1:11

Alexey

**This week, we'll talk about identity resolution and we'll also talk about building an open source startup. We have a special guest today, Sonal. Sonal is the founder of Zingg, which is a machine learning-powered identity resolution framework. This is actually not the first time Sonal appears in DataTalks.Club. At DataTalks.Club, we have a thing called Open Source Spotlight, where we invite open source authors to demo their tools. This is how Sonal and I got to know each other. Actually, her demo of Zingg is one of the most-watched Open Source Spotlight videos. So I thought it was a really good idea to invite Sonal to talk about open source, her startup, and large scale identity resolution. Welcome!**

1:55

Sonal

Thank you so much for this, Alexey. I think the video that we shot last time was really beneficial for Zingg and I hope people will enjoy today's talk as well.

# Sonal’s background

2:06

Alexey

**I'm pretty sure they will. Before we go into our main topic of open source and identity resolution, let's start with your background. Can you tell us about your career journey so far?**

2:17

Sonal

I have had a really long journey in my career. [chuckles] And all of it has been in tech – almost 24 years of working on various aspects of technology. I started as a programmer and an analyst at an investment bank . Here I am working on Zingg. I’ve had various roles – all technical – in different kinds of domains, like telecom, banking. Then, in the last couple of years, I was running a data consultancy, where we were solving all kinds of warehousing, data pipelines, and machine learning problems.

# How the idea for Zingg came about

2:58

Alexey

**When you were running your consultancy, you probably noticed that there are some problems that most of your customers have. And this is how Zingg appeared? How did it actually happen?**

3:13

Sonal

Yeah, very much. Honestly, I started my data consultancy in 2010, which was just the booming period – the first time all this Big Data stuff was happening. I have to be honest, I saw problems absolutely *everywhere*, just setting up Hadoop clusters, installing Spark, getting them to run on EC2, what is S3? I think those are fundamental questions at that stage. I remember back in 2010-11, we were doing something called cascading for ETL, which was a programmatic way to define ETL jobs. So there were a ton of problems and Zingg was something that definitely originated as part of the problems that I saw. It's a problem I saw early in 2013. And it's a problem that kind of flummoxed me even at that time. But it's only now that I feel that the base infrastructure for such a problem is ready and for the market to accept this problem. So here we are.

4:16

Alexey

**So it means that you don't need to figure out how to set up a Hadoop cluster, install Spark there? Basically, you go to your favorite cloud provider, click a button, and then you have a Spark cluster. Well, at least this is how it happens with AWS, more or less. I haven't experimented. I have also experimented with Google Cloud Platform. It's pretty much similar, right? So now, you're saying that tools that are built on top of these tools – it's easier to deploy them, to use them. Right?**

4:51

Sonal

I think just beyond the tools for Spark, it is also about the data infrastructure as a whole, which is what we are now calling the modern data stack. This is a very de facto set of tools for established patterns, for extraction, for transformation, for loading – having set patterns in your warehouse or your data lake. The data is essentially in one place. And when you have the data in one place, this is where the data management problems actually start appearing more and more. Because before that, you were actually bothered about your extraction, your pipelines, just running your flows, your observability. But the moment you have data in one place, and you want to glean insights out of it – that's where problems like identity resolution, actually, kind of just hit you in the face. [chuckles]

# What Zingg is

5:43

Alexey

**So can you tell us about Zingg? What is it?**

5:47

Sonal

Zingg is, as I mentioned, a tool for identity resolution. And what I mean by “identity” is very simple. It is establishing whether five different records in your warehouse actually refer to the same real world customer. These records, when we do ETL and we connect to different data sources, enterprises will have data coming in from different systems, and they will have records of the customers from offline channels from the online store, from various kinds of interactions – they have surveys, ticketing. What that leads to is lack of a well-defined definition of really who that customer is.

If you are counting five records as five different customer identities, your lifetime value reporting, any other kinds of personalization that we want to do – anti-money laundering, KYC – they all get affected. Now, this is just a customer. But when we talk about identity in a more general sense, it is establishing exactly who all those core things that the enterprise is dealing with are. These could be customers, but they are also suppliers, they could be vendors, they could be products, they could be B2B accounts, these could be locations. And establishing that single source of truth is essentially identity resolution.

# The difference between entity resolution and identity resolution

7:14

Alexey

**Is there any difference between entity resolution and identity resolution?**

7:20

Sonal

Identity resolution is used more in terms of the customer. Technically, they're the same concepts. But when we talk about entity – entity is a broader term, and it can refer to you know, any kind of noun. It can even be employees, it could be addresses or locations, it could be events. But when we talk about the customer, per se, or a person, per se, I think – like a citizen or healthcare provider – that is where people tend to use the term identity resolution more.

# How duplicate detection relates to entity resolution

7:52

Alexey

**What about duplicate detection? How is the problem of detecting duplicates related to this problem of entity resolution?**

8:02

Sonal

Duplicate detection is a sub-part. I would say it's how you consume the results. Let's say we have five records with variations in the customer name, address, first name, last name, date of birth and other details. We say that they belong to the same individual. So that is dissolving and saying identity. But now, what do we do with this result? Do we create one single record? Do we remove or purge out the other ones? That would be deduplication.

But when we say that, “No, we want all these records to be there. We want them to complete the story for us. We want to build a customer 360 – or a supplier 360.” That is where we use the term entity resolution or identity resolution. So I would say, in technical terms, probably the treatment is the same, but in terms of the consumption and the application, deduplication is actually an application of identity or entity resolution.

# How Sonal decided to start working on Zingg

9:08

Alexey

**Interesting. So the reason I'm asking about duplicate detection is because when I first got to know this problem, this was the name of this problem – duplicate detection. It was a competition on Kaggle from one of their online classifieds websites called Avito. The problem they had was – if you want to sell your phone, you go to an online classifieds website like OLX and then you just take a picture of your phone, put some title and then sell it on the platform.**

**If you *really* want to sell your phone and maybe you're not getting a lot of replies, what you can do is upload it multiple times, therefore creating duplicates. The item – the phone – is the same one, but you have multiple listings on the platform. They wanted to fight this problem with machine learning, so they created a competition. In this competition, the task was – given a pair of listings, you needed to detect if this pair is a duplicate or not. I took part in this problem.**

**Actually, this problem haunts me even to this day, because I took part in that competition, then another company contacted me because they had a similar problem. Then at OLX, I also needed to build a system like that. I actually didn't think of this as identity resolution or entity resolution – for me, it was always duplicate detection. But I never thought of taking this knowledge, or expertise, that had built over time by solving these problems, and somehow extracting this and putting this into a product, which is basically what you did.**

**I'm just wondering how it happened with you that you realized, “Okay, this is something big. I’m working on the same problem over and over again. I need to take all this knowledge that I have and put it into an open source tool, and then start a company based on working on this tool.” How did it happen to you?**

11:09

Sonal

I would say that part of the journey was definitely planned in terms of me choosing to work on the problem. This problem hit me as part of a consulting project that we were doing, where we were doing a data lake, and we had customer data coming from three different databases. We had to answer “What is the lifetime value? What is the likelihood of churn of a particular customer?” But for that, we had to have had the solid identity piece built in. So that was the first time I encountered it. And then very soon, I hit this problem again in a completely different scenario, which was enrichment of data coming from an external source and feeding your internal customer data with external CrunchBase Data. It was the same flavor of the problem.

I saw various use cases actually happening. That was the reason that I felt confident that if we solve it in a way that is generic, we will be able to attack a lot more use cases, with duplicate detection being one one of them. Zingg is now applied on products, on supply 360, customer, 360 – and all of them leading to different other kinds of use cases, like grants, like donors, patients. I think I was just lucky to see it in different scenarios for me to say that this is a problem worth solving in a way that is generic. It was honestly tough to solve it in a generic way, but luckily, we kept working at it. I think it was just persistence and a lot of hard work [chuckles] I would say. That’s what got me here.

12:58

Alexey

**Interesting, yeah. In my case, actually, all these three times that I worked on this, the solutions were quite similar. They were all in the classifieds domain anyways, I guess that's why. But for you – you described a pretty different use cases, but still the solution was the same. While you were talking about this, I remembered another term called “entity matching”. Is that similar to entity resolution as well?**

13:25

Sonal

Yeah, it is. It is similar to entity resolution. I think these are broad terms. I would say that entity resolution itself has so many duplicates – you call it record linkage, entity matching, in some cases, I think… [cross-talk]

13:37

Alexey

**Record linkage, yeah. Indeed.**

13:38

Sonal

Record linkage is there. I think there are at least 5 or 10 different terms that we use to talk about it. There's entity disambiguation, which is more in terms of NLP. Entity matching, I think is more in terms of matching unstructured to structured. But yes, I think they're all flavors of the same problem. Eventually, I think we'd like to solve many more of these kinds of problems.

# How Zingg works

14:02

Alexey

**So we talked about the problem, more or less. We have data coming from different sources and we want to reconsolidate or join it. Or we have duplicates because our users generate duplicates, so we want to detect them. There are other use cases that you mentioned, like patient/donor matching and things like this. So this is the problem. But how do we actually solve it? Is there a framework that all these problems follow that you also implemented in Zingg? How does it work?**

14:35

Sonal

Because I was fortunate to see this problem happening in different scenarios and with different entities, I wanted to create a system that will be able to actually just absolutely work with any kind of data. We didn't want to have a system that would just work on one specific thing. I think, if you solve for a person that in itself is a big enough market or usage. But if you solve it for even more entities, I think it becomes more powerful. That was the design goal that we set out with.

The second challenge – the second design goal was scale. One was being able to handle different kinds of entities, the second is really, “How do you scale this problem?” Which I think, at the heart of it, is one of the toughest challenges that entity resolution suffers through. The reason is that, if you don't know what to compare, then you have to compare every record with every other record. And that completely blows up. If you have 10,000 records, you're comparing 10,000 against 9999 – every single record against every other record. The moment you increase the size of your data by 10 times, the number of comparisons is going to go up 100 times. At a few million records, it absolutely blows up. So that's one of the fundamental challenges with identity.

These were the goals in terms of solving this problem. Machine learning kind of became an automatic way to do that, because if we train on the data that the user gives, we can actually get it to run on absolutely any kind of entity. ML, although is not really associated with scale, but in Zingg’s case, we learn how to distribute and how to do very smart indexing or “blocking,” so that the comparison of every record with every other record doesn't happen. Let's say you have 10,000 records – based on the training data that is provided and that Zingg helps you create (the training data) Zingg would really break those 10,000 records into maybe buckets of 100 each or 150, or larger or smaller sizes, based on a combination of fields. That is very powerful, because then you're not doing all those comparisons. It can be very fast. What we're seeing is that, when we released it, we tested with 15 million records (that was the maximum) and our users were able to run it at 80 million records, the last I heard, without absolutely any help from us. This is something I'm very proud of. It's scaling very well.

All of this is actually completely baked into the product. If you download Zingg, it’s very simple – just configure what fields you want the matching to run on. It can be any entity. There’s no need to define any rules or algorithms – you just need to understand what should be a “match” in your case. What's your business rule for a match? Don't bother about scale, don't bother about rule definition, don't bother “If A matches with B, and B matches with C – A and C should actually also match.” Internally, all that is completely baked inside the tool – that's the open source Zingg for you.

# What Zingg runs on

18:13

Alexey

**Can you maybe talk a bit about the implementation details? I know that the last time we spoke, you showed a Command Line Interface application and then internally, it was using Spark for computing all these things. Has that changed? What do you use to actually run it?**

18:35

Sonal

Yeah, a lot has changed since last year. I'll get there, I think, at the end of my answer. Fundamentally, at the heart of it, Zingg is a machine learning-based identity resolution tool. To do machine learning models, we need training data, and users will not have training data sitting around in their offices or their laptops. The command line utility that I showed you last time was a way in which this training data also can be generated through Zingg.

We configure what fields we want Zingg to look at, in terms of matching, and Zingg shows a few pairs very selectively for the user to say whether they are matches or non-matches. Around 40-50 pairs are good enough to train a model running millions of records. So Zingg shows you some pairs, you label, Zingg goes back and refines its models, and you label a few more and in a few iterations, you get a fairly trained, accurate model that can scale.

Internally, we use a combination of inbuilt machine learning, classification, graph processing. We've been using Spark for distribution, but we're also building out a Snowflake-native implementation. Some of our users are users of Snowflake but have not been on Spark, so the compute will be pushed on Snowflake. We connect with absolutely everything that has a Spark connector, so BigQuery, RDBMS, flat files, flat files in Parquet, in Avro, in JSON, XML or text files – you name it. In terms of the interface, we're kind of building out the UI as well for data storage and other functions, but we've also released a Python interface so that people can, instead of doing a command line JSON interface, they can use Zingg as part of their data pipelines.

20:41

Alexey

**I guess the command line interface didn't appeal to everyone, right?**

20:46

It did appeal to everyone, but to a lot… I wouldn't say that it was just a lack of appeal. It's more about usability. I think the way we look at Zingg is, it's not always about building the tool in terms of just market capture, it's also about “What is the best way that we think that the user would like to access it?” As Snowflake customers right now running those Spark clusters, but is there a better, leaner architecture for them so that they are not worried about two separate infrastructures?

That's the prompt, I would say. Again, with Python, it was the same thing. And I think with Python, the power of Zingg really increases, because now we have the option of integrating with Databricks notebooks easily and with tools like DBT, where a lot of action is happening for us, honestly.

# Switching from consultancy to working on a new open source solution

21:51

Alexey

**So you were working as a consultant – you were running your own consultancy – and then you saw that many of your clients had this problem. You then realized, “Okay, now I just want to sit down and solve this problem.” So did you just take some time off from your main work and then just bought a lot of coffee and started coding all these Spark jobs. How did it happen?**

22:15

Sonal

[laughs] Lots and lots of sleepless nights, honestly. [chuckles] Yeah. Some bits and parts of Zingg had already been created as part of my consulting, but they were kind of custom. Yes, I stopped consulting. I said, “I can't do everything. I need to focus. And I definitely have to take the plunge.” So I absolutely shut myself down from all communication, hands down – coding, getting stuff done. And that's how the open source came out. [chuckles]

22:53

Alexey

**How long did it take you to actually implement the first proof of concept?**

23:00

Sonal

Zingg has been long in the making. It's taken me at least a year and more to build out what we released last year. Honestly, I spent a lot of time tuning – I think out of that one and a half year, I must have spent at least six months just tuning the algorithms. It was crazy. It was tough. It was even tough getting the test data to run Zingg and to test it out. But I wanted to call it a scalable product and I couldn't do that unless I tested it at scale. So yeah, it took a lot of time. I think it was well worth it.

23:51

Alexey

**So you were running a consultancy and at some point you stopped. It took approximately a year and a half to release the first open version of Zingg. Right? That's amazing.**

24:06

Sonal

Yeah, I think one and a half would be close.

24:09

Alexey

**Okay, so you weren't working with any clients during this time?**

24:12

Sonal

No. Absolutely not.

# Why Zingg is open source

24:14

Alexey

**Wow. That's amazing. Why did you decide to actually do this in open source? After spending a year and a half, instead of doing it closed source and proprietary, you decided to do everything in the open? Why did you make this decision?**

24:32

Sonal

One reason is that I had personally been consuming open source – my data consultancy was built around open source tools and I wanted a way to be able to give back. That was a driver for me. It was personally important for me. I also wanted to establish that community – that feeling of people being able to just use I – the joy I had had in using so many other products. I wanted to kind of give that back. But just beyond those personal reasons, it was also a business decision. I feel that Zingg, as a product or as a technology, is something that a lot of companies, large and small, need. Some flavors of something like Zingg – identity resolution – are, in some ways, not as powerful, but are baked into products like CDPs and MDMS. These are… [cross-talk]

25:24

Alexey

**What is that?**

25:25

Sonal

Master Data Management Systems (MDMS) and Customer Data Platforms (CDP). To some extent – not as powerful and as full-blown as what we are doing, and these tools are very expensive. They easily run into six figure plus into multi-million figure annual licenses. I feel that open source Zingg can really get to a lot more companies than a closed source version, which needs sales – which needs a different kind of distribution.

It also has helped us find a lot more use cases compared to what I could do by maybe knocking on people's doors, having them look at the product, and get them to use it. So I feel that in terms of adoption, in terms of business, in terms of market, in terms of discoverability of use cases, especially if you are a team that is not really based in the heart of where most of the tech is happening (like Silicon Valley) I think it's a far better decision for a company like ours.

26:36

Alexey

**Were you afraid that somebody would just take all your code, rename the repo and say, “Okay, it's not Zingg – it’s Pingg. This is our new product!”**

26:51

Sonal

Yeah. I was worried about that. I have to be honest. I was worried about so much of my hard work.

26:59

Alexey

**One and a half years!**

27:00

Sonal

Yeah, it was a long “labor of love” as I call it. I was afraid of the IP being free, to be honest. But at the same time, I was also very upbeat about the potential. You know, you can control something, but then when you open it up, you realize there's so much more to what it is. That thesis has worked in our field, where I think the kind of welcome we've got from the community, from the lead leaders, from the practitioners is far, far beyond what I had ever expected. I also think, in terms of IP – ours is an AGPL license, which is not a classic license that somebody can just bake into the product. You can use Zingg, you can do whatever.

If you're a company, you can use Zingg internally, and if you're a solution provider, you can use Zingg and give a solution around that. But you can't bake Zingg into an existing project. For that you need a different license, unless you open source everything. So that's one thing that I think is a protection layer that we have. Also, at the same time, I think there's so much knowledge, and so much complex code, honestly, I would love people to contribute even for pet projects [laugh]. Take it further. That would be great as well.

# Open source licensing

28:41

Alexey

**Speaking of licenses, for me, this is the most difficult part of open source. There are so many different licenses. The one you mentioned, GPL license – I know that there are Apache licenses, which are pretty permissive, right? Let's say I have some proprietary code base – I have a closed source solution – and then I can just take this Apache license project, and then start using it and make money. I think the MIT license is similar to that but GPL is different, right? With GPL, I cannot just take a project and start using it without open sourcing my entire code base. Is this right?**

29:22

Sonal

No, no. AGPL, the one that we use, is very permissive. It lets you use it internally, it lets you provide solutions to your customers like SI – a solutions company. I can very well build out a solution and my client can install it and they can use it. Only if I distribute Zingg as part of a product, it's only then that those permissions kick in, which isn't any of the users. It doesn't really matter to anybody in terms of usage.

I think Apache and MIT is permissive to the extent that you can even build out a SaaS using the open source and absolutely fork it and use it. With AGPL, providing a network service with something like Zingg as part of your product service is not possible. That's the only difference. But for an end user, it really doesn't matter. There's absolutely no difference. [cross-talk]

30:37

Alexey

**If a cloud provider decides to offer Zingg as a service, they will not be able to do it. I know that there's one provider who decided to offer ElasticSearch as a service and then they ended up renaming the whole thing and called it Open search. So something like that is not possible with Zingg, because the license does not permit doing that, right?**

31:00

Sonal

I think so, yes. That's my understanding of the essence. [chuckles]

31:05

Alexey

**Did you need to hire a lawyer to actually make this decision? How did you make this decision?**

31:10

Sonal

No, I just did my research. And I talked to a few other people who were already doing open source. Honestly, the license is not the biggest part of open source – I think it's the philosophy. The code is all out there. The IP is in the code – it's all out there. It's not just a matter of what classes we've written, it's the algorithms which are there in Zingg, which are valuable. I wouldn't worry that much about this, I would have worried more about that discovery or innovation being open source. But I think it's a cool new way to reach a lot more people and help a lot more people. So I think it's well worth it.

# Working on Zingg initially vs now

32:00

Alexey

**When I asked you how exactly you started Zingg, you said that you took a year and a half to release the first public version. But you also said “we,” so I’m just wondering – were you doing this alone, or was somebody working with you on the first version?**

32:18

Sonal

The first version, I had done it mostly by myself, and towards the end I honestly needed help. [laughs] So I hired a consultant to help me towards the target.

32:33

Alexey

**I guess, before that, your responsibilities were mostly coding, then finding the product market fit, also tuning – you said that you spent a lot of time tuning the algorithm. What do you do now? How is it different from what you were doing for a one year and a half, coming up with the initial version, and what you're doing now? What do you do?**

32:59

Sonal

One is definitely coding the product, adding a lot of new features, planning how we want to do a tighter integration with Databricks, how we really do their APIs – so it's not just Zingg alone, but Zingg in the ecosystem, which takes up a lot more of my time, compared to just Zingg alone (working on just the product and the feature). It’s also now about “How do we tie in with particular technologies and make a whole solution?” So that is one.

The second thing is – a lot of time goes towards learning about different users and their experiences, “What is their feedback? How are they using Zingg?” Being active with the community and helping people out, talking to people, evangelizing Zingg, writing content, and getting the word out on Zingg. There’s also hiring, which is something that I spent a lot of time on. I think I would say that, from a pure Dev role, it's more of a company building, CEO/CTO role, founder role, which is a generalist of various kinds of activities that need to be done. Even taxation or incorporation or funding.

34:26

Alexey

**So what's your title now? Are you a CEO or a CTO or what?**

34:34

Sonal

[laughs] I’m the do-whatever-it-takes person.

34:36

Alexey

**[chuckles] But what do you write on LinkedIn? Is it “founder”?**

34:39

Sonal

I say Founder, yeah. Some places I say CEO.

34:45

Alexey

**CEO sounds cooler, probably.**

34:37

Sonal

In some places, yes.

34:50

Alexey

**But I guess CEO, to me, implies that this is not a technical role. So a person that calls themselves CEO, they are not coding anymore. Which is not always the case, but I guess this is like a rule of thumb – it's usually correct. But you still code, right? You still create code.**

35:10

Sonal

Yeah, I write code. Yes, definitely.

# Zingg’s current and future team

35:14

Alexey

**How large is your team now?**

35:17

Sonal

We are four people right now. We are actively hiring. We also have some consultants who are helping us with content and some of the marketing stuff.

35:31

Alexey

**How did you decide on the first hire? It was you and then a freelancer for some time. But how did you decide who exactly you needed to hire as a first full-time employee of Zingg?**

35:44

Sonal

I evaluated exactly in what bucket my maximum time was being spent and whether that activity was worth a full time role, or whether it was a good enough role for somebody to be able to do and enjoy, and also whether it was something that I could hand over to somebody. I think those were pretty much the prerequisites. That's the way I look at it.

I look at my calendar, “This is how I'm spending most of my time. What is the best way to free up my cycles?” And then “What is the demand that is coming in from outside? Who are the best people who can do it?” Because, obviously, I'm not an expert at everything. So those are the two parameters by which I look at who we should hire next.

36:39

Alexey

**Who did you hire, eventually, as the first employee?**

36:44

Sonal

It was actually a developer.

36:47

Alexey

**A developer, okay. So you realized that you spent a lot of time developing after going through this exercise that you just described. Who else do you have on the team? Who are the other two people?**

36:58

Sonal

Right now, it's all development. We have one person who's product marketing and writing content. Again, we are looking at more developers, because at the heart of it, Zingg is a technical product. It needs a lot of engineering. So we are hiring for engineering.

# Sonal’s biggest current challenge

37:21

Alexey

**What is your biggest challenge right now?**

37:25

Sonal

What's my biggest challenge right now? I would say that my biggest challenge is definitely hiring. It's a time-consuming process in terms of finding the right fit. You have to be very conscious of the other person's career aspirations. Basically, you are kind of getting somebody to bet on you, right? I would say that's a responsibility in and of itself. At the same time, you also have to evaluate the skills and ensure that that person would be able to deliver in whatever team environment that we have created. So those are things that definitely are challenging that way.

38:14

Alexey

**Are you hiring remote employees? Are you fully remote or how do you work?**

38:18

Sonal

We are fully remote, but all of our hires are currently in India. In terms of Indian salaries, I think that's working out well for us. [chuckles] We do have some interesting people who've reached out from different geographies and I think as we grow, we'll probably hire internationally as well.

# Avoiding problems with entity/identity resolution through database design

38:43

Alexey

**I’ve realized that we have quite a few questions from the audience. So I'll start with the first one from Bes. “How can a team avoid dealing with entity/identity resolution challenges from the start? Is it proper database design? Would it be enough if we design our databases in the right way from the start to actually solve this challenge? Or is this not sufficient?”**

39:11

Sonal

Designing proper data governance or ensuring that you have the right way to capture the data at all points, and to reconcile it with existing data, is a good first step. That should be done, but that would not solve the problem. The problem will still persist, because your marketing team would use multiple tools, your sales team would use different tools, your procurement is going to use different tools.

As the company grows, it will not be as simplistic as just capturing data from one single form and updating it in your database. You're bound to have different channels of customer acquisition, of lead generation, of customer interaction service, support, billing ticketing – not just for customers, it’s the same for vendors, etc. So the problem, by and large, definitely at the entry level, we need to be conscious of it. But the problem would still happen.

40:27

Alexey

**It's just that you can control the extent of this problem, but you cannot completely avoid it. Right?**

40:32

Sonal

Yes, absolutely. Completely right.

# Identity resolution vs basic joins, data fusions, and fuzzy joins

40:36

Alexey

**How is identity resolution different from using basic joins and data fusions? I don't know what data fusion is – I’m assuming it’s some sort of fuzzy join or something like this?**

40:51

Sonal

Yeah, even I'm not sure what the definition is. [chuckles]

40:54

Alexey

**I know that there are certain tools. There's one from Microsoft called Integration Service, I think. What they have is, you can just visually drag and drop to design your data pipeline – your ETL. In this tool, you can do a usual join, but there is also a fuzzy join – or fuzzy lookup – where a record doesn't match exactly, but maybe it looks for typos, it accounts for other irregularities in your data. I assume that’s what it is. Maybe I'm wrong.**

41:29

Sonal

To answer that question and with whatever our limited understanding of what data fusion is – what I would say is that if the data is simple enough for you to be able to just join something like an identifier – like an email that you trust, which is consistent, or an SSID, or a passport, or whatever number that you have – and you know that it’s consistently populated across all your sources, I think that's a great approach. If it works for you, there's nothing like it. Unfortunately, in most of the cases that we see, the data is not like that. Real world data is not like that. Even when we put in our identifiers – we have multiple identifiers: a driving license, or SSN, or passport, and even for KYC scenarios, we would put in different IDs.

Real world data generally tends to have a lot more variations compared to what a simple email join or first name, last name join. If it works – if the data can be trusted – probably there is already a curation step and your data is like that, then those joins would definitely work. Otherwise, you have to think beyond – how do you define those rules? How do you manage the scale? How do you compare every record with every other record? What do you choose as the threshold? Let's say all the databases have a fuzzy kind of thing – even ElasticSearch has a fuzzy lookup. But then, really, how do you decide at what threshold do you want to consider that as a match? How many matches should you look at? Those are, again, questions that need to be thought about.

43:17

Alexey

**While you were speaking, I remember a funny case that happened to me recently. I signed up for a webinar using my private email. I attended the webinar, and then after the webinar, the company that was running the webinar contacted me on my work email saying, “Hey, thanks for attending the webinar!” I thought, “Okay, they must be doing something smart.” They somehow figured out “Okay, this person who signed up and this person who they know from somewhere – it’s the same person.” And then “It's better to use the work email for contacting this person.” I thought something shady was going on, but now I understand what might have happened there. Maybe they had a record of me – maybe I signed up for another webinar some time ago and then they just combined this – linked these two records.**

44:15

Sonal

Yeah, identity resolution is everywhere. I think the moment you see it, you just can't stop noticing it. [chuckles]

# Deterministic matching vs probabilistic machine learning

44:25

Alexey

**What was the uplift from switching from deterministic matching to probabilistic machine learning?**

44:31

Sonal

So deterministic matching is a set of rules, which say that “I'm sure that these are the ones that I care about. And these are the ones I'm sure about.” This is what it is, and then [cross-talk]

44:45

Alexey

**Like if social security matches then…**

44:48

Sonal

Yeah. But as I mentioned in my last answer, real world data is not like that – especially customer data – unless you're in a heavily regulated industry. Even companies that are in life sciences and healthcare companies, when they have to do the Sunshine Act, and they have to report how much spend they have done on healthcare providers, their sales reps have actually been putting records of purchases and spends on physicians. Even that data has to be reconciled.

The Sunshine Act is a massive year-end activity that happens in those companies. So it ends up being that kind of fun exercise. Deterministic matching, as I said, if it works – if the data supports it, there's no need for probabilistic. But in a lot of scenarios, probabilistic *is* what you need because there will be variations of the data.

# Identity and entity resolution applications for fraud detection

45:50

Alexey

**Another question that I'm reading from my memory, “What are the applications of identity and entity resolution in fraud detection?”**

46:08

Sonal

In fraud detection? Okay. What happens in fraud detection is that people will create different accounts with slightly different name and address information. They would give different identifiers for their KYC. And when you want to track the flow of money, you would be actually counting them as separate individuals. But having those identities resolved, actually gives a very clear picture of how those transactions are happening.

If you look at all the graph databases, that's one thing that they talked so much about, because they are primarily used in all these fraud detection scenarios. And the first thing that they talk about is entity resolution – you look at all the Neo4j or TigerGraph. So when you are establishing those networks, having the identities established on those notes is critical. That's where identity plays a central role.

47:13

Alexey

**I have another example – something I saw in my experience. I was talking about duplicate detection of listings in online classifieds. What sometimes can happen is – imagine that you're renting out a flat. Let's say you have a flat, an apartment, and you want to rent it out to somebody so they can move in and live there. What can happen is somebody can just take your listing – all the pictures, all the information – change it slightly, (for example, instead of Berlin, it could be some other city) then upload it to the website and then pretend that this is a genuine listing.**

**Then people would contact them, “I want to see the flat.” What can happen next is they can say, “Oh, sorry. I'm not in the city now. But we have a lot of people who also want to see the flat. We have a lot of people who want to rent the flat. If you give us an advance deposit of 100 euros, we will reserve this flat for you.” Actually, when I was looking for a flat, this exact thing happened to me, like 5 or 10 times. If you are able to understand that this listing is actually a duplicate of another listing, then you can see, “Okay, this is created from different accounts and the cities are different. Something is wrong here. Let's figure out what's happening there.” So that's another example.**

48:42

Sonal

Wow. That's a new one for me. But yes, I think it makes absolute sense. We've also seen this in the case of e-commerce companies. One of these ecommerce companies will say that they have a promotion on all of these phones, and you are allowed to sell it at a particular discount. Sellers would pose as different buyers themselves and buy in bulk, and then sell it in retail. We had worked with an e-commerce company to identify said sellers. [chuckles] So yes – various kinds of fraud. People are very creative about fraud. [laughs]

# Graph algorithms vs classic ML in entity resolution

49:23

Alexey

**I know that for fraud detection cases, graph machine learning is quite useful. [Sonal agrees] In your experience, do graph algorithms outperform classical machine learning models in entity resolution, or no?**

49:39

Sonal

We do use graph algorithms in our case. We do pairwise matching, and then we use graphs to detect the network of records that actually belong together. We use that combination, but if you actually look at the output of Zingg, it's actually a graph that you can consume. You can consume it as a table, but you can also consume it as a graph. We say that we are the fundamental building block of your fraud detection algorithm. So you take this graph, which is your identities resolved, lay over the transaction data, and then do your classical processing.

50:20

Alexey

**What type of data can Zingg use? For example, if there are no common fields present, how does the tool know that these entities are the same? How does it work under the hood in these cases?**

50:34

Sonal

Right now, we don't have a column to column – figuring out which columns actually match to each other. That's something that we definitely want to build in the longer run. For Zingg to work, you have to specify what column to map to another column. It could be spelled differently, like “F name” or “first name,” but there has to be some notion of fields that are common in some way for it to figure it out. What we see sometimes is that people, in some cases, will have three address columns in one dataset, and a full concatenated address in the other. And they would mostly concatenate the address in the first dataset, and then match it with the other. But that kind of mapping is something that we don't figure out right now. The user has to specify it. But it's not very complex to specify. It's a very simple config.

51:27

Alexey

**So there is a way in your config to say, “Okay, this field and this field are related. Go figure out if the data there is the same.”**

51:38

Sonal

Yes.

# Identity resolution success stories

51:39

Alexey

**Okay. Another interesting question is about some success stories of implementing identity resolution in products. Maybe I can start with fraud detection. We didn't use Zingg for that at OLX, but there is a nice article at OLX’s tech blog (tech.OLX.com) where we talk about fraudulent rings of people who've done fraud – different people. There's a good success story. We were able to identify that there is a cluster of people who are actually the same person, or the same entity, and just ban all of them. So that's one of the success stories, but I'm sure you have a lot more.**

52:23

Sonal

Yeah, I have a lot of very good success stories. I can go on and on about them, but I will pick one which is really a very public good story. This is not a usual enterprise data story, but it's an open data case study, where the North Carolina State has come up with open data on their campaigns. You know who was donating, how much, who the donors are, who the recipients are, and what the flow between recipients and donors is. This data has been captured historically and people donate through online channels as well. There has been a digitization of those records – the historical and the existing records.

But particularly, all the donors don't actually have identities. The recipients also don't have identities. The same recipient has been entered multiple times by multiple donors in different ways and it’s similar for the donors. There's a consulting company (a nonprofit) and the state, who have used Zingg to establish how much spend donors are really doing on recipients. There is a case study that has also been published. And what they have seen is what kind of clusters they have been able to get, and then do the spend analysis for every donor. It's very easy to figure out affiliations – that it is all open. It's a very nice way to educate the voter on what's really happening in their constituency. And it's something I'm super happy about.

# What Sonal would do differently given the chance to start over with Zingg

54:11

Alexey

**Interesting story. Thanks for sharing. Again, these are questions that we prepared – this is not from the audience. I'm pretty curious – if you had to do this over again – let's say you're now working at the consultancy company and you think you want to solve this problem. What would you do differently now?**

54:38

Sonal

I think one thing that I would do differently is be on the lookout for a co-founder the day I decided to start. [chuckles] It's honestly a lot of work to do all alone – to do the funding, to be part of all the conversations, to be all over the place. Even now I'm actually very open to having somebody on board in that capacity, or more people on the founding team that way. I think that is definitely something that I would have been more open to. I just kind of thought, “This is a problem that I need to solve. I’m a developer. Let me just get down to it.” And I forgot how much time I spent there. So I would do that differently.

But having said that, a co-founder match is something you have to be very, very comfortable with, so I'm not sure whether it would have happened. No regrets there, but yes, I would have been definitely more receptive and thoughtful about that. [chuckles] Secondly, I think I could have open sourced it a bit earlier than I did. I was too busy polishing things. I was too busy getting it to perfection. I could have done it a bit earlier. Again, it’s not a regret, but looking back, those are two things I could have done differently.

# Advice for those seeking to realize their own solution to a data problem

56:07

Alexey

**Indeed, spending half a year on tuning is impressive. [chuckles] But probably, you indeed could have done this earlier. However, the demo you did with DataTalks.Club was really amazing. I saw that you put a lot of effort there. It was polished – it was really, really good. Okay. So let's say I have a similar problem, or maybe not a problem, but an idea. Similar in the sense that I see that there is a problem and I want to make a product out of this – an open source product.**

**How would you recommend me to proceed? First thing you said was to find a co-founder – I guess that would be a recommendation? Is there anything else that I should do to actually check if I should lock myself in the room for a year and a half before showing something? Or is there something I should do before that?**

57:09

Sonal

I think you should experience the problem in different scenarios. If you are a data person and you see it multiple times happening as being a professional, I think that's a strong sign. You should definitely watch that sign. Figure out if there is some potential to that problem. Having more people interested in solving that would be a great step as well. You could talk to a few people and figure out if they would be interested in using it. When we ask for feedback, everybody says, “Hey, go for it! Why don't you do it?” And that's nice, but maybe having very pointed questions on “Would you really use it? Do you think you would pay for it? If you would pay for it, how much do you think you would pay for it?`` would be nice questions. I wouldn't say they should be absolute blockers, but they would prepare you to think ahead and say whether you really want to do it.

I think it's more about building that inner conviction of whether you are so interested in solving this problem. It's going to be a tough road. Building a product is a lot of hard work, unless it's a shortage thing that you do over the weekend, and then release it and it becomes a success overnight, which is generally not the case. You have to really say or feel that this is something that you want to solve that much – you need that inner conviction. And then you really have to ask yourself how you would distribute it. Who are the people you think you would be able to easily reach out to? Are they in your network? Are they people who you can count on for trying out your product? Where would your early users be?

Even when you do open source, there are various distribution places, right? You can go to Slack communities, Discord, Twitter. Would you do content? What kind of content? I think distribution also has to go hand-in-hand with building the product. But absolutely, build a smallish thing, test it out – you will always learn. I see no harm in building things. It's absolutely the way to go.

# Reading suggestion from Sonal

59:26

Alexey

**And it's fun too. [Sonal agrees] Last question. One of the listeners, Joanna, suggested that we should ask every guest some recommendation – like a book recommendation, for example. I was wondering if you could recommend any book or some other resource to the listeners?**

59:45

Sonal

Absolutely. We completed one year at Zingg and I treated myself to a book that I've been meaning to read for a long time. It's called Creative Selection and it talks about Apple's design process during the time of Steve Jobs, especially while building the iPhone. And I absolutely loved the book for the way such a large company, at that time, was still operating as a startup – doing a lot of iterative development, very lean processes, but very strong focus on outcomes, on polishing the product, on focus, on usability and quality. We know about this because we use Apple products, but really what goes inside it was, again, something that I really enjoyed reading. So I hope people will really enjoy reading too.

# Conclusion

60:38

Alexey

**I haven't heard about this book. I have a few credits on Audible – the thing I use for listening to audiobooks – so I’ll look it up. Okay. Thanks for joining us today. Thanks for sharing your experience and expertise with us. You had an interesting journey and thanks for sharing it with us. And thanks, everyone, for joining us and for asking your questions. Enjoy the rest of your day.**

61:05

Sonal

Thank you, Alexey, for having me. Again, it’s always a pleasure interacting with you. We'll, again, probably catch up offline as well as maybe do some channel things like this. And thank you all for your very nice questions. Feel free to DM me or message me if there's something I can help you with. Thank you.

61:25

Alexey

**Maybe we should do another demo because it's been a year already or something like that, right?**

61:29

Sonal

Yeah. Yeah, we can do that.

61:32

Alexey

**Okay. Talk to you soon. Goodbye.**

61:34

Sonal

Thank you. Bye bye.