2:24

Alexey

**This week we will talk about Data Mesh. We have a special guest today, Zhamak. Zhamak is a principal Technology Consultant at ThoughtWorks and she's the inventor of Data Mesh, the thing that we will talk about today. So, welcome.**

# Zhamak’s background

2:39

Zhamak

Thank you, Alex. Thank you for having me. I have to just make a quick adjustment. Late last night, I did actually make an announcement that I have left ThoughtWorks. And I have started a textbook around Data Mesh. But we didn't get a chance to sync up. But, yes – I was a consultant at ThoughtWorks for more than a decade.

3:01

Alexey

**The first question I have for you is to tell us about your career journey. You briefly mentioned a part of your career journey just now. But maybe we can go a little bit back and you can tell us how it started?**

3:17

Zhamak

Yeah, absolutely. I mean, it has been a journey, as in I haven't stopped in one place, in one country, or one industry segment. I have moved a lot – I have moved around. I started as a software engineer, and I think I remember I was 14 (going too far back, but very quickly) I was 14, I lived in Iran, and it was a time of war between Iran and Iraq and there were sanctions. We weren't getting much into the country, not much getting out. My dad went to the UK on a work trip and bought a Commodore 64 and he came back with two basic programming books. [chuckles] On top of one of them had a picture of a computer that was hanging up handing out coffee or something and I thought, “Oh my god! Computers can do that.”

Since then I fell in love with programming. I became a software engineer later on. I’ve been in the tech industry for 20… 24-something years and the first half – more than the first half – of that was dedicated to deep tech products and R&D. I've done everything from firmware level, producing custom hardware in-house, to the largest scale distributed systems where data was a very important ingredient of the solution. With a larger scale, critical infrastructure monitoring before streaming was the thing, where analytics on streaming was a thing – we had a full stack system, building it. For the last 10 year, I came out of deep tech and went, I guess, across the board. I went to ThoughtWorks and worked with many larger scale companies that run the real world – infrastructure, communications, healthcare, and so on.

My focus has been mostly on distributed systems, initially, on the micro services and how to scale computing and solutions, I guess, applications for organizations that are complex. For the last five years, again, I made a transition and came this time to solve with Data Mesh. I worked with organizations to solve the complexity that surrounds data and getting value from data, and empowering people autonomously to get value from data. That led to the hypothesis of Data Mesh, so I was building solutions around that.

As of a couple of weeks ago, I've decided to leave ThoughtWorks. I realized there is a gap in the technology, mostly around enabling the experience of data folks, whether they are data producers or data consumers, to have a very peer-to-peer analytical data sharing model. I've now started very, very early days, a tech startup here in the Bay Area to build that new, reimagined developer experience and build the platform for it.

6:17

Alexey

**Sounds like a lot of fun.**

6:19

Zhamak

It has been. [smiles]

6:22

Alexey

**I think you mentioned what you did as a principal technology consultant. It was mostly consulting other companies about how to extract value from their data – how they should design their systems in order to make it easier for them to extract value from them, right?**

6:42

Zhamak

Yes. That involved all the layers of stack. Sometimes people think consultants are people that go and build a bunch of slide decks and wave their arms and then leave the company with a pretty slide deck. But that's not the case for ThoughtWorks. With ThoughtWorks, we did execution where companies needed to either introduce new capabilities or didn't have enough people and so on.

So yes, with my teams, I worked on building data infrastructures from the ground up, kind of like a data platform. And then on top of it, data products, and ML/AI use cases to take advantage of those data products. You can imagine it as more vertical slicing of all the layers of the stack involved.

7:35

Alexey

**So a customer would come to you saying, “Hey, it's a bit of a mess here. We cannot make any sense of data and what's going on. Can you please help us introduce some order here?” And then you would say, “Okay, just use Data Mesh. Here is a book.”**

7:51

Zhamak

[chuckles] It usually doesn't. I hope that it works out that way, but it doesn't. It doesn't work that way. Customers usually come saying, “Look, we've been trying this for a year.” The hypothesis around Data Mesh came with very specific questions. These are technologically very advanced companies here on the west coast of the US, where I was located. That's where I was meeting clients. They come and say, “Look, we've had a data strategy. We've done all of the on-prem, past generation of Hadoop and so on. Then we went to cloud and we had a data strategy – we moved all of our data to a big data warehouse on the cloud and we hired all of these data scientists, the head of personalization, we're using AI. These people are waiting and still not getting access to the data they want. Or if they build something we can’t really deploy.”

It was the process of data to value, and the value – whether it being an applied ML model or reports and dashboards that people can act upon – that process was full of friction, broken and very, very long. The customers usually look for (people are always looking for) silver bullets, “What can you sell me so that I can solve all my problems?” [laughs] And then usually, the process of engaging is getting involved and seeing “Okay. What are the use cases that you have in your organization? What are the friction points? Where are you in your maturity of technology? Do you have technical people in your organization to afford building some of these solutions?” and so on and so on. Then after that discovering assessment, you basically get in the trenches with the organization to deliver use cases and value while building their infrastructure or the team structure restructuring and so on. It's always a very, very tight collaboration.

# What is Data Mesh?

9:49

Alexey

**Does it have anything to do with Data Mesh? What actually is Data Mesh?**

9:56

Zhamak

[sinister chuckle] Well, maybe. I guess the only connection of that past is that – once you work at a level where you see a lot of similar problems repeat, you become a great pattern matcher. The patterns of problems, and then the patterns of solutions emerge from these many touch points. I think that the hypothesis of Data Mesh came from seeing these repeating problems. What Data Mesh is, is really an answer to some of the core challenges that we've had for a really long time. But they're becoming more pronounced because now we're talking about application of data *beyond* a bunch of BI reports in a BI team.

We're talking about application of data in every function of our products or services. Data Mesh was really an alternative way to get value from data involving how we structure our teams, how we even imagine or share data – the infrastructure and then the governance – so all of those pillars. It's a decentralized approach, meaning it's based on giving autonomy to independent teams without compromising quality or integrity or connectivity. So it's a decentralized socio-technical approach in managing, sharing, accessing data, particularly for analytical use cases at scale, in an organization between different business units or tech units or across organizations.

What it attempts to do is remove this “pipeline thinking” that data constantly needs to move to a pipeline, gets put in a pile, thrown processing or metadata and semantic and so on added, and then throw processing at it to get value from it. It really challenges that paradigm, because in that paradigm the time from data to value is very long. Data Mesh says, “Okay, how can we decouple these pipelines into smaller, self-contained units that encapsulate – whether it's a tiny pipeline or some sort of a computation and data – and metadata with data sharing API's that allows you, as a data user and ML engineer or data analyst, to analyze this directly (peer to peer) without a middle layer of a data team, directly access to data and run your analytical workloads distributedly and have a direct relationship with the people who are *actual* producers of the data without middlemen?”

12:56

Alexey

**You mentioned quite a few things and I think most of these things did not involve tools – it's more about how exactly you structure your team, how you structure your process, and how you build things rather than “Okay, use this, this, and this tool, and you will be good.” Right? So it's more about how exactly you plan your work, how you organize work.**

13:20

Zhamak

Absolutely, absolutely. But it’s also about how you organize your architecture. Honestly, it pains me when people say, “Oh, here's a database. Here's a bunch of tables in a warehouse. Go knock yourself out and get value from data.” Because that's a very tightly-coupled, fragile way of using data – that data cannot change. The contract that exists today for data sharing with a giant pile of files or tables in a lake is a very tightly-coupled and fragile contract between the data producer and the user.

When you think about that, these autonomous, independently moving teams building and sharing data, then it goes beyond just the team to say, “Okay, what is the future of data sharing contracts between these entities? What is the future of data computation? What is the future of governance?” Then it really very quickly leaks into your architectural thinking and very quickly leaks into “Okay, what kind of tool can I reintegrate in this new model?”

14:28

Alexey

**So it's not only about every team doing things independently, but also keeping the big picture in mind and also introducing some standards. If you want to use data, you have to have some sort of recommendations – it shouldn't be just a bunch of tables that nobody knows how to use, but a proper set of documents that describes what is there, how to use it, and so on. Right?**

14:55

Zhamak

Absolutely. You said a key word there that I want to double-click on, which is “big picture thinking”. In the data world, we are fairly isolated and every one of us thinks… [off-topic side-tracking] Yeah. So if you think about that big picture thinking, right now the big picture that I see over and over again that’s put on diagrams – whether it's an architecture diagram, technology vendor diagram – is about a big picture that is organized with this pipeline thinking.

The big picture we want to shift to is a mesh model, which is a kind of a graph model in a way, where value is generated through the links between these data products, through the interactions and exchange of value between these data products. Unless we are able to see that big picture, we can't really make any changes. You can't say, “I'll just make a change in one team and that will give value.” No, because no value is – the formula is (n\*n-1)/2. There is a number of interconnectivity and exchange of information between the nodes (nodes, of course, are architectural concepts and people). Yeah, that's *super* key.

# Domain ownership

16:34

Alexey

**I took a look at the table of contents of your book. I don't think I actually mentioned that you wrote a book about Data Mesh. But anyway, I took a look at the book and it's organized by what kind of Data Mesh principles there are – and the first principle was Domain Ownership. I think it's related to the thing we discussed, that teams work independently. There are units (maybe domain units) – can you tell us what a domain is? And what is this principle about?**

17:10

Zhamak

Sure, before answering that question, let’s step back for a minute. I started writing and talking about Data Mesh by first principles – by stating first principles. That was very key, because in today's world, we are thrown at so many tools and technologies that remove the ability for us to think for ourselves and we are constantly being reshaped by hand-fed, “Use this tool and magic will happen.” I really wanted to take a different approach. So, you're right. I started with this first principle and said “Okay, if we agree on these first principles, then each organization can come up with a novel and new way of bringing these principles to life and the implementations of data may look very different. Through that generation of ideas, in terms of implementation, the great ones bubble up.” That was the purpose of it.

There are four principles and all of them work together. There is a reason there are four and we can talk about that after. The domain ownership principle is about aligning data work – data generation, data consumption – with groups of people, with a team of an autonomous group of people. The domain is one. And the domain is often an aspect of your business that some business person is thinking about, has a specific set of languages, words, vocabulary, and an outcome. We want to organize the data sharing model or data generation model in a way that each of these business units, that are the direct producers or direct consumers of the data, can work independently and yet interoperate.

Let's describe that with an example. In the book, I use a hypothetical digital streaming company – I call it Death, but you can imagine a company like Spotify, SoundCloud, Apple Music, etc. If you have a business team and a tech team aligned with it, where their job is generating playlists – automatically curated playlists – the outcome of that team, that business domain of “playlist” is to really give an immersive and personalized experience to the listener. Then, that team with that outcome has a set of data that it’s generating, which is automatically classified, personalized, targeted playlists of music, and that team to do that (that's a machine learning model) requires data from *other* business domains. There is a business team that is thinking about the best class digital experience, so they're building the music players. They're a source of data, which is play events, or play sessions, how a user interacts with that player device.

There is a business team whose objective is onboarding more listeners. As part of their onboarding process, they have more information about the profile of their listener, such as maybe their age – not exactly individuals, that gets too creepy – but as classes, a distribution of the listeners, like the geographical location. All of this information can be directly consumed by these business teams. People get confused when I talk about domain. Really, domain is just, if you zoom out and look at your business and the various objectives of your business – and if you're a modern digital business, you probably have technical teams aligned with those business objectives and business units.

Let's go through a few cases. You have your listener team that's onboarding customers, you have your artists management – people that are managing the artists that are coming on the platform – you have your artists payments, people that pay the artists. These are all business function domains and they all generate data and they all generally take and they consume data. So we want to make a data production sharing model aligned with this unit and the reason for it is, that *this* model then scales out as you create more business functions, you expand, you grow your business into new areas, “Well, I'm going to create a new domain.”

Let's say we decided to work with partners at this company. We're going to work with these yoga studios to play their music or the Peloton auto cycling companies, for example. Then you bring your partner team and they have data capabilities internally. They're responsible, again, for sharing the data and consuming data from others for their data-driven solutions.

22:25

Alexey

**Maybe I'm jumping a bit ahead, because I haven't heard the other principles – you mentioned all these different teams, like music player team, artist management team, onboarding team, partners team. Let's say each of these teams has a schema in our data warehouse and they just publish data there, in the schema. Would it be a Data Mesh or is it too early to call it that?**

22:51

Zhamak

Um, I guess there are levels of maturity in terms of getting there. You might start with saying, “Look, we're still gonna have a centralized warehouse, but we're going to try to give *some* organization to how we structure the files or tables within this warehouse and structure the schemas.” That might be a good start, but it's not going to give you the outcome that you want. The outcome that you want is a *very* loosely coupled model of data sharing. The playlist team can change their schema, that can change their data – you can almost real-time share this information without really breaking anybody else’s. The data warehousing model of data modeling, interconnecting data so the correlations can be discovered and queries over multiple datasets can be executed – it's very fragile, again, it’s a tightly-coupled model. You know exactly what table and what columns – if you change that column or change that schema, your solutions that are doing cross-team or cross- schema correlations become broken.

In addition to that, if you want to remove any sort of friction in terms of discovery, in terms of understanding, there's a lot more involved in data sharing than a schema and a bunch of tables. You've got to share all of the other real-time (and when I say real time I don't really mean events) but things that are real-time can change and we want to share that, which is “Okay these are the guarantees of my data product. I update this monthly (or every second or every millisecond) the integrity of it. This is a near real-time data product.” But it actually has low integrity – it has missing information, it has duplicates versus “No, this actually gets reconciled nicely.” So there's a level of additional information that needs to be provided. Frankly, I'm not sure a data warehouse or data technologies that we had is the best way to share that information, because they’re built with a very different set of assumptions. But what you just described would be a good starting point.

# Determining what to optimize for with Data Mesh

25:26

Alexey

**Because I was thinking about this scenario – so the playlist team needs to access data from the music player team, and then from the onboarding team. If it's just one data warehouse then it's just a join, right? You have this data here sitting in this table, in this schema, then you have another table sitting in this schema, you just do a join, and then you have your data.**

**But in the case where it's distributed and everyone has their own tools, data warehouses (whatever they have) then making these joins becomes a lot more difficult. So how do they do this? Do they pull data first to their intermediate storage, from this team, then from this team, and then they do join? Then they also need to make sure that they actually have something to join on – like, there is a common key.**

26:16

Zhamak

Yeah, exactly.

26:16

Alexey

**How does it happen in practice? Yeah,**

26:18

Zhamak

Again, there are different levels of complexity. We have to decide what we are optimizing for. So far, what I'm seeing with our solutions and the biases that we have, is mostly about optimizing for machine performance. But if you're in a really complex organization, and in fact, you lose a lot of cycles to deliver value because you optimize too much for the machines, we have these very tightly integrated keys that you can't forever change, then no matter how fast your machine is, your final outcome is suboptimal, because you actually, those joints are not really delivering much value. You can't even build new use cases.

So just to start by saying there is a spectrum of what we are optimizing for. In the case that you just described, one might say, “Look, I'm doing Data Mesh within an organization. We happen to have standardized on some warehouses.” Maybe all of the data users are happy to use the warehouse, (which I find hard to believe), but let's assume that. Then, at the physical layer, one way of getting access to this data from different domains is warehouses. But at a logical layer, when people request access, when they discover even which table to go to, there is a dynamic set of API's that they have to call. So we can put a layer of indirection on top before you even get to the join to say, “If you want a playlist, actually, you call a different API that then will decide based on your version of the playlist you're requesting to which table to even go to.” That is a possibility.

On the other end of the spectrum, you're actually working with organizations that haven't standardized on one platform, one tech, or even maybe, to take it to the next level, it’s sharing data *across* organizations. It'd be crazy to say, “Hey, everyone, go on the same platform, same cloud.” It doesn't make sense. So we have to solve for that solution. Today, what we have – which is not great, but it's a stepping stone – we have federated query engines. You can have tables in different data warehouses and let's say you have that first layer API, it directs you to the right table and database and schema and then you can run your query in a federated way and do your joins and do all of the things that you want to do. Yes, you are probably sacrificing some performance, even though those federated query engines are getting faster and faster. You're sacrificing some sub-millisecond or sub-second performance, but that is giving you a level of freedom and scale from the human perspective, that you couldn't have before if everyone had to go on the same platform.

Then there are modes of consuming data that have nothing to do with SQL. There are modes of consuming data where you are reading structured streaming of data frames. Yes, you push some of your computation upfront to just pick the bits of the data that you want, but you're bringing in data to do downstream processing. And if you want to get really futuristic, that feature doesn't exist today, but I think that's where automation needs to head and that's why I thought, “Somebody needs to start a company to solve these problems.” That’s the future, where we put a stop in this kind of data movement model and say, “Okay, if where we want to get to is independent, rightful owners of the data independently changing, independently evolving – but yes, we want to have cross-cutting analytics and machine learning models – we have to define new data sharing API's that are in fact about receiving the computation, pushing the computation further and further up into those data products.

If I'm training a machine learning model, and I'm doing matrix operations, can I do this distributedly on the right source of the data no matter what platform they come from?” I think that goes beyond just having a SQL type that is about sucking data out. It's about putting the computation onward and only sucking the bits that are *really* valuable to the outcome that you're generating. I don’t think we’re there yet. There are a few startups that are thinking about this kind of federated machine learning training model and so on, but I think it's a movement that needs to happen for this model to be practical.

31:05

Alexey

**I guess the simplest way of implementing this is when every team has some way of accessing data and you (the playlists team) just pull it to your place, crunch it, and then create your product. Right?**

31:19

Zhamak

Yes. And then you have to think about the simplest way in terms of “What is the minimal set of guarantees and information that I need to publish and provide to other teams so that they can discover, so that they can understand?” It was fine, maybe to go to one centralized data team and say, “Hey, knock on the data team – can you run this analytics? Can you find this data? Can you create this data?” But that model won't work when you're in this distributed way, so standardizing those APIs that describe and provide metadata and so on is also key in addition to the data itself.

# Decentralization

32:04

Alexey

**So it's decentralized but there are some sort of central parts, like this API that you mentioned, this layer of indirection. We should define this in advance so that teams know how to communicate to each other. When we create a new domain, a new team, it follows the same API.**

**So if somebody needs to consume data from this new team, they already know how to do this because it's the same across all the interacting domains.**

32:28

Zhamak

Exactly. Absolutely. Decentralization and centralization are two sides of the same coin. If you and I are on two separate computers, two different time zones, but we managed to have this conversation with all of those folks on the Q&A and on the chat – the reason for it is that we didn't say, “Oh, hang on, everyone. We should all be on the same server to talk to each other.” No, the reason was because we've got a TCP/IP stack that communicates and standardizes the seams, the interconnectivity, and then it gives us autonomy to be in whatever stack that we want to be locally.

Absolutely the same thing exists with Data Mesh. We've got to standardize some cross-cutting concerns and one place to start with anything is about interconnectivity. Interconnectivity of these nodes – you can imagine them, as you said, like discovering them or joining them. This involves information about discovering this thing – what information it shares to say what it is in a standardized way. What language or what modeling language it uses to describe its data, its interfaces for the data sharing that you mentioned, or computation sharing. It’s identity management.

The thing that sucks most, frankly, right now, in the real world is the security authentication, authorization and just modes of validating access to data. Every technology has its own. Even on a single cloud provider, [chuckles] that is not sorted out so that you can access resources from independent services, independent accounts. So we've got to come to an agreement around that. There's a ton of learning. I mean, we've done the internet – so we can always go back and see what were the key innovations (a small number of innovations) that allowed this federated model of capability sharing and apply that to data.

# Data as a product

34:36

Alexey

**That's an interesting metaphor with TCP/IP. I think you mentioned this set of guarantees multiple times. I suspect it's related to the second principle, right? The second principle is “data as a product”. Can you maybe tell us more about this principle? What is it about?**

34:59

Zhamak

Yeah, Absolutely. What drove Data Meshes was autonomy, independence – it was more about moving and being able to have an infinite scale of different domains, different parts of the business and just domain-oriented ownership. *But,* very quickly, that can turn into a siloing problem. “Well, I'm in the playlists domain and I have the data that I want. I will suck in somehow the data from other places and keep it for myself.” How are we any better than the siloing of application databases that we have today? Data as a product was to invert our relationship with data and think about data as a product that we share. We measure the delight and happiness of the data user and that's different from a relationship we have today, which is that data are assets that we collect and it's precious and we don't necessarily want to share with anybody.

Data as a product is a set of underpinning practices and, again, technology that really focuses on the consumer first. Let's say, I am listener onboarding – I build listener registration apps, whether through a call center or through web or mobile – I receive listeners and I capture information about them. My job is to really optimize that process or the conversion from “Oh, I'm interested in this app!” to “I want to get free access and then try it and then pay for it,” is very smooth. So if I'm that team, then I'm generating a ton of data – the touch points with a user at the time of interest or registration. If I just collected that data – I got events from the plays, players, or applications, or web pages, and then I put them in some database or stream them and maybe someone downstream started using them with difficulty, we want to change that and say, “Hey, listener team. Of course, your responsibility is optimizing the process of engaging with listeners through registration. But also, you're responsible for the data that you generate to share that data with the rest of the organization based on their needs.” You very quickly realize that and you get measured.

As a product, you have a set of KPIs that measures the success of your role as a data product owner. From that, then you can think about, “Okay, if I'm giving a product or sharing a product or selling a product – I’m building a product – what are the characteristics that I need to be able to articulate and share for the audience to do a self-assessment, to see whether it's the right product for them or not?” And one of those characteristics is about all of the information that you need to share for a consumer to *trust* your data, and then self assess its usability. When you think about establishing trust, it's about bridging the gap between what you know and what the customer needs to know.

To bridge that gap, a set of information that you need to share are guarantees of your data products and then you can think about the guarantees as quality and time limits and integrity and completeness and a whole set of information. That changes all the time, right? It's not a static, “Oh, I shall have this level of completeness.” No. In fact, that constantly changes – so allow for that and then measure whether you are meeting those guarantees or not, and adjust your implementation.

39:02

Alexey

**I guess if we come back to that example of the playlist team consuming clicks from the player team – as the playlist team, you want to be certain that there are no big problems with the quality of this data and this is one of the guarantees we expect from the player team.**

**If there are problems, we also expect this team to tell us about these problems. This is the contract between us, our domain, and their domain. If we have this, then their data (the clicks) is the product that we consume. Right?**

39:36

Zhamak

Exactly. That's exactly it. That assessment is really a conversation. The data product owner or the data product manager is a role that Data Mesh introduces. Their job is to have that conversation with users (domains) that are interested in that data and say “Okay, look. Today 80% of consumers are happy with kind of low integrity, but real-time because they've been building dashboards about anomaly detection and how the player is failing or working. But now we have these playlist folks and they're actually not interested in every single event or every single click. What they're interested in is a bit more high integrity sessions of an interaction. From the moment that somebody starts playing the application, what music they listen to, in what order, which ones they skip, which ones they listen to. So they're only interested in a holistic view of maybe all of the listeners. They’re looking for aggregates – they don't even look for every single customer – they want to see *all* of the customers and *all* of the tracks, and the relationship between those.”

Then, in those conversations, you go, “Okay, actually, I have a data product about player click stream, but I need to now create another data product, or I need to create a new way of accessing this data product that gives the aggregate views that they're looking for. Yes, it's going to be *less* real-time. The guarantee becomes maybe this hourly, or whatever the processing window that makes sense for that aggregate. But the integrity is high.” So through that conversation, you then decide “Who should be building this product? Is it really the player team that should be building it? Or is it someone in the middle or the consumers themselves?” And then you manage your product ecosystem in that way.

41:39

Alexey

**And then each team has these data product managers that you mentioned. It's their job to make sure everyone agrees on who is doing what. Right?**

41:47

Zhamak

Yeah, exactly. Their job is managing their data as a product for the entire spectrum of consumers. Yeah.

# Self-serve data platforms

41:58

Alexey

**Okay. So what's the next principle? I have it in the notes. It's the principle of “self-serve data platform”. Can you unpack this? What does self-serve mean? What is a data platform?**

42:11

Zhamak

I must say, the platform, by definition, should be self-serve. Self-serve is almost a redundant word in that phrase, but I wanted to kind of *really* emphasize and make it bold so we don't forget it. If you think about this model of autonomous business focus, I take people that are focusing on a particular domain – teams are now accountable for data products and their data computation pipeline, their data API – it's a lot of responsibility. The way the data systems or data platforms or data infrastructure is provided today, that requires a *high* level of expertise. That almost makes Data Meshing impossible from day one, because we can't recruit all of these people.

What does that even mean? I put that there as a placeholder there to say, “If we really want to empower and enable these embedded data people in the domains, we need to rethink and reimagine our data platforms to make life of someone like a vanilla developer really easy to generate data products and use data products – or the life of an analyst really easy to consume that data and be able to play with it so we don't have to constantly introduce this intermediary role of analyst engineer or analyst as an engineer or ML engineer.” Maybe we can make that a little bit smoother and that requires really rethinking our approach to data technology in a way that removes that high degree of specialization. But, of course, you still need to have development experience and so on, and understand the characteristics of data, statistical modeling and things like that. Of course, you need to have capabilities that depend on your role – producer or consumer. But that proprietary tech expertise that you need to have needs to get reduced.

The surface of this platform needs to kind of go up a little bit in terms of abstraction. So it was really a placeholder. In the book, I give examples of value stream like a working model – the day in the life of a data product developer, the day in the life of a data product consumer, and what is their level of abstraction that they can expect from the platform in order to do their job easily, to do their job faster, to do their job right? And I think that's missing. A lot of organizations are building that right now because it just doesn't exist. And it's my mission (the next, next, next mission) is to build the technology that delights the experience of data developers, (I’m just using the umbrella term ‘data developers’) whether they're consuming or producing. That's the third principle.

45:15

Alexey

**If I can attempt to summarize, a data platform is a place where somebody who is not necessarily a data engineer – who doesn't know how Spark works, who doesn't know how Kafka works, who doesn't know all these things – they can just come, find the data they need, query it, do some analysis, and then maybe pull this data and start using it for their team, for the products. Right? So they don't need to hire a team of data engineers to be able to do this.**

45:47

Zhamak

Yeah, and I think we've got to be careful with that because that very quickly becomes, “Oh! I need to have a no-code/low code platform that nobody can test and understand afterwards.” So I’m definitely not advocating for that. I think software engineering practices, or good engineering practices, are evergreen and have to be built. But I think there is a set of services – someone might say, “Look, I am producing a data product (or I'm consuming a data product) and I want to work with data frames, or that.” So they might still use Spark as part of their work, but they don't have to worry about scaling it or running it or scheduling it. And they don't have to worry about, “Oh, this is just a vertical. I'm just writing the pipeline. And then I have to worry about the storage part of it. And then I have to worry about the security part of it.”

There is a new experience generated that says, “Look, if you initialize a data product, I will give you all of the aspects involved in a data product and you just focus on that little part, the Spark code that you have to write for data processing (or that SQL that you have to do). You don't have to worry about anything else.” So you might still code and have visibility to these tools, but your experience using those tools changes quite a lot.

47:14

Alexey

**When you said that we need to reimagine our data platforms, it was plural – platforms. Right? Each team has its own platform, or we have one global one, or how does it work usually?**

47:26

Zhamak

Yeah, I think everybody wants one giant, big, data platform that serves everybody's needs, right?

47:32

Alexey

**Yeah, that sounds cool. Right?**

47:35

Zhamak

[laughs] Is that reality? I don't think so. I think, again, it comes to… you see, Data Mesh, at its heart, embraces chaos and complexity. If you embrace chaos and complexity, then your solution has to be able to deliver value fast, reliably, and responsibly, despite the human chaos and complexity. If I said right now, “Oh, you decentralized all of the teams and have one platform to rule them all.” That is against that principle that I just mentioned. So I think, again, with the platform, the reality that is happening is that large organizations where Data Mesh makes sense – complex organizations – almost every country, sometimes, each business, they have their own technology and tooling and they don’t want to standardize.

The parts of the platform, again that you have standardized and you have common capabilities are what interconnects these platforms in terms of data sharing. And the parts that you probably don't care about – like, if one team wanted to use Prefect, or Airflow, or whatever for their data processing, so be it. Another team might be using some serverless technology for their data product (the computation with a data person) that's okay. But as long as those two can discover, understand, share, use and connect data, then have as many platforms as you want.

# Data governance

49:13

Alexey

**Is this related, by any chance, to the next principle of data governance? What was the next principle, actually? I don't remember. It’s governance, right? Is it related?**

49:25

Zhamak

Absolutely. All of these principles – I have a diagram in the book that describes the interrelation of these things together and why there's four and not three. In fact, I didn't have the first one for a while and then I added it because, based on experience, I realized, “Oh, God. If you don't get this right, we're doomed.” [chuckles] Where there are the cross-cutting concerns that we need to agree upon as an ecosystem of independent nodes on the mesh, and need to be able to implement in an automated computational fashion. “Governance” is a word we use a lot in the data space. In fact, it's a word that is not as much used in the microservices operational world – we just call them “policies” or “cross-cutting concerns”. So I use the same language that the data community uses – the word governance.

To me, it's one of those scary words that is very hard to implement without putting controls that slow people down, so I had to add the “federative and computation” to it in order to fit in this model. The problem that can arise from diversity of the platforms, or diversity of the data products and independence of those is, again, a lack of trust at a global level. How do I know that, now that every team is doing its own thing in terms of data sharing, the policies that are important to the company (not just to one team, but important to the company) are implemented? How do we know that we are all talking the standard language when we're exposing data? Those are those common cross-cutting concerns around all of that – all of the existing governance concerns like privacy, security, various policies, as well as the standardization for intercommunication, need to be implemented, but need to be implemented in a way that, again, it embraces the complexity and chaos of the organization. It embraces moving fast, but responsibly, without breaking things.

Then it says, “Okay, if we apply engineering to the problem – I did come from a very software engineering-heavy view of the world and I admit that – how did we solve this before?” You can go back to the internet or go back to my source of “How did we solve that before?” We solve that with computational-heavy as in heavily-automated capabilities that get embedded into every individual service in the microservices or in individual data products. There are many examples of that and there are architectural ways of doing that. That's the computation part.

For example, if a team says, “Oh, I want to share this product in this mesh. I'm going to run this command and get all of the things that I need to have in a standard way.” One of those things would be a way of configuring your privacy level and have the platform or that computation part take care of encryption at the right level, access control at the right level. Then the question of “Who defines these policies?” That's a federated operating model. These domain data product teams need to be part of that global conversation.

52:48

Alexey

**I guess this is related – we talked about data sharing API – having this API actually belongs to this principle, right? Not “belongs,” but this principle says that you should have one. Is that right?**

53:02

Zhamak

Yeah, this principle influences the common pieces across all of your data. API is one of those common pieces, but there are other common pieces such as the policies that impact the data itself – like, what's the retention policy? What's common is not that everybody has the same retention policy, but everybody has a retention policy with different values. Some data can be kept forever, some data, perhaps because of the sensitivity, has to be kept only temporarily, like for a few hours.

Having a thing that says “Every data product shall have a retention policy” is a common characteristic, but then automating configuration and enforcement and validation of that policy is the platform piece. And then giving autonomy to the teams to define what their value for that potential policy *is*, is the domain-oriented piece of it. Then exposing that policy as part of your data product discovery or understanding the information is the data part of it. So it does impact all of these pieces.

# Understanding Data Mesh

54:20

Alexey

**I apologize that I forgot about the questions. We have a couple of them. I think this could be related to what we just discussed – all these four principles. The first one is “What is the most important thing about Data Mesh for us to understand, learn, and adopt?”**

54:37

Zhamak

What is the most important thing to learn? Is that the question?

54:41

Alexey

**“To understand, learn, and adopt.” We can start with the “understand” part first.**

54:48

Zhamak

Yeah. I had a lot to do last year, but I decided to sit down and write this book so that I can put all the information that you need to know in order to understand it. I think if you – I’m not just not trying to sell my book – I can share a link that you can freely access with a short period of time on O'Reilly. That's what O'Reilly offers. But I think when I wrote this book from the perspective of what people need to know to understand (before this concept was completely bastardized by the industry into something unrecognizable) was that you first you need to understand “why” as in “What conditions have led us to think about Data Mesh in the first place?” And if those conditions don't apply to you, don't even bother about the rest.

Then second is the “what” like, “What are the first principles that drive this definition so that you can think for yourself about how to apply those first principles to *your* exact implementation?” And then the third is “how” and the “how” involves both technology, architecture, and organizational change. There *is* an organizational change. Yeah, I think you've got to kind of understand it top-down, today. If we were five years in future and Data Mesh was just business as usual – everybody did Data Mesh and there were so many tools that help you bootstrap – maybe as a practitioner, the question around “why” and “what” has been answered, and you only need to focus on “how”. And even then there is a platform that serves you – you don't need to get philosophical about it. [chuckles]

You can just run a command and get data products. That's just how life is implemented. So then Data Mesh really moves into the background. It becomes “just the way we do things,” right? We don't even talk about it. But if you ask me that question today, we still need to be informed about “why” and “how” because there is *so much* misinformation and “opportunistically published” (incorrect in many cases) content out there that I would, as a learner, want to protect myself against and have a way to think and judge for myself.

# Adopting Data Mesh

57:11

Alexey

**Then there was another part, “What is the most important thing about Data Mesh for us to adopt?” And I think it's related to a question from Jeffrey, which is, “What is the best way to start implementing Data Mesh from scratch?” We don't have a lot of time, maybe you can summarize it?**

57:27

Zhamak

Yeah. I think part five of the book is actually just that. I introduced the whole model around a kind of iterative, end-to-end, business-driven, use case-driven way of implementing this. And I give a ton of tools around even how to measure whether you're doing the right thing, how to select the first domains, how to select the first capabilities of your data. So there's a lot of content in the book that I recommend for you, if you're interested to have a look. But in short, I would say – start with the first self-assessment. Are you ready? There's a spider graph in the book that says “If you want to apply Data Mesh today, you really need to have top down support.” Because you're talking about a transformation – it's not a little Skunkworks project that you can do in a corner somewhere and say, “I've got this.” It’s a scale problem.

So yeah – do you have your executive support? Do you have the type of technologies that you need to have? Do you have the right DataOps or DevOps practices? There's a lot of software engineering involved and engineering involved in doing Data Mesh. So start with the self-assessments, find allies, find parts of the organization that lend themselves to this model first, and start small – with use cases that touch one or two domains. Don't start with a marketing use case, because they need data from all of the domains to deliver even the smallest piece of value. And you build your platform iteratively and smartly.

Don't start with completely decentralized models. Started with a team of different disciplines that come together, like different domains, or a platform team or governance team come together, but they work *very* collaboratively initially. Then, once you establish your ways of working in your interoperability layers, then you can become fully autonomous and decentralized.

59:20

Alexey

**Do you have a couple of more minutes?**

59:22

Zhamak

I have to check. I think I do, but let me just be responsible and check my calendar quickly. [chuckles] It takes a couple of minutes to find my calendar. Let's bring up the next question.

# Resources on implementing Data Mesh

59:51

Alexey

**Yeah. This should be a short one. “Do you know of any good reference implementation for Data Mesh where we can look at this and learn from this?”**

60:03

Zhamak

Super good question. There are not many. I don't. Short answer – nothing that's publicly available that I can point to and say “Go look at that implementation. That's a good one.” But there is a talk that a colleague of mine and I gave a while back. ThoughtWorks has a Data Mesh website and there is one around, I think “lessons learned” and in that one, we share a little bit about the technology that we use. But there isn't a Git repo that I can point to and say, “Go have a look at what these guys have done. Looks pretty good,” unfortunately.

One other place that I would say go and have a look is the Data Mesh Learning Slack channel and Data Mesh Learning Meetup. There are a bunch of case studies people have shared in terms of their implementation. Every implementation is slightly different and nothing looks like this future that I'm painting for you here. So we are *really* in the early stages. I kind of went through the microservices API revolution in my career and it was like 2011-2012. We were excited about this, but nothing existed. Containerization didn't exist, embedded web services didn't exist – none of it existed. So we were doing these kinds of fugly micro services on top of a big giant web application. So that's where we are right now. We are kind of banding it and hacking it together. But those two places should give you some ways of thinking about how people have started with the technology we have today.

61:46

Alexey

**Yeah, I started imagining how it would look – how it *will* look like in 10 years. It probably will be quite exciting. With some special [cross-talk].**

61:54

Zhamak

Absolutely!

61:57

Alexey

**Okay. Yeah. Thanks a lot for sticking around for a couple of more minutes with us and, in general, for sharing your experience, expertise, your knowledge, and for answering questions. That was a fun chat. I learned a lot. I think everyone who was listening in also learned a couple of new things. So yeah, thanks for joining us today.**

62:19

Zhamak

Thank you for having me. And thank you for the questions and sorry that we didn't get to all of them.