EPISODE 796

[INTRODUCTION]

[0:00:00.3] JM: Data engineering touches every area of an organization. Engineers need a data platform to build search indexes and microservices. Data scientists need data pipelines to build machine learning models. Business analysts need flexible dashboards to understand the trends and customer uses of a product.

Max Beauchemin is a data engineer who has worked at Airbnb, Lyft and Facebook. He's the creator of two successful open source projects; Apache Airflow and Apache Superset. In a previous show, Max discussed data engineering at Airbnb and the usage of Airflow. In today's show, Max discusses the engineering of Apache Superset. Superset is an open source business intelligence web application. Superset allows users to create visualizations, slice and dice their data and query it. Superset integrates with the Druid, a database that supports exploratory, OLAP style workloads.

One reason that Superset is distinctive is that it is a full open-source application. Many open source projects are tools like databases, command-line tools and web frameworks. Superset is an open source application that can be used by individuals who are not developers, so the potential audience is wider than the typical open source tool built for engineers.

Max joins the show to talk about his experience as a data engineer at Airbnb and Lyft and the open source projects that he has started and led. A few events that we have coming up, we have a fireside conversation with Haseeb Qureshi at CloudFlare on April 3rd, 2019. We also have a hackathon at App Academy on April 6th, 2019. These can be found at softwareengineeringdaily.com/meetup for the April 3rd meetup and softwareengineeringdaily.com/hackathon for the hackathon. The hackathon is for find collabs, a product that I've been building that allows you to find collaborators for your open-source projects, your other projects, artistic projects, musical projects.

The hackathon is also virtual. You can find online collaborators to enter into the hackathon, or you can attend the App Academy in-person hackathon. In any case, the prize purse for the

hackathon is $5,000, so you're not competing for nothing. You can enter by competing with any project, any cool project where you're looking for collaborators. We'd love to have you involved, so go to softwareengineeringdaily.com/hackathon, or findcollabs.com/hackathon. Let's get on with the episode.

[SPONSOR MESSAGE]

[0:02:59.3] JM: Kubernetes can be difficult. Container networking, storage, disaster recovery, these are issues that you would rather not have to figure out alone. Mesosphere's Kubernetes-as-a-Service provides single-click Kubernetes deployment with simple management, security features and high availability, to make your Kubernetes deployments easy.

You can find out more about Mesosphere's Kubernetes-as-a-Service by going to softwareengineeringdaily.com/mesosphere. Mesosphere's Kubernetes-as-a-Service heals itself when it detects a problem with the state of the cluster, so you don't have to worry about your cluster going down. They make it easy to install monitoring and logging and other tooling alongside your Kubernetes cluster.

With one-click install, there's additional tooling like Prometheus, Linkerd, Jenkins and any of the services in the service catalog. Mesosphere is built to make multi-cloud, hybrid cloud and edge computing easier. To find out how Mesosphere's Kubernetes-as-a-Service can help you easily deploy Kubernetes, you can check out softwareengineeringdaily.com/mesosphere, and it would support Software Engineering Daily as well.

One reason I am a big fan of Mesosphere is that one of the founders, Ben Hindman, is one of the first people I interviewed about software engineering back when I was a host on Software Engineering Radio. He was so good and so generous with his explanations of various distributed systems concepts. This was back four, or five years ago when some of the applied distributed systems material was a little more scant in the marketplace. It was harder to find information about distributed systems in production, and he was one of the people that was evangelizing and talking about it and obviously building it in Apache Mesos.

I'm really happy to have Mesosphere as a sponsor. If you want to check out Mesosphere and support Software Engineering Daily, go to softwareengineeringdaily.com/mesosphere.

[INTERVIEW]

[0:05:18.2] JM: Max Beauchemin, welcome to Software Engineering Daily.

[0:05:22.1] MB: Thank you so much to have me on the show again.

[0:05:24.9] JM: Yes. Our last episode was about Airflow and your work at Airbnb. You are the creator of Airflow, you're also the creator of Apache Superset, which we're going to talk about in a little more detail today. You've worked in engineering at Lyft and Airbnb. Your background is at the high-scale technology companies building data infrastructure, data visualization tools. Superset is this open source BI tool that was originally created when you were at Airbnb. Why did you create Superset?

[0:06:01.1] MB: Right. Yeah, so for context, I started probably working with data about 15 to 20 years ago. I've been working, using all sorts of data tools, data platforms and yeah, more recently, I worked at Facebook, Lyft, Airbnb where as part of the data tools team building all sorts of data tools. The history or the genesis of Superset is really it came from a place where at the time at Airbnb, we were running a Druid cluster, a POC on a Druid cluster. For hackathon, I decided that would be neat to just create a small tool to slice and dice the data that was in Druid.

At the time, Apache Druid did not have any front-end for it. It was assumed that you'd probably write your own front-end. For hackathon, I just build this tool. That's how it all started. Quickly after this, decided to make this work, to make the hackathon tool basically work with Presto as well and with other databases that we use internally and that grew from that point.

[0:07:05.0] JM: There are lots of different people within an organization that could potentially use a business intelligence tool, like a Superset. Was there a type of user that you were optimizing for when you started working on Superset?

[0:07:20.5] MB: Originally, I think we were just targeting at anyone who needs to explore data at Airbnb, right? We had these great data sets and Druid and we wanted for people to just create their own visualization and result sets out of it. I think it catered naturally early on to analysts and data scientists and engineer, but the tooling is so fast and easy, or the dev tooling we were building is so accessible that was more about which data sets are in the database that this is more what would dictate who would use it at the time.

[0:07:57.8] JM: There are so many BI tools in the world. Most of them are closed source. There are open source options. I'm not familiar with all of the open source options, but I know there are at least primitives for building BI tools, like JavaScript components, like D3.js, the charting library for visualizing data. When you started working on Superset, what was the landscape of available tools and why did you need to build your own new one when there's already so many BI tools out there?

[0:08:35.9] MB: Right. There are a lot of open source libraries out there and especially in the data visualization space, one thing that's a lot less common I think in the open source space is just these full-on products, right? It's very common to have building blocks that are open source libraries and things like that and I think it's more rare to see a full-on product that caters to an array of users, or something that you would run in production as a service.

I think there's also a lot of infrastructure that's open-source, but we don't see, I think Airflow to a certain extent. Superset are a little bit of an outlier as there's not that many of these full-on product, open-source solutions. The reason why, or one of the advantages that we have working with open source, or having a product that's open source is that we can shape it to be exactly what we need it to be, which is very different than when using vendor products.

Clearly, one edge that Superset had or has at Airbnb and at Lyft over the vendor tools that they use like Tableau, Locker, things like, Mode is that we can make it exactly what we need to be and we can integrate it with the databases and the solutions that we have internally. More directly while I was at Airbnb and part of the reason why Superset took off internally as an internal tool was that Tableau did not work very well at the time with Presto and Druid, which were our databases of choice.

Then we come up with this open source tool that we can make work with all of the tooling and systems in our infrastructure. Another example of that is integration in regards to data access policy, or security, or authentication. We have full control over how this works, because it's open source, because we can alter and modify the code.

[0:10:33.7] JM: You touched on something there that what makes Superset unique in the category of open source BI tools is that it feels like a fully-fledged product. In the open source world, there is a wide range of types of software. There's command-line tools that are barely formed and barely have any documentation and you pull it from a repo and you have to figure out how it works. There's back-end tools that have more documentation, but they're still hard to use by their very nature. Then there are open source schools where there is so much documentation, there's such a good onboarding experience that it really opens up the tool to a wide range of people who can use it.

One shining example that comes to mind is WordPress. Everybody can use WordPress, whether you want to hack on WordPress itself, or you want to be somebody who just wants to run a blog, or wants to run an e-commerce site. Why is it so important to have a highly accessible, fully formed product around that open source tool that you're building?

[0:11:43.7] MB: Well, I'm trying to think like first, there's not a lot of examples of these open source products, I think. It's an uncommon thing. The question is why is it important to have something that's well-formed?

[0:11:58.7] JM: Yeah, exactly.

[0:11:59.9] MB: Or why creating something like this?

[0:12:01.6] JM: Exactly. I think it's underrated. I think it's underrated what WordPress has done, what you're doing with Superset, the idea that you're not just creating this one-off random tool that maybe some people install, but it's a really fully baked thing that is open source.

[0:12:20.0] MB: Let me give this a shot. I think the reason why open source in general is building blocks is that it's just easier for someone to approach it this way, right? To say, maybe

I'm working on a piece of software, or I'm working on a platform or something that's internal and I need the library and I'm just going to go ahead and create a building block for myself. In this building block, I can share with the world.

I think it's much more involved to go and create a full-on product that works for everyone, right? There's definitely a challenge in writing a piece of software that's going to work in multiple environments, that's going to integrate with their infrastructure. I think logically, it just makes sense for people to share building blocks as opposed to sharing full-on pieces of software.

When you manage to put something together that is a full-on product, then of course, it's attractive for organization and it's a full-on solution that competes with potentially event vendor products. That becomes an interesting proposal for people to use and run. I would say it's just more complex and harder to do, but then the bounty, or the value proposition is also higher.

[0:13:39.2] JM: When I think about the trends in data engineering and consumption of BI tools on the other end of that data engineering, there are all these layers between the point of data creation and the point at which a user is consuming that data on a dashboard in front of the screen. This has become more complex with the rise of importance in streaming data, because before you might have – in the days of batch, you might have been able to get all your data and then you batch it into a visualization system.

Now we have systems that are rapidly updating. If you're a business analyst at Lyft, or at Airbnb, you want to have some dashboards that are updating in real-time. How has the movement from batch processing and batch data to stream processing, how has that affected the front end, the dashboarding BI side of the world?

[0:14:45.9] MB: I think there's definitely a movement there. I've seen a lot more streaming computation over the past year and especially because I was that Lyft, and Lyft is such a real-time business, right? It's a timely thing. It's heavily operational. We need to make sure that everything is working well right this minute, right this moment. Part of it is observed at Lyft.

I think if you look at Airbnb in context, there are still growing needs for real-time data, but I would say maybe not as much just by the nature of their business, right? I think in general, we do see

a movement towards more real-time data. I think some people have been arguing that ultimately, everything is an unbound data set and all computation should be operating on streams. I think that's a little bit controversial. Sometimes you just don't need that, right? If you're doing a growth and engagement type of analysis, you might not need to know exactly what happened in the last minute. If you're looking at more operational types of data sets, it becomes really important to know what's happening right now.

I've been with people on this that saying that what we need from a real-time perspective often is not exactly the same dimensions and metrics as what we need in the batch world, which I think there's some common dimensions there and there's some common metrics. Because the very nature of that, sometimes it does make sense to have different pipelines altogether on the real-time front and on the batch front.

One thing too that we've seen that that's driving change there is that the tooling has gotten so much better, right? With much better computation frameworks, like Flank and Spark streaming, with Kafka becoming so prevalent and solid and with databases, like Druid or Pinot, it's becoming a lot more possible than it used to be to actually do this.

Now how does that impact data visualization, right? I think in data is we've seen probably more and more of these streaming charts, though I think it's not necessarily super important to have your chart tick and move to the left a little bit at every second, but that's definitely a trend to have these visualizations that are showing what's been happening in the hour.

It's not new to have operational datasets, but traditionally these things have been served in time – off of time series database that typically don't offer a lot of dimensionality, or slice and dicing by their very nature. More and more, we see the capability of being able to slice and dice on real-time data, which I think is really interesting.

[0:17:29.5] JM: You mentioned this a little bit earlier that one motivation for creating your own BI tool was to be able to connect to whatever database you wanted to. Why was that a constraint on you prior to having Superset? Why was the constraint of being able to connect to certain databases a limiting factor?

[0:17:51.2] MB: At Airbnb at the time where Superset was born, we had chosen these databases. We were in the process of choosing what has become now Apache Druid and we had selected and made a huge investment in Presto and off of Redshift and onto using Presto and Hive. These databases were not very well supported in Tableau, which was our main data visualization tool at the time. There was just no ODBC driver, or any sorts of driver we could use from Tableau and into Presto.

Druid was a NoSQL database at the time. Since then, it has evolved to support SQL as well. There was just no way Tableau would ever probably integrate with Druid. Those were the databases that we had selected and that we wanted to work with on a daily basis. We're just at the mercy of these vendors to make their tools work inside our infrastructure properly. That was the driver originally. I think over time, that has become better so more and more tools are connecting to more databases now, but that was an original driver. I think the motives have changed since then. The reason why Superset is relevant and interesting today have changed and have evolved too.

[0:19:08.4] JM: Superset does integrate with Druid and we did a show about Druid recently. Why is Druid an interesting database?

[0:19:18.4] MB: Druid's got a lot of interesting property. I'm an ex-Facebook, so I spent a few years working at Facebook and there was this really interesting database plus frontend and that was tightly coupled with this database called Scuba. Scuba is this real-time – it's often sampled, sometimes probabilistic or best-effort database, right? With the guarantee of serving queries in less than a second or so. I think for Scuba, 90% plus of the query would take less than a second and often, just milliseconds.

There's a prevalence of real-time, which we touched upon and I think real-time in some cases typically operational use cases is really important. Now slice and dice is also important. You need to be able to do these analysis where you can group by different things, filters on different dimensions, right? Explore cubes of data, or explore multi-dimensional data sets.

Then I think this latency, I guess there's data freshness and data latency, but in both cases, Druid much like Scuba, or much like Pinot, which is a similar database in this space, offer really

good freshness and really good latency. I think the thing that's really transformative to me is if you can run queries and always get an answer in less than a second or less than a few seconds, it completely changes the way you interact with data and the number of questions you're going to be able to ask.

A comparison I often make is if a Google search would take, say 30 seconds, or 60 seconds, or an hour, how would it impact the impact of Google itself on the world and the usage of a system like Google? I think it would completely change the way people interact with it and the prevalence of the tool itself. Having these databases that guarantee very fast results is completely transformative to an analyst. Of course, there's concessions there, right? You cannot have just super speed along with everything else that more – perhaps more slower databases are able to guarantee.

The tradeoff there is okay, sometimes you use sampling, sometimes you do probabilistic, you use sketches to get probabilistic results and that's good enough. Sometimes you give up on being able to do joins, or sub-queries and things like that, all at the benefit of getting instant answers to your questions.

[0:21:50.4] JM: Druid is a good example of a database that is built for specific domains, specific type of queries. Can you go a little bit deeper on what are some examples of query types, or use cases where people need Druid?

[0:22:11.5] MB: Right. I would say that the main drivers for using Druid are the context in which you would select this database is where you care about data freshness, right? It's real-timeness, and so this measures is the time to event to the time of – the time from the event to the time at which you can consume and analyze this data. With the database like Druid is often counted in seconds. Then the query latency, which is from the moment where you ask your question how long is it going to take for you to get an answer?

For use cases where these things are important, I think using a database like Druid is becoming more and more of an obvious choice. Of course, there's a rise of data products which we could probably do a whole show on, but which is this idea of modern companies are building subject specific tools to power their business, right?

At Airbnb, it might be a tool to analyze the way that search ranking is working and analyzing user search session and really understanding what's happening, or at LinkedIn there might be tools like who viewed my profile and which companies are they from and things like that. I think we see more and more of these data products and these data products have – they require to get web scale, so they require to be sub-second type of latency and they often have this component of doing slice and dicing. Those are the cases where a database like Druid becomes really interesting.

I think the places where it does not shine is if you want to let's say an army of analysts to do some very deep analysis and do some complex joins and things that would typically require sub-queries, then for that use case is a more advanced database engine, or more traditional database engine is probably a better choice.

What we started doing is having the same data sets stored both in a database like Druid and in Hive and Presto, right? Would share the same storage layer. We always have the choice of depending on your use case, you can hammer a different database engine that will satisfy your use case and sometimes you can even pivot from one to the other. Maybe your data set is sampled in Druid and your results are probabilistic to a certain point. If you need accuracy, you'd be able to use almost the same query against a different and joining engine and just go back and forth depending on your use case.

[0:24:49.3] JM: In that answer, you hinted at just how expansive the use case for BI is. I've heard this referred to as embedded BI, where you gave the example of LinkedIn, where maybe a user logs into their account and they want to know how many people viewed their profile in recent history. This data might potentially get served from Druid and maybe it gets rendered in some nice BI-looking tool, some graphical tool. Maybe it gives you a breakdown of where these people are, their demographic information.

You're illustrating that you might not only want this BI, and I throughput, fast querying in internal tools, like you also give the example of I'm a business analyst within Airbnb, I want to see where people are querying, what they're querying, what they're entering into search. This is also something that you would want to offer as part of a UI layer to your frontend engineers.

[0:26:02.0] MB: Certainly. I think it's definitely something we see more and more of. I'm delighted as a user of the internet to see more of these things, right? Maybe on if you write a blog post, you're able to see how many views you're getting, how many people are reading through your – what's the percentage of complete reads, right, on your blog posts. If you're on LinkedIn, you know who's viewing your profile. I'm assuming that if you pay for higher tiers of services in LinkedIn, you can do even deeper analytics into their data as a service.

You can imagine too that as a Lyft driver, or as Airbnb host, you're interested in understanding how you're listing, or how you're performing against perhaps your peers, or people in your area. I think this is becoming a competitive edge too for commercial web companies, right? To share some of this data that's relevant to their users, or service provider on their marketplace, or on their platform.

Historically, I think for this we would assume that we would know the shape of the queries and perhaps, we would pre-calculate this data, or have some streaming aggregation process that would store data in a key value store, right? That would assume that you know the shape of the queries that will hit your system. Sometimes you may not know that, you might not for a service that allows for all sorts of shapes of queries right where the user might want to slice and dice and you want to evolve that offering over time. When that's the case, you need a different type of architecture, or different type of database. That's where Pinot, Druid, or even Scuba – this is not open source that will shine.

[SPONSOR MESSAGE]

[0:27:58.7] JM: Logi Analytics is an embedded business intelligence tool. It allows you to make dashboards and reports embedded in your application. Create, deploy and constantly improve your analytic applications that engage users and drive revenue.

You focus on building at the best applications for your users, while Logi gets you there faster and keeps you competitive. Logi Analytics is used by over 1,800 teams, including Verizon, Cisco, GoDaddy and JPMorgan Chase.

Check it out by going to logianalytics.com/datascience. That's logianalytics.com/datascience. Logi can be used to maintain your brand while keeping a consistent, familiar and branded user interface, so that your users don't feel they're out of place. It's an embedded analytics tool. You can extend your application with advanced API, so you can create custom experiences for all your users and you can deliver a platform that's tailored to meet specific customer needs. You could do all that with Logi Analytics. Logianalytics.com/datascience to find out more.

Thank you to Logi Analytics.

[INTERVIEW CONTINUED]

[0:29:26.2] JM: There are listeners out there who all their data is in a PostgreSQL database, or it's getting written to Parquet files. There are people who are in different phases of their data infrastructure build-out. Maybe they don't have Druid set up, and so the idea of setting up a custom database just to be able to serve to a new custom BI layer, that might be an intimidating idea. Can you talk about the deployment process of being able to use Superset? Are there some other options that allow me to set up Superset to be used with other data infrastructure platforms?

[0:30:09.6] MB: Right. I think there's a component too on the same side for on the database side of things, right? On the platform side of things that's pretty prohibitive and it's hard to set up, manage and maintain open source software. For Superset, we're trying to make it as easy as possible for people to do that, but it can be pretty complex with the architecture includes things like having an array of superset worker to compute thumbnails, right? To do things like scheduling e-mail reports and we need to have a caching layer and all these things. It's always a fair investment to run any open source platform, or open source product.

I think that's why we're seeing companies rise to offer services and manage, can hosted SaaS offering around these open source product. That's directly relevant, because I just left my previous job at Lyft to go and start a company around Apache Superset, so that we can make it really easy for people to get Superset running and to get it in a host – a fully hosted way where they don't need to have say a data infrastructure team, or an infra team that is spending cycle, installing and just maintaining this piece of software.

Similarly for Druid, there's this company called Implied that make it very dead simple to go and create a cluster and scale it and use it. I think it does make sense for the people who are building open source, or for someone the ecosystem to offer this first-class offering around having this managed software that's always going to be updated, that's going to be super reliable and built and managed and maintained by the people who brought you the open source solution.

[0:31:59.4] JM: Can you talk a little bit more about the plans for your company and what you would like to offer people? For those people listening, if you're not in the data engineering world, Max is an all-star. He's somebody who people have been looking up for a long time and being like, “Wow, I really hope that guy starts a company so he can solve some of my data infrastructure problems.” You finally done it, which I know there are a lot of people out there who are like, “Yes. Now I only have to wait a few more months, or a few more years until I can use whatever Max is going to charge me money for.” Tell me a little bit more about what you're – not to tee you up too much, but what are you going to be charging people money for?

[0:32:44.8] MB: Well, so I think the really obvious thing is to offer software as a service on the cloud, right? To have a multi-tenant, super-efficient place for people to come and get Superset probably for free at first, but then offering avid product offering where we need to make money, so eventually to sell upper tiers of services. We're still in semi-stealth mode and we don't know exactly the nature of the service we're going to offer and what we're going to charge money for, but I think what people are ready to pay for is things that will prevent them from having the need to have an infrastructure team managing, maintaining the software.

Fully hosted, reliable in the cloud, perhaps some services around a better offering around security, authentication, flexibility around managing your data access policy and your data governance, right? Having really good audit as who is accessing which piece of data in your organization.

I think in some cases too, what we've seen out of companies like Databricks, Confluent, Imply, right, is in some cases they'll have a multi-tenant offering. You can imagine a large cluster of say Spark running where there's multiple tenants sharing resources. In some cases, there's these

managed solutions too that are – you can pay us for us to go and run this infrastructure on your cloud, or on a new VPC that we pair with your VP, or your virtual private cloud.

Beyond that, I think the goal of the company much like Confluent is the strong sponsor behind Kafka, the intent is really to grow at least for me, to grow say Superset and make it a very competitive open source offering that's competitive in this space, right? That lines up, that looks at this very fragmented space, the BI and data visualization, data exploration tooling and get open source to win in this space, or to become significant in this space.

I would say the goals of my company are first and foremost, to push Superset and make it something competitive and really appealing and just a great product. Then to have a very comprehensive, easy, well-managed, well-maintained secure SaaS offering, so that people can just easily use it and get the full value out of it.

[0:35:21.0] JM: I want to take people through a few examples of using Superset, just in case they're eager to get started and don't have access to your product quite yet. Let's say I'm running a service where I'm running a podcast and I have a data storage system that's logging all of the events of users interacting with my podcast. It logs when they start playing it, it logs when they stop playing it, I'm storing all of these events let's say in Mongo, just in a Mongo database.

Then I want to have Superset render interesting analytics on these user events. I want to go from just having this stuff in Mongo, logging these events in Mongo and have a Superset interesting visualizations, querying this data system, building a dashboard with some stats for that data. If I was to want to set that up today, what would be my series of steps for setting it up?

[0:36:20.4] MB: Great. I'll start just by saying describing a little bit what Superset does and what's the candy at the end of this journey, right, once you get your data accessible and Superset what you get. Superset is – so it's really a web application that runs in your organization, where different users are going to be able to log in and first to go and slice and dice data, so we have this very easy to use place where you can just pick your metrics, pick the type of visualization that you want and without writing any SQL or writing any code, you can go

and apply filters and visualize, slice and dice your data, explore it. Ultimately, assemble your visualization into interactive dashboards.

Superset as this exploration feature, then the dashboard view. On top of that, we offer a SQL IDE, so that's a place that where if you do know how to write SQL, you can go explore your database schemas, write some SQL and ultimately, visualize the results and put them in a dashboard and share them with your – share these visualization with your team. This is what you're going to get at the end of this journey.

At the very beginning, so you're starting from data in Mongo and then probably one of the first question and one of the first thing you need to do is get your data into a more of an analytics database. Mongo is probably not a great place to run analytics types of workloads, so you probably want to think about where you're going to store this data. Perhaps on one side, you might want to scrape Mongo and export the data to analytics database and here, I have some popular choices today are things like BigQuery and Redshift and perhaps Amazon Athena.

Then you probably want to ask yourself too, are you interested in real-time? Yes or no. Or is that something that is important enough for you, so that you will spend extra cycle dealing with real-time computation and data streams. In any case, so you'll have to choose analytics database and write some data pipelines. You might even need on top of scraping your database, you might need a little bit more of an event framework where in your application you might want to be meeting more events, right?

On one end, you can think of Mongo as a place that stores the state of your application and you want to scrape that periodically into a database. You might also want to have an events framework where you stream events to your database of choice. Once you have this, then it's simply probably a matter of installing Superset, configuring it, adding your connection string to this database that you've just set up and then hammering away at it, right? From that point, once you added the connection and Superset it's pretty straightforward to point to a table and to just start slice and dicing, creating the visualizations that you need and then assembling those into dashboards that you can use or share with other users in your organization.

[0:39:32.3] JM: That's the greenfield, early stage set up process. Let's talk about Superset at Lyft. I think that's a different example. You've got mature data infrastructure, you've got Kafka set up, it's doing some stuff, you've got a data lake somewhere, you've got more data infrastructure. Describe how Superset was used at Lyft.

[0:39:57.1] MB: Right. First, I want to say it's been awesome working at Lyft. Lyft has been an awesome sponsor for Superset and they have really interesting data and use cases around real-time and geospatial. They have a really good data infrastructure team and just a reliable foundation to build upon. What the systems, or maybe what the architecture, or infrastructure in and around Superset at Lyft, maybe to describe that's on one end, there's the pipeline, there's something called analytics events, where the different applications running at Lyft are emitting events, so user, actions and time. You can think of the mobile app generating events.

These events somehow get collected. They get sent to Kafka. Within Kafka, there's different ingestion frameworks to either send this data on one end to the real-time infrastructure, so that's using Apache Flink to do perhaps data enrichment and data filtering and that sort of things, and then sending the data over to on one, in Druid and on the other end to Parquet files and S3 that ultimately become viable through Hive and Presto.

Now Superset at Lyft is connected to these databases, so to Druid and to Hive and Presto and allow people to just go and hammer at these things. Of course, there's a layer of Airflow too, so orchestration, or I would call it data pipeline orchestration, where a lot of the ETL, the extraction transformation load of data is orchestrated and executed. There's always a need for taking your data, denormalizing it perhaps in some cases, enriching it, curating it, cleaning it and ultimately, making it viable to the consumption layer, which would include Superset.

[0:41:55.0] JM: The thing about BI that makes it different than some of the other areas that we explore on Software Engineering Daily is that non-programmers are using it. Your tool is not just for programmers, it's for the data analysts, the business analysts. How do non-programmers use Superset?

[0:42:19.2] MB: Yeah. I think where we're aiming at there is just any information worker. I think that makes BI and Superset really interesting in terms of how much reach it has into

organizations and within these organizations, right? Here we're talking about open source that's extremely visible, because people – that many people, maybe like 80%, 90% of the workers in a company, or in a tech company for instance will be information workers, which do need access to some data on a daily, or weekly basis.

These tools become really prevalent. They're just for everyone. Sometimes I'll be asked which persona is Superset targeting and really, it's pretty much anyone in the organization, right? There's a whole spectrum of let's say data literacy. In organization, there's people that are – we have data scientists that are super data savvy and they know how to interact with notebooks and create tables and write very complex SQL. That's one end of the spectrum.

The other end of the spectrum is just someone who wants to come consume a dashboard, interact very lightly with the dashboard and perhaps asking a few of their own questions. With Superset, we're really aiming at satisfying everyone. If you do know SQL and you're very data literate, we want to offer you the best tools. Also, if you don't know exactly what you're doing, we want to offer you also a solution to come and just interact with data in a way that is relevant and easy. I think it's challenging to offer these tools that work well for everyone, but that's definitely the Holy Grail and that's what we're after.

[0:44:08.8] JM: There was the show we did recently with Netflix about notebooks. You and I talked a little bit about this offline. That show made me feel like I have no idea what is going on in the world of data science, because in that episode, the guest Matthew Seal, he just talked about how Netflix has been completely overrun with the use of these Jupiter notebooks, which are a shareable software development environment. It allows you to do visualization, allows you to share the code, as well as the results of different data science jobs that you might want to run.

I just like to get your perspective on the changing tool set of the exploratory data scientist, or the data scientist who is figuring out how to put a machine learning model into production. The tooling that they're using and how these UI layer tools, like Jupiter notebooks, or Superset are affecting their workflows.

[0:45:13.3] MB: That was a really interesting show, by the way. It's very much confirming trends that I've observed over the past five years, right? Of just dashboard becoming more prevalent and dashboard as unit of schedule, or unit of things that can be productionized for data pipelines. Also, we've seen a rise of notebooks as a place to define and assemble dashboards too.

I think, notebooks are interesting because ultimately, they're the most flexible to [inaudible 0:45:42.9] code, so you can from a notebook environment you have full access to anything that programming languages will expose, so that ultimately, they are as flexible as can be. The reality is there's something prohibitive about them is that you do need to go someplace and write code. Out of the maybe 80% to 90% of people would need access to data daily in organizations. A lot of these people are either not able to write code, or not interested in writing code depending on their use cases.

Of course, if you need that flexibility, then the notebook is a great place to do that, especially in environments where notebooks are building block, right? They're reusable components that are predictable and can run in production. In other cases where you're just trying to slice and dice data, right? Say if I work at Lyft and there's a spike, or there's some step change in the number of rides, or there's something happening right now that is different than what happens usually, or typically at this time of day and I need to figure out the dimensionality of what's happening, firing up a notebook is probably not the right thing to do, right? You probably just want to start from a visualization that provide some context and then easily start slice and dicing.

For many operations, you don't really need that level of flexibility. That's where well-designed BI tools shine, right? It's more the pivot table type of operation where you say, "Group by this, filter on that, drill into this dimension," these sort of things.

[SPONSOR MESSAGE]

[0:47:39.8] 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.

If you want to get some additional help for your engineering projects, go to softwareengineeringdaily.com/g2i.

[INTERVIEW CONTINUED]

[0:49:31.7] JM: Another aspect to that show that stood out to me was that the changing nature of these data rolls at a company like Netflix, so there's a cardinality of different roles that's increasing. You've got data analysts, business analysts, machine learning scientists, etc., etc., etc. At Lyft, have you seen the expanse of new data rolls? Did that stand out to you in that Netflix episode?

[0:50:01.4] MB: To me, it was – I think it's the model that they were talking about really matches what we've seen, what I've observed at Airbnb and Lyft. Maybe need more newer data rolls, or a way that say data engineering is evolving is to do a lot more streaming computations. I think this is not about to change, right? We're probably going to see a lot more information workers climbing up the ladder of complexity and abstraction.

That means if perhaps a few years ago you were using Excel, maybe today you're writing a little bit of SQL. If perhaps you were able to write a little bit of SQL a few years ago, now you're pushed to start interacting with notebooks and writing a little bit of code. Maybe if you're a strong data engineer that's been doing this for a long time, you start doing a little bit more ML. We see these roles climbing just the ladder of the complexity of the work that they do. As things get more complex, it creates opportunity for more specialization.

I think we're just going to see the people, companies that are successful investing more and more in data, data teams and for these roles to specialize more as things go. One trend that's slightly different here is the data infrastructure type of role, where we're seeing some consolidation around the cloud. Either, SaaS companies, like the company I'm starting, right? Instead of having teams of people running a lot of open source software and managing it from within your organization, you might rely more on specialists and people really know that software to run it.

Similarly with databases, right? A lot of people are moving from either on-prem, or their own, managing their own EC2 instances of databases, moving to things like BigQuery and just enjoying not having to manage and maintain these pieces of software.

[0:52:04.2] JM: The world of vendor selection is becoming a really confusing set of problems for a CTO, or a CIO. It's also becoming a dogfight for the infrastructure vendors. If you're a bank and you're looking for what vendor to go with, you're looking at AWS, IBM. Also Databricks and Imply. What's changing in the world of vendors and vendor selection and software infrastructure companies?

[0:52:38.5] JM: Yeah. I'm not sure if I'm the best person to ask this question to, but I can talk about this space and about fragmentation, right? I think we're in an era where people are investing a lot in data and data infrastructure and cloud infrastructure. There's definitely – that creates a gold rush and a lot of companies are popping and offering services. Usually, we see these phases of fragmentation followed by phases of consolidation around the tools that win the market.

I think we're still very much in a state of fragmentation. An example of that is for setting up my company trying to figure out which SaaS vendors to use for things like HRs, HR, payroll expenses. It's extremely complicated to pick anything. It's very quickly evolving too. Hopefully, we'll see some consolidation. One thing to that I say specifically about the BI space is that maybe it is okay to offer within your company different tools, right? Maybe a company can have a Superset, Looker, Tableau and multiple offerings. There's pros and cons there, but maybe it does make sense to offer that to your internal audiences a whole array of tool.

[0:54:00.6] JM: BI is one of these verticals that's usually not win or take all. There are these verticals like log management and operational analytics, where you see multiple players succeeding, like you see in the in the world of log analytics for example, like Datadog and Dynatrace and 15 other log management companies that are all doing just fine. At Lyft, did you guys standardize on vendors like that, or did you have more of a engineers can choose what tools they want?

[0:54:34.4] MB: I think it's a little bit of both and it goes with the lifecycle of maturity of an organization. I think, Lyft is probably entering more of a phase of consolidation on all the tooling that's used internally. For a different set of reasons, we decided I think to standardize and to limit the number of tools and to go with the tools that were more flexible and that we wanted to commit to for the long run.

It's normal in the face of growth of a company to go and buy all sorts of things and figure out what sticks. Similarly at Facebook, something similar was happening with just building tools, right? People are building all sorts of stuff at hackathon, for instance. Then over time, there would be some clear winners and we would settle on a more cohesive and subset of the original product.

I think it's normal for organization to go through these phases of expansion, consolidation and different areas at different times, right? I think, say Lyft will consolidate around their BI tools around Superset over the next I'd say year, but they might still be in a phase of growth and research on other areas.

[0:55:53.1] JM: When you started Superset, you had spent time working on Airflow, so you had considerable open source experience. What else have you learned about open source software since you started superset?

[0:56:07.1] MB: Yeah. What have I learned on open source software? I think I've reused a lot of what I've learned on Airflow working on Superset; one thing being to grow open source software. It's all about daily interactions and really winning one user at a time through just presents, say on GitHub and processing GitHub issues, honoring the pull requests that are coming in and just being – doing all the things that are required for your project to become legit, right? It's things like having a really nice appealing readme on your GitHub and things like working on documentation and building the features that the community wants.

Beyond this, what have I learned working on Superset? I've learned things about software foundations too, so that software foundations are somewhat important, but also can be fairly distracting an area of empowerment on one side and then friction on the other. I'm sure there's been a lot more lessons than this, but ultimately, it's about community, it's about people, it's about execution, it's about product. From that perspective, it's very similar working on Superset as it was working on Airflow.

[0:57:28.6] JM: Question about data infrastructure, there was a time – well, I guess we're still in this time where you have certain general purpose data infrastructure tools that are being used for a wide variety of applications. Things like Spark, for example. You could use Spark for many of the workloads that Druid gets used for, but Druid is a domain-specific tool for this, I think it's typically read-only data workloads with high throughput and fast query response time.

You built Airflow of course and people could have used other our scheduling tools, but you built a purpose-built scheduling tool specifically for the data infrastructure workloads that you were seeing. Are there any other domains in the world of data infrastructure that seemed underserved today, where you would just be like, “Oh, I wish there was an open source tool that would solve this specific domain within data infrastructure.”

[0:58:41.9] MB: Right. I think there's a handful. I'll talk about two. One is more of a family of things. I've given this talk before. That's probably online. That's called advanced data

engineering patterns with Airflow. Really can remove the Airflow component of this and really just look at this talk from this idea of advanced data engineering patterns. What we're seeing is that people are building the same pipelines, data pipelines over and over. There's families of pipelines that every company has built at least once, if not multiple times; things like computing, growth, accounting and engagement metrics on a data set that's primarily user actions in time, right? Or things like an AB testing framework pipeline, or things around user segmentation, right?

We keep rebuilding these same pipelines and really often, we start from scratch. When I originally build Airflow, I thought people were going to start sharing these higher-level constructs, or these higher level abstractions on top of Airflow. I haven't really seen that come through yet. I think the reason why, it's just because there's so many assumption in this code, it usually assumes that say you use Airflow or something else, or that you use Spark, or Hive, or Presto, or all of these things and sharing this code would mean that the people on the receiving end would need to add a very similar infrastructure to yours. That's one thing we haven't seen very much.

Then another area that's very dynamic, I think now that we'll see solutions around are things like Amazon sage maker. Just a way to run notebooks on Kubernetes and all of the building blocks are pieces of infrastructure that build upon this assumption. That ties into our conversation earlier about the notebook-centric company, or data team where the notebook becomes the thing that you schedule and maybe the notebook becomes the thing that you share, or reuse to do data science.

Kubernetes hosted notebook kernels in containers that are reproducible and shareable. I think we'll see some really good stuff come out of there in the next few years. I think there's already solutions. I think Kubeflow and then I believe Airbnb and Lyft are probably on the verge of open sourcing things in those areas too.

[1:01:15.4] JM: Awesome. Okay, one last question. You've written a lot about how to succeed as a data engineer, just general philosophies, infrastructure strategies, you've given some great presentations. Anybody who is a data engineer should go check those out if you haven't seen

some of Max's stuff already. If someone is listening and they're getting started as a data engineer today, what general piece of advice would you give them?

[1:01:43.3] MB: Yeah, I've had this question before and people asking like, “Hey, I want to be a data engineer. What should I do?” My journey into data engineering was a very long journey coming from data warehouse architecture, started in the early 2000s. I have an unconventional path to get there. I think in general, you want to follow your passions and you want to be in tune with what your organization needs.

Maybe a key is to get close to data analysts and data scientists and really understand what their daily struggles are and what data structures would be helpful to them. First, it's understanding the domain, knowledge and the business problems you're trying to solve. Then about the mechanics of how to do this, I think this highly depends on the environment that you're in, right? If you're in a certain environment, you might have to learn Spark, if you're in a different place, you might have to run Vertica, SQL, scripts orchestrated by Airflow and some other place, it might be take a totally different shape. It depends in the environment that you're in.

Probably one of the first thing is well, join a data-driven company if you haven't already. If you're at a data-driven company, maybe find a mentor, find some use cases and learn what you need to learn to have an impact. Maybe one or two things – I haven't really followed the literature very much, but I know the Kimball books, so Ralph Kimball wrote in the 90s I think about data warehousing and star schemas and the dimensional modeling, which I think some of that literature is still relevant nowadays, but not a 100% fully still relevant, but I would say most of it helps to put things in perspective. That might be more for someone who's done some data engineering, who's looking to understand a little bit more the fundamentals and that could put some things in perspective.

[1:03:41.3] JM: Max, thank you for coming back on Software Engineering Daily. It's been a pleasure.

[1:03:45.2] MB: Yeah, it was a pleasure to be on the show and the happy talking to you again sometime.

[1:03:49.8] JM: Oh, absolutely. I mean, well there's so much – I actually had a lot of questions around data engineering, data infrastructure, but I restrained myself. I kept it mostly to Superset.

[1:04:00.6] MB: Well, I'd be happy to come back on the show and talk about all these things.

[1:04:03.8] JM: Okay. Hey, and one other thing, you're hiring. People who want to potentially work with you, where should they go?

[1:04:11.6] MB: Right. I'm starting a Superset company. People interested in working on open source software around data visualization, dashboarding, offering tools for analysts, data scientists, or anyone who works with data, we're going to be building some really exciting things and push Superset forward. Probably the best place to reach out is to connect with me on LinkedIn and send me a quick message and I'll be super happy to connect and talk about what we're doing.

[1:04:40.9] JM: Okay. Max, thanks for coming on the show. It's always a pleasure.

[1:04:43.7] MB: Thank you so much.

[END OF INTERVIEW]

[1:04:48.7] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source, it's free to use and GoCD has all the features that you need for continuous delivery. You can model your deployment pipelines without installing any plugins. You can use the value stream map to visualize your end-to-end workflow. If you use Kubernetes, GoCD is a natural fit to add continuous delivery to your cloud native project.

With GoCD on Kubernetes, you define your build workflow, you let GoCD provision and scale your infrastructure on the fly and GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team, and they have talked in such detail building the product in previous episodes of Software Engineering Daily. ThoughtWorks was very early to the

continuous delivery trend and they know about continuous delivery as much as almost anybody in the industry.

It's great to always see continued progress on GoCD with new features, like Kubernetes integrations, so you know that you're investing in a continuous delivery tool that is built for the long-term. You can check it out for yourself at gocd.org/sedaily.

[END]