The Science of Blockchains

Ms. Jalashree D. Trivedi
Student, Computer Engineering Department, L.D College of Engineering, Ahmedabad, Gujarat, India

Prof. Karishma A. Chaudhary
Professor, Computer Engineering Department, L.D College of Engineering, Ahmedabad, Gujarat, India

Prof. Tushar.JRaval
Professor, Computer Engineering Department, L.D College of Engineering, Ahmedabad, Gujarat, India

It's interesting that traditionally, cryptography primarily had one primary customer, which was really the internet. And what's interesting is if you look at cryptography on the internet, what it was used for was primarily for securing communication.

So if I send my credit card to Amazon, I want to make sure that nobody can steal my credit card on its way to Amazon. So traditionally, cryptography, sort of on the internet, primarily focused on what we call confidentiality. How do I send messages across the internet so no one can eavesdrop and no one can tamper with those messages? So this raises cryptographic mechanisms. This uses cryptographic mechanisms like encryption, key exchange, and things like that. What we've seen with blockchains is that all of a sudden, cryptography now has a new customer. These blockchains are basically a huge consumer of cryptographic technology, which is why it's so much fun to work in this space. But what's interesting is that blockchain cryptography is a little bit different from internet cryptography. So on the blockchain, as you know, most data is just available in the clear. We're not encrypting anything. When you write transactions to the Ethereum blockchain, the transactions are just written in the clear. What we care about is not confidentiality. But rather we care about integrity. So really, when you look up about blockchain cryptographic primitives, by and large, we look at things like signatures, commitments, zero knowledge proofs. These are all integrity mechanisms to prove that the data is valid, not so much about encrypting the data. So it's been very interesting to see that the cryptography that's used on blockchain is somewhat different from the cryptography that's used on the internet. And as a result, that led to a wealth of new problems to work on and new things to think about. But more generally, I want to mention that there's this whole area which we call the science of blockchains. So science of blockchains is I use that term to differentiate it from what traditionally is called blockchains, which you kind of hear all the noise about in the press. The science of blockchains is the technology that underlies all these blockchain systems.

And it's an incredible collection of fields. In particular, there are lots of research questions that are being raised by the science of blockchains in economics, and distributed systems, of course, and cryptography, and programming languages, verification, and so on and so forth. So it's kind of a collection of super interesting areas. And I have to say, it's a fact that I don't think that there's ever been one area that kind of connects all the areas that I listed here into one discipline. And that's kind of what makes this kind of a fun area to work in. I have to say that, from my experience, the community is a very welcoming community. If you have good ideas in any of these areas, people are very eager and interested in collaborating with you, and maybe even deploying what you propose. So it's kind of a very friendly community. And it's been kind of a lot of fun to work in this space. So what I want to tell you a little bit about today is basically sort of where I see this space going. And we'll talk a little bit about our work. Before we do that, I do want to mention that because of this kind of confluence of scientific areas, at Stanford, we created this Stanford Center for Blockchain Research. It's called CBR, the Center for Blockchain Research. There's a bunch of faculty involved in it. If you look at the list, you can see they're from all over the University, focused on, again, the underlying technology behind blockchains. And again, it's been a lot of fun to collaborate with all these folks. And we run a lot of activities which are open to the public. All of you are welcome to attend. In particular, at the end of August, we're running the fifth Science of Blockchain Conference. This is a free open conference for anyone to attend. If you just search for the Science of Blockchain Conference, you'll find a website right away. You can register. and come if you'd like. We also run seminars.

So you can join our mailing list to hear about seminars. And again, these are free and open to the public. So we're very happy if you'd like to attend any of those. I also wanted to mention very briefly that we run a bunch of classes, technical classes, on blockchains on campus. So the classes are kind of an introduction to blockchains, which is called CS251. In fact, that's a class that's starting in a month and a half. I think this is the class that Petra mentioned. Again, this is televised. You can take it remotely. If you're interested in joining in the winter, we teach our cryptography class. And in the spring, we teach a consensus a class specifically on consensus. Maybe I wanted to just drill in, for just a second, and tell you about the blockchain class. So by the end of this class, the goal is to teach the students how blockchains work. If they need to, they can kind of even build their own blockchain. But primarily, they should learn how to write blockchain applications, applications that are run on top of a blockchain. I kind of listed the topics here. I don't think I'm going to go through them very carefully here. You can see them on the website. All the information is available on the course website. There's quite a lot of interest in zero knowledge proofs and proof systems in general, and their applications the blockchain. So the second half of the course, we spend quite a lot of time on that. So if you're interested in learning about any of these topics, and learning how to build your own blockchain applications, you're very welcome to take the class. It's available to be taken remotely, as I said. OK, so with that, let me switch gears, and talk about what I wanted to talk about, which is I guess we'll start with a kind of an overview of what blockchains are. I'm not sure what everybody's background is. So I wanted to start from the beginning, and kind of tell folks how I view blockchains. And then we'll talk about some of our recent work.

So what is a blockchain for? What is it good for? So the short answer that I usually give is that blockchains really are a way for a large number of people to coordinate when there isn't a trusted party isn't a single trusted party that everybody trusts. The corollary of this, of course, is if there is a single trusted party, typically you actually don't need a blockchain. And so for example, if everybody is happy to use, I don't know, Google as a trusted party, there's no need for a blockchain. You can just use the Google systems, the Google databases, and store all your data there. And everybody is happy. If there's a government that everybody trusts, then you could just use government services. There's no need for a blockchain in those cases. But if you're in an environment where, actually, don't want to be beholden to a single party, that's kind of where blockchains might be useful. And that's where they typically live. Now, there are currently four application areas that blockchains have been useful for. Yes, these are four applications where they seem to be a good fit. Maybe in the future, we'll see more. But for now there are these particular four application areas. So let's go through these slowly. The first one I'm sure you're all familiar with, is what's called DeFi, decentralized finance. Basically, this is the use of blockchains for the payment system. Yeah, payment systems and all the financial assets that go with that. I think you guys are fairly familiar with what DeFi is. I don't think I want to get into too much details here. The second application area is what I call managing digital assets. In general, any digital asset, be it art, or sports cards, or any sort of thing that you want, any sort of digital asset that you want to maintain ownership of and you want to prove provenance and originality of, you can kind of manage on a blockchain. Often, these are called NFTs. And it turns out there's actually a good reason why you want to manage these things on a blockchain.

I'll talk about that a little bit later. So NFTs, and more generally, digital assets, are kind of the second application area for blockchains. The third application area is basically games. And it's quite interesting that there's a large fraction of the gaming industry that's starting to use blockchains now in traditional games. And again, the idea is to sort of manage game assets on a blockchain. So if you win a sword in a particular game, the fact that you own that sword is recorded on a blockchain. And no one can take it away from you, unless you give it away. And the fourth application area is what I call decentralized organizations, DAOs. Maybe the easiest way to explain what a DAO is, is just to think of the real world analog. And the real world analog is basically a partnership. So a bunch of people come together. They want together to make decisions on how to manage assets, how to manage funds, and so on. So they can form this DAO, and together, vote and decide on what the DAO does. And there are many types of DAOs. We'll talk about those in just a minute DAOs for investments, for charity, for collecting art, and so on and so forth. And one of the interesting questions is basically, how does governance work? That is, how does the group of members of the DAO how do they actually make collective decisions? And there's a lot of active research going on here. And again, this is kind of one of the topics I wanted to talk about today in greater depth. OK, so those are kind of the four application areas. Maybe there will be more in the future. So I would challenge you to think about other application areas where the technology would be a good fit. And maybe next year when we give this webinar, the list will be longer. But these are kind of the current four application areas that we currently see.

OK, so let's go back to the beginning and just talk a little bit about, what is the new capability that blockchain enable? What is it about? Why did it happen now? Why didn't it happen 20 years from now? Why didn't it happen 30 years ago, and so on? So it turns out, of course, as you know, the first blockchain that came out is called the Bitcoin blockchain in 2009.

And what's interesting about Bitcoin is not the fact that it introduced the Bitcoin currency. That's sort of a derivative of what Bitcoin did. Really, the new capability that Bitcoin gave us is sort of something we didn't have before, which is what I call a public, append only data structure. So it's a data structure where anyone can write to that data structure. And anything that's written to that data structure will stay in that data structure, and can never be removed. At least in principle, that's the idea behind what Bitcoin gave us. So it's an append only data structure that anyone can write to. And once you write something to it, it'll never be removed. How do we secure that data structure? Well, how do you build such a thing? Well, you build it basically by replicating the data structure across many, many different machines. And the minute you start replicating a data structure across many different machines, you start to get problems of consistency. How do we know maybe one machine has one version of the data structure, and another machine has a different version of the data structure. How do we know which one is the correct one? And so that's exactly where the consensus problem comes from, which is how do we replicate a data structure across the entire planet, and make sure that all the replicas are actually in sync? So security is done by replication, and more importantly, by incentives. So everybody is incentivized to actually make sure that the data structure actually satisfies the property that it claims to satisfy. Once you have such a data structure, you can actually use it to build a currency, right? So if I have an apend only data structure, and let's say that I want to pay $5 to Petra, I can record that transaction on the data structure.

And no one can remove the fact that the payment was made. Yeah, once it's recorded, it will stay there forever. This is why a cryptocurrency was a very natural application for this data structure. But I do want you to think about the data structure as the primary thing that Bitcoin gave us. The currency is enabled by this data structure. So that's Bitcoin. The next big evolution in the space there have been many ideas in the space. But the next big step in the space is the development of Ethereum back in 2015, which introduced what we call the Blockchain computer. It's a fully programmable environment where you can run arbitrary applications. I have to say that in our course, we kind of go and talk quite a lot. We do quite a lot of hands-on experiments with building programs that run on the blockchain, on this Blockchain computer. These programs are written in a language called Solidity, which I imagine many of you are familiar with. It's kind of a fun programming language to write in. It's always interesting to write applications in this language. What's interesting is it's quite difficult to make sure these applications are secure. So I have a lot of stories I can tell there. But maybe we'll move on for now. Yeah, so Ethereum introduced this idea of a Blockchain computer. So these are public programs. Anyone can inspect the code. And they can manage digital and financial assets. Yeah, that's what these kind of programs do. The other interesting aspect of Ethereum is that they were very concerned with composability, which means that different programs running on the Ethereum blockchain can actually call one

another. So if I develop something useful and somebody else wants to use it, their program can just call my program. So this was, again, a very important idea that came out in Ethereum. So of course, as you all know the story, once the Ethereum came out, after that, we saw the growth of these four application areas, basically DeFi, NFTs, gaming, and DAOs.

And there's been tremendous growth in the space. And that's kind of where we are now. All right, so this is kind of a quick overview of the core ideas that enabled these blockchains. And there's this statistic there's this graph that I want to share, that I find this graph really interesting about where the development effort is being spent these days. So this is a graph that's due to Electra Capital. And what it does, if you look at the x and y-axis so the x-axis basically measures the number of days since the project basically launched, since the first commit since the project was born. So you can see Bitcoin is the oldest blockchain, which is why it's furthest to the right. And the y-axis basically measures the number of developers who work on the project, so people who commit code relating to this project, either building applications on top of it, or contributing directly to the blockchain itself. And it's really quite interesting. If you look at the entire space, there is roughly 20,000 developers worldwide that work in the space. So 20,000 developers, that's kind of not a small number of developers. But you can see, for example, the number of developers that work on Bitcoin has somewhat stabilized. The number of developers who work in Ethereum continues to grow. This is kind of where a lot of the development activity is taking place. You can see the up and coming blockchains. There's Polkadot over here. Cosmos over here. Maybe Swan over here. You can kind of see the next level of blockchains as they grow. And the numbers are kind of interesting. So if you look at Ethereum, there are roughly 4,000 developers right now building Ethereum applications. For Polkadot, there are 1,500 developers, and so on and so forth. It's a very interesting graph to show the level of activity. And the other interesting fact here is you can see that during what's often called winter periods, you can see the number of developers as this area of the graph here. The number of developers stays roughly the same.

And then during growth periods, the number of developers grows dramatically. So it's very interesting how this tracks the ecosystem overall. But anyhow, it's, of course, useful to see where developers spend their time because probably, that's an indication of where all the activity is. So I like this graph. It's an interesting, very informative graph. OK, so the next thing I want to do is just talk very briefly about how blockchains work. And then I want to switch gears and talk about the particular topic that I'd like to talk about today. So in my mind, blockchain is divided into four layers. At the bottom layer, we have what we call the consensus layer, that makes sure that all the replicas of the data structure, they're all in sync with one another. On top of the consensus layer, often we have a scalability mechanisms. As you know, there's a big scalability issue with popular blockchains, in particular in Ethereum because it's so popular. There are so many transactions being submitted to the Ethereum blockchain that the price per transaction sometimes gets to be quite high. And so there's a need for scaling these layer one blockchains. So often, there's a scalability mechanism that runs on top of the blockchains. These are called layer two systems. And rollups are kind of the typical examples of that. Maybe if we have time, we'll talk about how rollups work in a bit more detail, towards the end of the talk. But that's basically the consensus layer, to

make sure all the replicas are in sync with one another. And whatever is written into the data structure stays there and is never removed. Above the consensus layer, we have the compute layer, which is actually what we call the blockchain computer. An example of that is the Ethereum Virtual Machine, the EVM, right? This is the operating system on top of which all the applications run. So on top of the blockchain computer, we have applications that developers build. So students in the class, for example, build a lot of EVM applications, a lot of these decentralized applications, dapps.

And they run on top of the EVM computer that's provided by the Ethereum compute layer. And then on top of the applications, we have the user interface layer, which is basically the thing that allows end users to actually interact with applications that run on top of the computer. So I kind of think of a blockchain is divided into these four layers. And so if you look at the picture in slightly more detail, you can see the blockchain computer basically manages a lot of on-chain states. Yeah, so there's a lot of storage. This is actually the data structure where things run. So the computer manages a lot of on-chain states. These applications run on this computer. And then when an end user wants to interact with an application, typically they'll interact with a cloud service, traditional cloud service hosted at Amazon or Google. The cloud service will have a copy of the state that's stored on the blockchain. So the state is stored both on the blockchain and in the Cloud. Although, the authoritative source for the data is whatever is stored on the blockchain. The copy that's stored in the Cloud is more of a cache that's just used for accelerating operations. And then end users will issue instructions and interact with the blockchain application through this interface. Now, what's interesting is actually interacting with the blockchain, it turns out to be somewhat complicated. So there's entire companies, like Infura Alchemy, that are devoted basically to making it easier to interact with these applications. So keep in mind, if you're going to build something, most likely you won't be talking to an application on the blockchain yourself. You'll probably be talking through an intermediary. And those are available as well. So that's kind of how these things work. And I guess the last thing, maybe the last general thing I'll say, is this question of, why do we even need to decentralize applications? Why do we want decentralization in the first place? So it's kind of really quite important to understand that decentralization is not a goal on its own.

Decentralization is sort of a means to an end goal. And so what is the end goal? Well, you can think of the end goal as basically making it very easy for people to deploy their own applications. For example, if you wanted to deploy an application on a bank system or an existing social network, well, if the bank has a competing service, they might refuse to deploy your application on their infrastructure because they don't want you to compete with them. In a blockchain environment, actually, there is no such thing. Anyone can deploy anything they want. And no one can prevent you from deploying your application on the blockchain. If things were stored in a centralized system, if things were stored in a centralized database, basically, the users are stuck, in some sense, with whoever controls that database. In a decentralized environment, the data is replicated all over the world. There's no one person that owns all this data. So if for some reason you're using a service, and you're not happy with the service that they give you, you can just take your data and move elsewhere. Yeah, so those are the benefits that come from having this blockchain

architecture. And finally, I guess I'll say that one is regional. The other one tends to be global. But I'll skip that. I think the most important thing is that no one can prevent you from deploying a service. Anyone can deploy whatever they want, whatever program they want, on these blockchain systems. And then it's up to the public to decide whether they want to use what was deployed or not. All right, so I'll leave it at that. This is kind of what I wanted to say at a high level. Now let's kind of dig a little deeper into how DAOs work. And in particular, I want to talk a little bit about what a private DAO is. So in the past year or so, we've been working on this concept of a private DAO.

So let me tell you a little bit about that. So first of all, let's start with what is a DAO? So again, DAOs stand for decentralized organizations. The idea is that a DAO is literally just an application. It's a decentralized application that's deployed on a chain at a specific address. So it's really just a program. It's not even a very big program. Typical DAOs are like, 1,000 lines of code that are deployed on Ethereum blockchain. And then once the contract is deployed, once the DAO is deployed, it's deployed at a specific address. Anyone in the world can actually send funds to the DAO Treasury, and contribute to the DAO, and become basically a member of the DAO by doing that. So as you can see, the DAO builds a Treasury. The DAO typically has a particular goal in mind. We'll see that in just a minute. And then once the DAO is deployed, anyone can deploy funds to the DAO Treasury. And once they become members of the DAO, they can also submit proposals as to what to do with the DAO Treasury. And more generally, they can submit proposals on how to manage the DAO, how governance should work, as we said, what to do with the DAO Treasury, whether to accept particular members or not, and so on and so forth. So anyone can submit proposals. The participants in the DAO vote on these proposals. And once the proposal is approved, basically the proposal executes. And again, there are many infrastructures available out there for for managing these proposals. Snapshot is a very common one for managing votes and proposals off chain. So anyone can submit a proposal to Snapshot. Then the members can vote. And if the vote is approved, then the proposal actually executes. OK, good. All right, so today, basically, if you look on Snapshot, you'll see there are thousands of DAOs already deployed. There's something like 6,500 DAOs already today, possibly even more. So it's actually quite a popular thing. Many DAOs have been coming into existence.

Excuse me. And so let's see what they're for. So one type of DAO is what we call a Collector DAO. Collector DAO is one that collects various arts, maybe contributes to fashion and other things. So examples are PleasrDAO, flamingoDAO flamingoDAO is a fashion DAO, for example. They contribute to fashion artists. ConstitutionDAO here's a nice example. ConstitutionDAO is a DAO that wanted to buy a physical copy of the Constitution. So they would collect that particular physical piece. And you can see, for example, PleasrDAO is a DAO to collect art. You can go look at their gallery and see all the art pieces that they've already bought. Yeah, so again, people join the DAO. They together decide what art pieces they want to buy. They decide what to bid on those art pieces. And once a decision is made, they go ahead and execute. And if they're successful in the auction, they buy the art piece. So PleasrDAO, for example, just to drill in for a second, they have about 100 members. You can see these members manage the Treasury of the DAO. It also has full time employees. Yes, so think about that. You have a contract, a program running on the blockchain, that has

full time employees. So these employees' salaries is paid by a program running on a blockchain, which is kind of interesting. Together, they deliberate over Telegram whether they want to acquire a particular piece, what to bid for that piece. And then once they reach a decision, they go ahead and use the DAO Treasury to bid and to buy the piece. So it's very interesting how this works. It literally works like a partnership where they make decisions together and jointly use the DAO Treasury. So that's one example of a DAO. Another example of a DAO is what's called a Charity DAO. And one of my favorite Charity DAO is called gitcoin, which actually supports the development of open source software. So gitcoin has about 42,000 members. The DAO Treasury is substantial. And then these members basically make proposals to decide which open source projects to support using the Treasury that they raised.

So I wanted to give you one example. Here is proposal number 21. So here, they look to spend, whatever. They look to spend a portion of the Treasury on some marketing campaign or so. You can actually track this proposal. You can see exactly the rate at which votes were collected. So this is the voting pattern on this proposal. The line indicates when enough votes were collected for the proposal to pass. And so you can see that as soon as the line was reached, the proposal passed and could be executed. But people kept on voting. What's interesting is that for this particular proposal, only 8% of the DAO members actually participated. We'll talk more about that in just a second. The participation rate is relatively low in this type of governance. And then the proposal got executed. You can see exactly the Ethereum block number where the funds were actually spent. So you can kind of track exactly from the moment the proposal was issued, to when it was voted on, to when it passed, and to when it actually executed on the chain, and the funds were actually spent. Yeah, all of this is completely visible. So Charity DAOs are another category of DAOs. But there are many others. There are what are called Protocol DAOs, DAOs that are meant to govern and manage protocols. Uniswap DAO, for example, has 29,000 members. Compound DAO has 4,000 members, and so on and so forth. So again, they vote on how to manage these systems. There are Social DAOs, which I won't talk about here. There are Investment DAOs, which also I won't talk about here. So there are many, many categories of these DAOs. And there's actually quite a lot of infrastructure that has evolved around the management of these DAOs. So one question that's kind of interesting is how does governance actually work? And that is, who can vote? How do you vote? What the exact voting mechanism is, how much does your vote count for? There are lots of governance questions around this.

If you're interested in the governance questions in DAOs, which is actually really quite a beautiful, beautiful question, there's a nice paper that just came out by one of our CBR members, Andy Hall, in collaboration with Porter Smith. Andy Hall is one of the members of our center. So Andy Hall is actually a political scientist who's very much interested in the governance of DAOs. For political scientists, DAOs are like an amazing playground because they get to use all their theories and see how those theories affect governance on DAOs. So they wrote this paper on basically how DAOs can learn from what's been learned in the last 10,000 years of human governance. Quite an interesting read, so I highly recommend reading this if you're interested in how governance works. And generally, what DAOs are,

they basically provide a wonderful platform for experimenting with lots and lots of governance mechanisms. So for political scientists, this is like a OK, so one thing that I wanted to mention is, again, the typical governance method, what's called the one token, one vote. So for every token that I own, I get to count votes. This is very similar to how votes work in the stock market. If I own stocks in a company and the company runs the votes, then the more stocks I have, the more weight my vote carries. Yeah, so basically, members receive tokens. And then they can vote using their tokens. And you can see on proposal 21 that we just saw, you can see how the different members voted here. You can look at this graph here. And you can see here, some people have many tokens. So their vote actually carries a lot of weight. Other people have fewer tokens, and so on. These governance tokens are actually implemented using what's called OpenZeppelin's Governor's contract. Again, this is just relatively simple Solidity code. You can just look it up and see exactly how the Governor's mechanism works. Here I pasted in the castVote function.

So when you want to vote, literally what happens is someone calls the castVote functions on chain, with the proposal ID, the voter, whether they support or do not support the proposal, and the reason for their decision. And so you can just look at the code and see exactly how it works and what it does. It's fairly simple code. The problem, though, with these kind of governance mechanisms, is this is what we call a direct democracy. Everybody votes on every proposal. And what we've learned is that direct democracy even the Greeks already knew this. Direct democracy doesn't scale very well to large populations. So in fact, if you look at the participation rates in different DAOs, in a different vote, you'll see that the participation rate is actually quite low. So here we have a graph that we generated. This is, by the way, based on joint work by myself and Andy Hall, with two wonderful students who are collecting this data. What this shows what this graph shows you is basically, a number of DAOs out there, very famous DAOs, for example, gitcoin, Compound, and others. And it shows you the average voting rates for each proposal. So you can see that for Compound, gitcoin, PoolTogether, Uniswap, the participation rate is under 5%. Yeah, so not too many people are voting on the different proposals. For some reason, Nouns is able to do better. Participation rate is around 17%. But still, you'd like it to be much, much, much higher than that. And right now, using the governance mechanism that's being used, the participation rate is simply not high enough. So what do we do? Well, these Governor mechanisms actually support a delegation mechanism, where you, for example, rather than voting yourself on every proposal, you can delegate your coins to somebody else. And they will vote on your behalf. Yeah, so this is intended to increase participation support, and in fact, Governor contracts supports delegation. So just as an example, if we look at Element, this is how the delegates in Element are distributed.

And you can see, there's one address here. I actually don't know who this address belongs to. But you can see, there's one address here that 300 other voters delegated their tokens to this address. And so this address now can vote and has quite a lot of power when it's actually voting. Yep, so you can see how delegation actually works. It seems to be a good design. We know representative democracy is kind of how things work. And delegation is a step towards representative democracy. But there could be many other governance mechanisms that are deployed. And all of those are sort of being experimented on in the

context of DAOs. All right, so what I wanted to do is tell you a little bit about some recent work that we did on this question of private DAOs. In particular, I'll talk about a private DAO Treasury. So the story here this is kind of a fun story. So the story begins with the ConstitutionDAO. So what is the ConstitutionDAO? I hope many of you have heard of the ConstitutionDAO. It's kind of a remarkable story of what happened. So basically, in 2021, there was a physical copy of the Constitution of the United States that was made available for sale. I think there's 10 copies of the Constitution available. And one of them was put for auction at Sotheby's. And anyone can bid. So when that happened, the ConstitutionDAO formed. And it was quite remarkable. Within five days or so, this DAO basically was able to raise $46 million to bid on the US Constitution. This money was raised from around 20,000 participants from all over the world. This, again, shows the power of DAOs running on a blockchain, that really, you can reach anyone in the world. People will contribute funds for a particular cause. I'm not exactly sure how you would get 20,000 participants to contribute funds without a blockchain-like architecture. And so this is kind of a good fit for this technology. So this ConstitutionDAO raised those funds. And now it was ready to bid for the Constitution.

So they were actually able to make their bid. It turned out that they had to reserve some funds for managing the Constitution itself if they won the auction. And so the highest bid they could bid, if I remember correctly, was around $42 million. So they could only bid 42 million. And the remaining 4.3 million were needed to manage the Constitution if they won the bid. So it turns out that somebody else actually somebody else wanted the Constitution more. And they lost to another bidder who bid $43 million. So this was more than the ConstitutionDAO could bid. And as a result, they lost the bid. They lost the auction. And they did not win the Constitution the physical copy of the Constitution. So if you step back for a second and you ask yourself, well, what just happened here? You realize that something actually went very wrong. Yeah, the problem here is that everything about the ConstitutionDAO was managed on a public blockchain. As a result, everybody knew exactly what their Treasury was. And everybody knew exactly what is the maximum bid that they could offer. And as a result, the bidder who wanted to outbid them knew exactly how much they should bid in order to outbid the ConstitutionDAO. Yeah, it's because of the public nature of the blockchain that they could not participate in this auction in a fair way, in a way that would actually not reveal what their maximum possible bid was. So as a result, they lost the auction. So the question is basically, how can a public DAO that runs on a blockchain how can it participate in an auction, when everybody knows its complete Treasury? Everybody knows exactly how much what's the maximum amount that it can bid. So this is a very interesting question. I'd say that in some sense, the auction house should have anticipated this. And maybe they should have changed the rules of the auction to be more friendly to these public DAOs. But I think we live and learn. And hopefully, this will not happen things will go better next time.

But nevertheless, it raises this interesting question of, can we build a DAO that has a private Treasury? So the world that looks at the DAO doesn't actually know how much funds it has. So you wouldn't know what is the maximum bid that the DAO can issue. So this is what we worked on. This is with one of my former students. Yeah, the knife and so let me explain

how this works. So the idea is basically to have a single DAO platform that's going to manage many, many different DAOs. So for example, there are such platforms. JuiceBox is an example of such a platform. So, many DAOs will actually run on the JuiceBox platform. It's a single Ethereum contract that manages a large number of DAOs. Now, when someone wants to create a DAO, we'll call that the DAO manager. They create the DAO. And all they do is they basically publish a public key, some short key, which we'll call a public key. They publish that on the DAOs website. And they tell people, whenever you contribute to my DAO, just use this public key. Now, when a contributor wants to send funds to the DAO, what they'll do is they'll send funds to the contract. The contract manages many, many different DOAs. And they'll use a blinded version of the DAO public key so that when you look at the actual transaction, you have no idea which DAO on the contract is the one that's receiving the funds. You'll know that some DAO on the contract received the funds. But you wouldn't know which one. And so every contributes contributes their funds to the DAO in this blinded way. So nobody really knows which DAO has what Treasury. So an observer will see the total funds that are available on the DAO platform. But they wouldn't know how much Treasury each individual DAO had. So the hope is that this will make it so that it's not possible to tell, for example, how much funds the ConstitutionDAO. And as a result, someone who wants to outbid them would not know how much to bid to outbid the ConstitutionDAO. So that's the idea.

And then later on, when the DAO manager actually wants to withdraw their funds and participate in the auction, they can use their secret key to withdraw those funds. And they can only withdraw funds that were submitted to the particular DAO that they manage. So that's kind of the idea for how the system works. So the contributions are blinded. And even though they're blinded so the public doesn't know where they went but even though they're blinded, the DAO manager can later withdraw the funds that were sent to that DAO, and cannot draw funds that were sent to any other DAO. If you're interested in the details of how this works, we actually have a blog post that we wrote about this. It's kind of a fun design, kind of a nice cryptographic trick to make this all work. So I would encourage you to maybe look at the details of the system. So this is kind of how we build a DAO with the private Treasury. But it turns out there are many other privacy questions around DAOs. For example, maybe people want to participate in the DAO. But they don't want to reveal that they're a part of the DAO. So how do we build a DAO where the participant list is private? Maybe even the number of participants in the DAO is private. Today all that information is public. It's available on the blockchain. So how do we build a DAO with private participation? Obviously, how do we build a DAO with private voting, so the voters don't have to reveal how they voted on each proposal? Today, all those votes are public. And finally, we also talked about delegations, where people can delegate their voting tokens to other people to vote. So can we do a private delegation where I can delegate my tokens, but nobody will know who I delegated my tokens to. So there are lots of fascinating privacy questions around these DAOs lots more research to do. And of course, all of that has to be done while complying with all the relevant local laws. Yeah, so generally, privacy with compliance is quite interesting from a cryptographic point of view.

Often that's done using what are called zero knowledge proof systems. And that's why again, that's why this area is so much interesting. It's so interesting because there are so many beautiful questions to work on and so many different things to design. Yeah, so I should mention that some of these questions are going to be answered generically by general privacy platforms like Aztec, and Aleo, and others. And so while we can build custom protocols to answer these questions, there are also generic solutions being developed to answer these questions. And so in the future, we will likely see many more private DAOs that are run out there than we do today. So that's all I wanted to say about DAOs. I was going to talk a little bit about scaling the blockchai and talk about rollups. But I see that I'm actually running out of time. So why don't I just jump to my conclusion slide, and just tell you my concluding thoughts. And then I'd love to open it to questions, and hear your thoughts. So as I mentioned, I kind of talked about the science of blockchain here, basically, which is kind of the underlying technology that makes blockchain work. This science of blockchain brings together very different areas of research. It's cryptography, distributed systems, economics, programming languages, verification, and so on. So these are very distinguished, very different areas of research that all come together to make these blockchains work. So it's really quite amazing that all these are needed together to make these systems work. As you can see, this design, this science of blockchain area, generates new problems in all these areas. So there's a lot of room for more ideas. And as I mentioned at the beginning, the community is very open to new ideas in this space. So everybody's welcome to participate. So hopefully, you can think about these questions too. And if you have ideas, we would love to hear about them. In addition, the other thing that's changing is there's a lot of improvements to the blockchain infrastructure that's coming actually, not only coming, that is actually becoming available.

So for example, nowadays, it's a lot easier to do to interoperate between blockchains. So if you want to send funds or assets from one blockchain to another, this is becoming a lot easier. If you want to solve what's called the data availability problem, that is, store data in a way that it's guaranteed to be available later on, there is general infrastructure now available for doing data availability. So the problem is that in the past, every project had to solve for themselves. There is now general infrastructure that makes that much easier to do. And as a result, it's becoming much easier to develop better and safer blockchain applications. Yeah, so that's another big change that's happening in the space. And the development effort is becoming a lot simpler thanks to all this new infrastructure. So despite all of these developments, there are lots of open problems left to think about. There are lots of challenges. In particular, I'm sure you hear a lot about all the thefts that are happening in the blockchain space, where there are just economic losses, where projects, due to bugs, due to social engineering, they just lose funds. And so there's a lot of interesting questions around how to make that better. There are still questions around scaling the blockchain that remain. And so there's still many, many challenges to work on. By far, this area is still quite young. And there's a lot of opportunities to make an impact. So I think I'll stop here. And as I said, I'm very, very happy to take any questions. I hope you enjoyed the talk. And I look forward to seeing you in a future talk. So thank you all. Thank you very much, everyone. Thank you so much, Dan. Thank you. It's a fascinating area of

studies, and everything that is happening in the area. So thank you so much. There are so many questions coming in.

If you have some more questions, you should go and submit them in the Q&A box. So for you, Dan, there are a few clarification questions about the DAO Treasury. And one of them was on the user end. So since the contributors are blinded, how you, as a submitter, can guarantee that your funds actually go to a specific DAO? That's one question. And the other one is on the other side. How do you actually specify which private DAO receives the donation then? How do you apply the funds? Oh yeah, great, excellent. These are terrific questions. So let me jump to that slide. So these are terrific questions. Right, so what happens is, basically, the DAO manager is going to publish this public key on a website. So say I create the ConstitutionDAO. What I'll do is on the ConstitutionDAO's website, I'll say, this is the public key that you should use to send funds to the ConstitutionDAO. And anyone can download this public key. And when they want to send the funds, they'll basically, as I said, they'll use the public key to create some sort of a blinded version of the key. And then that will be used to push the funds onto the DAO platform. And once the funds are available on the DAO platform, people will know that those funds were contributed to the platform. But they wouldn't know which DAO on the platform received those funds. Now, the DAO manager has a secret key. Yeah, that's the whole point. There's a secret key that only the DAO manager knows. And that secret key allows the DAO manager to prove to the contract that these funds belong to this DAO. And therefore, the contract is authorized to release those funds to the DAO manager. So the DAO manager can actually use its secret key to just look at all the incoming transactions, and see which ones are for the DAO, for its own DAO, and which ones aren't. So the DAO manager will very easily tell what the current balance of the DAO is, just by summing up all the transactions that came in. And then once it knows the total sum, it can ask the contract to release those funds to it, using its secret key.

But what's interesting is without the secret key, everything looks just like noise. Without the secret key, you can't tell which funds went where. And as a result, this basically gives you the privacy, the private Treasury, that you'd like to run. And again, I didn't want to go into too many technical details here. But if you want to see the technicalities of how this works, please check out this post, where we explain kind of the cryptographic mechanism that's used for this. This is based on Diffie-Hellman tuples, and what's called the Diffie-Hellman random self-reduction. So it's actually very I think it's a very cute cryptographic mechanism that makes this all possible. So yeah, thanks for asking the question. It's a great question. Great, thank you so much. And there was a very positive comment about your answer. What do you think are the main barriers for the deployment of DAO? And maybe if you think about emerging markets, and some other areas outside of the US, like, what do you think about that, about the deployment outside? Well, it's actually quite easy to deploy DAOs. There's infrastructure, actually, that will make it possible for you to deploy a DAO quite easily. So depending on the DAO you want to deploy, you can find the right infrastructure. And use that infrastructure to deploy the DAO. For example, if you want to deploy an investment DAO, there are companies out there that will make it easy for you to create the DAO, manage the investments, manage the Treasury, and so on manage the votes, and so on. So it's actually not difficult at all to deploy DAO. This is why there's over

6,000 of them already deployed. Now, one of the issues is, again, how do you run the governance? So how do you decide who can vote, what can they vote on, and so on and so forth? So that's very important to specify at the beginning, when the DAO is created.

You want to specify exactly, what is the power that the DAO members have? So that's also something that needs to be done. And then after you create your DAO, the interesting part is actually attracting members, right? How do you get people to know that your DAO exists? How do you get people who believe in the mission of your DAO to actually become members of your DAO? And so that's more of traditional marketing. And so you would have to do some marketing work in order to make that happen. And that's actually, I have to say, one of the hardest parts is basically, making sure that people who are interested in the goal of your DAO become familiar with your DAO. And then finally, once people join your DAO, now you have a bunch of participants in your DAO. And you have to manage those participants. Participants often have a lot of requests. They might not agree with everything that you do. And so this is actually called the participant management question. And so there are typically, often, DAOs will hire people to do participant management. And they will coordinate among the participants, and resolve disputes, and so on. So yeah, all of that requires people. And a lot of these DAOs actually, literally, have employees working for them. And again, I mentioned this in the talk. But it's one of these fascinating things where you have a program that's running on a blockchain. That program hires people. And then it pays them a salary to do their job. And of course, if something goes wrong and they don't do their job properly, then the DAO can vote to maybe replace them by somebody else, and so on. And so yeah, there's a lot of work that happens behind the scene to make the actual DAO function seamlessly. It's quite a lot of work. If you're interested in looking at specifics of DAOs, one of the larger DAOs out there that's actually run quite well is called the MakerDAO. And you can actually look at the decisions they run through. You can look at the mechanics of how the DAO actually runs.

Everything is public. And so that's just an example of one DAO that you can learn from to see how that actually works in the real world. Yeah, it's quite fascinating to see how this actually operates. Thank you so much. Excellent question. Thank you so much. And you covered the private DAO Treasury. And now there are some questions about the privacy on blockchain. Isn't it against the actual basic idea of the blockchain? And maybe if we look outside of DAO, why would companies be actually interested in privacy, and how they can actually achieve that? Oh yeah, that's a very good question, too. So really, in some sense, the fact that everything on the blockchain today is public, it has some benefits because we have transparency. We can see exactly what's happening. And we can see how the system kind of evolves and operates. So that's kind of on the plus sign. And the minus side, complete transparency also means that it can't be used for a lot of applications. Just to give two examples, if companies want to use cryptocurrencies to manage their supply chain so maybe I'm building a car. Say Ford is building a car. And Ford wants to pay its supplier for tires. If they did that on a blockchain, basically everything would be public. So everybody would know exactly how much Ford is paying for tires for their cars, which is typically something that companies want to hide, right? So if you're trying to use this technology for managing supply chains, your entire supply chain would be public to the whole world.

Everybody would see exactly how much you're paying for your parts, who your suppliers are, and so on. This is typically things that companies don't want to reveal to the whole world. And as a result, they simply can't use public blockchains for this purpose today. Similarly, another problem is, if companies wanted to pay their employees' salaries in crypto, today, that's quite difficult because it would mean that everybody salaries is open and public for the whole world to see.

And again, often, people would prefer their salaries not to be public. And so there is definitely a need for privacy in the blockchain ecosystem. I think I mentioned that there are a couple of companies going after the privacy aspect of blockchains. And so companies like Aztec, Aleo that I mentioned down here, they're building sort of general privacy platforms that run on a public blockchain. And yet, the transaction data remains private. So I often refer to this as running private transactions on a public blockchain. And this is actually where this whole area of zero knowledge proofs comes in. You want to prove that a private transaction follows all the rules of the blockchain. But you want to do it in such a way that it doesn't reveal the contents of the transaction. That's exactly what zero knowledge proofs are. And we talk about that at quite a bit of depth in the course. In fact, this is kind of, I think, one of the most exciting areas in cryptography in recent years. And so the fact that it has these tremendous applications to the world of blockchains is kind of remarkable. So if you want to learn more about how this zero knowledge proofs works, sometimes they're called snarks zero knowledge snarks, we devote quite a lot of time in the course to that. And so that's a good resource. Yeah, so there is a way to do things more privately. And yeah, we'll see more of these applications come along.

References

Ahmed, S. (2008), Aggregate economic variables and stock markets in India, International Research Journal of Finance and Economics, No. 14, Vol. 141-164.

Bailliu, J., Dib, A. & Schembri, L., 2005. Multilateral Adjustment and the Canadian Dollar. Ball, L., 2000. Policy Rules and External Shocks. NBER Working Paper.

Borio, C. & Disyatat, P., 2011. Global imbalances and the financial crisis: Link or no link?. Bank for International Settlements.

Brunie, C. H., Hamburger, M. J., & Kochin, L. A. (1972), "Money and stock prices: the channels of influence", The journal of Finance, Vol. 27, No. 2, pp. 231-249.

Dong, X. and Yoon, S., 2019. What global economic factors drive emerging Asian stock market returns? Evidence from a dynamic model averaging approach. Economic Modelling, 77, pp.204-215.

Döpke, J., & Pierdzioch, C. (2006). Politics and the stock market: Evidence from Germany. European Journal of Vuchelen, J. (2003). Electoral systems and the effects of political events on the stock market: The Belgian case. Economics and Politics, 15(1), 85–102. doi: 10.1111/1468-0343.00116.

Engle, F. R., Lilien, M. D., & Robinson, P. R., (1987). Estimating Time Varying Risk Premia in the Term Structure: The Arch-M Model. Econometrica, 55(2), 391. https://doi.org/10.2307/1913242.

Fan F, Su L, Gill M K, et al. Emotional and behavioral problems of Chinese left-behind children: a preliminary study, Social psychiatry and psychiatric epidemiology, 2010.

Galloway S. (2014) Instagram 2014 Intelligence Report, L1,"Instagram 2014", [online] http://www.l2inc.com/research/instagram-2014, (22.11.2015).

Gurley, J. G., & Shaw, E. S. (1955), "Financial aspects of economic development", The American Economic Review, Vol. 45, no. 4, pp. 515-538.

Henry, P. B. (2000), "Do stock market liberalizations cause investment booms?", Journal of Financial economics, Vol. 58, No. 1, pp. 301-334.

Hondroyiannis, G. and Papapetrou, E. 2001. Macroeconomic influences on the stock market.

Hooker, M.A. 2004. Macroeconomic factors and emerging market equity returns: a Bayesian model selection approach. Emerging Market Review.

Johansen, S (1995). Likelihood-Based Inference in Cointegrated Vector Autoregressive Models (NewYork: Oxford University Press).

Kumar, Arun. Indian Economys Greatest Crisis: Impact of the Coronavirus and the Road Ahead. Gurgaon, Haryana, India: Portfolio/Peguin, an Imprint of Penguin Random House, 2020.

Lobo, B. J. (1999). Jump risk in the U.S. stock market: Evidence using political information. Review of Financial Economics, 8(2), 149–163. doi: 10.1016/s1058-3300(00)00011-2.

Rossi, B., 2013. Exchange Rate Predictability. Journal of Economic Literarture, pp. 1063-1119.

Saikkonen, P. and Lütkepohl, H. (2000) Testing for the Cointegrating Rank of a VAR Process with Structural Shifts. Journal of Business & Economic Statistics.

Saikkonen, P., & Luukkonen, R. (1997). Testing Cointegration in Infinite Order Vector Autoregressive Processes. Journal of Econometrics, 81(1), 93–126.

Sharma, Tanya, and Tapan Kumar Shandilya. Impact of COVID-19 on Indian Economy. New Delhi: Shandiliya Publications, 2021.

Tishgart L.(2013) As Instagram Rolls Out Ad Platform, Brands Are Seeing Record Engagement, Business Wire, October 29, 2013 , [online] http://www.businesswire.com/news/home/20131029005603/en/Instagram-Rolls- Ad-Platfor m-Brands-RecordEngagement%20-%20.U32_-5RdWQo, (22.11.2015).

Vuchelen, J. (2003). Electoral systems and the effects of political events on the stock market: The Belgian case. Economics and Politics, 15(1), 85–102. doi: 10.1111/1468-0343.00116.