EPISODE 560

[INTRODUCTION]

[0:00:00.3] JM: Social networks can make you feel connected to a global society, but these social networks are controlled by a corporate entity. The profit motivations of the corporation are not directly aligned with the experience of the users. Mastodon is an open source decentralized social network. Eugen Rochko started building Mastodon in response to his dissatisfaction with centralized social networks like Facebook and Twitter. In the mastodon model, users can run their own nodes and other users can connect to them. You could follow users whose accounts reside in other nodes. This model is the federated social network model.

Eugen joins the show to discuss how Mastodon works and how its thousands of users interact on the platform. We explore the open source community that is building Mastodon and we speculate on the future of social networks.

I want to quickly mention before we get started, Software Daily is a place to post your own software projects and discuss them with other people. We built softwaredaily.com with our own open source community, and you can come be a part of that at github.com/softwareengineeringdaily. With Software Daily, you can find collaborators and feedback for your own software project. If you have an open source application or a side project or a computer science paper that you're writing and you want to get feedback from software engineers, come to softwaredaily.com, post your project, we'd love to see what you're building.

With that, let's get on with this episode, an interview with Eugen Rochko.

[SPONSOR MESSAGE]

[0:01:51.8] JM: Users have come to expect real-time. They crave alerts that their payment is received. They crave little cars zooming around on the map. They crave locking their doors at home when they're not at home. There's no need to reinvent the wheel when it comes to making your app real-time. PubNub makes it simple, enabling you to build immersive and interactive experiences on the web, on mobile phones, embedded in the hardware and any other device connected to the internet.

With powerful APIs and a robust global infrastructure, you can stream geo-location data, you can send chat messages, you can turn on sprinklers, or you can rock your baby's crib when they start crying. PubNub literally powers IoT cribs.

70 SDKs for web, mobile, IoT, and more means that you can start streaming data in real-time without a ton of compatibility headaches, and no need to build your own SDKs from scratch.

Lastly, PubNub includes a ton of other real-time features beyond real-time messaging, like presence for online or offline detection, and Access manager to thwart trolls and hackers. Go to pubnub.com/sedaily to get started. They offer a generous sandbox tier that's free forever until your app takes off, that is. Pubnub.com/sedaily. That's pubnub.com/sedaily.

Thank you, PubNub for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:34.9] JM: Eugen Rochko is the creator of Mastodon. Eugen, welcome to Software Engineering Daily.

[0:03:39.9] ER: Hi.

[0:03:40.6] JM: So Mastodon in the simplest terms is an open source, decentralized social network. Explain why you started working on Mastodon.

[0:03:52.7] ER: I was frustrated with the other options. I was a heavy Twitter users since 2008, and so that format was really close to my heart, but the platform was increasingly becoming bad from a technical side and from a social side as well. So at some point I just started looking for other options, because as a software developer, I always had an interest in open source software and in various things that are more complicated, I guess.

So I heard about federated social networks a long time ago, back in 2012, when they just started essentially. That was the time when Google itself was interested in pursuing open standards for social networking. It was before Google+. It was the time of Google Buzz, if you remember that artifact.

[0:04:45.1] JM: I do.

[0:04:45.7] ER: Yeah, that was the first time to learned about them. Well, they didn't look particularly attractive back then, so I just put that to the back of my mind and went on with my life. Then in 2016, I just became curious what happened to that, and given all the circumstances, I decided to contribute. But instead of contributing to an existing piece of software, like social, I decided to basically just start from scratch because it was more interesting. As a Ruby developer, I wasn't super hyped about contributing to PHP code base that was started in 2012.

[0:05:25.0] JM: We will get into the design of Mastodon. Let's talk a little bit about the problems of the social networks that you are starting to identify back in — you started this in — What year was it?

[0:05:40.3] ER: 2016.

[0:05:42.5] JM: 2016. So back in 2016. We were starting to see problems of social networks back then. They've obviously become more acute today. What are the problems of the closed source, centralized social networks?

[0:05:57.4] ER: I mean, it was clear for me for a while that something like where it's supposed to be a public utility, because it's just like the telephone. You've supposed to be able to talk to anybody, and the fact that something like that belonged to a commercial company that had — It's not just that. It is also a commercial company — it is also a company with a history of mismanagement and a company with looming doom essentially.

For the longest time, Twitter had like the prospects of being bought out by Disney or closing down or these things. So I was just thinking, “What happens if Twitter disappears? This will cripple online communications in a way.” I think that's the main problem.

[0:06:46.3] JM: When we use a closed source centralized social networks like Twitter, it may seem like a public utility, but there is a lot of work that goes into the hosting of it. There's the reliability of the service. That's not easy to accomplish. In return for that, Twitter gets to serve ads to us. This seems like a fair trade. There’s something complex to do. Twitter takes care of that, and in return they get to monetize us through advertising. Why would I want to have a different exchange?

[0:07:22.5] ER: Well, is that really an ethical way to monetize though? Because you're not exactly — it's not exactly clear what you're giving away in this exchange. It’s not a straightforward exchange like I'm giving you $5 and you provide me a service. It's more like, you get to use my service for free and I completely exploit everything that you do. I make models of your behavior. I sell it to anybody I like. I try to predict your behavior to manipulate you into buying more products that you don't need, etc etc. So advertising-based monetization is not necessarily a thing that's friendly for the consumer.

[0:08:05.3] JM: More to your point of starting something, I think my devil's advocate argument probably overestimates how hard it is to run something like Twitter these days. All due respect to Twitter, but it's not that hard to run that kind of social network these days at a bare minimum level of reliability.

And the system that you've developed is something that’s somewhat different and has, I think, a more granular level of reliability depending on what node or what collection of nodes we’re talking about being reliable, because people can stand up their own partitions of Mastodon. I think we can start getting into a discussion of the architecture of Mastodon, but I guess before that, we should talk a little bit about like what the user experience is.

So mastodon, like we said, it's open source decentralized social network. Describe the basic onboarding experience for an average new user who is getting started with Mastodon.

[0:09:11.4] ER: When you're getting started, the first step is deciding which server to pick. Which server to become your home essentially. I'll give that - it's a complicated step. It's a step that confuses a lot of people, because it is new. It’s not that new, because everybody has an email address and at some point they had to pick an email server. This is essentially the same thing. Now, you could talk about how everybody basically just picks Gmail. I am not a Gmail user myself, and there is definitely a large number of people who don't just use that. So that system is not necessarily centralized itself.

I know there are doomsayers who say that email has become centralized in Gmail. I don't actually think that's true. Anyway, back on the main point. You pick a server and then you sign-up like you sign up on any other website ever. It's very simple really. You pick a username. You've got your new account. You get a little tutorial stream with a cute elephant on it. That explains, “Here is your complete username.” It includes the username that you picked and the server that you picked. Is essentially just like an email address, and that's what you show your friends. That's the only difference when you're in Twitter you say, “I am Gargaon on Twitter.” When you're in Mastodon you say, “I am Gargaron at mastodon.social,” because that's the server I’m on.

[0:10:37.7] JM: To be clear, when you connect to Twitter, you don't have a server you're connecting to, because you're signing up on Twitter's backend, which is a collection of a bunch of different servers that Twitter is managing and hosting.

[0:10:52.6] ER: There is a line of separation here, where it doesn't make sense to view them under a similar — hey’re not on a same level of abstraction, right? So when you sign up on Twitter, twitter.com is like the server that you pick essentially in the terms that we’re talking in. Of course, in the background, Twitter is hosted on thousands of machines in various data centers, etc., but that's a different level of abstraction.

Now when you sign up on mastodon.social, Mastodon.social is your server. Again, in the background, it runs on a couple of different machines, but the difference is that Mastodon.social is just one of the servers that you could pick. You could pick octodon.social, you could pick [inaudible 0:11:37.8]. Something like that. There're over 2,000 of them and I can’t list them all. But the way that they are run individually can be very different and can go very much in depth.

Multiple data centers, etc, depending on what you're talking about, but the main difference is that there are many and you have this top-level separation.

[0:12:01.6] JM: So an imperfect analogy would be that we all set up separate Slack teams to have our different companies with our own Slack conversations going on in them. Of course, this is an imperfect analogy, because Slack is all centrally hosted once again -

[0:12:20.2] ER: Yes, exactly. Imagine your have all these Slack rooms or channels, or perhaps [inaudible 0:12:27.0] servers may be more familiar to some people, or perhaps you may think of subreddits, but imagine that each subreddit or each [inaudible 0:12:34.5] server or each Slack room was actually completely independently hosted and operated.

But at the same time that you could still talk to all the people in those channels, rooms, servers, subreddits or whatever with your account.

[0:12:50.6] JM: Right. Users can host their own instances of Mastodon, or you could connect to somebody else's instance and create your account on somebody else's instance, and you can discover people within that instance, the instance that you created your account on, or logged in on, but you can also discover people within other instances.

So there is a way of connecting these different instances to one another so you get that global twitter style experience that makes Twitter so magical. Can you describe the model of interaction there? How are these different servers? How are these different instances of Mastodon connecting to one another?

[0:13:33.4] ER: It is a subscription-based model essentially. You've got all these different actors on different servers, and the basic idea of Mastodon is that you follow each other. So you have a following relationship. As soon as you have a following relationship, the server, where the content originates from sends that contents to the servers of the followers, right? That's how content mainly travels over the network.

There is additional ways that kind of spreads to the network, such as boosting, which is our term for re-tweeting essentially or re-blogging. There are other ways such as — there're ways that we pull down data instead of pushing it when we want to just sort of get more context on something. For example, if you're following somebody and they respond to somebody else, your server fetches that conversation, so when you click on it, you see the full thing and not just that one response, right?

At the same time, when somebody responds to somebody you follow, that response is also forwarded to their followers, so when you open the conversation, it's full. So those are the main ways that content travels through network.

Now there are various different terms when we're talking about decentralization. So there's different abstraction layers such as, as you mentioned, Twitter itself is hosted on different data centers and machines. So within that framework, it is decentralized, but it's also a single service operated by a single company. So it is centralized.

Now Mastodon is decentralized as a federated network. That means that there is a difference between that, and for example, a distributed network. Where, for example, I would say that the blockchain is a distributed concept, because you've got essentially the same data, but stored in different nodes.

On Mastodon, it's not the same data that is stored on different nodes. Every node stores its own data and then they exchange that data, but you don't have to contain the entire network if you don't want to. It's all based on what the users actually do, who they follow.

[0:15:44.3] JM: So every Mastodon server can set up their own rules around what's allowed. So rather than having this global policy that's set by a single organization, like Twitter, you can have a variety of communities with different policies and all of these communities can interact with one another through this federated model, as you described.

Could you explain that term federation in a little more detail? Because I think this is the key idea here that this is a federated social network. I think we should define that term federation.

[0:16:22.5] ER: Okay. I don't think I can give you the textbook definition if you're asking for that, but federation —

[0:16:28.0] JM: Oh, that's fine.

[0:16:30.8] ER: It is quite a simple concept. It's just you have nodes that can operate completely as standalone, right? Imagine like countries in the European Union. They could continue to operate completely independently of each other. If all countries but one disappeared, that country could still go on. The same is true for Mastodon servers. If all other service didn't exist, the one server that you have would still be able to go on.

The way this is different from a collection of centralized services is simply that there is a protocol of inter-operability between those so. For example, like in the European Union, there are institutions and laws that are compatible between the countries and that allow, let's say, free travel between them, right? So the same is true for Mastodon. There is an API that allows people to follow each other from different nodes and for content to reach one node from the other, when it needs to.

[0:17:29.7] JM: Once these nodes are set up, they're initially disconnected. What's the process of them discovering one another? Because I imagine if there were some index of the different nodes, that would arguably be a point of centralization, but maybe it's not so troublesome just have an index as the point of centralization. So how do these different nodes, these different private social networks, how do they discover each other so that connections can be formed between the different federated divisions?

[0:18:07.7] ER: We do not actually have a centralized index exactly for the reason that it would be a centralized service. Now, of course, that's kind of a hard problem. When you're starting a new server and you don't know anybody, it can be pretty empty, unless you have a local community that is ready to just start using it, right? Then you still got the problem that it's pretty disconnected from the rest of the network.

So the advice that I give to new people is do not start a new server until you've made an account on an existing one and sort of got a feeling for who you want to follow, and basically you join in existing server that has a large active community and that has already discovered various other servers, and then you find people that you want to follow and then you can export that list of people to follow, and when you start your new server, you import that and then you kind of bootstrap the connections to the other servers.

We mainly rely on organic discoveries. So when people share comments from each other, when people talk to each other, as I previously mentioned, we pull down some content when it's in conversations and when it's shared. That's how more content is discovered and as people see that content and they follow more people, more subscriptions are created and so it becomes more and more interconnected. That's how it works.

[SPONSOR MESSAGE]

[0:19:38.5] JM: We are running an experiment to find out if software engineering daily listeners are above average engineers. At triplebyte.com/sedaily, you can take a quiz to help us gather data. I took the quiz and it covered a wide range of topics; general programming ability, a little security, a little system design. It was a nice short test to measure how my practical engineering skills have changed since I started this podcast. I will admit, although I've gotten better at talking about software engineering, I have definitely gotten worse at actually writing code and doing software engineering myself.

But if you want to check out that quiz yourself and help us gather data, you can take that quiz at triplebyte.com/sedaily and in a few weeks we're going to take a look at the results and we're to find out if SE Daily listeners are above average. And if you're looking for a job, Triplebyte is a great place to start your search, fast tracking you at hundreds of top tech companies. Triplebyte takes engineers seriously and does not waste their time. I recommend checking it out at tripledbyte.com/sedaily. That's triplebyte.com/sedaily.

Thank you, Triplebyte, for being a sponsor.

[INTERVIEW CONTINUED]

[0:21:36.1] JM: That makes complete sense, because when I'm using the internet, typically, if I'm going to a site that I — all the time in conversation over coffee, somebody will tell me, "Oh!

Have you checked out indiehackers.com or have you checked out wikipedia.com?" I'd be like, "Oh! No, I have not. I will check that out."

Similarly, if I was running my own Mastodon node and somebody told me in person, "Hey, have you followed Gargaron? He's this awesome guy. He's on mastodon.social. I might say, "Well, I've got my own Mastodon node and I don't follow people in other nodes, but if you're telling me that this person on mastodon.social is worth following, I'll go reach out to his federated node and I will follow him so that now I will be subscribed to the activity that takes place on that node that is created by Gargaron."

[0:22:39.9] ER: Exactly.

[0:22:41.6] JM: How resource intensive is it to run a Mastodon node and how many people do you have setting up their own nodes to run them?

[0:22:50.2] JM: Well, the bane of our existence is kind of Webpack actually. If you remove Webpack from the equation, Webpack is a tool that's used to compile JavaScript and CSS assets, and it's used only during setup or upgrades only one time, but it consumes so much memory that people complain constantly about how resource hungry it is. If you remove that from the equation, Mastodon does not require a lot of resources and it could run on a $5 node from DigitalOcean or your host of choice. Essentially you just need like one CPU and one or 2 gigabytes of RAM and you're good to go. Of course, if you want to host more and more people, you'll need more and more resources, can't get around that.

[0:23:34.9] JM: What's the scalability story there? If I started a — let's say I started a Mastodon node. I start Mastodon.jeff, and mastodon.jeff starts with just me and a couple people I know that are in my apartment building and we all tell all of our friends to come join mastodon.jeff. Everybody else joins it. We start to get a thousand people on my node and it's starting to get a little too resource intensive. How does the node scale, or can I scale a node or do somebody else —

[0:24:10.4] ER: Yes, absolutely.

[0:24:10.8] JM: Okay. You can scale it.

[0:24:11.7] ER: Mastodon is a Ruby on Rails application, and luckily a lot of big companies run Ruby on Rails application servers. A lot of documentation on how to scale that, how to run such applications. Essentially it scales horizontally. You can just keep adding app servers that run the code, that it's split in, basically three processes. There is a streaming API process, which is coded in node.JS. There is a sidekick process, which is background jobs and there is the web process that serves web requests.

So you can you can individually add more and more of those processes, and the one bottleneck of the whole thing is the database. If the database is getting close to being maxed out on usage, you have kind of no choice but to buy a more resourceful server for database. But beyond that, load-balancing between app servers, that's how it's done.

[0:25:12.4] JM: Database, by the way, is another thing that a centralized service like Twitter is going to take care of for me. I don't have to worry about backing up my tweets, because Twitter has many, many backups of my tweets. Whereas if I was running my own Mastodon node on my laptop, if my node fails, I'm guessing that I'm going to lose all of my Mastodon toots, that the terminology for a Mastodon message. Is that right or is there any sort of backup model where peers are sharing toots and there are some durability, some distributed durability built-in?

[0:25:58.7] ER: On one hand, there is. On the other hand, there isn't. It depends on what you want from it. So every server where you're followed from obviously stores a copy of your toot. So if you disappear overnight because you didn't keep your backups, the people in those servers are still going to be able to read the content that they received previously. If you, for example, get in touch with the admin of that server, you could, with some efforts copy over those toots back into your own database and restore it somewhat. I don't think it would be perfect. Well, at least that would take a lot of effort to do that. But on the other hand, if you're not keeping backups on your own node, if you lose it, you lose it.

[0:26:42.7] JM: So if somebody subscribed to me, their node is pulling off all of my toots or is it only the most recent 30 or 50?

[0:26:52.9] ER: Currently it's not pulling down anything at the moment of when you're followed. It just receives the new ones that are pushed. In the future, basically we didn't implement this exactly because it was not clear whether we should fetch down 50, 30, a thousand or all of them, because it's such a [inaudible 0:27:11.7] shedding each issue that don't really want to touch it, and it's not really a big deal, because most of the time the way this this thing works and the way it works organically is that most of the time when you find a new person, you want to follow — Somebody else has followed them from your server.

So you already have some backlog of content that you can see. In the case that that is not the case, when you find something completely new, well, this it's not such a big deal either. You just click on their profile to open the public view, and on that you can see everything.

[0:27:47.8] JM: That makes sense. I am starting to grasp the benefits of this federated model and the elegance of it. It sounds like something that works beautifully and at scale and with enough people running node setups that are built with some minor tweaks for scalability, like you said, load balancers and horizontal auto scaling set up. This seems like a very durable system.

The one thing I'm curious about is the fact that in the different decentralized systems that we've covered on the show, which are numerous. We've covered bitcoin and Ethereum and Scuttlbutt and IPFS, but in all of these systems there is a pressure to centralize for convenience at various points in the system.

As you've already mentioned, it sounds like you are pretty vigilant about not centralizing, because you don't have this central index even of node addresses, which is you have to be pretty vigilant to not include that kind of index as a centralized source. But, for example, if you wanted to scale Bitcoin transactions, you can just build a big off chain system that would occasionally reconcile with the main bitcoin chain. This is the whole lightning network idea, but the more transactions the you put in that off chain system, the less decentralized bitcoin actually become.

That's just one example of how centralization, is there's always these pressures towards becoming centralized off of these beautiful decentralized systems that are unfortunately not so scalable.

It seems like Mastodon maybe doesn't have that problem so much, because Mastodon is — The model is a little bit more scalable it seems. Are there any pressures that pull you towards centralizing parts of Mastodon?

[0:29:54.0] ER: Not me personally on a software level, but there is of course the social problem that people tend to join bigger servers. This is the case, for example, at the start of the podcasts I mention Gmail as an example for the centralization pressures of the email system, and people do flock to bigger servers on Mastodon as well, so mastodon.social, octadon.social, [inaudible 0:30:17.7] they tend to grow and they tend to have much more users than the other ones.

On the software level, no. I'm not currently thinking about any centralized features or anything and I'm pretty, as you said, vigilant about not putting that in. I think that the federated model is very scalable indeed. As you mentioned, as I mentioned as well, the bitcoin thing where there is one central piece of data essentially that is distributed between the nodes. It means when that central piece of data grows, when more people use bitcoin and more and more transactions get added to the blockchain, everybody has to download that, everybody has to keep a copy of that and it grows and grows and grows at a faster pace.

On Mastodon, each node grows separately and only according to the needs of the people actually using it based on those subscriptions that they perform. I think that's a little bit more future proof.

[0:31:12.8] JM: I mean, I could see this issue of the history becoming kind of a scalability issue that could lead to pressures on centralization, right? Because if you have a large log of people that are sending lots and lots of toots and they want those toots retained and durable and persistent and censorship resistant over time, Mastodon starts to look more and more like a blockchain, and you throw in multimedia things like video that are really high bandwidth. Could you foresee that becoming more of an issue?

[0:31:52.8] ER: I can't really say that. I mean, I had those worries when we’re just starting, and at this point it’s still fine. I don’t know. As I said, the nodes grow according to the needs of the users on those nodes and I can't think of anything more elegant and more scalable than that, because where else you’re going to go? No nodes stored, nothing. I just don't see anything else that could be even more scalable than that.

[0:32:23.3] JM: That’s a fair point, and this is kind of like — it's not exactly — Like I'm just thinking of it, comparing it to the bitcoin blockchain and maybe that's not even the best comparison I should making, but in bitcoin it really matters that everybody is playing against the same system of record, because I wouldn't want somebody on another bitcoin node to be able to create a transaction out of thin air that I don't support. Because it is a creation of money that should not have been created and it’s not backed by a proof of work.

On Mastodon, it’s not such a big deal if somebody creates a new toot or if somebody removes all of their toots from history and nobody else had a backup and they were willing to remove all their toots from history. Not exactly the same system.

[0:33:17.9] ER: Yes, indeed. The difference is Bitcoin is immutable — the blockchain is immutable. So everything that happens gets added on and on and on. Mastodon does not have the same thing. Nothing has to be immutable. You can remove stuff, and I think that's actually really important for a social network, because people are going to post stuff that you would want to remove and not being able to remove it is going to be a huge problem for any social network that once use the blockchain as the basis.

Also, back to storage requirements, just the fact that you can clear out the cache essentially of content from other nodes just to minimize your requirements so you can indeed only store the things that your own users post. That is also possible.

[0:34:07.6] JM: On some social networks, there is this blue checkmark that allows people to identify as being famous, or it's at least this authentication mechanism, this proof of authority. Does Mastodon have any kind of blue checkmark?

[0:34:25.5] ER: No. People could use the green checkmark emoji and put that into their name, but no. It doesn’t really serve the same purpose. There is a complexity in implementing something like that in the decentralized system verification, because that would essentially have to be a central authority of some sort that you show them your passport and they sort of confirm that you are the person you say that you are.

On one hand, I can see that being valuable, because impersonation is an issue. On the other hand, email doesn’t have verification either, and people have managed so far. So it's a bit of a paradigm shift back from networks like Twitter where you do have that checkmark to rely on, but people just have to treat usernames like email addresses, and that if it's from a different server, that's not the same address.

Yeah, and I think there are other issues with the checkmarks too, like the feeling of importance that comes with it, the feeling of authority and endorsements, which is not necessarily warranted.

[0:35:33.7] JM: The ability of the platform to remove the blue checkmark and it's almost like a scarlet letter has been placed on you and once you get de-authenticated by a platform. It’s just a strange power for the platform to have.

[0:35:51.4] ER: It is strange only because there was confusion about what it really means, because, I mean, if people perceive the blue checkmark as an endorsement by Twitter, then they can take it away. But if it's supposed to mean that the person is who they say they are, then in a way makes no sense.

At the same time, Twitter really mess this up, because they give extra tools and extra features to people who have the checkmark. So it is not just you saying who you think you are. It also benefits. So it's a mess. It's a mess. I don’t want that on Mastodon.

[0:36:29.7] JM: Have you looked at Keybase at all?

[0:36:31.7] ER: Yes, I have. I think that's a good alternative or a temporary solution if there ever is going to be a permanent one for verifying, and many people use it that way. You essentially just have to sign some texts and post it as a toot and that's your verification. You can really claim that your profile is yours. It'd be nice if Keybase actually implemented a real integration with Mastodon that did not require people to manually post anything, but it works.

[0:36:58.1] JM: I think they would love to do that. They've just got out along the road map, and I like what Keybase is doing. I think it's something that's pretty innovative and I think it's needed. So if a government wanted to censor Mastodon, could they do that? What would they do? How much censorship latitude do they have?

[0:37:20.2] ER: Essentially, they would have to — I mean, it depends. Do they want to censor all of Mastodon or a single node? I think they wouldn't have a problem censoring a single node, because you just lock the domain address of that particular node or maybe you — I don't know. If you control the hosts where it's hosted, you can just take it down, but if you want to block the entire Mastodon network, I don't think that would be possible, because there — the network is large and it is sparse, and that means you would need to call it to find all the instances and you would still not find all of them because there could be just completely separated, disconnected bubbles that are like private Mastodon instances or like completely separate.

[SPONSOR MESSAGE]

[0:38:15.1] JM: You're a successful developer and you couldn't have gotten to where you are without help in your education and career. Maybe you're thinking about ways to give back in the community where you live. The TEALS Program is looking for engineers from across the country to volunteer to teach computer science in high schools. Work with a computer science teacher in the classroom to bring development concepts to life through teamwork and determination. Pay your success forward by volunteering with high school students in your area by encouraging them on the computer science path. You can make a difference. If you'd like to learn more about the Microsoft TEALS Program or submit your volunteer application, go to tealsk12.org/sedaily. That's tealsk12.org/sedaily.

Thank you to the TEALS Program for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:39:20.8] JM: Let’s talk about usage. So people use different social networks in different ways. I use Twitter differently than I use Facebook or Quora. How do people use Mastodon?

[0:39:35.2] ER: Well, as I said, I was a power user of Twitter. So when I made Mastodon, it sort of filled that niche for me. So it's very close in use to Twitter. There are minor differences there, because the system for posting is a little bit more rich. You can set the privacy granularly rather than setting your entire account to locked, you could just post the individual message to your followers or public, and there are content warnings, for example, for spoilers and there are ways to hide images individually rather than marking your entire account as not safe for work.

These small differences and many other small design decisions sort of influenced the way people use Mastodon, subtly. I mean, it's a large network, and I don't follow anybody. I don’t follow everyone, but from the people that I know from that social bubble that surrounds me, I know that people talk less about news, and when they do they use content warnings to warn that it's going to be about news, because people are really tired of that and don't want to interact with that sort of content, normally.

[0:40:49.2] JM: Really? So news is not the focus, because news has become like the de facto focus of Twitter and Facebook. It's almost like — I guess Facebook has been trying to push things towards back to personal content, whatever that means, but I don’t even think anybody — not many people are using Facebook for updates on new stuff, but certainly Twitter. People are using Twitter to see the news all the time. So what would — I don't even know what I would be tweeting about if it wasn't centered around the news.

[0:41:22.7] ER: Yeah. That’s weird, isn’t it? Yeah, on Mastodon, people don't talk about news as much, certainly not political news. People do sometimes post tech articles and stuff. Again, that's within my bubble. My bubble is a bit more technical and I can’t guarantee that that's the same for all of Mastodon, which is like more than the million users nowadays.

Yeah, people post less about news and it's a lot more personal. People do post just personal updates, what they're doing and there is a less celebrity culture, like on Twitter you are from the start encouraged to follow famous people and there are —

[0:42:01.1] JM: And you're encouraged to become a celebrity or to act in a way that would manifest yourself into some sort of strange celebrity even if it's just within your own social network.

[0:42:11.9] ER: Indeed, and the checkmark thing that gives you extra features also contributes to that. So there are just like two classes of people on Twitter, the famous ones that have the extra features and the normal ones that have to follow the famous ones.

On Mastodon, there is no celebrity culture. Everybody is sort of the same in terms of — everybody gets the same features essentially, because there is no checkmark.

People — I don't know, people don't appreciate celebrities just for being celebrities I guess. People prefer to just find people who post interesting content and people they like to talk to. There was lot of talking to each other on Mastodon. A lot more than on Twitter. I've been told multiple times that people say, "I get a lot more engagements on Mastodon than I ever did on Twitter." People just talk to each other.

[0:43:00.1] JM: Social networks often have a spam problem. Does Mastodon have any kind of spam problems and how do they deal with it?

[0:43:07.8] ER: There we go back to the federated model. So each server is operated independently, can implement its own rules. Side effects of that or maybe not the side effect, but the cause of that is obviously that it's operated by different people. Those people are the administrators, moderators. So each server has their own team of admins and mods ready to responds to the needs of the users on their particular instance.

What this means in practice is that, for example, in a company like Twitter, you get these billions of users and a team of customer support sitting somewhere, I don't know, 20, 30, 100 people. Mastodon, you have may be a thousand or like 10,000 people for like five mods or something like that. That ratio is a lot higher. So it's a lot easier for those local mods to enforce policies, to keep spam out, to keep harassment out, because there's just — It's just a lot more manageable.

[0:44:09.0] JM: The perverse thing about spam on Twitter that does not get talked about a lot is Twitter has some incentive to leave spammers and bots on their platform, because those spammers and bots consume ads.

[0:44:30.1] ER: Absolutely. On multiple levels, actually, they consume ads and they prop up their stats, and Twitter is a publicly traded company with shareholders, so they needed their quarterly reports to look good. The more activity, the better it is for them.

On Mastodon, you don't have that incentive. If somebody is spamming, yeah, your stats look higher, but they're also spamming your server resources and they're also like — they're spoiling the community essentially. They're making it less friendly, less interesting to the real people there. There is no incentive to keep spam around. We just ban that.

[0:45:10.2] JM: That's great. What about mobile clients? Do you have mobile clients for Mastodon?

[0:45:14.3] ER: Absolutely. I really didn't like when Twitter closed down their API for third-party clients and made it practically impossible to have nonofficial clients. So when I made Mastodon, I made it my goal to make one API that is the same for all apps. Including the ones that I make. So the web UI you see when you log on to Mastodon is using the same API that any other app could use.

So there is a rich ecosystem of various third-party apps. There are no official apps, because there is no such thing as an official app in this context. For every platform, essentially, if you search for Mastodon on Google Play Store, on the App Store, anywhere, you will find a multitude of applications that allow you to browse Mastodon.

[0:46:01.8] JM: So this is Software Engineering Daily. We should talk some about the software stack of Mastodon. We've covered the contours of it so far. It's a Ruby on Rails application. I believe that the frontend is and ReactJS, and beyond that, I don't have a good understanding except for the server architecture. Maybe you could talk about what are the more interesting aspects of Mastodon. What are the areas of technical debt that you're most focused on? What are the unique technical challenges of Mastodon?

[0:46:36.2] ER: I think the most unique technical challenge is just the fact that when you are releasing features, you have to make sure that they work on a variety of different setups and that they work with each other when you cannot guarantee that the same version is rolled out everywhere at the same time, because every admin is going to take a different amount of time to upgrade eventually.

This is especially important when you want to push outs protocol changes for the federation itself. So far we've been managing fine. It just involves introducing features gradually and sometimes introducing the support for the receiving end before you release the thing that actually allows end-users to use that feature. So when that version comes out, everybody has upgraded already, so it works. I think that's the most unique challenge probably, because other apps don't have to deal with that.

[0:47:32.0] JM: I suppose you've got all these people running their own nodes. You have to have some sort of enforcement for them updating their nodes or you have to make the protocol remain back compatible with the old node software.

[0:47:48.0] ER: Yeah. It's always back compatible. There was actually a huge feat that we accomplished and really proud last year, sometime around August or something like that, when we switched from one federation protocol to the other in the entire network. Mastodon started out as an OStatus server. OStatus was the original federation protocol developed in 2012, but it's a little bit outdated. It is lacking a lot of features that people were requesting. It uses HTML. It's really outdated, and ironically it's never left the draft stage as a standard. So it's never become an actual standard.

So at some point we decided that we want to switch to something more future proof and more up to date. And so the activity pub standard was proposed. It was on draft stage when we started implementing it, because we would have been the first software to implement it and we helped shape how that protocol is designed. Throughout the implementation, we were giving feedback to the team that was designing the protocol at the W3C. Eventually it became an actual recommended standard by the W3C.

Yes. So the feat was that we switched the protocol for the entire network and nothing broke. I think that's pretty cool. We're still somewhat compatible with those status, but Mastodon servers talk to each other over activity pub nowadays.

[0:49:25.9] JM: So I can tell that you're a thoughtful guy and you've put a ton of work into this project and it's very much a labor of love. This is not it is an enterprise with a billion-dollar outcome anywhere in the foreseeable future.

So I kind of want to get a little bit softer in terms of these questions, because I can tell you've got a broad thesis on social networks, humanity, socialization, the internet. The world, our world today, we are still so segregated into different geographic populations. We have different cultures. It doesn't feel like the internet has blended us and globalized us that much yet. You look at Twitter, you can feel that there is something there that humans want to connect with each other and we want to break down these artificial barriers to some extent, but there are still these frictions. There are colloquialisms and differences of culture that separate us in ways that sometimes seem impenetrable.

What do you think that we're going to see in the next 10 or 20 years? How is the internet going to affect us as an international community?

[0:50:48.7] ER: I do not think that I am qualified to answer that question at all. I'm sorry. I cannot predict what social developments there will be in society.

[0:50:59.9] JM: Are you are you optimistic? Given your time in working on Mastodon, it seems like you are, or maybe your just not — Do you not have some kind of deterministic outcome in mind? Are you just building tools that you think will be useful to people? Or do you have a mission? Do you have a vision for what you would like to see?

[0:51:18.9] ER: I do have a vision, but it is kind of on a different level than what you're asking me about right now. I can't really predict - I can't really say how humanity is going to change and how they're going to treat each other in terms of separate groups.

What's my concerns are, are a little bit more practical and down-to-earth. There is a future where basically the internet becomes one single website like facebook.com and there is no escape from ads. There is no escape from tracking. They control everything. They make money off every movement that you do.

That's not a future that I want. I want a future where the web is free and open, where everything is operated independently and still talks to each other. I want people in Germany to have servers that operate in Germany and I want people in France that have servers that operate in France and are not dependent on a company in the United States that enforces United States laws across the entire globe. That's kind of the things that I'm thinking about.

[0:52:32.0] JM: Do you see any sort of intersection between the Mastodon technology and the suite of technologies coming out of the Ethereum Project or IPFS or Gollum or any of these other decentralized, computation storage platforms?

[0:52:50.7] ER: These questions are getting more common. I've heard questions like are you going to integrate a blockchain into Mastodon, like a lot in the past half year I think. I don't really think that there is a need for a blockchain in Mastodon. At the same time, I'm curious what other projects are going to bring to the table. I've been looking at the Dat protocol, because I remember it has a social network project running on top of it.

[0:53:13.8] JM: What is that? The Dat protocol?

[0:53:20.1] ER: Yes, I think Scuttlebutt may actually be running on that if I remember correctly. There are ones based on IPFS and there are people who propose that media storage of Mastodon should be based on IPFS. I don't think it’s quite time to talk about that yet, because it's a little bit untested technology and the integration is a little bit untested as well.

It certainly cannot get more storage magically out of using IPFS because the way it operates, there are still nodes that pin the content and that's basically your server and the others simply cache it. So it's kind of the same as what we’re doing right now except less obvious like that.

Yeah, I'm curious. I certainly realize that projects that use peer-to-peer technologies or the blockchain have their limitations that I think they have not solved yet. For example, the problem of moderation in a peer-to-peer network. In a peer-to-peer network, you essentially are on your own and you're going to have to encounter all the spam, all the bad content, all the harassment on your own.

On Mastodon, you have your admins that would solve that for you. You have your community of your neighbors on the same node who would report spam and bad content and there is this communal way of solving it that you wouldn't have in the peer-to-peer network. With blockchain, you have the issue that you cannot — sorry. I'll let you talk in like a second.

[0:54:48.6] JM: No. Go ahead.

[0:54:49.8] ER: In blockchain, you have to problem that you cannot delete content. For example, I heard recently, there was new story in The Guardian that child abuse images were encoded in the blockchain which essentially makes — in the bitcoin blockchain which essentially makes downloading the blockchain illegal in most countries.

Yeah, I think for now I believe in Mastodon, in my own project, and I think it's the best solution right now, but it's not the end of everything. One nice [inaudible 0:55:17.0] about Mastodon and using activity pub, the standard protocol, is that anybody who can write server software that interacts with Mastodon becomes part of that network.

There are in fact servers that are not Mastodon that are part of the social network. There is one called the Pleroma, there is one called [inaudible 0:55:36.3]. There are various different ones being worked on right now, such as Rustadon, which is a play on the programming language Rust, which it's written in Mastodon.

Yeah. So eventually, I'm sure the software can come along that is much better than Mastodon and it will pick up right where we left off. It will integrate with Mastodon and nobody will have to switch again and bring all their friends over. That's the point I'm making.

[0:56:05.2] JM: Agreed with all of that. One direct application I could potentially see would be a censorship resistant node. If you want to set up a decentralized censorship resistant Mastodon node, you could probably do that on some was some suite of blockchain technologies, and then you would be able to evade the government censorship problem we talked about earlier.

[0:56:27.8] ER: Yes, probably.

[0:56:29.2] JM: Yeah, maybe. Vaporware. Okay, vaporware at this point. That's why I like talking to you. It's not vaporware. I just spent a month talking to blockchain people. So I guess I am acclimated to vaporware. Tell me more about vaporware. Tell me about your ICO. You're not having an ICO, by the way, for those who are wondering.

Just to conclude, and this has been a great interview, but you're working on this full time?

[0:56:54.0] ER: Yes, I am.

[0:56:55.2] JM: Okay. What's that experience been like?

[0:56:57.4] ER: On one hand, it's paradise, because I work at my own pace and I can work when I have inspiration. The process is a lot like art to me. On the other hand, of course, I feel a lot of pressure to work. So as to not disappoint the people who pledge in my crowd funding. So mostly I wouldn't trade it for anything. It's really nice.

[0:57:22.5] JM: I mean, I kind of feel the same way about this podcast, because I get to do it when I want to. But on the other hand, one thing I don't like is the isolation. Do you have any issues with just like sitting at home and the irony of reaching millions of people while simultaneously working in an isolated fashion?

[0:57:39.2] ER: I'm a bit of an introvert by nature, so it's not much of a problem for me. Although I will admit, I don't go out that much. There's another level of isolation that is different to that. It's that not that many people lead the same lifestyle. Not the same people work for themselves, like self-employed on old project. Not many people can afford, have the luxury of being crowd funded to be able to work full-time on something that they love.

So a lot of my friends just don't have the same problems and not the same experiences. So there is that sort of a little bit of isolation there.

[0:58:17.9] JM: I hear you. Okay, last question. Have you talked to anybody at Twitter or Facebook or the other centralized social networks? Have you gotten any feedback from those places on what they think about Mastodon?

[0:58:31.3] ER: I think they like to pretend that we don't exist. Did you mean users or did you mean like somebody involved in Twitter and Facebook?

[0:58:39.8] JM: Somebody involved, like executives. I'm imagining —

[0:58:42.0] ER: No. Absolutely not. They do pretend like we don't exist.

[0:58:46.5] JM: Weird.

[0:58:47.6] ER: I think they're hoping that we'll go away if they just close their eyes.

[0:58:51.2] JM: All right. Well, let's hope that doesn't happen. Mastodon is an awesome project and I'm really impressed with it. I am impressed by your ambition and your humility as well. So Eugen, thanks for coming on Software Engineering Daily. It's been fantastic talking to you.

[0:59:06.1] ER: Thanks for having me. Thank you. It was a nice interview.

[END OF INTERVIEW]

[0:59:12.3] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use, and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plug-ins. Use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on-the-fly. GoCD agents use Kubernetes to scale as needed.

Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on GoCD with the new Kubernetes integrations. You can check it out for yourself at gocd.org/ sedaily.

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

[END]