EPISODE 1498

[INTRODUCTION]

[00:00:00] ANNOUNCER: 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 need 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 is automating data synthesis to advanced data privacy. Their solution gives your production-like data for development and analytical purpose without compromising on data quality or privacy.

In this episode, we interview Tonic's CEO, Ian Coe, and Head of Engineering, Adam Kamor. This episode is hosted by Sean Falconer. Sean has a PhD in computer science, was a postdoctoral student at Stanford's Medical School, and as an ex-Googler and startup founder, now serving as Head of Developer Relations at Skyflow, an architectural solution for data privacy. Sean has published works covering a wide range of topics from information visualization, quantum computing, developer experience to data privacy. You can find more of his work by following him on Twitter @seanfalconer.

[INTERVIEW]

[00:01:26] SF: Hi, Adam and Ian. Welcome to the show.

[00:01:28] IC: Hey, how's it going?

[00:01:29] AK: Glad to be here.

[00:01:30] SF: It's great. Thanks. I'm really excited to talk to you guys today about Tonic.ai. My day job is actually working in the data privacy space. I'm always excited to talk to people that

are doing innovative things in the space. And there's actually a lot of amazing work going on in privacy enhancing technology world right now. I know you've been on the show before. But for anyone listening that perhaps missed those episodes, can you start off by explaining what Tonic.ai is?

[00:01:56] IC: Absolutely. Well, you should obviously go back and listen to the other episodes. Tonic is a data transformation company leveraging synthetic data, differential privacy and distributed computing. We deidentify sensitive data while preserving all the value of that data. So, developers and data scientists can use it for building and testing software or developing models. By giving developers or data scientists production-like data, teams offer fewer defects. They ship faster. And the whole organization gets to a much better security posture.

And I think one of the unique things about Tonic is that since we started off with developers as our first set of customers, we have a lot of powerful integrations and a robust API allowing for a lot of convenient orchestration of your data pipeline.

[00:02:38] SF: I'm sure you get this all the time. But what exactly is a data or synthetic data for sort of the uninitiated?

[00:02:46] IC: We actually kind of have these arguments internally about this, too. There're all these terms that come up in this space, like masking, synthetic data. And there's not obvious distinctions between them. I think people have a lot of preconceived notions. So, people think of masking as something fairly trivial. Taking data and replacing it with a bunch of nulls, or putting a bunch of X's in place of the data.

But what happens when you're doing sort of more intelligent masking, like looking at the statistical properties of the underlying data and replacing with something that reflects those statistical properties? Is that masking? Or is that synthetic data?

Where we kind of come down is that synthetic data is something based on a model or a rule set that can generate sort of infinite scale, whereas masking tends to be more of a one-to-one relationship between the underlying data and the new data. Adam, I don't know if that sort of aligns with –

[00:03:43] AK: I like that definition a lot. I think data is deidentified if the deidentified row can be tied back to a source row. It's like, “Oh, okay. Yeah, this de identified row corresponds to row whose primary key ideas seven in the source database.”

Synthetic data, on the other hand, you can't tie a synthetic row back to any individual row in the original data source. But each of the columns in that synthetic row are generated from like, essentially, the aggregate properties of the respective column.

For example, let's say you have a column that is the – Maybe it's a user's table, and it has their job title, right? Well, okay, if you were to make a synthetic row, you might say, “Okay, well, look at all the possible job titles and pick one at random,” right? You can't tie that back to an individual row. But you can tie it back to the aggregate properties of the source column.

When you use that approach, you can actually generate data really at any scale you need, because you now have an algorithm for outputting a new row of data, and you can invoke that algorithm as much as you'd like. Whereas with deidentified data, typically, it's tied back to an individual row. So, you're limited by the scale of the database, which oftentimes is perfectly fine, right? Because if you're a developer trying to test your new feature on the staging environment, oftentimes, having a staging database that's on the same scale as production is more than sufficient. And it's really actually desirable, because you can accurately test how your feature is going to behave once it's in production.

[00:05:04] SF: Deidentification is really, as you mentioned, this like one-to-one mapping. If you had a thousand rows in your database, you're going to end up with a thousand rows, essentially, in your deidentified database. Whereas with synthetic data, you're essentially trying to figure out what is the pattern of the column, and then apply some sort of algorithm so that you can essentially generate infinite data from that. Is that right?

[00:05:27] AK: I that is a good way to say. And I think it summarizes what we said. Well, I'll give one caveat to all of this. When you deidentify, the upper bound is however many rows you

originally had. But Tonic actually has a really cool feature called database subsetting, which allows you to generate subsets of your data, and you can de identify on top of those subsets.

Now, the subset of data set will never have more rows than the original data set. And it'll, in general, have fewer rows. And the purpose of that, like, I'd say it's probably our most popular feature behind just like pure deidentification and synthesis, it allows our customers with really, really large production databases. Like, many, many terabytes in size, to kind of subset down to a more useful size for development and staging environments. And then deidentification rules would typically be applied on top of that. And that's a great way, for example, for our customers to like give each of the developers their own database, right? You couldn't do that if you had to give every developer in the organization a five-terabyte database. But if you subset it down to 10 gigs, then, hey, that fits in a Docker container that you can run locally. And that's really awesome.

[00:06:28] SF: You mentioned this idea of giving each engineer on a team basically their own database to QA against. And I'm probably dating myself here somewhat. Like, when I started out engineering almost 20 years ago, it was pretty common for engineers to essentially run tests and even QA things directly against the production database. And there was even one company I worked for, I won't name them, but you'd actually use a sequence of keystrokes on a live company website that was deployed with the software that would open up an admin panel where no login required, and you could actually edit the live database. Obviously, we've come a long way since then. But as we've been a specific industry trend that has led to the need for companies to essentially have fake data available for developers to run these types of tests.

[00:07:14] IC: Yeah. I mean, obviously, there's sort of what are the alternatives to using something like Tonic? One is, as you're saying, ignore it. One is sort of onerous access controls. Like, have a thousand developers, 10 have production access, and they have remote desktop in, or something annoying to get there. And then the other is kind of build something yourself, have DevOps build your own sort of Dev environment that's deidentified. There's sort of pluses and minuses to each of those. Obviously, one big downside of the like do nothing, just let everyone have access, is the security posture is very questionable.

I think things like GDPR, CCPA, also, more and more folks are going for SOC2's, all of that I think has led to folks paying more attention. I think there's just also – There's been some high profile incidents that have, I think, change people's perspective on whether it's worth it. And our hope is that Tonic actually makes it a lot easier to elect yourself into a sort of a data minimization bucket, where you don't necessarily give everyone all the data just so they can be productive. That they can be as productive as if they had the production access.

I think it's sort of shifting the mindset, regulatory pressure. And then also just some pretty embarrassing incidents. There was one where a large company, their support engineers were actually going in and looking at sort of significant others, and their activity, and figuring out certain things. It can get pretty uncomfortable, obviously, when everyone has production access, and there are reasons to just sort of remove sort of – Really, just protect your customers at a higher level.

[00:08:51] AK: That's right. We don't see customers using production data in staging and Dev environments as much anymore. The company has been around for over four years now. And that's certainly less and less common. We see it from time to time, and those are definitely companies whose privacy posture we can definitely improve and help them improve while not really worsening the developer experience, because the fake data we provide is such high quality.

There this other bucket of customers that have already done away with production data in staging and Dev environments. But then they're like, "Okay. Well, hey, what do we actually put in these environments?" And they end up having to hand craft a small set of rows to insert into a database so they can do their tests, or that they're trying to maintain this. It gets overly complicated. And typically, they ended up building a very poor imitation of production that takes a lot of hours to maintain. And for those companies, Tonic is less of a privacy company. It's more of a Dev tools or development efficiency company that actually like makes developers like significantly more productive in their day to day.

[00:09:48] SF: in terms of like your sort of customer breakdown or profile, are most of the customers coming to you with this developer need? Or are they coming with maybe like a

privacy need or perhaps like a need for a data scientist to be able to run machine learning models essentially over data that can train models over data that's not actually customer data?

[00:10:08] AK: Short answer, it's all of it. But I think Ian has a more nuanced answer.

[00:10:12] IC: Yeah, yeah, it's definitely all of that. I was actually going to also add that you weren't really dating yourself. I mean, as Adam said, even the lifecycle of our business, I think we've seen a pretty big shift in the mindset here. But I would say, probably, it's about 50-50 in sort of the privacy versus developer productivity. It kind of just depends on where they are in their journey. As Adam said, that there's folks that have gotten pretty far in their journey and may be really looking to be more efficient with their processes. There're other folks who are starting their journey. And they're investigating, like, “Do I do this myself? Do I buy something here?” And, obviously, our belief is there's a pretty compelling reason to buy something that we can save companies a lot of time and give them a much better result. But I think because this is a relatively new field, people are still doing a lot of investigation when they sort of start the journey.

[00:10:58] SF: Yeah. Is this something that you've had to educate some of the market on in terms of the value of fake data and why a company might need that or why it doesn't make sense for them to kind of like roll their own solution?

[00:11:10] IC: Yeah. Yeah. I mean, we get feedback all the time that sort of says, “Hey, I tried this before, and it was really hard. And how do you guys do this? Is this really going to work?” And so, we're used to that conversation. And I think we're glad people have a healthy degree of skepticism here. And that's why we let people try the software, things like that. We work with customers to make sure that they can feel really comfortable that this is going to be effective for them.

I think it is certainly something that people have thought about. And I think we're now sort of seeing people really execute on in earnest. But yeah, there is certainly – I think the category is still being built. But it is kind of night and day four years ago in terms of just the acceptance that, “Hey, we should do something here. And that let's figure out how, as opposed to are we doing something here?”

[00:11:59] AK: We still do talk to customers that are like, “I had no idea a tool like this existed, or that it was even possible.” There is education from that point of view. And if I had to guess where that comes from, I think most engineers that have been in the industry long enough have attempted a solution like this before. And like, typically, what happens, and it's not always the case, is on day one – And I'm kind of summarizing this a little bit. On day one, everything's going good. You're writing your Python script to generate some big data. And then day two comes around, like, “Oh, wait a second. If I do this for this table, well, hey, there's this foreign key. Let me go handle this thing over here.” And then this goes on for a while.

And by day five, or six, you're pulling your hair out, and you realize, “Wait a second, I'm essentially trying to put into this little Python script all of the insane complex business logic that's in the application and the assumptions that the application has on the structure of the data in the database.” And very quickly, it just becomes an intractable problem. And I think everyone's been there. I've personally been there before prior to coming up with Tonic. I think that's where some of the doubt comes from. But as Ian said, I think doubt and skepticism are good.

[00:13:04] SF: Yeah, I totally get that. I think there's a tendency a lot of times with engineering to kind of want to own a problem at the beginning and build your own solution. But I think one of the things I always say is don't DIY privacy. It’s just very complicated field that spins out of control very, very quickly. It's like a game of Whack a Mole, you put one solution in, and then it’s not going to fit every use case. So, then you need to do another thing, and another thing, and another thing, and it spirals out of control pretty quickly.

[00:13:29] IC: Yeah, totally. And unless you're a weirdo like us, generally, you might not want to spend too much time on data pipelines. I think that's another big advantage, is that we save people time and headache there.

[00:13:41] SF: Right. What kind of expertise the typical user of Tonic.ai need? What's the kind of user profile? Are these data scientists, application engineers, data engineers?

[00:13:51] IC: Just to get started, I would say you sort of need the expertise that you need to kind of run almost like a BI product. But then quickly, it really gets into more kind of a core dev

tool. As Adam was saying, there's sort of a large spectrum of users. But I think the folks that end up kind of running us on an ongoing basis are usually more sort of like DevOps, developers, QA automation folks.

[00:14:13] AK: It also depends a bit on the use case. If your goal is to deidentify an application database for like development and testing, then it's typically going to be like dev and QA involved initially. Maybe DevOps is also involved just to get things kind of like working in an automated fashion after the configuration is done.

Tonic also supports data scientists in deidentifying data for like machine learning, training and validation. And on that side, it's typically the data scientists that kind of owns the modeling that's using Tonic. It depends some on the use case. But typically, it's a technical person. And it's someone that kind of at least has an idea and understands the data and the shape of it.

[00:14:49] SF: I see. Can you may be starting to talk a little bit more in depth about the actual getting started process? Like, let's say I need a solution like this. Where do I go to begin? Can you kind of walk through that process of signing up with Tonic.ai and then integrating that into an existing system for maybe one particular use case?

[00:15:08] AK: Sure. I mean, I think, typically, it begins with going to the website, which is tonic.ai, for anyone interested. From there, you can sign up for a quick demo. The demo, it'll typically be one of our account executives on the call with possibly a solution architect. We can give a demo. We'll talk about use cases. We'll kind of understand what you need. And we'll have a discussion if we think that tool is right for you. If it is, then, typically, we would enter into a pilot of some kind. The pilot would typically last between two and four weeks. Essentially, it would give the potential customer a no risk opportunity to try out the tool on their own prem. Because I mean, Tonic is typically working with an operating on very sensitive data. And as such, we actually typically don't host Tonic for our customers. Our customers take tonic and install it on their own hardware and run it in their own network.

In fact, some of our customers even run Tonic completely air-gapped. The tool itself has no need to phone home and doesn't do so by default. It can run completely on their prem. They'll

try it out for some period of time. And then if everything's working well and they're happy, then they'll proceed to a more standard annual contract.

[00:16:14] SF: What's involved with the on-prem deployment model?

[00:16:17] AK: Sure. We package up Tonic as a set of Docker containers. And then our customers are given access to a private Docker repository. They can pull these containers and run them however they see fit. We typically advise our customers to run these containers like you run all of your other containers. If you already have a Kubernetes cluster that you use for vendor tooling, run it there. If you guys are really familiar with EKS or ECS, run it there. If you don't typically use containers, we'll provide you with a Docker compose YAML, and you can just deploy that on, for example, an EC2 instance, and we'll help you do it. We always help our customers install even though some of our customers, frankly, don't need our help, because they're already pretty good at managing their own containers. But I'd say, like, in general, it's maybe a 30-minute setup, and then you're rocking and rolling.

[00:17:02] SF: And then once things are deployed within the Docker container, am I going through a process of sort of connecting my existing data sources so that the Tonic.ai can actually analyze it to figure out how it's actually going to generate synthetic data based on the schema that I support?

[00:17:20] AK: Yeah, that's absolutely right. The product ships with its own UI. You'll typically do all of this, at least initially through our UI, even though everything can also be done through an API. But we typically see our customers start using the API kind of like after the fact more for like orchestration and automation.

But yeah, they'll use the UI. And the experience is very similar to any like SQL clients. Like, with a typical SQL IDE, the one I use is DataGrip from JetBrains, you select the database you want to connect to, and you provide the connection information. Typically, something akin to a connection string.

The Tonic UI is very similar. You point Tonic at the database. Tonic scans the data. Identifies what it considers to be sensitive and what's not sensitive. The user can go in and kind of

validate that and make adjustments. And then you go down and you apply the transformations that are needed on the columns that were deemed sensitive.

And at that point, you're ready to generate an output database. And Tonic will – Essentially, you point tonic at where you want the output data to go. And then Tonic will create an output database at that location that is identical to the source database in every way, structurally, schematically, even statistically. But the columns that were deemed sensitive are replaced with fake but realistic-looking data. And then you can use that in your staging environments and development environments, etc.

[00:18:29] SF: And if I have, say, like petabytes of data, and I want to reduce that into something that's like reasonable for my developers to test on, like, how long does it take to sort of go through that analysis process to actually generating the database that my developers are going to use?

[00:18:44] AK: Sure. It's super hard to come up with rules of thumb for how long the process takes. And the reason I say that is the actual job of moving and processing data, which is really similar to like an ETL job, right? That job is typically network and disk.io bound. And it depends a lot on the hardware you have both on the source and output databases, as well as your network connection, and then like the Tonic server itself to a lesser extent.

But we deal with two customers that I know of on the petabyte scale. And the tool is able to do what it needs to do in a timely fashion. But those data skills often do require some pretty careful thought. We more commonly see customers that are like sub-10 terabytes, I would say. That's where the lion's share of our customers sit.

It's still a lot of data. But I'm only saying that when you get to the petabyte scale, it requires special thought. And typically, like, there will be more than one conversation with the Tonic engineering team before we start the process, just because that's so unique.

[00:19:45] SF: One common problem that companies face a lot of times when it comes to data privacy is essentially the replication problem. So PII essentially ends up like everywhere, on

multiple databases, warehouse, lakes, log files, what have you. How does like your solution maintain consistency across these different stores?

For example, if I have my name, Sean Falconer, in multiple databases, how do you sort of consolidate the fake data that, generated, it's actually going to maintain consistency across those databases? Or is that something that you're not able to do?

[00:20:19] AK: Oh, no. That is certainly something we do. And it might be our most appreciated feature by our customers that deidentify data with Tonic. Earlier, Ian was giving the examples of deidentification. Like, "Oh, okay, you replace the name with X's, or you null it out, or everyone becomes John Smith, etc." Like, our customers typically need a more like advanced style deidentification that Ian alluded to. And part of that is consistency.

I think your example is a very common one. A name appears in your Postgres database, it appears in a log file, and it appears in some parquet file sitting in S3 that were perhaps the output of some ETL job out of Redshift or something like that, right? I mean, the point is, the name is everywhere.

Tonic can absolutely consistently deidentify your data. And our notion of consistency means that a given input is guaranteed to always yield the same output. We'll guarantee that Sean Falconer always goes to, I don't know, Malcolm Smith, right? We'll guarantee it. And it works at any data scale. We do not achieve this by like maintaining some huge HashMap in-memory that has to be maintained and checked all the time. These consistent values are able to be computed statelessly and on the fly. So, it works really at any scale.

[00:21:26] SF: And during the process of generating the synthetic data, are you doing things to identify, essentially, the PII or maybe PHI elements from the data versus just regular application or operational data?

[00:21:40] AK: We absolutely do. I think the first thing that Tonic does when you connect to the database is we run what we call a privacy scan, which is going to help you identify where your sensitive information is. And oftentimes – I mean, I'm using the word sensitive here. But really, I mean that means PII and PHI. And we identify like a great many types of sensitive pieces of

information. And then we can suggest to the user, “Hey, these are the columns that I would suggest transforming. And here's how we think you should transform them based on what our understanding of the column is.”

And like all of the algorithms out there, we get false positives and false negatives. But it does a very good job of identifying a significant majority of all of the sensitive information. And it really is like such a crucial step for our customers, because they typically come to us with like pretty complex databases with like hundreds or thousands of tables and thousands or tens of thousands of columns. Like, no one person in the organization even knows where all the sensitive data is. So, having a tool right at the outset that kind of tells you where it is, is really helpful.

[00:22:37] SF: Yeah, I think like one of the first chores of any privacy engineer is essentially discovery, because the point that most businesses are starting to focus on this problem, they've kind of lost track of where and what they're storing.

When it comes to, essentially, the tool providing this discovery piece and other things that I'm assuming are configuration options for the person using the tool that's doing these integrations to generate synthetic data, is there anything that helps them sort of turn the right knobs to make sure that the data is being anonymized or synthesized in like a privacy preserving way that's going to be compliant with various regulations like GDPR, or HIPAA, or PCI?

[00:23:18] AK: We do a few things. One, we try very hard to guard against customers unknowingly moving data from production into a lower environment when they potentially have not deidentified all of the necessary columns. For example, databases change all the time. Let's say someone adds a new column to the production database that contains something sensitive, and then someone else goes and runs a Tonic job without knowing that, right? You could potentially have just moved PII into your staging environment. Tonic can be configured, for example, to not run jobs when there are schema changes and when they haven't been kind of acknowledged, and confirmed, and dealt with. That's the first thing.

The second thing is the privacy hub. Tonic does a very good job of showing you, after a job runs, what's your privacy report look like? Like, how many columns that are sensitive that do not

deidentify? And it kind of keeps track of that for you. And it also kind of tracks these changes through audit trails that you can kind of request from the server. We need to understand what changes were made? When were they made? And who made them.

And then on the flip side of all this, we also have what we call a synthesis report, which will give you an idea of the quality of the output data set, like from a statistical point of view. So, you can understand, "Okay, did I actually preserve that numerical relationship between these two columns? Like, in the source, the correlation was .85. What is the correlation in the output?" We run these types of checks as well. We definitely try to attack it from all sides. Like, we want to make sure that you don't move sensitive data to your lower environment without protecting it. That, holistically, you have an idea of what like the privacy is of that output. Then you also have an idea of how high quality the output data is relative to the source.

[00:24:51] SF: Cool. Yeah, that makes sense. What are some of the techniques involved with actually creating high quality fake data?

[00:24:59] AK: I would say – I mean, the glib answer is let Tonic run its privacy scanner and then just do what it tells you to do. It does a pretty good job. And if you do that, you're mostly set. Other techniques? That's interesting. I think it's kind of like understanding the ramifications of various actions. For example, we talked about consistency a few minutes ago. Consistency is a really, really powerful feature within tonic. And I mean, I kind of assume, I think probably almost all of our customers use it to some degree.

And what consistency does is it increases the utility of your output data, right? But it does that at a cost. And the cost is privacy. Because the data is less private when you make it consistent, because you're open yourself up, for example, to frequency attacks. Tonic is really – Like, think of it as a knob you get the turn. You turn it one way, you get more privacy and less utility. And you turn it the other way, you get a little less privacy, but you get more utility, right? And it's kind of like up to the user to understand for their use case like where that knob needs to sit, like, on the privacy utility scale?

I think like best practices with Tonic are like understanding what these tradeoffs are. And we do, I think, a pretty decent job within the tool and through these different reports I was telling you

about kind of explaining these things. But there's always room for improvement there. And having someone use the tool that at least has a rough understanding of these things I think makes the job a lot easier.

[00:26:19] SF: Mm-hmm. In terms of starting with an integration, what kind of integration do you support out of the box in terms of connecting to my database, my warehouse, my lakes? Is it SQL? NoSQL? Different types of technologies? Are there limitations that exist?

[00:26:36] AK: Well, of course, there are limitations of some kind. But we support all of the popular database vendors at this point. On the SQL side of the house, that includes – And I'm going to list some databases. I'm not giving you an exhaustive list, because I don't frankly remember all of our connectors at this point. But it includes things like Postgres, MySQL, SQL Server, Oracle. We even support various flavors of Db2, talking about getting ourselves in age.

And then on the NoSQL side, we support technologies like Mongo and DocumentDB from AWS. On the warehouse side, we support things like Redshift, Snowflake, the BigQuery. And on the data lake side, we support the processing of files via Spark. You can bring your own Spark cluster. We can connect to Amazon EMR, which is their managed Spark service. And we also connect to Databricks. And I know there are databases I'm missing, but those are some of our more popular ones. And we're constantly adding new databases as needed by our customers. The list is always growing.

[00:27:26] SF: And how does a company that's utilizing fake data for the interest of privacy? Like, how does that your approach relate to other techniques in the sort of privacy-enhancing technology space of like encryption, tokenization? You mentioned, data masking governance. Is this a replacement? Or are these like complementary technologies?

[00:27:48] AK: Definitely complementary. With that being said, a lot of our transformations kind of utilize some of the things you've said. For example, we have tokenization generators that our customers can use. And they're really – In my mind, I really liked them. They're really clever tokenization generators, because they give unique tokens for every value. So, two different values will never get the same token, which is I think in most tokenization schemes is a requirement. But not only that, we accomplish this through a technology called format

preserving encryption, which guarantees that our tokens have the same length as the original values themselves.

If you want to tokenize your 32-bit integer column, we can do so by providing you with 32-bit integer tokens, which is a really – I think it's a really nice thing for our customers, especially those like using like relational databases with like really structured table types. Like, if you're in Mongo, and everything's a JSON document, that might not matter as much. But in SQL server, for example, it matters a great deal. We also utilize technologies like differential privacy, which is a privacy framework. And some of our transformations, not all, but some are designed to be differentially private. And this is something that's understood by our users.

[00:28:56] SF: For the listeners that are less familiar with the idea of differential privacy, can you give a little bit of an explanation of what that is?

[00:29:03] AK: I certainly try. But I would encourage everyone to go read about it for yourself, because it is really interesting. And I'm, by no means, an expert. I like to think of differential privacy mostly as giving plausible deniability in case of a data breach, where the attacker, he might learn certain things, but there's only some probability that what he's learned is true. And there's some other probability that it's not true. And it kind of leaves them not knowing.

And differential privacy has several like mathematical traits that make it really, really useful. For example, you can have a notion of applying differential privacy on multiple transformations within a given table on different columns. And then you can kind of understand that, “Well, if each of these columns is differentially private, I can kind of get a sense of how private the overall table is going to be,” through this notion of like what's essentially called a privacy budget, which is represented by this parameter in the literature called Epsilon. And there's other great traits, like, it has this notion of composability. But not to get didn't do it too much. But in short, I like to think of it as giving plausible deniability in the case of data breaches, which is, I think, a very nice feature.

[00:30:07] SF: Yeah. And this is – I think, differential privacy, I am, by no means, an expert either. But it seems like a very, very like sort of hot area that a lot of – Both on the academic side and in sort of commercialization, there's a lot of interest in it these days.

[00:30:21] AK: That’s definitely true. And I've seen the interest from our customers grow over the last four years. Just like we were talking about, like, four years ago, we were seeing folks still using production data where they maybe shouldn't be using it. And that number has gone down. Well, references and questions related to differential privacy has probably climbed up by the same amount during that time period. So yeah, I agree. It's a pretty popular area right now.

[00:30:41] SF: We've talked a lot about this idea of taking maybe a large amount of data and then sort of shrinking it down into something that an engineer could use to test locally or test on a staging environment. But what about the sort of reverse situation? Do you ever have situations where a customer has no data or very little data, and they want to generate more data? Maybe for load testing, or specific use cases, like clinical trials data, there's usually not a huge amount of data. There's maybe a hundred or thousands records. Is that something that people use Tonic to address?

[00:31:13] AK: They do. But it's not a super popular use case. But it is something the tool supports. If you have no data, you only have, for example, a table schema, and you want to fill it with data, Tonic can do that. But it's not really purpose-built for that exact use case. Like, in general, Tonic takes a set of data and gives you a new set of data that looks and feels just like the original, but it's fake. Right? And when you have no data, you don't have anything to base this output data off of. But Tonic can be configured to basically fill the columns of a schema with essentially nonsense. I mean, that's really what it is, right?

Like, you can say, “Okay, fill this column with one of these three values. Pick them at random. Fill this column with timestamps between this date and that date, etc.” Right? And you can build things that look kind of real. But there are perhaps other tools in the market that might be more suited for that. I don't know.

But yeah, when it comes to having a small seed set of data and wanting to grow it, that's definitely something that we can do. When we talked earlier about synthesis, that's like a pretty common use case when using some of our various synthesis techniques.

[00:32:17] SF: Kind of moving away a little bit from like applications of the technology to the actual company of Tonic and how maybe the product and engineering teams are set up, can you give some background of how you're sort of structuring your product engineering teams today? What's the product development process look like?

[00:32:33] AK: It's changing all the time as we grow, I'll say that. It wasn't too long ago where it was, I don't know, pretty chaotic, the Wild West. I'm not sure what you want to call it. But we've definitely grown up a lot in the past, I'm going to say, two years. And we continue to grow a lot.

We have a product management team that's really, I think, helped get us into shape. And that team has been, I think, growing pretty commensurately with our engineering team over the past few years. And I'd say, our engineering process now is kind of boring in a way, which is a good thing. It's much like the engineering processes that I think in other traditional software companies. We have feature requests come in. We have product managers doing customer discovery. We have bugs that come in. Everyone's structured by team. The teams typically own various components in their code. And they own the bugs that come in. There's a triage committee that kind of looks at the day's bugs and farms them out to the teams. Maybe if there's a critical bug that comes in, it'll go through a different process that is still a little chaotic. Like, we'll just pick the best person that's available to go kind of fix the problem or plug the hole. But in general, now, I'd say we're a pretty standard development organization.

[00:33:36] IC: One thing that I would say is a little unique that – I mean, it's not super unique, but it's something that we decided to do, a little uncommon, is we did decide to have product report into me. I'm the CEO. So, we decided not to have product report after engineering. And our thoughts around that were basically, we really want product to be the voice of the customer.

And so, often, that comes from sales, customer success. That's where sort of some of that data comes from. And we wanted product to really kind of be able to arbitrate between engineering needs in terms of scheduling, and consistency, and structure, and the need to really serve our customers.

I think that's been a really helpful choice. And it kind of allows this tension between – Sort of, there's always a tension between sales and engineering in terms of what can be accomplished.

But this way, our product really can be the kind of the neutral voice and figure out really what's best for our customers considering the needs of the organization and the needs of the sales team.

[00:34:35] SF: Given that you we made that decision, are there decisions that you've had to make as a company maybe in terms of how you think about product and engineering that relate to the fact that you're selling a solution that's actually related to privacy? Has that changed maybe the way that you do things internally, or maybe you spend a little bit more time thinking about those things about your own product versus maybe what another company might be thinking?

[00:35:00] IC: Definitely. I mean, obviously, when we started, the in vogue thing was cloud first, cloud only. We, early on, from talking to our customers, just heard what they needed. And we realized that was not the right way to start Tonic. And so, we did have on-prem out of the gate. And I think that was the right move. I think that's been really important for a lot of our customers. I think it will continue to be important. We're obviously going to do things to make customers feel really comfortable using our cloud instance. I think, especially for small or medium-sized customers, there are huge advantages to not having to maintain things themselves.

But I think what it does do is it makes Tonic really approachable for a wide range of customers. We have some very, very large customers with really stringent security requirements that we just nail, because we're not asking them to send any data outside of their network. And then for certain medium-sized customers that need a little bit more convenience, we can serve that need as well.

[00:35:59] SF: Actually, in the cloud deployment – And this is something that we touched on earlier. But how does that work? Is that essentially single tenant environment? Multi-tenant? How does deployment for the cloud work?

[00:36:11] AK: It depends a lot on when customers join us in our hosting environments, we have a conversation with them for exactly what their requirements are. And it's a little bespoke at the moment, just because most of our customers do opt to deploy on-prem. the answer to your question is it depends.

[00:36:26] SF: And then the company was founded in 2018. Is that right?

[00:36:30] AK: Yes.

[00:36:31] SF: Yeah. It's been about four years. During that time, has there been anything that's really surprised you in terms of maybe a substance that you had about the type of customers that would need a solution like this? Or maybe the technologies that you use turned out to be wrong? Or that you had to change gears? Or build new features?

[00:36:48] IC: Yeah. I think, I mean, there's so many things that surprised me, specifically, in those – I mean, there are other things that I do throughout the sort of journey of starting a company. I think, specifically around verticals, we kind of went in thinking, "Hey, our first customers are all going to be in financial services and health tech." I mean, I think there is a lot of interest from those verticals. But we started getting outreach from customers reaching out to us from all different types of verticals. Particularly, early on, we had a lot of interest from Ed Tech. We still see that. And now, if you look at our list of logos, it really does span across industries. And I think that was something that did surprise us, is sort of like just exactly how ubiquitous this problem was for people building software. And I think, for us, now we're kind of thinking this is really applicable to anyone doing the SOC2 who's building any kind of data-driven product. It's a really wide range of folks that this applies to.

[00:37:45] SF: You mentioned SOC2. I saw that you recently became SOC2 type one compliant. Congratulations on that. I know that a lot of work goes on into that. Can you explain what that might mean for your customers or potential customers interested in a solution like yours?

[00:37:59] IC: Yeah. Part of this is just wanted to be really clear and transparent about our practices with everyone. And I think it's a very accepted standard. So that was part of the rationale. But also, we wanted folks to feel comfortable leveraging our cloud instance. We figured this was a good way to build a little bit more confidence there.

I mean, obviously, we recognize that I think there's always going to be need for on-prem. But we want people to not have any sort of security concerns if they're trying to make a more convenient choice for themselves regarding their own infrastructure.

[00:38:32] SF: Right. What's next for Tonic? Is there anything you can share in terms of future roadmap items? Or perhaps areas of privacy enhancing technologies that you're really excited about?

[00:38:43] IC: Sure, yeah. I'd say there's a few things. I mean, obviously, we started with developers. We're now having – We have a lot of technology now to support data scientists. I think that's going to be something we continue to invest in. Really making sure that we can serve that community as well as we're serving the developer community.

And then, I think, we're going to continue to release features and product that supports the increased complexity that we're seeing from our customers. We're working with larger customers now. More complex needs. There's going to be a lot of work there to support that. And then, I think, the other thing we're going to be doing is certainly making Tonic more and more accessible to folks. I'd expect more ways that you're going to be able to start engaging with Tonic in a lightweight way. So, think self-serve, freemium, things like that.

[00:39:31] SF: Right. Adam, do you have anything to add to that?

[00:39:33] AK: No. I think what Ian said is spot on. I'm personally most excited for the investments we're making right now in our data science offering and all of the kind of like new capabilities we're beginning to offer data scientists specifically. And I'm really excited to see how that continues to land with people.

[00:39:47] SF: Yeah, I mean, data science is another one of those areas that really seems to be taking off right now. Is there anything else you'd like the audience to know?

[00:39:55] AK: I'm going to say something. I guess, you asked us a few minutes ago what surprised us when we started the company. And I didn't give an answer. But I'm going to give one now because I'm going to turn it into something that's entirely self-serving. The thing that

surprised me the most was how hard it was to recruit talented engineers to join our team. I had this like very naive idea of, “Oh, if you build it, they will come.” And that's just not true. Recruiting is hard work. And we, as founders, spent significant amounts of our day actually working on recruitment, believe it or not.

I want to give a special shout out to all of the engineers listening to this podcast, and to let you know that we are hiring, and we are growing, and we have really exciting challenging problems for you to work on. If you're interested, come on over to Tonic.ai. Let us know you found us through Software Engineering Daily. And let's talk.

[00:40:41] SF: Awesome. All right, Adam and Ian, thank you so much for joining us today. And I'm glad that you added that call to action for all the engineers that are listening to this.

[00:40:49] AK: Absolutely.

[00:40:50] SF: Great. Thank you.

[00:40:51] AK: Thank you.

[00:40:51] IC: Thank you.

[END]