Appendix A

Interview Transcript Expert C

Interviewer
OK, let's quickly start because I have some frameworks to discuss. I'm really excited to show you some. Yeah, I can first give you some kind of small introduction about myself. OK. I'm Tom, Dutch, and I'm doing a master in data science and entrepreneurship, so it's a bit like combining the business perspective with the technical perspective from data mesh. And I'm especially focused on data engineering practices. So if we deep dive more into data mesh. That's what I really like. So more like everything that's going on before the data scientist. Everything that is happening before. So yeah, that's also interesting to know before being involved in data mesh, well data science, I was really interested in Industrial Engineering. So I come from a business background and yeah, I think that's also why I like data mesh so much because it's somehow organizational, but at the same time there are some technical components in it so I don't completely shut off my other expertise, but I like to combine data science and the business perspective. So yeah, so that's a bit about myself. Can you perhaps give a short introduction?

Expert C
I, like you, studied engineering in my in my undergraduate studies. I studied chemical engineering. Umm, and I didn't want to work in an industrial setting, so I worked with * at the * in an operations and technology rotational program. So I learned about data early on, but mostly around how to manage delivery of data and how to manage systems that that build data. There's a lot of project management, program management. I got my MBA at Duke. Um for two years, studied strategy and went into consulting after my MBA. And then found out that I was consulting quite a bit with *. Many of my projects were there. So I lined up just converting to a full time employee and I've been focusing on while I'm at the bank, I've been focusing on production stability of our core platforms, strategic road map, strategic road map planning, budgeting and the capital processing cycle. And then this data mesh project turned out to be a key item on our strategic road map, so back in 2021, we started this this effort to build a data mesh. We had some third party consultants come in and help us map out the strategy. We have a whole organization focused on data governance and data management. So we're building this thing simultaneously. That's where I sit in the organization and the governance. Yeah, enterprise data and analytics part of the organization, which has governance.

Interviewer
You're trying to connect all the pieces together, governance, data management.

Expert C
I'm the product owner, product manager, so I sit between the business representative for the mesh and I sit between enterprise analytic and technology.

Interviewer
No, that's what happens to a consultant quite often, right? That you end up working for one of the clients?

Expert C
Yes, quite often.

Interviewer
Nice. Well, first of all, thank you for participating in this interview session / discussion. Really looking forward to discuss all the models with you. So let me share my screen real quick. Can you see my screen?

Expert C
I can.

Interviewer
Great. OK well let's first start with the high level overview. So I constructed the framework right here which covers all the decisions you need to make for implementing a data as a product. Well, in my perspective, I think it's useful to quickly introduce these decisions first and at the end that we decide more like is this the right way or are there some missing decisions over here, etc. So yeah, the first decision in my perspective you need to make is what kind of data product do you want to develop? So what type. There can be multiple types in terms of derived data, raw data products, algorithms, BI tools, etc. Next we would choose the approach show up, and what I mean with approach is do we want to start from scratch, so Greenfield Development or are we more like we currently have a legacy architecture and we want to migrate towards the data mesh, so that requires some different kind of solutions. At the same time, we need to think about the architecture around the data product. So how does the data product communicate with other data products, communicate with self-serve platform, the management layer and of course the consumers. Furthermore, we will deep dive more into the data product anatomy. So what's happening inside the data product and at the same time we can think about the interface. So what I mean with interface is more like what kind of ports do we have on the perimeter of the data product, how can we make it accessible for the consumers, data analysts etc. And at the end, which is often the end of the software engineering cycle is how to deploy the data product. So that's more on the technical side. And if we start with the first one, so the data product type, I identified 3 main types of data products. So the first one would be to expose your data product as a raw data. The second one is to expose your data product as derived data and a third one is to expose your data product as an algorithm. And we can think of two variants on this. So for example the optimization based decision support system, so BI tool like power BI or we can expose the data product as an AI / machine learning model. And there can also be some hybrid products, so where we use the raw data as well as the derived data or the raw data as well as the algorithm. These kind of variations. And yeah, of course we can also create some kind of composite products. So my main question about this framework is what do you think of this framework? Uh? Are these other options or have you experienced some other options as well?

Expert C
Some of the things I'm interested in what are you looking to do with these frameworks? I have a feel, but I just want be sure.

Interviewer
These frameworks serve as some kind of guideline to help you think of all the architectural design decisions you need to make while implementing a data as a product. So this is just one of the frameworks in a series speccialized on data as a product. Later on I will create a different framework specialized on the self-serve platform. And the last one is about the Federated Governance. So all these frameworks are related to the data product principle.

Expert C
So are there assumptions going into this where you have something in mind like where, where are we in terms of like are you starting from scratch with data? As a product or are you starting with a framework that is assuming you're on like in a mesh?

Interviewer
Yeah, this is actually the second framework. So you mean like we're starting from scratch like Greenfield Development or that we already have a master data management, yeah, component in architecture?

Expert C
That's where I was starting if you go back to the first, the first decision. The inter-decision. Before I think about what type of product can be developed, we think about it as well. What's the purpose? Why am I building it? What's the use? Is there a consumer? Is there a need to build something? I guess I view it as a product. You're creating a product that you can mark it with value like if you're bringing something to market, then you want to have that you know be distinguishable, be differentiated and kind of have the all the critical elements of something that would be in demand. Yeah. So what I think, I think start with demand then you think alright what is, what is the consumer's need. Then you say, well, what then? Then it starts to be more about the conversation of what that product looks like, where it comes from. How you develop it. So that was the first thing that popped to mind.

Interviewer
Ohh OK, but do you also mention that this framework should actually be implemented before this one so that we first need to consider the orchestration decision. So do we want to start from scratch or do we want to migrate from a legacy architecture? Do you think that should be considered before?

Expert C
Well, I think so. I think the sequence here is, is something to be considered because just like you bring any product to market, I mean we're treating them, we're treating our data mesh as a product itself. So with users with the user-centered design focus that says. I have a need and my product meets this need. So it's not if you build it they will come but it's building for a specific tailored need or to fill up a spot in the market or just like if you were taking something to market you'd market size it and say here's my addressable market for this type of product. Here the demand is going to create. I would start there so then I think if I have that, if I start with the why and the consumer piece. Then I start to think well how would I do this? Do I have a fabric? Do I have a data mesh? Do I have? Is this all assumed to be on cloud platforms? Because, you know, you think about, OK, a product exists somewhere where does it exist, how is it accessible and how does it have all the critical elements of the data product like accessibility descriptiveness, the metadata associated with it and you know it's got to be searchable and discoverable. So I've this ecosystem mindset because we're building this. You know, set up set of technologies and processes in place but I think that's where from an

industrialization standpoint that's where we're starting.

Interviewer
Yeah. Ohh, that's a really good consideration indeed. So perhaps I started too early with the data product type before even mentioning the why question. OK. So perhaps we can just start with this framework.

Expert C
Let's do it, let's do it.

Interviewer
Because I identified 4 migration techniques, there can be more, I'm sure of that. And of course, the Greenfield development approach. Have you encountered one of these or how did you start with your data mesh? Was it from a migration perspective? Or was it like starting from scratch and building everything from the ground up?

Expert C
We did primarily Greenfield, but when I think about, let's just talk about each one of these. What do you mean by master data management?

Interviewer
Master data management is really centralized. So it's a program like Snowflake where we have centralized data warehouse where, yeah, where the consumer is taking all the data from really considered on.

Expert C
Then what's the strangler fig.

Interviewer
Strangler Fig is a technique that's slowly decomposing your monolithic architecture. So piece by piece. And at the same time you're building your data mesh architecture. Perhaps I can show you a picture that's much, much easier to explain. So well, you're decomposing your monolithic architecture. You slowly build up your data mesh. So one component at a time, one service at a time, not a component. Strangulation pattern. Yep, yeah. It's like doing everything piece by piece over a certain period of time.

Expert C
Got it. OK, then let's go. Let's keep going through. This is really helpful. Zero trust.

Interviewer
Zero trust is that each data product is really autonomous. So to get access to this data product there are certain access controls and it doesn't mean that if someone can access the entire mesh that he or she can also access each data product. So you can only access the data product where you have access to. When we say there is a customer service domain and a sales domain, someone from the customer service domain only has access to their data products and not to the sales data products.

Expert C
OK then CQRS.

Interviewer
CQRS is about segregating your read and write functions, You maintain some kind of stateless. Uh, let me again show you a picture. Pictures tell us more than a 1000 words.

Expert C
I'm a visual person too

Interviewer
It's really by separate segregating the query and the commands service. OK, you'll have a read and a write function.

Expert C
OK. And the last one? Yeah, that's the Greenfield. How do you define Greenfield in this space because my initial assumption has changed after these types of conversations.

Interviewer
Greenfield is just starting from scratch and that you don't depend on your current legacy architecture, so like starting everything without restrictions.

Expert C
So this may sound odd, but we honestly have the first three. We're implementing those in different places today. CQRS uh, I'm still a little foggy. I have to think more about it. But we started this thing totally Greenfield. We had no applications. We had a use case in mind, but we were building from scratch. But we decided to build on Snowflake. So we're looking at a snowflake warehouse that serves as our central storage and transformation. Well, location and the first data product was more of a zero trust architecture in place with access controls because security is a top component. So we have all of these patterns in place I would say security runs the game like every product we build is based on zero trust architecture. We have a central pattern on snowflake. We're looking to potentially do a secondary pattern on Azure native tooling. So that would give us just different economics, different capabilities. Folks would be able to deliver a product on Snowflake and a product on an Azure tool that's interoperable. Like we have to figure out that solution as well if we're going to have two channels. Umm. And again that Azure pattern would be Greenfield because like none of our legacy technology sits on Greenfield. So the challenge then becomes porting your data from legacy or you know operational stores to the cloud. Uh, so we're building patterns that do that. And, and I think one of the patterns is the Strangler-Fig pattern, where we're taking them monolith. And we're looking at decomposing you know decades of logic and uhm queries that kind of come together to create this reporting engine and we're breaking down that reporting engine and saying all right for one use case, here's what it's going to be. We're going to move that use case of the mainframe technology and on to the cloud and we're going to decompose that a little bit over time so that eventually most of the traffic is engaged with the new platform and then the other one can be put to sleep. So I do see kind of each one of those pieces with the exception of zero trust, all of our products are limited or control based on you know fit for purpose consumption. So the producer has access and the data owner and our governance model provides access to the consumer. So you've got a shared ownership model. The custodian provides access to the owner, who provides access to the consumer.

Interviewer
Yeah, yeah, perhaps I constructed this framework a bit wrongly, because it does seem like you have to choose one of these options. But of course you can choose two at the same time, like

the Master data management and the Strangler fig. That's really good feedback already. OK. Then we can perhaps go on to the next one. So the data product type, we briefly discussed this one already but do you? Uh, yeah. Do you recognize these three options? So the exposing data is raw, etcetera.

Expert C
I do and we've debated about this in our situation where you know, if you expose something as raw there could be two different ways to approach this. If you expose it as raw, then you have to use your storage technology to do any transformations that are required. Um, which in our case is Snowflake. What's your definition of derived?

Interviewer
Derived is doing some small calculations like returning the top ten highest grossing customers, et cetera, OK.

Expert C
And that's done in advance of a product being published?

Interviewer
Yeah.

Expert C
OK. So when I think of derived the transformation part is here we look at? You know. I'll speak into like 2 lenses again, the raw product, all data untransformed? A derived product could be multiple raw data sets untransformed or pre-transformed data. So if you're taking data from a warehouse that exists in a system that's difficult to access or it's on premise and you need to publish it for analytics. You publish it to the cloud. Well, maybe that data already has business logic that's built into it. From a lineage perspective before it hits that warehouse, the on Prem warehouse. So you've got raw data, you've got a warehouse and then you've got a published product that's a warehouse that's duplicative, but it's duplicative for the sake of taking it off of a system that's that that is difficult to access. It's publishing it once for mass consumption because we find point to point and interfaces with our legacy systems and rather than point to point interfaces it should be more about published ones for many use cases. So when we think about derived. It's kind of those two veins. But you could also, you know, you could also take two raw data products, combine them with a set of transformations or different views that would give you a derived product.

Interviewer
Yeah. So then you mean more like a hybrid product or a composite product?

Expert C
Yeah, and hybrid is the combination of data and an algorithm. And composite is the combination of raw data and derived data, then that's a differentiator to me. I just see hybrid, hybrid and composite can kind of be similar. Same terms synonyms with hybrid.

Interviewer
I mean like the data product has two ports, one port for the raw data and one port for the algorithm, for example. So it's not like merging them.

Expert C

Yeah, yeah, OK. And I didn't even think about a data product as an algorithm itself. We are looking at an analytics channel that's solely for the sake of developing models which in turn would be an algorithm which would be that middle product. So I've seen all three, it just comes down to, you know a kind of sequence it and say to me it's raw derived as one option and then raw or derived and/or algorithm um to kind of to help organize it. Hmm. Because you have your data segmented just like you have your algorithm segment.

Interviewer
Yeah, yeah, exactly. OK, that's a really interesting perspective. I can perhaps do that to make this framework more understandable for everyone. Because now it sometimes looks a bit like a mess with all those arrows, but we have to start somewhere, right?

Expert C
Yeah.

Interviewer
OK. And if we get more to the technical part, which I really like. The infrastructure around the data product. So what do we need around the data products besides the data product itself to help it communicate with other data products and self-serve platform, etc. A very important one is the schema registry which I read a lot about in literature. Uh, the schema registry together with the event streaming backbone and the central data product catalog make sure that there is some data lineage. So that we can keep track of all the transformations. That's happening to the data itself. So the event streaming backbone pushes all the events to the central data product catalog. And the schema registry make sure that the events in the central data product catalog are somehow converted to a more understandable format where the user can, uh, yeah, observe what's going on. So that's what these three are supposed to do. Furthermore, we can speak of some kind of shared storage. There are different perspective on shared storage because, yeah, a disadvantage of shared storage is when you try to version one of your data products. Well, in that case it has to wait for the other data product because they are somehow linked together, you know? So there can be a versioning problem, but there are also certain benefits to having a shared storage because otherwise there can be so many different storages and you need so many different storage accounts to keep of. So there are some trade-offs you can make. I identified some API invocation so there can be multiple options to make your data product accessible for the consumer. For example there can be graphQL, REST api, gRPC. When we consider the shared storage that can be a cloud storage API and a storage read API? Yep. Uh, another option is the SQL Access point and yeah, this was mentioned so many times, it's really, it's really industry standard to have a SQL access points otherwise the data analysts will panic. So that's why I included it in the framework and on the bottom we see more like non-functional requirements, so the data product policy enforcement mechanisms. Yeah. So you need to have some kind of security controls with fine grained access. It's not really important to really deep dive into this. It's just a good to know that. In memory catch and a query catalog. And what I mean with the query catalog is that we have some kind of sample notebook. Where we can observe all the possible, uh, queries for the data analyst. So the data analyst can observe all the possible queries he or she can realize to get access to the data. So it's more like a guideline, you know.

Expert C
Interesting, yeah. So. I look at this it's obviously all important right. There's no point in publishing something if you can't give somebody access to it. Exactly and again the point of publishing something is so that it can be trusted like that's how we view it. It can be a trusted

and well understood data set. So that the first three pieces, first two pieces. Um, in conjunction with the last piece. So schema registry, central data, product catalog and policy enforcement, that's our governance plane if you will. So we look at we're using a data management tool for most of that. Like Collibra as an example, we put Collibra over top of it and it allows us to manage our data or metadata about the product and the business domain.

Interviewer
Is that your central data product catalog or that that is where our data product catalog lives?

Expert C
Yes. OK. So we have, you know, we're treating these as enterprise data products, so therefore use across the business. So we're looking at, you know, each business line can have their own data. But the point is to show where a business line's data is then connected to a data product. To say this field in my business glossary refers to this context in this data product and this data product can be used for this purpose. So when you search that data product, you say, alright, I know what this owner was intending to publish when they put this together so that that brings everything together. It's like the most important piece, but you're fine grained.

Interviewer
So I think that's actually the start of your data mesh journey, right. So starting with your central data product catalog, that's the first thing to do before even throwing out the data product in the mesh.

Expert C
Yep. And you know, does it exist already? And if not, then, OK, what about what, what, what's the need and what's the, you know, then you then we haven't even talked about data product lifecycle yet because there is a whole ecosystem around that life cycle, so that you don't proliferate the number of products, right? The goal should be that this product should serve most of the needs. If you truly need something else then there's a whole series of governance that you know you have to redo the cycle, right? Why build something new if you could just extend something that you already have, if that makes sense. So there's a, there's a give and take there. The thing that I want to go into the event streaming piece of this because we're seeing products kind of in two or three different patterns so far, event streaming being one of them we like Kafka it's is one of our pubsub technologies. There are. I'll pause because you've got OLTP use cases or transactional and ODP. So you've got transit, transactional and analytics. We're obviously focusing on analytics for this product for the mesh. Because the mesh introduces latency. So there are operational systems that today have a pub sub type of structure where you go to them when you need something and you're getting your real time transactional data as it comes off the system. We're struggling with some consumers now who say I want that real time data. I need data in certain specific real time. I understand that, but as they come to the mesh. The mesh does introduce an extra hop and a pattern. Which has latency just inherited it. So we're trying to balance the folks who are coming to us for that demand to say well you know there is a cost for real time data via a data mesh and there's a pattern that you can go to for a transactional subscription. Umm. So when we see that as a smaller use case, we're trying to do event streaming either. So it's not constantly streaming, but if it's event streaming, there's a listener so that you pick up data when an event happens. So it's an event on top of an event streamer or you stream it at a more in a batch manner. Which then leads me to what we also have a file-based ingestion pattern or we have an event that triggers an API. So if a fund reaches a certain status, that status will trigger an API to send data to our product. So we kind of seeing those three so far. Because it's primary. But I see your point in saying hey some event requires publishing of data but then you also have

API there too, so but that's an access perspective.

Interviewer
Yeah. So with event streaming, I primarily mean the, uh, data lineage. So if someone is changing the data, it goes through the change data capture. Then the change data capture is connected with the event streaming backbone. The event streaming backbone goes to the central data product catalog, well, sends the event to the central Data product catalog and the schema registry. Makes it accessible for the consumer. So that's gonna read so that the event is, yeah, converted into readable format? Yep, so that's actually what the. Event streaming, backbone does and I think the data flow, so with the input port and output ports, that's mainly done through the API invocation over here. But you meant some more like real time data flow. That's controlled by the event streaming backbone. Did you mean that?

Expert C
Well and I look at maybe changes from two regards, so change being new data or a change being a structural change to a product. So you know new data would have affect the calculations that of the algorithm with the product and you might have a different end result, new structural changes. Say I have a new field or introduce a field to take away a field which is more life cycle based, product lifecycle based. But to your point it all has to go through a structural change and the registry to make it readable. I was talking event streaming in my mind talked about the the hydration of a product with data. Because you have everything but structure. You have access, you have consumption, um. So the only piece you're missing is what happens when data is fed to.

Interviewer
That's what's happening over here. So the anatomy. So this is what happens all inside the data product itself. OK. So we just discussed the change data capture. So if something is changing in the data set, yeah, this component will get active. The immutable chain audit log. I'm not sure, perhaps we can excluded this from the anatomy part and more like in the infrastructure layer, but I have to think about that. Sure, the data catalog. So we have two data catalogs. We have the data catalog like a local data catalog for the data product itself. And there is an enterprise data catalog which is like the catalog of catalogs keeping track of all the information inside the mesh. Yeah, so. There should be an internal storage for the data product to make it more autonomous so that it doesn't depend on yeah like others. We have some kind of observation plane. And this observation plane is kind of interface where the consumer can observe the data quality happening in the data product. The control plane is like what the management layer is connected to so that we can check if the policies apply the local policies and the global policies and then add that we can adjust some. Some methods if they do not adhere to global policies, etcetera. And data onboarding. What I mean with data onboarding is really the data ingestion and all the transformations and everything that's happening to the data. So that's a short summary of this anatomy. I'm really curious if you recognize all these patterns and practices yourself in in the Northern Trust data architecture.

Expert C
I do, I do see it and it's just a different way of thinking about it, which I find interesting because like I mentioned, this change data capture pattern we have. We have our data ingestion patterns here. The immutable change audit log. It could be infrastructure. It's also helpful to have, you know, what's your version, what are your version notes, your release notes, maybe for a product. The data catalog is certainly something we've talked quite a bit about domain driven design is associated here like where does this product fit in the domain across domains

and then when I think about. Uhm. The planes. Zhamak talks about the three plane architecture, the infrastructure plane, the data experience plane and then the observation plane. And we've structured our mesh capabilities in that manner. So we have organized it a bit differently. But when I look at observation and control, there's also an like orchestration plane that our engineers have built to say we need to monitor the tool chain so that you know when the system or the machine is running, where is it in the status. So we've got an orchestration layer that has monitoring, alerting and ultimate like you know, just operations or orchestration management associated with it. So what you have here is observation from a consumption pattern. You might want to think about observation from an engineering pattern as well. Um, depending on your toolkit. But again that's more how it's industrialized and less about how you construct a product, which I think you've captured here.

Interviewer
Yeah, OK, because this is really related to the interface decision where I encountered 3 different ports. Well, actually, yeah, there is of course an input port and output ports, the most obvious ones, I should include them here. But let's now focus on these three. So the observation plane is connected with the observation port. Yeah. And this provides some kind of integrated experience for monitoring. And we have another one, which is the discovery port. And the discovery port should be connected to the data catalog inside the data product. So not the enterprise data catalog, but the local data catalog. And this discovery port shows you some kind of sneak peek of what you can find into the data products. So that's what the discovery port is for. Do you encounter more ports? Or do you agree with these tree over here?

Expert C
Those three are key, we have compliance monitoring and quality. I know you've talked about that before, but you obviously want to observe that as well as an interface. So whether that's observation or control. So we're looking at certainly discovery, there are you know central tools that have it. We're looking at building our own potentially. But the marketplace is key because it allows you to discover what you captured. Let me think quickly if there's anything else that's here. You know, like we think observability, we think cost, compliance access, you know data access you know like lineage as well. So I think you've captured each of those and you just bucketed them differently. So I think the way you've got it here is pretty mutually exclusive.

Interviewer
Yeah, I think these three are very important to have. Otherwise, yeah, you decrease your observability in the mesh, well, your overall monitoring experience, etc. Yeah, and after this one, we only have one framework left, so I think we're right on time. I don't know if you have any extra time left, otherwise we will hurry a bit? What's the deploy decision? Yeah, the deploy is a I think a bit obvious. So what do you need to deploy your data product? For example, containers. A single container design. I encountered a lot, but there can be multiple options. I just had another interview this morning and that was someone from Microsoft and he told me that there are many more possibilities except for containerization with Docker and Kubernetes. Yeah, still have to make the transcript because it was such a short time period. Yeah, this is the model as it is right now, and it needs some reconstruction. But Kubernetes and Docker, those two were mentioned a lot in my literature research. So that's why I implemented them here in this framework.

Expert C
Yeah. And I think this is not my area of expertise. Certainly this is more on the architecture engineering side, but you're going to get a large permutation of patterns. Umm. Like

our goal is coming up with this pattern using infrastructure as code to quickly provision the pattern for our use in the mesh and then using an enterprise CI/CD process that will that will help orchestrate end to end. So we've got CI CD orchestration and a release and build level we've got orchestration within our mesh or mesh architecture and again this like we do have Kubernetes today a cloud version of it but to put to the point I think maybe you can try to generalize the patterns and give examples of technology. Because there will be many examples of technology.Tthe pattern should be there and it should be probably replicable at some point.

Interviewer
Yeah, I don't think only these two options are out there. There are definitely many more. So I need to do some reconstructing over here, but I think it's a good thing to start thinking about these two, yeah. And yeah, for example, when you want to increase your CI CD process, it's good to have some kind of templated data pipeline. So yeah that you create some kind of standardization. So, yeah, this was a really a high level overview on the deploy decision. But if we get back to the inter-decisions because we have seen now every decision separately. What do you think about this high level framework? Do you think we need some more decisions or do I capture already most of them?

Expert C
I do think the why question. What type? What approach? I'll have to think more about this one now that I've seen everything together and I'd love to, I do have to run in a few minutes before my next meeting, but I'd love to talk a little bit more like just it could be briefly about holistically and then to just interested in your industry research and experience about how widely this is used because I see it now as something new and introduced and somewhat novel. And I think it's been great to have this conversation with you. So I would love to just see where you think we are in the maturity curve and we could talk a bit about that but yeah let me come back to this because I will think more about, you know, what we left off given all the context.

Interviewer
Yeah, sure we can do that. Perfect. Um, yeah. Later on I will also make the second framework so for the self-serve platform and we can perhaps have some kind of interview again or discussion.

Expert C
Talk about that one too. Yeah, I have when you mentioned those three, I kind of started smiling because they're certainly what's top of mind for us which again is why it's so exciting.

Interviewer
Yeah, yeah and especially in these self-serve platform that's yeah that's some kind of component that you can entirely implement in the the snowflake technology, right, yeah, Snowflake is some kind of self-serve platform and it gets really interesting when you look at the data product and the management layer because there are so many options out there. Well, so many other technologies I mean. So yeah, you also need to look for the best technology because Pyxis is using AWS and Snowflake at the moment. So there needs to be a third party technology on the central data product catalog. Your central data product catalog is?

Expert C
Collibra.

Interviewer

Yeah. OK, I will check that.

Expert C
There are multiple tools there too, but you could look at the Gardner 4 quadrants and say you know where do these tools lie in the quadrant. I think Collibra is up there and they just provide different features.

Interviewer
So is that an open source tool or?

Expert C
It's commercial. I don't know. I don't think there's an open source angle to it. There might be a small piece, but limited.

Interviewer
Have you used many other third party technologies? So besides Snowflake and I think you were using Azure.

Expert C
Azure, Air Flow, definitely commercial versions, astronomer and then GitHub for our repositories and. Let me try to think what else. Us. Yeah, I'll have to look at that too and we should have a separate session. Well, put that on the agenda for the next session

Interviewer
I will do, I will do everything we can meet in a short period of time. I mean, I'm available like every day. I'm working on this like every day.

Expert C
So let's do it. Let's plan an early session and next time you have a round of feedback. Let's plan on it. Let's exchange emails, Tom, and thank you. I'm sorry, I have to, I have to run quickly, but this is great.

Interviewer
Well, that sounds good. Thank you. Thank you very much for this interview and we will keep in touch. Thank you. Bye, bye.

Interviewer