# EPISODE 1221

[INTRODUCTION]

**[00:00:00] JM:** As the volume and scope of data collected by an organization grows, tasks such as data discovery and data management grow in complexity. Simply put, the more data there is, the harder it is for users such as data analysts to find what they're looking for. A metadata hub helps manage big data by providing metadata search and discovery tools and a centralized hub which presents a holistic view of the data ecosystem. DataHub is LinkedIn's open source metadata search and discovery tool. It is LinkedIn's second generation of metadata hubs after Warehouse.

Pardhu Gunnam and Mars Lan join us today from Metaphor, a company they co-founded to build out the DataHub ecosystem. Pardhu and Mars and the other co-founders of Metaphor were part of the team at LinkedIn that built the DataHub project. They joined the show today to talk about how DataHub democratizes data access for an organization, why the new DataHub architecture was critical to LinkedIn's growth, and what we can expect to see from the DataHub project moving forwards.

[INTERVIEW]

**[00:00:59] JM:** Guys, welcome to the show.

**[00:01:01] ML:** Thanks, Jeffrey, to have us on your show. We were very following up your podcast for a while and we are big fans of like multiple data related especially podcasts which came across.

**[00:01:10] PG:** Glad to be here, Jeffrey.

**[00:01:13] JM:** Well, you guys both work on a company, Metaphor, that is based around the open source DataHub project, and DataHub came out of LinkedIn. It's based around metadata search and discovery. Could you give some context on what metadata means in the context of this conversation?

**[00:01:35] ML:** Sure. I can give you a sort of a broad definition of metadata, which is the sort of the definition we usually use. Everybody will probably give you the canonical as of, “Well, it's data about data,” but that is of course very, very abstract. And so to us we feel like this is a more concrete definition of that would be context about data. So obviously, no doubt, nowadays companies collect huge amount of data, but the flip side of that is a lot of time these data are collected and the sort of the context or processes and whatnot, a lot of these contacts associated with these data are lost. So when people actually try to make use of the data, they have to kind of kind of scramble and then go through various means to gather all these data. So any data that was helping a data scientist, or data engineers, or ML practitioner, to better understand the data they're looking at or finding. That all qualify as metadata.

And if you want to dig a little bit further into it you can sort of categorize them typically into three areas. There will be one that's typically about technical metadata. These are the things that people might think of metadata immediately when you talk about metadata, things like schema, the shape of the data, the quality of the data, et cetera. These are kind of like the technical metadata you gather from the source. And there's also business metadata. These are things that sort of people put context, put the data in the context of businesses, right? So it describes what kind of business entity you're talking about. Who owns the data? Et cetera. These are things that people sort of additionally augment the data in the context of a business to help understand the data.

And then finally there's the operational metadata. These are things that lineage, job that ran the status of the job. How many records was produced? And so on so forth. These are operational things that happen as you manipulate data and then give you further context about the data.

**[00:03:40] JM:** Can you give me an example of a piece of data being created somewhere deep within LinkedIn and that data being indexed, stored and registered in DataHub so that it could be searched over?

**[00:03:59] ML:** Sure. I mean, at LinkedIn – And by the way, of course, we're no longer with LinkedIn right now. So we're speaking from what we learned at LinkedIn before we left, and things might have changed since then, but it's based on our experience so to speak. So at LinkedIn, pretty much every piece of data was essentially indexed in the search index so it can be found. This includes both your online, your near-line, your offline data sets, right? So it's not just this data living in data warehouse.

And then actually majority of them are sort of called derived data, right? These are things that are the raw events and the raw user profile and what not that get processes, processed then and to transform into other things. And those are the ones that are a lot of time even harder to find because these are things that generally doesn't have a sort of a prescribed way of storing them or organizing them so to speak. And all of these things is gathered and then surfaced through DataHub to help people to find the sort of data that they would like to use.

**[00:05:04] JM:** So how does the storage system of DataHub compare to something like Elasticsearch, where Elasticsearch can index vast quantities of data? How does that compare to DataHub?

**[00:05:18] ML:** So one part ofDataHub is actually powered through Elasticsearch directly. That is kind of the main search experience as you speak. But as we talk , if you take a look at the DataHub architecture, it actually has multiple indices. One of them is the search index and the other one is the graph index, which allow you to link relationship between different entities, as we call them data assets, different type of data assets and so on. And there is also the sort of the value key document store which are serving the raw metadata without the index, right? These are raw metadata that surf through API, then people can make use of them directly. So it is kind of an amalgamation of different indices and different systems into one and surface that through an application known as DataHub.

**[00:06:05] JM:** Give me a description for the top level usage of DataHub. If I'm a data scientist, how am I accessing a piece of data in DataHub?

**[00:06:18] PG:** I can take that. So data habit LinkedIn majorly consists of two parts, right? One is the core metadata engine part and then the application, which is really the DataHub, which consists of the UI. So most of the users like data scientists analysts come to that for various applications, like one is the primary thing is search and discovery. So as a data scientist I am interested to know like, for example, anything that related to page view data. Where is this located? Like how many variations does it exist? Or what different data systems? Like do I need offline data warehouse replica or like an HDFS data set or a dashboard? Like what kind of form does it exist and like what can I leverage already what's available in the system? This is how a typical classic use case of how data scientist uses DataHub.

**[00:07:10] JM:** Let's give a walkthrough of the architectural components of DataHub and kind of explore how it fits into an overall data engineering stack. And then we can talk a little bit more about the use cases that we've outlined. So maybe you could just put DataHub in context of a typical data engineering stack with databases, a data lake, Kafka, these different things, and then talk about the architecture of DataHub specifically.

**[00:07:39] ML:** Sounds great. Yes. So DataHub by its name is the hub, and then we hope it to become the central hub of gathering all these metadata. And if you look at the data ecosystem, you have various things that either store data and at the same time metadata or produce metadata. For example, your database will be a typical place where it contains all sorts of different metadata. Your job, the airflow jar, for example, right? It will contain, will produce a bunch of metadata in the form of lineage and whatnot. These things are not necessarily stored explicitly but they are produced as the job was run.

So all these different metadata sources, they produce metadata and then send it to DataHub for indexing and processing through Kafka. So all the messages come in near real-time into the DataHub, you can, we'll call the injection layer, where these things come in and then get processed and then store in the formation document store, the data key value store in its raw format.

As it's stored, it also goes into this indexing layer where it basically index the interesting part of the metadata that you would like to search. Or if you like to traverse the graph, it goes into the GraphDB as well. So that's kind of the indexing layer. And one thing to note as kind of a major difference between DataHub and other systems out there is the stream-first approach, right? So the ingestion like I mentioned is already streamed coming in. And then you have the indexing layer, which also depends on this metadata change stream that's coming out of the system itself. So everything is stream-based if you will in that sense when we communicate different changes in metadata. And which means you can potentially also create additional application that depends on this metadata change stream. For example, if let's say hypothetically if you care about the ownership of a particular data set, if that gets changed, you want to be able to notify immediately. So instead of pulling the API and try to wake up every day to see whether the ownership has being changed, you'll get notified immediately through the change stream. So that's kind of the sort of ingestion and indexing layer. And then finally there's the serving layer of the metadata. So you have a REST-like API that basically serve these metadata for the application themselves. And then also more and more so it's moving towards kind of a GraphQL-based API because that sort of fit nicely with the metadata graph that powers DataHub.

**[00:10:18] JM:** Could you differentiate what DataHub is versus a data lake?

**[00:10:27] PG:** Sure. I mean, one way of looking at it is you can say, “Hey, look, DataHub is essentially a metadata lake where you basically put your raw metadata in and it processes it so you can then use the processed data or, in this case, metadata to further your various applications.” That's one way of looking at it, but they serve very different purposes, right? Your data ecosystem – Your data lake is the place where you put most of the time unprocessed or unstructured data for further processing. And DataHub in this sense is capturing the metadata as you sort of process your data through various data components including your data lake, right? I mean, if it's data warehouse, it's also the data warehouse metadata coming through.

So it's not easily comparable. I don't think they are the same thing. So it's not quite as though the comparison can make. It's more like a symbionic relationship. You have your data. Great, you can store as much data as you want to. But if you don't organize it and if you don't have a way of indexing it, then your datalake quickly become the so-called data swamp.

**[00:11:33] JM:** So you use DataHub in collaboration with a data lake.

**[00:11:39] ML:** Yeah. Actually, a slightly better analogy would be think of your data lake as your library, shelves in your library where you basically put all sorts of books there. But if you don't have that catalog or the index of your library, then it's going to be very, very hard to find things in there or unless understand how things are categorized or related to each other. So that way it's kind of augment your data lake and help you to get the best, most value out of it.

**[00:12:09] PG:** I just want to add. At the core, DataHub started purely as a data catalog, right? It was cataloging all the data ecosystem within your system including data lakes, data warehouses or metrics, dashboards, models, everything, right? That's how the primary purpose is to catalog everything and build insights on top of it for increasing your data productivity and governance.

**[00:12:34] JM:** How do you define the term data catalog?

**[00:12:37] PG:** So data catalog is really a kind of like an index on top of your entire data ecosystem, right? Like you aggregate like many times like based on your use cases and the technology choices and things like different organizations. Even within the same company use like different data technologies for specific things. Like for example, your machine learning architecture could be a machine learning data stack, could be slightly different than your analytics stack, or your dashboarding for business users could be very different from then your dashboarding for business metrics for experimentation and things like that. So the data catalog is something which can aggregate all this metadata into one place and like connect all the necessary relationships between them to make like meaningful insights on top of this metadata.

**[00:13:25] JM:** Got it. So I think we kind of glossed over the actual architecture of DataHubs. We talked a little bit about the ingest. We talked a little bit about the indexing, but we didn't really go into the actual architectural components and, for example, how it serves queries. Can we just dive a little bit deeper into the internals of the architecture?

**[00:13:49] ML:** Sure. Like I mentioned before, there is the API layer which basically serves the metadata. So the API layer, think of it as kind of trying to present this metadata graph back to you. So as the application itself, you can sort of traverse the graph to find the stuff that how things are related and what not. So, internally, it will sort of delegate a different sort of query pattern to different indices. Right now it's more of a kind of a somewhat manual process. As you program, you have to choose which index you have to use depending on the sort of the query pattern. So to sort of make it most efficient to answer that sort of question. But in the ideal state you want to be able to have all these things automated. So depending on the query pattern come in, you go hit different indices just to best serve these queries.

**[00:14:42] JM:** But can you talk a little bit more about like what databases does it use under the hood and those are managed?

**[00:14:48] ML:** Sure. I mean, when we design this, we try to design it in such a way that some of these things are pluggable to a great extent where you can implement specific instances of it. And this is also due to the fact that LinkedIn has a lot of its own proprietary stuff internally, right? The proprietary data technology. For example, we have our own proprietary document store called Espresso, which if you open source, then nobody will be able to use that particular technology per se. So what DataHub architecture, when you look at it, it does have all these abstraction layer where we call it data access object, DAOs, which you have right explicit implementation depending on the specific data technology you want to use. So in the open source example, the document store is built on top of MySQL for example. And then the search engine is built on top of Elasticsearch. And then the graph engine is built –So the GraphDB is built on top of Neo4j. But internally all three of them actually has its own proprietary implementation. So that's how we make this thing work between internal and external so to speak when we're working on LinkedIn.

**[00:15:56] JM:** So the data that gets ingested by DataHub is typically coming either through Kafka or through just batch jobs? How does the data get ingested by DataHub?

**[00:16:12] ML:** Sure. The standard interface is through Kafka, but there are certain things that are best written as batch, right? I mean, not every system will have a real-time metadata string coming out of it. So they are called a “crawler-like system” which wakes up regularly and then scan the system and get the metadata. But what it does is turn around and basically meet a bunch of Kafka events pretending it to be kind of a stream-based producer as well. But of course not real-time in that case. There is also API. Like I mentioned, API way of updating metadata and whatnot, but these are more oriented towards human-authored metadata, right? So people who come to DataHub and then update certain metadata that go through API. That doesn't go through stream typically. So that's how you can get the ingested metadata from various different sources.

**[00:17:09] JM:** Can you give another example of a user developing an application on top of DataHub? Perhaps a machine learning application or some application that would need to fetch more data from DataHub on a regular basis.

**[00:17:27] PG:** So a very good example, I can give a couple of them. So the metadata is like common part of like multiple data applications, right? One example is like your dashboards. So a dashboarding system at LinkedIn requires like, for example, like what are my upstream metrics and who are the owners for it or whatever is created into this dashboard? Which of these dashboards are certified or not? All this is part of the metadata. And like in the dashboarding application itself it would have like a repeated patterns of like a search index requirement or a key value store and things.

So at core, a dashboarding system at LinkedIn was also built on the same DataHub backend and while surveying a very specific use case of dashboarding. A similar example is the AI DevOps system at LinkedIn, which also requires like, for example, search indexes of all your models and what are the features connected to it or what are the data sets? Who are the owners and things? So there is a lot of repeated pattern of metadata across these systems. So they also use the core metadata engine of DataHub, which is called generalized metadata system. Yeah. So these are two examples which were very popular like beyond multiple other things which were used at like LinkedIn built on top of DataHub.

**[00:18:48] JM:** Let's talk about the company that you're building around DataHub, so Metaphor. What's your intention with the company? How do you expect to productize DataHub?

**[00:19:00] PG:** Good question. So when we open sourced DataHub, and we see a lot of patterns of requirements across metadata management and the kind of applications which come across. So they ranged all the way from like search and discovery applications, to some governance applications, or using it purely as a metadata store to power up like other data applications like what we mentioned. But as you can see like not just with DataHub, like there is a growing need for a good data catalog with like discovery aspects across the industry. You can see like other popular projects like Amundsen, which is also around the similar data catalogue and discovery space.

And what Metaphor really identifies is a vision of like – Is to bring this data cataloging or a discovery solution to multiple companies with minimal effort. Most of the open source solutions, what we built still require like a good amount of like effort in terms of integrating or even on ongoing basis of maintaining these ecosystems. So Metaphor would definitely focus as a company would focus on building a hosted, , fully hosted cloud solution which could give many more companies a direct access to a good discovery solution right out of the box.

**[00:20:21] JM:** Gotcha. And what would be the process of onboarding? Because it's a lot of data, and typical company has you know data all over the place or at least they have a data lake with a ton of information in it.

**[00:20:38] PG:** So usually like our integration touch points are based on the underlying data technologies what you have in your stack. So our primary focus would be obviously to go behind the most popular data technologies like, for example, Snowflake, or DBT, or dashboarding things like Tableau, Looker, or even like direct integrations with Databricks, kind of like data lake solutions, so that users would need to do like minimal, almost like a one click button kind of approach to integrate with Metaphor's solution so that they can get the discovery out of the box.

**[00:21:15] ML:** Yeah. And for a system that require customization, we also make it easy for you, for people to be able to publish custom metadata that are suited for their particular companies and whatnot. But that obviously will require some additional work, but we will try to make it as easy, as smooth as possible for them to integrate there as well.

**[00:21:38] JM:** Tell me a little bit more about metadata management at LinkedIn and how DataHub gets used in practice.

**[00:21:49] PG:** So at LinkedIn, it primarily started with purely from a search and discovery perspective. This is something which we talked in like in multiple places where when the data explosion started happening at LinkedIn, the primary problem there as these data engineering teams who have to answer, who produce like multiple data sets and like there are a lot of data consumers who would come back and ask like very repeated questions around these data sets and like, "Oh, where is this coming from? Or what is this schema about? Do I have this or is there any recent change?" etc.

So they created the original predecessor of data project, which is Warehouse, which was primarily to index all this information into one place and like create a simple such interface on top of it and like display this content. But as multiple other privacy initiatives came up at LinkedIn, LinkedIn took a very interesting approach of using a metadata system to tag all the compliance-related information to make like very data compliant ecosystem decisions. Like for example what kind of data is PII? What is not? What can be exposed to in terms of access? What can be transferred between one system to another system? What kind of permissions require? Etc. So all this ecosystem is kind of built around the same metadata system, which was Warehouse, which transformed into something called TMS, the metadata store.

Then the next level of like data explosion happened with rise of ML and data democratization across LinkedIn where a repeated pattern of the same metadata management requirements started appearing across like different data verticals, like ML, versus data science, versus even GraphQL APIs, etc. So we realize that general metadata management system can be like built, which can be tailored to like multiple use cases under like a repeated pattern, and hence built the back end of DataHub as generic as possible and started using it for all these use cases.

The beauty of it was – A side effect of like using a similar pattern of architecture also created helped us to create a well-connected knowledge graph of all these data assets across like different verticals. Like a typical example would be like if you are thinking about, “Oh, some even stream coming from defined as part of my user clicks and things or page views that gets translated in data lake data set, that gets translated into data warehouse data set and that is also connected to a metric, or a dashboard, or an experiment, etc. All of this metadata is kind of used and linked together to create end-to-end insights of entire LinkedIn's data ecosystem.

**[00:24:49] JM:** So let's go deeper into that example of a click stream. So I'm on LinkedIn, I'm clicking around on stuff, and it's important for LinkedIn to know that I click on this person, I click on that person, I click on this company, I click on that company, and maybe that's useful for filtering through some machine learning model to see what kind of ads to serve me or what kind of job opportunities to serve me. So that click stream is getting created. Maybe it's getting dumped into Kafka. Kafka is dumping it into a data lake. Maybe some job is getting kicked off for that data lake click stream to be put into the data warehouse. And you're saying that all along the way there is additional metadata attached to this click stream information that might, for example, identify it as PII, and this kind of metadata would be indexed in DataHub somewhere along the way. And this indexing in DataHub is going to provide utility in a number of different ways.

**[00:25:55] PG:** Yes. The very typical name for this in the metadata ecosystem is lineage. So every data is set all the way from a click stream to a Kafka data set to all the way into an ML model is kind of linked to like something called as data lineage, and at every stage like different types of metadata. Sometimes it's including PII, like compliance-related metadata. Sometimes it's ownership. Like, for example, if you see some changes at, let’s say, Kafka level and you want to talk to an owner like what is the reason that changes occur. You would need ownership metadata at that stage. Or you are looking at some delay in processing of a delta lake data set, you would like to talk to the operator of that data set like, “Hey, why there is like operational delay in this particular aspect?” Or a quality check with respect to a metric, like you would like to talk to that particular subject matter expert or something like that. So different types of metadata are attached to these different data assets and you build like an end-to-end relationship graph of the entire life cycle of your data ecosystem. Then that gives you like an end-to-end lineage.

**[00:27:08] JM:** Got it. So lineage is a problem that DataHub also solves. We hadn't touched on that previously.

**[00:27:15] PG:** Yes. Actually, very interestingly, we use the word in general the knowledge graph, which kind of subsumes the lineage part of the thing. So if you think about knowledge graph, like take all your data nodes, and there are key players like, for example, people, like different roles of like ownership, or subject matter experts, or operators, or consumers at each of these data sets and you connect all these nodes together. That gives you a knowledge graph. If you take like one particular item and like traverse it like upwards or downstream, that becomes your lineage. So like in a way, lineage is kind of like a subset of your knowledge graph.

**[00:27:58] JM:** Do you have a variety of ways that DataHub can accept queries? So there are document-oriented queries, graph-oriented queries, complex queries that involve joins and full text search. Can you tell me about the query patterns of accessing data in DataHub and what you've had to build to satisfy those query access patterns?

**[00:28:26] ML:** Sure. So I'll take uh take it to a little bit high-level in terms of categorizing the query into the source that are sort of generating these queries rather than the exact query patterns. So there is one source. I think it's pretty apparent at this point that there is the sort of the human-generated query, right? These are essentially people clicking through the DataHub web app and then trying to find out things. And these things are typically – There’s a search aspect of it, which obviously heavily skewed towards the search index. There is a lot of debugging related stuff, like the linear stuff that we do mentioned before where you have to trace back and see where did something break and all that. That is more of a kind of a graph oriented query. And of course the raw kind of the document faction, all that, that is always going to be there almost in this case.

But one thing to note is this particular source of querying also a lot of times include mutation changes as well, because people generally are the one that's going to really come in and then enrich the metadata through the web app in most cases. So that is one area, one sort of source of query that comes through DataHub. There is also the programmatic patterns, right? These are things that like Pardhu mentioned before, compliance was a big use case for DataHub where there will be jobs and services that essentially have to query the metadata in order to perform certain work. And this sort of thing generally comes more like a sort of a key value retrieval sort of pattern. You already have a very clear idea about the sort of things that you need metadata for and then you just go and fashion the metadata.

Occasionally, there is graph involved when the metadata need to be propagated through various lineage and whatnot. But most of the time, it's more of a kind of a key value retrieval sort of pattern in that case. Your typical REST pattern if you will. And then there's the third pattern which is kind of the analytical patterns, and these are things that you're trying to ask kind of a global questions and trying to perform most of the time asynchronous actions, right? It's okay to take a while to analyze most of your metadata.

And the way DataHub doesn't directly serve these sorts of patterns through its API. What happens is the various data in DataHub get dumped offline to your data lake and then the analytical work sort of work through those data warehouse data sets. And then at that point all sorts of different query pattern, but the performance is not an issue generally in those cases. So those are kind of the three main sources of query pattern that comes through.

**[00:31:09] JM:** How does DataHub compare to the other kinds of metadata solutions that have been developed at companies like Airbnb, or Lyft, or Google?

**[00:31:23] PG:** I mean, if you're talking about like Lyft's Amundsen, or Airbnb's Dataportal, I think DataHub is very much in like similar area, but the only difference was that the primary use case what it solves especially in the open source wall is search and discovery. But the primary difference is DataHub's backend architecture is kind of really next generation level of architecture. There were recent blogs published from LinkedIn as well on this aspect how it compares with like all these solutions out there and how its architecture is like evolved on top of like flaws of like existing like use cases or existing architectures, what we did. In comparison to like other architectures, like this is definitely the next generation. Trying to solve the downsides of what we have experienced building the previous generations of metadata management solutions.

**[00:32:21] JM:** So one thing I'm still a little bit confused about is the difference between DataHub and a traditional data catalog. If I'm looking for a system to index the fields of my data, isn't that the same thing that a data catalog is doing?

**[00:32:44] PG:** Yes. And the primary difference, like I said, DataHub at its core is still a data catalog, right? But the difference is that beyond just indexing the things, you would also index on top of like relationships and things trying to create that knowledge graph and so that you can build like multiple different applications. Not just a search and discovery application.

**[00:33:08] JM:** I see. So it's the interface that you're building on top of it, the data access interface that you're building on top of it that is trying to be the differentiation.

**[00:33:19] ML:** And also just to add a little bit. So the traditional data catalog lacks a couple of things, right? And typically it doesn't do automation very well. So there is a lot of manual work involved in registering data sets and whatnot. Some of them obviously evolved since then. But one thing for the sort of next generation data, kind of sort of the requirement there is to be able to auto ingest metadata from various sources as quickly as possible, as near real-time as possible. But the other thing is sort of the dimension of the breadth, right? So a lot of data catalogs are very much focused on one thing, right? For example, a dataset catalog. So I'm going to catalog just the data set. Or maybe a specific area of metadata for that I think. For example, if I'm a data quality data catalog, I will obviously focus just on the data quality angles of it. Or maybe I'm an axis granting data catalogs, I'll focus a lot on the access side.

But what we discovered over time is that to truly make a data catalog useful, you kind of have to give this 360 degree view of the things that you care about, and that means also, like Pardhu mentioned, the relationship to other data assets beyond just the pure data sets, right? The human aspect of it, right? Who's creating it? Whose technology experts of it? The actual knowledge itself, all the conversation that had happened related to this thing. All of that need to be presented in a way that the user will immediately – Help the user to immediately understand what is going on. So the sort of the next generation data catalog is both responsive. Sort of reflect the latest state of your data ecosystem as quickly as possible, but also rich in that sense it will give you all sorts of different metadata from various different sources to help you better find the things. Or once you found the thing better, understand what state it is and so on.

**[00:35:20] JM:** What happens when a component of DataHub fails? Is there some mechanism for sustaining the system when the flows of data get stopped for a moment?

**[00:35:34] ML:** So I think this is where the sort of stream-based approach helps a lot, because you are able to – For example, if your injection pipeline go down, if you have a pure API-based process, then of course you cannot ingest any data and then where does those data go when the system is down, right? So having something like a stream that allows you to queue up these things and process them based on your processing capability, that helps you mitigate a lot of these sort of transient issues. But then at the same time a lot of these systems that we use are kind of highly available to establish, right? Your search index, your GraphDB and your documents, these are highly available. So we generally don't face any issue in terms of reliability on the raw storage layer per se. But of course things can go wrong at a higher level at many levels. But having sort of this asynchronous nature of the system allows you to sort of isolate yourself from points of value in your systems.

**[00:36:39] PG:** Just to add on top of it. The ability for DataHub to support both batch system updates as well as real time updates even if there was any down time in terms of real time updates on things, there is also like a capability to update all of it through your batch crawler and things. It would still be the same. And both the search index and the graph index can be rebuilt on top of your – What's in your key value store information. So there are like multiple levels of like redundancy and like replicability built into the DataHub ecosystem as well which helps that to be like an operating as a separate metadata layer on top of your actual data sources.

**[00:37:21] JM:** Could you give more detail on the stream-based approach that you've alluded to a few times and more context on the advantages of it?

**[00:37:32] ML:** Sure. I mean, at this point, probably all your listeners who are very, very familiar with Kafka and the advantage of Kafka anyway. But, I mean, stream-based in general give you several advantages, right? So first of all this asynchronous nature is of it. So the producer of the data doesn't have to worry about whether the consumer of the data is ready to consume per se. You can just put it in the queue and then just walk away. There is also this buffering effect. So that smooths out spikey traffic, right? So you don't have to worry about provisioning your consumer so that what is API and what not? And to make sure that you can cope with the peak traffic that will be produced by your producer.

And then, of course, Kafka itself is very, very scalable. The stream itself in general is pretty scalable in that sense. You can easily put in more shards and then you get parallel processing going on pretty easily. So there's several advantage in going through that. And then finally of course the example that I gave for the previous question to do was downtime, right? Once you put in the queue, then you can process it at a pace that you want to and then your processor can easily – If it goes down and has to come back up, there will be no direct impacts on the producer. So this dependency isolation is definitely a huge strength of the stream. And of course stream is designed for real-time as that, of course, go without saying that that is a huge advantage over something that is less real-time.

**[00:39:00] JM:** What's been the hardest part of designing DataHub?

**[00:39:05] ML:** I think in the case of DataHub inside of LinkedIn, it is a process of evolution, right? So we started off with something really simple and then eventually evolved into something that's generic that can support many things. And throughout this entire process, even though it seems as if we were rebuilding our architecture over and over again, it's a continuum. It's always an evolution. So we never say, “Hey, we have dual stack,” and then you know have dual stack working and then turn off yours. It's always the old stack being gradually evolved into the new stack. And that has of course has its own advantage in disadvantages. And the challenge there, the biggest challenge there is how do you make sure as you evolve you don't break people. That is always a very hard thing to do. When you're able to start something that's completely green field, and then of course you have a lot of latitude in terms of breaking changes and whatnot. But if it's in continuous evolution, evolutionary process, even though you have less disruption, but then you have to basically make sure that it's always backward compatible and move people, various player in your ecosystem along with you as you have to make those – Inevitably have to make those breaking changes.

**[00:40:27] JM:** I’d love to hear a little bit more about the company. So as you're starting to talk to users or potential users, what are the problems that they have that they're hoping to have addressed by DataHub or by Metaphor? Or have you started to have proof of concept stuff with customers yet?

**[00:40:49] PG:** Kind of. Like we have been continuously engaging with bunch of customers like beyond what we have seen within DataHub or with like connections like from understanding from LinkedIn. So an interesting pattern is that like multiple companies all the way from like SMBs to like big enterprises are going through a data explosion phase. So like maybe the last decade was all about like data democratization. And we came up with great technologies in various places to enable multiple data users across the company to create new data insights or new data artifacts or assets. But this also created like a second degree of problem where too many data assets in the ecosystem are needed like a little bit more control like or a governed way on top of like how the people identify this existing data sets or how they use or how they interact with between data users and data producers.

So this is where Metaphor would like to focus, like bringing the discovery solution as part of its first phase to multiple of these companies and solve, improve the data productivity and data trust or understandability space for the companies.

**[00:42:09] JM:** Well, I'd love now to zoom out and get both of your perspectives on the data engineering landscape as a whole. What do you see as changing most dramatically in the next five years?

**[00:42:22] PG:** I think it's a hard question, but one thing for sure is the popularity among technologies like Snowflake and Databricks, which is traditional like data warehouse versus data lake thing, is kind of heated up, right? So we can see both the ecosystems trying to like solve like use cases which were across like data science analytical all the way into machine learning models and like AI-based use cases. So while these things happen, people would like standardize a little bit more on Datastacks across companies. And this this would also mean that like a lot more users would be like creating lot more data assets and like important aspects like quality or observability or data testing, like governance. These things would become like much more important aspect of your data decisions or data ecosystem not just like creation of the data.

**[00:43:25] JM:** Well, thank you both for coming on the show. It's been a real pleasure.

**[00:43:26] PG:** Thank you, Jeffrey.

[END]