EPISODE 1453

[INTRODUCTION]

[00:00:00] KM: When most people hear the phrase 'Internet of Things', it typically evokes an image of the connected devices we install in our homes. While this is a common and growing use case, the true winner to date in IoT is probably industrial automation. While small improvements can yield huge returns, and small errors can result in huge losses, it's critical to capture and elegantly handle telemetry data from industrial systems. The solution many turned to for capturing their streaming data is InfluxDB. In this episode, I interview Brian Gilmore, Director of Product Management at InfluxData about how real time applications achieve success built on top InfluxDB.

[INTERVIEW

[00:00:42] KM: Brian, welcome to Software Engineering Daily.

[00:00:44] BG: Thanks. Thanks, Kyle. It's awesome to be here.

[00:00:48]KM: Where does your experience working with software begin?

[00:00:51] BG: Oh, wow. Well, I mean, it goes back. I mean, early in my career, I was a hospital administrator, and I was scraping AS/400 data to help build practice plans, largely in Excel, but beyond that. I started my real technology sort of career in the industrial automation space through an operational role I had held in public aquariums while I was sort of exploring my life as a marine biologist. But you know, 10, 15 years ago, connecting up all of that operational technology directly to analytics, and customer facing systems was a little bit of development, a little bit of systems integration, but it was always challenging and always fun. That was a great place for me to get my start.

I joined a big data and analytics company in 2013, when they were rather small, and spent eight years with them working the sort of operational intelligence and the analytics of log files and was excited to be able to join InfluData last summer, to sort of start thinking about some of the same problems, just from the metrics, and eventually metrics logs and traces perspective.

[00:02:03] KM: What were some of the things that first interested you in taking the job at Influx?

[00:02:07] BG: It's a great question. I was aware of InfluxData, I had heard a lot of great things. We had run into them competitively once or twice. I really was interested in sort of exploring the open-source side of things. I mean, I think a career in technology is obviously a great career. I think one of the things that all of us have the opportunity to do is to really figure out how the exposure, the resources we have can help others. I think, sort of digital transformation hit a lot of the frontline workers sort of last in. It was going to give me the opportunity through that open source to really work on capability and on outcomes with customers before we really had to think commercially, and it is good mix of both, but I do really enjoy sort of that experimental sort of hybrid academic type engagement with our customers and partners.

[00:03:03] KM: Just in case listeners haven't heard some of our previous interviews with your colleagues or gotten exposed to InfluxDB before, could you describe what the software is?

[00:03:15]BG: Sure. Well, it sort of comes in three packages right now. We have an open-source package, which anybody can access on GitHub. It is fully featured, fully functional. It is incredibly powerful. You can build good scale. It's a time series data platform. It accepts information, textual information, metrics, and strings and things like that. As long as it's time stamped, that's great. If it's not time stamped, we'll assign one. But our sort of secret sauce is the volume that we're able to take that data, handle it, and store it and keep order of that data to make it very, very quickly accessible to the developers, to the operations teams that use our software in the open-source side.

What or we're really excited about is that our engineering team has spent significant amount of time over the last year or two, sort of moving the capabilities of that open-source software that everybody loves. I mean, there's like 450,000 installed instances of the open source out there that we know of. It's got to be higher than that. We moved it to the cloud in a way that, as -- the difference between, say, an IoT application and maybe like an enterprise monitoring application, whether it's for IT or OT, is that, for most or a lot of IoT, the data is born in the cloud. Asking customers to sort of build, run and manage themselves in the cloud. That was a place that we could help those customers in a pretty fundamental way.

We've spent hard work and a lot of time, getting everything to sort of feature parity in the cloud. Now, we've started looking at sort of a hybridization of those two platforms, the cloud and OSS, considering OSS sort of our Edge solution. We've recently released a number of capabilities that allow somebody to -- where they have IoT devices or industrial IoT devices at the Edge. They can capture that data, they can store it locally at the Edge, they can process it with flux, which is our sort of stream processing time series analytics language. Then they can either act on it locally, or share it transparently with their InfluxDB Cloud environment. It is really the best of both worlds, like truly hybrid type situation. I think it's going to be a fundamental building block for the types of distributed IoT and industrial IoT applications that will be built over the next five to ten years.

[00:05:51] KM: Given your experience working in industrial automation, can you contrast what you might do today with the latest releases and using InfluxDB to the solutions available to you at that time? What are the major enhancements that have happened?

[00:06:07] BG: Yeah. I mean, I think it's the functional sort of fundamental idea of storage and trending and recalling is there back then. I think the big difference is sort of the precision at which we're able to do this in terms of like the amount of data we collect, but more importantly, the amount of data that somebody can keep now. The sort of idea of capturing metrics as text, transmitting it with very precise, like nano precision or nanosecond precision on the data, and putting it somewhere that can actually maintain that precision in storage, so that you can use it for the analytics. I like to sort of consider it like a high-definition television, whereas 10 years ago, we were definitely not high definition, right? You had maybe like second-by-second data capture. Because of bandwidth and other technology constraints, you didn't have the ability to send the millions of different series of information that you can today.

That combination of the two things, sort of the breadth of the of the data points that you're able to capture and store as well as the precision, like the number of samples that you're able to gather in a second has gotten to the point that it's so high, that it's almost like an analog recording of whatever the system is that you're trying to work with. Whether you're building customer-facing applications on top of it or internal customer facing operational applications. Iit is really the difference, at least in my experience, between high definition today 4K and back then, like small screens and blurry pictures.

I think having that very, very precise, high definition view of operations, as they're expressed through those metrics, traces, and logs and things like that makes the sort of whole sphere of halo applications, like machine learning, artificial intelligence, augmented reality, everything that like the cloud is really enabling in terms of like advanced analytics, advanced user interfaces, et cetera. It's making those work better, number one because training machine learning models, you have to have a lot of data to do that and you have to have highly precise data to do that. It's just a great building block for the types of use cases that we were all working towards back then, but still working towards now, which is like, how do we optimize operations, how do we increase human safety, how do we reduce waste, how do we improve quality, how do we create new revenue generating services and how do we save money for the company. All of that comes together now and I'm like super excited how that's going to change over the next year or two with some of the stuff we're working on.

[00:08:52] KM: Do I answer those questions directly with Flux queries or is there a more complicated data stack that goes on the Influx is a part of?

[00:09:01] BG: I mean, most or a lot of our customers community use InfluxQL, which is actually a simpler query language It's kind of very SQL like for folks that really like that. You can't think of Flux as really a query language, it's more of -- I mean, it's not turning complete by any means, but it is definitely a much higher-level scripting language. It's not just about going to the index and getting out the time series data and doing the aggregations, the normalizations, the enrichments that you would want to do. But then it's all of the processing that you can do, all of the integrations. You can reach out to SQL server databases and pull in metadata to enrich your time series data. You can call an external libraries. You can do all sorts of different things to basically do that sort of data prep component directly with Influx.

The nice thing about Flux is it's a pipe forward operation language, so it is very iterative. You can sort of write your first line, and hit enter and be like, "Okay. That is what I'm looking for" or "That's not what I'm looking for, let me make an adjustment." You iteratively build these Flux scripts that give you the information you want. Then at that point, you can schedule them to run as tasks, and those tasks not only can just run them and output the information to a report or to another time series, but can also integrate with things like MQTT, or other external libraries, or systems and push information out to something else that might be responsible for supervisory control or can send information to a notification system, whether it's Slack or an industrial specific one. I mean, it's a super powerful language. I mean, I've been here, like I said, since last year, I'm still like digging deep Influx, and just learning all of the wonders there are in there. You asked earlier about 10 years ago, 15 years ago, in the beginning of my career, if I had had Flux, oh, my gosh, I would have loved it. I mean, I may not have had the same career path, because we would have been done back then, but it's a great language.

[00:11:11] KM: If I'm developing like a Web3 app. I don't want to say like scalability is no problem, but I have the benefit of the cloud. It's pretty easy to scale something like that up and to know typical performance metrics and just a basic web app. I'm curious if you could share what sort of challenges people bump into as they move further from the cloud, not towards the edge?

[00:11:32] BG: Yeah. I mean, especially within the world of Web3, where everything is becoming more and more decentralized. I think the fact that we still support an atomic sort of version of our database and datastore, which can be addressed either by custom code or can be sort of managed through our own tools. That allows you to see databases nearby the points of origin of the data that you would want to create. Whether it's like a distributed Web3 application, or even if it's just like a set of microservices that are running in like a particular cloud, you can keep the database close to the workers in that situation. Then, you reduce all the latency, you don't do a lot of like cross cloud ingress and egress fees. Your ability to very quickly use that data, even buy those microservices, which are generating the data, and then allow them to take action based on the information that's stored locally, is pretty seamless, pretty transparent.

I think as you look at something that's totally globally distributed, there's an initiative that we've been working on here for I think two or three years now, which we're calling IoX, which is sort of like a next generation of the time series storage mechanisms. Which initially, we're going to deploy it through our cloud service. Ideally, it will eventually trickle all the way down. That is going to move the file or the storage from like a proprietary time series index, like we have now to something that's based on parquet, and arrow flight and like other things that will allow us to think about the data itself and the way it's stored like a more traditional piece of content of file or whatever. If you think about the power there, if the database is just a file, you can have roaming data, you could have roaming compute, you could have all sorts of different interlocks between the distributed workers of your application, and the local data. Whether it's ephemeral storage, or whether it's secure storage somewhere, you'll have a way to really do that in a truly distributed manner, where you might have hundreds or thousands of files spread around the world representing a single database.

[00:14:00] KM: If I were going to do some big warehouse scale industrial project, I imagined, I'd start with a proof of concept where I get some sensor installed, make sure I can read its telemetry, I can get that stored in InfluxDB in a nice way, everything's routing. Once I've got that kind of proof of concept going, what's a typical roadmap to go from that to production scale?

[00:14:21] BG: Well, I mean, your systems integration piece is going to be one wild card. I think, when we work with industrial customers, sometimes they are very sophisticated in terms of their ability to get data to a point of consumption. For example, there's multiple factories, global distribution, you've got all of these electrical cabinets that have PLCs, and RTUs, and all these other little bits of operational technology there. Sometimes, in and each one of those cabinets, there's a computer that you could install InfluxDB on. But generally, people are trying to figure out how to haul that data back from the Edge to a centralized sort of monolithic data store. They've been doing this for 40 years with process historians, and now they're really starting to look at, wow, there's like other technology that solves these same problems that is more advanced. It's built on more like modern libraries, like Go and Rust as compared to like C# or whatever the earlier program. I don't think any of them are in Pascal, but that's kind of what I feel like sometimes.

They're looking at InfluxDB, and they're saying, "Okay. Well, this is a legitimate replacement for maybe even in its open-source form for a piece of software that they may have gotten from one of the big industrial historian vendors that cost them over a million dollars to start, and it cost them several $100,000 a year to run. They're sort of like -- they're suddenly thinking, "Wow! We've been doing this wrong. There's almost like a mindset that changes as well, that's another like piece of that transformation to go. It's not just like a technical roadmap or a technical expansion to production. It's like, how do we show the proof of value from that proof of concept? How do we explain to people within the organization that this type of technology and this approach is going to allow you to seamlessly integrate with the future of solutions, whether it's like machine learning for predictive maintenance? I mean, a traditional process historian can be a little bit rough to both get data in and get data out, like modern time series databases like us, that's the thing that we try to solve first.

For example, we were talking to a customer just a few days ago, who has a real interest in doing sort of modeling and predictive maintenance anomaly detection based on heuristics of assets in the field. It's just like a quick conversation to show like, "Okay. Here's our Python client, here's how you use it in Jupyter notebooks, here's how you build a Flux query that will return something that can be consumed by data frame, or pandas, or as a data frame or whatever it might be. Then, you can hand that data off and that process off to your data scientists, and the people who are already super comfortable in that environment, whether it be the Python or whatever.

Just showing that oftentimes sort of just like closes the door on any future experimentation, because now it's just a scaling exercise. It's saying like, "Okay. How do we capture the data from X, Y, or Z for this pilot?" Then, how can we just do that for every other piece of equipment, every other sensor, every other asset process, et cetera, because they can see the top end value that it will deliver through those other integrated services. Some of them are like other open-source services, some of them are commercial applications that just have the hooks and link directly into InfluxDB or Influx Cloud. We're seeing a ton of success there. I feel like we're sort of crossing a chasm in terms acceptance of open source and acceptance of technology that may not have the 40-year history, but is clearly the right choice from a cost effectiveness and an ability to achieve outcomes type perspective.

[00:18:23] KM: When looking at value, one of the -- someone who's rolled out in flux dB, I think, to your point, there's some intrinsic value, like we can now query data we previously couldn't query, I don't know how you put a dollar amount on that, but there's just being able to do analytics is a breakthrough. Could you speak to some of the higher order advantages and values gained? You touched on anomaly detection, what else can people do with this time series data?

[00:18:47] BG: Yeah. I mean, I think the first sort of thing that everybody goes after is, okay, now that we can grab data from a far larger number of sorts of sources, and we can do it at a much higher precision of capture. What happens just by giving our operational experts visibility into that data in this new way. It starts off with like, we've got a great sort of data exploration tool in the platform that they can sort of truly explore their data and ask questions of their data to get answers. As they start to find those early insights, they're almost like positive reinforcement clicks, where they keep wanting exploring. Because if they found a problem that they never knew they had, or they were able to do something like forecasting that's showing them that things are not headed in the direction that they were thinking.

When they do that, what happens is that, the scope of the data, it expands very quickly through those dashboards and the reports. They start to get shared out to other areas of the organization. You start to get like business leaders who want to be able to integrate their BI tools with it and all of that. As more and more consumers of the data come in, whether it's business people, or application developers, or IT people, whatever it might be, the diversity of questions, and interest, and queries and everything expands. That oftentimes drives more data, sort of like collection, because somebody from accounting will be like, "Oh! It would be great if we had this other supply chain information here, so that we could put this particular trend in perspective or whatever." All of that sort of like search, find, explore forensic type stuff is really like, that is the first level of maturity.

The second sort of level of maturity is the stuff that people I think start to do with Flux, which is, they know the stuff that they need to check on every day, so they start to write these Flux tasks, or these dead man checks, or whatever they might be so that the system is almost kind of doing their job in the background, and outputting new metrics related to the performance of an asset as compared to watching one particular voltage related to that asset. When they start to create metrics of metrics or key performance indicators of metrics, that's when they really start to see the value of like, "Okay. If we can just keep this gauge in the green, we know our manufacturing line is working well." Now, that gauge is representative of a KPI, which is probably surfacing thousands or tens of thousands of metrics or trends from the downstream plant, but it's suddenly graspable in a way that that the business stakeholders have never had before.

Once they get that, then that's when they start thinking about, "Well, what can we integrate with this? How can we make this automate parts of our business processes?" or "How can we pull in our data science team or our machine learning teams to do the clustering, the forecasting, the predictions? How can we integrate with our augmented reality system and the provider so that when people are out actually operating on the floor, the real time data they're seeing through their phone, or through their glasses or their -- whatever they're using for their augmented reality is partially or all from this sort of new source of truth, which is their time series database in InfluxDB.

[00:22:21] KM: Providing a time series solution is, from an industry perspective, about as generic as electricity seems like everybody has time series data. But with that in mind, do you see any patterns or trends about who's adopting?

[00:22:33] BG: Yeah, it's interesting. I think that there's two types of organizations. If I think if I figured out exactly what the patterns were there, the firmagraphics or demographics, we would be in really good shape. There are two types of organizations. I think there's the risk takers and there's the not risk takers. I think an organization that's willing to take risks will at least explore the idea of sort of escaping that status quo. My gut has been that the organizations on the industrial side, where there's better sort of collaboration and communication between the OT and the IT teams, especially when it comes to a cybersecurity perspective, like cybersecurity is literally like the biggest obstacle to digital transformation. It's a righteous obstacle. It definitely is something that everybody should be super concerned with. But when you run into organizations, where there's just the person at the end of the table saying, "Nope, it won't work in our cybersecurity policy." You don't really have a good chance of thinking about what is democratizing this data across the entire organization bring.

It's kind of like finding those organizations that are willing to experiment. We're really lucky because our open source lets them in a way come to us. What we do is we work through our community and everything, to find those organizations that are using our open source in a way that's transforming them as a business, allowing them to grow better, enabling their employees and all of that. Then, we have a great DevRel team, we've got a great community manager who's doing all kinds of stuff with our customers, and we do everything we can to support them. If we're lucky, and it moves into a commercial relationship, that's fantastic. I think we'll have lots and lots of commercial services to offer those folks in the future too.

I think once we get to that, they are hooked on our product before we really even deeply engaged with them, then it's just literally like mentoring the end users to sort of help them be successful. It's presenting them with use cases like through our Influx days, and through the webinars that we present, and just giving them all the information they need to experiment and to solve problems that they were never able to solve before. The only thing we really ask in return is contribution back to the code, and maybe being public about it through a blog or something like that. It's kind of a nice, positive reinforcement circle that we have with this sort of community of early adopters.

[00:25:12] KM: Well, I've been following IoT for a number of years, and something I've had a problem doing is getting my pulse on where I think we are in the evolution of IoT. You must have some unique experience and perspectives here. Are we on the hype cycle, the growth cycle? Where do you see the current state of it of IoT?

[00:25:29] BG: Yeah. I mean, that's a great question. I used to joke even 10 years ago. I had IoT in my business card, but I really could not wait till we went back to just calling it the Internet. I think it's a convenience in terms of a term. But it doesn't really specifically describe any one thing. I think what I'm really liking and what I'm seeing at our customers, the customers we're working with, especially is that, people are starting to think of IoT as a means to an outcome as compared to the outcome. I mean, I would get called in for consultation at organizations seven, eight, ten years ago. The first thing would be like, "Okay. We budgeted this amount for IoT, what do we do with it? It's like, "Oh, boy! This is going to be a lot of work."

But what organizations are starting to do is they're starting to say like, "Look, IoT, along with cloud, along with blockchain DLT, all of these kinds of great emerging technology trends that are truly contributing to the success of businesses these days, those are all like tools in the tool belt. Starting from the top, and like talking to the stakeholders, the people who are responsible for the plant floor, the oil rig, or the safety of the operations, things like that. Figuring out what their biggest pain points are, what keeps them up at night, what they got called in for the last time they were called in off hours. Then, helping them understand how you can solve those problems with technology, where one of those technologies could very likely be IoT. One of those technologies will very likely be a time series database. One of those technologies will very likely today in modern times be something related to machine learning or artificial intelligence. If you can show sort of how those different technology trends align to support them in these business needs, and how they can measure the success of those technologies as deployed in terms of like real ROI and outcomes, there's really nothing to argue there, right? It's literally like, "Okay. Here's what we're trying to accomplish, here's the technology we can use to accomplish that."

You don't run into what I know a lot of people called over the last maybe five or six years, this idea of pilot purgatory where you've got a budget for Iot and so you're going to like loop through every single one of the 675 IoT platform vendors, and you're going to try out five different time series database companies and you're going to talk to 10 different companies that have black box algorithms for predictive maintenance. Companies don't think of it that way anymore. Organizations don't. They think of it as like, "I got a problem, I got to solve it. Technology will help. Here's the best of breed vendors in each of these sorts of like categories of what I need. Then for us, we're lucky they come to us, they check out our open source, they try it out and we get called in at a point where we can really deliver value.

[00:28:21] KM: When I've worked with -- and this is fairly limited, but experience with IoT devices, I found that they're often broadcasting pretty noisy and frequent data. There's some challenges that around managing that. Are there any best practices you can share? Do we clean that up at the edge or store full fidelity and worry about it later? What are the typical processes?

[00:28:42] BG: Yeah. I mean, I think there's a lot of talk I think, especially in the industrial space about this idea of a unified namespace, which is the layer at which the semantics should be applied to the data. I think those semantics can be viewed from a data quality, like, are we using floats or integers? They can be from a content perspective, are we binary encoded or are we just plain text. It can come from like a structure or data model perspective, where we're building some type of hierarchical tree to represent this information or whatever. I think leaving that sort of semantic layer as something that sits in between the sort of the start of the data, the actual device, and the sort of point of consumption for the stakeholders is the right way. I think a lot of folks are using MQTT brokers for that these days, which, if you can imagine, if you have a MQTT environment, you could use something like Sparkplug, which is a pretty well-defined semantic for MQTT of industrial operations, it works with the ISA-95 sort of framework.

Or if you want to create your own model for the way that you want the consumers have all of this data to be able to interact with it, you can just like in any other sort of type of interface, you can strongly type and define the way that broker works, what the names of the topics are, what the names of the assets are, what the data types are, things like that. Then, as long as that sort of schema is documented really well, ideally published and self-discoverable through the through the broker, then the devices and the machines that connect up to them sort of have the set of rules that they needed to be able to put data into the system, as well as get data out of that system.

The nice thing about those brokers is, is that they do scale very well both horizontally and vertically. You see a lot of that in MQTT. We see quite a bit of that in Kafka. You're seeing -- Tim spanned the team over at -- I forget the name of the company, but the Pulsar team. They're doing great stuff there. There's a number of companies out there who have really started to think of the message broker is not really just a holding pen for data. But actually, the place that sort of defines what that data semantic, data quality looks like, and it seems to be working very well. I think I've seen that really accelerate adoption, because it sort of clarifies a big question to your point.

[00:31:24] KM: What's the state of machine learning? That's obviously interesting use case, but one that I find some companies have a difficult time achieving and working their way up to. Have you seen a lot of successful deployments?

[00:31:37] BG: Yeah. I mean, and not just now. I think, even going back five, ten years, you have companies that were incredibly successful with it, and then you would find another company that just was literally completely out of their element. The thing that I would want people to know about machine learning, and especially when people brand machine learning is artificial intelligence, which I think is a little bit confusing sometimes. It even borders on irresponsible others. But it is math period, right? If we think about statistical processing of data, whether it's detection of anomalies, or clustering data together, forecasting prediction. This is a domain that's very well defined. We've been using it in all sorts of things, like hurricane predictions and stock market predictions for literally decades.

I think the thing that has happened is that the sort of the organization's awareness of those capabilities is out in front of sort of consumerization of those capabilities for enterprise. You have these tools, which are still ultimately very scientific. Company thinks, "Oh! I can just download Jupyter notebooks, and I can do all my machine learning. They don't understand that there's an entire layer of things, like to your point that, semantics, data formatting, data quality, of course, algorithm selection, how are you going to train the model, all of that kind of stuff and they don't have the internal expertise to really do that.

I think we're seeing good success from companies that are really in the sort of machine learning from an operations perspective consulting, where they're just going in and they're helping companies. I think hiring data scientists should be like first priority right now. I like to say, the younger, the better. That kind of sounds bad, but it's true, because I think the kids that are coming out of college right now, with backgrounds in data science, they're literally up to speed on the latest and greatest of the technology. They're also like digital natives, right? They're coming out, they've literally spent their entire teen and adult life with a phone in their pocket and they just tend to look at things a little bit differently.

I think, beware of the black box, that's sort of always my first recommendation. Nobody can take your data, having never seen it before, and can run it through some type of like magic machine learning algorithm, and can tell you when all your stuff is going to break. It just doesn't work that way. You have to be ready for experimentation and iteration and all of that. The organizations that understand that and they know that it's going to be a process are doing really great things. I mean, there's a lot of work around sort of using machine learning to behaviorally baseline assets in the field. Literally, you take these streams of time series information that you would capture with and store in InfluxDB. Then you use that semantic layer to organize it into something like a thousand cars or a million power cells or whatever.

Then, you use machine learning to actually build models of regular normal operation over, and over and over again. Then the models start to really understand like what is the total picture of regular normal behavior, which is ultimately, what most people want. Everybody sort of looks for optimal behavior, but in the real world, with actual physics, that just never seems to work out. You just baseline and then you understand like, “Okay, it didn’t catch on fire and it didn’t kill anybody. This model is trained for that scenario, and then you can test your new, your real time, your latest data that you’re getting from your time series database against those models. I think you can start to detect, when very subtle things start to slip. You might have an RPM that changes over time slowly in a way that a human would never recognize and you don’t have a chart that sort of has the, the extent of X access to be able to do that. These models and these machine learning processes can detect those very subtle things and sort of create these like events, which in their own right are time series data. But those events and sort of be classified just like an IT log or alert or whatever would be classified. Is this informational? Is this a warning? Is this an emergency? Then, it can be presented to the people who need to make the decision.

I think the holy grail of that would be able to would be being able to sort of automate even that second part, the response part. And I think we got a lot more experimentation to do with machine learning to really have systems where the machine is both detecting and then reacting to the issues. It’s not because it’s impossible. If you look at very sort of strict domain application of machine learning, like the vision stuff that goes on in in Tesla’s and whatnot, it’s there. The problem is that the domain of driving is a very limited domain, believe it or not. There’s a lot of external factors and internal factors that can affect it. But generally, if we can teach a 16-year-old kid to do it, you can teach a computer to do it. I think when you start to get into the area of industrial operations, there’s a lot of nuances, there’s a lot of moving parts, there’s a lot of physics. I still think we’re a little bit of ways from just being able to buy a controller that you strap on your factory and it runs itself.

[00:37:18] KM: Absolutely. Great points. I mean, there’s certainly a lot of elbow grease, those data scientists you mentioned are going to have to work on here. A lot of that effort is like feature engineering, right? Just getting, let's say, a temperature reading off a sensor. It's great. But I actually need like the rolling average or something along those lines. Could you highlight any of the features that InfluxDB has that are useful in the feature engineering process?

[00:37:42] BG: Absolutely. I think the first one is just -- we talked about that sort of like Edge component, right? If you have a fully functioning time series database at the Edge, either embedded on the device, we have customers that are doing that on mining equipment, tunnel boring machines. Tesla does some stuff right on the power walls, with InfluxDB. Hhaving that database there to really capture that raw data. But again, that raw data is just a discrete signal. It's not labeled in any way that's really consumable by either a person or an algorithm. You can run those queries at the edge just to do simple things, like roll ups and aggregations to take a million data points and turn it into maybe a thousand data points that's more easily consumed by an application or a person. But then there's also like stuff you can do right at the edge, like anomaly detection, just monitoring, having upper limits, lower limits, monitoring the trend when it goes up or above by a certain number of standard deviations. Take the information from a minute before, to a minute after and ship that somewhere where it can be processed.

I think understanding the operations of the equipment and the assets that are being worked, as well as the processes that those equipment and assets work in, it's key because somebody's going to, for the first time say, "This is what we need to monitor. This is what it's supposed to be doing. This is how we would summarize that data. You've got a lot of flexibility there. I think enrichment is another one. Enrichment usually happens, I think sort of like maybe the second level, whatever we're going to call what's between the edge and the cloud. I think some people are calling it fog. I've got some other ideas there, but sort of that in stream processing to do some of this additional, like hard work. Enrichment oftentimes falls in there, right? Whereas, maybe it's reaching out to a database, maybe it's just attaching a highly precise timestamp, maybe it's making sure that the data is formatted correctly, contains all the right fields, the right features to your point.

There is that flexibility now to do that sort of wherever it makes the most sense, right? The old days of having to do all of that in the cloud, oftentimes, when the cloud had no sort of way to respond, or big way to reach back down into the edge to actually ask questions of like, "Okay. Well, what was this other feature or this other sensor reading?" or "Who was operating the equipment at that time?" or whatever those features might be that go into building the model Having a cloud call back the Edge to get that stuff in a late binding manner, forget about it, it's no, it's just not the way you would do it.

Where the information is created, and then where those points of enrichment are possible, like having sort of an architect, a data engineer there to define that and say, "Okay. This is what it's going to start like and this is what it's going to end. This is the things we're going to do, whether it's in two steps, or 2000 steps in between." Somebody needs to make that decision, somebody needs to define it, somebody needs to document it and somebody needs to monitor it. Because when you get to that type of complexity, if you're not monitoring it, you have no idea what's happening when it doesn't work.

[00:41:08] KM: We could imagine a pretty senior data engineer who's experienced just happens to have not overlapped with Influx before, maybe they've been on SQL Server and Oracle and stuff like. Smart person, know how to write software, know how to do engineering, they get this idea that I worked at a factory, I should be doing things at the Edge for all the reasons you mentioned. I don't know that the roadmap is entirely clear. Are there any principles of success you've seen or ways to decide what should be computed at the Edge versus shipped to the cloud and done there?

[00:41:40] BG: I mean, I think the principles sort of align with the point of creation, and then the point of consumption. I think, as we like to say in computer science, nothing's binary, but it's the truth, and that you really have a few options there. I think you just have to decide in terms of latency, in terms of availability, reliability, security, whatever it might be. At what point in that sort of like continuum between the edge and the cloud is your best place to apply it, like where do you have the appropriate compute. For example, if you need to do something, say parallelized in Spark and you want to run it on GPUs against your data, you probably don't have a nice little rack of high-performance computing right next t your conveyor belts.

That type of stuff, you're going to need to move in full fidelity to the cloud. But if it's just some basic anomaly detection and things like that, again, that expert, that data architect, the data scientist will be able to say like, "Look, we have the compute power available locally to create that particular insight. Doing it locally makes more sense, because number one, it's faster, latencies are reduced. Number two, it's more secure, it's not an ingress or egress point where somebody would be able to take advantage or intercept the data. Three, it might literally be like -- it's just needed for local decision making, like as soon as possible. It's sort of making those decisions, just based on your particular implementation of storage, network, compute, all of these sorts of very typical architectural decisions that developers, engineers, IT folks are making already.

[00:43:31] KM: Are there any industries or use cases you're particularly excited about the opportunities in in the next, I don't know, one to five years? Where can Influx really shine?

[00:43:41] BG: Yeah. I mean, I think we're seeing great uptake in manufacturing all sorts of all the typical industrial use cases. There's a number of companies out there now I think factory.io is one of them, that are actually embedding InfluxDB as part of their industrial IoT platform or solution as the primary historian part. Companies don't start, at least in my experience, building products around your products, unless there's a big customer need, right? Because they're sort of -- they're scrambling to solve for a customer and to integrate a third party's technology gets them to where they need to be going quickly.

It's exciting to see those companies sort of finally escape the handcuffs of like the older legacy processes historians and the companies that come with them, because when they're unbounded, they're doing amazing things in manufacturing, like optimizing quality. We're seeing a lot of adoption to in like renewables like solar, wind power, like water power, where you have assets that are producing electricity and then feeding that data to either a local sort of private grid for completely off the grid type facilities or feeding it back to the larger smart grid for regular consumption by other people. I think that's exciting. If you're being used in power, you know you have a powerful time series database, because power is something where electricians and electrical engineers, they don't have time or tolerance for databases that aren't looking at things like in the cycles are more precise. A lot of the process historians don't handle that. A lot of the SQL Server databases break down in that situation, because when you're talking about hertz, or kilohertz or megahertz, you really do need something with a significant amount of granularity in terms of its storage.

But the place that I'm like, I would say I'm most excited, is what we're seeing in terms of uptake and adoption with this sort of like these new space economy companies. We've got customers like Loft Orbital, several other customers that are involved in everything from launching to satellite design, satellite delivery, space exploration, like the Vera Rubin observatory uses InfluxDB as part of the system that's literally mapping the night sky, like multiple times a night. It's just absolutely incredible. All of this coming together, like seeing that affinity for those sort of like hybrid engineering/scientific/

academic people who are really driving that sort of totally cutting-edge technology in space exploration. I think that's amazing. Because if we have a fit there, and we can help them do their jobs better and faster. Ideally, our company, our products, our people will be making a pretty significant contribution to our success there in terms of whether it's just exploration or if you're into the idea of like colonizing another planet like Mars. I think it's great, because in that situation, like our current work with the industrial companies is actually a really good primer for what the folks who are doing stuff in outer space are demanding.

I think a lot of people think of science fiction, and they look at like, they have an idea of what a moon base or a Mars base would look like. I mean, I can tell you, they're much more likely to look like an oil rig or an iron mine than they do the traditional data center. If we're standing up to those operational use cases in the factory, and the academics, the scientists who are doing that space stuff love our product as well. We sort of have that I think that magic synergy there that's going to help us be successful in those truly out of this world use cases.

[00:47:39] KM: Absolutely. Where's the place listeners can go online to learn more?

[00:47:43] BG: Sure/ I mean, check out influxdata.com. Everything is there. Also look for us on GitHub, which is, if you're interested in the open source of either our current InfluxDB technology that we've been talking about, or the IOx technology that I mentioned briefly. Yeah, it's all there. t's ready. We also would love folks to contribute. I think that's one of the big conundrums for us open source folks is that, we love when people use the software, but we love love when they contribute back. Especially when it comes to like these novel one- off use cases. If there's something you can add, please let us know and put in the PR and we'll try to get it pushed.\

[00:48:19] KM: Sounds good. Well, Brian, thank you so much for coming on Software Engineering Daily.

[oo:48:27] BG: Yeah. Thanks, Kyle. This is great. I appreciate it.

[END]