EPISODE 1351

[INTRODUCTION]

[00:00:00] KP: Companies that gather data about their users have an ethical obligation and legal responsibility to protect the personally identifiable information in their dataset. Ideally, developers working on a software application wouldn't have access to production data. Yet without high-quality example data, many technology groups stumble on avoidable problems. Organizations need a solution to protect privacy while simultaneously preserving aspects of the data which are important.

Tonic.ai is automating data synthesis to advance data privacy. Their solution gives you production production-like data for developers and analytical purposes without compromising on data quality or privacy. In this episode, I interview Tonic's CEO, Ian Coe; and head of engineering, Adam Kamor.

Adam and Ian, welcome to Software Engineering Daily.

[00:00:59] IC: Hey, great to be here.

[00:01:00] AK: Likewise.

[00:01:02] KP: Can you introduce listeners to Tonic.ai? What is it?

[00:01:06] IC: Sure. So Tonic is a data transformation company that leverages synthetic data differential privacy and distributed computing. We do that to de-identify sensitive data while preserving all the value of that data so developers can use it for building and testing software. By giving developers production-like data, their teams offer fewer defects. They ship faster all while protecting their customer privacy and having security. Be a lot happier with their practices.

To put a finer point on it, when we do sort of case studies with customers, we see sort of astounding stats like they ship three to five times faster and with huge reductions in defects

just by sort of moving a lot of their testing to higher quality data sets and making their CI/CD pipelines more dynamic.

[00:01:56] KP: That's a big speed up. Where does it really come from in the process?

[00:02:00] AK: The speed up comes from a few places. Primarily, it's really just like the shift left methodology of trying to find bugs earlier in the process. And the reason Tonic helps with that is because engineering teams today have a problem, and the problem is that they have production data that they can't use in their lower test environments, in their development and in their staging environments. So what they end up doing is creating some poor mimicry or poor replication of their production database. It'll have a small handful of rows, whereas the real database can have millions and billions of rows. And it has nowhere near like the plethora of options and complexity that really is in that production database. So they have this basically bad development or staging database and then they have to go develop and test against it.

But because that database isn't really a good imitation of what's out in the real world, they miss a lot of issues that they would otherwise have found if they were testing directly against production. So Tonic gives you all these speed ups, because it's giving you an actual production database to use for development and testing. But that production database is devoid of any sensitive information.

What tonic is really doing is taking that production database and making a copy of it. The copy of it looks identical to production, but all of the columns that contain sensitive information, or PII, or PHI, depending on your industry, have been de-identified and replaced with new data that looks and feels real, is statistically accurate, but it's completely fake.

[00:03:28] KP: Very neat. So I'm in my mind picturing something like an e-commerce platform. I obviously don't want the developers having access to the actual purchases, or their credit card numbers, or the people, but it might be nice if the volume of orders coming from cities was a pretty good match for what's in production. Do I get things like that out of box?

[00:03:49] AK: Yes, you get that out of the box. And you get it out of box in a few ways. First off, the first thing Tonic's going to do is it's going to scan your database and identify where the sensitive information is and give you an automated way of de-identifying that data. Replacing it with something that looks and feels real, but it's fake. And for the things that don't come out of the box in the sense that they don't immediately work in some automated way, the tool is highly configurable and very interactive. And really you're only limited by your imagination in terms of the types of transformations you need to apply to your data.

[00:04:21] IC: Speaking of sort of large e-commerce platforms, a customer that we've been partnering with for a really long time is eBay. And I can't go into like all the specifics of their requirements, but certainly to create data that helps them develop their application, we have to maintain a lot of complex important relationships within the data we're producing.

[00:04:42] KP: It seems to me you have a challenge in that scanning process you described, going in looking at a client's database, exploring their schema and identifying the PII that they've captured, because developers are free to write crazy schema structures and weird tables and weird column names. What's the process for that discovery?

[00:05:02] AK: Sure. So first let me say that Tonic is deployed on-prem. So I mean, as you know, as an input to the tonic process, you need to connect to a database containing real data. And as a result, customers typically will deploy Tonic on-prem. It's a really simple installation. You're given basically a Helm chart or a Docker compose file and access to our private Docker repository. And installations are typically done in maybe 10-15 minutes.

And in terms of being able to understand or introspect any given database, you're right, it really is a big challenge. Most software engineers are comfortable writing applications that connect to their application database typically through some type of ORM. Tonic, though, we have to basically make no assumptions about the database, and we rely very heavily on introspection queries to understand the columns, the data types and the overall structure of the database. So for the developers out there that are familiar with, for example, the

information schema in MySQL, or maybe PG catalog in Postgres, or the sys tables in SQL Server, we make very heavy use querying against those types of tables in those schemas.

[00:06:06] KP: Let's talk about some of the integrations you connect to. There's a wide forest of options out there for where I might persist my data as a developer. How many of them can you integrate with?

[00:06:16] AK: So this is changing all the time. At this point, I would say most. And to give you a less glyph answer, we kind of lump databases into a few different categories, right? There're the application databases, your Postgres, your SQL Server, your Oracle, your DB2. These are the databases that are typically backing an application. Oh, and I'll include Mongo in there as well. Mongo is actually our latest no-SQL database that we're supporting. And we support all these flavors, whether you're hosting them on bare metal in your own data center, or if you're using some type of managed service like Amazon RDS, or Google Sql, or anything similar really.

And in addition to those types of databases we also support data warehouses. So when I say like a cloud data warehouse, I'm thinking of a Redshift, or a BigQuery, or a Snowflake, or a Databricks, or just flat files sitting on a file system like Amazon S3 being operated on by some type of Spark cluster, maybe like Amazon EMR, which is Amazon's managed Spark service. So yeah, that's what we support today. And I think in the last few months actually we've brought online Snowflake, Redshift and Mongo, which has been really exciting for our customers.

[00:07:22] KP: So you've got all the major places I would think of. When you're on boarding with a new customer, is it typically a developer that's looking for this as a tool and a good way to protect production? Or is it more of a CIO, chief information officer, entry point? Who first brings Tonic into an organization?

[00:07:40] IC: So it's often someone in leadership who ultimately sort of gets the purchase done. I mean, that's for budgeting reasons and other things. But as you're alluding to, it's also somewhat aligned with kind of incentives. But we do see a lot of DevOps engineers, QA folks, other individual contributors who are tasked with potentially creating their own

companies sanitized data systems and they're looking for solutions and thinking if there's better ways to do it than spending their own time. So we often see folks like that reach out and then ultimately the purchase goes through leadership. And at the end of the day I do think leadership is often the impetus behind the effort, but we do see a lot of IC developers and other folks like that reaching out.

[00:08:24] AK: I think now is a great time to call it out and to kind of advertise something I'm really passionate about. Just recently, over the past couple weeks, we've actually rolled out our own hosted platform. So like I said earlier, most of our customers use Tonic on-prem. But we're actually now hosting it ourselves at app.tonic.ai. And anyone is welcome to come and create an account to give Tonic a try for themselves.

[00:08:46] KP: So what's the hosted experience like? Obviously that makes it easy to sign up, and you guys are probably doing the best in class deployment. What else would be the reason for me to choose this as a service?

[00:08:57] AK: So if you go with us as a service, you're getting a few benefits and a few downsides. I mean, the obvious benefit is you're not having to manage it yourself, and there's no need to have that initial install meeting because it's already installed for you on our servers. The downside is it can be twofold, but not necessarily.

The primary downside, and this isn't a downside for all of our customers, but when we're hosting it the data is being sent to us for processing and then sent back to be written to your output database. So depending on the customer, the industry, the rules and regulations for that industry, some customers aren't comfortable doing that, which is why we primarily do install on-prem. But for those customers that are comfortable using our hosted platform, we do have best-in-class security, and we can provide various services on top to ensure that their data is remaining protected while it's being processed by Tonic.

[00:09:48] KP: If I'm not using Tonic, I don't know if there's a standard alternative, but it seems like the alternative is no alternative that I just maybe replicate a production database to a staging area once in a while. What are some of the risks I expose myself to if that's my policy?

[00:10:03] AK: Sure. Interestingly, I'd say like it's changed a lot over the past few years since we started Tonic. That type of approach was more common a few years ago. And it's becoming less and less common today. I'm on customer calls. I'd say many, many customer calls a week, and very rarely do I hear folks taking that approach now. The more common alternative I'm seeing is customers that are attempting the de-identification and managing of that the identification themselves. And that can work for some customers, but it doesn't work for many. And I think it's why we've been successful.

And the reason it doesn't work is because it is very complicated to de-identify your own database. And it's really complicated for a few reasons. The first reason is that you need a database that once it's been de-identified is still useful. When you think of a database, it really represents the state of the users in your system or whatever the objects are in your system, right? And oftentimes you have foreign keys in your database and those kind of enshrine the relationships that have to exist between different rows. But there can be a lot of things not encoded in constraints that need to be true for the application to actually work when you connect it to the database.

So what ends up happening is you build some type of in-house system for managing this de-identified data, and over time this system grows in complexity and you end up basically reproducing all of the business logic of your application in your data de-identification system. And very quickly you realize how difficult that is to do and how untenable it is to manage. And that's what we see customers doing today. And we see them coming to Tonic, because tonic provides basically a solution that just works, and then they don't have to worry about it anymore and they can get back to their business.

[00:11:51] KP: If I wanted to generate some fake data that looked like my production data, what it means to look like it can be I can value different things in different organizations. Maybe as I said earlier, I have a desire to have it geographically similar. Someone else might care that it's age distributed across customers or something along these lines. What are some of the common needs and how do you service them in maintaining more than just a fake data, but in mimicry of the actual production?

[00:12:18] IC: So one thing I would say there is that Tonic actually allows a lot of different possible configurations. So we often refer to Tonic as sort of a model of models where you put together a lot of different things to get the data that you actually want for your use case. So we actually just launched a feature called Smart Linking that allows you to essentially have variational auto encoders figure out all the data, figure out all the relationships that you want in your data. You can also do a lot more manual configuration and specify a lot of relationships between columns, between rows, things like that, within the product. So it's something that we see as sort of a strength of the platform that it's not just one style of creating that data. It's many options. And that does mean that there's a little bit of human work, but we ultimately see that as kind of a huge win for folks trying to create data for different purposes.

[00:13:07] AK: That's absolutely right. And when it comes down to like the nitty-gritty of what type of transformations Tonic can do to your data to maintain the relationships you care about, I'd say the most important relationships you care about are the ones I talked about earlier. Those hidden constraints that exist in your data that the application assumes will be there and are essentially required to be there for the application to work. That's really the primary challenge.

Beyond that, Tonic provides a variety of different generators for working on all different types of data. You can very broadly kind of lump your data into two types, categorical and continuous. Categorical would be things like the makes and models of cars or the city and state you live in. And continuous would be things like your income, or perhaps your latitude and longitude even.

So Tonic provides a variety of basically like very statistical generators that are very good at preserving statistics on those types of columns. In addition to that, and this is a relatively new feature, we have something called Smart Linking, which instead of using what I'm just going to call not really like elementary statistics, but just standard statistics, Smart Linking actually makes use of machine learning models trained on your own data to generate new data that looks and feels real.

So it's actually you deploy Tonic on-prem, you connect it to your database. There's a given table that you want to really preserve the high statistical quality of that data. You point Tonic at it. Tonic will actually train a model on the rows in your actual database and then we'll generate new rows that go into your output database that are entirely synthetic, but statistically they are going to be very, very similar to the source rows.

The use case that we envision for this in the use case that we're actually seeing happen is actually a bit different. So we've primarily talked about application development, but there're other people out there that need Tonic. They need to de-identify data but still make use of it, right?

One group of people are data analysts and data scientists. So data scientists need de-identified data that they can train. Or rather they need the trained models and they only have access to data that is de-identified, or rather they can't have access to their production data. So Smart Linking is actually very successful at training models on de-identified data. And the output of those models is statistically the same as if the models were trained against the real data itself.

[00:15:31] KP: That's very interesting that that model holds up that way. I wouldn't take that for granted, but it's promising that the methods hold up across an anonymization process.

[00:15:40] AK: It's certainly something you can't take for granted. You're absolutely right. And it's something we're continuously improving. I think the real trick to it is actually training the models against the data in the table as opposed to just having one base model that is shipped within Tonic, if that makes sense.

[00:15:56] KP: Are there things that people are typically customizing and massaging over time about this? Do I have to fine-tune? Or is it kind of a one-time setup?

[00:16:04] AK: So it's primarily a one-time setup. But of course your database is going to change over time. Tables are going to be dropped. Columns are going to be added. New PII, or PHI, or sensitive data is going to be introduced. And you need to update your Tonic configuration as those changes are occurring.

For example, let's say someone adds a new table that contains social security numbers, right? What tonic is going to do in that situation is it's going to actually give you an alert in the UI letting you know that a new column containing sensitive information has been added. And optionally, Tonic will prevent generations from happening until someone, a human, has acknowledged that this new sensitive column has been added, because at the end of the day we have to always prevent sensitive data that's not been de-identified from entering a non-production environment.

[00:16:52] KP: In that regard, it seems to me that Tonic must be a tool used in part to achieve compliance and regulation needs. Can you speak to the way that it's been deployed in that regard?

[00:17:03] IC: Absolutely. We see a lot of customers using us to help with GDPR compliance. And sort of two big ways that they're doing it, one is in the event of a data breach, GDPR basically says, “If you make your data substantially resistant to reverse engineering, you're not subject to fines.” So if you think about sort of a data minimalization effort, if you reduce the footprint of your data and just have it on fewer laptops, fewer lower environments, your risk of breaching sensitive information, sensitive customer information that would lead to GDPR fines, or CCPA fines, it's just much, much lower.

The other thing that we started to see is that a lot of folks are de-identifying data so they can actually maintain it and not be in violation of GDPR and other regulations like that. And it's something that GDPR sort of continue being interpreted and the best practices are kind of consistently being revised, but something that we're seeing more and more of are folks who are getting uncomfortable with maintaining 10 years of customer data with sensitive information in it. And so Tonic can help you maintain just one year and then allow your data science team to still leverage the value of the last, the previous nine years or your marketing or sales team without any of that sensitive information that sort of creates some complexity around GDPR compliance as you're sort of thinking about your plan.

[00:18:23] AK: That's right. We're especially seeing that use case that Ian just mentioned come up with our customers using our cloud data warehouses like Snowflake, Databricks,

Redshift, etc. The ultimate goal is to be able to use all those years of data but in a safe way without breaking your analytics. And Tonic can really help with that.

[00:18:43] KP: What are some of the major industries who have adopted tonic?

[00:18:46] IC: Yeah. So when we first started we thought we're going to be selling to healthcare, we're going to be selling to financial services, these highly regulated industries. And we certainly see a lot of folks in those industries reaching out to work with Tonic. But it's not limited to that.

I think you know as we think about who should actually be using Tonic, essentially it's anyone building a data-driven product that would consider doing a SOC 2, because those are the companies that need to protect their customer data and they need to develop off of reasonably complex data sets to get their work done. And so obviously we have folks like eBay and dozens of other consumer-focused companies who've reached out, a lot of folks in ed tech. So we see it as really pretty ubiquitous across industries with maybe a little bit more pressure to adopt at an early stage in kind of financial services and healthcare.

[00:19:37] KP: Thinking about it from the point of view of the software engineer who now has the benefit of a more proper staging and development environment and data that's more reflective, I can think of just off the top my head a couple of simple ways that might illuminate a bug early. For example, like pagination. If I just go in as a developer, make 10 fake records and I end up on one page, it might be hiding the fact that in the real environment where there're lots and lots of data there's something about pagination that doesn't work. Do you guys see any common repeating patterns or things like that that Tonic will help really solve and take care of in people's development process?

[00:20:14] AK: I kind of break or rather lump the two types of bugs that our customers find with more accurate production data into two categories, and these are super broad strokes I'm about to paint. So let me just say that. The first category is that that scale and load problem that you just kind of outlined, right? I mean, yeah, the SQL query runs really fast against this table when there're five rows. What happens when there're five million rows like there are in production? Does it actually still run fast? Like, "Oh, are those joins

working now?" Well, of course. There're only ten rows in each table. But when there're a million rows in each table and there's no index on this column, like, yeah, they're going to be a lot slower, right? So like there's the scale problems.

And then the other side of the coin is I guess what I'll call like relationship problems. Think of any like moderately complex application. There's typically like 50, 100 tables, maybe even more. The plethora of states that a user can find themselves in and how those states get expressed across all those 50, 100, or a thousand tables, or whatever they are, I mean, it really just like combinatorially explodes. And it's really just impossible to kind of represent all of that complexity when you're crafting your database by hand like our customers did before they found Tonic.

But with Tonic, yeah, you're not getting every possible state, but you're getting all the states that existed in production at least. So that's pretty good. And those are kind of how I view the types of bugs that our customers find with Tonic.

[00:21:36] IC: The other thing that comes up sometimes as folks are sort of getting acquainted with this approach to solving this problem, sometimes people say, "Well, why not just encrypt my data?" And I think you know this may be obvious to most of your audience, but one thing to obviously point out is encrypted data isn't particularly useful for, as you're saying, debugging UIs and things like that. So a lot of the common methodologies for securing data kind of fall apart when you're trying to use that data to test and develop software.

[00:22:02] KP: Makes sense. Let's get into data warehousing. When I think of an analytics professional, I guess I could build a case that they should be looking at real data, because we need true numbers for the business. Or if they're looking at very high level trends, those trends might be preserved and perhaps for security alone, maybe other reasons, it might make sense for BI teams and analytics people to be looking at fake data over real data. Where do you guys weigh in on that?

[00:22:30] AK: so that's a really tough question to answer without getting into like the real specifics of what the data is, what questions are being asked. And then, also, kind of, I mean, who's asking those questions?

What I will say is that a tool like Tonic is basically a control knob. And you turn it counterclockwise and the data becomes more private, more secure. And you turn it the other way, clockwise, and it has higher utility. So really, it's like a balancing act, right? Like on the one hand you can make the data utility go up, but the privacy goes down. Or you make the privacy go up and the utility goes down, right?

So for analytics use cases, just like for application database de-identification, you have to decide where that knob' is going to be, right? What are your threat models? What is your tolerance for risk? What is the worst case if people see this data that shouldn't have seen it? And Tonic gives you all the control in the world to kind of make that decision.

With that being said, analytics is only useful if the queries you run, or rather the results you get from those queries are truthful. So when you use Tonic against, for example, like a data warehouse that's being used by data analysts or by BI tools or things like this, you do of course only want to apply transformations on the data that are going to preserve the queries that you care about. And Tonic ships out of the box with a variety of transformations that can help you preserve the analytical queries that are typically of interest to analytics teams.

And in fact, we have a blog post coming out pretty soon. So folks, go to our website, tonic.ai. Subscribe to our blog and you'll get notified when it's released. And we go into this exact topic in a lot more detail, especially in a lot more technical detail actually.

[00:24:11] KP: Good resource. So I think I have a clear vision for how I would use Tonic at the transactional level where I'm building my application. What's the story for Tonic as we go up that stack and look at data warehousing and things like Snowflake?

[00:24:25] AK: I mean, Snowflake, along with Redshift and BigQuery, are all data warehouses we support. Typically we find our customers using these databases for

analytics. Sometimes also the databases though are backing their application. And Tonic works the same as it does on any of the other databases that we support. And really like the different use cases that exist for Tonic, data analytics, application testing, machine learning model training, etc., it really just comes down. It's all the same experience for solving those problems. It all just comes down to the specific transformations that you're applying on your data.

[00:25:00] IC: Yeah. I mean, the other thing that we're seeing a lot of customers do in sort of the cloud data warehousing area is essentially setting up different environments for different teams and use cases. So imagine you have three environments. One is sort of for DB admins only. The next environment is for the data science team, who needs a little bit higher fidelity to be able to get their job. And then maybe the next environment is available to the entire company so that sales and marketing can do aggregate statistics and other things that they need to get their job done.

[00:25:35] KP: Faking user data, protecting PII is obviously a great use case for you guys. Is there anything else you find you're mocking for customers, reason to mock their inventory data or things like that as well?

[00:25:48] AK: I haven't seen inventory data come up at least in terms of if I have a database of just data inventory. I don't know if we have customers doing that exactly. One other interesting use case that I'm particularly fond of, and I know some of our customers are as well, is actually generating fake data for sales demos. So it would be pretty embarrassing if during a sales demo a customer were to see, for example, data pertaining to one of their competitors, for example, or something like that, right? Like that would kind of show a lack of respect from the person doing the demo.
So some of our customers will actually generate fake data that they can use for demos.

[00:26:23] IC: I will add that one of our customers that we have been partners with for a long time, Flexport, is in sort of shipping logistics. And I suspect they actually have a fair amount of inventory data and that they're using Tonic to allow developers to build on top of that. But yeah, I guess to Adam's point, I don't know exactly what that data is, which is sort of by design.

[00:26:45] AK: That's a good point, Ian. And to Ian's point about it being by design, I mean, recall, Tonic is shipped on-prem. By design and very intentionally, we don't you know really have any insight into how the tool is being used unless we're having a conversation with our customer and they tell us. And this is very much to preserve our customers' privacy.

[00:27:05] KP: It seems like a standard rollout would involve making a duplication of the data set, one per each environment if I go in the multiple environment route we just described. Do you see anyone having to make careful tradeoff decisions about cost of storage versus the amount of volume that they're going to fake?

[00:27:22] AK: That certainly can be an issue, but it is often an issue whether or not you're using Tonic, right? Storage and the storage costs are something that most engineering teams have to take into account. I will say, and this is great, because I would love to call this feature out, we actually have a feature to deal with this exact situation. So imagine your production database has a terabyte, or five terabytes, or 10 terabytes of data in it, right? That is a tremendous amount of data. If you want to give each of your developers their own local version of the production database that's been de-identified, that's really not tenable at that data scale.

I mean, maybe you could do it. It would be extremely cost prohibitive, and it would just be a very like – That's a pretty heavy approach, right? A heavy-handed approach. But Tonic ships with a feature called sub-setting. Sub-setting takes a database and it generates a new database that is structurally schematically the same. It has the same foreign key constraints and references, but it's smaller, but it's smaller in a way such that like statistically it's going to be a very good representation of the original database. And like I said, all of the foreign keys remain intact. And it's really, really common for our customers to heavily subset their very large databases so that they can basically give every developer their own local copy of a Tonic database to develop and test against.

And in fact what we're starting to see is customers actually taking these databases that have been reduced in size, shoving them into Docker containers, and making it super easy

for anyone in the organization to very quickly stand up a development database that's been de-identified so they can just develop or test against really any version of the product.

[00:29:05] KP: Very interesting. Is that a common thing you see most customers doing? Do you envision that being the standard use case? Or is that simply what works for most organizations?

[00:29:15] AK: Are you referring to the sub-setting or the Docker container approach?

[00:29:18] KP: The Docker container approach.

[00:29:20] AK: I would say that most of our customers that are using sub-setting are likely not doing that right now. I don't know what subset of our customers are doing it. It's certainly more than a handful. And it's actually something that I suspect will become like a more popular like pattern that you see in Tonic as we add additional like first-class support to that workflow into the product itself.

[00:29:43] KP: If I wanted to try and tackle this problem without you guys and I had let's say a MySQL base, I might think of doing something clever like going to my users table and sampling all the user IDs modulo 100 or something like this, so 1-100th the size. But as you point out, I'm going to lose, or I'm likely to lose a lot of foreign key relationships that way because the things that relate to that user may not also be aligned at conveniently that right place. For a developer that has those concerns, can you convince me that Tonic is going to take care of that for me?

[00:30:15] AK: Yes. Well, I mean the first thing I'll say is Tonic takes care of that for many, many customers every day and has been doing so for several years now. If you're interested in how sub-setting works, and it really is interesting, I'd encourage all of your listeners to go to our blog. And I think maybe, God, it's probably been two and a half years, maybe two years, but something like that, we actually wrote several blog articles that go into the technical details about how sub-setting works. But the long and short of it is that you pick a table or multiple tables that you want to specifically target.

For example, you said the users table. So you say, “Okay. Give me only the users in the city of Atlanta, or give me every user whose ID modulo 10 equals zero, or something like that, right? Like you somehow select the users you care about. And then Tonic is going to basically traverse the graph of all of the tables in your database. Think of the tables as the nodes and the foreign key constraints is the edges, and it's going to traverse that graph in a very special way so that it opportunistically only brings over rows and other tables that are required to maintain referential integrity. And it doesn't bring over essentially any superfluous rows, or at least by default it won't bring over any superfluous rows. You can actually configure it to bring over additional rows if needed.

[00:31:27] KP: What does growth look like for Tonic? Where do you want to expand in the future?

[00:31:31] IC: Obviously, we want to keep doing things that are most valuable to our customers. So we sort of see some customers already doing things with the product that we didn't tell them they could do, but they're kind of figuring out. So we're seeing customers get data into the cloud. And so obviously we're going to try to help them do that really in a first-class way. And there’s sort of what we consider sort of these upstream and downstream use cases. So customers are finding sensitive data with Tonic. That's not something that we set out to do initially when we were building the product, but it's something that became necessary and that we're going to continue to support.

So we see folks potentially using tonic as a bit more of kind of like a data catalog and understanding the provenance of their data and really where all their sensitive data is across their whole ecosystem. The other thing that we kind of see kind of downstream as we get a lot of data to developers, we're seeing data go for sales demos. We're seeing data go to the cloud. We're seeing data could be used for data partnerships. We're also seeing data go to data science teams. So there's just so many different things that you can do once you have this technology implemented into your ecosystem and you start expanding upon some of the initial capabilities of Tonic.

[00:32:42] KP: Yeah. I’d love to explore the data scientist angle a little bit further. It has a strong appeal, because if I were in that role, I wouldn't want to be given a lot of PII data. It

doesn't seem to me that my model should learn anything about an individual. That would be some form of over-fitting. So it seems natural that that would go hand-in-hand. Do you think that this will become a data science first service at some point? Or is it a rising tide for all ships?

[00:33:08] IC: So I see it more as a rising tide for all ships certainly. Data science I think is likely to be a key customer for us down the road. And I agree, most folks feel more comfortable if they're able to be productive in an environment where they're not exposed to sensitive information. Obviously, the complication for data science is that the fidelity of data that they need, it tends to be a bit higher than what you need for software development. I mean, they actually need kind of anomalies that may not even be obvious to a user of the data, but something that a detailed modeling effort would pick up. And so the data that we produce has to be able to support that and support a variety of really specific inquiries.

So we certainly see that as an important challenge for us and something that we want to address, but we also want to make sure that what we're shipping satisfies the needs of all our customers. And I think we're kind of at the beginning of – And I think synthetic data overall is kind of at the beginning of sort of its ability to serve that audience in a way that's really satisfying.

[00:34:10] KP: Yeah. I feel like in a perfect world I could train my model on the fake data. I could submit that model to be run in production, and it's almost like a perfect holdout set. I can then verify my model does or does not hold up on the real data. Do you have any advice for setting up an operational procedure around that? Surely, there should be a process for the day that comes when for some reason the model in production isn't performing as well. How should teams integrate the product and set up a good process?

[00:34:41] IC: If you're looking to sort of validate models, there are a lot of different technologies you can use to try to do that. And you could certainly you know build your model on top of synthetic data and then run a varieties or tests in production to validate that it's giving you the results that you want. I mean, obviously, at some point you're going to probably expose someone on your team to sensitive information doing it that way, but it may be far fewer employees than if you didn't have a synthetic data environment at all.

Obviously, there're other approaches that probably beyond the scope of what we're talking about here, federated learning, things like that that come into play. But I would say the very basic idea would be build an environment where everyone can work and then have a more limited test environment if you really need to do some specific model validation where you can have the model that's been built on a synthetic data run on top of production. Effectively, it's sort of in and out of sample, right?

[00:35:36] KP: What about lag? Is there any time I have to wait if recency of getting the access to the latest fake data is important to me?

[00:35:44] AK: Performance is something that's really you important for our customers, and it's also important for the engineers at Tonic. In fact, since this is a Software Engineering Daily, I spent the last week working on some performance improvements in MySQL. So when it comes to customers that are sensitive about how long it takes to generate a new set or a new database containing fake data, there're a few options available to them to kind of increase or improve performance. But let me say first that the bottlenecks that we typically encounter when de-identifying data are rarely actually on the Tonic side, in the sense that it's not the computation happening within Tonic that's the bottleneck. It is typically the read I/O and the disk I/O coming out of the source database and into Tonic, and then in particular, the network I/O and the disk I/O on the output database. And almost always it's actually the bottleneck is really writing the data to the output database. So we typically think of it as a write-constrained process.

So things that we can do to speed that up are, for one, deploy on beefier hardware. Two, subset the data when appropriate. Three, run tonic in – This is going to kind of get into the weeds a little bit, but you can run Tonic in different modes, and we call them table modes because they can be applied per table. Some table modes are going to re-identify the entire table every time you run the job. Other modes are going to de-identify the table data once. And then the next time you run the job we're only going to de-identify whatever has been the delta since the last time you ran Tonic essentially. So the first time there's this big, big bang, you copy over a terabyte of data. The next time you run, it's really only going to copy

over maybe 10, 20 gigabytes of data. So in that way you can really reduce subsequent job times.

[00:37:29] KP: What does it take to get started if a developer or a team wants to explore using Tonic in their project?

[00:37:34] AK: Well, they have two approaches. One, they can go to tonic.ai and they can book an appointment with one of our sales people. They'll go through a demo. They'll get to talk to someone on our engineering team where they can ask additional questions, and we can kind of ensure this is a good fit for them. And then they can get started with Tonic after beginning that conversation with one of our sales reps. The other option is to go, again, to tonic.ai and sign up for one of our new free accounts. They'll be given essentially unlimited access to Tonic for some set period of time and they can use that with any of our sample databases that they'll have access to. Or they can connect it to their own databases. It's really their choice.

[00:38:12] KP: Very neat. And what's adoption been like? Are you finding any particular trends or different sectors discovering the tool?

[00:38:19] IC: Yeah. Like I said, it's always been kind of surprising and kind of in a good way the number of folks that are interested in what we're doing. Like I said, when we first launched, we really thought this was a healthcare and financial services thing. And pretty soon we had ed tech customers. We had customers in all these different areas. Like I said, Flexport is one of our early customer partners. They're in shipping logistics. So we've just kind of seen, like I said, like anyone building a data-driven product, this is just a problem that you have to confront at some point in your organization. And I think depending on your industry, you're going to confront that at different points. Like if you're a healthcare company, you might be confronting that when you're 10 people. If you're just kind of an average b2b company, maybe you're confronting that when you're 200 people. Maybe you hire a CSO or go through a SOC 3 or sign a big enterprise contract with a data covenant in it and you have to start being a little bit more serious about your data practices. So we really see this as just very widespread. And we've always been kind of flattered and excited about the interest from a variety of different sectors.

[00:39:22] KP: Well, congratulations on the recent race. I'm eager to hear about how you intend to make use of that.

[00:39:28] IC: Thank you. Thank you. We're super excited. We're really happy to bring insight partners on to help us build the company. And we're obviously entering kind of a new phase. I think what the VCs call it now is scale up, which I think is the next phase of startup. And it really does feel that way for us in terms of sort of what we're focusing on now. It's much more about how do we do things as fast as possible for all our customers. There're just many fewer existential questions and just a lot more questions around how do we go faster. And to that end, I think what are we going to use the money for? It's going to be hiring really talented folks. So if you're interested in any of the problems that we're solving, we'd love to talk to you. We're hiring aggressively across multiple functional areas, certainly in engineering and data science. We're hiring in sales, marketing, you name it. So we'd love to connect.

[00:40:19] KP: Where's the best place for people to see about open positions?

[00:40:23] IC: Yeah. You can just go to our website. And I think right under the about section there should be a listing of quite a few jobs.

[00:40:31] KP: And before we go, remind listeners, what's the getting started story if they want to try out the product?

[00:40:37] IC: So if you want to try out the product, there's a link I think right directly on the home page. So go to tonic.ai, click free trial, and you can create an account and get started. There're tutorial videos. There're a bunch of other things that will help you kind of get acquainted to the platform. And don't be shy. We are happy to chat with you. A lot of our customer success folks spend a lot of time with customers doing a free trial. And it's not a high pressure sales thing. It's really just, "Hey, let's help you make sure that you learn what this product can do for you."

[00:41:12] KP: Well, Adam and Ian, thank you both for taking the time to come on Software Engineering Daily.

[00:41:17] AK: Thank you for having us.

[00:41:18] IC: Yeah, thank you. Really appreciate it.

[END]