



EXPERT INTERVIEW

UPDATED  
01/18/2023

# Oscar Beijbom, co-founder and CTO of Nyckel, on the opportunities in the AI/ML tooling market

## TEAM

Jan-Erik Asplund  
Co-Founder  
[jan@sacra.com](mailto:jan@sacra.com)

## DISCLAIMERS

This report is for information purposes only and is not to be used or considered as an offer or the solicitation of an offer to sell or to buy or subscribe for securities or other financial instruments. Nothing in this report constitutes investment, legal, accounting or tax advice or a representation that any investment or strategy is suitable or appropriate to your individual circumstances or otherwise constitutes a personal trade recommendation to you.

This research report has been prepared solely by Sacra and should not be considered a product of any person or entity that makes such report available, if any.

Information and opinions presented in the sections of the report were obtained or derived from sources Sacra believes are reliable, but Sacra makes no representation as to their accuracy or completeness. Past performance should not be taken as an indication or guarantee of future performance, and no representation or warranty, express or implied, is made regarding future performance. Information, opinions and estimates contained in this report reflect a determination at its original date of publication by Sacra and are subject to change without notice.

Sacra accepts no liability for loss arising from the use of the material presented in this report, except that this exclusion of liability does not apply to the extent that liability arises under specific statutes or regulations applicable to Sacra. Sacra may have issued, and may in the future issue, other reports that are inconsistent with, and reach different conclusions from, the information presented in this report. Those reports reflect different assumptions, views and analytical methods of the analysts who prepared them and Sacra is under no obligation to ensure that such other reports are brought to the attention of any recipient of this report.

All rights reserved. All material presented in this report, unless specifically indicated otherwise is under copyright to Sacra. Sacra reserves any and all intellectual property rights in the report. All trademarks, service marks and logos used in this report are trademarks or service marks or registered trademarks or service marks of Sacra. Any modification, copying, displaying, distributing, transmitting, publishing, licensing, creating derivative works from, or selling any report is strictly prohibited. None of the material, nor its content, nor any copy of it, may be altered in any way, transmitted to, copied or distributed to any other party, without the prior express written permission of Sacra. Any unauthorized duplication, redistribution or disclosure of this report will result in prosecution.



Published on Jan 18th, 2023

# Oscar Beijbom, co-founder and CTO of Nyckel, on the opportunities in the AI/ML tooling market

By Rohit Kaul



EXPERT INTERVIEW

**Oscar  
Beijbom**

Co-Founder & CTO  
Nyckel



## Background

Oscar Beijbom is the co-founder and CTO of Nyckel. We talked to Oscar about the opportunities in the AI/ML tooling market, challenges for data labeling companies, and the cambrian explosion in the ML non-expert market.

## Interview

**Can you talk to us about the state of AI in 2023? What are the different key segments in the AI market?**

It's useful to look at it from the perspective of who you're selling the AI product to.

There are two main personas you could be selling to—One, an in-house machine learning team where you assume that the users involved are machine learning experts who control the ML process and want full freedom to experiment. This could be



a senior director, a senior innovator, or a contributor that actually makes a purchasing decision.

The other persona is a non-expert, like a product owner who just wants to solve a particular problem and doesn't care about the inner workings of things.

In the first trend, you have the traditional machine learning ops (MLOps) ecosystem. You need someone like Scale, to help with data labeling. You also need a company like Aquarium to help you with the data engine visualizing issues with your model and identify what data to send for labeling. You may or may not want to employ something like an AutoML engine or even network architecture search like NAS but probably want something like Weights and Biases for monitoring your experiments. You also probably want some sort of repository for your trained models. Ideally, that's tied to the data sets they were trained on. So, you have the lineage.

Then, you need some infrastructure for deploying your models and having that be elastic. You also need monitoring infrastructure on top of that to see how these models are improving over time or if they are degrading. There's a really big, messy ecosystem out there in AI right now, and it's a challenge to figure out what to buy and what to build.

The other trend comprises the non-experts, who don't care about any of that. They have problems and they want to solve them. For them, the abstraction layer is the data itself. If I give you this input, I want this output. That's the abstraction. Then, you do everything—train, deploy, and monitor. You even tell me which labels I should be reviewing next and so on.

This can be done by a company like Nyckel with APIs or with an in-house ML team. It is similar to GPT-3 and the OpenAI interface, where arguably, you're not even doing any training. There is some prompt engineering that can be likened with training or fine-tuning, but the interface itself is built for non-experts.

As the deep nets become more sophisticated, not only in terms of their architecture and their size but also, in terms of how they are pre-trained, they become better and better, and an interesting trend begins to emerge. It used to be that you needed 10,000 label data points to even get something



interesting. That's why companies like Scale existed and became big. But I don't think that's going to be the case anymore, at least for the majority of these cases.

You're going to have these foundational or pre-trained models like GPT-3 and need much less data. That changes everything because then, you don't need all this machinery around it. You just need to do a minimal amount of fine-tuning and train it very quickly with just a handful of examples.

With regard to the MLOps space that caters to experts, my guess is that it's going to shrink or at least not grow quite as fast as the space that it caters to—the end customer, the non-experts, and the product owner.

**Data labeling companies have a usage-based pricing model. Historically, creating data sets to train AI has been a bottleneck to building it. Is that still true? Can you talk about how few-shot learning could impact data labeling business models? How has the recession impacted usage? Are there any 80/20 opportunities to do more with smaller data sets?**

The data annotation space isn't going to keep growing like it did earlier because pre-trained models are going to get better and better. The rise of those companies, Scale, in particular, was tied to the autonomous driving industry that was extremely data-hungry.

There are two trends pulling in different directions. There is the fact that if you think about a self-driving car that operates in an open world, there is a combinatorial number of edge cases. What I mean by that specifically is you imagine, you might struggle with people that are dressed in some sort of glossy rain gear. That might be a problem if your AI is a computer vision system. You might also struggle if it's raining. You might also struggle if it's kind of dusk or if the sun is shining right into the camera.

Then, you get a hundred of those kinds of edge cases, and I have all the combinations of all those hundreds. That's a combinatorial problem, and that's a good thing for Scale because it means that these companies are going to keep sending you an infinite amount of data. That's one trend.



The other trend is that, as ML gets better at architecture and pre-training, it generalizes better. You have much fewer of these edge cases or situations where it degrades performance catastrophically.

I think the autonomous vehicle industry is struggling to grow. Well, Cruise and Waymo may pull it off, but clearly, there's not as much money being put into it. So if I was running a pure data annotation company, I would probably want to either go into other parts of the stack because a lot of stuff that Scale was offering, like offshore labelers for hire, APIs to define instructions and versioning, labeling, and annotation, you don't need it anymore.

If you are building in a future world, and that's where we see Nyckel, you have the domain expert like the person building the product, the product manager, or the developer labeling themselves because they know the data intimately and they can get rid of so much overhead in terms of writing instructions and payment incentives to get these outsourced data labelers to be motivated.

**Another trend we are seeing is companies like Flexport, Convoy, or Workrise using a lot of computer vision to scan physical documents like invoices, purchase orders, and bills of lading and digitizing them by using AI. Do you see that growing as a use case for data labeling companies to a similar scale as say, the AV companies?**

Fundamentally, it's the same. You have to ask yourself—what is the magnitude of the trend? How open is the problem? How diverse is the dataset that you are going to see?

I would argue that there's no problem as open as driving around the city. It's an order of magnitude more open than scanning documents, even though the documents could have oil stains on them or they could be crinkled. That's nowhere near the same levels of data diversity.

Then, you still have the other trend where the models themselves are becoming more robust and better and better, so you don't need thousands of images or documents to train the model. So, no, I don't see that driving a second wave of data annotation.





## **Tell us what inspired you to build Nyckel. How did the company start?**

Sure. My co-founder Dan and I come from different backgrounds. He's a software engineer, and I am an AI researcher and engineer. He was building a website called What's That Charge, which is Urban Dictionary for credit card statements. He was like, "I just need to classify these pieces of text. Is this a charge or not?" He was frustrated by the lack of an API for that. There were APIs that were pre-trained. But that would typically be this generic spam, generic offensive text, or content moderation. That's not what he wanted because this is a credit card statement, so it's a very specific structure.

Then, there was the whole go-to-Amazon SageMaker and build your whole MLOps-pipeline-from-scratch issue. He thought that there should be a simpler solution for this where the API layer is exactly the data layer. You just provide the desired inputs and outputs, and then, everything happens behind the scene. You call with new inputs and you get the predictions.

From my perspective, every time I do a new AI project, I've to do the same thing, start building the whole pipeline from scratch—the annotation, the training, the data engine, the infrastructure, elastic, and deploy. I got tired of doing it every time and wanted to see if I could build it in a very general way so as to do it once and do it right. That's how Nyckel started.

## **How does Nyckel work? What does your AI stack look like? Do you own your AI stack end-to-end or build on top of other AI models?**

From the customer's experience point of view, I think, we've taken it to an extreme point where we don't even talk about models. The customers have no idea what model is powering their data. What we do give them is, we train in real-time, in maybe five to 10 seconds and we do cross-validation behind the scenes. They just upload their data, and they can immediately see how well they're doing on their own data.

So instead of giving all these fancy stats, like ROC curves or precision vs recall, or mean AP, that people don't really understand, we show them the actual data that they uploaded,



using cross-validation. They can see if this is actually doing the right thing.

Now, the way we make that work is a highly, highly parallelized distributed AutoML engine. We have several deep nets that we developed in-house, and we have several that we adopted from open-source channels. We basically try them all at the same time on their data. Then, you split it up into the tiniest component, spin up a node to do that, and then, consolidate the results back as quickly as possible.

### **Help us understand the differences between AutoML and ML-in-a-box?**

AutoML is a pretty well-defined term. The way it works is, if I give you a set of annotated data—inputs, and outputs, you give me the best possible model for that data. An AutoML system is a system that searches the space of all possible ML models, or AI models, and returns the best fit for your data. Searching, meaning also training. But you do not just train one model; you train a big space of models. An AutoML engine is a part of any machine learning stack.

When I say end-to-end ML or ML-in-a-box, I mean the type of company that Nyckel is building where you, as a customer, don't actually touch the ML models. It means that the APIs that you experience are at the data level. You're just giving the inputs and the outputs, and don't concern yourself with triggering training—you're not selecting models or tweaking any parameters. You're only concerned about your own data and what you want the model to do.

Most ML systems today rely on so called “transfer learning” where a AI model is pre-trained on a large general body of data and then tuned to work for your specific problem. Basically, the networks are so “smart” that with just a few little inputs, they learn to do what you want.

### **What was your initial product-market fit? How did you get your first handful of customers, and what did they use Nyckel for? What does your core customer profile look like—including company type and buyer role?**

The core customer profile is basically either a CTO or a senior product manager at a small to medium-sized startup.





One of our first big customers was a Brazilian fintech company, Foregon. They had a problem with spam where people signed up, and they used profile pictures that weren't their own profile pictures. They were able to train a Nyckel function to say, "Is this profile picture actually a real person or one of the fake ones from the library of fake photos that they use?" The customer there was their tech lead. He just created an account, uploaded his own data, trained the model, and put it in deployment in a day or two without talking to us or asking us for help. He did it in a pretty self-serve manner.

I think the beauty of machine learning is that we're seeing very diverse use cases at Nyckel. We have a company called Gardyn that builds in-house intelligent gardens. They are equipped with cameras and stuff, and they take pictures of the plants, and they send them to Nyckel to check if it is wilting. Does it have fruit on it right now? Is it pests? That's on the computer vision side.

We have a company called SpyScape that builds an augmented reality game where you walk around the world taking pictures with your phone. They also use Nyckel to add intelligence and a semantic layer to the game.

The last one I want to mention is a company called Taimi, which is a dating community of the LGBTQ. They use Nyckel for content moderation—Is this comment suggestive or not? Are they trying to scam you, or is it too explicit? In their case, because it is a dating community, they can't really use off-the-shelf content moderation models because the bar is just a little different, so they needed a custom solution, and we were able to help them with that.

**Do these companies bring their own data to Nyckel, train the models and then, take the models to production?**

Yes, that's right. They all bring their own data. I believe very strongly that this is important. Here is why:

Before you put any ML model in production, you should at least test it on your own data because the state of AI, as glorious as GPT-3 may seem, is still always an issue with domain shift. My position is always to provide enough data to convince yourself that it is working now, and that's maybe ten or 100 or so data points.



What we realized with Nyckel is once you have 100 points, you can use cross-validation and actually train on those points as well. You can split it up into chunks, train on a subset, and evaluate. Now you've got two things at once— you fine-tune the model to your data and you convince yourself that it works for your data.

That's why we tell our customers that you have to upload your own data. You have to actually sit down and annotate it. Some of them have their own labels from some database, but more often than not, they just annotate in the UI. It takes 20 minutes or so for 100 samples and then, we train in a few seconds, deploy it immediately, and they're done.

I think one of my biggest pet peeves with certain aspects of GPT-3 is that if you're doing generative modeling like you want something to start generating content for you, maybe it's okay to use it directly. But if you're using it for classification, you need a data engine on top of it. You need some way to check if it's working for you. This is not really sustainable to just put it into a prompt and hope that it works for other situations. You need a way to define—here's what I want it to do, now go do it.

**There's a trend of companies building on top of existing foundational models rather than building their models with proprietary data. Do you see this as a headwind for Nyckel?**

I think, fundamentally, you shouldn't build a company around a particular model or model architecture. You should build the company around the function type. If you offer machine learning text classification as a service, that's one of the things that Nyckel does and one of the things you can do with GPT-3.

With Nyckel, we just tell them to let us know what input and output they want, and we'll take care of all the AutoML stuff. There's nothing stopping us from calling GPT-3 if we think that gives us the best value for our customers, and then it's up to us to do the engineering and so on to make that happen. The customer's responsibility is to review the predictions based on their own data and see if it's good enough for them, and then, they might tweak the label set, add more data, and so on.



We think that is the right abstraction and again, we can do whatever needs to be done behind the scenes. Just saying, “Here's one model, it's magic, call it and hope it works for you”, is not the end-all-be-all for machine learning.

**Critics believe that LLM/GPT latency would have to come down and costs would have to reduce about 10x before it would be feasible to integrate AI-generated results into search. How much of a bottleneck is pricing and latency for Nyckel?**

Most of the models we're using right now are not as big as GPT-3, so we don't have the same issues. None of our customers have issues with pricing or latency.

The way we've set up our architecture is that we have a suite of deep nets that are shared among all the customers. They do some of the processing, and we can amortize the cost of keeping those nets warm. Then, there are secondary nets that customize and make it do exactly what the customer wants, and those are very light. So the cost of deploying those is almost negligible. It allows us to spin up and have a very elastic infrastructure on those shallow nets or those smaller nets while we amortize the cost of the bigger nets across the whole customer base. I think that's what GPT-3 does as well. It has one deployed net that powers all queries.

The problem is when I fine-tune that GPT-3 model, it becomes very expensive because you have to have a whole GPT-3 warm for every customer, which is probably not very feasible.

I think the trend line there is very hard to predict. Now we're getting into research territories. Can you get a model that generalizes as well as GPT-3, but it's like a hundredth of the size? I don't know. It's a big leap, but that would probably change the game for them, at least.

**We see companies like Labelbox, Snorkel, and Scale moving into offering not only labeling but also APIs and services to train ML algorithms for the programmatic classification of images and text. What advantages do they have from vertical integration, if any?**

My perspective is that they should all be packaged together as a single product, and the customer should only interface with



their own data. To give the best value or experience to the customer, you should integrate everything into one product—the whole ML stack. In that sense, I think they're doing the right thing.

I can't see a future where ML teams will cobble together these pieces the way they have been doing it. There are so many synergies to be had—the right data engine, coupled with the right AutoML engine, coupled with the right discovery engine. It's all the same thing and it's very hard to build an effective product around just one of them. I think these companies are doing the right thing.

I'm just hoping for their own sake that they can find a way to really integrate all these different products. It's not enough to have ten different products. Even though those ten products cover the whole space, they also need to talk to each other in an effective way. I think that's where the challenge is for, maybe, a bigger company to get that right.

**Which are the companies that you believe are building a cohesive user experience spanning across different stages of the ML lifecycle?**

One of our key competitors, the product we think has it most right, is Vertex AI from Google. We think there's a lot of ergonomics of the UI and API and there's a lot of things they could do better. But, as far as the big picture is concerned, what they're offering is the right thing.

There are also a lot of startups doing something similar. There's a company called Roboflow, although they cater more to machine learning experts than we do. Akkio does kind of what Nyckel does, but for tabular data. There is a company called Levity from Germany that also tries to do the full abstraction. In short, there are these big cloud players that have products, which are good but not truly great. Then, there's a pretty big field of startups trying to do the same thing.

**Drawing a parallel between MLOps and DevOps, a lot of DevOps people actually came from Unix and were used to working with these point solutions. Then, they switched to GitLab with versioning, CI/CD, and all of that built into one platform. Do you see something similar happening in**



## **MLOps? Why did MLOps practitioners start with point solutions? What are the drivers for them to change?**

It all happened very quickly. ML started taking off just five or six years ago, and all these point solutions started coming up. Aquarium came out five years ago, for example. It's the same thing as Scale's Nucleus product. They try to solve the piece where you have billions of data points. Which ones should you send for annotation? Which ones are at the corner cases? How do I find more data that is similar to the corner cases? This is just one example.

People realized that "Okay, machine learning is starting to work, we need to really prioritize it." Then, all these companies started cropping out to help with pieces of the puzzle. For instance, Weights and Biases helped with monitoring the models.

All these companies were built to solve a specific problem for the ML teams, and then, it was left to the ML teams internally to piece this all together to make something cohesive so they could train and practice things.

The next step in MLOps is what happened in DevOps with these toolchains merging into one because it's really just one thing that they're doing, different aspects of the same thing, and it's best provided in a single package.

## **Scale seems to be doing something similar to Nyckel with their data annotation studio and APIs to pull the results into your app. How do you see Nyckel positioned vs. Scale?**

I think they're Nyckel's competitors. Even though it pains me to say this as an ML engineer, I believe that ultimately, companies will hire fewer and fewer machine learning engineers as the technology matures. They will be outsourcing more and more of the stack, and it makes sense for them to do so.

Imagine you want to do content moderation. Then, who is the right person to train a model—Is it the content moderator who understands the nuances of the users of the platform? Or is it a machine learning engineer who has no context on the data they are training on?





We really think that the person training and defining what the model should do is not necessarily a machine learning expert but the product owner, and hence everything else behind the scenes should be productized and commoditized so that it can be outsourced.

I think at the high level, we are essentially selling the same thing as Scale is. It all comes down to execution—How is it packaged? How quick is it? What are the limitations? If I am right, in the future, the data will be the right API layer, just inputs and outputs, and then, all the companies that do MLOps will basically consolidate and start providing API experience to their users.

**Do you see this trend as similar to Webflow selling to the marketing teams so that they don't have to run back and forth to engineers to spin up marketing websites/landing pages? Will Nyckel, Scale, or others abstract the complexities of running/managing ML so product owners from marketing, finance, or other orgs can build on top of them in the future?**

That's right.

In retrospect, it seems crazy that if you want to add text messaging to your app, you'll build that engine yourself. You are, of course, going to go to Twilio. The same thing in ten years will come to ML. If I want to add some text classification to my app, it would be madness to set up a whole machine learning ops for that. I'll just call Nyckel or Scale. The reason that's possible, again, is that I don't need 10,000 annotated samples anymore. I need 100. Once you need 100, you can just do it yourself in an hour, and now you're done.

**What do you see Nyckel becoming five years from now if everything goes well? What does success look like for Nyckel?**

The good news and the bad news are that our use cases are extremely broad. I think every company in the world could potentially be a Nyckel customer because every company in the world has some text or image or some tabular data that they want to classify, search, or index. So we are doubling down on the platform play.





Bigger companies already have their own ML team, and those companies are a little bit hesitant to throw it all away and use a Nyckel API instead. So most of our companies are earlier in their trajectory and our goal is to find more and more of those. Our goal is to become the household name for machine learning among non-experts, the same way people think of Stripe for payments or Twilio for telecom.

The way we achieve that is to do two things. First, we double down on this simple abstraction where we hide all ML complexity behind a simple API at the data layer

Second, is to broaden our ML product offering. There are many different types of Machine Learning use cases. Do you want to classify something, search for something, detect things, or do named entity recognition? We try to build out these function types and have as much coverage as possible for anything people want to do that falls under the machine learning umbrella.

## Disclaimers

*This transcript is for information purposes only and does not constitute advice of any type or trade recommendation and should not form the basis of any investment decision. Sacra accepts no liability for the transcript or for any errors, omissions or inaccuracies in respect of it. The views of the experts expressed in the transcript are those of the experts and they are not endorsed by, nor do they represent the opinion of Sacra. Sacra reserves all copyright, intellectual property rights in the transcript. Any modification, copying, displaying, distributing, transmitting, publishing, licensing, creating derivative works from, or selling any transcript is strictly prohibited.*