

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

Lecture – 27
Summary Blockchains

Hello and welcome to everyone for the sessions in the last week of blockchain technology and applications course on NPTEL. So today we are going to have a 3-part week. So in the first part, I will talk about a little bit of summary of all the different blockchain technologies that we talked about and will also mention a few other ones that we did not have time to cover. So then in the second part, we will talk about two applications, particularly in the space of land registry and land record and also in the healthcare information on blockchain.

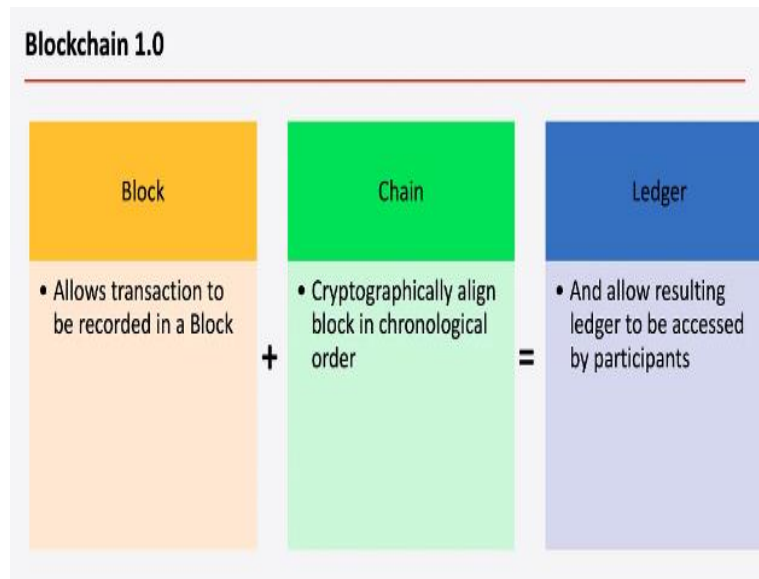
Then in the third part we will talk about some misconceptions related to blockchain technology and discuss those misconceptions.

(Refer Slide Time: 01:23)



So, I want to kind of summarize the different blockchain technologies going from blockchain 1.0 to blockchain 2.0, all the way to blockchain 3.0.

(Refer Slide Time: 01:38)



So in blockchain 1.0 which came about with the introduction of bitcoin blockchain back in 2009 and so there the idea is that you would create blocks with transactions and so these transactions are recorded into a block and anybody can create a block and the block that is selected has to solve a cryptographic puzzle as way of proving their proof-of-work, and then once the block is accepted it basically is connected to the latest block before this block by a cryptographic hash function and this is what constructs the chain, right.


So you have blocks and blocks are connected to each other by cryptographic hash and that is what is called a blockchain and this allows you to create a decentralized ledger and this ledger has all the history of all the transactions and this can be looked up by any of the participants in the blockchain you know as new transactions happen and they can be validated by looking up information in the blockchain and also the integrity of the past data is assured because of the hash that is being chained together creating a long history which requires to be completely recalculated.

If anywhere in the past you want to make a change making it virtually impossible to change any of the past blocks and that is what gives the temporal resistance property of the blockchain.

(Refer Slide Time: 03:54)


Bitcoin Blockchain

1970 Merkle tree:




1992 Proof of work

PROOF OF WORK



2008 Bitcoin



Now Bitcoin blockchain did not come you know out of the vacuum. People have been working on some of the techniques that are used in the Bitcoin blockchain such as creating Merkle tree, the idea of proof-of-work and then this of course the PKI, the public key cryptography infrastructure, use of digital signatures all these things where very cleverly put together into a single technology which is kind of revolutionary and that led to the idea of blockchain in 2008 white paper which eventually 2009 it got published and then the first blockchain implementation of Bitcoin came about in two thousand nine 2009.

(Refer Slide Time: 04:53)

Bitcoin blockchain:

- Proof of Work + Transaction Validation
- Bitcoin rewards as incentive for mining and Keeping them Honest
- Permission-less Blockchain
- Idea that Block chained by Hash could be public ledger
- **Idea that <50% attack could be sustained**

So just to summarize what is there in the Bitcoin blockchain is proof-of-work and transaction validation. This is how the blocks are created and then once the block is distributed throughout the network, everybody has to repeat the transaction validation and then they will put their copy of the block. Once they are satisfied with the transaction validation and the

validation that the puzzle was solved by the creator of that block, they actually attached this block in their copy of the blockchain.

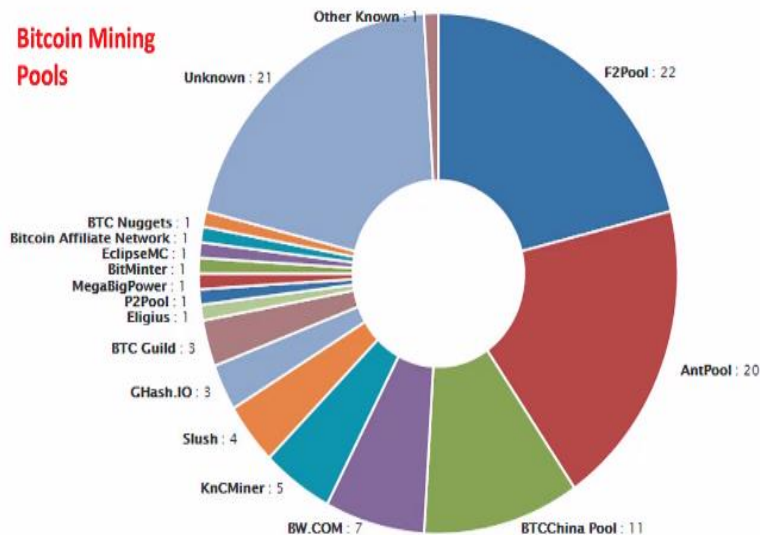
So the blockchain of the ledger is available at many different nodes and that is why it is quite difficult to actually make it tampered with, to make it unavailable and also to make transactions that are not authorized to be done by somebody. Now what is most interesting part of the bitcoin blockchain that we have seen is that it mines these blocks and along with their mining a certain reward amount of bitcoin is given as a transaction in the block to the person who actually or to the node that actually successfully created the ledger block and miners say go on doing this mining.

Not all the blocks are created by the same miner because it is very expensive and there is some amount of stochastic randomization in the process. So a lot of the miners will accumulate this bitcoin, and as they accumulate the bitcoin, their stake in the bitcoin network increases and therefore they will not mine a block that is invalid or that basically is doing a double spending and so on because they will then destroy their own bitcoins.

Because once the value of the bitcoin goes down because of this kind of activities, they are not going to be able to cash the bitcoin that they have won as block rewards and that is basically working against them, so in a game theoretic sense, these incentives keep them honest. Now bitcoin blockchain was purposely built to be permission less and the idea of blockchain by hash is a public ledger obviating the need for banks to keep track of the total amount balance and transactions and so on.

Everybody can have access to the ledger and also it was shown in the white paper that unless more than 50% of the minor of the hash power goes to a specific miner or a group of miner colluding together the tamper resistance and everything in the blockchain will be sustainable.

(Refer Slide Time: 08:51)



Now, we also discussed that the mining is very expensive, computationally very demanding and therefore not any random participant could actually win the process of mining blocks because the amount of computational hardware and power that is required is very high and that is why we have seen that miners pull together to create this blockchain pools, although we did not discuss in detail how this pooling happens.

So, there is quite a bit of technology behind this so that you know when the reward comes how the different pool participants actually distribute the reward among themselves and so on but that you can study, you know there are lot of material out on the internet, you can find that, but what we see here in this picture is that a is a few of the mining pools actually yield a very large amount of hash power. So this thing is always changing, so this is a snapshot from about 2 years ago.

But if you look at the distribution pie chart like this now, it may be slightly altered but may be some of the pools would be renamed or some pools may have come together, some pools might have split, but altogether if you see that about if 3 large pools come together, for example these 3 or I mean 3 in this case may be sufficient, these 3 together will yield a 51%. So 22, 20, 42 and 11, 53. So these 3 pools together yield 53% of hash power which is means that if they collude together, which has not happened, but if they collude together then bitcoin could actually lose its credibility.

So that is one of the big problem with the current situation with bitcoin and we also saw that many of these cryptocurrency blockchains also has this similar situation where they are

actually you know getting more and more centralized like we showed you some of our own analysis with respect to Ethereum where we saw that about 100 users out of millions of users actually yield most of the power mostly yield do most of the transactions, they own most of the currency, so this is decentralization that was thought to be not happening with this cryptocurrency seems to be not working so well.

(Refer Slide Time: 12:15)

Problem

- High transaction time- 10 min/block
- Limited size of block
- Wastage of energy
- Pseudo-anonymous

So the other problems with Bitcoin is that the each block takes about 10 minutes and if you look at the average number of transactions that are put in one block, we find that per second about 7 transactions on an average gets confirmed, but even then it is not confirmed because actually you have to build more blocks on top of the block in which this transaction was accepted, at least 6 other blocks before you can be sure that there will be another 4 that would eliminate your transaction.

So in order to be sure that your transaction is actually almost permanent, you have to wait for 10 times 6, about an hour and that makes it very slow for any kind of e-commerce activities, that is why bitcoin has become less like it from a transaction medium and more like an asset-building activity. This block is 2 megabytes and unless there is another 4 this will not change and therefore there is a problem with this by increasing number of transactions per block you cannot really increase the throughput.

But even if you go from 2 megabyte to 4 megabyte, then you will actually only increase the throughput by about you know 100% which would mean that 14 transactions per second type of throughput. Wastage of energy we already talked about that the amount of energy that is

used for running the bitcoin blockchain system is the amount of energy many countries as a whole, smaller countries as a whole, use as their energy usage.

Then the other thing that the original author of the bitcoin blockchain thought about the anonymity turns out this anonymity is sort of pseudo-anonymity which certainly you still cannot you know find the real world identity of a particular miner or a particular participant, but at least you can correlate because all the data is available. So, there are a lot of correlation based way to figure out the activities of a particular bitcoin address even if they actually rename and recreate multiple identities, they also can be correlated. So, all these problems are there in the bitcoin blockchain.

(Refer Slide Time: 15:13)



So, Algorand is another instance of blockchain 1.0 which came about 3 to 4 years ago and here the idea came from Silvio MiCali and his group in MIT and they said that well the bitcoin blockchain uses too much energy for mining and also it's a little too slow because it has to solve the cryptographic puzzle and therefore the transaction throughput is very low and therefore they came up with something based on very sound cryptographic techniques as well as sound distributed algorithms ideas especially an unimproved Byzantine fault tolerance algorithm or BFT.

We did not cover this because it requires a little more involved cryptography to actually go and cover all of this, but I just wanted to show you that something like this exists in the blockchain 1.0.

(Refer Slide Time: 16:39)

Algorand: The Solution

- New Byzantine agreement using proof of stake
- Uses verifiable random function and cryptographic sortition.
- No Forks: due to explicit consensus
- No proof of work: which means scaling easy
- Proven security bounds, about % of malicious users who may disrupt majority of network for certain amount of time.
- Very low CPU load, can potentially run on modern mobile phones

The basic idea there is that first of all it uses a new Byzantine agreement and this Byzantine agreement is done by a randomly selected set of participants, not by the entire set of participants, and this is done by some kind of a voting mechanism. So how do you select random participants from let us say millions of participants that are involved in this blockchain. They use a very clever idea of something called a verifiable random function and it is actually a combination of cryptography as well as the idea of zero knowledge proof technique.

The idea is that every node runs a function called VRF functions or verifiable random functions that depends on its private key and then based on the result of that computation of this VRF, the result they get will tell them whether at this moment they are part of the committee or not and this process of computing the function also produces a zero knowledge proof, a proof in which you do not give out your private key based on which this function was computed but you sort of provide a cryptographic proof that anybody can check that you indeed computed the random function correctly.

That the result tells that you are right now part of the committee and this committee formation happens in rounds. So every round, so there are multiple rounds are called epochs and during every round, this random function is computed by everybody and then based on the result during that round you will either be part of the committee that votes on the next block or you may be out of the committee in that round. If you have millions of them and if you use only let us say 1000 committee members, then everybody at some point may get a chance to be part of the committee.

However, in order to ensure that you know nobody creates thousands of identities so as to get a higher chance of getting into the committee all the time, so they use this like who gets to selected like this based on how much money you won at that time. So if you have already accumulated certain amount of Algorand currency, then your chance of getting selected is kind of weighted by that and this is kind of use of in an alternative way of using your stake. So if you own more currency in the system, you have more stake in the system.

Therefore you will try not to sabotage the system that is the basic idea of proof of stake in this case. So you get a higher weight age in the process and then if somebody wants to create thousand identities, he has to then distribute the amount in order to be selected, he has to distribute his total stake into these thousand identities, therefore he does not get any extra probability of being part of the committee because your stake has to be now split into smaller stakes.

So that is the idea that they use and then this committee votes in multiple different steps and depending whether the network is at this time synchronous or that it has not been, you know some adversary did not delay stuff in the network out of bound, etc., then it takes less number of steps if the network is being under in some kind of an attack by adversary and the network becomes a synchronous, then it takes a lot more steps to actually converge on the voting.

Once the voting happens, the block that is being voted on whether it should be accepted or not or whether there is an alternative block proposal that should be voted in, either everybody the committee agrees that or at least if all the honest members of the committee, honest in the sense that they are not affected by a Byzantine behavior, if all the honest committees agree on a block, then that block becomes a final a selection for that epoch. In the other case, it may become tentative block and a tentative block will then has to wait until later epochs in which there will be a proper final block selected.

Once that is done, then all the previous tentative blocks get confirmed. So, this is the process and this voting or the or rather agreement is done by a Byzantine agreement algorithm that was not a standard Byzantine agreement algorithm that was and Byzantine agreement algorithm called BA star that was invented for this purpose. So, there is no proof-of-work which means the scaling is much easier, more and more nodes can be added because we are

not even making everybody vote, we are always selecting a fix size committee.

Every time I form the committee, I select different set of nodes or some of them may overlap, but you know everybody gets a chance and with higher stake you get a better chance, etc. The idea in the Byzantine guarantee here is that the nodes with two-third of the stake are honest and so at the max one-third of the stake holding nodes would be corrupt or could be adversarial and this is unlike the bitcoin them the mathematical theoretical analysis has been done by the authors of Algorand and there is a pretty good bounds on everything that was done.

Since the process is always done by selecting only thousand nodes and they do not have to do any proof-of-work, this is very low on CPU load and then they can potentially run on modern mobile phones.

(Refer Slide Time: 24:38)

Algorand features

- Sybil attack not possible
- Double spending (safety) not possible
- No Forking (safety)
- Safe from DDoS attacks
- Transaction Efficiency

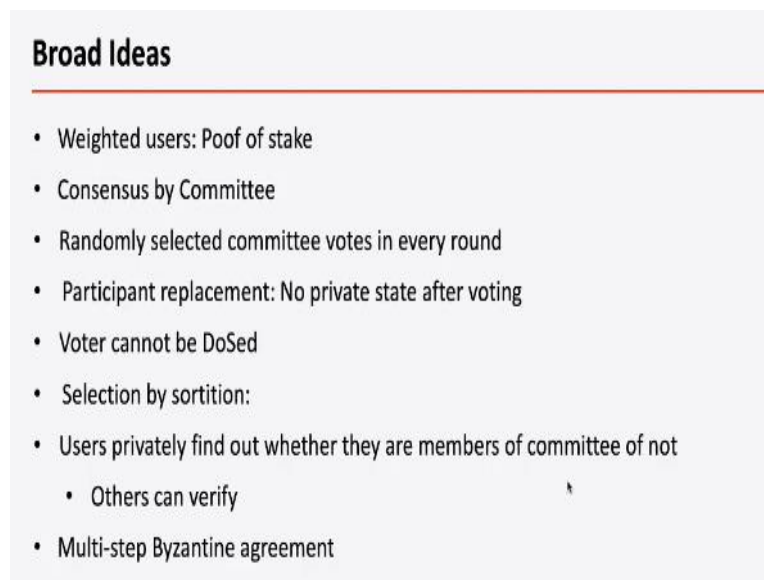
So, as I said that the way the Sibyl attack is not possible in Algorand is because you weight the nodes by the stake that they have, therefore creating multiple identities but having a fixed amount of money at your disposal does not help you with the sybil attack. So, one more thing is that in Algorand, there is no forking possibilities because everything is done together unlike bitcoin blockchain where everybody is creating their own, anybody who solves a puzzle may create its own block and then they get percolated in the system at different speeds.

There may be temporary forks and then the forks has to join together depending on highest

length chain to which people connect their next block, etc. Here, there is no forking and since there is no forking, double spending is not happening because there is always a single chain that is being created. DDoS attack is harder to do in this because normally you do DDoS attack on the nodes that are minors in order to stop the blockchain from growing, now here since you are choosing randomly about let us say 1000 nodes, but they are different from one epoch to the other and this is epochs are very fast.

So, by the time the adversary knows that this is member of a committee, so let us DDoS this one, the next epoch comes and then when the next epoch comes, the committee member is no longer a committee members, so DDoSing that committee member does not stop the process of building the next block from happening and therefore it is kind of safe from DDoS attack. Then the transactions are efficient in the sense that the transaction rates can be much higher because you are doing it in a scalable way.

(Refer Slide Time: 26:55)



Broad Ideas

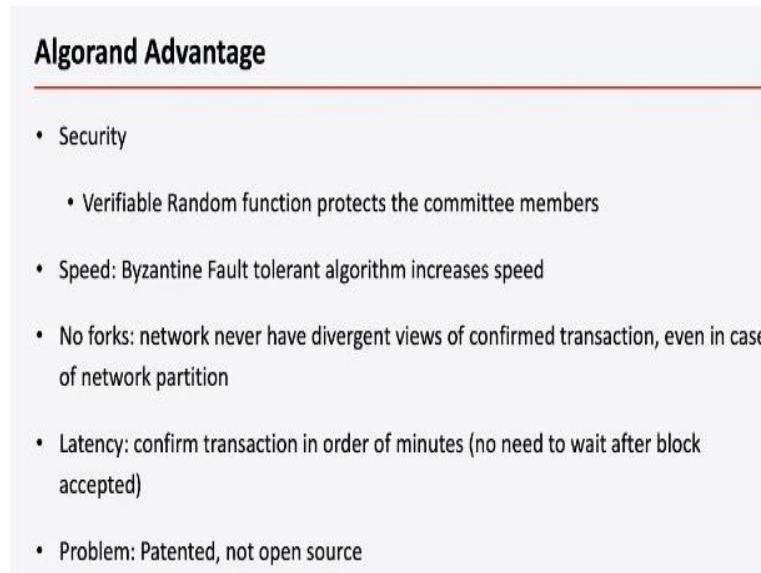
- Weighted users: Proof of stake
- Consensus by Committee
- Randomly selected committee votes in every round
- Participant replacement: No private state after voting
- Voter cannot be DoSed
- Selection by sortition:
 - Users privately find out whether they are members of committee or not
 - Others can verify
- Multi-step Byzantine agreement

So sorry, here is a spelling mistake here, it should be proof of stake. So users are weighted based on their stake. Consensus is done by committee. Randomly selected committee votes in every round and participants get replaced after they participate in the committee. They do not keep any private state after voting, so they have no knowledge that they were part of the committee after that epoch. So therefore, voters cannot be DoSed or subject to denial of service. Selection is done by this process of using VRFs or verifiable random functions, this process is called sortition.

Everybody self selects itself based on the computation of random function and then

everybody can verify because during this process of generating this information, there is a proof that he belongs to the committee or not can be verified and there is a multi-step Byzantine agreement called BA star that was invented for this purpose.

(Refer Slide Time: 28:06)



Algorand Advantage

- Security
 - Verifiable Random function protects the committee members
- Speed: Byzantine Fault tolerant algorithm increases speed
- No forks: network never have divergent views of confirmed transaction, even in case of network partition
- Latency: confirm transaction in order of minutes (no need to wait after block accepted)
- Problem: Patented, not open source

The advantage is that it has a lot more security from DDoS kind of attack, from Sibyl attacked and changing the refreshing the committee every epoch makes it very hard to target more than one-third stakeholders to become corrupt, etc., etc. The speed this Byzantine fault tolerant algorithm increases the speed, you know the mining process and bitcoin is much more time consuming and computationally intensive. There is no fork, so therefore it is difficult to do, even network partition can be tolerated in this because of the way this happens.

The latency to confirm transaction is in the order of minutes, so you can actually confirm blocks in the order of minutes and you do not have to wait for multiple blocks to build on top of your block before you feel confirmed. So everything gets done very fast. The problem is that it is patented and it is not open source. So therefore, we cannot really you know use it like Ethereum or Hyperledger, etc.

(Refer Slide Time: 30:08)

«Traditional» contract

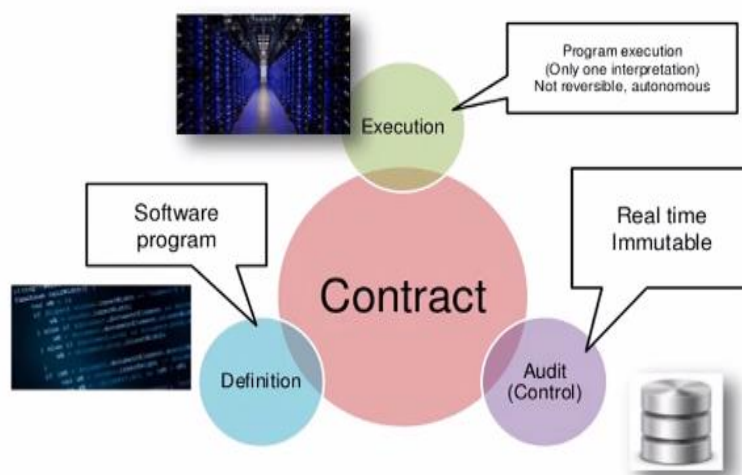


So now, then we talked about blockchain 2.0 where the idea of the block and the connecting blocks with hash to create a blockchain and then having them replicated all over the places all this nice ideas of blockchain 1.0 is added with the idea of the smart contracts. The smart contracts as we have discussed earlier is basically a computer equivalent of having a contract to do something and that that may also lead to exchange of money or exchange of goods versus money or exchange of labor versus money, etc.

The execution of the contract is often done under the purview of the law. So that is what this picture kind of shows and then an audit trail is created by keeping the contract information when it was created, when it is executed, etc., etc., so then auditor can check them.

(Refer Slide Time: 31:01)

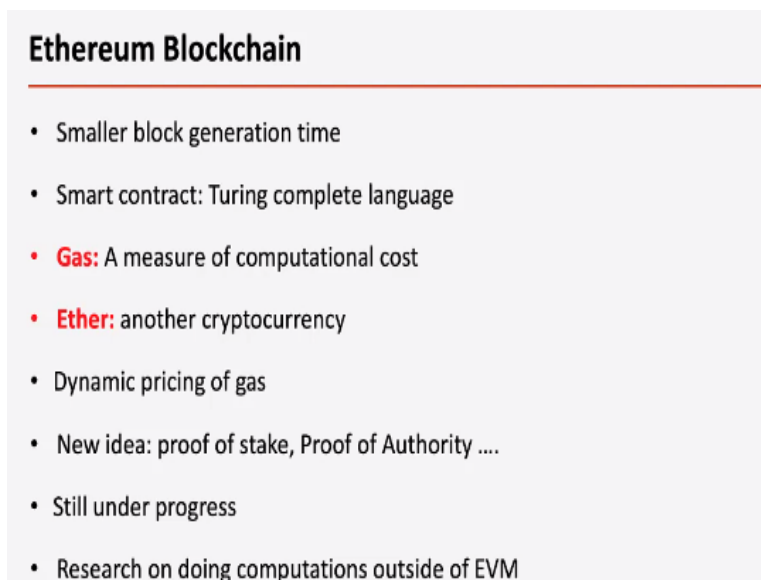
Smart contract



So, this can be replaced in a smart contract based scenario where all these things are done by

computer program. So execution is done through program execution and program execution is done on every node that wants to validate the transactions on the Ethereum virtual machine. The contract itself is a software program or called smart contracts and then the ledger basically allows you to do the audit, so that is the idea of smart contract that basically made it Ethereum the forerunner of the idea of blockchain 2.0.

(Refer Slide Time: 31:41)



Ethereum Blockchain

- Smaller block generation time
- Smart contract: Turing complete language
- **Gas:** A measure of computational cost
- **Ether:** another cryptocurrency
- Dynamic pricing of gas
- New idea: proof of stake, Proof of Authority
- Still under progress
- Research on doing computations outside of EVM

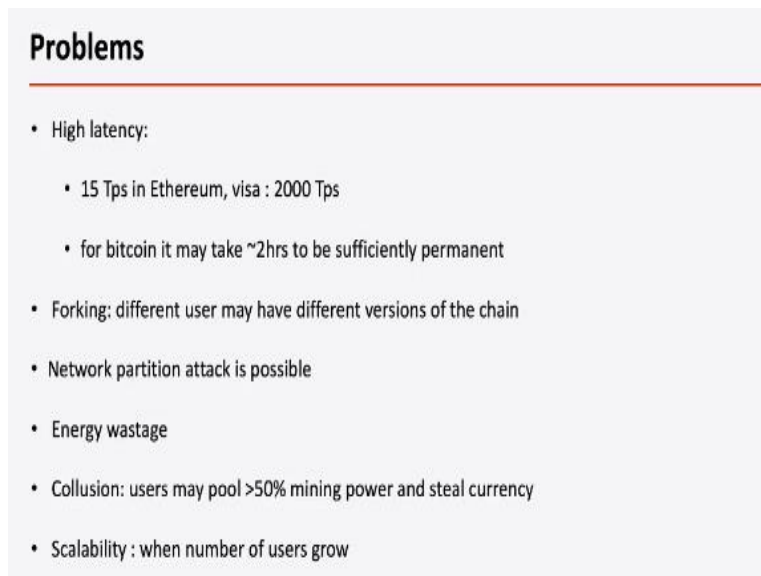
Ethereum blockchain has much smaller block generation time, although it uses a proof-of-work, but it is faster because everything is done pretty autonomously. Smart contracts, Ethereum made sure that the smart contract is done to a Turing complete language which leads to the question of termination going into infinite loop and so on and therefore they introduced the notion of gas. So gas is a measure of computational cost and then if you allocate a certain amount of gas.

Then your program even if it is written in a way that it could have the possibility of going into infinite loop because of the running out of gas that cannot happen. Then it basically has its a native cryptocurrency called Ether and the gas is dynamically priced and you can actually give more price for your transaction to go through for gas and stuff like that. Ethereum now has been for a while is toying with the idea of proof of stake, proof of authority and there are certain implementations already but Ethereum main net is still using proof-of-work.

So these things are still under progress and then the other issue is that EVM is obviously you know it implements execution engine for a Turing complete language, but it is still going to

be much slower. So the question is can you do more intensive computation outside the EVM, these kind of ideas are floating around.

(Refer Slide Time: 33:33)



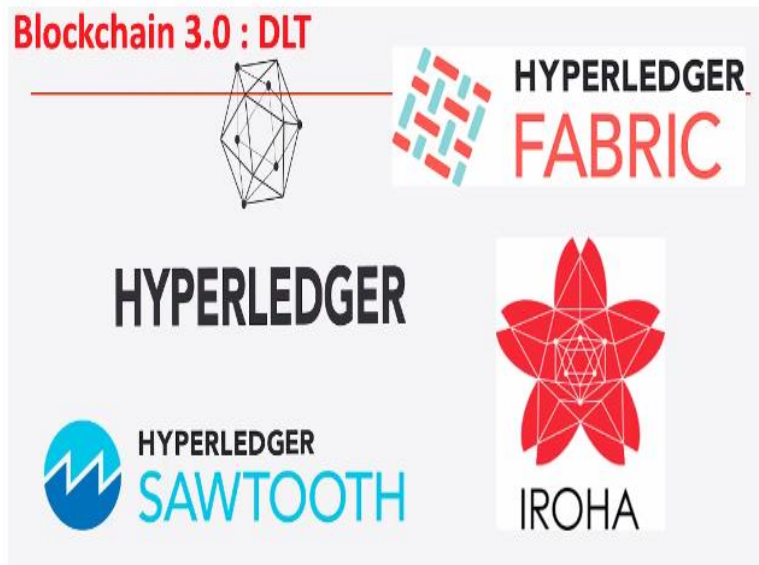
Problems

- High latency:
 - 15 Tps in Ethereum, visa : 2000 Tps
 - for bitcoin it may take ~2hrs to be sufficiently permanent
- Forking: different user may have different versions of the chain
- Network partition attack is possible
- Energy wastage
- Collusion: users may pool >50% mining power and steal currency
- Scalability : when number of users grow

So it is still of high latency, it is about 15 transactions per second in Ethereum compared to 2,000 transactions per seconds by say Visa, MasterCard, etc. So therefore, there is still a long way to go for it to be competitive against standard ecommerce you know transaction choice for. Of course, Bitcoin is even much slower, so therefore Ethereum is slightly in a better position. Forking can happen. Network partition is possible which can basically create problem in terms of creating forks in the system or double spending to happen which needs to be eventually invalidated.

Energy is still a concern because you are still doing proof-of-work of course in the newer ideas of proof of stake and proof of authority, etc., but reduce that. There is still a 51% attack possible and as the number of users grow, these throughput and everything could actually become much more problematic.

(Refer Slide Time: 35:11)



So the blockchain 3.0 came about in order to introduce, first of all it keeps the idea of the block in the chain, hash chain, the idea of you know double spending proof technique for accepting new blocks. It also keeps the idea of smart contracts all that stuff, but it actually first of all was designed to be permissioned block chain and also the idea of not only permission but also private blockchain because in order to use blockchain for enterprise, finance, or any kind of e-governance application, you cannot have all the data that everybody in the world can see.

So you may need transparency to certain extent, but auditability to some extent, but you do not want each and every data item to be free for all to download and analyze, etc. You also do not want to spend enormous amount of energy to actually generate your blocks and may ensure that your blocks have integrity. You also do not want that any random person can actually take part in transacting or creating blocks for your enterprise information system and therefore this idea of this distributed ledger technology.

So DLT or distributed ledger technology is often you know distinguished from blockchain even though it is kind of like an advanced version of blockchain and Hyperledger is one such example.

(Refer Slide Time: 37:13)

Hyperledger Fabric

- Started from IBM
- Currently owned by: The Linux Foundation
- Open Source-Apache License
- Flexibility
- No mining

So, it started from IBM, but currently it is a consortium called the Linux and it is owned by the Linux foundation and it uses an open source licensing model. It is flexible and there is no mining. It uses for the ordering service which allows you to actually choose based on your need whether you want to be a single node or solo ordering or you want to be a crash fault tolerant ordering like you know Raft type of ordering or you want a Byzantine fault tolerance model ordering. So, ordering processes where a consensus is needed and that is what a Hyperledger does.

(Refer Slide Time: 38:02)

Characteristic	Ethereum	Hyperledger Fabric
Description of platform	– Generic blockchain platform	– Modular blockchain platform
Governance	– Ethereum developers	– Linux Foundation
Mode of operation	– Permissionless, public or private ⁴	– Permissioned, private
Consensus	– Mining based on proof-of-work (PoW) – Ledger level	– Broad understanding of consensus that allows multiple approaches – Transaction level
Smart contracts	– Smart contract code (e.g., Solidity)	– Smart contract code (e.g., Go, Java)
Currency	– Ether – Tokens via smart contract	– None – Currency and tokens via chaincode

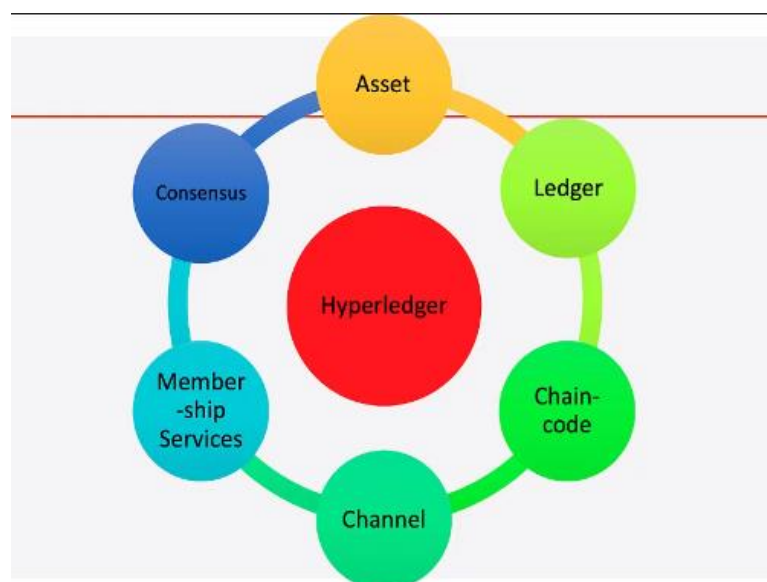
So, this slide basically shows you the various differences in the Ethereum and Hyperledger fabric because this question comes up often that whether to, because Ethereum also allows you to create a private blockchain that you can run as an Ethereum a network administrator which will be different from the Ethereum main net, but you can do that. So therefore often

this question is asked like you know whether to go for Hyperledger type of BLT or Ethereum. So, this kind of comparison often comes about.

So obviously, Ethereum is more generic whereas the Hyperledger is very targeted for enterprise systems. Ethereum is governed by the Ethereum developers, the Hyperledger is governed by Linux Foundation. Ethereum is permissionless, it could be public or private whereas in Hyperledger you are always permission and private. Consensus is based on proof-of-work or in the process currently it could be also with proof of authority or proof of stake and the consensus is at the entire ledger level whereas in Hyperledger the consensus first of all is flexible.

You can plug in your consensus and then it is actually at the level of transactions. The smart contracts are in available in both, but hyper ledger allows you to go to go write this code in any generic language that you want. It is basically a piece of code that needs to actually execute some business logic. There is a native currency in Ethereum and then you can also create tokens on top of them. Hyperledger has no native currency and the currency and token can be developed based on smart contracts or the chaincode.

(Refer Slide Time: 40:22)

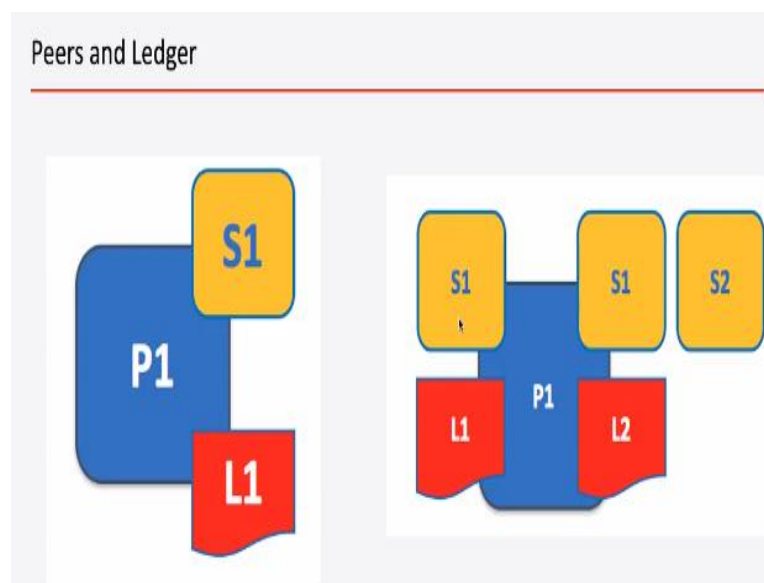


So Hyperledger has all this different concepts in them. One of the important concept of the membership service that basically gives identity to people and it is also pluggable. You could use a standard digital certificate technique or some kind of even enterprise you know authentication and access control mechanism. Hyperledger also is known to have smart contracts which are called chaincode. Hyperledger also has the concept of channels you can

run on the same Hyperledger network multiple private channels between different nodes which will have their own ledgers, right.

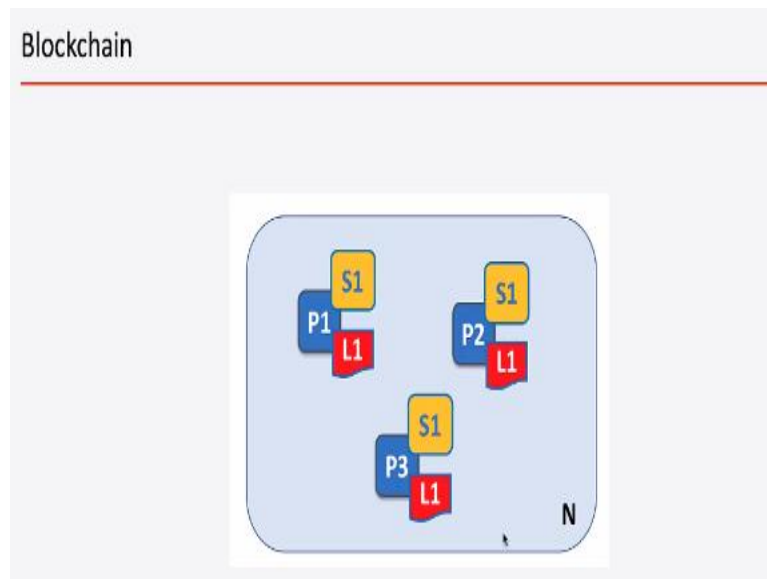
So, it can actually run in such a way so that you have multiple ledgers and you can actually query across ledger, so you can actually create a much more complex enterprise information system. There is this notion of digital assets, basically these are the ones on which transactions happen and then consensus is basically the ordering that happens.

(Refer Slide Time: 41:38)



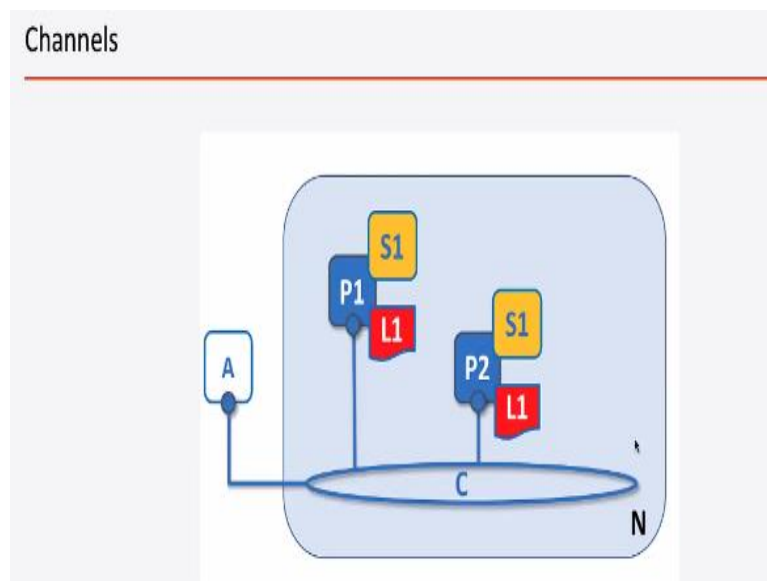
So, this is just a few pictorial representation from IBM's own slides. So every peer will have a docker running in them and the smart contracts run in their own dockers and then they have a copy of the ledger and the world state database and then you can have multiple smart contracts and multiple ledgers. So this peer in this case is participating in two channels, in one channel the ledger is called L1, the other channel ledger is called as L2 and within the namespace of this channel you have 2 smart contracts running on two docker containers and in this channel it runs a single smart contract in a single docker container.

(Refer Slide Time: 42:34)



Then the entire blockchain looks like this. Every node in this particular chain or the network, we see 3 peers. They run smart contracts, one one smart contract each and they are on the same channel that is why they are all sharing the same ledger.

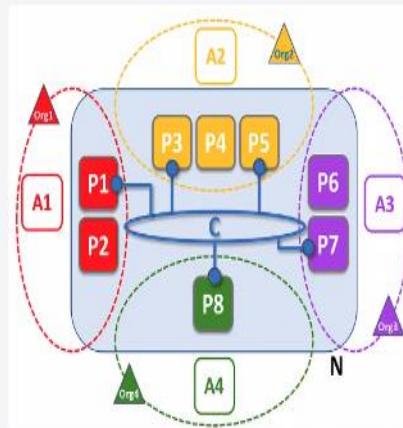
(Refer Slide Time: 42:53)



Whereas in case of a channel business, the same peer may participate in multiple channels like in this picture. The same peer is in 2 channels more, in one channel it sees this ledger, the other channel it sees this ledger. The same idea is in this channel.

(Refer Slide Time: 43:13)

Organization



So here, we have a single channel with participants, peers they all have the same ledger, but the Hyperledger also has these notion of an organization. So you can have multiple organizations like if multiple banks are using a Hyperledger based distributed decentralized ledger on which they do interbank transaction, etc. Then some peers will belong to this organization, let us say bank 1, this would be another, this may be auditor, this may be another bank, this may be the central bank whatever, right.

So, there are multiple organizations and in the trust idea is that within your organization all the peers you trust whereas across organizations you do not trust the peers. So, there is this distinction of the participants not only in terms of, you know when we studied Hyperledger, we saw that there are nodes that are called peers, there are nodes that are called validators, there are nodes that are called orderers, etc. So already there is a distinction of the type of nodes as compared to bitcoin and Ethereum where every node is at the same footing.

Then in this case also, we are seeing that there is organizational belonging of the peers and then the trust levels are also different.

(Refer Slide Time: 44:59)

Endorsement policy

Describes condition by which a transaction can be endorsed

- A transaction can only be considered valid if it has been endorsed according to its policy
- Each Chaincode is associated with an Endorsement policy

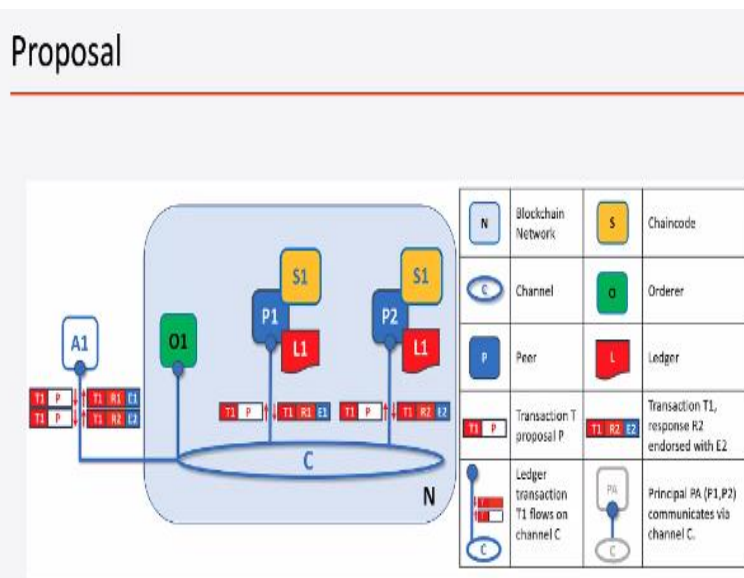
Examples of policies

- `AND('org1.member','Org2.member','Org3.member')`
- `Or('Org1.member','Org2.member')`
- `Or('Org1.member',AND('Org2.member','Org3.member'))`

Then we know that Ethereum uses this idea of endorsement. So when a transaction is initiated by a client, then it sends it to certain number of peers which are in the endorsement policy and this endorsement policies might say something like organization ones organization 2 and organization 3 should have one-one peers which would actually all of them must endorse. In this case, we are seeing an Or endorsement where organization 1 and organization 2, either of them can endorse this.

Then we are seeing the idea of and more complex endorsement policy where we have either an organization 1 member or you have to have both the organization 2 and 3 members.

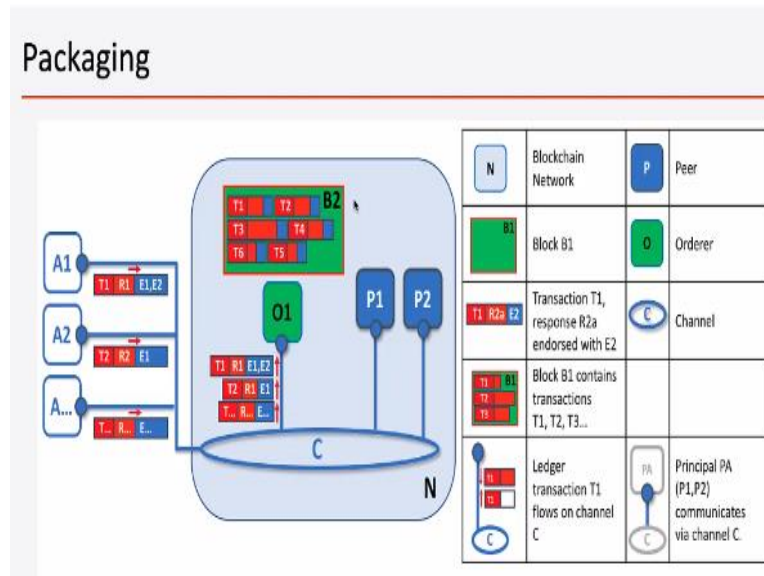
(Refer Slide Time: 46:02)



So, a proposal looks like this. So a proposal will have the transaction proposed by a client. The transaction proposal goes to the peers who will execute the smart contracts, create the

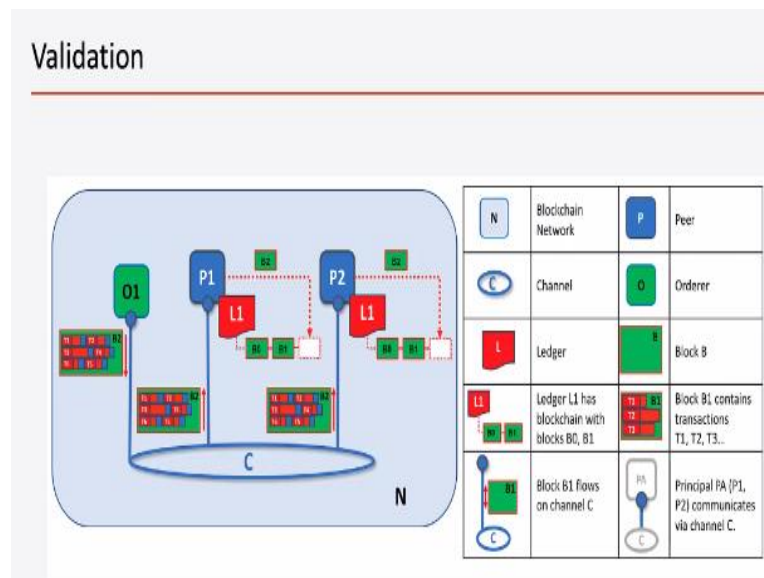
read write sets and then send them and with their endorsement and digital signature, they will send it back to the client and then the client will then send it back for ordering. So that is the idea of a proposal and the proposal being endorsed and then the endorsed transaction collected by the client and then sent back to the ordering service.

(Refer Slide Time: 46:47)



Then once there is sufficient number of these transactions, the orderer will do whatever is required to first of all reach a consensus on the ordering of the transactions across the different transactions, then it will create a block. So that is what the block is, B2 is a block that is being packaged here.

(Refer Slide Time: 47:16)



Then finally, every node that wants to validate, they will validate the block and then if they find that there is a mismatch between the read write set versions, etc., they will invalidate the

transaction, but this invalidation will be marked on the block, but it will not be actually, you know it is not like the invalid transactions will be thrown away, they will still be in the block but they will be marked as invalid and then every peer will now update their database, take a world state database as well as their ledger.

(Refer Slide Time: 48:04)

Pros

- Enterprise backing
- Open source: Apache license
- Modular architecture
- Private channels
- Smart contracts
- No PoW: Kafka Crash fault tolerant consensus

So, the pros of Hyperledger are that it has enterprise backing. It has many application examples. If you go to the Hyperledger dot org, you will find a lot of real-life applications. It is an open source. Modular architecture. It has an idea of private channels which helps you to keep certain information flow from another set of information flow and information systems. It allows smart contract and there is no proof-of-work, so you can have multiple different types of consensus as your situation demands.

(Refer Slide Time: 48:48)

Cons

- Permissioned
- Identities of parties must be known
- No cryptocurrency, no notion of reward

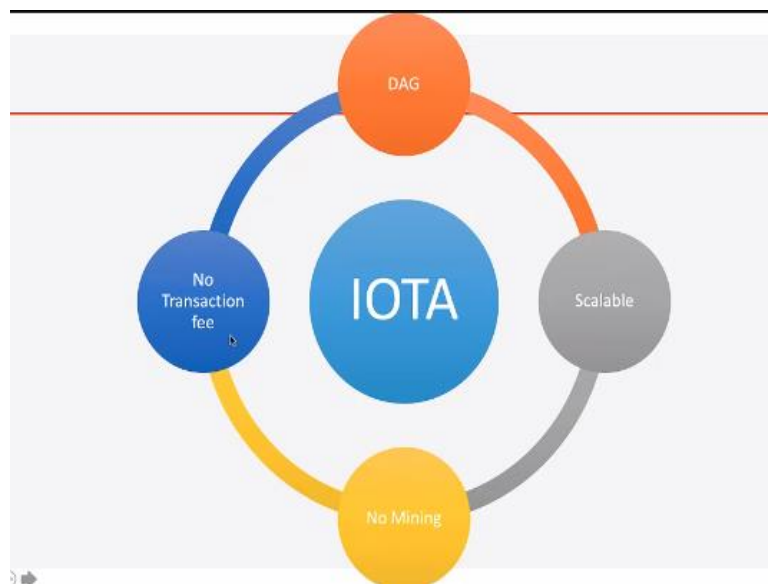
The cons are that it is permissioned. So you have to actually get you know some identity, so identities of the parties are known. Of course that may be considered as cons or may be as a pro depending on whether you know what your needs are in terms of trust. There is no cryptocurrency and there is no notion of reward, which I also do not see as a con because that is not what it was designed for.

(Refer Slide Time: 49:22)



So, then we also showed you blockchain without block and without chain which was the example was IOTA.

(Refer Slide Time: 49:32)



The idea was that there it is not really a chain of blocks, it is actually a DAG, directed acyclic graph, of transactions. It is highly scalable. There is no real mining involved and there is no transaction fee.

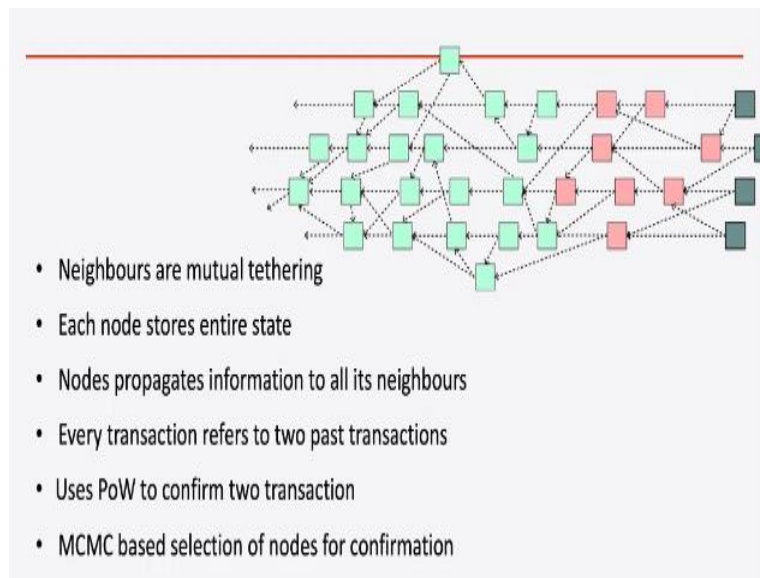
(Refer Slide Time: 49:49)

IOTA Blockchain

- Public Blockchain
- Genesis transaction creates maximum number of IOTA tokens that could be ever circulated
- Transactions are called sites, unconfirmed ones are called tips
- Sites are weighted in proportion to computational work needed
- Does not support smart contracts

So, it is a public blockchain. The genesis transaction creates some maximum number of IOTA tokens that could be circulated and never ever a new IOTA in that particular IOTA instance will be created. Transactions are called sites, unconfirmed transactions are called the tips. We saw the tip selection algorithms and sites are weighted in proportion to the computational work needed and therefore we saw this idea of the confirmation confidence and it does not support any smart contract.

(Refer Slide Time: 50:28)



So, the neighbors are basically transactions which are tethered to each other. Each node stores the entire state. Nodes propagate information to all its neighbors. Every transaction refers to two past transactions and it uses a little proof-of-work to confirm the transactions in order to you know avoid spamming and a Markov chain based selection of nodes is used for choosing

the tips. Now we discussed that a lot in the context of IOTA before.

(Refer Slide Time: 51:08)

Pros

- Good for micro payment
- Quantum immune
- No hierarchy of nodes
- Public Blockchain without rewards
- Network becomes faster as it grows
- Offline transaction is possible
- Parallel vs Sequential consensus

Pros is that it is good for micro payment. It has its own invented cryptography which is claimed to be quantum immune, which means that it does not depend on computational problems that are solvable fast in a quantum computer, you know theoretically possible to solve fast on a quantum computer. There is no hierarchy of nodes. Every node is basically a transaction. It is a public blockchain without any reward. Network becomes faster as it grows. Offline transactions are possible, we showed that when we discussed this.

You can have a lot of parallel activities going on at the same time because you can select different tips, different new transactions select different tips and or same tip but they can all work in parallel versus in the bitcoin on Ethereum, etc. Even in Hyperledger, the consensus is sequential when the block is being made. No other new block can be made in parallel because there is this sequential consensus process.

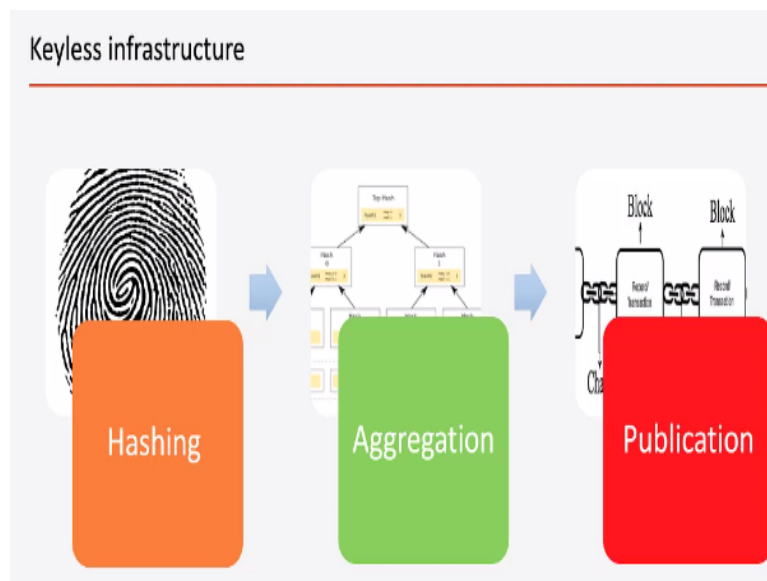
(Refer Slide Time: 52:32)

Cons

- Still uses Proof of Work
- Use ternary number system
- Crypto is not verified or tested: violates “Do not roll on your own crypto”
- Cannot handle smart contracts

It still uses proof-of-work. Uses ternary number system, The cryptography being their own, there is a possibility that the crypto is breakable. There is this idea in the cryptography parlance that do not roll your own crypto, you always use well-tested, well-validated crypto that has not been done and of course it does not have smart contracts.

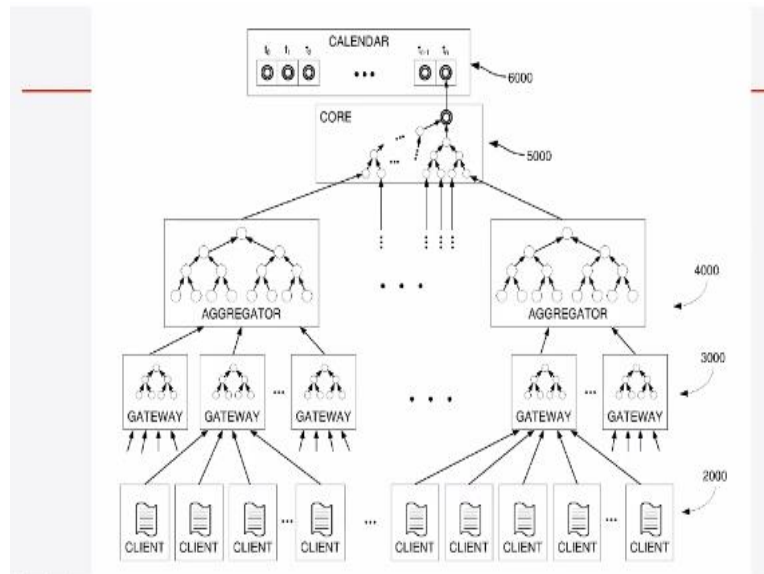
(Refer Slide Time: 53:52)



The other blockchain that we did not talk about in this course is the keyless signature infrastructure blockchain of Estonia. So it is actually an Estonian project that has gained a lot of traction and here the idea is that you basically depend completely on hash. There is no private key, there is no public key or there is no secret key and using the hash technology and the hash graphs which are kind of like Merkle trees, they actually have created a system where you can have integrity and time stamping of any document that is created in their government system.

So, we would very briefly look at what this is. So, it is based on the idea of hashing, aggregation of hash in the form of a hash tree or kind of like Merkle tree and then publication of root hashes. So what that means?

(Refer Slide Time: 54:08)



So, it means that say you have all these government organizations. Each organization will have some gateway and for every second of the day, any document that is created for e-governance, be it the creation of a license, creation of an identity, a registration of a property, etc., all that stuff are always being created right. So at any point in time, at every second whatever is created is hashed, all the documents are hashed and put into a hash tree at the gateway. Then for the next second again, the gateway will create another hash tree and percolate it to aggregators.

So it may be multiple layers of aggregation. At the end, all these different locations of the government and offices and everything, all the documents that are created the hash eventually gets you know rolled into a hash tree and then eventually those hash tree roots are again aggregated from various locations and eventually it goes into a core hash tree and then finally the root of that core hash tree is now put into something called a calendar blockchain. So for every second, there will be a single root hash that gets into the calendar block chip.

So, now you have a kind of new block every second of the day that is permitted here and they are hash connected and then every Sunday or something, they will publish the latest hash. Now see what happens that suppose this is my property paper that was created at certain date,

so I come back to the office to validate that this has not been tampered with. So what you have to do is they have to, by the way when they gave me this document and then the gateway hashed it, they gave me a token and this token basically contains the hash of the let us say this is my document.

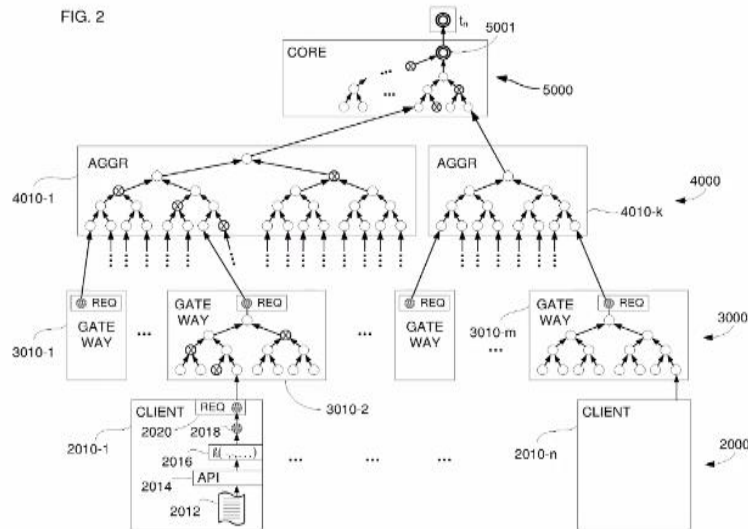
In order to check whether the document hash is correct, what they have to do is they have to give me in the token they give me this hash, this hash and that is it. So when I come back, what they will do is they do not need these documents or these documents, all they need is look at my token and see whether it matches their version of my token and then they will take my documents hash and then this is in the token, so together they will create this hash. Then from my token, they will retrieve this hash.

Then by this together they will create this hash and then they will check whether the hash here is actually matching this root hash. Now in order to check whether this root hash matches with the root has now being calculated when I bring back my document, they have to basically have the right set of you know complementary nodes hash values which will eventually lead me to that particular seconds calendar hash. So, integrity of this calendar hash has to be guarded critically.

Because if the integrity of this hash values are compromised, then certainly somebody could have changed some of these hash values as well and then accordingly change the integrity of the corresponding top hash value. So in order to ensure that none of the hash values that were stored for that particular second has ever been tampered with, integrity of this hash value is very important and that is why this is actually a block chain with a hash connection between all the subsequent nodes.

In order for public trust, in order to enter enhance public trust, what they do is that every so often they will publish in all daily newspapers and all other channels the latest hash periodically. So if anybody has tampered with anything, then if challenged, people can say that on so and so date you told me that was the latest hash, now go and show me that that is still the case and then if they are matched, then that means nothing before that has been tampered with, so that is the idea.

(Refer Slide Time: 59:36)



Then idea is been implemented a very painstakingly throughout the country with every location having the gateways and then aggregators throughout the entire country's network. S, this is a very extensive project and it is actually taken care of by a company called Guardtime and it is a commercial company and they actually have implemented this very successfully. So everything in the e-governance in Estonia, so if you actually get time to look at the Estonia on Google, look at the use of this hash technology, this case either, or this is called a keyless signature infrastructure.

Because they are only doing hash, there is no public key, private key or anything and then a whole idea is based on the idea that the all government documents should be kept you know in a tamper-proof manner and there should be a way to prove that nothing in the government information system has been tampered with and therefore the trust of the citizen on the governance is much higher which is not the case in most countries because there has been lot of corruption.

The IT systems are centralized under the control of NIC or some other government organization and they could tamper with all this data and then when you go back with your version of the document, it might not match with what we are giving you and what they have in their system and this leads to a lot of litigation and all kinds of stuff and now certainly reduces trust of citizens in the government. In this case that is what they have achieved is that their trust, the citizen's trust in the government has been enhanced significantly by use of this technology.

(Refer Slide Time: 01:01:38)

KSI: Pros

- Privacy
- Publication of the root hash guarantees immutability
- No mining: Highly scalable
 - 10^{12} registration per second
- Quantum immune
- Indefinite key expiry

So the pros is that it is private because none of the data leaves the government, only the hash that gets percolated as well as the token gets only the complimentary hashes, so no data gets in the hands of somebody else. Publication of the root hash periodically guarantees the immutability of the calendar hash chain and that is what gives the trust. There is no mining involved and they are scalable. So they do actually 10 to the 12 which is basically a million millions which is a billion registrations per second.

The hash is quantum immune and there is no notion of key expiry because there is no keyed SoS to speak to keep. So when you do a registration of a document or something you get a token and the token is basically the complimentary hashes in the hash tree.

(Refer Slide Time: 01:02:46)

KSI: Cons

- Private/permissioned
- Patented by Guardtime
- Smart contract not available
- Loss of token

It is private or permissioned in the sense that for finally the calendar blockchain and the entire

hash chain system is actually provisioned and organized by the government with the help of this company called Guardtime. It is also patented by Guardtime, so that is a problem and smart contract is not available in this case because here there is no other thing that is being done for this system except for the integrity of the documents and there is possibility of people losing their token in which case they cannot go back and verify things.

(Refer Slide Time: 01:03:27)



So in summary, what we have seen throughout the class, of course we discussed something today that are not part of the class that is Algorand and KSI blockchain and also another thing that we did not discuss is the Zk-Snarks or the idea of Zcash which is built on top of bitcoin and the idea there is to use zero knowledge proof to actually give guarantees of proper you know anonymity rather than pseudoanonymity, but in general these are some of the concepts that we have discussed in the concept and we did not discuss some of them.

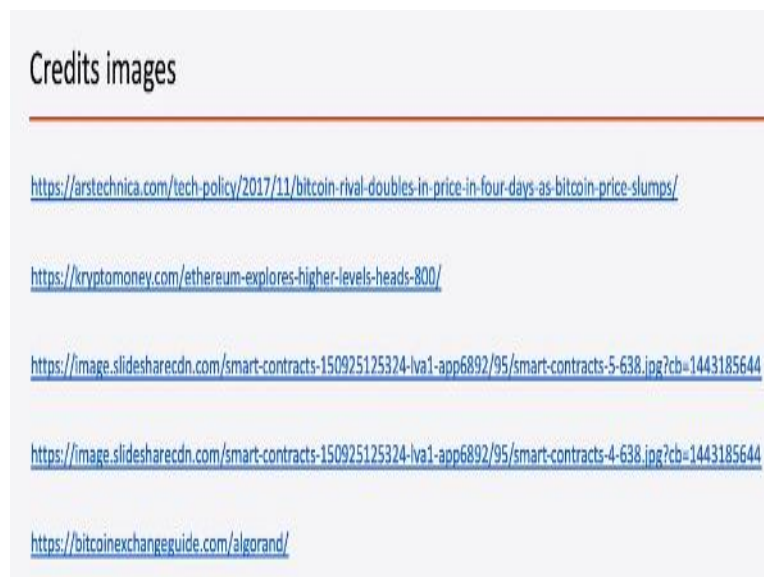
For example, we talked sometimes about the game theoretic sense why incentive mechanism works in keeping people honest and so on, but actually game theory people will analyze them as games cooperative or non-cooperative break games actually and then they will argue that this particular strategy achieves the Nash equilibrium and all this concepts and cryptographic sortition this idea we saw that the case of Algorand. We saw extensive use of Byzantine fault tolerance or BFT. Zk-Snarks is the technology of zero knowledge proofs behind the full anonymity of transactions in Zcash which we did not cover.

We talked about permissioned block chains and obviously un-permissioned block chains, etc. So there are lots of different concepts that are either covered or mentioned in this class and of

course as I said from the beginning, this class is not about learning how to program a blockchain which you can learn very quickly with the help of online tutorials, here the idea has been always to actually give you a very good idea about how to conceptualize understand the differences between different blockchain technologies.

The salient differences so that you have a better knowledge of how to choose what is the most suitable blockchain technology for your particular problem or whether blockchain at all is the solution to your problem, etc. So in that sense, you know we have covered a lot of grounds, but of course there is a lot more to learn and you can probably have an entire master's degree program in blockchain which obviously in an 8 week course we are not covering, but you get hopefully some basic idea that we covered.

(Refer Slide Time: 01:06:39)



At the end of these slides, I have given a number of credit to various images we have used in this set of slides.

(Refer Slide Time: 01:06:51)

<https://www.pinterest.co.uk/pin/5066618308753012/>

<https://chriscapria.wordpress.com/2013/09/02/bitcoin-mining-explained-like-youre-five-part-2-mechanics/>

<https://medium.com/@ihartikk/a-blockchain-in-200-lines-of-code-963cc1cc0e54>

<https://medium.com/hashtaagco/blockchain-say-what-8e16ed29e543>

<https://patents.google.com/patent/US20170033932>

<https://medium.com/iota-et-tangle/presentation-iota-tangle-2d6e63e3a70>

<https://www.bitcoinbeginner.com/blog/what-is-iota/>

But also in this if you go there, you will find other interesting documents and the blogs etc., that can help you to enhance your knowledge.

(Refer Slide Time: 01:07:00)

<https://medium.com/@philipsandner/comparison-of-ethereum-hyperledger-fabric-and-corda-21c1bb9442f6>

<https://www.cryptocompare.com/mining/guides/how-to-mine-zcash/>

<https://www.criptonoticias.com/aplicaciones/zcash-celebra-primer-aniversario-impulsando-privacidad-intercambios-atomicos/>

<https://slack.com/apps/A0P9ZU35Z-crypto-currency-coin-market-cap>

<https://www.wired.com/story/how-to-set-up-twitter-lists/>

<https://www.vectorstock.com/royalty-free-vector/crypto-currency-zcash-silver-symbol-vector-19123605>

<https://atozforex.com/news/zcash-price-long-term-forecast-2025/>

<https://www.dreamstime.com/stock-illustration-hank-you-text-illustration-social-icons-tablet-computer-mobile-cellphones-cyan-digital-world-mag-background-image48170847>

Hyperledger Documentation

So when we come back, we will talk about some applications that we have worked on on the use of blockchain in e-governance and then after that we will talk about some final thoughts about blockchain and try to correct some of the misconceptions people have about blockchain. So, we will come back soon and till then stay tuned. Thank you.