Appendix A

Interview Transcript Expert E

Expert E
So. Recording in progress? OK so. Yeah. So, so, so just as a side note, I go by Jean-Georges, OK. So that's because being a French in the US, you've got to. . . .

Interviewer
Difficult name to pronounce.

Expert E
Say that my parents didn't plan me like that. OK, so. So I'm not blaming them, but that's how it is. OK, so literature would say outside? Everything every time I read something which was not written by Zhamak. I had a problem with so for example there's this very beautiful site called datamesharchitecture.com or whatever. But I think they're taking, they're taking a lot of liberties with a few things and they're also opinionated. So it's good to be opinionated I think but it's also I think they have their own views on it. So I'm not saying I've got the truth but the thing is you know there's this book called: What would Google do? OK, I tried to think what could Zhamak do? OK and I'm not saying I've got the answers but anyway, yeah. OK.

Interviewer
I can I can perhaps give a short introduction about myself that you also know my background, etcetera.

Expert E
Yeah, sure, go ahead.

Interviewer
OK. I'm Tom. I'm Dutch and 24 years old at the moment. I'm doing a master data science and entrepreneurship, where we try to combine the technical perspective with the business perspective. That's also why I like data mesh so much, because I can really resonate with those principles during my studies. Before data science I was more like a bachelor student in industrial engineering. So I came from a business background and now I'm really deep diving more into the technical aspects of data science. Well, when we specified even further in data engineering. I'm doing an internship at Bain and Company and I'm looking at architectural design decisions related to data mesh, really from the practitioner view.

Expert E
OK, that's great. So I also looked at your discussion content document. How many people

have you interviewed so far?

Interviewer
Four, actually. A really diverse group of people, there was a data mesh product manager. Someone from Microsoft and he wrote the book: Data Management at Scale for O'Reilly. And I interviewed someone from the data mesh learning group. I really like the different opinions. So for example someone was talking about the data catalog that you have inside the data product and he wasn't really a fan of it because often you duplicate a data product and when you duplicate the data and the metadata, it can create some kind of yeah, it can create some kind of semantic problems because when you copy the metadata. For example, if you have ownership and you copy the metadata, then there's two similar data products with different owners, this wouldn't make any sense? And on the other hand, we have someone else who was more a big fan of the data product catalog inside the data product and it's telling me more about discovery, etc. So the first one is more like only make a central data product catalog and * is more like make a central data product as well as the data product catalog inside the data product itself. So there were a lot of different opinions on that.

Expert E
Yeah, I think there is. Uh. There is a little bit too views. And I see that at my company as well. There is enterprise view of things like, OK, you know when you're a big enterprise, they've got some enterprise wide departments like enterprise data governance, enterprise data quality, enterprise data platform, OK. So, they try to find a model that suits everyone and their thinking is by definition very centralized because they see that if it's there, they have the controls, they can see it OK. Where if they give the, if they give the right to the permission to the business units, OK, it's kind of they're losing some kind of grasp over it, OK. So. So I think there's this these two approach. Yeah. And I often compare them to like the grassroot approach and the and the centralized or enterprise approach. OK. So I'm more on the grassroots side. I think when you're thinking about the architecture it's always going back to the four principles of data mesh right. When you're thinking about the domain ownership. A lot of people ask me, hey, what's the domain? How do you define a domain? Blah blah blah. And The thing is, you can spend hours, days, brainstorming on what a domain is, but a very, very simple thing is a team OK when you're when you're thinking about, OK, I've got this data, it's the data my team owns. OK? So that's OK well, that's your domain. OK, that's not. Please don't be more complicated than that. So I think this is really where this approach of that allows you to say, OK, this is my domain, OK? My domain contains everything I know. It's my vision of the world, you know, like in the middle-aged Europe, OK, the world was Europe, OK. And the thing is, that's all I know. OK? So I'm putting everything together, all I know. I'm putting the governance there. I'm putting the definition I'm putting, I'm documenting everything I know at this level. OK, at the data product level. So. What's the point of centralizing? OK, you've got to centralize because the user that is looking for your data is the citizen data scientist or not even the citizen data scientist but anyone that wants to consume your data as long as they have the right to do so. You want them to find it easily. OK. And yeah, so there's still something centralized that contains everything, but it's only a copy. It's a stupid copy of the reality. The reality being in the domain. OK, so. And then when you think about search and discovery, well, yeah, you're not going to ask every domain "do you know, Tom"? You're going to ask the main registry, hey, yeah exactly but that's the principle of federation. OK so you're federating all that and so. . . .

Interviewer
We can perhaps have a look at the frameworks and then yeah have a perspective on each one of

them. So if I share my screen. You told me you have seen a couple of them already, but yeah, I'll give you a short introduction right now. So this is the inter-decision links. These are the inter-decision links and you can see this as more some kind of high level overview. OK, yeah, go ahead. Yeah, yeah. So this is a high level overview. Yeah. So according to Grey literature, I found some kind of chronological order in the decision making where we first start with the type of data product. So first we take it highly conceptual and make a decision on that part. After that we can deep dive more in. Do we want to migrate from legacy architecture or do we wanna do Greenfield development, so starting from scratch, there's a decision on that. Also we need to think about the interface. So what is the outside layer of the data products and not the data product itself, but what's happening around it? So how does it communicate with other data products? How does it communicate with the self-serve platform, the management layer and of course the consumers? Thereafter we have the data product itself, the anatomy. So what's happening inside the data product? What kind of architectural components do we want over there? The data product interface and with the interface I mean like the ports. That place only data product, so it's like more reports on the perimeter side of the data product. What kind of ports do we need? And yeah, most of the time we end this software engineering cycle with deployment. So how to deploy the data product. So if you start perhaps with each individual decision and at the end we come back to this inter-decision framework and check if there are some decisions missing or if we can.

Expert E
Yeah, yeah, we don't we don't work this way.

Interviewer
OK. Yeah, this is really from the academic perspective. OK, you can just start with the first one. So the first one is what type of data project can be developed. I identified 3 main options. So uh, you can expose your data product as raw data, you can expose it as derived data, so the top 10 customers, etc. You can expose your data product as an algorithm and there can be 2 variants based on the algorithm part. So there can be an optimization based decision support system like a BI tool or we can expose your data product as an AI / machine learning model. And these two options over here are more like a hybrid data product. And with a hybrid data product I mean the data product can have for example two parts, one for the raw data and one for the derived data. So there's some kind of combination two options, but we can also use the data product as a composite product. So how do we somehow merge for example the algorithm and the raw data? This is what I've identified. So three main options and there can be some specializations between them.

Expert E
I don't think I don't think, I don't think you're missing out. I think it's very detailed. We don't ask ourselves that many questions. Maybe we should. We are. And I remember having the discussion with our enterprise data governance team which are great people I guess though. Then they had a strong need for qualifying everything. And this is always a little bit not bothering me but kind of tickling me in a way. OK. Like we mostly I would say we mostly do if, if we base on your diagram where we have mainly two types of data products, OK. I would say we have raw data, all right and we've got derived or curated data. OK, that's what we have. But overall. Why do we care? The thing is, when you want to consume the data right at some point. I'm a data consumer. OK, so. I just need the data to be described to me. And then I can see the description. Like when you go to cattle, OK, like, you know, like a. You're probably a little too young to know this mail order catalogue you are in in the past, right? You open it, you see the description, you say, OK, that's what I want. I'm going to order this

thing. When I'm talking to the data scientist. They don't really care if the data is raw data or if it's curated data. What they want. They want, they want to know, OK, if there's some transformations that happen to the data, they want to know it says Alize, OK. They want to know the data quality that is applied to it, and they want to know when something goes wrong. OK, so I don't see I believe that was the ones I'm dealing with. I don't see for them the need to understand that level of details when it comes to their product. We're pushing on them. So if I think that if I'm adding this extra layer on top it's not making things easy for them. So that's what kind of inspired me. Let's continue. I want to have a kind of a global view before I can…

Interviewer
We can go to the to the next step For sure. Yeah, yeah. The next one is more about do we want to migrate from a legacy architecture or do we start from scratch, do we wanna start from scratch. So I've read some of the articles of *'s and he mentioned four particular patterns. The first one is about master data management, so that you have some kind of master data management data mesh. So you keep your master data and you create a mesh around it. The other one is to strangle fig. So where we slowly decompose our monolithic architecture and slowly create applications that are decentralized. And this zero trust architecture, yeah, I think it speaks for itself and the CQRS about segregating your write and your read functions OK and the main feedback from * of on this diagram is that I, uh yeah that it looks now like you have you can make one of the one of these as a decision, but there are certain combinations so you can choose the master data management and strongly fit together you. And use both of them in your migration part.

Expert E
Yeah. It's never black and white. Yeah. Yeah. Uh, OK, OK, I could see that. We are mostly for our self in Greenfield.

Interviewer
Mostly in Greenfield, so you started completely from scratch with nothing at the beginning.

Expert E
Very specifically for the data product and the data mesh part, not the data, right. The thing is yeah, the data and some pipelines and data quality and all that was already in place. OK. So it's a bit the add-on, OK. So the myths, the 'meshisation' of that OK or the quantification when you building data quanta from that was Greenfield, OK. Yeah. So there are two parts yeah, because like the data management part and the data part.

Interviewer
Yeah, because I can imagine if you choose for Greenfield development that there is so much more flexibility in your choices so that you do not stick to some kind of old legacy system, you know?

Expert E
Yeah, it's never that's straightforward. It depends on the company and the company culture and infrastructure that is available to you. But yeah, for us it was mostly, it was mostly Greenfield. OK, I'm not saying that and The thing is it's probably going to stay Greenfieldish for going forward for the management parts, OK for the data I can see us doing this kind of patterns to create the new data products, OK. So now that we've got the management tools creating new data products based on that could be I see a stronger thing we use MDM. I

think we're going to MDM, but it's not like we're coming from MDM, OK, it's more like the opposite way, OK, so other things that based on the data product lifecycle you can more easily go towards MDM design, building your MDM and then exposing it.

Interviewer
Yeah, my, my perspective MDM is uh, is a bit centralized. So it's it feels a bit like we're doing the opposite of data mesh. So I'm not really sure about that pattern. What's your perspective on that?

Expert E
Yeah. I wonder if I'm MDM is not like more marketing than. It's like a zero trust architecture. Our industry, and you'll see that more and more as you see as you're growing, is really keen on having this kind of fancy marketing buzzwords, OK. I think even worse, even worse than other industries because a lot of the things we're doing are pretty complex. Though marketing comes in, find a quick sentence like zero trust architecture here or something like that. OK, so you make a slogan out of it. The thing is it's, you know, it's a bit like data fabric. What is the data fabric? OK, so thing is they're fabric. It's kind of exciting name, you know, but the thing is, you know what's in there, OK, it's all the all the ******** you rebundled together that are in your catalog before and you put everything together. OK. So it's not really innovation like data mesh is.

Interviewer
Yeah, yeah, because the next one is more, uh, well, this is at least a more extensive diagram. We can slowly talk about this one. Yeah, this is really about the infrastructure. So what's happening in the layers around the data products. So yeah, from *'s articles, I saw the schema registry. And the schema registry is slowly, well, slowly is connected with the central data product catalog and the event streaming backbone. And actually what it does is it converts the incoming events from the event streaming backbone into a more readable format for the user. Hmm, so the user can discover some event data in the central data product catalog. And this is more readable. It's more readable converted by the schema registry. Umm. Yeah, and there can be a shared storage implemented on the outside layer. So where some data products can store their data in. This is connected with a storage read API and a cloud storage API. Some practitioners mentioned the problem with versioning. So if you have a shared storage between two data products and one of the data products is getting versioned. Well, the other data product has to wait for that data product to be versioned. Otherwise, it will lead to some versioning problems with the shared storage. So it also decreases the autonomous part of the data product I think. So it's not really autonomous anymore, it's dependent on the other one.

Expert E
You're recreating some mini-central. Uh, we don't do shared storage. It gets his own internal. it's shared in a way that, OK, it's the same database or the same engine or something, OK. But there's no dependency that would prevent from one to move to a new version versus another one. OK. So it's all this practice of data engineering and we use tables and views and things like that because yeah a lot of things is read only for now. So yeah so but the shared storage for two data point that we don't do. I think that that would be more a nightmare. Yeah.

Interviewer
And another option is the API invocation. So there can be multiple kinds of APIs. The rest API, the graphQL, gRPC, but of course the cloud storage API and the storage read API are also connected to the API invocation. Actually I should create an arrow between the rest API's

and these two because it's like of variant of the REST APIs.

Expert E
We use a lot of API's but not for data. OK, what do you mean with that so we're exposing controls and our metadata and observability via APIs, but not the core data. The core data is. I'm not really at liberty to talk about it but it's not API, right? We don't think that the APIs are going to be fast enough for our use case. It generates some latency, yes, and it involves also data movement which we're trying not to have so OK that's a bad practice. OK, so we have a different question. But I can't talk about it. No.

Interviewer
Sure, no worries. OK yeah. Furthermore, there should be a SQL access point, because most of the data analysts nowadays only know SQL. Well, not only, but most of them. You probably know what I mean.

Expert E
SQL is still a language frangula for a lot of people for sure.

Interviewer
Yeah. And also with the introduction of Snowflake which is entirely SQL based. Yeah, it's even more important than ever. So yeah, on the bottom we see some kind of non-functional requirements. So data product policy enforcement mechanisms, yeah. What I mean with Query catalog is that we have some kind of sample notebook. Or some kind of query guideline where the data analyst can find all the possible queries he or she can use to access the data. It's some kind of manual. Security controls which is also non-functional and yeah in-memory cache. Do you resonate with some of these?

Expert E
Yeah. So we have, of course we have a provision for government well. I would say we've got some basic governance built in, OK. So that's something we already have. There's some kind of company-wide policies that are kind of non-debatable. OK, that's all. Yeah OK and therefore it's really hardcoded in our product. However, in global policy. I would say strategy. OK, so when someone decides something, we want to be able to push it to our data product. OK, so that's why. That's for us part of the control plane and the control services. Via APIs, but that's not something we've done yet. OK, so it's not. OK, we don't have this, we don't have this push mechanism from a central data governance tool. I would say like an extensions to Collibra for example, or something like that, that would be able to push the rules down to the down there. OK, so let's say hey we. We want to implement GDPR. On this product. OK, so if you want to have GDPR. We are just going to say to push the rule. OK. So there's a provision for that but we've not done it yet so I see your NFR there and that's a good thing to mention for sure.

Interviewer
Yeah. Yeah great. Do you have by the way a few minutes extra because we're almost out of time already.

Expert E
Yeah. I can go to 9.

Interviewer

Ohh until 9. OK great. Now we don't have to hurry. OK yeah. Well, if you see this framework, you'll think I'm missing something when I'm considering the outside layer of the data product. So this is really infrastructure layer, but I think perhaps the event streaming backbone is something, yeah?

Expert E
This is a way we're interacting with a data product. We've defined 5 words. Five sets of words. OK. So there is the observability client.

Interviewer
That's what I have over here with the interface. We can perhaps quickly go to this one and then go back to the other one. Yeah. So this is really based on your article actually on medium. So I saw the different kind of ports. So the observation port, the control port, the discovery port. I noticed from the interviews before that I forgot to mention the most obvious ones. So the input port and output port. Should implement those as well in this in this framework over here.

Expert E
These ports accesses metadata, OK in general, OK. So whether it's active metadata or it's passive metadata or it's the management of the data product itself. OK. So I think that's what I described in the medium article and it's still there and it's still just there. OK.

Interviewer
One question about this one because I showed this framework to ** as well and he's really like yeah, if we have 10s to hundreds of data products and we have all these kinds of ports for each data product. When we want to check the data quality through the observation port, yeah, we have to send some kind of federated query to the system which increases some kind of latency. So he taught me to not focus on these ports, but perhaps create some kind of management layer where all these ports are connected to and that we can centrally manage all these things.

Expert E
Because he's got the centralized view. It's not wrong OK. But it depends. It depends what you want to do. OK. So what we're doing with this observability port, we give the power to the data scientist to check the observability, OK. So for the data scientist it might not be that important. You know which data quality rule has been broken or whatever. OK. But yeah, we have different level. We've got kind of a synthetic view saying OK, like we mimic the Amazon ratings. OK. So based on all the data quality rules and also observability. We have this not very complex algorithm that creates a rating of five. OK. So that's like Amazon five star rating OK. So now the data scientist using our product if it's not that into it can just check hey. So it's a data quality and it's there or it's a 5 or 4.6 or well, this one's 2. OK. So when you're choosing your data product, you want to work with this rating, this system rating will tell you immediately whether it's a product you want to use or not. OK. So and that's what Zhamak in her book called the System Feedback Loop, OK, So this is a feedback loop from the system itself. So at this at this level. I don't really care if it's centralized or not, OK, because I want to inspire trust. For a specific data product. OK, so and see also the evolution of the data of the data quality of my system metric now the data scientist if he wants to drill down into that say hey I really want to know more about why you get this bad score then you can drill down and look at what are the rules that have been violated. OK. So does that make sense to have it at a central location? I don't think so. OK, as a central, as a central location. It's for OPS people, it's for enterprise level people which should have their own monitoring tool over all these databases down or this server is up or whatever. OK, it's not our problem. I'm

not serving them and honestly I don't care about them. I'm caring about my users. OK. The thing is it's really a different perspective on who are you serving. OK. And how you're serving yourself because your data management and your enterprise data management so you've got this big view or are you serving your users? Alright.

Interviewer
Ohh, wow. It's a really good perspective. I haven't, I haven't really looked into that one. So the different kind of perspective the user perspective and more the data management at scale perspective.

Expert E
I'm trying to focus on the user. OK, like being a user champion, I guess so, because when you think of it, you're building a product OK, you're building something. If you don't care about the user, your user will not care about you and will not care about your product. OK? And it's not that. Oh, I want to have 10,000 users or 1,000,000 user on my product. That's not what I'm after. What I'm after is making things in life easier, OK? I want them to be focusing. So the thing is, every decision we make is days on the user and yeah, and then the first thing we did when we started this project is. Is trying to identify who our users are. And that's this persona-oriented developments that we started with, OK and at the beginning we had I don't know 10 different personas and you said really, we really have 10 different kinds of people in this using our product and discussions you create smaller groups, OK. We serve two kind of people. We're serving data scientists, right, data analysts OK and we're serving engineers. So and the data engineers are actually the one buildings that they have products and the and the data scientists are the ones that are on the list, are the one consuming the, the, the data, the data products. So based on that you've got two different set of features. That are coming towards your product. OK. So we first because we've got more users and engineers, we started by satisfying and looking after the engineer, the analyst and the scientist, yeah. And now we are in the phase of hey because we see that there's a lot of traction that the scientist wants more data products. So now we're building tools for the engineers to be able to produce more data products and add it to the mesh. Yeah, but we didn't lose the other way around. The focus was first on the data science on the user. Yeah, the user. Yeah. So, so that's why when someone says, oh, I need to centralize, blah blah blah blah blah. I don't give, honestly, I don't give a ****. OK. So figure yeah, maybe for your career or for your advancement or whatever, it's good. It's not for my user. OK. So no, I understand. So that's why we've got this three sets of ports Zhamak in her book merged the discovery and observation together. I think it's significantly it needed to be split. So that's why in our design we split it right

Interviewer
Why do you think that?

Expert E
I think there's this notion it's all about metadata, but just this notion about passive metadata, OK, things that don't change that much, which is really the dictionary, which is really static, OK. You're not changing it very often and you've got things that are pretty dynamic and that's your observability. OK. So that's why we wanted data quality as result of the execution rules, all those things. OK. So that's pretty dynamic that can evolve differently than the dictionary, so that's why we've got this two sets of APIs.

Interviewer
Yeah, because these ports I also recognized them inside the data product and it's more like this

layer over here. So the observation plane, this is actually also mostly based on your article. So the observation plane, the control plane, the data onboarding and the catalog is, yeah, it's like the discovery plane, right. This is the dictionary in your in article.

Expert E
Yeah, yeah. In the article it was still plane OK, but because I renamed them to services now. Because we have other things which are planes so, so in our current architecture. So we so needed to do things a little bit differently so but yeah I think you're nailing it, OK, so. Yeah, because they also included the immutable chains audit log, which is the thing that * is placing emphasis on. To make sure that there is, uh, yeah, some data lineage. Yeah, and they change data capture which is connected with the event streaming backbone, I mean. Yeah, I may not agree with everything with * says, but yeah. Yeah. So another thing we did is. So we don't. Yeah. So, yeah, OK because you identified it as a problem, but we don't have this. If you look at the component, for example, the change data capture for us, it's not, it's not a component, it's a way of storing the data. So it's the way we store the data. It's an enterprise practice, OK. So it doesn't have to be here, it doesn't have to be done, OK. It's not a pattern. Well, you can say it's a pattern in the way we're thinking, but it's not specific to the data product. It's an enterprise level way of doing things. OK. So it's more like a best practice kind of things. Yeah, I can change that. So it might be different for other people. We don't have this immutable log. OK. I can see why it's interesting, but we don't have it. And then we have two kind of two kinds of storage. OK, we have the storage. Let me see if I have something I can actually show you. I can probably show it to you. It's something I use at a conference, so nobody will tell me if I if I showed that to you. Let me get my screen for two seconds, yeah, sure. But OK, so you should see this. That's where it's slightly different and that's not in in my article. It's not exactly like that but all we could have done it. OK and I think that's kind of our desired goal, OK. You see our services. So I don't have the name of the services anymore here, but I should have put some, OK. So you've got basically you've got observability, you've got dictionary here and you've got the control services here. OK. So. When you look at how we stores the data, we have this internal data which we called quantum DB. OK, so each quantum DB is actually the local store for anything that happens within the quantum. OK. So if the quantum is on our desire to have the quantum completely independent, OK, because it's independent, it needs to have its own instance where it stores its own metadata execution. So it could be something like you were doing the log here. If we need it, we could add it here if we want to as well, right? Yeah, so. So that's this one, and it's independent from the interpretable data. OK. That's why it's, it's yellow, it's gold. That's a precious, that's the pressure stuff, OK. That's what your customer wants. Your customers, they don't give a **** about this part, they want that, OK. So but basically and that's why it's just kind of no arrows towards it because everything is kind of converging to it. OK. So you're making a service call. It's being logged here, you're making, you're running, you're running a data quality rule, it's being logged here, you're having a new customer connecting, it's logged here, etc. OK, so this is really the nervous system in a way, or the memory I would say of the history of what's going into your data products.

Interviewer
Yeah. Oh wow. Yeah, really interesting. Yeah. So the quantum internal data over here, I see a lot of similarity with the immutable log.

Expert E
Yeah but the thing is because we have this storage capacity, we can store whatever we wanted. OK, so now let's say we want this symmetrical change log because our customer demands it or something. Well, we can implement it. Alright, and everything that is blue here. Hmm.

Everything which is blue. Yeah, is independent from the data, so it means that I'm creating a new product. It's kind of a rubber stamp. OK. I'm taking that. ****. Yeah. I'm just configuring it. I'm configuring it for my new, for my, for my new, for my new data. So that's how we designed the system.

Interviewer
OK, yeah. We can perhaps quickly talk about the last framework because I see a Docker over here and well the two main technologies deployment components or how do I call it the deployment parts where Docker and Kubernetes is. But I think there should be many more but they weren't mentioned in the literature. But let me let me share my screen that's more convenient. So I saw the Kubernetes engine as some kind of function as a service, but it can also be an infrastructure as code and Docker can be an infrastructure as code and it's more like a single container design, so really fine grained with yeah data products.

Expert E
So I think that the difference between Kubernetes or Docker is really minimal OK, it's really when you're thinking about it basically it's a container platform OK so whether exactly so whether you're. You're using Docker or Kubernetes being the orchestrator for the container in a way. So I think, there's that or there's something else. OK. So unfortunately or fortunately for us there is we had to use something else because there's ways of doing things at PayPal, OK. But what you have here is good. OK, so basically you can keep in mind is that you want your execution to be as close as the data as possible. Yeah, right. So to do that, the easiest way to move things around is the container. You've got your container. You can execute it where your data sets are. OK, so when you're talking about a small company. I don't think your data mesh really applies with that, but when your company gets larger, larger. OK before working for PayPal, I worked for a company where we were pretty technology savvy, in retrospect. But we didn't have the luxury of having only one cloud, OK? So when you're thinking about it, you've got this really poly cloud architecture and when I'm saying poly cloud it includes this hybrid thing as well OK because the thing is if you still have a few servers, data center, whatever. It's part of your cloud. OK, it doesn't have the same service, it doesn't have the same thing, But it's basically everything in the cloud. So now you're thinking about, OK, how do I have a unified mesh over all my data sources. Then it makes sense. I don't see much of a of another option than using something like containers. OK, because you're going to deploy your same container. Wherever it needs they need to be the underlying system you need to run your container, while it could be Docker Enterprise, it could be Kubernetes, but it could be whatever, or Openshift. Openshift is a great practice as well. Openshift from Red Hat. So you've got, you've got two things. You've got your code in the container and you've got the orchestrator. OK yeah and before Kubernetes you have other things you had Missiles as well OK so you've got so you've got, you've got Missiles, Kubernetes, Docker Enterprise, Openshift all those goes into the category of orchestrator to your container. So that's yeah I think there's no way I think it's similar. It's very it's going to be very difficult to have a successful data mesh without having that. So yeah. OK, I think so because we only have two minutes. I think you're on a great start. Yeah. And The thing is when, when I, when, when we said that I didn't, I didn't have this process in mind, OK. It's because it doesn't start by what type of data product can be developed. It starts more like: what are our users are expecting. Yeah. So more from the business perspective, yeah, absolutely. So, so and then from that, OK, the question is more like. Is the data available OK and then you can have you can have this this you're starting this this more classic data engineering process saying I need to make my data available and then when my data is available how can I turn it into a mesh. How can I turn it into a data product? And when I've got a data product how can I add it to the mesh so it's a

little bit different that's why. When I was a little bit troubled by some nuance. Yeah.

Interviewer
We can add some nuance to it. Yeah. OK. Thank you so much for this interview. I won't take it. You're very welcome. I'll share it with you, the recording as soon as. Yeah, yeah. Thank you so much. will share the transcripts with you

Expert E
You're very welcome. I would highly recommend you look at as the flow of data as well, OK, because you're not really talking about the flow of data there. So basically when you had the lifecycle management, you know. The thing is you've got, you've got the input, OK, all the data that is flowing in your data product and then which is exposed. OK. So the notion of port that you have, but specifically for the input and output, OK, what kind of data can come in? OK. So is it streaming data, is it virtualized data, is it a copy, is it a parking dump, etc. So trying to understand a little bit what's what it's feeding your data quantum and what your consumers are expecting, OK, because that's that's also two categories of ports. I think that would be a neat addition to what you've been doing.

Interviewer
Ohh, thank you so much. Yeah, that's great advice. I will definitely take that into account. OK. Yeah.

Expert E
Thanks so much and then life cycle is a completely other thing. So, but yeah, send me updates and I'll send you feedback on your documents. OK. That'd be nice.. When do you when do you plan to give that away?

Interviewer
Yeah, well. My thesis the deadline is June the 1st, but I also will make two other frameworks, so for the self-serve platform and for the federated governance. So perhaps we can talk about his self-serve platform in a while. It will take one or two months. Yeah. If you want, we can talk in a couple of months again.

Expert E
Yeah, for sure. Yeah. Great. Thank you. We will keep. Well, Tom. Yeah, we'll definitely keep in touch and good luck to you. OK. Bye.

Interviewer
See ya. Bye, bye.