Appendix A

Interview Transcript Expert G

Interviewer
Start this recording. Okay, it's recording now. I will first give you some short introduction about myself and perhaps you can give a short introduction afterwards and then we can deep dive more in the frameworks. I've sent it to you, I think, before, but yeah, it doesn't matter if you didn't have the time to have a look at it. Yeah, no worries. But okay, I'm Tom. I'm 24 years old and I'm doing the Master Data Science and Entrepreneurship at JADS. At the moment, I'm working for Bain & Company. Well, actually, I'm interning for Bain & Company and we're looking at some data mesh migration patterns. And to be more specific, we're really deep diving into the architectural patterns related to data mesh. So yeah, that's in a nutshell what I'm doing right now.

Expert G
Very cool. I mean, it's very a buzzword everywhere. So, the entire data management community is talking about it all the time. Yeah, let me do the same. My name is *. I now work as a cloud architect for the *. So basically, it's a very new art where it's established this January to create a data foundation for both machine data and enterprise data for *. So basically, our positioning is a central platform for data and to provide the, we call it trustee data set for all different platforms and consumers of like persons, like data scientists. It's very critical because we serve the data for machine critical platforms to do diagnostic monitoring, but also we serve for R&D department, which means the innovation on the machines. So, generally speaking, we have been in Data Lake, not the data mesh, but still is needed for business context, which I will elaborate later. Okay. I'm 31 years old. I started, I did my master in *, but I also had a lot of data science track because I think like eight years, like it was very sexy to become a data scientist. So I was trying to be a data scientist. Probably not. So a lot of people want to become data scientist. As I found out, it was not really my thing after my thesis. So I started as a data engineer or software engineer and I was trying to work together with the chief information architect in the *. So he became my manager. I basically worked together with him for another five years. After that, I doing the corona, I felt like maybe it's time to do something else, like I really bored at home, like it's just having teams meetings every day, then I switched my job to *. But I think 85knowledge of skills are transferred, like can be transferred from * to * because everything we do is basically based on cloud, based on the similar architecture patterns, having the same architecture challenges. That's the reason why Data Mesh is serving as a principal to help different organizations to step up on the more decentralized data management world. I do have done several open source solutions with IBM that is linking to the implementation of the Data Mesh because Data Mesh is holding a, I don't know, we call it architecture principles or guidance. It's not directly linking to technology. So we do have done open source solutions with IBM on proving the Data Mesh implementations. That's still in progress. I can use a link

about the open source project in my health from an ITB vendor perspective, like IBM, how they are actually to actually wait, rather than looking at the conceptual models because they are still there.

Interviewer
Yeah, sure thing. It'll be amazing.

Expert G
Yeah. So that's something I can send to you first. Where should I begin with? Do you have specific questions?

Interviewer Yeah, I can just share my screen and then we can talk about the great literature, etc. That I've read and I created some kind of framework. So this is the inter-decision framework. This is like the high level concept. It explains which decisions we need to make during our data as a product implementation. The first one I've identified is what type of data product can be developed. Do we think about raw data? Do we want to expose derived data, things like that? After that, we can make a decision on the creation, so which approach do we want to take towards creating a data product? Do we want to migrate? Or do we want to do green field development, for example? The other one is more the infrastructure layer. So how does the data products interact with other data products with the self-serve platform and of course the management layer and consumers? Then we deep dive more into the data product itself, so the in the anatomy. So what kind of architectural components do we need inside the data product? So within the perimeter of the data product, then we can also look at the perimeter of the data product, so the interface and contracts. So when you think about the different kinds of ports you need, a discovery port, things like that. And most of the time the last decision is how to deploy your data product. So if we start with the first one and go over all these decisions, and perhaps we can, if we have some time, can come back to this inter-decision framework at the end, so we have a clear understanding of all the different decisions.

Expert G
Yeah, can you zoom in a little bit? Yeah, sure. Yeah, that'd be much better. Thank you. Yeah, like this.

Interviewer
Okay. So the first one is what type of data product can be developed, because this also has some influence on your decision-making further downstream the creation of your data product. So I actually identify three options. So to expose your data product as raw data, to expose your data product as derived data, and the third one is to expose your data product as an algorithm, and that can be two variants of algorithms. So for example, we can do some kind of BI system, an optimization-based decision support system, so in BI to like Power BI, or we can create some kind of AI / machine learning model, and on top of that, there can be two other data products. So the hybrid product is more like we want to do, we want to expose raw data as well as derived data. So perhaps we can create two different output ports. That's more like a hybrid product. Any composite product is really like a merging derived data with an algorithm, for example, it's really like the result of merging different aspects. So these are all the options I've identified, and I'm curious if you resonate with some of them or...

Expert G
Yeah, for me, I think the data product is only one of the concepts in the data mesh. It's this particularly being focused for your thesis, or you are also touching other different...

Interviewer
Yeah, this is the first framework in a series. So after this one, I will deep dive more into the self-serve platform. And lastly, we have the organizational aspect, the federated governance.

Expert G
Yeah, for me, those four principles are so tight together. It's very hard to judge one framework without the existence of other things, because one thing is because the data product is basically based on the context that you want to create a domain-driven design and data mesh framework. I think the first tough question for the organizations is you really facing how should I divide the business-driven domain in a more structured way that is really business-facing, really value-facing. I think the data product is just naturally going to be created once you have a proper design domain. Without such a good domain development, you won't have a data-free product framework to be deployed or implemented. But coming with such domain is extremely difficult for big organizations. For small or medium organizations, data mesh is maybe not even useful or valuable. I think the data mesh concept is mainly targeting for big companies. But for big companies, the first big question is how should I divide the domain? How should I make the domain businessdriven? Because it's not only about organizational structure, it's about how should you fund. Basically, you have a bunch of money. You want to create several domains. Each domain should get what money. How should they quantify the domain values? Since usually what will happen is there are a lot of politics on protecting their own kingdoms in the organization, which really slowing down the proper domain design. I think that's basically my first observation that companies are really facing. Creating such a domain will take time effort and top-down enforcement. It's not very easy. Without such domain, you cannot even create a data product because data products will be based on their domain interfaces. So that's my first feedback based on it. I think for you, this framework is already very detailed. I think you have years on no can and cannot. But in the real business context, I think it really depends on the organizational data proposition. Because I worked as * industry, but I also worked in the * industry. Both industries have totally different data organization, data value proposition, as well as data management requirements. For bank industry, it's highly regulated, which means it's policy-driven. Everything they do around data is basically being written out by the central bank or ECB. What you can do, what you cannot do. And those things, you really cannot be framed in your framework, because there is already a very detailed framework from central bank on how to do things. It's not like you can create a data product and do this, do that. It's basically that central bank or ECB tell you, this is the first step you have to do. It's the second step you have to do. This is how you generate your balance sheet. And that is basically the whole finance and the risk framework guiding users. But the * industry is less regulated. It has so much freedom. Basically, you can do whatever you want. It's purely focusing on the product innovation, right? If you look at *, we make machines for making the *. We only make several types of machines. So we don't have a diverse portfolio on our products. We probably only have, let's say, a term-made product. And they actually all create very standardized data sets. We're not having 1,000 different data products. Probably we're just getting term-made streams of the data product from the machines, even though each machine sends all hundreds of different data sets. But still, they're all standardized, which means you can actually build one standardized interface to collect all those data. Coming back to your question, if you think in industrial, you can build one interface to collect the data. And the data can be standardized. You actually can easily build a raw data layer to make sure you have such a physical storage layer to put the data together. Since collecting the data is not really issue, but for banking, it's an issue. But for *, it's not. So the first layer of the raw data is usually depends on how do you develop. For bank, it's different from financial industry. For banks, it's different from *. For *,

the raw data lake can be centralized. You can put them in one bucket, but for banks, you have all local banks. They all have their own small data lake. It cannot be physically centralized. So you can see the deviation on the raw data lake design. Banks are doing in more domaindriven methods, but * is still doing more centralized data lake-driven. So that's basically the business context that you have to think.

Interviewer
Because these frameworks are really specified on the architectural design principles. Not really the business logic.

Expert G
But they are correlated. With all business context, those frameworks, you cannot really return values for architects to adapt. It's always business correlated. Because even as an architect, your stakeholders are business stakeholders. You have to do your value proposition, what can benefit them. I think without business proposition, it's very difficult for me to charge your framework. It's right or wrong. Because I don't have a context to charge. That's why I'm giving you the context of the * and *. Because they all have their own context. I think it gives you a specific direction, whether your years and no, in that context, can be applied. Yeah, clear. That's you can... Yeah. I think I expose the data as raw data layer. Raw data practice, that's basically your upper one I already touched. As I mentioned, for banks, you have no choice. Because it's the business structure. If you think of the bank, each country has their own central bank. Like Belgium has their own central bank. They have their own policy makers. They do things with their raw data. For Dutch, they do their own raw data processing. In a domain treatment design, they are actually already creating raw data products in their own business structure. Which is policy driven once again. Because they have such business value to be created. That's reason why I always say business value. Because it's always value driven. But if you see ASML, we are not distributing the raw data layer. We're centralizing it. Because that's the easiest way for us to create a value. That's the reason why in other actual data match concepts, we are not exposing the raw data layer into different domains. We are creating such raw data layer as a single standalone. Isn't it already?

Interviewer
Perhaps we can go to the orchestration decision famous. Because this is really much related to, for example, master data management, where you centralize everything.

Expert G
I was referring to all those concepts. They are all into practice. Each other is particularly one. Another one is very difficult.

Interviewer
Yeah, sure thing. Because in this framework, we segregate on migration and greenfield development. But for example, these patterns, most of them can be implemented at the same time. Do you know Strangler-Fig, for example? I don't think so. I quickly show you. Because it's really about decomposing your monolithic architecture. At the same time, building up your data mesh architecture. I really like this picture, for example. You first have your monolithic architecture. Then you decompose a surface out of this monolithic architecture. But you keep your monolithic architecture at the same time. And you grow your decentralized architecture. As well. Evolutionary way rather than revolution. Yeah, it shrinks over time instead of just doing everything at once, like in greenfield development. Just starting from scratch. So this Strangler-Fig. Yeah, master data management. I think that speaks for itself. Zero trust archi-

tecture. So that we make sure that every data product has its own access controls. And CQRS is about segregating the read and write function. So that when we read something out of the data product that we don't overwrite stuff in there. So those are the form migration patterns I observed in data mesh. I don't know if you recognize them.

Expert G
I don't think there will be big bomb migration for all the big companies towards data mesh. If you think about the basic contact, you don't want things to be disrupted. Because you have products to be sold, you have the organizational, I don't know other priorities as well. I think what I usually see like * is taking a completely different approach versus *. So * is more evolution approach, like the Strangler-Fig part that you mentioned. Because the banking business is coming back to the business again. It's very policy driven. We have to make sure every month we are creating the risk report to the central bank to make sure we have enough capital buffer to differ the risk, otherwise all the deposit problems will be gone in the financial crisis. So those things are very static. You cannot change over days. But for * is different. For * we create the new whole organization. It's called digital platform. So this new digital platform is posing itself in a more disruptive way. We are going to create this new digital platform, the data mesh way. We are going to create the domains. We are going to create the data ownership on the domains to make sure we have such structure facilitating this transformation. So it all depends on what business value that this data mesh is going to bring. Whether it's impacting the business continuity allows how it brings a value in a faster way. You can see the value that this data mesh is creating. So the master data management is very traditional already. It's like 20 years. Zero trust architecture. I think it once again tied to the business. If you have a banking industry then probably you need a zero trust among different local banks. I'm not saying within one bank because if you think it's a group, it's not only in the * bank. It also has a retail banking, wholesale banking over 50 different countries. They have, I don't know even like a retail banking in *, they are all on their own legal identity. Which means probably they should not trust each other by sharing the data, or sharing the data product. Especially if a country has a very strict personal data protection policy. Sharing data is not allowed across the physical border of the country.

Interviewer
And that makes your data product more autonomous. If they all have their own access policies.

Expert G
In that business context, zero trust architecture is more relevant. But for *, we don't have such requirements than we don't have.

Interviewer
Okay. Okay. Clear. But at *, you did more some kind of greenfield development. You started from scratch.

Expert G
It's basically the whole company. Both members decided we're going to establish a new organization. It's called Digital Platform Disruptive Way. This new digital platform is going to deliver the whole data value chain for * in the next five, 20, 10 years. And all the organizations will follow. So they appointed very high level manager to do that. But for the bank, I will never imagine it can happen. Yeah. Okay. It depends on the business context to look at the actual deployment model. Like what kind of business value is actually bringing? Yeah, exactly.

Interviewer
So if we think in infrastructure terms, because we also talked about infrastructure as well a bit, this is a more extensive diagram. Let's just start on top. So we have the schema registry. But schema registry actually does. It's really connected to the central data product catalog and the event streaming backbone. So when something is flowing through Kafka, the event streaming backbone, it reaches the central data product catalog. And the schema registry makes sure that this event is converted in a more user-friendly format, you know, so that the user can understand what's going on. Because yeah, the event streaming backbone, it's written in a less user-friendly way. So yeah, that would be how data products communicate and the central data product gets of course, very important in terms of discoverability of your data products in the mesh, etc.

Expert G
Yeah. But if you look at the schema registry, I think the most important feature is a schema enforcement, a schema evolution. So basically, if you, so schema itself is like a handshake of the data contracts, like how the data format should be like, like what this is, I have agreed to be shared with you. But the most important thing is if you change the tiny thing, all the downstream will be impacted. And that will happen very often in the technical interface, which means, I because business is involved, which means the schema will involve, but what kind of agreement that you should make with your downstream consumers is controlled with this schema registry on the schema, we call it enforcement or evolution. So yeah, but this is already very technically contacted with with this federated data governance.

Interviewer
Yeah, yeah, indeed. So yeah, everything is overlapping a bit, but this is what I found in gray literature related to data products. For example, there can be a shared storage and some practitioners mentioned the pitfalls of shared storage, because when you have two data products, for example, and they both have a shared storage, one of them is getting versioned, the other one has to wait until the first one is versioned, because yeah, they are somehow connected with each other. So this really drops the autonomous ability, you know, so there are some pitfalls, but at the same time, another practitioner mentioned the fact that if we would have 10,000 data products and each data product would have an internal storage, then we have to manage 10,000 different storage accounts. So there are pros and cons with a shared storage and with internal storage.

Expert G I'm just curious, like, because you are doing it for Bain, do they actually offer you a specific industry case for you to study? Or they just threw you a very theoretical assignment? Because for me, you can build this framework for financial industry, but they cannot be applied for chemical household industry, because all different business contacts do you think?
Interviewer Yes, it is actually meant for startup within Bain, and this startup is called Pyxis, and they are involved in transactional data. So they have e-receipt data, and they have credit card data, and they try to draw some kind of customer profile with it to support the case teams within Bain. And yeah, they would definitely benefit with data mesh, because there are so many different operational databases when we think of credit card data, etc., and e-receipt data. So there are so many different operational databases, and they can perhaps create different kind of data products to classify customers, for example, in a specific way.

Expert G
I think that you can make it more, I don't know, explicit, financial business contacts. I think it's allowed to be making it more stronger framework.

Interviewer
Yeah, the framework is meant as a guideline for every petition, so it's not really focused on one particular area.

Expert G
But that's also the danger of the framework, if you are too generous, because you are creating a framework, I don't think it's already framework, it's already very detailed, like execution plan to do things. So that's the danger to make it a search for framework. What is the boundary of your framework? Are you going to touch the implementation already, like schema registry, or you are floating on top of the implementation?

Interviewer
These are all options. So it's not mandatory to have an event streaming backbone. You can, for example, do more things like batch, that you do some batch processing, like putting your data, the data that flows out of your data product in a Query table, or in a blob storage, things like that. So I should mention those two as well in this framework. Now I'm thinking about it, because now it's really focused on streaming data, but of course, you can do batch processing as well. And there can be different kinds of APIs. So these two are not kinds of APIs. These can be REST APIs, for example, but if you focus on these three over here, so we can choose graphQL, GRPC, and REST APIs. Those three were mentioned the most by the practitioners. I don't know if there can be more. And yeah, for example, the SQL access point was really relevant, because most of the data analysts nowadays, well, not only, but they prefer SQL over other methods. So that was a really important aspect to implement in your in your data products, to make it accessible for the consumers. And on the bottom, we have more like non-functional requirements. So what kind of security controls can we implement, like fine grain access control, and can even specify it further. In memory catch, and on the bottom, we see a query catalog, and a query catalog actually is some kind of manual that's where all the possible queries are stated. So the data analyst knows which query he or she can use to access the data. So it's more like a dictionary. So these are all the things I encountered when checking the infrastructure layer around the data product.

Expert G
It's not directly related to the source of platform, but it's more like what's going around in between. Clear, I think those are very, I don't know, outstanding cases, like typically you see like API, SQL, events. But it would be a question because those are actual interfaces. You can have as options. Yeah, these are options. But what is our intention? You want to come up as complete as possible.

Interviewer
The first one. We want to show practitioners what their options are when choosing this route of data mesh.

Expert G
But it will be requirements driven, right? Like what is the case for streaming? What is the case for SQL? What is the case for batch or event?

Interviewer Yeah, so you think there needs to be an extra decision over here, another one?

Expert G
Yeah, because you give people options without giving them why you should go to this option.

That's the value of the framework rather than showing the option itself. Yeah, if I'm a customer, I'm your client, like you are giving me a voting case. I think the help that I need is from your framework that I know which path I should choose rather than having a complete path.

Interviewer
Yeah, definitely. I should still write context now.

Expert G
Yeah, because the streaming then you have a boundary like you have you firstly need to have the sources stream like you in the bank, you have a quick stream or transactional database. They are real time database as a source. And you have a consumer that actually need to take actions in milliseconds like for detection. Then you have a driver for guiding them to go to the even streaming backbone. Yeah, that is self explainable. Then you if you have API, then it would be more different. I don't know.

Interviewer
Yeah, that's a really good point actually to focus also on the sub-decisions. I can call it like that.

Expert G
I think the value of this framework is helping people to make decisions rather than the whole landscape because that you will never get there. I don't know. I didn't even do solutions, but there are also solutions. Thousands of them. Yeah, it's not focused on technologies. Yeah, because even for events, you have different schemas can be used. You can use JSON. You can use text, use binary, of Avro. Then where people should go, you will never get a complete list. I think the best value you will offer is helping them with the decision making in the context. Those are the check list you should get. You have a streaming source. You have a streaming case in the end. I agree. You have a streaming data engine like Flink or Spark streaming, those kind of things to facilitate your event streaming. I think that your business clients can judge if it's a fit or not. I think all of these things are quite already complete. I don't think you should add more, even because I think it's not really adding more value sometimes. If you can cover 85successful. The corner cases should be always out of the framework because you cannot cover all the corner cases.

Interviewer
Yeah, that's a really good point. I think in terms of time, we can quickly go on to the data product itself. This is really what's going on in the data product itself within the perimeter of the data product. We already discussed the streaming line, how do we call it? Streaming data principle. I should also focus on the batch processing as well, because the change data capture is really related to events streaming. Immutable change audit log, what this actually does is keeping track of all the transformation processes within the mesh. If something changes, the change data capture sends, well, publishes the event in the event streaming backbone in Kafka. All the data products are subscribed to a specific topic in the event streaming backbone, and they will get notified. It will be stored in this immutable change audit log. When the data is transported to another data product, this data product already knows about the changes. This is also a bit like a data lineage, keeping track of all the changes and storing this. That's the immutable log, and immutable because it doesn't change. It's like an append-only. The internal storage is, I think we already covered this quite a bit. The data catalog, so we have a central data product catalog, and each product will have also its own data catalog to show the metadata.

Expert G
We already call it small like technical catalog to various metadata with data gravity. It's very important because sharing data is expensive in the organization. Not business domain should actually be encouraged to share the data, especially between different cloud vendor solutions. The data catalog is usually working as data steward in subdomains to make sure all the metadata are captured and all the data governed within the domain.

Interviewer
There was also someone who mentioned that we should separate the data catalog and some kind of meta store. That's so that we should have both. A meta store for, for example, active metadata. What's active metadata? Active metadata's data that changes over time. For example, schema versioning is a really good example. We have a local data catalog. He mentioned to fact that he doesn't store all the metadata in the data catalog, but he also separates this in the meta store so that we should have both. I don't know if you agree with that.

Expert G
It's basically the fact that all the companies have multiple data catalogs because most of the IT products are coming with a nested data catalog. If you buy IBM suite or you SAP, they all have their own data catalog to capture the metadata naturally in a native way. There is no way you get rid of them because it's better for the product itself to manage the metadata. It's like I mentioned, it's just gravity. There will be data staying in IBM or data staying in SAP.

Interviewer
It's more like the data catalog inside the data product is more like a static thing. It only contains metadata that doesn't change and the meta store inside the data product would kind of feed the data catalog.

Expert G
It depends on what kind of metadata you're talking about. It's the technical metadata or it's business glossary like business metadata. So business metadata is a more about definition of your data. Then on the business data, you can have the technical metadata representing that business metadata. Are you going to have multiple business metadata catalog? Or you have one business metadata catalog linked to 10 different technical metadata catalog. So it depends on your governance setup.

Interviewer
Yeah. And what would you recommend? Which will store the business glossary in site data catalog or in the meta store?

Expert G
If you look at the big companies, their preference is to have one metadata definition like what is the definition of customer? Because it can be really translated to different things within the big organization. Like taking the example for the bank, the customer from mortgage department is completely from wholesale banking doing the deals with big companies. The customer entity is completely different. That's the reason why I think they all prefer to have one centralized business glossary definition. But then who has the decision power on making such list is also very important whether you have a got the company's body to execute on it to make sure everyone great. This is the definition customer. This is the database that I have. I will link my this table, customer table to that business glossary definition. If I have a wholesale banking customer, then we are wanting to a different business definition. That would be the complete

ideal world for companies. But when it comes to the implementation, it's becoming difficult and compact.

Interviewer
Yeah. Okay. Because now we would have three kinds of meta data stores. So we have the data catalog inside the data product. We have the meta store, meta data, meta store inside the data product. And we would have a central data catalog. Perhaps I think that the data catalog can be removed and that we only have the central data catalog and a meta data registry inside the data product. Because I think you have to think from the sales perspective, if you're IBM, do you want to get the rid of your product and create a value for another product? It's becoming protecting their own business kingdom, protecting their business offers, which means they will try as hard as they can to make sure their catalog is fully embedded with their data products. If you buy IBM project, you have to buy their data catalog. There is no way back. So then it's becoming the actual implementation perspective for the companies. How should I implement it in reality? That is definitely very difficult. From architecture, everything can be a dream. By the way, it's coming to reality. Nothing is working. That's basically the quality of being architects.

Interviewer
Yeah. Yeah. That's the cost benefit aspect as well. Yeah. Okay. And if we look at the last three over here, we have the observation plane and a control plane. The observation plane is really for the data consumer that he or she can observe the data quality inside the data product. And the other one is the control plane. And this one is related to governance so that we can enforce certain policies within data products. Yeah. And yeah, the last one is data onboarding. So this really about ingestion and transformation, et cetera. Yeah. And the interface is actually really related to these three. So the data catalog, the observation plane and control plane, because I observed an observation port, a control port, and a discovery port. And of course, there would be the input port and the output port. I forgot to draw them here.

Expert G
Can we give a little bit of a bigger because I should not really see the tags?

Interviewer
Yeah. And one of the practitioners mentioned that all these ports can have negative influence on your data management, because when a central entity wants to check for the data quality inside the entire mesh, well, then he has to write some kind of federated query that is going along all these data products, all these different data products, and that would increase some kind of latency. So he mentioned the fact that we can have an overarching management layer where all these ports are actually integrated and that we don't have these ports on site. Well, yeah, on the data product perimeter. So there are two perspectives, actually, do we want to make the product data product as autonomous as possible with the with these ports. Or do we want to take more centralized approach or more some kind of data management at scale approach where we choose for an overarching management layer?

Expert G
I think the answer to you without business context is impossible. Then it depends on this case, like what kind of data you're talking about. Are you talking about the customer data? Like really sensitive that you have to have a port to intercept what they are doing with customer confidentiality. Probably in this case, you want the most centralized way, but if it's just technical data, like, I don't know, click streams, like you are looking at the customer behaviors,

there is no sensitivity to engage. Then probably you don't care, you don't want to implement such centralized way. You don't want to have overarching thing to command you on doing the behavior analysis without confidentiality. Exactly. Did you encounter different kind of ports during your implementation? It's very difficult to answer. So, first,

Interviewer
If it's too confidential, that's no problem.

Expert G
No, there is no confidentiality, because everything we talk about is very like, very high level. Yeah, it's like impossible to give a concrete detail. Like, like discovery port is definitely valuable to have centralized way, because then you want everyone to discover what's over there. Or using it, then it's more making sense from business perspective, but control plane or control port, it depends on what exactly you are controlling. Like, I'm controlling the frequency, control the governance.

Interviewer
It's meant for the governance team.

Expert G
Governance can be like, the * governance manual is like 200 pages about governance, like what kind of governance actions you want to control, what kind of governance action you don't want to control. Then it all depends on the details that can be controlled or what can can be or what cannot be controlled. And observation port, of course, you won't have the footage that the lineage to be generated. I think it depends on whether those lineage you are generating is valuable for the business. Like, if you are a business owner, you don't want your employees. And like, I don't know, one, two days per week to generate data lineage. It's not going to sell anything with the data lineage. And probably the central bank people, they come to the bank, they say, you're a beautiful data lineage, it's perfect. But that means you are spending money on generating such things. So then it depends on what kind of observation you are capturing. Are you capturing high level data products contracts? Are you also capturing the real data lineage? Like, what kind of transformations you are doing with your data?

Interviewer
Yeah. Yeah. Yeah. The business context is so important around these kind of decisions.

Expert G
Yes, exactly. Because everything in the end, you are doing business inside of the organization, even as a data governance or data lake or, I don't know, data product team. They are all doing business within the company as well. I think everyone has to sell something. If there is no value for other teams, you won't get the buying for the things that you want to do. So that's basically the dynamic dilemma of the architects.

Interviewer
Yeah. Yeah. Yeah. This brings us actually to the last one. And that's the deployment decision. I encountered three options and I shouldn't choose proprietary names here. I should choose container orchestrator and containerization over here. The function as a service can, for example, be a third option. Yeah. And of course, the lake house architecture can be used to quickly generate data products. Yeah. These are actually the options I saw. Did you use some kind of containerization or were you more like using a lake house architecture and building data

products on top of that?

Expert G
Yeah. For *, we're combining the two. Okay. Or a * bank, a more containerized word. That's basically tied to the introduction that I did. Like for *, we have the standardized raw data collection, which means you can actually build a lake house as a serving layer. And on top, each domain, they can build their own containerized data product to consume the lake house. So it's not like they're conflicting with each other. There can be a combination of the two. But for *, the data sources are so diverse that it's very difficult to have such lake house to be in delta format. Yeah. Because the data is so diverse, the interfaces are very locally tied, depends on the local banking technology stack. So it's more containerized. Or API driven, they build APIs to be served with each other. But inside is more like microservice architecture. You have your own containers, you expose your traffic outside with APIs. All right.

Interviewer
Yeah. That's about it, I think. If we go back to the last, yeah, to the inter-decision framework. So these are the six decisions I encountered. But do you think there are more if we look at this diagram?

Expert G
I think the important thing, if you ask me if I want to say it, same as long as for framework, if I'm a customer to buy your framework, the most valuable thing is helping me with decision making on the framework. When should I make what kind of decision? That's something you are going to sell without decision making facts. It won't be really valuable for your end consumers. Because if you want to sell your framework, they have to get the value from it. Keep that giving options does not bring value. But helping with decision making, that's a valuable thing, valuable thing without framework. So I think it would be really nice for your framework to make sure you are detailing out for each step. Why you are making this decision? Because you give people options, then you have to explain why those options are exist, in which scenario you should go to which option. I think that's the most important thing of your framework.

Interviewer
Actually, actually, I can show you something really interesting because that's what I have been doing. I haven't had the chance to show you this. This is what you mean, I think. Sorry. Current slides. There are too many slides on this one. Current slide. This is the framework. This is again the inter-decision. It's like an overarching table where we can observe the decisions we just mentioned. So these are the evidences. So the articles I read and these decisions were mentioned in there. So ten distinct articles. These evidences are related to this solution, for example. There are forces. You mentioned actually the forces. When we choose, for example, for an immutable change audit log, and we look at force 25 over here. The pointer. So force 25, I observed this as positives. These plusses and minuses are like interpretation of the article. When we look at force 25 over here, we see traceability. So choosing for an immutable change audit log would increase, well, would have a positive impact on your traceability, on your data architecture, traceability, for example. So is this more what you mean with the why question?

Expert G
Yeah, but I think this is not very customer friendly, let's be honest. Yeah, it's a bit complex. Maybe you can use it as a theoretical framework from academic perspective, but you are not going to sell with your real clients from paying out from marketing, I don't know what kind of clients you are having. Otherwise, it will be like, if you're asking a director to see the stable,

they will go crazy with the headache. It's perhaps too extensive for practitioners. Yeah, but I mean, again, I think the most valuable thing like the diagram you are creating, I think there are logics in aids, but you have to enhance the logic on the diagram to make sure people can actually follow it with the decision making process. Yeah, because the next step would be to write a context about each decision framework. You should, yes. That's something that's a great value.

Interviewer
Yeah. Okay, but you don't think I'm missing out on some decision or do you think?

Expert G
Yeah, I think there can be. So I think quite enough, because I like, I don't know, I like simplicity in the business, because that's something you can easily sell if you bring everything on the table. Yeah, I don't know, just from selling perspective, I'm not going to buy it, because I don't say the, I don't know, like MVP or like unique selling point. I think it's already quite good at the starting point. I think you already know almost everything on the generic framework. I think it's very important to detail the business context to make sure it's based, I mean, for potential readers, it can get the, I don't know, reasoning behind it. So like why, yeah, for in different parties.

Interviewer
Okay. Yeah. Thank you very much. Thank you. Thanks for the, yeah, I think we covered everything. I've actually, one last question, because I will be developing the self-serve platform framework in like two months, and then we have some extremely like this. Can we do perhaps a second, second discussion on that one?

Expert G
Yeah, because I've also worked on self-serve platform. And now I'm also working partially on self-serve analytics in here as well. So hopefully I can help. Great. But let's see. Okay. But please, I'm saying a lot of things about the organization, which I should not be by just purely for your own, like, I don't know, reference perspective. Yeah. So keeping the recording for yourself, do not distribute because

Interviewer
no, sure thing. No, I have sent you an informed consent to make sure that everything is anonymous, and after transcribing it will be deleted. So okay, please, and the transcribing is just with not your name, but it's like expert D or expert E.

Expert G
Yeah. Things like that. Because I'm saying a lot of things about companies and they are all very sensitive.

Interviewer
Yeah, sure thing. No, you will be safe. You will be safe. Okay. And thank you. One last question. The IBM framework, which presented to me, I think it will be really valuable for implement. No framework is actually technology implementation. I will share with you. Okay. Thanks, gone. See you. Bye bye. Have a good weekend.