Appendix A

Interview Transcript Expert B

Interviewer
OK. I will first share a short introduction about myself. I'm Tom, 24 years old, a student at the Master Data Science and Entrepreneurship. Before I was in more involved in Industrial Engineering in my bachelors, but I like the technical side more, so I decided to be more involved in the Data Science and to be more to be more specific, I was really interested in the in Data Engineering practices. So now my thesis subject is about in data mesh which is really on the data engineering side. Can you perhaps..

Expert B
Yes. My name **Expert B**. I currently work for * in the role of * for the *. And prior to that role, I worked in a highly complex environment at * where I was responsible in the * for the whole data management end to end architecture together with a small team. During that period I also wrote a book * and now I'm working on the * of that same book. I give an awful lot of keynote speeches and workshops at large customers, and before * I worked for 10 years as a consultant for * and *.

Interviewer
Great. Thank you **Expert B** for the introduction and thank you for participating here in this interview. Well, as you can see over here, I designed some Frameworks, six to be more specific and I encoded them so everything is created in Python. And the reason I did this, is because I expect some iterations before it can be considered "perfect" and when I need to make a new draw.io model every time it's would be really time consuming. So for this reason I encoded everything in Python. Yeah I made some kind of generator in Python and this generates this model. Yeah first of all we see the Inter decision framework over here. So this is more the high level concept of all the decisions I saw from grey literature and grey literature is actually all literature that is not officially published. So medium posts, YouTube videos, conferences, etc. Everything that's not officially published in an academic journal, etcetera. That's that can be considered. . . .

Expert B
And a book is something you then would consider as something which is officially published. Yeah, that's officially published. OK, so if I give you at the end of the session the link to my latest book then.

Interviewer
Yeah. That would definitely be useful.

Expert B
OK, great. OK, yeah, perfect.

Interviewer
Yeah. So all these decisions are based on the grey literature that I observed during my research. So the first decision was about considering what type of data product you want to develop. After that, there was. Yeah, more like a migration approach. So what kind of approach would you consider for the creation of your data product. We could do Greenfield Development or more like migrating from your legacy architecture. Furthermore there is the infrastructure around the data product. So how do you want your data product to communicate with other data products and with the self-serve platform, with the management layer and of course the consumers thereafter we have the architectural elements that should be inside the data product. So then we look more inside, yeah in the data product anatomy. There's also a perimeter around the data product where we define the data product interface and contract, so it's more like what kind of ports do we need. Yeah. On the perimeter side of the data product. And yeah, the final step is of course the deployment. How do we deploy the data product, what kind of decisions do we need to make here? What kind of options do we have? Alright, so if we start with the first one, I will zoom in a little bit. Actually there are three options over here. So what type of data product can be developed? We can expose the data products as raw data and I saw in one of your articles that you shouldn't expose your data product as raw data.

Expert B
I feel for ad hoc exploratory use cases but it's in a way temporarily because you could not build a sustainable solution on top of raw data, because raw implies it's tightly coupled, linked to the original structure. If the original structure changes because someone decided to change the application and therefore the database structure as well, then yeah, you're broken and the whole solution goes down.

Interviewer
Yeah, I agree. Yeah. The other option is to expose your data product as derived data. So you perform some kind of transformations. Well not really transformations, it's like the top 10 most revenue or the top 10 customers etc. So. Some small calculations hmm. And the last one is to expose your data product as an algorithm and this can be further specified to an optimization based decision support system, so like a BI tool. Or the data product as an AI / Machine Learning model. And we can combine these options over here to deliver some kind of hybrid data product which can expose raw data as well as algorithm for example or compose it product.

Expert B
Do you have a clear definition of what you mean with a data product? I feel this is really important because I talk to many practitioners. And they all use slightly different definitions of what they think is a data product.

Interviewer
So yeah, OK and I, myself, define data product along with the product thinking principles. So a data product needs to have value on its own. It needs to be interoperable so that it can communicate with other data products. It needs to be trustworthy. So keeping one source of truth is really important for your data lineage. It needs to be secure. It needs to be natively accessible, so it needs to be accessible for the data analyst as well as for the data scientist, other data engineers, etc. That's how I would define a data product

Expert B
If you take a step back and look may more conceptual from an enterprise architecture standpoint. There are different viewpoints you could use. So some practitioners, they use more the data management or data viewpoint and they say well, data product, it's really about the data you make available. So it's a data set that has been stored somewhere that has been optimized so you could easily get access to it. It therefore matches many of your requirements or principles you just mentioned. So I think one way to approach it, the other way what Dehghani does in her book, she approaches it more from a technology architecture standpoint, where she says well, it's the combination of data, metadata, infrastructure, and code. To me that's an entirely different, um, viewpoint, because that would also mean you include the infrastructure, so the services for instance, the metadata that describes the actual data itself. Also maybe the code with how the data set has been developed. And I feel it's important to first get clarity on what viewpoint you decide to use, because if you go for more the latter, maybe this whole diagram changes. If you go for the first one, maybe I could add some additional variations of the data as well, so. It's just a thought, maybe.

Interviewer Yeah, yeah, I agree. Definitely. But at the moment do you do you think there is something missing here that we can expose data products differently or what's your perspective?

Expert B
Yeah well but the difficulty I have with the viewpoint of Zhamak Dehghani combines data, metadata, infrastructure and code and in practice I see this leads to really big problems because it requires organizations, I work with,, to manage lots of infrastructure. If you take that fine grained approach and each data set would come with its own infrastructure, so its own storage account and maybe its own spark engine to do the data processing and its own data modeling factory for serving it out as a model. It would mean a lot of infrastructure to manage. So yeah, that's one first problem I see. Another problem I have is she glues metadata and data tightly together. So you see sometimes on the Internet data product architecture and then you see infrastructure data, metadata and code and then the metadata is assumed to be inside the data product. But then what would happen if you take the same semantic data but you duplicate it. For instance, you have consumers sitting somewhere else on the globe, so they would like to get the data more closer to themselves. So then the best practice is to duplicate the data and bring it to their side on the other side of the globe. But then in that viewpoint she uses you would, because you clone, duplicate the whole data product, you would also then duplicate the metadata. This could be problematic especially when you start duplicating lots of data and I would assume in a data-driven world. You are likely to duplicate a lot of data because maybe you would like to the same semantic data, serve it out as a graph and maybe as a document and then a relational database type or as a file, a column source, etcetera. So you end up adding lots of data but each time you copy the same metadata then if the metadata changes because the definition of something changes or the classification of something changes, then you need to update it on many different places. So that would make your architecture again very problematic. It could also mean that that in that particular case, so maybe the ownership could change as well because you. Yeah, copy the metadata. You could potentially end up in a situation where you update the ownership on one side and the ownership on you leave it like this is on the other side. So that would mean the same semantic data would belong to two different owners, and I find that very confusing because it originated somewhere that data was owned by one individual party, so it should not be long long for that reason, too many different data owners, because then somehow others could suddenly claim the ownership of data that they were never accountable or responsible for. Huh. So my viewpoint is a bit more different. So I rather would define a method. I would use a method more

holistically describe with metadata what a data product is, so it's more logical entity to me. And then that same metadata you would link to multiple physical representations or locations. So abstract. Maybe in a data dictionary or data glossary you could describe it, but then it could have different variations or implementations on different locations. And if you take that approach then you end up with also. Data Product architecture. And a more abstract data product and data product data. So when I talk to customers I explain my best practice would be take more holistic approach. So logically you define a data product and then there's data product data. That's more the data viewpoint so how is the data ordered? Is it raw, is it being optimized? Is it already modeled for the purpose of using data on the consuming side? So you should use more that viewpoint and then there could be underlying technology architecture as well. And this is what I call on data product architecture. And the benefit of this I feel is you could use this architecture maybe for building up multiple data products, which again makes your architecture also more efficient. So not. You have to overprovision and create lots of infrastructure with these expensive and hard and difficult to manage. So you could strive for your reusability. And this is yeah my new book and I will again share you the link but this is more the approach I feel is necessary because the way Zhamak Dehghani defines it, I feel it longer term leads to implications she maybe not have thought of. And these other definitions to me, very much focuses only on the representation of the data itself. And then yeah, there are different things you need for managing and establishing all of these principles you just described.

Interviewer
Yeah. Well, interesting. I saw, I saw some similar things in the Gray literature with the Enterprise data catalog. But also some mentioned a data catalog, so that a data product would have a data catalog and there would also be an enterprise data catalog in the infrastructure who's keeping track of all the metadata in the data products, do you know what I mean? So the data product itself has a data catalog and there is some kind of overarching enterprise data catalog who's keeping track of all the data lineage of all data.

Expert B
Yeah, yeah, yeah. I see something similar.

Interviewer
So perhaps we can discuss this framework because this is the central data product catalog I was just talking about. And well, the decision here is what is happening around the data product. So not inside the data product but more like the infrastructure. There can be a schema registry and event streaming backbone like Kafka to enable communication between the data products. And in terms of data lineage to communicate with the schema registry and the Central Data product catalog, we can have a shared storage between the data products. There can be API invocation while there are several options like in like REST APIs, GraphQL, gRPC. Attach SQL access point another expert I interviewed was mentioning the SQL Access point can be integrated with the API invocation. It's actually also some kind of API. I'm not sure because I don't think SQL is an API, so perhaps we can discuss that. And this is like more than non-functional requirements, so the data product policy enforcement mechanism like a query catalog which is more like a sample notebook or some kind of guideline where all the possible queries are defined for the data analyst so he or she can easily access the data product. Security controls with encryption, fine-grained access control, so each data product should have a access control itself. With attribute based access control this is more I have read a lot of articles from * as well and I really liked his perspective on all these different components and he wrote an article about each component separately, so. Yeah, a lot of things he wrote about are also present in my frameworks. But do you think I'm missing something here or do you

think something can be implemented differently? Like do you agree with an event streaming backbone?

Expert B
Um, I think the lineage. Hmm, perhaps misses here. The data lineage, so if you take the left side so how does the data product interact with other data products, self-serve data platform, management layer and consumers. Data will never suddenly become a data product. It always should come first. from somewhere so an application um so how I see it. We started with a blank sheet and we have a business problem and that business problem we would like to solve with technology. And so we give our engineers the task. Well, you should help our business people. Let's develop an application. So what typically then happens is you start listing out what are the requirements and the business terms. So this is the conceptual data model. Preferably you would like to capture it as well. Then that conceptual data model is used as input for a logical design, bit more abstracted, but we now know how these entities relate to each other and this will be input for really the application development. So next we start developing an application. This is the operational application. Chances are quite high that the underlying data structure is highly complex. We normalized it for transactional interaction. So each data field attribute you only see once in the data model. It could be very well that complex business logic sits inside that data model for catering for that analytical workload. At that point we don't have a data product yet. Now imagine situations where someone else would like to get access to that application and instead of giving users direct access to that application we copy the data to another location and make the data available as a data product. So what you need for that is you need to extract the data first from that application we designed. Let it land somewhere, but then still it's raw. Because we copied the original structure, so we need to modify and remodel it in a way it makes sense and users will easily understands its structure and they could see for themselves how to use it best for their own analytical use case. Yeah, if we start with that starting point, what we need then is. Users would like to capture that context as well as remember the business requirements and the entities. This is something what you would expect in a data modeling or in a data catalog as well. So that's I think first part we need to capture ensuring all consumers understands how the application was designed, where the requirements originated from, what these business terms really mean, because with that information they could relate also how the data was created originally. So for that I think we need a catalog or either data modeling tool, you could do both and then also use the data modeling tool as input for the data catalog, so instead of typing it in manually you import it from the data modeling tool, that's also another option. Preferably ideally, I also want to know the translations being made between the conceptual model, logical model, physical data model. Unfortunately not so many practitioners know this anymore these days, so you don't see that a lot. But in highly regulated institutions or within banking we also paid attention to that part and try to capture all of these translations then you would like to see the lineage. So where what data are you? Selected from that application and how did you model transformed it into a data product. Not a part you don't hear much about is historicization if data changes in the transactional application. Often you don't keep track of that, so you see most often the most accurate records only or you see some historicization, but the historicization then is only retained for the purpose of that operational application, so you cannot try to travel back many years. However, on the consuming side you see modeling and reporting these kind of things. Usually, yeah, you need to have tons of historical data for that. So how do we retain all of that historical data then? I feel the data product should be designed for that as well. So each time when we make an extraction, we should add a process where we look already so what data is inside the data product doing comparison and update records, so we get and also a trail of the history inside the data products. I think that's another part we should add to

that on the ingestion side. So then you have input and output. Yeah, I think you mentioned most part. So you could do real time ingestion via APIs via streaming, you could do batches. We need to transform that data and on the output side, so when we hand it over to consumers, you could see very well the same patterns again. So you could do request response with the data product you could say OK give me a stream of events with all the changes you made to the data product give that to me on the consuming side or the whole data product as a whole. So that will be again a batch data movement you could query it with like SQL access. That could be something. Maybe inside I think that's something you don't have here yet is yeah, the inside is in the next framework.

Interviewer
That's about anatomy. Yes, this is more about the infrastructure around it. So everything that's happened outside of the data product.

Expert B
I think you could also use all of these aspects also next to each other. So imagine we take the data from that operational application, move it into a secondary location where we make it accessible for other users. You could store it in an file format, but you could also store it in database format and in databases you have Q value stores, RDBMS, columnar stores, documents, graphs are there are many kind of databases and here you should also look at the consuming side. So what read patterns do they need and what works best for all of their use cases and maybe decide to add a variation of these so it could be very well that you develop or design the data product architecture that it would contain several of these standards or interoperability patterns. Yeah, however that's I think more in most advanced and mature scenario. In reality most customers they only choose one interoperability format. Most of the time it's SQL or Delta or Park as a file format and then with some lightweight.

Interviewer
Yeah, interesting! OK let's have a look at it. This is more about what I should consider....

Expert B
Because I think what you miss here is metadata management. So there's a lot more you want to describe for a data product, so in what domain was it created? The context we mentioned, but maybe you want to know what people in that department work in there, how long, what is the data quality? So data quality here is missing. You want to see the data quality trends over time. Master Data Management is a very essential aspect. We mentioned data historicization, so that's whole data lifecycle management part. So how do you ensure that all data is also cleaned up? Meeting regulations. So if a customer come to your organization, how do I ensure my data is really deleted also from data that sits inside these data products? Yeah, regulation part is missing as well. Um. Yeah, data security you already have. Yeah. So metadata management, data lifecycle management, master data management. You also have something what I call reference data management. Country codes. Currency codes. Oh yeah, simple example. Imagine we develop multiple applications inside an organization here in the Netherlands. Chances are quite high someone for the Netherlands uses another. One uses NL, another person uses NLD. If you would bring that data together, how would consumers ever combine that data? So you need to have a discipline for that as well, keeping tracks of where do you see all of these, what I call reference lists and you need to apply reference data management for that as well to ensure data at some point. Cleaning I would consider that more part of data quality management. OK, but this is really an area of data management of master data management. And within master data management you have two disciplines.

So there's master data, customers which you create on many different places, and these could have different addresses and slightly different names. You all want to standardize these, but you also have reference data and that's more for giving caller to data, reference data. So how you could describe data and that's another discipline you need to manage.

Interviewer
Hmm. Ohh, this seems like some similarities but in the interface decision, because I observed some kind of observation port, a control port and a discovery port. And of course we have the input and output port. But these would be the, yeah, the ports on the perimeter of the data products, how we can access the data? Well, with the observation port we can observe the quality and check if the quality is in line with the service level objectives and things like that. The control port would be more relevant for the for the enabling team. So the ones who are observing, yeah, if the data product. Is adhering to all the policies or the defined policies, so it's more like the management layer,

Expert B
Yes. And the security is also part of that part.

Interviewer
So the control. So who could access that data for what purpose? And the discovery port is more some kind of sneak preview for the consumer to observe what kind of product is, it's like the port who is providing you access to the data catalog inside the data product. So it's like the metadata that describes what's happening in the data product and with the data to it.

Expert B
Honestly, the control and observation port. Imagine we design a data product exactly like this. So how some Zhamak Dehghani feels it should work. So if you want to know the freshness of a data product. We should query the data product itself and then we see here whenever it was updated. Hmm. Imagine a large scale organization which has thousands of applications. I used to work at ABN Amro we had more than 2000 applications there, 300 DevOps teams maintaining all of these applications. Assume we go down this hill and we ask all of these teams to create small data product architectures which you need to query in order to see the freshness. How would I see the overall freshness of all data products in my organization that would require me to run a Federated query maybe against 10 thousands of data products? But I don't consider this as realistic, so yeah. I think that the control port, observation port and probably also part of that discoverability port will move to another layer which is more overarching. So it will probably be part of a central data management solution and data catalog or data quality. And how it then would work if you do data quality? So you check against data and then the observations you publish somewhere in a central data management solution. And you access that to see the freshness. Not a good reason why you would like to do this. Imagine we don't give guidance to all of these different teams and they all use different data quality frameworks. How do I see end to end consistency of data quality if they all use different frameworks and different ways of collecting statistics, metrics, designing the rules for data quality. It will be impossible as an organization to see the overall health of your entire architecture. So strongly what I miss is you should take a more global, holistic approach to this. Yeah, and then probably if you start implementing this and you want to scale up, look more realistically. People aren't there yet, but I strongly believed that maybe after two years from now they will look back and they say, no, this is the wrong approach, we should, yeah, it's just more my perspective.

Interviewer

Yeah, I also think everyone has a different method of yeah of implementing data mesh and there are so many options, options.

Expert B
It depends on. This could work. If most of your consumers. Require only a few datasets or one data set. That could be very well the case for lots of machine learning use cases, yeah. But if you would like to do enterprise, so group reporting, financial reporting, risk modeling, fresh reporting, what most large enterprise do then require lots of data from many different consumers and you all need to combine and integrate it and know the quality not from a single data product but more end to end and then such a fine-grained design will lead to problems.

Interviewer
OK. OK. And if we now consider the data product itself, so if we look more at the anatomy, so what's happening inside the data product, I observed several patterns. So again the control plane which is connected to the control port and this is partly what we have discussed before, the observation plane is connected to the observation port. Um, yeah, the data on boarding is yeah, it's more like the data ingestion. Uh, the data catalog itself. So this is the place where we register data sets that can be a difference between domain datasets and core datasets. Domain datasets are the datasets that are being shared within the domain. So for example, the HR domain is sharing datasets inside the domain itself, but only with HR but there can also be core data sets and these data sets can be shared by HR together with sales and they can combine some kind of recommendation engine etcetera. It's just about sharing between the domains. That's what I mean with the core datasets. Um, yeah. I've read a lot that the data products should each have an internal storage to be autonomous. And these two, so a change data capture is connected to the event streaming backbone. So if something is changing inside the data product it's sent to the event streaming backbone and at the same time there needs to be an immutable change audit log which converts all these events in more readable formats and it's keeps track of all the data lineage that is going on in the mesh. Perhaps I can say that the immutable change audit log is more like a pattern that can be implemented outside of the data product. So. At the moment I'm explaining this to you, I'm I'm doubting my decision to place it inside the data product, so inside the anatomy decision.

Expert B
Change data capture is more the technique of moving small chunks of data from one side to another side, so. They mentioned we, um, develop an application and we designed it in uhm SQL Server, that's the database where we will store all the data in and instead of reading the entire database each time, we only would like to capture the changes. So for that their software you could use and that software is called these days, change data capture. So you change only the changes, yeah. What you would then will do with these capture changes, it depends. So you could very well say I deploy another database on a secondary location and I will just. streamline all of these events and we have two databases keeping each other up to date. Yeah, you could also use these changes as input for an event streaming platform. So then the change data capture software would spit out JSON Files or AVRO messages and we'll throw that at an event streaming platform. And then from there you could route it to other applications. Or you could store it in an immutable audit log for instance. Or you could write it to another location. Or I could store it in a data lake and let all of these changes then lend and pick them up from there. So I would be cautious maybe here in this model that that change data capture, it is really about the software you use, it's supporting for what you probably would like to do here. So you would like to differentiate between you have full data movements. And you have changes or deltas. So, so delta or increment data movements, I think that's the difference here.

So if you would design this, probably your users in your organization, they would either ask for give me an architecture that moves in all data and does the comparison and then I keep tracking history of all data I received. And another could be I have this need to be up to up to date to quicker faster, make decisions on changes I detect and see and that's where I think. Umm. Yeah, event streaming pattern. And to implement that, yeah that could. One way could be change data capture, but you could also speed out the changes with API calls. Or you could if you have a modern application. Some modern applications could trigger or issue a function and that function then would spit out a message and would directly deliver it to an event streaming platform without a need of change data capture. So change data capture is software to one of the scenarios to make it faster and provide real time streaming data to customers, but there are more patterns.

Interviewer
OK, yeah, I haven't thought about the API calls to be honest, but that's a really good point. I can deep dive into that even more. And what do you think of the other components I would implement inside the data products? So for example internal storage, do you agree that each data product should have an internal storage to be autonomous on its own.

Expert B
There again, it's much more nuanced. If you would give each data product each own dedicated store, it again could dramatically make your architecture more complex? Yeah, so if you have in that scenario I discussed, 10,000 data products, you would then have 10,000 storage containers you need to manage. Umm. On Azure, so I work as a cloud architect if you would like to use Azure data factory, the data orchestration tool for importing data, you need to register each storage account individually if you would give each data product its own storage account. And I need to combine lots of data that I would probably spend my first days spending configuring storage account configuration and links. Yeah, that's probably not what consumers want so I would then take a quite a different approach. Look, I would decouple the two and maybe you have centralized or shared storag. Yeah. And you have logical partitions or containers in which then the data is stored. But I feel what misses here is the whole data pipeline, her movement. So it's when data comes in first? Yeah, it's raw and you need to transform it.

Interviewer
Yeah, that's what I summarized in the data on ohh, just onboarding.

Expert B
Yeah, but the data onboarding could be more. So yeah, maybe in the onboarding you check for quality or you check for the schema. Is it still what you expected? If not, you may reject the data. You could also apply security on the onboarding part. So if suddenly somehow the location of the application has changed and you start pushing in data, maybe in a large enterprise. You see what data is no longer coming from the internal network, but from an IP address which is not known to the security department. Maybe you drop all incoming data. So in data onboarding you could also be. security measures be implemented. Yeah, maybe in the data onboarding you would also would like to check against the catalog, so before you ingest data, did you make yourself known in the catalog is all the conceptual data there that you link all the business terms to the data attributes that you even make yourself known as a data owner before pushing in data. Otherwise we have data floating around in our organization not knowing where it's coming from, who's owning that. So that could be I think also part of that. And then yeah, there will be probably lots of interaction between these flows. So that control plane could very well then interact with the data onboarding part. And again with the data

quality.

Interviewer
Hmm. Yeah, OK, so I should specify the interactions between the patterns here.

Expert B
Yes. Even more ohh yeah yeah. What else? Infrastructure provisioning. So I'm actually refers to that as well. So you have that infrastructure plane maybe? I process my data, but for data I need temporary some infrastructure, so I need compute, a spark engine to be provisioned somewhere, yeah, or I combine and analyze lots of data and for that I need an high performing storage layer, but that not necessarily always must be there, so I want to only spin that up and make that available. The moment I start processing data. Hmm. So I think an infrastructure, maybe orchestration component or part could be added to this. Maybe a metastore metadata store.

Interviewer
Yeah, that's more like the data catalog, right?

Expert B
Um. Yeah, it depends on how you engineer and design this. So usually the data catalog, it's quite broad. So you not only see data product data, but you also see transactional operational applications. You want to see all the models, the information about the organization itself or the reports that are being developed and using what data products. I feel the data catalog has a very broad scope, but maybe for this architecture and what we do here, we want to have a meta store metadata. Or more close where all these activities would run so the schema versioning and the and the contracts, I store that not in the catalog, but in components which are separated out. And it's more likely than that these components, these metastore components deliver their metadata to the catalog. For the processing of all this, you would use them directly. OK. That's more in practice what I see. Yeah, they yeah. So then that metadata management should be added.

Interviewer
I could have a look at that. That's because you also mentioned that during the framework over here that we should have some kind of universal metadata repository in the infrastructure layer, and that the central data product catalog on its own and isn't enough. So yeah, I could look more in general in the metadata.

Expert B
Yeah, it can spend a bit. So honestly I see more smaller organizations they used to central catalog. For almost everything, yeah. The somewhat larger organizations. They make standalone metadata repositories or components. For complementing this way of working. Yeah, the requirements more often exceed So what these data catalogs currently offer in terms of functionality and features.

Interviewer
But this also depends on your legacy architecture I think. So if we have a look at this framework over here, well, when you start from scratch or when you do some Greenfield development, there's much more flexibility in your decision making. But if you have a master data management approach, where for example, my company is using Snowflake and in my perspective Snowflake is really a master data management tool. It's really centralized. So yeah, migrating

towards the data mesh requires some different decision making compared to, yeah, when we start from scratch and have all the opportunities to choose from.

Expert B
Yeah, on master data management, um. If you. Look back in the theory and all the books, um. What you've been told is that, UM, data management works best. When you centralized a lot of these activities and Zhamak Dehghani also in her data mesh article, she describes that as well. So Central team, central data model, central data warehouse, central database. Central integration layer, etc. Yeah, she now strongly advocates for decentralization federation in a way, she says. There's no single data model anymore because we have autonomous data products all using their own structure and languages according to domain driven design. Yeah, this is in theory. In practice, well, yes, you could say there's no single data model, but this doesn't mean there are no shared entities between all of these different departments so if you have an order management system. When you create new orders you need to link it to a customer. So it's highly likely that there are also proportions of customer data sitting in that order management system, so these transactional systems. It's not a single shared data model, but there are lots of shared entities floating around in your organization. And she misses this point a bit and this could lead to lots of problems on the consuming side, if you forget about that and you ask all your individual domain teams to start individually themselves all of these data products and then on the consuming side I see all these data products and you see, well, the identifier from the customer management system is also being used in the order management system, although they transformed it to a Global Unified Identifier and therefore it's no longer recognizable, so I am unable to integrate it and join it with the customer data. If that's the case, none of your consumers will be happy and could use any of that data for their analytical or consuming use case, so. Yeah, you still need to add some of the discipline of master data management to your data mesh architecture. Yeah, to ensure data can be integrated and joined. That Strangler-Fig,. that's more like where you provide some compatibility from the previous scenario. So, what I often see is customers they designed an reporting solution or data warehouse they would like to move to this target architecture. But it's in a transition, so they need to. It's not, yeah, if you make external reports to your stakeholders, shareholders even, you could not say, well, yeah, I'm on this journey, so next seven years. I don't provide any insults about the financial health of my organization, so you need to ensure some continuity in that respect, so. Um, that's where that second pattern makes sense, where you have the old architecture running side-by-side, really decomposing slowly. You gradually move away. And maybe the output of these newly created data products somehow follows the same structure or compatibility of these old data structures you have created over the past.

Interviewer
Do you have, by the way, a few minutes extra?

Expert B
I have more time

Interviewer
Great. Thank you. Time is going so fast right now. Yeah. Yeah, sure. Yeah, the zero trust architecture. I don't know, in my perspective, we can perhaps combine some patterns over here, for example, master data management and zero trust architecture. We can somehow combine that and...

Expert B

Yeah. What I feel here, it's. It seems like it's one already now or another. It's mostly combination of all of these. So yeah, zero trust focuses on security. Security you always need to have, so you could argue well all large corporates or enterprises. They need to adhere to security policies, so surely zero trust will be always part and zero trust it's non-related to data mesh so zero trust could be an applicable pattern very well in the already existing situation. CQRS is another software architecture pattern. It stands again apart from data mesh, so you could very well apply it within the spirit of data mesh. But you could also implement it when designing an operational high performing application with lots of reads and limited writes and therefore you decouple the reads from the writes and implement CQRS. Umm.

Interviewer
Yeah, I feel like this is not complete.

Expert B
My view, um, what I see in some organizations. They start indeed in Greenfield and adopt this principle federation, decentralization, centralization from the start. Well, other organizations. Have a more nuanced approach or they start with centralization first, OK, in order to decentralize these activities at a later stage? And why they start with centralization first? Because they feel in the beginning it's more easy than to have control and overview over your architecture. In the beginning it's more difficult, so you need to support all of these decentral teams. With the right guidance and right technologies, if these technologies aren't there in the beginning, it is maybe sometimes better to start with a centralized approach in order to federate later. I see some organizations today. choose an approach what I call hybrid. So more mature domains, they work according to the principles of data mesh, but the more traditional domains or domains with lots of legacy. They continue to operate and run much longer in using the old methodology. So then you have side by side and this is what I call hybrid.

Interviewer
Yeah. So yeah, it sounds a bit like the Strangler-Fig like small yeah, slowly decomposing your current architecture with the yeah components that are suitable for data mesh at the beginning that are yeah, that can be easily transitioned towards data mesh and well while keeping your legacy architecture present. And slowly, yeah, decomposing it. So slowly transition towards the data mesh and not doing everything, yeah at the start, but doing everything in certain time periods.

Expert B
Yeah,. I also see your own approach. Smaller organizations, they look at data mesh, they very much like these principles, but they don't feel they are ready yet to federate at least the responsibilities. So the responsibilities are kept much longer central, although they adhere to many of these data mesh principles. So no longer data first is integrated. Now they design data products like how it has been described in the theory of data mesh, but they do that on the central level. And they do some virtual alignment within the organization to envision the domain orientation, but still it's all central. So it's one large central team, but they do virtual alignment, virtual domain orientation within that team. OK, you could have a conceptual debate. Is it really data mesh? But at least the four principles are being adhered to so there's domain orientation, they develop a data products in that respect, following the guidelines, observability, everything is there. They establish Federated computational competence because they have policies, they enforce lots of these things and ensure the catalog and observability are always there. Self-serve, you have that central team could also do it self. Where on the business side they could do self-serve and look into the catalog. So yeah, yes in a way it's I

feel, um, they adhere to data mesh. Hmm. But yeah, you could also have a conceptual debate. Well, yeah, it has not been fully Federated because those domains are not autonomous. And the platforms, uh, aren't decentralized. So it's not that these all of these different teams own their own infrastructure.

Interviewer
That reminds me a lot about one of your articles about data mesh topologies, where there is a really distinction between coarse grains data mesh and more fine grained data meshes and those are really separated between each other and also the Federated component and the more governed component and centrally governed component. I really like that approach and I see a lot of similarities right now.

Expert B
My recommendation would look again, maybe at the right hand side. Hmm. And maybe approach should be, um, that word should be better defined, so yeah. Is it like the transition approach or um implementation approach, infrastructure approach. There are different viewpoints.

Interviewer
Yeah, I agree. There are also some different implementations correlated with the deployment decision when considering your migration over here and that's actually where my last framework is about. So I actually observed two, yeah, two main components here. So Docker of course for your containers and Kubernetes will. Yeah, these two are really aligned with each other because you can't have Kubernetes without Docker container because Kubernetes is some kind of organized Docker orchestrator. Um, yeah, actually. Most of the articles were mentioning these two, but and. Yeah, sometimes it's a function as a service and infrastructure as code. Do you think there are more or do you agree with this? Yeah, with these two practices?

Expert B
It does to me is really the whole the fine-grained way of doing

Interviewer
With the single container design?

Expert B
Yeah. So yeah everything is a container always, yeah. But then where's the data stored? Um containers? Best practices are stateless, so you should not store the data itself in a container. So you need to have central or individually managed so the infrastructure. So the storage infrastructure part is missing. Yeah, the storage. Um. Yeah, how to deploy? Yeah, you could also do it without Kubernetes or containers. Just publish in your catalog that you have stored data somewhere. It resides on this location. So what I see? Um, almost all my customers I work with honestly they don't use Kubernetes and Docker containers for doing this. All of my customers I work with they use a Lake Hoise architecture. Hmm. In which they design these data products? And then these data products reside in folders or containers within their data lake. OK, and they make a reference. In the data catalog of where the data then is being stored? Or they make a reference in a central metadata repository? So they created a small tiny application. It's a metadata database. There's a nice front end on top of that where you can self-service, register your data products and look them up. In the database they keep track of where all of these data products then are being stored within these different data lakehouse architectures. So what you then see is, each team they have their own data Lakehouse archi-

tecture, so it's their own data lake in a way which services you need for moving data across and the lineage and the transformation of the data pipelines of these services are there for each team. So each team gets their own environment and inside this own environment they onboard sources themselves. They transform their data and then the output data is what they call data products and they store these in containers or folders.

Interviewer
OK. Yeah, because I have seen lake houses before as some kind of uh yeah component in your data mesh but I haven't really deep dived into it

Expert B
I will give you a link to my book you can read for yourself but so this is the most futuristic way of doing this. Yeah yeah, so how Zhamak Dehghani describes this in practice. Honestly, I see no company doing this, so they all use now lake houses or data lakes? Or storage accounts in which the data is stored.

Interviewer
Yeah. OK. Yeah, thank you very much. That's, that's actually all about uh my frame, my individual frameworks I've actually won last question, it's more about the inter decision link.

Expert B
So if we maybe go one back before I forget what you could also do here, what I sometimes see organizations, they use a streaming platform. For instance, Kafka. Kafka, that's what I meant with the event streaming backbone and event streaming and instead of using Kubernetes, you could also use a shared Kafka environment and then each data product they say it's a topic in my Kafka cluster. Some of these topics are internal, only belong to domain. And other topics are classified as formal data products, so these are categorized and listed out in the metadata framework or the catalog for this event streaming platform. And then you could use these and. Yeah, then there's no container or Kubernetes part of that, that way of working. So that's I think a secondary or maybe third alternative to this.

Interviewer
Yeah, because now I only encountered two options, but there should be more than more than two it.

Expert B
Yeah and I think look at data mesh more conceptually. So to me it's a methodology, it's a framework. Sometimes what I also say to customers, you could even do it on traditional mainframe systems. Hmm. If on the mainframe we spin up a secondary part and there we let the data lend and surface from there and we apply all of these methodologies of discoverability and self-service and product thinking. It's standalone from the technology part. Yeah, but that's often the risk I see. We have this notion of jumping straight into technology, but it's, um, unrelated to technology. You could do it on on, you know, technologies from 20 years ago as well, yeah, if you follow and implement the standards and frameworks.

Interviewer
That's also in line with one of my questions overall, because what's your perspective on applying all these frameworks in terms of difficulty? Do you think it's still too general and it should be more specific. Or do you think it's really understandable for practitioner and the options I discovered are really relatable and can definitely be implemented?

Expert B
Yes, my view. So what I feel these days is. We already have done this data management for quite some time. Yeah so. Um, I'm a data practitioner myself and now I feel with data mesh we start to give new names to things we already did long time ago. So input port it's data ingestion.

Interviewer
It makes it confusing

Expert B That's I feel makes it's very confusing for lots of people so we now are confronted with all of these terms and the problem it's in a commercial organization ThoughtWorks. So a bunch of consultants, they advocate all these new things. And it's, yeah. They do this for a living. So it's for them their revenue model, so these consulting activities and I think that this makes it for data practitioners difficult and it's more nuance, so it's not one and another. You could do data mesh as well data fabric, something maybe not part of your thesis, but you could complement these different approaches. You could do data warehousing also within data mesh or you could build an operational data store also in larger architecture and that follows the data mesh principles. So it's not one on another, all of the data management areas and methodologies are still there. And complement each other. This is more what I see on top, so it's a bit of more federation. So instead of a central team doing all that work for the entire organization, you move these activities across. And how fine grained you do this. Yeah, it has impact and influences, the underlying infrastructure architecture and the way you configure things and set things up, but the principles itself. Aren't changed and you know, sometimes what people forget it's all new.

Interviewer
No. Yeah, we. Yeah, it sounds like I'm also missing some decisions. Hmm. So I observed six main decisions over here, but it sounds like I'm missing a few. So the first one is the data product type we discussed. That's the first decision you need to make before creating your data product. The next one is about Greenfield Development or migration. Then we can at the same time make a decision on the infrastructure, so the data, product communication, the self-serve platform, management layer, etc. Thereafter we have the anatomy, so what's happening inside the data product and at the same time the data ports. So the data product interface / contract. And the final step would be to make a decision from the, yeah,

Expert B
If we go to the left, what I see at customers, So what types? Yeah, there could be a raw type so we have just the raw representation of the data. It's in an application we throw it at somewhere on a location and we say, OK, that's a data product, it's raw, you could use it. There's also this type where it has to be modelled more read optimized, but still in the context of a domain. What I have preferably would call that's more the data product data and the way you would like to just subscribe to it. Sometimes within I see the classified data like you already also observed, it could be core or internal. On the consuming side I see different variations so you could model data after consuming data products and some customers call that a consumer-aligned data product. I'm no big fan of that concept because it's tightly coupled again to the structure of the consuming side. If you start to expose and share that to others, they are tightly coupled, bound to the same problem you would have with raw data on the other side. So I'm no favor of that but some customers classify data project for that as well, some organizations they enrich data on the consuming side, so they create new data and expose that again as a new data product. So that's another variant. Umm, you could say you construct,

so you take data from different places, you compare, you combine, you aggregate, you create business calculations rules, and that becomes a new data product then. So you construct and construct a domain. I see organizations they also combine data and they master that data, so they combine customer data. So you have different customer applications inside your organization. They combine all of that and they create a new mastered data product, they call that. Customer data but then mastered because it has been combined. We take data from three different masters. Yeah. Customer administrations where the master data sits within, they combine, integrate that, they compare, they see differences. So Janssen is there spelled with two S's in another application with one S, that probably should be a mistake. So we correct it and we have one unified version of the truth then and they call that an mastered data product.

Interviewer
OK. It's new to me to be honest.

Expert B
So sometimes the composite is another variant. So imagine you have lots of consumers on the right hand side of your architecture and they all in a way combine and integrate more or less the same data, would you opt for scenario. They all do this themselves on the right hand side. If you have many teams that to me would be a waste of time because all of these teams would do the same transformation steps. So what you could also think of is. Let's cook the data upfront and we create this aggregate or composite as input for these consuming domains. So some organizations take all these aggregate data products so that yeah, listing that out, you have many types of data products, so you have raw. provider aligned, consumer aligned, aggregates, constructor, master, shared core. And I'm giving a few examples here and what organizations do, and maybe instead of listing out um, I would more talk in terms of a framework or methodology so they list out these different types of data products and then next attach the principles to all of these. So they say if it's a mastered data product you should use master data management services as well data quality services. You should in the catalog then classify these data product as being mastered and you should always add the lineage. Knowing so where the data is coming from. For raw data, yeah, it comes without guarantees so lineage shouldn't be there. The business terms not necessarily have to be provided, so. Organizations, often they create playbooks or documents listing out what kind of data, products or types they foresee, including the principles and best practices and way of working attached to all of these. Yeah, if you make it more concrete, sometimes I see they even link this with the whole data governance framework. So within data governance they listed out well, there's some data owner, there's an application owner and data steward, data advisor, all of these different rules. And then in these playbooks they're very fine grained and describe all four consumer aligned data product. The user always must submit the use case, describe the scenario he would like to use the data for, the purposes, the notion he intended to share the data with other parties if so. Well, you see, quite concrete, yeah, the role names, the instructions for each type of data products. This has been spelled out. It's a lot of work, but honestly, I see, many organizations. Yeah, taking this approach of providing guidance playbooks to the organization, aligning this with the data governance framework as well.

Interviewer
OK. So not to stick only to the source aligned data products, aggregates and consumer.

Expert B
Yeah, so maybe so maybe for your thesis. So instead of. So yeah what types are there maybe you could consider what may be best practices or methodologies. Do you see a correlation

between these different functional areas based on the types? And if so, yeah, maybe categorize a little bit on that.

Interviewer
OK, yeah, that's a very interesting one, because now I only defined the algorithms raw data product and a derived data product, but there, yeah. We can make it more extensive. With the, yeah, with the provider aligned, the product, consumer lines, etc. Well, thank you. And the other components do you think? These are OK or? Are there still some missing gaps over here? So this was the migration. This is the infrastructure layer. Hmm. This is the anatomy of the data product itself. The interface showed the ports. And the deployments. Can there be more decisions we need to make?

Expert B
The metadata? The catalog? The descriptions? The context in which data products are developed. Should we capture and describe that maybe as well? Yeah, it's a separate decision. Yeah, the lineage. So knowing where exactly the data originated, transformations that were applied before data became the data product.

Interviewer
Yes, it is mostly in the context of architectural design decisions.

Expert B
So yeah, perhaps metadata is not really architectural, but governance if you go to the left. So what was your first question?

Interviewer
First was about type. But there will be two other frameworks. So I will develop some kind of self-serve platform framework and a governance frameworks of federated governance framework. Hmm. So this is really about the data product itself, what decisions are connected to? Yeah to this principle?

Expert B
Yeah, for sure you need infrastructure and services and technology. I think there could be indeed a relation to the type and then the kind of services you would need for. Like that master data management? Yeah, master data product. That would require different services than something you would do on the consuming side for example. Yes, self-service. Sometimes I see also organizations, they create data products within a particular scope. And they say, well, we make it available, but not for all use cases. So there are limits or restrictions attached to some of these data products depending on the scope and the quality and the classifications they apply. I think maybe classifications and labels.. What are the elements of that product? Yeah, there's input port, output port, the. Yeah, streaming API, batch, the pipeline,, the metadata needed for, you know, what I would maybe do more is peel it off so data product it's more conceptually, yeah. It's a logical view you have underneath. There's an architecture, I would call that the data product architecture. There's data product data. Hmm. Data product metadata. That could be stored in the architecture itself, but you could also share it with others and put it in a shared repository, yeah. Yeah, the central data product catalogue, yeah. And then the data product architecture, yes, you could closely align it to the data and the metadata and then it follows more naturally what Zhamak Dehghani described in her book and in the articles. But you could also decouple the two and you could use the architecture for multiple data products and then you have a shared environment. Yeah. Maybe if you start with that, I think it then

would also be easier to map these decision trees on all these different classifications you did so. So what type of data product? So the data product data and there you have consume data, source system aligned data, master data, core data, raw data. But if you would the type of data product, you would plot it on the architecture, yes, there could be shared data product architecture, standalone data product architect, maybe combination of these. So yeah. And on the data you would normally have classifications, but not on the architecture. The architecture is for services, so. And then these principles. And I think now because you not make concrete up front, So what viewpoint you used, it could touch upon all of these different layers and that makes it a bit difficult.

Interviewer
OK. Thank you so much. Yeah, this is a really valuable feedback. OK. I will stop the recording now because I think we have discussed a lot, actually everything I wanted to discuss. OK, that's a good thing. So let's stop the sharing and stop the record.