Transcript provided as a free resource by 

Data Mesh Radio Episode #46: Designing a Data Literacy Approach for Data Engineers

Interview with Interview w/ Dan Sullivan

Listen (link)

Transcript provided as a free community resource by Starburst. To check out more Starburst-compiled resources about data mesh, please check here: https://www.starburst.io/info/data-mesh-resource-center?utm_source=DataMeshRadio

Scott Hirleman

A written transcript of this episode is provided by Starburst, for more information you can see the show notes.

Starburst

With a massive move to distributed data architecture, it's essential to have access to all of your data wherever it is. A data mesh emphasizes domain driven data ownership, data as a product, selfservice infrastructure and federated computational governance, giving you faster time to value without needing to transport your data. Starburst allows you to achieve this distributed architecture by allowing you to run SQL queries across distributed data that connect sources, regions and clouds. For more information on how your team can benefit from a data mesh strategy, check out our data mesh resource center on our website.

Scott Hirleman

Welcome to Data Mesh Radio, produced and hosted by Scott Hirleman, the founder of the Data Mesh Learning Community. Data Mesh Radio is a vendor independent resource for learning more about data mesh. Let's jump in.

Bottom line up front, what are you going to hear about and learn about in this episode? I interviewed Dan Sullivan, Principal Data Architect at 4 Mile Analytics. I asked Dan to be on to chat about data literacy as he has put together a number of online classes around database and data engineering topics. Dan's three key pillars for driving data literacy for data engineers are: one, domain knowledge, two, learning and three collaboration. Data engineers should pair with business people to acquire domain knowledge. They should be given the opportunity to spend time doing things like online training so they can actually learn new skills and new approaches. And they should collaborate across the organization instead of just being ticket tacklers.

Zhamak has mentioned in a few talks that data engineers, at least as many are currently used, may not be a thing in data mesh going forward, and possibly in the

whole industry. There's a lot of people I talk to who behind closed doors will say that they don't think that data engineers will be doing what they're doing or exist in something like five years. Dan actually somewhat agrees, as right now, there's a big rush to build out the initial iterations of data products in the industry definition, not just the data mesh definition of data products. Going forward, Dan thinks there will be a need for data engineers that can really understand consumer needs and build the interactions such as the SDKs to leverage data. It might be like what happened with distributed systems engineers in the early to mid 2010s, and what is starting to happen with streaming data engineers now. Per Dan, not all data engineers are the same depending on background. Some come from a data analyst/data science background, but many come from a software engineering background, so we can't treat training all data engineers as if it's the same, but we do need data engineers to have a well rounded understanding of what's going on.

So a big need is for them to understand more about the data consumers and or the producers, so embedding them in the different domains can really help to round out their understanding and their experience. For driving buyin with data engineers, Dan points to the problems typically being around incentives. Data engineering is often also hampered by organizational issues and a lack of clear direction. So if you can tackle those, you can often win over the data engineers. In any organization, but especially in one implementing data mesh, standards, protocols and data contracts are all very important, however, most data engineering teams are not given the time to actually create those and really manage those well. They take a lot of effort and are hard to get right. So we need to give people the space to really tackle those different aspects. A very, very key point that Dan brought up is around tech debt and data. Taking on tech debt should always be a very conscious choice, but the way most organizations work with data is much more of an unconscious choice, especially by data producers who are often taking on that tech debt that the data engineering teams will have to pay down. So again, what choices you're making should consciously lead to what tech debt you take on, but when it comes to the way most organizations are handling data, that's not at all the case and we have to move past that.

We have to find ways to deliver value quickly, but with discipline, so that we aren't just taking on, especially unconscious, but even the conscious tech debt when it's not necessary. Dan also talked about how data can take a lot of useful practices from Agile, especially the fast cycle feedback loop, and that data people really need to think more about the user experience for data. The interview that I had with the predictive UX folks, Karen and Steve, talked about this too. There just really isn't nearly as much of a necessary focus on user experience when it comes to data. And to do data mesh right, or data in general right, that user experience really does need to be part of what you're considering. So I think you'll get a good approach to

working with data engineers to get them to a place where they need to be to really help implement your data mesh or whatever data projects that you're looking to kind of move forward with. With that bottom line up front, done, let's go ahead and jump into this interview.

So very excited for today's episode, I've got Dan Sullivan here, he's the Principal Data Architect at 4 Mile Analytics, and Dan has come across my radar multiple times over the last couple of years for a lot of different things, Dan has put out a lot of really, really good content around data and things like that for training people. And so I first became aware of him relative to Apache Cassandra working at DataStax, but he's done a lot of things around helping people learn around data topics.

And so what I wanted to get him on today to talk in general about data literacy, but especially as what we're seeing, Hello Fresh had done this on their meetup, they talked about how they've got their internal data literacy program. I think one thing within data mesh that is pretty easy to overlook is that you have to train your people to be more informed around how to use data to inform their own decisions. So how do we get people to that capability to actually use data? If you just give them access to the data and don't tell them how to use it, it's not really gonna go all that well. They're not gonna be able to really provide a lot of value from that. So with that as kind of the introduction, Dan, if you don't mind giving folks a bit of an introduction to yourself and your background, and then we'll kinda jump into the topic at hand.

Dan Sullivan
Sure. Oh, thanks Scott. Yeah, and first of all, I appreciate being here. I really enjoy talking about this topic. Yeah, so as you mentioned, I am a Principal Data Engineer with 4 Mile Analytics. 4 Mile Analytics is a Google partner, data specialist, so we do a lot of work in data analytics, analytics engineering, data engineering. And I'm relatively new to 4 Mile. Prior to that I worked in FinTech, and prior to that I was at a company called New Relic, which is an application performance monitoring company. And so as you might imagine, they deal with a lot of data, large volume, very high volume data ingestion issues and analytics problems. And prior to that I worked in genetics research, mostly in Infectious Disease genomics. And prior to that, I just did a lot of Oracle Consulting, a lot of data warehousing, going back to the early Kimble days and before. So that's the short version.

Scott Hirleman
Yeah, so you've been on the analytics side for quite a while. It's funny, Tim Tischler was just on. For people who will be listening, the episode isn't out when we're actually recording it, so Tim was also at New Relic, so I think it's something that's very interesting that that space around, Hey, we've gotta get our arms around what's actually going on, kind of transitions well into data mesh, because of all the DevOps

type of concepts that really flow well into data mesh. And that we're trying to make this not something that it's that continuous delivery of data is an interesting kind of approach that I don't think people have really thought of before. A lot of times data is a one off delivery because it's a one off kind of transformation that we're not creating this in that kind of factory way of we're treating how we create data, so.

Dan Sullivan
Yeah, it's really interesting. Yeah, 'cause there's sort of like two sides of this coin, where there's what you just described, it's like we're creating all this data, and it's not just... And certainly with things like New Relic, but also IoT and just other sources of data and just like Clickstream data, everything we do on the web, many of the things are tracked. So yeah there's this constant stream of data, and so some companies are really good at capturing that and collecting it. And then the flip side is, well, okay, now we have all this data. What are we gonna do with it? And how do we use it?

And I think this gets to the point you were mentioning earlier, it's like, how do you start to think about data? And as a data engineer, I tend to think about things like from the ingestion point of view, it's like, okay, how are we gonna scale this up? How are we gonna keep low enough write latency so that we can keep up with an ingestion pipeline and things like that. But that really has nothing to do with people at the other end of the data production and use pipeline who are trying to make decisions about, Well, how should we tweak our product? Or what's the best way to appeal or break into a new market? And things like that, and these are all questions that are really important from a business perspective, but they have nothing to do with low latency writes to Bigtable or Cassandra.

And I feel like sometimes we live in two different worlds, like those of us in data, like sometimes you're in the data analytics camp, and sometimes you're in the data engineering camp. And in fact, I was just talking to a data analytic manager earlier today about kind of bridging that gap. Like how do data analysts become more like data engineers? And I think it's equally important, it's like, How do data engineers learn more about what data analysts do and what's important? Because with so much data, basically, whatever data you want or whatever topic you're interested in, you could probably go out and find data about it, and the question is, so what? Of that data that you find is going to be useful for whatever you're trying to do, and how do you link that up?

And that's really... For me, at least right now, the really interesting kind of touch point is like thinking... I often spend a lot of time thinking from the data engineer, ML engineer, data architect perspective, but also it's like, well, from a consumer of the data, why would they care? Like why do they care about what kind of clickstream data might be available in the data lake? And so getting from the business problem

to using and consuming data is an important step, and it's like, how do you formulate the questions or why would you even formulate it in terms of data? And if you're a quantitative person, you probably had a statistics course at some point, or you had a Coursera course on stats and you understand some basic descriptive statistics, and that might be a starting point, but maybe that's not...

Maybe you don't really understand, like, what's a normal distribution? I keep hearing this word distribution, what does that really mean? Why do I care? And stuff like that is kind of like... That's somewhere in that middle ground between data engineer and data analysts and building the business question.

And so it's really... I find this hard for me to think about, how do you talk to somebody at the data consumer end from their perspective and walk them back, walk back with them to, here's how you use the data, this is how you formulate the question, if you have this question these are the kind of statistical tests you might wanna use. And then this is where we'll go get the data and kind of build the bridge from where they are to where they wanna be to answer the question that they're interested in.

Scott Hirleman
Right. Well, and then there's the other aspect as well of teaching the application developers about how to actually deal with data, how to serve data, what data engineers have historically done of just trying to ingest and then actually formulate the data from there versus pushing that to the far left and saying, how do we train them as well? We can jump into both sides of helping the... Or all three, because as you said, the data engineers need to be able to understand more about the application side, especially in data mesh, to enable the team, whether you're embedding a data engineer in that team, or you're just, the data engineering team is there to create the platform or however that works. Or Zhamak has talked about maybe five, 10 years down the road, do we even have data engineers? Because so much of what data engineering is is maintaining the status quo of making sure this stuff is still flowing, and if we have enough tooling that prevents us from having those problems when we have one off very, very high scale stuff. Sure, we might break that down, or we might have people that are very, very dedicated to that, but that creating the pipelines and stuff should not be nearly as hard as it is and maintaining those and things like that.

So would you prefer to start with helping the application engineers really understand how to serve data with the data engineers of both sides of getting closer to the consumer and getting closer to the producer or the data consumers, and really training them up on how to use data and inform their decisions?

Dan Sullivan

Well, how about if we start with the first one, with the data engineers and how do we think about the consumers and how do we get there? And I think you're absolutely right about data engineers, we're needed now, it's like, this is like a big public works project and we're gonna build the interstate highway system in the United States, kind of level of investment with building data products. But yeah, after a certain point, it's gonna be like water infrastructure in a country, it's just like the pipes are built, the water flows, the water will keep flowing and there will be issues, things will break, but for the most part, it's gonna be like, we're not gonna think about it.

I think there will be a shift away from kind of data engineering as we do it now, just as we automate more things and we get better at developing patterns that we see are really useful. I think one of the things that I have trouble with is when we're building data products is thinking in terms of the person using like a developer who's gonna use this data product. So there's some API and they're gonna send a request and I'm gonna send back some data, and it's like, well, what is the data, what's the conversation we're having? Because that will determine our vocabulary and how we structure the data and how we organize it. So what kinds of sets of data do we pass back?

And so I think initially a lot of the work for a data engineer is learning about the needs of the consumer and kind of crafting APIs around that, and then on top of that, building things like SDKs or libraries that make it easier to use the APIs.

I feel like that's where more work will continue. I think the bulk of data ingestion, moving data around, getting it written to a data store, something we can query with SQL, there's definitely a lot of need for that right now, that's the kind of data engineering that's going to taper off and become more automated and easier to kind of scale up with a small number of data engineers. Kind of like operating a Public Water Works System or something like that. There aren't that many people that have to work on it day-to-day. You know if you get closer to the consumer, we need people working doing plumbing at the finer grain, detailed level all over the place, like just that's a constant need. I feel like that's gonna be something similar, we'll see from data engineering, we're gonna be more focused and more tightly coupled with people who are the consumers of the data.

Scott Hirleman
And how have you thought about with the data literacy which is what we're talking about today. How have you found or what have you found is useful to approach them with from an information standpoint and a training and teaching standpoint, is it that embedding them into the teams, is it just putting them through a lot of classes, or what have you found that makes it so that they can actually have those useful conversations with consumers. I've had a few episodes around domain driven design

for data, and a huge part of that is just getting people to exchange information in a language that both can understand, and so how do we get the data engineers to really understand what the consumers need, because within data mesh that we are putting those data engineers into the domain teams, or is it that they're building the platform or whatever. But really understanding what people need. How have you found that to be... Like what have you found is effective and maybe what's not effective, so people can go down and prevent themselves and going down that path.

Dan Sullivan
Yeah. Yeah, I'm not sure. I have things that I would definitely say, yeah, do this and don't do that. I don't have... 'cause it's still... I feel like this is still early, but I do feel that it's almost like there's two kinds of or two broad types of data engineering tasks, there's the building the heavy infrastructure for moving the data, persisting it, doing data lifecycle management that we need to focus on. So there might be people that are really good with cloud data flow and Google, which is like Apache Beam building batch and stream processing pipelines. And then there are other data engineers who are more focused on delivering data sets tailored to consumers, and those people, I think really need to be embedded with the domain experts, the people that they're going to be serving.

And I've seen this, especially working in life sciences, where you have people who have backgrounds in Computer Science, we could have a computer science vocabulary natural to us, and then all of a sudden you're sitting down with people who are geneticists or microbiologists, and they speak a totally different language. And when they talk to each other, there is almost like a shorthand or there's an understanding that may not be clear to us, the people outside the domain, and it's really crucial for someone who's trying to tailor data products and build like APIs that you really understand fairly deeply, what's going on in that domain, so I think it makes perfect sense that domain driven data modeling keeps coming up around data mesh, I think that's like a key tool and methodology that we're gonna be using and depending on quite a bit, so.

Scott Hirleman
If people are saying, okay, I'm going to try to train my team on this stuff, how do you recommend they set that up? Is that something where every company has to build all their own stuff internally or... Obviously, as somebody who delivers classes, you probably want people to go and take your classes, but you know what I mean. Is it that there needs to be specific tailoring or is it more pairing or what have you found is how those people can actually get to... 'Cause data engineers, you immediately assume that their data literate and it's like they kind of are, but they kind of aren't, and so how do you get them to know what they need to know, what have you found is effective there?

Dan Sullivan
Yeah, that's a great point about the data engineers are not necessarily data literate, a lot of data engineers come from software engineering, and so people with background in software engineering, really good in CICD, they know GitHub, they are very good at software engineering practices, are well versed in design patterns and things like that, and maybe not so much on statistics side. So yeah, I think that is a really important point that we do not think that data engineers are also people who consume data and manipulate data the way a data science person does. I think the critical thing for getting a functional group is that you have people that, first of all, they have some specific functional knowledge, like the data engineers understand data ingestion and data modeling and things like that, so there's a specific knowledge, but also the ability to learn, and this is really crucial, and this is one of things like sort of one of reasons that I like certifications is that it's like a concrete thing that people have to demonstrate, yeah, I can learn. So next year, when something new comes up, yeah, I'll be able to keep up with that.

And then the third leg of the stool is people on a team need to be able to collaborate, so I'd say the first thing is, you wanna put the right people into this position, because if someone is not really wired to be very collaborative, that they prefer to work by themselves and do specific tasks and things like that, that's great but they're probably not going to be able to go through a training course for a couple of days, whether it's in company specific or a general... Take some stuff, courses on LinkedIn Learning, you're not gonna come out the other end of that and be able to function well in a team where there's going to be a lot of uncertainty, there's... Depending on the domain, it could be a fairly long ramp up time before you actually understand. So I'd say the key is really focus on the team and making sure you have the functional, like the knowledge domain is covered, but the other two pieces, the learning and collaborating are really key sort of personality characteristics or skills, I'm not sure what they are, but those are must have.

Scott Hirleman
Okay, so you'd say it's domain knowledge, learning and collaboration?

Dan Sullivan
Yes, yeah, those are really sort of the key things. And for domain knowledge, none of us are gonna know it all, so that's why we need to collaborate, and it doesn't matter how long we stay in this field. I've been doing this for decades, and there's no way I don't know. I know a small fraction of what I would need to know to do anything interesting in the work that I do, which is why I end up working with teams all the time. So I think we need to recognize that as well. Both on ourselves, but also people like if you're in an organization, you're a manager and you're trying to put together

teams, I think it's really important to be aware of the mix of skills in terms of functional knowledge that are required.

Scott Hirleman
Do you have any advice, if someone were to ask you, How do you accelerate the domain knowledge acquisition? Is that just kind of pairing with people, is that trying to be in a lot more business focused meetings that have absolutely nothing to do with the data? If you were to say, How can a data engineer get to a reasonable useful level of domain knowledge?

Dan Sullivan
Yeah, I would say for me, and people have different learning styles. It's kind of multitiered. There's definitely the orientation with the business people to kind of understand what's the scope, because it really doesn't matter what field you're working in, you could just walk in and say, Oh, I need to know about X, and X could be commodities trading data sets or something in life science or something in marketing, with direct to consumer marketing, any of those fields are so broad you could spend a lot of time like trying to get a foundation and sort of lose the marginal gain that you get from working on your own. So I feel like the first thing is get a sense of what's the scope of what you're trying to understand and be in meetings where people are discussing, like other marketing domain experts are talking about what it is, so that you're starting to learn the vocabulary because that'll help you when you go back and you do the, Okay, now I need to actually go drill down and learn what do these terms mean, and what is the data associated with these? And so I find it's a mix between being around the culture and learning what the specific subset of things that we need to do, sort of book learning, online learning, going and finding YouTube videos on this particular topic, and then also working with the data.

Again, it's like those three things, and you kind of shift depending on where you are in your learning cycle. Sometimes some of those are more important than others, but definitely start with the people you're gonna be working with would be my suggestion.

Scott Hirleman
Well, I think we've given a good path for people to kind of help their data engineering team. Which would you prefer to talk about as well on the getting data consumers up to speed, and you talked a little bit about actually, leveraging the data, but I think part of data mesh is about putting data in front of folks in such a way that they can answer their questions without having to do like the heavy statistics or the heavy SQL and things like that as well. Or the other side, which I think is one that's really weighing on a lot of people on data mesh is how do we get the domains, not just getting them bought in that they want to share their data has been a little bit

difficult for a lot of folks, but just even understanding or getting them to understand how they could appropriately share that data. Because some of it is people are like, hey, you need to be able to share this in a super complicated way so that our data scientists can use it today, and it's like you gotta kinda build up to that, build that muscle and things like that, but which you think is is that you'd rather jump into more.

Dan Sullivan
Yeah, how about if we jump into that latter topic about how do we get data, how do we get people to buy in, because this is really like a... It's like a social good, it's like you're gonna contribute something. You have this really valuable data that you use for whatever it is that you work on, and it turns out that that data is also useful in different ways when combined with other data for other people, so we have a problem around incentives because if you in your organization, if you're incentivized to just work with your data as you are and produce what you're producing, but there's no incentive to share that, then that's a problem because you're gonna focus on where you basically have been told by the organization, this is what we want you to focus on.

So one of the problems at a certain level, it's an organizational issue, and how do you as an organization reflect the value of that kind of data sharing and collaborating at an organizational level about sharing your data, and so there need to be incentives or there needs to be clear direction or strategy or kind of acculturation about this idea that, No, data sharing is really important.

And so part of it is we as the data people working with the data, have to buy into the idea, and that's an organizational problem, and then we need the means to do it, and this is, in a lot of ways, this is the hard data engineering part because this gets also into collaboration, because now what we're talking about is we're talking about standards and protocols and basically coming to agreement and making contracts with people that this is the way we will share the data, and I know we've talked previously briefly about like in the field in Infectious Disease genomics, one of the reasons the COVID-19 vaccine was able to develop so quickly is because there are data standards around sharing genomic sequences and understanding the data and how you can upload them into repositories and other people can quickly get access to them, all of those standards, it took years to develop those, and so what we're doing sort of every day where I work is kind of doing that at a microlevel within an organization, trying to come to agreements about how we're gonna share data.

So I think we need to understand that that is a piece of work. And it's not something that's necessarily gonna be done in one or two sprints, and it's not gonna be done by just one or two teams. This is actually more like, how do we extend Agile

methodologies to easily accommodate put structure around this kind of work that we need to do.

Scott Hirleman
Yeah, I think that's really important when you... The through lines that I see in a lot of these conversations is... It's kind of like the, So tell me about yourself. That's kind of just the worst interview question because there's no direction versus if you give people a, hey, here is a couple of different ways that you could do this, and you give them the agency to choose how they get to that, that works much better, but if they have to invent everything from scratch, if they have to invent their own standards, and then every domain is gonna have their own standard and it's not gonna be usable across, but it's also gonna be a much harder mountain to climb because you're not giving them the resources to actually do it. It's a major pain in the butt for them because they're just like, I have to spend so much time doing this, this toil work that if you provided a blueprint, it really would, but how have you seen educating those folks? Again, we're trying to talk about the literacy angle, and literacy isn't just reading and things like that, but when you are talking to data engineers that maybe came from the software engineering world, we're trying to teach data to the software engineers in data mesh when you're teaching data engineers about data and how to do that, how do we extend that into the software engineers, the application engineers, to give them the understanding of data, what have you seen has been effective with that?

Dan Sullivan
Well, I think there's a couple of pieces. When it comes to the architecture and how we're building the pipelines and where we're getting data, I think it's really helpful talking to software engineers in software engineering terminology, so it's really easy to talk about tech debt and software engineers completely understand. The idea that I'm gonna just plug this thing, I've got this need, I'm gonna fix it with some code, it's not great, it's not really efficient, I don't wanna do it this way, but I'll come back and fix it later because it enables me to do other things that I need to get done to meet some particular deadline, and with data, I think what we haven't realized is that by thinking that we're silos and, Oh, I'm just gonna go somewhere, I'm gonna go to the data warehouse or I'm gonna go to an OLTP system and get some data out of there and build my own pipeline and produce my report or my dashboard. We're basically accumulating tech debt or data debt, but we don't realize it, because what we really need is like a platform in the middle that is a way of dealing with large volumes of data and governing it and making it accessible and searchable and secure.

And so I think talking to software engineers about tech debt and why we build... Sometimes we take shortcuts and go around platforms that are maybe like the unifying platform to get data just to get something done, but really what we wanna

be doing is building, you know expanding on a platform that will be usable years down the road, and again, going back to the life science example, you put the investment upfront, it's gonna pay off later. And so part of it, again, is just like understanding at an organizational level that we need to be building these platforms, software engineers understand this idea of platforms, and it's like, Oh okay, we're gonna have some APIs and SDKs.

There's going to be common design patterns that we use, so I think talking in terms of things like design patterns and common interfaces really helps because it's basically a software issue then, and then it's like, and data is just the thing that's going through all of these pipes. So that's one piece, and that's sort of the easy side, 'cause it's very easy to talk about data in terms of software kind of methodologies and practices. And then there's the side of thinking like the data analyst or the data consumer, and that's a little trickier because now it's like, okay, we basically have to try and empathize with the person who's using our system, and this in a lot of ways, it does have an analog with software engineers, especially those who do like front ends and are dealing with user experience and things like that, and that is definitely an area I am not good at.

I couldn't design a decent user interface, if my life depended on it, so I don't wanna speak to that any more directly because I'm sure I will do a bad job, but somehow there's that idea of being able to sit next to the consumer and put yourself in that data consumers position. And if you think about it from the data consumers position, well, they want this data for some reason, so the big question is, start with that, what's driving them to come to look at this data to begin with. So they're clearly data literate, they understand that the data can help them solve a problem, so I think we as data engineers, the first thing we need to understand is what is that problem that the person is trying to solve, and then we're kind of like the bridge builders between... It's like, Okay, I see what you're trying to solve. And then if you're a data analyst, you probably understand enough about statistics and modeling to be able to start to formulate in your mind, it's like an approach to solving that problem, like building a model that will solve that problem. Doing a statistical test that will give us the answer to a particular question.

And if you understand that then you can start walking back to the more of the data engineering kind of questions like, What data do we need? Where is it located? What's the latency? How much is it updated? How often will I need to rebuild this model or re check if we're doing some kind of a statistical test? And think in terms of that. So yeah, I think a lot of empathy and just thinking, starting with the business question that's driving the consumer, put yourself in their shoes, I guess is what I would suggest.

Scott Hirleman
I liked what you said about the tech debt without realizing it, 'cause I think this is the thing of the enterprise data warehouse is just, even before it's deployed, it's tech debt And that's what data mesh is trying to do is remove that from an Agile perspective, but that you do take on tech debt, but that it's a conscious decision because that's exactly what taking on debt should be. It's not that you go... If you see the horror stories of 18, 19 year olds, they get a credit card and don't realize that they have to pay the money back and things like that, right?

Dan Sullivan
Right. Yeah. No, that's a great analogy.

Scott Hirleman
So we need to really help them to understand where can they take on that tech debt and where shouldn't they and how to manage it going forward and things, and that we give them the tooling and put it in their language, as you said, and that kind of UX for the consumers, the user experience, it's funny how many of these conversations, it's that either it was that people are playing telephone through the data engineering team. So the producers were talking to... The consumers come to the data engineering team, the data engineering team goes to the producers and says, "Hey, here's what we need", and then they do the transformations when they get the data ingested into whatever, whether it's the lake or the warehouse or whatever. And then they pass that on to the consumers. The consumers are like, "Well, this isn't what I really wanted. But okay, it took me three, six, nine months to get this, so I'll just use, I guess, what I've got."

But just getting the consumers and the producers in the same room to figure out how to share that common language. That it's not that you're going to be able to pass tickets back and forth and understand each other. And you talked about something a little bit earlier, which is kind of against where I think Zhamak was seeing the original idea of data mesh, but it's where I'm seeing a lot of people are moving forward, which is having those one on one, you don't think of... If you're building super bespoke data products in data mesh, you're doing it wrong, because you're only answering very specific questions and you're not setting up the data to be capable of answering multiple questions.

But starting off in that bespoke one on one way with an idea, as to grow it into that data set that can answer multiple questions, seems to actually be a very, very good pattern that I'm seeing emerge because it's... "Okay, I am going to say I'm going to have that one to one relationship, 'cause I've got somebody who's saying, "I need your data, producer, I need your data", and the producer's going, "Okay, I can work with just this consumer. And I don't have to think about every potential use of all of

my data ever. I can make sure I satisfy this need and then start to build out how we would satisfy other needs and how we really wanna display how the business, our domain works and that and what's going on, that context, but that I don't have to say, "Okay, I have to do that from the very beginning." Because again, it's, "Okay, so tell me about your domain, right? So tell me about yourself, tell me about your domain", instead of that one-to-one. So I liked a lot of what you said there. I think it's very useful.

Dan Sullivan
Yeah. I think... And a lot of this I think we can learn from software engineering and Agile methodology, is this idea of just sitting down, get rapid feedback, be with people who are gonna be using the system and getting feedback, so it's not what you describe and which I have been guilty of, and especially early in my career when Waterfall was the thing, or Spiral was the thing, and we built stuff for a month and then show it to somebody. And that often ended in train wrecks. If it wasn't a terrible disaster, we were feeling lucky. And I feel like now we know better, collectively we know better, and we have these techniques which thanks to the Agile community and people who practice Agile, it's like there's collective knowledge about how to do this kind of thing, and I think we could definitely adapt that in data engineering, and I hope we all do.

And I really like your idea about this idea of building one thing and it's not... It's more of an instance of how you'd like to see it, but I'm gonna build just enough to get one piece done and kind of like, "I'm gonna put one thread in and then I'll come back and do another thread." But you have in mind what the weave is gonna look like while you're building it, and I think that's super important. It's like it's worth investing in building your architecture and understanding how you wanna work and setting up the mechanics of how you're gonna collaborate and document and make decisions about, "Alright, well I'm going to make this API or I'll structure the JSON in this way." And those come out, those sort of practices get built incrementally and evolutionarily, you learn as you go. But definitely delivering some value soon is better than trying to plan the entire structure and build it all at once.

Scott Hirleman
Yeah, I think that delivering value soon is the thing that just keeps coming up more and more. There's these grandiose ideas of what data mesh could be, but all the people that I see that are having success are having that practicality and Zhamak is very practical as well. There are people like, "Oh, these articles in the book and stuff. Okay, this is a lot of theory", and it's like, "Yeah, but she's been doing this with clients. She knows that you've gotta deliver that fast value. "

Dan Sullivan

Right. And the other thing I've seen is that people will start up, it's like, "Yeah, we're gonna follow principle, do data mesh, and then the clock is ticking and you start, "Well, forget it, forget it, forget it." And you end up just building something filled with tech debt because it also requires discipline to just stick with it. And it's a balance, and there's no right answer, it's like when do you take on tech debt? It's a personal decision in your own life, it's you as a team. And this is another reason why that kind of close iteration with them, either like the project owner or the person consuming the data is so important, 'cause you're gonna have to decide which is more important, that you're sticking to the principles and you're building out your data mesh platform, or you're gonna deliver something for someone who needs to make a deadline, and you're just gonna take on more tech debt?

Scott Hirleman
Well, in that kind of what you talked about with data tech debt and the enterprise data warehouse. We need to re-evaluate how... This is something that I've talked about with data consumers, is that data consumers are like, "Don't ever change anything that you're delivering," because they're so used to a change meaning breaking, right?

Dan Sullivan
Right, right.

Scott Hirleman
And with data mesh, we have to set it up so that those changes are versioning and that it's not...

Dan Sullivan
Yeah, exactly.

Scott Hirleman
And maybe going from one version to the next does break what you've got and you've gotta rebuild it, but we're gonna give you the extensibility and the ability to know when that's coming, and we're gonna give you the help to make it so that you can make that transition and we're not going to break things out from underneath you for no reason, which is kind of what's been happening with the application engineers, is that things just change 'cause they need to evolve their model and they don't know who's consuming what, why, how. And so they break stuff for people and they just didn't know, and so...

Dan Sullivan
Right, yup. And I think it's important for us as data engineers to think about the different kinds of breaking, and there's definitely the one we probably, all of us know

all too well this idea of the API changes. Something underlying changes in our code, made assumptions, or we had a contract and the contract's broken and now our stuff breaks. But we also have to understand those changes weren't done just on somebody's crazy idea and wanted to try something out. No, there was some business driver or organizational driver that required them to change something, and so we as consumers also need to understand that the stuff we depend on is also in flux, right? It's like gears, and the gears aren't static, and because we're consuming from a stream of data doesn't mean that that stream is going to stay the same. So we need to be adaptive in the way we consume because breaking from our perspective means I went to get the data the way I always do and it's not working. And then there's another kind of breaking, which is the business needs to answer a particular question, and they look at the recording there is, the data is not there. This is not working for me. The function, this report, whatever the function of this report, it's not working, and so now we need to fix it.

And so there's different kinds of breaking depending on where you are in the data, in the spectrum or the data pipeline, and I think we can't think of ourselves in this, "I'm the center of the universe, and everybody revolves around me." We need this more Copernican perspective. It's like we're all moving. And yeah, it's not breaking. I don't know how I feel about this, whether I wanna keep calling it breaking, 'cause I keep saying that, but it feels not right, 'cause it's not breaking. Things evolve, nothing stays the same. There's entropy, there's evolution, like the one thing that is constant in the universe we live in is that things are changing. So I think we also have to adapt our mentality as data engineers to accommodate that.

Scott Hirleman
Yeah, there's an episode coming out on Wednesday, which listeners will have heard it a couple of weeks back that I talked about this with consumer locking is kind of what I'm calling it, of that they'd say, "You must lock how this is served to me versus we have to enable the evolution." If we don't have evolution, if your application isn't evolving, your application is stagnating. If it's static, it's stagnant. And so if you're not having that evolve, that's trouble, but you don't want your data model necessarily to evolve exactly one-to-one with your application. That's what breaks things. So we need to create that ability for the application to evolve and the data model to also evolve as the business evolves, but that there isn't this tight coupling between those two.

And I haven't seen very good frameworks on how to do that. How to really... 'Cause people are still so used to just trying to share as it lives in the database for the application, and it's like, that's not at all useful. I'm trying to get... Working at DataStax and the Astra DB offering and stuff, I'm trying to get us to think about how can we make it so that we can offer something to the application engineers so they can

understand what's gonna change or what this might break if I make this change for the data model, and is that actually an okay thing where it's like, "No, that's an evolution, so I need to evolve my data model at the same time," or, "Oh, okay, if I do this, it's a refactor, but I could just do it this other way and it wouldn't actually break the way that the data model is consuming."

So then that would be a breakage 'cause it's just kind of on a whim of, oh, I'm gonna do this refactor and there's not really a real need for it to do it in this way, so that it would be breaking it versus I'm making an evolution and the way we do things just has to change. And that might mean that just the way that we consume the data to then share it changes, or it may be what we share changes. And so it's just kind of all wrapped up.

Dan Sullivan
Yeah, and exciting. That's part of the contracts when we may make contracts. It's like normally we talk about, alright, I'm gonna skew Columns A, B and C from the data, and it'll be in a JSON structure, and we assume it's fixed and we don't talk about, well, how is this schema gonna evolve and how will we adapt to it? And maybe we'll just agree, it's a JSON payload and we're gonna have maybe some metadata and whatever. But I think, yeah, if we talk about schema evolution as part of our contract, that's a step. But it's a hard problem, this is not trivial. I think that's good. It's gonna be one of the interesting areas over the next few years that we figure out, because it is... We're building standards, like electrical engineers and mechanical engineers, there's all kinds of standards that have been built up over the years, and so we're still immature as an engineering discipline, and so we're in those early phases. And I sometimes wonder, like the engineers who built the aqueducts, like in 2000 years ago in Europe, how did they do it? How did they come up with standards and things like that, and how do you promulgate to standards and get everybody to buy in?

Scott Hirleman
Yeah, it's all just kind of craziness and that you've gotta have standards that are extensible. There's certain things that you may wanna have rigid, but a lot of it needs to be extendable, extensible, where you can just say, "Hey, here's the basics that you have to adhere to, but if you wanna add additional, great. If you wanna augment this greatly, then that's a new standard and you have to own that new standard. But that's fine, if you wanna do things under a new standard." And yeah, we've had a few episodes on data contracts 'cause it's just... There is the schema side, but there's also the actual metadata and meaning side, the semantics as to, okay, you were measuring in inches and now you're measuring it in centimeters. So the meaning has completely changed, but.

Dan Sullivan

Yeah. No, right. And that's another layer of the problem. Even if we all agree on this, you and I agree on these things, there is this idea, it's like the old, we have two reports, column names are the same, and they have two different values and you're like, "Well, which is the source of truth?" And how do you talk about that? That's another long conversation.

Scott Hirleman
I did a SQL class and the data was just barely unclean, and if you went and you did joins in A to B to C to D, you got this number that was like $2445.15. And if you went from A to E to F to D, you got $2445.12. And so it was like, okay, so they're just slightly off, but which one's correct? And it's like, ehh, they kind of both, kind of aren't. But also, we're unfortunately not gonna have time to really go super deep into the literacy for the consumers, but I think that's another aspect for the consumers of directionally right versus fully correct, of like even though those types of things of, "Well, does it really matter?" If you're talking about accounting, yeah, it probably does matter which one of those is correct, but if you're talking about, "Okay, what were our sales yesterday and what does that look like compared to the week before?" "Okay, it's up 15%. So is it up 15.02% or 15.18%?"

Dan Sullivan
Right.

Scott Hirleman
Ehhh, it doesn't really matter that much.

Dan Sullivan
Yeah. Yep. Exactly. Right. We're not measuring the mass of an electron here, we don't need it out to 10 digits. It's like, "Pi is about three." Close enough, when I'm setting up some big circular thing in the backyard, that's close enough for what I'm doing. Yeah, knowing the context of the business question will help us understand when we're... How much precision, how accurate does it have to be? Yeah, I'd love to do that. I have to say, I have a course I'm about to put up on Udemy, on data literacy for non tech managers and it really, it talks about some of these things. That's another... It's just like, Yeah, there's so many side topics related to that, yeah, we don't have time to talk to you about it, but they are definitely interesting and will cause us pain in the future.

Scott Hirleman
Yeah, and I'll drop that link in the show notes if it's out by the time we publish the episode.

Dan Sullivan
Oh, great. Thanks.

Scott Hirleman
And so we are unfortunately at time because of hard stop. But where can people find you? What do you want people following up with you about? If people wanna continue this conversation or have additional incremental conversations, and how can people get in touch with you and what do you want them reaching out about?

Dan Sullivan
Oh, sure, yeah. Feel free to contact me, LinkedIn's a great way. I'm Dan Sullivan, PDX on LinkedIn.

Scott Hirleman
I'll drop the link.

Dan Sullivan
Okay, great. Also my email at 4 Mile is dan.sullivan@4mile.io. Feel free to email me, if that works better for you.

Scott Hirleman
Okay. And any specific topic that you want people reaching out to you about, anything that you think is...

Dan Sullivan
No, if you have any questions about data or cloud... I do a lot of work in Google as well, Google Cloud. So anything along those lines, just feel free. I'm always... People always have interesting issues they're dealing with and interesting challenges, so I'm always up for a good conversation about those.

Scott Hirleman
That's kind of part of the podcast, right?

Dan Sullivan
Yeah.

Scott Hirleman
It's just people sharing context and just having a useful conversation. Well, Dan, this has been super awesome. I really appreciate you taking the time, so thanks so much for that and thanks everyone for listening.

I'd again like to thank my guest today, Dan Sullivan, Principal Data Architect at 4 Mile Analytics. As discussed, you can find a link to Dan's LinkedIn as well as his email address in the show notes. Thank you so much for listening to this episode of Data

Mesh Radio. Hopefully, it was useful to you. If you'd like to connect with the show, you can find us on LinkedIn or Twitter, if you'd like to connect with me, you can do the same. If you have feedback, or especially if you'd like to be a guest, we've got some links in the show notes to tell you how to do that. I'd love to hear what questions people have and how I can be useful.