1:01

Alexey

**Today we will talk about open source and creating startups in open source and we have a special guest today, Johannes. Johannes is a data scientist and engineer and he is a co-founder of Kern. He likes machine learning, data management, and all the things that people build with these things. At Kern, they help data scientists with labeling and managing data effectively on a large scale. So welcome to our show.**

1:28

Johannes

Hi, thanks for inviting me.

1:31

Alexey

**The questions for today's interview were prepared by Johanna Bayer. As always, thanks, Johanna, for your help.**

# Johannes’s background

1:36

Alexey

**Before we start – before we go into our main topic of starting an open source startup, let's start with your background. Can you tell us about your career journey so far?**

1:46

Johannes

Sure. Happy to do so. I've had this typical technical background. I studied business, computer science, and data engineering. I worked as a data migration consultant during my undergraduate study. So I've already been quite early in my career in the data path, so to say. And I've been in the field of AI for roughly eight years now. Not professionally, but I started eight years ago with my very first touch point, which back then was this little web app called style transfer.

You could upload an image of a dog and then choose an image of Picasso and it was merged. From there, basically, I was super curious about how this works. And in my Master's degree, I first started a consultancy. This was like the first time I did something on my own. And this consultancy, at some point, kind of turned into Kern. We did a couple of consultancy projects and found interesting topics that were repeating and we started building software for that.

2:53

Alexey

**You mentioned that you studied data engineering. So it was actually a part of your degree?**

2:58

Johannes

Yes. Data engineering, basically, the name of the course is data engineering, and it's like a mixture of data engineering and also data science. So both fields, basically,

3:14

Alexey

**Which university was it?**

3:17

Johannes

It's the Hasso Plattner Institute in Potsdam. That's where I did my Master’s and Bachelor’s. I chose the smaller university.

3:30

Alexey

**Are you still in Potsdam? Or you moved?**

3:33

Johannes

No, I actually moved back to Bonn. I studied in Potsdam, but I moved back to Bonn, close to my family. [chuckles]

3:42

Alexey

**I'm surprised that there are actually courses for data engineering, because usually universities are… It's not always possible for them to catch up to these things. So that's really cool that you can do this already. Okay. So it started with style transfer?**

4:01

Johannes

Exactly. Exactly. It was a curiosity for me. It was basically that I just started generic programming roughly eight years ago – a little bit more than eight years ago. I was just still in the mindset of, “Okay, the thing I can program is if/else and so on”. I was thinking in that sense and then realized that you could do something like style transfer – it was just too interesting for me to dive into. So that was cool. [chuckles]

# Johannes’s Open Source Spotlight demos – Refinery and Bricks

4:33

Alexey

**So, Johannes, this is not the first time you appear on our YouTube channel. You appeared twice before that. We have this thing called Open Source spotlight, where we invite open source authors and ask them to demo their tools. You already did two demos. Can you tell us about these demos? By the way, we will of course, make sure that we add links to the show notes so you can also check them out.**

5:00

Johannes

Yeah, sure. I've been a big fan of the series myself. So I’ve already known about it before. When we did our first open source launch of Refinery, which is our flagship project that we're working on, I reached out to you guys and asked if we can show it. Refinery is basically a data visualization labeling tool specifically designed for engineers. If you, for instance, want to build a really cool natural language processing application, and you have just a web scrape, a ton of data, and you now want to look into how you can refine it (that's also where the name comes from) that's a really cool tool for it. One of the fuels, so to say, or the drivers of Refinery is automated heuristics.

For this, we have the second open source project, which is Bricks. These are the Bricks with which you can build your project, which is basically, a content library, so to say, where we collect for abstract things like sentiment analysis, multiple algorithms, both open source and both machine learning and regular programming, and so on. So those are the two open source projects – the bigger ones we have so far.

6:20

Alexey

**Can you tell us a bit more about Refinery? You said that it's a data visualization labeling tool, and it kind of refines your labels, or refines your data right? So how does it do this? What does it do exactly?**

6:33

Johannes

Yes. We basically had two questions in mind. Also, previous to the launch, we also already worked with a dozen of data science teams. One idea was basically, “What can I do if I just have the raw data and I want to quickly label a lot of it?” That was the first idea. We saw that if, for instance, you want to do something like sentiment analysis, you typically can use some techniques like active learning or integrating something like GPT or other tools to use a heuristic that can label data automatically but in a “noisy” way, so to say.

How can you combine that so that you basically have a framework? This is called weak supervision, which we did not invent, but we do make use of it. Basically, it helps to relatively quickly get training data instead of raw data. That’s one of the cool things about Refinery because it really shortens the gap between “I have an idea for some kind of project I want to do,” and “I have to baseline implementation.” That's the first thing.

The second thing about Refinery is that as soon as you have some automated labels, and you now want to dive deeper, you may also already have lots of manual labels. But we typically have seen that, in the real world, even the manual labels are extremely messy. Even if they are called “ground truths”, oftentimes, they're not the truth. [chuckles] It's also super, super valuable to dive deeper into the existing neighborhoods that they have – which of those are most likely wrong? Or what are the subsets of my data set where I'm really having headaches because there's so much mess going on? You can actually use those heuristics that, for instance, are part of Bricks, to quickly dive deeper into this.

Let's say we have a sentiment analysis and we see that the sentiment analysis is performing really well for short texts, but it's very, very difficult for texts that are complicated to read, for instance, or a long, long text. That's where you want to dive into and see what you can do there. For this, you can use Refinery perfectly.

9:00

Alexey

**I'm wondering why… This problem is not new to me personally, and I think for many data scientists. Ever since I started doing data science, I think even before my first real full-time job, it was already a problem. I remember solving this problem with this iPython Notebook widget – back then it was called iPython Notebooks – this Jupyter Notebook widget .There are these little tools, one of which is called Pigeon. It allows you to quickly create a simple annotation tool with these iPy widgets. But it never occurred to me to actually make a proper tool from this.**

**I would use these widgets, label data quickly and then I forget about this. So how did it happen for you that you realized “Okay, there is some potential. There aren’t really any good tools for that. Let’s make a tool and then people in the community will benefit from this.” How did it happen to you?**

10:02

Johannes

Yes. One of the things is that it's mainly designed for natural language. For natural language, it was a bit more difficult than what we see… [cross-talk]

# The difficulties of working with natural language processing (NLP)

10:14

Alexey

**So these widgets didn't really work for NLP, right? It's much harder.**

10:17

Johannes

Yeah, it's much, much more designed, in many ways, for structured data. When you, for instance, already have something like an Excel spreadsheet. But if you have a .txt file, which only contains textual data, something that we've seen is that, oftentimes people are missing those valuable metadata insights such as the language and so on, so forth. We have been coming from a research perspective, mainly. We were studying and looking into what kind of new interesting technologies are coming out.

I think one of the big things that also made this product possible is the general pace at which natural language processing is currently developing. There are so many new techniques that you can use that make this tool not just nice, but really strong. Maybe a decade ago that would not have been possible. So I think there are two factors. Basically, the general interest in natural language processing is rising so much, especially in the last months, and also what's possible in the end – what special pain points that language processing has.

11:25

Alexey

**Yes, especially these tools like ChatGPT. I remember having a lot of fun. I would ask it to summarize an article and then I would ask it, “Hey, can you please write it, as if Donald Trump wrote this article?” [Johannes laughs] It's so much more entertaining. Unfortunately, right now, they changed something. When I ask ChatGPT to rewrite an article as Donald Trump, it would say, “Sorry, I'm not going to do this.” That's very unfortunate. But I guess indeed, NLP is quite a nice place to be right now, with all these tools.**

12:02

Johannes

It's super, super interesting. We've been in this field – we've been focusing on natural language processing purely for a bit more than three years. Pre-November 2022, people were always asking us “Why the hell are you focusing on natural language processing? This is too complicated.” Of course, we didn't know that ChatGPT would be there, but we already saw, because of the being so close to research, that things like Hugging Face were happening. Transformers were becoming more and more strong. Ever since then, ever since ChatGPT, it feels like the whole perspective shifted by 180 degrees. Now they are saying “Okay, of course you're focusing on natural language processing.” So that's interesting to see. [chuckles]

12:54

Alexey

**For you, you started doing it before it was popular, right?**

12:59

Johannes

Yeah. [chuckles] I was really hoping for something like this to happen. But of course, I didn't know that it would happen. I mean, and natural language processing already was becoming more and more popular, even before ChatGPT, but I think mainly for the developer base. Developers were seeing that something like this was happening, but not for the broader masses, I would say.

# Incorporating ChatGPT into a process as a heuristic

13:22

Alexey

**GPT-3 was around for quite some time already when ChatGPT was released. [Johannes agrees] So your research was about NLP right? Probably for this, you needed to label data. Right? And that's how Refinery appeared?**

13:37

Johannes

Yeah, as we were working in the consultancy, we basically saw this quite often. We were trying to figure out with people in workshops, what kind of use cases would be helpful and if you could ever have this completely data-centric perspective, in the sense of, “Okay, what kind of data do we already have and what kind of use case can we build on that?” Or you could take the other approach thinking, “Okay, what kind of use cases would be valuable? But do we have the data for that?” We typically went for the second approach of, “Okay, what would be helpful?” And then try to figure out how we can build the data for that.

At the same time we did this, we were in a research project at the university, where we also already saw, “Okay some things are happening.” This weak supervision approach is not absolutely magical. I think it was hyped a bit more than what it actually offers, but it's a really strong framework to bring together any kind of heuristic that you can imagine. For instance, ChatGPT is also an amazing heuristic that you can use, right? We saw it back then and realized that very, very quickly, it could not only bring interesting research insights that would actually be helpful in business applications. So that's where we came from.

14:57

Alexey

**I recently came across a paper that says… What I think the researchers did was compare ChatGPT with Mechanical Turk (a crowdsourcing platform). Did you see this paper? [Johannes agrees] ChatGPT was basically better at labeling the data than crowdsourcing. I'm wondering, for you, does this possess any risks that people will not use Refinery and they will just go directly to ChatGPT to extract sentiment from the data and use it for labels?**

15:35

Johannes

Honestly, I love it. I love it because I rather see it as one kind of heuristic. I know from business that there are still many applications where it's not that easy to just plug in ChatGPT and have it work.

15:51

Alexey

**By heuristic do you mean that you can use it to create an initial set of labels and then refine them with Refinery?**

15:58

Johannes

Yeah. For instance, very simply, let's say we want to build an intent classifier. We have an email inbox, emails are coming in, and we want to understand, “Are they about cancellations? Are they about feedback?” Whatever. We want to understand the intent. One really good approach if you want to build a classifier now would be to say, “I want to use the power of ChatGPT. The first 10,000 examples I have – I would just ask ChatGPT to label them for me and then I will train a model on top of that label data.” That is one very solid approach. With that, I think you can already get something like (this is a random number, of course) but something like an 80% precision, let's say.

If you now say, “Well, I have ChatGPT as one heuristic, and I also use an active learner, like a Hugging Face model, for instance, that I train on manually, and maybe I also use a crowd labeler. Then I would achieve a 90% precision. So I think it's like one ingredient in a complete dish and I think it's an incredibly cool ingredient. In many cases, it will already be super helpful to just use ChatGPT. But I'm thinking that it becomes even stronger when you combine it with other techniques.

17:26

Alexey

**Okay, so it's not a competitor, it's one of the things you can take advantage of and use it together with your tools.**

17:34

Johannes

Yeah, I think so. As I mentioned initially, I think we finally would not exist if, for instance, Hugging Face would not exist, because that's something that we already see as one of the key foundations where we finally said, “We can build really strong embeddings and the embeddings have a data management part.” Of course, you could already also say, “I have the Hugging Face models. I want to fine-tune them.” And that's good. That's a super valid approach.

I think with Refinery, it just makes it a bit easier to get there faster, and especially for business users to get there much faster. So I think that GPT is something that we actively make use of honestly. It's already a brick. [chuckles] We use GPT as a couple of bricks. It just helps so much in prototyping and everything. So that's amazing.

# What is Bricks?

18:33

Alexey

**And by “bricks,” you mean this other project that you mentioned in the beginning and the other project we also demoed at Open Source Spotlight, which is this collection of different recipes that people can just grab and use?**

18:43

Johannes

Exactly, exactly. For instance, just to give an example, we have a Brick for sentiment analytics. One implementation could be, for instance, using Textblob very simply, as a module that you can just pip install. Another implementation could be a prompt that we have prepared for GPT, which basically says, “The following examples are positive/negative. What is this example?” That's kind of, again, the idea.

If you want to implement a sentiment analysis for your own project, one idea could be to say, “Okay, let's look into what kind of Brick implementations I have. Let's just take all of them. Let's just see how well they perform. Because if each of them perform at an 80% precision, together they will perform at a 90% precision.” It's a very simple idea, but it works. [chuckles]

19:38

Alexey

**That's the idea behind Ensemble methods in machine learning, right? You have a bunch of quick classifiers and when you put them together, they perform a lot stronger.**

19:48

Johannes

Exactly. And the thing with weak supervision, mainly, is that – of course, weak supervision has different kinds of implementation, so you can have different kinds of strategies of how you want to make use of those single workers. But the idea roughly is as if you have a random forest, but instead of decision trees, you have concrete implementations – concrete heuristics. One decision tree can be GPT, one can be Textblob, one can be Vader, and so on and so forth. And one could even be crowd labeling, right? So that's kind of the big idea behind weak supervision.

# The process of starting a startup – Kern

20:22

Alexey

**Okay, you have this cool project – multiple cool projects – and at some point, you decided that you should start a company to actually focus on this project. How did this happen? Can you walk us through the process?**

20:37

Johannes

Yeah, exactly. First, actually, it was the other way around. We already knew what we were building. We started with the consultancy, and we started the consultancy already knowing that at some point, we want to build products. But we didn't know what yet. We didn't know whether Henrik and I (we are the co-founders) would get along well or not. [chuckles] And we also needed some initial capital. Those were the three reasons why we started the consultancy. It quickly turned out that Henrik and I worked along really, really well together, which already was a cool thing.

We initially started building a software that completely failed. The first product that we built was really bad. [chuckles] It was a complete, no-code, machine learning builder, basically. The idea was, we were working with business units quite a lot and we wanted to help them basically say, “Just label a bit of data and then you have the model and you can use it.” That didn't work for two reasons, basically. One, of course, the training data that the business has to put into the auto-AI was really bad. [chuckles]

And the second thing was – and I don't know if this is now shifting or not, but it was the reason back then – they had a big fear of building AI themselves because they were not the experts. Or they didn't want to have the responsibility for building something like an AI. This was not a technical problem, but one rather from a mindset or perspective. It’s just as I want to read a legal document, but I don't want to write a legal document myself. It's kind of how I like to put it. So what we wound up doing back then – we still had a consultancy, so everything was going well.

We realized that we had this research project and we just said, “Okay, let's shift the user from the non-technical user to the technical user and let's take care of this specific problem that we realized is causing bad models, which is training data.” From there, basically, we started building Refinery, the very first version. We had specifically designed everything for engineers. We already knew there are already many labeling tools that are already really good, but what we were missing was something that gives engineers more control. That’s also the idea of Refinery.

23:10

Alexey

**Why did you specifically focus on engineers?**

23:14

Johannes

Because that's what we saw – they were the ones that were confident in building AI models.

23:25

Alexey

**Your first product was focused on business people, and they weren't confident in that.**

23:32

Johannes

Exactly. They wanted to use the model in the process and they wanted to say “I have the following problem. I have incoming emails and I want to classify them. I want to have them draft the response for me.” But they don't want to build AI. [chuckles]

23:49

Alexey

**So then they actually ask engineers to do this. Then the engineers would discover Refinery, for example, and just go ahead and use it in building their solution. Right?**

24:00

Johannes

Exactly. That's kind of the shift that we had. One of the insights that we had with the first bad product was that we didn't yet understand… We understood what our users wanted, but not that they didn't want to build it themselves. So we took that insight and tried to shift the user perspective to different people. We said, “Okay, can we help you build those models in a better way?” And it was interesting, because we then realized that engineers had problems with training data. The first team that we went to, who we are still actively working with, basically had a process in which they were supposed to build this very AI. It was a key process – something that the company has been working on for more than a decade. The training data has been an Excel spreadsheet with 10,000 rows and two columns. That was the foundation for their work. [chuckles]

So we asked them, “Is this for real? This is actually your training database?” And it was. That was basically how we realized, “Okay, there are teams that have almost no control over the training data that they have, so they have no insights. And there are also teams that quickly want to prototype things so that they can discuss with the business unit if it makes sense to follow through and if they want to look deeper into it.” This is also why Refinery is now the product that it is, because we tried helping those teams that quickly want to set something up as a prototype. But it also is specifically designed to help engineers who already have a labeled dataset and now want to dive deeper into what is going on there.

This is also where the heuristics are super helpful, because you can do things data management like looking for those records with two heuristics that your domain experts gave you, where they collide – where they say different things. That's basically that's how we got to the product. We've been going to developer conferences and developer conferences and developer conferences – many of them – and just try to be super close to the users of it. At some point, we realized, “Okay, this product that we've made, we have to open source it. That's the best way for us to get more and more feedback and to distribute it faster.”

# Making the decision to go with open source

26:22

Alexey

**I was just going to ask you how exactly you made this decision. So you attended conferences. Was it your realization that you need to open source so that more people try it? Or was this something you heard from other people at the conferences?**

26:38

Johannes

It was a combination, honestly. Basically, we already had the feeling that open source could be an interesting approach. But I also have to say, I was a bit afraid of open sourcing at first, because I thought we now have these really cool products. It has been coming from research, so we did a lot of cool things, and if we now open source it, we're basically showing exactly how it works.

It's something that was my initial mindset. I was realizing it more and more as I spoke with more users. I also had kind of “help” in the sense that I had a discussion with a befriended co-founder, who was also going through that phase. He was also thinking “Oh, should I open source?” We just had a beer together and talked a lot about it. At the end of the discussion, we both said, “We have to open source what we’re building.”

27:36

Alexey

**Who is this other founder? What’s the product they have?**

27:40

Johannes

It's actually a product specifically designed for other developer toolings, which is called Crowd. It's basically a tool that allows source projects for developers to measure, “Where's my community? How big is my community? Where are the people that are most engaged with my product?” He's basically the one who convinced me to take the insights that we had from users and say, “Okay, let's open source it.” [chuckles]

# Pros and cons of launching as open source

28:11

Alexey

**So you had some fears of opening all the code. Did any of these fears, actually, realize? Did any of the things you were afraid of actually happen? Now everyone could just go and open the code and get the insight of your years of research.**

28:32

Johannes

It comes to both advantages and disadvantages, honestly. I think open source itself is not a business model, but open source changes your business model. As I already mentioned, we were coming from first the company, then the open source perspective. We already knew that if we want to survive as a startup, we have to make money at some point or find a way. What we realized is that you “lose” some customers, in a sense, and you “win” some customers. You have to get an understanding of what's bigger is – the losing customers or is the gaining customers bigger.

29:25

Alexey

**How exactly do you lose customers by opening the code?**

29:31

Johannes

Yes, precisely. That’s a good question. We know that if you're working with really small teams, like other startups, in many cases, the open source version is everything that they need.

29:51

Alexey

**Oh, you mean they remain customers, but they don't pay you. Right?**

29:53

Johannes

Exactly. Right.

29:56

Alexey

**They basically use the “free version”. The open source one.**

29:59

Johannes

Exactly. The open source version, I really must say, is a really good product. It's basically already offering so many things, because we really tried to make it a big open source. We didn't want to make this as a “marketing gimmick,” so to say. We really wanted to commit to this open source idea. So, yes, we already have lost revenue in that segment, because people who would otherwise have paid for the software just took it and use it. And that's perfectly fine. That's cool, because they use it and we get much more feedback and everything.

I think what we see more is that we are rather making money from the business unit side. When we are working, for instance, with enterprises – we try to discuss, “Okay, what kind of use cases are you working on? What kind of use cases do you want to implement?” Then we talk about very specific details, for instance, “What kind of use cases are there in the insurance industry?” It's much more that we are now selling to the business units, but we get like the biggest support we can imagine from the developers inside those companies, because they say, “Okay, I downloaded the open source version. I played around with it. It's much better than what we had before.” We, as developers, say, “Yes, that's good.”

The business typically pays for something different. They pay for some things like support –the knowledge that you have from certain use cases. That's something that we see. So we still have small companies also paying for the product, but I would say the distribution shifted for sure. So we lost some and we gained some.

# Kern’s business model

31:47

Alexey

**So what's your business model? From what I understood regarding what you just said, you have the open source library and then there are some enterprises that use this library – you support them and share the knowledge you already have. I guess that’s one of the income sources, right? You consult on how to use your product. Is this correct?**

32:10

Johannes

Yes. Mainly, at the moment, we have the Refinery open source version as a single user version. That's at the moment of the case. What that basically means is, if you want to implement a project, you can just download it and use every feature but you can just work in a single user setting. For instance, if you want to invite me so that I can label your data, that can’t happen in the open source version at the moment. The commercial one basically works like this – the engineers don't want to label themselves, but rather they want to say, “I want to have control over my project. So what I'm rather doing is setting those things up and then asking my colleagues to label it in the subset where I really have headaches.”

That's what I meant so far, for instance for a pre-seed data product that started, where they have one or two engineers at a time – that's okay, they don't need that. They sometimes find workarounds that we didn't know existed to work with the product. On the other side, you have the team that tries to think of the opportunity costs. They say, “Okay, if I pay 150 euros per month right now for this set up infrastructure for the multi-user setting and everything, for the knowledge, we're basically saving a lot of money.” Then they would rather pick that option.

33:53

Alexey

**So you do not do consulting? The income you have in your business model is this extra thing, which is collaboration?**

34:03

Johannes

It's both. I would say for the developers, it's mostly that they contact us and they say “I want to build the following,” then it's mostly just giving them the software and supporting them. Then they reach out to us with use cases that we've never heard of, or they can give “process knowledge,” so to say. But when it's more on the enterprise segment, and we reach out to clients ourselves, then we reach out to them with very specific pain points and very specific industry use cases. For example, how you can automate certain high value underwriting and insurance. Then we basically say, “Okay, we're going to work very closely with your development teams.

We're going to give you the software – basically the whole platform – but we're also going to do things like workshops and the like, for you to dive deep into it and make sure that everything is exactly as you want.” What I mentioned, initially, the business units want solutions. In the end, they wouldn’t care if it's… maybe they care a little bit, but they don't worry too much if it's AI or not. They want a solution for their use case, so you really have to make sure that that works.

As for the developers, they get super creative. They play with a lot of things and they want to implement their own things. It's a bit different around the segments. That's what I'm trying to say. I think open source pushes you more, more, and more into understanding what kind of segments there are. So that's something I definitely see impacted us.

35:44

Alexey

**The workarounds you mentioned that the engineers find – these are workarounds to not pay you, right? [Johannes agrees and laughs] What do you do with these workarounds when the developers share them with you? [cross-talk]**

36:00

Johannes

Honestly, when they share it with us on Discord – and I'm not kidding, you can look into our Discord – we help them with that workaround. Basically, we had one that I'm going to share because you can look into our Discord and see it anyway. One way, basically, to work around it is that you can create a “smooth user” setup. If you have multiple users, every one of them downloads the open source version of Refinery and if user A starts labeling, what they can do is create a project back up, which is like a snapshot of the whole project, and export that. It’s not a regular export – a regular export looks a bit different. But you can create something like a backup. If you import that backup into the other machine of the other user, they can continue labeling. That's basically one workaround that you can use.

36:56

Alexey

**But not at the same time, right? They cannot label at the same time. [Johannes agrees] Basically, this is not convenient. Right?**

37:02

Johannes

Exactly. It's not convenient. For very early startups, it's rather that you can say, “Okay, I will work two more hours today and we'll save 100 euros. That sounds better so I'm going to do it even if it's not convenient.” That's something different from corporate saying “We have way too little resources, but we have money.” So it's two different pain points and two different perspectives. The startup has a lack of money. Corporate has a lack of time. So it makes perfect sense. We want to help people using our software. In the end, even the people that find workarounds, they give us some super interesting insights. We want them to use our software. We want them to find workarounds, and we will help them.

37:55

Alexey

**You mentioned the Discord channel. Do you have the link? This Discord is in your GitHub, right?**

38:04

Johannes

Yes. And the description is basically about our software. But we also like to share things generally around data-centric AI, so for people that are also just getting started, don't feel shy. There’s a super, super happy team and we’re happy to have you.

# Working with enterprises

38:23

Alexey

**You mentioned that for this second income source (consulting), you find enterprises, for example insurance companies, and then you come to them with a use case already. You say, “Okay, we think that a company like yours will benefit from this use case because we saw that other companies already use this and they get a lot of money or save a lot of time.” Right? This is how you would do this. [Johannes agrees] And how do you find these clients? Or you just take a segment – let's say insurance in Germany – and then you just start calling them?**

39:00

Johannes

I think that's the difference between commercial software and purely open source projects. You really have to try to understand how you can get in contact with them, because it's not as simple as building a really great product and then just calling and hoping everyone will say yes. [chuckles] That's not the case, sadly. Honestly, it's a lot of trial and error. We see that different approaches work for different segments.

One example, for insurance, especially in Germany, it's really a lot about who you already know, what kind of networks they are. We are located in Bonn, and many, many insurers in Germany are located in Cologne, which is very close to Bonn. So what we basically do there is try to go to meetups or get in contact with people that have a good network and can help us. That's one approach.

Other approaches, in other segments, completely different things work. It feels a little bit like not only building a product, but you also have to build some kind of machinery to get in contact with people that could find this interesting. There are many approaches that you just have to try out, I guess.

# Johannes as a salesperson

40:21

Alexey

**Do you do this sales part yourself, or are there people in your team who take care of that?**

40:29

Johannes

It’s both. Basically, up until March, it was completely on my own. We now expanded the team a little bit. We didn't grow like crazy, but we hired a few more people and added more people into this, but it's still 100% a very big part of my job to do that.

40:51

Alexey

**As an engineer, how did you learn sales? For me, it's so unnatural – it's so different. It's a totally different world.**

41:02

Johannes

I think this sales thing has a bad connotation for engineers and I understand it. But I think it's not true, honestly. Let me give you an analogy where I currently see something super interesting happening. We have in our team, a so-called Developer Relations Team. These are people that are very much focused on creating really good content on YouTube, that also have people on Discord, go to meetups and everything. Those are developers who do marketing. There’s this joke for developer advocates, that you should never tell them that they are doing marketing, but they are doing marketing. Because that's some approach that helps developers a lot.

Developers hate being sold something, but they love learning new things. So what you try to do is go to conferences, talk with them, try to educate them about new things and talk about your really cool open source project that they should check out, because they will learn something from it. And that makes sense. I like to think of sales very similarly. If you have a really interesting product, and you see that it's helping people, then you can feel confident about it. Then you don't feel shame when trying to sell something that they don't want, but instead you actually want to try to help. You get in contact with them with a “big chest,” a happy face and try to help them and try to understand what their problems are and how you can help them. And then it's practice.

I'm also not perfect yet in that field. I used to make many excuses and will continue to do so for the whole of my career. But I think it's the moment you try to shift the perspective of trying to be someone who says something that people don't need – to something like, “Okay, I'm trying to first understand what the problems are and how I could potentially help you.” Then I think it's a whole different topic.

# The team at Kern

43:12

Alexey

**Okay, that's interesting. But right now you have people who help you with that. Who do you actually have on the team? What kind of roles do you already have?**

43:22

Johannes

Yes. I think at this moment, we have 11 full-time people and 2 working students. It's a small team. You can fit everyone at one table. [chuckles] We kind of have three… this term sounds way too big for the team as it is, but we have but we have three “departments” so to say. One being the development/product team – these people work with Refinery, mainly, Bricks and the other products that we have. Those are very engineering people. They can build stuff. They are very, very close to the developers. So that's one department.

The other department is the developer application department, which is basically just trying to focus on “Okay, how can we spread the word, how can we engage with people, and how can we be helpful to them?” And “How is what we offer bigger than our product?” For instance, they focus more on the topic of data-centric NLP instead of just Refinery. Then we have our “go-to-market” team, which is basically a team that is fairly close to sales. But instead of just doing sales in the sense of calling people and selling the product, this team rather tries to understand, “What kind of segments do we have? What kind of use cases are there? What kind of things are people currently seeing? What kind of trends do we have?”

To give you an idea, something that we are seeing since November, there are strategy and innovation departments in companies – they all have the same task, which is basically to understand what GPT means for their business. It's people that previously haven't worked with NLP. What the go-to-market team does is try to understand “How can we help them? What kind of knowledge do they need to have? And how can we help them?” Another part of the go-to-market team is setting up campaigns, trying to reach out to them, having conversations with them, and so forth.

That's kind of how it currently looks like. Something that I think we are really good at is having the teams work together really, really well and I think that's super critical for a startup. For instance, if the go-to-market team identifies that logistics companies are struggling with email communication because vendors are telling them what kind of shipments they need to send to Sweden the next day, and they need to parse email data, then the dev team will have to set up some Brick the next day. Then the dev team will make sure that this use case that the go-to-market team wants to showcase is part of the playground. So that's how they work together really well at the moment.

46:23

Alexey

**So then for the next logistics company, you can already showcase it and say “Okay, this is what we do with our logistics companies. Do you want to have this too?” and then you cover them.**

46:34

Johannes

Yeah, basically. Kind of that way. Very often, depending on the enterprise size, if it's a smaller enterprise, you can have recurring exact templates, so to say, with pre-build AI models, because most logistics companies are struggling with identifying unit dimensions in the emails. But if it's, for instance, going into a bigger enterprise, they typically have very niche problems. Then you need to take an example reference in this case, and try to narrow it down to what exactly they are facing. I would say the bigger the client, the more you need to really… I mean you should always understand what your client is facing – but the bigger the clients, the more niche the problems become. That’s something that we’re seeing.

# Johannes’s role at Kern

47:20

Alexey

**Interesting. And what exactly do you do? All three and more?**

47:26

Johannes

Me personally? [Alexey confirms] I'm jumping in between. Now I think my main task is mainly in the go-to-market side. But I'm still programming. I do my side of the programming as well, but the prototyping part. For instance, if I'm in the go-to-market area, I'm trying to talk with potential clients to see what kind of problems they are facing, and I'm hearing more and more certain things, then I'm trying to very quickly set up some kind of prototype that I can demo to our development team. I then try to understand and discuss with them what kind of problems the potential clients are facing and how we could build something for that or how we can nudge features. But I rarely do any fixing stuff or building PRs for Refinery. Refinery is something that sadly, I don't touch at all anymore. [chuckles]

48:26

Alexey

**But it wasn't like that all the time, right? I guess at the beginning, maybe you focused more on the engineering side of things, then maybe developer advocacy and now the go-to-market part.**

48:39

Johannes

The coding part is becoming less and less, but it's still there. But I'm also perfectly fine with that because I see that the overall speed in which we are developing is just growing a lot and we just see that more and more cool things are happening. This happiness of a developer seeing stuff becoming reality, that's completely true. And it's becoming more and more true, even though I'm coding less and less.

49:09

Alexey

**What's your actual title? I don't think I asked you. Are you the CEO, CTO, or?**

49:15

Johannes

Yeah, but I'm also the person that's cleaning the office. [chuckles] Many roles. No, I'm just one of the co-founders. Just for the official title, I'm the CEO and my other co-founder is the COO, because we kind of have these roles where he's working more on the stuff that's happening internally, like processes that we need to set up and security stuff, whereas I'm rather the outside position, like talking with clients. But yeah, that's just about it.

# How Johannes and Henrik separate responsibilities at Kern

49:51

Alexey

**How did you come up with this separation of responsibilities? Was it something natural or you just decided to sit down and discuss it?**

50:00

Johannes

No, we didn't know that. When we started the consultancy, we didn't know what our strengths and weaknesses would be because we didn't know too much about each other yet. The setting basically was –I was looking for a co-founder for many months for the consultancy and found no one. So I've started the consultancy myself at first. I met Henrik during university and I was talking about a project that I just landed that I realized I couldn't do myself. [chuckles] But I already signed the contract, so I was kind of in a difficult situation.

We were talking a bit and I just asked, “Hey, do you want to join me for this project and see if that works. Then we can potentially work together.” And he was super enthusiastic and I'm so, so happy that he worked on a project with me. But we didn't know our strengths and weaknesses. We figured that out, I think over more than half a year later. That took a lot of time. At some point, we realize that I'm a messy person. [chuckles] I jump around between different things and Henrik is super calm. He's super structured and he is more analytical than I am.

I'm also an introvert but I'm still trying to talk with people to validate ideas a bit quicker. That's something that we at some point realized that that's kind of good that we are different in that sense. So Henrik is focusing more on the internal stuff and I’m focusing more on the external stuff. But it took us a lot of time to realize that.

51:44

Alexey

**Both of you come from a technical background. [Johannes agrees]. And neither of you do any coding right now. Correct?**

51:57

Johannes

Henrik doesn't code since the beginning anymore – basically, since the beginning of our software consultancies he coded. And I code, essentially, if I want to prototype something to showcase to the development team. At the moment, I'm working on a prototype for native PDF labeling – basically extending Refinery, which is currently for natural language processing. I'm currently prototyping. I don't know whether we're going to continue on it, but it's just something that I want to discuss with the dev team. I'm working a bit on that prototype, but it's definitely not my main focus. That's kind of what I'm doing.

# Working with very niche use cases

52:40

Alexey

**That sounds like a cool project. I guess one of your clients has this use case, right? Or maybe multiple clients – they have a bunch of PDFs and they need to do some NLP on these PDFs, right?**

52:54

Johannes

Yeah, it's many, many, many, many clients that have these PDF problems, but it's just also so difficult. [chuckles] But we are already seeing that we made incredibly good progress. It's not the first time that I'm touching this prototype. I've touched this prototype, I think, already five times. This time, (I hope, let's see) I cracked the code on how we can use our weak supervision approaches. In the past, I already also looked into it and couldn't figure out any good way. [chuckles] But basically, there's something manageable.

53:28

Alexey

**There are companies who specialize on just that – and not just on PDFs, but on PDFs in a specific niche. For example, I remember five years ago I spoke with a company in Berlin that was processing PDFs for real estate in Germany. It's a very specific niche, real estate developers. They have specific processes, specific documents. And it was enough just to start a company on this. I guess it’s not a simple problem, right?**

54:01

Johannes

Yeah. That's also something that I find extremely interesting, because it's something that we are seeing when we're working with clients. You're working with the development teams and they typically try to think about what helps the company a lot and you see that there are different sets of use cases in a company. I like to think of the long tail, so to say. There are very repeating use cases.

Every company processes invoices – every company. Of course, if you look at it from a company perspective, the total market for generating invoices is huge. And so, of course, there are like many vertical solutions for general invoices. But there are also dozens of documents, as you already mentioned, that have different categories. And from a total market perspective, they are tiny – but for the very specific company, they are huge.

For instance, just to give one example, it could be that a certain type of invoices like ones made by doctors or clerks or whatever – they may be a huge volume of the general invoices that a company processes. So what they're doing is, the data science teams are working on creating internal solutions for those niche-y kinds of documents. There's no vertical solution that is out-of-the-box, but we can have those data science teams come up with a niche solution much quicker. That's also what we at Kern see often.

55:39

Alexey

**If I understood you correctly, there is a company (an enterprise) that already has a data team, and they have this problem with processing invoices – they have a lot of PDFs. What you do is you help them set up or improve the processes by using Refinery to label data, so then they can train models and optimize these processes.**

56:03

Johannes

Exactly. That's not just for documents. But for documents that's what we're currently trying to figure out a bit. But it's also something for natural language processing. Very often, you go into the companies, and you see that there are already data science teams working on a long list of use cases. It's not just one use case. Basically, it’s internal processes that are high in value. So if you optimize them, it's beneficial for the company.

What we typically try to do is just work together very closely with the developer team to figure out ways to improve that. And the reason that having an open source project is helping us so much to start at a good baseline, basically, on a human to human level with the dev team, is because they already know that we are contributing to the open source community. Every developer likes that. That's one of the big advantages of our open source launch.

# The short story of how Kern got its funding

57:02

Alexey

**I just realized that I did not ask you about that. First of all, congratulations on getting funding of 2.7 million – very nice. I wanted to ask you, how did you actually get it and what the process was. I don't think we have a lot of time, but maybe you can quickly tell us. Was it difficult, or because you already had a good product, it was only natural that investors just wanted to give you money?**

57:32

Johannes

I’ll try to give the short answer even though it’s possible there would be a long answer as well. [chuckles] So in short, I think we had a relatively good standing because we were interesting for a very niche kind of investor. If you think of 100 investors, (it's also just a random number, but just to tell the story) 10 of them are specifically interested in open source for machine learning and maybe five of them are also specifically interested in developer tooling. The number of investors we talked to was relatively small, but those that we talked with, we had the benefit that they were very interested. A big part of this was also that not all of them, but many of those investors basically contacted us, because of the open source launch. In a very short way, that’s how the funding happened.

58:35

Alexey

**I just realized that I actually completely forgot about that. So after we did this Open Source Spotlight, I got a message in LinkedIn from one of the investor families – there are actually families that invest in, surprisingly, in open source ML tools – and they told me, “Look, we are big fans of your open source spotlight and we want to invest into this company called Kern. What do you know about them?”**

59:00

Johannes

[laughs] Really?

59:02

Alexey

**Yeah. I hope I actually went through. I don't know what happened after that. But they were also asking about other startups. They said they are big fans of this Open Source Spotlight, which I never realized. It was never the intention to do these things for investors. It was just very surprising. Maybe we need to get one of these investors on the podcast. It would be quite interesting to see how they make decisions.**

59:29

Johannes

That would be super interesting, I guess, because it's something that you see so much in the last five years or so – funds (not all of them, but many of them) are building their investment thesis specifically for open source. I think that the time of “commercial open source” companies is basically just going to go bigger and bigger and bigger. So yes, that's a good idea. [chuckles]

# Johannes’s resource recommendation

59:58

Alexey

**Maybe the last question before we wrap up. So you know any resources or books or anything that you want to recommend to our listeners about the topic of starting an open source startup?**

60:10

Johannes

Not on that precisely, but one book that I would recommend for people to read, when talking about machine learning and natural language processing more from the economic perspective, is a book that's called Prediction Machines. This is basically a book I think it's like five years old or something, but it's looking at machine learning from a very dry economic perspective. It's still fun to read. But it just gives you so much of an understanding on what kind of business applications machine learning has. It tries to view this topic from not just the technical perspective, but also what automating one part of a task means for the other tasks in the process. It just has so much to get a better overview of applied AI.

61:04

Alexey

**I am checking it on Amazon. It's called Prediction Machines, Updated and Expanded: The Simple Economics of Artificial Intelligence.**

61:12

Johannes

Yes, exactly. That one. That's good.

61:15

Alexey

**Okay. Thanks, Johannes, for joining us today, for sharing your experience with us. It was really nice talking to you. And thanks, everyone, also for joining us today and listening in.**

61:28

Johannes

Thanks for the invitation.