
Transcript provided as a free resource by 


**Data Mesh Radio Episode #41: Winning Over Application Developers: Agency not Autonomy**

Interview with Jessitron (AKA Jessica Kerr)

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

**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, self service 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**

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

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 learn about and hear about in this episode? I interviewed Jessica Kerr, more widely known as Jessitron, who is a principal developer advocate at Honeycomb.io, or as many affectionately refer to her, the “Empress of Software.” You're going to enjoy this one, I bet. I asked Jessitron to be on because she had a tweet I loved calling Data Mesh "conscious design for unexpected use," and I thought that was an amazing summation of it, especially when you think about producer aligned or primary data products. And the second part was the three most common questions I get about Data Mesh are one, how to do data discovery, two, how to do domain driven design for data and three, how to get application developers to freaking give us their data.

Obviously, I don't love the framing on the third question, but it's a pain point, and she knows application developers incredibly well, having been one for 20 plus years.

Jessitron started off by discussing one of the biggest issues with application development today. Despite the tooling and process advancements over the last 20 years, it's all somehow only made application development that much harder. You have to understand so many more aspects of the application development than people had to 10, 15 years ago. So the starting advice is, don't just add more to application developers' plates. It probably won't go well.

Her biggest point, I think, was giving application developers agency in how they share their data, I think that's the key. Autonomy, as she puts it, is kind of BS, it's just passing the responsibility over without the help. Basically, here, you do this, I don't want to take care of it. Application developers want guidance and direction, especially to the target outcome, but they want to make the choices on how to achieve the target outcome while being given the resources to do so. In general, they want others to have influence but not control. The information and capability to do their job is key, work with them to give them both in spades. We haven't done this historically around data, so figuring out how to do that is not necessarily an easy task, but you do have to think about just giving them the responsibility to do this without the resources, the information, the capability to do it, it's just not going to work very well.

For driving buyin, don't start with the ask, but start with the why, let them know why their data is available, and be specific, as in, your data is used this way by these teams to power X and Y and Z. The conversation should be about their potential impact, not just the negative of, you changed this and it broke X or Y or Z, but the aspirational side of, here's what we could build if we had this data, or here's how your data could really improve the organization.

You want them to start thinking about how you can work together to enable them to share their data in a high context, highly meaningful way. Data Mesh is going to be a big cultural shift for application developers. You need to not just put something high priority on the backlog, you need to give them the space, meaning that they have enough points or whatever on their backlog to actually do this, but to really understand and learn how to share their data well. You have to focus on teaching them how, possibly via an internal hack-a-thon to start building that muscle, or even a cross functional pair programming like initiative to show other ways, to show each other your ways of working and sharing knowledge.

So really, again, what I'm saying here is you need to teach them what they have to learn and give them the time to actually learn. You can't just say, you need to start doing this and here are some resources, but not change up the way that the team is working or change up the KPIs or whatever for the team. They need the time and space to learn something new, otherwise it's just not going to happen. Also, show

them the impact they're having along the way as they get going, that will motivate them to do even more. Be very conscious of language. The interview with Audun and Gøran at NAV talked about this a lot. Application developers and data people don't speak the same language, work with them to put what you're trying to do into their language. In general, Jessitron, AKA the “Empress of Software,” has some really great insights in this fun conversation, I think you'll really enjoy.

Quick editor's note that all of the previous information should be attributed to Jessitron and the conversation that we had, and that it's not just my conclusions from it, but a lot of it, or pretty much all of it is from directly the conversation with Jessitron and her points that she made. With that bottom line up front done, let's go ahead and jump into this interview.

Okay, super, super excited about this episode, we have the “Empress of Software,” I'm with Jessitron here. Jessica Kerr is also the principal developer advocate at honeycomb.io, but I'll be referring to her as Jessitron, which is her online handle.

**Jessica Kerr**
It's my unique ID.

**Scott Hirleman**
Yes. So it's kind of...

**Jessica Kerr**
We have five Jessicas at Honeycomb now, we have more Jessicas than Mikes.

**Scott Hirleman**
Wow.

**Jessica Kerr**
Right, for a software company.

**Scott Hirleman**
I went to one company, I had never worked closely with another guy named Scott. And then we had an office of eight people in San Jose and three of us were named Scott.

This one's a little weird, like that's not the best, it's not.

**Jessica Kerr**
Statistics, man. It's going to happen sometime.

**Scott Hirleman**
Yeah. Oh, yeah. The whole, like, multiple people in a room having the same birthdays...

**Jessica Kerr**
Yeah.

**Scott Hirleman**
So, but on the topic that we're going to be talking about today, as Jessitron is a principal developer advocate, she understands the developer mindset. And so very excited to kind of... One of the key things that is blocking a lot of people is trying to essentially extract the data out of the developers, which I think from a lot of the other episodes that we've had would kind of be a throughline of don't do this, of try to extract instead of try to enable them so that they can share their data, and make it a part of their workflow, make it part of their role, but also give them the resources to actually do that, whether that's additional heads, or tooling, or yes to both. So if you don't mind, maybe we should start with a little bit of background and introduction to yourself, and then we can kind of go through this very, very large topic, where we'll just kind of bounce around and just, hopefully, share some jokes as well.

**Jessica Kerr**
Okay, okay. Yeah, I'm Jessitron, I've been a developer for 20 years, been speaking at conferences for the last eight or so. And I've become more and more fascinated by software development over the course of my career, because you would think after 20 years, something would get easier, but it has not. It's only gotten harder and more interesting. And some of that is, a good bit of it is really the industry. We keep taking something that's hard, or at least hard in terms of effort and time and making it easy. Starting with compilers, instead of writing machine instructions, you can write something that more expresses what you need. And then we keep layering on top of that with higher level languages. And then we add infrastructure in it. And we've got tools like Kubernetes, that abstract over the hardware. And this, this just gives us different problems, right?

**Scott Hirleman**
Very much so.

**Jessica Kerr**
And so first, we had the problem of not enough data, and then we have the problem of way too much data. And now we're like, can we get the right data? And that's a much harder problem than either.

**Scott Hirleman**

And at the right time in the right format, in front of the right people, in front of the, structured in the... Yeah, yeah.

**Jessica Kerr**
Yeah, so we keep solving our formerly hard problems and finding the next one, which is really exciting.

**Scott Hirleman**
It's one of those things where when people are like, "Oh, the machines are going to take over," you know, or it's like...

**Jessica Kerr**
Yeah, software developers laugh.

**Scott Hirleman**
Yeah.

**Jessica Kerr**
Because there's so much manual work that goes into... Manual, like, different every time kind of work that goes into keeping software running.

**Scott Hirleman**
Yeah, and just, and being able to understand that complexity and that it's like, okay, when is it different this time, and when isn't it different this time and can we create for use and...

**Jessica Kerr**
Right. So like in software, I wonder if the word manual will go from meaning repetitive, meaning labor, to meaning custom, different every time, bespoke? Because that's something only a human can do.

**Scott Hirleman**
Yeah, that's an interesting insight. I have to think about that one.

**Jessica Kerr**
It could happen.

**Scott Hirleman**
Yeah. So one of the big things that... Again, as I mentioned, people keep running up across is when they're looking to implement Data Mesh, the conversation between of... Coming from the data side and going to the application developer, the domain side and saying, "Hey, we need you to participate. We need you to give your data in

this way." It's kind of framed in that way and it hasn't been going that well for a lot of folks. So how do you start to think about if someone were to come and speak with you about, hey, we're looking to use this data, what do you think is an initial good way to start that conversation? Not even to get everybody on the same page, but is it about sharing how it's used, or is it about, hey, we're going to support you in this first and foremost, and that we're not just tasking you, we're not throwing things on your backlog?

**Jessica Kerr**
I would start with, you have something valuable. The developer who writes the application, the application owns the data that you want to be able to use in other circumstances. Is that accurate?

**Scott Hirleman**
Exactly.

**Jessica Kerr**
So the developer has at least access or through the software that has some control over where this data flows, and they might not realize how valuable that is to the organization and new ways that they can add value to the business by providing the data in a meaningful way. And the meaningful is, to me, this is the essence of Data Mesh, is providing data that's meaningful as a product, that conveys what the data really is and how to use it well, as opposed to, I used the phrase earlier, extract the data out of the developers. Which they will do, if there's a Jira ticket for it, and they're told their job is closing Jira tickets, but then you're going to have to tell them exactly what you want. And what you really want is understanding, and you want the data and the way it's offered to you to express that understanding, right?

**Scott Hirleman**
Yeah, it's having that conversation back and forth of, "Hey, here's what I think you have. Here's where we're trying to get. How do we bridge that gap?" And to pop-up to a higher level, I forgot to mention why I asked Jessitron to be on. She put out a tweet that I think expressed the way of  data as a product, thinking better than any way I've seen it, which was just... I think it was... What was it? Conscientious design for unexpected use or something like that. I think that was what it was. Was that it was...

**Jessica Kerr**
Plausible.

**Scott Hirleman**
Or conscious design or something like that. That you're thinking through how you would share your data and to share that meaning. So when you're talking with

developers, as you mentioned, developers are now having to do more and more and more things...

**Jessica Kerr**
Yeah, that's true.

**Scott Hirleman**
So if you start to throw... You now have to serve out your data to people as well on their plate, like if someone were to come to you and say, "I need you to share your data," explaining why it can be useful, but what tooling do you think is there or do you... Is that not even the right way? 'Cause is it that it's not tooling, it's not work... Should it be workflows? Should it be just platform, or should it be blueprints, or how do you think about working with the domains to make this easy, an easy transition, that it's not... You need to share this thing in the most complicated way, right up front.

**Jessica Kerr**
Okay, yeah, we want to give them the abilities to solve the problem that we set for them, but we want to set for them an additional problem. In addition to keeping this data secure and consistent for the use of the application, we also want to offer it up in a way that's useful to the business in ways that we can't predict. So not just exactly one way, probably. It's not meet these specific requirements, it's broader than that. It's meet these needs, which is way more difficult, but also way more interesting. So we want more than a piece of action from the developers. We want some mindshare. We want them to be thinking about an additional product, an internal data product. And for instance, when the meaning of the data changes later in the course of development work, we've added a new feature, and now people can restore deleted records, so we're not going to really delete them, their entries, say, we're going to mark them as deleted. Well, that's a piece of data that you'll want if you're retrieving entries, that I just made up.

And so the developer who's implementing that new don't really delete, just mark deleted feature, you want them to think about the data that they're offering to the business internally and recognize that this is going to have an impact there. So it's not a one time action that you want. For this data to continue to be meaningful, it needs to belong to the team. And you've used the word the domain side a couple of times, and I like the focus on the domain here, because what I want an application developer to have, and as an application, what I want is that understanding of the domain. Which means I know what the data means. Which is a culture shift for a lot of teams. Because if you've got teams that are used to being told what to do, this field is called this, pass it from here to there. This is very common, because people are just asked to close Jira tickets and not to understand and they're not given the

opportunity to understand the data.

But what I want as a developer is to know what I'm doing. This helps with security, you can actually validate your data if you understand what it's for and what the meaning of it is, and it's totally necessary to be able to publish a data product, to put it out there into the Data Mesh in a meaningful way. And that's what I want to ask of developers. Now, whether their manager asks that of them or encourages it or gives them the ability, like the contact with business people, the feedback of what's really happening in production, for instance, if developers don't have those abilities, it's not their fault, and you're just going to have to tell them what to output and get it in their backlog and negotiate with whoever assigns the backlog. That's one way a lot of companies work. Like a lot.

If you do have a team that really owns the application and the domain and the bounded context of what they're building, then you've got to give them the opportunity to create more value. And if you can show them what providing this data will do for the business and give them visibility into that, then you can get buy-in and they can say, "Oh, you asked for this, but you probably also want to know whether it's been deleted or maybe you want us to exclude those, let me know." Yeah, 'cause you're not asking for the bytes, you're asking for something more, something that a developer of an application hopefully gets the chance to really understand what they're building and share that with you.

**Scott Hirleman**
And how it might be used in that, of that back and forth conversation, I think...

**Jessica Kerr**
Yeah, what customers really do with it. Maybe there's this description field that if developers have visibility in the production, they might notice that the length of the description is either zero or it's 200 and some, and you can get an idea of what customers are really doing. An app developer who has that kind of that visibility and that feedback loop can provide you a lot more insight on what it is you're asking for.

**Scott Hirleman**
Yeah, I think this is a common throughline of... I had an interview with a couple of people from Thoughtworks, Danilo Sato and Andrew Harmel-Law on domain driven design for data, and Danilo at the end just said, “Hey, just get people to talk to each other.”

**Jessica Kerr**
'Cause talking to each other is the important part, and talking to each other, being conscious of language. Ooh, ooh, I heard something really interesting yesterday, a

friend of mine who's doing some software work for a bank, said... And then they want the domain driven design so that an external account means the same thing everywhere, across the business. And I'm like, wait, wait, wait, wait, you know... Do you know that you just expressed the opposite of domain driven design? And they were like, Well, that's what they call it. Oh, God. Well, you know, like Agile, a word has reached its peak, like DevOps, when people use it to mean the opposite of what it originally meant.

**Scott Hirleman**
That's happening in certain places with Data Mesh, like people going, Oh, yeah, we're doing a Data Mesh, so we're creating a centralized data warehouse, it's like... What? No, that's... Yeah, it's...

**Jessica Kerr**
No, that's a data net and you're caught in it. Yeah, that part of talking to each other, when we come from different domains, which is just different modules in the software usually, and we talk to each other about a piece of data, the assumption needs to be we're not talking about the same thing when we use the same name, until we establish what each of us means by customer ID entry, deleted.

**Scott Hirleman**
I like that. I think that's a helpful initial tip as well is to go in when you're kind of getting people in the same room to go, Hey, we don't talk the same language. We need to. We need to be on the same page, so we're going to work together to get there, and it's not that you application developer have to speak in our language, but also we're not going to necessarily understand your language as is. So like let's back up and...

**Jessica Kerr**
Which is being explicit, we can use both languages when we need to, but names based stuff. This is the accounts' team customer ID, and this is the diary team customer ID. I don't know.

**Scott Hirleman**
Yeah, I've talked about... Some places are having five underscore names where it's, this word this word this word this word this word and people are like that, ohh, that's so awful...Well, but it's helpful, right?

**Jessica Kerr**
Yeah, it's real.

**Scott Hirleman**

Yeah, and what was it like... I think it was like one of the old SQL servers. You could only have 64 characters in the name of a column or something like that, anything else.

**Jessica Kerr**
Oh, back in the day.

**Scott Hirleman**
Yeah, we're not in that world anymore.

**Jessica Kerr**
And we're not typing this stuff into our command line SQL clients either. Well, okay, not often, usually we're using tools with auto complete. And if we are typing, there's cut and paste.

**Scott Hirleman**
Yeah.

**Jessica Kerr**
That does work in my terminal. So I was talking to Eric Evans the other day, speaking of DDD, and we came to the conclusion that there's a particular value in acronyms, and acronyms where you started with a term like external account customer ID, and then you use that. You've got a real term that means a very specific thing within a bounded context, within a piece of software. It's not generic, this is referring to a particular abstraction in the software, not to a physical thing in the outside world. And that gets shortened into EXCID, and that gets shortened into excid, or whatever I just made that up. And then an excid is a thing within your company, and on one hand it's like, oh, acronyms, why can't we just call it customer ID? Because customer ID is not specific, customer ID is like it's a common noun, it's a lowercase thing that people think of as... Or at least customer definitely is a lower case noun, it's something general.

Whereas, an EXCID is something very specific, it's a proper noun, like Scott and well, actually, in this case, not like Scott or Jessica, which is too inspecific, it's like Scott Hirleman or Jessitron, it refers to a specific abstraction within a specific context, and that's better. Yes, it takes people a long time to learn them when they come in, but at least it's signaling, you don't know what we're talking about with all the TLAs.

**Scott Hirleman**
And put documentation in, so you have empathy for those people who have to learn it, right, that you have that...

**Jessica Kerr**
Yeah, but do not use them, because you really are talking about that specific thing and not about some generic customer that someone might think they understand what they mean, but they're from a different team. So no, they're talking about yes, okay, there's a connection between the customer and the external account team and the customer in the... What was I using? Diaries team. There's a connection maybe in the outside world to the same person, but they're not going to have the same data. And we need to think about them differently. So...

**Scott Hirleman**
I worked at a place that used the whole wrapping around a customer, was a customer container, and then we also used Kubernetes. And so just talking about containers, everyone just was like...

**Jessica Kerr**
Oh, no.

**Scott Hirleman**
Which container do you mean? Like which type of container? And it just made everything very, very difficult. So I liked even that you said cut and paste or auto finish. I know some places that their security doesn't allow for cut and paste, and they don't have that finish, that auto finish. And so it's like, "Don't do that." These are having empathy for folks as to how they're actually going to use this.

**Jessica Kerr**
Yeah, there's a difference between preventing people from doing things wrong and helping them do it right. And it's like those password fields that you can't paste into. Thanks, my password is now garbage. You broke it.

**Scott Hirleman**
Yeah, I used something rather insecure because I could remember.

**Jessica Kerr**
Yeah. And we were talking earlier about, the point is that we want the data that is offered by the domain teams to be useful. We want it to fit with other data in the company, not be identical to, but have clear relationships to. Can you connect a customer ID over here to a customer ID over here? Yes, if they're this referring to the same person, they will have the same customer ID maybe. Probably not, they each have their different ones, and there's a different field in there that is referring to some other generalized ID. But can you correlate these? And what is the relationship and how do you navigate that? And maybe there's always going to be customers in each system that aren't in the other. So we want the meaning encased in that data to be

accessible to people, to be friendly to humans. And sometimes that means really long names.

**Scott Hirleman**
Do you think when we're thinking about making that data accessible, in a lot of ways, sometimes developers are asked to get a little too... I don't know if "cute" is the right word, that kind of sometimes feels derogatory. But around trying to create too much normalization or abstractions or...

**Jessica Kerr**
Yes, I think we love doing that.

**Scott Hirleman**
It's a fun, interesting challenge, right?

**Jessica Kerr**
Oh, yeah, normalization of database schemas is so entertaining.

**Scott Hirleman**
But stitching it back together, when you actually want to share what actually happened. Like, how would you go about actually working with a team to do that? I talked with somebody this week that was... It's a data platform team. And what they've said is, you absolutely should have somebody, at least one person on your data platform team, that is focused on actual developer application platform, because if you're trying to make it self serve for your developer...

**Jessica Kerr**
Okay, this is the tooling that makes it easy.

**Scott Hirleman**
Right.

**Jessica Kerr**
Yeah, yeah, because you are going to want it to be compatible. You don't want, like, Well, I offered you some data using SOAP, and then the other team is like, Here you go, you can call this with CORBA, to pick two that you definitely don't want. Yeah, so making it easy for them to do the right thing, and then leaving them with the much more important problem of expressing the data meaningfully.

**Scott Hirleman**
Yeah, and understanding their workflows. If it's data people trying to expose how to do data to application developers...

**Jessica Kerr**
Oh, my gosh, talk about different languages.

**Scott Hirleman**
Yeah. What, you don't love YAML?

**Jessica Kerr**
Oh, God.

**Scott Hirleman**
You don't have a YAML tattoo or something, where it's like, I love YAML? .

**Jessica Kerr**
I prefer QR codes. At least you know they're inscrutable and you need tools to read them.

**Scott Hirleman**
Yeah. The number of times, dealing with Apache Cassandra and stuff, that's... I think they cut it down, but at one point there were like 400 YAML choices and it's like, "Oh, these like 265 you should never touch."

**Jessica Kerr**
Oh, no.

**Scott Hirleman**
Why are they there? They were in there. And so it fails if it doesn't have them, but...

**Jessica Kerr**
Oh, my God. Wow. Yeah, yeah, that's for computers to mess with. Right, so I agree with you on a data platform to make a nice happy path for the mechanics of exposing that data. And then you need to get the developers on board with it, and they need ownership of it, right, so that they'll keep it current. I think my last real blog post was about, if you want people's help, don't just tell them what to do, get them involved in the decision, and then they'll actively contribute and help you. But that means you may not get exactly what you planned on them giving you.

**Scott Hirleman**
Yeah.

**Jessica Kerr**
You may get something better. You may get something that's equivalent, but they

feel buy-in and that's worthwhile.

**Scott Hirleman**
Yeah, I think that... Or also that you, a lot of people are trying to start with their most pernicious, their most difficult problem first, right, of... And that's a very, very bad antipattern for Data Mesh, because you have to build the muscle, right. You have to get people in the habit. So like, that's where I'm trying to figure out how, like what workflows you might model after to say, okay, this can help people, this can help application developers to learn. And it's different for every organization, every person, but...

**Jessica Kerr**
That's a very good point, okay, 'cause sometimes you take on an end, a goal not for what it gives you. Maybe you already have this information in another place, but if you ask the developers, what would be easy for you to provide? And they're like, well, we could tell you how many diary entries each person is making every day, and that would just like fit with the way their data works now. Great. Make a little dashboard for that, because even if that activity itself would be low priority, it has the effect of teaching them how to put out their data, of integrating that as part of what they produce, part of what they garden and maintain and provide, the capabilities they provide to the business and then expand.

**Scott Hirleman**
Yeah, exactly. And I think there's this kind of perception from the data side that application engineers have no idea about data modeling, which I don't think is true at all.

**Jessica Kerr**
It's true of some and not others. I mean, back when I started, design basically meant data modeling.

**Scott Hirleman**
Yeah. And I think there's aspects of it in everything they do, but is there like something that you're seeing now that would, you think could provide a gentle path, right? You know, you talk about the happy path, which is developers, at least most people that talk about developers say they want to be able to achieve things, right. So, and learning new things is not a bad thing for most developers. Most developers are really excited to do that, that's part of the job, and that it's interesting if it helps them tackle new challenges.

**Jessica Kerr**
If not ruined by externalities, yeah.

**Scott Hirleman**
So like, is there something that you could see as being a kind of a happy path to learning about how to do data modeling? Is there a framework or is there something that you've seen out there?

**Jessica Kerr**
Some people do internal hack-a-thons for this.

**Scott Hirleman**
Okay.

**Jessica Kerr**
You know, like the internal hack-a-thon of, Hey, here's this tool, which team can use it to produce some cute little dashboard that other people in the company find entertaining or useful or cute. That could be a start.

**Scott Hirleman**
Yeah, okay, I just didn't know if there was, like if there was a framework that felt like it was, it kind of forced you to do a little bit more data modeling than others, that it might not be as popular, but that...

**Jessica Kerr**
Yes. Some of them totally do, Elm, for instance, really makes me think about the modeling of state, whereas you need to do all of that in React, but React will let you get away with being mushy about it for longer, but React is way more strict about it than, say, vanilla JavaScript, where it just turns into a pile of scare. So writing code in Elm really helps me write React, for instance.

**Scott Hirleman**
Interesting.

**Jessica Kerr**
As far as data modeling, man, I don't know. I learned it by writing create table statements, and so should they. Not like literally, but it is surprising how we've abstracted away SQL in a lot of cases and how incredibly useful it is to be intimately familiar with SQL and explain, analyze and...

**Scott Hirleman**
This has been an issue for me that I see a lot of people on the data side again are saying, oh, we should just expose DBT to the application developers. And it's like, they may not...

**Jessica Kerr**
What's DBT?

**Scott Hirleman**
DBT is like a thing for transformations in, and you just write SQL and it kind of creates your transformation pipelines and stuff. So it's very, very, very popular on the data side. But like, yes, as somebody who's intimately familiar with application engineers saying, what is DBT, that should be something that people realize that is not...

**Jessica Kerr**
So you speak different languages? The like worlds that we live in are entirely different, because an application engineer's world is built on React maybe, and then JavaScript, and then the browser, or a node or Java that the languages the web frameworks, Spring, if you're in Java, Express, if you're in node, these are our world that we build things in, these are the, like the physics that we use. And maybe we get down into SQL sometimes 'cause we store things in a relational database, or not, maybe it's Dynamo, who knows.

But then, yeah, on the data side you have a completely different tool set. On the application side, like at Honeycomb, our purpose is to give application engineers visibility into what's going on in production, and they can set that up for themselves with tracing and then look at our graphs, y'all have your own ways of looking at at graphs that we don't even use. So, man, like, it would be fascinating to pair on stuff and just see each other's world, the IDE versus DBT. The tools the data engineers use that I don't.

**Scott Hirleman**
I kinda like that idea of pair programming completely across worlds, where like, you can have a shadow program of, Hey, I'm going to walk people through this, and it might be one data engineer with like five application engineers on the same one, so it's not one-to-one, just because there tends to be far more application developers and engineers than the data engineering folks, but like, Here is how I would take this and express this, here's how I do this, here's the manual bits that suck, and so we want to make sure we don't expose these manual bits that suck to you, but we want to expose the power of how to accomplish the goal. And then they can start to show, Okay, and here's how we're thinking about this workflow" like give us that live feedback of "Yay" or "Boo" and just the... Okay, you exposed the SQL, and it's five thumbs down instead of like working in my own language, my own space.

**Jessica Kerr**
Yeah, yeah, so probably the best way to establish these shared goals and how are we

going to talk about this stuff together, and what do I really mean when I see this, is to sit down and do some active work together, pair program or ensemble programming with more people. And can you accomplish a thing together and let each... What do I want to say? I want to say side, but I don't want to make it about sides. Let each denizen of different tool worlds show the others what their life is like.

**Scott Hirleman**
And those aspects and just like that understanding build that, build that context rather than put it on somebody's backlog, right, like if you just put it on backlog, you don't have that context that comes with.

**Jessica Kerr**
Yeah, you've got to sit together and experience each other's little universes. Because we do work in different worlds, because we run different programs on our computers, and Lord knows, the screen is our world these days.

**Scott Hirleman**
And I talked to somebody this week, the folks at NAV, and they said what they're starting with, I'd like to get just your feedback and not, if you hate it, that's totally fine. If you love it, that's great too. But, what they're asking is to say, Hey, here's what we're trying to accomplish here, can you please go out and do this, and then what we're going to do is see how you do this and then automate your work. We're not even starting with, Here's the platform to do this, it's like, Hey, we're going to start with something that's more of a lightweight task for this, but let's automate every level of complexity that we go down.

So it might be that you're just creating a table or a view, a very lightweight table or a view of data that you're going to be pushing out, that you're not doing anything with heavy transforms, you're not doing anything kind of crazy. And then they're watching how the application engineers are doing it and then saying, Hey, we're going to create the platform to automate this, or We're going to watch your workflow the way you think about this, and then we're going to build the tools to make it so that you can just do that work in the way that adds the value and not adds the toil, but that it's part of your workflow, that it matches your workflow, instead of exposing data engineering tools to developers.

**Jessica Kerr**
The activity of doing that itself will have a lot of value for the people engaged in the activity. The automations themselves you might consider throwing away. You will gain a lot of insight from that activity. I haven't talked about those principles of collaborative automation, but there are dangers to if you automate the simple stuff, then when the hard stuff comes up, people get even more stuck by it.

**Scott Hirleman**
I hadn't thought about it that way, that's...

**Jessica Kerr**
On the other hand, the activity of automating something drastically deepens your understanding of it.

**Scott Hirleman**
Yeah, well, I think that they're trying to create... I don't know if it's necessarily fully automation versus it's the platform to enable them to do it, right, and so building out that platform to help the application engineers to do what they need to do, I don't know that it's necessarily automation automation versus it's drastically reducing the amount of toil, or should we be putting...

**Jessica Kerr**
Right, you do want to automate away toil and stuff. Be sure you don't do this with just one team, work with several of them, because they're all going to be different, and leave plenty of escape hatches. You can make people a paved path of, "This is how another team did it", and "these are what we have that makes it easy", that I've seen in some companies where the central team makes a library and you should darn well use their library for this, because that's how we do it here, and then that library doesn't do everything that you need and maybe you just sit there and don't accomplish what you wanted, more likely you work around it, but it's incredibly painful. Automation can hold you back in that way too.

**Scott Hirleman**
Yeah.

**Jessica Kerr**
I'm a huge fan of automation, also of developers automating their own work, but developers automating the work of other developers is just way harder than it seems.

**Scott Hirleman**
So think about your ROI on doing this work and...

**Jessica Kerr**
Yeah, but also think about the side effects, what does this teach you, what version of you exists because you did this work, and that might be one who can design libraries that are not frameworks, that are useful, or sometimes it's just documentation. In the end, you wind up, this is the example that you can cut and paste from, and that winds up being your reduction of toil, here's a template kind of thing. It's not in

YAML. So sometimes there's stuff like that that's low tech, but high knowledge, and sometimes you learn about that by trying to automate stuff.

**Scott Hirleman**
I like that low tech, but high knowledge. I think that that knowledge share, you kinda talked about it as to... People sharing their context. How is this used? How is this... It's just the more that you can put knowledge in front of people so then they can make decisions instead of putting tickets in front of them, putting, "I want you to do this," versus, "Here's where we're trying to go. Can you please help me get there." That let's go together type of attitude.

**Jessica Kerr**
Yeah. And to be fair, developers need space for that, and they don't always have it.

**Scott Hirleman**
Yeah, I talk about not putting additional responsibilities without putting additional resources. That's a... not nice move. I typically use a different word when I'm not on the podcast, but...

**Jessica Kerr**
Developers might be totally cool with learning things, but also that's not their job as stated. That's not in their OKRs or whatever. Or they are contractors. They're not supposed to learn things on the job, or specifically, they're not supposed to be trained. There's all kinds of weird incentives that happen, and I don't want people to feel bad if they're not sitting there, "Oh, my God, I can't wait to learn something today," in their day-to-day work.

**Scott Hirleman**
Well, and that's another aspect of Data Mesh, I think is, it has to have enough buy-in at the manager level to actually put this on people's plates. That they're able to allocate the time to it, and it's not just they feel bad 'cause they're not helping out their fellow employees, but they don't have the capability to. And the same thing, I talk about this with tooling too... Application engineers have to change their application. They have to evolve their schema. If they don't know what that's going to break downstream, they can't have the empathy for it, because otherwise they're stuck. They're in analysis paralysis, 'cause they say, "I don't know what this is going to break. I have no way to test what this is going to break. So I just kind of have to break it."

**Jessica Kerr**
Yeah, or do nothing. Right, yeah, and that's another thing that's why I care about observability is because that's another way that you might be able to figure out

what's going to happen and who your data is connected to.

**Scott Hirleman**
Have you seen even in a testing side of somebody's doing their development or on the, I guess, the development platform instead of the prod version or whatever? That there is even, "Okay, this is going to actually create this from a data perspective, and there's any tests there." Have you seen anybody doing that.

**Jessica Kerr**
Like integration testing?

**Scott Hirleman**
Sort of, but before you deploy, right?

**Jessica Kerr**
Oh, before you deploy. Generally, I've seen, before deploy, if you integrate with other things, it's in a dev environment that everyone else is also integrating with other things, and so everything is chaos and nothing works, and if something breaks you're like, "Ah, it's probably just dev. It's always down."

**Scott Hirleman**
Yeah. That's kind of what I thought. I don't know how we can get there. I don't know if we have good tooling to get there.

**Jessica Kerr**
IThere are some places where you can run...So the alternative to that is test and prod, one way or another. Either you're connecting to prod or some places have a QA environment that they think is sufficiently realistic, it's not. But the thing is, if you can make it safe to test and prod, if there's a mode or feature flags that none of this data gets saved, it just flows and then it gets dropped right before the database, for instance. I have heard of, at least, places that do that. A Stripe, there... This isn't internal, it's external. The Stripe test API runs through Stripe production, but with that little test attribute so that it doesn't actually go to any payment processors at the other end.

**Scott Hirleman**
Yeah, 'cause I'm thinking of that integration testing on the data side as well. How could you actually do... How can a developer know, if I make this change it's going to break things, unless there's frameworks to make that easy and that they don't have to manually test everything?

**Jessica Kerr**

Right, unless they're working with real data and they can run it either excluding test records in real production, or not excluding test records and then look at them. If you have feature flags that are huge for this, also, you have to be able to turn stuff on for a specific set of users for you, but not the public, kind of stuff. So yeah, no, I've never seen a realistic dev environment where people can really test things and find out whether they're going to break. You can find out whether some things break, you can find some, and some of those will actually matter and some of them won't. But usually, you'll spend a lot of your time trying to get things to work at all so that you can hope that it might work in production. Yeah.

**Scott Hirleman**
So I have two questions that I'd like to wrap up on. So if you only want to do one, great, but choose the order. But one, what do you think is the thing that if you give to developers, is it autonomy? Is it help? Is it tooling, whatever?

**Jessica Kerr**
It's agency. Agency includes both a goal that you can reach and the abilities to reach that goal. So you let them choose, help them. Give me direction, give me information, but let me choose my direction and then give me what I need in order to be able to move. So there is autonomy in that, to some degree. It's not aloneness, though, it's input, it's influence not control, people don't have control over their own lives, but we need to influence them.

So I'm going to decide on my priorities for the month with my manager. I'm not going to hear my priorities and I'm not going to just make them up, [chuckle] we're going to collaborate on this. Influence, not control. Yes, I like the term agency a lot better than autonomy. Autonomy is, you figure it out. Autonomous teams are, we can't get our stuff together, you figure it out, and we're going to be generous and blame the team instead of the individual, that's our big... Okay, that is progress, but it's still just dumping the problem on a smaller unit of the organization, when really our organizational design needs to give people the information they need to choose a good direction and the capabilities that they need to move in it.

**Scott Hirleman**
I really like that. I think that's very, very apt and pithy. And then the second one was about... So there's a lot of talk within Data Mesh about maybe data engineers kinda go away, there's the data platform developers, but are those necessarily data engineers or are they data engineers and application engineers, or are they just software platform engineers and that then the... Because the application developers are now doing data that they all just become software engineers, and everybody is just a software engineer instead of this specialization. And I think you're, from... If I can read facial expressions, you're as skeptical of this concept as I am.

**Jessica Kerr**
Oh, well, there's the first part, there's titles and I don't care. Whatever makes people happy and whatever gives them the influence that they need. A title is part of that ability to do your job sometimes. As far as... I do think blurring the lines is really good. If people cross over from one area of relative specialization to the other is excellent, but it's the same thing with a full stack developer, what is a full stack? And do you really expect people to be an expert in React and MySQL performance? Because you better pay them a lot if that's what you're looking for, and everything in between.

**Scott Hirleman**
Unless you, again, provide them the capability, but then again, you're just abstracting, abstracting, abstracting away. Then they don't really know exactly what's going on or what they're doing. So when they do hit those hard problems, they don't know... I think that's what you're saying here.

**Jessica Kerr**
Right, right, we can't know everything. We can't know everything. There is a limit. Cesar Hidalgo in his book *Why Information Grows* calls it a “person bite,” that the maximum amount of information, of knowledge, you can cram into one person's head, doesn't matter how much it is or whether yours is bigger. What matters is that it's finite, and that's why we have teams, which adds overhead, but increases the maximum amount of knowledge we can have on the team. And we talked earlier about opportunities to learn, what gaps does the team have. You can hire into some of those, but also, can you choose to take on problems that will develop those skills?

**Scott Hirleman**
Yeah, I agree with that, and I also, I worry about the everyone is a generalist approach, because I also worry about that for junior engineers. I think that having to come in as a generalist for a junior is much harder than having a specialization and so...

**Jessica Kerr**
Yeah, you can only get good at so many things at a time, and I want to get good at the domain, that's unique to every team.

**Scott Hirleman**
And to me, it's also especially worrying around DEI and stuff like that, where it's like there's been a lot of pressure for underrepresented folks and they kinda a lot of times get shoved out from the industry or kind of... So we need to be able to do that to evolve as an industry of software, and that, it kind of frustrates me that we're not thinking about how we actually set ourselves up to be a more equitable society and

industry and all of that stuff, and if we're just trying to make everybody generalist, it's like...

**Jessica Kerr**
You know what, I want to be a software engineer who can speak data engineer, or a data engineer who can speak software engineer, and if we work together enough to know each other's language and be able to have a conversation that's accurate, that's the level of generality that we need.

**Scott Hirleman**
I think you're spot on on that. I'm glad that we're on the same page, 'cause that one's been a little bit worrying to me of how many people... And a lot of times it is the people who have learned so many things because they've been in so many different roles, and so they are able to be that generalist and their “person bite” is much larger than my “person bite,” and so it's like, you may be able to do this, but...

**Jessica Kerr**
Yeah, but what really matters is being able to talk to people who can do that, and my grandmother used to say, know enough about a topic to have a conversation with people who are interested in it, and then every time you do have that conversation, you learn a little more, it's fun.

**Scott Hirleman**
Well, this has been so phenomenal, I think this will help a lot of people on figuring out how to interface with application developers and engineers. I think a lot of it is going in with the, Hey, yes, we're going to ask you to do additional things, but we're going to provide you with the information and the capability to do this, and that we're... And that capability is...

**Jessica Kerr**
And we're going to work with you.

**Scott Hirleman**
Yeah.

**Jessica Kerr**
Some of that we're going to provide you the capability is, I'm on your team for the next two weeks while we work on this.

**Scott Hirleman**
And that we're also going to give you the space is another capability, we're going to give you the time on your backlog, and I think that that's important. So the “Empress

of Software," where can people find you? Do you prefer Twitter or LinkedIn?

**Jessica Kerr**
Yeah, Jessitron on Twitter. I exist on LinkedIn, but I won't answer your messages. Yeah, or jessitron@gmail, jessitron@honeycomb.io. In fact, if you go to honeycomb.io/office-hours, you can sign up and have a chat.

**Scott Hirleman**
Awesome. Yeah, and I think talking about observability in the data space and how there's overlap and things like that, I think is interesting as well, of data observability as space. That's very, very kind of exploding, but I think there should be more of a kind of talking to people like yourself that have been...

**Jessica Kerr**
Yeah, and data engineers have a lot to add on the software engineering side as far as observability, because y'all have an analytical mindset, and software engineering starts out with the graph the logs mindset, and that's different.

**Scott Hirleman**
I love it, I love it.

**Jessica Kerr**
It's been great talking to you, Scott.

**Scott Hirleman**
You as well. Thank you so much for the time. I'd again like to thank my guest today, Jessica Kerr, aka Jessitron, who's the principal developer advocate at Honeycomb.io. She'd mentioned a few different things, including her contact information, which is all linked in the show notes, as well as a book and a couple of presentations and blog posts that she had done. So please do check those out as well.

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.

**Starburst**
With a massive move to distributed data architecture, it's essential to have access to all of your data wherever it is. Data Mesh emphasizes domain driven data ownership, data as a product, self service 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.