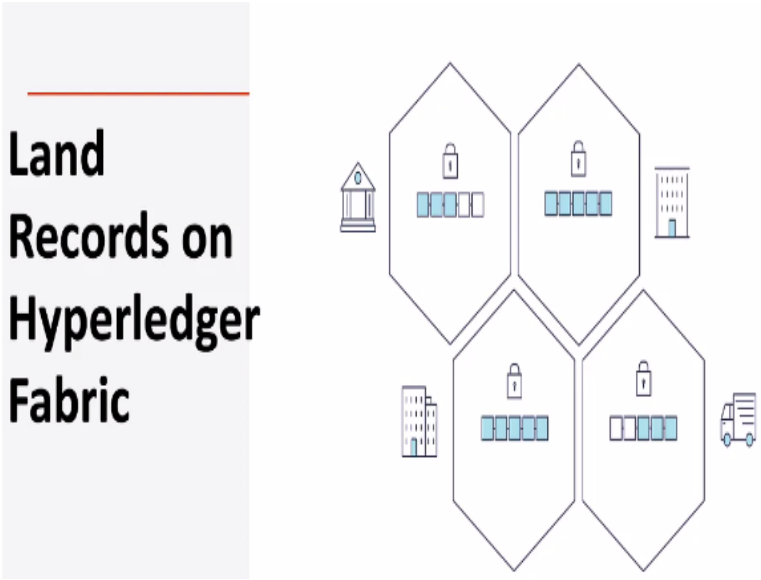
Blockchain Technology and Applications
Prof. Sandeep K. Shukla
Indian Institute of Technology – Kanpur

Lecture – 28
C3I Center

Hello and welcome to the second part of week 8 of blockchain technology and applications on NPTEL. So in this part, what we are going to do is to expose you to a couple of applications that we have been working on using blockchain and as you will see that these are all e-governance applications and so the applications that we are particularly looking at is land record registry and the other one is on the health care information.

(Refer Slide Time: 00:43)



So, land records on Hyperledger fabric is one thing that we have done and I will give you a very bird's-eye view of what this entails. So, as you know that in India the states actually maintain the land records and usually there is a department of land records or the board of land records and this is called also board of revenues in certain states and then there is a registry and stamps department which basically registers any change in the land record.

Any kind of transaction on land records like somebody selling a piece of land, somebody is buying a piece of land or transferring the ownership of a piece of land. All these things has to be registered and this registration actually requires to pay registration fee. So usually these are 2 different departments and what happens in most places today is that these are 2 different IT systems that are maintained. The board of revenues or land records department maintains a different IT system and then the registry and stamps maintain a different IT system.

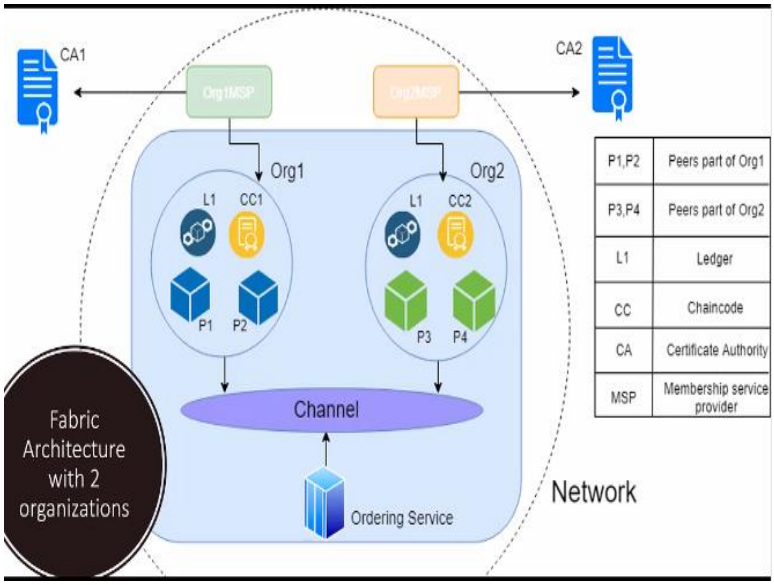
Then when people go to registry pay their fees and do the registration, then certain information gets passed along to the land records department, where the final decision about the sale or about the transfer of ownership or other kind of transactions are adjudged to be legal or not and then the land records are accordingly fixed. So now what blockchain gives us here is if we use a blockchain and put all this different organizations on that blockchain.

Then there would be a seamless IT system, would remove any kind of lost in the middle kind of situations as well as any kind of manual errors that may occur when 2 different departments used 2 different IT systems, but more than that what happens today is that as I was saying in the previous part of this lecture is that trust in the government is very important for a functioning democracy and many times, in the land records, we find that there has been some problems in the maintenance of the land record, there has been unauthorized changes in the land record.

Also the document that is kept with the government and the document copy that the customers have may not match exactly. There may be some integrity violations while it is in the custody of the IT system of the government may be because of corruption or maybe because of mistakes. These problems can be solved if we use a tamper proof system in which any kind of tampering of the records would be easily detected.

So, with these in mind, we will show you how we go about doing this land records and as you can imagine, this is a problem that requires a decentralized ledger technology rather than a blockchain technology which supports cryptocurrency kind of applications. So Hyperledger is one possible choice for this kind of decentralized ledger technology.

(Refer Slide Time: 04:49)



So this is the idea and the idea that Hyperledger is pretty suitable for this kind of application comes from the fact that Hyperledger fabric architecture is such that you can define multiple organizations and you can have channels and these channels would allow you to keep information flow confidential and on the other hand the ledger that is being maintained by multiple peers belonging to different organizations are consistent and therefore this consistency and integrity of the ledger gives you more trust on the IT system that the government maintains and the citizens can be better served this way.

So, as you can see is that there may be multiple organizations for example this could be the board of revenues or the land record department, this could be registry and stamps department and they would be the organizations that are involved in this blockchain and there could be multiple channels or single channel as we will see and there may be different peers belonging to the different organizations even the ordering service, the members of the ordering service, the nodes of the ordering service could also belong to the different organizations.

The organizations might have problem trusting each other in terms of whether they be one maintaining the records consistently or not, etc. So those kinds of mistrusts between the organizations can be handled in this kind of an architecture. So, we will do this with multiple channel architecture.

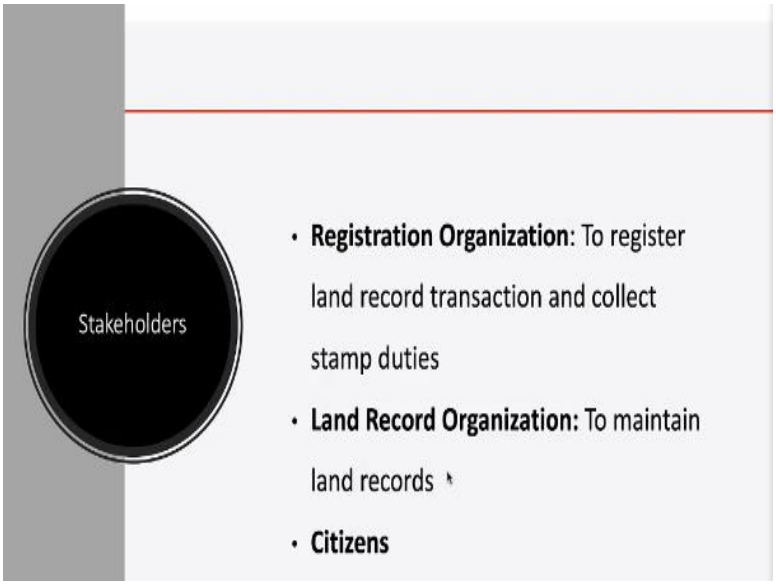
(Refer Slide Time: 06:58)

So, the problem statement here is that we want to design a decentralized system to capture land records. The integrity of the data should be maintained and privacy depends on what privacy requirements you impose. Certainly, right now the land record data is not really confidential, anybody can check for a piece of land who wants that kind of information, but the PPI or personally identifiable information, personal information should be protected. So, we should not reveal the let us say Aadhaar number or PAN card number or bank account number of the person even though that may be in the system.

The verifiability is required because owner should be able to verify his claim of ownership. So right now, it is a piece of paper or some kind of a digital document that they maintain with them and then the idea is that once he presents that to the board of revenue, board of revenue has to actually verify it to know whether this piece of paper is trustworthy or not and in many cases what may happen is that if the data has changed in the IT system of board of revenues, then the owner would be in trouble.

So, on the other hand if the owner has been forging the document in that case and has bribed somebody internally to change the record accordingly, then the rightful owner would be deprived from his rightful claim on the piece of land. So, the ecosystem here is the customers which are usually buyers or sellers and then the registration organization and the land record organization.

(Refer Slide Time: 09:11)

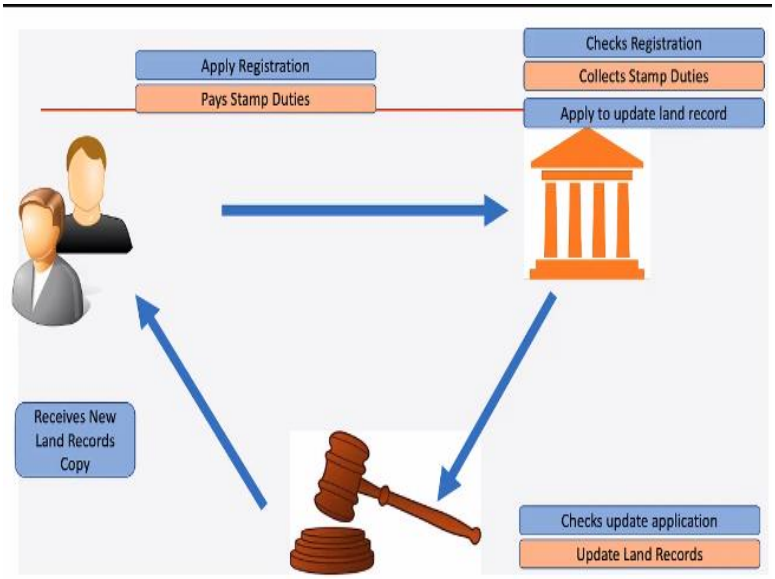


The registration organization is usually there to register land record transaction and collect the stamp duties and land record organization is there to maintain the land records. Citizens who actually own lands or want to buy lands or want to sell lands, they are usually the other parties that will be using this system. So, these are our stakeholders in the system.

(Refer Slide Time: 09:41)

So, land record management happens through this registry and update. So, registry basically is a transaction where a land transaction is registered, buying a land, selling a land, changing ownership of land, etc. etc. succession, these are some of the transactions that are registered at the registration office and then based on the registration, the land record has to be suitably updated. So normally, we know this process mostly as mutation where the data in the ledger of the government has to be updated accordingly to that transaction that has taken place.

(Refer Slide Time: 10:33)



So, the customer, in this case let us say a buyer or seller, in this case buyer obviously will apply for registration and pay for stamp duties and this is the registration department. So, the registration department will check the registration for completeness and correctness, etc., will collect the stamp duties and most probably do a KYC on the client and then apply to update the land record. So, at this point the document goes to the department of revenues or the department of land records and the land revenue court will make a decision.

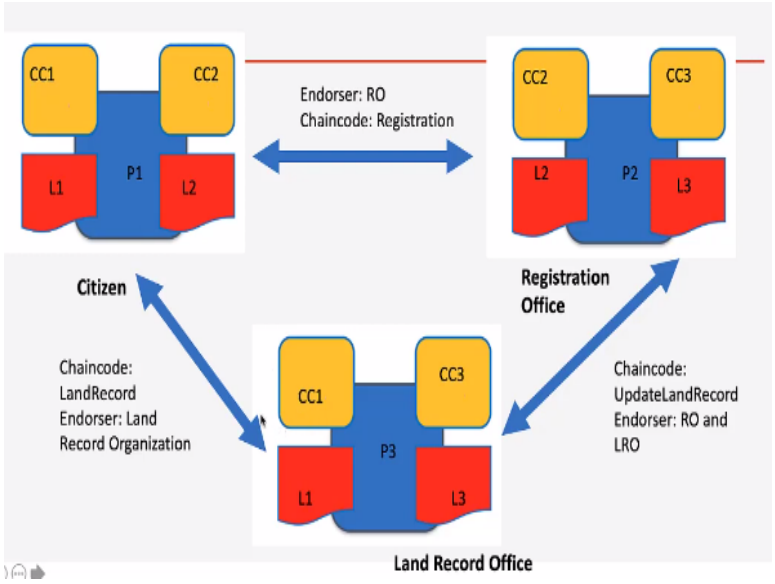
Usually it will give sort of a wide advertisement about whether anybody has any concern about this particular transaction and because sometimes the land is owned by multiple parties or when there is a succession transaction, maybe there are other successors of the original owner after his death the succession has been applied and the other should be able to get some time to know that their land is being served by somebody else, so therefore there is a time given.

After that the revenue court makes a decision as to whether to allow this land to be recorded to be in the name of the person who applied for this registration or names of the persons who applied for this registration. So, that is the checks update application process. It is a legal process and then the land record gets updated. Now this board of revenue has multiple hierarchy inside it with the local registers and so on and so forth, but they all can be part of the same organization as the board of revenue or land record department, and then the customer eventually gets a copy of the land record.

Now when we say if he gets just a printout that is not enough, right, because as I said that you know the printout can be forged and other things. So therefore, it has to be something more secure and probably digital and probably has enough information like as we said in case of the KSI blockchain in Estonia that there has to be a token which has some complimentary hash values embedded in it, so therefore it can be checked, etc. So, some such token has to be given.

Now, remember that this is not tokenization of the land, this is just a token to verify that the land record the copy that this person has is indeed the copy that was registered and updated, so that is the process.

(Refer Slide Time: 13:57)

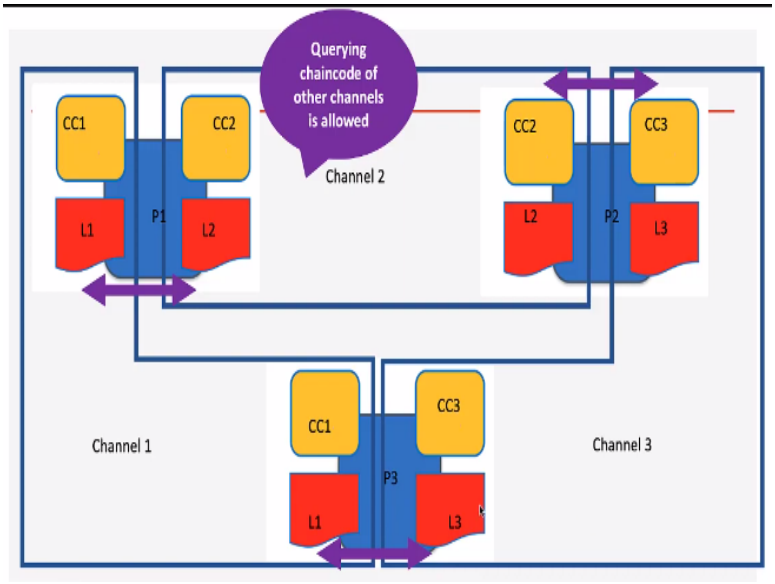


So, here what we can do is we can have all these different peers and they are working in different channels. So, the citizen will be basically a client which connects to peer to actually apply for registration and his registration has to be endorsed by the registration office and then when the registration office has endorsed, then it will update, it will send it to the land record office which will be making the necessary checks and make the decisions about the land records, you know validity of the registration and then accordingly it will do that and then send suitable digital proof to the citizen that the report has been properly updated.

Now, here you see that there are 3 chaincode involved. So, one chaincode is for the citizen to actually do its function, second is the land record that basically is part of another channel and a different ledger and this ledger is between the citizen and the registration department and then there is another chaincode that is between the registration department and the land

record office, and then finally there is the L1 which is the ledger between the citizen and the land records office. So, this is the multi-channel solution.

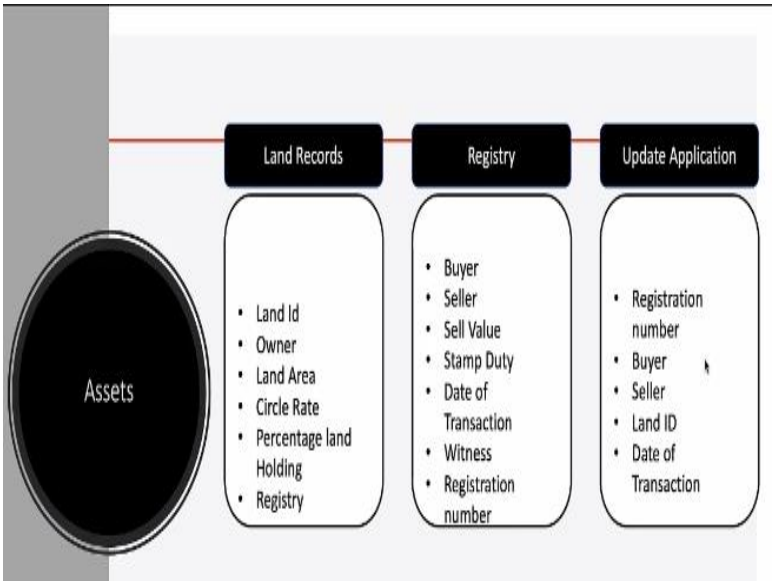
(Refer Slide Time: 16:02)



So here, you see that there are 3 different channels, one channel on which a ledger L2 that is between the citizen and it is basically keeping track of all the transactions and payments that the citizen does to the registry department, all the necessary documents, etc. Then the registry department when it endorses this transaction and then it basically uses the information that it glints from this transaction which is necessary for the land record department to do the investigation and accordingly update the record that is being shared through ledger L3.

Then the citizen and the land record department they actually are on this shared ledger L1, through which the citizen can always check the integrity and validity of the records that he actually owns. So, querying the chaincode of other channels is allowed, that is why this kind of situation is possible. So here, there is a query between this channel and that channel, so the channel 1 and channel 2, similarly between channel 2 and channel 3 also there is a similar querying possible, and then between channel 1 and channel 3, so this is something that it is.

(Refer Slide Time: 17:43)



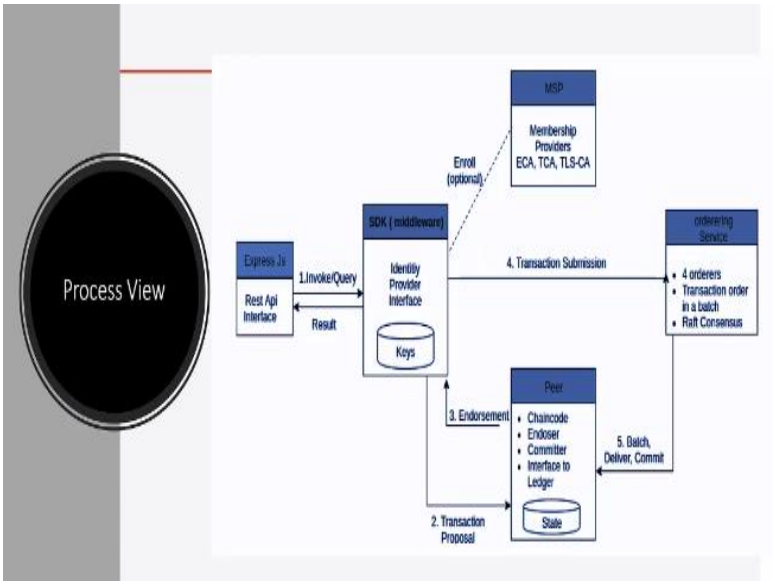
So, there are multiple assets and assets is a concept of the land records of the Hyperledger. So we have identified three asset types. So, one is land record, another is the registry, and another is the update application. So land records obviously we will have a land ID, normally like in Uttar Pradesh, the land ID is actually Aadhaar like unique number for every piece of land owner. That would mean that owner's name, there may be Aadhaar number, maybe PAN number, etc. etc.

The information about the land, the land area circle rate, percentage land holding, this kind of stuff and registry related information. Registry will on the other hand will have the buyer, seller, sell value, stamp duty, date of transaction, witness, registration number and all that information. Then update application will have the registration number which is basically being sent from the department of registry to the update land record department. So this kind of information will be there.

(Refer Slide Time: 19:06)

So, the transactions that are designed for this are applyRegistration, validateRegistration, applyUpdate, approveChanges, updateLandRecord, fetchLandRecords, getLandOwner, etc. So, applyRegistration is done by the citizen. ValidateRegistration is done by a transaction initiated by the registry office or RO. ApplyUpdate is also done by RO and then approveChanges and updateLandRecord these are done by the department of revenue or land records department and then fetchLandRecords and getLandOwner, this kind of transactions are done by citizens.

(Refer Slide Time: 19:58)

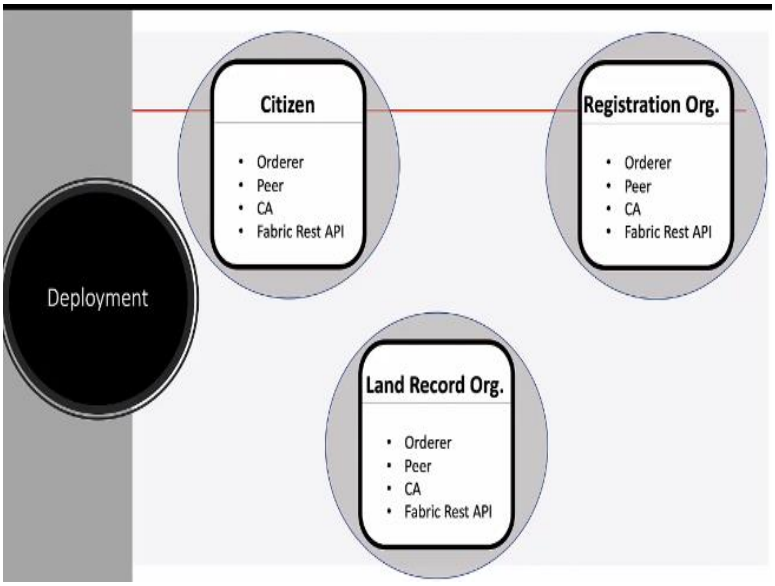


This is the process view. So, process view basically says you know how this software actually was implemented. So this shows the peer and the SDK, this is the identity service provider which basically maintains the digital certificates or keys for all the individuals and all the servers that are you know participating in making transactions because each

transaction has to be signed and then each endorsement has to be signed and that digital signature requires the digital identity.

Then this basically may depend on some membership provider like it may depend on a certification authority or some other kind of authority to derive the identity and then there will have ordering service which orders you know Byzantine fault tolerant way or crash tolerant way for making sure that there is an agreement on the order of transactions snd this is the basic structure of this implementation.

(Refer Slide Time: 21:24)



So deployment will be that registration organization will have one peer, which is an orderer, there will be a peer in the registration organization. There would be a CA for the registration organization and then it should have the rest API for interacting or invoking transactions. Same thing with the land record organization and the citizen. So that was sort of like a very high-level view of what a land record or land record registration and transaction system implemented in Hyperledger.

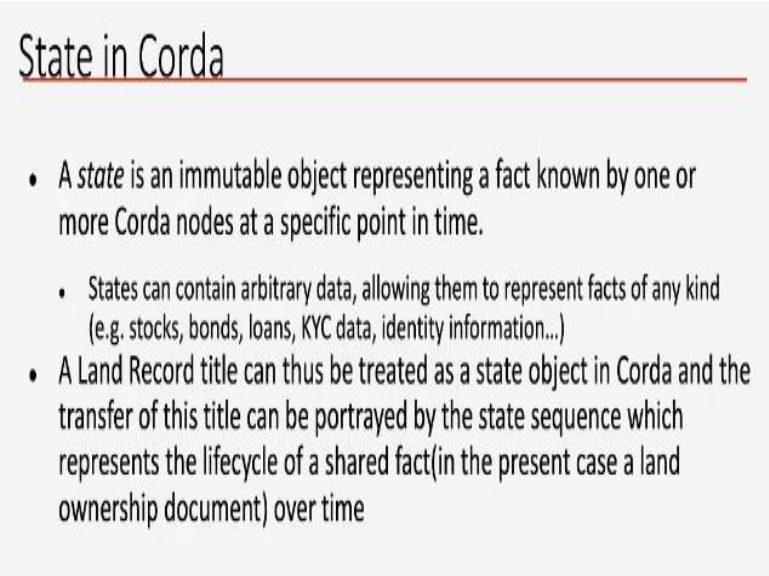
Now I want to show you another alternative way of doing this is to Corda because we said that Corda is another example of a decentralized ledger. So it is possible that one would prefer Corda over Hyperledger, but one has to actually first look at not only the convenience of designing the system because Corda as I discussed before is more geared towards the contractual obligations and contract obligated transactions, so that the contract requirements are fulfilled in the financial domain, but that does not mean that the Corda cannot be used in another domain.

Same thing as said before is that Hyperledger can be properly designed to actually do what Corda does, whereas Corda is more focused towards a particular sector, Hyperledger is more generic, but the choice has to be made based on the convenience of the architecting the system given the primitive conceptualizations of various functionalities that are available in that particular platform. Also once you implement it, you have to do proper performance testing, and to do performance testing you have to basically create a lot of transactions.

Random database transactions which are similar to the real world usage of this similar system and also the transaction rates and everything has to be tuned so that it kind of mimics the transaction rates and the inter-transaction arrival time, etc., for the real-world application in which this is targeted and then you have to see what the transaction throughputs are, what is the latencies are, then also you have to look at considerations such as security, ease of use, etc..

Then you have to make a choice between whether you want to implement this on Hyperledger or in Corda or in some other like maybe a private Ethereum. So these kinds of decisions you have to make or quorum which is a private Ethereum variant you have to do that based on those things.

(Refer Slide Time: 25:16)



So here is our attempt at using Corda to do this same thing as we said is the land record transfer using Corda. Now in Corda as we said is that one of the primitive concepts is the state. State is an immutable object which represents a fact known by one or more Corda

nodes at a specific point in time. Remember the transactions in Corda take states as inputs and produces states as outputs. So, states can contain arbitrary data, so that they can represent facts of any type like stocks, bonds, loans, their ownerships, KYC data, identity information and all that stuff.

So land record title can then be treated as a state object in Corda and the transfer of this title can be portrayed by a state sequence in which before a transaction the state who recorded who owns the land at this point in time, what are the other owners, and all that stuff and then as the transfer takes place if it succeeds, then the next state will be recording the name of the new owner and other relevant information.

So in that way, any land will be associated with a state so at any point in time at present or in the future would be actually a state information about the land, right. So, the virtual representation of a piece of land is actually a state object in Corda.

(Refer Slide Time: 27:10)

So, the participant, Corda nodes in this case would be buyer, seller and the government and government can create a new land record document with specified owner. Buyer can request to buy a specific land record basically in the requested state change associated with the virtual representation of the land. Seller can sell or transfer ownership of the land record and thereby basically changing the state that is representing the land.

(Refer Slide Time: 27:50)

So land record would have as I said, even in the last case also we saw that land record would be having land ID, the owner name ID and depending on whether a piece of land can be owned by multiple people, it could be actually a list of owners, previous owners which should be also a list of previous owners and there may be other relevant information about the date of transfer in the past from one owner to the next owner and so on so forth.

Buying requests would be that is the list of potential buyers. So whenever somebody makes a buying request transaction, it may go to an intermediate state where a list of potential buyers could be recorded.

Refer Slide Time: 28:42)

So in Corda, we also talked about the notion of a flow, right. So flow is basically automates the process of agreeing ledger updates, right. So flow is basically orchestrated the different

activities at different places in the Corda system which would have to take place in order for the update of the states and the ledger. So, there would be 3 flows here that is create land record flow, request ownership transfer flow, and then transfer ownership flow. This is kind of you know similar to what we were doing in Hyperledger where we had 3 channels, so here we have 3 flows.

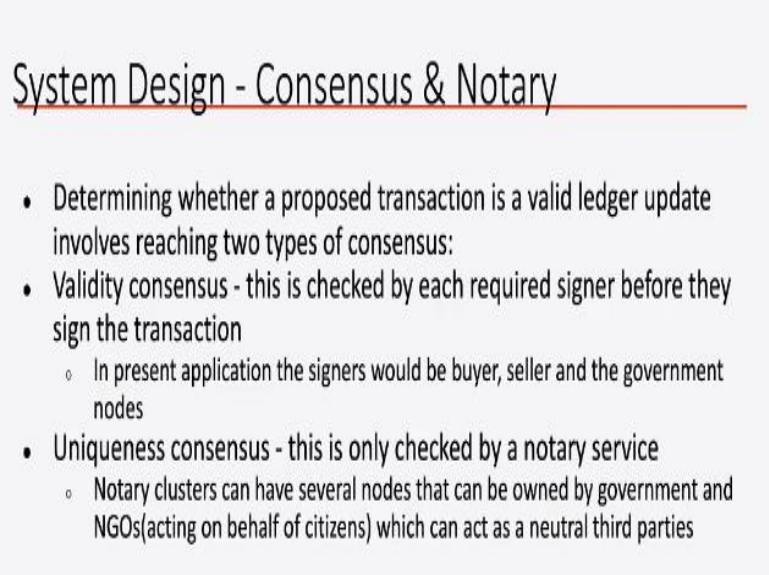
So, create land record flow would be a buyer and then government would be participants and it can be initiated by government only. Request ownership transfer flow can be initiated by buyer and the participants are seller, buyer and the government. Transfer ownership flow would be also seller, buyer, government and can be initiated by seller only. So, as you can see that this is when the government actually starts a new land record, this is when a registration request is made and this is when the actual mutation happens, that is when the actual land record is updated.

(Refer Slide Time: 30: 20)

So, to contract in this case is basically will take transaction as input and states where the transaction is considered valid based on the contract's rules. In our application, the contract should check the following. The create land record contract will check that there is zero input state and one output state. It should be only be enabled to be initiated by the government and there should be 2 signers, namely buyer and government. Request ownership transfer contract would be one input state, one output state, initiated by any buyers and 3 signers namely buyer, seller and the government.

Transfer ownership contract would be one input and one output state, initiated by the current owner and there should be 3 signers namely buyer, seller and the government. Now this is highly simplified, I mean we are not exactly going by a specific state government's exact procedure because we are not fully privy to the entire process that the government follows, but it can be customized according to the government rules and processes as in case we actually do it for a specific government. Here, we are just giving you some basic idea about what it might be like.

(Refer Slide Time: 32:00)



So, whether a proposed transaction is a valid ledger update would require two types of consensus, right, one is of validity consensus and other is the uniqueness consensus. So, validity consensus is checked by each signer because before they sign if the buyer says that you know, I am getting this piece of land from this seller, then buyer should first verify that this is a valid transaction, government should verify this is a valid transaction and buyer must verify that he or she actually owns the land and then actually sign this validity consensus.

Here, the idea is that hopefully the government actually will be the most important one here because you know if the buyer is not telling the truth, the government has had to actually first do the proper investigation. So this cannot happen like overnight, the government has to put out an ad about this update and then anybody who has any kind of you know reservation against this update should be able to go to court of revenues and put objection and so on. So once all that is over, only then the government will sign.

The uniqueness consensus has to be checked by a notary service. So we talked about notary service in the context of Corda. So notary basically checks whether this particular state of land currently is unique in the sense that somebody did not use the same to make another update transaction before, right, because then the land has been already updated in somebody else's name, then this seller is not really the correct owner and this should not be allowed right.

So, this notary service could be actually several nodes that can be owned by the government and maybe NGOs acting on behalf of the citizens which can act as neutral third parties. So therefore this is what actually the court of revenue does, and then once this thing has been actually cleared only then the uniqueness consensus will be achieved. So that was a Corda version of the land record, very superficial view of how this can be done.

(Refer Slide Time: 34:49)

Then finally, I want to talk about the electronic health record application and again we are using Hyperledger fabric here.

(Refer Slide Time: 35:08)

So the problem statement here is that, so let me first explain what the problem is. So when we go to a hospital, normally what happens is that the hospital checks us in. Then it takes some vital information like blood pressure, temperature, weight, etc. It records them and then it sends us to the actual doctor. Doctor makes whatever checkups that he has to do and then he might order various tests right, pathological tests, blood tests, urine test, etc., and then it has to go to the pathological laboratory.

When the pathological laboratory collects samples, they return all the results of the tests, then you go back to the doctor and then the doctor takes a look at this information and then accordingly may prescribe. Now doctor may also prescribe before the tests are done for temporary relief, but may actually make a more specific prescription after that information comes. Anyway, so once you get a prescription, you have to first pay the hospital for having seen the doctor, etc., but then you go to the pharmacy.

The pharmacy then fills your prescription and then gives it to you, and then if you have insurance, then the insurance has to actually get involved during your payment to the hospital and maybe the payment of the medicine, may be payment at the pathological tests, and also they might have to be initially involved when you actually first make the appointment that they are agreeable to this particular location, this particular hospital or they actually make cashless payments or reimbursement based payments whatever, all this information has to be gathered in the beginning also.

So, this is a pretty involved multi-party activity that happens. Now usually what happens is that we have all these documents and either in a more advanced hospital, this documents, this information that is taken about a patient like his vitals on a particular day, his pathological test reports, his doctors collected information, the information about the prescription being made, the state of the insurance payment all that stuff, these are either maintained on a central database by the hospital system or it is maintained in bits and pieces at various systems like in the pharmacy, in the pathology lab, etc. and they are not integrated.

There in the first case, it is integrated, but it is centralized and then when you go to a new hospital system getting that information quickly from one hospital system to a different hospital system would be time consuming, either through some email or something they will send your past history to the new hospital or you have to go and collect printed copy and then bring it to the new hospital. So this is quite cumbersome.

But bigger issue is that all the patients information if it is maintained centrally in a hospital in an IT system, then anybody who has access to that IT system, be it a doctor, be it a nurse, be it a front desk person, etc., the pharmacist or pathologists they all get to see all your information, that gives a privacy concern and in the United States there is a law called HIPAA which is very strict law about patient health information privacy protection act, HIPAA, and that kind of act is needed in India as well.

So if that kind of an act comes, then hospitals cannot be just leaving this data you know unprotected, cyber unsecured on some server or on a cloud server and then be happy about it and the patient certainly would be happy about it because all that information what disease they have and what kind of you know maybe certain diseases they do not want everybody to know, their neighbors to know, etc., all that information should be protected by law. So another issue is that, one is either the data is in a centralized database in the ownership of the hospital which is not already a very good thing.

Then on top of that there is privacy concern. Now data that is regarding the patient should be data principle is the patient himself and then if hospital becomes a fiduciary which maintains that data, then hospital will have a big onus once a data privacy act comes into existence that the hospital has to be always afraid that if the data gets leaked, then the patients are going to

sue them and they might have other legal liabilities. So therefore, we need a system which is decentralized and to capture medical records.

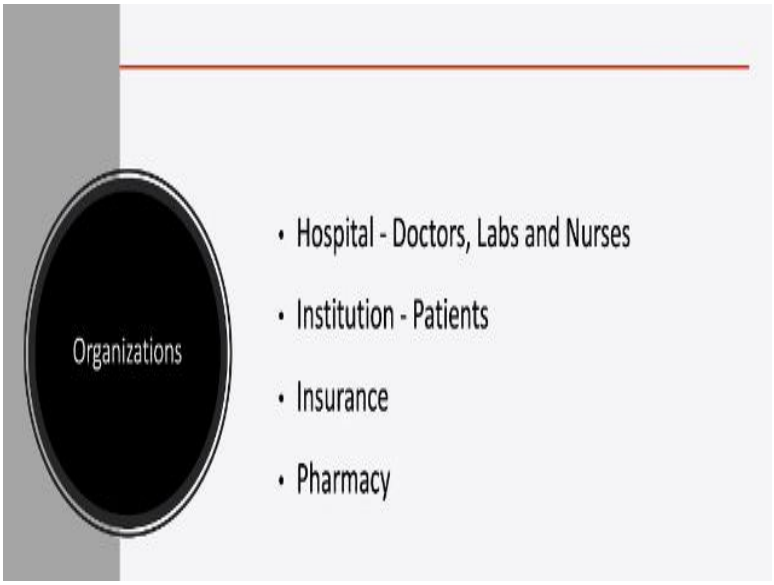
Also, the other issue that I forgot to tell is that in a centralized system like that anybody can go and change your data. So there have been movies about it that some patient did not have diabetes, but the data showed that he has diabetes. So when he was unconscious in a hospital in an emergency situation, the doctor gave insulin in his fluid that intake and then they had an insulin shock and they died. I do not know about a real case like that but this kind of scenarios also worry people that if the integrity of the information is not maintained, then anybody can manipulate that information and get you into some trouble.

So therefore, the question is that the centralized ownership is not good either for the data fiduciary that is the hospital because they have the huge liability for the patient because patients confidentiality could be breached and also if the data integrity is manipulated, then the patient may be actually in danger. So to solve all this, one has to consider using a decentralized system to capture the records, the integrity of the data should be guaranteed and then patient privacy should be preserved.

That is if the patient as the data principle allow somebody to look at the data, only then that data can be looked at, otherwise it should remain unreadable to everybody that is it should be remaining encrypted. So now there has to be a solution in which the patient can selectively show the data based on need-to-know basis. For example, a pathologist does not need to know patients vitals and other information or pharmacists does not need to know even what actually is wrong with the patient because their job is only to fill the prescription.

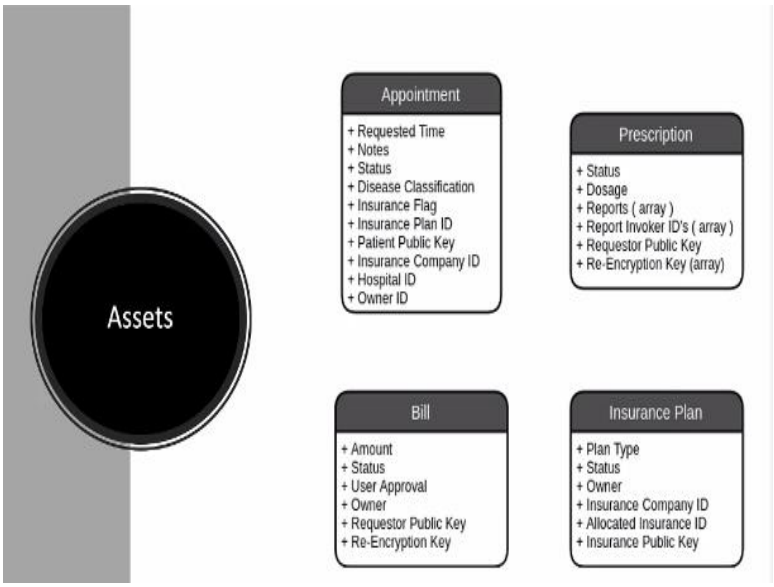
Then the integrity of the prescription is important because if they get tampered with, then the pharmacists will give the wrong prescription to the patient. So, we want a complete healthcare information system in which all these problems are addressed and then it is a decentralized system, data integrity, historical data about the patient that's integrity should be maintained, confidentiality of the patient should be maintained and patient should have the full control on who sees the data and for how long.

(Refer Slide Time: 44:18)



So, the design that we have is that we have organizations like hospital where there are multiple parties which interact with the system like doctors, the pathology labs, the x-ray labs, the nurses and the front desk people. Institution that is patient and then the institution means if it is an institutional hospital, insurance companies and the pharmacy.

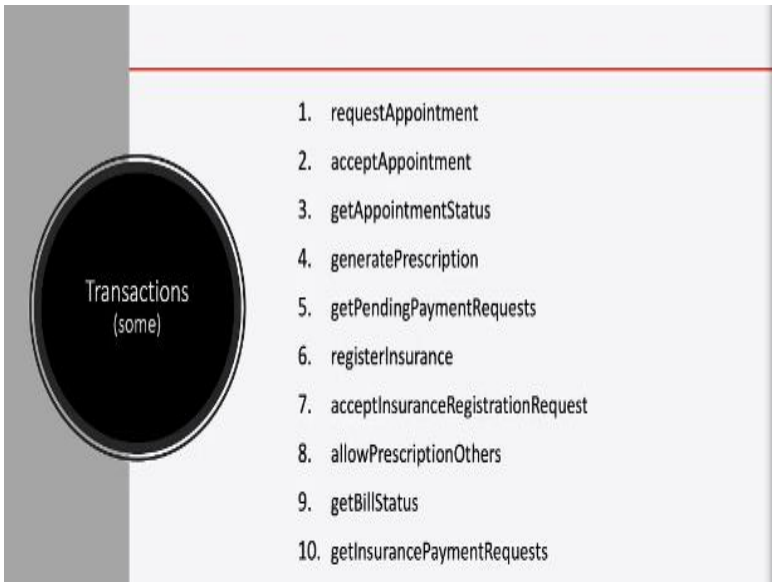
(Refer Slide Time: 44:54)



So, there are multiple different assets involved here. So appointment is one asset, prescription is another asset, bill is another asset and insurance plan is another asset and there may be some few others you know, I am showing as very simplified version. So when the patient books an appointment and instance of the appointment asset is created, which is associated with the patient and then when the patient comes and sees the doctor, then prescription is created and it is actually encrypted with the public key of the patient and therefore only the patient can see this.

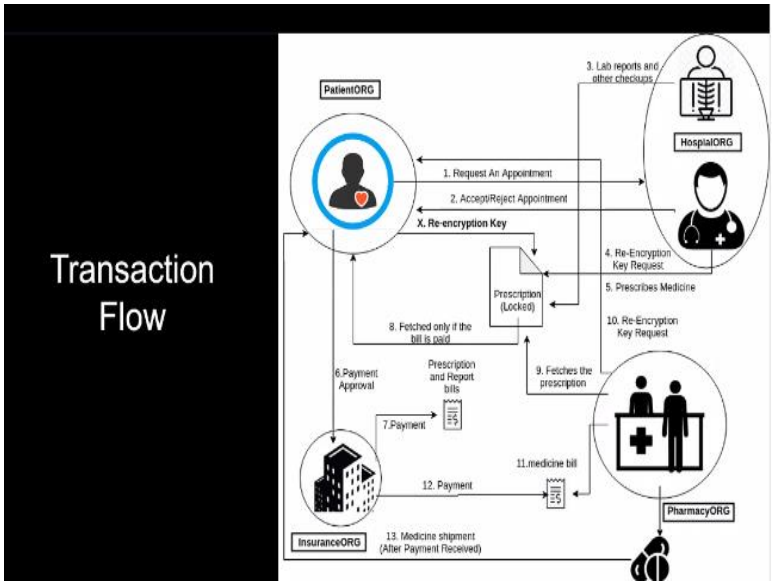
But in this case you can also use something called a proxy re-encryption by which the patient can generate a proxy key that allows say insurance or a pharmacy to actually look at its prescription information which is the one of the vital personal information that needs to be confidential.

(Refer Slide Time: 46:04)



So, different transactions like requestAppointment, acceptAppointment. RequestAppointment is done by patient, acceptAppointment by the hospital front desk, getAppointment status can be done by the patient, generatePrescription can be done by the doctor, getPendingPaymentRequests is by the patient, registerInsurance is by the patient or the insurance company, acceptInsuranceRegistrationRequest is doubly endorsed by the insurance company and probably the hospital, allowPrescriptionOthers is by the patient, getBillStatus by the patient, getInsurancePaymentRequest is by the insurance company and so on and so forth.

(Refer Slide Time: 46:53)

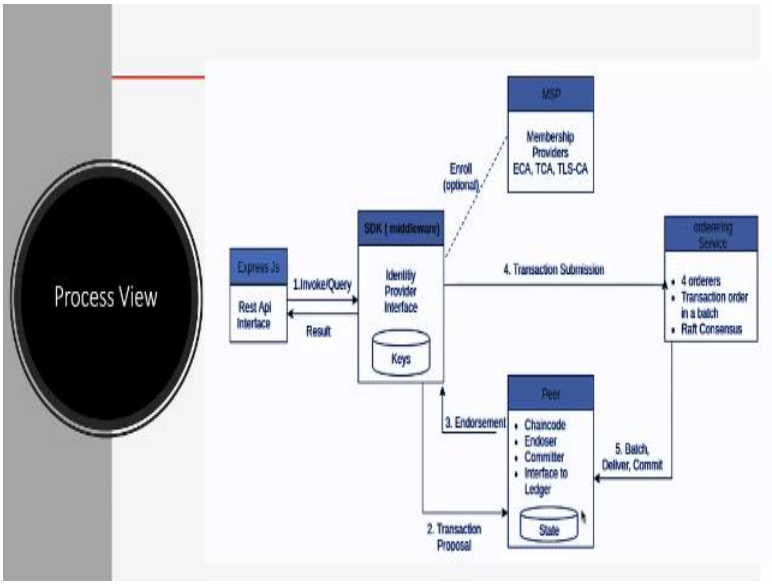


So, the transaction flow we are just showing a very simplified picture of the transaction flow. So you have the patient, you have the hospital, so there are multiple organizations right. So patient organization, this is a hospital organization, insurance organization and pharmacy organization. Then idea is that the patient makes request for an appointment and then either gets accepted or rejected the appointment. When it is accepted, the patient goes to the hospital and then his data is basically sent to the doctor.

Then the doctor sees him and then may actually request lab reports and other checkups, but once that lab reports and other checkups come in, then the prescription is created and this prescription is locked by the public key of the patient, and then when the pharmacy needs to see the prescription, the patient basically gives a re-encryption key, it sends a re-encryption key to the patient and the patient sends re-encryption key. Then this re-encryption key could be time bound so that the pharmacy can only see, only unencrypted for some time, and then it can fill the prescription.

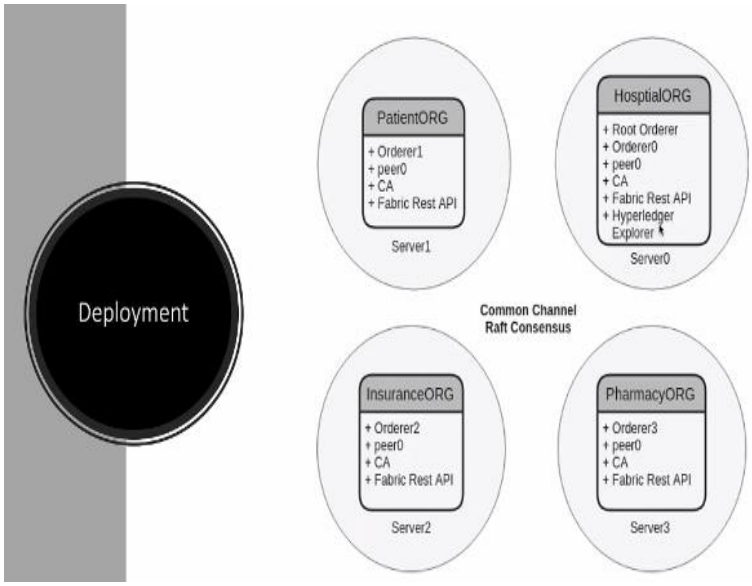
The hospital and the pharmacy, etc., will interact with the insurance as well as the patient, so this is pretty. Even in the simplified form, this is a pretty complex flow.

(Refer Slide Time: 48:35)



This is a process view which have peers, which has membership service provider, may be a some kind of a certification authority or if it is an institutional hospital like an IIT hospital, then it could be the institutional LDAP or some other way of giving membership proof or identity to the patients. Then you have ordering service which is maintained by the different organizations involved and then the peers.

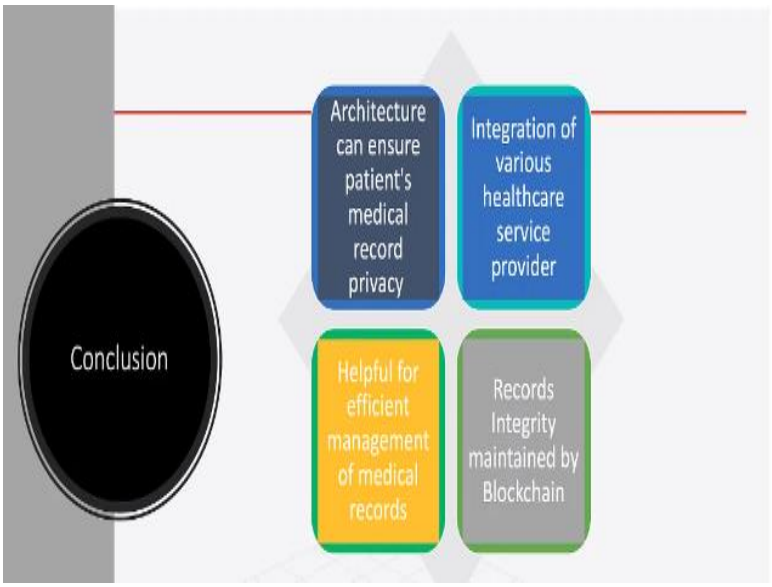
(Refer Slide Time: 49:10)



So, the deployment view is that the orderer is maintained by all, so there are 4 orderers in this simple implementation which work for a crash tolerant Byzantine fault tolerant consensus and so it requires at least 4 orderers. So here, I have orderer belonging to each organization, but the root orderer or the lead orderer is under the control of the hospital. Then you have different peers associated with in each of these and then you have the CA and then the fabric

rest API for in for transaction interaction. Then the hospital can also use a Hyperledger explorer to look into the ongoing activities through this entire system.

(Refer Slide Time: 50:14)



So, this architecture can ensure patients medical record privacy, integrate various healthcare service provider into one simple decentralized system and they can maintain a decentralized ledger. The prescription is always in encrypted form on the ledger, so only the patient has the ability to unlock it for some particular organization. So, it is helpful for efficient management of medical records and records integrity is maintained by the blockchain.

So this is a very simplified view of what one can do with this third-generation blockchain or distributed ledger technology for creating multi-organization integration of information systems with the property of information integrity and with the ability to transact across different organizations by maintaining a decentralized but shared ledger among all the different organizations. So, when we come back, we will talk about some of the final ideas and thoughts about this course as well as some misconceptions about blockchain that would actually help us conclude this 8-week long course.

Obviously as you know that after this, you have a long way to go, you have to actually learn at least one or two individual blockchain framework, how to use them, how to develop projects based on them which was not possible within the time frame we have and also the goal of my course was not to actually do this individual blockchain, you know nitty-gritty details and how to develop or program smart contracts etc.

So for that, there will be other courses, but this one is to get you a very conceptual standing or you know clarify your concept about what blockchain really is, what this technology is good for, where it is not applicable and where it is applicable, and how it is applicable. So, see you in the third part of this week and then we will discuss the final thoughts. Thank you.