1:56

Alexey

**This week, we'll talk about being an MLOps architect. We have a special yesterday, Danny. Danny and I met more than one year ago – slightly more, I guess. I wasn't sure who that guy was and what he wanted from me. I thought he was trying to sell me something. He actually did try to sell something. [laughs] We then interacted more and more – over time, I realized that Danny's quite a technical person. So he's quite hands on, he can code, and then the workshop he did with us a couple of months ago was quite technical.**

**Danny’s a very technical person, but I must admit, I still don't know what Danny does. I admire people who can do many different things at the same time, so that's why I invited Danny to talk more about this. Danny is an MLOps architect at WhyLabs, and in this episode, we'll learn what that means and what MLOps architects do. Welcome, Danny.**

2:58

Danny

Hi. Thanks, Alexey. I'm excited that you decided to invite me on. And I feel so honored that you say that you respect people who can do a lot of things, because I do a lot of things in my personal life and in my professional life.

# Danny’s background

3:12

Alexey

**Yeah. So before we talk about the main topic – MLOps architect – let's start with your background. Can you tell us how you ended up doing so many things and what was your career journey so far?**

3:24

Danny

Yeah. I'll give you the long version. Because we're here for an hour, I'm sitting back in my chair. Growing up, I grew up in the Bay area – in Mountain View, actually – which is the headquarters of Google and tons of startups. I was surrounded by tech and I never thought I wanted to get into tech. I was never that good at math, my dad tried to teach me programming, and I was so incredibly bored. I really loved the humanities – I really loved history and political science. I thought the way to do that and make money was to become a lawyer.

So I dropped out of high school, I went to community college, and I started taking classes about how to be a paralegal, which is kind of like what a nurse is to a doctor – a paralegal is to a lawyer. It’s somebody who has some autonomy, but mostly is helping somebody with a higher level of certification. And I realized very quickly that I absolutely hated law – that it really was not the right profession for me. I thought it was like sitting and discussing big ideas.

4:21

Alexey

**Like in Suits, right?**

4:22

Danny

Yeah, exactly right. Exactly. I thought it was like Suits. Turns out – it's paperwork. It's all just filling out paperwork and if you mess up the paperwork by switching a date, you are going to get somebody in jail. So I realized that the law is not for me and I had a big identity crisis. I applied to colleges wanting to study political science – to university, I should say – wanting to study political science, because I thought I had wanted to be a lawyer. I got accepted for political science, and then really had no idea what to do. But I had taken a stats class in community college and I really liked that statistics class.

Then I found out UCLA, the school that I went to, had a statistics major. So I started taking more and more stats classes for that major. Originally I was just going to do a minor and then I liked it so much that I turned it into my major. It turned out that I actually really liked programming. I just wanted to do object-oriented programming instead of what my dad was teaching me – a different paradigm of programming.

5:21

Alexey

**Basic, right? “Go to ten. Go to 20.”**

5:26

Danny

[laughs] Maybe I wouldn't… No, I definitely wouldn’t have… yeah, my dad really wanted to start with logic gates. I mean, he has an electrical engineering background, so he wanted to start at the hardware-level, logic gates, firmware.

5:35

Alexey

**Like assembly.**

5:36

Danny

Assembly, yeah. That was too deep for me and I really like the abstractions that we get to use these days. Especially when I learned – I started with C++ and that was interesting, but it was annoying having to think about objects and having to keep track of everything. Then I learned about R, because we were using that in our stats courses and I was like, “Oh, this is cool. I can write English and the computer will understand.” And then when I discovered Python in my job, I was like, “Oh, man. Okay, this is where it's at. I can write English, the computer will understand it *and* people won't make fun of me for writing everything in R.”

So that was my academic background and how I discovered machine learning, because machine learning was really emerging at the core of the statistics classes that I was taking. All of the interesting, cutting edge stats classes were really machine learning-oriented. I went back to the Bay area to live with my family and I worked at a startup called Qubole. I started there as an analyst, working on the product team, and then got a promotion to be a product manager for our data science and our machine learning products, because the company was pretty focused on data engineering, but we wanted to expand into data science and machine learning.

From there, the story gets a little bit rocky. I wanted to move to Singapore, so I took a job working for Boston Consulting Group as a data scientist in Singapore. And I got accepted and whatever. I was on my way there and then this crazy thing happened where there was a global pandemic. So on my way to Singapore, I got stranded in the Philippines for a couple months, had to come back to the US, and had to look for a new job. I ended up working at a database company called Imply and then I joined WhyLabs about a year after working at Imply. So it’s a very convoluted story for how I got there.

7:26

Alexey

**So what did you do before WhyLabs? What was your role?**

7:30

Danny

Before WhyLabs, I was a field engineer. There's a bunch of very related, kind of abstract titles that are all related in the space, like Solutions Architect, Solutions Engineer, Sales Engineer, Field Engineer. Basically, what they mean is, “You work for a vendor and that vendor’s product is so technical that a salesperson can't sell it by themselves, so you need somebody who specializes in the technical side of the product to be able to talk to customers.” My joke that I would make when I was working at Imply was that I was there to answer questions when my sales rep didn't know what they were talking about anymore.

# What an MLOps Architect does

8:11

Alexey

**Was it a joke or? [chuckles] Sounds like a job description. Okay, so you first did law, then stats major, then you worked as an analyst, a PM after that, then as a field engineer, and now as MLOps architect. So you’ve worn quite a few hats, right? You’ve tried quite a few different things. So what do you do today as an MLOps architect?**

8:42

Danny

Alexey asked me this question over Slack before we'd scheduled this podcast and I think my very vague non-descriptive answer is how I managed to land getting interviewed on the podcast, because the answer is that it’s still very much blinding. The very short answer is that now, my role is primarily focused on doing something very similar to what I was doing before, but not as extensively customer-oriented. Basically, my skill set is that I am very comfortable communicating. I'm very comfortable navigating customer interactions.

I've spent a lot of time in this market, so I understand what's happening in the market. So I can both do the business stuff *and* I understand the technical stuff well enough to be able to get by. I haven't actually written any very serious code in a while, but I can throw together a demo, and importantly, I understand the basic concepts that are necessary to be able to have these conversations. So anything that requires both a technical understanding *and* business understanding is basically what I do now.

9:41

Alexey

**I think you're being quite humble, because like the demo you did – the webinar, the workshop you did – About WhyLabs was quite technical. You would create a Kafka stream, right? You would explain how profiles work, how they are mergeable – there's a low level, I would say, to some extent. But still, it's not “Oh, just some technical stuff.” Not every data scientist knows these kinds of things.**

10:08

Danny

I'll tell my dad you said that. Thanks a lot. Maybe he’s tuning in. Probably not.

10:14

Alexey

**Maybe. Let's see if there are questions from him. So this is what MLOps architects do, right? It’s a person who can talk with both customers and business and technical people in the team.**

10:32

Danny

Yeah. I mean, I don't know if this is what the role would be in general. This is what the title we decided on for me was. Because when I joined WhyLabs, we were still a seed stage company. Everybody else's title was engineer or I think just Bernice – I think we had just the one data scientist – and then the founders, the C suite. I was the first person who didn't found the company, whose job wasn't building the product – which meant that I was doing really anything and everything that wasn't building the product. The “architect” in the title is really meant to convey the fact that I work with our customers, with our partners, with even internally within our company, with helping people understand the landscape.

I don't do a whole lot of engineering these days. I'm not building scalable things. But I understand the trade-offs involved in different architectural choices and in different tooling choices. And I understand the tooling market out there from having been in the space for so long. I can help people navigate and help our customers navigate their tooling choices in a way that's helpful for them. Because I don't know if you've seen the 2022 Big Data and AI Landscape diagram that Matt Turk puts out. It’s a little bit scary to look at these days.

11:47

Alexey

**Is it the same one from Gartner or is it a different one?**

11:51

Danny

Oh, no. Here, I'll pull it up. It's this massive – we like to call it a NASCAR diagram, because it literally just looks like the side of a NASCAR with so many different logos on it. Here, can I share my screen?

12:05

Alexey

**No, because for people who listen to this, they will not see this. But maybe you can share the link and I will post it in the chat and in the show notes.**

12:17

Danny

Alright. People can just look up “Matt Turk data landscape” and they’ll find it.

12:20

Alexey

**So it’s just like a big image with a lot of small logos, right?**

12:25

Danny

Right. And it represents all of these companies that do ML, MLOps, ML as a service, data, data engineering ETL pipelines and [cross-talk]

12:35

Alexey

**And you really have to zoom in to see the logos, right? Because otherwise, it's just a big blur.**

12:38

Danny

Right. Exactly, exactly. That’s the one.

12:41

Alexey

**So you're saying that you actually came up with this title – you invented it for the job that you were going to do.**

12:49

Danny

I would say so, yeah.

# The popularity of MLOps Architect as a role

12:50

Alexey

**Have you met other MLOps architects?**

12:53

Danny

I think the closest is AI Architect, and those are people that I tend to work with pretty closely at our customers that are large enough to have somebody in that role. But no, I mean, I've met ML Engineers, or MLOps engineers, and I tend to work pretty closely with those people too, because they're usually building out the MLOps platform at a larger company. But yeah, I think MLOps Architect – we can look on LinkedIn, but I think I might be the only one.

13:20

Alexey

**Yeah. Because my next question is, “How many people are out there with this title?” But I guess you don't know this. I was also wondering how many people there are who do similar work as you. I guess, in some companies who are also in this field – the MLOps field – maybe I don't know, Iterative or others, who have some solutions and a Sales Engineer or Solution Architect, who are doing similar stuff? Right?**

13:50

Danny

I would say so, yeah. I think there are some limitations in terms of thinking about my role purely in the same way that my role was at Imply. At Imply, really, my job was pre-sales. I would go in with the account executive, I would do the demo, I would run the proof of concept with the customer – which required a certain amount of technical skill. But when things started to really take off, after the deal was signed, when it was time to really do implementation, I would just hand it off to a solutions architect and go on to the next opportunity.

At WhyLabs, I was the first person hired on the go-to-market team and I was the only person on the go-to-market team for about six months. So I was doing things like our DevRel, evangelism, developer advocacy, I was doing our product marketing, I was doing our social media and event and community management and like all of these different go-to-market-related roles. That was very overwhelming.

Thankfully, since then, we've hired a Customer Success Data Scientist to do post-sales, a Solutions Engineer to do pre-sales, an Evangelist to take over a lot of the community management, DevRel Dev Advocacy stuff. So I'm mostly still just helping them onboard, helping them get their footing, giving them guidance and direction in terms of how to be successful and then doing a lot of the product marketing and kind of figuring out how we want to decide the messaging directions on our product. It's still a little bit challenging to strictly define the role.

This is like an ongoing conversation that I have with the leadership of my company to figure out “How can I be most helpful given the skill set?” Because I know I can be flexible. I can do a lot of different things and the question is just “How can I help the most with the skills that I have?”

# Convincing an employer that you can wear many different hats

15:35

Alexey

**With this flexibility, when you say you can do whatever needs to happen, so you need to do DevRel – you just go and do DevRel until you hire a DevRel person, right? Then you need to do community management before you hire a community manager. And now you're taking care of product marketing, and then you hire a product marketer who will take care of this. And then I guess you could just chill and I don't know, do interviews. [chuckles] [Danny laughs]**

**But yeah, with this flexibility, the question I have is “How can you convince a potential employer that you can do all these things?” Because it's a lot of different – not random things – but they are different things. How can you convince someone that you can do these things?**

16:20

Danny

Totally. Thankfully, it's not a question I have to ask myself right now, because I love my job.

16:26

Alexey

**[chuckles] you’ve already managed to convince them.**

16:29

Danny

[laughs] Yeah. Well, I think the great thing about working at really early stage startups is that you do really pick up this “whatever needs to get done” mentality and it's not 100% whether you can pick it up with somebody from talking to them. But there are some signs that hiring managers at small startups can use to figure out like, “Is this person (I hate this term) “go-getter”” It’s so dumb.

16:57

Alexey

**Go-getter. [chuckles]**

16:58

Danny

Well, yeah. But that's what it is, right? It's like somebody who's willing to pick up the slack and do whatever needs to happen, right? This person who isn't like, “Come in and tell me what to do, like ‘Assemble this part in this way,’” but somebody who will look at the big picture and figure out what needs to be solved there. At least for early stage startups, I think they can look back at my track record, not that I'm looking for a new job anytime soon, but I don't think it would be that challenging if WhyLabs had to go down and I had to look for another job. I don't think it would be that challenging for me. Now, if I wanted to go work at something like a FAANG company, I think that'd be impossible. I also like… [cross-talk]

17:38

Alexey

**They will torture you with these LeetCode-style questions first before you can even show that you can do other things.**

17:45

Danny

Exactly, exactly. And I don't think I would really enjoy working at a company like that, because I also wouldn't be able to do everything in the way that I can do everything at a startup. The issue when you've hired somebody for every role, then that means that everybody needs to have just one particular role. And I think it's kind of exciting and fun to wear so many hats and to get pulled in so many directions. That's obviously also stressful and hard and there's disadvantages to it, too. But I wake up every day not knowing what I'm going to do that day and checking my Google Calendar to figure out what meetings I have.

# Interviewing for the role of an MLOps Architect

18:21

Alexey

**That's exciting. Do you remember how your interview actually went? What were the questions? Who conducted the interview? What kind of questions did they ask? How did they decide that you're good for this?**

18:37

Danny

Yeah. The interviews got pulled in a couple different directions because they wanted to validate a number of different skills. We did the Amazon style, intensive interview process where I had five interviews spread out, I think, over two days (but true Amazon style would be five interviews in one day).

18:56

Alexey

**It’s a lot of leadership principles, right?**

18:59

Danny

Yeah. A lot of the way that WhyLabs operates comes from Amazon.

19:04

Alexey

**Most of the founders, if not all of them, came from Amazon, right?**

19:08

Danny

Three out of four, yeah. The interview that stood out to me a lot was with our data scientist, Bernice, because I was just so excited about the company after this interview. That doesn't always happen, right? It's not every conversation you have with people, even a good company or a good role, that will get you excited. But Bernice asked me some really interesting questions. We got to really dive into some machine learning theory and our mutual statistics backgrounds, and that way of analyzing machine learning was really helpful in that interview.

She introduced me to some new concepts, which I found really, really interesting about particular ways of doing splitting in random forests and decision trees, and how to speed that process up. And I just realized that I met somebody who was smarter than I was, in the same way that I was. That's always a very exciting thing for me to find in other people, because it means that I can learn a lot from them. So Bernice asked me a lot of data science- and machine learning-oriented questions. Maria, the Chief Operating Officer, who's the person that I directly report to, asked me a lot of business- and sales-oriented questions. The CEO asked me kind of directional, high level overview questions. And then I think I also had interviews with our Chief Product Officer, Sam, but I don't totally remember what those questions were. And I think Andy, our Head of Engineering also asked me questions, but I don't remember them either.

20:41

Alexey

**So basically, the entire company – all the staff – interviewed you. Because you said there are four founders, and…? [cross-talk]**

20:49

Danny

Yeah, so it was the four founders and Bernice. There was a big engineering team – not huge – but there were five or six people on the engineering team at that point, and I only got interviewed by the Head of Engineering. I'm sure if I practiced and read Cracking the Coding Interview and did the code for a couple of hours a day, I could do a traditional software engineering interview. But like I said, that's not really what my skill set is. The coding that I do is really, I would say, not engineering. I program and I code, but it's like scripting, rather than trying to build scalable processes and software that will scale.

21:31

Alexey

**I guess, at some point, they gave you an offer and then you needed to decide, “Okay, do I want to join this company and do this particular thing – join this company in *this* role or not?” How did you make this decision that you wanted to do all these things? Wasn't it scary? Like, “What if I couldn't do this thing? What if I'm not a good product marketer?” You didn't have that much community management experience either back then, right? Wasn't it scary for you?**

22:04

Danny

Yeah. I remember when I made the decision. There was still the negotiation period afterward and all this stuff. But there was a point that I mentally committed to it. It was about 50 weeks ago – my anniversary is coming up, so it's a fun time for us to be having this conversation. Yeah, it was the summer solstice and I was backpacking in Denali National Park for four days – completely disconnected from the world. Seeing grizzly bears and getting to the tops of mountains and hiking on glaciers. Spending the time out there, I had to struggle with what I wanted to do.

I was at Imply – I was being very successful as a field engineer. I basically had an offer in hand to get promoted to a management role in that field engineering organization. But I thought about what that would mean for me to be a manager of field engineering. And I realized that although I wanted and still want the experience of management and understanding what that's like, I would get a lot more exposure to a lot more skills and a lot more interesting experience by joining such an early stage startup. Then the big gamble is that early stage startups tend to go out of business, right? It's pretty hard. So you're making a really strong bet primarily on the people.

I thought about the conversations that I had in my interviews – I thought about the people that I'd met at this company and I realized that I was just so impressed with them. I knew that no matter what the market was going to throw at us, no matter how our product evolved, that this group of people was capable of making the right choices and that they were going to build a really strong company and organization. I figured out what I wanted for myself, which was – I want to do a lot of things and experience hypergrowth at a really early stage startup and also, I want to be working with these people and I want to have them in my company and in my life.

23:59

Alexey

**And it all occurred to you in the mountains, while watching grizzly bears, right?**

24:03

Danny

Yeah, I think it was more on the mountain tops rather than… about the grizzly bears, I was mostly just afraid, rather than thinking about WhyLabs. [laughs]

24:12

Alexey

**[laughs] How far was the grizzly bear? Was it a metaphorical expression, or you actually saw one?**

24:18

Danny

No, it was certainly not metaphorical? No, no, no. There's *a lot* of grizzly bears in Denali National Park. The closest I ever saw them, I would say it was maybe (you probably want this in metric, right?) I would say like 40 meters, but on the other side of a river.

24:38

Alexey

**Oh, okay. But that’s pretty close, right?**

24:40

Danny

Yeah. Well, so that was pretty close. It’s the closest I’ve gotten to see them. Apparently, one walked through our camp while I was in my tent – right by my tent’s door opening, according to my friends. So I guess I was within like a couple of meters of a grizzly bear at one point. But yeah. That I did not see.

# How Danny prioritizes work with data scientists

25:04

Alexey

**I guess these things make you wonder what the future holds? And “Is what I'm doing for me or not?” Right? [chuckles][Danny laughs] Like, “Do I want to be a manager of field engineers or do I want to do and try many different things.” Okay, I see that we have quite a few questions, so maybe we can cover a couple of them. The first question is from Eminy – sorry if I mispronounce your name. The question is “MLOps Lifecycle Management has a lot of topics to cover and tooling to optimize – from training to inference. How do you prioritize your work with data scientists?”**

25:49

Danny

Yeah, great question. WhyLabs… Alexey, I promise this is not a product pitch. [laughs] WhyLabs is a company that's really focused at the end of the lifecycle. Or rather, there's not an end of the lifecycle because it's a cycle and it's continuous – at the end of the line, if you were to draw this linearly. Because what WhyLabs does is provide monitoring and observability for models that have been deployed into production. After you've taken your model, you've trained it, you've experimented on it, you figured out the model you want to deploy, you deploy it. Now, after you've deployed it, you want to make sure that it's still operating effectively, that its accuracy is high (I mean, hopefully, you're not actually using accuracy here.) That like F1 is high – whatever accuracy metric you're using. That's really where WhyLabs comes into the picture. I would say what I think about what I need to prioritize in terms of *my* conversations with my customers, what I need to prioritize in terms of my understanding of the tooling, is basically the further away it is from productionization, the less I care about it.

When I'm talking to customers, what we spend most of our time talking about is “What does your inference process look like? Are you using BentoML or using SageMaker, UDops, TeachableHub – there's like all of these different tools, be they open source self-deployed tools – FastAPI, Flask – or managed service providers around deployment, spend a lot of time on that, less and less time as we go through experiment tracking, weights and biases – don't really have to think about them very often. Even further back, like the model training – I don't even know what people use for models. Probably still Scikit-learn, TensorFlow and the old basics. I don't know, maybe this fancy mixture. [cross-talk]

27:34

Alexey

**XGBoost.**

27:35

Danny

Yeah, right. That's what I was using in college. [chuckles] I don't think it's changed that much. So I really focus on the tooling that's kind of very late in that process because it tends to be what's most relevant to me. As we're expanding, and this is very early to be talking about this, but we're exploring more how we can help not just data scientists, but also data engineers. And how, even for data scientists, we can help them understand the entire data pipeline. Because when you determine that there's a problem with your machine learning model, normally, the problem is much earlier in the data pipeline. When you've got data drift, it's because it's happening in the real world. When you've got a data quality problem, it's because something upstream of your model has gone wrong, which means that to have observability into the model, to be able to solve problems in the model, you need to be able to look at everything upstream.

So what I'm staying really up to date on is, “What are people doing in terms of pipelining? What are they doing in terms of orchestration?” And really making sure that I still understand the whole data engineering process end to end. Fortunately, my work at Qubole was really helpful in making sure that I had a strong grounding in that. But yeah, understanding all of the ETL and transformation and all of the work that happens *before* data even gets to a model has been really near and dear to my heart these days. But that's kind of how I prioritize – based on what's most relevant to WhyLabs.

# Coming to WhyLabs when you’ve already got something in production vs nothing in production

28:59

Alexey

**Would you say that you usually talk to customers who already have something in production? Customers who have already trained a model, who figured out how to best deploy this model – they already deployed it, and now they’ve started to maybe experience some drift or models going crazy or something like this. Then they think, “Okay. How can we solve this problem?” Then they go to you. Would you say that this is usually the case? Or sometimes companies/clients come to you without having anything.**

29:31

Danny

Those are definitely the easiest conversations – when they come and they've already deployed a model and they've already felt pain. Because then, they have pain that they need to alleviate and WhyLabs can help with that. More often than not, we try to be proactive and preemptive instead of REactive. So, if we think companies have models in production at all, we'll talk to them even before they've experienced that pain to kind of coach them through our approach and our thinking. Sometimes they understand and they're like, “Oh, yeah. We monitor our applications in production. Of course we should monitor our models.” Other times, they're like, “Oh, it's not a priority for the organization.” And then we say, “Okay, we'll talk to you in six months when one of your models goes down.” [cross-talk]

30:14

Alexey

**This is what you told me.**

30:16

Danny

Yeah, basically. [laughs] And go figure – you did come back and talk to me about it again later. I don't know if it was quite six months. We have some conversations with people who are still at the model development stage. But you know – it's just not the most useful stage for us to be involved, because they're trying to solve different problems.

# Market awareness regarding the importance of model monitoring

30:39

Alexey

**Okay. You said you’re trying to be pre-emptive and this is also part of your role, right? You are trying to spread awareness about this problem by talking on different podcasts about model monitoring, why it’s important, what can go wrong if you don't monitor your models, things like this. This is what you used to do, right? Or are you still doing this?**

31:05

Danny

I would say I still do it to some extent. We have a full time evangelist now. Our data scientist, Bernice, who I talked about earlier, does this to some extent too. I would say we've seen a really good increase in market education on this topic. People are much less often asking “Why do I have to monitor my model?” And are more often asking “How do I monitor my model?” So that's a big improvement and a big change, but there's still some market education to be done, I would say. Especially outside of major tech hubs.

31:37

Alexey

**Yeah, there are quite a few companies that do monitoring and all of them are spreading awareness. People now know about this problem more than, let's say, one or two years ago.**

31:50

Danny

Exactly, exactly. Actually, funnily enough, one of the reasons that I was so intrigued about WhyLabs is because while I was working at Imply, I was actually (this is back in 2021. Second half of 2020, first half of 2021) I had actually been going around and doing a little bit of evangelism for Apache Druid, which is the open source project that Imply is built on – it's basically a fast analytic database. And I was going around and saying, “Hey, you can use Apache Druid to do machine learning monitoring.” I would speak at conferences and meetups and talk about how you could architect Apache Druid as the database to do anomaly detection and alerting and figure out analytics for machine learning.

Then I talked to WhyLabs, and I was like, “Oh, you guys are using Apache Druid for machine learning monitoring. I literally have been going around and talking about people doing this for the past year.” So it seems so serendipitous, and that's another big reason why I joined WhyLabs.

32:55

Alexey

**That's what you use under the hood, right? Druid?**

32:58

Danny

That's one of the things that we're using under the hood. It would be a little bit hard to just use Druid for this because Druid is just the database… [cross-talk]

33:07

Alexey

**And then things built on top of that, right?**

33:08

Danny

Right. So you need things like dashboard UI on top of it, to be able to see these changes, you need anomaly detection… But yeah, in terms of having a really fast, scalable database, Apache Druid is really useful. And we have some talks where we talk about this as well. Our data engineer, Drew, talks about how Apache Druid is a really good fit for this particular problem.

33:32

Alexey

**And that's not part of the open source thing. It's not a part of WhyLogs, right? You just use WhyLogs for creating profiles, and then these profiles are sent somewhere, and this somewhere is your proprietary solution, where you keep track of all these things. And then you under the hood, you use Druid, right?**

33:49

Danny

Yeah. If you go to a WhyLabs.ai/observability, you can see actually everything that I'm talking about. There's an architecture diagram that explains exactly what you're saying. So WhyLogs is open source – it takes in data, it creates data profiles, which are statistical summaries of that data. Those profiles get sent to WhyLabs. Under the hood, they get sent to… Well, there's an intermediate process that cleans them and does whatever, but then it lands in this Apache Druid database, and then we've got visualizations, anomaly detection, trigger alerts – all of these things built on top of it.

# How Danny (WhyLabs) chooses tools

34:25

Alexey

**Another question we have is… I don't know if you can answer this because for the previous question, I think you clarified that you're a bit outside of these decisions. The question is, “On what basis do you make your choice of tools? Do you decide to use Kubeflow (which is open source) versus Vertex AI (which is commercial) versus MLflow, etc.” I don't know, do you get to make these choices? Do customers talk to you when they are not sure what to use?**

34:54

Danny

Yeah, it's a really, really good question – especially the builder vs buy question is a really early important decision to make. Like, “How invested are we in building out this feature or this tool ourselves when WhyLabs costs $50 per model, per month – and you're paying your engineer God knows how much money?” It's a question we get a lot, especially for our space. Now, obviously, I'm not the one in charge of making that decision. But I work with people – I work with data scientists and with machine learning engineers – so that they can make the case to their managers.

These are people who have probably never bought software at a commercial level before, so I help them make the case to their manager and talk to them about what their manager will care about – KPIs, MVOs, company risk – all of these things that are like garlic to vampires for machine learning engineers and data scientists, but which are really important for navigating a business or an organization. And I think that's how I can be most useful to the customers that I work with.

36:02

Alexey

**But I guess this is more in the context of whether they need to go with WhyLabs or with some open source alternative. But when it comes to model serving, which is not what you do as a company – let's say you can use Kubeflow for pipelines, or Vertex AI, or I don't know, a company can just build this in-house.**

**Do your customers also ask you for an opinion about this? Or do they usually already have something? Like, they already have Vertex AI when they come to you and they say, “Okay, I have this pipeline on Vertex AI. I have this model deployed in this way. Now help me monitor it.”?**

36:47

Danny

Yeah. WhyLabs is platform agnostic and we integrate with a lot of different tools – basically, any tool that can run Python or Java – so anything with Spark, or Scala, or Java, as well as anything in Python. We tend not to be too strongly opinionated and we think our customers should pick whatever is best for them. Yeah, usually people at least have an early choice made. By the time they want to do monitoring, usually they're already doing it inference. So I don't think I have a ton of influence with the customers that I work with on what their inference architecture looks like.

Now, by the time we've had a relationship for like six months or a year – if they're doing an architectural review, and if it's time for them to switch up the work that they're doing, and they're examining their toolset – at that point, usually, I can be more helpful. Because most of the time, they're pretty in the weeds using this one thing – they're very deep and not very wide on tooling exposure, versus I'm very, very wide and not at all deep on these tools. I understand each one of these tools and how they interoperate and what the advantages and disadvantages are. But I have maybe launched them or played around with them once. I don't have the deep expertise that people who really spend their time in this tool have.

# ONNX

38:01

Alexey

**Another question we have is about the ONNX standard. Have you heard about this? It's getting wider adoption, so would you say many of your customers use it or is it still Scikit-learn models and pickle deployed in Flask?**

38:20

Danny

Yeah, I remember when ONNX came out, it was very interesting and very exciting because we had been using PMML as kind of the standard markup language for this. But ONNX kind of helped to bridge that gap for interoperability of these libraries. I don't see a ton of people using ONNX. But again, I'm really not that involved in the model development stage and I think that's a lot of where ONNX would be seen.

I think ONNX is really cool in theory and probably gets used a lot at really big organizations that have a need for really diverse tool sets. But especially with the SMB and commercial customers that I work with, they tend to pick one tool set and run with it – versus ONNX, which is really meant for interoperability and the ability to communicate across different ways of training models.

# Common trends in tooling setups

39:10

Alexey

**What kind of setup do you usually see more often than other kinds of setups? If you can talk about this, of course.**

39:18

Danny

Yeah, it's nothing propriety. It's not that I can't talk about it because I have a contract. But it's just that there is no “usual” in machine learning operations.

39:27

Alexey

**So everything is different?**

39:29

Danny

Yeah, I would say the only slightly common trends that I see is people who really like AWS will use an entire AWS stack, people who really like CP will entirely use a Vertex stack, people who really like Azure really will just use an entire Azure machine learning stack and not touch too many other vendors. But that's such a small… it's not that there's consensus – certainly not that there’s consensus around only using cloud vendor tools. Because a lot of those tools are kind of lacking in a lot of ways. There's a lot of feature gaps in a lot of those technologies.

Yeah, I definitely see super diverse setups. This is one of the things that IA (the AI Infrastructure Alliance, which both DataTalks.club and WhyLabs are associated with) tries to help customers navigate is, “How do you fit all of these different tools together? And which tools are swappable?” As you mentioned, there are other companies out there – we're not the only machine learning monitoring and observability company. There are other companies out there as well. How do you figure out which tool is right for you? And even before that, how do you figure out where the tool fits in the landscape? And that can be really daunting to navigate, especially for somebody just starting to get their MLOps architecture together.

# The most rewarding thing for Danny in ML and data science

40:49

Alexey

**Yeah, thanks. What do you like doing most in machine learning and data science? What is the most rewarding thing for you in the field?**

41:00

Danny

Good question. I really like getting exposed to new, interesting ideas within the field – I miss doing research basically. Not that I was ever that strong of an academic, but it gets me really excited when I find a topic that I can obsess about and then I'll just spend hours figuring out and researching. Most recently, that topic for me was fairness and bias in machine learning. I talked for a while about how explainability is the thing that everybody uses for this and then I thought about it and was “But why? How does explainability actually relate to fairness and bias?”

I put on my old dusty paralegal hat again and started researching the law around discrimination and bias and I ended up getting really deep into this and realizing that, actually, the best things we can do to prevent bias in machine learning models is, rather than try to focus on explainability, to focus on segmentation and the ability to separate – to be able to track disparate impact. So yeah, I really love the ability to just deep-dive into a topic. A few years ago, AutoML got me really interested – and hyper parameter tuning. [chuckles] It turned out that they were the same thing, which was pretty funny to realize at the time. So just getting to like really geek out and go really, really deep on a topic in ML is still the most exciting thing for me.

# Danny’s secret for staying sane while wearing so many different hats

42:26

Alexey

**You do, or you took part in doing DevRel, product marketing, community management. Also you just mentioned that you're doing a bit of research, like figuring out how fairness and biases work, like “Do we need to have explainable AI or not?” Things like this. That's quite a lot of things and it's pretty wide, right? So how do you manage to do all that? How do you manage to do all these different things and still stay sane? What's your secret?**

43:00

Danny

That's like a big assumption that I stay sane. [laughs]

43:06

Alexey

**[laughs] Okay. So that's your secret?**

43:07

Danny

Yeah. [chuckles] No, no. I have a lot of productivity management techniques that I use. I try to stick to inbox zero. Right now, my inbox is a little bit of a nightmare. But I really like using my inbox as a tracking to-do list. Then I also have another to-do list in Google calendar that I keep associated with it. I try to keep my tabs under control, too. Because I know it adds a lot of mental stress to me.

43:34

Alexey

**You try. Right? [chuckles] Has it worked?**

43:35

Danny

It's decent, I would say. It's not the worst.

43:41

Alexey

**How many tabs do you have open now?**

43:44

Danny

Well, the thing that I do, which is something I don't see a lot of other people doing is, I actually (this is crazy) use Windows. I don’t mean the operating system, but I have multiple windows open and each window is a project, effectively. So I have one window for something [cross-talk]

43:58

Alexey

**Like workspaces, right?**

44:00

Danny

Right! Yeah, exactly. That's how Apple refers to it. So I've got kind of my catch-all Gmail, Google Calendar, whatever article I want to read – one-off stuff. And then, I've got one workspace that's dedicated to explainability, fairness, bias, and that research that I'm doing. One that's related to getting ready for our evangelist sage to come on DataTalks.club in a couple of weeks and making sure he's ready for that. And I just like using these workspaces in this way. I would say that I definitely think I could be more effective at each thing if I was only doing that thing, but I don't think that I've hit the limits of my ability to wear multiple hats and do multiple tasks.

# T-shaped specialist, E-shaped specialist, and the horizontal line

44:46

Alexey

**Would you say that sometimes you wished you were more like a specialist – you would just focus on explainability or you just focus on DevRel – or would you quickly become bored?**

44:59

Danny

I don't know whether I would become bored. I'm not sure. But it's not an appealing idea to me right now. I think maybe it will be later. We were just talking about AutoML and hyperparameter tuning – are you familiar with the explore/exploit trade-off and optimization theory?

45:16

Alexey

**I've heard about this, yeah. It's in reinforcement learning, right? I'm more like an exploiter. Like, exploration would be trying different kinds of food – you go to different restaurants and me as a person, I find that, “Okay, I like this type of food – Spaghetti Bolognese,” and I will just stick to this. So I'm more of an exploitative kind of person. But an explorer kind of person would just keep trying different types of food and wouldn't stop just on one thing. So you're saying you're more like an explore kind of person?**

45:49

Danny

Well, the really useful finding from optimization theory is that you need to balance the two of these. If you've got a bounded amount of time over which you can optimize around a search space, your optimal strategy is to start off very exploratory and then to start taking more and more advantage of the things that you've learned in order to be able to exploit and get the maximum value. This is like Thomson resampling –one of the easiest and best solutions to the multi-armed bandit problem. It's *way* simpler to implement than a full reinforcement neural network. And also, it's Bayesian, which is cool. It works really well for this because it does exactly that. It starts off by exploring the search space and then as it learns more and more, it starts exploiting more and more, and then by the time it's done, it's just exploiting full-time.

This is pretty abstract – optimization theory, Thompson resampling, whatever. You can think about this in our lives, too. I've got a bounded time that I can use to try to figure out what makes me fulfilled, happy, joyful, ecstatic – whatever it might be, right? Whatever I'm optimizing around, whatever that optimizing function is looking for, I can think about the period of my life that I'm in and whether I should be focusing on exploration and exploitation. And at this point in my life, I am really interested and I get a lot of value out of learning more and more, and trying out lots of different things.

Actually, there's some good research that says that as we get older, we seek novelty less and less. We naturally evolve into being more exploitative, which sounds very negative, but not like exploiting people, but exploiting our knowledge about the world. We evolve into being more exploitative as we get older. I think that's really powerful and a really useful mental framework for me to keep in mind as I think about “Oh, would I be happier picking one thing and specializing in it?” I think, likely – probably. That’s what everybody else does. So I don't think I'll be like this for the rest of my life. But right now, it's really powerful for me to get to explore and try lots of things.

47:51

Alexey

**What's cool about this is there are places where you can do this. It's not like people hire for a role that is super defined, like “You need to do X, Y and Z.” They know that it can be pretty different and then you just pick whatever needs to be done? It's cool that there are positions like that – where you can actually do this.**

48:16

Danny

Exactly. Yeah. I'm super lucky to have developed the skill set that kind of bridges the gap between two very important, but often very disparate things. I used to have a joke that I'm neither scared of numbers, nor scared of talking – being scared of neither of them is a rare combination.

48:37

Alexey

**Okay. Would you say there is a demand for people like you who (pardon me saying it this way) people who can do a lot of things, but not well? [chuckles]**

48:56

Danny

[laughs] I think so. I think the consensus that I've seen in the industry – it’s kind of a cop-out – but there's this concept of being T-shaped as an engineer. That means kind of doing both – having a broad overview and then really deep expertise in something. I would say that I'm kind of like a horizontal line-shaped, rather than a T-shaped, but the nice thing about being horizontal line-shaped is that it's easy for me to find something that I want to really get deep into. I would say there's enough demand for me that I'm not that concerned about not being able to find a job in the future. And I'm very lucky to be in a position to be able to say that.

49:34

Alexey

**I also heard about the comb shape –like a horizontal bar with many things sticking up. When you know many, many different things, not super deeply, but enough to get the job done. [cross-talk]Maybe that’s you. Or maybe like an E shape, but rotated horizontally. Yeah, okay.**

# The importance of background for the role of an MLOps Architect

50:00

Alexey

**If somebody wants to follow your path and wear many different hats and do pretty much whatever needs to be done, what would you suggest they do? How should they go about finding a role like this? What kind of background should they have? Is it even important? Does it really matter?**

50:23

Danny

Good question. I think, background – not in the sense of academic achievement, or previous titles – is not important. However, background in the sense of being able to do these things matters. If you want to be successful and if you want to provide value to a company, you don't need to know exactly how to do it, but you need to know that you can figure out how to do it. None of us know, off the bat, how to program, or even when we start a new project – I spend most of my time Googling and on StackOverflow. But I know how to find the answers to the question that I have. I think that's a very important skill – being able to find answers to questions that you have and knowing how to navigate the massive world of information that's in front of us.

Being an effective Googler is a really useful skill. More broadly, I think it's definitely very possible to break into this. If people are genuinely really, really interested – if an engineer wants to try out getting more into the business side, product management can be a really rewarding role. Sales engineering can be a really rewarding role. You will find that there's a lot of demand for being able to bridge that gap. If you're on the engineering side, you'll need to develop more of your communication and business skills. For people wanting to go the other way, I would say it's much more… not easy, but it's not the hardest thing to do. Going the other way, I think it is much more of an uphill battle. If you're an account executive and you want to become a sales engineer, I think that's really challenging to do. But it's certainly possible, because you can learn and you can upscale.

People who are on the business side but want to be more tactical, there are tons of resources out there, like DataTalks.club has Zoomcamps, there's like unlimited courses and books and all of this information out there. It's really just about like sitting down and committing. I think a big component of this is a growth mindset – looking at yourself in the world, not as a static thing that exists and will always do the same thing. Our brains are incredible. Neurons that fire together, wire together – it's self reinforcing. And you're capable of changing who you are very fundamentally, if you really put the intention and the practice into it.

52:37

Alexey

**So you need to have coding skills, you need to have communication skills, you need to be good at Googling things, and you need to have a growth mindset. I'm really curious about this “being an effective Googler” because with coding, you can find a course on freeCodeCamp or Code Academy or whatnot and learn this, or go into a boot camp. But how do you learn to Google? There are no courses about this, right? Are there courses about being an effective Googler? I haven't seen such things.**

53:07

Danny

You know, if there's not, we should start one because it's a very valuable skill. I think those of us who grew up with technology have a little bit of an unfair advantage, because when we're younger, our brains are more plastic. There are studies that show that millennials and Gen Z people don't memorize things as well as people in previous generations, but they're better at memorizing the path to be able to retrieve information. Like they'll remember the series of jumps that they made via referencing in a book or Googling how to get somewhere, but they won't remember the actual thing itself. Which is very nice when you can plug your brain into the global human connection of the internet, and very inconvenient when you're backpacking in Denali National Park and you don't remember whether grizzly bears are attracted to the smell of food. Turns out they are.

53:57

Alexey

**Yeah. Or the internet simply doesn't work and then you're stuck at home with a computer that works – it's connected to the power line – but like, you have your BS code open, you have your Python, but “What to do next? How do I start?” [chuckles]**

54:16

Danny

Yeah, I mean, I've tried coding on planes before and it's *very* challenging. I'm just so used to Googling everything. But I would say… there might be a course. If not again, Alexey, you and I, we should get on this because I think this is gonna be a big moneymaker for us. But I also think that this is one of those skills that you can just pick up. Stop asking your coworkers things unless it's things that aren't Google-able – like if it's internal things, then you're not gonna be able to find them on Google. But if you want to figure out a particular Excel thing, take the extra time.

In my childhood, growing up in elementary school, the teacher would not tell us definitions of words. She would tell us to go look it up in a dictionary. And I think it's kind of the same. If you want to get good at using a dictionary, if you want to get good at using Google, just stop asking other people for answers and start finding them yourself. And it's hard. And it's a challenge – like learning a new language. It's an uphill battle, but your brain will wire itself in a way that makes it easier.

# Key differences for WhyLogs free vs paid

55:17

Alexey

**Okay. I think we should be wrapping up and then maybe – since we spend quite a lot of time talking about WhyLogs and then your name here is WhyLabs team, and we know that because you pay just for one license. [chuckles] The question is about WhyLogs? What are the key functional differences between WhyLogs free versus paid?**

55:42

Danny

Is this really a question we got, Alexey?

55:44

Alexey

**Yeah, this is a question. I'm not making this up. It is a question from an anonymous person who is asking that.**

55:50

Danny

Okay. So there is no paid version of WhyLogs. WhyLogs is our open source library. So it's totally open source. Everything about it is completely accessible and completely free. Obviously, as all open source libraries are. WhyLogs takes in data – it can be text or tabular data or images, whatever kind of data – and it generates statistical summaries of that data. And those summaries are called Data Profiles. Those profiles, you can do a number of different things with them. In a course earlier, in this workshop earlier with Alexey at DataTalks.club, I showed how you can set constraints in order to get alerted when data looks different than you're expected to, how you can visually explore your data using the profile visualizer. One of the things you can do with these profiles is send them to WhyLabs.

WhyLabs takes in these profiles that you've generated with the open source and with these statistical summaries of data, it allows you to track changes in your data over time, which sounds very innocuous, but actually has a very powerful impact both for data scientists and data engineers. For data scientists, you can pick up on model performance degradation, model failure, data drift – all of these types of problems. For data engineers, you can pick up on data quality issues, like a spike in null values, breaking changes of data schema or data definitions. So you can pick up on these types of problems in a solution that's totally SaaS, so you never have to deal with infrastructure.

It's got a self-serve option – you can go to WhyLabs.ai/free and sign up for an account yourself and you'll never have to talk to me or a salesperson or anybody like that. You can just start using it right away and if you have just a single model or just a single data set, you can use it for free. So I guess the important distinction between WhyLabs and WhyLogs, is that WhyLogs is open source – it takes in data and generates profiles. WhyLabs is software as a service, and it takes in those profiles and it generates alerts, anomaly detection, and allows you to explore those profiles.

57:46

Alexey

**Yeah. Do sign up now, so maybe the team will finally afford having multiple licenses. [laughs] Sorry, I couldn't help but make this joke.**

57:59

Danny

[laughs] I'll start a Kickstarter, Alexey. If you want to donate to WhyLabs having multiple zoom accounts. [chuckles]

# Conclusion and where to find Danny online

58:07

Alexey

**Okay, yeah. Anything else you want to mention before we wrap up?**

58:12

Danny

No. I guess just that… I so love my career and it's been really, really rewarding professionally. It's been so engaging and interesting. I know it can be scary to make the jump, just like it's scary to see grizzly bears on the other side of the river. But it's really powerful to be able to take yourself into the edge of your comfort zone, and to push yourself to do things that are hard. It's fulfillment as a human being to be able to grow and expand and improve. To me, I think it's really valuable to be able to do that.

58:47

Alexey

**Okay. If you want to reach out to Danny – he's first in DataTalks.club Slack, so you can find him there. Also WhyLabs also have their own Slack community. I remember when I joined your community, the first thing I saw was a message from you. I think it was automated, because in the message, it assumed that we didn’t know each other, but we did know each other. So you'll see a message from – is it still you, or?**

59:11

Danny

Nah, I don’t send those messages anymore.

59:13

Alexey

**Oh, you don't send this. But yeah, there's another place where you can find Danny. Or on Twitter, I guess, right? Where else can people find you?**

59:20

Danny

Not huge on Twitter. I’m trying to get more active on it. But LinkedIn – I’m on LinkedIn all the time. Yeah. And you can just email me, too. Danny@WhyLabs.ai

59:31

Alexey

**Okay, I guess that would be it for today. Thanks, Danny, for joining us. Thanks, everyone, for joining us, too – for asking questions. Tune in – in a couple of weeks we'll have a workshop from WhyLabs. So keep an eye out.**

59:47

Danny

Cool. Thanks, Alexey. Thanks for having me on. Good seeing ya.