EPISODE 569

[INTRODUCTION]

[0:00:00.3] JM: There are a hundreds of different databases. There are tens of continuous delivery products. There's an ocean of cloud providers and CRM systems and monitoring platforms and sales prospecting tools. The range of available software products is so diverse that it can be overwhelming to figure out which products to buy.

Siftery is a company that was started to index the software products that exist in the world and to help buyers make decisions. Siftery can build a data set from your website, or from your Google account, assess your software stack and compare those software products to others on the market.

In a previous show with Ayan Barua, we discussed how engineers should explore the question of build versus buy. In today's, episode Ayan joins the show to discuss how Siftery has evolved and the engineering behind Siftery itself. A newer Siftery product that they recently built is called Track and it can ingest banking transactions and QuickBooks records and other transaction histories and you can use that information to then assess your cost structure.

Ayan is a good friend and it's awesome to see his company Siftery doing so well. We spent most of the conversation talking about the product development process of Siftery, and then the latter part of the conversation talking about how Siftery turns product development ideas into engineering processes.

I hope you enjoy this episode with Ayan Barua.

[SPONSOR MESSAGE]

[0:01:42.6] JM: Azure Container Service simplifies the deployment, management and operations of Kubernetes. Eliminate the complicated planning and deployment of fully orchestrated containerized applications with Kubernetes.

You can quickly provision clusters to be up and running in no time, while simplifying your monitoring and cluster management through auto upgrades and a built-in operations console. Avoid being locked-in to any one vendor or resource. You can continue to work with the tools that you already know, so just helm and move applications to any Kubernetes deployment.

Integrate with your choice of container registry, including Azure container registry. Also, quickly and efficiently scale to maximize your resource utilization without having to take your applications offline. Isolate your application from infrastructure failures and transparently scale the underlying infrastructure to meet growing demands, all while increasing the security, reliability and availability of critical business workloads with Azure.

To learn more about Azure Container Service and other Azure services, as well as receive a free e-book by Brendan Burns, go to aka.ms/sedaily. Brendan Burns is the creator of Kubernetes and his e-book is about some of the distributed systems design lessons that he has learned building Kubernetes.

That e-book is available at aka.ms/sedaily.

[INTERVIEW]

[0:03:17.9] JM: Ayan Barua, you are the CTO at Siftery. Welcome to Software Engineering Daily.

[0:03:21.7] AB: Thanks, Jeff. Thanks for having me again. It's great to catch up again.

[0:03:25.4] JM: Last time we spoke, we talked about the state of Sass and what you're building at Siftery. Explain what Siftery is for those who did not catch that episode.

[0:03:35.3] AB: Sure, Jeff. Siftery, we started the company in 2015 June, the mission and the vision of the company is to help buyers of software, which is pretty much everyone, effectively buy, select and manage their Sass. We built a discovery platform, which we call Siftery Discover. We launched in 2016 in early Jan. It was a great product launch. We launched in

product hunt and the product itself was a way to understand which company is using what product and inversely how products are doing.

Using this data set, we built out a recommendation engine that since then have been used by 25,000 companies, big and small. That's where we are with Siftery Discover. Lately, we have also launched on the product Siftery Track. It's an effective way to manage the spend of your software footprint. I can go into more details later.

[0:04:34.4] JM: Sure.

[0:04:35.4] AB: These are the two products we have and we are seeing some traction in the market around these two products.

[0:04:41.9] JM: My experience with Siftery has been that it integrates with your stack some way either through, depending on what Siftery product you're using can integrate through your Google account and can look at the products you're using through Google. It can look at your website and just tell from the website that it's using certain products. The newest product, the Siftery Track thing is basically a way to integrate with your accounting software, or through your bank so that it can look at your transaction history and make associations with products based off of that transaction history.

All of this is useful because it is a self-creation. It just makes a vision of your stack that you can look at and also generates the amount of costs that you have associated with that stack, and potentially gives you other stacks that you can compare it to, other products that you can compare it to. It's a way of maybe assessing your stack and looking at other options, perhaps ways to save money. I think of it as a category creation.

I think, one thing I've taken away from it is that there are a ton of options for things that you can buy in the software world. Engineers probably should be buying even more software than they are, because rather than building stuff from scratch – this is something we touched on last time, the whole idea of do you build it or do you buy it? Are engineers buying as much software as they should be, or do you think people are maybe saving money a little too much? Should they be buying more software?

[0:06:25.5] AB: Interesting question, Jeff. I think that specialization around software and what it can do has just increased in the last two years. We've seen this with almost every category that very niche products come out and they are so good at automating that part that you would want a couple of engineers to be building out.

What we've seen is proliferation of software, and what we've seen is more categories and more niche products that have come out and effectively served a very strong use case. That is just a one way direction according to me from – the data also tells us that we started with a database of 3,000 products. We've now gone up to 40,000. I think it's all around us, I guess, that you are using more and more tools to automate specific use cases, and that's an irreversible process according to me.

[0:07:25.9] JM: I talked to a hiring manager recently from a really large tech company, one of the biggest tech companies out there and he said that most of the new grads that they're hiring, their job is about stitching together external API's, it's about open-source software, it's about copy pasting from Stack Overflow. It's not writing stuff from scratch.

It really does seem to be a sea change, even in how people are writing software. The software they are writing ties in with the software that you're purchasing, because there are these very good high level APIs that that you can purchase off the shelf of course, and they just make writing software much easier.

Siftery has a variety of user types from what I can tell. You've got CIO-type of people who are the buyers, who might potentially be approaching Siftery from a cost management standpoint, or from a product discovery standpoint. You also have engineers that are looking for just products that they could potentially use to build off of. How do the different types of users approach Siftery?

[0:08:35.5] AB: I think yeah, you covered two of these, which is the CIO, CFO, or any decision-maker and such, basically people are trying to understand what they're using at their company and this problem of what we are using at my company is really compounded with the

scale of the company, right? A public company would end up using thousand vendors, thousand plus.

A smaller organization might use 50, so definitely understanding what my software footprint is, especially because the decision-making around purchasing and around adoption of that software is very decentralized today. Trying to understand what the footprint is, trying to understand what the shadow ID is, where we didn't really know that we are using this, but here let's standardize. That's one.

Then the other part is understanding how much you're spending. If you can cut down and you can basically reduce waste and reduce the footprint in general in terms of licenses and renewal around licenses. You can do a lot of, look at all that data and you can take a lot of intelligent decisions. That is one.

The other one is as you mentioned, engineers and other groups of people looking into it, one is social proof as well. Let's say a new product has come out and the moment Siftery is surfacing that and you go to the product profile on Siftery and you see some decent customers using that product, creating good NPS score around it, writing some good reviews, they are also being surfaced in good comparisons. That's social proof for others to understand that this is a product that is gaining market, and it is a well-designed, well-architectured product. Satisfying the use case that it is intended for.

Use case understanding is also something that people do on the platform. We have gone really deep in terms of categorization. It's still a work in progress, but we have I think over 700 categories. We have broken down many categories, which haven't been broken down before. It's okay if all those categories have five or six products.

[0:10:52.8] JM: What's an example of a niche category that you've broken down?

[0:10:55.6] AB: If you look at social media management overall, right? I can give you examples in the show notes, but there are certain products that have come out, which do a very specific task. It could be as simple as understand your LinkedIn connections. It will fall

somewhere in the CRM bucket, but it's so niche and it's like 15 bucks a month. It is really effective in understanding your LinkedIn graph and then what you can do with that graph.

[0:11:27.9] JM: By the way, this tool might sound boring to especially – maybe to some engineers listening. Why would I care about that? I can just tell you from my experience doing a podcast and a podcast is a weird niche business, and we do weird niche things. The way that we interface with podcast advertising buyers, for example is very strange. We use unusual products for that. That's why it's actually quite useful to have a way to discover an index very niche products, because the number of niche businesses with niche needs is increasing.

[0:12:06.0] AB: Absolutely. I think this is the data supports this, but every day new products are coming out. Earlier you just had a salesforce, or Microsoft, or a big category winner. You don't see that very often these days. Even the big companies are decentralizing themselves and there are many products adding up to their revenue cycles. I think this is an increasing direction.

This is a direction that as I mentioned is just one way. You will find specialists working with really good products. That combination is very powerful too. Many years ago, Instagram got to that success with 13 people, is because they had a really great team, which knew how to architect good systems and they also had really great systems at their disposal. I think the winning formula here is going to be great people working with great set of tools. Our job is to – Siftery's mission is to help all these great people to be really effective with their toolkit, in spite of the human progression, I guess that's how we've gotten here.

[SPONSOR MESSAGE]

[0:13:33.4] JM: We are running an experiment to find out if Software Engineering Daily listeners are above average engineers. At triplebyte.com/sedaily you can take a quiz to help us gather data. I took the quiz and it covered a wide range of topics; general programming ability, a little security, a little system design. It was a nice short test to measure how my practical engineering skills have changed since I started this podcast.

I will admit that, though I've gotten better at talking about software engineering, I have definitely gotten worse at actually writing code and doing software engineering myself. If you want to take that quiz yourself, you can help us gather data and take that quiz at triplybyte.com/sedaily.

We have been running this experiment for a few weeks and I'm happy to report that Software Engineering Daily listeners are absolutely crushing it so far. Triplebyte has told me that everyone who has taken the test on average is three times more likely to be in their top bracket of quiz scores.

If you're looking for a job, Triplebyte is a great place to start your search, it fast-tracks you at hundreds of top tech companies. Triplebyte takes engineers seriously and does not waste their time, which is what I try to do with Software Engineering Daily myself. I recommend checking out triplebyte.com/sedaily. That's T-R-I-P-L-E-B-Y-T-E.com/sedaily. Triplebyte, byte as in 8-bytes.

Thanks to Triplebyte for being a sponsor of Software Engineering Daily. We appreciate it.

[INTERVIEW CONTINUED]

[0:15:31.1] JM: Has anything surprised you in the data that you have all these people that are plugging in their stacks into Siftery. What surprises have you seen in the data about how people are buying and using Sass products?

[0:15:43.9] AB: The data is very multi-dimensional. We don't tend to analyze data to the extent where we are making data business out of it. What we do, we don't have customers who are trying to understand their data get better We dig in at times trying to understand what's going on. There are a lot of things that have surprised us in terms of how people are – there are certain segments of the industry, which are very reliant on certain – other types of software.

For example, all these e-commerce companies, right? You could take stitch fakes and all other companies. They're very reliant on shipping software, which is something that the customers don't see. There is this really interesting group of companies, which are not the darlings of

venture capital, or not really well-known, but they are almost pumping hundreds of millions of dollars of worth of shipment across the board almost every month.

We've seen these categories of software, which do not tend to identify easily, but then you realize, okay these are powering economies. Background checking services, background verifications; I check I just raised hundred million dollars. .They are powering so many on-demand companies and similarly shipping software as I mentioned in the e-commerce business. They're also powering a lot of that segment. Very interesting data set, I think if you have specific questions, I'm happy to answer them.

[0:17:18.9] JM: Yeah, these products that are niche, and it's always cool to hear about a niche product that is widely adopted within a category. You see these niche products actually end up making lots and lots of money, because oftentimes their niche categories end up to be bigger than you anticipate. It's also cool to see, when a company like Slack makes a widely accepted product, a flexible product, a company that is not a salesforce, or a Microsoft, or a Google, a dominant company adopts, creates something that is so flexible.

Are you seeing any new flexible products come to mind? Do that tons and tons of companies are adopting across all kinds of industries, things that are – that your layperson may not know about? Are there any subtle products that you think are sleeping giants?

[0:18:12.0] AB: Yeah, I think if you look at Slack, it's a communications tool at the end of the day. You will have work software, which are very tightly integrated with your workflow. Slack is one of them. I'm sure that Front is another one.

I think if you look at software which deeply integrates with your workflow, and if that workflow is very company-wide, then you will see products that are very flexible, that are very customizable, that are very good overall in terms of experience. It's just given that you will see them really win the market, because adoption today is very quick and it's very easy to grow fast, just through product experience and word of mouth through marketing.

[0:19:15.3] JM: Wow. Do you see any interesting observations about cloud providers? Because you obviously also can have visibility into the – it's not just the Sass, but it's also the

platform as a service and the infrastructure as a service. Are you seeing interesting observations in a Google Cloud versus AWS, or more long-tailed cloud providers? What kind of observations do you see there?

[0:19:37.0] AB: Yeah, I think the data is pretty sensitive, so I wouldn't want to give you numbers in those what we are seeing. What we are also seeing is a tiny bit of the actual spend. We have only one company, so extra pulling to a industry-wide benchmark may not be the right way to do it.

One observation that I have is that unlike what people think that companies like – smaller companies and smaller hosting companies are dead, they are not. AWS is probably really great when you have a toolkit. You have a product that needs a lot of other tools, so you need a database, you need a cloud front like steward service, you need you need maybe auto-scaling built-in. When there's sophistication, you need a lot of these platforms.

For a lot of products you don't need these sophistications. The line nodes and the OVH, they are doing really well, because overall, more and more platforms are coming online. Not all of them have a strong complex ecosystem, so what happens is that you're perfectly fine just hosting your line node.

A lot of the experts actually don't even rely on AWS. I do see that the usage around a lot of these products have been gone down. They are equally keeping pace with the bigger giants. That's one thing that I think I personally have noticed. We haven't really published any report around that. It’s just an observation that I’ve had.

[0:21:10.7] JM: Man, and I bet those businesses are getting better and better, because they get better and better economies of scale, even if their costs, or even if their prices stay flat and their users are useful to Lanode, or to HostGator. I still use HostGator. I started using HostGator in high school.

[0:21:29.8] AB: You wouldn’t need to go to AWS if you a use case. I think C3 is one of the machine as to bring clarity there. There's a lot of hard mentality as well. Two years back also, I've mentioned that you don't need to be a Docker if you don't have to be a Docker, right? I think

with Silicon Valley engineering, we all want to play with cool products, and that's great. That pushes the boundaries around innovation.

We've gone from three-people company to 20 plus people. What I see organization is a challenger on how do we understand that within this plethora of tools that are very easy to adopt, how do you choose better? That that problem is going to reflect in a much bigger way as time progresses.

[0:22:18.7] JM: All right. Well, maybe I won't ask you about Kubernetes today. I started using your product Siftery Track, which is the accounting tool. I use Siftery Track and it integrates with your accounting, or your banking software, and then it tells you what you're spending money on, and it helps people control their Sass spend. I guess, my first question is it hard to get people to feel comfortable authenticating with their accounting software, or with their banking software? Getting you to authenticate with Google or Facebook is not too hard, but I feel like OAuthing with your bank account is a little bit, makes people a little more uneasy.

[0:23:04.4] AB: That is certain. Before I answer this question, can I just talk a little bit about how we came about to this software? It ties back into the answer that you're looking for.

[0:23:16.0] JM: Sure.

[0:23:17.5] AB: While we were building sifter.com, our vision was always to cater to the buyer side and less to the vendor side. While we started growing siftery.com, a lot of people, a lot of our own users gave us feedback that this list is great. I can see what we are using on the platform, but I am not sure how much I'm spending, and also if I'm spending the right way.

The other question that a lot of users post to us was the fact that do we even know if we are paying the right amount? Because the moment you go into the enterprise deal, pricing is just basically going to be boiling down into negotiation. How much are we spending? Are we spending the right amounts? Are we organizationally looking at reduction of waste very concerted, very structured way?

These were certain questions that people were throwing at us comes consistently. I think from there, on all these feedback, all the discussion that we've had with our users, it seemed like there could be automated ways of pulling that data in and not really getting users to verify. While we started brainstorming this, we realized that the best source of truth around how much you're spending is your financial source of truth, so whether it's an accounting system, whether it's a bank, all your credit cards, your expense management systems are finally resided within your bank or your accounting system.

I think we started a small beta. We tried to understand which accounting systems we should go and tap into if at all, which banking institutions should we go and pull data from. We polled our user base, try to understand what their security concerns were, what their privacy concerns were. From all that discussion, I think emerged Siftery Track, which is a very simple way to connect your financials just to offshoot to our database, and then we build a beautiful, very magical dashboard out of it.

This product is kept separate from sifter.com, simply because we didn't want to immediately merge the two data sets. Track has been given its own bubble. Track has been given its own structure. We have started at a very secure zone to start of it, and then the privacy of the data is also paramount.

We've looked at the architecture, the engine architecture the way we go about doing things in a very different way. Track is a completely different product that's why. To answer your question around security, it's a big concern so we are under SOC 2 audit today. This product will have to be very, very secure and –

[0:26:24.5] JM: What is the SOC 2 audit?

[0:26:26.5] AB: There's a bunch of compliance that you can go and get yourself certified with. SOC 2 is one of those industry standards, which look at your entire infrastructure, look at your InfoSec policies, look at your personnel. This is a formal audit that you go through, looking at your entire footprint of what you're doing.

[0:26:50.1] JM: This is not necessarily related to the fact that you have to integrate with banking APIs. This is just a general audit that's for your own company health?

[0:26:58.6] AB: For our company health. We are also working with secure APIs. For our banking integration, we are using plan Yodlee, so these are two top-of-the-line products. For QuickBooks and Zero these are all OAuth and they have thousands of such apps on the marketplace. We are using OAuth and SAML2; they're very – it's needed. The fetch of the data is very secure, and then it's the onus is on us to make it really, really bulletproof. We are focusing on the second part.

The first part we're integrating with the banking source of truth, we are not doing that because we would need PCI, DSS compliance and all that. We are reliant on fantastic APIs like Plaid and Yodlee, which are working with the financial organizations and they are – their security is much ahead of the curve.

[0:27:53.7] JM: As you said, you this first product was Siftery Dscover and this is where you can browse different products and find related products and compare prices, compare usability, compare popularity. Then you built an entirely separate product, which is Siftery Track. You probably were able to reuse some aspects of the first product, like the index of products, but then you had to figure out how to connect certain receipts, or transactions in my banking history, or in my QuickBooks history with the products that they correspond to.

Maybe you could tell me about the process of building a second product, because this is something that it is pretty hard for a lot of companies, because they start with one product, they build a core competency in that one product. To go off and build a second product can often be a challenge. What was the process like? What was the evolution like of developing the second product?

[0:29:06.2] AB: Yeah, this is actually a very interesting question. I don't get this question often, even when I'm talking, chatting with my friends. We had to rethink the way we are organized, because siftery.com is a fast-paced consumer product and it moves at a certain pace, features are getting developed in a certain way. The data is open by default and for Track we had to completely rethink the way we do things.

It was not easy, because this is the same team which is building a very different product and the core audience has shifted from more browsing the behavior that sifter.com encourages more browsing, more understanding of you looking at a much wider footprint of data. But here, you're just looking at your own data anywhere.

One is catered for breath, this one is catered towards depth, which meant that the product thinking itself was a little different. Every product is different. We had to rethink a lot of the ways we did things, including architecture. This one had to be a different architecture. We had to build security from the ground up. We're not saying that sifter.com is not secure, but here the data security and data privacy is so paramount that we had to err on the side of paranoia.

We had to bring a couple of people in who are experts here and we had to carve out certain members of the team, had to work on this exclusively. I think it is very hard to build a second product, which is not just a feature, but it is a completely different product. It caters really well to the mission and vision of the company. We want to a really big business on the buyer side and hold off or not make that grade of business on the vendor side, where you can sell data, you can you can sell lead gen. I think that two years ago, I also I had mentioned that some of the core competencies platform for it to be the platform of Discovery, we need to have that healthy balance where it is not totally catered for vendors.

We need to give we need to build the business out on the buyer side, so that the marketplace is more neutral. I think this product fits in really well into that thesis that we need to build a product, which every company uses and it's on the squarely on the buyer's side. It's a Sass product. We have marketplace so sifter.com plus track.siftery.com, these two are a combination of Sass plus marketplace. The Sass is where the business model is emergent and the marketplace stays more neutral in that case. Even though this was hard, I think it was tied – it fell really well in the arms of our mission and vision and. Yeah, and sometimes you just have to do the hard things, I guess.

[0:32:12.8] JM: Indeed. You mentioned hiring there. Hiring is one aspect of scaling. You've gone from three to 20 people probably since we last spoke. Maybe you had more than three people when we last spoke, but what have you learned about that scaling process? I'd love to

hear about it, both on the personnel/management side and on the infrastructure engineering and product development side.

[0:32:38.7] AB: That's a crazy question. I think that every day. I feel like I am so not up to the mark. Even though the product is growing and Siftery is growing, because every day it throws a new challenge. Within the team also, we started with a group of people, there is a lot of expertise that have been built up now. We are also doing new things and that expertise at times, that is not helping us as much that you would want it to help us.

We had to look at security in a very fundamental way, and we didn't have anyone. Thinking what security the way we needed to think about Siftery and about security, which means that we needed to go and find people who could re-architect parts of the platform. That is always hard. It is just not track.siftery.com. It is also going to be reflective of the entire platform.

New challenges are thrown at us every day, and most of the times we are under the impression that maybe today we are not going to hold it together, but we end up doing it. It's been crazy actually.
[SPONSOR MESSAGE]

[0:33:55.7] JM: Software workflows are different at every company. Product development, design and engineering teams each see things differently. These different teams need to collaborate with each other, but they also need to be able to be creative and productive on their own terms.

Airtable allows software teams to design their own unique workflows. Airtable enables the creativity and engineering at companies like Tesla, Slack, Airbnb and Medium. Airtable is hiring creative engineers who believe in the importance of open-ended platforms that empower human creativity.

The mission of Airtable is to give everyone the power to create their own software workflows; from magazine editors building out their own content planning systems, to product managers building feature roadmaps, to managers managing livestock and inventory. Teams at companies

like Conde Nast, Airbnb and WeWork can build their own custom database applications with the ease of using a spreadsheet.

If you haven't used Airtable before, try it out. If you have used it, you will understand why it is so popular. I'm sure you have a workflow that would be easier to manage if it were on Airtable. It's easy to get started with Airtable, but as you get more experience with it, you will see how flexible and powerful it is.

Check out jobs at Airtable by going to airtable.com/sedaily. Airtable is a uniquely challenging product to build, and they are looking for creative front-end and back-end engineers to design systems on first principles, like a real-time sync layer, collaborative undo model, formulas engine, visual revision history and more.

On the outside, you'll build user interfaces that are elegant and highly customizable that encourage exploration and that earn the trust of users through intuitive thoughtful interactions. Learn more about Airtable opportunities at airtable.com/sedaily.

Thanks to Airtable for being a new sponsor of Software Engineering Daily and for building an innovative new product that enables all kinds of industries to be more creative.
[INTERVIEW CONTINUED]

[0:36:15.7] JM: How do you scale the hiring process? I mean, there's so many different ways you can hire. You can do it completely in-house, you can use executive recruiters, you can use hiring services. This is a topic that I have not covered as much on the show as I would like, but somebody like you who's quite busy, well how do you approach hiring?

[0:36:36.3] AB: Increase the team has a cost, right? Once you raise a little bit of capital, all you're making a lot of money through different products. You're hitting profitability, whichever way you are getting capital. It's easy to think of hiring as let's throw more people at this problem. I do think that's not going to scale.

Even though we raised a bunch of money and we have parts of our team in engineering in India, we haven't really grown to a point where it's all crazy, so it's still a very well net-tight team.

Attrition has been really low, close to zero. Where we have tried to – the way we've tried to scale is being very, very sure about the candidate.

I think, I spent more time in understanding an incoming request of candidature than actual coding with those. What we've seen is we've tried to understand the incoming candidate really well, whether they fit into our team, whether –

[0:37:42.0] JM: Culturally.

[0:37:42.6] AB: Yeah, whether we'd fit into their – because anybody who's coming in doesn't only keep the culture, but extends that culture and adds to that culture, so that whole culture thing has – we have a certain system of values, where new people are constantly adding and extending that culture, what we stand for remains the same, but the nuances are always in a state of flux.

I think we spend a lot of time trying to understand who this person is, so the engine hiring and even the growth hiring, the product hiring, design hiring is all based on discussion, so purely from an engine point of view, because I'm leading that org. We don't have a very structured interview. If we get a lead through a network referral, or somebody's just e-mailed me, or there's a job description that has been floating on an Angel List or some other platform and there's an incoming person.

We spend a lot of time understanding what they have done, why they like Siftery, what are they trying to accomplish. Even before we test technical skills, we try and understand if this person is going to really enjoy working with us, and obviously, vice versa. That's one way to scale and rather getting the right people.

If you have the right people, you are having to tell them less of what they should do. Instead, they are telling us what we should do as a group. Terrific individual responsibility is what we're looking at. If you have a group of people who are really good at executing their own plans and actually influencing their peers in a really good way, I think operational overhead really goes down.

We don't have a managerial layer. As such we mostly have a group of people who are pushing product and pushing code. That has scaled till now, whereas I mentioned it's too early in the lifecycle to understand whether this will scale to 500 employees, and I really hope we can get that ratio.

[0:39:44.4] JM: Last time, we talked about the engineering side of things. We talked about the stack of Siftery. I'm curious if anything has developed, what new technical learnings do you have? I mean, don't take this the wrong way, I think the more interesting questions are probably around product development, rather than engineering, if I recall last time, the engineering stack is pretty simple, right? Or has it gotten more complex? What are the difficult engineering challenges that you have started to run up against?

[0:40:18.6] AB: Last time when we spoke, one thing we didn't have was the sea of data. We had just launched, we had a thousand companies using us, or maybe even less. Then we've gone from there to millions and millions of data points. How do you store all this data? Then we have our own crawlers crawling the internet, so terabytes of data can pour in every day.

I think from an engine standpoint, we've had to really rethink and rebuild a lot on the data architecture, the storage of it, the analysis of it, and then how do you parse all that data and bring out the relevant insights out of it? I think we've gained a lot of expertise on that front. It may not be necessarily a tool thing. We've iterated on that as well. We didn't have a warehouse before this. We had to go and get our data into something like Redshift.

We had a single database, we have to look at other caching options. .There was some infrastructure components to do, this but I think the 80% of the rebuilding and the rethinking was around the data architecture itself. With Track, what we saw was a different problem, because how do you footprint and how do you fingerprint software from transactions?

Different vendors are billing you in different ways. Sometimes your AWS payment will be through amazon.com, sometimes it will be aws.amazon.com, sometimes it will be AWS. It can seem like a simple problem, but it is not always a simple problem, because it is clearing positives right with sifter.com, because the data set is so diverse and it is partly crowd-sourced.

There is a bit of error margin. With Track, this is your data. I cannot tell you that even we have a error margin of point 0.001, that even that is not acceptable, because it gives a wrong idea about your own spend. we have to get the products right, we have to get the spend right and there are lots of nuances when you are integrating with these API.

Not only your banking APIs, your accounting systems, your expense management, so you also connect Expensify, or you can connect bill.com, which is your accounts payable in the marketplace system. Then you can bring in your custom uploads, so there is duplication problems, there are identification problems. Really great ones to have, but I think that the initial bit of data was really hard to build.

That is siftery.com itself was very helpful, because we could cross all and validate initial – our initial biller customers, we could cross-validate that they are indeed using it. It's on Discover, but we are not able to find it. Maybe there are better ways to algorithmically understand the feed itself.

I think there will be Sass spend management is an emergence fee, so with other competitors where we really would stand out is the fact that we have a really wide footprint like siftery.com, which is also a really created a funnel to validate and cross politics, so all that data cannot coexist, but they teach either each other things.

[0:43:41.8] JM: All right, so first of all I definitely take back what I said about the engineering problems. Plenty of interesting stuff to discuss. You talked about two disjoint sets of problems; the first one is the volume of data and adjusting to the volume of data. The data warehousing solutions that you integrated with, or purchased, or stood up to respond to that influx of data. Is this a solved problem, or was there anything – I mean, can you just look up how to migrate my PostgresQL database to Redshift and what queries to make against Redshift. Can you just look it up, or was there anything that you knew that you had to do from scratch?

[0:44:28.5] AB: You wouldn't migrate your PostgresQL to Redshift in totality, because both of them are different sorts of – one is your analytical data, where you have tons and tons of data. You're analyzing them. Not all of them are relevant. Your PostgresQL will have your app data,

which is very relevant and your app is constantly querying that data set. One is the OLTP, the other one is the [inaudible 0:44:53.4]. You wouldn't migrate one into another.

[0:44:57.1] JM: Okay, sure. One column, one or two columns of –

[0:45:00.3] AB: What happens is that along the way, the – initially, your PostgresQL is what where all the data is, your relational database is where all the data is. Maybe you have something that won't go in place where all your data is. Then when the volume increases, you realize that, "Okay, I need to chunk this data out and put it into some warehouse where we can analyze this better."

That evolution is quite fascinating. It didn't happen in one day, but we were thinking about and suddenly we had to say that, "Okay, it's getting to a point where we need to chunk it out and add more data to it," which we are not saving today. We could save more data, but this data we shouldn't save it in our relational database, because it slows down the app, query time increases. Okay, chunk it out into a warehouse.

Again it's less about the tool, I guess, more about what we are trying to do. Something like Redshift is very easy to get started with and doesn't have a lot of learning curve, and is again on the PostgresQL engine itself, which RDS uses. I'm sure that there are other companies like Snowflake and some really cool – and you have these products that have come out.

We haven't had the chance to play with them yet, but I guess that's part of scaling as well. When you get to a point where your Redshift is throwing trouble, you set up in a way where you're not getting enough leverage and maybe along comes a way, along comes a new tool that can do it really effortlessly, and then you sort of tend to might be. I guess, forward-thinking is very difficult. In hindsight, it's easier I guess.

[0:46:41.1] JM: The other product engineering challenge that you mentioned tackling was the fact that when you have somebody who integrates with this new tool, the Siftery Track tool and they're importing all of their transactions, all of these banking transactions and you want to be able to parse all those banking transactions and turn them into a report on how much you are spending on particular types of products.

You need to be able to correlate each transaction to a product and there's all kinds of malformed transactions that could turn up, there's different transactions that could correspond to the same product, and what you were saying was that you were able to use Siftery Discover, which is the first thing that you built, this set of companies and the products that they use, you were able to use that to validate different transactional, the wording of transaction receipts basically to companies that had verified that they use certain products on the Siftery Discover product.

If you had a company that used both Discover and Track, you're able to look at the transactions from Track and look at the predictions you're making and if you have a transaction from Amazon that is strangely formed, but your mapping system maps it to being an Amazon transaction and it indicates that somebody's using AWS, then you're able to look at that same customer on the Discoverer product and validate that they are indeed using AWS.

I imagine there are a lot of other – I mean, there's probably a lot more to that story of how you develop that transaction analysis system. Could you talk a little bit more about how you built a system that learned to identify those transactions and associate them with specific products spend categories?

[0:48:46.4] AB: Yeah, we've gotten a scale where we don't need to do that anymore, but the initial bootstrap that we had to do, because looking at a transaction, you're looking at – we were looking at our own transactions, so we are huge users of Siftery Track at Siftery, so I'm constantly looking at how my engine team, our marketing teams and our design team want they spending on how we are doing that. Is it a new product that we're using, because that's also an alert that can go out from Track, the product set?

We saw that it is not always straightforward to fingerprint wrong actions as you mentioned> Malformed wrong actions, creating false positives, we were able to really validate that data set initially with our own understanding of what we have put data we've put on siftery.com. This was possible for a few initial customers who agreed to be a participant in the beta program.

That was a really, really good learning experience overall, where that 40,000 products that we have, you could inject that information into a transaction system and then a machine learning algo can figure out, “Okay, these set of transactions most likely are going to point to this product.” That was really helpful initially.

Once we got to a point where we are able to parse through 90% of the products in any transactional system, we don't need to do that anymore. It was great to use that data set and validate on particular company down metrics.

[0:50:36.1] JM: All right, well we've run out of most of our time. I’d love to close by hearing a little bit more about where the company is going, what your long-term goals are, and I guess what you're focused on building today.

[0:50:49.0] AB: Yeah, I think the mission and the vision of the company I think we’re breeding is data to buyers, or software and enable them with all the data and insights that they need to create decisions. Discover is one of those, so we are going to build more features on Discover itself, Siftery Discover, so that discovery is easier.

On the management side, I think we have to Track, we’ve seen great traction. Lots of companies have started using us, and you're almost adding 50 to 60 companies every week. It's hard to keep up with this growth, because it creates a lot of support footprint, and we’re still 20 people, we’ve not really added a lot of people.

Where we really want to go with the Track product is to get utilization going. It's already a work in progress. We have some beta customers there too, but it's not really rolled out to the entire user base. Utilization is where you can bring in your product-specific information and really understand how much are you really using a product. If you port 25 salesforce licenses, how are these licenses being utilized? A lot of the data is in the salesforce even itself.

You can bring in your identity provider like Octo or OneLogin and get to understand how many users are even active on this third-party software products. I think utilization is a big game that we'd like to go and build. There are other problems that also can really help in this whole b2b software management space, which we also want to tap into. We just try to focus on really

improving Discover and siftery.com's experience, build a really great robust tool at Siftery Track and then build utilization on top of it and like to monetize and build a revenue channel there as well.

[0:52:50.3] JM: All right. Ayan Barua, thank you for coming out Software Engineering Daily. It's been great to have you once again.

[0:52:54.7] AB: All right. Thanks Jeff. It was lovely chatting with you.

[END OF INTERVIEW]

[0:52:59.6] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plugins. Use the value stream map to visualize your end-to-end workflow. If you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on the fly. GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team, who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on GoCD with the new Kubernetes integrations.

You can check it out for yourself at gocd.org/sedaily. Thank you so much to ThoughtWorks for being a long-time sponsor of Software Engineering Daily. We're proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]