EPISODE 1181

[INTRODUCTION]

[00:00:00] JM: When Tim Wagner worked at Amazon, he invented AWS Lambda. After working on the early serverless infrastructure, he joined Coinbase and worked as vp of engineering. Since leaving Coinbase, he has started a new company called Vendia. Vendia combines his learnings from the serverless space with the innovations around blockchains to work on the problem of data sharing. Tim and David Wells join the show to discuss what they're working on with Vendia.

[INTERVIEW]

[00:00:34] JM: Guys, welcome to the show.

[00:00:36] TW: Hi there. Welcome to you and thanks for having us on today.

[00:00:40] JM: So you both work at Vendia, and before that you respectively worked at AWS, Tim. And, David, you worked at Netlify. Those are pretty cutting edge companies doing a lot in serverless and cloud computing. How did you respectively come to the problem set that Vendia is working on?

[00:01:00] TW: I can kick us off here a little bit. So CEO and founder of Vendia. Had been at AWS for about six years where I started what's now known as Lambda. Way back in the day when it was just kind of a six pager on how can we make the cloud a little bit easier for folks to use. And then after growing the serverless team there, had been down in the Bay Area for a year as the vp of engineering at Coinbase. So got a chance to also see some of the distributed ledger technologies kind of in play and maybe some of the places where they actually don't play quite so well.

So Vendia is in some sense for me the culmination of those two things. The name stands for as a kind of a contraction of Venn diagram. This idea of the coming together of distributed ledgers and serverless technologies together. So that's a little bit of the path by which the journey by which I got here. We're very excited to have David as part of the team. I'll let him tell a little bit about his own sort of Vendia origin story.

[00:02:04] DW: Before working at Netlify, I actually was working at Serverless Inc., and that's actually where I kind of started dipping my toes into this whole serverless world using Lambda functions, etc., all the stuff that Tim kind of pioneered. And yeah, I've always been focused on like developer experience and making tools as accessible as possible. So after working at serverless on the serverless framework for quite a while, I switched over to Netlify where they have like a really nice kind of onboarding flow to the serverless world as well.

But yeah, I heard Tim was working on this kind of new kind of idea with distributed ledger tech, and at the time as well I've been doing a bunch of research into just how you can use the kind of embarrassingly parallel serverless compute as kind of a new way to do things and a new way to do things at a massive scale kind of like a fraction of the time. So yeah, I joined up over here of Vendia, and that's kind of how I came over here.

[00:03:13] JM: Tim, can you go a little bit deeper on the problems that Vendia is trying to solve?

[00:03:17] TW: Sure, absolutely. And I'll kind of kick this off by saying just like I think the word serverless has had an interesting amount of debate as to whether or not that's actually a good moniker. We feel a little bit about the same way about blockchain in part because there're some great ideas there, tamper-proofing, distributed ledgers, information and data sharing. It's also kind of been associated with maybe some of the darker side of crypto and certainly some of the performance challenges and some of the failed enterprise experiments. So we can chat about all of those pieces.

But the one thing I would say here is that no enterprise has a blockchain problem, but every enterprise has a data and data sharing problem. Technologies like electronic data interchange, or EDI, at this point go back almost half a century. The problems of sharing data and data silos and information consistency certainly predate computers. Enterprises have been struggling with how to work with their partners, with their customers. How to share accurate and up-to-date information since pencil and paper was kind of the reigning technology here. And so all of that is in some ways is a very old problem. Think about things like TP monitors that were kind of classic enterprise solutions for many, many years.

So that's the part that's been interestingly different of late. There are all these incredible disruptive technologies now. Obviously the public cloud has changed things. But then even within that, these new modern ideas about distributed ledgers, the advent and the escalation of serverless and all the things that it's capable of doing. Not just the sort of the original idea, but all the things that AWS and the other clouds have continued to add to that and grow to that set of capabilities. And now with Vendia, the idea of kind of bringing the capabilities of both of those things together.

So we help customers with everything from supply chain optimization to being able to, in the case of one of our companies, the Best Friends Animal Society, actually helping to get pets back to their owners. So being able to kind of match up where a dog or cat might be in a pet shelter versus all the people who are looking for those and create a single consistent up-to-date database of information even though there are thousands and thousands of pet shelters spread all over the United States and even within large cities, could be many dozens of places that an animal might mind end up. To things like a European auto manufacturer who's using our technology to track part and chassis assembly as it kind of passes from hand to hand and kind of agent to agent. So that if there's some kind of problem or damage detected, they can work back to who was responsible for that and ultimately who should pay for that. So lots of places where people are trying to share either code or data or both across clouds, accounts, regions and technology stacks.

[00:06:12] JM: So it sounds like a shared database, which is what I've heard from a lot of different kind of blockchain vendor companies. Tell me why is this different than just a shared database.

[00:06:24] TW: I mean that's actually a great way to think about it. When you walk up today to a technology like Oracle or even something perhaps that's kind of cloud native like Amazon Aurora, you bring a data model, in this case in the form of a SQL DDL statement. You give that to that service or to that enterprise application and then a few minutes later you end up with a table, a database that's capable of giving basically one account, one company, one account, one region centralized storage. And Vendia does something very similar, except when you give us a data model, what you get a few minutes later is a database that can span companies, clouds, accounts, regions and technology stacks. And so that's kind of the beauty of the decentralized mechanism here. So that lots of different people can potentially participate in that. And different people doesn't necessarily have to mean different companies. Although when we think of things like Ethereum, that we often think of lots of different people kind of participating in it. In the enterprise case it could also be multiple departments. It might be an automaker that has IOT sensors in a factory in one country but needs to process and analyze that data in a different cloud center, in a different country, and just needs a simpler way to move that data and keep those things consistent and operate on them across departments, accounting units, cloud regions and so forth.

So there's a whole spectrum out there of enterprise use cases that require information or code to kind of operate over these disparate regions, accounts, clouds and so forth sometimes within a company, sometimes across companies. But yeah, distributed database is a good way to think about it and you can we can chat about smart contracts and code on top of it. But at the base layer is always a shared consistent up-to-date, as we like to say, acid representation of data that can be kept the same everywhere.

[00:08:24] DW: Yep. And kind of immutable aspects of kind of similar blockchain technologies where you have a full transaction log of everything that's happened. Who's done it and what they've done?

[00:08:39] JM: Well, tell me more about what you might want to build on top of this shared database model.

[00:08:47] TW: Sure. I think maybe David and I can each take a stab at this, because we do have people who come at it from different perspectives. You can think of it as just an easier, more complete, more kind of application ready way to build a great serverless app. We also have a lot of customers, kind of our more top-down sales model where we've got an enterprise that is looking for – Has a particular problem to solve and is looking for uh looking for a solution.

One of our customers, for example, in the travel industry settles airline flight payments and associated kind of ledgering information, ledger in the financial sense in this case. So we help them out with a single consistent representation that can span different airlines, different payment and financial institutions as well as just basically keeping now one set of books consistent, correct and always up to date. By doing that and doing that quickly, you can cut down on the amount of money or the amount of – Sort of the amount of money that has to be carried there over time. We do the transactions more effectively. Everyone has a consistent, always up-to-date view of that information. That both lowers cost and increases the monetization opportunities.

You can imagine, for example, with the airline loyalty programs, how having that up-to-date information both helps with fraud detection and prevention but also with the opportunity to give customers the kinds of opportunities that those respective companies are looking to offer them especially at a time when the travel industry is looking to monetize the recreational and opportunistic upsell in that case. And so we're a piece of plumbing for all of that, right? We don't build the application piece of the stack. We're the part that makes sure that hundreds of different airlines, different financial institutions and some of these settlement agencies in the middle can share that information in a way that is trustworthy, secure, reliable and always up to date. Replacing a lot of these very older school, kind of DIY systems or ERP systems that tend to be expensive, slow and require a lot of a maintenance. I mean let's face it. Most of them

were invented 20, 25 years ago before the cloud was even a concept. And, David, let me hand this over to you because I think it would also be useful to take a look at the developer experience and perspective here.

[00:11:07] DW: Yeah, for sure. Another like really common use case that we see, and you see this across like multiple iterations of the blockchain, is supply chain management. So when you think about it, you order something, comes overseas, it travels through so many hands and there's so many desperate processes that goes through that we're even talking to some companies that they're still using paper, handing around pieces of paper. And you can imagine that this is costly, takes time. It's prone to human error, and there's a lot of chance for forgery and stuff like that. So that's really one of the big use cases that we're seeing. And the way that we're approaching that with what we're building at Vendia is – So instead of every intermediary in that chain, having to have an IT team with a DevOps infrastructure setup to manage and maintain their own blockchain in their own stack wherever they might be running their stuff, we actually provide a kind of a turnkey solution. So anyone that – You basically hand us your data model. So what is actually passing through will actually spin up those stacks Tim mentioned. And anyone can join the chain without needing to basically be an expert in any one particular cloud.

So what we're trying to do there is really take that paradigm, that kind of the serverless word and paradigm or shift brought up was, “Hey, we want you to focus on your core competencies, not basically the low-lying infrastructure.” So when you use these different pieces of serverless technology, you don't need to worry about patching your servers, scaling your servers, all the security implications of that. So really what we're trying to do is make that onboarding experience much simpler for developers and companies to adopt. Because one of the use cases Tim mentioned is the Best Friend Society, and there're a lot of animal shelters that are not very technically sophisticated. And we want them to have access to the same data as everyone else has without needing basically an AWS cert, right?

So that's really how we're approaching it. Basically making it very simple and anyone can join the chain. There's multiple ways that you can consume the data. The layer that developers

actually interact with is GraphQL. So if you're familiar with GraphQL, you can consume and put stuff onto the chain. And then from that, on top of that, these different use cases can build different clients for the – So going back to the supply chain example, you can build multiple clients for multiple different use cases. So truck drivers might have an app that they see a particular part of the data on chain versus the car manufacturer need a different UI, a different experience around that. So we offer that kind of base layer, a GraphQL interface on top of their data model. And then underneath that is the typical blockchain. Well, it's not typical, but a blockchain-like setup where you have that immutability and you can verify everything on there.

[00:14:35] JM: Can you guys tell me more about the underlying engineering you've had to build to create this platform?

[00:14:44] TW: Yeah. There's a really interesting innovation that we've had to create here in order to make Vendia possible. So one interesting place to start here is kind of think about kind of what was missing from some of these technologies. So serverless tech, things like AWS Lambda are awesome. But you know as we all know, they're specific to a particular cloud. So you're building a Lambda function. You don't necessarily have an easy way to also hook that up to some data that might be coming in that you might have stored on, say, Google Cloud, or to exhaust it to maybe something on Azure where you need to take action on it. And that might not be a big deal for a startup, which could be built entirely on a single cloud. But when you get to the fortune 500, through acquisitions, through growth, they generally have a multiple clouds whether they like it or not. They certainly have partners and customers in some of those cases, which are going to necessarily also make the multi-cloud.

So part of what we were gunning for here was take some of the best of serverless and help people get to it from wherever they happen to be and help them reach out to wherever they need to go. And that's a little bit of a different multi-cloud strategy than kind of the single cloud serverless efforts. And the thing that was key to that was having a distributed ledger. So we sort of filled in that missing piece of having a data model for things like Lambda functions. And then also using that data model, kind of the trick was use that data model to also reach out across to the different clouds, because it gives us an obvious way that we can have an acid

level representation of equivalent, consistent, correct data in all the different places that that needs to be.

So at the core of Vendia is a – Think of it as a blockchain built by someone who grew up in the cloud. So it's built, no servers, no containers, built entirely out of managed services. So things like Lambda functions, AWS step functions, on the AWS piece of this for instance. Things that have built into them, the fault tolerance, scalability, kind of managed outcomes that also let us very easily scale this up to thousands or even tens of thousands of transactions per second. And that was one of the big challenges here, was we looked at all the existing tech out there, Ethereum, Hyperledger Fabric. In the case of the public chains like Ethereum, 14 transactions per second is just not going to cut it for the enterprise.

And then all the hacks that people do, the side change, the off chain work, it kind of blunts the value, the impact or even the correctness of that. And we don't want any of those outcomes. We wanted something where all of the data all of the time, all of the code all of the time could operate on chain whether that's data – Classic scalars in a database or more blob and file-oriented information stored in something like S3 or Azure Blob store.

So we built something that allows kind of all of those pieces to stay consistent together, fully cloud native, kind of in the cloud, for the cloud, by the cloud, and yet has all those properties of a blockchain. So we do a multi-phase commit at the heart of this that keeps all of the different representations in sync. We use a NoSQL database to store the scalar side of the data. We use blob stores like S3 or Azure or Azure store to hold the unstructured data. And then we do the tamper proofing by combining the information from all of those pieces. So think of it as kind of being able to construct a Merkle tree across all of the scalar data as well as all of the blob data and put all of that together so that you have a single tamper-proof ledger that speaks to it regardless of how we've chosen to store it in the cloud.

And then because it is in the cloud, the scalability and also the ease of access makes it very easy for us to turn this into an application platform, because the other side of this picture isn't just, “Can you have a tamper-proof ledger? Can you write a smart contract as a Lambda

function?" It's also, "Can a developer use this effectively?" And for that to happen, you need things like S3 or like Lambda functions that are really easy to use, that have the right operational properties, the right scalability properties and then could also integrate with existing enterprise solutions.

One of the challenges we found in practice, and this was certainly true in my experience at Coinbase, like running an Ethereum fleet, it's expensive, it's time consuming, it's personnel consuming. We wanted to deliver something in a SaaS style venue that would be really easy for people to get up and running and could accommodate enterprises and SMB customers that might be as small as a few people in a startup all the way up to the fortune 100, because they have such varying IT needs in terms of the fluency and the operational support. And so that was kind of one of the guiding principles of our architecture.

David, I know you've also kind of been front and center of a lot of this. Maybe add in there to that kind of your own story here and some of the ways in which we make that happen.

[00:19:48] DW: Yeah, definitely. I mean we're definitely dog fooding everything that we're doing. A lot of the stuff is we're building mainly in a serverless fashion using a lot of the AWS CDK. Yeah. Almost every service we have – Well, I guess every service we do have is a serverless service. And yeah, that kind of gives us this kind of flexibility. We're doing some interesting stuff with how we build out this multi-tenanted model. But yeah, it's fun thing to build. And I think Tim wrote a blog post a while back. I recommend people go check out on just like the scalability concerns around traditional blockchains and how he was kind of thinking about this when him and Shruthi started Vendia. We're trying to address that bottleneck of the transactions per second. And by leveraging these cloud services, we're really addressing that and we're seeing some promising results.

[00:20:55] TW: One of the nice things about that performance and scalability is when you switch from kind of a server to serverless you can escape the limitations of a single machine. I think a lot of people wonder or may suspect like maybe Ethereum is slow because of some complicated math problem that hasn't been solved yet. Like it has nothing to do with that. At

the core of it, it's a peer-to-peer data sharing network that runs on PCs basically. And so once you hit the memory, the CPU, the network, the nic limits of a single machine, Ethereum can't – And hyper ledger fabric or even some of the more recent chains. Like none of them can scale beyond that, because that's inherent to their architecture. And as we all know, a single machine is never how you build a distributed system, right? True, scalable distributed systems like Lambda run on millions of machines.

And so we said to ourselves, we need an architecture that puts a supercomputer into every node of a blockchain, and that's where serverless plays an important role here. It's not just that it makes our company go faster, makes us able to do more with less, although it's certainly all of that. The key architectural outcome is that it let us put essentially infinite power into every node of a blockchain by building it that way. And that was kind of a revolution, right? That's the thing about Vendia that's unlike any of the other similar distributed ledger concepts out there and gets us around these inherent storage, compute and network limitations.

[00:22:19] JM: So were those the limitations of the other shared distributed ledgers? Was it a bottleneck in the speed and transactionality? I thought it was more kind of a difficulty in finding the right customers, finding the right product market fit. I mean I remember talking to – Like Microsoft had a pretty big industrial blockchain or what is it? Shared between industries blockchain initiative. I mean it always seemed like the customer demand was not really there. Have you found enough customers that are interested in the kind of data sharing that you're building?

[00:23:01] TW: Yeah. What we found is there's no shortage of interest in building better and more effective ways to share information across companies. One sort of in-market alternative and encumbrant that you can look at there is SAP, right? SAP's business is this ERP, a big chunk of which is helping companies share and process information that kind of comes and goes in and out of their four walls, right? The supply chain information, logistics information, partner information. In some b2b cases, also customer information. That and some of the large kind of cross data sharing applications even the kind of things that Snowflake does today, that's a collectively speaking tens to hundreds of billion dollar market. The amount of money

and time that people put into building and buying API solutions that sit on the surface area of their enterprises and then all the challenges they have around trying to make what's flowing on those APIs look like you know consistent, correct, up-to-date information with their partners upstream and downstream. Those are huge markets today, huge amount of tam that's out there.

What I think has happened is that the first generation of blockchains spurred a lot of interest as a potential way to tackle some of those problems and then also kind of suffered an equivalent level of burnout when the TPS wasn't there, the difficulty of deploying those technologies became kind of obvious and prohibitive. I mean even just figuring out how to scale a file storage on the Hyperledger Fabric is challenging. Even a managed service like Amazon Managed Blockchain I think has something like a couple of hundred steps to get going. And that's and that's the easy button solution.

So all of that is obviously – It's prohibitive and it's a challenge for an enterprise that's looking to deploy that and turn those technology concepts into solutions. One of things that I think we did that was interestingly different was try to get closer to where customers are today in terms of both their needs, their problems, but also their existing technology solutions and offer them less of a kind of DIY gap between where the tech is and where their application solutions would have to start.

Obviously, as you said, you see a lot of that kind of early interest in blockchain kind of sparked. And so a lot of it fizzled out. It’s sort of that trough of despair now. I would say as an entrepreneur, I'm actually very interested in the troughs of despair because they are often where you can build value and then actually get that value to write up not on hype, but on real actual production deployments. And we've been we've been uh delighted to start signing actual contracts with actual customers and get those great proof points that there is in fact a need out there that if you provide the right product and product market fit that customers will indeed be interested in it.

[00:26:10] DW: I'd add that the onboarding ramp of kind of the blockchain as we know it thus far, it's a pretty steep one, right? Like if you want to do smart contracts, you have to write in a new foreign language that you don't necessarily know. So those are the type of things that we're trying to address. So like with Vendia, you can write a smart contract in any language that you want so you don't have to learn a new syntax and way of doing things. So we're trying to smooth over the rough edges so people can adopt these things. And the implementation of this stuff takes weeks instead of many, many months or years to actually do. And then every new partner being onboarded, that kind of expands. So we're trying to make that as easy as possible again around the developer experience.

[00:27:06] JM: What would be an example of a smart contract that somebody might want to write I guess on top of your platform, on top of the data sharing network?

[00:27:14] TW: Sure. I think a lot of us kind of hear about smart contracts and end up getting familiar with them in the context of Ethereum and solidity and therefore start to think about things like Uniswap, kind of complex financial derivatives, kind of crypto related outcomes. But a better model for this and I think a great way for people to think about smart contracts and really understand them is database triggers. If you look historically, one of the reasons people wrote PL/SQL and wrote database triggers is because they wanted applications or at least parts of their applications to operate or expand upon a consistent view of the data rather than having to pull for that data periodically or maybe operate on an inconsistent or incomplete representation of it. And that's the way you should think about smart contracts especially from something like Vendia, is the modern version of a database trigger now just a database that can be decentralized, replicated and distributed across clouds, across accounts, across regions and yet still have all those properties that you would want in a database trigger. That's one of the reasons why we try to get away from these kind of more arcane and specialized languages and say what matters is not that it's written in solidity. What matters is that the inputs to a function, in this case a cloud function like AWS Lambda, that the inputs to those come from the chain or they come from the blockchain or the distributed ledger. And if it computes something interesting, it can return them, it can return them back again.

And so you know anything that would be a natural part of kind of the data or close to the data processing layer then can become a smart contract in that world. So let's say you're in the airline industry and you're doing loyalty programs. You want to put some policies in place. Maybe that say you're somebody who is a member of your loyalty program can fly on airline A but not on airline B. They can fly on airline C but they only get 50 credit when they do so. And you can take those policies from a business, in this case a travel oriented business, and turn those into a consistent, computable representation that looks like basically policy enforcement without having to worry that you'll be computing on old data, without having to do nightly batches or work on spreadsheets or, God forbid, pick up a phone and call someone to check on something and turns all of that back into an automated system.

We've also had customers of Vendia who've used smart contracts to do things like SLA enforcement on logistics deliveries. So if a delivery doesn't get to its intended recipient within four hours, the person doing the delivery pays a penalty or fine, and that happens automatically now so they don't have to go back and call in the lawyers, call in the accountants, call in the engineers to figure out what went wrong. It's all computed off a single share of representation of the data.

So policy enforcement, application data layers, things that look like shared SLAs and automated workflows, all of those are fantastic candidates for representation as smart contracts. And by lowering the barrier to entry to being able to write and read and operate those contracts, we make it possible for all of those parts of existing enterprise applications to now have a shared and consistent view of that data rather than having to be either shared publicly, which obviously could be very scary or written into something that is obviously hard for developers to understand.

[00:30:45] JM: Tell me more about what you've seen in the early customer deployments of Vendia.

[00:30:50] TW: Sure. I can give you a little bit of some insight. One of the things that we've had the great fortune to be able to have Slalom as one of our delivery partners. And they've been

working with us on several of our early customer accounts. That customer journey often looks like as typical for an enterprise sale. Looks like usually a pilot or a proof of concept. One of the things that we try to make, and it's been a very I think delightful experience for our customers, is come in, help them put together a simple schema, a simple data model representation and then built them a working production system in the span of five to ten minutes. And for someone who maybe has been struggling to deploy your Hyperledger Fabric or a homegrown solution for once, that's a refreshing change of pace. So those pilots and POCs often run very quickly for us within a week. We can usually get the customer to see some initial proof of value to learn about the system to get excited about that. And our delivery partners like Slalom are fantastic and also helping to bridge, kind of close the gap between Vendia's platform technologies and the customer use cases. I think something like the Animal Society where we want to be able to represent what the dogs or the cats kind of look like on chain. Some of the information about them, the state that they're in, where they are in terms of the shelter workflow and process and then ultimately help build an application that can allow people to find pets, to help in the pet recovery system and of course hopefully get those pet owners to be able to reconnect with their lost pets and get them back to where they belong. So that often looks like POC or pilot, then a production implementation which can range anywhere from you know a few weeks to a few months depending on the complexity of that, and then at that point the system is in production.

One of the nice things about having built this on a serverless core is that we can offer customers a very simple clean experience where they have almost no operational overhead. And we can do that in a cost effective way inside of Vendia because serverless itself is such a low operational footprint for us. No servers to worry about no. No operating system updates to deal with. So that we're able to provide a SaaS experience even though this is a deeply infrastructure-driven sale and a deeply infrastructure-driven integration experience. And that integration for customers will typically look like you're hooking up either web and mobile apps. And David can speak a little more to kind of what that experience looks like. Or on the cloud side and the backend systems, perhaps things like pub/sub, streaming data or other mechanisms that they might have today to get the data either into that system or exhaust it from the system in order to take more action on it.

David, you might want to take this from a little more of a DevEx perspective here, because one of the things we also do is stand up some of the GraphQL capability to help enable web and mobile development.

[00:33:56] DW: Yeah. So I mentioned this before, but yeah, so you give us your data model, right? So you give us your JSON schema. With that data model, we will generate a GraphQL backend for you. And we're working on automatically generating client SDKs to make the development experience even nicer. So once you start actually putting data through the blockchain, you can query that, whether it's the actual list of transactions or like list all dogs or whatever it might be, whatever your data model is. You can do that via the GraphQL interface. So you get kind of the nice like auto completion of graphicool type experience and it's pretty straightforward to actually build out clients from that.

Additionally, with the GraphQL kind of interface, you have the ability to subscribe to changes. So if you want a real-time application that's listening to updated or new transactions through your system or new dogs or cats added to the system, etc., you can do that as well in the client interfaces that our partners like Slalom are building out for our end customers.

[00:35:16] JM: So, Tim, I'm just curious. If you talk to the OG blockchain people, who I'm sure you have you know plenty that you're friends with because you worked at Coinbase. And you tell them about kind of the private ledger or semi-private ledger startup you're working on, are they skeptical? Because I find that like when you talk to blockchain purists, they tend to be kind of skeptical of private blockchains. What do they think of Vendia?

[00:35:48] TW: No. It's a great question. And look, having also been a part of the crypto community and working at Coinbase and understanding some of the philosophy there, there are certainly folks on that side of the fence who look at enterprise blockchains and enterprise technologies like Hyperledger Fabric and Vendia scans in this in the same way that the converse is true here. I think when you get past some of the theology, if you will, of public versus private or permission versus permissionless, there are some really interesting

technology choices and options under the covers there that actually make for really interesting concepts. For example, it's not the case in Vendia that customers simply trust us with their data, right? One of the things we didn't do and can't do is simply stand up a database in sort of classic SaaS fashion if you will and just take all the customer data from all the different enterprises and slam it all together, right? That doesn't meet our customer expectations. That doesn't meet their compliance expectations. That doesn't provide operational isolation. We still create an individual account for every customer. All their resources are completely isolated at the cloud account level from all other customers, and that requires us to still build a decentralized solution.

And so we actually have a lot in common with even some of the newer technologies like Algorand, Hetero. We share with them a belief system that customers should be able to operate in a mutually trustless fashion and that information shouldn't simply be the repository of any one agent or agency on there. It should be something where any of those customers can seek and have and maintain their own unique representation. So I think in that sense, Vendia very much adheres to kind of the blockchain ethos if you want to call it that. We try to find a way to package and deliver it in a fashion that is more comfortable, familiar and accessible to enterprises, hence the desire for SaaS, for cloud-based and so forth.

There's probably the biggest point of division you might have here is this question of are managed services versus just infrastructure rentals sufficiently portable or disconnected from the interests of an individual CSP? And certainly if we were to build an AWS only solution, I think a lot of blockchain folks would not find that sufficient. One of the things that we're very excited about of course is having a multi-cloud solution so that customers can experience the best and the brightest in all the different clouds but without having to go to the lowest common denominator in terms of what it is that they have to use and port across those.

And as I've kind of written about and talked about in other venues, at the end of the day if you're relying on AWS EC2, that doesn't tie you any less to AWS than, say, making a Lambda function call or using AWS step functions. Like you're using all the same data center, all the same data center personnel, all the same building mechanisms under the covers that is really

not a strongly differentiated architectural perspective in terms of whether or not that cloud vendor has a connection to you or to your business outcomes and so forth.

And I'll tell you from experience being at Coinbase that the vast majority of Ethereum processing takes place in US East 1. So if US East 1 goes down, Ethereum's having a very bad day. So it is not divorced from cloud vendors or cloud vendor outcomes in an operational or practical sense even though that is sort of the religion of some parts of that community.

[00:39:30] JM: How do you see Vendia maturing? Like what do you see is the future of the platform?

[00:39:37] TW: So we're doing I think much like AWS Lambda did in its early days. We're really growing in three different directions here. One is what we call and much of what we've been talking about here today, what we call Vendia Share. That's kind of our mainstream product. Enterprise focus, kind of classic enterprise sale, very platform focused around code and data. Another way in which we're growing is creating things on top of that platform. One of the ones that I think we're both really excited about, we call the Vendia Virtual Data Lake. Think of it as taking uh file sharing and then putting all the metadata, the lineage, the permissioning and so forth for those files into a blockchain. And so obviously we do that on top of our existing platform and it's a really easy way for people to be able to share unstructured data files in the cloud but without losing control. Imagine giving away that information just like you might on something like Dropbox, but then having a button that says you want to call that information back again. You can kind of revoke that access down the road. And so we can do all of that through the benefits of keeping a tamper-proof ledger of information about what's been shared with whom along the way and over time.

And then the third one here and the one that David's also been spending a lot of time at Energean is we want to create a bottom-up developer experience. Part of what made serverless so exciting and frankly so successful is that lots of people out there in the community found uses for it that might not even have been imagined by us at AWS back in the day when we were first working on it. And we think the same thing is true here. We want

Vendia to be the easiest, fastest, simplest way to experience and develop serverless apps for any cloud and even across clouds. And to do that and to really be successful at that, we need to get that into the hands of millions of developers. So we're working on our preview release which we'll have out in a private beta in December. And then we'll be opening that up to additional folks in January at the start of the New Year. And I'm looking forward to getting lots of feedback and seeing what people build with it.

[00:41:45] DW: Yeah. And I'll add to that. One of the things that we're trying to do, there's this big educational component around this, right? Like a lot of uh developers out there, myself included up until you know six months ago, I wouldn't have considered using any type of blockchain for my data store. I typically use DynamoDB for something. But just the properties you get from putting something into a database and having a full transaction history, an immutable transaction history of what's happening, gives you a lot more flexibility around what you can do. I don't think too many developers are thinking about that. I think a lot of folks out there are still in the right in place of mentality around their database. And there was a time and place for that, because like data storage was expensive. We're in this world where like compute is getting cheaper every day. Storage is getting cheaper every day where there's really no reason why you couldn't have an immutable history of your database. What it’s looked like at like basically every point in time and kind of the benefits that fall out of that and just kind of the compliance regulations and all kinds of GDPR stuff that falls into that.

So we're really excited to kind of get the word out there. You can build quite a number of different types of applications. We're going after a lot of the traditional kind of distributed ledger blockchain stuff first because that makes sense. But yeah, it's pretty interesting what you can do with this. Again, we're providing you with an API. That's a GraphQL API. It would be indistinguishable from just you setting up your own AppSync client or what have you. So you could use it for some traditional use cases as well and then you get all those kind of benefits I mentioned just out of the box.

[00:43:41] JM: So just to revisit it one more time, because I know that there's going to be blockchain sticklers in the audience. What is the difference between what you're building and a shared database?

[00:43:52] TW: The easiest way to think about it is Hyperledger Fabric with every node having essentially unlimited resources. So the same tamper-proofing consensus, multi-node, multi-party guarantees that you would get with – This is sort of the leading contender in the space with something like Hyperledger Fabric. We provide all those same guarantees, all those same outcomes and capabilities. But we've built this and architected this in a very different fashion built off of a very different technology core and that gives us a unique set of properties and obviously a different set of customer outcomes especially when it comes to things like performance integration and ease of use. But the guarantees around replicated data that say Hyperledger Fabric would produce or even that, say, Ethereum, Algorand, Hetero, take your pick of kind of the newer chains, all similar outcomes, right? They all guarantee a tamper-proof ledger. They all guarantee that all machines maintain a replicated representation at the same time. And all of them in some way shape or form allow you to take some kind of code action on top of the current state of the database, a.k.a. a smart contract. And so we've done all of those same pieces.

The form factor can be seen in a very different way. For us, for example, most of our customers want us to manage their accounts for them even though we're still building it in a decentralized fashion. But there is no single centralized database. In fact, Vendia does not want and does not allow itself to see, does not want to see customer data. Doesn't want to own the customer data. It's very important to us that that be a database that our customers own and operate in an isolated account. Not something that we share and not something that we would take that data and multi-tenant it ourselves and manage that on their behalf. So that piece is very different from just running a database implementation.

[00:46:05] JM: All right. Well, as we begin to wind down, I want to ask uh each of you kind of your vision for the future of something. Tim, maybe you could give your vision for the future of crypto infrastructure since you worked at Coinbase for a pretty long time. And, David, you

could give your vision for the future of serverless since you worked at Netlify and Serverless Inc.

[00:46:26] TW: Sure. Let me kind of start by saying I think crypto has been and continues to be I think very successful as an economic tool as a store of value and somewhat but certainly not fully realized yet as a transfer of value and a kind of a transfer and payment mechanism even though it holds great promise for that. One of the challenges for crypto infrastructure I think is going to continue to be the challenges of scale, and particularly scale with integrity. I think as you kind of move to off-chain and side-chain and kind of increasingly divorce mechanisms that are really under the covers, just good old-fashioned centralized apps again, you start to dilute and diminish some of the value of those systems. And so I think what you're seeing Ethereum and others wrestle with here is the limitations of that kind of single box deployment methodology and what they're going to do about that and how do they kind of go from the 14 TBS to 14,000 or 14 million TBS. So that that feels like the single biggest challenge ahead for that community and it continues to be something that is kind of deeply wired into the architectural limitations today. And so I think they're going to continue to wrestle with not just how to slightly tweak a consensus algorithm, but how to kind of fundamentally rethink some of the things that are holding back the transaction power and capabilities there.

I think the form factor stuff, people will figure that out. I think you're already seeing a lot of companies and startups trying to make Ethereum easier to consume. I think that's a simpler problem to resolve than some of the fundamental architectural ones. But over the course of the next 10, 20 years, I think that's where crypto is going to go. I think it's going to be a long difficult road. But some of those pieces like shifting from proof of work to proof of stake are already well underway and I think you'll see more kind of sea changes like that coming down the road just based on some of the newest crypto stuff that I've seen people working on. So very hopeful for the future. Also very hopeful for the economic role that crypto can play. I think done right, it helps bank the unbanked. It helps make people resilient to things to wars and kind of loss of personal property and relocation. And so I kind of think forward maybe to sort of the future here. We've all had a very trying 2020. I have a lot of hope that crypto will be able to address some of those challenges in the future.

[00:48:50] DW: Yeah. And on the serverless front, we've come quite a long way. And I'm surprised that we have the father of serverless here on the call. So I'm sure Tim has ideas on this as well. But we've come a long way since serverless kind of came to be with Lambda, then API Gateway. Back in the day, you'd have to upload zip files of your code. It wasn't super straightforward. Again, like kind of that development hurdle was there, but the tooling around the industry has come quite a long way. So now with just a couple lines of code and a single command you can get your Lambda functions up and running in the cloud. It'll scale for you, paper execution. The whole serverless kind of spiel.

The other kind of interesting thing that we're seeing is the kind of arbitrary limits of the cloud. Like, previously, Lambda functions would only run for a max of five minutes. That's been lifted to 15 minutes. I don't have any inside baseball, but I don't see like why that couldn't lift even further. There's just like a lot of the limits. Cold starts have gone way down. They're doing a ton of stuff around that. So it's very interesting to see uh all these different improvements from the cloud providers.

Additionally, there's all these companies like Netlify that are trying to kind of streamline over the intricacies of these different cloud UI consoles and how they're kind of not as easy to approach. So a lot of developers are picking up these new technologies. Going serverless, using best of breed tools to actually like deliver results instead of kind of, “Oh, we need an email tool. All right, let's spend six months building a transactional email service.” They're leveraging these tools that are out there to get things done and actually focus on their core competencies.

I see infrastructure as code where it's been around for a while, but more and more developers are adopting that model, which is very promising to see. So, again, you can deploy you know a number of different stacks with a single command into multiple stages without this kind of error prone bash scripts and all this other stuff that we used to do. And then the other thing that I’m super excited about and super bullish on is just, again, back to this idea of the serverless super computer. So there's a bunch of interesting projects out there and a bunch of interesting

research from Stanford and other places where they're taking these processes that typically run in a single machine and it's like a large instance of something and breaking those jobs down into thunks or tinier pieces and just distributing that out, shotgun into many, many, many thousands of Lambda functions that are running in parallel.

So things that would normally take 13+ hours to do, they can do uh relatively quickly with just, again, an army of Lambda functions doing that compute. I think we're going to see more and more use cases of that. And yeah, I have a blog post on that. Serverless supercomputer, if folks are interested in in learning more about some of those projects. And hat tip to Tim to kind of tipping me off to that world.

[00:52:21] JM: Tim, I don't know if you have time to go over a little bit, but if you have any perspective on serverless, I'd love to hear it.

[00:52:28] TW: I always happy to opine on serverless. Look, I think David has some of the important highlights here. Probably by the time viewers are listening to this podcast, some of the new stuff Reinvent will have come out from AWS. I mean I obviously can't tip my hat on that, but I will say that that team that I started there continues to do amazing work, continues to think about how to bring serverless to more people, more foreign factors, more use cases. And I think you're going to see that continue to happen.

One of the most important pieces is how to selectively enable state. You're seeing Vendia tackle that in one way by providing a replicated data store on top of which serverless can run in the form of smart contracts or database triggers. I think you're going to continue to see that in streaming data. How to make it really easy to use things like Amazon Kinesis or Kafka with serverless. Today the binding between the compute state and the streaming data information, say, if you want to do windowed computations is really challenging to make work. And so I think there's going to be some continued progress in that fashion.

As David said, the operational aperture, if you will, continues to widen. So we've seen this kind of growth of serverless being something that's good for cron jobs to something that's good for

event handlers. In the case of things like the BBC, it's now a mainstream application for their customer, their mission-critical customer rendering outcomes. And I think you're going to continue to see that. E-tailers who use it for flash sales and other things, that mainstreaming of serverless into the real-time, synchronous business use cases. That was certainly one of the hottest growth areas when I was still at AWS, and I'm sure that continues to be a use case that's expanding kind of by leaps and bounds for Lambda and the other serverless outcomes.

And then as David said, there's some incredible research going on here. We don't have time to kind of dip into a kind of a full survey of that. But some really interesting outcomes as people look at, for example, how could you take, say, Python Pandas or something similar and reimagine that as a serverless implementation and how can you get the power and benefit of the cloud so that these large complex data pipelines that even within them have very varying needs of memory, of compute power and so forth, can start to become more cost effective, more scalable, more resilient and easier to use. And I think that's going to be just transformative. Same thing that kind of Databricks did to the Hadoop world, I think serverless computing is going to ultimately do to some of the existing companies and technologies over time. And so that's a really exciting area to keep watching as well.

[00:55:10] JM: Well, all right. Guys, thanks for coming the show. It's been a real pleasure talking to you, really wide-ranging conversation. I look forward to seeing where Vendia takes you.

[00:55:19] TW: Awesome. Jeff, thanks so much. Our pleasure to be here and really appreciate the opportunity to chat about the company and the perspective and direction. Check us out or learn more at www.vendia.net and please sign up for the developer preview if you're a developer.

[END]