

**EPISODE 1233****[INTRODUCTION]**

**[00:00:00] JM:** Cryptocurrencies like bitcoin are electronic currencies with a complete transaction record stored on a blockchain. A cryptocurrency blockchain is a linear record of all transactions between users for a given currency. This record is public and distributed across thousands of computers, which makes falsifying a transaction nearly impossible. People use digital contracts for code that dictates the terms of a transaction and executes those terms on the Ethereum blockchain. dYdX is a decentralized exchange allowing people to exchange cryptocurrencies without a centralized broker, such as in the case with Coinbase.

Antonio Juliano is a Founder of dYdX, and he also founded Waypoint and worked as a software engineer at Uber and Coinbase before launching dYdX. He joins the show to talk about the crypto community and their recent release of layer 2 with dYdX, as well as the future of cryptocurrencies and blockchains.

**[INTERVIEW]**

**[00:00:59] JM:** Antonio, welcome to the show.

**[00:01:01] AJ:** Hey. Thanks so much for having me.

**[00:01:03] JM:** Many people have probably heard the term DeFi these days, decentralized finance, and this is probably a newer term for something that has actually been going on for several years. I think dYdX was pretty early to the world of decentralized finance. Can you explain what you saw coming in the world of Ethereum and crypto technology when you started dYdX?

**[00:01:30] AJ:** Yeah, absolutely. You're right. dYdX has been around for a little over three and a half years now. By crypto standards, that's ages basically. But I got really excited about Ethereum and what was possible to build there. When I started my career at Coinbase back in 2015, that was basically my first job right out of college. I was a software engineer on their payments team. While I was at Coinbase, one of the cool things that happened there was we

had a really great front row seat in terms of seeing a lot of the exciting new developments in the blockchain space, and the main one around that time back in 2015, 2016 was the invention of Ethereum.

Kind of once we kind of collectively wrapped our minds around what Ethereum was, how it worked at Coinbase, I got really excited about it personally. I was really excited about it because it really just seemed like a new paradigm of computing to me. For the first time, you can write programs that just execute totally deterministically, totally autonomously, and without being controlled by anybody. This just seemed fundamentally different than anything that had come before. Around that time, I was basically convinced that there would be something interesting to build on top of Ethereum because there basically always is when there is just new paradigms for technology, new paradigms for computing. But I didn't really know what would be interesting to build at that point.

Eventually, I kind of came to the idea of dYdX because the main way people are using cryptocurrency is back in 2016, 2017 and even still to this day is for trading and speculation. That was right around the time mid-2017 when the very first decentralized exchanges were starting to come out, so things like ZeroX and Kyber. These are just kind of more simple decentralized exchanges for spot trading. Spot is just basically a fancy word for buying and selling. I took a look at that and felt like the logical next thing to build after that would be more advanced financial products, so things like margin trading and derivatives.

That was kind of the genesis behind the idea, and I think one thing that's critically important when you're thinking about starting companies especially is not just coming up with an idea that will be valuable at some point. It's really more timing and coming up with an idea that actually is valuable in the market right now. I think that's what we stumbled upon and what we're able to find with kind of the genesis of dYdX.

**[00:03:45] JM:** Could you give an example of an application? Well, actually first we should probably explain what dYdX actually is. I think we've done enough shows on Ethereum that people are probably familiar with Ethereum, but describe the interface between dYdX and Ethereum.

**[00:04:01] AJ:** Yes, absolutely. dYdX is built on top of smart contract, which run on Ethereum. Smart contract is a fancy word for programs basically. So we write smart contracts which effectively know how to do derivative contracts. The type of derivative we specialize in for the most part right now is the synthetic contract called a perpetual contract. It's not something you find so much in traditional finance, but perpetual contracts are extremely popular in crypto because they're relatively simple as synthetic products go to trade. You basically can synthetically create a new asset that tracks the price of another asset.

For example, if you're trading a bitcoin perpetual, you can trade on the price of bitcoin without any actual bitcoin being involved. The main reason these are so popular and the reason people like to trade these is because they can be traded with really high leverage. Leverage basically just means you can multiply your gains and your losses, and you can trade as if you had more capital. Imagine you came to dYdX with \$100. You could immediately start trading if you decided to use 10x leverage with as if you had had \$1,000, and you could buy basically \$1,000 worth of the bitcoin perpetual or even more if you want to. That's the reason why these products are so popular in crypto.

We're really excited about the market for synthetics and for derivative products because normally in the traditional financial markets, you see derivatives trading at kind of like 5 to 10x the volume of the entire rest of the financial market combined. So it's quite literally the biggest market in the world, and one of the interesting things that's happened in crypto recently with the trading markets is for the first time derivatives volumes on crypto have surpassed all the other trading volume on crypto combined. It seems to be well on its way to hitting that 5 to 10x the volume mark. It's a massive market.

Kind of to answer your original question, yeah, we interface with Ethereum just through smart contracts that that run directly on Ethereum. One other thing I'd love to dive into, and I'm sure we will a lot more just a minute, is we've recently launched a new product which is built on what's called layer 2. Layer 2 is just a general term for kind of a new more scalable type of blockchain technology that allows us to process many more transactions, process many more trades but then kind of rolls up or settles everything on Ethereum, so retains a lot of the same security and decentralization properties. But that's what we've been working on for the past six

months or so, and we just recently launched this new product that we built on layer 2 and are kind of building from there.

**[00:06:32] JM:** When an entire system is actually built on top of Ethereum, are there some kind of engineering concerns that you need to take in? Well, I'm sure there are engineering concerns. Just tell me about some of the challenges of actually making a system work on top of Ethereum.

**[00:06:51] AJ:** Yeah, absolutely. There are definitely a lot of challenges in terms of building a real production and, for us, a real professional application on top of Ethereum. I'd say overall the biggest challenge is just everything about Ethereum and everything about blockchain more broadly is still pretty nascent, still very new. Ethereum was just invented a few years ago, so a lot of the tooling around it is not nearly as good as what you would experience as a software engineer if you're just using more standard programming languages or practices.

For example, like testing frameworks. We had to build like all of our own testing frameworks. Couldn't just use like something off the shelf like you would experience like a node or something like that. That was definitely one consideration. I'd say the biggest overall concern or biggest overall kind of time sync for building smart contracts, especially for us, is just you have to have a real change in mentality when you're programming because security has to come first and foremost. Basically, the way our smart contracts and most smart contracts and DeFi work is that they quite literally hold money. The smart contracts on Ethereum quite literally are in possession of the assets that our users are placing into them, and they'll only disperse those assets in accordance with the smart contractor in accordance with the program.

That's all awesome if the program has no bugs in it, right? But suppose there was a bug in the program or suppose some other security incident that then users could potentially lose hundreds of millions or billions of dollars, which is just a massive price tag, and that's basically a death knell to any given protocol that has a security incident like that. For us at dYdX, really what that means is just trying to make our smart contracts as simple as possible, so they're really easily code reviewable, really easily auditable by third parties.

Another thing that we just enforce everywhere for our smart contracts is just 100% branching test coverage in terms of our solidity code that we write for our smart contracts which, again, like

slows coding down, but it's really a trade-off you have to make. I think one reasonable comparison that I didn't come up with but I've heard before and I like is that kind of coding smart contracts is a little bit more akin to designing hardware because once you ship them, they're not really upgradable. If there is a bug in them, it's a serious problem. So it's definitely you're just going to have to change your mentality. It's not so much like you can ship your MVP version of your smart contract and then just continue iterating from there. It's something that cannot have any bugs in it from day one, and that just makes it have to have a change when you're coding them.

**[00:09:25] JM:** Give an example application of dYdX. What would a user of dYdX actually be doing?

**[00:09:34] AJ:** Yeah. Users on dYdX are just trading for the most part, and they're trading our perpetual contract. We're kind of marketing or our target user is more a sophisticated trader on cryptocurrency, so it's not someone that has just, say, bought their very first bitcoin on Coinbase. It's really someone who's been in the space for a while. Still mostly dominated by individual or retail traders but a lot of individual traders and especially in the crypto space have gotten pretty advanced now and want access to more advanced financial products like margin trading and derivatives. That's really what they can find on dYdX.

Of course, we're not the first exchange to offer this. Different centralized exchanges have gotten really popular in crypto in terms of offering leveraged products and more advanced products. But I like to think we're one of the leaders that's doing it in terms of DeFi. In terms of what they can expect, it's similar to like the UI you might see on something like a Coinbase Pro or any other exchange that the listeners may be used to. You see things like the order book, which is basically all of the prices that people are willing to buy and sell the given products at. You'd be able to place trades through kind of the product to be able to see a lot of advanced statistics on your position, the size of your position, how much leverage you're taking on your profit and loss, things like that.

It's a bit kind of like a Robinhood, which I guess may be a little bit more mainstream product but something that's even more advanced and targeted at more advanced traders than that.

**[00:11:02] JM:** Why does it matter that dYdX is decentralized? Why not just have users be interfacing with contracts that are offered through centralized services?

**[00:11:14] AJ:** Yeah, definitely. It's a good question. I think there are a number of different reasons why people are getting pretty excited about decentralized exchanges in dYdX being a part of that. The first I think is security. I'm sure people have heard a lot over the years about various centralized exchanges getting hacked, and this just continues to happen to this day oftentimes for hundreds of millions of dollars. Pretty familiar with this as well because one of the things I worked on at Coinbase was I was one of the main engineers that worked on their hot wallet service, which is basically the service which custodies all of their private keys. Yeah. If you can just store your private keys in your own user controlled cryptocurrency wallet, which you can do on a decentralized exchange like dYdX, it's just much a much more secure way to trade. That's the first thing.

I think the second thing is transparency. One of the things that's really important to traders and especially important to traders who are trading more advanced products like derivatives is transparency. In addition to centralized exchanges, custodizing your funds in cryptocurrency, if you're trading, say, a derivative product, the exchange is also responsible for the operation of that product. Not to get too technical but things like liquidations, things like the funding rate, which kind of drives the price of this derivative product and everything else that goes along with that. It's really kind of a black box to users what's going on in these exchanges. Whereas on the decentralized exchange like dYdX, because everything is built on top of these open source smart contracts, everything is verifiable, auditable, and everybody can understand what the rules of the exchange that they're trading on are.

Kind of the last and I think most important reason that users might prefer a decentralized exchange over a centralized exchange is access. Users of smart contracts and DeFi are just interacting with these open source smart contracts, which are available on Ethereum everywhere to everyone. This is something that's pretty different than the way most of the financial system works. If you're in America, you can't really just start trading like Chinese equities. If you're in China, you can't really just start trading European bonds, things like that. Whereas because cryptocurrency is a decentralized asset, it can be traded anywhere and kind of by extension of that because decentralized exchanges are just open source code that runs on

the blockchain. That's available to everyone and that's fairly different than most of what people are used to in finance.

**[00:13:34] JM:** Tell me about what happens when a transaction occurs on dYdX.

**[00:13:41] AJ:** Yes, absolutely. Maybe I'll talk about the system that we built for our new product, which even goes beyond Ethereum. Then I'll talk about kind of the life cycle of a transaction or trade that happens in that new system. The new system that we built and we built this in partnership with the company called StarkWare is a layer 2 solution. Layer 2, again, just means like a new kind of more scalable blockchain solution that's based on top of Ethereum. The thing that we're building in partnership with StarkWare is what's called a zero-knowledge roll-up chain.

Basically, the way that this works is you take a bunch of data. In our case, let's say this is all the trades that you might want to settle on dYdX for a given time period. Then every once in a while, so say once an hour, you have kind of this list. You have this buffer of all the trades that you want to settle on the network and you run this thing called a zero-knowledge proof on them. This is basically just some very fancy math proof, which not that many people, including myself, in the world understand. But this is kind of why we partnered with StarkWare to go out and build the solution. But anyways, you have basically this array of trades. You run this zero-knowledge proof on them, and then kind of the magic that happens is that out of that proof you get this constant size data object.

It doesn't matter how many trades went into that batch. There could be one. There could be a thousand. There could be more. This constant size data object that you get out of this zero-knowledge proof is the same size. This constant size data object, what it is, it's what's known as a zero-knowledge proof. You can basically take this proof and you can present it to a smart contract on the blockchain which knows how to verify the proof. Importantly, verifying the proof is much, much less computationally intensive than generating the proof. So it can be done on chain, and this happens just on layer 1 Ethereum by a smart contract, which is running on top of Ethereum.

Then once that proof is submitted to the chain, you can basically update the state of all of the accounts in the system which are stored in a Merkle tree, and I can dive more into that if there's interest. But basically, you just store the root hash of that Merkle tree in the smart contract on chain, so you only have to store a very small amount of data, and that's where all the scalability comes from. The reason why this was really important to us and important to our traders is one of the things people are probably very aware of with blockchain and specifically Ethereum is that it's not very scalable, which is definitely true.

This kind of manifested itself for us in terms of our users having to pay very high fees to get their transactions and their trades included in the blockchain. This has gotten pretty ridiculous honestly, so it's usually costing these days about 100 to upwards of 1,000 dollars to execute a single trade on our old system on layer 1 Ethereum. That's what really drove the need on our side for moving to this new more scalable solution which, again, scales by this zero-knowledge proof technology. That's how it works right now.

I guess maybe just to summarize the life cycle of a trade on new dYdX on this layer 2 system. A trade comes in. It's matched with – Say you want to buy like one synthetic bitcoin. You'll be matched with somebody who wants to sell one synthetic bitcoin for the best possible price. Then your trade will be buffered in kind of that batch for about an hour or so. Then once an hour, the zero-knowledge proving engine will come along, will prove that your trade happened, will basically prove that the cryptographic signatures that you provided saying that you're willing to enter into this trade are valid, everything like that. Then that'll be submitted to our smart contract on the blockchain, which will update the state to reflect yours and everyone else's in that batches trade.

**[00:17:26] JM:** Are there any race conditions that can occur that can be problematic across the platform?

**[00:17:33] AJ:** Yeah. There aren't any race conditions right now. The way that dYdX operates is that we're what's known as a hybrid exchange. That means we have some decentralized components, which are all the smart contracts which run on the blockchain which is kind of the stuff I've been discussing so far. Then we also have some centralized components, which are basically just servers which run on AWS like any normal company. But these centralized



components, what they are is they run our order book and our order matching service, our matching engine. Because we run the matching engine which happens off chain just on AWS, we really don't have any finality, don't have any race condition problems because we can just order trades as they come in. Then they'll just be settled on that once per hour basis through the zero-knowledge proving system.

**[00:18:20] JM:** Can you explain the division of labor between what you use AWS for and what you use the Ethereum blockchain for in a little more detail?

**[00:18:28] AJ:** Yes, definitely. In terms of what we use AWS for, so let's maybe just dive into an example of a little bit more detail about what actually happens when a user places a trade on dYdX, and I think that'll be helpful. Say you're on the dYdX interface. You want to buy, like I said in my example, one synthetic bitcoin. You place an order to make that buy through the exchange. The first thing that happens, and this happens not on our servers but it happens in basically the client code of our website, is the client code of our website will call into the user's wallet, and it will generate a signature, a cryptographic signature basically attesting to the fact that the user wants to make this trade. The thing that's being signed in the signature is the fact they want to buy one synthetic bitcoin, also kind of the worst price that they're willing to buy it at or the highest price they're willing to buy it at.

Now, we have this cryptographic signature and we have the details about the order. The next thing that happens is the website calls our API, and this is just a normal rest API and a post endpoint, and it hits our servers. Basically, what happens once this order hits our servers, the first thing we do is we validate the signature just to make sure it's valid. Then the second thing that we do is we route the order to our matching engine. What the matching engine does is the matching engine just stores all of the state of the currently active orders on the order book. Basically, what an order book is is just an ordered list of the prices and the amounts that people are willing to buy or sell at at any given time.

This new order to buy one synthetic bitcoin comes and hits our matching engine. It will look at the sell side of the order book because a buy has to be matched with the sell to make a trade. It'll look for the best possible price on the order book, so basically the first order in this ordered list that we have stored in our matching engine, and it will match that with the order to buy that

one synthetic bitcoin that just came in. Then it will just update everything, which is kind of a cached version of the state that exists on Ethereum in our Postgres database. Then once we've cached that on our side, then and only then does kind of the blockchain of the Ethereum stuff start to happen.

After this point, we've kind of reflected this trade in our system, but we need to actually settle the funds on Ethereum. What happens after this is our servers will call into StarkWare servers. StarkWare is just another company, again, that we partnered with to build our layer 2 solution, and it will send to them basically this trade. What this trade is is details about the trade, so what amount is being traded at what price. Importantly, it will also include the signatures for both the buyer and the seller. Basically, there's a one-to-one mapping on dYdX. Every trade or every order has an associated signature with it, cryptographic signature.

Then all the stuff that I mentioned before starts to happen in terms of StarkWare batching that, kind of buffering it, that trade for up to an hour. Then it'll run the zero-knowledge proof on it and will submit that to the blockchain once an hour to basically settle our state with what actually exists on Ethereum.

**[00:21:37] JM:** When I purchase a position on a dYdX contract, does that mean that somebody else is taking the opposite position?

**[00:21:46] AJ:** Yes, exactly. That's exactly what it means. Contracts on dYdX are always zero sum. For every one unit of long, there's one unit of short. So for every one buyer, there's one seller.

**[00:21:58] JM:** Is there consistently enough volume on each side of the order book that your orders are getting filled reliably?

**[00:22:05] AJ:** Yes, absolutely. You're kind of getting at the concept of liquidity, which is critically important to an exchange, so it's a great concern. If you want to buy, there has to be somebody sitting there willing to sell, and kind of one of our main classes of customers at dYdX are what's known as market makers. Market makers are just operating basically trading bots, programmatic trading bots, which are placing resting orders on the order book.

In my example, the buy order that or the user placed had to be matched with that sell order. Well, like 99% of the time, that sell order is going to be placed by one of our market makers. What market makers are doing, and this is actually one of the other pretty interesting technical challenges with running any type of exchange, is they're programmatically placing resting orders on the order book. Suppose the price of the dYdX synthetic bitcoin contract were \$50,000. A market maker may place an order on either side to, say, either sell at \$51,000 or buy at \$49,000, which is a good deal for the market maker, right? Because if the price is actually \$50,000 and on either side they get hit, they're kind of making a small spread when they make those trades.

But the thing that we have to deal with as an exchange is market makers are placing and canceling these orders extremely rapidly, like multiple times a second, often times tens or like hundreds of times a second. These market making bots are hitting our rest API. Our posting orders are kind of canceling orders all the time. Those are always being routed through our matching engine. Also very importantly, none of this stuff until trade execution actually has anything to do – Well, or actually is it's basically not settled on the blockchain, and we don't need the scalability for that.

The types of events that are hitting our matching engine I would say are just probably like 99.999% just placing orders that don't match with another order on our order book. Suppose you were a market maker. You wanted to place your offer to sell one synthetic bitcoin at \$51,000, but nobody wants to match with you. Nobody wants to buy it from you yet because the price is only 50,000. In that case, it just rests on the order book, and that's what 99.999% of the events on dYdX or any exchange are, just market makers placing and canceling and updating the prices for these various orders.

**[00:24:31] JM:** When I make a trade through dYdX, where is that trade being stored? What is the database that is storing the information of who has what trades?

**[00:24:42] AJ:** Yeah. There's kind of two different states of the world. The first state of the world is off chain and, again, that just exists on our servers, and basically we run a Postgres instance on AWS. All of the information for all the trades that have ever been placed or have ever been

matched exists in that database, but that database doesn't actually have control over anyone's money, and all of the settlements for the trades happens on Ethereum. So all of the settlement logic and specifically what everybody's balances are, that's the only thing that exists on chain or exists on Ethereum. By only putting kind of a subset of the information like the important information in terms of like what is your balance and how much money do you have on Ethereum, that's one of the other ways that we get a lot of scalability.

**[00:25:28] JM:** How does dYdX fit into the broader world of DeFi?

**[00:25:34] AJ:** Yeah. The broader world of DeFi right now I'd say is mostly still dominated by spot exchanges. But, again, just being kind of a fancy word for regular old buying and selling of cryptocurrencies. I think the reason that this is usually in any type of financial market, and this really happened in centralized exchanges on cryptocurrency, there's like things have to be built in terms of like, first, there's the asset. Then there's kind of spot exchanges which trade the asset. Only after that can there be more advanced types of exchanges like derivatives and margin trading exchanges.

In terms of how we fit into the landscape right now, I'd say we're one of the market leaders in terms of decentralized derivative. But kind of the decentralized market right now by volume is still mostly dominated by spot exchanges. Specifically, Uniswap is the most popular decentralized spot exchange right now, and there are a couple others like ZeroX and others that are popular as well. Right now, our volume is probably roughly like one-tenth or like one-fifteenth of Uniswap or so. But we're really excited still about kind of the derivatives market, and I think it's pretty likely that the same thing that will happen in the centralized cryptocurrency market will happen in the decentralized or DeFi markets as well, where at some point we'll see a shift in terms of the volumes for derivatives just really overtaking those of spot volumes. It's mostly just kind of derivatives exchanges, spot exchanges.

The other thing that's really popular in DeFi is borrowing and lending protocols. So these are things like Compound or Aave are probably the most popular ones at this point, and those are valuable as well. But our niche is kind of the more advanced financial products.

**[00:27:14] JM:** You did mention a few other products there such as Aave. Can you contrast a little bit more how dYdX compares to the other DeFi products on the market?

**[00:27:25] AJ:** Sure thing. I guess just diving in, let's maybe just take a few specific examples. Like I said, I think the leader in the spot decentralized market right now is Uniswap, and then we can take Aave as an example of kind of the decentralized lending market. Maybe I'll take Aave first because that's a little bit easier or like a little bit different. For Aave and more generally just a lot of decentralized lending products, the way that they work is through a pool-based model. There's basically some decentralized market where users can borrow and lend a bunch of different cryptocurrencies, and there's a dynamic interest rate for this market that users can borrow and lend at.

The important concept that all of these are built on top of is the concept of over collateralization. That means and basically the way that you can lend to somebody without understanding who they are beyond just have an Ethereum address is you can make them lock up more money to make a borrow than the borrow is worth. Then if they default on their borrow, you just take the rest of their collateral. That's kind of the way Aave and most of the other lending protocols work.

In terms of the differentiation between us and Uniswap, the leading spot exchange in DeFi, so Uniswap is built on what's known as an automated market maker model. This means that all the stuff I was talking about with our order book and our matching engine doesn't exist on Uniswap. Instead of trading with market makers on Uniswap, you're basically trading with the contract itself or trading with people who have deposited to offer to basically market make as per an algorithm that's just coded directly into the smart contract on Uniswap.

This has tradeoffs with the order book based model. The main trade-offs are that it's a lot easier to just bootstrap and kind of build liquidity for new markets because it's way easier to be a market maker in the Uniswap or automated market maker scenario. All you have to do if you want to provide liquidity is you don't have to deal with the complexities of building this really complex trading bot kind of that I was alluding to before that our market makers run. All you have to do is deposit to the smart contract, and then the smart contract will market make your funds with the algorithm that's coded directly into the smart contract.

The other thing, main difference between us and a Uniswap is what I've been talking about for the most part. It's that we offer synthetic products and specifically just more advanced financial products. We also offer leveraged trading on the exchange, which is the concept where you can kind of multiply your gains or losses through trading these synthetic contracts.

**[00:29:58] JM:** How do these different types of DeFi products get combined to build more powerful trading systems?

**[00:30:06] AJ:** Yeah, definitely. I think composability is really important to a lot of different DeFi products. From my perspective, it just kind of makes it a lot easier to build certain features. For example, one of the features we wanted to build at dYdX was the ability for users to deposit different types of funds into dYdX. One of the kind of properties of our smart contract is that we only support one type of collateral or one type of like value locked in the contract, and that's what's known as USDC. USDC is just what's known as a stablecoin, which is a cryptocurrency that's targeted at a price of one dollar. That's the only coin that our smart contract knows about basically.

But suppose somebody came to dYdX, and they had a different cryptocurrency other than USDC, but we still want to make it as easy as possible for them to onboard to dYdX. The thing we can do is we can integrate with a totally third party decentralized exchange. What we're integrating with is called ZeroX, and ZeroX is basically a spot exchange. A spot exchange means that we can trade this other coin that somebody comes to dYdX with for USDC which, again, is the only coin that our smart contract knows about kind of atomically when the user deposits to the platform, and we can make that all happen under the hood.

We're able to do this because this product that ZeroX has come out with it is just a smart contract itself. It's totally permissionless. We don't really need ZeroX's permission or anybody else's to start using it, and it's really easy to kind of combine smart contracts together as building blocks. For example, you could create this deposit transaction where, say, like the thing we want to do in this transaction is like the user comes to dYdX with Ethereum. We want to use ZeroX to trade that Ethereum for USDC. Then we want to deposit that USDC to our smart contract.

You can make this all happen in one transaction or one kind of user experience interaction through just combining these smart contracts together. So you could, say, basically make a proxy contract or kind of like a third contract that just sits in front of the dYdX contracts and the ZeroX contracts, which once it gets the transaction sent to it, the first thing it'll do is it'll take the Ethereum that was sent along with that transaction. It'll call into ZeroX. It'll trade that Ethereum for USDC. Then the second thing it'll do is it'll call the dYdX smart contracts deposit function and then deposit USDC for the user. It's fairly complicated. But from the user perspective, all they know is they deposited Eth, and now they have 100 USDC in their dYdX accounts, and it's really easy for them.

It's actually like kind of once you understand the concept not that hard to chain these smart contracts together. From Ethereum and from solidity, you can basically just start calling any arbitrary smart contract which exists on Ethereum, and you can kind of use this concept of a proxy contract, which is really able to chain together just multiple interactions for different smart contracts into one atomic transaction.

**[00:33:11] JM:** Now, my understanding is that if you write a smart contract that chains together several different protocols and you, for example, margin trade and then you use your margin to do other things and then you can't pay back your margin, the entire trade can get unraveled. The entire transaction can get reversed. Your entire smart contract can get reversed. Explain this like phenomenon of how a transaction can be reversed in a little more detail, assuming I have it right.

**[00:33:47] AJ:** Yeah. You do have it exactly right, and I think maybe the best framing to think about blockchain transactions for software engineer certainly is the concept of a database transaction. Suppose like in SQL, like if you just start making a transaction and then you do some inserts to the database and then, say, you just like try to update a record that's not there or something like that, the entire transaction will be reverted, and it'll be as if those inserts never actually happened. It's basically exactly the same thing for a blockchain transaction where, say, in your example, if you want to write a proxy contract that, say, chains together like borrow from Aave. Then you, say, borrow some funds into your account, then you make a trade on Uniswap. But, say, there's not enough liquidity on Uniswap or something like that, and your Uniswap part of the transaction reverts.

In that example, like all of the state changes that you would have made to Aave never actually happened, very much in the same way that just kind of a database transaction in SQL or something like that. All those inserts that you made like never actually happened, but still that transaction gets mined onto the blockchain, and you still have to pay fees for it and things like that.

**[00:34:57] JM:** What is a bug in dYdX that was incredibly hard to solve?

**[00:35:04] AJ:** Yeah. That's a good question. I think we've been fortunate to not have bugs in our smart contracts, and that kind of gets at some of the testing practices that I was alluding to before, a bug that's been really challenging to solve. I think – So maybe this isn't exactly a bug, but one thing that has been challenging for us to continue to solve is more of the scalability of our off-chain components. It kind of gets at what I was talking about before with just a really high rate of throughput that we have to serve for our market makers, and this is just kind of a really common problem that anybody building an exchange might have to deal with.

The basic problem is that the matching engine component of our exchange has to be very fast. Also, kind of by the rules of an exchange, the matching engine has to be deterministic and serial. The serial part is what makes it especially hard. Basically, the reason that it has to be serial is because you have to honor kind of the first in, first processed API request that you get. For example, if you get an API request to place an order at like time one, and then later at time two you get a different API request to place the second order, you have to process in your system kind of that order that happened at time one before the order that happened at time two.

This basically means that from the matching engines perspective, everything has to be processed serially. So the matching engine has to be very fast. This isn't exactly like an original thought. I mean, there's a lot of like algorithms, a lot of like software engineering that goes into building matching engines anywhere. But it's always kind of challenging to think about like what data structure you want to use to store your order book, things like that, as you're processing these orders. One of the things we had to think a lot about from a software engineering perspective was how do we make these matching engines really fast, and that was a hard technical problem to solve.



**[00:36:59] JM:** Can you talk more about the differences between your two products, perpetuals and margin?

**[00:37:05] AJ:** Yes, definitely. Our two products right now, like you said, are perpetuals, which are the synthetic contract that I've kind of been discussing, and then margin trading, which is not a synthetic contract. Basically, what margin trading is is it's leveraging, borrowing, and lending to basically just allow users to take on leverage. The difference between them is just that that the perpetuals are synthetic contract, and the margin trading is physically settled. When you're margin trading Ethereum or whichever other currency, you're actually buying and selling real Ethereum, but you may be borrowing some amount to do that. Maybe walking through an example would be kind of the easiest to illustrate this point.

Suppose you wanted to go 5x leverage long on one Ethereum on the margin product, 5x basically just means that if the price of Ethereum moves 10%, your position will move five times that. So your position will move 50%. Say you want to do that on the margin product. You come to the exchange with your, say, one Ethereum as a margin deposit, and you want to buy four more to kind of turn that into the price exposure of five Ethereum. Basically, what you can do, and this all happens automatically and happens under the hood on dYdX, is that you can borrow the amounts that you need to buy the remaining four Ethereum and you can put up your one Ethereum that you came to the exchange with as collateral to do that. You come to the exchange with one Ethereum. You make a trade to basically buy for more Ethereum. Now, you're left with a positive amount of five Ethereum in your account, but you had to borrow some amount of stable coin. Usually, that's a USDC on our platform to be able to do that. Now, you have this borrow of however much Ethereum based on your trade price that later you'll have to pay back to kind of unwind your position. That's how margin trading works.

Synthetics are completely different. When you're trading synthetics, it's more has to do with kind of what we were talking about before where it's zero sum. For every one unit of someone who's long the contract, there's one unit of someone who's short the contract. The only way you can buy into a contract or, say, go along a contract is if there's somebody that's willing to sell it and go short. That's kind of the way the perpetuals product works. Like I said, there's no actual bitcoin or Ethereum that are involved in these trades. It's just the stable coin, just the USDC.

**[00:39:33] JM:** Could you explain the term cross margining and why that's important for layer 2 systems?

**[00:39:38] AJ:** Yes. Good question. What cross margining means, it means that you can have one pool of collateral or one kind of margin account that margins all of the positions that you might want to take on dYdX at the same time. For example, you could, say, deposit 100 USDC to dYdX using cross margining. Then you could use this 100 USDC to collateralize both a position in Eth and a position in bitcoin at the same time. The counterpoint to this is you could have an exchange that only offers isolated margining. For isolated margining, what that means is each margin account can only collateralize one position at a time or one market at a time.

If you try to do the same thing where you want to take a position on BTC and Eth at the same time with isolated margining, then you would have to put down a pool of collateral for each of those markets separately. So you might have to deposit, say, like 100 USDC to the bitcoin market and 100 USDC to the Ethereum market totally separately. The advantage of cross margining is that it's much more capital efficient because you only have to put down collateral one time, as opposed to one time per market that you want to trade on. So it's much more capital-efficient for traders. Importantly, it's also much more capital-efficient for market makers, even way more than for normal traders.

Imagine if you were operating an exchange and it only supported isolated margin, which was the case on our older kind of version one perpetuals product, and you are a market maker that wants to quote all of the products that dYdX offers. Let's say like in that example dYdX has 50 markets on the exchange, which isn't that unreasonable. If you had isolated margining and you're a market maker, you would have to put down collateral for every single one of these 50 markets separately if you want to quote on and provide liquidity for all of these different 50 markets. That's really untenable. Market makers aren't going to put down collateral 50 different times to quote on like all of the different markets.

This was from a product perspective what was really holding us back from being able to launch a lot of new markets on the platform. But now with cross margining, it's actually really easy to launch new markets from a liquidity perspective because as a market maker, you can just

deposit your collateral one time. Then you can just start quoting on all of the different, say, 50 markets that dYdX wants to support, and it's much more capital efficient from that perspective.

**[00:42:12] JM:** What are you trying to build within dYdX today? What's on the feature roadmap?

**[00:42:17] AJ:** Yeah. As I've been talking about, the main thing we've been working on for the past six months has been this layer 2 product, and it's not just the layer 2 that we've been working on. It's not like we just took our old product from layer 1 and port it to layer 2. We really rebuilt everything from the ground up, so we built the entire front end, completely redesigned, rebuilt our entire back end, so totally new matching engine and everything like that. Of course, the protocol smart contracts as well.

We are currently live in private alpha for that and we'll be launching publicly with that in just a few short weeks. Most of the team is probably going to continue to work on that product for the next two to three months or so at least, just adding new features, things like that. Kind of the next things we're going to be focusing on for the company as a whole is just decentralizing more parts of dYdX. This kind of means like how can we no longer operate the order book and order matching in a centralized way or kind of on these AWS servers. How can we make dYdX only run on the blockchain? This is a pretty hard problem as well because as I was kind of talking about, a lot of the components for our order book and our matching engine kind of require like 100 to 1,000 times more scalability than the actual order settlements do because, again, like most of the activity on a given exchange is not trades. It's just market makers kind of placing and canceling orders all the time.

On our roadmap, we're going to be focusing on solving this problem over the summer kind of from an architecture perspective. We do have a couple of different ideas in mind in terms of this, how we can accomplish this and then spending probably the next like six to nine months or so after that just going out and building like the version four of dYdX, which will incorporate just this full decentralization.

**[00:44:06] JM:** Give me your perspective on the future of DeFi right now. When does it go from people basically speculating to products that may have more real world applications?

**[00:44:22] AJ:** Yeah. It's a great question. I mean, I think for kind of the entire space, I think it will continue to be very trading-dominated for at least like the next five years or so. There are certainly some other interesting things that are going on, but I really think that the very first killer use case for cryptocurrency in general is trading and is speculation. It's kind of the perfect thing to trade or like the perfect things to speculate on if you really think about it, which kind of has to do with what I was talking about before with like how easy it is for exchanges and users to access cryptocurrency and especially access it globally. It's really this global trading market, and that doesn't exist for very many other products in a really efficient way.

I don't really see it moving away from trading anytime soon. Some of the other interesting things that I think are going on in DeFi right now, there's a lot of hype around this thing called NFTs or non-fungible tokens. This is basically at a high level like taking things like trading cards, stuff like that and tokenizing them on the blockchain. Again, this just makes it a lot more accessible. It makes it a lot easier to trade these types of NFTs for real world value, and it's just kind of a 10x improvements over the market for artwork or the market for like trading cards in game like video game items, stuff like that operates today. That's something a lot of people are excited about.

I think in terms of like some of the stuff people were excited about with blockchain in like circa the ICO boom in the 2017 era, so things like building a decentralized social network or like a decentralized Uber or something like that, honestly I think that's still pretty far away. Just like the blockchain even with some of the new scalability solutions is not quite there yet in terms of being ready to handle like the scalability that a lot of those technologies need. I really think of kind of blockchain as just decentralizing value, whereas the Internet, kind of the main innovation there was decentralizing information or like access to information. Whereas for value, I mean, that's basically just money, and that's why I think we've seen just blockchain be really used for disrupting the finance industry to this point.

**[00:46:34] JM:** Well, Antonio, it's been great talking. Is there anything else you want to add about dYdX or the space in general?

**[00:46:41] AJ:** Yeah. No. I think those were great questions, and thanks so much for asking and diving into the technical details there. Just in terms of things to add, so for dYdX right now, we are live with our current layer 2 product on mainnet. If you're curious to check that out, just head

over to dYdX.exchange. We are hiring for a bunch of different software engineering positions, so mobile front end and full stack as well. I think it's just kind of a really interesting space to be in as a software engineer. It certainly captivated me for the past five years.

One of the interesting things I think about it is that everything is so new and is changing so quickly, so you always have to be on top of the game in terms of like understanding new scalability solutions, coming up with solutions to things that like people have never built before. Nobody's ever built like a decentralized derivatives exchange before dYdX has, so I think it's pretty interesting from that perspective. But if you're curious to learn more or curious to kind of comment and chat with us or ask us any questions, just head over to our website, dYdX.exchange. Join our discord. Apply for any of our job listings or check out the product there.

**[00:47:46] JM:** Awesome. Thanks for coming on the show.

**[00:47:48] AJ:** Great. Thanks so much for having me.

[END]