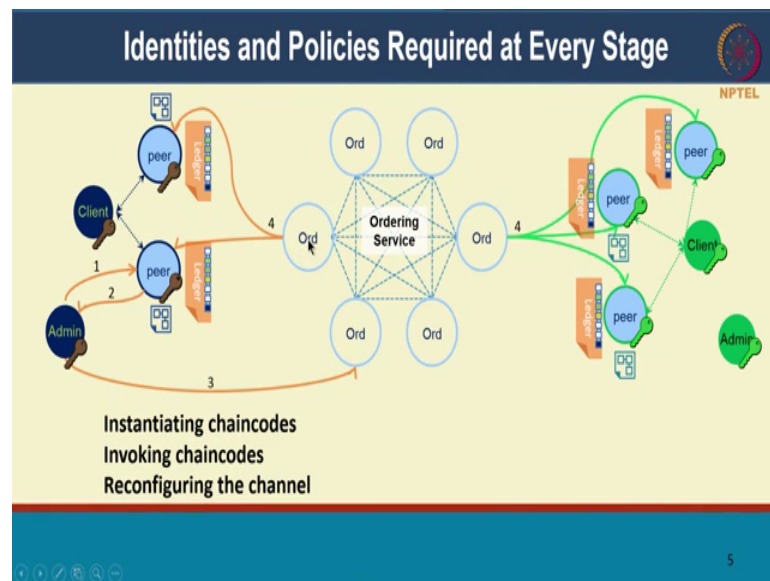


**Blockchains Architecture, Design and Use Cases**  
**Prof. Sandip Chakraborty**  
**Prof. Praveen Jayachandran**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture – 44**  
**Blockchain Security – II (Membership and Access control in Fabric)**

Hello everyone, and welcome to the Block Chains course. Then looking at some of the security aspects, and we look at a deeper dive into membership and access control in hyper ledger fabric, and how hyper ledger fabric provides this. We looked at this very briefly in one of the earlier sections just so that you understand what the membership notions are. But we will look at it as a full lecture now.

(Refer Slide Time: 00:43)



Hyper ledger fabric requires identities and policies at every stage of processing. So, there is every within an organization you can have admins, and members and they have identities, organizations can run peers, and the peers have identities. The ordering service also has identities, and organizations can participate and run order notes. And this could be just showing multiple organizations.

So, there are policies involved when you are installing chain codes and the identities are verified. So, only an admin can actually deploy or install a chain code on a particular pair. So, the peers admin has to be installing chain code. If anyone else tries to install a

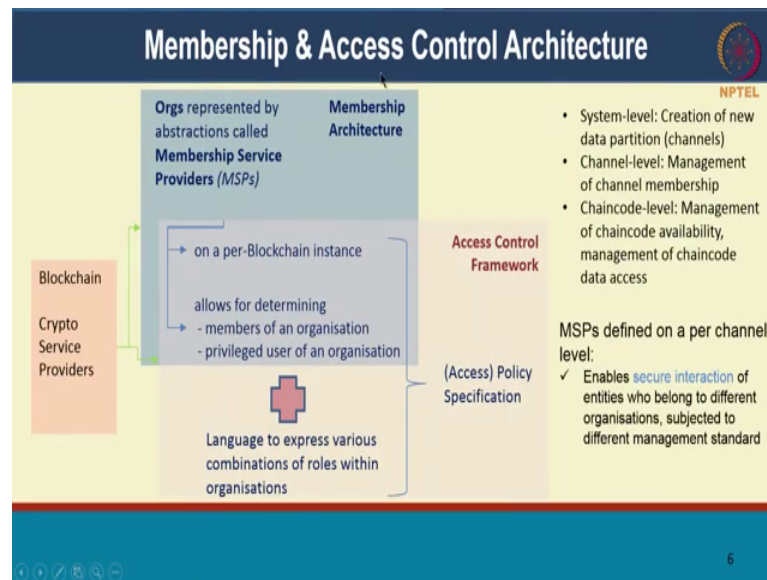
particular chain code on a peer, it will do it, right. So, that is at the step of installing chain codes.

Then there is instantiating chain code. So, this is done at the channel level. So, once you have the process chain code process itself running, you will have to tell which all channels on which it needs to execute. So, then the admin is going to call invoke this instantiate chain code call. So, it is going to call in turn instantiate the chain code for a particular channel. The channel already has information about who are all the peers who have joined this channel.

So, it will go and communicate to all the peers who are part of the channels so, in this case it has these 5 arrows. It communicates to all the peers saying, hey, for this new for this channel there is this new chain code which we all have to instantiate. So, all the peers will go ahead and instantiate that and the channel is now reconfigured to include that chain code also. So, that chain code also part of the channel; so, all of these also involved.

So, every step of this there is identity verification. So, only a peer only an admin can instantiate a chain code on a particular channel. And to do that that admin has to be part of an organization that is part of the channel. And the order will be performing this check to make sure this who is invoking this is actually part of the channel. If you not part of the channel you should not be able to instantiate chain code in the channel. And again when the order communicates this message back to all the peers, the peers validate whether this message actually indeed came from the order. So, that identity check is done also.

(Refer Slide Time: 03:05)



So, this is overall the membership and access control architecture. It is built as again a pluggable framework. You could have different crypto service providers. So, we will talk about towards the end of this lecture about the crypto service providers themselves.

So, these are ones that are implementing different crypto algorithms. The membership architecture itself; so, the membership service provider is really an abstraction of membership or identity management on blockchain. So, regardless of what kind of membership you have you can bring your own certificate authority you can use the fabric certificate authority that fabric shifts with.

So, with regardless of what implementation you are using for your certificates. The MSP provides the uniform abstraction from separating the implementation of the identity management of how that is run from the block chain platform itself. So, MSP provides that unifying interface. So, that allows you to plug in any kind of identity management structure any kind of certificate authority into hyper ledger fabric.

So, this membership architecture is there for per block chain instance; which means that every pair every order, they all have an MSP within them. And this MSP will be provide the unification across the different kinds of certificate authorities different instances might use. So, even within peers themselves, organization a's pair can have a different certificate authority from organization b s pair, and MSP provides that unifying interface.

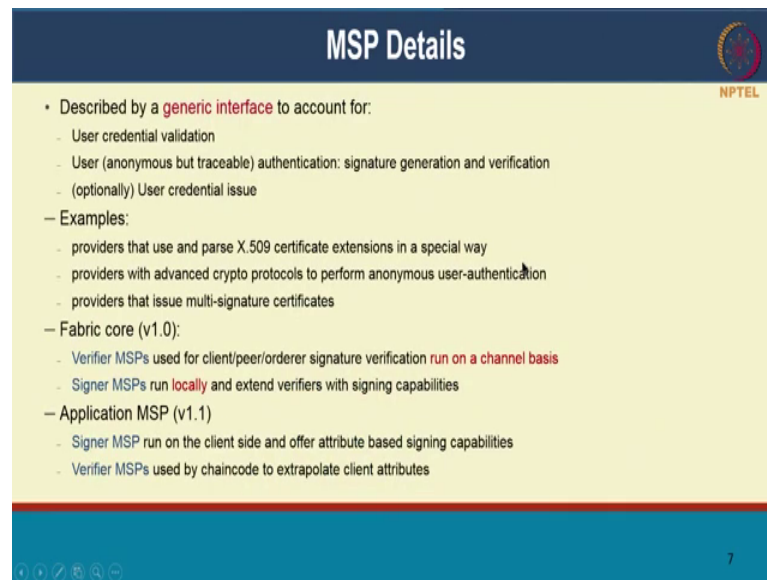
So, this allows for determining members of an organization, who are the privileged users of an organization. So, remember the only the privileged users can perform certain things like installing chain code, and configuring reconfiguring a channel then so on, right. And then there is an access control framework based on the identities that are provided; like, for instance this is the what I just mentioned about privileged users for being able to perform certain functions, it is actually a form of access control.

And we populated fabric also gives you a language to express various combinations of roles and organizations. So, it comes by default the default fabrics a comes with 2 rows that gives you admin and a and a member role. But you can also have sub organizations like what we call as organizational unit. So, you can also have that defined, and you can provide access control mechanisms on some of those roles and organization unit. In together all of this allows you to give the access control policy.

So, all of this membership and access control is present at the system level which is used for creating new data partitions or channels. It is that at a channel level where the management of channel membership is managed based on this MSP and the access control. And it is also there at a chain code level based on like managing chaincode availability whose running these chain codes managing the access to the data that the chain codes provide. So, all of this at each have level there is membership and access control.

And MSPs are defined on a per channel level. So, it enables secure intake interaction of entities so, belong to different organizations. And each organization could have their own identity management standard right. So, they can all use their own identity management standard, but MSP provides the unifying way of how each of them can talk to each other; so, going to some details about MSP itself.

(Refer Slide Time: 06:53)



The slide is titled "MSP Details" and features the NPTEL logo in the top right corner. The content is organized into a bulleted list:

- Described by a **generic interface** to account for:
  - User credential validation
  - User (anonymous but traceable) authentication: signature generation and verification
  - (optionally) User credential issue
- Examples:
  - providers that use and parse X.509 certificate extensions in a special way
  - providers with advanced crypto protocols to perform anonymous user-authentication
  - providers that issue multi-signature certificates
- Fabric core (v1.0):
  - Verifier MSPs used for client/peer/orderer signature verification **run on a channel basis**
  - Signer MSPs run **locally** and extend verifiers with signing capabilities
- Application MSP (v1.1)
  - Signer MSP run on the client side and offer attribute based signing capabilities
  - Verifier MSPs used by chaincode to extrapolate client attributes

At the bottom of the slide, there are navigation icons and the number 7.

So, it is defined as a generic interface, anyone can implement that interface. And it provides an interface for credential validation to check whether somebody has so, the person whose who is performing this transaction is actually a valid user. It provides a anonymous, but traceable authentication. And it also does the signature very signature generation and verification.

In addition, it can also do credential issue by that is optional. Like, you can have the, you can bring in your own CA which is already issued credential and you can plug that in to an MSP.

Some examples of how one might implement an MSP is if you could have one certificate authority, that has a special kind of AX509 certificate and maybe attach some extensions to it and has and modifies that in certain ways. Or it could have it could be because you are you are applying certain advanced cryptographic protocols and you might perform anonymous user authentication.

So, you will actually talk about an example of this called identity mixer. I think I have briefly mentioned it in the previous lecture. We will talk about identity mixer in more detail in the next lecture; where it provides anonymous anonymity unlinkability, it is actually a special kind of a certificate authority that fabric is providing.

And it could it can also have issuance of it party signature certificate. So, all of these can be provided as pluggable components using a common MSP interface. And fabric core itself implements 2 things it implements verifier a MSP; which this is used by each client pure or order to verify the signatures of messages coming from other participants.

And this is this is unique for each channel. And there is signer MSP that is run locally at each component. And each component will use that to store and sign based on their private credential. So, only the signer MSP will hold the private keys. The verifier MSP only going to have the public keys used to verify whether signatures are authentic.

Now, the application MSP is going to again implement a signer MSP and verifier MSP. So, the signer MSP is going to be the client side or the user certificate. There sorry, users private key right, and it might have special attributes part of that that it uses those attributes to sign, who when it sign. And there are also verifier MSPs that are used by the chain codes to figure out whether the attributes provided by the user are actually authentic. So, these are the different places where MSPs are used.

(Refer Slide Time: 09:55)

### A Standard PKI based MSP for Fabric

**VerifierMSP (on a per channel basis):**

- Identity = **standard** X.509 certificate
- Governed by standard PKI hierarchies (root/intermediate CAs, CRLs)
  - Setup = (list of root CAs, revocation list, admin certificate)
- Support for ECDSA keys/limited support for RSA keys
- Identity Validity Conditions = signed by a root CA
- Offers **no anonymity** support
- Signature generation/verification = **standard** public key crypto operation
- For certificate issuing, can leverage commercial CA (off-band), or our custom fabric-CA (online)
- Used by clients, peers, orderers for client/peer/orderer **signature verification**

**SignerMSP (only on local basis):**

- Verification aspects same as VerifierMSP
- Includes SigningIdentity = (std X.509 certificate with public key **PK**, private key **SK** for **PK**)
- Used by clients, peers, orderers to **sign messages & authenticate off-chain messages**

The diagram illustrates the architecture of a Standard PKI based MSP for Fabric. At the top, an 'MSP API' box contains two components: 'Fabric-CA' and 'External CA'. The 'Fabric-CA' is connected to 'Fabric-CA Certificate Authority (clients)'. The 'External CA' is connected to 'External Certificate Authority (clients, peers, orderers)'. A 'Peer / Client / Orderer' is shown interacting with the 'MSP API'.

NPTEL

8

So, fabric implements a standard PKI based MSP this is this is implemented by default by fabric. So, you can have other implementations as well. It uses a standard X509 certificate as a notion of identity. And this is governed by standard PKI high hierarchy.

So, this kind of certificates are used routinely in internet applications. And what they are governed by is the fact there is a root CA there is who is trusted. And that CA will sign and say this certificate actually belongs to Praveen. And if the CA could have multiple intermediate CA s just for hierarchical purposes and it could also maintain a certificate revocation less.

So, periodically these certificates might refreshed and new certificates might be issued, and the CA will maintain which certificates has been revoked in the past. And what fabric supports is ECDSA which is elective elliptic curve based digital signature authentication. And it also provide some limited supports for RSA based keys. But the predominantly is uses an elliptic curve based digital signatures. And the validity is based on whether the root CA has actually signed it or whether it is verifiable right.

So, you need to have the public key of the root CA to make sure that this it is actually signed by the true CA. This standard implementation of MSP provides no anonymity. So, if you look at the certificate you will know exactly who perform the transaction. So, it does not give you any notion of anonymity at all.

The identity mixture which is the advanced notion and it is a it is a fabric CA implementation it is a. That provides anonymity and unlink ability. But the standard MSP implementation does not give you a anonymity for your transactions.

Ah So, signature generation verification is standard public key crypto operation. So, you are going to use your private key to sign, and you will and people who have your public key can verify through is. In fact, who provided that digital signature or perform that transaction. So, like I mention you can have a you can use your own you can use a commercial CA of band. Or you can use fabric CA that comes shift along with fabric, and you can use that fabrics CA in your application. And IBM block chain platform right now does not allow you to plug in your own CA, but you will have to use yours use the fabric CA implementation. But I guess at some point in the future they will allow that capability as well.

That is on the verifier MSP or if you look at this part, this does not have any private information at all. It only has public information required to verify signatures that others have signed. The signer MSP is where the private key is held, and it includes the apart from the verifier MSP portions apart from the X509 certificate with public key. It also

has the private key that you will use to sign. So, that is the, what the function of the signer MSPs. And this signer MSP is a side as part of the clients orders peers every component in the network, they will all use this to sign their transactions or sign their messages.

(Refer Slide Time: 13:23)

**MSPs: Building Blocks for Access Policies**

Each MSP allows for the definition of three type of principals:

- Role-based: **member** of an MSP, **admin** of an MSP
- Identity-based: a specific **identity**
- Organizational-unit-based: a member of an MSP that belongs to a specific OU

A policy can be defined as a combination **N** MSPPrincipals and may **be satisfied** when identities and signatures corresponding to **t** of them are present

**Examples:** Assume the existence of three MSPs, **AliceCo**, **BobCo**, and **CharlieCo**, a policy can have the form

"AliceCo.admin **AND** CharlieCo.member"

"AliceCo.admin **OR** CharlieCo.admin **OR** BobCo.admin"

"CharlieCo.OU.FinanceDivision"

NPTEL

9

And MSPs provide the core building blocks for you to then get to access policies. And I like I mention there are access policies at every stage of fabric weather it is channel creation, channel membership management, chain code management, chain code data access everything is controlled can be controlled through access policies.

MSP it allows you to do role provides role based. So, you can right now it implements 2 rules member and admin. You can do identity based on specific identity, or you have the notion of an organizational unit; which you can think of it as a department within a large end organization, or a sub organization with a large end organization.

So, what do some examples look like? For instance, AliceCo dot admin and CharlieCo dot admin means that, both of these have to be have to have a signed for you to accept this. These AliceCo, BobCo and CharlieCo are existing MSPs existing organizations on the network.

And this rule says that both Alice and Charlie have to have Alices admin from Alice and a member from Charlie should have sign this transaction the second one is AliceCo dot



admin or CharlieCo dot admin or BobCo. So, any one of these if they assign any one of these admin sign that it is.

The last one shows the example of an organizational units. So, financed division of Charlie company if someone from their signs this transaction then it is acceptable or same transactions, but it is for any kind of message exchange and. So, on so that is the kind of access control language that fabric provides you can use any kind of these and or rules.

It also provides the n odd of k rules, but I have not talked about it or shown examples of that, ok.

(Refer Slide Time: 15:17)

The slide is titled "Blockchain Crypto Service Providers (BCCSP)" and features the NPTEL logo in the top right corner. It lists four key features of BCCSP, each with an icon and a brief description:

- Abstraction** (Icon: Padlock): An abstraction of cryptographic operations used in Hyperledger Fabric.
- Pluggability** (Icon: Interlocking puzzle pieces): Alternate implementations of crypto interface can be used within the HPL/fabric code, without modifying the core.
- Multiple BCCSP** (Icon: Multiple server racks): Easy addition of more types of CSPs, e.g., of different HSM types.
- International Standard Support** (Icon: World map): Pluggable crypto service provider. Potential to support more fine-grained confidentiality features.

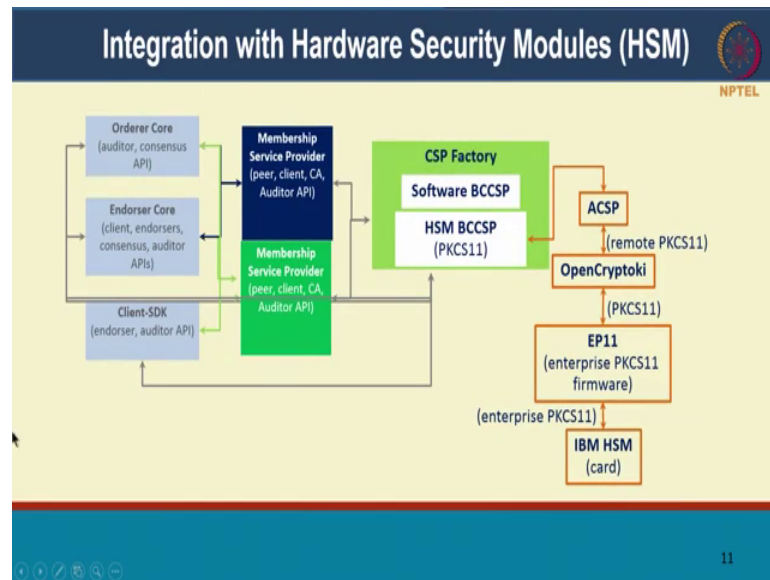
At the bottom of the slide, there is a navigation bar with icons for back, forward, search, and other controls, and the number "10" in the bottom right corner.

So, that was all about identities and access control. The next aspect of it is the crypto service provider. So, these are the cryptographic algorithms or the operations that you would use based on the identities that you have, right. So, one of it that is implemented is the elliptic curve based digital signature. So, that is one part of it. So, apart from that what even this part of it the crypto service provider can be a pluggable it is actually a pluggable components.

So, you can come with a come up with your own implementations of these crypto operations. And anything about this is you can also use this to connect to hardware security modules. We will talk about that in the next slide, where some of the signing is

happening in hardware. So, that key is not even exposing in software. So, no one can hack and steal your key. And it also provides international standard support. And it is it is it gives you the ability to provide even more fine grained confidentiality features be on what is what is implemented today in fabric.

(Refer Slide Time: 16:19)



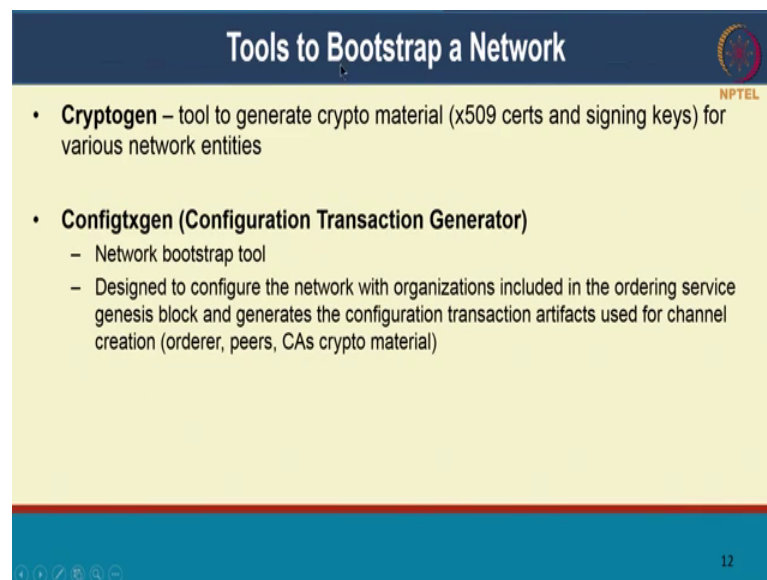
So, this is an example implementation of integration of hardware security modules with hyperledger fabric. So, this is the usual fabric components order endorser client and so on. So, as I mention there is a membership service provider who along with this. And that membership service provider interfaces with a crypto service crypto service provider CSP. So, there could be a CSP in software or it could be a HSM based CSP right.

So now HSM is a hardware security module, which securely stores your secret information. And that secret information will not leave the hardware at all. So, the, this signing would actually be performed in hardware. The secret never leaves that. So, which means that even someone hacks into your software they will not be able to find what your private key.

And there are open standards for this something called PKCS Public Key Crypto System. And it defines it is basically interface and a standard for interfacing with crypto graphic tokens, such as HSM hardware security module or a smart card.

So, there are standards for this and the standard implements interface called crypto key crypto token interface. So, there are open interfaces and standards for how you would interface with HSM. You can actually use those to connect a membership service provider and a certificate sorry a crypto service provider to HSM module.

(Refer Slide Time: 17:55)



The slide is titled "Tools to Bootstrap a Network" and features the NPTEL logo in the top right corner. It contains a bulleted list of tools:

- **Cryptogen** – tool to generate crypto material (x509 certs and signing keys) for various network entities
- **Configtxgen (Configuration Transaction Generator)**
  - Network bootstrap tool
  - Designed to configure the network with organizations included in the ordering service genesis block and generates the configuration transaction artifacts used for channel creation (orderer, peers, CAs crypto material)

At the bottom of the slide, there are navigation icons and the number 12.

Fabric also provides certain tools to as a as a convenience feature for you to help you bootstrap a network.

So, one of the tools is Cryptogen it is a tool to generate crypto material; for your for your users in your organization, for your peers your, you need to be able to generate certificates. So, all that can be generated by crypto gen. So, you can just tell crypto gen I have 4 organizations I have 3 peers, generate material for it. So, it will go and generate the crypto material needed for those organizations. Another tool is cryptotxgen.

So, this configuration sorry, configtxgen which is configuration transaction generator, it is a network bootstrap tool which is designed. So, that organizations can be included as part of the ordering service genesis block.

So, you it is basically to tell the ordering service these are the organization server be part of the network. And it generates the configuration transaction, and the genesis block to instant to create a channel it. So, it has all the crypto material for orders peers certificate authorities that are needed.

You then goes and creates the genesis block and also have a configuration transaction, that tells the order and the channel that these are the organizations that are participating these are the peers who are part of the network. And in future that configuration transaction you can you can create other instances, you can modify that configuration through channel configuration from (Refer Time: 19:29).

So, that is also possible, as long as you are an admin of channel. So, that is an important access control that you will have to over you have to satisfy. So, those are some of the features in hyper ledger fabric with respect to membership and access control.

(Refer Slide Time: 19:45)



**Fun Reading**

- Membership Service Provider, Hyperledger Fabric docs: <http://hyperledger-fabric.readthedocs.io/en/release-1.1/msp.html>
- Hardware Security Module, Wikipedia: [https://en.wikipedia.org/wiki/Hardware\\_security\\_module](https://en.wikipedia.org/wiki/Hardware_security_module)
- PKCS 11, Wikipedia: [https://en.wikipedia.org/wiki/PKCS\\_11](https://en.wikipedia.org/wiki/PKCS_11)

NPTEL

13

So, some reading material on this. So, the MSP itself is it is it is a slightly sophisticated concept, I would say about how to abstract different notions of identity and membership on hyper ledger fabric. So, you can read up the doc fabric documentation on that there gives you some examples as well.

If you are not aware of what hardware security modules are, I would encourage you to look up the Wikipedia article on it gives you a good overview of that. And the PKCS standard is also good read just look that up on Wikipedia you will understand what there is a problem. That concludes this lecture. On the next lecture we will actually dig deep into privacy aspects, and how hyper ledger provides some of those features.

Thank you. See you soon.