# EPISODE 440

[INTRODUCTION]

**[0:00:00.5] JM:** How would you build a system for indexing and monitoring the entire internet? Start by breaking up the internet into IP address ranges. Give each of those address ranges to servers distributed around the world. On each of those servers, iterate through your list of IP addresses sending packets to them. Depending on what sorts of packets those IP addresses respond to and what those responses are, you can build a map of the devices on the internet, what is running on those devices and what they respond to.

Qadium is a company that indexes and monitors devices on the internet to help organizations understand the devices that are within their corporate networks. If you're a large corporation, Qadium can probably do a better job of figuring out your internet footprint than you can. Matt Kraning is the CTO of Qadium. In today's show, he describes the process by which Qadium maps thee internet.

Matt used to work on data infrastructure at DARPA, and he has deployed Hadoop in Afghanistan. So the infrastructure of Qadium seems relatively manageable to the environment he was in in Afghanistan when apparently there was incidents where a bullet head hit a disk and messed up the Hadoop cluster a little bit.

So our conversations in this episode span from talking about Storm and Hadoop to more modern infrastructure, like a Google BigQuery, Bigtable and data flow. Matt gives us a great picture for how Qadium works. I had a great time in this episode. I think you're going to really like it. We'll certainly have Matt on the future. He discussed a little bit about moving some of Qadium's infrastructure to Lambda, AWS Lambda or Google Cloud functions, the server list functions. But there is plenty of infrastructure in his current infrastructure worth discussing and I think you'll like this episode.

[SPONSOR MESSAGE]

[INTERVIEW]

**[0:03:40.5] JM:** Matt Kraning is the CTO at Qadium. Matt, welcome to Software Engineering Daily.

**[0:03:44.4] MK:** Thank you so much for having me. I'm very excited to be here.

**[0:03:47.5] JM:** Today, we'll talk about the challenges that Qadium is solving. Let's start with network security. What are the most common misconceptions about how to secure a large computer network?

**[0:04:02.4] MK:** I think the biggest one starts with the fact that for pretty much all large organizations. Foundationally, they do not even know the full extent of their own computer networks, at least in a centralized sense. It's very hard to impossible to protect that which you don't know about. That really is the foundation upon which Qadium is based is you need to know everything that is you in order to be able to protect it.

A lot of trends in modern computing, specifically around Cloud computing and the internet of things, mean that kind of be a traditional enclave security models of the 1980s and even into the 1990s that have not changed much today, are not convergent with the way that modern enterprises run. Therefore, are incapable of fully securing really any modern large-scale enterprise.

**[0:04:59.1] JM:** How much visibility do corporations have into their own networks? If they're not seeing everything, what are they seeing?

**[0:05:06.6] MK:** It definitely varies. Our experience with our customers, which are some of the largest organizations in the world, it can vary from 80% feasibility over their assets too, in some cases, actually under 10%. By this, what we mean is that they typically will have local people in their IT staff that will have visibility for most parts of their network. But it's not being effectively aggregated into one place.

More importantly, the estimates that they themselves have of their own visibility, very widely with some organizations being very confident, they have everything handled. When in fact, they can categorically do not based on our unique observations on the internet, to others that know they have a problem, but really until we invented the first solution to it, did not know how to actually implement a real solution other than stacking bodies against it and just trying a lot of consulting efforts.

**[0:06:02.8] JM:** So explain what your company Qadium does.

**[0:06:05.4] MK:** Sure. So at a high-level, what Qadium operates is an internet intelligence platform. What this platform does is discovery and monitoring of all assets on the public internet before the purpose of alerting organizations to the security risk that they pose. So going down the level, what we do is among other things, actively communicate with every IP address in the world. We can do this faster than ever hour now.

What we do is we gather lots of information about all the devices that respond to signals that we send. What this boils down to in a lot of cases is IOT devices, servers and infrastructure, we in almost all cases will not see personal devices, such as phones or laptops. Those would be behind in that and not respond to an active signal. But all the parts of a corporate network that they care about will respond and they will respond a lot more than the people that own them think they should be.

It's that critical difference of what – it's a term that we call an enterprises network boundary. It's the difference between what they think it is and then what it actually is as we see it on the global internet. That difference is everything and can be very, very large and very important to organizations. Those have differences and those risk there are what we are able to service in order to allow our customers to actually understand their true network boundary and understand all of the risks on it.

**[0:07:41.0] JM:** From your website, it says – why would you have people monitor an imprecise map of one network when you could have computers monitor the entire internet? You just said you communicate with every IP address on the internet. When you say you're monitoring the entire internet, what does that mean? What exactly are you doing?

**[0:08:04.5] MK:** So this involves sending out a variety of different signals, so packets to every IP address in the world and then recording the responses. So we likely clear – it's radar. We send things out and then we observe what returns back to us. We are not monitoring traffic or anything like that. It's what responds on the internet to signals that we send that are over a wide variety of standards and protocols.

This really allows us to understand for the things that respond, the software that's running there, the protocols that they speak. From that information, we're able to deduce a lot about both what's running there. Is it vulnerable or is it potentially vulnerable? Who owns it or who is ultimately responsible for this? By tying all of those together for all organizations in the world, essentially in real-time, we can provide them with a set of all of their exposures across all of their computer networks.

The important thing is that this is done independently of a view that they have of themselves. There are a lot of good reasons why using only older technologies, they're going to be very limited in what they can do and get limited visibility out of that. We don't have those limitations. Therefore, for every single and a lot of our customers, we have been a super set of information that they have had about their own assets and their own asset status.

**[0:09:27.2] JM:** Help me understand how that works. If you're pinging the entire internet with the signals that you can get from just pinging or sending some packets, and getting some responses and measuring those responses, then you have customers, particular enterprises that each of those enterprises, some subset of the internet resources belong to that enterprise. Are you doing some kind of diff between a enterprises, internal network and then the global internet pinging? You could tell me as granular level as you can what's going on there.

**[0:10:08.5] MK:** Sure, absolutely. For us, it's the difference between their public network presents and what they think their public network presence is. We really prefer the term 'publicly accessible.' Again, we are not hacking in any way, but we are seeing what is actually accessible on their networks that belongs to them from the public internet. This can be everything from now their core autonomous system and IP ranges that they are authoritative for, all the way to things like Cloud infrastructure that they set up.

The trick is that there are a large number of modalities where their traditional monitoring systems will not work or will not be integrated or just are not fully integrated, and therefore they lack visibility. I'll give a couple of examples of this general trend. One is MNA events. In a merger and acquisition, especially between two large companies you'll typically have vastly disparate IT resources and IT infrastructure. Getting those two speak a common language can in some cases be a multi-year or even multi-decade affair to actually integrate them and provide total visibility.

In contrast to this and we've seen this for multiple customers, we don't care that the systems that they're operating internally are not mutually inoperable. They all speak the same protocols, and therefore we are able to both identify assets that belong to both the acquired company and the acquiring company. Then also show them where all of those gaps lie. In a lot of cases, these will be blind spots for all of the companies on both sides, because just they have these gaps. They persisted, and then they never took the time to fully integrate it.

Another example would be something like Cloud hosting, where again there is a verging market for secure Cloud provisioning systems, but there is also a huge area called shadow IT, where you'll have developers for these companies that will actually go outside of normal deployment processes. And spin-up instances on AWS, spin-up instances on Azure or Google or Digital Ocean. They will persist out there and they will have many, many, many risks associated with them.

When we say we're monitoring the Cloud, what a lot of companies will say is, "We have a solution for that." That means that if their developers follow policy perfectly and always, and this will never be the case always at any large org, they think they're okay. We say, "What happens if somebody does something out of policy?" They'll say, "Wow, we would have absolutely nowhere to see that." Then we will usually have examples. We'll say, "Well that's actually interesting, because for the past eight months, we have seen the following eight resources out there."

It really is one of those where you have to understand the internet in its totality in order to be able to pick a part and understand all those individual parts a bit that are relevant to a company, even though they themselves didn't know it at first.

**[0:13:00.9] JM:** Okay. Let's say I'm running a big multi-national organization, and some developer within my organization has decided he's going to work – he or she is going to work on some data engineering application. They spin up an AWS instance and they load some proprietary company data into it, so that yeah this is a sensitive asset now. This AWS instance, they've spun up, and they're going to do some data science on it.

As the big multi-national corporation, you would like to be able to index that computing asset to know where your risks lie, to know where the surface area of your network lies. How do you identify that that developer has spun up that instance? How do you have that in a place where you can – your big multi-national corporation can index it?

**[0:13:53.1] MK:** Well, as a multi-national corporation and one – Firsts of all, you won't. The worst case is you won't even know the risk is out there, until it winds up on the front page of the newspaper. In contrast, what Qadium does is – because we search over everything, as long as there is some discoverable signal that is publicly accessible on that easy-to instance, that we are able to tie back to that multi-national org.

We do this unconditionally. So we don't make the assumption that it has to be my enclave for it to be mine. It is, no. It just has to be relatable to you and associated to you in some way. .As a foundational principle of our systems and our methodologies, we assume that we might always never know the full picture, which means that we're constantly iterating towards a more complete vision, rather than thinking, "We have everything locked down. We don't need to keep searching." That is what leads to be the most complete solution that any of our customers have ever seen.

**[0:14:57.2] JM:** Can you give me some signals that might indicate how would you – this shadow example where some rogue developer, but not to miscreant, but just somebody who is trying to setup an AWS instance to do some data engineering. How would you be able to map between –

**[0:15:16.5] MK:** Sure. So we use a lot of high-assurance signals. There is obviously a lot here that goes into the secret sauce, but I'll give you a great example. Let's say that your developer really likes SSH, because it's secure, it will do everything and he likes to manage public keys, and he has a small set of public keys. He's used this on attributed infrastructure before. So imagine, there is an SSH public key that Qadium detected on a corporate asset of mega global corp, say.

We detected it two years ago. We now detect that same SSH public key on a Microsoft Azure instance. That is a very high-assurance signal that that asset is very strongly associated with the organization, because the same public key we can actually – so in our case we gathered all of them. Well, we can make sure it's not a manufacturer default or anything like that, and very quickly we can explain a way, other hypothesis and say, the only real reason this would be here is either the private key that used to belong to this company has been compromised, in which case you also have another problem. Or this is actually a pretty strong association. Then we can report all that information explainably to our customers for why we are associating this asset with you across a number of different modalities.

It's because we have this richer data set and because we have this data set of associations for corporate assets that we're able to find these needles in the haystack. In our case, they're explainable and every single case, our customers eventually say, "Well, I didn't know you could do that, but you were right."

**[0:16:54.5] JM:** Well that's pretty cool. You can look at the public key infrastructure as a way of drawing a map of the internet.

**[0:17:01.7] MK:** Absolutely. For us, the main trick is everybody asks us what is the one thing that you guys do that enable this? The real key is it's not one thing. It's many, many dozens of things, but they inter-operate together holistically and they allow us to basically continue to ask and answer these questions that allow and truly find the true extend of organizations network boundary throughout.

A good anecdote we have is early on when we were much, much smaller even, but our systems were at a good level of maturity. They're even there now obviously. There is, I'll just say a very large European reinsurance company came to us and said, "That's great. Our networks are super locked down. We spend hundreds of millions a year on security, and we office gate ourselves as well. The best we've ever been able to get from any other vendor or consultant is 20% of our network space. We have a view of our own network space. The best we've seen from everybody – everybody pitches us is 20% and that includes everyone out of the sun. We don't think you kids in California can do this."

We said, "Challenge accepted." We came back with a 110%. They agreed to the other 10% was shadow IT and AWS. They asked, "How did you find this?" A number of it was along the methods I just outlined, who are looking for these associations. They can be many different things. So an SSH key is one example. Just the name on a webpage or stylistic information that we put in, that we can then filter down in a high-assurance way.

The main trick for us is that our false positives are basically zero specifically, because we're not just searching for the name of a company, say in a web banner. It's actually much higher assurance and using signals the machines give off rather than those that people type in.

[SPONSOR MESSAGE]

**[0:18:59.4] MK:** Simplify continuous delivery with GoCD, the on-premise, open-source, continuous delivery tool by ThoughtWorks. With GoCD, you can easily model complex deployment workflows using pipelines and visualize them end to end with the value stream map. You get complete visibility into and control over your company's deployments.

At gocd.org/sedaily, find out how to bring continuous delivery to your teams. Say goodbye to deployment panic and hello to consistent predictable deliveries. Visit gocd.org/sedaily to learn more about GoCD. Commercial support and enterprise add-ons, including disaster recovery are available. Thanks to GoCD for being a continued sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

**[0:19:59.4] JM:** Okay. You've give us a picture for some of the data sets that you're building across the internet. I'd like to talk a little bit about the indexing and the data engineering challenges. By the way, the whole story behind Qadium is pretty interesting, and I don't know if we'll have time to get to that, because I'd like to go to the engineering stuff, because I haven't seen you cover that as much in other shows, other interviews and presentations and whatnot.

Let's go through the engineering process. So you've got a index, every device on the internet you're going to build a topology of some different networks and there is variance in these different topologies that you're going to build from network to network, and it's not like a standard set of devices, and there's not a – You're talking about you want a really high-assurance rate. Give me an overview of this – I think you mentioned every hour, you're able to index the entire internet. What are you doing? What does your crawler do?

**[0:20:59.1] MK:** Sure. Absolutely. From the first part, it's similar to a crawler, but it actually does more than the internet. So a lot of people, even software engineers they're used to experiencing the internet as pretty much just web or HTTP or HTTPS. It's what's in your web browser, or on your iPhone. We definitely index all of the web, but there are also all of these other protocols that are very weird and strange.

I'm sure a lot of your listeners are familiar with SSH or FTP. Other ones go really down into the weeds and get very manufacture specific. There is actually for example a protocol called Ethernet IP, which is a protocol. It is not Ethernet and it is for industrial control systems. So we have a huge, huge, huge library of these payloads that we launch.

Then there is the question of where is our deployment infrastructure? So our infrastructure itself is in many, many different places around the internet. We have lots of points of presence globally. A lot of this is that you will see different parts of the internet from different source locations. If you're looking at Europe from the United States, there are actually going to be responses that will differ than if you're looking at Europe from Europe, of if you're looking at South America from Europe or South America from South America.

We have a geographically distributed presence, and everything is dock or containers that we run. Then we have a lot of our own infrastructure code that will be written in go to really orchestrate all this background, because it's a globally distributed heterogeneous system that operates and heterogeneous would compute and networking environments.

We need to make sure if things, like if there is a BGP rule that gets borked and we can't communicate for some set of our nodes for five minutes, we still don't want to actually DDos anybody, for example. Step one is have infrastructure. Set it up very well, monitor it and have really, really good dev ups around how we all should deploy and launch these payloads. Step two is then when it all comes up, you need to both make sense of this.  There is a lot of great monitoring, so for the tools that we use, some are the standard big data suite. For us, everything goes into Kafka, and then into our own ETL topology.

There is a lot of monitoring around and making sure that things like that pressure don't add up. That all of our workers are keeping up. Then we started off actually using H space and we've since moved to Google Cloud, so we're very big users of Bigtable and BigQuery infrastructure as our source of truth. It's really pretty amazing that even as we're now a series B startup, but obviously even though as a series A startup, you can operate at petabyte scale now, and that's the way the world works and that's what enables a lot of these be possible is us utilizing cloud hosting resources.

A lot of these goes to in contrast the scales of something like a Google or Facebook we note there. I've taken huge amounts of video rich media. Most of what we deal with is text, which is one highly compressible but also much smaller. So it is now possible to not a small budget, but also not a nine figure a year budget to actually store and index everything on the internet as a reasonably well funded startup.

**[0:24:14.5] JM:** For sure. That was a great explanation. Does the problem statement from a high-level, is it like let's – we need to ping every IP address that we have on record and we're going to break up these IP address ranges into a bunch of chunks, and then we're going to break these chunks across dock or containers that are distributed around the world, and each of these dock or containers is going to go through their IP address range and they're going to ping all of those IP addresses with some different types of packets. Am I already getting something wrong?

**[0:24:46.8] MK:** No, no, no. That's exactly right. I think for us, I would just not use the word ping, because ICP ping is something that we do, but it's one of these large modalities of protocols that we use. But it's exactly saying, we'll be able to deploy those against the internet, against the IP addresses, against a large variety of the reports with a large variety of different payloads.

**[0:25:08.7] JM:** Are you running the same test against every IP address, or do you have some internal mapping where you've said, "Okay, this particular IP address tends to respond to what was – Ethernet something, or whatever?"

**[0:25:22.7] MK:** Ethernet IP. There are a whole variety of very weird industrial controls and protocols that we test for that, actually a huge value for some of our customers. Generally for us, it's a hierarchy. There are some things that you want to constantly test the internet for, because just because something has never had something before, it doesn't mean that that will be the case going forward.

So you want to test for those continually. You also want to test for known customers in different ways, and when someone becomes a customer we can do more with their IPs under waivers. Then on top of that, there are other events that if something changes, we can react accordingly and say, “Hey, if something’s changed, is there something interesting here maybe we should gather a bit more information?”

**[0:26:05.8] JM:** Do you have a – You have a mapping for each IP address, like what are the things, what are the protocols that this IP tends to respond to?

**[0:26:14.8] MK:** Yes.

**[0:26:15.6] JM:** Okay. Interesting. All right, so you get to each IP address. It’s getting hit with some packets and it’s responding with some interesting information, you’re gathering all that information, buffering it in Kafka. How is that Kafka – is there anything interesting going on with the Kafka buffering layer? Where do you have that stuff?

**[0:26:40.4] MK:** Yeah, so all of our backend is in Google Cloud. All that is auto-scaled. We were on Storm – we love data flow. It’s pretty amazing. How that works, so a lot of our systems are very much scaled to handle global enterprise workloads that we deal with, but it’s pretty amazing where the systems are definitely not trinky and we have absolutely amazing engineering teams on them. But again, my main point especially for your listeners is it’s not that it’s a push button. It definitely is not, but you also don’t need a team of 50, which is what you used to need.

It’s very interesting to see the scale on leveraging you get with software combined with a lot of these very large cloud hosting services, where with one thing some of our larger customers ask is, “So you guys must gather a huge amount of data. Where do you store it?” Our answer is, “Well, we store it in Google and then we also have back-ups in Amazon.”

They’re like, “Wait. How do you do that?” It’s like, “The scale of petabytes is quite large for a startup. But it’s rather trivial to run on those platforms.” Compared to YouTube cat video uploads will just dominate all of our data costs. It’s very nice for us that we can ride those cost curves, not need to solve those problems, and instead basically specialize in the things that matter to our customers, which is really taming the complexity of what a modern IT environment is like, and I'm showing them that vision of themselves.

It's funny. For some of our customers, the largest it doesn't fit in an Excel spreadsheet, which if they're still in Excel 2003 with 65,000 lines. If they're on Excel 2011, I guess makes a million lines and that's large. Then for other customers largest like, "You're only at petabytes. You're not yet at exabytes?" I'm going to say, "No." It's like, "Oh, that's reasonable," we say. Thank you. It's still very valuable, but the scale is now achievable today with a good set of tools that has been improving quite dramatically, even as we've been doing this for the past three years.

**[0:28:43.9] JM:** I'm sure you have an amount of gratitude. It's probably tremendous, because I was watching some of your videos and I think you were responsible for Hadoop clusters in Iraq or something working, or DARPA. Like monitoring Hadoop instance back in the day, I mean it's so much worse than figuring out data flow today.

**[0:29:09.4] MK:** Yeah. Those were in the pre-Qadium days, I was the lead data scientist for DARPA in COBOL a lot of interesting things.

**[0:29:16.7] JM:** Afghanistan, okay.

**[0:29:17.8] MK:** Afghanistan. I joke that I would study abroad for my PhD program. I had actually stopped out at Stanford at the time briefly to go to this, because I very much believed in the mission and the people there. Yeah, it's a very different environment because there everything had to be on closed networks, things would go down all the time.

We had all sorts of bandwidth issues. Occasionally when we be getting data, a hard drive would come back with a gunshot through it. There were very, very different problems then. Yeah, it's pretty interesting. You're usually thinking of missing data as your experiment –

**[0:29:48.5] JM:** Facing the generals.

**[0:29:50.2] MK:** Yeah. I haven't thought of that analogy. That's actually pretty close in – at least in geography. Yeah, it was a different prompt set, but it actually inculcated a lot of the same ways of thinking, and specifically for the problems that we were looking at, it was how do you tie together many, many, many disparate pieces of information that in in themselves were actually not meant to by tied together.

It's how do you build the equivalent of an approximate data warehouse when you don't necessarily have primary foreign keys. But you still know that these links exist, but you need to be able to find them. A lot of that that we started doing, and Afghanistan actually translates very well to cyber security where there are many, many different pieces of information you can gather. When you start to have all of them, they start to create a very good picture if you know how to link t hem together.

**[0:30:44.2] JM:** Well tell me more about learning that. Because you're basically saying, given enough data, you can asymptote towards being able to find something that resembles – like it's 99% a signal. Let's treat it like a signal.

**[0:30:59.9] MK:** In general, yes. I think the main key in all – unfortunately, I'd have to be a bit vague about some of the Afghanistan work. But the main point is when you're doing this form of analysis, it's all going to be in a regime where you're typically not going to have labels, because you're just gathering data from different sources.

It's also the scale and complexity where defining what a good output is is also typically not possible in every case. But what you actually want to do is you want to generate a number of reasonable hypothesis for here is a pattern. It seems strange.

Here are four main explanations for it. One is bad, three are benign. Is there a way that you can generate other analytics to basically explain a way of why it's none of the three that are benign? If you do that and then you apply that scale, you're left with a very good story where you say, "Here is a pattern. It has multiple explanations." Here is also why we do not think it is a normal pattern or an aberration. If that is credible and then you can validate the credibility of that at scale, what you're left with is actually a very high-assurance battle-tested hypothesis that has been verified does not have a good alternative explanation. When you iterate that a lot, you get to very powerful conclusions.

**[0:32:19.0] JM:** Let's go back to the data infrastructure discussion, because we stop short of getting to the end of it. The data flow point. That's pretty interesting, so I did a couple shows about Google Cloud data flow, and a couple takeaways I remember from that were like, let's – this whole idea of batch versus streaming is kind of an illusion, because if you just got data that's going over a TCP socket and that's just data. It comes in spurts and sputters all of the time.

You can look at it as batch. You can look at it as streaming, but really it's just data. I think data flow looks at that as a first principle rather than things like the Lambda architecture. It sounds like you were on Storm for a while, so maybe you were doing something like a Lambda architecture where you had some data that was streaming and you had other batch data that was being resolved and you migrated from that. If you could talk a little bit about that streaming data infrastructure.

**[0:33:27.4] MK:** Sure. I think a lot of it for us is a – well, we go back to what are the tools we need and what are the actual problems that we have. For us, being in an enterprise space, it's quite nice that saving 50 milliseconds does not somehow reduce a click-through rate by some percentage and lead to millions of dollars of losses.

When we think about streaming in big data, it is not a every millisecond matters in terms of latency, and that's really why dataflow is so great is that the difference between streaming and batch, for us it's really get there in a reasonable amount of time, which is for us typically – ideally tens of seconds to maybe a few minutes. Sometimes hours is okay if it's a very, very huge job that just come back, but then we don't have to focus on the differences of that, because I think everyone agrees that the data processing and the data storage infrastructure is extremely important and it's very necessary. But at the same time, it is definitionally a means to an end.

For us, the degree to which we can make that more just a means to an end and abstract that away has been a lot better. Like when we ran Storm, really there were lots of issues, instrument and things. There were all sorts of issues with back pressure, all sorts of memory allocation issues, and that's not to say that Storm is bad. Storm is fantastic. We were doing some very complex topologies early on the – I had some architecture changes that we were very happy to make as we iterated it.

But the main thing is we didn't actually care about running it. We wanted to have it get out of the way and focus on where we're great, which is really managing lots of complexity and variety of signals at what I would call reasonable scale. So a petabyte scale and with reasonable speed, and that's where we came to a data flow was after trying pretty much every other framework under the sun. We tried Spark SQL for a while. We tried a lot of others, and then we finally hit on something where the system broadly got out of our way.

We didn't need to deal with it and that was really, really nice because that freed up a lot of extra capacity and just said this is a solved problem that with one of the largest infrastructure companies in the world is invested in and we'll continue to invest in. That's great for us, because before that, just the number of things that they gave us when we tried to run them on their own, do not take care of on their own is rather astounding. It's pretty amazing that these cloud obstructions exist. In our experience really do work and we have confidence in.

**[0:35:54.9] JM:** NoOps.

**[0:35:57.0] MK:** Exactly, yeah. We've started to look even at some sort of those things, like AWS Lambda. Some of the equivalence, we think it's still a bit really, but that's still where we want to go to is ideally NoOps, or if we have Ops, manage it very specifically for specific reasons, because it's a core competency of ours.

[SPONSOR MESSAGE]

**[0:36:23.6] JM:** At Software Engineering Daily, we need to keep out metrics reliable. If a bot had started listening to all of our episodes and we have nothing to stop it, our statistics would be corrupted. We would have no way to know whether a listen came from a bot or a real user. That's why we use Incapsula, to stop attackers and improve performance.

When a listener makes a request to play an episode of Software Engineering Daily, Incapsula checks that request before it reaches our servers and it filters the bot traffic preventing it from ever reaching us. Botnets and DDoS attacks are not just a threat to podcasts. They can impact your application too. Incapsula can protect API servers and micro-services from responding to unwanted requests.

To try Incapsula for yourself, go to Incapsula.com/2017podcasts and get a free enterprise trial of Incapsula. Incapsula's API gives you control over the security and performance of your application and that's true whether you have a complex micro-services architecture or a Wordpress site, like Software Engineering Daily.

Incapsula has a global network of over a 30 data centers that optimize routing and cashier content. The same network of data centers are filtering your content for attackers and they're operating as a CDN and they're speeding up your application by doing all of these for you and you can try it today for free by going to incapsula.com/2017podcasts and you can get that free enterprise trial of Incapsula. That's Incapsula.com/2017podcasts. Check it out. Thanks again, Incapsula.

[INTERVIEW CONTINUED]

**[0:38:13.4] MK:** Lambda and Google Cloud functions, that stuff makes perfect sense for your application, because you're just like sending these random stateless pings every five minutes or every hour or whatever and like just passing the data onto dataflow. That's going to be massive cost reductions.

**[0:38:31.1] MK:** Yeah. We're very much looking forward to that and we're doing a lot of experiments with tooling around that. Would love to come on a subsequent and show our experiences with that once we've done that migration. That's where we see everything going.

**[0:38:45.1] JM:** Yeah, let's do that. Can you give me a preview like what – So you said it's too early. What makes it too early?

**[0:38:51.0] MK:** Just the current system actually works quite well. Part of it is a – we see it as even better, but there are higher priority items and –

**[0:39:04.6] JM:** your margins are good enough.

**[0:39:05.8] MK:** Yeah, exactly. Once some sort of marginal rate gets tripped, we'll definitely re-prioritize that. But right now, a lot of it for us is really both find the right tool for the right job and right now definitely good enough. Everything is humming along. That's where everyone wants to go. The interesting thing for us is what's the time scale involved in that?

Also, at some point I'm sure one of our infrastructure engineers is just going to want to try this as a fun project and surprise all of us, and then I would be like, "Okay, yeah you were right." I look forward to that surprise.

**[0:39:43.9] JM:** All right. Yeah, that would be a nice 20% time project. Data flow is doing what for you exactly? It's like calculating something, bucketing something. What kind of stuff are you doing in a data flow?

**[0:39:58.4] MK:** Very large amounts of things. We just have RN-type system for some of these things. A lot of data flow jobs are actually instantiated and enforcing type correctness. It's also piping a huge amount of information through our systems from a raw state as we gather it into unpacking protos, parsing them and then putting everything into Bigtable in terms of how we're actually storing and persisting our single source of truth.

It is a very nice series of pipelines that ideally just work. From that, they take a lot of our raw results and then put them into our tables. We then increasingly have those tables also go into flow jobs that go to our APIs and frontend as well, and it's basically how we are codifying all of our jobs that are any sort of ETL. Then it's nice because all of our engineers just need to learn one tool in order to do these jobs and it's a consistent framework with a consistent set of APIs.

**[0:40:56.8] JM:** Refresh me on Bigtable, and then we'll talk about BigQuery. I'm a little more familiar BigQuery, but is Bigtable, is that in memory thing, or is it a disk thing. What exactly does Bigtable do for you?

**[0:41:08.0] MK:** Bigtable is essentially Google's hosted Hbase. That's actually being unfair, because Hbase was based off of the Bigtable paper that – so Bigtable is the original Hbase. It was Hbase before Hbase, it is hosted by Google. So initially, we actually ran our own Hbase cluster internally on actually hardware that we got. I can say that that was a decision that in retrospect was definitely not a great one for us and it's been fantastic to migrate that to Bigtable.

It is definitely not a memory. It is a hosted solution that basically allows us to specify certain access patterns that we have that are efficient for data. It's also pre-spanner at Google. It was one of their largest systems for storing many, many different data sets across a large number of team.

**[0:42:01.1] JM:** Then BigQuery is based off of Dremel. I guess, that puts your data into a columnar format, so that you could do aggregations really aggressively, is that right?

**[0:42:13.6] MK:** Yeah, exactly. Again, it's a successor to Dremel, but it's one of these in-memory hosted solutions. We also used to run Impala ourselves hosted and then have switched to entirely a cloud infrastructure there. That's really for interactive analytics. It's quite cool that you can actually query over terabytes in seconds.

We use this extensively on our data science team when we are doing a lot of exploratory analytics. This is a key feature for us, because no one has actually gathered this data set at scale, so we are actually the first organization to really have this data set at any kind of scale and have the ability to introspect across it. We both have to ask and answer the questions. Being able to ask many questions iteratively is very important.

I joke that it's much like how GP use enable a deep learning, basically just by allowing researchers to try more things after they would see the result of an experiment and therefore iterate faster. We use that query in the same sense on our data science team to understand it and ask questions and get answers much faster where it's a very different experience if you can query over, say all web servers that we've ever seen in 20 seconds or less, rather than submitting a batch job, waiting for it to come back, getting coffee. Everything that you were thinking about for your hypothesis is just gone.

It really changes the game for iteration speed and how you think of this, because now all of a sudden you can actually ask questions of everything, and then you can ask another question of everything based upon the answer to your previous question and you can chain this very, very quickly. It's absolutely phenomenal.

**[0:43:59.3] JM:** Those are like adhoc queries though, right? You're not talking about things that are displaying on dashboards.

**[0:44:04.1] MK:** Absolutely. Yeah. If you're doing dashboards, that would get very expensive very quickly. These are ad hoc queries that will typically generate new features and product insights for us. One way to think about that is we've seen the ecosystem of all data of something. Like in the example of SSH keys, let's take FTP servers instead. There's actually a broad categorization, so it's like the file genetic history of all FTP servers that we can do.

In doing this, I should be able to track different insights by saying, "Okay, actually all these category is actually going to be both vulnerable, and also we'll have other risk or other things that our customers might want. Let's tag it, let's structure some data around that." But then, what BigQuery gives us the ability to basically partition the entire space of a different data set into these different categories and see the forest and the trees and the leaves simultaneously, which is really quite nice for us, because then we can go back.

Again rather than having to run – having to say, "I have one idea. Waiting a couple hours for this batch job to finish." Instead, "Oh, I have an idea." "Okay. Cool. Wait." "It came back with zero results. Why is that?" "I added a space to a batch X incorrectly." The small things, it's one of those that come up all the time. Nobody talks with them, but that's actually the rate when the need step is you should be able to make a few small mistakes, tweak things iteratively because that's really how all development and research and science is actually done.

When all of the other stuff gets out of the way, that's when we really let people use the right tools and then say – at Qadium we say you're just a SQL query away from internet discover. It really is true, because we empower our teams with these tools.

**[0:45:48.3] JM:** Cool. That's so much better than the Hadoop and COBOL days when you have the space, actual space and it just dismantles your query and there goes three hours.

**[0:46:01.8] MK:** Yeah. I actually add something to that, in COBOL but this actually want to beam was that because we would not have trust, or the systems would be very long, we would actually ship in pelican containers very, very large workstations with us. So this would be now HP Z800 Workstations. They would lots of memory, like 200gigs of memory. But this fundamentally limited you, because it meant that it would scale up as much as you could on a single system, because you can bring it out with you, you can plug it in anywhere.

But it also meant that if it did not fit in that memory, like I remember just getting anxious and having almost heart palpitations when I would be in country and I'd be like, "Oh, my God. I need to run this on Hadoop." Like, "Why me?" It's a four-hour job later, but that actually affects what you're going to do and affects the conclusions that you're going to do when instead of you have just one thing that says, "It works, and it works at global scale." That actually really is a game changer for us.

Some of the enterprises we work with, so for most of our customers we're the poor company usually by two to three orders of magnitude of market cap. Even then when they have those same problems when they're trying to do analytics, even over their internal structures and were able to say, "Oh, why don't you just run it over everything always." That is actually cost-effective now.

We get this day stares. Sometimes we're like, "Wait, you can do that now?" We're just like, "Yes, we've been doing it for years." I really do think that the tools you use, it's similar to how the languaging affects how you think about the problems. The tools you use do as well, because if you're not used to searching, or you're used to saying, "I can only do this problem if it fits in this system," you're going to be missing out on all of the things that don't fit into the system that you didn't know were there, and we'll try to eliminate all those biases as much as possible.

**[0:47:51.8] JM:** Couldn't agree more. Let's close the loop on the data infrastructure and how that turns into the product that is Qadium. So you got this data, it's going through stuff and data flow, it's getting put into Bigtable, you have the new adhoc queries with BigQuery.

Help me understand how you get from that data infrastructure and then you layer on products that people are actually using within an enterprise and what they're doing in the enterprise.

**[0:48:20.7] MK:** Sure, absolutely. With all these data, one of the key features and also requirements of Qadium is that we let an organization see everything that's relevant to them and nothing else. The example is for and I would take – if you have fuko and barko. Fuko is allowed to see everything of fuko. Fuko is not allowed to see anything of barko and similarly for barko.

The way that we enforce that is through this association engine of basically finding all of the assets and finding the network boundaries of all these companies. We give them total visibility of that, but we do not allow them essentially a total global query access or anything close to that on the rest of our data. That's where the customer part comes in, because we have this big global data set. It's very, very lockdown.

We then filter that on a per customer basis by all of their assets and how they change essentially in real-time. The customer view is basically the global data set filtered down to their assets and things that are relevant to them to protect their own network. Then that again is a data job for that filter. Then that information gets populated in both APIs that are accessible in our systems, as well as our frontend product called Expander, which gives them a lot of powerful ways to interact and view our data both currently and historically going back over three years. Expander has lots of different, let's say aggregates of workflows and a lapse that allow them to drill down very carefully anywhere and do discovery over their entire ecosystem with assets.

**[0:49:58.0] JM:** Correct me if I'm wrong here, but this product sounds in some ways a little bit like the Palantir product, because Palantir is like this really high-value product that's purchased by really high market enterprises. I think once the company gets onboard with it it's like, "Wow, this product is beautiful. It totally changes how we do work." But there is perhaps an onboarding process, because there is a lot of data, there is a lot of integration. Things are really sensitive. I know you have some similar investors in Palantir. Am I portraying things correctly? Is the deployment process similar? Is the engineering process similar?

**[0:50:35.2] MK:** I would say that no. I should say I'm kind of laughing, because this is the first question we get asked by almost everybody. It's a very legitimate question, because the answer basically is it's entirely different, but only because of what happens behind the scenes. I would say that Palantir is fundamentally a horizontal middleware company. That they don't own any of their own data, they don't generate any of their own data.

They take organization's existing data from a bunch of different silos, create lots of middleware, create ways to link those data sets that the company own. Not Palantir, but their customer owns, link them together and then provide workflow and interaction tools on top of those. How would you connect you sales database with your supply chain database and make a better decision if you're a global manufacturer would be a classic example for them.

We in contrast are entirely vertically integrated. We own the full stack. We say packets to pixels. Say no, we send out hundreds of terabytes a month of outbound traffic, and that moves pixels on screens for customers. I think the big differences start with we own our own data. It also is we don't install anything. We actually don't even get special access upfront from our customers. So onboarding for our customers is they give us their e-mail addresses and we send them a login specifically, because all of our product is 100% web-based and software is a service. So they login over the internet and then are able to see all of their assets, all of the exposures over their assets, all the historical data we have on them all through a web browser.

It's actually a very different experience where integration for us is do you have a modern web browser? Some of it is support JavaScript. If so, great. Send us your e-mail and we will send you credentials and you login. That is it. Everything else has been done already. That is onboarding. There is zero ramp time, there is zero install. A lot of our customers are – especially those used to more a traditional enterprise model. So like, "Wait, when did you get started?" It's like, "No, we've already been gathering data on you or the section of the internet that is you for three years. So we already have everything that we need."

After someone becomes a customer, there is definitely more we do with them in terms of integrating with their other systems. A large fraction of our customers give us more data that they have about their internal systems to enhance a lot of things that we do. A lot of them also really like our user interface, which has been nice and they've asked us if we can put more data in that that they give us. The answer, of course is yes.

That's the main differences. Palantir is horizontal and does a lot of things we do. Like we do not do supply chain in terms of understanding a carton of milk goes from Los Angeles to Tokyo. We don't do that, but we vertically own our own data and we process it and we provide the user experience and we don't need to install anything in order to get that done.

**[0:53:30.1] JM:** But during the onboarding process, are they giving you some like labeled data, or are they telling you what they know and you're able to –

**[0:53:36.2] MK:** Absolutely they can. It's not required usually, but once happening as part of an early onboard in most of our engagements is we ask for a list of where they are tracking their own assets and then we do a diff and we say, "Okay, here is where you thought you were." The diff is interesting for two reasons. One, where do we think you have presence and you don't? The other ones are where do you think you have presence and we don't?

The both the false positives and the false negatives. We have had a number of customers, and actually we've learned quite wide spreading the history. Actually, erroneously claims something like an IP range that they no longer own. This is actually very problematic, because if they're doing things like penetration testing or even vulnerability scanning where they're actually launching exploits or having a team basically try to offensively hack them as the entity, they're actually doing that to someone else's assets inadvertently, which is quite serious and definitely a problem.

The other side is just as bad where they don't know to protect other things that definitely are them and they were not tracking them centrally in any way. We basically show them where both of those sides are, and then say, "This is how you actually operationalize the Qadium view of here is your network boundary, here are your assets, this is how it changes overtime. View us as the source of truth and then let's work together to implement processes in your staff." Like a lot of our customers have 10,000 people IT staff, so that everything gets found and fixed in a very short period of time.

**[0:55:08.6] JM:** All right, man. I know we're up against time. Time has flown by a really interesting stuff and we'll have to do another server list show once you've made that migration.

Let’s close off with a simple as you can think of one, or as complex as you want to make it. Tell me about a subjective engineering decision that you made at Qadium.

**[0:55:26.6] MK:** Sure. I’ll actually tell you about one that was one that I own and I think was in hindsight definitely a mistake. So early on, we subjectively thought that the only way our enterprise customers would trust us to store their data would be to have it locked in on our own enclave, and that should led us to buy our own hardware and run our own servers and actually run Hbase for everything.

When we said we don’t think they’re going to trust cloud, that wind up being wrong in a number of dimensions. We know it was going to create engineering headaches, but we thought it was necessary. So it turned out it did create engineering headaches for having to host everything ourselves. We spent a lot of time on Ops.

It also turned out to be wrong when we started to talk to a larger ranger of customers. They were actually okay with Cloud and had started to store things on their own. That was one where I think a lot of people like to say the good decisions they’ve made, I think we’ve made a number of those. I also like to be honest and own a mistake like that where we made an engineering decision based on a customer perception that was very real at the time.

We were able to migrate off it. Obviously we survived and thrived, but I think those are the big ones where you’re going to make an engineering decision based off of business perception. Again, I would not have done anything differently, because the knowledge I had at the time said that. It’s very good to own those reasons why and say that we made this with information we had. It was very subjective. It was based entirely on feel and nuance and conversations we have had with other executives. Then ultimately, it wind up being not something that was a death nail, but it was painful to move off of that and thankfully we never have to go back.

**[0:57:08.7] JM:** All right. Well that’s instructive. Matt, thanks for coming on Software Engineering Daily. It’s been great talking to you.

**[0:57:12.8] MK:** Thank you very much, Jeff.

[END OF INTERVIEW]

**[0:57:16.3] JM:** Thanks to Symphono for sponsoring Software Engineering Daily. Symphono is a custom engineering shop where senior engineers tackle big tech challenges while learning from each other. Check it out at symphono.com/sedaily. That's symphono.com/sedaily.

Thanks to Symphono for being a sponsor of Software Engineering Daily for almost a year now. Your continued support allows us to deliver content to the listeners on a regular basis.

[END]