1:57

Alexey

**This week, we will talk about growing a data engineering team in a scale-up. We have a special guest today, Mehdi. I know Mehdi as a content creator and a data engineer. We live in the same city, Berlin, and this is how we met. It was actually near Alexanderplatz, where I work. He has over eight years of experience in data and engineering. Now he works as a staff lead engineer. Today we will talk about scaling data teams in a scale-up. Welcome.**

2:27

Mehdi

Yeah, thanks for welcoming me on the show with such praise on my YouTube channel. I need to keep up the game now.

# Mehdi’s background

2:35

Alexey

**Before we go into our main topic, let's start with your background. Can you tell us about your career journey so far?**

2:42

Mehdi

Yeah, sure. I started, as you mentioned, about eight years ago in the data world, doing classic BI with Microsoft tooling, and mostly click, and drag-and-drop tooling. And then I had quite quickly the opportunity to jump early on a Big Data project – an on-premise Adobe cluster back then – and I just surfed on that wave, because back then there were few resources to do such work at scale. A lot of companies started to invest into their on-premise cluster so I had a kind of R&D project. But again, remember, cloud was there but, not as popular and without free tier. Basically, if you wanted to have a playground, it already cost you a lot of money. [chuckles]

I just surfed on that wave, built my skills around data engineering, mostly Big Data infrastructure. I worked in corporate and then moved out from Belgium about three years ago, trying new tech companies, like Klarna and the bike market. Those experiences, the last work experience was mostly in a scale-up company. I think the environment is pretty different from a standard company. And that's where I am today – I just started as a staff data engineer now.

4:22

Alexey

**Do you miss the days when you had to have an on-premise Hadoop cluster to do analytics or Big Data? [cross-talk]**

4:31

Mehdi

I don't. I hate it. I think the barrier to entry was so high – you needed an entire team, there were so many skills needed, like networking, setting up servers… I mean, networking you still do on the cloud, but not as heavy when you have to order the machine to the data center and configure your network infrastructure itself. So yeah.

4:58

Alexey

**I remember we had four servers from Hetzner – four machines – and then we had a Hadoop cluster there. Not only was it constantly breaking – we didn't have a team to look after it – but also, most of the time, it was idle, and we (data scientists) would just SSH to one of these servers to do some data science stuff. If somebody wanted to run a Hadoop job there, you had to prioritize [chuckles].**

5:25

Mehdi

Exactly. Before, when you were running a big job, you were blocking everybody, so people were coming to your desk. Now you’re just burning your credit card. So it’s at the end of the month that people come to your desk.

# The difference between startup, scale-up and enterprise

5:41

Alexey

**[laughs] So what does it mean to be a scale-up? I know what a startup is – a startup is a company that just started up. I also know more or less what an enterprise is – it’s a huge company with a lot of people working there. So what is a scale-up? Is it something in between? Or what is it?**

6:00

Mehdi

I think there is a different understanding of what a scale-up is. But, roughly speaking, it’s a company that is in hyper-growth. So it's usually a company that has already achieved first revenues and they have another big funding series to basically scale-up their team because the pace to which they get the user is faster to the engineering team. At a certain point, they have other challenges, so they need other skills – so they are heavily recruiting.

It's usually a company that goes from, I don't know, something like 80 people to 400, in like two years or one year even. When I was at Klarna it was like 30 people joining every month. I would say that it’s mostly the environment. Connected to that you have decisions that need to be made quickly, so it's not an environment where you have time. You still need to prove yourself on the markets, or you’re in competition, so the driven decision for certain features are going faster.

7:28

Alexey

**When you describe it, I imagine an environment where it's just a bunch of people, a lot of users, and everything is all fire and everyone is running around screaming. [chuckles] Is it similar to that?**

7:41

Mehdi

Yeah, it's roughly correct. Yeah. I mean, I think if I look at the different companies I have been working with, they have different challenges at different times. Maybe they’ve launched a new product, maybe they’ve launched something else. A classic challenge is when a European company scales to the US. It's a completely different dimension of users and of their usage. If you do, for example, B2B and you have one customer that has shopping and you serve something on the e-commerce over there – it has so much impact because it's so big that it's *way* bigger than what you can have on the European market. So that brings other challenges downstream. This is one example.

Another example is a new product feature, which is quite challenging. I think it really depends, but there is always a lot of stuff on fire and you need to basically make compromises to see what the biggest one to focus on is. [chuckles]

9:02

Alexey

**And not only that, you also need to grow a team, right? You need to quickly fix things, like your production issues, in such a way that maybe you will regret later, but you need to fix it up anyway. Then you also need to hire a lot of people and then all these people come and also add some code fixes. I guess it's quite a challenge, right?**

9:26

Mehdi

Yes. Exactly as you mentioned, you need to make compromises between speed and quality all the time. [cross-talk] Otherwise, you're just collecting technical debt if you just use speed. This dimension of onboarding people – for me, I think that was one of the biggest challenges. You have so many people coming in, not just in your team, but as stakeholders. I think for me, it's the two biggest things like, “How do you make compromises all the time?” And “How do you keep your onboarding process smooth?” And also “Is your product ready to scale internally or externally?”

# Hypergrowth

10:21

Alexey.

**So, in this case, the hyper growth that you were talking about refers to both growth in terms of the user base, also in terms of traffic, consequently, and also in terms of the number of people who work at the company. And it all happens at the same time, right?**

10:36

Mehdi

I mean, it depends. But usually, they want to hire a lot of people because they get a lot of users and they have new challenges.

10:49

Alexey

**What are some of the recent companies that you would describe as scale-ups? Maybe you heard about Gorillas – this is a grocery delivery company. Would it be considered a scale-up?**

10:59

Mehdi

Yeah, definitely. In Berlin, there are a lot. I think what I would *not* call scale-up, even if they’re tech companies like Zalando, for example. It's more established. Spotify is another example.

11:11

Alexey

**That would be an enterprise, right?**

11:12

Mehdi

Yeah. Because they're already rich – they're still growing, but not at the same curve. They have more established revenue which makes them more comfortable to survive such a barrier that we have now, for example.

11:33

Alexey

**Scale-ups typically have investors as well, right? They are not necessarily earning all this money – they usually just closed an investment round, so they have borrowed money. So it's necessarily their own.**

11:49

Mehdi

Yeah, most of the time. I'm not such a huge fan. With what the landscape has been like in these past years, we've got huge funding. Was it reasonable or not? That's another story. But yeah, usually there is funding. That's what I was saying, that we achieved some first revenues. But the value is still yet to be proven. That's why basically, you bet on investing into a lot of features and a lot of engineers to reach a state where you're comfortable.

# Data platform engineers in a scale-up environment

12:30

Alexey

**And what do data engineers do in a scale-up environment? How is this world different from your typical enterprise?**

12:40

Mehdi

I would say that we put the landscape in place, so here it is. You're joining a team. A data engineering team or a data platform team has multiple consumers internally – data analysts, data scientists – and let's say you have 100 of them today (or maybe way less, like 50) but the growth over the month is really big. So aside from the people, you would expect a lot more use cases, different ways of using your products as a data platform engineer, so you need to think when you build stuff and how you can scale the usability directly and remove yourself from the equation.

That was mostly my role – I forget to mention in the intro – but there are multiple definitions of data engineer. But I have been working more as a data platform engineer, providing tools and services. Before, I was doing more things like writing pipelines and business logic, in order to deliver that to other internal data stakeholders. There will still be those kinds of people in a scale-up. But it's the same thing – they need to think about how they can scale. Like, “Okay, I'm doing a small pipeline that uses a Google sheet. It's totally fine for this small use case. But then suddenly that Google sheet blows up because there is way more data.”

Giving a certain context, let’s come back to going to the US market, for example. So what do you do, right? So that's the thing – you need to anticipate more about the fact that things are going to break way sooner than they otherwise would. I would say in a startup, you can be a bit more of a cowboy because you can say “Okay, let's just have users. Do a shortcut, and maybe there is some technical depth down the line that we're going to regret. But it's fine. Let's just bring value in and go for it.” And in a scale-up environment, it's a bit different. Because if you do the shortcuts without thinking too much, it's gonna cost you way sooner than expected.

15:11

Alexey

**As I understand you, as a data platform engineer (or a data engineer working on the data platform) your role is to let others build data pipelines. You build tools for that so then the data scientists, data analysts – whoever needs data – can use this platform to do whatever they need. And you need to build it in such a way that it can withstand 10 times more traffic than today.**

15:40

Mehdi

Yeah. Coming back to the onboarding thing, I think there is a lot of education to do around those different people, because they all come from different backgrounds. I guess the data roles in general – the names – they're super confusing, right? Data scientists, analytics engineers… How deep is the technical knowledge of the person? Does he have experience with *that* technology or *that* one? So your job as a data platform engineer is also really to make it as simple as possible, so that if you assume a certain technical level of your data consumers, it has to be “Okay, I think they're going to be able to do that.” But not that much and it has to be pretty well-documented.

You may have an onboarding session, a support channel that is reactive, that listens to the user. So it's really kind of like a startup inside a scale-up where you really need to listen closely to your consumer, because they're growing quickly. If you just behave like, “Okay, I'm gonna do that pipeline myself because it's faster.” Like you could do, for example, in other companies – you will regret it down the line because you're still putting yourself in the equation as a dependency. Here, it's really about leveraging other data people.

# What a data platform is and who builds it

17:22

Alexey

**In practical terms, what does having a data platform usually mean? Is it like an Airflow cluster and then some other things that you can use from this Airflow cluster? Or what?**

17:34

Mehdi

I think it's more than that, definitely, because that’s just one piece. It's like having a car without a driver’s license. [chuckles] You cannot do much. I mean, you could – but then things are probably gonna break. [cross-talk]

17:52

Alexey

**So Airflow is the car, right? What is the driver’s license then?**

17:56

Mehdi

The driving license probably would be, for example how you should structure the thing, how you should handle sequence in Airflow – what's the naming convention? Maybe also configure some Airflow pipelines as a YAML file, if they're generic, rather than having to rewrite them all in all. Again, this is a classic thing where you take an Airflow DAG, “Yeah, it's okay. We can do it manually and copy/paste. They are slightly different. We are not going into the struggle of having a YAML file that generates DAGs.” There are easy frameworks, by the way, to help you do so.

But the point is, again, you need to think ahead and say, “Okay, are we going to have 10 pipelines like this? Or 20 pipelines next month like this? Maybe we should better invest that time now.” It's always about compromise. To come back to your question, the driving license is basically all the best practices that you can put from the start so that things do not go bananas and become hard to control. That's usually the case – if you have a lot of users, they have so much imagination that at the end, you end up with a lot of *interesting things* on your platform.

19:25

Alexey

**So in some ways, the data platform is the car, the car park is Airflow plus I guess some connections maybe to databases, data lakes, and so on, so you can read from somewhere, write to some other place. Then, in addition to that, you have a set of rules or a playbook or something that people can use in order to effectively use the Airflow part – the car park – so they can drive the car.**

**You, as a data engineer, need to make sure that, first of all, these things are up to date and users know how to use it. I guess you also help them figure out how exactly they should use it.**

20:09

Mehdi

Yeah, that's exactly it. Yeah.

20:13

Alexey

**Okay. And who do we need to actually build this thing? This type of data platform?**

20:20

Mehdi

Yeah, I think I can tell you about some mistakes in the past. For a scale-up, you probably don't want to compromise on seniority. You know, there is this saying, “If you pay peanuts, you get a monkey.” [chuckles] The point is, you should try to recruit a lot of senior folks first to set those best practices. You don't need a ton of them, but if you are just relying on juniors or new folks who are starting their career to set up things that scale at a huge fast pace, it will be really difficult. The problem is that learning on the job is really challenging, because the challenges fall down on your desk. It's like, “Okay, now we're going to the US, so you need to scale this.” “It's not just “Okay, what if we build that feature?” No. It's happening next month. [chuckles] Also, you’re trying to get people with niche skills, if you need.

Again, it's the same reason – because it's really difficult to ramp up on certain technology. Let's say you have a use case with streaming, and you need to implement a Kafka cluster. Don't expect people that have never worked with Kafka, even if they're brilliant engineers, to just learn by themselves and set up the thing to scale up. Even if it's temporary, hire an expert in that – someone who has huge security. Same if you go into the cloud. Most companies are already in the cloud, or if you are changing a cloud provider, just get someone that has experience with that and that will help to put the best practices in place.

I mean, even you and me, we're both experienced in the domain, but whenever we tackle a new technology, it's always super hard to say, “Is it the right way to use it?” There is a value in experience of having to work before with the technology if you need to scale it fast. Yeah.

22:48

Alexey

**You mentioned “We need a Kafka cluster.” But if we don't have experience using Kafka, how do we know that we actually need it? It would be funny to have to hire a person who has this experience with Kafka only for them to start working and say, “Hey, we shouldn't use Kafka here.”**

23:09

Mehdi

[chuckles] Yeah, no. That's fairly true. There is an architecture decision that should also be driven by some senior people. But I think [cross-talk]

23:20

Alexey

**So you need somebody like that, too, when you’re designing the platform.**

23:26

Mehdi

Yeah. But the point is also at the lower level, let's say you have an engineer or software engineer that's comfortable with that subsystem, maybe RabbitMQ, and they want to consider Kafka. Some of them have heard of it, some of them have been playing a bit – they consume the Kafka topic. But the point is, just having high level knowledge on this to possibly make some decision, and even some good decision, is not enough to say, “Okay, now we're going to implement it and serve it at scale. We have a team and we're going to maintain it,” right? That's a huge difference.

The point is that you're going to start with one or two topics to try it out and then the next round, you're probably going to have 15. By the end of three months, you're going to have hundreds of topics. And if you didn't have the Kafka expert at the beginning, then it's really hard to untie everything. A concrete example I can give you – I was mentioning Kafka just off the top of my head. But for data folks, it's a data contract with users – or a software engineer pushes data to Kafka and they have used it from service to service, as a Pub/Sub. And they often really don’t care about what's happening downstream. Then, the data team is going to apply to Kafka downstream to S3 and then the transformation.

One best practice to do in that sense is to actually put some data contracts in place with the data folks and say, “Okay, this is the schema that we're going to have. Everything is going to be typed. So we are not going to put random JSON. It’s gonna be Avro, for example on the schema registry. You can consult the latest version. If there is a change in the schema, this is what we're going to allow to change. This is our process for changing the schema.” As a whole those things will go smoothly for the things down the line, because then if you have a new user of Kafka, you just read the guideline and that's it. But if there is no guideline, again, people are really creative. [chuckles]

In the end, you’re just suffering, consuming that stuff. Whereas the data platform team can say, “Oh, there is a data type change in that schema. That was allowed.” Or “There is not even a proper schema, it's just a JSON that's evolving. It's really hard to parse. We have iCompute cost on our Spark cluster or data warehouse, because the data is not super well-modeled from the source.” So basically, it's just a few examples that create a lot of downstream problems that you could have avoided with some best practices from the start.

26:15

Alexey

**So I guess if you don't have experience in Kafka, then you might not know that schemas change, or you might not realize it and just say, “Okay, I'll just be sending JSON. What can possibly go wrong?” But then in a couple of months, something happens, the data changes, and you just didn't think about this. The person who had experienced this – who probably lived through this experience of having to somehow manage this ever-changing JSON files (perhaps maybe you had this experience) that's why you're bringing this up, right? [chuckles]**

26:52

Mehdi

Yeah. [laughs] Definitely. It was a pain.

26:54

Alexey

**Then these people that know about it go, “Hey, wait a minute. We first need to think about this thing. Because when there are 100 topics, then it will bite us later.” Right?**

27:04

Mehdi

Yeah.

# Managing the fast pace of a scale-up while ensuring personal growth

27:05

Alexey

**Okay. We have quite a few questions. The question we have is, “Hi, I recently got a senior data engineer role and I expect it to be fast-paced. Any idea as to how I can manage the pace while ensuring personal growth?”**

27:24

Mehdi

That's a really good question. I've seen depressed data engineers in a scale-up environment sometimes. We were talking about that trade-off between speed and quality. Sometimes you're not really in control of that decision – there is politics in place, and you just redo. So if you're in a place where you keep redoing things, and you just keep extinguishing fires without having a proper vision, and it’s always the same fire – that's hurtful. It's also not really interesting for learning because you just redoing the same things and the same fixes. I think it really depends on the general culture.

I would say that in a scale-up, once you get above 200 people, you basically have a macro culture of the company and the micro culture within the team. Those could be quite different. So I would say talk with your future teammates, your engineer, to feel it. I understand that you already got the role, so you just started. I think it's a discussion within the team to basically make those compromises together. I think it's important. If it is just a part of negotiating – if you just keep fixing all the things without having the time to build stuff that is more interesting, where you can learn, then it's just that there is a problem somewhere else where someone is just accepting everything and not putting a buffer.

Again, it's not just saying, “Hey, let's see each other in a month and we will build our stuff and come back with rockets.” It's just about compromises. Just say, “Okay, you need this thing tomorrow. Can you have it in three days? I want to refactor this part.” For bigger projects, it's a lot of communication with the stakeholder to say, “Hey, we want to get away with this. This is all the pain points that we've been notified of.” If you're intentional with what you want to improve and the time that you're going to save, then I think you're going to have a lot of space to learn and grow. But it's true that it can be challenging to have to make those compromises, and it highly depends on who is supporting you and which team you end up with.

30:20

Alexey

**You also brought up this topic of culture and I guess a company that recently was a startup, but now, all of a sudden, is experiencing growth, then they might not get a chance to adapt to this new environment, right? Then maybe as a senior data engineer, you can try to somehow influence the culture?**

**You can say, “No, this is not how we should do this, because we're no longer a startup. We expect that next week there could be two, three – 10 times more traffic.” As a senior data engineer, or as a senior engineer in general, I think sometimes it's kind of expected that you bring these topics up, right?**

31:07

Mehdi

Yeah. It's a really great point that you brought up. Basically, you have to know that the general company is in transformation – it’s a huge shift – and they should be open in terms of culture. They should say, “Okay, we used to do one release every week and that was fine. But we have too many breaks. We need to change our process. We need to change the way we work.” You have space to speak up, but you have to take that space. It's true that sometimes people end up in a scale-up environment a bit randomly. They may not really be aware that it's a company that's growing a lot.

You should take a step back and say, “Okay, there is a lot of stuff that’s not going to work.” It’s exactly as you mentioned, because people think, “Oh, we're still a startup. We can do that.” Again, that doesn’t work. Even a simple thing like sending a birthday or a goodbye email to the whole company. [chuckles] As the company grows into a corporation, you're not going to say, “Hey, it's my birthday.” You’re laughing, but I saw some scale-ups still doing that, and you feel like, “Okay, we probably shouldn't do that anymore.” I mean, it's a really minor thing. It's just a small story. But you get the point – there are a lot of internal things that need to change and you just need to embrace that change and suggest how we can move forward into this.

31:45

Alexey

**This question is a bit ambiguous and maybe on purpose. It says “while ensuring personal growth” and personal growth can mean different things for different people. So what kind of personal growth do you think you can expect from scale-up?**

**Is it more like this culture-changing sort of personal growth? Leading the team – I guess you had experience with this. Or is it more like new tools, scaling problems, or all of the above?**

33:21

Mehdi

Yeah. When I heard “personal growth” I think it's both on technical and on individual contributor track, or the manager track and all about leadership and everything else. But I think it's both sides. Again, it really heavily depends on where you end up. But you do have a huge opportunity over there. Let's state the problem the other way around – if you are in a slow-growth company, they may not have that much budget. They might hire maybe one or two people in your team. It's completely different.

Here, you have opportunities to work on the onboarding process, get other people up to speed, improve your tool sets, and improve the data platform that you serve, because you have a lot of use cases and people are screaming. Even if it's a constant fire, and I agree it might not fit everybody, depending on their mood, but at least you know that some people are using what you're doing. In a slow-growth company, maybe you're gonna work on that small project, or that existing project and it's harder to see your impacts. Whereas, in a scalable environment, you have so many opportunities that you're going to see an impact of what you’re doing at certain points.

# Should a senior data person consider a scale-up or an enterprise?

35:05

Alexey

**So what do you recommend to somebody that’s a senior and that person is considering multiple offers – and one of these offers is for a scale-up company. Let's say another offer is for an enterprise. Would you suggest that the person goes with a scale-up or with an enterprise?**

35:26

Mehdi

I think it depends on personal life. [chuckles] I've heard both things. I mean, you and I both have kids – I feel like when you have kids, depending on the situation in life, it's different. What you want, what you do next with your job. People are really fine to do just nine to five, and have their family time and they don't look further or look to learn something new. There is nothing wrong with that. Everybody's different. Everybody has different ambitions. So I would not advise on this. I think it's a really opinionated choice.

I can say, if you want to grow your career faster, then definitely a startup or a big FAANG company is a good place. At a big FAANG company, you will meet incredible people, because there is a base level that is really high. You're going to work on a really niche product, so the problem is that maybe your impact is not going to be that big. You're going to work on that Google doc comment color – like it's going to be really, really small, right? But you're going to learn so many technical things and build a network that is insane. So there’s a third option, I would say – besides a startup or a big tech company like Facebook or Netflix.

But I found that a scale-up is actually a good compromise to a startup, because they have money to run a couple of years, so there is interest. Normally there is first revenue generated. But they’re still, of course, not at the same size as a big tech company. So you could have a project that has a huge company-wide impact. So I feel like it's the best of two worlds, between a startup and a FAANG, for growth. But now, it can be really stressful. It can be time consuming – extra hours here and there, depending on the culture. In the US, it's different from Europe. But yeah, take that into consideration.

Again, I always do a reverse interview. After the process, I do one more interview with the manager that I'm going to work with, or some direct teammates and we can just chat a bit more informally about my concerns and so on to grasp things like, “What's the mood over there?”

38:12

Alexey

**And what do you ask?**

38:15

Mehdi

“What's the most boring thing at work?” [chuckles] I usually even ask that during the interview process, because I don't have the offer yet. The people giving the interview are not sure if I'm the right fit. I mean, in general, because there are multiple people asking, but I think there is a question that you can ask like that, or “Are you used to working during the weekend or not? What time do you stop?” It's an informal question, but if you ask that to everybody in your interview process, you get the grasp of the general culture and how it's driven.

# Should a junior data person consider a scale-up or an enterprise?

39:02

Alexey

**Would your answer be different for a junior specialist? Somebody who is just entering the field of data engineering and maybe has less than one year of experience? For them it may be just too boring to work nine to five?**

39:20

Mehdi

Yeah, indeed. Coming back, it depends on where you sit in life and in corporate vs scale-up – that's concerning if you already have a couple of years of experience. If you're a junior, then definitely – that's the best way to dive into the cold water. You don't have, let’s say “hard commitments” on different things, so you can really spend a lot of time on your work, at least for the first years. You just ramp up. I've seen people growing super super fast in a scale-up. In a lot of scale-ups, or in tech companies also, there are around two promotions per year. And I've seen some unique talent being promoted two times a year because they were just delivering *so* much.

Again, everybody's different. Everybody has life contexts that are different, but the point is – you have the opportunity. In a classic company that would never happen. That would never happen because you won’t have the opportunity. It’s not about the promotion round. It's also about having the projects show up. And I think in one year, in a certain scale-up, when you see their products before and after the year, a lot of things could happen. So yeah, why not?

# Sourcing talent for hyper-growth companies and developing a community culture

40:51

Alexey

**Yeah. Another question we have, “I am leading a data team in a hyper-growth scale-up company in Lyon, France. How do you guys source degreed engineers?” I guess this question is “Where do we find talent? How do we find the engineers?”**

41:07

Mehdi

We don't find them? There are not enough of them. [laughs]

41:11

Alexey

**So you will not answer this question? [laughs]**

41:16

Mehdi

No, no, no. [laughs] I think there is something that definitely will help, and something that scale-ups need to start working on is really to start to work on is the engineering culture. A good example of an established tech company is Airbnb – a lot of engineers know them, of course, because of the service, but they also know them for the engineering part. They have so many data projects, they have an insane blog, they speak at conferences. So you know them because of what they do on the engineering side. I think scale-ups need to invest into those things – into contributing back to the community – to show “this is our engineering environment.” I feel that's the best way to attract talent.

Engineers in general are mostly attracted to engineering problems. [chuckles] So if they see there is this mindset in that open source project from that company, when they are passionate, they actively look at that. It's also a way to filter out some people – to get people that are more curious. I think curiosity is a really great value to get when you are seeking talent. It's really hard to evaluate. You can ask, “Okay, do you have side projects?” Do you often read or write technical books?” I think when you get people contributing to your open source project, reaching out to you after a tech conference because they heard you speaking, you know that those people are curious and willing to learn and passionate about what they do. This can easily show up as a talent that can grow really well.

43:23

Alexey

**At the beginning, you also mentioned that you sometimes met with depressed data engineers. I guess, in this environment, people might have already felt like squeezed lemons and then, as a VP of engineering, you realize, “Okay. Now I want to improve culture because we want to change it – because we want to attract more talent. We want to have a nice employer brand.” So then you tell all your squeezed engineers to also write articles. So they’re like “Okay, I’m already working 12 hours per day. So where do I find the time?” Again, I guess that’s a question of culture, right?**

44:05

Mehdi

Yeah, it's a balance. But it's a really good point. Coming back to reality, not everything is green and nice. But, normally, for those kinds of opportunities of conferences and blogs, it should be led by more senior people. If we come back to our example with the junior person, it's not necessarily him that's going to write, but he's going to have the opportunity to write a blog about it. It's gonna take extra work. But I believe that the reward is really insane. To give another example – I know it's also really opinionated, and I don't think that they do it anymore – Google’s 20 percent. I don't know if you remember that, where you have 20% of your time to work on any tech project.

But you have to justify everything – I never looked at it in too much detail. A lot of people were saying, “Oh, but it's not 80 and 20% – it’s 120%. You have to work on the weekend for this project, where you need to show up with something.” *But* Google Maps was built on those 20%. So the point is there was an opportunity there and someone took it on the side. The point is that you have opportunities to show up, but it will require quite some work. I think at a certain point, you can find a balance where you're not squeezed lemon, as you mentioned.

46:02

Alexey

**I also happen to work to be an editor of our corporate tech blog. What I noticed is that sometimes, somebody works on a very cool thing and I ask them, “Hey, can you write something about this because this is so cool?” And I get a bit of a pushback, “This is nothing special. I’m just doing my work.” So I'm wondering if you have a suggestion for me – how do I convince them that what they’re working on is actually not just the usual stuff they’re doing, but that it's also quite interesting. Many other people will also be interested in learning this thing – about the problem, the solution, and things like this.**

# Generating content and getting feedback

46:44

Mehdi

Yeah. The advice I recommend for any junior getting into industry is writing content or doing content in general. When I say content, it can be a PR on a repo, it can be your project that you push as open source, it can be a blog, video, anything. The best way to learn is actually to teach it to someone else. So that's the first thing. And the second thing – I think there is a lot of value in writing in general because it forces you to summarize things and you get to a review process when you get feedback from people. Then afterwards, as you mentioned, when you share it, you also get feedback from other people.

The fun fact is that most of the time, you get more feedback from people outside your company than inside your company. Because your things are so niche and you are so connected online in 2022. I think this community is a really good example. You will never be able to launch a mostly online community like five years ago. [chuckles] But as we get a bigger network online and feedback online, it becomes a really good opportunity for you to get feedback on this. “Oh, actually that design – it sucks.” And it's okay. You're gonna have haters anyway.

But it puts things into perspective. Other people can say “Actually, this is really great. We did exactly the same thing, but slightly different.” I had long talks with people that read some of my blog posts online – we booked an hour – and I learned so much from them. I learned as much as they gave. People see that as work, like, “Oh, I have to do this for others.” But you're going to learn a lot through the process – pros and cons. It's really the best way to evaluate where you stand.

48:53

Alexey

**How did you get this call? Somebody read your article, reached out to you on LinkedIn saying, “Hey, amazing article!” And then you asked them, “Can we get on a call? I want to ask what you think about this.” Is that what happened?**

49:06

Mehdi

[laughs] It was actually the other way around. They asked me directly because they wanted more insights. Sometimes in the article you need to take some shortcuts. Or sometimes they reach out by chat and they give a precise technical question, like, “Okay, you didn't mention this. Can you please elaborate?” I get back to them because I know they have a similar implementation and I’m like, “Oh, by the way, I fixed that issue. Why do you think…?”

It's a win-win. I really want to insist on that. Most of the time, it's a win-win when people ask you things, at least, when you go further than the Hello World phase of the project. When it's a company blog, it's often not a Hello World project. So I think there is way more value in writing those blogs. A personal blog is also super great to do, but a company blog can bring a lot of value.

# Generalization vs specialization for data engineers in a scale-up

50:17

Alexey

**Yeah, thanks. In a scale-up, people usually need to work on a wide range of tasks, right? You’re pretty much a generalist. But then, as the scale-up grows and evolves, these people might need to specialize later. Do you think this is okay and how does one deal with specialization?**

50:39

Mehdi

Yeah, I don't think you're that much of a generalist. I mean, a data engineer is already a generalist. We put everything that we can into that role. I think it's the same for most of the data roles. Data scientists might do a lot of pipeline work in scale-up, until they find a proper data engineering team to help them. Again, I think it depends on the setup. I didn't feel like I was too much of a generalist. As the teams are growing, your responsibility will be scoping down and more niche and then, consequently, your skill set.

To give a concrete example, let's say you are the data platform engineer that maintains a data warehouse, the infra-Kafka cluster. Now you have a lot of users and you want to build a framework like DBT, to enable people to execute SQL in a production manner – so what DBT does. It's an open source project – okay, it's available. But let's say you have to build this thing in-house. It doesn't exist. Then it's going to be a dedicated team, right? You cannot manage to do it in-house plus that at the same time, so your scope will go down. You go from more of an access management role to really properly developing a service that involves a different set of technology.

It’s the same thing for streaming. Let’s say now you will be involved in streaming. Based on the project and how things are going to grow, you're going to be scoped down and be more niche, like we were talking about Google Docs comment features. I think you need to have specific skill sets, like real-time, web socket, and so on, for those kinds of teams. So you're going to build a certain profile basically based on the project that's gonna come over.

# The ratio of work between platform building and use case pipelines

52:55

Alexey

**At the beginning, you also mentioned that there are different kinds of data engineers. One type of data engineers are those that work on platforms, and the other kind works more on use cases. The question that we have here is, “What is the ratio of work between platform building and use case pipelines?”**

53:21

Mehdi

It's a really good question. Again, it depends on the size of your company. In the context of a scale-up, I would say 50/50, because you're gonna build as many pipelines as features that you develop in order to ease the development of the pipeline. [chuckles] There are going to be a lot of requests that are similar, basically. You need to build frameworks – to build something generic around it, so that's more of the data platform work. But yes, I would say 50/50, in general.

54:10

Alexey

**So you start getting requests and then you see if there is any pattern, and when there is a pattern, you build a solution for this. Then the people who request these things will just be able to self-serve.**

54:29

Mehdi

Exactly.

# Being proactive in order to progress to mid or senior level

54:31

Alexey

**What would you say is the most important attribute for a data engineer who wants to get promoted from a mid-level role to a senior role?**

54:55

Mehdi

I think you need to be more proactive. As a mid-level engineer, you're basically solving problems for your team and as a senior, you're stepping back and looking at what the closest team is doing in terms of the data platform or users. Both teams have projects, so you start thinking about what you need to do in order to help two teams, for example, your team and another one. It's the same thing for staff. When you get staff, it's more about the whole engineering scope, “Okay, how do I have a project that solves issues that are present for all the engineering teams?”

I would say the more you grow, the bigger the impact you should have. Basically, as a mindset, you should always kind of step back from your tickets and what you have in the backlog and say, “Okay, why are we doing this? What's the problem?” Go talk with people from other teams, book a chat. I think it's way harder now, for me at least. Before I would just go to the coffee machine and learn stuff. But book intentionally and say, “Hey, I just want to chat informally about what you’ve been up to.” And maybe you're going to see “Oh, there is something to do.”

# Caps and bass guitars

56:34

Alexey

**Are you looking for a band? Somebody's interested if you are because it appears they need a bass guitar player.**

56:23

Mehdi

[laughs] A bass guitar player? No, I'm not. I have my own project. And I've been putting that aside. So, no.

56:49

Alexey

**I think they want to have somebody who is both a data engineer and a bass guitar player.**

56:58

Mehdi

[laughs] Yes, maybe. I don’t know. [chuckle] I unfortunately don’t have time for that.

57:05

Alexey

**How many caps do you have?**

57:08

Mehdi

Is that your question? Or is it a question from the audience? [chuckles]

57:12

Alexey

**[laughs] This is mine.**

57:14

Mehdi

Okay, because I was thinking maybe there are some colleagues trolling me. [chuckles] It's a common question. Say a number and I’ll say lower or higher.

57:26

Alexey

**A hundred?**

57:28

Mehdi

Lower.

57:31

Alexey

**Okay, I'm disappointed. [chuckles]**

57:39

Mehdi

[laughs]

57:35

Alexey

**Twenty?**

57:38

Mehdi

Um. Still a bit lower.

57:41

Alexey

**Okay.**

57:42

Mehdi

Okay. We’ll stop there. I wanted to leave the mystery. [laughs]

# MehdiO DataTV and DataCreators.Club (Mehdi’s YouTube Channel and podcast)

57:48

Alexey

**[laughs] Yeah. Actually, I wanted to talk about your channel. But then we have quite a few questions. So I wanted to cover them, too. So tell us about your channel.**

57:59

Mehdi

Yeah. It's a channel about data engineering, mostly – also coding. I mean, any software engineer is welcome. I think data engineers are software engineers at first. It's mostly me sharing whatever I can about my experience around data scope. I now have a bit of a data engineer roadmap as a project, so a roadmap with a specific project. I'll be showcasing those projects. So really, my intention here is just sharing my knowledge and getting feedback. When people come back to me and say, “Hey, maybe you did that wrong.” So yeah, that's the channel. It's called MehdiO DataTV.

I also have a podcast now, which is called Data Creators Club, which is also on the YouTube channel. There are links to Spotify and so on. DataCreators.club is the website I've done to find data mentors. You can search any data mentors, on any channel, YouTube, LinkedIn, Medium – it's a pretty simple website, but it gets a lot of traction. I guess it's me just learning from people and putting a list of those. You are, of course, inside that list.

59:30

Alexey

**Thank you.**

59:32

Mehdi

Yeah, and that's mostly where I'm active. For writing, I'm using Medium and LinkedIn too.

59:41

Alexey

**So how much time do you spend on creating a video? When I look at this, it's amazingly high-quality videos. I know that there are other creators and they have an entire team to actually do this. But you don’t have a team, right?**

59:57

Mehdi

I’m cloning myself. We have this one [points to duck] but it's two people. But [cross-talk]

60:03

Alexey

For those who are just listening and not seeing the video, you're pointing at a rubber duck that looks like Darth Vader.

60:12

Mehdi

Yeah, exactly. Sorry. I definitely spent too much time at the moment and still figuring out the process. I started in December with roughly one video a month. It’s taking a lot of time. I'm getting better. The first video, it literally took me two months, because you don't know things like the banner, creating a YouTube channel, the intro, the outro. I think it takes me more time to produce content – I don't have the flying wheel yet because my expectations for production are high. So it's not really a good thing. Because you were mentioning “Oh, the quality is good.” But maybe I don't spend enough time on the core content.

But I believe I will when my process is figured out. Then I’ll be like, “Okay, now I can pull high quality video in less time, so I can spend more time on the content.” The point is, don't be discouraged – if anyone is listening and wants to go on a YouTube channel. Today, you just need a smartphone and you're good to go, as long as you have a good story to tell. But for me, it's definitely taking too much time at the moment. I’m still getting into the process to lower that and get to the right place.

61:33

Alexey

**You mentioned cloning yourself and I think in one of the videos – was it about different types of data engineers? But it was actually three. [chuckles] I know we should be wrapping up. I don't know if you need to go somewhere.**

61:50

Mehdi

I still have a couple of minutes. That's fine.

61:53

Alexey

**How did you do this? You filmed three different things and then you kind of stitched it through a program?**

62:01

Mehdi

Yeah, it’s as simple as that. The only thing you need to really pay attention to in the setup is the light. The light needs to be consistent because if you have a shadow that's coming over, when you're going to cut each piece you're going to see a dark one, so you're going to see where the video is being cut. So this is the only thing. It’s a simple trick.

There are a lot of tricks, which are way more simple than people think. It's just doing some research on YouTube and figuring that out. I love doing those. I'm having fun doing those. If you don't have fun, you're gonna give up. But for me, those effects and being a bit more funnier than would say the average learning resource – that’s what excites me.

62:56

Alexey

**I wish I started asking you about this a bit earlier because I wasn't watching the time. But yeah, thanks for sharing all this. Thanks for telling us about all this. Thanks for sharing your experience. And thanks for telling us about your channel. Everyone will appreciate the effort you put in the videos. So for everyone who's listening, please go check it out. I will put the link in the description. I guess that's all for today. Have a nice weekend.**

63:22

Mehdi

Cool! And I'll see you soon, I guess!