EPISODE 982

[INTRODUCTION]

[00:00:00] JM: Edge computing is the usage of servers that are geographically close to the client device. The first common use case for edge computing was the CDN, the content delivery network. A content delivery network placed media files such as images and videos on multiple servers throughout the world. These are big files and they take lots of bandwidth to transfer. By placing them at CDNs, the files will be closer to any user throughout the world. The early use case of the CDN for edge computing was pretty much about storing large media files, but the vast majority of compute still took place at the central application servers.

Overtime, users have required faster and faster application experiences. Today, an increasing amount of compute has been moved to the edge in addition to the existing storage applications. More user data is being cached at the edge to make for quicker transactional processing. Machine learning model training and hosting at the edge makes for a faster, more responsive machine learning feedback loop.

Jaromir Coufal is an engineer with Red Hat. He joins the show to talk about modern applications of edge computing and how the demand for edge computing is creating a market opportunity for companies that have lots of servers at the edge, such as telecoms. These telecoms can repurpose their widely distributed telecom infrastructure as edge servers that they can sell usage on. This creates a business opportunity for Red Hat to provide software infrastructure to the telecoms.

It's an interesting discussion of business opportunities and engineering challenges that take place at the edge. As we head into 2020, we are hiring for Software Engineering Daily, a couple part-time roles. We're hiring a writer and we're hiring an operations lead. The writer will help us create content for Software Engineering Daily. It will be writing about computer science and software subjects, and the operations lead will be somebody who helps us run our business more effectively. This will involve working with spreadsheets and various business productivity tools. If you are interested in either of those part-time roles, send me an email, jeff@softwareengineeringdaily.com.

[SPONSOR MESSAGE]

[00:02:27] JM: Today's episode of Software Engineering Daily is sponsored by Datadog, a monitoring platform for cloud scale infrastructure and applications. Datadog provides dashboarding, alerting, application performance monitoring and log management in one tightly integrated platform so that you can get end-to-end visibility quickly, and it integrates seamlessly with AWS so you can start monitoring EC2, RDS, ECS and all your other AWS services in minutes. Visualize key metrics, set alerts to identify anomalies and collaborate with your team to troubleshoot and fix issues fast. Try it yourself by starting a free 14-day trial today.

Listeners of this podcast will also receive a free Datadog t-shirt. Go to softwareengineeringdaily.com/datadog to get that fuzzy, comfortable t-shirt. That's softwareengineeringdaily.com/datadog.

[INTERVIEW]

[00:03:32] JM: Jaromir Coufal, welcome to Software Engineering Daily.

[00:03:34] JC: Thank you very much. Good to be here.

[00:03:36] JM: We're going to talk about edge computing today, and particularly Red Hat's strategy around that. Give me two recent applications that you have seen of edge computing.

[00:03:45] JC: Two very often reasons why the applications want to run at the edge is latency, which means you want to have very fast request response. The second one is reducing bandwidth over the network. Those are the very two very often mentioned ones. They are way too many other reasons why to do that, and for me these two reasons are mainly the consequences of what I want to achieve. If I want to improve user experience, those are probably the two – The latency is probably the first one. If I want to reduce cost, probably reducing the bandwidth is another one.

But then there are other reasons, for example, risk factors. I want to improve resilient. Therefore, I want to run my applications at the edge and have them running in very completely isolated environment. There are really many reasons, but the two most often mentioned ones is definitely latency and bandwidth.

[00:04:41] JM: The type of edge computing we've had for a pretty long time is CDN infrastructure. CDN infrastructure is fairly basic. It's if I make a request for an image, for example, that image is going to get cached at the edge and then subsequent requests to that resource will be sent to the edge, because the edge is going to be closer to the user. The edge might be a content delivery network that is pushing out that content to a lot of different locations so that a user in Uzbekistan has rapid access, a user in Texas has rapid access. The kinds of edge computing that we want to do today is very different than that type of simple request response for an image, for example. What kinds of computation do we want to do at the edge today?

[00:05:27] JC: I don't want to repeat myself, but it's always it depends on the use case. There are many industries which are doing edge computing for different reasons. Let's take for example industrial IoT in the production lines where they are having multiple sensors in their production lines to control quality production, for example, of their product.

These sensors are generating a lot of data, and this data is being analyzed and it's being corresponded, is the product quality okay? Can I continue in the production line or do I need to interrupt an adjust the process? This is kind of application which needs very fast rapid response. You cannot rely on sending that over the network to the centralized location. Do the processing there and then get the response back. This is very often reason why you want to push the edge computing to the location of the IoT industrial.

But there are also reasons why do you want to centralize. You don't want to do everything only at the edge. There are reasons why you want to do a centralization. Imagine I have the industrial plants and I have them over the country, but I need to train my machine learning models from different examples and I want to gather those examples from different locations. Therefore, I need to centralize them somewhere than do training model very likely in the

centralized location and then push out only the trained models to do the fast decision making at the edge. This is one of the examples.

[00:06:52] JM: Right. That architecture makes a lot of sense, because in order to train the machine learning models, you need all the data centralized in a particular place or you need some large subset of the data, or you need the new training examples, whatever. The data to train the models takes a lot less space than the actual models that can make a decision that can improve application infrastructure. We need to start deploying these models to the edge.

Now if we talk about the types of CDN infrastructure that has been there for a long time where you're caching an image and then you can request the image at the CDN layer, if we're talking about hosting machine learning models versus hosting basic images, do we need different infrastructure to run those machine learning models instead of simple CDN type of infrastructure?

[00:07:45] JC: Absolutely. We do, and it depends on the context. What kind of application I want to run there for the machine learning, very often, or for example video processing of similar request. We might need fast data processing. We need real-time kernels. We need GPUs very often to speed up the processing and reduce the cost of the processing power.

The type of the hardware which is enabling the edge use cases is specific for the application which needs to run on top of that. The CDN is very simple use case as you mentioned. You don't really need much of the processing, specific processing. But if you need to run, for example, containerized network function, it does very high requirements on the fast data path and networking, or it needs to access to real-time kernel, which typical CDN networks don't need.

[00:08:35] JM: Already talking about machine learning stuff, I've also heard of widespread edge computing use cases in the telecom industry. Can you tell me about those applications?

[00:08:47] JC: As for the telco, they want to do edge computing for multiple reasons. I would separate it into two categories. First one is to run their own radio access networks, and for that, that is the containerization of their network functions. As you know, there was a progress of

running the network functions on the bare metal itself with specific hardware. Then it moved to VMs and now it's moving through the containers. Here at KubeCon we actually saw a demo of the proof of concept how to do the 5G fully running on Kubernetes.

That is one of the kind of the application because at the antenna level, you need to have the processing as close to the signal as possible so that you, again, reduce the latency and you don't have to push all the data through the network to reduce the bandwidth as well. That is the main requirement for 5G or main features of the 5G.

The second use case for the telcos is to monetize your infrastructure which they are building for the 5G. That is to provide mobile edge computing platform for all the independent service vendors or other enterprises running their applications. I can be an enterprise which has branch offices, and in order to get access from those branch offices to fast processing, I might want to run my applications closer to those branches, not necessarily centralized in my data center, which might be private cloud one national data center, but I might take an advantage of my telco provider to run the application closer to my branch office.

Another very common use case is gaming or augmented reality where you take advantage of being as close to the end user with the cellphone or the smart devices possible so that they have the good user experience and then they don't experience the lags.

[00:10:42] JM: I understand you correctly. A telecom like Verizon, they need to build essentially data center infrastructure or telecom data center infrastructure to support 5G for their basic cellular customers, and in order to get additional value out of that infrastructure, they are adding the capability to essentially lease out those resources to enterprises that might need similar functionality to that same 5G infrastructure. Basically, they're already building data centers and they're like, "Well, we might as well reuse this for edge computing."

[00:11:23] JC: That's exactly right.

[00:11:25] JM: Amazing! As Red Hat, you basically have an opportunity to help a telecom like Verizon that has not historically thought of itself as a cloud provider, to the extent that I understand it, become a cloud provider essentially.

[00:11:45] JC: Yes. I would not say necessarily cloud provider, but mobile edge computing platform provider, absolutely. Any of these telcos or service providers in general. It doesn't have to be telco only. It can be internet service provider as well. Think of Comcast or others. That is to reuse their infrastructure which they need to build to support 5G. We can think also of the other use cases when they want to place a device to the end user premise itself.

You can think of smart stadiums where you have broadcasting of different games, for example. You need a server which needs to run their – You're going to need high bandwidth. You need to broadcast it. What Verizon or other telcos do, they place the server there at the stadium at the end user premise and they use benefits of their infrastructure behind, or as an end consumer, for example, I'm a Comcast user, I have my set-top box at home. Currently, the set-top box is being used for very specific services, for example, to get my TV streaming on my screen.

But in order to reuse this device to be more generic, we start talking about the universal customer premise equipment to turn these very specific use-case-oriented devices into more generic computing platform to turn on your home into smart home and connect the IoT devices to it and provide different kinds of applications running there.

[00:13:14] JM: Okay. That's a very different kind of edge computing.

[00:13:16] JC: It's very different kind of edge computing, but it is still edge computing.

[00:13:19] JM: Okay. What role would Red Hat have in that kind of application? The like, "I'm Comcast. I've given you a dumb set-top box and now we want to turn it into like a smart connected home thing."

[00:13:33] JC: Comcast is still owner of that server or of that device which is running at your home. They need to run some operating system there. They need to run some platform, which is enabling to run applications on top of that. Guess what? The applications are very often containers. We can think of, for example, take it to the extreme, Kubernetes in a box.

[00:13:53] JM: Wait. Real quick. The applications running on my dumb Comcast – Sorry to call Comcast dumb. I'm calling the box dumb. Not calling Comcast dumb. Those applications that are running on the set-top box, they're running in containers.

[00:14:08] JC: They might. Some of them might be running in VMs depending on the vendor.

[00:14:11] JM: Interesting. Okay. Sorry. Go. Continue.

[00:14:13] JC: Then you can think of the device to have a marketplace of different applications which can run at the device. It doesn't have to be only TV streaming. You can now have audio streaming or you can now connect your light bulbs and have the light bulb manager running on top of that box. Be the really generic computing server which can run any kind of applications. Then the internet service provider can give you, "This is the marketplace of apps. These are the vendors who are contributing there." You can deploy your app or run your app there.

[00:14:44] JM: Is that to say that Kubernetes would be useful as essentially a consumer operating system in that environment or maybe OpenShift if the operating system or is it just a Linux, just a single Linux node? You don't actually need a distributed system, or you need – Well, it's a distributed sense that you have containers running on the same node, but it's just a single node.

[00:15:10] JC: It is just a single node in our homes. In our home, you probably don't want to have free servers to have physical HA. You have one single server, which is the transform set-top box which is a single box running operating system, which is always the core of everything you need to enable the hardware in some way. Then you need some platform or you need something which is orchestrating the workloads on top of that. Depending then on the level of orchestration, autohealing or other things you need there, then you need different set of services from CNCF or Kubernetes to run on top.

[SPONSOR MESSAGE]

[00:15:56] JM: When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by

engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

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

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

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

[INTERVIEW CONTINUED]

[00:17:44] JM: Let's go back to the telecom premise. What role does Red Hat play in that business expansion opportunity for a telecom? If I'm a telecom and I say, "I'm building out this 5G infrastructure. I want to be able to lease out my excess capacity to businesses or I want to be able to offer services on top of that excess capacity." What kinds of arrangement am I making with Red Hat? How is Red Hat helping me accomplish that goal?

[00:18:24] JC: Red Hat will support the telecom industries to build the infrastructure and put the platform on top of that. The telco builds the infrastructure from the hardware. They deploy very often operating system running on Linux [inaudible 00:18:39] as the top enterprise graded software operating system. Then on top of that, they need some orchestration platforms so they can then provide the space or the hardware, at least the hardware, to the independent service

vendors. Running OpenShift on top of their hardware at the edge locations is providing them this platform to enable to spin up containers from different providers.

[00:19:12] JM: I'm not sure I totally understood that. Different providers – What are the different providers you're talking about?

[00:19:18] JC: By providers, I mean application providers. Be it my enterprise. I can be my own application provider. So I can have business-critical applications running on top of that, or I can be gaming industry vendor and I can run my gaming applications also there. Those providers, I'm talking about the application providers.

[00:19:37] JM: We're really talking here about Red Hat is essentially giving the telecoms their domain- specific infrastructure needed to become an edge competing provider for the gaming companies or the businesses that might need access to edge computing infrastructure.

[00:19:56] JC: That's correct.

[00:19:57] JM: Okay. Now, I understand that OpenShift plays a role here, because OpenShift is sort of like a usability layer over Kubernetes. We've done plenty of shows about OpenShift. We've done plenty of shows about Kubernetes platforms. You've been at Red Hat for 7-1/2 years, and during that time you saw the rise of OpenStack, which was an ecosystem that looked in a lot of ways like the Kubernetes ecosystem. As I understand, OpenStack remains popular among telecoms. Why is that? What purpose does OpenStack serve for telecoms?

[00:20:35] JC: It's a great question, and in fact most of the time in Red Hat I was involved in OpenStack. I was product manager and engineer also on the deployment and management of open stack. What we've seen is the transformation of the physical hardware applications into the virtual network functions. This got adapted across the telco industry in a very large-scale. It is still being adapted. Many telcos are just applying these virtual network functions across their infrastructure. Now they start pushing it out to the edge also.

We see transformation of their vRAN sites into the virtual network functions and deploying them, and OpenStack is basically enabling you to have the infrastructure as a service, the ability to

virtualize your infrastructure and run virtual machines on top of that where the network functions are running, eventually, and that is also supported in a large-scale, which we are talking about.

Telco, you can think for example in the United States, we can be talking about thousands of thousands of sites of cellular networks, cellular towers, and each tower, if they need to have a site nearby, we talk about massive scale, and open stack allows you to scale your infrastructure in a virtualized fashion and spin up the virtual machines on top of that.

[00:22:05] JM: I don't know if this is in your domain at all, but do you know what is the relationship between the cellular towers and the telecom data centers? Do the telecom data centers need to be close the cellular towers?

[00:22:22] JC: It is not exactly my domain, but the telco providers run multiple applications or multiple tier applications. At the cellular tower, you need to have certain application which is doing some processing of some packets. But then you need to have some core of that application of the network function which does not need to reside at the cellular tower itself, but the core needs to reside either closer to the edge or it can resides a little bit at the regional edge. It depends. Usually it is at the access point. For more details, I really cannot talk to that. It's way too complex for me.

[00:23:00] JM: No problem. It's definitely not telecom engineering daily. It's very much software engineering daily, but is there some latency sensitivity between – I guess it's no surprise. There's some latency sensitivity between the data getting routed between the cellular tower and wherever the telecom data center is.

[00:23:21] JC: That's correct.

[00:23:23] JM: Going back to the virtualization side of things. The OpenStack, I guess pre-OpenStack, these telecoms, they had data centers, but they were just running less efficiently managed infrastructure, and OpenStack gave them a better open source management platform. I guess, what were they running before then, like VMware? What was the story before OpenStack?

[00:23:51] JC: Very often, these telcos are running very specific hardware switches or other specific devices. You have basically a vendor lock-in in that context because you are locked into that hardware vendor, whatever that hardware was doing as a function. In order to get out of this old infrastructure, you put a generic computing server there, which then you need to somehow enable to run the virtual machines on top of that. They are replacing their hardware appliances, which were very specific function-driven and they are making it more generic computing platform.

[00:24:30] JM: Okay. That was the big deal with OpenStack. Then why didn't OpenStack succeed as a more general computing platform? Because I think a lot of people had the vision that this would be a general computing platform. I know there were a lot of enterprises that deployed like OpenStack in non-telco circumstances where now they're kind of thinking, "Oh, we don't really need open stack, or we want to deprecate OpenStack. We eventually want to move to Kubernetes and replace OpenStack." Why was OpenStack useful for the telecoms but less useful long-term for other domains?

[00:25:14] JC: I wouldn't say it's not useful for the other domains. It heavily depends on the applications you're running on top and how fast you can actually move in the adaption curve. Every technology has its adaption curve. The enterprises are very often better in driving the adaption of their applications or of the technology in a little bit faster manner.

With the enterprises, we still see a growth of enterprises adapting OpenStack if their applications are written in the virtual machines. As you know, the trend is to move into containers because of the microservice infrastructure, cloud native and all the other reasons why we are doing Kubernetes. Therefore, the enterprises are looking into migration into the new technology world.

With the telco industry, they have heavy investment into the infrastructure and the VNF or the virtual network function vendors, application vendors, are a little bit slower in moving into the containerized microservice world. Once those application, so those virtual network functions were running on bare metal, when you want to move them into virtual machines, they did not really do very good job in decoupling that application into multiple smaller services. It is still very big fat VM which is running there.

Now they are looking into completely reconstructing that application, that VNF, into microservice-based architecture, which takes time. Currently, the telcos, they are still in the business. They need to run their business. So they invested a lot into the infrastructure to run the VNFs in the virtual machines. Once the technology with the containers is ready, I believe in few years we will see shift of the telcos moving towards container native functions as well, and therefore Kubernetes.

[00:27:10] JM: Cool. You're largely in charge of thinking about edge computing at a strategic level for Red Hat, and the purpose that you've just described to me, for example, helping telcos build out data center infrastructure that is easier to manipulate and easier to potentially build their own businesses on top of. That makes complete sense to me.

What's difficult about that is as we're talking through the infrastructure at a telco data center, it's non-uniform, very non-uniform, right? You've probably got plenty of telcos that are still on these old hardware switches, plenty that are on various versions of OpenStack. Some that are probably more forward-looking and they've got – I don't know. They got a hybrid cloud or something. In any case, it's not homogenous. It's going to vary from customer to customer.

How do you, as Red Hat, devise a strategy for how to work with these telecoms when it's always going to be so variable. It almost seems like the strategy has to be we're going to give you a bunch of solutions architects from Red Hat. They're going to figure out your strategy. It's going take a while. That's our strategy.

[00:28:39] JC: It's great. As you know, Red Hat open hybrid vision is to run any kind of application, be it an application on bare metal. Be it an application to virtual machine or in the container. It doesn't matter what kind. If you deliver the application on any kind of footprint. Again, we are talking about the physical hardware itself. It's going to be virtualized. It's going to be public cloud or private cloud.

With the edge, we are extending it that you can run this at any location, be it your core data center up to the very edge. Our vision is to provide you a single uniform platform where you can achieve all of these. We started some projects in order to achieve that. To achieve the hybrid

cloud vision, you know that OpenShift can run on any kind of those applications. It's going to run on bare metal. I can run in public cloud or private cloud or in virtual environments.

If you need to combine the containers and VM's, we are looking, for example, into projects like KubeVirt where you can use OpenShift APIs or Kubernetes APIs to spin up your VMs as well. You will have a single uniform platform where you natively run containers, but if you're in the hybrid world where you still have some legacy VMs or applications, you can use the same set of APIs to run your VM's on the KVM underneath.

[00:30:14] JM: Is that a new project? KubeVirt?

[00:30:15] JC: It's not a very new project. It's been around for a little while, but it is a part of the CNCF.

[00:30:22] JM: Cool. Coming back to just because you have adjacent expertise in telecom, you hear this term 5G. I know that 5G is like a bundle of technologies. It's no one thing. Do you know what technologies go into 5G? What does that even mean?

[00:30:41] JC: Well, the 5G as I know is an evolution of the network, cellular network itself.

[00:30:48] JM: That's descriptive.

[00:30:49] JC: That's very descriptive, right? The main characteristics there are we need to enable – We need to support the amount of data which is being generated in this fast-paced world.

[00:31:02] JM: Totally.

[00:31:03] JC: There are more and more smart devices, which are generating more and more data, and we have more sensors which are generating more data. We're getting more insights and the 5G network will allow you to handle all this amount of data, because you will have larger bandwidth, which it can go through and you'll be able to support new use cases.

An example of that is connected vehicles, differential healthcare equipments, for example, which are getting smarter and smarter in terms of monitoring –

[00:31:31] JM: And data intensive.

[00:31:33] JC: Data intensive, definitely. Augmented reality, we mentioned that already.

[00:31:38] JM: Another super high-bandwidth thing.

[00:31:39] JC: It's another super high-bandwidth thing. Really, the bandwidth demand and the latency is two major factors driving this.

[00:31:47] JM: Okay. We're at KubCon. What are the developments in the Kubernetes ecosystem that are improving edge computing?

[00:31:56] JC: I would say they are fairly slow, because every technology goes through different phases and it faces different challenges. Edge computing is very hype and very intensively mentioned area. However, the platform itself is still facing several challenges how to easily run on bare metal, for example, or how to easily operate that infrastructure in an automated fashion. We see improvements in different areas which are helping the edge computing, but there can be definitely more.

Good examples of that is when you're doing edge computing, we are talking about scale. Then there is a question if you want to have a single Kubernetes cluster, which is scaling to thousands of nodes, or if you deploy multiple small lightweight clusters. If you do that, then you are having different kinds of challenges because you need to, A, manage those clusters, and we are talking about thousands, tens of thousands, easily hundreds of thousands of clusters. Then you need to talk about the size of the footprint it runs, because you need to be very space efficient to give enough space for the applications which are running on top of that.

The next one is if you have multiple independent isolated devices or servers, not devices, but servers, which are running independent Kubernetes clusters, how do you make them collaborate together so that they can provide the computing behind that so that the application

can be distributed across those clusters and can communicate with each other. You need to create mesh between those clusters.

[00:33:43] JM: Or Federation, right?

[00:33:44] JC: Or federation. All of these things are out there, and we see different challenges across different projects. Storage is another one. How you distribute the storage out to the edge across multiple clusters. How you do replication across those clusters. All of this – And every single project needs to evolve. We can do edge computing today with Kubernetes, different use cases, but it can get much better.

[SPONSOR MESSAGE]

[00:34:17] JM: Over the last few months, I've started hearing about Retool. Every business needs internal tools, but if we're being honest, I don't know of many engineers who really enjoy building internal tools. It can be hard to get engineering resources to build back-office applications and it's definitely hard to get engineers excited about maintaining those back-office applications. Companies like a Doordash, and Brex, and Amazon use Retool to build custom internal tools faster.

The idea is that internal tools mostly look the same. They're made out of tables, and dropdowns, and buttons, and text inputs. Retool gives you a drag-and-drop interface so engineers can build these internal UIs in hours, not days, and they can spend more time building features that customers will see. Retool connects to any database and API. For example, if you are pulling data from Postgres, you just write a SQL query. You drag a table on to the canvas.

If you want to try out Retool, you can go to retool.com/sedaily. That's R-E-T-O-O-L.com/sedaily, and you can even host Retool on-premise if you want to keep it ultra-secure. I've heard a lot of good things about Retool from engineers who I respect. So check it out at retool.com/sedaily.

[INTERVIEW CONTINUED]

[00:35:53] JM: The cluster “meshing” versus the federation, this, as I understand, is problem of you’re a big enterprise, you've got thousands of engineers and different sets of engineers, different teams are spinning up different Kubernetes clusters. You'd like to have these clusters communicating with one another. You don't know if you should have a higher level Kubernetes cluster that federateds functionality to those other clusters or just have these Kubernetes clusters being communicating flatly. That's the broad architectural difference between the meshing and the federation, right?

[00:36:42] JC: That’s the higher-level idea. Yes. It depends what phase of evolution we are at. What we are trying to achieve eventually is resilient independent systems. It is difficult to get there, but if you introduce a single point of failure somewhere, which can in this case be the federated high-level cluster, then you will not have access into all the other clusters. You have single point of failure.

If you have those clusters being completely independent, interconnected, talking to each other, then we are building truly distributed system and that's eventually where we want to go, but it takes time in order to build a technology to support that.

[00:37:29] JM: Is that to say that the federation model is not appealing to you or you’re just saying that if you architect it wrong, the federating higher-level Kubernetes cluster could be a single point of failure.

[00:37:41] JC: I'm just saying it can be a single point of failure, but there are different applications which might benefit of that and different kind of use cases, which it might be too big of a risk for it.

[00:37:50] JM: OpenShift has become a key product for Red Hat, and OpenShift is this management platform on top Kubernetes. How does OpenShift compare to the Kubernetes management systems on the cloud providers? You’ve got Amazon EKS, AKS for Microsoft, Google Kubernetes Engine. How does OpenShift differ in product design?

[00:38:13] JC: OpenShift or overall Red Hat’s vision is, again, the open hybrid cloud and no vendor lock-in. All those public cloud providers which are running Kubernetes on top require you

to stay with that public cloud provider and you've really don't have flexibility to migrate your workloads across different providers. You are locked in.

For example, if you build your business on top of Google, or Amazon, or Azure, they increase your price. You don't have any other choice than just go with them or completely re-architect your business applications to run on different service provider. What OpenShift brings you is if you deploy OpenShift on top of Google, Azure, on top of AWS or even on bare metal, you have the same set of APIs, you have the same set of functionality which is running on top and you can easily migrate your workloads or you can easily move across your providers as you need to.

[00:39:17] JM: This is confusing to me , because I thought Kubernetes was the lock-in free open source platform for containerized computing. I thought Kubernetes was the portable thing.

[00:39:32] JC: Correct, but different cloud providers are enabling different services on top of Kubernetes, which is basically making you the vendor lock-in. It's not the Kubernetes itself. Those are the services which are running alongside Kubernetes.

[00:39:46] JM: IBM acquired Red Hat recently. What lessons have you learned about how a large technology acquisition works?

[00:39:53] JC: Honestly, for us, at least from my perspective, this being fairly transparent and fairly well- working for us because IBM really led Red Hat to be an independent company. We are not being pushed to embrace or do only IBM products or IBM synergy projects, but we are still partnering with larger ecosystem system and IBM is enabling us to grow faster, scale faster. All the core values of Red Hat which we have had over the time embracing of open sores, the culture at Red Hat and the openness, which is at Red Hat in general. That is still being embraced and I am actually very surprised that this didn't go south.

[00:40:40] JM: Much of your background is in user experience design. How does that apply to building infrastructure products?

[00:40:48] JC: I would say in every single product, management in general, you need to think of usability and ask questions why. Why would the consumer want to do it? What is the path they

want to go from point A to point B? Building the infrastructure is very the same. You ask about the use case. Why I want to do what I am doing? Then you need to understand how it applies and then you can translate it into the infrastructure. This is the architecture which supports my use case.

I'm not building it from the technology bottom-up, but I'm going from the top, the use case itself, down to what technology can support that. That's probably what got me the most experience from the user experience design to focus on the user itself. What is the motivation? The why, and then translate it into the set of requirements and then build the solution from that.

[00:41:53] JM: Let's say I am a telecom. I feel like I am ready to get my edge computing going. I am looking for a technology company they can help me bring edge computing to my telecom infrastructure. I come to you. What's your strategy for me? What is the set of steps that I'm going to take to deploy edge computing infrastructure to my telecom infrastructure?

[00:42:23] JC: The first question we will always ask is why you want to do it. What is your use case? What drives you? What drives this direction? Is your direction to virtualize your obsolete hardware applications and move them to VNFs, or do you want to investigate the container native functions, or do you want to enable your infrastructure to lease it out for mobile edge computing use cases? Because we need to understand what drives your motivation. Many customers come say I want A, and in fact what they want is B because that's how it works.

First, that's what we do. We ask why. Then we look into very important partner relationships we have in order to support that solution which needs to be build, because no single vendor can provide you end-to-end edge computing solution. You need to have hardware vendors. You need to have the software vendors. You need to have maybe networking SDN vendors, storage vendors which might need to support you, some consulting services, maybe managed services.

We look at the partner echo system. We work with the customer also because very often they have their own tenders for different kind of vendors which they are allowing. We look into with whom we can collaborate. Then as soon as possible, we go to POCs to prove how the technology can support this kind of use case and evaluate if there are some gaps from the minimum viable product or if we can go full production. Then it is the rollout. To start small, you

start typically with couple of sites or small amount of sites and then you start growing as so you are comfortable that your system can support or your operations team also can support eventually.

[00:44:25] JM: How long does that take typically? How long does an end-to-end rollout take?

[00:44:30] JC: I would say it never ends, because the technology always evolves. It is a lifecycle. We can see many companies going from point A, just submitting RFPs up to the production and it varies definitely company by company, but it can take easily from half a year up to a year or two. It depends on really how complex that infrastructure or that architecture is. It's going to take a long time just to do the decision-making process. It can easily take multiple months just to decide what kind of technology to use. What kind of vendors we want to use. Then the rollout itself, it can be fairly fast. It can be very aggressive. It can be a matter of weeks easily. But the preparation before is very crucial and it usually takes most of the time.

[00:45:27] JM: The telecom application, it's going to be very useful. It is very useful, I'm sure. It's not as thrilling as the machine learning training and federated deployment system. Have you seen anybody actually training their models in a centralized place and then deploying them to the edge, or do you think this is more of like a "future thing"?

[00:45:54] JC: It's definitely not a future thing.

[00:45:56] JM: You're seeing it today.

[00:45:57] JC: We are seeing it today all over the industries. Definitely, industrial is very important driving that one. Yeah, I would say industrial is very good example of that to just minimize the amount of waste which can happen if you make a production mistake somewhere along the process until you catch the problem. To minimize the time in between, having the machine learning mechanism with different sensors monitoring the state of the product along the way, it's very important and we see it in production already.

[00:46:32] JM: This is like in manufacturing or is it predictive maintenance or what kind of –

[00:46:38] JC: It's mainly in manufacturing. Predictive maintenance is also another use case. I personally haven't seen any production predictive use case itself, but we see it coming already in the automotive industry that you see more and more smart vehicles advising you what kind of components need to be maintained based on learning how you are using that vehicle, based on the other factors.

[00:47:04] JM: The smart vehicle application, are you talking there about like I've got a vehicle and models are periodically getting updated on the vehicle that are telling me like my brakes need to be replaced or something? Can you give me a clear application, where have you seen this edge computing machine learning model deployment? What is a clear application that you've actually seen?

[00:47:35] JC: An example with the connected vehicles, you can have different kinds of applications running there. Apart from the media infotainment system, which is being also kind of edge computing. The autonomous driving is a very good example, because you have multiple sensors which are sending you the information from the environment around. Then you have trained models in the car which are responding to those inputs and you need to do very fast decision-making.

This is a very good example where you need to have a trained model which is not being trained exactly on the car itself, but you need to have a trained model. This trained model needs to be updated overtime because we are getting more and more information. We are getting those application smarter. They are being pushed to the car, but the decision-making is happening on the car itself.

As for the predictive maintenance, it is the – For my believe, I honestly haven't seen it in in the real-life, but the model itself is getting information from the different sensors, in the brakes, in your driving habits and it is basically building the prediction on this might be running towards the end of life. It needs to be looked into. It does not have to be, but it's the risk potential.

[00:48:55] JM: Cool. Last question. Actually, no. I have a couple more questions. We touched on this idea of the "edge" in the home, the idea of my set-top box becoming smarter and containerized application ecosystem developing in my set-top box. What other edge computing

applications that are not owned by like the major tech companies, like Apple, Google, whatever, Facebook? What are the more open opportunities for edge computing in the home?

[00:49:41] JC: Apart from the smart home where you can actually regulate your temperature, your lights and other things, smart things at your home, we can think of the edge computing in the home also in a way that there are already some applications and systems which allow your own computers or servers to lease out your space for storage, for example, which you can monetize on. You can be –

[00:50:13] JM: Wait, like cryptocurrency stuff, right?

[00:50:16] JC: Like a cryptocurrency stuff, but it's not about the cryptocurrency to do the computing on your computer, but basically you're leasing your storage for someone else. But it's basically yes, similar model.

[00:50:27] JM: Sure. Okay. All right. We're talking about a very distant future stuff. Okay. So maybe that's how edge computing changes my life in 20 years or in the distant fictional future. How will edge computing change my life in the nearer term? Let's say 3 to 5 years.

[00:50:41] JC: I would say it is in a multiple ways. As we mentioned, there are more and more smart things coming. For example, the healthcare and wellness industry is really booming. Gathering information about your body, understanding your body, how it works. If something goes wrong, for example, if you have a heart attack or something, your smartwatch can notify 911 right away so you can get to the treatment as soon as possible.

I would say overall it is improvement of our well-being that's from this perspective. Also, the entertainment industry is another one. Being able to play different kinds of games, being connected more and more, being able to augment the reality. For example, in retail, or if I am designing my own house, be able to design it and see it right away without actually buying the product itself. See it on the on the place where it's supposed to be. That's another application. Autonomous car driving is also booming trend. All sorts of different use cases just to make your life easier in the end. I believe that's what edge computing is helping with.

[00:52:00] JM: Jaromir , thanks for coming on the show.

[00:52:02] JC: Thank you very much.

[END OF INTERVIEW]

[00:52:12] JM:

As a programmer, you think an object. With MongoDB, so does your database. MongoDB is the most popular document-based database built for modern application developers and the cloud area. Millions of developers use MongoDB to power the world's most innovative products and services, from crypto currency, to online gaming, IoT and more. Try Mongo DB today with Atlas, the global cloud database service that runs on AWS, Azure and Google Cloud. Configure, deploy and connect to your database in just a few minutes. Check it out at mongodb.com/atlas. That's mongodb.com/atlas.

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

[END]