered it to the extent that you can, but is there something that could have been done differently to make Parse a more successful acquisition?

[0:25:13.4] CM: Honestly, Facebook would have to be a different company, I think ultimately. There just wasn't alignment. That became more clear as time went on. I mean, there are a million choices that Facebook could have made differently that would have made it a good

acquisition, but the company that Facebook is today, Parse is never going to be successful there.

What makes me angrier is that I have heard that both AWS and Google wanted to buy it when Facebook was thinking of shutting it down. Facebook declined, it wasn't worth the paperwork to them, so they just shut it down instead of selling it. If true, that makes me furious. Furious, because it was a very valuable platform. It meant a lot to a lot of people.

It helped a lot of people get their businesses up and running fast. It really changed the way we thought about platforms as a service. We had done the work to make it compatible with open source, open platforms, all this stuff. It strikes me as the thing that they would do, honestly.

[0:26:21.0] JM: I mean, I thought the shutdown was protracted though. Didn't it get open sourced, or it's still maintained or something?

[0:26:28.6] CM: That's a period of time in which as I understand it, AWS and Google reached out and were like, "We'll take it."

[0:26:36.6] JM: Well, that's super perplexing though. Because I mean, there was some taking on of technical debt, right? If you are open sourcing it – It is open source at this point, right? It's open source and the original hosted service.

[0:26:48.8] CM: Yes and no, but it's not the original code. It's a Nodejs. The original code was in Go. It was highly performant. It was multi-tenant, but it was so customized to things about our environment. This is just a single – it's for handling a single app. It's not for handling a million apps.

[0:27:10.6] JM: Well, that's bizarre.

[0:27:12.6] CM: Yes and no. I can understand why they did it. It's also a little bizarre. Yeah.

[0:27:21.9] JM: Okay.

[0:27:22.7] CM: Yeah, it's not the original code. No.

[0:27:25.1] JM: Weird. Yeah. Were you gone at that point? Were you gone when they –

[0:27:29.1] CM: Yeah, I left.

[0:27:29.8] JM: You left.

[0:27:31.4] CM: The big thing that I open parse, when it was proposed to us – and in retrospect, I think that this is when the founders, like they were not transparent with us about anything. I think that open parse was when they knew this was likely to come, open parse was their effort to bring it in for a good landing, right? So it would live on.

I remember when Kevin was telling us about open parse and I was just like, “This doesn’t make sense. This doesn't make sense. What am I missing? This doesn't make sense.” It drove me nuts. He kept telling me that I was crazy. No. Just spinning these fairy tales about how it would make sense. It didn't make sense.

Finally, I'm going to disagree and commit. I'll execute on this. I'm a good soldier. I take my team of people. This is the second time that I've taken them on a mission and then it's just been like the rug has been yanked out under my feet. I was tired at that point. I was just like, “You know, I'm going to take this. We're going to finish it, because parse is fucked if I leave.” Then, I'm going to need to go, because I am – this is impacting my health, my sanity. This doesn't make sense and it's driving me nuts. I'm drinking too much and I can't sleep. Then they just told me to leave.

I think that he was trying to do it for my sake, but it was the worst thing you could have done, because I knew that they were all fucked at that point. There was no way that – I had all the institutional knowledge of everything. In retrospect, I realized that I was inappropriately propping up morale for the entire engineering org. They believed that I was there, so they believed things would be okay. They trusted me. I shield them from all these things. As soon as I left, everything did just crumpled. People just evaporated and left.

In reality, I kept them there too long. Those were amazing engineers, great people. They should have been off doing something better with their lives, but because I believed in this company and stayed there too long, everyone else did too.

[0:29:45.3] JM: Isn't it crazy how cognitive dissonance, when a person at the top has cognitive dissonance, but there's – I mean, you were in the top at relative to the engineers. Somebody was above you. They may or may not have had some –

[0:30:00.9] CM: They knew. They knew or they suspected. Yeah.

[0:30:04.3] JM: They knew it didn't make sense.

[0:30:06.7] CM: They knew or suspected that Parse was getting shut down. They were in the rooms where like, they were looking for executive sponsors and all this stuff. They knew. They weren't telling me.

[0:30:17.7] JM: I see. They knew that Parse was going to get shut down and they were selling it to you as if, “We're going to do open parse. It's a next phase of the parse life story.”

[0:30:28.9] CM: Yes. I was like, “This makes no sense.” I wasn't cynical enough at that point to read between the lines. Yeah.

[0:30:39.8] JM: That's brutal. Yeah. Because the cognitive dissonance thing, it grinds at you and then you're experiencing it and then you're like, “It doesn't make sense, but they're selling me this story, like it fits together.”

[0:30:54.5] CM: I trusted them and we’ve been doing this together for so long and I could make myself believe it, if I squint.

[0:31:00.8] JM: Right. Then there's something in your soul that then it just starts to emerge and drinking too much and not getting sleep. It probably rubs off on the engineers beneath you, which is what's terrible. It’s wow, brutal.

[0:31:13.5] CM: It is. I don't know. I do think that some people are better equipped than other people to enjoy the –

[0:31:19.6] JM: Compartmentalize.

[0:31:20.0] CM: - lessons in front of them. Compartmentalize. To not have to understand why. If it's a great project, fun team and you're just enjoying your life, not everyone is cursed with the need to know why all the time. Unfortunately, I am. Yes. I don't know.

[0:31:43.5] JM: Weird.

[0:31:43.8] CM: Yeah, it was brutal. It was brutal. Took me a long time to recover from that.

[0:31:47.3] JM: I still don't quite understand. There's somebody above you. They're literally saying – they knew Parse was going to get shut down.

[0:31:53.8] CM: Well, they suspected. What I think is that they suspected. They're in these rooms where people are discussing the future of Parse and they're seeing that there's not a lot of support, or reason. They're doing their best, right? They're all doing their best. They're doing their best to try and split the pie, or chart the best possible course for the people, the product and the team. They just don't really believe that transparency is the way to do that.

[0:32:25.3] JM: Interesting. It's funny, because that's the second note on transparency in the conversation. Well, I guess it does make sense, because – so if they're basically in a position where they're like, "Okay, this thing is going to get shut down."

[0:32:41.5] CM: Maybe not, right? Maybe there won't be a Hail Mary. You don't want to just tell the team and –

[0:32:45.4] JM: Right.

[0:32:47.5] CM: I can see that from their perspective. It's not the choice I would make, but I can see it.

[0:32:53.2] JM: You see, that makes it how about. Seeing from their perspective and I'm going to do an interview with Ilya, so maybe we can shed more light on this.

[0:33:01.3] CM: Oh, boy.

[0:33:04.7] JM: Maybe I shouldn't have told you that.

[0:33:06.2] CM: It's fine.

[0:33:07.9] JM: If you think about it from their perspective, okay, this thing is going to get shut down. We can't tell –

[0:33:13.9] CM: Probably.

[0:33:14.6] JM: Probably. We can't tell the engineers it's going to get shut down. If maybe there's a vision for making it an open thing and that will make the engineers less spiteful when we finally tell them this thing is going to be shut down, but thankfully we've been working on the escape hatch, the open source escape hatch for the last three months, aren't you relieved that I had you work on that open source escape hatch?

[0:33:40.2] CM: Yeah. I don't think that they expected anyone to be pleased. I just think that – it's like, "What are you going to do? You've got a team. You got to tell them something to do. You don't want to –" I'm sure they feel if they tell one person is going to get out and everyone's going to know or something, they saw us all as being – the very founders versus everyone else. There weren't other levels of privileged information.

Until it's a 100% for sure, you're always hoping that it's not going to go that way. Maybe you've seen near-misses in the past and you're hoping this will be one too. I can totally see it from their perspective. They're good people and they were doing the best for us. I can't really – it was just hard.

[0:34:29.9] JM: We've been doing this series on Facebook engineering. We've told a bunch of stories about how Facebook engineering is different from a lot of other companies. Most of the stories, the engineer who I'm interviewing is a great fit at Facebook and they're able to do some of their best work. They're really happy with the things that they've done at Facebook. It seems like you were less of a good cultural fit. You were not comfortable. You were not totally happy or gratified. I'm sure, a lot of that had to do with the commute, by the way.

[0:35:06.2] CM: Yeah. Not a great start to the day.

[0:35:07.9] JM: Not a great start to the day. Tell me your memories of the Facebook engineering culture and why was it discordant with who you are?

[0:35:14.7] CM: Yeah, that's a great question. In some ways, I think I got off on the wrong foot, because I was so pissed about having to be there. Would you choose into something, you just have a different attitude that when you get pushed into something for sure. There's that. Also, there is a lot of arrogance. I remember one of the first days I was there, I get called into the tech lead. I get called into the head of engineering's office. He's like – I get this feeling like I'm supposed to be super-impressed and supposed to be just wide-eyed like, "Ooh, everything's so cool." It's not and they're pissed about it. He's like, "Oh, so when are you going to get off AWS?" I'm like, "Well, we're not."

[0:36:02.4] JM: Oh, no.

[0:36:04.0] CM: He's like, "Well, of course." I'm like, "No, we're not. We have the ways of why we're not. Our data model means that we rely on this elastic storage 12 terabytes, etc. You don't have the right hardware. It would cost 20 million dollars minimum to run our databases in your infrastructure."

Down the line – the biggest reason of all is at the time, they had a perfectly flat network. We're letting developers all over the world just run random third-party and trusted code on our systems and queries on our systems. On Facebook's network, they could just call out to any Facebook other data. There are so many reasons why we're not moving off of AWS and on to Facebook's network.

He just like – he literally rolls his eyes and shrugs me. He's like, "Uh," dove in these detail. We'll solve it. We're Facebook. Literally he said, "We're Facebook. We'll fix it." I'm like, "I'm pretty sure you won't. I'm pretty sure you won't and you shouldn't and let me repeat my reasons again." I made a lot of enemies by saying this. I held out. My boss to his credit was like, "We're not going to force you to do anything you don't believe in." I held out for a solid two years of the raid force – forces of all of Facebook engineering management being angry at me for this thing.

Finally, I'm like, "You know what? I might be crazy. Maybe they're right. They are Facebook. Maybe all these concerns that I have are I'm just too provincial. I can't get my head around it. Whatever. You know what? It would be amazing if we could expose Facebook services via the Parse API. That would be super awesome. You know what? I'm onboard."

Not only that, I got my entire team stoked about this miserable multi-year project of moving an infrastructure from one to the other, right? Nobody see that fun, but I sold them on it and I carved out this –

I crafted a role for each of them, that was going to push their boundaries, to help them learn new things and set them up for great success at their next – challenge isn't everything. My team got excited. I sent out a status update that we're doing this. I sent regular, bi-weekly status updates of our project progress. We're doing everything in full site. Two months into this, I get a ping from the same director of the structure who's like, "What are you doing?" I'm like –

[0:38:20.5] JM: Oh, no.

[0:38:21.8] CM: I'm doing what you guys have been telling me to do for two years. We're moving into FB infra. Don't worry, everybody knows, the past infrastructure knows, they provisioned it. All the teams know. Everything is going great. He's like, "Well, we can't spare the resources from these other [inaudible 0:38:36.8]." I'm like, "No, no, no. It's cool. We got it covered. We're doing it ourselves. It's not going to be any burden. We've already done this shit." He just threw in – starts calling meetings. My team is getting concerned. I'm like, "Don't worry about it." These dudes have been leaning on me so hard for two years. "It's fine. Don't worry about it. Do your shit."

Finally, my boss pulls me aside. He's like, “I'm so sorry. You don't understand what's going on here. They're not going to let you succeed, but nobody wants to be the one who pulls the plug on your project. It has to take the fall for it. They need you to do that.” I'm like, “Well, I'm not going to do that.” He's like, “Well, for as long as you don't do that, your team will just continue to waste their efforts on things that will never ship.” I went, “Motherfucker. Okay.”

I pulled the plug on it. I got a bad review that quarter for my mistakes in doing this. Imagine get good reviews for my folks, but that killed something in me. That was really what let – it wasn't actually the open parse stuff that led to my – After that, I just started to have ulcers and just – I made the best of it with my team. We printed out a mission accomplished banner and we threw a party, where we – we had a black cape and booze and shit. They were impacted, but not too demoralized. It hit me a lot.

Facebook is a very non-confrontational, very smiley, happy friendly to your face aggressively so, to the point where they won't actually tell you things straight up. You have to find out in the hallway afterwards with somebody who actually knows something. I didn't know everybody, so I just took people up at their word. I regularly had things like this happen in a lower volume. It's the thing where if you're at Facebook and if you're one of the senior engineers who's been there a long time who everyone defers to and gets out of your way and you know, I think it's a glorious place to work.

If you're a junior engineer who wants to get a well-rounded education, get from junior to senior in three years flat, their machine is optimized for that. I think that they're very good at that. For a senior engineer who comes in through the side, it's not always the best.

[0:41:17.4] JM: This the thing about – this is a characteristic of large corporations, the corporate – When people talk about corporate politics, this is what they're talking about. The fact that you have these people who have –

[0:41:31.7] CM: Would do anything.

[0:41:32.7] JM: They will do anything. They will destroy you. They'd want to destroy you. This is the one area of their lives where they can exercise some amount of power and they're going to use it to their full extent. They're going home to an unhappy marriage and I don't know about that exact person.

[0:41:49.7] CM: Maybe. In this case, I think that they just changed and nobody really paid attention and nobody was willing to take responsibility.

[0:41:58.0] JM: The thing is these organizations are – just a completely on point. I don't know anything about that manager specifically, but there are people who are so bitter with their lives. They've gone to a place and – or for some other reasons, there all kinds of perverse reasons where you can end up in a situation where you are the middle of an organization and you are not happy with your life and the way that –

[0:42:19.3] CM: Building your petty powers.

[0:42:21.6] JM: Yeah, your pettiness. It's heartbreaking to hear your side of that. At the same time, I'm glad you told that story because there's somebody out there right now who works at some other big corporation and they've just been a victim of the same thing and they're feeling the same cognitive dissonant thing that you felt. They were like, “Why is this happening? Why does it have to be this way?”

[0:42:46.0] CM: The bigger business gets, the more it has to depersonalize, right? It has to become the machine. Otherwise, things don't work. I get that. I really do. You can find lips of humanity and teams and one-on-one time and everything, but the organization itself is a machine and its job is to depersonalize you. I don't like that, which is why I prefer startups, or everyone is relentlessly human, in-your-face human. You have to get used – it's like, your siblings almost, right?

You spent all those real human time with each other. I think that there are ways to love your job at a big company, but I think that you have to first learn how to depersonalized to – to depersonalize and dehumanize and detach and make it a compartmentalized part of your life

where you just go there and you do the things that the org requires, as well as whatever brings you joy.

You have to learn to separate those. That's my theory at least and I don't know how to do that, or didn't do that successfully, or – I think I could do that successfully at a big company if I had to go and got required. If I had chosen into it. If I made the decision to go into this with my eyes open, knowing that I had a choice. Yeah.

[0:44:09.9] JM: Well, so the last time we talked about the –

[0:44:14.3] CM: I don't mean to be such a downer. I'm sorry.

[0:44:15.1] JM: No, no, no. Actually, I really like that story. That story is so good, because it's – I worked at Amazon only eight months. Your loyalty exceeds mine. Your ability to stay for the year-and-a-half stints at these companies, or your 13 months –

[0:44:33.6] CM: Oh, three years.

[0:44:34.8] JM: I have been more of an eight-monther. I saw this in Amazon. I saw people be victims of this at Amazon. I've heard about people being victims of it before. I've read a number of books where this thing is highlighted. There just seems to be this part of humanity that maybe it is, maybe we do need to depersonalize in these large organizations in order to be more productive.

[0:45:00.7] CM: Yeah. I mean, if I wasn't putting my whole self into that job, if I was just taking my three weeks of vacation every year, if I was coming in at 10:00, or 11:00, checking out at 5:00 and just putting my work aside when I went home, it shouldn't have been heart-crushing. It was because I identified so much with my work and it was so intensely a part of me and it was so – I put too much of myself into it, even after I should have stopped doing that.

[0:45:32.7] JM: The thing is that Facebook is so cohesive. What I wonder is was this a rational action by the organization to freeze you out, because they identified that you were going to be some – you're going to be too gritty? You were not going to be –

[0:45:49.9] CM: I don't think this had anything to do with me. I think it literally was just that they realized that the amount of work it would take to support – it was the technical reasons, which I identified before them on day one. It was too much work and too little time, they couldn't do it.

[0:46:04.8] JM: Oh, it was literally –

[0:46:06.4] CM: That's literally what I told it was.

[0:46:07.8] JM: Overtime you were like, "Yeah, okay fine. I can see the idea. I was excited at this. Whatever. Who cares?"

[0:46:13.0] CM: Yes. Maybe they're right.

[0:46:14.4] JM: By then, they had actually –

[0:46:16.2] CM: No, it was literally what I told them it was.

[0:46:18.9] JM: That's actually just hilarious then. The other thing I wanted to compliment you on is the ability to see that other side, instead of digging in. You could have just as easily dug in and dug in and dug in further, but instead, you did find the other side of it, which happens so often at these software companies, where nobody really knows what they're doing. We're all building these weird products. There's no precedent for. You can see things from different perspectives, if you have the humility to.

[0:46:51.1] CM: Yeah. Well, what was best for Parse.

[0:46:53.6] JM: Yeah. What lessons from the Parse and Facebook experience have you brought to Honeycomb?

[0:47:02.3] CM: Oh, boy. Well, I wrote a whole blog post actually right after they announced Parse getting shut down where I talked about my acquisition lessons. If we ever go through an

acquisition, it will be things like have a C-level sponsor, make sure there's alignment. Look for a period of independence, etc.

As far as what we brought to Honeycomb, well, one of the – a couple of them I'd mentioned already. Christina and I feel very strongly about transparency. We see this not as our company. We're the founders. There's some big bright line between us and the employees. This is all of our company. We've tried to be more generous with stock than – as much as our investors will let us, because it doesn't seem fair for us to get orders of magnitude more than people who have been here almost as long.

We are not a democracy, but we democratize access to information as much as possible legally and ethically. We do try to be transparent with people about where we're at, including the couple times that we've considered acquisitions. We've told people, “Hey, you're going to see some folks walking around. We're not planning getting an acquisition.” We're not wanting an acquisition, but we are talking to some folks. If you have any questions, let us know, that sort of thing. It will never be a shock if and when something happens. It will never be a shock to our team. I can guarantee that.

Another lesson that we took away was the importance of not giving away our product, at least not too much, not too early on. We do have a free tier now and probably waited a little too long and maybe overlearned that lesson. We did not want to have the army of entitled freezers eating up expensive storage that we did a Parse. Sorry.

[0:49:03.5] JM: What's true about building startups today that was not true when you were working on Parse?

[0:49:12.2] CM: Oh, that is a great question. Well, a boring answer is that from the investment side, the landscape has changed a lot. Back then, there was a lot of willingness in the side of investors to take risks and big plays that there was a sense that there was this – there are going to be some big players and there wasn't really this straight-up recipe, like at series A you must be this many dollars of revenue and this much of a [inaudible 0:49:43.6]. Series B, it must be this much and this much. It wasn't that strict as we're encountering today, which makes things a

bit harder. The rules that New Relic was raising money under were vastly different than the rules that we are raising money under today, which is unfortunate, but is what it is.

Another is that these platforms of as service have a couple on serverless is now the big hot thing. That the margin for people, cost for building a really cool product just keeps dropping like a rock. The developer, another thing that's changed is how much developers can and should be in the driver's seat for their own code, right? They need to be – I think I was the last of the generations at Parse where it was reasonable for me to be on call, instead of the software engineers who were writing the code.

It was reasonable at the time. It was the right decision, but I would tell anyone starting out today no, no, no, no. It's almost flipped. I would almost say, if you have one SRE and a bunch of software engineers, the SRE should not be on call, because they're always backed up on call. Instead, the engineers who are writing and shipping code every day should be in the rotation.

[0:51:12.2] JM: You were CEO of Honeycomb originally and you decided to switch to CTO. What drove that decision?

[0:51:20.7] CM: CEO is the worst job in the world. It's the worst. I never wanted to be CEO. There are originally three of us that started Honeycomb and we had to let one of them go a couple months in. I hadn't really realized the implications of that. I thought that was shit. I mean, someone else has to be CEO, and so that was me. I tried real hard to give that job away for a year. I never loved it. I always hated it, but I did it because it needed to be done and I'm fundamentally very motivated by what needs to be done. I will do not what's fun.

I also believe that you don't really do a good job at something, unless you enjoy it. That was okay for a while, because often, I start out not liking something, but then I grow to like it over time. That never happened with the CEO job. I don't think I was doing a very good job. I hate spreadsheets, dude.

Christine is so much better at spreadsheets. She's better at meetings happen after happen, at the same day, same time every week after week after week. All of these things, the predictability, the dealing with finances. I'm terrible with finances. All these things –

I got us through the period where nobody knew about us, and so the ability in fact that my name is a little bit known. I could draw some attention. There was just a lot of ad-hoc running around and crap. Now that we're settling into a more predictable cadence and we have big enterprise, customers that need a steadier hand on the sales motions and everything. I have no patience for sales either. It's just a much better place for Christine.

[0:52:58.8] JM: All right, final question. This is a broad question. It can span over any part of your career. What do you wish was different in the tech culture?

[0:53:13.0] CM: What I wish different in the tech culture. That's a really interesting question. I love tech. I feel we criticize from a place of love, but I feel we often gloss over the fact that things are getting better for almost everyone intact. It's so easier for us to see that the failures in the places where people are being hurt. It's good that we can see that. Years ago, we couldn't see that.

When I grew up, I became an engineer because I wanted to be where all the woman worked, because I grew up in a fundamentalist home where women just had babies and cooked and I didn't want to do that. I ran as far away from home as I could get. I have been learning to be a much better feminist in my adulthood.

I will say that tech doesn't have enough sparkles. I really enjoy aggressively feminizing male-dominated spaces, so I always tried to bring a bit of sparkles and pink. We need more nail polish swag, basically. Just girly shit associated with tech. That's my off-the-cuff light-hearted answer.

On a deeper level, I think that it's – the thing that I would change about tech is just – instead of making it everybody's chasing the few X Facebook, X Google people and funding them, just make it a lottery. Honestly, I think most hiring processes can be replaced with the lottery and [inaudible 0:54:51.7] off. I think there's too much policing and guarding around access to these things that are the best team, or the best company, or the best engineers and the best. Fuck it.

You're not actually – It's the idea of having an investment advisor, versus just putting your money in mutual funds and how the mutual funds usually out – I honestly think that a lot of would out-earn and outperform most of the processes that we have around hiring and early stage investing.

[0:55:24.1] JM: Well, that's interesting. All right, well very good thoughts to leave us with. Charity, thank you so much for coming back on the show. I have really –

[0:55:31.5] CM: Thanks for having me.

[0:55:32.4] JM: I have really enjoyed the –

[0:55:33.2] CM: Hope I wasn't too much of a downer. That was interesting conversation.

[0:55:35.3] JM: No. Not at all. Not at all. I mean, if you've listened to any of these interviews, they've mostly been glowingly positive reviews. It's just nice to know that Facebook is not a place that is insulated from the underbelly of human nature driven by corporatism.

[0:55:52.4] CM: Not at all. Not at all.

[0:55:54.9] JM: Okay. Well, thank you –

[0:55:55.7] CM: Thanks for having me.

[0:55:56.2] JM: Thank you, Charity. Great to talk once again.

[END]