EPISODE 1206

[INTRODUCTION]

[00:00:00] JM: Over the past few years, the conventional wisdom around the value proposition of big data has begun to shift. While the prevailing attitude towards big data may once have been bigger is better, many organizations today recognize that broad-scale data collection comes with its own set of risks. Data privacy is becoming a hotly debated topic in both the technology industry and in regulatory agencies and governments. Bigger private datasets are more attractive targets for hackers, meaning that an organization must invest heavily in security as well to avoid a breach. Every organization faces a tradeoff between the value of the insights produced from large datasets versus increasing storage costs and privacy risks.

Tonic is building a synthetic data platform to address these tradeoffs and help organizations mitigate data risk. Tonic takes in raw data perhaps from a data lake and transforms it into a more manageable de-identified dataset for ease of use and user privacy. Tonic can create statistically identical structured data sets that allow software engineers and business analysts to extract the same useful insights that drive an organization's progress without the risk of working with identifiable private user data.

Ian Coe, Andrew Colombi, and Adam Kamor are co-founders of Tonic, along with their fourth co-founder, Karl Hanson, they have all worked at Palantir Technologies where the idea for Tonic was born. They join the show today to talk about the value of synthetic data, the risks and rewards of big data and how compliance, privacy and security are driving innovation in the data management sector.

[INTERVIEW]

[00:01:31] JM: Guys, welcome to the show.

[00:01:33] AC: Thanks for having us.

[00:01:34] IC: Thank you.

[00:01:35] AK: Yeah, likewise.

[00:01:37] JM: Simple question. What is synthetic data?

[00:01:39] AC: Oh man! That's like simple question.

[00:01:42] IC: It's actually not a simple question. It's not a simple –

[00:01:44] AC: Simple question, man. Wow! I'll start and I'll let Ian and Adam chime in too, but like the short answer is there's no consensus to what that means. I think, colloquially, when people think of the word synthetic data, they're thinking, "Oh! We're going to create data based on some sort of statistical model of the underlying data. And the reason we're going to do that is we think it'll help protect the privacy of the underlying data." There's that reason and maybe another reason would be we can create as much of the data or as little of the data as we want because it's from a statistical model. So I'll just sample that model more if I need to. So I think those are the things that are going through people's heads when they're thinking about like what synthetic data could do for them.

But why do I say it's not a simple question? There are just so many ways you can approach that. And even ones that seem like straightforwardly, like, "Oh, this should definitely protect the privacy of my users or whatever," aren't actually as protective as you think they might be. So there isn't like one simple answer there. And what our product does is try to give you like a set of like a tool box, a little bit of a toolbox to produce data that provides the utility you need, meaning like it can create data that's useful for testing, if that's what you want to do with it, or sales demos, if that's what you want to do with it while also protecting the privacy of your users, which means you don't reveal any sensitive information about your users. And so like I said it's a bit of a toolbox in that way because there are different ways you can configure it depending on your needs.

[00:03:28] AK: That's right. I would add on to that that like what you mean by synthetic data can also – Like it can depend on what your use case is for that data, right? Where like the use casing kind of imply or suggest different techniques or algorithms that should be used to generate that data and also kind of help you determine what the privacy bar is for the data you generate, which again is going to change what techniques and algorithms you use.

[00:03:56] JM: So the main motivation or at least easiest to explain motivation that I can see is like I'm a developer working at a company. I have access to a database with customer data. I'm going to make a query to that database. I should not be able to see actual customer data, right? I should be able to see samples of the data that looks like it's real.

[00:04:20] IC: Yeah. I mean, certainly that's very aligned with our philosophy. Obviously it depends on your specific organization and the types of data that you're working with. Obviously things like medical data are more sensitive than certain other types of data. But long term, if you're looking at your data governance strategy and you're trying to keep your organization and your users as secure as possible, it's going to help your company and put you in a better place in it from a security posture and just general liability perspective if your lower environments don't have all kinds of user information scattered throughout them. And then, conversely, that your developers have a data source that's really useful and actually allows them to have test cases that are valid and everything. So you sort of want both to protect your users but also have something that doesn't slow engineering down.

[00:05:18] JM: And is there also a motivation to getting synthetic data to increase the volume of data that you have available to machine learning models?

[00:05:31] AC: Well, that certainly could be a motivation. Another version of that, increasing the volume, is for scale testing. Actually our main focus is not in getting synthetic data into the hands of ML practitioners, but rather getting them into the hands of developers, QA people, designers, product managers, people that are more like in the software development side of things rather than the model development side of things. We know that there're a lot of use cases out there for synthetic data for ML, and I think that's like a hot topic right now. It's not where we've decided to focus. Yeah. And if you look at our website and you look at like our customer lists you'll see like, “Oh, yeah. Yeah, that kind of makes sense.”

[00:06:15] JM: So walk me through the process of actually creating some synthetic data.

[00:06:22] AC: Sure. So the way our approach kind of divides the problem is what users are doing in our platform is annotating data in their database looking at the different tables and the different columns and annotating those things so that the system can understand like what the appropriate model is for that kind of data. So like a very simple example to give you. Let's say you have a table of users and they've got like geographic information in there. We can annotate the columns that, “Hey, these columns are related to geographic information, or these columns are categorical in nature.” And then the system will understand, “Okay, that's what this has been annotated as.” And by the way we have systems in place to automatically annotate some of your data. So for geographic information, for example, we can automatically detect that.

And then the system creates a composite of models. So by that I mean we've got many different kinds of models in our system. It's not just like one model that rules them all. We've got models for geographic information. We've got models for categorical, for continuous, for event kind of data, all these different kinds of models, and the system puts those models together to create one giant composite model and that aggregation of all the models for all the tables and columns in the system become like the super model or the meta model that is what's used to synthesize data going forward. Yeah. So that's kind of like how it works in an overall perspective.

[00:08:02] JM: What is difficult about building that?

[00:08:07] AC: From a user's perspective or from our perspective. I’ll start like from our perspective. I mean, when we first started this company about three years ago now we definitely had a different set of perspectives than we have now. We were naive and idealistic and we thought , “Okay. Well, we can just make one model to rule them all.” And as we encounter different use cases at different customers and the complexity of the data – One thing that we haven't talked about, I'll just briefly mention. We focus on entire databases. We don't focus on datasets. We focus on databases and many databases. If you go to your typical tech company, like eBay is one of our customers, they have thousands of databases and tens of thousands of tables as you might imagine. And so a system that works on a couple CSV files, while can be very impressive and useful in certain use cases, is not going to be what you need for software development.

So that was one of our early learnings, is that like we need to focus on databases, which also means focusing on what makes that problem unique. What makes that problem different from the problem of working with a couple CSV files? Well, it's the relationships across the whole databases. It's the relationships between databases. So those are some of the things that we focused on early on I think helped us get traction with the software development community. It's very helpful there. So that's like the large scale problem of doing a whole database, but there's also the detailed problems of, "Hey, we've got a text field. It's got sensitive data in it sometimes maybe somewhere." And how do you find the needle in that haystack? So there are problems both at the large scale and at the small scale of how you create synthetic data.

[00:10:01] JM: So can you tell me more about the problems that this solves for typical companies? So you've got problems in testing and security and compliance. Tell me a little bit more in detail how generating synthetic data alleviates problems that they have.

[00:10:20] AC: Sure. And Ian, or Adam, you should definitely chime in if you have more thoughts. I mean, some of it's regulatory, right? There are things like CCPA. There are things like GDPR which have kind of a detailed description of what kind of problems you can face if you don't adequately secure your data. And then there are other like more specific regulations in particular industries like HIPAA for healthcare data and there're equivalent ones for education data. It turns out that we have a lot of customers in ad tech and in health tech and in fintech, because fintech is another example of a specifically regulated industry. So that's like if you're in one of those industries you know the laws that apply to you and you kind of need to find a solution.

Then there are more like what's the best practice or what's the kind of company culture we want to have or what's going to do right by our customers. And you find that that's the kind of thing that hits companies as they grow. They're starting to scale up. They're beyond 20 engineers now. Maybe they're at 100 engineers or something and they're just sort of, “We should do the right thing here. The data being exposed to too many people isn't good for our customers.” And that's like another level of thing that can come in as a company grows. I don't know, Ian. You've probably thought a lot about this too.

[00:11:44] IC: Yeah. Yeah. I mean, Andrew, I agree with everything you said. I think what's interesting is when you actually talk to a lot of our customers they say things like, “Wow! We've been wanting to solve this for years and we've just been too scared to take it on because it's an inordinate amount of work.” And they're really glad to see that there's someone out there who's going to solve this in a first class way and can actually deliver results in a month and let them essentially transition their dev team from a dependency on this data with a lot of sensitive information into a dataset that is much, much more secure and poses much, much less of a risk. Because, I mean, the thing is people secure production in most cases. I mean, it's access control logged, what have you, but then if someone has to copy that data somewhere else, typically wherever they've copied it to doesn't have that. And that gives CSOs a lot of heartburn.

And I guess know sort of going a level deeper on what Andrew said around industry is in the fullness of time we believe anyone developing software of a certain level of complexity and at a certain level of scale needs this type of solution. The way we kind of think of it today is that if you're a healthcare company, we should be talking to you if you're 10 people big. If you're a fintech, maybe 50 people because there's quite a few regulations on fintechs as well. A standard B2B company, and this is I think kind of the sweet spot and where many of our customers live, is around 200-ish people. At that point you've maybe hired a CISO. You're considering something like a SOC 2 audit. The other thing is that you now are signing data covenants with customers saying how you're actually going to use their data. You might have your CISO going to meetings and explaining your data practices and how you're protecting the data. And so that's something that gets – Even as you reach that sort of SMB level, that's something that becomes increasingly more complicated to manage and handle appropriately.

If you're B2C, it's often a little bit bigger just because you're not having to deal with those kinds of things as early and might not be as – But again, all this, as Andrew said, is dependent on your company's culture. But I guess those are sort of the frameworks that we think about when we think about who's going to adopt this and when.

[00:14:08] JM: So if I'm running Tonic over my dataset, I mean there's all kinds of things that can change in my databases. Like the schema can change. New kinds of data can be added. How do I know that the Tonic system is going to be consistently updated with the changes to my database?

[00:14:35] IC: Well, glibly, I would say that's one of the things that does make this hard to your previous question, but I'll let Andrew provide more detail.

[00:14:43] AC: Looks like Adam was ready to –

[00:14:46] AK: Yeah, I can take this one. I mean, in a rapidly developing product like a lot of our customers have it's really common for schema changes to happen, new columns to be added, columns to be changed. The meaning of columns can change over time, etc., right? So what Tonic does is basically alert users to when new data has been added that hasn't yet been acknowledged that this new data is there in the Tonic system. Tonic can be configured to essentially not allow a new generation of data to occur until someone like a user of tonic has acknowledged the presence of this new column and has either said, "Yeah, that's sensitive data. Let's deal with it," in one of the ways Tonic supports. Or, "Oh no, that's not sensitive. We don't need to do anything about it." Because at the end of the day we really want to make sure our customers don't inadvertently take sensitive data from production and then move it into a lower environment without at least having the option first to transform that data.

[00:15:41] AC: And the one quick thing I’d add is this is another example of a learning that we had from working with large databases as opposed to just a few CSVs here and there.

[00:15:53] AK: That's right.

[00:15:55] JM: Any other frustrations of working with those large databases? Integrating with them?

[00:16:02] AK: I can throw one out, that is it's not necessarily a frustration. It's actually we've kind of turned it into an opportunity. But our job, or rather Tonic's job can be done slightly easier when there are foreign key constraints already in the database. Because foreign keys, they kind of tell you what relates to what, right? But often times, and for very good reasons, large database systems might not have foreign key constraints. And initially that was painful, right? It was like we weren't able to like make decisions on certain things without these foreign key constraints. But we've actually, in newer versions of Tonic, and we've been doing this for I guess well over a year now, come to think of it. We actually allow the user to add what we're calling I guess like virtual or logical foreign key constraints on top of the system that don't actually go into the database layer, but you're telling tonic where those foreign keys would be. And then we can treat them as if they were real and still do all of the magic that Tonic does. And actually we're going further now and actually helping users detect where they are for them, because oftentimes in large database systems you won't even know where they are. So that's one example. Andrew or Ian, any others?

[00:17:11] AC: That's a great example. That’s good.

[00:17:15] IC: The only other thing I was thinking of is just performance. Many of our customers depend on regular refreshes being able to move large amounts of data quickly. So I think that's been another learning that was really important, is that everything we do needs to be very, very efficient. And we spend a lot of time thinking about how we optimize performance.

[00:17:38] AK: Yeah, that's a good one. I think when the company was much smaller than it is now, Andrew and I spent many hours together looking at profile traces. Figuring out where we could eke out the most performance and speed.

[00:17:51] JM: When you talk to companies like Flexport or eBay, these companies that have gigantic sets of data and lots of problems that they need to tackle with the data, what are the other problem sets that they come to you discussing?

[00:18:09] AC: I think the big one that I can think of right away is subsetting. So when you look at eBay's data, and like I said, it's spread across many databases, many tables, and it's massive, right? Petabytes and petabytes of data, and you want to create something that's useful for development. Likely the answer is not another petabyte and petabyte dataset. In fact, if you want to be able to load that database on your laptop so you can do local development and not really step on anyone else's toes while you're doing it, you need a much smaller dataset. And so this is one of the features that we invested in early and are continually improving is our subsetting algorithm, which what it basically does is it kind of understands the web of relationships in your database and across databases, I would add, through foreign keys and through the logical foreign keys, as Adam was explaining earlier, so that when you extract a single entity, like let's say in eBay's case, it's like an item that's for sale, you don't extract just that entity, but you extract all the metadata associated with that entity. So that would be the users that are involved with that entity. Any like bids on that entity and the myriad of other data that is associated with an item or any entity in a large database system. So we invested in that early. It's something that's used heavily at eBay as well and also at our other customers. And Flexport's an example where they specifically did want a database that's suitable for development purposes on their local laptops. And that's another example of where like being able to pair the data down is very helpful.

[00:19:57] JM: So you have this this problem of data de-identification. So when you are trying to create synthetic data for users, you want to create data that is anonymized, if I'm understanding it correctly. And I remember reading a paper about this a while ago, the anonymization problem, and that it's really, really hard to make datasets where you shadow certain fields and try to anonymize the data, like removing the address from all of the users in the database. Have you guys actually made breakthroughs in de-identification or are there some new techniques that are available?

[00:20:42] AC: Yeah. I mean, de-identification, it's maybe a pejorative. It's tough, going back to your earlier questions about like what is synthetic data. What is the identified data? What is masked data? What is obfuscated data? All these things have like squishy definitions and they can be dangerous because you can say like, “Well, I’ve identified the data.” And exactly what that means is going to really matter because it may or may not have accomplished the de-identification that you were hoping for.

There's this famous example of Netflix, the Netflix challenge where Netflix released a bunch of data having “de-identified it”, and then sure enough internet sleuths were able to re-identify certain users based on those users activity that was masked in the Netflix dataset but not masked in the real world. And so you can go on imdb.com and maybe correlate activity of a user in Netflix with a user in imdb.com and then reverse who that person was.

Yeah, I mean, it's a difficult thing to do. And there are techniques though. There are known mathematical frameworks for thinking about how to properly really de-identify data. The main one that comes to mind is called differential privacy. It's been around – It’s discovered or invented depending on your preference in, I think, 2006 out of Microsoft Research and it's been a real mainstay of data privacy research going forward. And then there are other techniques as well and there're other approaches. It's not it's not just differential privacy, but that there are a handful of approaches, and we incorporate those in the product.

So our product is very configurable, like I was saying before, there's a toolbox kind of approach and there are ways you can configure it to maximize the privacy which can have a detailed deleterious effect on the utility. And then there are approaches that can more maximize the utility. And the approach you take is going to depend on what you think your adversary is, right? Is your adversary other people in your company or is your adversary the world? And then of course even if you're not worried you're not going to like publish the data to the world, it may get inadvertently published to the world because, although you trust your employees, the employee loses the laptop or whatever. So it depends on what your threat model is and what the data is exactly that you're trying to protect. But there are ways of being very mathematically rigorous about what it means for the data to be private, and we incorporate those in our technology as well.

[00:23:19] JM: As you've mentioned, this is a product that is going to get used by data scientists, by QA people, by customer service people, by a wide variety of people. How do you keep the interface widely accessible?

[00:23:37] AK: I would say that I think all of those groups of people could certainly use the tonic interface, right? I think we accomplished that by just having same defaults, a clear UI, helpful documentation, etc. But really it's mostly larger groups of people use the data that Tonic generates. It's not necessarily the same group of people using the Tonic interface to like generate the configuration that later on generates the data. So I just wanted to call out that distinction. But I do believe that a wide variety of people and backgrounds do use Tonic successfully. And Andrew, you were going to say something as well.

[00:24:16] AC: Yeah. No. I mean, I think that covers a lot of it. The product, you can go on our website, you can see like little vignettes and snippets of it. It is a UI-focused thing. It's not like an API product. And we believe that's a valuable thing for us because it helps users visualize what they're doing very easily. And I’ve been a coder for more than 20 years and I love me some code. But if I have the choice between coding and just like clicking on something, often I will pick the click because it's quicker and often gives you like easier feedback and stuff like that. So we have a UI and it's kind of WYSIWYG to borrow a term that's not used very much anymore. Yeah.

[00:25:02] AK: Right. Really, I mean, just to emphasize it once more, you can use Tonic without writing a single line of code.

[00:25:11] IC: Yeah. I mean, all our backgrounds, I mean, we come from sort of the data analytics and you know data BI space. So actually that sort of ingrained in us to make interfaces that are intelligible to users and help you visualize and understand your data in a clean and clear way.

[00:25:32] JM: Take me inside the product development. What has been the strategy for organizing your engineering team and what are you focused on right now?

[00:25:44] AC: Sure. Well, just to give a little more background, like our company is currently 20 people and hiring. And so if you're interested in any of the problems that we've been talking about so far, you should definitely drop us a line. But yeah, I mean, our team isn't super big yet. And so it's been a very flat structure. We hope to continue the flat structure. But everyone does everything. We don't hire frontend only or backend only. Like you get a feature, that feature is yours from the SQL that you need to generate to do it to the React, TypeScript component that needs to be created to take advantage of it. We find that to be a like an empowering way for people to work and also giving people like the whole problem is more efficient in many ways because you're not like trying to communicate between different teams or different people trying to like agree on an interface or whatever. So I think that's like the way we've organized the engineering team so far. And then on the question of like what are the big challenges or the things that we're working on, I like to divide it into three main categories of problems that our engineers are working on. The first one is data pipeline stuff. So it's like we do a lot of grab data from a database. Do some analysis of it or even take it wholesale and then writing data to an output database. And actually I want to – As a brief aside, like I said take it wholesale.

Another learning from doing this on databases is there are many tables in a database that you don't need to make private. They're like the schema version, because it automatically gets updated. Or there are other things in a database that you don't need to actually synthesize or make private. Anyway, data pipeline is the first category. Second category is ergonomics. So that's just like making Tonic easy to use. And it's not just easy to use but like fit with a corporate workflow. I mean, when you've got a company like eBay that has thousands of engineers that are going to be working on data that you create, there are a lot of stakeholders in the synthetic data that you're creating. And so there're a lot of people that want to have visibility into it. There's like compliance that wants to be able to approve things. So there's a whole workflow and that's where I put in like the ergonomics bucket of like it's not just ergonomics for the person but it's ergonomics for the organization. The whole organization needs to feel comfortable using this tool.

And then the third category of things that we work on is the more mathy stats and the privacy utility. We want to create the most private data with the best utility that we can. And doing that is research grade problems. And we're definitely borrowing from academia as much as we can and building some of our own stuff as well. But yeah, if you're interested in doing math and stats, that's definitely another area that we that we focus.

[00:28:48] AK: On the project side, the first thing that Andrew spoke to, I'd like to add one more thing. It's definitely true that our engineers, or rather an engineer, will take a single feature and kind of drive it from start to end. And part of that process is also communicating directly with our customers. We have a tight communication channel with almost all of our customers. Typically sometimes over email but oftentimes on Slack or Microsoft teams or the chat app Du Jour. And our engineers will work with our customers on a very regular basis helping them sometimes with configuration changes, but oftentimes collecting specs and requirements on new features that they're going to drive forward in the future. So in a way, at least for now, our engineers are also acting like their own product managers. But that obviously changes somewhat as the company grows in size.

[00:29:40] JM: I'm curious about taking some of this this research like we discussed with the data de-identification and productizing it and verifying it, verifying that you've productized it correctly, because these algorithms can be really complicated to actually implement. Can you tell me a little bit about the engineering behind an algorithm like that? Like what programming languages you use? How you test it? How you verify that it works as intended?

[00:30:10] AK: I’ll give an example of an algorithm we use and I can talk about how we verify that it's correct. So one type of like algorithm that we make fairly heavy use of is something called format preserving encryption, which is a way of doing a two-way. Meaning you can undo it, encrypting a piece of data, but encrypting it in a special way such that the ciphertext, that is the output of the encryption algorithm, resembles in some way the plain text, the original values, right?

So for example, if you have a column of 32-bit integers, when you encrypt them, you would like them to remain 32-bit integers. Or if you have a column of ASCII values –

[00:30:52] AC: I think a nice example is like credit cards. If you have like a column of credit card numbers, you want the output to also be credit card numbers.

[00:31:00] AK: Right. Meaning, if there's like a check digit in that credit card number, which there happens to be, you would like to ensure that that check digit is satisfied even with your encrypted ciphertext, right? And that technique is typically called format preserving encryption. And we have to implement our own algorithms there. You asked what the coding languages are. For this part of our backend, as for most of our backend, it's written in C# using .netcore. Or rather now it's just called .net, but it's cross-platform C#. And to verify the validity of these algorithms it's actually fairly simple for encryption algorithms. Simply decrypt and ensure that what you've encrypted when decrypted goes back to the original plain text. So that is one way of verifying as well as doing other checks, for example. I mean, over 32-bit integers you can actually kind of test this over the entire space if you need to ensuring that you have one to oneness as well is another good check to do.

[00:31:59] AC: Yeah. And like for some of the other algorithms like when we implement differential privacy for example, it is challenging to verify the correctness there. But we have like a data science team that that's kind of what they do is like take a look at these algorithms and try to look for various attacks on it. There's like the implementation and then there's the theory, right? So like the theory, it's all great, of course, but then you implement it and you may have implemented it with a bug. And so like the theory doesn't actually hold up, right? And so verifying that the actual implementation upholds the qualities that the theory says it should such as being resilient to reverse engineering, et cetera, et cetera. Those are things that that team looks at. And it's a lot of – We have the algorithm. Let's do some Jupyter Notebooks and let's try to crack it.

[00:32:51] AK: That's right. Some of the go-to testing mechanisms for developers are unit tests. And this is a conversation Andrew and I were having recently. A lot of our results are probabilistic in nature, right? You run it twice, you're going to get different results. But you know if you run it enough times you know what the distribution of results should look like, right? We know what that is, but for any given value we don't know what it's going to be. So creating unit tests that can test statistical and probabilistic features is something that we've put some amount of thought into, and I think we're happy with our framework there.

[00:33:29] JM: And so as far as testing it, can you give me a little bit more detail into how you test the overall system and make sure that it's not de-anonymizing at at any level?

[00:33:44] IC: I mean, I think one thing to understand about our product is that we take data from production or whatever secure source and then we make a net new dataset out of it. So there aren’t sort of like artifacts and things that it's not in the sense of like there aren't sort of like things that you will be able to poke at in in the process sort of if you're thinking of it as sort of a pipeline. So there's sort of I think what Andrew and Adam were kind of alluding to is that they're sort of an entering and exit, and that's the main focus and really the important part to test.

So if for some reason you’re – We offer basically on-prem and then also a cloud-hosted version of our product. And so I think for folks who are using the cloud, those kind of questions become a little more important and we do have assurances that we give those customers and we never retain data, things like that. But for folks that are especially concerned about sort of the intermediate steps, that's a really good reason to use the on-prem and then you're sort of in control of the data throughout the entire processing and it all stays within your VPC.

[00:35:04] JM: Gotcha. So as far as the compliance use case, if I’m getting my company audited, how does having anonymized data help me pass that audit? I guess I’m not super familiar with data privacy law. So I’d love to know a little bit more about how this kind of system is useful.

[00:35:29] IC: Yeah. Yeah. So there's actually a tenant within SOC 2 that says thou shall not use production data for testing and development. So that is a specific thing that you actually have to check a box on if you're going through a SOC 2 audit. And so that's one that's really, really specific. GDPR and CCPA both have slightly more vague sort of language around this. But basically the main thing that we advise folks to think about is that if you breach under CCPA or GDPR, you're subject to very large fines except if you have de-identified the data. And as we were discussing before, they do not define what de-identification is but they use a language similar to the act of something like substantially resistant to reverse engineering.

So that doesn't necessarily mean that if you gave an intelligence service a decade to reverse it that you had to withstand like that kind of adversary, but substantially. And a lot of this is up to prosecutorial discretion. So if you look at sort of the big privacy laws and security frameworks out there, they talk about this and you are in a much better place in the event you have and have an adverse event if you've taken steps. You're less likely to get fines. You're less likely to have issues with your customers.

[00:37:04] JM: Gotcha. That makes a lot of sense. So what's in the future for you guys? What are the next products that you think you'll be building or features that you'll add on to Tonic?

[00:37:17] AC: Yeah. I think going back to those three main categories of development work, the pipeline work, the ergonomics work and the stats utility privacy work, those are the areas that we're working on every day just to like go maybe a level deeper there where we mostly work against like basically the most popular relational databases, right? Like Postgres, MySQL, SQL Server, Oracle, that kind of thing. Expanding beyond that I think is really interesting. We already have a product actually for Spark. So if you have massive amounts of analytics data that you want to quickly and scalably do some sort of data protection on, that's something that we can already help you with. But expanding beyond that I think is an interesting spot.

Also, increasing the ergonomics, right? Like I was saying before with enterprise customers, they have demands around how customers – How their different teams can work together, and I think making that, giving people a workflow, giving people a way to almost like codifying what is good look like for a company working on protecting their data. Not just from the like algorithmic perspective of, “Oh, we're going to use these algorithms to protect the data,” but just from a process perspective. Who should be involved in making the decisions? What kind of visibility? What kind of reports? What kind of audits should be available so that if there is an incident you can track down exactly what happened? Those kinds of features I think are going to be really important in the future and especially as like larger companies take this very, very seriously, and smaller companies too.

And then the last thing, the statistics and stuff, like I said before, we're always trying to increase utility and privacy. It's a very difficult challenge. They directly try to defeat one another. The more utility you squeeze out of a data, the more privacy you're likely sacrificing, but that doesn't mean that there are ways of different approaches that can improve the score on both privacy and utility. And so finding those compromises. It's a compromise. It's not like a silver bullet always, but finding the right compromises that fit our customers’ needs is what we're looking for.

[00:39:44] JM: All right. And zooming out, given that you work with a variety of companies and you see some of the data problems that they have up close, do you have any more general industry perspective about changes to the data landscape that we'll see in the coming years?

[00:40:02] IC: I think one thing that we're seeing a lot of is microservices. So I expect that we'll continue to see more and more of that. And obviously that's something that if you're going to solve this problem, you need to solve it in a way that supports those type of setups.

[00:40:23] AK: I think another interesting challenge that companies have been coming to us a lot with lately is satisfying all of the new laws and regulations that are coming out, or maybe some of them aren't so new at this point, but they're relatively new. And I think we're going to see more and more on that front, and hence additional features and capabilities and Tonic just to always allow our customers to meet the compliance issues that come up.

[00:40:48] JM: Cool. Well, any other topics you guys want to explore or do you think that's – I think we've covered everything in the scope of Tonic and your perspectives on data infrastructure.

[00:41:02] IC: We just want to reiterate that, as Andrew said, there are some really interesting problems to work on here at Tonic. And if they sound interesting to you, we'd love to chat.

[00:41:13] JM: Awesome. Well, thanks guys for coming on the show and talking about Tonic and synthetic data and what you're building.

[00:41:20] IC: Thanks so much.

[00:41:21] AC: Thanks for having us.

[END]