EPISODE 983

[INTRODUCTION]

[00:00:00] JM: Cloud infrastructure is usually consumed in the form of virtual machines or containers. These VMs or containers are running on a physical host machine that is also running other VMs and containers. This is called multi-tenancy. Servers across cloud providers such as AWS have a high-utilization because there are multiple virtual instances running on each physical server host.

Cloud computing has led to a low cost of compute infrastructure, but in some cases this low cost comes at a price of not being able to control the underlying hardware with as much precision as the user would want. Some users want specific types of hardware. Other users want to be using dedicated hardware that does not risk the noisy neighbor problem of sharing a physical server with some other application that might be hogging most of the other resources.

Packet is a company that provides remote access to bare metal infrastructure. The user experience is similar to that of a cloud provider, but with more control over how a given physical host will be used. Zachary Smith is the CEO of Packet and Nathan Goulding is the chief architect. Zach and Nathan join the show to talk about the business and the engineering behind Packet as well as the future goals for where they want to take the company.

If you are planning a hackathon, check out FindCollabs hackathons. Whether you are running an internal hackathon for your company or you’re running an open hackathon so that users can try out your product, FindCollabs hackathons are a tool for people to build projects and collaborate with each other. FindCollabs is a company I started to allow people to find collaborators for their software projects, and our new hackathon product allows you to organize your hackathon participants to make your hackathon as productive as possible. Check it out at findcollabs.com.

[SPONSOR MESSAGE]

[00:02:00] JM: Looking for a job is painful, and if you are in software and you have the skillset needed to get a job in technology, it can sometimes seem very strange that it takes so long to find a job that's a good fit for you.

Vettery is an online hiring marketplace to connect highly-qualified workers with top companies. Vettery keeps the quality of workers and companies on the platform high, because Vettery vets both workers and companies access is exclusive and you can apply to find a job through Vetter by going to vetter.com/sedaily. That's V-E-T-T-E-R-Y.com/sedaily.

Once you're accepted to Vettery, you have access to a modern hiring process. You can set preferences for location, experience level, salary requirements and other parameters so that you only get job opportunities that appeal to you.

No more of those recruiters sending you blind messages that say they are looking for a Java rockstar with 35 years of experience who's willing to relocate to Antarctica. We all know that there is a better way to find a job. So check out vettery.com/sedaily and get a $300 sign-up bonus if you accept a job through Vettery.

Vettery is changing the way people get hired and the way that people hire. So check outvettery.com/sedaily and get a $300 at bonus if you accept a job through Vettery. That's V-E-T-T-E-R-Y.com/sedaily.

Thank you to Vettery for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[00:03:51] JM: Nathan and Zach, welcome to Software Engineering Daily.

[00:03:53] ZS: How is it going?

[00:03:55] JM: Going great. I am a millennial and I don't ever want to touch a server. I like infrastructure as a service. Why would I ever go anywhere other than Amazon, or Google, or Microsoft for my cloud provider?

[00:04:12] ZS: And you're a millennial, yeah? I mean, solid like 1983.

[00:04:16] JM: '88.

[00:04:17] ZS: '88. Wow! Okay. VCRs were not in your world, huh?

[00:04:21] JM: No. They were. They were.

[00:04:22] ZS: Well, you asked a really interesting question.

[00:04:24] JM: Even LaserDisc. LaserDisc also. I remember LaserDisc.

[00:04:28] ZA: All the good ones. Well, that's a good question. When we started Packet we said, "How can we automate infrastructure for a new generation?" Which kind of boils down to millennial developers, frankly, because you and your friends are slowly, if not so slowly taking over the world.

We thought how we could give an out-of-box experience that was highly automated and yet, and this is really cool part, allowed you to innovate all the way down? Our point was could we give you not crappy, really awesome programmatic control of hardware sitting in somebody else's data center that you'd never have to go and touch.

What people do with us here is mainly they innovate one layer deeper where they have a strong amount of opinion on the software that they write, the hardware that they run it on, or the places that they put it, and that's where they come to us versus one of the hyperscale clouds that frankly are really awesome experiences but push you in the high amount of software abstraction.

[00:05:27] NG: Yeah, and I'll add to that. I think that a few other emerging trends, specifically around container and container deployment. Really, you kind of obviated the need for the hypervisor to kind of be there in terms of you have the bare metal. Then on top of that you layer on your hypervisor, and then on top of that you layer on your container runtime environment. Then finally, ultimately maybe you have your application there.

Being able to handover a bare metal system in the same time that it takes progression of virtual machine on one of the hyperscaler clouds and in some cases actually even faster allows people to run their containers on native metal without having to have a bunch of kind of orchestration layers in between, which just makes it much easier to kind of maintain and control and also at the end of the day gives kind of a better performance experience as well.

[00:06:13] JM: The overhead of the orchestration layer that you're referring to, this is epitomized by Amazon EC2. If I spin up an Amazon EC2 instance, it's a virtual machine. It's sitting on a hypervisor on an Amazon box somewhere in an Amazon data center and that virtualization layer, the hypervisor – Hypervisor is a term for basically a virtualization, a VM manager, and these different VMs that are being managed by the hypervisor, well, these VMs are competing for resources. If you have a VM on that hypervisor that's eating up all of the space on that physical host, then when you get your EC2 instance provision on that same physical host as some other greedy application, you have what's called the noisy neighbor problem.

When you have more control over the actual physical infrastructure, you're not going to be subject to that noisy neighbor problem. What are the kinds of companies that are sensitive to this noisy neighbor problem? I mean, most applications, can I just go on EC2 anyway and like if there's a noisy neighbor, I can work around it? Why do I care about this problem?

[00:07:25] ZS: I'd love to take this one because I thought of the coolest millennial kind of description right now you were talking. It's kind of like the difference when you're coming back from JFK late at night and you're like, "Should I take Uber pool or Uber X?"

[00:07:38] JM: Uber X every time.

[00:07:38] ZS: Every time, right?

[00:07:40] JM: It's late at night. I don't have time for pool.

[00:07:43] ZS: No time for that sharing. No noisy neighbor situation. Anyways, for the vast majority of developers and companies actually using an Amazon style service where you get a ton of scale – I mean, it's a pretty generic product, right? You've got it in U.S. East or U.S. West or whatnot. You've got relative kind of menu to choose from related to these services. This is totally fine. This is actually completely good, right? It's like 80% bell curve of internet infrastructure, or as I like to use an analogy, it's the retail checking account of IIS. Almost everybody needs it. It's totally fine for them, except if you have a billion dollars in your checking account and then it's the wrong product.

What we find is that people who are really pushing the limits on use cases, they almost always start to have a strong amount of opinion about the software that they run, including all aspects. Who they share it with? What it's running on? What runtime? Maybe they invented their own thing. Maybe they're using a very specific down dom0 kernel or whatever it is, right? Also, kind of where they put it, right? You have an architecture question or a location question or anything else like that.

This is the type of thing that you don't get to control within a hyperscale cloud, which is making the vast majority of the decisions for you so they can give you a great price on a retail checking account. What Packet is looking for is who are the customers? We kind of think of it as the next 1,000 or 2,000 technology-enabled enterprises of the next decade who are really winning their business or building their experiences or growing their markets based upon a technological advantage.

I think that once you're building a really big business based upon technology, you end up getting a lot of opinion about these things that you're running on the software that you're using or you might be sharing with or where you're putting it. The initial use cases on our biggest customers today might be something you're carrying around in your pocket, which is a specialty amount of very specific hardware and very specific software called your phone, which is not generic at all. It's actually super specialized.

Some of our biggest customers are running large scale 5G networks where they have an ultimate amount of opinion around the hardware that they're going to run a very specific application called a 5G network on and where they're going to put it, which might not make sense for almost anybody else. For example, putting wireless infrastructure into Tukwila probably doesn't matter to most people, but it matters a lot to a few people.

That's the real differentiator of Packet, is we give you kind of a very portable layer of automated hardware that we'll take anywhere for you. We have a model called our edge model where we partner with a real estate and we'll put that in any city town around the country or the world and run it for you.

[00:10:29] JM: Can we go a little bit deeper into the telecom use case? Because 5G and telecom infrastructure more generally is something that I think a lot of software engineers, they understand that there is a heavy, heavy software component to this. But it's very much a black box. I know that like when I make a phone call, that phone call is being routed through like cell towers and stuff, but like what is actually going on there?

[00:11:00] ZS: Millennial place phone calls. I thought they just text, but they –

[00:11:01] JM: I had to call an Uber driver yesterday. I call my mom. You have to use the communications API that your forbearers used. But anyway, so like the telecom infrastructure. What is this software? Is this a big C application or C++ application that's routing telephone calls that needs to be like super-duper fast because it needs to route from one phone number to another? Can you give a little bit of context so I have a little bit more understanding of what the actual application that the telecom provider is running looks like and why they cannot run that on a heavy multi-layer virtualization stack that they don't control?

[00:11:47] ZS: Well, let me take a stab and then Nathan will correct me. First and foremost, we're going through a really, really big shift in the world of mobile right now. We're going through not only a technology change, e.g. from 4G to 5G, which is causing a massive infrastructure upgrade, technology upgrade, etc., the cloud is kind of hitting these guys.

If you think about when was the last time big wireless providers, which is a multi-hundred billion dollars a year industry, upgraded their networks really? Well, that was pre-iPhone, right? So LTE was just coming out. It was planned in the early 2000's. Rolled out improbably what? 2007, 8, things like that. If you think about this, this is primarily a network that was built for voice. That was what most people were doing. Maybe a little text messaging, and things have just dramatically changed since 4G started being rolled out or LTE was rolled out.

At the time, you think about the world of the cloud, I mean, AWS started in 2006. There is no cloud when people were designing 4G. It was just like really hardcore IT. People built 4G networks with highly centralized appliance black boxes from the likes of Ericsson and Nokia and whatnot.

I guess it was Alcatel, Lucent or something at the time. You have these highly proprietary Silicon, maybe ASICs, or line cards or whatnot that have embedded software on it that do like one thing called like packet-switched voice, and they do it really well. They do a lot of metering. They calculate every single call that you're doing every minute is going on. How you roam between radios, like lots of math is happening and it's being done in a very embedded and proprietary manner.

What's happened over the last 10 or 15 years is that not only have our usage wireless devices kind of completely changed. I mean, I remember laughing probably with Nathan in 2007. I had a Blackberry and I was like, "I would never watch a video on this thing." It's true. I ended up getting an iPhone and now my kids don't even know what a TV is. We've kind of changed our habits and how we use wireless in general, and that's continuing to evolve. In fact, I think it will change even greater over the next 10 years. I certainly hope that in 2030 if we're having this podcast that we're not walking around Manhattan staring at our iPhones running into each other all the time trying to text something. My suspicion is that wireless will continue to evolve rapidly.

Not only of our use of the devices changed pretty incredibly. We're no longer doing voice so much. We're doing all kinds of other stuff that look a lot more like the Internet. Then the other thing that's happening is the cloud hit the wireless guys. They're building wireless networks right now that look a lot more disaggregated. They started to say, "Okay. What if I have hardware? Commodity hardware like x86 or ARM-based architectures with software that we could innovate on more rapidly than once every 10 years?" We could do updates on a regular basis and put out new features and react and change and maybe even buy from different vendors or maybe write some of it ourselves.

The world of portable software has hit the mobile carriers as well. That's kind of the big trends that I'll give you in terms of how it actually runs. It's a lot less sexy and I think these are actually starting to look more like cloud data centers. They're running servers now with some sort of software either virtualized software running in say an OpenStack environment or whatnot or containerized software that's basically accessing Linux substrate directly.

[00:15:10] JM: Real quick. Just to respond to that or sort of summarize it and make sure that I understand, because I thought it was a beautiful answer, and then we can kick it to Nathan for –

[00:15:18] ZS: To correct me.

[00:15:19] JM: Well, for his own perspective on what –

[00:15:21] NG: That was a beautiful answer. That's a high bar to clear.

[00:15:24] JM: Well, maybe you can give a bit of a lower-level answer and I'll try to bridge the gap between you two. But I think, Zach, what you said is basically the telecoms are starting to realize that they are the layer between this highly dynamic cloud infrastructure and this highly dynamic mobile infrastructure. In the mobile infrastructure world, we know how fast our smart phones are getting better. We know that we need to like update our phones on a regular basis or else it's like you have an out-of-date magic wand and it's like your magic wand can't like cast the spells that you need to cast as fast as you want to cast them. Whether it's summoning your Uber or ordering your groceries or –

[00:16:06] ZS: Or catching your Pokémon.

[00:16:07] JM: Or catching your Pokémon. You want to be able to do all these things really fast. As you are doing these things, your phone is generating packets of information that are being sent across the world to a cloud provider or being routed between different cloud providers. On the critical path of that routing infrastructure, that information passage infrastructure, our telecoms. Telecoms are basically routing just these bits and bytes that are in certain patterns. They're coming in certain bursts.

If you just think of it from a very low level point of view of streams of bytes and bits going between the kind of mobile client and the cloud provider backend, the heavy cloud provider backend, there's a ton of logic. There's a ton of room for optimization and the algorithms for doing that optimization are always going to change as long as the cloud providers are changing and the mobile applications are changing.

It makes sense for them to have this basically like say we don't know what the future is going to hold. The best way to plan for that future is to have dynamic infrastructure, and dynamic infrastructure means software infrastructure. If we want our software to be as perform and as reliable as possible, if we want to keep our moat, these telecoms have a great position right now.

If they want to keep their moat, they need to make their performance idealized, and idealizing that performance probably in many cases means factoring out as much risk, as much complexity, as much noisy neighbor problems as possible, which would mean having somebody that is very focused on the performance of the hardware, like Packet, to be handling that later.

[00:17:49] NG: Yeah, and I can speak about a little bit. I mean, obviously, it probably goes without saying that these telecom networks are incredibly complicated. But, fundamentally, there are a few different components that go into it I'll say. There is obviously the radio, which is RAN, radio access network. That's actually a thing that communicates with the radio that's in your wireless device. Does all that really complicated math to allow you to jump from one tower to the next tower to the next tower and send all the bits.

Then from there, it actually connects to what's called the EPC, Evolved Packet Core, which is actually where you get your S gateway and P gateway which is kind of where the packets and voice data will flow out to the network. There's been a big push to – Again, I will say that there's only a few very large operators. There's fewer that have spectrum, but there's been a big push to move the Evolved Packet Core closer to the edge.

What that does is it allows – Instead of saying all of the traffic that's going to be coming off of my mobile device or all the voice data is going to be going to a handful of CO, central offices for the telcos and saying, "Okay. Well, can we actually push this closer and closer to the edge? Can we bring –" Instead of everything routing through one large data center somewhere, can we actually say, "Well, it's going to egress from the point where that's closest to the end user so that that gives the end user –" Well, it enables two remain primary things.

One, it gives the user a better performance. The latency to say loading that video or the throughput of that doesn't have to go across big transcontinental links, but can actually happen and be served directly from or very close to where the actual end user is.

That's the first component of that, but it also allows – If you do that, it allows you to actually put other compute next to that as well. You say, "Okay. Well, we're egressing to the Internet or appearing to a provider very close to where the end user is." But that also allows, because of that, we're accessing the broader Internet right there. We actually have the ability to put things that require a lot more compute. Whether the CPU or traditional CPU, but it allows us to put the application extremely close to the end user, which unlocks a lot of capabilities around obviously AR, VR implications but also other things that are maybe just a little bit more traditional. Things that will just naturally need to scale more as there are more users downloading more videos, playing more games, doing things that are being done.

Again, instead of kind of upgrading massive, huge data centers, which are going to have to happen anyways, but it kind of decouples that a little bit in terms of scaling this beyond is to say, "Well, that's actually allows us to be a little bit more fluid to allow us to put kind of hyper-dense solutions and compute into locations where we need that without having to kind of forklift the entire thing or do this massive upgrade all at once. There's a lot more flexibility that's being introduced that allows this to kind of take place in the telecom space.

[SPONSOR MESSAGE]

[00:20:46] JM: When you start a business, you don't have much revenue. There isn't much accounting to manage, but as your business grows, your number of customers grows. It becomes harder to track your numbers. If you don't know your numbers, you don't know your business.

NetSuite is a cloud business system that saves you time and gets you organized. As your business grows, you need to start doing invoicing, and accounting, and customer relationship management. NetSuite is a complete business management software platform that handles sales, financing, and accounting, and orders, and HR. NetSuite gives you visibility into your business, helping you to control and grow your business.

NetSuite is offering a free guide, 7-key strategies to grow your profits at netsuite.com/sedaily. That's netsuite.com/sedaily. You can get a free guide on the 7-key strategies to grow your profits.

As your business grows, it can feel overwhelming. I know this from this experience. You have too many systems to manage. You've got spreadsheets, and accounting documents, and invoices, and many other things. That's why NetSuite brings these different business systems together. To learn how to get organized and get your free guide to 7-key strategies to grow your profits, go to netsuite.com/sedaily. That's NetSuite, N-E-T-S-U-I-T-E.com/sedaily.

[INTERVIEW CONTINUED]

[00:22:34] JM: I've heard this term colo. I think this is like a pre-millennial sort of term.

[00:22:44] ZS: Well, the cloud is just somebody else's big computer of the data center.

[00:22:49] JM: Colo, there's a company, I think Equinix. That is a popular colo.

[00:22:56] ZS: That's a big one.

[00:22:57] JM: What I have heard about a colo is basically – I mean, the pitch for a colo that I've heard is like very similar to the pitch for Packet, as I understand, if you need dedicated servers. Colo's have been around since the 90s as far as I understand. You rent server infrastructure, basically, but you just have to rent it in kind of a less API-driven on-demand way than a cloud provider. Can you explain what a colo is and give me like a brief history of colo and how that's different from Packet?

[00:23:32] ZS: Yeah, totally. Well, colocation facilities or colo effectively allow you to colocate your equipment in somebody else's building, right? The reason why this industry came about and why Equinix, for example, is one of the leaders, is because the Internet started by having a place to co-locate that wasn't the phone company. Phone companies had networks, but how could network stitch together?

I believe it was the PAIX, right? Palo Alto Internet Exchange that started, Nathan? Out in the West Coast in the mid-90s. Jay Adelson? It was the place where people came to interconnect and connect their networks, and this became kind of the fundamental part. You had AOL over in Ashburn and you had like other networks that were coming around. Well, it wasn't really the Internet, because people couldn't freely interconnect over standard HTTP and route packets between each other.

So colocation came about as just a building with good facilities like air conditioning, and power, and whatnot that also allows you to house a router and get you wires between each other's equipment. It since become supercritical, something like Equinix has. I don't know. They have like 100 or 200 locations around the world. We call them like the major football cities of the Internet. People like Digital Realty, Interaction, CyrusOne. These are the major real estate companies now that offer data center space to enterprises, service providers, cloud companies and whatnot.

What you get is you get like the ability to have a building to put all your stuff, all that physical maintenance and capital, but you also get a place to meet other customers. That's the really important part of why colocation exist and becoming ever more interesting so to that way people can interconnect and pass traffic to each other. As you probably notice, when you're coding in a webpage or something, there's a lot of different services that are loaded, hundreds probably to load up a popular webpage. Interconnecting and routing that traffic is super important to the overall performance of the application, the economic, etc. Okay?

[00:25:37] JM: This is totally different than what I envisioned. That the main feature is like if Twilio has to do a ton of routing to AWS, then maybe they would want to share some infrastructure in a Colo to accelerate. I mean, there's kind of a win-win situation there to colocate.

[00:25:55] JM: Exactly. Instead of us reaching over the Internet as it were, you can – Most likely, you're interconnecting with AWS there in Equinix's Ashburn, Virginia facility. Equinix gives you a neutral place to let that occur. It's very hard, to me, like an Amazon doesn't want you walking into their data center with your router and saying, "Could I bring my fiber and router and put it in your data center?" They don't want that, right?

They're like a big, giant private data center. Somebody has to play the role of the neutral party to allow Verizon, and Twilio, and Amazon, and level III to get together and pass bit between each other. That's the major reason why service providers meet at something like the Equinix.

Enterprises like to be there too, and this is kind of a trend that's been happening over the past 20 years, is obviously some enterprises had made the shift to the cloud, but most are still just getting out their corporate data centers. By corporate data center I mean literally their IT closets or data center they built on their corporate campuses, which are just for them. That was kind of the style for a long time, is I have a very important data. I do critical things. I have lots of applications. I'm going to run them where I can control all of that.

But now so much of the value that enterprises find out technology has to do with accessing technology ecosystems. Different providers, different SaaS companies, having your data to being able to use the machine learning service on it that maybe you don't run at some third-party, right? How do you do that in your own private data center? You have to bring in like MPLS lines and fixe transit and kind of connect to people using dedicated lines or the Internet, which can be extremely variable or not encrypted or other things like that.

People are finding enterprises are moving into those neutral places, taking their equipment out of private data centers and into neutral data centers, like Equinix, so that they can exchange and kind of reach those ecosystems.

A lot of them move other applications into the cloud directly and kind of access those ecosystems at a higher level resource. But if you say have a couple hundred petabytes of storage in a legacy sand, you’re not like shipping that in Amazon, right? You’re putting it someplace where you can use applications that might sit in Amazon against that data. You’re going to do that over some sort of a private interconnect.

[00:28:14] JM: Okay. If I were to boil down what Packet is and how it fits into the competitive landscape, it's basically a cloud provider with no virtualization layer. Would you say that’s an accurate way of describing it?

[00:28:32] ZS: Yeah. I mean, I would say yes. I mean, we basically, as I tell my wife when I come to work every day, I’m like, “Oh! Just go back to work to turn the servers on and off all day long.” What we basically do is there’s this layer of the Internet which we just described called data centers. I said data center as a service. You can get access to a data center without having to build your own building, and that's – You go to an Equinix or a DRT, right? Get some colo space. Put in your own equipment. Off you go.

There’s this layer called infrastructure as a service, which is anything from turn my server on or my virtualization and my VM in EC2 all the way up to load balancer or a VPC or something else like that, object storage. That’s considered IaaS. Then above that you have SaaS, right? Give you my Salesforce, my Office 365, and maybe you have a PaaS system as well. Just run my app me. Here’s binary. Okay?

What we do and what we think is missing is between the data center as a service and the infrastructure as a service. we think there is a missing very, very large sandwich called hardware as a service, and that's basically targeted at millennias like yourself. How do we put hardware, generic or special, big or small, x86 Intel or an embedded ARM device in a place? Run it for you so that you can put the software, e.g. that infrastructures as a service layer. You want to run VM's on it, containers, serverless stuff. I don’t know. Something you invented yesterday. All good. How do you we give you that physical layer to mock? The plumbing of the Internet that fits between the data center and all the software you write? That's what Packet does. I would consider us to be a more fundamental layer. We don't offer virtualization or run times.

There’s an article we wrote a couple months ago called We Forgot to Build a Kubernetes Service, which means like we are never ever, ever doing that, right? That's the world of our ecosystem. Who writes Kubernetes services? Google, and Red Hat, VMware and thousands of other developers, right? We don’t need to do that. There are lots of people who could live on top of us to make that ecosystem infrastructure usable. Hopefully that helps describe it.

[00:30:40] JM: The really epic light bulb for me ticked on – I mean, I hope this is correct quote. I think this was – I was watching an interview with you and you painted a picture of a world where there is a lot of fairly custom, arguably narrow types of hardware that people would want to offer access to. I think the end of one example of this is the TPU, the Tensor Processing Unit. You have people building TensorFlow models and you want to do training for those models. You want to be training. You want to do inference for those models, and machine learning we can comfortably say is here to stay. If TensorFlow is your system of choice, your programming model of choice and it's really important to your company, which is for many, many, many applications, then you're going to want hardware that's optimized for that.

You're going to want to go with a cloud provider who can give you the hardware that is optimized for that, and it's fairly easy to see a world where there are more and more kind of narrowly scoped hardware application, or hardware units, and it's like, “Well, okay. If I need that hardware unit once a year, I need to train a model every once a year or once a month.” I don't want to buy a dedicated piece of hardware.

Also, if I’m somebody who wants to be a hardware designer and I want to design some narrow piece of hardware, there’s no hardware access marketplace out there. I can't say, “I want to make my own full stack machine learning framework system. I’m going to make JeffFlow. I want to have a JPU. I want to make the Jeff Processing Unit, because I have a different perspective on how machine learning models should be built and how they should run, and the full stack thing. I want to make that JPU accessible like a cloud provider would make an EC2 instance accessible.” Where do I go to do that? Well, potentially, Packet.

[00:32:55] JM: Yeah. It’s certainly one of our goals, and I totally agree with you on this. One of the things that I think is a misunderstood maxim is hardware is a commodity. Then we walk around with this like incredibly hyper engineered piece of hardware in our pocket that we update and buy new ones every year. When you get up on stage, people talk about there's a new iPhone. They don't talk about the new app. They talk about the new bionic processor, and they talk about battery life, and they talk about speed, and they talk about what it does for you and how you have to look at it unlock it, which three years ago didn't exist, right?

Most of this is unlocked by really, really special hardware being paired with a really, really special hardware. I think those two things combined, software and hardware, really make a magical experience. Packet believes that over the next decade technology is not going to be fully innovated in JavaScript. It's going to be innovative in the things that touch people's lives, and we think voice, data, wireless, IoT, cars that drives themselves, like energy that gets figured out in a more efficient manner. These are the things that are going to touch my kid's lives in my life in a more regular basis. They already are, and most of those magical experiences, they're not random. They're just really big.

When they're really big, what do you do? You optimize the hardware for it. If you want to do a lot of Jeff processing, man, you better make a JPU.

[00:34:22] JM: You definitely do not want to do a lot of Jeff processing. I'm telling you. It's not –

[00:34:26] ZS: Random thoughts. Random dimes.

[00:34:27] JM: Not a good place to be. Not a good place to be. You don't want Jeff as a service.

[00:34:32] SZ: Well, I think what's kind of cool, and Dan Khan gave a great talk at our rainy IFEX show in Vegas in a parking lot.

[00:34:40] JM: I saw him show up there.

[00:34:41] ZS: What he did was he kind of talked about like why do things happen at certain times? Why five years ago did Kubernetes happen? But also Docker Swarm, and also Mesosphere, and also like lots –There're like 15 other container runtimes all kind of –

[00:34:57] JM: Nomad and this and that.

[00:34:58] ZS: Why do they all pop out thin air the same time? He said there's this kind of concept that you build on the shoulders of giants and then at some point it becomes possible for this to occur, and then it's like becomes impossible for it not to occur. We can kind of say that this is happening in the world of Silicon, right? ARM architecture is now really usable. X86 is really competitive. RISC-V is open source. There are architectures that work with Linux, the world of software. I don't know if you're programming what language, but most of them compile against all different architectures natively now, which it didn't used to be that way.

Now you've got a rich, rich world of software that can work against all kinds of architectures and a competitive Silicon space that will make you a special JPU. If you've got [inaudible 00:35:50] license some stuff from ARM and go out and build a processor for not – You don't have to be like a multibillion dollar company to do that. You can just say, "Hey, I've got a really particular use case and I want to optimize the crap out of it." Maybe because I need to do it in two watts and put it in somebody's pocket, or maybe because I want to embed it in the wall so that I just have to talk at the walls or something. I don't know.

That's exactly where this kind of technology is going to end up in, and I think, yeah, people are going to need an operator unless millennials suddenly want to start doing it all themselves. But I just don't think they don't want to own cars or apartment. So I'm not really sure that we're going to get them to specialty JPUs.

[00:36:30] JM: Yeah, and you want the model where a hacker in a dorm room can design a piece of hardware and click a button and have that spec for hardware sent to Shenzhen and produced and then shipped to Packet and be accessible as a service. That world is totally attainable.

[00:36:30] ZS: I think so. I think we can create a place where there could be a lot more innovation for the world of software and that it won't be just up to the biggest 5, 10 companies of the world who get to create magical experiences that combined software and hardware, but open to thousands of people. Gosh! Darn it! We probably needed it because we can't like wait for a couple of companies to innovate our way out of the big technology in the societal problems that we have in front of us. We're going to need a more diverse answer in the world of open source, which has been a huge part of my career and something that me and Nathan and the rest of the team here not only support avidly, but use avidly, right?

We think that the 25 million developers in the world today are primarily going to develop open source software or portable software. These ecosystems have allowed this, the massive amount of innovation. Well, we need to allow a substrate for that innovation to touch hardware no matter what that hardware is, because we can't just build game-changing technological experiences based upon the same stuff we have, right? We're going to have to build it off of maybe new things. Wouldn't that be cool if the pace of software and the pace of hardware can move together for most people?

[SPONSOR MESSAGE]

[INTERVIEW CONTINUED]

[00:39:31] JM: It's a great vision, pretty distinct. The playbook, it almost sounds like you could run the CDN playbook. So like Cloudflare, as I understand, Cloudflare, like I'm not sure if they have their own data centers. They might have some of their own data centers, but I believe most of what they do is strike deals with data centers throughout the world, and they basically get – They lease enough territory at each of those existing data centers where they can –

[00:40:02] ZS: Colos.

[00:40:03] JM: Are they colos? Is that what it is?

[00:40:05] ZS: Yup. Yeah.

[00:40:06] JM: They're colos. They just like grab some space at every colo that they need access to and they get this huge network. They can do whatever they want with that space. You as the developer get the sense that Cloudflare is this omnipresent behemoth and it basically is, but that doesn't mean that Cloudflare has to own tons and tons and tons of assets. A lot of it is just leases.

[00:40:28] ZS: Yeah. I mean, they've definitely taken space in colos. I don't think they build data centers, right? I think they just rent space data center as a service and put their own kind of specialty hardware in those which move packets really quick and fast and efficiently.

Yeah, I think that that's really cool. The thing is though it took them years and a lot of money to be able to put their servers, which aren't that special. They're kind of somewhat special in a hundred locations. What I think needs to happen especially as we look towards like Nathan was describing around 5G which allows you to actually put your application closer to the wireless network and people might start to have experiences that count on that or that you can really innovate on, we're not talking about people's cellphones. We're talking about cars or different worldviews, totally different applications that we've got now. Maybe the analogy is my 2007 Blackberry versus my 2019 iPhone. I wonder what 2025X interface will be for me.

People need a substrate to deliver that. The problem is you have to have – You can kind of tell a CDN. A CDN is the ultimate PaaS platform. It's like Fastly is an amazing company that basically is incredible at running varnish code, right? Cloudflare is an amazing company that's like really, really good at running HTTP and maybe a couple other things.

I think that the question is that going to be enough or is there going to be more innovation in software and people are going to need a lower-level substrate, like hardware or fundamental run time that they can use? That's our goal. What we're trying to do is say, "Hey, can we partner with a real estate of the world? Those big data center partners, tower companies that are splitting and molding business, mobile businesses or commercial real estate, like where you're probably sitting right now in a WeWork or in a Brookfield office center or at the mall. Can we partner and make it so that if you, a 50-person startup in San Francisco want to deploy a specialty type of hardware with your software in 200 places around the world, that should be like easy thing to do. Should be a couple week process. Right now it would take you years.

[00:42:33] NG: That's really also – I think that there is a – This isn't really just a theoretical concern. I mean, there are – A lot of the new applications, when you're in the cloud, not a problem when you're just throwing off megabytes or gigabytes or terabytes of data every minute. That's a much different problem when you're trying to develop and deploy an application. Not even just 200, 2,000, 20,000 locations, because it's just a different problem.

You can do the things that you do when offloading everything to somebody else and they're just going to figure it out in the background and you're not going to have to think about it. Those are real problems and they require innovative thinking around how to be able to do this sort of thing at scale both in terms of actually getting the work done but also the expertise around it.

Hardware is not the easiest thing to work with. I mean, there's a reason why we've spent and it's taken 5 years to –

[00:43:24] ZS: I mean, it had the word hard in it.

[00:43:27] NG: Exactly. Hardware is hard literally and metaphorically, and that's something that there is the practical reality of knowing what to do and how to do it and the optimizations and the realities. There's power concern, profiles, backup, redundancy. Where you want to actually –

[00:43:44] ZS: Firmware.

[00:43:45] NG: Firmware. There are so many things that go into actually being able to turn the thing on and have it be useful. Any one of those things that goes wrong means that you're not able deploy or use the application, which degrades the entire experience and your awesome, amazing cool application that does incredible things with massive amounts of data, the edge is not just sort of sitting metal bricks that don't do anything because something went wrong.

That's something that really I think gets lost in all of these, which is that there's actually a lot of expertise, operational expertise that kind of goes into making something like that a reality because the physical constraints that you have are much different than when you are going into a colo facility where there's been 20, 30, 40 years of kind of institutional knowledge around how you design and develop and deploy data centers, and now how do we compress that and multiply it by 10 or 100 fold. Those are areas that I think that being in a hardware delivery as a service kind of business which is we can put whatever hardware that is wherever you need it very, very quickly across the world and allow you to deploy your application at that speed as well. There's an enormous amount of effort that goes into actually making that a reality.

[00:44:58] JM: What is that supply chain look like? Is it the Cloudflare playbook? Are you going to these colos and Equinixs and whatever and making deals with them or where are you getting your infrastructure from?

[00:45:09] ZS: Well, you certainly make partnerships. We're definitely a buyer of colocation. We don't build data centers. We do builder our own hardware so that we can make the delivery of hardware easier, better, faster. It's small-scale. Most of the hardware supply chain right now is built for hyperscale, I think 100,000 servers a month coming in via C containers into a big data center in Ashburn.

[00:45:32] JM: Whoa!

[00:45:32] ZS: That's not what we do. I don't think that's what anybody else except for a few people in the world do.

[00:45:38] JM: Okay. Sorry. I was enchanted for a moment.

[00:45:41] ZS: Oh, it's pretty amazing when you're running one, one big gap called like EC2, right?

[00:45:48] JM: I mean, how many servers are we talking coming over on a ship every –

[00:45:52] ZS: Well, I mean for us or for somebody like –

[00:45:54] JM: For you. Yeah, for you.

[00:45:55] ZS: Yeah. I mean, thousands, right? Which is still a problem, but we do it at a very different scales. We believe that the vast majority of the world is subscale, seven or eight hyperscalers who buy 100,000 of the same thing every month. Well, every other enterprise that I've ran into or use case, they need five really specialty things in these places. Maybe times 10 places or 20 or 50 or 100, and maybe they use them for one year or maybe they use them for 10.

We think the world of infrastructure in the future has a lot more heterogeneity to it and it doesn't look homogenous. We built a supply chain and expertise around deploying small things in places, in lots of places. A lot of that has to do with our hardware innovation. Basically, we use open source hardware project called Open19 where we remove the cabling from the time of installation. We think that's super important because it is pretty hard to get consistent and gypsum service to Jakarta and have them get plugged in appropriately and serviced and whatnot. We're trying to make them feel serviceable.

Then what we do is we partner with real estates. We partner with people like SBA Communications, the third-largest tower company in the United States to build modular data centers where they deploy data centers into cities for us and we fill them full of computers. We partner with the large wreaths colocation infrastructure partners to build delivery models and say, "How can we build the packet hardware delivery platform in all of your data centers and make that something that can be more directly accessible to your customers?"

Packet is just a small business. We're 130 people. We're going against the biggest kind of companies in the planet in a very capital-intensive business. What we do is we say we're ecosystem-based. We can't do this whole thing. All we are are these plumbers that make servers go into places and turn them on and off with consistent APIs the software could work. But we have some great friends in the real estate side or great friends in the software side and together we can make a more powerful ecosystem.

There was an article we'd put out a couple of weeks ago called Ecosystems are the New Oil. None of the big problems like how a car turn left and right get solved by one company, verticalizing it and saying, "Here, we solve the autonomous car problem globally." Well, there's probably going to be a whole lot of people playing in that. Same thing like civic infrastructure or IoT. You're just not going to go into every city on the planet and say, "It's okay. We're going to run everything our way." You don't work with them. It's going to different in different cities or different places, different players, different sizes or capital. Different types of software. Different regulatory environment. I think the world is a lot more diverse and that's the angle we're going for.

[00:48:30] JM: It's an epic vision. Really interesting. I have a lot of other questions that I could ask around cloud providers, and infrastructure, and debt and stuff. But you are both musicians.

[00:48:41] ZS: Yeah.

[00:48:41] JM: Zach I know you're a very serious musician in the sense that you attended Julliard. I don't know your background as well, Nathan, but I would love to just hear from both –

[00:48:50] ZS: I play violin at Harvard. It's not really that interesting.

[00:48:53] JM: Geez!

[00:48:55] NG: That's not entirely true. I just play violin for many, many years.

[00:48:59] JM: Jacob also, Zach, your brother is a musician also, right?

[00:49:02] ZS: Yeah. He's a bassoon player. Yeah, a little bit of a concert band going on here.

[00:49:07] JM: I guess to close off, for both of you, what are the commonalities between a successful orchestra and a successful technology company?

[00:49:17] ZS: Well, I'll give my 60-second. Nathan can give his. Unfortunately, I was a bass player. So I was always in the back of the orchestra. So I'm like taking my opportunity to get the baton and sit up in front. That's been my approach.

Listen. I mean, one of the things you have to have in a great orchestra like the New York Philharmonic, you have 100 experts who are all individually amazing at what they do but are also exceptional at working together. I think a great run company is no different. You have to have leadership and a vision. Know where you're going, and then everybody has to be experts at what they do. Have their unique value to add and then be humble enough to work together.

I think being a great tech company especially when you're trying to innovate and move fast in big markets, you have to have that kind of rapport. Great leadership up at the top hopefully within each section. The analogies can go further. Then really excellent independent people who can have their own mind, have their own way of doing things, their own kind of super human strength. Very own superpowers, but they got to in the end perform together, otherwise you just end up breaking apart and you go in different directions. You can't hit the downbeat at the same time. That's the analogy I take. How about you, Nathan?

[00:50:32] NG: I would echo that. I mean, I think you hit the major points. I would also say that you have to really find people who are passionate about it. I think that when you look playing an instrument or doing bare metal hardware, this is something that Zach and I have worked together for over a decades at this point. I think 14 or 15 years. It's one of those things where we learn new things every single day. We do it because we love it.

I think surrounded ourselves with like-minded people who really care and love and don't mind showing up to work every day and in some cases doing the same thing because knowing that's going to make us better every single day and we're going to learn something every single day. That's really certainly learning an instrument, practice, practice, practice, practice, practice, and that's what it's all about. You do the same thing over and over and over again and you make little incremental progress along the way.

I think that that especially with something that apparently has been a solved problem, bare metal infrastructure, we don't do that kind of thing anymore. It’s already solved. It’s really not. There’re a lot of things that we’ve obviously touched on that a lot today, but there's a lot of innovation. There are a lot of things happening. There are a lot of demands that are requiring that a business like ours exists. I think that finding people who are passionate about that is really crucial to making it successful.

[00:51:41] JM: Guys, thanks for coming on the show. Great conversation. Really ambitious company.

[00:51:44] ZS: Thanks for having us.

[00:51:45] NG: Yeah, thank you.

[END OF INTERVIEW]

[00:51:55] JM: Embedded analytics is the way to add dashboards to your application. Are the dashboards that are in your application engaging your end-users or are they falling flat? According to analyst firm, Gartner, the UX of embedded analytics has a direct impact on how end-users perceive your application. Fortunately, you don't have to be a UIUX designer to build impressive dashboards and reports. Logi Analytics has come up with six steps that will transform the user experience of your embedded analytics.

Logi Analytics is the leading development platform for embedded dashboards and reports, and unlike other solutions, Logi gives you complete control to create your own unique analytics experience. Visit logianalytics.com/sedaily to access six basic principles that will transform your dashboards. That's logianalytics.com/sedaily. That L-O-G-Ianalytics.com/sedaily.

[END]