EPISODE 1463

[INTRODUCTION]

[00:00:00] JM: Web3 is powerful, but difficult to work with. Deploying blockchain nodes, accessing data and performing staking operations are non-trivial engineering actions. To simplify Web3, Ankr hosts APIs for node deployment, RPC, and staking. Josh Neuroth from Ankr joins the show to talk about modern Web3 infrastructure.

[INTERIVEW]

[00:00:23] JM: Josh, welcome to the show.

[00:00:24] JN: Hey, thank you for having me.

[00:00:27] JM: You're working on Ankr, and it is a system of tools for improving access to, you could call it Web3 infrastructure. I think, Web3 has been accessible for a pretty long time at this point. Arguably six years since Ethereum got started. The higher-level tools, the APIs and the infrastructure is just now entering its more mature stage. Can you talk about the necessary building blocks for getting to a higher level, more accessible whole set of Web3 tools?

[00:01:12] JN: Yeah, definitely. The most fundamental piece in Web3 is really node access. A node is just simply the application that's running the blockchain that's sitting on a server somewhere. Anyone can run their own node. You can go download a node client, like Go Ethereum, which is commonly just called Geth. There's some other ones, like the Aragon client for Ethereum, or there's four or five more now, and increasingly more node clients.

Running a node, you can run a node if you’re using Ethereum, or an Ethereum virtual machine capable chain. You can always run your own node. Historically, projects have started running their own nodes. We saw early on that consensus was started by one of the Ethereum co-founders. One of the products that consensus offers has been Infura, is effectively building an API that runs on top of a cluster of nodes. That's also what we do at Ankr as well. We've seen really, just a lot of developer tools that frankly, just sit on top of cluster node technology that

make it easy for developers to just one click, or some command line in the CLI that gives you access to nodes.

That is the most fundamental building block of I'd say, Web3 infrastructure is node access. When I say node access, these are usually called full nodes. They're not validators. Validators build consensus on the blockchain, but everyone that's building anything needs to talk to the blockchain, send transactions, or a query data, on chain data would use a node access for that.

[00:02:55] JM: When you think about the older pieces of infrastructure, like Infura, or at this point, even Alchemy, Alchemy has been around for a while; can you outline in a little more detail what the older services have offered for a while and what some of the newer pieces of infrastructure are doing in contrast?

[00:03:22] JN: Yeah. I think ultimately, if you think about Infura or Alchemy, these are effectively Web2 business models, where effectively, just very similar to AWS, right? Where there's a subscription, or a pay-as-you-go model. As a developer, if you need access to these, you're literally just signing up for a pay-as-you-go subscription. There's no really governance from the Web3 community embedded into that. There's no way to pull that value back into the community. Historically, for both Infura and Alchemy, we've seen that they're exclusively backed by venture capitalists, and there's really no way for the community to capture that. As you're building out these developer platforms, or as the community, I should say, build out these developer platforms, I think we need to think about governance from a community standpoint.

We need to think about decentralization, because we're taking a decentralized system and we're frankly centralizing it in order to build these high-performance developer tools. One thing we've done at Ankr, and we're working on is we're frankly, just trying to – we started as a centralized service provider, and we're moving into becoming more of a protocol that's decentralized, that's community governed. The community’s setting everything from the payout to node providers, to the prices that the API is charged to developers.

I think that that's one big concept that we are still gathering our heads around in the Web3 space is this idea of being a platform versus a protocol. Platforms, we've seen these Web2, there’s as a playbook for building a platform. A lot of focus. You use the same venture-backed series A, series B, series C. Try to get as much traction as you can and then you capture that

and maybe have an exit, or an IPO, or something like that, versus a protocol that is decentralized, that is community governed, that just can't be shut down by a board member, or something like that.

[00:05:25] JM: In the case of Ankr, there are some building blocks that your team has built. I think, the place to start is with RPC endpoints. If I want to interface with various networks, like we talked about Ethereum, Avalanche, Celo, NEAR Protocol, Polygon, all these different networks, what is in the API spec for how we can interface with these different networks? What are the different calls we can make out to these networks?

[00:06:07] JN: Yeah, great question. First and foremost, there is – In order to build an API endpoint, like when we talk – I'm going to go back to the centralization versus decentralization before I answer the question. In order to create a single API endpoint, that is going to be, I guess, in a way, centralized. The way we accomplish decentralization is we pull together resources from the community on the back side, and then introduce governance into that process.

To answer your question, yes, the developer experience really starts with an RPC endpoint. The RPC endpoint is frankly just pulling in the official method calls that are available in the DOE client. Go Ethereum has a set of methods. You can look them up there. They're usually eth_, something like get logs is one that's common and used to pull on-chain data, or eth-call is usually used to send transactions, or pull some information off the chain. Those are the standard.

Then as you go into the Aragon node client, Aragon has a few method calls that aren't available on the Go Ethereum client. Then ultimately, what we see with Alchemy, Infura and now with Ankr as well is these services, or protocols are starting to build their own method calls to make the developer experience easier, right? There's an NFT, the one that's been popular as an NFT lookup. It just makes it easier to pull some of the information around NFT, the smart contract, who owns an NFT, maybe even the images on IPFS that are associated with that entity.

I think long-term, increasingly, while these – I will call them node infrastructure, developer infrastructure to really start with the same methods that you can get from running your own

node, but will increasingly start building their own method calls as well to improve the developer experience.

[00:08:08] JM: When I want to hit one of these networks, Ethereum, for example, why wouldn't I just use the out of the box experience for accessing the Ethereum network? Maybe you could contrast the out of the box Ethereum and experience with something like Ankr. Ankr offers, because you have your own system of accessing these endpoints.

[00:08:35] JN: When we launch it, what Ankr protocol offers just direct access. We are in the final stages of launching a decentralized API that expands those method calls. I think the biggest thing is that these calls are very rudimentary. I'll give you an example. We built a multi-chain block explorer called ankrscan.io. You can visit it in a web browser. It's still in beta. For a team of three engineers, that took about four months of building a standard of historical – frankly, it calls to build a historical view of the multi-chain space. Something we're going to be offering in the coming weeks is an API that takes those four months of time it took to build all those calls to go multi-chain and just make it available in one single method call to load a historical transactions and trace those back.

It really comes down to efficiency, right? You can definitely in a decentralized system, you can use what's available to the node client, but there are increasingly tools to make that experience better. That just improves time to market. We're seeing a lot of – Crypto is moving very fast. There's a lot of Web2 engineers and they're moving to space and starting to get started. I think we'll see as these newer APIs come to market, they ultimately make it simpler to build what the applications and dApps. We should see more innovation and faster innovation happen because of that.

[00:10:15] JM: Can you go in a little more detail, like if I hit – see rpc.ankr.com/eth, for example, for hitting the Ethereum network, what is actually happening when I hit that endpoint? Can you give me a breakdown of the call stack?

[00:10:34] JN: Yeah. Yeah. First of all, for DDOS protection and whatnot, we are using Cloudflare to mitigate some of the risk, because that's an open endpoint. One of the advantages of using the endpoint is you don't really need a sign up. You don't need to put a credit card in, or anything like that. It's a free community service. Certainly, if you use the Ankr token, you can

scale beyond the limits of the free service. We see a lot of MetaMask users also plugging that into MetaMask, or another type of – some other standalone, custodian self-custody wallet.

You can use that in a wallet app, as well as if you're an engineer building an application. When you call that, like I said, we have Cloudflare in the background. We have a very simple, I'll say, load balancer in place. This is very common. You'll see this in other platforms as well. Alchemy calls theirs a super node. We just refer to ours as a load balancer. Frankly, it's just a cluster of nodes. Ours is geography-based, which means we can offer a very low latency experience to the end user.

If you're in a place like Argentina, you'll be routed to our nodes that are running in Sao Paulo, Brazil. Rather than, if maybe on another platform that's hosted on AWS and might take all of the sub-C cables up to North America. The node at the backend is just a standard node. It's going to process that request, that compute power and then return the response back to the user.

I think, the cluster technology is really important, because if a node goes down and if it – which commonly, we just say it's out of block height, because in a peer-to-peer system, all of the nodes have to stay in sync with each other. If you get out of block height, the node is behind the latest state of the blockchain. Then that node is automatically withdrawn from the load balancer.

If you think about it, that's very important, because maybe there is an outage. Maybe someone's running a node in AWS. AWC has an outage, which is rare, but it does happen. The service is ineffective then, because there's multiple providers and it has geographic diversity built into that, and decentralization of providers and node providers offering nodes on the back side of that. It's a very resilient system and preserve some of the aspects that you'll see. The consensus side was validators, going peer to peer.

[00:13:15] JM: What kinds of applications demand lower latency access to a network, like Ethereum? Does the average application need to be super-fast? Does the average application, like can the average application tolerate just the basic Ethereum latency?

[00:13:34] JN: Yeah. Well, it really depends on how the developers are integrating it. Typically, transactions, I'd say the number one use case where latency is ultra-critical is when people are trying to do sniping bots, which is a very small percentage of the community right now. They

need absolutely, we're talking one, two milliseconds, where they're trying to cut as much latency off their transactions as possible.

Then for users that are using DeFi, it's very important. Primarily, because if you're doing transactions, like the more hops that the network is taking, or maybe you're accessing a node in another continent, then potentially, your transaction error rate goes up as well. We've seen that. I think, what's also important is that a lot of developers are just integrating the RPC endpoints directly into the front-end and to the client side. You see Web3.js, you see Web3 for Python, or other libraries and frameworks where it's all client side. It's not a server-side call. In that sense, the user's web browser is really calling the RPC endpoint.

Potentially, there's a lot of latency in that transact, or in all those requests as data is being pulled in. Then the website, or the dApp
is going to feel slower to the user. We've seen this same thing happen in Web2, where these investments in all these frontend technologies and making things faster with geographic diversity have really helped the user experience. That's what's happening right now in Web3.

[00:15:22] JM: If you look at the kinds of applications that are using Ankr, what are you seeing? What kinds of dApps your RPC infrastructure?

[00:15:37] JN: Yeah. Well, we have customers, like DeFi Pulse, who are pulling on-chain data in for doing some analytics and market research. That use case is a little bit different, because they're doing server-side calls. I believe, they're running a major cloud platform and pulling the on-chain data directly into their application. Then you see, we serve a lot of – probably the number one endpoint, or RPC endpoint for Fantom. We serve Spooky Swap, Spirit Swap, two of the largest decentralized exchanges, or commonly called DEXs on the Fantom Network. There's a lot of DeFi activity in Fantom.

In those cases, they're frankly dApps, or decentralized exchanges where they're connecting to the RPC and processing transactions and weaves. In those cases, you're seeing, like I said earlier, you're seeing the end user through the dApp querying against the RPC endpoint. Right. That's a major use case. We see NFT mints using our RPC. Maybe there's some new 10,000 NFTs that have launched. There's a mint going on. Those mints need access to an RPC endpoint. We see NFT marketplaces.

Really, I guess the way to think about it is there's definitely people that are doing server-side calls, where there's some backend process, like querying data, or sending transactions. Commonly, we see a lot – Any time there's a Kinect wallet button that's in some – in a lot of cases, those developers will just integrate the RPC endpoint to facilitate that. Really, I guess, you could say, every use case in Web3, in DeFi, in metaverse gaming, NFTs is going to require an RPC endpoint, right? This infrastructure is very fundamental. It's very necessary in order to facilitate that.

I will say, too, that I think in the previous six months, we've seen a lot more users and dApp developers using web sockets, specifically in mobile applications. They want those concurrent connections. That is increasingly becoming very important infrastructure, especially in the gaming community.

[00:17:58] JM: When I think about a gaming application, if I wanted to have a game where the entire – maybe the entire game state is represented on the Ethereum blockchain, you probably would need really rapid access. But you would also need fast write time. You need both fast read and write time. With an RPC system like Ankr, you're mostly just getting fast writes, right? You're not you're not necessarily getting fast reads? Or sorry, you're mostly in fast reads, not necessarily fast writes.

[00:18:39] JN: Well, yeah. Because the way that the blockchain works, you can pull data instantly off-chain at 10, 20 milliseconds usually. In write times, when you write a transaction, you need to pay a gas fee and there's a block confirmation that it's happened before that transaction will actually be essentially included in the block. It's going to stand in what's called the mempool, or the pending transactions.

Ethereum is actually a very slow chain. We see Ethereum has its global capacity right now is 25 transactions per second across the whole world, across all the users using Ethereum. That's very slow in terms of transactions. We see Ethereum being used more for a trust layer. Then, we see what are called layer two scaling solutions that are really running on top of Ethereum. Some names that you might be familiar are Polygon, which is frankly running a sidechain and other solutions as well. We see Arbitrum, we see ZK-Rollups now.

There's many ways they're being architected, but one way is that these secondary, or layer two solutions are making it very cheap and fast to do a lot of write transactions. They package that up and then they write that to the Ethereum network as a single transaction. Within that one transaction, you might have 10,000 database rights on the layer two. That's one way the community is building and scaling. You're not going to pay a $15 gas fee, or $20 gas every time you write to the Ethereum network. That's very impractical if you're building a game, right?

I think, we've seen layer twos in the last year just completely accelerate. Some of them are processing 2,000, 3,000, 4,000 transactions per second now. That's really going to be how a game developer frankly builds on those is using the layer twos. You still need an RPC endpoint there.

The other thing I'd point out, too, with that is that we see, when we talk about metaverse, when we talk about games, the games themselves are being processed using traditional technologies. I've seen games have built on Roblox. I've seen games built in Unity. Facebook's building the metaverse right now and their gaming industry, and you see some games built on the Unreal Engine. I guess, what's being – why they're being called metaverse games is because there's play-to-earn elements in there, where they're the game items are being given to players, or traded to players through NFTs. Those players have those items in-game. Weapons, skins, whatever they are, have real monetary value that can be traded on an open marketplace. That's how a lot of these games are starting to take shape, at least in the Web3 community.

[00:21:37] JM: We've talked about RPC infrastructure, at least as it pertains to Ethereum. You've got RPC protocols for Polygon, Solana, all these other networks like I mentioned. Can you just tell me, how does the engineering between those different RPC systems compare? Is it just you copy paste, basically, or are there different –

[00:22:06] JN: No. Not at all. Increasingly is different. In many of these chains, like I mentioned **[inaudible 00:22:12]** that are made for Ethereum and there's many other networks, like Fantom that are called Ethereum virtual machine compatible. That means a developer can use the same – those standard method calls, the same method calls for Fantom as they can for Ethereum. It also means the smart contracts are more or less compatible.

Of course, sometimes you have to do a little bit of modification as you go crossing. When a developer goes multi-chain, picking another chain, a secondary chain to integrate with is if it's Ethereum virtual machine compatible, that is a easier learning experience, or easier to integrate. They can use a lot of the code, reuse that and then you have chains like NEAR and Solana that are built. The primary language is usually Rust. The Solana methods are very different from the Ethereum communities.

Right at the moment, Solana is not EVM compatible, or Ethereum virtual machine compatible. There are some people working on that right now. NEAR Protocol has Aurora, which is working on Ethereum compatibility for NEAR. That makes it easier for developers that are already familiar with the Ethereum ecosystem to adopt those chains and integrate those chains into their applications.

[00:23:32] JM: The Ethereum compatibility, EVM compatibility, if I deploy a smart contract to the Ethereum network, then obviously, people can access that smart contract. EVM compatibility would allow somebody to also deploy that smart contract on Solana, if Solana was EVM compatible. Can you just talk about the importance of EVM compatibility? It seems to be a common denominator that enables a lot of these networks to be viable.

[00:24:10] JN: Yeah. Well, I think the community I mean, the most advanced developers in this space that you mentioned earlier, how Ethereum's about six-years-old now, maybe seven, depending on how we look at that. There's a lot of experience that has already gone into Ethereum. You have that first mover advantage on smart contracts that helps out a lot long term. Maybe EVM compatibility isn't that important. At the moment, there are – I'd say, the lion's share of developers are using solidity. They're using libraries, like Web3.js. A lot of the that knowledge is important for adoption.

I think a great multi-chain example of a DeFi protocols is beefy.finance. I know the team over there. They've integrated what, eight, nine, 10 different chains now. All EVM compatible. It's a great example of a platform where it wants to access more users. It integrates a new chain. Why would a developer be interested in going to a chain that's not Ethereum? It's because, we started seeing this multi-chain space that each chain has its own community, has its own passionate uses. It has all these assets that are on that chain, right?

An example I'd say is, you have in the West, the most popular chain is Ethereum. Some of the emerging markets in the East, we see Binance smartchain, now called the BNB chain is definitely the chain of choice. It's fast. It has little gas fees. Of course, you have Binance there as the major exchange in a lot of those emerging markets where they don't have Coinbase, or Kraken, or Gemini, or other exchanges like that.

If you want to reach those users in places like the Philippines, or Thailand, or Vietnam, Binance smartchain is probably the most popular chain there. You would go use that chain, integrate it really to get access to that community over there.

[00:26:17] JM: Do you know anything about how these different blockchain – I mean, my understanding is Binance smartchain is faster than Ethereum. Avalanche is faster than Ethereum. Do you have any sense of what the engineering differences between these two different chains are that leads to latency differences?

[00:26:42] JN: Yeah. It's, I want to say, latency. It's really about what determines a chain speed is frankly, how fast the next block on the blockchain forms and how many transactions can that block contain? That's really what defines the chain speed. Sometimes that's different than the transaction speed, but they are related. I think, what the Ethereum community has tried to do is make it decentralized. Basically, it comes down to the server requirements more or less that power the nodes.

There's a community that runs Ethereum nodes on Raspberry Pi. You can run a Ethereum node on your local machine. That helps the decentralization of the network, because the barrier to entry is very low. Go get a cheap droplet in DigitalOcean, or run an Ethereum node if you want. Now, in contrast, Solana is a very fast blockchain. It's currently doing, I think, about 4,000 transactions per second compared to Ethereum's 25. The Solana node requirements are crazy compared to Ethereum. You're looking at an enterprise grade server, NVMe disk, 256 gigabytes of RAM minimum even to just stay a block on the Solana blockchain. The cost of that server in contrast to maybe in Ethereum server, where Ethereum server might be $30 a month than DigitalOcean, or you can run a Raspberry Pi. Solana is probably, you're looking at a $1,000 a month, or $2,000 a month if you're just renting out an AWS or something.

Some of these chains that have frankly traded on a bit of decentralization for speed. I think, that's one reason why people are excited about building on the layer twos on Ethereum, versus going to a chain like Solana is because of that preservation of the decentralization.

[00:28:32] JM: If we go just a little bit deeper, so this seems like you're pretty well versed in the different blockchains, at least at a high level. It's going to take advantage of that a little bit. There's a wide variety of networks, like NEAR and Polygon and Celo and Binance smartchain and Avalanches that I have heard a lot about. I've seen a lot of traction from these networks. You mentioned one of the axes, or you mentioned a few of the axes that these chains can be compared on, particularly block size, which leads to faster finality, but different server infrastructure requirements. Are there other ways that, like I've heard Celo is good for mobile payments. Are there other ways in which we can compare these different blockchains to have a principal component analysis of them?

[00:29:35] JN: Yeah. Well, I think that like I said earlier, the two main ones are going to be the speed of the network and also, the cost of the gas fees. Those are related. In a lot of cases, Solana has very cheap gas fees. We've seen Polygon has averaged to a dollar, sub $1, sometimes more. Ethereum is $15, $20, $30. If you're writing to a smart contract, could be a $150 per transaction. Speed of the network, the gas fee is a big thing. The node requirements, definitely the server requirements factor into that. I think, the other overlooked thing sometimes is the incentives. A lot of these protocols have an ecosystem fund, and they're putting – they're almost acting like venture capitalists and they're putting money into dApps and tools.

The development of those ecosystems really accelerate the adoption of the chains. That's one thing that's happening, and Avalanche has been doing a great job, getting Avalanche integrated all over the place. Obviously, Binance they're the largest exchange by volume in transactions. They're seeing, they help get BNB and the BNB chain adopted everywhere. Whereas, Ethereum has this huge ecosystem, because it's decentralized and no one's really controlling it.

Then Solana Foundation is putting a lot of capital behind things as well. People follow the money, I think. Developers can follow the money. A lot of times, there's grants available for dApps you might get. Instead of having to raise an early seed fund as a team, you can go to the ecosystem. You can say, "Hey, there's a gap in your ecosystem. I'm going to fill it. Here's our

prototype.” You might get a grant. Sometimes these grants are $20,000, $30,000. I've seen them go up to 500K, like a small seed.

Then obviously, if you did get that, then you get a lot of support from that foundation of community. Then you might use an NFT sale, or a token sale as well and you raise some more money on top of that. Incentives have a lot to do with what's happening. I think, Solana has been doing a great job. There’s a huge backing of FTX behind it and you have a lot of Silicon Valley VCs behind Solana. They're using their insider connections to get adoption happening. It's a great time to be a Web3 developer. So many options. I think you've got to find a community. You got to find a chain to start with that really, you feel connected to the community and then you can go multi-chain as you get traction and really build the user base.

[00:32:21] JM: You're saying that the average application wants to be on a variety of blockchains?

[00:32:30] JN: Well, I think, frankly, I mean, if you think about it, let's just think about this use case. Let's say, you develop on Avalanche. Maybe I develop a dApp on Avalanche. I've only integrated Avalanche. I'm limited to the total capital that's on Avalanche. Now, there's some bridges and some cross-chain things that can help users, but that's really limited to Id’ say, DeFi power users that are really going to use any bridges-to-bridge assets from Ethereum to Avalanche.

Keep in mind, all these blockchains or siloed, right? If you have assets on Avalanche and you want to go to Ethereum, if you're not going to use a cross-chain bridge, you've got to sell those assets, which triggers a taxable event in the US at least, and then you're going to trade those for ETH, right? A lot of times, as a dApp developer, you want to go where the money is.

We've seen platforms like OpenSea that started on Ethereum. Now, they have Polygon. Polygon is helping with the lower gas fees on there. You see some of the games, or platforms will pick two different chains to start with and then go from there. In Web3, money is embedded in the experience. It's embedded in the infrastructures. You got to think about wallet access, where are my users going to be. Long-term, I think these problems of having to pick what chain you want to go on will get easier and easier as some of the cross-chain, multichain bridges get better and the wallet technology gets better.

For now, it's still pretty early on. You have to pick. I see most developers in the United States at least, especially in the traditional Silicon Valley influence tech crowd start with Ethereum. Definitely is like a rich person's network with the higher gas fees. You start with that and you get access to Ethereum community, and then you start branching out into a layer two, or one of these other chains.

[00:34:39] JM: If we zoom out from just the RPC systems that you've built, there are other pieces of infrastructure that you've worked on. Well, I guess, we should just talk about the Ankr token, and explain in more detail what the token actually does. Can you describe what the purpose of the token is and what it actually empowers?

[00:35:09] JN: Yeah. For the Ankr token, we've been – so first and foremost, our token is about four years old right now, traditionally. Prior to October 2021, we really just used it for governance. It wasn't widely used. What we're moving to is all of the protocol revenue is now tracked through the token, and then the token is paid out to the independent node providers. Money in through the token, money out through the token. Then we're working on some features where you can lock that token up and it's almost like an Amazon Prime style membership, where by locking the token, you maintain ownership of the token, though it's locked and it gives you exclusive access to some benefits.

Developers, instead of having to pay a monthly fee for something, they're going to be able to stake the token, or lock it in that case. Instead of earning passive earnings, get access to a feature in the product. Then obviously, as I talked earlier about the Ankr DAO, we can use the token to submit governance, where we're going to use snapshot of the work. We're also looking at Aragon in the future to really bolster that community and make it easier to do governance on that, and get users of the product engaged through the token.

Ultimately, I think the token in a Web3 business model, the token is essentially what makes people excited about ownership, about being engaged. Frankly, the rise of that token. I'm a product manager. One of the hardest parts of Web3, especially in building a protocol is when you have a token. It's easier to build centralized service provider for Web3, whether you're serving data, or even building RPC endpoints. Building a product where there's a micro economy at the base of it, there's a lot of math, there's a lot of economics.

Unlike software features, when you build a tokenomics and how the token is integrated and distributed to users, that's a feature that is very hard to unwind, or change. I think, there's been some protocols out there that have gone. They've done it very fast. They made some mistakes and it's really came back to bite them. We've taken a much more conservative aspect of rolling out the token in stages and being very, very slow. But also, in my opinion, doing it right on the decentralization, transition to a decentralized model.

There's not really a playbook for tokenomics yet. There's a very defined playbook, I think, for building a Web2 company these days, or software as a service, for example. When you integrate a token, it's a completely different dynamic. There's also a lot of gray area in the regulatory environment of how these things can progress. More to come on the token.

[00:38:03] JM: If I look at the most success full blockchains, typically there's a slow ramp up to the network being recognize as being important. It usually doesn't happen overnight that a network gets widely adopted. I'd love to get a sense for how the growth has progressed over time with Ankr and how your infrastructure had scaled in response to that.

[00:38:41] JN: Yeah, absolutely. You're absolutely right. I think, the thing to think about, if you zoom out, like you said, and look at the macro thing, it's very hard to deal – It's very hard. You might be familiar with the concept of product market fit. It's when you really build the right product for a market and it feels like it's flying off the shelves, so to speak. You're getting adoption, you're getting traction, you don't have to work too hard to sell it. Product market fit is historically, has been built by very small focus teams. In the Web3 community, building product market fit is very difficult to do with a decentralized community; trying to organize a community around building something.

What I see a lot of projects, like Ankr, or compound have done is they've built a centralized service with a small focus team. They get product market fit and then they begin handing off pieces to the community, and facilitating governance. As you do that, as you get the community engaged, community gets excited about it. Usually, you should see a snowball effect happening. We've seen that with Ankr. I think, we're still early days. In 2021, in the calendar year, we saw 20X traffic growth. We ended the year with doing around 200 billion with a B, request to the

blockchain from our users on our RPC endpoints across all of our RPC endpoints. We started the year, what? Under 10 billion in early 2021.

A lot of that came from the growth in DeFi, in NFTs, in decentralized exchanges. I think this year, I told my team that our goal is to get to 1 trillion, which I think is going to be easy for us this year. 1 trillion monthly requests to the blockchain. A lot of that is going to come through games. We have several games building on our RPC endpoints right now integrating NFTs into traditional games that are already available in the App Store and the Google Play Store, on mobile. Those will use a lot more requests.

Certainly, requests are just one way to measure, I guess, growth. Because it varies widely based on the use case and the dApp, or the game that's integrating into that. It is one good match to show. I think, we've seen in the last year, to answer your second question around infrastructure, to go back to the concept of infrastructure to power that growth. We do have about 30% of our nodes at the moment that are served by the community and contribute to decentralization. Ankr still runs about 70% of our nodes. Our goal is to fully turn those over to the community.

As the server requirements, especially for chains like Solana have increased the average sysadmin that has a full-time job and is running nodes on the side, or maybe we have several DAOs that run nodes for us, they don't know how to go out and get service out of the supply chain. We've helped them in a lot of cases do that. Frankly, we built a custom server. We do everything on bare metal, at least when we help provide those node providers with hardware, we built a custom server.

When I say custom, it's a custom configuration. Obviously, we're not at that scale where we're inventing our own CPU, or modifying ARM architecture, anything like that. We built our custom bare metal server. That just means it's a server that’s running in a standard data center right down the hallway from where AWS runs their servers and powering traffic. I think, that's a good model for the community. My background, I've worked in many infrastructure and service companies prior to being Web3.

It gives us a huge advantage cost-wise over platforms like Infura and Alchemy that run exclusively on AWS. It makes it very easy for us to plug and play nodes into our load balancers, because they can be ran from anywhere in the world as long as there's good hardware.

[00:42:42] JM: As we draw to a close, can you talk about some of the other systems that Ankr has built, just particularly around DeFi infrastructure?

[00:42:54] JN: In addition to our RPC endpoints and our developer APIs, we also have what's called liquid staking. Liquid staking is really when you stake – First of all, if you're not familiar with what staking is, staking, a lot of the earlier chains, like Bitcoin use proof of work. They use all those, the miners. Proof of stake doesn't use miners. It uses a financial mechanism to help maintain consensus and frankly, security as well on the blockchain.

In the case of Ethereum, that means every Ethereum validator which takes the place of Ethereum miners, and once Ethereum fully transitions to proof of stake, hopefully later this year, all the Ethereum miners will be obsolete. Then the Ethereum validators will take over building consensus. What is going to happen in that case is that when you launch an Ethereum validator, you'd be 32 ETH that is staked, locked up for each validator. We've seen since December 2020, when Ethereum started transitioning to this beacon chain as it's called, we've had about 60,000 Ethereum that have been staked across in almost 2,000 validators or as so, 32 ETH per validator.

The community, we built a decentralized solution where the community can run some of the validators. We call them side cars and contribute to the decentralization of that. Ultimately, it means that you are issued by Ankr, a liquid staking token that represents your stake in the network. What this allows you to do is exit your stake at any time.

When you stake on Ethereum at least as of March 2022 here, your deposit in if you will, is going to be locked until Ethereum transitions to proof of stake. We don't know when that's going to happen. It's been delayed several times. When you have a liquid staking token, you can sell your stake, or exit your position at any time by just selling your liquid staking token, which is linked to your underlying deposit. That's what liquid staking is. We compete with other platforms like lido.finance, or Lido as they're commonly known as. It's another liquid staking solution out there. Smart contracts-based.

There's other reasons to use liquid staking as well, because there's a lot of incentives now in the DeFi community in DEXs. You can take your liquid staking token and then join it into a liquidity pool, provide liquidity, earn sometimes farming rewards in that case, or even borrow against it if you're if into that.

We see early on when liquid staking started and Ethereum staking, I think the price of Ethereum was around $400, or $500 per ETH. Now, it's a lot more expensive. The other solution that liquid staking allows us to do is to create micro-pools for staking. If you're staking with Ankr, I think the you can stake with a minimum of 0.5 ETH now on the product, instead of having the full 32 ETH.

Keep in mind that this is a DeFi-based solution. You use your self-custodian wallet to do this and all that, versus if you were staking with Coinbase. Frankly, Coinbase doesn't have liquid staking and it's a centralized service. They have all your ETH as your custodian in that sense. We're really DeFi native in that sense. We're expanding to other chains and we've launched Polygon, we launched Fantom, and now Binance chain as well, the same type of solution. I think, we'll see a liquid staking be used in a lot of really interesting ways in the future. There are some ways we can integrate it with the developer tools as well. That's exciting. But coming on a roadmap later this year.

[00:46:49] JM: Can you just, before we close off, talk a little bit more about staking? If I'm comparing different places where I can stake my money, why wouldn't I always just choose the highest ROI place to stake my coins? Is there a risk when I stake tokens?

[00:47:12] JN: Well, yes. In proof of stake, there's always risk, because if you stick to a validator that is malicious, that meaning that validator starts trying to fudge, or fake transactions on the network, that validator could be slashed by the network, which would mean this underlying deposits would be lost, or would be slashed. They'd be minimized, or some of the deposit would be taken, frankly. There's always risk when you're staking.

I think, the reputation of the protocol, or the platform matters a lot in that. For people that just want to stake and forget about it, honestly, if my dad was like, "Hey, where should I stake?" I'd probably just say, stake in Coinbase. If you don't care, you want to keep it locked up for the next

year, potentially on Ethereum, then stake there. It's going to be simple for you. All of that for DeFi users though, products like what Ankr has done with what we call Ankr Earn now is a great solution, because you can get added benefit through yield farming and liquidity pools and have that instant liquidity as well. There's some exciting news we have in the future.

We've done some partnerships in the past where we've had extra incentives that have taken the APR on the asset from 6%, 7%, up to 15%, 20%, 30%. Those rewards are only available if you use a true DeFi-based staking solution. That's a big reason why you'd want to do it. If you don't care about DeFie and you don't want to maximize, or a 100% maximizing your rewards, centralized platform is fine, I guess, in that sense.

[00:48:56] JM: Well, Josh. Thank you so much for coming on the show and exploring the variety of blockchains and staking and other subjects that we explored. Very interesting platform that Ankr is developing into.

[00:49:09] JN: Yeah, protocol. Trying to get out of the platform terminology.

[00:49:13] JM: Sorry, protocol.

[00:49:14] JN: Transition to a protocol. Yes, thank you so much for having me. We're excited. Again, you mentioned it earlier, but if you're a developer, we have these RPC endpoints. You don't need to sign up for anything. They're available out there to the public and you can start playing around with the blockchain. We got some tutorials now, whole developer section on our documentation. Certainly, we have community channels, too. If anyone has a question about dApp development, we can always point you in the right direction on our community channels.

[00:49:45] JM: Awesome. Thanks again.

[00:49:47] JN: Thank you.

[END]