B.5 Participant 5

Interviewer
Okay, great. It's recording right now. Okay, I will first give you a short introduction about my background. That's useful to know. Okay, I'm Tom. I'm Dutch. I had a bachelor in industrial engineering, so I came from a more business background. And now I'm doing the master data science and entrepreneurship, where we try to combine the business aspects and the technical aspects as well. So that's also why I like data mesh so much, because I think there are a lot of similarities between business and the technical side. Yeah, that's what I recognize in that data mesh a lot. And now I'm doing some kind of internship at Bain and Company. And we're really deep diving into the architectural components of data mesh. And now we're trying to build our first framework on the data as a product principle. And after that, there will be two more. So the second one will be about the self-serve platform. And the third one about the federated governance. So yeah, that's a short introduction about me. Yes, you can provide me a short introduction.

Participant 5
I can introduce myself also. So I'm *, I've been yeah, I work from *. I've been working in the data space since *, started as a consultant and building data warehouses for small and medium organizations here around Barcelona and the east of Spain. Then I moved to an organization that's called Shipstead, which is a non-line classified groups where I did a bit of the same. So we had a lot of brands and I help them building their own local data warehouse until we realized that we also wanted some data at global level to be able to compare the different brands in the different parts of the world. So we started some global initiatives on that. And then we suffered a bit the pains of distributed ownership and then the different like multiple sources of truth, like different definitions at the local level and global level. And this is where we started thinking on a different approach than just centralizing everything. And we started moving for this global or central data warehouse. We started moving into some kind of hybrid approach, hybrid approach, where some things were kept local, some things were kept global. And we built a lot of documentation and FAQs and examples on how to use the data. So we also embraced the data as a product principle somehow. And then the article from Zhamak Deghani came out and realized that we were doing a lot of these things. And we were suffering from the pains that she was writing about. And it's like, okay, yes, with one of those companies that she's signaled and it's nice to have some words to explain what is happening in the organization, right? Because up until the point we didn't use data as a product or data mesh or whatever.

Interviewer
Yeah. Oh, well, so you were some kind of early adopter of data mesh, even before it was even data mesh kind of.

Participant 5
Yeah. So that's we always say that we took a bit of a hybrid approach in the sense that we don't follow or we have not followed data mesh like in the pure explanation of data mesh, we just did what worked for us at that time. And some of the things that worked were part of the data governance data as a product and distributed ownership. But we also kept a lot of things or like the most important things or data products maintained by a central team, for example. And then, well, I, the company changed the name to *. I kept continuing in the same role and basically exporting the same challenge or having the same challenges and exporting solutions for that. And yeah, it, like an opportunity came to change jobs. So I accepted it and I'm now working at *, which is an online supermarket, like a pure online supermarket that we are, we

are present in Norway, Finland. And we have launched in Germany this month or we are going to be public in Germany in February in Berlin. So yeah, I changed industry, but we are having the same challenges. So I'm working a lot in the same areas of data as a product, distributed ownership and such. And I have in the past three years, I have also gained some experience on experimentation, so A/B testing. And so that's what I'm doing today at *. Okay, great.

Interviewer
So you're still doing something in Data Mesh because you implement Data Mesh in *, but now you're doing the same at *

Participant 5
Yeah, it's, it's not my primary responsibility at this point, but I advise a lot. So we have been working on, there is a medium article for that about the data and insight principles and how we operate in * about like using data and such. And for example, distributed data ownership and shared data governance is one of the principles. Data as a product is another one. So I have been contributing a lot to that. And actually, I think that next in February, I will publish another article on how do we treat data as a product at *. I have the draft, well, it's finished. I need some thumbs up before publishing it. Yeah. But yeah, even if I'm focused on experimentation platform, I see as a strong contributor on the data as a product and data measure principles. And I'm helping on that. I'm not managing the data platform team. Let's say.

Interviewer
All right, interesting. Okay. So I'm really curious about your opinion on my frameworks where we deep dive more in the data as a product principle. So I will quickly share my screen and I provide you with a short explanation. So I have, yeah, well, I have created seven different frameworks. Actually, there are six different frameworks. But I identified six decisions related to your data as a product journey. The first decision should be what type of data products can be developed. And all these decisions are in the context of architectural design decisions. So it's really about the architectural framework behind the data as product so not about the business logic. So that's why I identified the data product type here as first decision. After that, it's more like, do you want to migrate towards data mesh from your legacy architecture? Or do you want to do greenfield development? So starting from scratch. At the same time, we can make a decision on the infrastructure. So how does our data product communicate with other data products, with the self-serve platform, the management layers of the governance and the consumers. Then we can deep dive more into the data product anatomy. So what's inside the data products, what kind of components do we encounter? The data product interface slash contract is more about the perimeter that's around the data product. And what kinds, what kind of ports do we encounter? And the last decision is about, yeah, how do we deploy the data product? And we can perhaps now start with the first decision. And at the end, we can go back to this inter-decision framework and check if there are some decisions made decisions missing or there are some wrong decisions over here. So let's quickly start with the first one. It looks a bit like a mess, but actually there are three options over here. So I identified the data product as raw data. So you can expose raw data. Some practitioners do not agree with this because raw data can be really messy and unstructured, etc. The second option is to expose your data product as an algorithm. And the third one is expose data product as derived data. And there can be some variance on the data product as an algorithm. So we can look more in BI tools, so optimization based decision support system, or more like the machine learning. So expose data product as an AI / machine learning model. And on top of that, we can do some kind of hybrid product, create some kind of hybrid product. And what I mean with this

is that we have, for example, two output ports. One output port is throwing raw data into the mesh. And the other one is throwing derived data into the mesh. So it's like the best of both worlds. And the last one is a composite product. And this is more like merging data sets. Yeah, it's like merging the raw data with the derived data, for example. So this is what I have identified. And according to gray literature, and gray literature is actually everything that hasn't been officially published, like an academic journal is officially published, a medium post and YouTube videos and all those kind of stuff are not officially published. So this is really from the practitioner view. Really curious what you know,

Participant 5
This is great. I have some ideas to improve it based on practice. But I think that it's going in the right direction. I think that if you go to the previous diagram on like, I think that maybe I would add here one step before all of this, it's why are we building the data product, right? So for example, like, what is the business need that we are solving? Or like, I don't know, like, what is the question that we want to answer? What is the problem that we want to solve? And because here we go first with, okay, it seems that building the data product is already like a decision that we have made, right? And sometimes we enter into implementation mode, but we don't think about, okay, is there a something already available? Do I really need to build this, right? I don't know if it's taking into consideration in some other, like before jumping into this framework, but I think that it's a good thing to keep in mind. How do you want to represent that. Like deciding on what is the product type, but also, for example, you could have one of these sub diagrams, like, okay, what is the problem that we are trying to have, that to try and to solve? And then you can choose, okay, can I solve it with an already existing data product? And then I need something to discover data products in the organization? Or I need to build a new one, for example. And then if I need to build a new one, then I will go into choosing what type of data product can be developed, for example. Okay. Maybe it can be more complex, these previous diagram, but I think it's...

Interviewer
Yeah, I think it's a good starting point, actually, to start with the business problem. Yeah, exactly.

Participant 5
And here, I don't know if you have considered dashboards as a type of data product?

Interviewer
Yeah, it's more like the variant of an algorithm here. So more, the optimization-based decision support system is more like a BI tool, so a dashboard.

Participant 5
Yeah. I wouldn't consider... It would not be obvious to me that if I want to build a dashboard, I need to choose to expose the data product as an algorithm. I think that you have the different options here. Maybe it's a matter of reordering things. Yeah. What I've used to work with when regarding data products is... Okay, I need to create a data product. And for us, the options we were choosing at Adevinta was low-level aggregations, aggregations, a dashboard, or a machine learning model. So there were these four. And... Okay. Yeah. And for example, in Oda, we don't consider raw data as a data product, in the sense that it's very dirty and we really don't want people to... I mean, we offer raw data as a data product, but it has been enriched, it has been cleaned, it has been filtered. So let's say...

Interviewer
For these source align domain, you know? So a previous step for the consumer align domain before it's getting true.

Participant 5
So if that's the case, then raw data can be a data product. And then with distinction between two different types of data products in the derive data, it can be low-level aggregations. For example, a user's table where you store a lot of signals about your users, like have they entered the website during the past week, during the past three months, during analysis? And then you have a data product type, which is aggregations, and that are, okay, how many users did we have yesterday on this type of things? I think that... Yeah. That would be data as a product, right? And then for algorithms and dashboards, sometimes you need this data as a product to be available. So you might need to... You may want to expose your data product as an algorithm, but you don't have the data first to do that, right? So sometimes you need to expose the data product as raw data, clean and print, etc. And then build another data product, which is the algorithm on top of that data, on top of that raw data. So this is my experience on that. I don't know how I would put this into a diagram, but yeah, that's been my experience.

Interviewer
Yeah, I can perhaps rephrase this one over here, because optimization-based decision support system, yeah, it's not immediately clear that this means a BI tool or a dashboard, etc. Yeah. Which is a more familiar term for most people.

Participant 5
Yeah. I think... I think yeah.

Interviewer
Yeah, okay, great. Then we can perhaps have a look at the next step. So do we want to migrate or do we want to greenfield... Yeah, do we want to start from scratch? This one is mostly based on the articles from *. Perhaps you know him, *. He liked these articles on the medium. And yeah, he identified four migration patterns. So first, you have master data management, where you create some kind of master data management data mesh. For example, in Snowflake or another master data management tool. Strangler-Fig is really about decomposing your current monolithic architecture. And at the same time, slowly building up your data mesh. So it's about taking away one service from your monolithic architecture and converting this service to a data product. Do you know what I mean? Or...

Participant 5
I haven't read the article, but it sounds very quickly slow.

Interviewer
Let me show you. It's really this one. So you have a monolithic architecture, you take away two services and you start building up your data mesh slowly. So this is really monolithic. The zero trust architecture. So you make sure that each data product is autonomous and has their own access controls. And CQRS is about segregating your right and your read function to make your data product stateless. In greenfield development, yeah, that's just starting from scratch and provides you with a lot of flexibility, of course.

Participant 5
And this is the decision that you do every time that you build a data product or is an architectural decision that you do once and then...

Interviewer

I think you would do it once at the beginning of your data mesh journey.

Participant 5
Exactly. So that's probably if that's the case, then I wouldn't put it here, for example. So this is something that goes... It's a decision that you need to make before starting the first data product, for example. Right. So maybe it's... I think it makes a lot of sense and you need to define the strategy. But it maybe complicates... For data product developers, it complicates their journey and it's not something they need to care about. But maybe it's relevant for the data platform, folks and architecture, architects and... Yeah.

Interviewer
Sure thing.

Participant 5
Because if I am a data product developer, if we are already in the Strangler-Fig kind of pattern, I don't have any option. I need to continue on that. Yeah.

Interviewer
Yeah. Okay. So these are not mutually exclusive. So you can do the strangler-free and the master data management at the same time. So it's not really like a one-go option. You can choose multiple. Perhaps I should change that as well. But do you recognize some of these patterns when you started migrating to our state mesh or you did some greenfield development?

Participant 5
Yeah. I think that we did unconsciously think about those patterns. What happened in our case, for example at Adevinta and Shipstead, is that there was a central team that was doing everything. It was not a monolith from the architectural point of view. Well, at the beginning it was a monolith. So all the pipelines there were in the same repository. But then as a central team, we decided to split the pipelines into different repositories. So we broke the monolith, but we were a central team. And then what happened is that we started to distribute ownership of those smaller repositories to other teams. Right? So yeah. I think that it's a bit a Strangler-Fig, but we didn't put a goal into it because we broke the monolith and we didn't distribute the ownership. And then when we distributed the ownership, we didn't distribute the ownership of everything. And we didn't have that very clear from the very beginning. So it's kind of a child or adapting. Sometimes what happened is that, okay, this marketing data products makes a lot of sense for the marketing department to own this, but they don't have anyone to own this. Right? So we keep we keep them in the central team. So yeah, this kind of things happened in our journey. So I don't know how this could be represented or like if you choose an end state. I think that definitely I wouldn't put this in when creating a data product. It just needs to be more on your data strategy, maybe. Like how do you say, have a strategy around this. Probably.

Interviewer
And do you perhaps think there are some patterns missing that I haven't looked into?

Participant 5
Well, maybe we could. Yeah, the things that I was commenting, right? So approach is chosen for the creation of data product.

Interviewer
So yeah, perhaps I should rephrase that as well.

Participant 5
No, that was, yeah. Yeah, I think that it's a bit difficult to relate to these. Because on master data management, for example, I understand that some data products in your organization will be master data. That's one approach, but then you will have master data, but then you will have also non master data data products. Right? So any of you have master data data products, you can build a Strangler-Fig kind of pattern where you prioritize extracting from the monolith the master data products. I think that the approach to be chosen as, yeah, depends on the prioritization also on.

Interviewer
Yeah, perhaps you can have a look at the infrastructure because it also explains some elements of, for example, the zero trust, etc. Yeah, this is more extensive diagram. So I will slowly explain this one. This is really about the infrastructure. So how do we communicate with other data products, etc. I identified a schema registry and a schema registry actually converts all the incoming events from the event streaming backbone and essential data product catalog in a more user friendly format. So that user is able to see what's going on in the mesh. And the central data product catalog is keeping track of all the data lineage. It's like one stop shop for everything that's going on in the data mesh. The event streaming backbone is something like Kafka. So it's really about event streaming real time, yeah, real time event streaming. There is a, yeah, we can choose for shared storage. But I heard from other practitioners that there can be some versioning problems when you choose for shared storage. So for example, you have two data products. And one of them is getting versioned. And the other one has to wait for the first one to be versioned because they somehow depends on each other due to the shared storage. And this makes them less autonomous. So there are some, of course, some pros and cons considering this component. Some pros are that you don't need as many storage accounts as you would have when you choose for an internal storage for each data product. So there are some pros and cons for this option. API invocation is about all the possible APIs we can implement. So REST APIs, GraphQL, gRPC. And then we can also, yeah, this is not really a kind of API, but it's more like, for example, a variation of the REST API. So Cloud Storage API, Storage Read API, SQL access point. And yeah, on the bottom, we see more non-functional requirements. So for example, security controls in memory catch. And what I mean with a query catalog, a query catalog is more like a sample notebook or some kind of query guideline where the data analyst can check which queries are possible in your data product. So it's more like some kind of manual. Yeah, so that's what I've identified for the infrast, yeah, it's more like an infrastructure layer. It's what's going on outside the data products.

Participant 5
Yeah, I think it's a good list to do not forget anything because we didn't have this very well specified. And like, for example, when for the data product developers, I think that this helps a lot. So I agree on the schema registry, the like the entry on the data product catalog on the event is streaming. Just a comment that not all events will be. So not all data will be flowing via an event. So stream, this is a batch data. Yeah, so you have a batch data that you can import via... And these types of things, like for example, from the backend systems or like the databases of your product, you will need some more like a blob storage. Yeah. So that could be another. Okay, source or like an option, it's a batch or event, or like, it's more like, it's kind of a pull or push, right? So, in an event stream, things are pushed to you. When you are retrieving data from database, you are pulling from that. Then you have the storage. And you can, yeah, I think that some of these arrows should be optional. Sometimes you don't need an API or you don't implement an API. Yeah, but I agree on the SQL access point and the policies. And

here, I don't know if it's included in the schema registry, but I would be explicit on the data model also. Somehow in the sense, you might have an schema when you read the data, like the source data schema, and then you have kind of a target data model or target schema. Maybe putting something like target schema or target data model. It helps when thinking about the data model that you choose like for the data product, because this is important. How do you name columns or attributes? It's important.

Interviewer
Yeah, to standardize. Yeah, to standardize. Okay, that's a great point.
Participant 5
And then for the data product policy enforcement, security controls, like access controls, yes. One thing here to be taken into consideration also is retention policies. So how long can we store the data, for example? We have in Europe, we have some strong regulations on that. And you can also choose the purpose of this data product. And yeah, and this will, like, for example, if you store, or maybe this is an option that you can add here, like if you store personal information or like personal identifiable information, then you need to have these retention policies. You need to have takeouts and deletions on this data set and have a strategy on how you are going to implement that apart from access control and on this.

Interviewer
Yeah, okay. Does that also relate with the central data product catalog? Because it sounds a bit like metadata.

Participant 5
The central... Yeah, it's metadata, but as a developer, you need to do something also, right? So I don't think that you should expect the central data product catalog to solve that for you. Maybe the data catalog will have an option to say for how long are we storing this data? But the data catalog probably will not implement the deletion of the old data, or like you will need to do some development. The data platform team will need to do some development.

Interviewer
Okay. Okay. That sounds fair. Yeah, I think that's all we can discuss about the infrastructure layer. I don't know if we are missing something, but I think we covered almost all of it. So great then we can look more inside the data product. And yeah, for example, there is a central data product catalog and also a data catalog for each data product. So it's more like a meta-store registry. On top of that, we have the change data capture, who's keeping track of all the changing events in society data products. And this one is pushing it to Kafka, to the event streaming backbone. We can have an immutable change audit log, so who is accessing the data products, keeping track of that, versioning, alerting, things like that. So it's really about using some kind of persistent log. Internal storages. So this is really important to maintain a single source of truth and to make your data product autonomous. Furthermore, we have some kind of observation plane and a control plane. An observation plane is for observing the data quality, keeping track of the data lineage, seeing all the dynamic metadata, etc. It's really about observing what's going on in the data product. The control plane is more for your governance team to make sure that all the policies are enforced. And the data onboarding is related to data ingestion, transformations, everything that's happening to the data itself. So yeah, that's really that.

Participant 5
Yeah, I think here architectural element. I think it's good. I think some things could be in the previous diagram also, depending on how you look at it. So like, for me, like this one on the

previous one, it feels like a list of things that you should take in mind when implementing a data product, right? And you would choose like, there are some mandatory ones or like some, and some others that are optional maybe or not. But it feels a bit part of the same list. If we need to split it, yeah, I don't know how now. I don't know how they would split that. Because the data catalog, for example, implementing a data catalog component, this one for me, it feels duplicated with previous, not like a registering data set, or at least I understood that the central data product catalog was referring to this part here also. So there might be some duplications, there might be some room for consolidating into a single list. And then my last observation is that sometimes you enter into a lot of detail and some others you don't like observation plane, control plane, and these things are very data mesh specific, and they are not explained. And in some other boxes, you actually explain much more. So I don't know what is the right level of explanation, but maybe keep it consistent across the different diagrams. So for example, instead of putting observation plane, if you at one sentence there, using the words that you used to explain it to me, maybe that would be more helpful.

Interviewer
Yeah, I should still write the context regarding each framework. So I can do that in the text or perhaps, yeah, or some kind of architectural components here.

Participant 5
Yeah, or you have a specific diagrams on each. Like, for example, sometimes the audit and data access that you mentioned in the previous one, like data access control, like row level, and this could be in the control plane or the observation plane, I think. But also in the previous diagram, you had that very, very detailed. I don't know if you need that amount of detail in that diagram compared to these others. So I don't know, I think that in general need to find the good balance of detailing in the diagram versus detailing it in some other place.

Interviewer
Yeah, I have also one question about this one because, for example, before *, he wrote a book about *. And when I interviewed him, he was really missing the meta store. So he would separate the data catalog and a meta store to make sure that your meta data is stored somewhere else. What do you think about it? Would separate meta data from the data catalog, because he feels like it's more like a universal thing and not really focusing on metadata management?

Participant 5
Yeah, I'm not sure. I think that well, there is this open metadata framework now that what you can store metadata in kind of an open shared format. And then if I understand that line of thought, the data catalog would consume some of this metadata and expose it in a UI so that people can search and browse the metadata of the data product. It's more like centralized approach as well because he's really like, yeah. I'm not sure. I think that I would be comfortable just having data catalog, but I guess that it makes some sense to separate also, like putting metadata and storing metadata next to the data product. And then the data catalog consumes some of this metadata, for example. Okay. One of the things we considered metadata, for example, is in our case, if there was a validation error, so for example, if you have the target schema, and you have a validator that for every single event validates if this event matches the schema. And if it doesn't, we attach the error as metadata in the event itself. So this is metadata that will not go into the data catalog, for example. So maybe on that sense, it makes sense to, on that regard, it makes sense to separate.

Interviewer

Also thinking about the difference between static metadata and dynamic metadata. So for example, you could store your static metadata in your data catalog and that barely changes. It almost never changes. And your dynamic metadata, you could store it in a meta store, which is really focused on keeping track of this. Yeah, keep track of these changes. I don't know if that would make sense.

Participant 5
Yeah, I agree. Yes, that makes that's another difference on, yeah. Yeah, so the difference between static and meta, yeah, static metadata and dynamic metadata. Okay.

Interviewer
Okay, great. I can also deep dive into that more. When I'm reconstructing all the frameworks, I made this all in Python, by the way. So everything can be changed really easily. Instead of making thousands of draw.io diagrams. Yes. Diagrams, that's code. Like, yeah, it's code. Yeah. Okay, let's look at the interface decision. So what kind of ports do we need? Well, I identified three, but some practitioners mentioned that I should include two more. So the input and the output port as well, which are the most obvious ones. But besides that, I only think that there can be five ports. So the input port, output port, a discovery port, which is connected to the data catalog, and the control port, which is connected to the control plane. And, yeah, the governor's team has control over this one. And the observation port, which provides some kind of sneak preview for the data analyst or data scientist about what's inside the data product. It's about monitoring. Yeah.

Participant 5
Yeah. But then you missed the input and output or, yeah, I think it's also a great suggestion. Like the output would be how do you use the data product, right? Yeah. Yeah. Yeah. Yeah. I agree. Yes. I mean, depending on the type of data product, you will have a different place to enter. Like, if you build an algorithm, it's different than if you build a table or a dashboard, right? But, yeah, like should be, yeah. It's like a generalization. Okay.

Interviewer
Okay. Great. Yeah. I think this one is pretty complete, to be honest. Yeah. I have some doubts about the deploy decision because I only encountered two options and both have to do with containers. So the first one is about containerization. I should change the names because they are really proprietary. Yeah. And so I identified a containerization and a container orchestrator. But I think there should be a few more options to deploy your data product. Yeah. If you deep dive a bit more into the Docker. So, yeah, using a single container design for each data product, you can use a templated data pipeline, which helps you to standardize everything in your data mesh and, yeah, quickly to CI/CD.

Participant 5
I think it's here. Yeah. It depends on the technology that you use in your organization also because you can build data products with Oracle also, right? Yeah. So which doesn't use Kubernetes or, yeah. So this, I think that this dashboard or this diagram can be very extensive. Yeah. Yeah. And it's also where data mesh has not entered into, right? Right. Because it has been kind of more of a, not technical, right? And now Zhamak is launching his, her company on how to deploy a data product easily. And maybe that should be included here also, right? But yeah. I don't know if I would include that diagram, to be honest, because it can be so extensive that, you know, Yeah. Okay. But if you don't provide more examples on this, like people that looking at this on the implementation side will look, okay, but I want a template

or I want, like, how do you do it? Yeah. Yeah.

Interviewer
And they would think that containerization is the only option, because yeah, Kubernetes and Docker, they both have to do with containers. And I think function as a servers, that one could be a third option. So I think I miss this misinterpreted here. Kubernetes is not, of course, a function as a service. So perhaps I can identify that as a third option. Do you agree with, but it will be very specific to machine learning or? Yeah. Yeah. For example.

Participant 5
I think that how to deploy a data product, this is, you should probably split that one second. All right. Because when deploying, you need to choose several things, right? So where is the data product going to be stored? Like Snowflake or like the database system? What is the, where is where the code is going to run? Right. So which is what you refer here now? Yeah. What kind of CI, CD process are you going to have? Like for continuous integration and continuous delivery? I think that the deployment part has several parts that, like, even if you use Kubernetes, you still need to choose, like, you can use Kubernetes on top of a Snowflake or you can choose, like, you can just Kubernetes on together with Snowflake or with Dredshift or with BigQuery, blah, blah, blah. Right. So yeah. I don't know. This can be, or very big and very open or like a completely different project to focus on all the many different options.

Interviewer
Yeah. Okay. Yeah. I think this diagram needs the most reconstructuring of all the other diagrams. So yeah, I will definitely deep dive in all the articles again to find more deployment options. Okay. But yeah, those were all these six diagrams. So if you look at the inter-decision framework, yeah, we already mentioned that the, yeah, for example, choosing between migration and green field development can be placed further upstream.

Participant 5
Yeah. Or at least like a first, yeah, on the Y, right? So why are we doing this data product and, yeah, perhaps it could be good. Yeah.

Interviewer
Do you think I'm missing some decisions over here or would you skip some decisions?

Participant 5
Yeah. For example, you would perhaps skip the deployment decision, but at least for an initial, like first iteration, I would skip it. Yeah. Because that's completely, maybe it's not a, yeah, it's a different type of decision, right? And how do you deploy it would be given by you from how the organization deploys things probably. And it's a different type of journey on decision and very complex.

Interviewer
Yeah, sure thing. But do you think there are some decisions missing that you encounter during your data mesh implementation, well, to be more specific during your data as a product implementation?

Participant 5
I think maybe speaking some about the domain here, like, I think that one challenge we have encountered is about the ownership of the data product, right? So who will be owning this

data product? And what are the rules you will choose to determine ownership of this data product? Because it might be very clear when you build it, but then after six months or one year, the organization changes, and then you don't know who is owning that anymore. So kind of, yeah, I think that thinking about explicitly saying about who owns the data product, like with, and which domain this data product belongs to, and then ownership can be tied to the domain, for example, that would be one. And then also, when is this data product going to be deleted? Because we don't need it anymore. Right? So it could be that, yeah, one thing that we are encountering is that we build, build, build, but then we don't delete, right? But defining some rules on when this data product can be deleted, I think that's also something to be considered in the, yeah, in one of these, like the second, third, fourth, yeah, maybe like choosing some policies around the life of the data product, and being explicit in the creation at creation time, when this data product can be deleted. Like, for example, when nobody is querying it, or like, if nobody has queried in the last 90 days, or something like this, I think that sometimes people do not think about this when creating the data product.

Interviewer
Yeah, but I'm also making it another framework for the federated governance principle. Do you think a retention policy you just mentioned should fit in that diagram, or can we already consider that option in the data product diagram?

Participant 5
I think that the shared governance needs to tell you that you need governance on that aspect, right? But as a developer, when creating the data product, I need to, like the governance program needs to give me a checklist that I need to use in order to implement my data product, like choosing the type, yeah, access control, retention policies, which is storage, the registry, etc., right? So, building the checklist is kind of on the governance side. Follow the checklist for me would go in the data product implementation.

Interviewer
Yeah, okay, great. Yeah, I think that's, oh, I have only one question left that's more about the usability and difficulty of this, yeah, all these diagrams. Do you think a practitioner can use this as a guideline to implement it? And how difficult do you think this diagram is constructed? Is it understandable for the user?

Participant 5
I think that it's understandable. But it's a bit, I would say it's not the easiest thing to understand. But if I have to, I think I would prefer some, like, the idea is super great. Maybe it can be made a bit easier. It will be more used or more effective. And maybe you could have the same questions, right? But in a different, like, I think that once you settle with the flow, let's say, like, migrating this into some other tool or something that it makes it a bit more appealing for the user to use. I think that would be nice. But I think that once you enter, it's easy to understand. Something could be simplified, maybe. But also it's about the experience of the user when starting entering into this.

Interviewer
Yeah, I try to do some generic terms because I don't want to be too technology specific, you know? So for example, for master data management, there are tons of options. But master data management is more the generic term instead of a proprietary term.

Participant 5

Yeah, I think that's great. Yeah, it's in the right direction.

Interviewer
Okay, thank you so much for this discussion. I think I learned a lot from it and I will transcribe it and share the transcript to make sure everything correctly stated. Yeah, thank you so much. And I hope you have a wonderful day.

Participant 5
Yeah, you too. Let me know about the progress and updates. I would like to follow that work. Yeah, sure.

Interviewer
I will keep you up to date.

Participant 5
Thanks.

Interviewer
Thank you, Xavier. Bye. See you.

Participant 5
Bye bye.