**EPISODE 1001**

[INTRODUCTION]

**[00:00:00] JM**: Data infrastructure has been transformed over the last 15 years. The open source Hadoop Project led to the creation of multiple companies based around commercializing the MapReduce algorithm and the Google file system, both of which came out of Google as papers, and then cheap cloud storage popularized the usage of data lakes.

Cheap cloud servers led to wide experimentation of data tools. Apache Spark emerged from academia and Apache Kafka came out of the corporate challenges faced by LinkedIn. Over these 15 years, Ben Lorica has been following the world of data engineering as an engineer, a conference organizer, and a podcaster.

When he was host of the O'Reilly Data Show, his material served as inspiration for some of the episodes of this podcast, and today he hosts the Data Exchange Podcast and writes the Data Exchange Newsletter. Ben joins the show to talk about modern data engineering and his opinion on the past and future of data infrastructure.

If you enjoyed today's show, you can find all of our past episodes about data infrastructure by going to softwaredaily.com and searching for the technologies or the companies that are mentioned and we also have mobile apps that you can use to find all of our past episodes, all 1,500 of them. If there's a subject that you want to hear covered, you can feel free to leave a comment on the episode or send us a tweet @Software_Daily.

[SPONSOR MESSAGE]

**[00:01:36] JM**: Today's show is sponsored by StrongDM. StrongDM is a system for managing and monitoring access to servers, databases and Kubernetes clusters. You already treat infrastructure as code. StrongDM lets you do the same with access. With StrongDM, easily extend your identity provider to manage infrastructure access.

It's one click to onboard and one click to terminate. Instantly pull people in and out of roles. Admins get full auditability into anything that anyone does. When they connect? What queries they run? What commands are typed? It's full visibility into everything. For SSH, RDP and Kubernetes, that means video replays. For databases, it's a single unified query log across all database management systems. StrongDM is used by admins at Greenhouse, Hurst, Peloton, Betterment and SoFi Control Access.

Start your free 14-day trial of StrongDM by going to softwareengineeringdaily.com/strongdm. That's softwareengineeringdaily.com/strongdm.

Thank you to StrongDM for sponsoring Software Engineering Daily.

[INTERVIEW]

**[00:02:59] JM**: Ben Lorica, welcome to Software Engineering Daily.

**[00:03:01] BL**: Glad to be here.

**[00:03:03] JM**: You are working on a number of different things that we're going to explore, and the first one I want to talk about is Anyscale, which is a company that recently came out of Stealth, and Anyscale works on technology associated with the Ray Project. What are the problems in distributed computing that Ray is solving that were previously unsolved?

**[00:03:24] BL**: Broadly speaking, the two main constituencies that Ray can appeal to are machine learning people. These could be data scientists or machine learning engineers who are probably doing training and related activities around machine learning and Python programmers. I both cases, I guess a common thing, there's Python, because as you know, ML and data science is mostly dominated by Python. Most of the libraries make sure they appeal to Python users and so forth.

Anyscale for both communities, the goal is to provide tools that will let you scale and run your code fast and pretty much almost the same code will run on your laptop on a cluster. If you want to scale machine learning or you want to scale Python, in the future, that will probably happen

using Ray and the Ray ecosystem, which is now comprised of a few libraries, as you now. There's a library for reinforcement learning and hyper-parameter tuning. I think in the next few months we'll see libraries for model serving, distributed training.

Basically scalable ML and scalable Python, which actually I'm the whole organizer of the first Ray Summit, which will take place in San Francisco May 28 and 29. We thought a lot about the tagline for the conference and it's exactly what I just described, scalable machine learning, scalable Python for everyone.

**[00:05:03] JM**: What's unscalable about machine learning pre-Ray?

**[00:05:07] BL**: For people who want to run machine learning, let's say, on a lot of data, you might be able to scale it on one machine, but increasingly the future of ML seems increasingly distributed. As you know, the model sizes keep getting bigger and bigger and the training times get longer and longer. A lot of that is happening in a cluster.

With Ray, the opportunity there is that you have code that you've already written for a snapshot of your data, a small copy of your data and you can point that same code and it will run in distributed in parallel in a cluster almost unchanged.

**[00:05:48] JM**: Was Ray inspired specifically by the problems in reinforcement learning?

**[00:05:54] BL**: I think, originally. Yeah. Ray came out of I think a graduate seminar that Ion Stoica who's executive chairman of Anyscale and Databrix taught in Berkley. I think at the time they were trying to do I believe distributed deep learning on Spark. What two students, Ray and Philip who are now the cofounders of Ray along with Ion and Mike Jordan. Robert and Philip tried to do distributed training, deep learning on Spark and they got it to work, but they realized, "Oh! There's an opportunity here for doing this."

Then I think their co-advised by Mike Jordan. So they're machine learning people and they were also at that time interested in reinforcement learning. The original target for Ray was primarily reinforcement learning.

**[00:06:49] JM**: Why wasn't the Spark ecosystem sufficient to perform the scalable machine learning tasks?

**[00:06:57] BL**: The computation patterns for training is one thing, but also I think Ray is written in C++. Built for scale, built for speed. Originally, that was particularly for reinforcement learning when you need that low latency because you're doing all these simulations. That worked out well.

I think now if you were to do a distributed training for deep learning, you have roughly three options. I think TensorFlow has some tools, which even Rajat Monga, if people will listen to my interview with him on my podcast will admit it's a bit harder to use. Then there are some tools in the open source one from Uber called Horovod, and then one from ByteDance called byteps. The now Anyscale, hopefully shortly we'll be able to step in and make distributed training for both TensorFlow and PyTorch much easier.

**[00:08:03] JM**: You've been involved in the Spark ecosystem for more than 6 years. You've been an advisor to Databrix. The Spark programming model allowed for distributed in-memory working set for performing multiple computations and this was an improvement on the single batch run model of Hadoop.

When you reflect on those 6 years spent in the Spark ecosystem, what were the use cases that Spark has been made easier to solve? How has Spark changed that data infrastructure?

**[00:08:40] BL**: I think the incumbent was MapReduce. So at least early on, I mean, I don't think this was true when Spark really began to takeoff. You literally had to write MapReduce, right? There weren't any easy ways to do the data wrangling. Shortly after they had –

**[00:09:00] JM**: There was Hive, and Pig, and whatever.

**[00:09:02] BL**: Yeah, shortly after there was Hive and Pig. But then I think the attraction for Spark was the speed and Scala and then the unification of the different tasks that you wanted, right? I remember being at this San Francisco meet-up when Spark Streaming was announced and people were just blown away, "Huh? I can basically use the same engine to do batch and

streaming and the code almost looks the same?" Then they introduced other things, SparkSQL, right?

Then at least in those days before deep learning for traditional machine learning, Spark also had a machine learning library that people were using alongside other options. I think that has played out today. I mean, if you look at any data engineering job posting, any company doing any sort of data engineering, chances are they're using Spark in some form in their infrastructure.

**[00:10:05] JM**: How has Spark changed the process of interactive data science?

**[00:10:09] BL**: I think it raised expectations in terms of being able to get answers in a timely fashion, ease of use, because PySpark made Spark available to Python users. That means Python people could do data processing at scale. Then just more and more companies started building on top of Spark, including open source projects. But I do remember when I first started using Spark, it was mostly around the fact that it was fast compared using MapReduce. The other things on top of MapReduce, still, the execution engine was MapReduce, right? So be it Pig or Hive. Then the fact that you could do your data processing and also start doing some machine learning using the same tool was attractive I think unification is also a big part of what Spark has changed.

**[00:11:04] JM**: Both Spark and Ray came from the RISELab or from its previous incarnations. You also have Mesos and Alluxio that have come out of this lab. Has the RISELab given us a clear roadmap for how to take a research project to market or is there anything generalizable that we can take about the process of going from building a computer science system in an academic environment and taking it to market? Is there anything generalizable we can take away from the RISELab?

**[00:11:46] BL**: I guess if it were generalizable, you would see it many other places.

**[00:11:50] JM**: Clearly. You'll see it in almost no other place, right?

**[00:11:52] BL**: I think it has to do with their tradition of – Dave Patterson has introduced and now carried fort by people like Ion. But basically the few elements there, one, they have fixed timeframe. Whatever it is, 6, 7 years. Then they reboot. Challenge themselves to reboot and start over. Secondly, they're very interdisciplinary, which other computer science departments probably say, "Well, we have professors that work across different areas of computer science as well."

But then they also have this mindset that they want to build for tools that industry will use. These labs usually have industry sponsors and they have close working relationship with the lab. Some of them even send people to collaborate with the lab for a short period of time. But they have close communications. So a lot of the tools get used by the sponsors early on. Look at – Ray is already in production in a bunch of places including ad-finance which relied heavily on Ray for 11/11, which is the largest shopping day in the world.

The sponsors give them feedback. I mean, this is a very formalized process. They have retreats twice a year and they present the projects, status, the incremental updates to the project as well as kind of the more longshot projects that they're working on and then the sponsors will give them formal updates at the end of the retreat. So twice a year.

You get this constant feedback, you have people from multiple areas of computer science because a lot of these problems require people from many areas. Then the mindset that we're building for tools that – We want to build tools that people can use.

**[00:13:57] JM**: You've spent a lot of time talking to big companies; Twitter, Uber, Netflix. You did a lot of work organizing conferences where you would vet the talks and vet the developments that are going on at these companies. They're often building internal machine learning frameworks, data pipeline tools. Why do all these companies like Twitter and Uber and Netflix, why do they end up inventing similar systems at the same time?

You see Netflix has a streaming system at the same time that Twitter has a new streaming system. Uber has a new machine learning framework at the same time that Netflix has a new machine learning framework. Why do these things happen simultaneously?

**[00:14:44] BL**: There are two main reasons. One, I mean, arguably when they build these things, in some cases at least, these tools that they need might be very specific to their infrastructure of their needs and they may not exist and there might be requirements around privacy and security where they're not comfortably using external tools.

The first main reason is they have very specific needs and the tools just don't exist, and if they do exist, maybe it doesn't adhere to their requirements. Maybe they prefer to use open source tools. I don't know. Then I guess the other reason is if you have a company that's in the tech space and you want to retain your engineers, you've got to give them interesting projects to work on, right?

I think that will be the case for a while until, let's say – A lot of these companies are trying to build these end-to-end ML platforms. At some point, those platforms, like Databrix, will be enough for these companies. If not now, for most cases, they're probably enough. Then you have these full of engineers and they need to work on interesting projects, otherwise they'll move to other places. There's some of that going on too as well.

I mean, I think there are areas that are your core to your business that you can definitely point your engineers to, but there might be areas where the tools are good enough. Because it's one thing – As you know, Jeff, it's one thing to build a tool, but then you have to maintain, add features. What's exciting before is not kind of a drudgery kind of thing, right? There's an advantage of getting a tool from a vendor to specialize this in that problem.

**[00:16:39] JM**: Totally.

**[00:16:41] BL**: For example. Actually, one of the companies that I advise is a company called Anodot, which is doing machine learning for time series, which includes – The big areas are anomaly detection and forecasting. Well, I mean, you can build that too because that's an exciting project. But overtime, you have to maintain that thing, all the feeds. The algorithms have to be up-to-date and so on and so forth and then you have to scale it.

In the meantime, maybe you should be working on the core business problem that you have to tackle, be it whatever this logistics, media or whatever. I think at some point maybe the

shareholders will also probably also say, “Well, why are you guys spending all this money building tools that you can just get free?”

**[00:17:32] JM**: You see this same phenomenon in basic runtime infrastructure, like before Kubernetes won the container orchestration wars, every company was building something for container orchestration. In retrospect, there was a lot of blood that was shed as people were creating these different internal container management systems that they then had to tear out and replace with Kubernetes eventually. I don’t have any perspective, because I don’t work at any of these companies. But in retrospect, it looks like most companies probably would have been happier if they would have just focused on business logic. Ignore your platform issues. Stay on VMs, whatever, until one of these open source container things win and then just adapt the one that wins.

**[00:18:25] BL**: But then the question is would your engineers be happy doing that?

**[00:18:28] JM**: I mean, your goal is to build a business, not to make your engineers happy, right?

**[00:18:35] BL**: True. But I mean I guess in an environment where engineers are in demand.

**[00:18:40] JM**: Why do you want the engineers that you’re hiring just to make happy building some platform solution that you don’t actually need?

**[00:18:47] BL**: I’m just saying that there are some of these happening, because basically IT infrastructure is complex, complex enough that management may not precisely know what’s practical to build and what’s impractical to maintain over a long period. I’m just speculating. I think there’s some of this going on to this day.

**[00:19:08] JM**: Sure. I guess what I’m suggesting is as there are more and more companies building infrastructure solutions, whether we’re talking about for container management or machine learning or data pipeline stuff, I think the answer becomes more and more buy. Do not build. Never ever build. Just buy.

**[00:19:26] BL**: Yeah. You point out in the Kubernetes container orchestration, that's happening now.

**[00:19:31] JM**: Yeah.

**[00:19:33] BL**: I think in machine learning, it's slowly happening. I think it's happening in enterprise. It's just that we live in a [inaudible 00:19:40]. We're always talking to people at these companies and they're all building their own stuff.

**[00:19:47] JM**: Fair enough. The self-driving car companies, you got to imagine, they're solving some of the hardest problems around machine learning. Do you have any sense for what kind of infrastructure is being built to the self-driving car companies or is that all mysterious?

**[00:20:03] BL**: I don't have a good feel. I do know they're hiring some of the people that we know that came from the big data world, which makes me believe that they're building kind of similar streaming end-to-end ML infrastructure. Maybe of that, their needs are so specific. Their data types are different enough. Whatever they are, images, video, LiDAR and geographic data that maybe there's no off-the-shelf solution.

I do know that a good friend of mine who's an investor has long observed that there's an opportunity for someone to provide infrastructure for these companies, I mean in the sense that the infrastructure is so specific. You build kind of an infrastructure company around this transportation industry.

**[00:20:56] JM**: Yeah. How many customers do you have though? Are there enough self-driving companies that would want to buy this stuff?

**[00:21:06] BL**: Potentially, if your infrastructure becomes kind of dominant in the transportation industry. I mean, it's a big industry, right? I mean, look at how many people are building this kind of solutions for adtech.

**[00:21:21] JM**: Well, that's true. I mean, but there's a lot more adtech companies, like the barred entries for adtech companies are a little bit lower than –

**[00:21:28] BL**: Yeah, that's what I'm saying, is that if your infrastructure is so good and so specific at this domain, there might be a company –

**[00:21:28] JM**: Right. Well, I mean –

**[00:21:37] BL**: We're just used to all these super high valuation companies, but there's a lot of opportunities there for verticalized solutions, right? Yeah.

**[00:21:47] JM**: Yeah. I don't know if you saw that company ScaleAPI, the data labeling as an API. But they really found a lot of traction by selling to the self-driving car companies because they needed so much data labeling.

**[00:22:00] BL**: Right.

[SPONSOR MESSAGE]

**[00:22:08] JM**: As a programmer, you think an object. With MongoDB, so does your database. MongoDB is the most popular document-based database built for modern application developers and the cloud area. Millions of developers use MongoDB to power the world's most innovative products and services, from crypto currency, to online gaming, IoT and more. Try Mongo DB today with Atlas, the global cloud database service that runs on AWS, Azure and Google Cloud. Configure, deploy and connect to your database in just a few minutes. Check it out at mongodb.com/atlas. That's mongodb.com/atlas.

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

[INTERVIEW CONTINUED]

**[00:23:04] JM**: I've got a list of data infrastructure products that I want to get your perspective for how each of these fit into modern data infrastructure, or you can just say pass on any of them.

**[00:23:17] BL**: Okay.

**[00:23:16] JM**: Let's go through them. Amazon Redshift.

**[00:23:19] BL**: I hear a lot about them. I think they've made a lot of progress in terms of building out that product. But beyond that, pass.

**[00:23:33] JM**: Okay. Fair enough. Apache Arrow?

**[00:23:35] BL**: Apache Arrow. I think Wes McKinney, who you probably know, has really done a good job of pushing forward with that project. I think it's one of these projects that the outside world don't hear a lot about, but a lot of engineers are really happy with that project.

**[00:23:53] JM**: Are data science workflows mostly in Python or in Java these days?

**[00:23:59] BL**: Well, most data scientist are Python people, but the question is do –

**[00:24:06] JM**: This is driven by that Apache Arrow, just because I think it's this data interchange thing that allows you to share data between a Python runtime and a Java runtime. I think it's like the idea is you can switch between those two runtimes and use the same data format.

**[00:24:23] BL**: Yeah. I would suspect that most – Since most data scientists and machine learning people are Python first. At least in the prototyping phase, it's Python. It depends on the company how they productionize.

**[00:24:39] JM**: Is anyone doing data science in JavaScript?

**[00:24:42] BL**: Data science in JavaScript. I mean, there's machine learning in JavaScript, but I don't know many people who are. Yeah.

**[00:24:49] JM**: Google BigQuery.

**[00:24:51] BL**: Good product. I mean, I think that of the cloud products in that category, that's the one that I hear the most raving reviews about.

**[00:25:00] JM**: What is that category? Is it like unified data warehouse and data lake? Is that the idea?

**[00:25:08] BL**: I'm about to publish a post with the founders of Databrix around this topic. Basically the idea here is that – We'll have a term for it. I don't want to reveal the term for it.

**[00:25:20] JM**: Is it Delta?

**[00:25:21] BL**: Delta is a manifestation and open source. But basically the idea is here is, in the old days, most companies probably when they talk about data for ML was mostly structured data, for ML and analytics. You had data warehouses basically. Then you have the need to collect more data variety and velocity. Then you had Delta lakes. But then the Delta lake suffered from poor data quality. Basically, people just put data into the data lake.

Now I think what we're seeing a new generation of tools which combine some of the features of those two and probably add some more. But basically the main thing is that you separate compute from storage. You're able to support more data types, not just structured, but semi-structured and unstructured, including video, images, audio, right? Then you can support more diverse workloads, not just SQL. You can use Spark if you want to do something more complicated and you can use Python. You can point your favorite machine learning library.

Most importantly, you support transactions, right? So, ACID. In most companies, you'll have people working on data pipelines and touching data maybe concurrently. You want to be able to make sure that you can reason about data integrity.

Those are some of the core features, but basically that this new data management paradigm then would allow you to pull data warehousing, data science, machine learning, analytics from the same source. You don't need to move data around, because in the data lake model, because the data quality was not that good and you would have to basically pull data out of

there, create a data warehouse and do analytics and that was a complicated process for most companies.

**[00:27:26] JM**: Right. That process was everybody dumps their data, exhaust data.

**[00:27:32] BL**: Storage is cheap, right?

**[00:27:33] JM**: Storage is cheap. You just throw it on disk in a data lake that use S3 or HDFS, data lake and then –

**[00:27:43] BL**: Google fits into this category somewhat, but Google BigQuery fits into this category somewhat, but it's optimized for SQL kind of workloads. I think the Databrix platform is an example where you have all of the features I listed out so you can do BI to AI.

Then the open source files formats that you mentioned, Delta lake, which is Delta.io, Apache Iceberg and Apache Hudi can be the basis of this new data management paradigm, but you'd have to build out all of the other features I listed out.

**[00:28:25] JM**: Why is the file format relevant?

**[00:28:26] BL**: These file formats were introduced so that you can do some of the things I described. In the case of Delta lake, for example, you can do transactions.

**[00:28: 9] JM**: The problem of data quality, doesn't have to do more with the operator rather than the infrastructure that you're providing to the operator?

**[00:28:49] BL**: No, but your data probably at ingest is not going to be ready for ML. Michael Armbrust of Databrix has this good metaphor. He has the bronze datasets, silver and then gold. Each step, you're doing some refinement, but you might still be in this exactly the same data management system. You don't have to pull it out of a data lake, right? If your data management system has this capability that you can do the processing in place, then you don't have to maintain two systems and pay for additional storage.

**[00:29:24] JM**: This is not exactly a data infrastructure product, but Kubernetes. Has Kubernetes affected data engineering?

**[00:29:31] BL**: Probably every data engineer has asked for the skillset now in terms of most jobs that are out there, I would imagine, at this point in time, or at least if not yet, probably soon, right? There'll be part of the data engineering interview process to some extent.

**[00:29:52] JM**: Why? I mean, why does a data engineer need to know about Kubernetes/

**[00:29:55] BL**: Well, I don't think they need to know it in the level of detail as the DevOps person, but probably some knowledge of it, because a lot of the infrastructure relies on it.

**[00:30:07] JM**: Like for deploying machine learning models, for example?

**[00:30:10] BL**: Yeah.

**[00:30:12] JM**: Another tool that is not exactly a data infrastructure product, but I don't know if you've heard much about data science of data engineering people using it, GraphQL?

**[00:30:22] BL**: GraphQL. I've heard it come up a few times, but –

**[00:30:29] JM**: It sounds like a pass.

**[00:30:30] BL**: Yeah. It's a pass for me. Yeah.

**[00:30:33] JM**: The problem with data discovery within a large company, explain what data discovery is.

**[00:30:38] BL**: I mean, if you work in a large company, chances are there are many different data systems or even if there's this unified system like I described earlier. You may not know what data is useful and relevant to your peers even, right?

That's why we need systems that allow you to basically even navigate your data repository to show what's available, but also maybe even understand what people are using most. Also, be able to share. That's why these feature source are so popular these days, because data scientists, data engineers and ML engineers may spend a lot of time discovering the right feature for some machine learning model that they built and they want to be able to publish and share it with their peers. Because if you spent like a week figuring out what the right features are for your fraud model, maybe I can find some of those features useful for my other model.

**[00:31:44] JM**: Have you seen any data catalog products that have stood out to you?

**[00:31:48] BL**: There's a bunch. I mean, so Atlassian is a pretty good product. I guess a lot of them give you the ability to understand what people are actually using.

**[00:32:01] JM**: The datasets throughout a company that people are using.

**[00:32:03] BL**: Yeah, Atlassian, Okera are the ones that come to mind. There's another one that I forgot the name, but also focused on data privacy, so data catalogs and data privacy.

**[00:32:15] JM**: I was listening to an episode you did recently with someone from Rakuten, and the guest you had on talked about the fact that they outsource the early steps in the data engineering process. If there's some project that they want to do at Rakuten and involves data cleaning. The data discovery and the data cleaning part, they actually outsource to contractors. Then once the data – Did I mishear that?

**[00:32:44] BL**: Not at all. They have a position called data wrangler.

**[00:32:47] JM**: Data wrangler. Okay. Trying to outsource to specific –

**[00:32:50] BL**: Yeah. Let's just say at a high-level, they have three major positions. There's the data wrangler, data scientist and then data engineer, ML engineer. The data wrangler does a lot of the data prep, freeing up the time of the data scientist. Then the data scientist may build a model and then the data wrangler steps back into the project and does kind of the model testing, model validation. From their perspective, and actually it's quite actually clever and I

think other companies should think of adapting this instead. It frees up some of the time of the data scientist, but also gives a career path for this other role, because this other role overtime gets mentored by the data scientist and might become a data scientist. In fact, in that episode, he said that they just promoted some data wranglers into data scientists where he actually went and traveled to where they were located to tell them in-person. So it's a big deal.

**[00:33:54] JM**: A data wrangler does not need to be that technical.

**[00:33:57] BL**: In the beginning, yeah. Not familiar with ML, right? Then they can train them in-house and then that allows them to grow their team and then provide this career path.

**[00:34:11] JM**: There are a number of approaches to building data infrastructure for doing machine learning. There are some enterprises that adapt a collection of data infrastructure tools, like Kafka, Hadoop, Spark, whatever.

**[00:34:28] BL**: Stitch your own.

**[00:34:29] JM**: Stitch your own. Then there are these end-to-end machine learning platforms, like you can buy a large end-to-end machine learning platform from certain providers. How would you contrast these two approaches?

**[00:34:44] BL**: Yeah. I mean, I think it really depends on your priorities, right?

**[00:34:48] JM**: Also, who are the big end-to-end machine learning platforms?

**[00:34:52] BL**: I mean, the cloud providers probably will say they are, and then Databrix. Yeah. I guess it really – Stitching your own. I mean, if you have multiple open source projects, it's not going to be easy for a regular enterprise, because take just one of those systems, Kafka, just being good at using it and operating it, administering it, as supposed to buying an end-to-end platform where you can just focus on solving problems for your business units.

Then, I mean, that's just Kafka. Then you've got Spark. You got to tune all of these things. I think overtime most – I mean, I don't know if we'll ever – Jeff, we'll ever get to the point where

the tech companies will go to vendors, because like I said, they may be cutting edge. Also, it's just not the way these startups evolve. Because when you're a startup, you go, "Okay. We need to do this." Well, there are these open source projects. They try to stitch their own. Then before you know it, I guess they can move to Databrix.

I mean, if you were a startup and you wanted to move fast and you know you needed to do ML, that seems to make sense, and then focus on your business problem, right?

**[00:36:12] JM**: Exactly.

**[00:36:12] BL**: But that's not – The normal route for the startup is not that, because there's usually a technical founder.

**[00:36:19] JM**: It's going to become the norm. I mean, you look at Netflix. Netflix moved to the cloud. It was massive success. At first it looked crazy. When they first moved to the cloud, people thought they were crazy but then they realized, "Oh, it doesn't make any sense for Netflix to be in the data center business." Overtime, it's going to look ridiculous for a lot of these companies to be in the open source infrastructure business.

**[00:36:39] BL**: Yeah. That's the tech world. Regular enterprise, stitching together these things. Very few of them will do that. There are a few companies that probably you hear a lot about that try to experiment and do a lot of things on their own, like Capital One and Two Sigma to some extent. But I don't know if you can name 10 of those companies, right?

**[00:37:01] JM**: Right.

**[00:37:03] BL**: Yeah. I mean, you can just see also from the growth of Databrix how much this type of solution resonates with enterprise who cannot afford to hire armies and armies of engineers to build and maintain infrastructure.

By the way, for your listeners, I should disclose that I am now spending 50% of my time at Databrix and the remainder of my time, I spend some of it at Anyscale and other side projects

where I help friends, but also I have, now, I guess weekly podcast. You can find it at dataexchange.media.

**[00:37:44] JM**: Why, by the way, do you like to be involved in so many different things? Why not just focus on one specific thing?

**[00:37:52] BL**: I think that I just want to be able to see a wide variety of problems across many different stages of maturity of the company, but also, as you know, I spent over 10 years at O'Reilly and I left in November last year. In there, I shared 8 large conferences. I've always kind of had a broad perspective. I want to be able to see if I can maintain that as much as possible.

**[00:38:22] JM**: Definitely. I can certainly relate. We talked a little bit about interactive data science earlier. More recently, Jupyter Notebooks have made an improvement in interactive data science. Describe how Jupyter Notebooks have changed the data science workflows within an enterprise.

**[00:38:40] BL**: Well, it's basically also – I mean, I think Python benefits a lot of the improvements in Jupyter, because now most data scientists are trained in Python, which means that when companies hired data scientists, they're probably Python first. It helps that people who got trained in Python can use somewhat similar tools when they get to companies. I mean, I think that for people who got trained using Jupyter Notebooks, the fact that you can have your visualizations in there, your code and your documentation is really beneficial. The question is, is that how ultimately people will productionize these things. It seems like people are trying to think through what would be the workflow to streamline prototype to production and what would the role of notebooks be in that world. I think that overtime maybe that will blur a little bit just because companies want tools that will allow them to act fast. You don't want to write a complicated data pipeline in a notebook only to have to hand it to the production team who have to rewrite it into something else, right?

On the other hand, I am somewhat not sure the whole notion that data scientist can just push a button and deploy to production. I don't know if that's going to happen.

**[00:40:10] JM**: Why now?

**[00:40:11] BL**: Well, because I mean there's all sorts of regulatory reasons why you want to make sure that something is up-to-snuff. In the financial services sector, they have well-laid our organizational structures for compliance and things. I mean, even predating machine learning, just statistical model. They'd have people who would review the source code. People would test it. Yeah, make sure it's not touching a data that's not supposed to and so on and so forth. I guess if you can automate all of these tests, make sure that everything is up-to-snuff, then maybe it's possible. But I don't know if we're going to get there soon.

Also, who's ultimately responsible if after you deploy a model, it starts misbehaving? Who's going to wear the proverbial pager? Ultimately, some kind of ops and production team will be responsible. Then, if that's the case, then they want to know what is it that we're putting out there.

**[00:41:18] JM**: Yeah, it's the old throwing it over the wall problem for somebody else to take care off.

**[00:41:22] BL**: Maybe you can automate some of the tests and things like that, but you're probably still going to have a dedicated team to make sure that everything is ready. Also, I mean, which also brings me to something I've given talks about over the last two years, which is the whole notion of managing risk and ML, right?

I mean, the other way to frame it is responsible AI, but I like managing risk because some companies already have risk management teams and risk officers. So it's well-understood. What goes into this pile? Well, safety and reliability, privacy and security, fairness and bias, explainability. You have to make sure that depending on your use case, that you have to make sure that these are all accounted for, which at this point in time I think that means actually even having cross-disciplinary teams.

We're early on in this age of privacy preserving ML, but right now the tools still require some level of expertise in computer security topics, like encryption, and cryptography, and things like that, which are not topics that a data scientist knows about or frankly would even want to read about. You would have to set up teams that can work across these different functions, which

means your end-to-end platform should probably appeal to different personas, not just data people.

**[00:42:59] JM**: I mean, if you're at Netflix and you're rolling out a new recommendation system for movies, you can kind of ignore a lot of these questions. It doesn't really matter. But for the companies you've talked to, financial services, offering loans, maybe mapping software, stuff that's like super mission-critical.

**[00:43:18] BL**: Healthcare.

**[00:43:19] JM**: Healthcare. What are people doing? Are they even rolling out machine learning models or are they just kind of too afraid to?

**[00:43:26] BL**: No. No. They are, but they still have more processes in place than –

**[00:43:32] JM**: What does that look like? What are those processes look like?

**[00:43:34] BL**: I think the emerging term that people are using is model governance. They have basically that function for model governance to make sure that the machine has been tested, validated, that it's in compliance with whatever regulations that need to be accounted for, and that reproducibility is important for many of these domains, because if something goes wrong, they may have to explain to the regulator what happened. A lot of those have to be in place. It's not as simple as just rolling out a recommendation engine, which if things don't work out, then it's you the company that suffers, right? Not your users.

**[00:44:14] JM**: Right. This governance team, does it look like an old school QA process?

**[00:44:22] BL**: In the governance team, there are also people from compliance. Yeah, they'll be the equivalent of teams who are dedicated, model validators, investors probably. I mean, I think for most people, the best practice there is to separate that from the people who build the model. Let someone else test and validate the model.

[SPONSOR MESSAGE]

**[00:44:51] JM**: Today's show is sponsored by Datadog, a monitoring and analytics platform that integrates with more than 250 technologies including AWS, Kubernetes and Lambda. Datadog unites metrics, traces and logs in one platform so that you can get full visibility into your infrastructure and your applications. Check out new features like trace search and analytics for rapid insights into high-cardinality data, and Watchdog, an auto-detection engine that alerts you to performance anomalies across your applications.

Datadog makes it easy for teams to monitor every layer of their stack in one place. But don't take our word for it, you can start a free trial today and Datadog will send you a t-shift for free at softwareengineeringdaily.com/datadog. To get that t-shirt and your free Datadog trial, go to softwareengineeringdaily.com/datadog.

[INTERVIEW CONTINUED]

**[00:45:55] JM**: In the early days of Hadoop, there was this common problem where you have a business analyst and the business analyst goes to the Hadoop team and says, "Hey, I've got this query. Can you go run it for me?" Then it takes like 5 days and then they get their answer back. These lines of requests will build up for the Hadoop engineers.

There was this dream of self-serve data science or self-serve analytics for the business analyst. Then there was all these tools like Tableau, and Looker, and things like that. Have we reached a point where we have self-serve data science or self-serve business analytics? How much have things improved since the early Hadoop days?

**[00:46:46] BL**: I think they have improved quite a bit, but if you're asking if – The analyst we're doing the things that you described. Assuming that the data is ready to go, I mean, we have the tools for them to start doing interactive analysis, right? But the question is if there's some data munging and preparation involved. Some of these tools are also incorporated that capability.

That also goes back to that earlier new data management paradigm I described, which is if you still require that people make a copy, get it out of the data lake, put it in a schema, then of

course you still need IT for that, right? Most analysts won't be able to do that, but if the data is sufficiently ready to go with minor data wrangling involved, I think we're starting to have tools.

I think the area I'm excited about, which is quite early, is the use of ML for visual discovery, so intelligent visual discovery. The idea here is if you have so much data fields and columns and you don't even know what chart to draw. We're starting to see researchers and tool builders build tools so that maybe you get the presentation of the charts and things that you need to keep track off rather than you have to spend hours and hours.

For example, if you're a marketing analyst and you want to understand what's driving churn, well, maybe you can load all of the data in Tableau or Looker or whatever tool, but there might be – I don't know, like 250 columns. It could take you a while. What if you just use ML to weigh through that quickly? We're starting to see efforts around these lines.

**[00:48:41] JM**: As you've mentioned, you are doing a lot of media at this point.

**[00:48:49] BL**: Not a lot, man.

**[00:48:49] JM**: Not a lot. Okay.

**[00:48:51] BL**: One podcast a week that I do on the side.

**[00:48:54] JM**: Podcast plus newsletter, the data exchange. Explain what you're trying to accomplish with the data exchange. What are the areas that you're wanting to explore?

**[00:49:04] BL**: I said the same areas that I explored in the O'Reilly data show, data, machine learning and AI. Basically, the idea here is to ultimately provide resources so that people can make decisions around these three topics. Right now, my collaborator, Mikio Braun and I are starting with a podcast, but overtime, who knows? We might add other things.

**[00:49:29] JM**: What's your perspective on how software media is going to change in the next 5 years? I mean, you spent 10 years at O'Reilly. Now you're embarking on your own goals. What changes do you think are coming soon?

**[00:49:44] BL**: I mean, I think that podcasts are going to be important, I think, more so than people realize, I think. Part of it is just people have these mobile devices. The consumption patterns will change. Newsletters and podcasts are going to be more important. Curation is going to be important because we're drowning in information.

I think that the role of social media, I don't have a good feel for how that's going to play out, because I think the platforms obviously are under the gun right now for what happened in 2016. Who knows that will affect the way media virally spreads overtime.

**[00:50:29] JM**: You think podcasting is a durable format?

**[00:50:31] BL**: Yeah, for someone like me who doesn't have to feed myself using a podcast. Yes. I mean, Jeff, you would know this better. If you look at media companies, most of them are doing podcasts.

**[00:50:48] JM**: Most of them pivoted to video 5 years ago.

**[00:50:51] BL**: Yeah, but I think podcasts are much more realistic, because you can consume it while you're driving, while you're doing other things, multitask. Video, you have to actually watch, right? I mean, as someone who spent many, many years sharing conferences, I can tell you, with a podcast, you can reach probably more people than you would if you spoke at a conference.

**[00:51:14] JM**: That's music to my ears.

**[00:51:16] BL**: Just be a guest, Jeff in my podcast.

**[00:51:21] JM**: Indeed. Getting back to machine learning, how has AlphaGo affected the world of machine learning?

**[00:51:29] BL**: How has AlphaGo – I mean, I think it raised awareness about reinforcement learning is my answer. I mean, I think obviously it got people excited about the potential of

systems being able to learn by itself to some extent. But I think as far as specific tools and techniques, I think RL is the one that benefited.

**[00:51:53] JM**: Is reinforcement learning more difficult to implement than supervised learning?

**[00:51:57] BL**: Yes. Yeah. Yeah. Yeah. But I have a post coming up. I think Anycase will publish it about RL in enterprise. There, I'm not saying that RL is without problems, including it's more challenging than supervised learning. But in the post I cite two major areas where companies have revealed that they're using RL. The first one is recommenders and personalization. In the post I think I listed four companies; Netflix, YouTube, JD.com and Facebook who have revealed that they're using RL as one of the things that are part of the recommenders or personalization systems. It's not like they threw everything out. They're adding RL.

The challenge there is that while these companies have either published papers, given talks, or in some cases even published source code, there're still no tutorials and detailed for how to do this on your own. But I think that's going to come.

Then the other area that is probably less familiar to your listeners and data scientist is this area called simulation modeling. In turns out, there's a lot of software that companies now use to model different scenarios. Your factor floor, your retail store, your logistics supply chain.

Now, if you look at the software, some of it even has 2D modeling that looks like a game. Once you see this software you realize right away, "Oh! I get it. I see why reinforcement learning might play a role in this setting." Now we're starting to see some of these software providers of simulation modeling work with RL companies. There's a startup here in San Francisco called Pathmind, which basically took Ray RLlib and is using RLlib and integrating that into some of these more well-known simulation software, which allows the simulation software vendor then to run more complicated scenarios using reinforcement learning and their users won't even have to know that reinforcement learning is happening in the background.

**[00:54:20] JM**: Actually, I just was thinking about the self-driving car discussion earlier. I remember there's this big Atlantic piece a while ago about Waymo and so much of what they discussed in that piece was about the fact that Waymo has all these infrastructure around

simulation and they do these large scale simulations that are based off of real-world driving so that they can simulate a lot of iterations at a fast pace.

But my understanding is that simulation also fits into on-the-fly decision making, because basically in many cases these models, they're interacting with real-world situations and this comes back also the C++ Ray implementation, because you have a situation where maybe a drone needs to make a decision about where to go and if it can simulate the future on-the-fly really, really quickly, then it can make more education decision.

**[00:55:18] BL**: Yeah. I mean, as I noted, there are two main buckets that I think RL might start appearing in the enterprise. Simulation modeling and then recommenders and personalization, and they share a couple of things. First is maybe the reason you might use RL instead of supervised in this setting is, one, maybe you can't get labeled data or labeled data is expensive. Then secondly, there are certain applications where it's not like a singular moment and then someone decided to churn. It may be a sequence of events. It's almost like you have this notion of memory and that's where reinforcement learning excels, is in the sequential decision making. Then a bunch of little things happen on the website and then, "Boom!" someone decides to churn or whatever. There's no labeled data that you can get for that kind of thing, because it's not like you can say, "The reason this person churned is because of this event." No. It might have been based on a sequence of things.

By the way, there's one more topic I think your listeners might be interested in that I wanted to talk to you about, which is this whole area of machine-aided programming. I don't know if you – It's the use of machine learning to help programmers.

**[00:56:43] JM**: Oh! I hadn't heard of an application of this. Any – There's this thing called Kite. I remember Kite a while ago is something like this.

**[00:56:50] BL**: A couple of areas. One; program synthesis. The Ray Project has worked with another group in Berkley and they developed tools around program synthesis. The first manifestation is this tool called AutoPandas. Imagine a data scientist has many, many APIs that they have to keep track of. One of those APIs is Pandas. What AutoPandas does is the following. Here's my starting data frame. Here's what I want it to look like. AutoPandas will write

automatically the Pandas code to go from input data frame to output data frame, the most efficient Pandas code.

The idea here is that that same technique can be applied to a bunch of other APIs that a data scientist might be interested in. You can imagine in the future a data scientist may not need to know that details of every single tool that they need to use, because the Pandas API – I don't know, has hundreds of functions, right? Then programming becomes more like I want to write this program. I know I have these 20 JavaScript frameworks that are involved, but now I have these tools that can help me. That's one manifestation.

Then there's other kind of more –

**[00:58:14] JM**: Real quick. Have there been any practical applications of program synthesis?

**[00:58:23] BL**: AutoPandas is the first one that I've seen that looks promising, because basically a lot of people use Pandas.

**[00:58:30] JM**: Right. But this is still pretty much in very nascent stages, right? There're not people actually like deploying it and using it at companies.

**[00:58:36] BL**: Not yet. Not yet. Yeah. But the combination of deep learning and Ray allows them to be able to do it for Pandas. I believe they can do it for other APIs. The other areas where machine-aided programming beginning to appear is to auto-completion. This hasn't happened yet, but –

**[00:58:56] JM**: You mean in the – What is it called? Like your IDE.

**[00:59:00] BL**: Yeah.

**[00:59:02] JM**: Right.

**[00:59:03] BL**: Then what they want to do is basically use some ML techniques to analyze code repos and see what kinds of tools come out of – Because now we have all these new natural

language models. They want to see if some of that can be used to develop tools for programmers.

**[00:59:26] JM**: That seems plausible.

**[00:59:27] BL**: I think it's an exciting area and it could change the nature of programming.

**[00:59:34] JM**: Totally. I mean, that's a great way to avoid a lot of syntax errors.

**[00:59:38] BL**: Yeah. I mean, also, if you can imagine, like I said, if you're a web developer, you have – Whatever, 50 JavaScript frameworks that you have to keep track off, and now suddenly you may not need to keep track of them at all in too much detail.

**[00:59:55] JM**: Yeah.

**[00:59:55] BL**: That's somewhat future-looking, but since this is Software Engineering Daily.

**[01:00:01] JM**: Totally. Any other domains that are in your mind? I mean, I've listened to a bunch of your episodes and one of the episodes I really liked was one where you just explored the kinds of trends that you were looking forward to in 2020. There's further exploration of some of these things in that episode. But is there anything else that's on your mind?

**[01:00:21] BL**: Generally, I mean I think that there's still a lot of innovation that's happening at the moment. Across the entire data pipeline, including this new data management paradigm that I described, because I think for the most part, the only things that get written about these days are ML, say, in the popular media, right? As you know, a lot of the actual challenges are in infra and data engineering.

**[01:00:50] JM**: That's right. Yeah. Just a few more questions on the issues and ecosystem. Do you have a sense of how the PyTorch community compares to the TensorFlow community?

**[01:01:02] BL**: I think in the research world, PyTorch is ascendant. In the enterprise, I think TensorFlow still has a lot more users just because it came out earlier and there's a lot more

enterprise software vendors that rallied around it. But who knows? That may change. Just like – I mean, who would have imagined that Python in the enterprise would be this popular, right? I mean, I think if people are doing research with PyTorch, that tells me that the professors are using PyTorch, which means the classes are taught in PyTorch or more of them. It's not like no one is doing research or teaching in TensorFlow. It's just a matter of time then that enterprise tools for PyTorch become better. Maybe not completely on par with TensorFlow, because TensorFlow will always have a Google and Google Cloud. I think that we'll see more and more people use PyTorch.

I mean, for the most part now, if you talk to any enterprise software vendor in ML, they make a point in mentioning PyTorch. They'll say, "Yeah, we support TensorFlow and PyTorch." They're checking that box now. Now the question is are the tools that they're building for PyTorch on par with TensorFlow?

**[01:02:21] JM**: If you are running a cloud provider, would you put any restriction on the use of facial recognition APIs?

**[01:02:27] BL**: I think there's still a lot of discussions. So, yes. For the moment, yeah, I would. There's still a lot of discussion about the applications of facial recognition. I think there's some New York Times article that came out recently of a startup that –

**[01:02:44] JM**: I saw that. Yeah.

**[01:02:46] BL**: That is working mostly with law enforcement, but that's – I mean, once you're selling to law enforcement, what prevents you from selling to anyone?

In the west, in particular, I think there's going to be a lot of pushback with facial recognition. Look, I mean, we just went through GDPR and CCPA. Then you think that facial recognition can just deployed willy-nilly in mass? No. I don't think so.

**[01:03:14] JM**: Yeah, except for face ID.

**[01:03:17] BL**: I mean, you can imagine. If it's facial recognition but you are trying to comply with GDPR, does that mean if I send you an email saying scrub me from your facial database. You will be able to execute a MapReduce or Spark job to do that?

**[01:03:36] JM**: Certainly, I wouldn't want to have to answer that question.

**[01:03:38] BL**: Yeah.

**[01:03:39] JM**: You visited Israel recently.

**[01:03:41] BL**: Yeah. I go there every year. Yeah.

**[01:03:43] JM**: How do Israeli software engineers think different than American software engineers?

**[01:03:47] BL**: I don't know if it's different, but they're scrappy, because the population of the country is small. I think it's 9 million. They know that if you're doing a startup in Israel, you know that you need to target the U.S. market right away. They're just scrappier that way. Also, I think the community is smaller, so people tend to know each other. So the degrees of separation is quite low, but they're also helpful to each other. As you know, probably just tradition in science and math is just perfect for this age of data and machine learning.

**[01:04:25] JM**: Last question. If you had to start a company in the data infrastructure space today, what company would you start?

**[01:04:33] BL**: What company would I start? I don't know. I mean, if I knew, I would probably started. No. Actually, yeah, I have some ideas. But I mean I think I would probably be more vertical focused. Yeah. I mean, the challenge of course of a vertical is that if you go all-in the vertical and it's wrong, that's one challenge. Number two, you have to find a vertical you actually have passion about, because it's hard to – Startups are hard enough if you're not – Let's say you decide I want to build something in sports and you really don't like sports. That's going to be a hard slog.

**[01:05:11] JM**: Totally.

**[01:05:12] BL**: Yeah. What company would I start? I don't know actually. Yeah. I mean, I think RL would be interesting, but it's early. In terms of explaining to enterprise, you have to hide it, just like the simulations modeling software will ultimately have RL and you may not even know.

**[01:05:32] JM**: Ben Lorica, thanks for coming back on. It's been great talking.

**[01:05:34] BL**: Thank you.

[END OF INTERVIEW]

**[01:05:45] JM**: You probably do not enjoy searching for a job. Engineers don't like sacrificing their time to do phone screens, and we don't like doing whiteboard problems and working on tedious take home projects. Everyone knows the software hiring process is not perfect. But what's the alternative? Triplebyte is the alternative.

Triplebyte is a platform for finding a great software job faster. Triplebyte works with 400+ tech companies, including Dropbox, Adobe, Coursera and Cruise Automation. Triplebyte improves the hiring process by saving you time and fast-tracking you to final interviews. At triplebyte.com/sedaily, you can start your process by taking a quiz, and after the quiz you get interviewed by Triplebyte if you pass that quiz. If you pass that interview, you make it straight to multiple onsite interviews. If you take a job, you get an additional $1,000 signing bonus from Triplebyte because you use the link triplebyte.com/sedaily.

That $1,000 is nice, but you might be making much more since those multiple onsite interviews would put you in a great position to potentially get multiple offers, and then you could figure out what your salary actually should be. Triplebyte does not look at candidate's backgrounds, like resumes and where they've worked and where they went to school. Triplebyte only cares about whether someone can code. So I'm a huge fan of that aspect of their model. This means that they work with lots of people from nontraditional and unusual backgrounds.

To get started, just go to triplebyte.com/sedaily and take a quiz to get started. There's very little risk and you might find yourself in a great position getting multiple onsite interviews from just one quiz and a Triplebyte interview. Go to triplebyte.com/sedaily to try it out.

Thank you to Triplebyte.

[END]