A.9 Expert I

Interviewer
Okay, I will shortly give you some background information. I'm Tom. Something went wrong with the internet. I can't hear you. Now I can hear you. Okay, great.

Expert I
Okay, yeah, sorry. I pressed leave the meeting because I'm recording. Just while you're starting recording, I just want to say on record that I'm not representing *. Yeah, sure. You cannot use * name or branding anywhere in the output of this.

Interviewer
No, sure thing. No, I will make sure there are no proprietary names or anything like that. I'll get in trouble otherwise. I'm 24 years old. I have a background in industrial engineering. So I kind of came from a business background and the last two years I've been studying data science and entrepreneurship where we try to combine the best of both worlds. So we make, we try to make data driven business decisions. And I think this has a lot of similarities with data mesh as well, because it's also an organizational thing. Yeah, so that's what I've been doing the last couple of years. Well, actually, the last two years and now I'm doing my master thesis in data mesh. And to be more specific, I'm focused on the architectural design options that are related to data mesh. So what kind of architectural components can we implement in our data architecture? So, yeah, that's in a nutshell what I've been doing the last couple of years. Nice to meet you. Yeah. Can you perhaps share some background information as well.

Expert I
Yeah, sure. So my name is *. I'm a, I guess, a data engineer consultant for *. I've been in the role for over four years. I work with *'s most strategic customers to help them onboard onto * and make the most of their * investments. I specialize in data, specifically data governance and security. And I've worked with a number of large enterprises to help them build their data mesh vision on *. To migrate most of all, but also to help them align to the new ways of working on the cloud and use the cloud migration as an opportunity to reimagine and refresh the ways of working in an organization. And data mesh is something that I have been focusing on for the last two years. Started with a customer asking me to do data mesh on *, and I didn't know what it was at the time. So I said, sure, let's try it. And then here I am. I've written most of the go to market content on data mesh. So the white paper, the five parts article on data mesh on *. Yeah, so that's me. My background is in engineering, but I have a bachelor degree in psychology. I see myself as someone who can interface between the technology and the business side. And I use that skill regularly with my customers to help them with their move to *. So happy to help and participate in your study.

Interviewer
Yeah, thank you so much and great to meet you. Yeah, we can perhaps deep dive immediately into the material, because I think there is lots of things to cover. I encoded everything in Python, by the way, because I expect some iterations before the favors can be considered perfect. So, yeah, it's so much easier to do everything with coding instead of making thousands of Draw.io diagrams. But there have been some transformations, but I want to keep the interviews consistent. So let's look at the PowerPoint. I think that's more convenient. So this is the first one. This is more the high level concept of all the other decisions that are linked together. I identified six decisions you can make during your data product implementation. So this is only the first framework in a series because there is also a self-serve platform and of

course, the federated governance. The first decision I identified was what type of data product can be developed. After that, there can be a decision on migration or greenfield development. So do we want to start from scratch or do we want to migrate from a legacy architecture?

Expert I
When you say type of data product, is that the two types of data product or what are the different types of data product?

Interviewer
The way you can expose your data products.

Expert I
So it's a consumption interface, it's not the data product itself.

Interviewer
Yeah, that's already a good feedback point to rephrase that.

Expert I
In my head, I see a data product as a collection of data resources owned by a single data product on a team. And the data product has multiple interfaces that are being exposed directly to their consumer. And each interface is optimized for a particular consumption pattern and the interface should be accompanied by, for example, a data contract that provides guarantee about backwards compatibility, data quality, availability of time, consumption, the way in which you can access the data, etc. Just to clarify, I can talk about type of data products, so many types of data products, but I think what you're talking about is a data product interface.

Interviewer
Yeah, that's a good one. I think I can rephrase that. That's good to just have a quick high level overview of all the decisions I've observed during my studies during my grey literature study and grey literature is actually all the things that haven't been officially published in an academic journal. So medium posts, YouTube videos, white papers, things like that. That was all the content I used for my frameworks. So we can perhaps start with the high level overview quickly go over all the decisions and at the end we can go back to this one and check if there are some missing decisions over here, etc. Or something needs to be finetuned, etc. So, yeah. Another decision is how does our data product interact with other data products with the self-serve platform there needs to be a management layer. So how does that interaction turn out and of course the consumers. Then we can deep dive more in the anatomy. So what's happening inside the data product itself, what kind of architectural decisions do we need to make here. Then we can talk about the data product interface / contract. So what kind of ports do we need and the last decision is how do we want to deploy our data products. So yeah, this is the first decision I observed. Yeah, we talked about it a little bit already, but I identified three main options. These three over here. So you can expose your data product as raw data. You can expose your data product as an algorithm and that can be two variants. So it's an optimization based decision support system. So this is more like a BI tool, Power BI or dashboard tool, etc. Or an AI / machine learning model. And the last one is that you can derive your, well, you can expose your data product as derived data. So some small transformations are applied to this data set and the hybrid.

Expert I
Sorry to interrupt again. I think you could benefit from a clear definition of what a data

product is, what a data product interface is, what is a composite data product. And I think, you know, because the reason why I'm saying that is the definition influences the way you talk about the type of data product. Because here you're saying raw data can be a data product. An algorithm can be a data product. Derived data can be a data product. This decision system is a data product. Now, the aspects that are in common in all of these different data products that can be generalized. So, for example, when you say raw data and derived data, you are saying this is a, I don't know, maybe your interface is a table. Right. That's a very specific type of interface that can imply a certain way in which you can consume it. And the table could be in an operational system, it could be in an analytical system. Or is it an API? Right. Is it an API that I call to get, instead of all the customers in this region, this particular customer ID, all the information about this customer ID. And then maybe that's, you know, what are you talking about when you're talking about the algorithm. Right. So what does it mean to expose the data product as an algorithm? Are you exposing it as an API interface that somewhere you can interact with based on a standardized input output interface? Or are you taking an export of the entire model and then giving it to the consumer? And I think these aspects need to be clearer. And then, you know, as to the final point about BI tooling, is BI tooling a data product? That's a question that can be answered differently based once again on the definition of a data product. In my view, a BI tool is a way of consuming a data product. It is not a data product in itself. People may argue differently, but once again, based on your definition, you can construe any of this in multiple different ways. And my suggestion is to have this to converge on a definition before you start branching off onto multiple types of data products as you have here.

Interviewer
Yeah, that's a good one, because now I'm thinking by myself, yeah, perhaps a dashboard is not really a consuming data product. Because, yeah, it actually extracts information out of a consuming product, I think. So it's not a product on its own.

Expert I
Yeah, it's pulling data from somewhere, right? Like a table or an API. So that is a product interface, not the dashboard itself. Yeah, okay, yeah. I should really re- That's my view anyway, I'm sure other people can disagree and I'm happy to debate. But it all starts from having an agreement on a definition. Unless we agree on a definition, we can't agree on anything else.

Interviewer
Yeah, I would agree with that definition. Yeah, okay, I think this is one of the minimal...

Expert I
Another thing I would recommend that you read about is Zhamak's book. I've read that one. Okay, cool. Well, she has this notion of a source-aligned data product and a consumer-aligned data product. So in a data architecture, you have, you know, like the raw layer, the harmonized cleanse layer, the integrated layer and the presentation layer, right? That's the kind of four layers in the data architecture. And data can basically fit into one of those. Yeah, I think that's more granular than it needs to be. I mean, so Zhamak has very specific terminology. She uses source-aligned data for that and consumer-aligned data for that. In my view, you know, all kind of data architecture has four layers, which is raw, cleanse, integrated and presentation. And a data product can fit into one of those four layers, right? And raw and cleanse may be our consumer, sorry, source-aligned data product. They are data that is a somewhat representative copy of a source system. So maybe a database export with minimal transformation and then the consumer data, consumer-aligned data product have been transformed in such a way that facility, a consumption use case. And that, and the typical data architecture aligns with

the aggregator and, you know, the integrated or aggregator, whatever layer you want to call it, and the presentation layer. And then, yeah, I think, you know, the high-level mapping is source-aligned, consumer-aligned and then within that there's kind of four layers. So raw data is, you know, on the left side, derived data can be anywhere on the right side. You know what I mean? Because anything that's derived from the raw layer is derived data set. So what does that mean in practice? Like high-level aggregation and low-level aggregation? I don't know if that's a useful differentiation. You know what I mean?

Interviewer
Yeah, that was one of the practitioners who told me to distinguish between some small transformations you make to the data set. So, for example, the cleansing part you mentioned and the high-level aggregation. So we're really aggregating stuff from different domains.

Expert I
Yep. So I agree with that, right? So that means going from the raw layer to the cleanse layer. Yeah, exactly. And then if you are doing some aggregation, then that's, you know, going into the integration layer. So, yeah, I think just have a think about that.

Interviewer
Yeah, that's a good one. So you would perhaps do this source-aligned data product item over here. You would put that further upstream instead of these. That would be the first thing, right?

Expert I
Yeah. Are you building a data product that is aligned to the source system? Or are you building a data product specifically for consumption use case? A data product can be both. It can also be an aggregated data product, for example. I think that's a term that Zhamak used. But yeah, I think that's probably kind of the first decision and probably the most granular level of a data product that you would get. And then everything else is variations of one of those.

Interviewer
Yeah, yeah. Thank you so much. This is very valuable feedback. We can perhaps just check the next decision real quick because we're running a bit out of time. Yeah, so in terms of migration or greenfield development, we can, of course, start from scratch. But we can also choose to focus on, for example, master data management. That can be a master data management data mesh so that we really stick to that pattern. We can do Strangler Fig. Do you know what Strangler Fig is? I know about all of this. Sorry.

Expert I
Yeah, OK. Yeah, so yeah, this one of *'s articles.

Interviewer
So Strangler Fig is really about decomposing your current monolithic architecture. So slowly decomposing it over time, extracting, for example, just one surface out of it and making data product of it and building your data mesh. Yeah, at the same time. Perhaps I can show you a picture real quick. That picture always says more than a thousand words. Strangler Fig. It's a bit of a weird name, I think. As soon as you see the picture. Yeah, so it's like this. So it's about creating your data mesh over a certain period of time. Yeah, and the other one was to maintain focus on zero trust to make sure that each data product has its own access controls, etc. And CQRS is actually the last one. So it's about segregating your read and write function.

I don't know. Do you agree with these migration patterns or would you define it differently?

Expert I
I think they have very little to do with data mesh. OK. Sorry. Yeah. So master data management. Master data management is a whole field in its own with vendors trying to sell solutions. And they mostly go down to records management. So how do you tell if two records is the same? Now, in a data mesh, you have and it's about identifying the authoritative source for a particular record. In a data mesh, you have the data product is the fundamental building block that is an authoritative data source. And if you say I need a master data record for customer across multiple source systems and that is a consumer line or aggregated. And there's no such thing as master data management in the data mesh, in other words, because all data are massive data, quote unquote. Now, I mean, there's a caveat to that, which is all data products are vetted and verified and have an owner or master data. And there will be a good number of data products that are not that do not meet the minimum standard for a data product and therefore may not be considered authoritative data source. But that's a decision that we make on the governance structure. Now, on the governance structure, all data products by definition have a data interface that is shared with the consumer. So all data products needs accurate control. Right. So then zero trust has kind of very little to do with all. It's a one approach, I guess. But the way that zero trust is being construed is a little bit misleading because zero trust originated in networking where the paradigm is you do not fundamentally trust any network. You do not fundamentally trust any agent in the network. So if you are connecting as a machine from an on premise network or from a VPN, you're still treated as an external identity and you are verified in the same way. That's what zero trust means. I think there's a little bit of contortion here to apply it into the data domain. And I find you can invent new terms and whatever you want. I've read the article and all that boils down to is fine grained access control. And there is one aspect you may want to implement in the data mesh that is appropriate for a certain class of data that requires such level of management, right? Such as PII, for example. But you don't need to do all of that if you're managing PII. And it's just different level governance that you enforce across the data product landscape based on the sensitivity and classification of the data. Strangler Fig is, you know, could work. Now, Strangler Fig is about migrating. So once again, it's a bit of a contortion of ideas for data mesh because Strangler Fig is about migrating a monolithic application into microservices where each of the microservices services are owned by individual teams. Now, you can argue that that applies in the data mesh. You have one team owning the data warehouse, monolithic data warehouse, quote unquote. And you break it into multiple subcomponents and then build a sub-data product, so to speak, and have multiple teams manage that. But fundamentally speaking, Strangler Fig has nothing to say about domain ownership and how the data product should be owned by the team, either the source team or one of the basically not the central data engineering team. Yeah. So I think, yeah, you can use it, but it's like once again, it's just one piece of a much bigger picture. And CQRS, I think, is purely an application, kind of a way of interacting with the data. And it's one way to approach it. Yeah, it's just one option. You can choose any number of ways. I don't know. I wouldn't say that if you do in Greenfield, you have to do CQRS. There are many use cases where you should not use CQRS. It's not worth the complexity. And so, yeah, I think it's a choice, like all architecture choice that has to be made consciously, right? Like, we're waiting off the benefits and the trade off of that particular architectural decision. So these are just, to me, architectural decisions that can be made, but not really migration, fundamental to migration or even fundamental to data mesh.

Interviewer
Yeah, that's a good one. Yeah, I was at least confident about this one, to be honest, because,

yeah, most of the articles talk from the Greenfield development point of view, I think, so that we just start from scratch and building up a whole data mesh. But I think we should also take legacy modernization into account because the legacy architecture is already there and it's very costly to just put that away.

Expert I
So you can't ignore it.

Interviewer
Yeah, so that's why I'm thinking about some kind of migration patterns. But yeah, I need to rethink this.

Expert I
Yeah, yeah, I think the only thing that can be classified as a migration pattern is the strangler pattern. Yeah. But the others, they are more. Yeah, yeah, but I guess the other just architectural patterns. But yeah, from the perspective of migration. I'm not sure if there's anything that's special about data mesh, let's say, you know, you have a migration scope, you define the scope, you define a kind of factory pattern to complete the migration, you find the people who are responsible for each component, and you agree on a roadmap and you deliver on the roadmap. Yeah, I don't know, I just feel like that's the fundamental aspect of the migration and you can make it fancy by saying I'm using Strangler Fig, but at the end of the day, it's a standard migration project.

Interviewer
Yeah. Yeah, okay, I agree. Okay. I think the point is clear. I really need to rethink this, this framework over here. So we can perhaps go on to the next one. This one is really extensive, to be honest, but it's about, yeah, it's like the outside layer of the data product. So what's going on in the infrastructure? So how do we communicate, etc? So I'd identified several patterns, for example, the schema registry, I think that's one that one was really important to make all the change events user friendly, so the user can actually see what kind of changes are going on in the mesh. Yeah, the central data product catalog, an event streaming backbone, such as Kafka, but I heard from several other practitioners that I was a bit too focused on event streaming, and we can also do batch processing, of course. So for example, I could add the blob storage over here that we can, yeah, save, we can store certain events there, or the query table, things like that. There can be a shared storage. And one of the practitioners told me to really have a shared storage because when you do internal storages for each data product, that would result in like 10,000 different storage accounts when you have 10,000 different data products, so that can be very costly. We have API invocation, of course. So let's focus on the three down here, and these are just variants of these three. Let me pick my pointer that's more clear. So we can have REST APIs, GraphQL, gRPC, but at the same time, there can also be a SQL access point, which is really feasible for the, well, really accessible for the data analyst. And on the bottom, we see more like non-functional requirements, so security controls, in-memory cache, and a query catalog. And what I mean with the query catalog is that it's some kind of manual for the data analyst to know what kind of queries are possible to access data in your data product. So it's more to make sure that the data analyst is not doing some weird query, which will be very costly at the end. So that's more like a query catalog. Some articles mentioned that one as well. So do you think some of these patterns, you can resonate with them or are some of them just really out of the blue?

Expert I

So I think once again, there's a question about definitions, which is, what is the data product? How would the data product consume the self-serve platform? What is the responsibility of the self-serve platform? What's the responsibility of this so-called management layer? And how does that integrate to the self-serve platform? How does the data product work with this management layer? And also, fundamentally, what is a data product? So as I said before, data product can have multiple interfaces. Interface can be a stream, interface can be an API, interface can be a SQL endpoint or a table. So I think you're kind of in the middle of listing different types of interfaces. And each interface in my definition has to be accompanied by a data sharing contract that specifies the metadata relating to who owns the service, what the data contains, how you can use the service. Have you heard of data card? Data card? Yeah, let me share that with you.

Interviewer
Sure. I think I've seen it before in one of your white papers. I'm not sure. Yeah. Yeah. One of your Google Cloud white papers.

Expert I
So a data card is a document that describes information about the data set and how it can be used. I mean, you won't use everything in the data card, but that is in the data template, but it gives you a kind of exhaustiveness of all the things that could be included, including whether the sensitivity classification of the data, for example, data set, who owns the data, the risk of the data, maintenance plan, provenance about the data, how it was collected, which systems were used to adjust and process the data, how it can be used, which was the original intended usage of the data, things like that. And then finally, also access control, like how you can access the data, what are the restrictions on the data retention of the data, depending on certain types of, for example, PII data or some regulation in certain industry may require that you keep data for X number of years, but no more. So all of that needs to be outlined in the data contract and automatically enforced by the platform.

Interviewer
Yeah, it sounds like the enterprise data catalog. Do you see some similarities with that?

Expert I
So here's my problem with the catalog, right? A catalog is where you, it's a data swap for metadata. You dump metadata, no one knows whether you can trust that metadata and no one knows how to even use a data catalog. So that's all the data catalog I've ever seen. What is more important is how you manage the metadata, right? Just like you have to manage data as a product, you have to have a process for managing metadata for the data product. All that has to be standardized so that you can automate processes to verify and drive enforcement policy. For example, data set that are classified by the data owner as, I don't know, confidential, according to a certain privacy classification within the organization, will automatically have its access restricted by the platform automation. That is an example of metadata driving governance automation. And there's a clear responsibility of who should define the classification category, who should define the policy that ends up being automated and who is responsible for actually defining the correct metadata for the data product. The data product owner is responsible for the data product. They have to keep it updated. It is their responsibility, but the platform uses that metadata to drive both information to enforce all the things that you have at the bottom, as well as to make the data discoverable and accessible. Because if you're missing, for example, sensitivity classification, then maybe the worst assumption that we can make is that everything is sensitive. Therefore, no one should have access to that. No one can

access means no one can discover it. And that's the job of the data product owner. And I think this is where, once again, you need to have definitions and clear responsibilities of each of the components in the data mesh and how they interact with each other. And then once you have that, then all of this is driven from your definitions.

Interviewer
Yeah, that's a good one. So can we do a data card on top of this? Or do you think we should change the center data product catalog in a data card so that we should remove this one?

Expert I
You need a catalog. I think you need a catalog or you need an interface with people to discover data products, request access to data products, understand what the data product contains. And typically a data catalog is that place in an organization. The problem that I have with most data catalogs is that there's no process to manage metadata in the data catalog. And even data card itself is just a template of what metadata you can use. There are recommendations on how to define data card, how to obtain it, etc. It all boils down to automation. So I think data card can feature as a potential structure for a data contract. But it still needs to be stored somewhere. It needs to have a process to maintain an update. And usually that place is the data card. So you should probably have that there. Yeah. Even the schema registry. Yeah, schema is a property of the interface. If you're exposing an API, you need to have the API specs. And that includes the schema of the, for example, message body that you can send and the schema of the response that the API will give you. Similarly, if you're exposing a table that comes with the schema and if you're exposing a table, you're also saying, I have, as a data product owner, guarantee that this interface will not change in a backwards incompatible way. Because if I do that will break off my financial consumer. I will also need to put guarantees around the availability of the data, the freshness of data and whatever standards the organization may have about aligning terminology to key business terms. So access control, identifying a particular column as a sensitive attribute. So basically, in the data mesh, in my definition, the only aspect that is central to, I guess, all the principles, right, data product, data platform, federated computational governance, is this interface's contract. And that drives everything else. So the data product owner is responsible for defining the content of the contract. The governance policy owner defines the shape of the contract and the policy that should be driven from the content of the contract. The platform team automates and enforces the contract based on the policies and then the consumer have to, you know, all of that allows the consumer to discover the data product, to trust the data product and to find a consumer data product.

Interviewer
Okay. Yeah. And if we get back to the data card, I was just thinking about some practitioner who mentioned the metadata repository. I think that that has a lot to do with the data card you just mentioned, right? So to distinguish between your catalog, which is more static, so there can be static metadata, and this can be feeded by dynamic metadata that changes over time. And it's stored in the metadata repository.

Expert I
Yes, so Data Catalog is just a database for metadata.

Interviewer
Yeah, but it's static data, right?

Expert I

No, it's not static. It's a database which means it will be updated over time. Oh, yeah. Yeah. Okay. But the question is how do you update and what's the process of updating it so that you have verified and trustable metadata? Yeah, okay. I get what you mean. So just think of Data Catalog as a database with a pretty UI for browsing metadata. Yeah. But like, how do you manage the content of that metadata is very important. And that's central to the entire interoperability governance of the data mesh. And that data card is just one particular metadata structure template that you can use.

Interviewer
Oh, okay. Yeah, I think I could add that one on top of these because all these options over here, they are not mutually exclusive. So you can choose multiple of these.

Expert I
Yeah, so the one data product would have multiple interfaces. Yeah. And each interface needs to have its own contract to the consumer or in the mesh, so to speak.

Interviewer
Yeah, so we can perhaps deep dive more into the data product itself. So what happens inside the boundaries of the data product? I again observed a data catalog. So I think it's also really important that each data product has a catalog on its own to make it autonomous and that the consumer can check what's inside the data product so he or she can discover.

Expert I
What's your definition of a data catalog?

Interviewer
A data catalog in this way, it's more like a local data catalog. So this one only contains metadata of the data product itself. So not metadata of the other data products in the catalog. You know what I mean?

Expert I
So is your definition about the metadata or about the catalog, which is the capability to store, process and update metadata? So are you saying each data product needs to have the capability on their own independently from everybody else to store and process metadata? Or does each data product need metadata?

Interviewer
I think the first one. Yeah, to make it understandable for someone who wants to consume the data product.

Expert I
So when I want to understand a data product, am I looking at the metadata or am I looking at whether you have a data product?

Interviewer
I think I would look at the metadata to make sure what's inside, to make sure what kind of transformations are happening inside, etc.

Expert I
So it's a content that matters, right? Yeah. It's not whatever's holding the content, let's say.

Yeah. And the metadata could be, I don't know, a document or a Confluence page.

Interviewer
Yeah, okay. Sometimes I think I need to work on my definition of certain components. Yeah, because besides the data catalog, which is really focused on the discoverability port, there also needs to be some kind of observation plane, which is in close contact with the observation port, of course. And this really represents the data quality. So what kind of quality metrics, what is the data quality of the data sets in our data product, etc. The control plane is more for the control port, so there is some kind of governance team, which is keeping track of all the data products and they can control an individual data product by using this control port. Data onboarding is really about ingesting and all the transformations that are being applied to the data set within the data product. There can be internal storages to maintain one single source of truth. We can have a change data capture. When something changes in our internal storage, this immediately goes into the change data capture and an immutable change audit log to make sure that changes that happen within the mesh are also stored in here in some kind of S3 bucket or whatever. To make sure that it's immutable, so we can use some kind of append only mechanism where each change event is being appended. Yeah, that's all the options.

Expert I
That's nice if you can do all of that, but the reality is not always so clean.

Interviewer
What do you mean?

Expert I
You said once that data mesh doesn't exist on its own. There's something that can be followed. If you come along and say, all right, well, whatever you have, you have to turn that into an append only system. I need a CDC feed of every database that you have and you also need to run a data catalog. Do you think whoever owns, as a data owner, why should I do any of that?

Interviewer
Yeah, that's true. Yeah, I know what you mean. Not all of these are mandatory, let's say. Right. So how do we find a definition of a data product that allows different teams to easily implement and potentially make compatible with the existing paradigm?

Expert I
And I think once again, it comes down to definition and in this very case, identifying the essential aspect of a data product. These are to me a bit superfluous. Whether or not I have them or not, I don't care. The data products won't be fine by these kind of technical patterns. And I think, yeah, let's say what are the outcomes? On the other side, you're saying here are some of the outcomes I need, not everything on the right side, but in the third column, so to speak, on the left. Versioning, alerting, single source of truth, building a business-centric data mart. I would avoid using data mart, but the outcome is good. So I think if you define the outcome, so what are the key properties of the data product that need to exist? What are the different outcomes that are essential for an interoperable and governable data mesh? If you can define those, then you can define, okay, well, where are the minimum capabilities that are needed to meet those outcomes? Does that make sense? Like you're kind of starting from, if I do this, then I will get that. And I think that's the wrong direction. You need to start by saying, what are the minimum outcomes I need and then what are the minimum things, technical capabilities I need on the left side?

Interviewer
Yeah, so you mean I should start with the business logic instead of thinking about types already?

Expert I
Not the business logic, but the properties of the data mesh are required for a data mesh to, sorry, the minimum properties of a data product that's required for a data mesh to function and to scale in a way that is interoperable, secure. I don't know all the different ideas that Zhamak has. And you have some of these here, right? Actually, in addition to what Zhamak has, like versioning, like alerting, single source of truth, that's great. You can add those. But I guess the question is, like, what are the minimum requirements? And then there are multiple ways to implement a requirement. There are multiple ways to meet a requirement. If you say, for example, I need alerting, I can tell you 10 other ways that there's nothing to do with immutable change on the block. Yeah. Yeah, OK. You get what I mean? Yeah. And the whole point of a data mesh is the technical pattern, how I meet this outcome in a data product is the decision that's made by the product team. It's not made by anybody else. No. So they decide how best to meet that outcome. You agree as a mesh, every product needs to meet these outcomes. And they need to have these characteristics or maybe from a technical perspective, they need to meet these API specifications or have these minimum metadata template that allows the platform and the governance team to drive automation based on that. But how specifically they populate that metadata, for example, is up to them. You can't dictate that kind of thing in a data mesh where the goal is autonomy and choice of technology.

Interviewer
Yeah, that's a very interesting view on it. So not just thinking about all the possible options, but also take the minimum requirements into account.

Expert I
Just focus on the essential, right? Otherwise you're writing another book.

Interviewer
Yeah, indeed. Because there are so many patterns out there that this is just the top of the iceberg, I think.

Expert I
Right.

Interviewer
But yeah, the goal of my thesis is just to provide some kind of guideline and to provide some kind of overview which patterns are out there, which you can use. There's not a one way solution.

Expert I
So I guess the question I have to you is, why should I do this? Why should I do immutable change-up? And then why does that matter? And if you can answer those two questions, then great, include it. But otherwise, leave it out.

Interviewer
OK, that's a good one. Just, yeah, the immutable change audit log is, for example, very important in the case of event streaming, but it's not important when you do batch processing,

for example. Well, then it's not as important as in event streaming.

Expert I
Yeah, so if the data mesh needs event streaming, then, you know, Confluent will be very happy.

Interviewer
Yeah, no, it's not mandatory. Definitely. Yeah, definitely. It's not mandatory. Right. It's just an option. Yeah, thank you. That's a perspective I haven't taken before. So I will deep dive into that one. Just broaden my scope a bit. Just take a more general approach. OK, I see we have only seven minutes left, so I will quickly go over these last two. I think we covered this one pretty much already. This is really about the interface. So, yeah, of course, there are two ports missing, which are the two most obvious ones, the input port and the output port. But besides that, I think there can be five different kinds of ports, considering the data product. Do you think there can be more or don't you agree with some of these?

Expert I
So these are Zhamak's definition from her book, and I am inclined to agree with them. She's already done a lot of work to boil down all the concepts.

Interviewer
Actually, she did the observation port and the control port. She merged them. So what I did here is really separating them. And that's in line with an article from *. Perhaps you know him. He wrote a book about * and he really, he really evangelizes this concept about separating these two ports.

Expert I
I think it makes sense to. It makes sense to separate them. Yeah, because to me, the observation port is for. Basically, data quality monitoring. The control port is for access control management. But it's fundamental. You can use it for other things. And the discovery port is the metadata that gives you information about what the data product is. So I think I'm inclined to agree with that.

Interviewer
OK, great. I think we can quickly go on to the last one. Yeah. Most articles only mentioned the containerization way. I shouldn't use proprietary names here, by the way. This is, of course, the container orchestrator and Docker is containerization. But can there be more options out there instead of using only containerization?

Expert I
What do you store the data? In the container?

Interviewer
No, not in the container, in the storage layer.

Expert I
There you go. So that's a bit that's missing in your diagram.

Interviewer
Is that the only one?

Expert I
So, I mean, think about it. There's every application before data mesh has a storage layer, an application layer and a kind of an interface, right? The contract. So there's nothing different about the data product. There's where you store the data, there's where you process and expose the data. And there's an interface from which everything is consumable from the outside world. And what you're showing here with respect to Kubernetes, Docker functions, it's just the application layer. Yeah. Yeah. And then the data layer, you can even break that down into the infrastructure layer as you have here and the database itself. So infrastructure could include, I don't know, like in a Google Cloud context, like a project or an AWS account. Everything lives in an account, for example, with the billing ID. The database and exists on its own. And then the database lives on top of that. And then you have an application that are consuming the data and that application is exposed to data in some way. And there are some databases that will claim to do everything. And there are some databases that will claim to do only one layer that is not actually the data storage layer. For example, what is it?

Interviewer
Snowflake?

Expert I
Snowflake pretends to do everything. Like Starburst is Presto or Managed Presto or Trino. They only do the query layer. They say you can store your data where you want and whatever systems and they just do the query layer on top. So that's, you know, like I'm just showing you like all the different layers that could access and different kind of technology. You know, vendors essentially are trying to pitch different areas. There's also, you know, the whole, what is it, knowledge graph aspect. How do you automatically extract metadata presented as a graph and use it through automation, etc. There's a bunch of data fabric is the topic that comes up. Like what data fabric is data mesh. There are vendors that will try to push a data fabric view of the work. Yeah, so different vendors would push different things. So, yeah, I think you're asking me. I'm not here to represent a vendor. I'm not going to work with this. But yeah, I think that's so my feedback is like there are many different layers. I forgot what those layers are and what purpose they play in a data mesh and you know, just read the different pitches from the vendor. But you don't believe any one of them. Always be, always be, how do I say it? They're all biased.

Interviewer
Always be biased until the proof is out there. Okay, we can go back to this one quickly. So the inter decisions here. I don't think it's complete yet. Do you think this can be, well, if the feedback has been applied. Do you think this can be useful for the user itself when he or she wants to migrate or do some greenfield development in related to data mesh?

Expert I
I think you have to start with what they're trying to do. Who are the user or what they're trying to do. And, you know, is this useful? Maybe for some user, for some particular types of questions, but not for others. Yeah. So I think you need to define the scope of what this exercise is about. Like what is the scope? What are the questions that you're really trying to answer? Who are the users and what are the problems that you're looking to provide a structure to get to an answer for? And I think you already have a structure in mind. I'm not going to pull you away from that. But I just wanted to say that this is just one particular type of question that is not, you know, in my experience, I'm exhausted of all the questions I work with my customers to help them with.

Interviewer
OK. Yeah. Thank you, *. That was really helpful. This session gave me some different kind of perspective on certain decisions. I really need to think more about the user as well and about the definition of data product. Yeah. I think my next framework will be about the self-serve platform and I will, of course, implement the feedback on this one as well. Do you think we can do another session in two months about the self-serve platform that we can have a session on that, a new one? If you have time, of course, because I really enjoyed the session. Yeah. OK. I will. I will send you a new email by then. And if you have time, that would be amazing. But I already enjoyed this session. So thank you for that. Have a great week. Best of luck with your session. Thanks. I will send you the transcript, by the way, to make sure everything is OK. Sure.