EPISODE 1222

[INTRODUCTION]

[00:00:00] JM: Google Cloud, AWS and Azure are the dominant cloud providers on the market today. But the market is still highly competitive and there's significant overlap in the services offered by all three major cloud providers. Since all three offer a broad range of services, developers looking to choose a platform for their application must focus on providers domains of relative excellence on how those align with their needs. One domain where Google Cloud platform excels is with its database offerings. Google has data management baked into its organizational DNA and has been the source of several innovative technologies in the data space such as Spanner, BigTable and BigQuery.

Andi Gutmans is the leader of several engineering teams on databases at Google. He joins the show to talk about Google's database offerings and the engineering challenges of making them function at scale.

[INTERVIEW]

[00:00:48] JM: Andi, welcome to the show.

[00:00:50] AG: Hi. Thanks for having me.

[00:00:52] JM: You work on databases at Google Cloud, and that is a broad topic. There are a wide range of databases, Spanner, BigTable, BigQuery, Firebase, all kinds of cloud-hosted, Postgres and MySQL options. Give me a broad overview for what your day-to-day work looks like overseeing all these different database products.

[00:01:18] AG: That's a great question. Let me give you a bit of context of what our customers are doing right now and how they're transitioning into the cloud and then I'll talk about how that impacts my day-to-day. So you can imagine that we've been putting databases on

premises for the past 40 years, probably even more than 40 years. And according to Gartner, by next year, 75% of databases will be in the cloud. So if you can just imagine that kind of 40 years of databases moving into the cloud superfast, that's what's impacting my day-to-day.

And so what I focus on is kind of the journey that our customers are on into the cloud. I like talking about the three-step journey. You have customers who are trying to lift and shift and move workloads as fast as possible, and we kind of talk about that as the migration journey. We then have customers who want to modernize and actually move off of proprietary databases onto more open APIs or cloud native databases. And so that's actually changing their application. And then the third are the more transformative applications that they use BigTable or Spanner for. And so from a day-to-day perspective, a lot of my time is focused on talking to customers. Really understanding how that journey is going and then working with the product and engineering teams to make sure we're building the best services to really help our customers on that journey.

[00:02:42] JM: So there's really a wide variety of database products that people are choosing from, and I'd like to focus on, first, transactional databases. So I think, specifically, I think about Spanner versus traditional key value stores like Mongo or SQL databases like Postgres. How are customers choosing between transactional databases? What are the parameters that they use to choose what is going to form their core OLTP store?

[00:03:18] AG: That's a great question. I think a lot of the customers have already made some choices on the traditional databases such as MySQL, Postgres, SQL Server, Oracle, or also MongoDB, and the majority of the workloads today are on relational, and I'll talk a bit about the non-relational side also. So we see a lot of you know kind of built for web applications sitting on MySQL. We of course see the more traditional OLTP workload sitting on Oracle or sitting on SQL Server. A lot of departmental applications on SQL Server. And then Postgres is really in a huge renaissance today where customers who want those more sophisticated transactional capabilities but do not want to go down the more cost prohibitive proprietary path choose Postgres at the target.

So from a relational perspective, customers who want the more traditional relational model and SQL make that choice. Now what we're seeing though that as customers are building more mobile-first and web-first applications, they're dealing with a level of scale and availability needs that traditional databases weren't really able to address. And so that whole digital transformation of meeting the customer on the devices that they want to engage with, and obviously the pandemic is only accelerating that, is making customers needing to build a new set of applications with a new set of capabilities. And that's where I'd say a lot of the more modern app solutions such as document stores, key value stores, Spanner, which is relational but it gives you the scale and availability of non-relational, that's where those applications come into.

On BigTable we see a lot of usage in things like recommendations and personalization and targeting customers because BigTable is just so efficient at scaling at high-throughput volumes and collecting customer data. Spanner is being used by companies who need global transactional data. Uber today runs on Spanner. ShareChat also runs on Spanner and they needed to scale overnight by 5X. And so the capability of being able to scale without downtime in the cloud was very valuable

And then on the cloud native side, also Firestore, which is a document database, but it's really mobile-optimized with Firebase, helps organizations that are trying to iterate very quickly on their digital transformation and build mobile-first applications. So there're lots of choices out there. It really depends on the use case and the capabilities that customers need.

[00:05:53] JM: Maybe we could zoom in on some of those use cases there. So Uber, for example. You said Uber uses Spanner as their core transactional database?

[00:06:03] AG: Yeah. So Uber has actually finished transitioning all of their cities onto Spanner. What was really appealing to them, and they talked to one of our conferences, was the fact that Spanner has five nines availability. It's got the global footprint, but it is also fully transactional and has a SQL interface. So it's really the only database on the planet that gives you the best of relational databases but with very high-availability and near infinite scale.

[00:06:34] JM: Now Spanner has been around for a while. So I wonder like when you're looking at it from a management perspective, what kinds of features are you adding to it or incremental work are you adding to it? As a leader of the database product, how do you focus the energy of the team?

[00:06:55] AG: That is a great question. I think in the early days of Spanner we got the availability, we got the scale, we got a lot of those core capabilities that really made Spanner very unique. Where we fell short was we didn't really add all the features that enterprises needed such as backup and restore, point in time recovery, customer-managed encryption keys. And so what we saw was that Spanner was primarily adopted by digital native companies and not by the mainstream traditional enterprise. And then from a, I would say, SQL surface perspective, Spanner is SQL compliant, but it didn't have all the developer-oriented features that developers were looking for. So that has changed significantly over the past two years. We've eliminated most of those enterprise adoption blockers. We've added significantly to the SQL surface. And actually today Spanner is one of our fastest growing services, and that growth is actually being driven by financial services, retail. So a lot of the more traditional enterprises are onboarding now given the fact that we've removed a lot of those enterprise adoption blockers. And so the next two years is really about continuing down that path and making sure it's the most secure and best governed database and continuing to help democratize the interface to make sure that general developer population has an awesome experience with Spanner.

[00:08:22] JM: Okay. So another use case you mentioned is mobile-first. So Firebase and Firestore, I'm a fan of those products. I've used those products in a lot of my own projects. Firebase I think of is more of a platform. Do you oversee the Firebase product? Can you tell me how that is evolving and where you're trying to take it?

[00:08:43] AG: Yeah. So I don't oversee Firebase, but we work very closely with the Firebase team. So for those who don't know what Firebase is, it's a mobile development platform and it is used – The reason why it's not in my organization is because it really supports a lot of the

different Google businesses; Google Ads, Google Search, Google Maps, of course GCP and Firestore. And so it is Google's answer to rapid agile development for mobile applications. But Firestore is the document database that is best integrated into Firebase. And so when I look at kind of our customer base in Firestore, about half of it comes from Firebase users and the other half comes from GCP enterprise customers. That's pretty much what the breakdown is.

So a lot of our focus is working closely with the Firebase team because there're a lot of unique capabilities the Firebase and Firestore delivers such as supporting occasionally disconnected use cases, which is super important. Server-side push of data to the mobile devices, and then just a document model is just a lot more agile for developers, especially mobile applications where change is constant.

We continue to invest heavily making the Firebase/Firestore experience a very seamless experience, but then in parallel we're just seeing a general surge of interest in document databases. And so quite a lot of use by just in general purpose use cases of Firestore as a database.

[00:10:19] JM: So you mentioned these two categories of user, the digital native companies. Perhaps the companies that are born on the cloud and then companies that have a lot of on-prem infrastructure might be migrating onto the cloud. How do these two categories differ in their needs and what products do you try to offer them to satisfy those needs?

[00:10:46] AG: Yeah. So I think traditionally there's been quite some difference between these constituencies, and I'll talk about them, but I think what we're seeing is over time they're actually converging because the traditional enterprise is really digitally transforming now and trying to build those same digital experiences that the digital natives have built. And so this is why we're seeing this huge surge of adoption by traditional enterprises of Spanner and Firestore and BigTable because they're converging.

I would say historically the traditional enterprise, because they weren't as digitally-first, they did not have that scale that a digital native has. So they would be more focused on security, data

sovereignty, just kind of the traditional enterprise ilities that these customers would have. And they would tend to gravitate more towards a relational database model because that's where a lot of their applications were.

I think on the flip side, the digital natives tended to have much higher volume. And so really looking at getting the most available and most scalable systems have been most important. Some of the data sovereignty aspects are very important to them now, but maybe five years ago they weren't quite as important. And so they were really much more focused on operational excellence, price performance, availability, scale. And so we kind of had this bifurcated approach you know in these environments.

What we're seeing today is that they're converging. And so that's actually making it very exciting for me because we're seeing really great adoption of the more cloud native databases in the traditional enterprise, because as they're building these digitally-first experiences, they've got the same requirements around scale and availability that the digital natives have had.

[00:12:38] JM: Are there many users that you encounter that want to be multi-cloud? And does that create any kinds of interesting architectural challenges at the database layer?

[00:12:51] AG: So we definitely feel that multi-cloud is here and is here to stay. And as you know, Google has really been the leader in supporting multi-cloud deployment models. Actually it’s one of our core differentiation, is that we really are focused on what our customers want, and many of our customers do want to have a multi-cloud experience even if it's just for DR purposes. And so that's why we've worked on Anthos and an ability to really run Kubernetes across multiple clouds and hybrid clouds, and that has been extremely well-received by customers. And needless to say that databases is one of the next things that they ask for. And so we definitely continuing to think about how do we best support our customers on platforms like Anthos from a database perspective. And with some databases, the more traditional ones, like Postgres and MySQL, that's a bit easier. You already have open source operators in the community. But when you look at technologies like Spanner, which is

using atomic clocks and data centers, bring something like Spanner into a multi-cloud environment is actually not a trivial feat.

So we just continue to think about what our customers most need and then think about what we can do to best support them. That doesn't mean that all our databases are going to be everywhere, but we will definitely try and make sure that we give our customers good options either as a first-party or in partnership with third-parties so they can be successful in these hybrid and multi-cloud scenarios.

[00:14:26] JM: Do you spend time on the data warehousing products like BigQuery? Or are you completely focused on transactional databases?

[00:14:36] AG: So I'm completely focused on transactional databases, and then I have a peer who runs analytical databases. But in the past I spent a lot of time in analytical databases and he actually spent a lot of time on operational databases, which makes for a really great partnership, because I know his world really well and he knows my world well. So a lot of the conversations we have is ultimately customers don't want to have to self-assemble both the operational and analytical side. Customers are looking for solutions. They're looking for fast time to value, fast time to market. And so you're going to see an increasing amount of time and capability spent at Google to really make that journey for customers from generating their data in the operational databases to querying and analyzing them on analytics side as seamless of an experience as possible.

[00:15:31] JM: How deeply do you personally need to understand the mechanics of a database like Spanner, for example? Spanner is a deeply technical product. Do you feel you have to understand it thoroughly? Do you have to understand all the nuances of the atomic clocks?

[00:15:49] AG: So you've probably hit on our most complicated and hardest to understand database product. I'm sure that wasn't by chance. I have to have a pretty good understanding on how it works and how customer benefits are extracted from that and how the atomic clocks

impact transaction management and price performance and so on. But definitely I have folks on my engineering team that understand it way better than I do. And so my job is to know enough to help drive decisions and help make sure we have the right talent on board, but really we have a phenomenal engineering team. The spanner team is just top notch and has been working on Spanner in many cases for ten plus years.

[00:16:37] JM: Do you focus on the ETL process at all? Like the process of getting the data from those transactional databases into the analytic databases?

[00:16:50] AG: Yeah. I mean, thinking about the ETL process is very important. That's where me and my analytics counterpart, we spend a lot of time talking about how to make that easier for customers. We have products today that help customers get there with Data Fusion, Dataflow. We also have partners who are delivering easy ETL solutions. But definitely thinking about how we expose change data capture capabilities out of our databases and how we make it easy for customers to move that data into analytics is a constant area of discussion and our goal is to make it as seamless as possible for customers.

[00:17:31] JM: What's the biggest technical challenge that you're working on right now?

[00:17:34] AG: So I don't think there's one single hardest challenge that I'm working on. Our customers are continuing to push us to increase our availability, our scalability, our price performance. They're asking us for more features. They're asking for more simplicity and making the journey easier for them. So I'd say from an overarching challenge it is to continue to make sure that we put operational excellence and keeping our customers available and secure as our top priority. And there's always more we can do there. And so we continue to push ourselves to better for customers. And then at the kind of more innovation level, just continuing to iterate on solving specific customer use cases and making sure that we continue to get more efficient from a price performance perspective and making data management easier for customers.

[00:18:30] JM: Do you have any perspective on the ideal database to be using for machine learning applications?

[00:18:38] AG: That's a great question. I think from a training perspective, databases that can manage semi-structured data at scale, at low cost, and then create very useful materialized views for machine learning model training is extremely valuable. And we see customers using both BigQuery and BigTable for that because they're really optimized for those kind of workloads. And I think continuing to add capabilities in those databases that help customers very efficiently store and query training data is going to be critical. BigTable is often used at that kind of data fabric where customers are integrating lots of different data sources. And so that does become a bit of that live data lake style database that the customers use in the machine learning pipeline.

Then on the inferencing side, I think over time a large amount of databases are going to have inferencing capabilities within the database. Making that deployment easy and simple for customers is a challenge, and that's one that you know we're excited to solve in the future.

[00:19:56] JM: Speaking more about the present. Database migrations are a huge headache, either migrating from an on-prem database to the cloud or migrating from one type of database to another. Tell me about how database migrations can be improved or how they can be eased.

[00:20:14] AG: Yeah, that's a great question. And I think as I talked earlier about the three-step journey, we're definitely seeing that those 40 years of databases, they're all trying to move into the cloud as quickly as possible. Customers are looking to shut off data centers. And it is true that database migrations have never been easy. And it's not just the database, right? It's all the applications that are actually around the databases that also have to move.

One thing we've done it on the database side that we just announced was database migration service. We took a bit of a different approach than most other players. A lot of other players have basically done migration services based on more of a change data capture paradigm.

What we've done is we've used the native database replication capabilities. We've made it serverless. We made it super easy with a very easy kind of wizard-style experience. And so at least for MySQL and Postgres and soon SQL Server, we've probably made migrating the database itself as easy as it can get. And we've been in public preview. Just for a few weeks we've already seen thousands of migrations happening with the database migration service.

I think migrating the full application is still a challenge, but we do have capabilities that we've built out as part of our compute team that make it easy for our customers to discover the assets they have, the VMs they have and create a plan for how to move both the database with the applications into the cloud in as easy of a manner as possible.

[00:21:47] JM: So in architecting the databases on the GCP side, when you're building a database as a service – I mean, I guess this is another question that you may not have to be intimately familiar with, but can you tell me about the architecture of a database cloud service these days? Like how it uses Kubernetes or how it provisions disk? Just give me some insight into the mechanics of a cloud database deployment.

[00:22:16] AG: Let me kind of talk a bit about know why I think Google has been so unique in solving some of these big problems and in building out Spanner and BigTable. So the way we build our cloud-native databases is very different than what people would expect, right? Most people expect there to be a server somewhere with an attached disk. You bring up the database process and it's using that CPU, serving the request and everything is going to disk, right? That's kind of I'd say the traditional model of building these things. And we actually build our solutions very differently. We have completely disaggregated storage and compute internally. So basically very similar to Kubernetes environment, everything is built on containers. And this could be thousands or tens of thousands of containers. And actually storage is also completely desegregated. There is no local attached disk. Everything is disaggregated and we have very smart clients that know where to write the data to.

That gives us a lot of flexibility. It gives us great durability, great availability, but it also gives us compute and storage separation out of the gate just in how we built these databases. So as we

think about how to scale those environments, how to make sure they're available, we just get a huge amount of benefits from how these systems are built from the ground up. And with the differentiated networking we have, both know within a zone and also the fact that we own the global backbone ourselves, some of the problems around making sure we have the right SLAs between regions on replication and so on, because we're in full control of the network end-to-end, we really have a very strong ability to manage two SLAs.

So I would say if you asked me kind of what was the most exciting part of joining Google, it was the fact that we take a very different approach to how we actually build our systems bottoms up. That's where our ability of delivering products like Spanner that truly have global scale and are virtually unlimited in the capacity they can handle is because we're building it in a very disaggregated way.

[00:24:34] JM: And as far as making that disaggregated model scalable and fault-tolerant and resilient, are there a battery of tests that you run? Are there some architectural efforts to ensuring that it functions as intended? Because that can be quite a complicated architecture to make sure it's failure proof.

[00:24:57] AG: Yeah. I would say yes and yes, but there're actually some benefits to it that it removes some of the burden on you, and then there's some additional burden that you take. So if you think about durability as an example, right? In the traditional model, you own the durability. Meaning when you write a byte to disk, you have to deal with the replication and so on. Today when I write a byte or 4k or whatever to the disk aggregated storage, that gets automatically replicated for me across zones. I have checksums created. Like a lot of durability aspects happen that I don't have to design myself in my service, right? So there's a lot of, I would say, free goodness that we get that we don't have to worry about.

And in everything from like a bit flip or all sorts of edge cases, that just inherently get resolved for us by using the same common infrastructure that we can test once and we can prove correctness once and so on. So I'd say that Google has really done a good job of reusing those very core architectural components to make sure we have very, very dependable systems,

right? And actually the Google business also depends on those systems, right? It's not only our external customers.

And then as we build things on top, of course we have to then have the discipline of knowing what our dependencies are. Making sure we have good failure management, good monitoring, good testing and so on, so forth, end-to-end durability. So there is a lot of investment we make on making sure that the applications we build. And I think of a database as an application are resilient, durable and secure, but it does help us a lot that we have underlying primitives that are extremely well-tested.

[00:26:44] JM: I want to change the topic for a while. We did a show not too long ago with one of your your old colleagues from Zend, Zeev, who you collaborated closely with on PHP. I'd love to get your perspective on how web technologies have advanced in the last six years, seven years, since you were focused on PHP and Zend technologies.

[00:27:12] AG: Great question. Yeah. So Zeev has been my partner since the late 90s and he's awesome and we've worked on PHP together. We built the company together and of course we're still in touch on an ongoing basis, and that was really a fun and enjoyable journey. PHP and web in general, it's kind of interesting, when I started in 1997 working on PHP, Cold Fusion and Perl I think were kind of the leading technologies, a bit of ASP also, but the web really hadn't taken off. It was really at its kind of infancy. And so it was very exciting as we put out PHP3 that we saw that huge traction. And then companies like Yahoo and Facebook kind of building up PHP and really becoming successful. The thing that kind of kept PHP very popular was the fact that we consistently evolved it as new standards came out. XML came out, SOAP came out, REST came out, non-relational databases and just making sure that the out-of-the-box experience is helpful.

I think where you know things have changed over the past few years, I think in a few areas, first of all applications are becoming much more sophisticated, right? We're actually building kind of the thick client experiences now in the browser or in the mobile apps. And so whereas PHP in the early days was mixing presentation logic and business logic and of course introducing

frameworks with an MVC pattern, address some of that separation of concerns. Today I think we're increasingly seeing that the backend is really only focused on the transaction management and business logic and that all the presentation logic++ is happening on the client side. And that's also driven a lot of interest of course and know how in JavaScript, TypeScript and a bunch of kind of frontend frameworks.

And so I think what it's done is it's basically made the backend, whether that's PHP, or Node, or Python, right really more focused on some key primitives that interoperate with the databases and other data sources to serve the frontend. PHP continues to be very successful, I think, on the web, but no doubt that you see kind of some of the more event-based models like Node taking on what I'd say the use cases that are a lot simpler that don't need a huge amount of business logic at the backend and can benefit from the scale that an event-based model can handle. And actually PHP also now has a number of event-based models so you can actually use it outside of a web server context and get that higher throughput and scale out of PHP.

[00:29:58] JM: What do you think about what Facebook and Slack have done in terms of really scaling PHP and building additional languages on top of it?

[00:30:12] AG: I was very close to the team at Facebook as they were working on what they called HipHop at the time, HHVM. It was interesting because they put a lot of effort into a just-in-time compiler and they were able to get better performance, but then what we discovered as we were looking at it that the real issue was actually memory consumption. And so we put a lot of effort into – We reduced the memory consumption of PHP by about a half, and without a just-in-time compiler we actually ended up being faster on most real-world applications than Facebook HHVM. That was also one of the reasons why they actually stopped investing in PHP compatibility there because they suddenly realized that that technology was not the one that was actually going to displace kind of the mainstream PHP. And then of course now in PHP8 we've added a just-in-time compiler to PHP8, although I would say that in real-world applications like content management systems, web apps, the impact is still de minimus because these applications tend to be a bit more I/O and

memory-bound. But if you want to do some hardcore computing, then of course you get a significant boost. And so I think today Facebook is primarily focused on supporting Hack, which is their variant of PHP on the HHVM platform.

[00:31:37] JM: Taking you through more of your career history, you have spent some time at Amazon Web Services before going to Google. And I'd love to know, how do you think the perspective of AWS compares to that of Google Cloud in terms of how cloud services are architected?

[00:31:59] AG: Yeah, great question. My higher level view is that we're still early days in cloud and all of IT pretty much with only some exceptions are going to move their workloads to the cloud. So it's just a huge kind of market segment and a number of companies are going to be very successful. So I think from that perspective every company has its own nuances on how it differentiates and what it does. What excited me about Google and one of the reasons I joined was the fact that we kind of embraced multi-cloud, right? Really kind of we're waiting to be very open from that perspective. I think also how we architect our services. We talked a bit about the compute and storage separation and disaggregation is quite unique in how Google architects these systems. And then some of the other infrastructure components we have are quite unique. So I think both companies are obviously doing super well and we're kind of all in it together to help customers transform, but I did get very attracted to kind of some of the key pillars that we're focused on the Google Cloud to help our customers on their journey.

[00:33:14] JM: In the, last let's say, 10 years of cloud service development, there has been this I would say co-evolution of open source and cloud productization. So you have an open source product that gets developed and then cloud service that gets turned into a hosted version of that open source project. And you oftentimes have multiple vendors that serve that open source version as a product. So you can take examples like Elasticsearch and the company Elastic, or Kafka and the company Confluent. Do you see this as an ongoing model? Do you think this is something that will remain the same? Or do you think there's any reason why the current model of open source productization will change?

[00:34:06] AG: Yeah, that's a great question. I'll give you my personal opinion. This is not Google's opinion, right? Just being contributing to open source since 1997, I dedicated a huge part of my life to contribute to open source, and not only PHP, also contribute a bit to MySQL and some other open source projects. I'm a big believer in open source to me it was always very exciting that huge communities were collaborating. Kind of making source code and capabilities available in an unfettered way to companies and individuals. And so what I liked about it is it's really kind of a self-rewarding kind of cycle, right? Because every product you look at, whether proprietary or open source, is actually built on a lot of open source software, right? So if you take Elasticsearch as an example, it's built on Lucene, and Jackson, and Java, and Nettty, and you name it, and you take MongoDB. MongoDB is also using open source.

So I'm personally very excited about kind of the flywheels that open source creates. I also think that taking a proprietary approach to software development model is absolutely fine. So I don't think one is better than the other. I just think that it's very important to kind of make sure that when you work with communities, you keep the trust. What's open source is open source. What's proprietary is proprietary, and that you kind of keep things very clear. So customers can make the choice, and a very clear choice of what they want to consume.

By the way, just to answer your business model question, if you actually look at some of the very successful open source businesses, right? They all ended up being very successful, completely unrelated to any licensing changes they made, because the success actually in most cases came before. I don't equate licensing as much to the success. I think that's actually a bit – People overly focus on that. I think a lot of it is about are you delivering value to customers? Do you have a model for how to deliver that value? And are you differentiated, right? And that differentiation can come from features. It can come from how you operate a service. It can come from how you support customers. I don't think there's a one-size-fits-all model, but I do think a lot of the discussion around licensing is actually not directly correlated to value creation.

[00:36:41] JM: And one thing I think was misunderstood as this debate has been developing is that it's just very hard to build the best version of these products. If you look at Elastic, the

company, or if you look at Confluent, the company, they build versions of their products that are considerably better than the offerings from really big clouds, whether it's Google or Amazon. If you take a company and it's entirely focused on delivering a product experience, it's pretty hard to beat that company at that product experience.

[00:37:22] AG: Yeah. I think about it probably a bit differently, although I agree with what you said, but the way I think about it is, look, the IT industry is huge and all these market segments are huge. And the reality is that customers have different needs and buy for different reasons and choose to use open source, or use proprietary, or ask someone to manage it for them for different reasons. And so the worst thing that can actually happen to open source is not to be consumed, right? I mean, that is guaranteed to be a small pond. And so I think the bigger the pond is and the more customer use cases can be solved through open source, the more that open source actually thrives.

So I just think in general it's not a zero-sum game and there're lots of players that can be successful. I actually think Elasticsearch is a great example, because if you look at the actual results, Elastic is doing really well, right? They've built a really good product. They build an awesome company. They're a good GCP partner. They're doing super well. And then you've got lots of other companies. By the way, forget Amazon for a second. Just like ISVs and smaller companies who've been using Elasticsearch and they've built great businesses also. So .NET is Elasticsearch has been a net ad to the IT industry in lots of different dimensions. And what's really nice about kind of the flywheel of open source is, as I said earlier, a lot of these earlier on open source projects like Lucene and Java and Netty have enabled that kind of innovation.

[00:39:04] JM: All right. Well, as we begin to draw to a close, I'd love to get your perspective on the future. Any thoughts on how cloud services are going to be changing in the next five to ten years?

[00:39:15] AG: Yeah. I mean, I think we're definitely in the data-first world. I think data is at the core of digital transformation and we're seeing a convergence of the operational and analytical

side as customers want to make faster decision and better decisions in real-time on data. And so I'd say there's a strong convergence of these worlds. Data scale is growing significantly, which obviously put some price pressure and scale pressure on customers, and that's a really exciting problem for us to solve because we know that we have to solve the price performance equation for customers. Of course machine learning, which is only as good as its data, is continuing to evolve very quickly. And the thing that we're really excited about is how do we help our customers reimagine their businesses with data, with machine learning, even with some of the other Google services that we have such as Google Maps?

And so I'd say as you think about five, ten years from now, and the pandemic has of course accelerated this, it's going to be very hard for any business to be successful if it hasn't digitally-transformed. And so I just think that this next five to ten years is going to be probably one of the most exciting times in IT where businesses are really focused on innovation. And this is where I think cloud providers and partners can really help where we help our customers not deal with the infrastructure and not have to kind of do a lot of the heavy lifting themselves, but rather focus their limited resources on innovating on the business model, building new applications that helps their customers in new and differentiating ways.

[00:41:09] JM: We touched on something really interesting there that I just want to dive a little bit deeper into, but the convergence of transactional and analytic data. And maybe you could give me real quick the role that the database plays in terms of a data platform. I've done all these shows about data platforms and data lakes and data warehouses and data transformation systems, and it's an area that is still shifting a lot. So I guess I'd love to get your perspective from the point of view of somebody working closely on databases.

[00:41:41] AG: It's a great question, and I think you know Google has had a bit of that benefit of having been there probably 10 years before a lot of other companies or maybe even more than that, because they kind of scale, the precision also that was needed in the Google business led to innovations that ultimately led to Cloud Spanner and BigQuery. Because we saw that there's a high level of transaction processing happening and you still need to – Many businesses are still going to need those transactions. Making sure their data is durable, but

doing at a whole different scale. But then I think that the piece that is different is how that feeds the real-time decision making, right?

In the past, I mean, you would kind of refresh your data warehouse every 24 hours with ETL jobs. Or if you were really adventurous you do it every hour, but you weren't able to act in real-time on what's happening with click stream data or shopping carts or other data sources. And so now you're kind of getting into this interesting point where you suddenly need transactional systems that are actually petabyte scale, their kind of data warehouse scale, which Spanner by the way can do. I think it's the only OLTP-focused database that I think can do this at petabyte scale. But then you also want to run these petabyte scale analytics in real-time.

And so this is where I think these worlds come together. You're still going to have some level of best of breed, right? So BigQuery is going to be the best of breed on I would say large scale analytics, while Spanner is going to be best at OLTP. But the more we can make that experience seamless for customers, the less they're going to have to worry about OLTP, OLAP and what database choice they make. The more they're going to be able to just focus on the application they're trying to build, the transactions that are coming in and then the insights and machine learning that they're driving on the other side. So I think that's where a lot of the exciting innovation is coming in right now.

[00:43:46] JM: Well, Andi, it's been a real pleasure to talk to you. And you've been around for a while. You've had a lot of amazing experiences. So thanks for giving me your perspective.

[00:43:55] AG: Yeah. Thanks a lot for having me. I really appreciate it.

[END]