Blockchains Architecture, Design and Use Cases
Prof. Sandip Chakraborty
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 42
Blockchain in Government – IV (Hyperledger Indy)

Welcome back to the course on Blockchain. So, in the last class we have started looking into one use case of utilization blockchain for the government use cases; where we have looked into the digital identity as a use case, and we have understood the concepts of digital identity and a requirement for having digital identity for multiple agency usage; where different agencies or different domains, different service providers can utilize your identity for authenticative or to provide authorization to use certain services.

And there we have seen that to ensure digital identity, you need to ensure 2 aspect the first aspect is the saves over an identity; where as an individual you should have control and ownership on your identity data, and you need to have control over how your identity data is going to be usage used by different agencies which we call as the consent.

So, these control ownership and the consents are 2 fundamental concept behind providing self-sovereign identity. And the second requirement was the distributed trust management. So, where as an individual you have a trust relationship with the agencies, with which you are sharing your identity information with a trust that they will not reveal it to any other third party without your consent.

So, the example that I have shared with you in the last class like, it may happen that you are going for a summer trip, and during that summer trip you are check into a hotel, and you are providing your photo identity card a copy of the photo identity card to the hotel. So, the hotel management is utilizing that photo identity card to find out that you are a citizen of a particular country, and it gives your identity information to the hotel management.

But you have an inherent trust relationship with the hotel management that they are not going to utilize your identity information, or they are not going to reveal your identity information to any other third parties. So, that way you have a setup of our inherent trust

relationship with the agencies or with the companies, with the authoritative domains, with service providers with whom you are sharing your identity information.

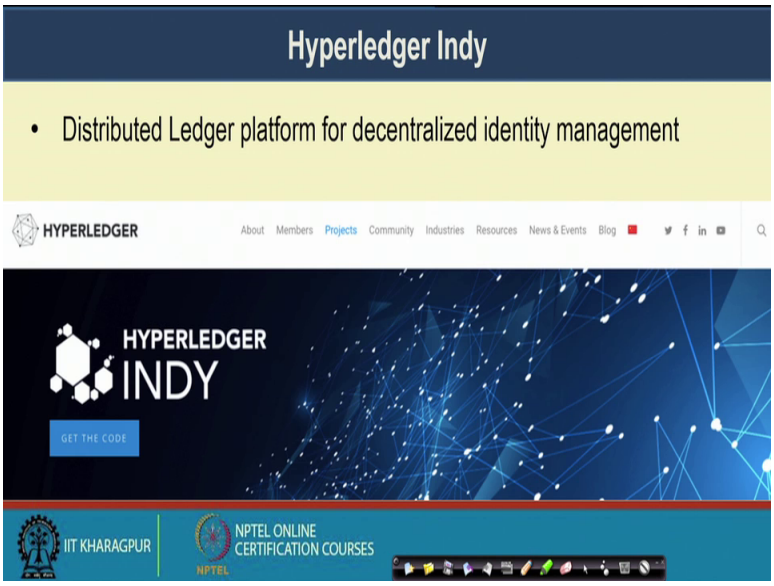
So, in a in a distributed domain or a decentralized domain where you can have multiple attributes that may cover your entire identity information and as we have mentioned that inherently this identity information is decentralized, where different agencies or different organization can use different identity attributes to verify you. So, for example, the passport is the means for verification during a say, during your foreign trip when you have to do a custom clearance; whereas, your passbook is your identity whenever you are going to bank.

So, that way multiple attributes can be utilized for identifying you at for getting multiple services. So, having this decentralized identity information among multiple agencies or multiple service providers while ensuring the self-sovereign identity and the distributed trust model that is the biggest challenge of our today's digital identity management.

And here we have seen that we can utilize the blockchain platform for digital identity management where blockchain can provide you a user centric design, the second is that blockchain can ensure that you can even verify your identity data with the help of a smart contract by not revealing the actual attribute information of your identity. And the third one is the information is tamper proof an auditable.

So, as an individual you can always check that how your data is getting shared with third parties, or whether they are getting shared with any other property without having your concept. Now in today's lecture we will look into an example of use case of a digital identity management with the help of a hyper ledger platform, which we call as the hyper ledger Indy.

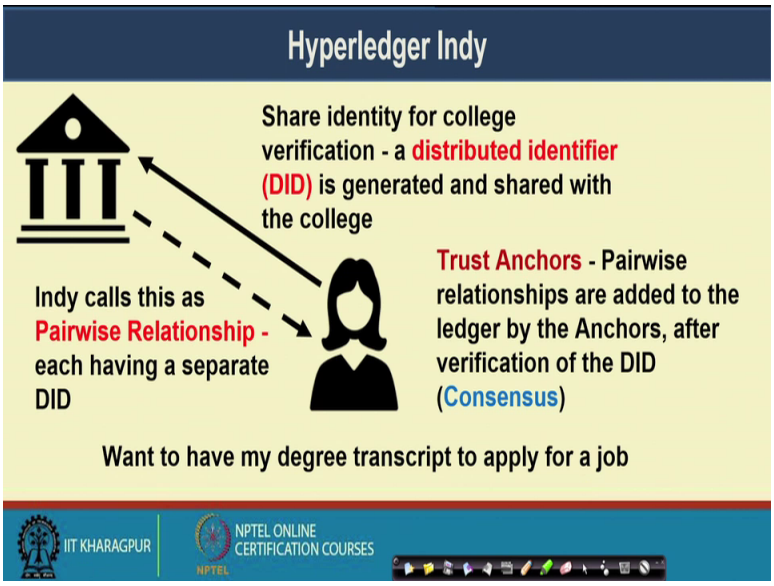
(Refer Slide Time: 04:54)



So, this hyper ledger Indy is for or a project sub project under hyper ledger project. So, in previous lectures which had been taken by Praveen you have already seen, the variant of a hyper ledger platform which is called as hyper laser fabric, and you know that hyper ledger was blockchain platform or better to say a permission blockchain platform for a smart contact platform, that was jointly developed by the Linux foundation and many other industries like IBM, who have participated in this hyper ledger fabric design.

So, under hyper ledger this hyper ledger Indy is a project or you can say it as a sub project, or sometimes we call it as a fork of this main project, which deals with or which provides a platform for digital identity management. So, we look into the details that how hyper ledger platform can provide you a ubiquitous environment to manage your digital identity for getting multiple services.

(Refer Slide Time: 06:05)



So, we will explain the entire thing using an use case, say Alice she wants to apply for a job, and when she wants to apply for a job to apply for a job she requires hard degree certificate or the degree transcripts. So, she need to apply to her college from where she has already passed out for a degree certificate or for a transcript.

So, Alice want to have her degrees transcript have to for applying a job. Then she share her identity information with the college for verification. So, the first term that Alice have to do that she has to share her identity information with the college. So, that the college can verify that this is the authoritative person named Alice to whom I can share a hard transcript so that she can utilize it for some other purposes.

So, here Alice wants to apply for a job. So, this identity information which is utilized in a hyper ledger Indy it is called a distributed identifier or a DID, DID. So, the idea is a distributed identifier so, this distributed identifier is like a public key of Alice through which the institute or the college can verify the identity of Alice.

So, this did is generated by hyper ledger Indy. So, this did it is shared with the college. Now in the next step the college it verifies the DID and establishes a connection. So, in Indy we call it a connection when the DID gets verified deed. So, in between there are multiple steps I am coming to those part that how this did is getting verified.

So, once this did is getting verified, it verifies the identity of Alice, and then the college can establish a connection with Alice, and share the transcript of Alice to her. Now this Indy this particular connection establishment with the help of did or a distributed identifier, in Indy it is called as a pair wise relationship. So, we call it as a pairwise relationship and every pairwise relationship has separate did.

So, if Alice wants to verify herself over the college, she will use one did one identity information based on the requirement she will set up one pair wise relationship with the college, and based on the DID and the DID will contain some certainty photography key information along with the identity information of Alice, maybe her name and her college identity number, identity card number certain information to which the college can verify the identity of Alice.

So, this and obviously, some form of digital signature of Alice, through which the college will be able to verify; that is the DID which set up the pair wise relationship with the college. Now if whenever Alice will set up another connection with some other organizations, say a company where she is interested to applying for a job by providing the transcript, she need to set up another pair wise relationship having a separate did.

Now, that did may contain some other information, like the transcript information that Alice has already obtained from the college. So, that way every pair wise information will have separate DID it will be managed by separate bit and every pairwise information will have one connection of and connection is a kind of secure channel through which individual can communicate with each other through a secure platform after doing the authorization or authentication for getting services.

So, here comes the point where how will you verify that the information that Alice is being provided to the college that is actually the correct information that is coming from Alice itself. So, here in Indy platform we utilize the concept of trust anchors. So, this trust anchors you can just think of as a minor to a bitcoin platform. So, the in a bitcoin platform the task of the miner was to verify ongoing transaction and then add that transaction over a block.

Now, on an Indy platform the trust anchors they verify a date, they verify a pairwise relationship, they find out that whether the information is coming from Alice or someone is trying to impersonate Alice. So, after doing the verification, this pair wise relationship

information, they are taken as a transaction in the Indy platform, and they are added to the ledger with the help of these trust anchors. So, this entire part the trust anchors they ensure the distributed consensus over this in the hyper laser platform; so, I will come to that consensus algorithm which is called as a plenum consensus.

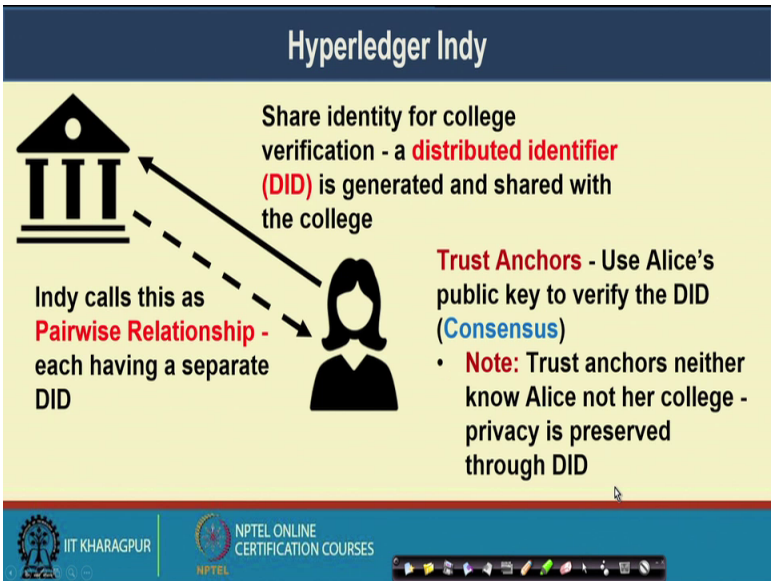
So, in that plenum consensus by utilizing that plenum consensus the trust anchors they ensure that the pair wise relationship that is going to establish between Alice and her colleagues, it is a valid relationship. So, the connection can be established and no one is trying to impersonate Alice while making this connection. So, after doing this verification by the trust anchors after reaching to the consensus; So, the consensus algorithm I will talk into details.

So, after reaching to this consensus this particular transaction information this pairwise relationship information is included in the ledger. Now that way this becomes a tamper proof record or the auditing log for pairwise relationship, where Alice can verify that which are the pairwise relationship that was established based on any of the attributes of an Alice.

So, it is it is just like that if the college is going to use say someone is there in the college who is a malicious user, and that malicious user is trying to sell or trying to reveal Alice's identity to some third party. So, that that particular malicious user need to reveal that did which Alice has shared with the with that third party with that college that they need to devil, but whenever they will try to reveal that particular date DID these trust anchors will prevent it, because the DID has already been utilized.

So, that did cannot be shared with a anyone else unless Alice is they are in the loop, unless you are getting a authorization from Alice.

(Refer Slide Time: 13:33)



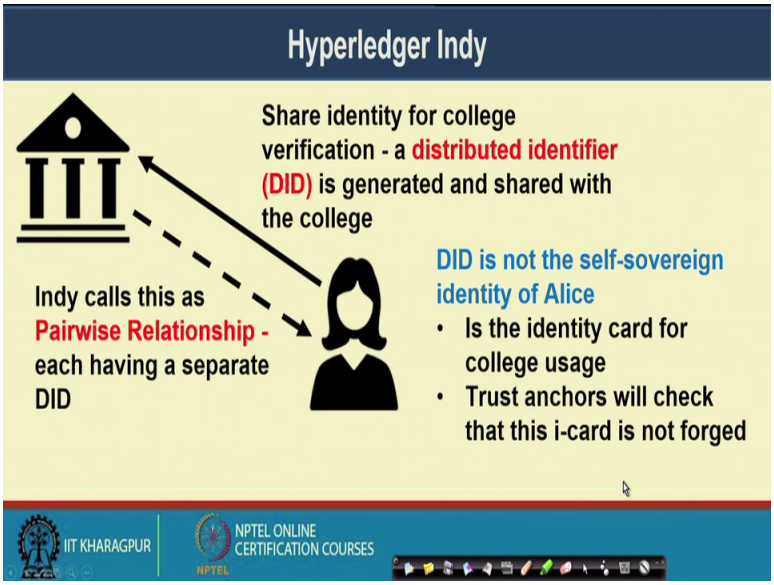
So, these trust anchors in the network, this trust anchors are neither no Alice nor her colleagues. So, that way this privacy is preserved through did, the only task of the trust anchor is to verify that the DID is indeed correct and both the persons like the Alice and her college authority they have agreed on setting up this pairwise relationship. So, that is the only task of the trust anchors, which is done with the help of some cryptographic keys.

But interestingly the way that things happens in bitcoin like you can see the transactions, but you cannot actually identify that who is the source of the transactions or which the originator of the transaction, and who is the destination of the transactions transaction because everything is put like an address. And with the help of the public key who were just verifying the address and the corresponding transaction by utilizing the bitcoin script.

But similarly here in the Indy platform you have certain identity information that is a public key information or you can think of just like a bitcoin address, a wallet address which is being utilized in bitcoin, you can think of it as an violet address through which a Alice is setting up the pair wise relationship or the connection with her colleagues. And a trust anchors whenever they are getting this information about the pairwise relationship, they can only see the address, but from that address it is theoretically impossible or better to say it is practically impossible to find out who that actual user is.

So, they are not able to identify neither Alice nor her college, but their only task would be to verify that this pairwise relationship is indeed valid; that means, they are both of the parties have agreed to share the information with each other. So, that is the only task for the trust anchors. And that way in the Indy platform you are ensuring this privacy you are ensuring that the privacy is getting preserved well.

(Refer Slide Time: 15:54)

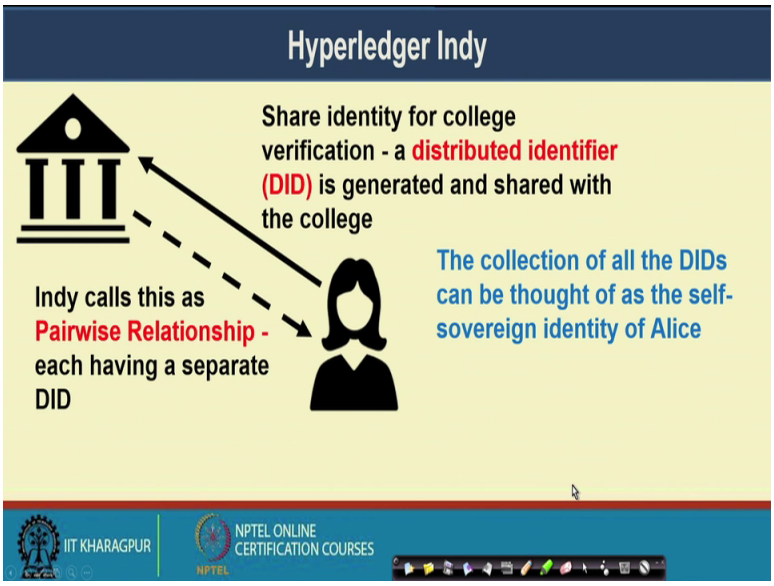


So, so, another point to note that did is not the self-sovereign an identity of values. So, did is you can just think of it like a identity attribute; which Alice shares with the her college. So, for this example it may happen that it is nothing but the identity card of Alice. Now the trust anchors that they will check this identity card which the Alice is going to share with the college that is not forced.

So, Alice is the authoritative person, and no one else is trying to impersonate Alice in this particular example. So, that would be the only task for the trust anchors, for checking that identity card is not being forced. So, this did you can just think of it as an identity attribute like the college identity card. So, the self-sovereign an identity of Alice would be all such DIDs that she is going to create and she is going to share with some other institutes or some other companies for getting different kind of services is sharing with different service providers.

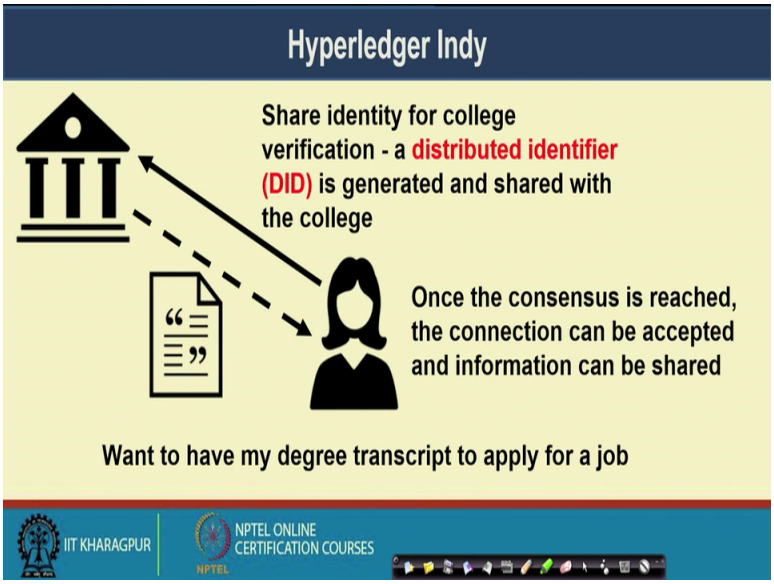
So, that set of did that will constitute the self-sovereign an identity of Alice. And every individual did you can think of as an identity attribute.

(Refer Slide Time: 17:23)



Well, the point that I have mentioned that the collection of all these DIDs can be taught as the self-sovereign identity of Alice.

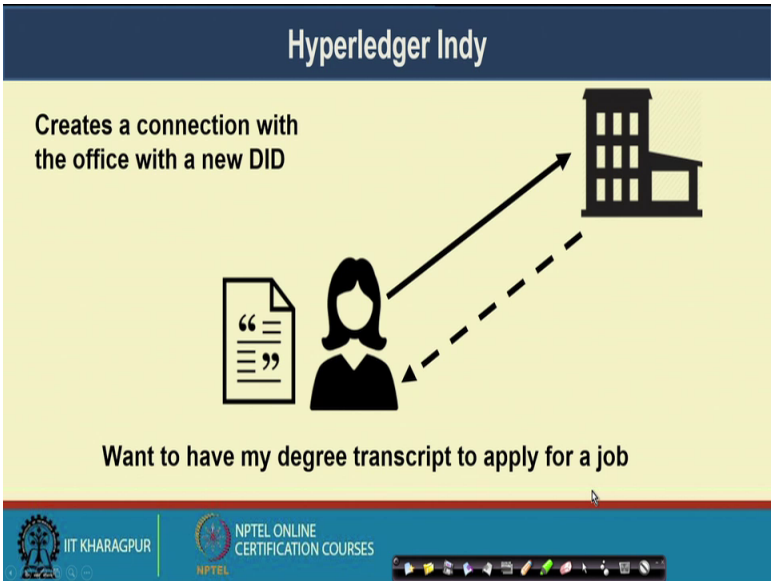
(Refer Slide Time: 17:32)



So, once this consensus is reached, the connection can be accepted by both the parties by Alice and the college, and the college can share the transcript with Alice. So, once the college get a verification from this trust anchors that the pair wise relationship that Alice is trying to set up with the college with the help of the DID it is indeed valid and the

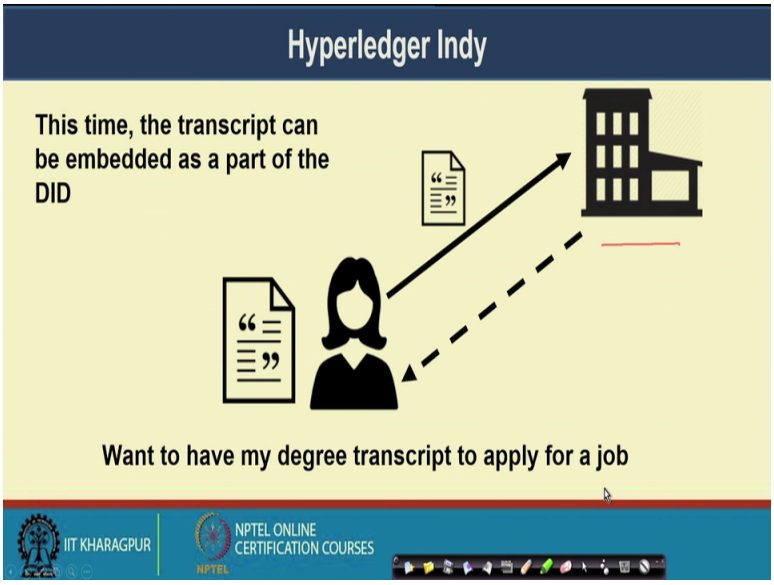
identity card that Alice has shared it is not getting impersonated in that case the college will share the transcript with Alice.

(Refer Slide Time: 18:13)



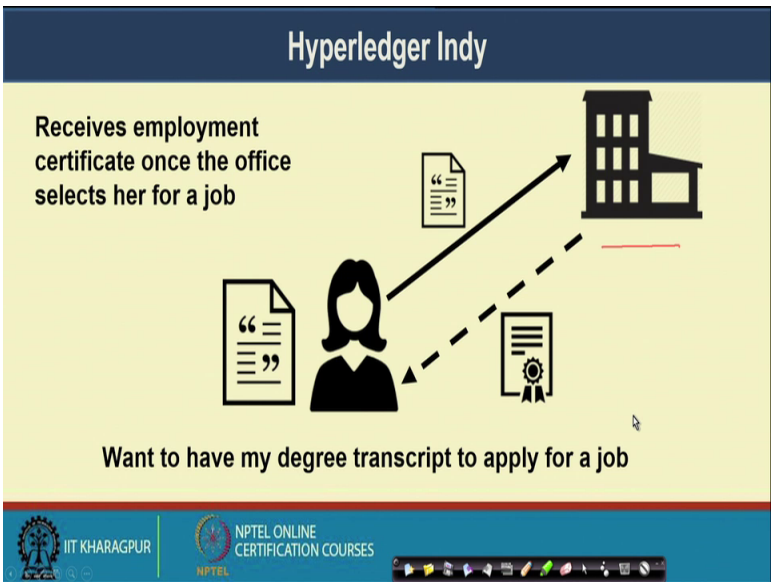
Well, once Alice get that transcript so now Alice can create another pair wise relationship with say the company where she wants to make an application. So, Alice can create another such pairwise relationship, and in this case you can use the another DID, she can set up another DID with this particular company for applying for the job, and in that case in that did setup she can provide the transcript that she already got.

(Refer Slide Time: 18:50)



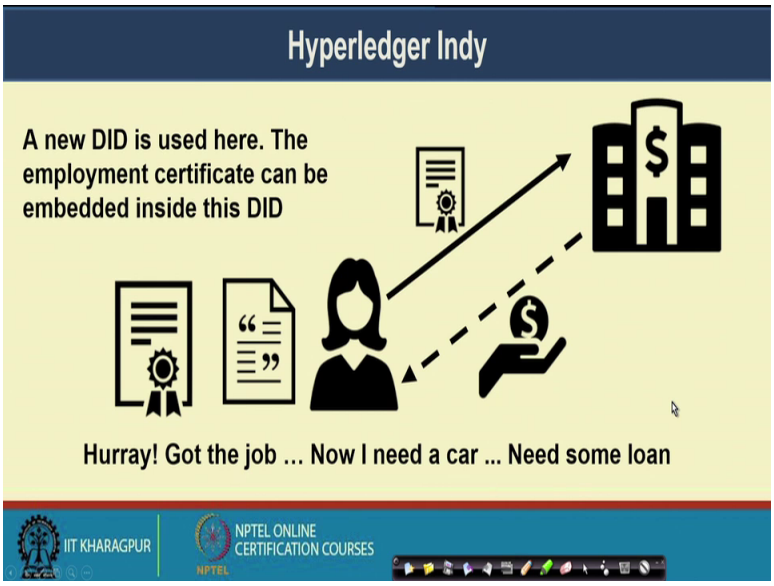
So, that transcript it is shared along with the DID it can be embedded as a part of the DID and it can be shared with the company to have a job applications. So, again that same process will get repeated up trust anchors they will verify that the DID that is being created for another pairwise relationship, between Alice and one company say the company ABC. When so, the pair wise relationship that is getting created, and in that pair wise relationship neither Alice nor that company ABC they are getting impersonated. And the both are agreeing to share the information with each other by doing this identity validation.

(Refer Slide Time: 19:33)



And then the company can offer a service, offer a job to Alice and they can provide employment certificate to Alice.

(Refer Slide Time: 19:43)



Now, Alice has this employment certificate. So, she is very happy now, she got the transcript from the college, she got the employment certificate by getting a job in a company, now she wants to buy a car. Now whenever she wants to buy a car she need to apply for certain loan in a bank. And in that case whenever she wants to apply for certain loan in a bank, she need to give a proof of her employment.

Now, the employment certificate that she got from her employer, the company ABC, that I have mentioned earlier. So, she can use that employment certificate to apply for a loan. Now Alice wants to apply for a loan, and again a new did is created a new pairwise relationship is created. And with the help of this new pair wise relationship, she gets a loan from the bank. So, that way in this particular example what we see that, with the help of this DID the distributed identifier which will be verified by the trust anchors, that whoever is creating that DID and trying to set up a pairwise relationship.

Those DIDs are indeed valid it they are not a kind of replay attack and did came from replay attack or a DID which has not been shared without the consent of any of the parties or the originator of that DID. So, here Alice is the originator for all this DID. So, and Alice has the ownership about those DIDs. So, these DIDs cannot be shared without getting permission from Alice, and without getting constant from Alice and this individual did still work like a attribute for getting authentication or attribute of the digital identity of Alice and together all these DIDs that Alice is going to create over her

lifetime. They constitutes the self-sovereign an identity of Alice. So, in that particular case the trust anchors they are verified that the DIDs are indeed originated from Alice and no one is trying to replace an older did.

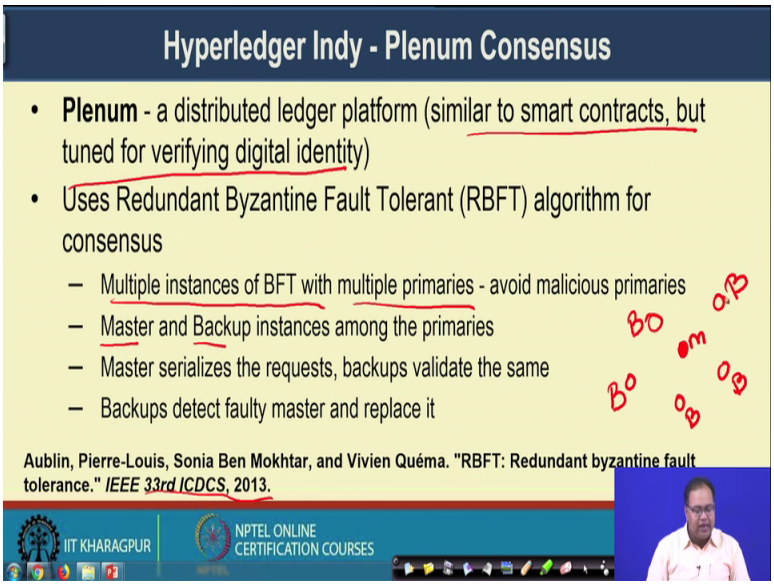
Now, say for example, whenever Alice has shared one did say with one service provider A, and that service provider A wants to says share or wants to reveal the identity of Alice for certain business purpose are doing trying to trying to sell that identity information without getting consent from Alice. Now if this kind of scenario happens in that case, that did will not get verified by the trust anchors, because the trust anchors will find out that DID that Alice has shared that contains the cryptographic key of Alice.

So, that particular did which is now coming from say another organisation or another service provider. There that did has not originated from the Alice so, they will not allow that did to get executed or to get verified. So, that did will not pass the verification check by the correct trust anchors, and it will not get updated in the Indy ledger and as a result thus pair wise relationship will not get set up. So, that way without getting the consent of Alice or without keeping Alice in the loop, none of the DID can be shared with any of the third parties.

So, this basically ensures the initial objectives that I had mentioned. The first objective was protecting the self-sovereign identities, here Alice has the complete control on the DID that she is going to create. So, only Alice can create did no one can impersonate Alice for creating the DID. The second to us the distributed trust relationship. In the distributed translation ship whenever Alice is sharing the DID with the college or with the company or with the bank.

So, neither college nor compact company not the bank they will be able to reveal that did any further. So, without getting it verified from Alice without getting the consent from Alice. So, that that way this particular in the platform which is utilizing the concept of digital ledger at the backbone, it is ensuring the requirements which we are there for managing the digital identities.

(Refer Slide Time: 24:23)



So, in hyper ledger Indy as I was mentioning in between that uses a consensus algorithm called a plenum consensus. So, this plenum it is a distributed ledger platform which is similar to smart contracts, but it is tuned for verifying digital identity. So, the smart contract which is getting implemented there with the help of the hyper ledger platform, and they are mostly suitable for verifying the digital identity. So, that distributed ledger platform which is utilized in Indy we call it as a plenum platform.

So, this plenum it uses a bagging time fault tolerant algorithm as a backbone. So, you can understand that this entire digital identity verification platform it is a use case of permission blockchain environment, because the environment is not an open environment that at the environment is closed. So, Alice the bank, the institute, the companies where Alice is applying for a job all of them have certain identity in the form of a cryptographic key. And they can register to some organization to some PKI infrastructure to get that keys or to obtain that keys; so, you have this concept of public key and a private key here.

So, you can utilize the PKI infrastructure to get the keys by authorized authorizing themselves with that PKI infrastructure. So, this is a kind of a closed environment or a permission blockchain environment. And here it is a nice example where we utilize a variant of a bagging time fault tolerant algorithm, which is called a redundant bagging time fault tolerant or a RBFT algorithm.

So, this algorithm was proposed in 2013 in a top distributed system conference called ICDCS. So, in 33rd ICDCS in 2013 it was getting proposed by Aublin Sonia Ben Mokthar and Vivien. So, this idea of this RBFT was something similar your standard BFT algorithm with the help of this primary and backups.

So, in case of your PBFT type of algorithm you had a primary and back up. So, what we have seen that if the primaries kind of faulty. So, the PBFT ensures the consensus by making a weak synchrony assumption when the PBFT is faulty. So, what when the primaries faulty. So, what PBFT does? That whenever the primary is faulty PBFT assumes that the primary will not send any further message. So, the backups they wait for the messages from the primary if they are not getting the messages from the primary, then they wait for a timeout period and then they initiate a few change procedure.

But what if the primary rather than being a faulty node it is it is just behaving maliciously; that means, it is sending the messages to the secondary, but say for a valid message it is marking the valid message as an invalid message. Now if it happens then a malicious primary they can get control over the entire consensus protocol. So, this RBFT algorithm the redundant bagging time fault ER and algorithm, they basically try to prevent the malicious primaries in case of BFT based consensus algorithm.

So, here the basic idea is that you run multiple instances of the BFT with multiple primaries to avoid the malicious primary. So, you have now rather than having a single primary in the system and multiple backups in the system, you have multiple primaries and multiple backups. So, different nodes can work like a primal. Now among these primary nodes which are running there, these primary routes you are also classifying into them 2 groups, one is called is master another set the backup.

So, here you have multiple primary nodes, and every primary have their own backup similar to the PBFT algorithm. But among these primaries we have one node which is termed as the master. So, say this M is the master and all others are the backups. Now in this case, everyone independently runs the PBFT algorithm; thus, the serialization of the transactions, and whenever they are doing the serialization of the transaction, the master is the only one who is proposing the final transaction.

So, the master is proposing the final transaction. But what these backups are doing? Because these backups they are parallely running the PBFT algorithm, and parallely they

are trying to serializing the set of transactions. So, what the transaction that the master is proposing? The backups observes whether the proposed transactions by the masters, they are same as the transactions which are generated by the backups whether the serialization orders are same or not.

If the serialization orders are same; that means, the master is behaving correctly, if the serialization order is different; that means, the primary which is working like a master that is possibly behaving like a malicious node. And in that case this primary backups they try to remove that master nodes, detect these faulty master nodes and replace this faulty master nodes with another primary node.

So, that is the broad idea of this RBFT algorithm; where you run multiple instance instances of the BFT algorithms in parallel, and every primary generates one serealized order of the transactions. And among this serealized orders the serealized orders which master is going to generate that particular set of transaction that serialized order of transaction; that is taken out as an output from that consensus algorithms.

But at the same time, all the backups validate that whether the serialization order that master is proposing whether that is matching with their own serialization order. If they are matching then that means, the primer is working correctly, if it is not matching then; that means, the master it is behaving maliciously. So, they replace the backups they collectively elects another master, and that master will take control over the entire system.

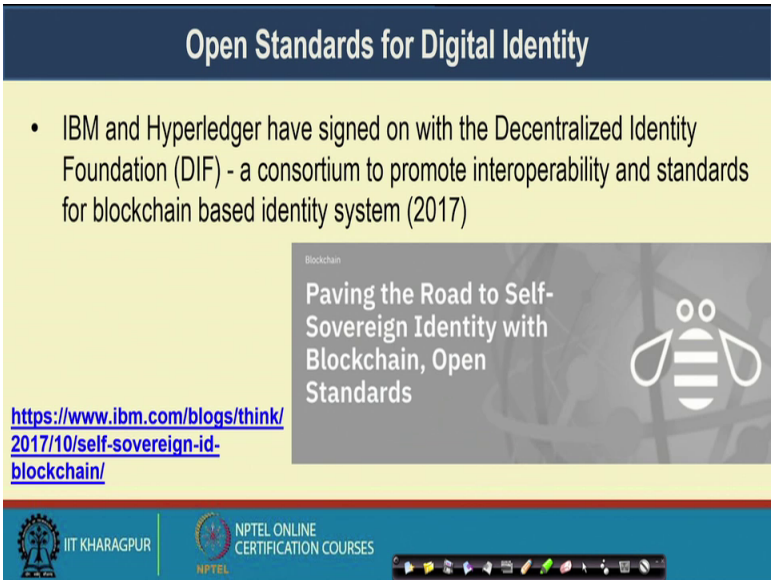
So, this is the RBFT algorithm, I have just given you a brief introduction about the RBFT algorithm. So, you should read this paper in a publicity in ICBS to get the details about this RBFT algorithm and the theoretical proof and complexities of that particular algorithm. Well, that is about this hyper ledger in the platform; which utilizes this concept of plenum as a digital laser and RBFT consensus algorithm to ensure preventing identity fraud, and ensures this self-sovereign identity and preserving self-sovereign identity and preserving distributed trust during the identity sharing and identity verification.

(Refer Slide Time: 31:55)

So, there are multiple startups which are nowadays working on this digital identity platform. So, this secure key they are developing one digital identity platform based on blockchain environment or the smart contact environment. So, many another well-known or popular startup which is working on this space of a digital identity management. So, sovereign also have their own white paper.

So, I encourage all of you to read that white paper to understand how this sovereign platform works.

(Refer Slide Time: 32:27)



And there are certain open standards for digital identity which is coming up. So, this IBM and hyper ledger, they have signed an relationship with the decentralized identity foundation; which is a consortium to promote interoperability and standards for blockchain based identity system.

So, it was very new it came in 2017; where they are collaborating with each other to have the standards for digital identity management.

(Refer Slide Time: 32:58)

So, that is one interesting read that I encourage all of you to go through. It is the sovereign protocol which utilizes blockchain platform for identity and decentralized trust management. So, you can get the nice concept about how blockchain can be utilized for digital identity management.

So, that is all about the digital identity management over the blockchain platform. Hope you have enjoyed this lecture with a practical example and platform for practical use case from which the government can get a lot of benefit. So, broadly we can or we should be able to tackle the problems which are there with the other database the debate; which are going on top of this other database. So, with this I would like to conclude this lecture. See you again in the next class.

Thank you.