EPISODE 776

[INTRODUCTION]

[00:00:00] JM: Edge computing refers to computation involving drones, connected cars, smart factories or IoT sensors. Any software deployment that is not a large centralized server installation could qualify as an edge device, even a smartphone. Today, much of our heavy computation takes places in the cloud, which is a set of remote data centers some distance away from our client devices. For many use cases, this works just fine, but there are a number of cases where lower latency and higher bandwidth requirements are occurring at the edge. A simple example is video.

Let's say you want to record a video stream and you want to detect people in that video stream in real-time. Based on who you detect in that video stream, maybe you want to do different things. Maybe you want to identify that person and send them a text message to let them know that they're being recorded on video, or maybe you want to report to the police that a dangerous person has entered the premises. This video stream could be captured by a drone, or by a smart car, or by a video camera mounted somewhere.

So where is the video stream getting stored? Where is the machine learning model running? How do you deploy new machine learning models to the operating system with the machine learning model? How could you cross-reference this video stream with other video streams?

This is a really simple example, but there are so many open questions as to how to solve such a problem, and since we're at the edge, we have increased constraints on the resources that we have accessible to us. So there's a need for new hardware and new software to power these edge applications. This led to the creation of LF Edge, which is a new open source group under the Linux Foundation. The goal of Linux Foundation's LF Edge group is to build an open source framework for the edge.

Arpit Joshipura is the general manager of networking orchestration, edge computing and IoT with the Linux Foundation. He joins the show to describe the state of edge computation and the mission of LF Edge. This episode was exciting for several reasons. After seeing the rise of

Kubernetes for a container orchestration, we know that a popular open source technology which solves a widespread problem can have dramatic influence on the software world, and when multiple large companies get involved in that open source project, it can gain traction quite quickly.

Edge computing has a large set of unanswered questions, but telecom providers, like AT&T and large infrastructure companies like Dell EMC are getting heavily involved with the Linux Foundation edge group. This represents a significant expansion of the open source model, and it also suggests that further investment into open source projects will occur in the near future. This was an interesting episode and it connects nicely with tomorrow's episode about AWS's internet of things platform.

Hope you enjoy today's episode.

[SPONSOR MESSAGE]

[00:03:22] JM: Triplebyte fast-tracks your path to a great new career. Take the Triplebyte quiz and interview and then skip straight to final interview opportunities with over 450 top tech companies, such as Dropbox, Asana and Reddit. After you're in the Triplebyte system, you stay there, saving you tons of time and energy.

We ran an experiment earlier this year and Software Engineering Daily listeners who have taken the test are three times more likely to be in their top bracket of quiz scores. So take the quiz yourself anytime even just for fun at triplebyte.com/sedaily. It's free for engineers, and as you make it through the process, Triplebyte will even cover the cost of your flights and hotels for final interviews at the hiring companies. That's pretty sweet.

Triplebyte helps engineers identify high-growth opportunities, get a foot in the door and negotiate multiple offers. I recommend checking out triplebyte.com/sedaily, because going through the hiring process is really painful and really time-consuming. So Triplebyte saves you a lot of time. I'm a big fan of what they're doing over there and they're also doing a lot of research. You can check out the Triplebyte blog. You can check out some of the episodes we've done with Triplebyte founders. It's just a fascinating company and I think they're doing something that's

really useful to engineers. So check out Triplebyte. That's T-R-I-P-L-E-B-Y-T-E.com/sedaily. Triplebyte. Byte as in 8 bits.

Thanks to Triplebyte, and check it out.

[INTERVIEW]

[00:05:11] JM: Arpit, welcome to the show. I want to talk to you about edge computing. Edge computing has existed for more than a decade. We've had smartphones. We've had factories with computers installed. You could classify these computers as edge computers. But something has changed in the last few years that has brought more attention to edge computing and IoT. What are the things that have changed in the last few years?

[00:05:41] AM: Thank you very much, and excited to be here. Let me hit on five technologies that are converging that has made edge computing great again or cool again I should say. They are in no particular order, but the first one being the maturing or at least coming of a technology, telecom technology called 5G. That brings a set of very low-latency support closer to the applications as possible, and we're talking 5 to 20 milliseconds of latency here. So really, really fast, and these latencies are needed for newer apps like autonomous cars, or factories that need quick turnaround on analytics, etc. So that's the first technology.

The second technology that has matured is the microservices Kubernetes container type app development framework for engineers, where you can quickly write the apps, make them portable, they identify containers that could store it and more it close to the actual asset, whether it's an IoT device, a microcontroller or a gateway.

The third one is really the maturing of AI and machine learning algorithms, frameworks and basically tools that allow predictive maintenance to happen very close to the device and the application. So that's number three.

The fourth is obviously the maturity of very custom hardware. So it's CPU, GPU, TPU, NPU, you name it, in all shapes, forms and sizes and cost that can actually cut across the technology

spectrum. Then the fifth missing element that has matured in the last two years is what I call on-demand NFE, which is the automation of a telecom network.

Today, everything on the edge was really around silos. There's an enterprise offering of an edge or a cloud offering of an edge, and now telecom has come in and kind of solved this edge cloud problem where you're bringing compute and storage very close to the edge application. So these five technologies as I call, convergence of these five technologies, is really, really driving a whole new set of agile applications and IoT devices that have really renewed the interest of pretty much all verticals.

[00:08:11] JM: You're discussing these trends in edge computing. What are the applications? What are the use cases? What are the concrete examples that we have seen of edge computing contemporarily?

[00:08:25] AM: So typically I always classify applications into sort of the latency sensitive and non-latency sensitive. So for example, if there is a sensor or a device or sort of a robot sitting in a factory that is dumping analytics data once a month into a public cloud hundreds of kilometers away, while it may be an IoT asset, it's not an edge application.

What we are saying is the applications that are in the 5 to 20 millisecond latency zone in terms of response time, and therefore obviously they are falling into four broad categories that we are tracking. Obviously, IoT and analytics, whether it's industrial sensors, home devices, retail and commerce or healthcare. Whether it's immersive experiences. So virtual reality, augmented reality, 360 video is very hot, variable assistance. Whether it's autonomous devices, so things like drones, vehicles, robots, etc., or straight telecom edge infrastructure applications, like virtual RAM, passive optical networks, or SD-WAN, or enterprise services. There's a whole new set of applications that emerge because compute and storage is brought close to you. So these are some examples.

[00:09:53] JM: My favorite example application that I saw when I was preparing for this episode is video. It's very simple – Video. Video is so high-bandwidth. It's such a simple one word application, but you think about it, like we need video on the oil rig. We have our smartphones that need video. The connected cars need video. The drones need video. What is video but this

gigantic, high-bandwidth data stream that is training data for machine learning. It's potential things that can be run through machine learning classifiers. There's compression that needs to be run on it. There's all these preprocessing that we can run between the edge and the cloud. I really feel like video is just the perfect prototypical application category that we can use to say, "This alone describes why we need new infrastructure at the edge."

[00:10:59] AM: So you're very right. I would augment it with the following, which is we had a couple of analysts do survey and research on what are the killer apps for edge, and yes, while we have a standard craft that says video content delivery is the top of the pyramid, and you're absolutely right. There are things like autonomous vehicles, ARVR, IIoT gaming, surveillance, supply chain, smart cities, etc. They're all on the list. But if I was to sort of summarize the killer apps, they fall in two buckets. It's nontraditional video. So it's not like the YouTube's of the world per se, but anything nontraditional. So video coming from a 360, video coming from a drone, etc. It's connected things that move, but not a phone. So things like autonomous vehicles, or drones with inspecting base stations and electrical grids, factories that need to connect in a very close and do predictive maintenance. So the two big buckets, nontraditional video and connected things that move.

[00:12:08] JM: What are the challenges of these domains on a technical level? What kinds of new infrastructure do we actually need to build and why are the present day operating systems and devices and application tooling, why are they insufficient?

[00:12:28] AM: All right. So to answer that question, I have to start off by sort of going back to defining the edge. So one of the things that LF Edge as an umbrella project under the Linux Foundation is striving to do is create a glossary of terms and define the scope of what an edge is. It's anything from a software perspective that starts above the hardware layer irrespective of the connectivity. So [inaudible 00:12:57], Bluetooth, Wi-Fi, it doesn't matter. Then stops at the edge data center, which is in a telecom kind of a network.

Anything that goes in a regional or a centralized data center is not considered as edge. So assuming that we define the scope correct of what an edge is, because that's where you are going to get that responsiveness of 5 to 20 milliseconds of latency, the three big problems that the community is solving right now is the following. So one of them is a non-technical problem

that we have hopefully a path to solve it, and that non-technical problem is the edge community and the development in an open source way is very fragmented, right?

Lots of people are building lots of little things for lots of different verticals that don't work with each other. Hopefully that problem is solved by taking LF Edge as an umbrella organization where you can have these communities across cloud, across telecom and across enterprise and IoT all collaborate and build out common plumbing and common APIs to make it happen. So that's the first non-technical problem.

The second technical issue is the lifecycle management of an app or an end device, whether it's an IoT device or an application, software application that is classified as an edge application. The lifecycle management means how do I install it, load it, boot it, restart it, log it? How do I store persistent data on it? How do I write down APIs that can connect to clouds and multi-cloud? How do I get to making sure the size of the load is correct so that it matches the hardware, but is hardware agnostic?

So it's essentially the lifecycle management or what we call the plumbing of edge infrastructure in a zero-touch manner. As you know, clouds through Kubernetes, for example, orchestration is zero-touch. Telecoms, they are pretty much getting to zero-touch through projects like open networking automation platform or ONAP as we call it, right?

Enterprises are getting to zero-touch through software define data centers, but they're not quite yet, because apps are kind of all over the place, right? Then the IoT landscape is not zero-touch, because you are either sitting in a public cloud ecosystem, like an Amazon, Azure or Baidu, or you're sitting in a private cloud, or you're sitting in some version of a hybrid cloud. So that's the second big problem that we are solving, which is the plumbing or lifecycle management, and that's a technical problem in nature. How do you get zero-touch lifecycle for edge applications and devices?

Then the third one is for future apps that I believe are hardware independent, OS independent, cloud independent. What are the set of APIs that can be exposed and standardized both on the southbound side of the drivers and the physical interfaces as well as the northbound side, which

is really around the cloud and the multi-cloud infrastructure, right? Those three technical – Like two technical, one non-technical, is what LF Edge is intending to solve.

[00:16:21] JM: In previous episodes we've done a lot of coverage over the dynamic between the "cloud and the edge", and this is a really interesting dynamic, like what kinds of processing do you do on the cloud? What do you want to do on the edge? What are you going to do about the fact that you want to have this high-bandwidth connection between these two places, but maybe there's intermittent bandwidth at the edge, these kinds of things.

But what I understood by reading about LF Edge, the Linux Foundation's edge computing set of projects, which we'll get into in some detail, but the importance of the telecoms and the fact that all of these processing or all of these data shuttling is going across telecom infrastructure. Can you explain the role of the telecoms when it comes to the dynamic of edge computing?

[00:17:17] AM: That's a brilliant question, because the moment we classify an edge application requiring storage and compute close to it with a certain parameter of latency responsiveness, you do get into the physical barriers of how bits travel, right? So at this point what telecom edge or edge cloud or smart edge or whatever, those are terminologies, but really what we're talking about is a set of automated compute storage and software infrastructure that is residing either at the bottom of the base stations or on a gateway in the basement of a building or a stadium or a little bit further out in a smart central office in a neighborhood.

What telecoms bring to the table are three things. They bring the location advantage. They bring the latency advantage both because of location, but also because of technologies like 5G, and they bring the mobility advantage. What that means is if there's an application, connected cars and autonomous driving and fleet transportation being the perfect example, if the application requires low-latency response with mobility, you do not – An enterprise or a cloud provider will not be able to solve that. Those are the three things why telecom is such a critical player in the edge compute space, and that's why we see a lot of collaboration between telecom players, cloud players and enterprise IoT players, and that's the mission of LF Edge.

[00:18:53] JM: And who – Not to sound flippant. Who cares about this? Like if I'm a developer, do I care? Aren't I just either I'm a developer like building something for my factory? My

company owns factories and I'm writing software for the factories, or like maybe I'm writing software that is doing processing on data that has made its way from the factory into the cloud, but in either way, I work for a candy factory company. Why do I care about telecom infrastructure, or are these open source projects, which we will get into, are these open source projects, are they only meant for the telecoms?

[00:19:39] AM: No. No, they are not meant for telecoms. That's the whole point. The projects are bringing the telecoms, the cloud and the enterprise verticals that their developers and their use cases altogether. So the three have to operate in harmony. The reason why a developer, an enterprise, a cloud player or a telecom player should care is what you described as an analytics sitting in a factory going into a cloud. That's IoT today. That's not an edge. That's not an edge app. That's not something that you'd traditionally categorize in the new world as – It's a vertical siloed of you doing analytics in the central cloud. You're not getting responsiveness on services, and you are solving the problem locally if the compute is at the gateway level. So that means your decisions are local. Your analytics are local. You're operating with limited set of information that you're not aware whether another factory in a different neighborhood is down or they have similar problems, or if it's a global fortune 500 with factories all over the place, you're not sharing anything.

So what is happening is what you described was what we call IIoT 1.0, or today's way of thinking. What we have to get to is there is a layer of common APIs and common visibility and control that allows these assets to share information. If you're connecting, say, take an example of smart cities. You're connecting the visual identity of license plates through the traffic lights and then passing them on to an edge application to respond in real-time. That's intersection of an enterprise, a government and a telecom cloud, because you're going to go through several networks.

Right now, what's in it for a telecom? Well, they provide a service through an SLA and they can charge for it. What's in it for the enterprise? Well, they are actually making money on this service, right? What's in it for a developer? Well, their service gets more and more adapted and it scales globally. So this is why I believe that the interest on such an initiative is very high, and you can always solve the problems in silos. What we have seen over the last several years with

the advent of open source is when communities come together and solve the problem, everybody wins.

[SPONSOR MESSAGE]

[00:22:14] JM: DigitalOcean is a reliable, easy to use cloud provider. I've used DigitalOcean for years whenever I want to get an application off the ground quickly, and I've always loved the focus on user experience, the great documentation and the simple user interface. More and more people are finding out about DigitalOcean and realizing that DigitalOcean is perfect for their application workloads.

This year, DigitalOcean is making that even easier with new node types. A $15 flexible droplet that can mix and match different configurations of CPU and RAM to get the perfect amount of resources for your application. There are also CPU optimized droplets, perfect for highly active frontend servers or CICD workloads, and running on the cloud can get expensive, which is why DigitalOcean makes it easy to choose the right size instance. The prices on standard instances have gone down too. You can check out all their new deals by going to do.co/sedaily, and as a bonus to our listeners, you will get $100 in credit to use over 60 days. That's a lot of money to experiment with. You can make a hundred dollars go pretty far on DigitalOcean. You can use the credit for hosting, or infrastructure, and that includes load balancers, object storage. DigitalOcean Spaces is a great new product that provides object storage, of course, computation.

Get your free $100 credit at do.co/sedaily, and thanks to DigitalOcean for being a sponsor. The cofounder of DigitalOcean, Moisey Uretsky, was one of the first people I interviewed, and his interview was really inspirational for me. So I've always thought of DigitalOcean as a pretty inspirational company. So thank you, DigitalOcean.

[INTERVIEW CONTINUED]

[00:24:21] JM: I am reminded of how I felt when I started covering the container orchestration space, and this was like four years ago or three or four years ago before Kubernetes had kind of become the de facto container orchestrator that everybody was using. I didn't even realize there

was a market for a container orchestrator thing, like one container orchestrator to rule them all. Three and a half years later it looks obvious. Everybody is like, "Of course I want this thing."

[00:24:53] AM: You just answered your question.

[00:24:56] JM: Well, let's go deeper there. So what's the analogy? How does this change? Looking back now I can see, "Oh, wow! Kubernetes is really changing the equation for these enterprises and giving them systemized cloud infrastructure, distributed systems getting solved." What gets solved when we take the edge computing world and put an open source spin on it?

[00:25:24] AM: Okay. So I have to go a little bit deeper on this. So what tends to happen, and just like you compared the cloud and the Kubernetes, and I think I mentioned this. What was a problem that Kubernetes solved in the cloud? It solved the container orchestration and the portability. Effectively, what that means is my apps don't have to worry about what is underlying. Same problem has now been solved in the telecom world through automation with projects like ONAP, right?

So now how do I take the best of those two worlds and put it close to applications that rely on this infrastructure being like literally next to me. When I do that, how can I automate it? So there's a layer of zero-touch, zero-automation. So the projects that we have are using – And specifically this is the Akraino Edge Stack. They are using portions of Kubernetes [inaudible 00:26:23] and ONAP to create an automated zero-touch load that would go and fit into an edge base station data center if you way that can actually serve these edge applications.

Effectively, what is happening is we're taking the telecom and the public cloud technologies and integrating them with automation and build both orchestration as well as automation and pushing it closer to the edge physically. So think of this as an edge cloud. It's really the culmination of what you were looking at three years ago where telecom was looking at two years ago and bringing them together and pushing it out for future expansions at the edge.

[00:27:10] JM: How does this compare to the proprietary infrastructure that exists today? So just to give you my perspective on – I've done shows on AWS IoT, Google's IoT stack, Azure's IoT stack, and they've got a bunch of different IoT offerings and they're pretty sweet. One

common pattern I see is this idea of a hub. So like in a factor you can have this IoT hub thing, which is like a server that's a small device in the factory and all of the devices in the factor talk to the hub and then the hub is the only thing that talks to the cloud. So you have this gateway that is the only infrastructure that's actually talking to the open internet. So it puts is a single focused point where you can have all the security centralized around that hub, and then talks to the cloud. Anyway, that's kind of my perspective for the interaction between the cloud providers and the IoT installations. Can you contrast that vision or tell me if that concords with your vision for the cloud providers and how that would differ from an open source world?

[00:28:22] AM: Correct. So what you're describing is probably 40% of the current IoT deployments, and I would say I would not classify them as edge, okay? Here's why. These applications do not have a latency requirement. They are IoT applications, for sure. They do go on a microcontroller with a gateway into an Azure or an AWS cloud where it's a single purpose, single vertical hub-based. I've participated in all of their conferences. You see an ecosystem of about 100, 150 member companies if you way from all verticals that would solve that specific problem.

The other 60% that would benefit from a lower latency from a proximity to clouds, to portability and things like that, and more importantly let's take the fortune 500. You have factories in U.S., you have factories in China, you have factories in Europe. You cannot have visibility across them. You cannot share datas. The APIs are different. It's a vertical silo. Wouldn't it be nice if you have standardized that with a layer of open source software that can plug into a multi-cloud environment? That's kind of one big use case beyond the other 60% of the apps that are low latency.

I just want to sort of emphasize, IIoT is not the same as edge. People confuse the two. There's overlap. Don't get me wrong.

[00:29:55] JM: And the difference is latency?

[00:29:57] AM: It is latency and it is the visibility and connectivity that you may need from the ecosystem around you. So if I'm operating in complete isolation in a factory floor with IIoT support, that's fine. That's one of the applications, but what if I have to send that data and

crunch it in real-time? It's not like it's getting crunched in an Amazon data centers hundreds of miles away. Crunching in real-time, because there are some issues.

Take home edge for example, right? Security issues popup, physical security, or disaster issues come up or something, and you got to communicate back and forth and process it in real-time, or you have on-prem factory there. The different kinds of hardware virtualized differently. Communicate on the backend with an edge device very close to their enterprise. It could be in a basement or something like that.

So you're right, it's the proximity, it's the latency, it's the responsiveness and it's the portability of the app across cloud. Just like Kubernetes and containers allowed app portability between sites and between different infrastructure, this is taking it to the next level.

[00:31:19] JM: Okay. Can you talk more about how – So when we think about the container orchestrator world before it's sort of settled out into this really nice world that we live in today, where there's still a lot of confusion, like who do I buy my Kubernetes from? What are the cloud providers doing? What are the vendors, like Mesosphere, or Cloud Foundry, Pivotal, what are they offering me? It's still confusing, but it's at least more optimistic than it was when it was like container orchestration wars and you had these competing standards between Mesos and Kubernetes and the other orchestrators. How does that compare to where we are today? What's the environment and where do you think things are going?

[00:32:10] AM: Okay. If you look at the way we have setup the projects, specifically LF Edge, we always believed in the Linux Foundation that it's all about harmonizing the communities. What that means is harmonizing other open source communities and harmonizing the standards and the consortiums that work in this area.

So we would go out and formally, legally setup collaborations framework or unofficially do that if it's an open source community. For example, if you look at the LF Edge today with projects like Akraino and EdgeX Foundry and others, there are working groups in that project and they're working alongside Kubernetes as well as ONAP as two projects, one for cloud, one for telecom, and both of those open source projects, Kubernetes has an IoT working group that is working with Akraino. ONAP has an edge working group. So IoT and edge, right? That both of these

projects have developers that are collaborating with the LF Edge and Akraino at stack to make sure that as they create the load, the write level of distribution at the – "Do I need hardware awareness? Do I need to know how much memory is there and then I can optimize the load appropriately? Do I load all of the modules from ONAP down at the access and edge layer including closed loop control, data analytics, predictive analytics, all of that, or do I just load policies that will sort of first figure out that I don't have enough compute cycles to crunch numbers here. Let me connect it back into a cloud."

Again, I'm going too much into detail here. But these types of collaborations are all formal and happening as we speak. That's the first set. So open source to open source collaboration with other open source projects. The second collaboration that is happening – And by the way, we are setting it up – We have learned the lessons over the last years. Public wars on standards and open source options, while initially they are great because we're still learning, when it gets to a point where people now really want to make this happen, collaboration is the best way to do it. What we have set up is we have set up a collaboration with, like let's say if there's an edge standard. In this case, it's [inaudible 00:34:29] for telecom, okay?

If there's a consortium, so for automotive. Now, here is where we get into vertical consortiums. Automotive, it's ACC. So ACC is a member for LF Edge as well. So Toyota is kind of sharing that, and essentially what it is is it's how do I take advantage of the edge for automotive industry. There's an IIC consortium, who is a member of LF Edge. How do I make sure that their devices and drivers and everything that EdgeX Foundry does is aligned to IIC including OpenFog now, etc.

So what we are saying is in my view, the market and the industry needs to be harmonized in order to get more adaption, because think about it this way. If you are an end user, a developer, a vendor, a supplier or a user, you're like, "Where do I go? Do I go here? Do I go here?" Who's going to win? This is more like, "Oh! They're all going to figure out in the open, and so it's a safe choice, and so now let me invest my resources to solve things."

I'll tell you a little secret having been in open source for so long. At the end of the day, if you put engineers in a room from anywhere vertical, any competitors, any companies. They will figure out the right solution irrespective of geography, irrespective of whether they work for a

competition. They don't care about the business aspect, and that's what we are facilitating here, which is developer to developer solving these hard issues, but providing a framework for collaboration.

[00:36:12] JM: Let's give an example. AT&T contributed the Akraino Edge Stack, and you've mentioned this a few times. What is Akraino.

[00:36:21] AM: All right. So Akraino Edge Stack is a software project which is bringing three things to the table under LF Edge. So the first – There are two use cases that they're going after. There is a telecom only use case, meaning if you're an AT&T, or an NTT, or a telecom or whatever. If you're a telecom provider, how do I automatically extend my reach and my software stacks in an automated manner to provide edge services and all the edge use cases for all my 5G deployments and all my apps that are telecom-centric, VPC, VNFs, things like that. So that's one use case.

The second use case that they are solving is how do I work with an enterprise use case that will allow for deployment close to the enterprise, maybe as a managed service. So then it's inside the demarcation point at the enterprise or outside the demarcation point, right? So that's the second use case. Then the third thing it's bringing from an edge stack perspective is a set of validated blueprints that cut across the entire edge and non-edge umbrellas. A blueprint for just terminology's sake is a set of validated reference solutions and implementations beyond the architecture that has a specific hardware software and a stack attached to it, and these blueprints are aimed to satisfy the specific deployment scenarios and the use cases.

So Akraino Edge Stack, the technical community has approved 19 blueprints right now, and these blueprints – And I'll give you an example. There are blueprints for a smart city that would probably start with a micro-MEC architecture based on, say, ARM or Intel, and then it can sit with a 5G or wireless connection on a light pole or a vehicle, and there's hardware and then there's an amount of software that sits on it.

There could be a blueprint for micro-CPE gear, which sits in a manufacturing floor that is for universal CPE for IoT appliances, which is a very [inaudible 00:38:37], very thin orchestration. There's a blueprint for 5G, which is the radio cloud if you may. There's a blueprint for

Kubernetes native industrial automation infrastructure, where you basically are taking native container workloads using a subset of Kubernetes technology and then putting it on a set of equipment that's sitting on the factory floor, etc.

That's what Akraino Edge Stack is doing. These blueprints, you can access the details of this through lfedge.org under Akraino. There's a process, there's a technical detail. So what effectively happens is if I'm an end user and I want to sort of say, "Hey, I have this user case. Has open source solved this issue?" They can go at it and download the code, get the support and just be ready to deploy. So that's what Akraino Edge Stack is.

[00:39:34] JM: This was contributed by AT&T.

[00:39:36] AM: Portions of it was contributed. Some of the seed code, but a lot of the other seed code is being developed both in upstream as well as in the community itself now on other use cases. So there are 60 members and there's like 28 premium members that are all contributing code. It's being developed right now as we speak.

[00:39:57] JM: This is being used by AT&T or is it like their vision for what they want?

[00:40:03] AM: This is the code that they had developed for their own use that they contributed, and now it is going to get deployed as part of the prep work for 5G. So they contributed it back with an intention of expanding the community, and then along with that, there's another 18 blueprints that are being created. So the code they contributed was specifically for probably two blueprints.

[SPONSOR MESSAGE]

[00:40:38] JM: HPE OneView is a foundation for building a software-defined data center. HPE OneView integrates compute, storage and networking resources across your data center and leverages a unified API to enable IT to manage infrastructure as code. Deploy infrastructure faster. Simplify lifecycle maintenance for your servers. Give IT the ability to deliver infrastructure to developers as a service, like the public cloud.

[INTERVIEW CONTINUED]

[00:42:01] JM: To get into an area that I worry is either a buzzword or it doesn't have enough to do with software, but can you explain to me what 5G means? Is that like a collection of things or is it a hardware? Is it software? What is 5G actually mean?

[00:42:19] AM: So 5G, to a layman, is just going to be, "Hey, I had my phone. It says 4G. Now it's going to say 5G. That means I have to buy a new phone with a much faster – I'll get more bandwidth much faster." That's the average assumption, and that is not incorrect. That is fine, but that only happens to be 10% of the use cases of 5G.

5G, although it is a radio technology. So it's over the air technology where you're getting effectively a 10th of the latency. You're getting 100 times the number of devices you can support in that. You are getting 100X the bandwidth. So it's a next generation of air technology, or radio technology, which it's bigger, better, much different.

Now, if we were on a linear curve where if you look at 2G, it was for straight texting. 3G was for like SMS and a little bit of web. 4G was for video and things like that. What 5G opens up is a set of machine-to-machine communications. It opens up all the entire IoT communications. It opens up a new set of applications that require low-latency.

So the reason 5G is so important is not only as a radio technology, but when you do that, your entire data center, your cloud, your backend systems need to be automated, because both orchestration-wise as well as lifecycle management-wise. Because if you look at it, you cannot

have an operator sitting on a phone saying, “I want to give this sensor or this car a service,” and then that car stays in service for a year. It’s not like a phone registering on a network. It is really things come up, things come down, things are active for just milliseconds, not years, and the backend and the glue needs to be fully automated. So that’s why you’re seeing 5G used as a proxy to upgrade the backend infrastructure of not just telecom, but even for enterprise networks.

So if you look at inside the data center of an enterprise, you got right now hardwire connections that are fully single pipes coming out into a carrier who is just a [inaudible 00:44:42]. Well, with 5G, you can actually eliminate a lot of wiring. You can actually go from device to device and still hit the cost curves, hit the automation and things like that. It changes the mindset for the enterprises as well. So that’s in a nutshell what how I see 5G. So it’s not just the technology, it’s the implications of the technology that are the talk of the town.

[00:45:07] JM: What I have heard about the CPU industry is the tik-tok term, where you have all these different players involved in the CPU days or in the CPU market and you have Intel, and then Intel says, “Okay. This is the plan for how CPUs are going to advance.” Then it has all these implications for other players in the industry, but it really helps to come to this agreement on like, “Here is what we’re going to do according to Moore’s Law,” and Intel kind of describes to everybody and then all the suppliers and everybody kind of falls inline and then you have this tik-tok kind of thing. Is 5G sort of like that? The 3G, 4G, 5G?

[00:45:48] AM: Yeah, it could be a good analogy. The only thing I would add is the implications of 5G are way more than the migration from 1, 2, 3, 4G, okay? 2, 3 and 4G were mostly linear evolutions. 4G was absolutely a step up, because it was an all data network first. It was not just voice and data. But 5G is a much more exponential change, primarily because of the machine-to-machine and the proximity to the edge that it can provide.

[00:46:22] JM: Can you explain in more detail? Why is 5G such a big upgrade?

[00:46:25] AM: Yeah. As I said, the reason it’s a big upgrade is – Okay. So think of it, what did 4G enable as a market? Fine. We got better iPhones and we got better laptops, and we got faster speeds and all that. The market that it enabled was mostly social media and videos and

YouTube's and all that. At the end of the day, it's still linear thinking. You could do that with 3G, but it was still slower.

5G, you can do all that and you can do – But what it enables even more because of the inherent specifications inside of latency and Jitter and all the network characteristics is the machine-to-machine communications. What that does is it gives a whole different meaning and a scale to connected devices. Here we're talking from 3 billion phone connections, or four billion phone connections to 20 billion connected devices. Now all of a sudden the connected devices have each separate use cases in many, many verticals. So the fleet and transportation department may use it differently. The homeland security and the traffic departments may use it differently. The industrial automation departments may use it differently.

So it has profound the implications in many, many verticals, as well as it has implications on infrastructure that needs to be redone, because some of these data centers are sitting below a base station. Some of them are sitting in a neighborhood. They're not all sitting in a massively scalable data center somewhere a centralized location. Does it help?

[00:48:06] JM: Definitely helps. What's interesting to me about LF Edge and who is involved is – So you have AT&T who is I guess they contributed, as you said, parts of the Akraino Edge Stack, but they are maybe the flagship contributor, or I don't know, a flagship contributor.

[00:48:25] AM: For part of one of the projects. Then you have EdgeX Foundry is the other project, which is it was seed coded by Dell EMC. Then you have another project for the on-prem edge, which has been contributed by a startup called ZED Data. Then we have Home Edge. That contribution seed code comes from Samsung, etc.

So those are just seed codes. Seed codes just help you jumpstart portions of the project, and then the community builds on it. Today, if you look at our community, we launched with 60 founding members, which is like phenomenal given just a short time. But it's a cross-section of hardware renders. So you got ARM, Qualcom, Intel, all of them participating. Then you have the carriers, so AT&T, NTT, etc. Then you have the cloud, Baidu, Tencent, etc. Then you have the industrial players, Dianomic, OSIsoft, etc. Then you have the system vendors, suppliers,

integraters, the usual, Dell, HP, Ericson, Nokia, etc. So a whole bunch of players are all coming together, and that's the implication of seed code, but more importantly the community.

[00:49:39] JM: Didn't Facebook try to do something like this with their open compute project a while ago?

[00:49:45] AM: Facebook has a small working group in the OCP. So there are bodies or organizations that are dedicated to hardware. As I said, LF Edge is a software only project. So everything that is done in hardware is complementary to us. So we work with – So there's actually an OCP hardware that will be part of the blueprint that is going to go into Akraino Edge Stack. So it's a complementary.

[00:50:12] JM: Oh, okay. Right. Okay. Cool. So Facebook sort of – I think if I remember correctly, we did a show about this a while ago, but they open sourced kind of their server infrastructure, like the hardware infrastructure.

[00:50:23] AM: Correct. But again, where is the software coming from? That comes from Linux Foundation. Yes.

[00:50:30] JM: Yeah. So what I was going to say is like AT&T, DELL EMC, these are – I mean, I think they have had some open source stuff going on for a while, but I imagine they've looked at the Cloud Native Computing Foundation, they've looked at the Linux Foundation. They've sort of seen where the tide is going over the last 4 or 5 years, and maybe is this them sort of signaling that they are making a big push towards open source?

[00:51:02] AM: Yes. In fact, not just them, ARM, AT&T, Baidu, Dell, Ericson, HP, Huawei, IBM, Intel, Juniper, Mobile AJAX, Nokia, Qualcom, Red Hat, Samsung.

[00:51:18] JM: There are people listening to this who have less context on the significance of this. Like maybe they've only been in the software industry for three or four years and they're like, "Yeah, open source. Whatever." But you've been in the industry for a while. Explain why this is a big deal that all of these major enterprises are all of a sudden getting into open source.

[00:51:37] AM: Very good question. I think we could have started with that, but that's okay. Leave the best to the end. But open source in the last 10 years has significantly changed the way software is done. The primary driver for this is having control and options with full interoperability. So if you ask an end user, "Why would you go with open source?" They're like, "I don't want vendor lock-in. I want to make changes and customize based on my needs, and I want to sort of control my own destiny, because I'm slightly different." That's the primary driver for open source.

The second driver is it is a complex plumbing problem and not a single vendor can solve it. Even if they put hundreds of millions of dollars of R&D. I mean, take the example of ONAP. You got 500 to 1,000 people working on it. Like our LF networking projects, open source projects. If we do a straight developer head count to business value, it's over $500 million of R&D that has been created in the open.

Now, imagine a single company investing that much money on a non-differentiating set of software. It's not going to happen. So that is what has really picked up in terms of why open source is such a big deal. Then of course, there is a cost equation here where if you're not investing and you're utilizing – So let's say I'm a Red Hat or I'm an IBM, I would put, say, 10 designers on LF Edge, for example. But there are, say, 200 people working on it. So not I've put in 10 people's worth and I'm getting 200 people's worth of work back to what I need. That's a straight ROI instead of putting 200 people. So this is the power of community.

I mean, there are lots and lots of papers and research done on this, but the key here is open source is here to stay. People have realized that. The business models have changed. The industries are picking up, and each vertical industry is on a different path, if you may. So cloud started off with open Bing, like the first in mind. But then you have these hundred-year-old industries. Telecom is 142-year-old industry, and that has just moved to open source in just the last three years. Automotive Grade Linux, which is another one of our open source projects. Automotive is a 100+ year industry. That just moved to an open source in the last two years. Banks, financial techs, they have started moving to blockchain technologies, which is now hyper-ledger and part of open source. So you would see different timing, but in general it's those values that are brought by the shared creation of intellectual property in a very seamless manner that is really proving itself.

[00:54:38] JM: The moment when I realized that open source was going to take over everything was actually in a conversation with my mom, who knows nothing about computer science, but she manages some properties. She owns some real estate and manages these properties, and she was telling me about a refrigerator that he had a problem with, and she's like, "I want an open source refrigerator." I'm like, "Mom, how do you even know what that would mean?" She's like, "Well, my refrigerator keeps breaking, and the parts are like $500 for like some niche screw that you need in your refrigerator. Then you need the person to come and install it correctly, because their refrigerator is highly integrated and it's just this thick piece of hardware." I was like, "Mom, thank you for making me realize that open source is going to eat the world, including refrigerators."

[00:55:32] AM: Can we sign her up for leading marketing here?

[00:55:35] JM: Exactly! I know! You've got LF Edge. Now you need LF Fridge.

[00:55:41] AM: But the concept is absolutely correct. If you put yourself and replace refrigerator with a piece of hardware, a piece of software, a piece of application, a piece of devices, that's exactly the notion, which is if I'm smart enough, I will repair it myself. But let's have interoperable parts. Let's not rely on just one particular bland. That's exactly open source, yes.

[00:56:05] JM: Well, Arpit, we barely scratched the surface. I'm sure we will be doing more shows about edge projects, and it's really nice to get acquainted. I'm sure I'll see you at some conference or something, and I look forward to doing more shows and covering the development of open source edge.

[00:56:22] AM: Yeah. No, absolutely. You're welcome to come over to Open Networking Summit, that happens in the Bay Area, April 3 to 5 this year, and there's a whole track on edge technologies. Just take a look at LF Edge –

[00:56:39] JM: Where is that?

[00:56:39] AM: It's in San Jose.

[00:56:40] JM: Okay.

[00:56:41] AM: Yeah. So you can go to Linux Foundation Events Open Networking Summit and you should be able to get details and register there. Plus look at lfedge.org, and there are wikis and every document is in there as well.

[00:56:54] JM: Sounds great. Okay, Arpit, thanks for coming on the show. It's been really fun talking to you.

[00:56:57] AM: Yeah, it was fun. Thank you for the questions.

[END OF INTERVIEW]

[00:57:03] 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]