EPISODE 1465

[INTRODUCTION]

[00:00:00] JM: Data as a service is a company category type that is not as common as API as a service, software as a service or platform as a service. In order to vend data, a data as a service company needs to define how that data will be priced, stored and delivered to users streaming over an API or served via static files. Naqeeb Memon is an engineer at SafeGraph and he joins the show to talk us through the mechanics of delivering data as a service.

[INTERVIEW]

[00:00:27] JM: Naqeeb, welcome to the show.

[00:00:30] NM: Thank you. Thank you very much for having me.

[00:00:32] JM: You manage teams at SafeGraph, engineering teams. We've done a number of shows on SafeGraph, I'd like to get an understanding for how a data as a service company is managed from an engineering perspective. Can you give me a top-down view for how data as a service teams are managed?

[00:00:55] NM: Yeah. Basically, to give you just a high level, I would say, we have what we call a sourcing team that's responsible for pulling in all the different data sources that we use for our data service platform. We also have a team that processes that data. Then finally, we have a delivery team that's responsible for getting this data out via our enterprise customers, or in my case, a self-serve platform.

[00:01:22] JM: From an engineering perspective, how are those different teams arranged in terms of responsibilities?

[00:01:29] NM: Yeah. From a sourcing standpoint – could you elaborate a little bit more when you say arranged?

[00:01:36] JM: Sure. You're a data vendor. There are different elements of that. There's data cleaning, data aggregation, data delivery. There's the front end of the website, I just like to get a get a sense of the different teams you manage how they're composed.

[00:01:55] NM: Yeah. Just a spoiler alert on this one, I don't oversee all of this, but I can give you just a high level. On the data sourcing side, what that team is responsible for is kind of bringing in the data, right? If you're thinking about a data vendor, there's a couple of steps. It's sourcing the data, bringing it in. Start with cleansing of it, right? That belongs in the sourcing team. Then when it comes down to where we actually take the data and normalize it that falls into what we call, we have an internal name called places, but it's more of like the – I'd say more of the transformation layer and matching. Then finally, on the delivery side, that's one of the teams I oversee. It's one of the teams that focuses on how do we get this data out. We have two primary ways to deliver it. One via our enterprise network, so all the clients that we have. Then, finally on the website. What we do is we take that data. We ingest it, put it into a database that makes it available. Then, finally, it gets served up by the front end.

[00:02:58] JM: Your main focus is on the delivery side of things?

[00:03:02] NM: Correct.

[00:03:04] JM: Okay. As I understand, that's like, I want a bunch of data about places, and I'm going to get the data in CSV or whatever kind of format that is going to get delivered to me, what are the prime engineering challenges in delivering that data?

[00:03:22] NM: As I say, some of the delivery challenges is the scale of the data. Imagine you as a customer wanting all the places in the United States, right? That's usually about, right now, about 10 to 15 million. Making sure that we're able to deliver that at a reasonable time is one. Also, on top of that, we also offer the ability to kind of filter, kind of build your own data set, in essence like, we have data about places, we also have data about other attributes, like geometry, like how does a place look like. Some of the other challenges there is also shaping the data in a way that you can consume it.

[00:04:02] JM: When you say data shaping, what does that mean?

[00:04:05] NM: In essence, you can pick different attributes that you would want about the data. If you think about places, places have a physical address location, but there's also other aspects of the data that you would want. When I say shaping, specifically giving you as a user the ability to be like, "Okay. I want to pick. I want to know about the address of the city, but also, I want to know how does this place look like in the physical world? What's the latitude, longitude?" Giving the ability to – when I mean shaping, I'm referring to like how a customer can pick the attributes that would want on the data that gets delivered.

[00:04:39] JM: Those attributes, what's the cardinality of those attributes? How many potential attributes could there be?

[00:04:46] NM: I'd say right now, they could be upwards of like 50, but we're always adding more attributes as we're continuing to add different data sources.

[00:04:55] JM: Do you have an understanding of the underlying architecture that goes into where that data is stored?

[00:05:04] NM: I mean, when you say data stored? Do you mean, the underlying data that we use to deliver off of or –?

[00:05:10] JM: Yeah. I mean, the core places data that you're vending to people, what's the database infrastructure or storage system that you use to store that data?

[00:05:18] NM: Yeah. For that, what we use is two different formats. We have like a variety way of deliver data. One, being an API. As an API user, you get to ask like, specifically I start with, what are the places around me? That data is stored in a Postgres database for easy access. Then there's also the, we want to receive not only just real time data, but you also want to receive, "Hey! I want to know all about the places in the US. That data is actually stored in parquet files, which is then consumed by Spark based on who's actually requesting it.

[00:05:55] JM: So you pull data from parquet files into RDDs and Spark, and then you vend them from the API?

[00:06:04] NM: That's for our larger delivery sets. We don't vend them via the API there. When we vend them via the API, it's usually through our Postgres database and paginate through that.

[00:06:14] JM: Okay. Got it. As an engineering manager, how are you interfacing with the engineering teams on who are responsible for the delivery?

[00:06:26] NM: Could you clarify when you say interfacing?

[00:06:28] JM: Yeah. What's an example of deliverables that you're giving to them?

[00:06:34] NM: Right. Okay. The deliverables we give the data team is usually the raw objects that represent the places. The data delivery team has a couple of functions. One is, ingest the finalized data set that we receive internally. Two, make it available both to our enterprise channel, as well as our self-serve channel. In essence, if you think about one of the interesting challenges here is that we have different release cadences. For our places, we usually do a monthly refresh. Part of that challenge is, as we're serving the current version of data, we want to make sure that the new version is being ingested, and prepared and then released. Then, as part of the ingestion process, what we do is we take that data, we read it into a Postgres database, so we make it available via our API. Then we go ahead and then package that data into parquet file, so that when we need to deliver via other formats like CSV, our Spark jobs can pull those Parquet files to make the deliveries.

[00:07:35] JM: Gotcha. You get raw data from data vendors and then you repackage that into Parquet files.

[00:07:44] NM: We get sources from many different. There are data vendors. There's also some crawling that we do on our side. I'd say, part of other teams responsibilities to bring all those data sources together, cleanse them, and make it available into one deliverable that we can call our places.

[00:08:05] JM: Got it. Can you give me a sense of how you maintain the quality of the ingested data that's changing over time?

[00:08:16] NM: Unfortunately, I can't really speak to the quality of data, because I think that speaks to more of the sources team. I'd say, my responsibility is taking the end state of data and making it available.

[00:08:28] JM: Got it. Is that like better understanding some specific customer needs and translating those into new kinds of data delivery or new facets of the data?

[00:08:44] NM: Yeah. A part of it is understanding what the customer needs. Part of it's also, how do we make this data acquirable and exposable via a self-serve tool. One of the primary teams I oversee is, you can come to safegraph.com right now and be able to take and ask the question, "Hey! I'm looking for all the Starbucks in Chicago. How many rows of data do you have?" Then from there, taking in being like, "Okay. I want some data about the places. I want about the geometry or how people are visiting that data." Visitation data and then being able to shape that.

There are two paths there. There's the, our enterprise customers that have a specific way that they want to deliver data to them and well as format. Then the second part is, your self-serve customers that have probably like a onetime use, maybe they're doing a site selection on a coffee shop in New York and want to know where all the competitors are. They can come into our shop and then pull that data so that they can make that decision.

[00:09:45] JM: Okay. Presumably there are improvements that you need to make to the platform and the platform is already pretty comprehensive in terms of how it can deliver data, what kinds of data it can deliver. Can you give a sense for the roadmap and what you're working on right now?

[00:10:04] NM: Yeah. One of the things that people have a hard time with data is understanding how it could be used. Being data as a service company, our secret sauce is the data. One of the primary questions that we get asked is that, "Okay. How can I use this data? What can I do with it?" We found that visualizing the data has been very helpful to kind of show people both what we offer as a product as well as how to interact with the data. One of the items that we're working on is, how do we make this much easier for people to understand what to do with the data. We found that putting some visualizations in our shop, was able to, it helps people to bring together that. Okay. This is the data and I can see how you can use it. One of the things we're solving for that.

The other thing on the roadmap is thinking about our API and the different types of users that hit it. Right now, what we're looking at is, how do we optimize for people who are trying to enrich their current own datasets or trying to – for example, you might have an address about a particular location, but you're not necessarily sure what's there. We offer an API for you to go and say, "Hey! What is at location 123 Fake Street?" We give you, "Okay. There's a Dunkin Donuts there or there's a Starbucks." They're trying to build more of a live API use case that app developers can kind of integrate into their platform.

[00:11:25] JM: In the example of providing a visualization around the data within the store, what was the feedback loop between SafeGraph and the customer to understand that that was a feature that could improve, I guess, browsability?

[00:11:44] NM: Right. I think there's a number of different inputs. Usually, one comes in from the sales calls that we have. As our sales people are talking to different clients, some of this feedback gets sent over to our product team. The other feedback is, we're also constantly talking to our power users and understanding what their use cases are and how they're trying to demonstrate the data. Third part was, I'd say, also just understanding internally, ourselves just kind of dogfooding our current platform and asking the question like, "If this is the first time you've heard of SafeGraph or you're not as familiar with our value prop, how would we be able to demonstrate the value of our data?"

One example, what we did was, we had like an internal hackathon, which we call [Hackera 00:12:29]. One of the things that we want to do is like, how do we go about like showing the value of this data? And we prototyped, "Hey! Why don't we build a tool? Let's call it a search radius tool? Based on a specific lat-long and distance around, here are all the data points that we have." Looked at that prototype, said, "Okay. This makes sense. Let's start talking to our customers about it. Let's get some feedback on this." Then finally, bring it to market.

[00:12:54] JM: Do you see yourself more as a product manager or do you find yourself more in the weeds on specific engineering problems?

[00:13:08] NM: I'd say a little bit of both. I think as an engineering manager, you kind of have to wear multiple hats. I do think that a good engineering manager also has a product lens. I think, to me, being a good engineering manager also make – you should be thinking about how the feature and how it's being used. But also, being able to speak intelligently and get into the weeds of certain problems. But I think, building the right team around you allows you to kind of not have to go in super deep on some of those technical challenges and making sure that you have architecture, a big picture idea of most of the features. But you can jump in when necessary.

[00:13:49] JM: Can you talk more about the engineering stack of the data delivery platform, how's the API written, how is it vended, what's the process for architecting and engineering an API for data?

[00:14:06] NM: Right. Okay. If we talk about the data delivery stack, we can talk about, when you say, are we focusing on like the API use case? Are we thinking about the larger deliveries or both?

[00:14:18] JM: Let's go through both.

[00:14:20] NM: Okay. On the API side, what we need to think about is like, when we have this data, we need to make sure it's available. We need to make sure it's – we have very reasonable response times when it comes to data, right? In essence, it needs to be available in some sort of data store and something that can be read very quickly. First part of that is making sure that we have the right data store. In this case, we're using Postgres to kind of surface this data. Second part of this is kind of making the decision on how to make this data available. There are different options like GraphQL, obviously. Gives you opportunity to kind of shape your responses as well as kind of go down how your data is related to each other. Then there's also REST endpoints.

Making the choice there on how do you want this data to be surfaced **[inaudible 00:15:05]**. We ended up choosing GraphQL as we want to expose it. As there are questions like, I'm for example searching for all the Starbucks in Chicago, but I also want to know, "Hey! What are different attributes on these Starbucks?" GraphQL allowed us to kind of shape the – not only allow our API consumers to shape the data, but also ask more questions about, "Okay. Now that you have all the Starbucks there, what are the other attributes and how are they related to each other?"

[00:15:32] JM: Can you talk about how the API is architected and how it trans – if we talk about the API just on-demand data, what is it typical request and how does that request get translated into a query against the database?

[00:15:50] NM: Yeah. Okay. I'm talking about the different queries. Our first most common entry point is our search. Typically, what people are trying to figure out is that, given a set of criteria, what places meet those criteria. Usually, what we tried to do is we expose a number of different ways to query that data. For example, you might be wanting to figure out a specific list of places by categories for next codes, or you want to do it by brand search. Usually, our entry point is there, and then we return back a list of different places that can meet your criteria. Then based on, for those type of places we have what we call place key, a unique identifier for you to reference that specific point of interest.

Typically, after that, then you ask the question about, "Okay. What specific attributes I want to know about this?" For example, we have, like I said, our geometry data, which talks about the physical place itself or like we have pattern, which is basically how people are visiting that particular place. Then the next question becomes, "Okay. Now that you have a particular physical place, now, tell me about how it's physically stored or what does the visitation look like over the last week, last month, last year."

[00:17:06] JM: Maybe, let's give the example of a GraphQL query. GraphQL query comes in. Do you have an understanding of what goes on the GraphQL server to translate the GraphQL query into a Postgres query?

[00:17:21] NM: Yeah. I guess, specifically talking about how we would translate that or –?

[00:17:28] JM: Yeah.

[00:17:29] NM: Yeah. Basically, what we do is, we would take the query itself, the resolver, push it down to our, what we call storage gateway or abstraction. In essence, you can ask the storage gateway, "Hey! I would like to know all these different places, specifically all the different places with the search filter, let's look at all the Starbucks in Chicago." That gets translated. Then we also have a separate query, depending on if you're asking for different data related to it, run a separate query. It's like, "Okay. Now, taking this ID and taking these place keys. Let me go ahead and find and fetch the related data to this."

[00:18:02] JM: Gotcha. Do you have a sense for the typical query patterns in the sense of frequency? Are people requesting bulk data just for ad hoc requests or are there cases where people are requesting data every single day to look for updates?

[00:18:26] NM: Right. I'd say I'd segment that., I think there are some users that are just wanting to, on demand want to know, "Hey! I want to find all the places around me" and just want to very quickly get the data there. Then there are also other users that want to – they want to take, for example, all the data that you have on Chicago or United States, and then pull that and run different algorithms on top of them. I'd say like, there are two specific use cases to kind of consider. It's like more of, "Hey! We just want the data now versus –" there's also the people who are trying to get a large set of data so that they can run their machine learning algorithms on top of.

[00:19:10] JM: Got it. It's pretty wide ranging.

[00:19:12] NM: Yep.

[00:19:13] JM: How have you determined pricing for these datasets?

[00:19:18] NM: I'd say that's honestly an evolving question. I think, on the data ingestion use case, we kind of look at it as per record type. Basically, if you're trying to pull all the records in the United States, looking at per record. But if you have more than live use case or autocomplete use case, that's still TBD. We're still kind of iterating over that and finding what the right price point for that.

[00:19:41] JM: What's behind the choice of Postgres as the source of truth database?

[00:19:48] NM: I think speaking to this, I think it's more of – it’s the ability to kind of service different use cases. We also have a transactional component with our SafeGraph self-serve tool. Then there's also that we want to service this data to a variety of different vendors. We found Postgres to be a good starting point for this.

[00:20:08] JM: Have you looked at any other candidate databases?

[00:20:10] NM: We have. But I think for now, I think we're sticking with Postgres, but looking at different options. I think, a couple of ways we've been thinking about this is like, "Oh! What's the most optimal way to serve this data? Because as we continue to build the dataset, there are different challenges. Like now, we're going to be growing in different attributes, right? There will be a point where Postgres will be the right fit. Looking towards other solutions, we're still kind of like internally talking about it. Don't want to talk too much about it at least at this point.

[00:20:39] JM: No problem. Are there places where Postgres is constrained?

[00:20:44] NM: I'd say they're definitely different. Some of the constraints we're running with is offering the ability to, basically one, the use cases on our self-serve tool is to give you a record count, for example. Where we're talking about, okay, like I said, the Starbucks example. But sometimes you might want to get a little bit more granular and more specific if you want to know all about the different restaurants in Chicago, and also full-service restaurants. Some of the constraints there is, now that we have a certain list of criteria and that criteria, for example, on data like **[inaudible 00:21:21]** brands, it's fairly common across our place dataset. Trying to summarize those in a very quick and easy manner has been a bit constraining on the database. In essence, some of the challenges about making sure that we get accurate row counts and being able to return those at a reasonable time, like in a couple of seconds.

[00:21:40] JM: Is there any work around enriching data that that comes in. Do you have a data enrichment layer, where incoming data from various sources gets enriched with other sources or you perform calculations based on existing attributes of those data?

[00:22:04] NM: Yeah. Honestly, right now, we're kind of in the exploratory phase of building in the enrichment layer kind of. Some of the things that we found our users are doing, especially on the API side is taking our raw data, and then making some of those summarizations. One of the popular use cases were, "For a specific place, I want to know the popular times of the day that they visit." That, from an API standpoint isn't available through any other competitors. What we're kind of in the process of doing is like, okay, we can offer.

One of the key insights I think we've been getting is that we have all this raw data, how do we make this available in a way that can be easily consumable. If you are trying to get, for example, one of the popular hours in the day for a particular place, you would have to hit our API, get data maybe for the last month and then be able to extract that insight yourself. The enrichment use case is kind of interesting, because that's where we're kind of, right now in the midst of building for that and saying, instead of having our consumers hit our API for this raw data set, why don't we just offer an API where we have, let's call them convenience fields. Not only do you get data about that particular place, but you can ask, "Okay. What are the popular hours of operation or what are the popular days for this particular place?"

[00:23:22] JM: What are the canonical problems that you see in the data delivery platform, the canonical engineering problems, things that you seem to run into again and again?

[00:23:35] NM: I'd say a couple of things. I'd say, there's the, how do we make sure that this data is available in a way that our end users can immediately find value in. Some of the challenges there is that with a large data set. Right now, we're offering a delivery via CSV. That's good for some users, where they can go ahead, and download and integrate into their workflows. But there are also the challenges of, how do you make sure that we decrease the – when we deliver a particular file, it's in the CSV. Our users have to take that CSV, read into a data frame, make it not even just reading a database, take that data, make it – probably reading it into their own data store, making it available via data frame, and then run the different operations they want. Some of the challenges there is like, how can we make this available? If they're in snowflake, if they're in AWS, how do we make this more seamless of a connection, so that our users can automatically start querying off the data versus like having to figure out where to store it, how to unpack it and then get their insights.

[00:24:42] JM: Can you talk a little bit more about – since you mentioned data frame, can you talk a little bit more about the usage of Spark across the data delivery platform?

[00:24:52] NM: More so how we use it and how we're actually delivering off of it or –

[00:24:56] JM: How you're using it, how you're running it? Just general usage of Spark.

[00:25:01] NM: General usage of Spark is usually for – typically, when a request comes into our system, usually, from our self-serve, we need to first, make sure that we understand what the data requests are. Then from there, we dispatch it to our Spark instance. Our Spark instance based on the specific data set, whether you're pulling in places or another data set pulls the right Parquet files together, shapes that, queries the Parquet files, applies the different filters for you on the Parquet files, and then joins that data into one data frame that can be then converted into a CSV file and then pushed over to S3 for delivery.

[00:25:46] JM: Got it. Spark is used basically for all the in-memory calculations you would need to do and all the preparation functions you would need to do for those large datasets.

[00:25:58] NM: Right. I guess I could speak to a little bit on the opposite side, so when we're actually ingesting the final product. Spark is used to actually ingest the data and making it available. Then also, on the data delivery side to actually do the in-memory calculations, join the data and then make that available as a CSV file.

[00:26:14] JM: As we begin to wind down, when you think about the broader data as a service category, it surprises me that there are not more data as a service companies. Can you give your perspective on why there are not more data as a service companies?

[00:26:31] NM: Yeah. I think the first thing is the moat that you need to build. I think with data as a service companies, there's a certain point inflection point where you need to be building, building, building, and then you get to the point where you have enough data so that people would come and ask those type of questions. I think one is making sure that you're collecting a decent amount of data before it becomes valuable to your end consumers. In terms of the other thing is like, what insights are you offering on top of that data? I do think that if you're just aggregating data, I guess what would make that different from any other company that offers web script data. I think what other insights and secret sauce you offer. Then finally, making sure that you have a more broader range of users that are aware of your data and understand how to use it.

I do think some data as a service companies are very geared towards more technical users. If you can broaden some of that appeal to – some of the people who don't are not as technical, taking the example of, you might be a user that's trying to build a coffee shop and want to figure out where your competitors are. Can you make it easy for someone to kind of look in your interface and take a look and say, "Okay. Here are all my competitors. Now, I want to pick a location between Wabash and Wacker because I find that to be the most optimal location. There's not any Dunkin Donuts around me. This, I was able to use this data to derive that insight without me having to pick up like pandas or any other data frame and then extract that information from there."

[00:28:12] JM: Okay. Fair enough.

[END]