EPISODE 806

[INTRODUCTION]

[00:00:00] JM: Ben Lorica is the chief data scientist at O'Reilly Media and the program director of the Strata Data Conference. In his work, Ben spends time with people across the software industry giving him broad perspective into every area of software, particularly cloud and data infrastructure. In the early days of the data engineering ecosystem, the Hadoop vendor wars were starting between Cloudera and Hortonworks.

Strata was a neutral ground for practitioners and open source contributors to meet and share ideas about the Hadoop ecosystem. Since then, the Strata Conference has grown to encompass topics such as data science, distributed databases, streaming frameworks, and machine learning. There are so many open questions in the world of data and data infrastructure today. What's the best path that an enterprise can take to build out a data platform? How should a software team be arranged to efficiently build machine learning models? Which distributed streaming frameworks should I use and for what purpose?

Ben joins the show to discuss modern data engineering, data science, and infrastructure. O'Reilly was kind enough to give us tickets to the O'Reilly Strata Data Conference, and O'Reilly has been really generous ever since we started Software Engineering Daily. They have given us tickets to several different conferences and set us up with some really great interviewees. I'm a big fan of O'Reilly, and Ben Lorica is a fantastic podcast host as well as a guest on this episode. I really enjoyed talking to Ben, and I hope you like it as well.

The FindCollabs in-person hackathon is this Saturday, April 6th at App Academy in San Francisco. If you are looking for collaborators for your software projects or for any other projects that you're working on, we'd love to see you at the FindCollabs in-person hackathon. FindCollabs is the company that I'm working on, and it's a place to find collaborators and build projects. The hackathon has a first place prize of $4,000 and a second place prize of $1,000, and several other prices, and you actually don't even have to attend the in-person hackathon. We're also having a virtual hackathon and it ends on the 15th of April.

So we'd love to have you involved. You can just go to findcollabs.com and you will see the information about the hackathon. One really cool project that is going on on FindCollabs is the Software Daily open source project. You can see the progress of that so far by going to softwaredaily.com. That's an open source community around Software Engineering Daily, and it's being built through FindCollabs with the help of David Cedric and the Altology team, and you can check out all of that by going to findcollabs.com.

Now let's get on with this episode.

[INTERVIEW]

[00:03:07] JM: When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

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

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

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

[INTERVIEW]

[00:04:58] JM: Ben Lorica, welcome to Software Engineering Daily. It's great to have you.

[00:05:01] BL: It's great to be here.

[00:05:02] JM: I want to start off by saying I have been listening to the O'Reilly data show before I listened to my own podcast, before my own podcast even existed, and it has served as a lot of inspiration for how I have done some of my shows, particularly the ones in the data space. Even like very early on, I would listen to your show and hear Matei Zaharia, or Alexio founder, that's Haoyuan. That was kind of the earlier days of the data space. It's really matured a lot since then. But I just want to say, your show has been a super valuable resource to me.

[00:05:38] BL: Great. Then I don't know if you know this, but actually it was an accidental podcast, because the original idea was YouTube. We would interview people on YouTube. But then I started thinking, "This is crazy. You'd have to schedule someone to come in to a studio," plus our video team has very exacting standards.

Then at some point I just decided we did end up recording a few YouTube videos in a studio, and then that's when it dawned on me this is not going to be workable, because it limits me to people who can go into a studio. So what we ended up doing was I just used the audio for those first few YouTube interviews and then that became the podcast. At that time, I think podcast were already big, but not as big as they are now. So then after that, it was just, "Let's just do podcast."

[00:06:30] JM: What I liked about your show from the early days was you were talking about very technical subjects, but you also were capable of maintaining a conversational tone, which I think is a hard balance to strike. But it seemed like you were, from the beginning, very familiar with who these people were. You were intimately familiar with the projects as well such that you could maintain a casual conversation about them, and I think that speaks to your background in the space.

Can you describe the data space to me from the point of view of when you started working on the Strata Conference. I think this was back in 2005?

[00:07:12] BL: Oh, no, no. So, me, personally, started getting involved in the Strata Conference in 2014. So I started getting really involved in late 2013 to plan the 2014 conferences. So at that time I guess the big move there was the new generation of big data technologies, which revolved around I think mostly MPP databases and Hadoop. So this was technologies that could allow you to scale out, run queries on your data on commodity hardware. I think those abilities scale out, but it would cost a lot of money.

So when Google first published the MapReduce paper, I think that led Doug Cutting and Cafarella to kind of take those ideas and start the Hadoop project. So that made some of those ideas available to startups, frankly.

So in 2014, you had basically the big – I would say the big topic was Hadoop, and then it was around that year that I did first started playing around with Spark. So I became close to the team in Berkley, Berkley AMPLab, that created Spark. So I started using it. I loved it. So I started really writing about it. I think in many ways, I kind of help get people excited about Spark.

Yeah. Then, obviously, there were also talk about machine learning. At that time, I think machine learning – Some people were using either R or some of the machine learning libraries in Python at that time, which it still is popular scikit-learn. But I think the resurgence of deep learning was around 2011 for speech and 2012 for computer vision and images.

Actually, now that I think about it, in 2014, the first Strata that I was a chair of, back then I had a track on tutorial in Strata called Hardcore Data Science, which was more – Because I come from the academic world, which was a way for me to invite my academic friends to speak at this industry conference.

So one of the speakers at that conference was actually Ilya Sutskever, who now is the research director at Open AI. So he was still at Google. So at that time, deep learning was really just limited to a few select groups. So you almost had to apprentice with one of these groups in

order to learn it, because there weren't really usable libraries. There was some good documentation. So you'd have to apprentice in one of these groups.

In fact, I think I may have a post about a talk that Ilya gave at a meet up where he basically said, "Yes. So this is all about oral traditions being passed from one person to another at that point." Obviously, fast-forward to today, you have well-documented libraries and researchers publishing papers right and left usually accompanied by code on GitHub from Pytorch or TensorFlow. Yeah, now that I think about it, actually the first Strata I was involved in, we had deep learning. Yeah.

[00:10:26] JM: When I think about Strata, for some reason the Hadoop vendor wars comes to mind, because I think of Strata as this true meeting of the worlds of open source, academia and industry. Hadoop was such a foundational environment for how that world has developed. How do you remember the Hadoop vendor wars, and what impact do you think they had on how the industry formed?

[00:11:00] BL: Well, I think if you look at the space back then, right? So Hortonworks had the Hadoop Summit, which was I think the original Hadoop event that came out of the group at Yahoo. Then Cloudera shortly after created Hadoop World. Then O'Reilly had an event called Strata, which was an attempt to gather in a vendor-neutral setting. Big data enthusiasts, but also this relatively new job title of data scientist.

So I think Hadoop Summit and Hadoop World were more focused no Hadoop rightly so, because that's in the name of the conference, whereas Strata had a bigger footprint. Obviously, I don't think Strata could compete with either Hadoop World or Hadoop Summit as far as the depth of Hadoop content. But we had content that wasn't Hadoop, so some of these open source MPP databases and also data science.

So it was a fierce battle between Hortonworks and Cloudera. I mean, I'll leave that to the historians to analyze. But I think there's well-documented outrageous blog post on both sides. I think a beef on the mailing list for various projects. But now they're one big happy family.

[00:12:22] JM: That's right.

[00:12:24] BL: So, yeah.

[00:12:25] JM: Obviously, since the worlds of Hadoop and HTFS, we've had a lot of maturity in the richness of the computation models that people have developed. So there was the early movements beyond just a big file system and a MapReduce model on top of that big file system. You had the interfaces like Pig and Hive. Then you started to have the real rich computation models, like Storm. I think that was around the same time Kafka came out, or maybe –

[00:12:58] BL: I think Spark.

[00:12:59] JM: Spark too?

[00:13:00] BL: Yeah. I mean, Spark was maybe – So I first started using Spark as I remember it in 2014 in my version 0.5, 0.6. I mean, at first I was a bit resistant, honestly, because it's Scala, right? So I don't want to learn a new language. But then in got kind of hooked into Scala a little bit, because of Spark, and also because Spark was just fast, because it was in-memory. Then suddenly the things that were a bit awkward to do in Hadoop, because they were slow. Like machine learning, you would do in a scale out fashion.

Yeah, actually the first Spark meet up I went to was when the AMPLab folks in Berkeley introduced Spark Streaming. It wasn't actually released. They just said, "In the next release of Spark, we're going to add this thing called Spark Streaming. Oh, by the way, it's the same API. So whatever you're doing in batch, you can just – It's the same programming model for streaming." I was just blown away by the reaction of the audience, because it was almost disbelief at that point.

[00:14:07] JM: Why was it so important?

[00:14:08] BL: "Well, wait a minute. So you mean I don't have to have two systems? I can use the same system to do batch and –"

[00:14:16] JM: Because that was the Lambda.

[00:14:17] BL: Yeah. Yeah. Then I think after that, I think Spark evolved to do many other things, but I think they were also first – As I remember it, they were also first as far as introducing this notion of SQL on top of streaming. So what they called structured streaming.

Then, obviously, today, as you point out, we went from this big distributed file system to now object stores being more performant. So now you can go to the cloud and maybe there's for many, many tasks, there's not much of a difference between object store and one of these distributed file systems, particularly if you have InfiniBand in the data center for some of these cloud providers.

[SPONSOR MESSAGE]

[00:15:09] JM: Manning Publishing has spent 20 years producing top-quality content for developers. Manning has books on languages like Go, frameworks like React, and entire paradigms, like Reactive programming, or serverless. Manning is offering 40% off their entire catalog, including these selections shown at softwareengineeringdaily.com/manning. Just use the code Software Daily 40 when you check out to save 40% on your purchases, or when you click on the add to cart button from softwareengineeringdaily.com/manning, Manning will enter the coupon code for you automatically.

Manning books are a great resource for learning about software engineering. Check out softwareengineeringdaily.com/manning to find a book that fits your needs. That's softwareengineeringdaily.com/M-A-N-N-I-N-G. Softwareengineeringdaily.com/manning.

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

[INTERVIEW CONTINUED]

[00:16:27] JM: You touched on the importance of streaming, and I think the way that you framed it was that it's perhaps a unification point for what was previously this Lambda architecture, where you had the fast data leg and the slow data leg. Can you tell me what's the importance of streaming and why streaming changed the world of big data so much?

[00:16:46] BL: Well, I think it was just also the fact that the data sources were streaming. So people were generating data in real-time. So they had to have a way to process that data in real-time, which also changed the way the business user started thinking, which is, "Oh, maybe I can start making decisions on more recent data as supposed to waiting for the thing to complete overnight and making decisions. I can make decisions based on –"

Honestly, in the beginning, it was more near real-time. So let's recomputed every hour, and then it became every 20 minutes, and then every 10 minutes. So I think a lot of these technologies, while they're cool, at the end of the day, they only get adapted if the business user can see a use case and a reason for – I mean, why isn't batch good enough? Well, maybe in some domains it is, but maybe not for recommendation systems, or fraud detection maybe. It needs to be more real-time.

[00:17:54] JM: Why are there so many streaming frameworks?

[00:17:57] BL: That is mostly an artifact of the engineering mindset of wanting to build something from scratch, I think. Yeah, it's a mystery, but I don't think it's something that we can stop. I mean, just engineers who've build these things, they always feel like whatever system is out there, doesn't suit their particular use case. So they have to rebuild it from scratch. That's not where I would put my efforts, because as you point out, there're probably enough options. There're probably other problems you can solve where you can make a more dramatic impact.

[00:18:35] JM: Can you explain Apache Beam to me?

[00:18:37] BL: I think Apache Beam was an attempt of Google to have a unified way to program all of these different streaming frameworks. So they had to have buy-in from the different communities. So I think they had this – The people who cooperated a lot with them were probably the Flink community. I think lesser cooperation – I think, to some extent, they all cooperated, but I think maybe lesser cooperation from the Storm, or maybe the Storm community, but Heron wasn't – By cooperation, I mean dedicating something to making sure they were compliant with being right. So I don't think the Kafka community was either. I think it's one of these things that, in practice, would have been good, but all these communities are busy fixing their own problems. Yeah. Yeah.

[00:19:32] JM: So it was like an API that you could use to run streaming jobs on Flink, or Spark, or Storm.

[00:19:40] BL: Yeah, I think that was –

[00:19:42] JM: Because the different streaming frameworks are making different tradeoffs, and so you want a compliant API so that you can just write to one API and have it fit to different use cases.

[00:19:51] BL: I think that was the intent.

[00:19:53] JM: What are those tradeoffs that they're exploring? That the different streaming frameworks are exploring?

[00:19:58] BL: Oh, I think they were just built by – I mean, different engines, so they have different plus and minuses. So throughput, latency are the main things I think as far as I understand. Of course, when it comes down to it, reliability and ease of operations.

[00:20:18] JM: One access is that Spark perspective on microbatches versus like the Flink perspective of every single item is a unit of the stream, right? That's a fairly notable contrast.

[00:20:34] BL: Yeah, I think so, and I think from the Spark community perspective – And they've had projects to make that distinction go away, but I don't know how much effort they've put into it, because I think the reality is – From their perspective, right? So from their perspective, they would say 90% to 95% of use cases can be covered by microbatch. So, yes, they can cover all 100% if they make a big engineering push to do so. But then the project needs other things.

[00:21:08] JM: Right. Is data flow – Do you see Dataflow as kind of that unified model, the Google Dataflow paper that perhaps would solve for every use case?

[00:21:19] BL: I think so, but you should talk to Tyler about that.

[00:21:22] JM: Okay. I'll talk to Tyler. I'm trying to get him on the show.

[00:21:24] BL: Yeah, he's here.

[00:21:26] JM: Yeah. Anyway, I think he prefers conferences to podcasts. When Kafka came out, did you recognize how important that section of the data stack would become?

[00:21:38] BL: So I guess I should say yes. Yeah, in retrospect, yes. So I think Jay Kreps wrote a post, a small book for us called *I Love Logs,* which I think made the case for a system like Kafka. But on the other hand, in retrospect, you had systems like that in the enterprise from vendors, but maybe they weren't the scale out systems and they weren't open source, right? Yeah, generally, a streaming application now in retrospect, you need a messaging layer, a processing layer, and a storage layer. But don't sleep on Apache Pulsar, which is also a great system.

[00:22:25] JM: I'm not sleeping on it. How would you contrast Pulsar and Kafka?

[00:22:28] BL: At a high-level, I think that Pulsar had the luxury of coming after. So they could obviously observe how Kafka was designed, and they could make some improvements and support maybe workloads that Kafka was not originally designed to do. Also, I think Pulsar also has a reputation for being easier to operate. I guess I shouldn't say this here, but it doesn't suffer from as much data loss. But you should talk to both communities. I shouldn't –

[00:23:04] JM: I will. Yeah, I'll talk to them more.

[00:23:07] BL: So I know that Pulsar people are here. There's a company commercializing Pulsar called Streamware.

[00:23:12] JM: So you are describing Kafka as the messaging layer, I think, but there is a lot of – As I see it, there's a lot of data gravity to that messaging layer. So that's why it makes sense to build things like Kafka streams, or KSQL on that. But when you start to do that, it starts to look – Well, correct me if I'm wrong, but Kafka starts to look almost like another data lake. Would you

say that's accurate, or do you think there's just considerable – Is there a considerable archiving from Kafka into the data lake?

[00:23:47] BL: I mean, maybe one can make the observation that it starts looking that way, but it wasn't really designed for them. So it wasn't really designed to archive data over a long period of time. So just like anything else, it could evolve. Yeah, we'll see. I mean, I guess that's the part of the streaming stack that I think benefits from having more options, like now that we have Pulsar. So we'll see how the different communities – It's good to have – As you pointed out in the stream processing, maybe we have too many. But if we only have one choice, then that's not good either. So it's good for two communities to push each other. Yeah.

[00:24:28] JM: Are there any other sectors of the data stack that you're seeing new open source projects in?

[00:24:38] BL: There's may sectors, but I don't think there's any open source projects at the scale of Spark, or Kafka in terms of popularity, but there's bits and pieces of things happening around data catalogues and data –

[00:24:53] JM: What's a data catalogue?

[00:24:55] BL: Data catalogue at a high-level is basically what data do I have? Who's using it? In order for you, as an enterprise, understand access patterns and what data is available. Also, I guess, one area where there's a lot of activity maybe is around these data science platforms, but most of them aren't open source. So that term generally means when you get to the point when you now have a team of data scientists, you should have a place where they can collaborate, use libraries that they like to use in a setting where they can collaborate, share, not only models, but maybe even share data pipelines and features. The more advanced data science platforms will have automation tools built in, so-called AutoML for automating the model selection, hyper-parameter tuning phase. But also even automate maybe some of the feature, engineering feature validation steps.

Then the ideal scenario is the data science platform actually is not just for prototyping that you can use it to push things to production. So that's one area. So the other area actually that just

caught my attention is the whole area of developing machine learning applications. So you have tools for software development. Now you're beginning to hear about tools for ML development.

So there's a company here at Strata called Comet.ml, and there's another startup called Verta.ai. But what's really caught my attention is this open source project from Databricks called MLflow, which it's 10 months old, and when it first came out, I thought, "Oh, yeah. So we don't have anything like this. It might have a decent chance of success," but I didn't really paid that close attention to it until recently.

So now, fast-forward to today, you have 80 contributors for 40 companies, and 200+ companies using it, right?

So what's good about MLflow is that it has three components and you're free to pick and choose. You can use one, two or three. But based on their surveys, the most popular component that people use is the one for tracking and managing machine learning experiments. It's also designed to be useful for you if you're an individual data scientist, but it's also designed to be used by teams of data scientists. So they have documented use cases of MLfLow where you have a company managing thousands of models in production. So they have many, many data scientists.

What else? Then going more into the AI application space, I'm excited by another project from Berkley called Ray, which is a distributed processing framework written in C++. But if you think about it, if you have a cluster, you have either kind of a lot of controls. So you could run Kubernetes, Mesos or whatever virtualization, or you have more specialized libraries, like Spark, Kafka. Then now it seems like there's something in the middle. So you have serverless, which are functions as a service, and then something – So Ray is also somewhere in the middle between lots of control and ease of use.

So what's nice about Ray is it's written in C++, but the API is small and it's Python. So, for example –

[00:28:39] JM: Is this what Pyron uses under the covers?

[00:28:41] BL: No.

[00:28:43] JM: Okay. I'm sorry.

[00:28:44] BL: So one of the things that people can use with Ray today is a library built on top of Ray called Modin, which is M-O-D-I-N. So one line of code you add to your Python code, and then your Pandas will run much faster on a single laptop. But more importantly, scales out the cluster. So one line of code.

Then Ray was also designed with this next generation or machine learning applications in mind, specifically reinforcement learning and AutoML. So the two most popular libraries on top of Ray after this Modin is RLLib, which is a reinforcement learning library on top of Ray, which means that – Actually, so reinforcement learning is not as popular at this point. It's also harder for a data scientist to get into. But what RLLib allows you to do is basically just use for enforcement learning just like you would from scikit-learn. Then they also are beginning to get into kind of distributed training of deep learning models using Ray and AutoML.

[00:29:52] JM: You alluded to MLflow, and I think that speaks to Matei Zaharia's capability of making machine learning easier to work with. I think that's been one of his focus is for a while. I've heard him say when he was kind of in academia and started to think about Spark, he realized he could go down the path of making more complex machine learning training systems or he could make machine learning more accessible.

What are the ways in which dealing with machine learning is still too hard? What are the biggest bottlenecks you see in the everyday developer?

[00:30:35] BL: Well, I think because it's really not a model at the end. So you have a model, which is an algorithm at the end of it. But, really, you have a pipeline which involves another algorithm, which is to train the model that you have at the end. So it relies on data. It relies on usually a computational graph embodied by this pipeline, which means you have to actually keep track of many things.

Even the model, even choosing the model itself can lead to many, many choices. What deep learning architecture should I use? What should architecture look like? How many layers? What parameters should I use? Just that a model, but then to get the data in place for the model to absorb is another pipeline.

So there's a lot of keep track off, which means that if machine learning gets to the point where it's deployed in many mission-critical applications, that means regulators are going to start looking at machine learning more closely, hackers who want to attack machine learning are going to look at it more closely, which means you need to be able to audit and trace your machine learning pipeline end-to-end. That's why there're all these tools that need to be built. So data lineage tools, tools for keeping track of what you did when you set up your experiment. Where did the data come from? And all these things.

It presents itself with a different set of challenges. So actually I was talking to someone yesterday about data lineage. I said, "In principle, it's easy to do if you only have one framework that you're dealing with." Imagine in a company, they have so many systems. Each of these systems can touch the data and modify and tweak it. Essentially, data lineage systems means you have to have logging built into each of these systems.

I mean, that might be easy to do if you have like a few frameworks in your company. But as you know, I mean, even startups, they start out with one or two frameworks, and then another team wants to move faster. They introduce another tool, and then suddenly you have 50 different open source libraries on what you're going to have build logging into each of these source in order to have a really good data lineage system, which actually I was joking with someone. I said, "Which means that a company?" Because if it's an easy problem, it's not a company.

[00:33:20] JM: Yeah. One of the things you touched on yesterday, I think, in your keynote, at least what I've heard. I didn't see your keynote. But the consolidation of data scientist and data engineer into machine learning engineer. Why is that happening?

[00:33:40] BL: Well, first of all, I don't think they're consolidating. I mean, I think that I would say the machine learning engineer is somewhere in between engineering operations and data science with maybe a stronger background in engineering, but knows enough machine learning

in order to build models. With the emphasis – The role emphasizes productionizing models, right?

So what I said in my keynote was I was curious as to what extent data scientists are rebranding themselves into this new role. So actually my question in my Twitter poll was quite precise. I said, "If you described yourself or use a job title data scientist two years ago, what are you using today?" So the options were data scientist, machine learning engineering, research scientist or deep learning engineer. So half said they were still using data scientist.

If you look at the number of people who said machine learning engineer, 32%, but throw in another 4% that said deep learning engineer, because that's just a machine learning engineer that specializes in deep learning. So you're at 36%. So machine learning engineer like I said has stronger engineering skills, anecdotally higher – Much more higher compensate, because it's much more mission-critical. It touches production systems, right?

So what to extent can data scientist just start calling themselves machine learning engineer? I think if they actually maybe improve their engineering skills, yes. But have a third of them are more done that over the last two years? I question that. So maybe there's just maybe somewhat of a rebranding happening there as well.

Then that title research scientist, [inaudible 00:35:23], because it seems to be more specific to people who are able to really go in there and build models and tweak them and really take your modeling to the next step.

[00:35:37] JM: And it's 2019, what are the other trends that you're seeing in this Strata Conference that you didn't see last year?

[00:35:47] BL: I think the highest level trend is the strong interest in data is crystalizing itself to strong interest in machine learning. In this particular conference, we've tried to kind of emphasize to companies that if you want to be serious about machine learning, don't just think about modeling. You have to build these foundational technologies in order to build a sustainable machine learning practice.

So I think that the big trend is people are interested in machine learning. Machine learning will impact how we build software. So then the question is how do you distinguish yourself from the pack? How do you go from being one-off machine learning team to one that really has a good practice around it? I think there's a strong interest among companies to be in this camp where they have machine learning practice that's strong and it can be sustained over a period of time.

[00:36:49] JM: Now, there's all these companies that have what I've heard referred to as a data mote. So you take a company like an insurance company that's been around for 50 years. They got so much data. They would love to be using it in practical ways and they could probably make a lot of money using it in practical ways.

They are trying to build out this "data platform" that you've talked about, and they could build it by composing together a bunch of open source projects. They could compose together cloud solutions. They could do some combination of the two. They could go to some vendor that's offering them an all-in-one data platform. How should they evaluate that decision landscape?

[00:37:32] BL: I mean, I think as you probably know, adapting new technologies, it's probably better to start with use cases. I mean, I think – Actually, when we survey companies and we try to get them to self-select level of maturity of machine learning. So people who are just getting started, their main bottleneck is just trouble identifying use cases and simultaneously convincing the rest of the org about the value of these technologies. So I would start there. What are some good use cases for these technologies and figure out how to build from those initial use cases.

Which brings up the point that one of the underappreciated things about machine learning and data and technology in general is it's you have to educate the business people too. Because in the previous generation of technologies when it revolved around BI, you had to get people to start making decisions using data, so with data in mind instead of just using intuition.

So with machine learning, it's the same way. You have to educate people in terms of what can be automate using this current generation of technologies, and which of our workflows are more likely to lend them self to automation. But I think – Our surveys bear this out, that changing the culture of a company is just as important as changing the tech.

[SPONSOR MESSAGE]

[00:39:08] JM: When I talk to web developers about building and deploying websites, I keep hearing excitement about Netlify. Netlify is a modern way to build and manage fast modern websites that run without the need for addressable web servers. Netlify is serverless. Netlify lets you deploy sites directly from git to a worldwide application delivery network for the fastest possible performance.

Netlify's built-in continuous deployment automatically builds and deploys your site or your application whenever you push to your git repository. You can even attach deploy previews to your poll requests and turn each branch into its own staging site.

Use modern frontend tools and site generators, like React, and Gatsby, or Vue, and Nuxt. For the backend, Netlify can automatically deploy AWS Lambda functions right alongside the rest of your code. Simply set up a folder and drop in your functions. Everything else is automatic, and there's so much more. There's automatic forms, identity management and tools to manage and transform large images and media.

Go to netlify.com/sedaily to learn more about Netlify and support Software Engineering Daily. It's a great way to deploy your newest application or an old application. So go to netlify.com/sedaily and see what the Netlify team is building. Also, you can check out our episode that we did with the Netlify CEO and founder, Matt Biilmann. That was a really enjoyable episode, and I'm happy to have Netlify as a supporter of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:41:04] JM: I think one way to think about top-level use cases is the operational analytics world, where you have people who are analysts looking at things that are getting served to them by Druid, Druid-based applications from Imply, or from like superset BI tools where they're getting up-to-date streaming data that's offering them real-time analytics.

How is the frontend analyst-facing data analysis world changing?

[00:41:36] BL: Well, I mean, I think for one thing, the tools are getting better. I don't know if you've gone to the Expo Hall. I invited a research project from MIT called Northstar. So they have basically like a minority report user interface. But it has AutoML. So you cannot only just do exploratory data analysis. You can start building models as an analyst.

[00:42:02] JM: How technical do you have to be?

[00:42:04] BL: There's no programing. So it's all – But you have to know the underlying data, right? Then you can start exploring. So it's got AutoML capabilities. You can build simple models. Then these simple models, they have an export function into Python. So then you can hand it off to someone in your team. I think user interface is going to be hugely important. For one thing, we're not at the point where many of workflows can be fully automated. So you still need domain experts. So we're talking about augmentation for the most part. So user interface, which might include in the future – Honestly, if machine learning is going to be impacting a lot of tasks and workflows, one of the elements of the user interface might be an explanation layer. How did this system arrive at this decision?

So user interface, not just for usability, but just for putting the end user at ease to use – Continue to trust and use the application. Actually, which brings me back to the original thing we were – Earlier thing we were talking about, this data scientist versus ML engineer. So one of the things that's happed is – Because data scientist title has exploded and gotten so popular, gotten so covered, is that there's been kind of, in my mind, somewhat of a delusion of the title.

So I was just talking with someone on Tuesday at this conference. He works for an unnamed car sharing service. There're many of them, right? He said in their company they have basically two types of data scientists. The data scientist that we kind of assume are the real data scientist. Then he said the people who used to be called business analyst who do most of the SQL, they're also called data scientist.

I told them, "Look, that's confusing, because that happened a few years ago with another unnamed tech company." Someone like us from the outside were confused, because these two people are data scientists but they really are different. He said, "Well, there's nothing we can do

about it then, because people like this title and that's a career path. So there's nothing we can do. They want to have this title."

I think what the machine learning engineer title introduces is it's a combination of machine learning and engineer, but engineer is in there. So engineer, I think by implication, requires more technical skills, which might be a little harder to inflate. So besides the fact that it emphases different things, it's much more production-focused as supposed to the prototyping.

[00:44:52] JM: Since you mentioned car companies, car technology companies, I've done a couple of shows with self-driving car infrastructure engineers. From what I can tell, those applications are going to drive some really big innovations in the data engineering landscape. One way that I think about that is the fact that these cars are driving around with big servers in them collecting so much data, which is a really weird model compared to the other kinds of applications we've had.

But you could imagine those kinds of advancements being applied to things like, "I'm carrying around a smartphone." What kinds of downstream impacts do you think the data engineering and the self-driving car space will yield?

[00:45:37] BL: I mean, I think for me, the more interesting thing beyond self-driving cars is 5G, because 5G is – I think the rollout is beginning to happen. So I think by 2020 we'll have a lot of 5G. Maybe not a lot, but we're going to have enough 5G coverage to start seeing what kind of applications come out of it, because now we're talking about machine-to-machine applications that we can't imagine at this point. I think a lot of the infrastructure that we built will be put to a test, put to the test with the rollout of 5G.

Obviously, there'll be a lot of machine learning built-in to some of these applications at the edge and at the data center. I think the more interesting thing for me to watch beyond just the specifics of the self-driving car are two things, 5G and specialized hardware for ML both for inference and training and both for edge devices and at the data center. Because both are going to – We're going to start seeing glimpses of both of them later this year. So Q3, Q4, we'll have new hardware for training ML and deep learning, which will be much faster than what we have now. Given that we're in a very empirical era for machine learning, there's not a lot of theoretical

understanding for how models work and when they work, when they work. So people need to explore a lot of models.

That means if we can accelerate training, the researchers and data scientists can explore more models. So maybe we'll have more interesting models. But the timeframe for both of these things is later this year. So the good news is that actually a lot of our infrastructure will probably be still be fine. But we don't know until both of these things come online. But I'm pretty sure that most of the data infrastructure we have in place will be able to handle some of these real-time applications. Yeah.

[00:47:46] JM: What's your perspective on the present and future of podcasts?

[00:47:49] BL: Present and future podcast? I think in the U.S. and in the west, or at least in countries I travel, it seems like we're still in the early stages. But there are other areas in the world where they don't seem to matter, like China. There are just short videos.

[00:48:11] JM: No podcasts are used in China.

[00:48:12] BL: Are they?

[00:48:13] JM: They're paid podcasts actually.

[00:48:14] BL: Oh, paid podcasts.

[00:48:15] JM: You'll make tons of money off paid podcasts.

[00:48:17] BL: But when I'm there, it seems like people are more interested in consuming short video.

[00:48:22] JM: But I mean that's the same with pop-culture here, I think. People are addicted to their Instagram.

[00:48:28] BL: Yeah. But whenever we've had conferences in China for the last few years – So I travel there regularly. I just don't hear people talking about podcasts.

[00:48:36] JM: I don't know if you're interested in this, but there's –

[00:48:38] BL: But there's podcasters here who talk about what's happening in China. That's big, but in China itself, I don't hear that much.

[00:48:46] JM: It's happening for learning. People use it for learning.

[00:48:48] BL: But here, I think we're still in the early stages, and I think – The one thing that I worry about is maybe too much consolidation too early. I mean, because some of the people might get absorbed and maybe too many show start becoming too slick and too alike. But we'll see. Yeah.

[00:49:10] JM: Ben Lorica, thanks for coming on the show and thanks for producing a podcast that's been really important to me.

[00:49:14] BL: Thank you.

[END OF INTERVIEW]

[00:49:19] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use, and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plug-ins. Use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on-the-fly. GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on

GoCD with the new Kubernetes integrations. You can check it out for yourself at gocd.org/sedaily.

Thank you so much to ThoughtWorks for being a longtime sponsor of Software Engineering Daily. We are proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]