UNIVERSITY of NICOSIA

Week 1 - Session 2

Blockchains and the Architecture of Trust

BLOC 512: Blockchain Systems and Architectures

Session Objectives

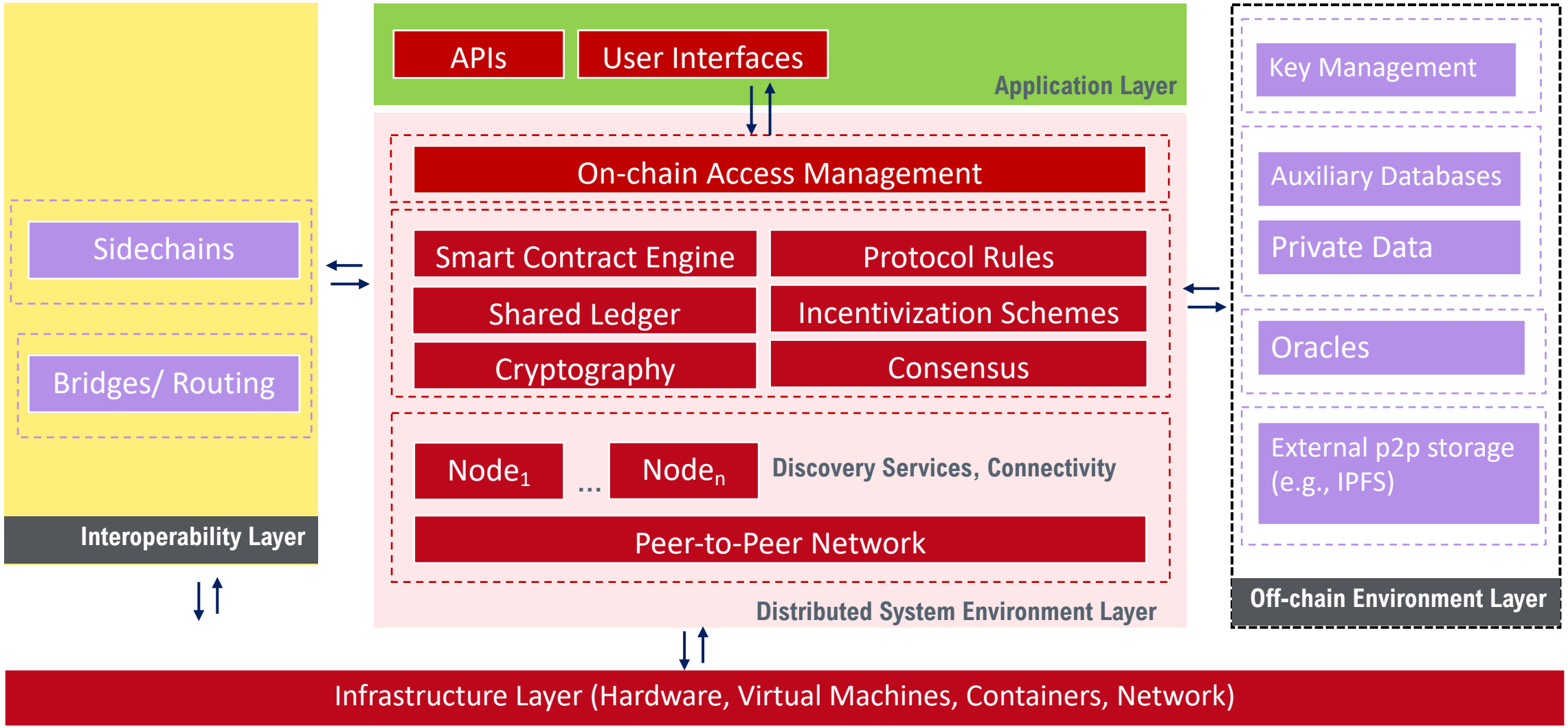
- The second session will cover basic methods of reaching consensus between mutually distrusting parties. We will review a *subset of such protocols*, discuss briefly how they work, what security can they provide and how scalable they are.
- While diving deep into the technical details is *out of scope for this course*, you are encouraged to explore the concepts further by following the many links provided.
- Introduce the idea of state machine replication
- Raise awareness of the importance of blockchain security
- Learn about original Proof-of-Work algorithm and examine optimization and hardening methods
- Understand advantages of modern consensus algorithms adapted for DLTs/blockchains and their weaknesses
- Explore other interesting or promising models

Agenda

1. The architecture of trust
2. Reliability and Decentralized Systems
3. Approaches to Consensus
4. Nakamoto Consensus
5. Improving PoW
6. Alternative Consensus Protocols
7. Blockchain's Big Bang
8. Conclusions
9. Self-Assessment Exercises and Further Readings

Flashback

DLT Layers & Architectural Components



Building a Trust Architecture

State Machines and Distributed Systems

- A blockchain can be viewed as a ***state transition mechanism*** whereby a state is modified from its initial form to the next one and eventually to a final form by nodes on the blockchain network as a result of a transaction execution, validation, and finalization process.
- Replicated state machines share several characteristics with blockchains:
  - **Fault Tolerance:** replicated state machines offer a mechanism to implement fault-tolerant services to a distributed system. The state of the system is replicated to various serves in order to cope with any failures and offer reliability of services.
  - **Consensus:** replicated state machines rely on a consensus protocol to synchronize the state between multiple clients.
  - **Voting:** replicated state machines depend on a quorum of voters in order to reach consensus on the correct state of the world.
  - **Communication:** Updates to the state are communicated to all servers/nodes. The system guarantees that information stored by one component is replicated and delivered to the other components even in the case of failures.
  - **Cryptography:** replicated state machines utilize cryptographic techniques to address faulty or Byzantine processes. Digital signatures are used to authorized the exchanges of messages.
  - **Ordering:** all requests from components are ordered in a similar fashion as with the blockchain data structure where blocks and transactions are ordered.

Design Considerations

- Decentralization
- Data Structure
- Consensus
- Access Management
- Off-chain
- Interoperability?

Reliability and Decentralized Systems

The Byzantine Generals Problem

Reliability and Decentralized Systems

- **Consensus** – how does distributed systems agree about the correctness of some state (e.g., shared ledger).

> "Consensus allows processes to reach a common decision that depends on their initial inputs, despite failures." – V. Hadzilacos & S. Toueg, 1993

- The ***agreement challenge*** is not new. It is a **fundamental problem in distributed computing** and multi-agent systems - on how to achieve overall system reliability in the presence of a number of faulty or misbehaving processes [1].

Such an algorithm enables a distributed system to:

- attain reliability of services
- tolerate failures (i.e., fault-tolerant)
- build a level of trust between different nodes

[1] Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. In ACM Transactions on Programming Languages and Systems (TOPLAS), vol. 4, no. 3, pp. 382–401.

What does it mean for DLTs/Blockchain?

- **Blockchain consensus** is an algorithm that ensures that all peers of a Blockchain network reach to a common acceptance or consensus about the real-time state of our shared distributed ledgers.

- For Blockchain-based distributed systems:
  - enables a unified agreement
  - aligns economic or other incentives
  - insures fairness
  - enables fault-tolerance
  - ensures that everyone works on the same state of the world

- **Variations:** each consensus algorithm is unique by design, and operates according to a different ground of objectives

Consensus and Blockchains

- Is proof-of-work the only option (consensus)? Do miners really need to be paid (thus requiring a native digital currency in some form of value-tokens)? How open should the system be (permissions and identifications)?

  - It depends on the context. Generally speaking, blockchain is not a panacea. For truly open payment systems, where everybody can join and nobody is required to identify themselves, proof-of-work is currently taken as the most secure. There are concerns about „openness“ of Bitcoin due to mining concentration and costs and there are potential candidates for other models which we will explore in later sessions.
  - If absolutely open and public blockchain is not of primary concern, then proof-of-work might be redundant, but some form of distributed consensus is still required for an agreement between parties that are each pursuing their own goals.
  - You are invited to read about „blockchain“ in private replicated databases by Gideon Greenspan: Ending the bitcoin vs blockchain debate.

Blockchain and Consensus

There are several types of consensus protocols:

- Crash Fault-Tolerant
- Byzantine Fault-Tolerant
- Incentivized

- For our needs, consensus will mean a **Byzantine agreement** between parties of a network *i.e.,* peers agree on the ordering of transactions that flow through the system, even when a minority of peers are dishonest or some simply leave the network.

- Note that by „transactions“ we can also imagine a more general „state-transition“: when we‘re dealing with smart contracts for example, they transform inputs to outputs and by doing so change the contract‘s state. If all peers start with the same state (genesis block), they just need to update the starting state by *all* transactions in the same *order* and they will consequently share the same current view of the system state.

Approaches to Consensus

Background – Part I

Distributed consensus is not an innovation of Bitcoin. It was studied for decades. One of the first implementations came in 1999 with Practical Byzantine Fault Tolerant algorithm (PBFT) by M. Castro and B. Liskov.

Traditional BFT approaches in computer science are successful, proven and performant (capable of tens of thousands of transactions per second), yet they usually have some challenges:

- scaling to a larger number of nodes*, say thousands,
- often work only with known nodes that bear the consequences for any misbehaving.

Proof-of-Work, Proof-of-Stake and variations offer a fresh look at the consensus problem, they don‘t always rely on traditional methods from the distributed systems field but offer economically incentivized and probabilistic solutions to reaching consensus. Such a consensus is not final, but it works as a self-enforcing loop that makes participants progressively converge to an agreement and re-writing history becomes increasingly more costly.

Background – Part II

Research into both old and new consensus mechanisms is providing a thriving ecosystem, with many new and improved approaches being revealed surprisingly often. Some will never get traction, but few of them might engage the community enough to be deployed in live applications.

While traditional consensus mechanisms often require knowing the identities of nodes (implying the management of identities in a more centralized way, which is often tied to KYC requirements), in Bitcoin's Proof-of-Work the majority of nodes to reach a consensus is measured instead with brute computation power.

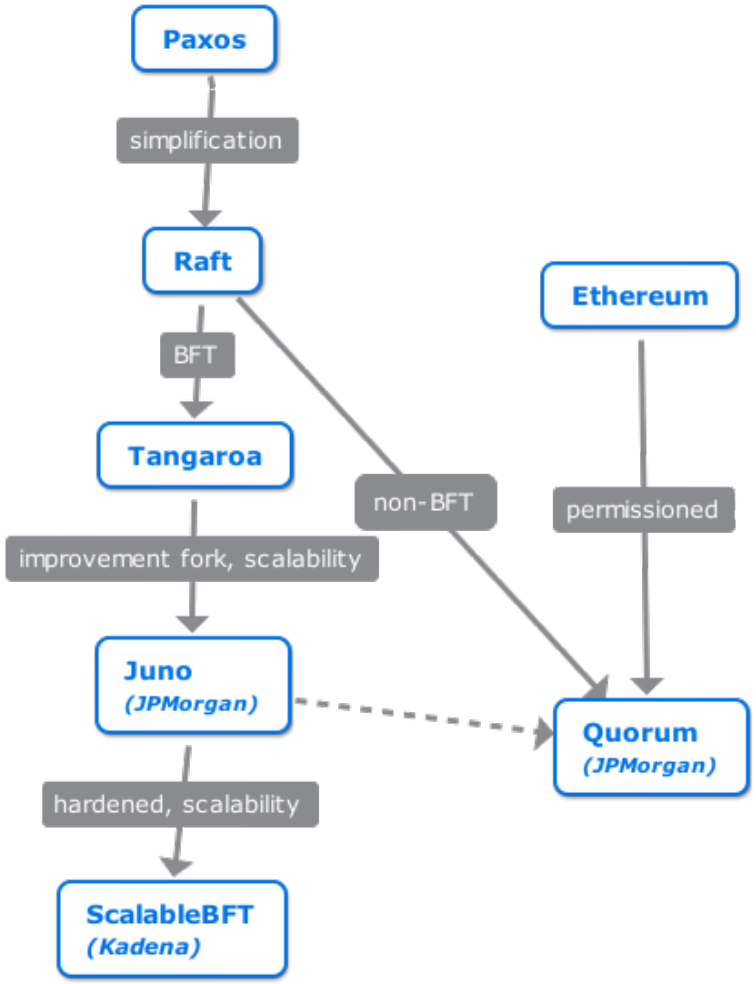
Nakamoto consensus: "It matters not who I am, judge me by what I do."

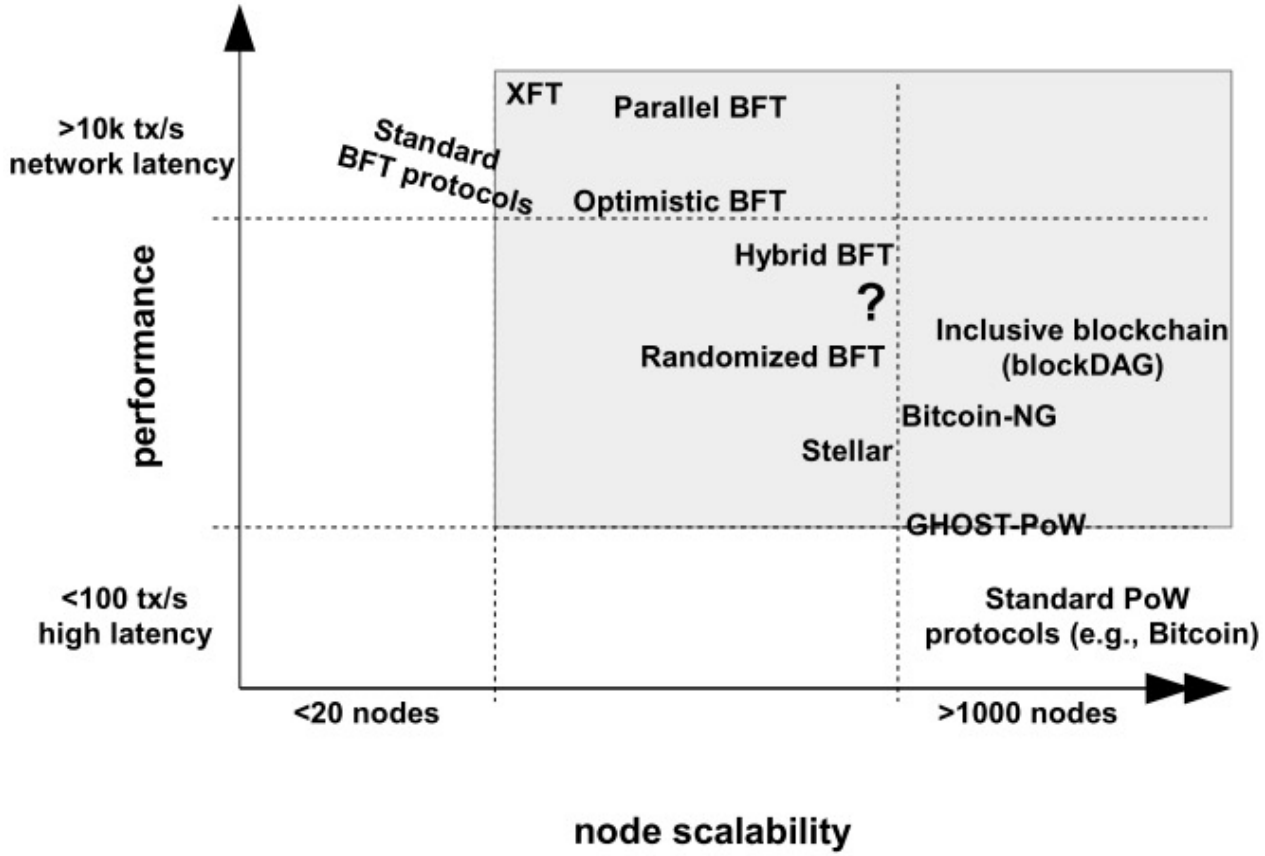
- This makes public blockchains suitable for *permissionless* participation.

Traditional BFT: "We are a small group who knows each other, and we have binding contracts to follow."

- This makes traditional BFT algorithms suitable for *permissioned* blockchains.

Consensus is hard to do right





Source: M. Vukolić, 2015: The Quest for Scalable Blockchain Fabric

Nakamoto Consensus

The Concept of Proof-of-Work

Sybil protection

- Hashcash introduced the idea Proof of work as a **Sybil protection mechanism**.

What is a hash?

- Hash is simply an integer (normally, a very large integer). Most hashing functions result in 256-bit hashes (integers between 0 and 2^256).
- Miner randomly tweaks input data (e.g., nonce) until the hash fits under specified threshold. The threshold (also a large integer) is established collectively by the network as part of the consensus mechanism. The PoW is only considered valid (solved) if hash fits under the threshold.
- Because hash functions are one-way, it is not possible to analytically calculate input data that would result in a small-enough hash. It is also impossible to tweak the data and repeatedly try to guess (in a brute force manner) the data that resulted to a given hash.

[1] Nakamoto, S., & Bitcoin, A. (2008). A peer-to-peer electronic cash system. Bitcoin – URL: https://bitcoin.org/bitcoin.pdf

The Concept of Proof-of-Work

But what do we mean exactly with Proof-of-Work?

- PoW blockchain protocols employ cryptographic techniques as a sybil attack mechanism where a ***prover*** seeks to convince a ***verifier*** that possesses knowledge of some secret key (that solves a mathematical relation)
- For this mathematical relation to hold true computational work is required. Thus, in essence the ***prover*** demonstrates to the verifier that it has performed a certain amount of computational work in a specified time interval.

Simplified PoW

1. Network provides a difficulty parameter δ e.g., 4 leading zeros
2. Miner frames the header of a block with the following info:
   - Bitcoin version
   - Hash of previous block
   - Merkle-root
   - Timestamp
   - Difficulty target
   - **Nonce (this is the puzzle the miner attempts to solve)**

- **Once solved the miner hashes the header with SHA256 and it becomes the current Block hash**

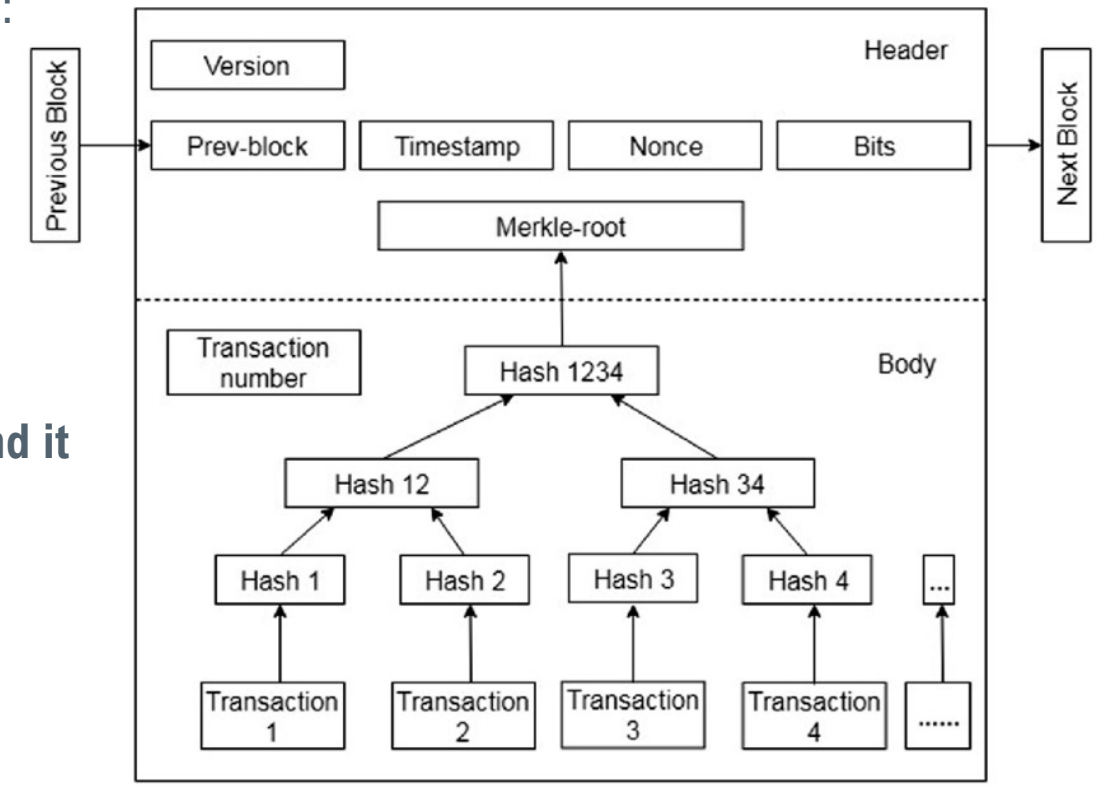


Fig. Internal block structure

Source: Wang, B., Chen, S., Yao, L., Liu, B., Xu, X., & Zhu, L. (2018, June). A simulation approach for studying behavior and quality of blockchain networks. In International Conference on Blockchain (pp. 18-31). Springer, Cham

… to sum-up

- This consensus algorithm has been proposed in the Bitcoin network, aka ***Nakamoto consensus***
- Each node in the network
  - is calculating the hash value of the block headers. Miners change the nonce frequently to get different hash values until the calculated value is equal or smaller than a given target value
  - the process repeats until a node hits that target value. In such a case the node broadcast the block to the other nodes
  - given the right nonce that derives the target has value, other nodes need to mutually agree on the correctness of the hash value. If the block is validated, then the new block is appended on the ledger.

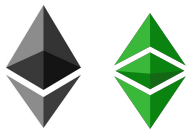
Examples of Proof-of-Work I

- Influenced by this idea many different hashcash-based PoW-methods have been devised over the years:

SHA-256

A function that belongs to the Secure Hash Algorithm (SHA) designed by the NSA and approved by NIST as a U.S. Federal Information Processing Standard. The SHA-256 function produces outputs of length L = 256 bits and is used by Bitcoin [1] and other related cryptocurrencies.

Ethhash



The Ethhash function [2] has been obtained from the Kekkak hash function [] combined with modified versions of the Hashimoto and Dagger [3] algorithms. Ethhas was designed to achieve the following goals:

1. **IO saturation;** to achieve ASIC-resistance by memory-hard computations
2. **GPU Friendliness;** since CPU algorithms have been vulnerable to botnets.
3. **Light client verifiability and startup:** to be able to verify the mining result and become fully operational and able to verify blocks quickly.

* **Note:** The heart and soul of the next generation of Ethereum is proof-of-stake (PoS)

[1] Nakamoto, S., & Bitcoin, A. (2008). A peer-to-peer electronic cash system. Bitcoin – URL: https://bitcoin.org/bitcoin.pdf
[2] Ethereum team, "Ethash", 2018, Available: https://eth.wiki/en/concepts/ethash/ethash
[3] Original paper of Dagger-Hashimoto https://eth.wiki/concepts/dagger-hashimoto

Examples of Proof-of-Work II

Scrypt

This function was developed to use a large amount of memory. It is a password-based key-derivation function that makes use of large vectors of pseudorandom strings [1]. Due to its memory-hardness it has bee widely adopted by many blockchain protocols as part of their PoW-algorithm aiming at ASIC resistance.

Cryptonote

This function relies on random memory access and has been considered as an egalitarian hashing algorithm because it can be computed by CPUs and GPUs but it is impractical for use by ASIC miners. It was originally proposed by the CryptoNote and Bytecoin.

RandomX

Was proposed as an update to Monero's Cryptonight egalitarian hashing algorithm. It is designed to be an ASIC resistant algorithm by using memory-hard techniques []

[1] N. van Saberhagen, "Cryptonote v 2.0", 2013, Available: https://cryptonote.org/whitepaper.pdf
[2] https://www.monerooutreach.org/stories/RandomX.html

Examples of Proof-of-Work II

Equihash

Equihash [1] has been proposed as another PoW algorithm which is memory-hard. In brief the algorithm has been proposed as a solution to limit the parallel implementations of memory bandwitdth.

Given some generic hash function *H* the *prover* is required to produce messages (m_1, …, m_k) with the following properties:

- The bitwise sum of the hashes is the zero vector:

$$\mathcal{H}(m_1) \oplus \ldots \oplus \mathcal{H}(m_k) = 0,$$

- The hash of the concatenated message has a given number of leading zeros:

$$\mathcal{H}(m_1 || \ldots || m_k) = 0$$

[1] A. Biryukov, D. Khovratovich, "Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem", Ledger, 2017, Available: https://doi.org/10.5195/ledger.2017.48

Examples of Proof-of-Work II

Concatenating PoW

Some hashcash solutions have been proposed in the literature by concatenating different hash functions. One example is the X11 hash that was developed for Dash [1]. This function combines 11 hashes i.e., Blake, Bmw, Grøstl, Jh, Keccak, Skein, Luffa, CubeHash, SHAvite-3, Simd and Echo.

The idea is to produce a heavier but arguably more secure hash function. Influenced by the Dash idea, many other functions have been proposed e.g., X13, X14

[1] E. Duffield, E.Diaz, "Dash: A Payments-Focused Cryptocurrency", 2018, Available: https://github.com/dashpay/dash/wiki/Whitepaper

Improving Proof of Work

Improving a Consensus Protocol

Proof-of-Work has some challenges, just like most of other consensus algorithms. We can classify them according to these main aspects:

- **Scalability** – to support the growth of the network
  - Number of transactions per second
  - Number of nodes involved in building on a consensus (miners/minters/validators/orderers)
- **Privacy** – to preserve open access and confidentiality
  - Preserving privacy of any node in the network
  - Preserving privacy of nodes involved in building on a consensus (miners/minters/validators/orderers)
- **Decentralization** – uniformity of the system
  - Technically, decentralization is imposed by the system's architecture.
  - Governance or coordination is a political aspect to decentralization.

Following, we will be focusing on the scalability aspect. The privacy aspect is in large parts still under heavy research, not everybody agrees on the suitable degrees of protocol-level privacy in the first place. Decentralization is debatable – the technical point of view is left for the reader to explore further.

Improving Proof-of-Work

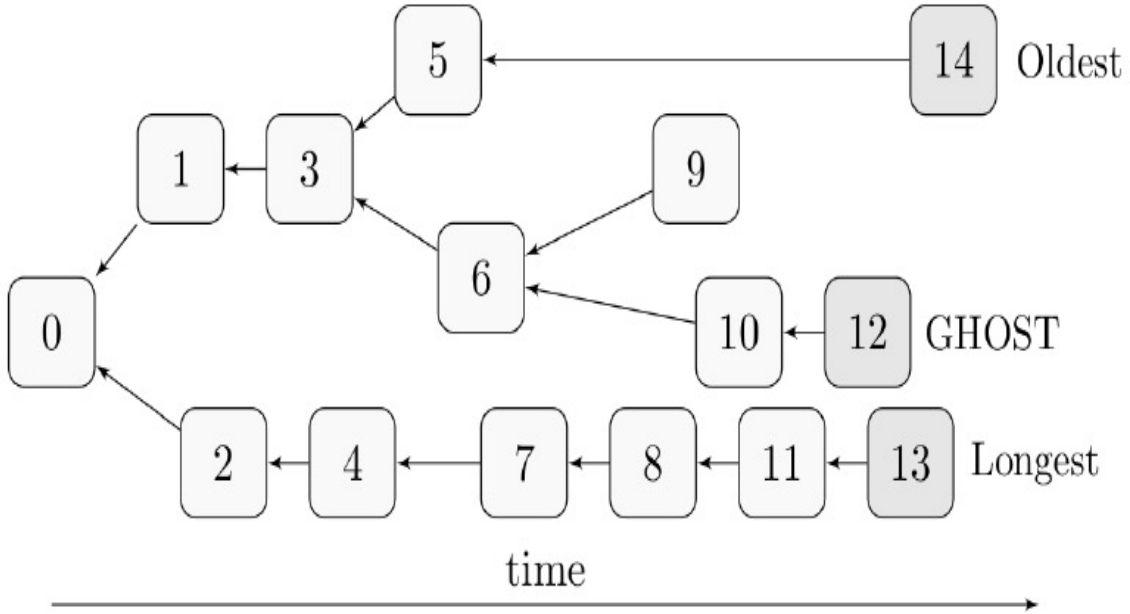
- The way consensus is reached via Proof-of-Work is of utmost importance to Bitcoin. Therefore, any improvements are extremely critical and painful to deploy. Proposals to change or even replace Proof-of-Work have been discussed since early beginnings, yet most of the time they were postponed or even dismissed.

- Lack of peer review and developer time is worrisome on one hand, but on the other it is not so pressing as such fundamentals are not supposed to be modified or replaced often, if ever.

- We will take a deeper look at the policy that proposed a ***different criteria for determining the valid state of the blockchain***, named GHOST. While it will not be implemented in Bitcoin in the near future, if ever, a modification of GHOST was successfully deployed in Ethereum. There are other proposals, but since a deeper consideration of them is out of the scope for this course, we are going to focus on some of them only, in order to see the general way of how they work.

GHOST

- In 2013, researchers Aviv Zohar and Yonatan Sompolinsky published [1] a modification of the Bitcoin protocol that would enable ***significantly faster generation of blocks***.
- The **Greedy Heaviest-Observed Sub-Tree (GHOST)** policy defines the best chain of blocks to include staled sub-chains, that are in Bitcoin discarded by design.
- In Bitcoin, the longest chain (more precisely, the one with the most work spent on creating it and is valid according to the protocol) is defined as a single-track of parent-child relations.
- What **GHOST proposes is that the best and most secure chain is actually the one that also includes blocks that are not on the main chain**, they are therefore in parent-subtree relations. Instead of the longest branch scheme, GHOST weights the branches and miners could choose the better one to follow.



[1] Sompolinsky, Y., & Zohar, A. (2013). Accelerating bitcoin's transaction processing. Fast money grows on trees, not chains.

GHOST vs. original Proof-of-Work

- GHOST was analyzed as being capable of generating one block each second (600 blocks in 10 minutes versus 1 block per 10 minutes as originally in Bitcoin).

- Bitcoin‘s 10 minutes per block are important!
- If Bitcoin as is would increase the frequency of new blocks, an important security gap between **the time that the block is generated** and **the time it is propagated through the network** would be diminished, perhaps dangerously so. Thus, some miners wouldn‘t be aware of a recently found block and would keep on mining on their older version of „the longest chain“.
- In other words, there would be more staled blocks due to **frequent reorganizations** of the chain. One consequence of that would be that a part of the work on mining would be wasted instead of used for securing the chain. Secondly, mining would be even more inclined to centralization, as miners with less hashing power would need to make costly switches to other chains more frequently. The main issue is **not the *exact* number** of 10 min/block or even 1 MB sized blocks for that matter, but the general size **range**. It could be interpreted that GHOST *hints* at some reserves Bitcoin currently has.

GHOST vs. original Proof-of-Work

- In general, absolute numbers are not that important. It is more sensible to consider a broader range of values, the order of magnitude, and what changes that implies.

  - Increasing the frequency of blocks reduces the security (faster blocks mean there will be more reorganizations of the blockchain because honest nodes will be less up to date with other nodes, while the attacker can gain advantage by not caring about the rest of the network and can build on his own chain with no efficiency lost). Equally important, the change of frequency would alter the ***emission*** of newly mined bitcoins, affecting its economics and predictability.
  - Increasing the block size is another path to increase the throughput of Bitcoin network and has similar effects: increasing transactions throughput, but reducing security for longer propagation times across the network. Despite heated debates on the topic, it is a fact that the propagation time of a block increases linearly with its size [Decker, Wattenhofer, 2013], it is also a fact that the bandwidth capabilities are increasing each year for about 14% [CISCO VNI].
  - Simulations are one way to evaluate such re-parametrizations. A recent one [Gervais et al, 2016] confirmed the option of a fairly secure increase in either a block frequency or size increase – as long as it is modest.
  - Theoretical analysis is another way and while it carries the most weight, it is also very complex [see Sompolinsky, Zohar, 2015]. A recent theoretical proof in June 2017 [Garay, Kiayias, Leonardos, 2017] showed the ***critical balance*** between the time the nodes synchronize over the network against the time it takes for solving Proof-of-Work round.

- However, not only the content of any change is important, a significant change of the protocol is in itself stressful. Hence, not to be taken too easily. Not just for the technical part, but economical and social as well. A store of value must be predictable and reliable.

GHOST vs. original Proof-of-Work

- To repeat: GHOST takes into the account also the work that was done on the blocks off the main chain and thus preserves the security of the protocol even with more frequent blocks. *What it does not do, is splitting the reward among main chain and sub-chains*. A modified GHOST is used in Ethereum, such that the mining reward (but not fees) is split up to the 7th level of sub-chains.
- GHOST was not added to the Bitcoin protocol and it doesn't look like it will be anytime soon. The most common technical concern is that higher rate of blocks would increase the risks of selfish mining (a.k.a. block-withholding attack). To mitigate this risk, a **Deterministic Conflict Resolution protocol (DECOR)** was proposed by Sergio Demian Lerner, also in 2014. It is used together with GHOST in Rootstock, a smart-contract extension to Bitcoin.
  - There are multiple versions: DECOR, DECOR+, DÉCOR++
  - You can read more about it here: How DECOR++ can eradicate selfish mining incentive by design .
- Even with recent improvements of GHOST, there are still open challenges, such as adjusting it to the changing network conditions, support for light clients, keeping block rewards proportional to the hashing power without unfair advantage to larger miners etc.
- As we have mentioned on the previous slide, changing any fundamental part of the protocol will be always met with resistance – not for the content of the change only, but for the very change happening in itself.

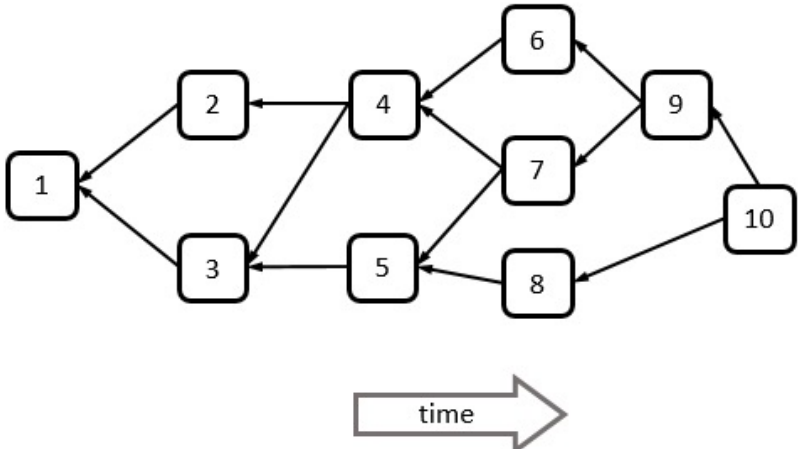
Improving GHOST – and changing the structure of blockchain

- Along with the GHOST protocol improvement for Bitcoin in 2015, Aviv Zohar and Yonatan Sompolinsky were joined by Yoad Lewenberg and proposed an alternative blockchain structure, called a **DAG (directed acyclic graph)** to be used as an inclusive protocol, where non-conflicting transactions from all blocks are included in the ledger, even the ones that are off the main chain. Fees are to be rewarded to the creator of the block, regardless if it's on the main chain or not.
- Very similar to Bitcoin, but exploiting a different data structure to drastically increase performance.

- SPECTRE currently received only some critics regarding usage with light clients, fees distribution, RBF transactions … and of course, applying it to Bitcoin would be quite radical for various reasons.

Improving GHOST – SPECTRE

- SPECTRE (Serialization of Proof-of-work Events: Confirming Transactions via Recursive Elections) is a security & performance improvement of the GHOST and the Inclusive protocol, published in 2017.
- It is constructed to be resilient to attacks of up to 50% of the computational power and can operate at fast block creation times (unlike the Nakamoto consensus as discussed before). In SPECTRE, transactions off the main chain are accepted by a pairwise voting, where the structure itself serves as a vote.



Source: Aviv Zohar, 2017: Serialization of Proof-of-Work Events, Confirming Transactions via Recursive Elections

DAG structure (Cont.)

IOTA Tangle is using a similar DAG structure, but has a different algorithm for detecting conflicting transactions. Instead of deterministic voting as in SPECTRE, IOTA uses a Markov Chain Monte Carlo algorithm that is probabilistic and can be exploited by certain types of attacks.

- IOTA itself is pretty special, as it has no blocks, just a DAG of transactions, that are processed at high speed. It also has no mining (therefore, no transaction fees and no difficulty adjustment scheme). We will look at IOTA in more detail in the session dedicated to the IoT. At this time, just keep in mind that it has raised some serious doubts as to the truth about its capabilities.

- https://www.media.mit.edu/posts/iota-response/

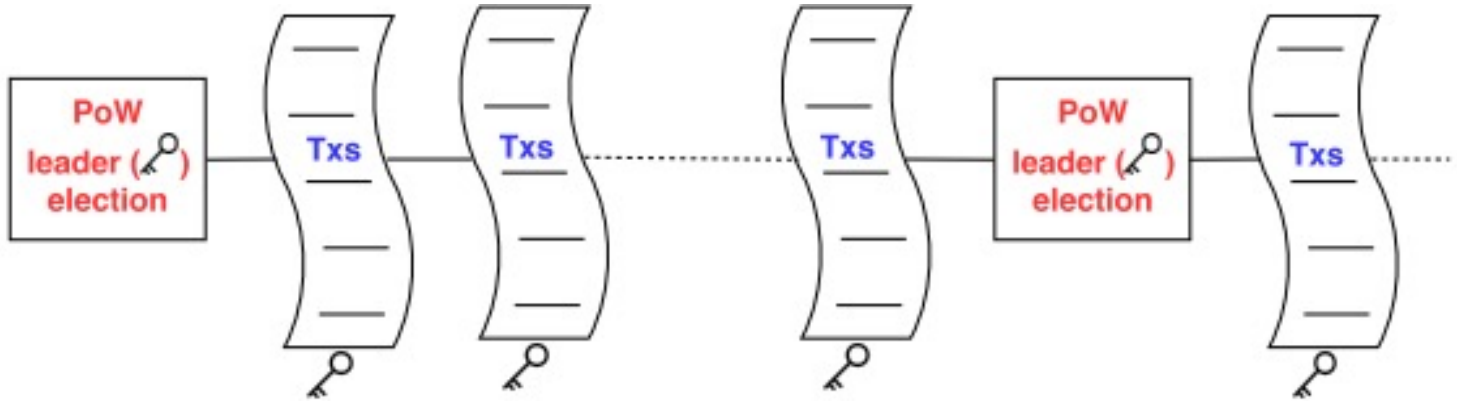
- In Ethereum, a DAG file is used, but in a completely different way – as a huge data structure which serves to cause a large memory burden for the mining algorithm, thus making it ASIC resistant.

Bitcoin-NG – elective Proof-of-Work with micro-blocks

In October 2015, the IC3 group presented Bitcoin-NG, a.k.a. Bitcoin Next Generation. Its idea is that at each 10 min interval, a leader node is elected by a Proof-of-Work algorithm. This leader can then construct micro-blocks in the next 10 min period, after which another leader is again elected. Hence, we would have 2 types of blocks:

- key-blocks – which would contain only the public key of the elected leader
- micro-blocks – which would contain as many transactions as the leader wants or can successfully package into as many blocks as she seem suitable

Bitcoin-NG is deemed to be compatible with the Bitcoin protocol and it might get another consideration sometime in the future, when less radical upgrades are already safely merged. It did raise a couple of comments, though, esp. with regards to double-spending due to reorganizations and transaction selection in mining pools.



Alternative Consensus Protocols

On replacing the Proof-of-Work

There are mainly two common criticisms of Proof-of-Work*:

1. Energy consumption: To reach an agreement of blockchain's state, Bitcoin's Proof-of-Work consumes a lot of energy to perform mining computation, cumulatively more than some countries (e.g., more than Quatar)**. It would be nice if there was a way to save on these costs, one could ask why do we need physical work to secure a completely digital protocol, can't it be done mathematically, cryptographically? Can we make mining be virtual? Expensive mining leads to optimizations in scale, consequently possible *centralization* in specialized mining centers.

2. Speed of transaction processing: Speed is a controversial limitation of Proof-of-Work because it has strict requirements for the time between blocks, as we have discussed in the previous chapter about GHOST.

* To read a counter argument, see "Nothing is Cheaper than Proof of Work" by Paul Sztorc, written in 2015.

** Bitcoin Energy Consumption Index: https://digiconomist.net/bitcoin-energy-consumption

Big Bang of Concensus Algorithms

Consensus Protocol	Cryptocurrencies
PoW	Bitcoin (2009), Litecoin (2011), Namecoin (2011), Peercoin (2012), Dogecoin (2013), Primecoin (2013), Auroracoin (2014), Mazacoin (2014), Monero(2014), Dash (2014), Titcoin (2014), Verge (2014), Vertcoin (2014), Ethereum (2015), Tether (2015), Zcash (2016), Ethereum Classic (2015), Bitcoin Cash (2017)
dPoW	Komodo (2014)
PoS	Nxt (2013), Gridcoin (2013), Potcoin (2014), Steem (2014), Tezos (2014), Ouroboros (2016), Algorand (2017)
DPoS	EOS (2017)
PoA	Ethereum Kovan (2019)
RP	Ripple (2013)
PoS	Dash (2014)
POI	NEM (2014)
PBFT	Tendermint (2014), Hyperledger Fabric (2015), Diem (2020)
DBFT	NEO (2014)
FBA	Stellar (2014)
PoET	Hyperledger Sawtooth (2015)
PoBr	Slim Coin (2014)
PoC	SpaceMint (2014)

- ***Proof-of-X***
  - Proof of Activity
  - Proof of Burn
  - Proof of Capacity
  - Proof-of-Authority
  - Proof-of-Weight
  - Proof of Reputation
  - Proof of Elapsed Time
  - …..

[1] Oyinloye, D. P., Teh, J. S., Jamil, N., & Alawida, M. (2021). Blockchain Consensus: An Overview of Alternative Protocols. Symmetry, 13(8), 1363.

How Proof-of-Stake works

Proof-of-Stake algorithms try to reach distributed consensus not through brute physical work, but by voting for transactions by *validators*, whose votes are weighted by the size of the stake they have in the system. Thus, securing a valid state of the blockchain is not computationally intensive.

validator's stake = the amount of base currency the validator has made available (locked) for consensus

The more stake a participant has in the system, the more he is concerned to preserve it to be functioning well.

The question then is, how to choose validators? They need to be "good" → honest, performant, live.

There is also a security aspect: directly attacking Proof-of-Stake could be more expensive than Proof-of-Work. If one were to execute a head-on 51% attack on Bitcoin, he would need to buy enough mining hardware to perform it and then pay for the electricity to power it for the duration. With Proof-of-Stake, in general, one would need to acquire more than half of all the coins. Even in the case of the pioneering Peercoin with market capitalization of less than $18M, that is quite an expensive attack. Secondly, an attacker is disincetivized to such an attack as he has the largest stake in the system and would be the biggest loser if it lost its value.

How Proof-of-Stake works

- First Proof-of-Stake implementation was made by **Peercoin** in 2011 by Scott Nadal and Sunny King. To select which node will be the next block generator depends on their *stake in the system*, more precisely: the age and number of coins they hold. Consensus is made for the chain that has the largest cumulative sum of coin ages of all transactions on it.

- ***Peercoin*** uses centralized checkpoints to prevent double-spending of old, accumulated coins. Therefore, reorganizing the chain (changing history) is only permitted on blocks that are newer than last checkpoint.

- ***Nxt*** was introduced in 2013 by bitcointalk user BCNext. Nxt uses weighted randomized selection of block generators, meaning that peers with higher account balance are chosen more likely. Nxt does not use coin age nor checkpoints, instead it allows reorganization only of the last 720 blocks.

„Nothing at stake“ problem

- Here, we focus on one of the most fundamental possible attacks for PoS called „nothing at stake“.

- In Proof-of-Work, block generators (miners) are putting hard work into reaching the consensus, so they make very good effort to mine on top of the longest chain – and on that chain only. With Proof-of-Stake, block generators can build on top of the longest chain and on all other chains at the same time, because it doesn‘t cost them anything besides signing the blocks.
  - They can do that on all possible chains they think that have a slightest chance of becoming the main chain. They can „support“ many chains at the same time, perhaps even some where they are double-spending. An attacker just needs to acquire a large stake in the system – in naive PoS designs they don‘t even need to have a large account at that moment, they can obtain private keys that held a large stake in the past and then construct a chain from then on.

- There are some ways to mitigate this risk, like checkpointing as in Peercoin or hard-coded chain states deeper than some level of blocks, but these mostly just move the problem down the chain. In next slides we will examine some models that are more effective at addressing this problem.

The concept of Delegated Proof-of-Stake

- The idea for Delegated Proof-of-Stake (DPoS) was presented in 2014 by Daniel Larimer. Its first implementation was in Bitshares. **It is a modification of Proof-of-Stake, in which only delegated parties can participate in selection for next block generator**. They are called elected witnesses to stress the fact that they are just validating (witnessing) the transactions in blocks, for which they are paid. They can‘t change transactions whatsoever. If a delegate becomes malicious or simply not performing his duties, he is punished by voting off and the next delegate in line will process new and old, pending transactions in the next time slot.
    - Bitshares is a platform that offers many blockchain services, such as user-issued assets, price-tracking tokens, decentralized crowdfunding, referral rewards, decentralized asset exchange …
- Decision about how many witnesses are necessary and who they will be is made by periodical (daily) voting of all stakeholders: the higher your stake is, the more weight it bears. Blocks are generated by randomized selection from the current panel of witnesses.
- DPOS in Bitshares also has elected delegates, who propose changes to the protocol (fees, block sizes etc.), for which they are not paid.

Pros and cons of DPoS

- Major benefits of DPoS aim to be:
  - **fast confirmations (3s), lower fees and support for light clients**. Since validators are elected in a decentralized manner (every participant can vote), most of real work is delegated to them.
  - DPOS protocols attempt to address the problem of „nothing at stake“ by the election of „good“ delegates. This then becomes a different kind of problem and hints at a fuzzy centralization-decentralization balance.

- Some problems raised with DPOS are:
  - **concentration of block generators** – where only a smaller, limited set of nodes have the possibility of validating transactions (but compare this to large mining pools in Bitcoin),
  - **voter apathy** – if many stakeholders are not interested in voting, there is a risk that only large stakeholders will vote and they will vote for their own benefit.

Potential Attack Vectors

Algorithms we've examined in previous slides are sensitive to different kinds of vulnerabilities, which can be exploited by attacks. In general, we classify them into these types:

- **Short range attack** – attacker waits until merchant accepts his payment as confirmed, then double-spending it by bribing validators to reorganize the chain and exclude his initial payment
- **Long range attack** – attacker with enough power can rebuild the chain as he choses, with the ability to build a blockchain however he desires
- **Coin accumulation** – limited to protocols which rely on the age of coins, where attacker could try to accumulate enough old coins to be able to reach a majority in voting
- **Precomputing attack** – attacker with enough power can make sure he is always chosen as the next block generator, therefore be able to organize the chain at will and collect the fees
- **Denial of service** – flooding nodes so that they can no longer relay transactions
- **Sybil attack** – flooding the network with bad nodes that isolate good ones
- **Selfish mining** – secretly building on a chain and publish it selectively in order to waste competitors resources

Other Interesting Ideas

Consortium Consensus

Permissioned = Governance is provided by a group of memberships controlled by some central entity
This is a well understood problem in Distributed Systems
    More than 700+ protocols out there [AFK+15]

Hyperledger Fabric (IBM's proposal)
    Practical BFT implementation

Tendermind, Juno/Kadena, JPMC Quorum, Iroha, RPCA (Ripple)

HoneyBadgerBFT
    Revisits the practical randomized BFT model
    https://github.com/amiller/HoneyBadgerBFT

Many different implementations out there:
    https://github.com/search?q=BFT

[AFK+15] Aublin, P.L., Guerraoui, R., Knežević, N., Quéma, V. and Vukolić, M., 2015. The next 700 BFT protocols. ACM Transactions on Computer Systems (TOCS), 32(4), pp.1-45.

Consensus in Ripple and Stellar

Ripple and **Stellar** started at the same base, as Stellar was forked from Ripple in 2014. Few months later, due to the problems detected within the consensus logic, Stellar designed a new protocol. Since Ripple has significant traction with financial institutions and Stellar is active in expanding financial services worldwide, especially to those underprivileged, we will briefly examine both of them.

Ripple‘s website currently lists tens of partnering banks and many active integrations. More details can be seen at: https://www.forbes.com/sites/thomassilkjaer/2019/11/06/ripple-surpasses-300

- XRP Ledger (previously Ripple Consensus Ledger) – a „blockchain-like“ distributed ledger with a native currency XRP
- Interledger (a.k.a. Hyperledger Quilt) - a cross-ledger protocol for settlement, it is not about ledgers, but about enabling payments between different types of ledgers
- RippleNet – a global payments network for real-time messaging, clearing and settlement
- xCurrent – the main product of Ripple, a real-time gross settlement system and exchange, this is the software that is operating on the RippleNet, but has no connection to „blockchain“ or XRP ledger or XRP tokens; it is using Interledger to connect to various ledgers of participating financial services entities
- xRapid – competitor to other ledgers which is using XRP tokens as a means to offer low-cost liquidity, it is still under testing, only recently was made compatible with the Interledger protocol
- Xpring Platform provides developer tools, services, and programs to integrate money into your apps - https://xpring.io/

Stellar has its own impressive list of partners, working together to enable micropayments, remittances, mobile payments and social enterprises: https://www.stellar.org/about/directory

Participants in Ripple Consensus

- **There are two types of nodes in Ripple:** tracking nodes that relay transactions and validating nodes, that relay and create ledger snapshots. Validators update the network state by selecting which transactions will be applied to the next ledger sequence.

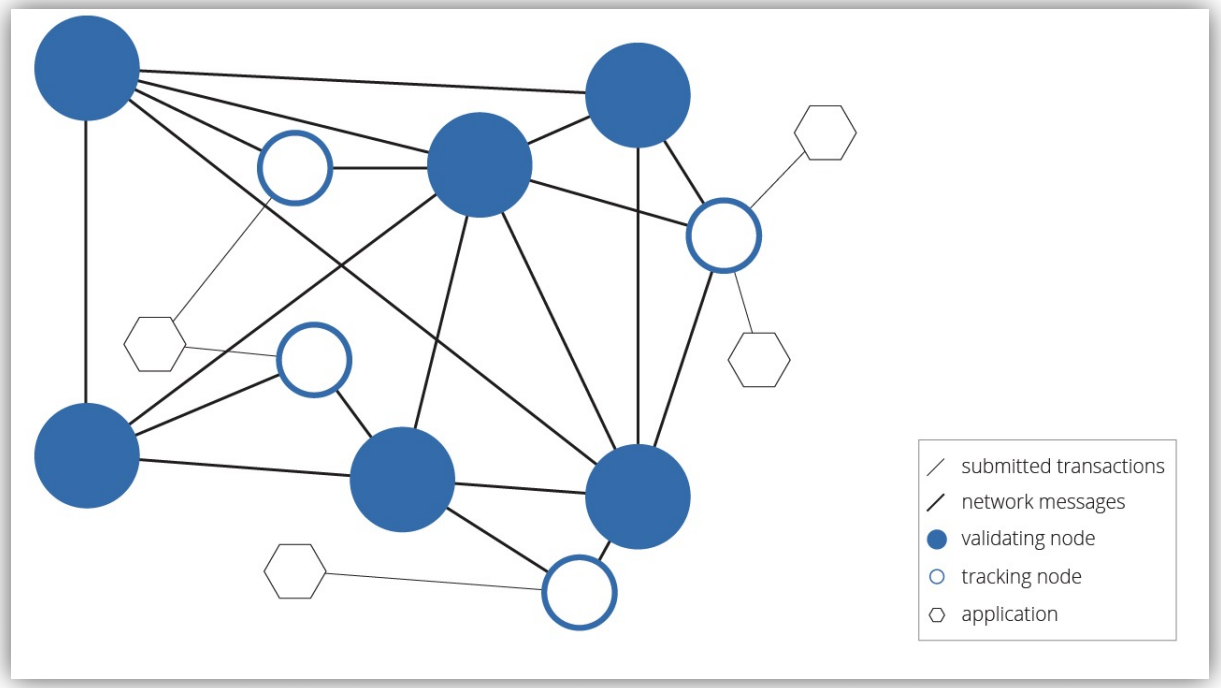


Image by Ripple: https://ripple.com/knowledge_center/the-ripple-ledger-consensus-process/

Ripple Protocol Consensus Algorithm

- When a client application generates a transaction, it sends it to the nearest tracking node, who takes care of forwarding this transaction throughout the whole network. Validating nodes evaluate these candidate transactions and if a transaction gathers enough positive votes from these validators, it is applied to the ledger.
- ***Key issue is having good validators.*** In Ripple, the set of "trusted" validators that each node has, is called **Unique Node List** (UNL). Participants can freely choose custom validators. It is not required to trust each particular validator on the list, but it is expected they will not collude in an attack. Most participants are likely using default UNL provided by Ripple.
- If users would have very different UNLs, there is a possibility of forking, as different validator sets would produce different ledgers. Ripple is stating that UNLs must intersect at least in 20% of validators. Chase et. al [1] points out that this should be at least 40% [1].
- Details on the XRP Ledger Consensus Protocol is available here.

[1] Chase, Brad, and Ethan MacBrough. "Analysis of the XRP ledger consensus protocol." *arXiv preprint arXiv:1802.07242* (2018).

Consensus in Stellar

- After forking issue detected from the initial version from a Ripple fork, Stellar designed and in late 2015 also implemented a new **Stellar Consensus Protocol (SCP).**

- To reach a consensus in SCP, an unanimous agreement among all nodes is not required. Instead, users choose only a set of *neighbors that they trust*, called a **quorum slice**. Different users trust different sets of their peers. They select them by testing them for *liveness* (they are not blocked during the agreement) and *safety* (not contradicting to others). Transactions spread through the network via intersecting quorum slices. Contradicting statements can be blocked, so they don't stop the consensus.

- The **problem** is *how to choose an honest quorum slice*. Stellar hints at openness, social-networks and the fact that countless trust relationships are used in our daily lives already, even when using our computers and internet. Stellar aims to make this transparent and controllable.

- An example of SCP with more description can be read here.

Hashgraph

Swirlds is a startup creating a development platform for distributed applications. Swirlds Hashgraph is one of the newest consensus protocols, though already published in 2016, that aims to be extremely fast and secure, asynchronous (non-deterministic!) BFT consensus. Currently focused on permissioned networks.

„*Gossip about gossip with virtual votes*“

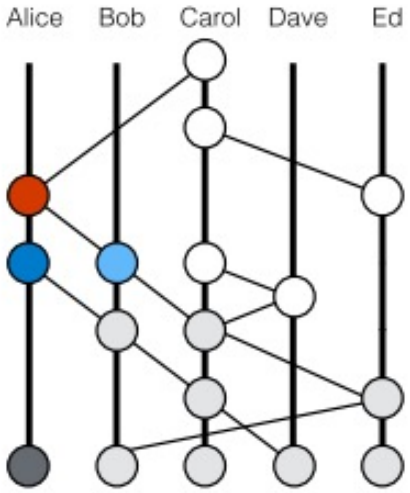


Image source: Hashgraph whitepaper

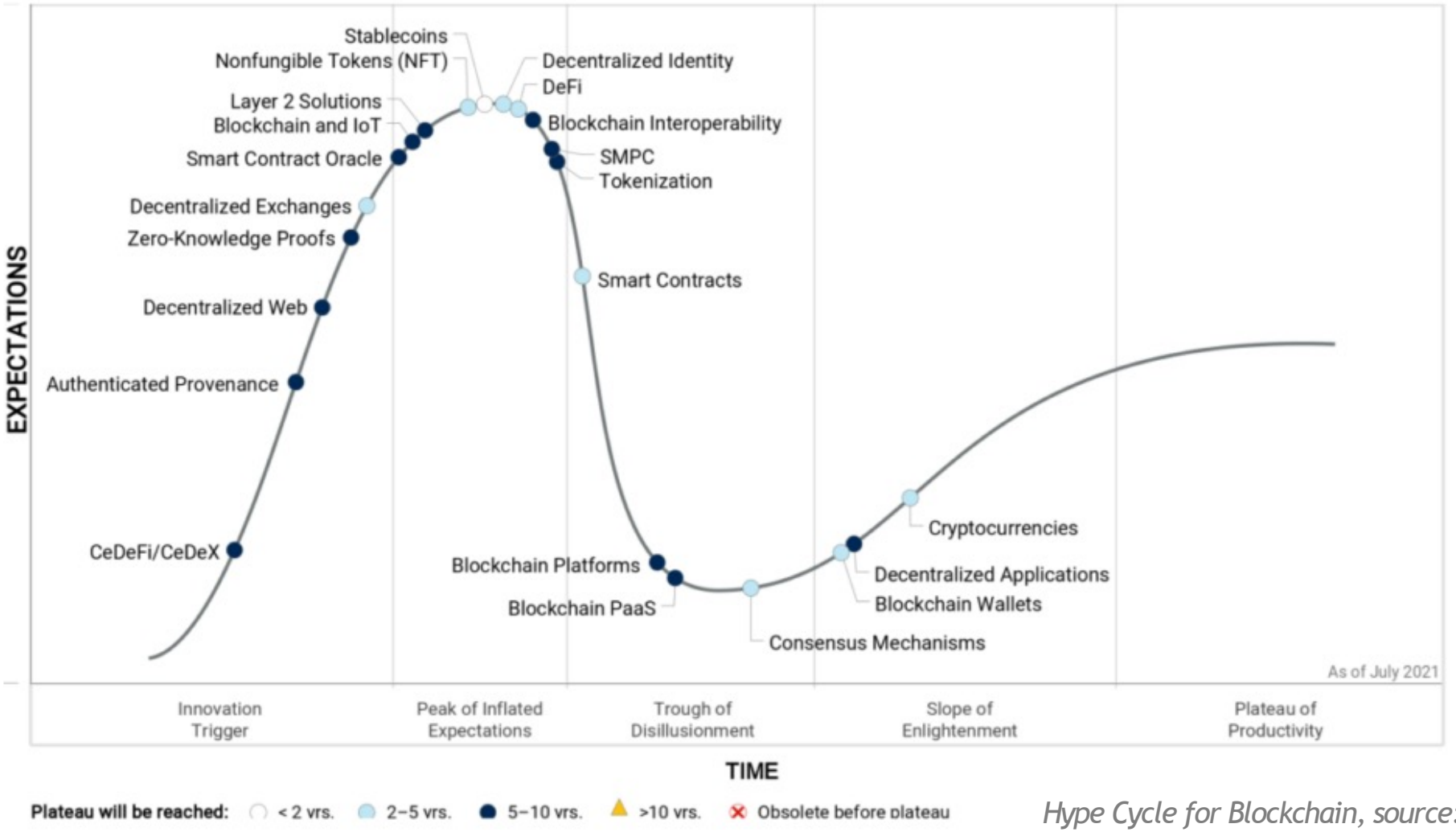
https://hashgraph.com/#whitepaper – official documentation
https://www.youtube.com/watch?v=IVImwsleu6c – a 17min video explanation of hashgraph by its author, Leemon Baird

Finally, PoS for Ethereum

- The Merge is part of what was formerly known as “ETH 2.0,” a series of improvements that restructure the fundamental aspects of the Ethereum network.
- ”The Merge” is considered a major milestone to Ethereum’s multi-part upgrade, to improve scalability, security and energy efficiency. It will accomplish this through sharding and rollup updates to improve scalability.

Blockchain’s Big Bang

The Growth of Blockchain Technology



Hype Cycle for Blockchain, source: Gartner (July 2021)

Types of Blockchains

- Public Blockchains: These types of blockchains are permissionless minning that all users have the same priviladges for using the system and/or participating in the consensus.

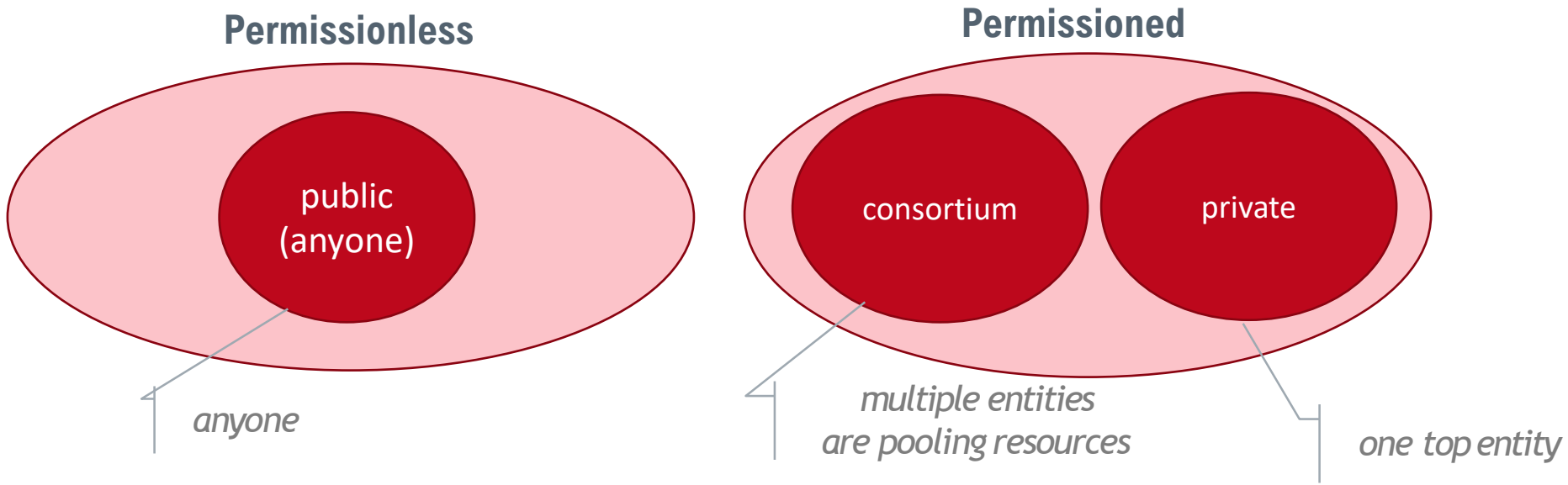
- We are familiar with Bitcoin and Ethereum, but there are countless other blockchains being announced and developed at this very moment. What can we make of it? There are many crypto-currencies, but what about those blockchains that can do even more, like run smart contracts or build with interoperability?
- Most new blockchains in the nearest future are likely to be permissioned, the reason being that public ones usually need incentive mechanisms to attract users and/or miners and it is a tough problem to compete against already established ones with large market caps. In that spirit, how many more public blockchains are needed if the current ones are still not that close to their potential.

Categories of Blockchains

- We started our course by emphasizing the ***openness*** of blockchain in Bitcoin. Its two main properties:
  - *commonly verifiable transactions*, and
  - *impossibility of changing historical data*,

are undergoing substantial evaluations of how they can be used in a closed, ***private*** environment.

- We can cluster blockchains in two main clusters based on their accessibility:



Categories of Blockchains

Property	Permissionless	Permissioned	
		Private Blockchains	Consortium Blockchains
Consensus Participation	Open, all nodes	One organization	A selected group of nodes
Consensus Algorithm	Variations of PoW and PoS	BFT adaptations/ variations	BFT adaptations/ variations
Centrally Managed	No	Fully	Partially
Read Permissions	Public	Read privileges can be assigned	Read privileges can be assigned
Write Permissions	High	Low	Low
Privacy	Limited	Could be preserved	Could be preserved
Immutability	Nearly impossible to tamper	As long as there is agreement – could be tampered	As long as there is agreement – could be tampered
Throughput	Low	High	High
Scalability	Poor	Good	Good

Categories of Blockchains

- **Public Blockchains**
  - provide the foundation for largest interoperability as they serve as a common hub for all connected services
  - they are censorship resistant, without entities with special access, consequently enjoy increased trust
  - suitable for global deployment, like digital currencies, global computation platforms or data repositories
- **Private Blockchains**
  - lower costs (vs. public), as validation is performed without external threats – minimized validation
  - high adaptability and high throughput as the system is easier to update and optimize
  - suitable for managing data coming from dispersed sources, like branch offices or complex supply chains, where the origin and the transport of data could be compromised
- **Consortium Blockchains**
  - lower costs, higher speed and increased security for: sharing data, communicating and reconciling between organizations in a certain ecosystem as there is one shared database, which is trusted by all participants

Private Blockchains - I

A use-case for blockchain within one company …

- **Who are we protecting our data from?**
  - We trust ourselves? What about database administrators, technicians? – There a several non-blockchain solutions, like real-time data encryption, non-disclosure agreements, etc.
- **What is not trusted?**
  - We trust our infrastructure, devices, sensors? – Hardened devices with signed software can help.
  - BFT mechanism is used in some critical environments, like in Boeing Aircraft Information Management System. If parts of the system do not agree on some data, there must be a mechanism to solve such conflicts. If Boeing would also store historical data from those sensors and sensors would cryptographically sign that data to prove the integrity, then we would have a ledger with signed records and it still wouldn't be a blockchain, unless the ledger was distributed.

Private Blockchains - II

- **If we look at one whole organization as one entity, then what sense is there in using decentralization-oriented technology?** Imagine we have blockchain in a company that produces and sells foobars. What nodes would run the consensus mechanism? A consensus for what?

- However, as soon as we consider multiple entities, then we might need some kind of trust-resolving solution. – No need for blockchain even in that case: we can add new policies, new layers, new protocols to our existing platforms … **but a blockchain-based system can have all these bells and whistles included deep in its architecture and because it is designed with certain features from the ground up, it might indeed be "better" (faster, simpler, cheaper …).**

Consortium Blockchains

As much as it was unimaginable just few years ago, some use-cases for non-public blockchains could indeed be imagined. Consortiums seem quite suitable for using blockchains, as in general **they share the same regulatory requirements and comparative business goals** and aren‘t able to blindly trust each other.

- Members of a consortium share a portion of their business data, but instead of each maintaining their own ledgers, they could all work on the same ledger, distributed among them in a secure way. Indeed, that is why there is so much interest from big players and some large blockchain-centered consortiums are already formed.

- There are some **key challenges** with consortium blockchains, some of which are quite hard:
  - agreement on common technology between (competitive) participants
  - cost of implementation of a radically new architecture
  - performance & scalability must support modern trading systems (or others besides trading)
  - privacy on one hand, transparency on the other

Non-public Blockchains

You might have noticed that R3 and Hyperledger projects talk about distributed ledgers instead of blockchains, as some systems don‘t require any „chained blocks“ and can be considered more general then for “transaction” processing. Here, we can look at a Distributed Ledger Technology from the aspect that companies are now becoming more willing to share their data to an extend never imagined before. Distributed databases have enabled this for a long time, but it seems that blockchain and open-source trend have finally triggered the willingness of people to do so on a larger scale.

- The fact that public blockchains are censorship resistant also means they are a promising tool for providing transparency in operations, e.g., tracking budget spending or preserving historical data with all modifications clearly visible.
- Non-public blockchains have one distinctive property: reaching consensus can be delegated to few nodes, as participants have known identities and validators can be held legally accountable.

There are many approaches to develop a blockchain system – in practice it is often done by taking the open-source code of Ethereum and adapting it, or making a new design from scratch, for instance in Hyperledger‘s projects fabric and Sawtooth Lake. Note that a consensus mechanism is but one element in a particular blockchain system, regardless of how important it actually is. So Hyperledger fabric works perfectly fine with a central node for defining “consensus” or can use a proven BFT algorithm as it will soon be able to.

Consensus according to Blockchain type – Part I

Public vs. private blockchains consideration – this is the distinction of granting access to view transactions that happened in the past, and creating new ones:

- to all willing to participate (public) or
- only to selected users (private).

It is important whether blockchain technology is applied in the open or on a private environment, especially regarding the consensus models chosen. While Proof-of-Work is a reasonable choice for public blockchains where parties are unknown and dishonesty must somehow be punished off-court, it can be considered wasteful in a private deployment where parties are known and operate under traditional legal system.

Decentralization as a security leverage – in this session, we will focus on the security of blockchain protocols regarding their consensus mechanisms, thus we will leave the decentralization aspect of the security model for our later sessions. Note that decentralization has ambiguous meanings and it might better serve our understanding if by decentralization we mean the technological aspect, network topology etc., and then we separately discuss coordination or governance of such technology.

Consensus according to Blockchain type – Part II

- Both permissionless and permissioned protocols have their advantages and disadvantages. For instance, one can operate in an open and hostile environment, but on the expense of poor performance, while the other can be very fast, but must have at least some trusted parties.

- Is there any proof that a consensus is really working as it is supposed to or is it just a “magic geeky thing” that might crash at any moment? – It may come as a surprise, but unlike traditional consensus protocols which have been scrutinized in detail and have a backing of formal proofs, many novel consensus protocols lack in analyses and formal verifications.

- While it may appear for a certain amount of time that some “new consensus on the block” is operating correctly, it also bears the uncertainty hazard and can crash dramatically when unexpected circumstances occur. And of course it did happen already and will happen again.

- Surprisingly, the Nakamoto consensus, that stirred the waters, only got its formal verification in 2017 with the already mentioned Bitcoin Backbone Protocol paper.

Blockchain Evolution – Part I

- **Blockchain 1.0 – Cryptocurrencies:** Blockchain 1.0 was introduced in 2009 and focused on solving the double-spending problem by introducing cryptocurrencies like Bitcoin. Soon after the Bitcoin launch, other cryptocurrencies (alternative coins) were evolved and joined this first wave.
- **Blockchain 2.0 – Smart Contracts:** 2015 marked the beginning of a new era in Blockchain with the introduction of smart contracts and the concept of general-purpose programmable blockchains, of which Ethereum is a prime example. Hyperledger and Corda also support this concept.
- **Blockchain 3.0 – Decentralized Applications (DApps):** In 2017, a new generation of Blockchain applications appeared. Dapps have their backend code running on a Blockchain (a decentralized network of computers) where storage and communications can also be decentralized.

Blockchain Evolution – Part II

- **Blockchain 4.0 – Interoperability and scalability:** Since 2019, considerable work has been done to:
  - (a) improve scalability,
  - (b) increase interoperability among different blockchains,
  - (c) integrate Blockchain applications with existing business solutions and IT infrastructures,
  - (d) integrate Blockchain technology with other advanced technologies (e.g., Artificial Intelligence, Internet of Things).

  All these aims seek to spread and speed up the adoption of Blockchain technology. Additionally, two emerging trends deserve close attention: the discussions and experiments with Central Bank Digital Currencies (CBDCs) and Decentralized Finance (DeFi). The former speaks to the efforts to complement or replace traditional physical money with digital equivalents issued by the central banks of countries or economic areas (e.g., European Central Bank, US Federal Reserve, Bank of China, among others). These currencies aim to retain the advantages of cryptos, like Bitcoin, while providing traditional fiat money security. The latter – DeFI – is a form of finance that does not rely on traditional institutions but instead on Blockchain and smart contracts.

Competitors and Complementors

Competitors and Complementors

What differentiating features have alternative coins and other models?

- To freshen our memories:
  - colored coin protocols are an „add-on“ to bitcoins
  - meta-coins build on top of Bitcoin, such as Counterparty, they have their own native tokens
  - alt-coins are stand-alone implementations of similar idea, but with different consensus algorithms or other parameters, like Litecoin or Monero … or Ethereum
  - sidechains are secondary chains pegged to the parent Bitcoin chain
  - subchains can mean 2 things:
    - meta chains embedded in the Bitcoin blockchain, like custom assets or smart contracts, relying on OP_RETURN transactions
    - a Bitcoin scalability improvement proposal using “weak blocks”, that would rely on intermediary mining of easier-to-find blocks
- Systems without underlying value-tokens are also an option and they could be used for non-payment purposes, as well as for payments (Open-Transactions, Hyperledger, Corda …).
- Lastly, the system could be open and public for everyone, or it could be private and permissioned, which would raise the necessity for identification of its participants.

Competitors and Complementors

What can be replicated ad-hoc and what cannot?

- Technical parts of open-sourced models can be replicated, tweaked, re-designed (e.g., Counterparty implementing a fork of Ethereum). Some changes are simpler (Bitcoin to Litecoin), while others can be very hard, for instance changing Proof-of-Work to other models. Besides technicalities, there are many other aspects to consider:
  - network effects (can they be re-used by allowing seamless protocol integration)
  - investment perspectives
  - potential mining investment
  - developers investing their time
  - active users community
  - new use cases that couldn‘t be covered on the original platform
  - specific market niches, etc.

Competitors and Complementors

- New use cases that are being discovered power the development of many new protocols, take for instance Steemit‘s monetization of online discussions. However while the whole ecosystem is evolving rapidly, Bitcoin is also adding new capabilities, some that are quite fascinating in their design and their potential, like Segregated Witness for increased transaction throughput or Lightning Network, enabling off-chain payment channels. Many competitors simply fail to engage the community enough to gain traction (as is the case with some bitcoin‘s proposals, of course).

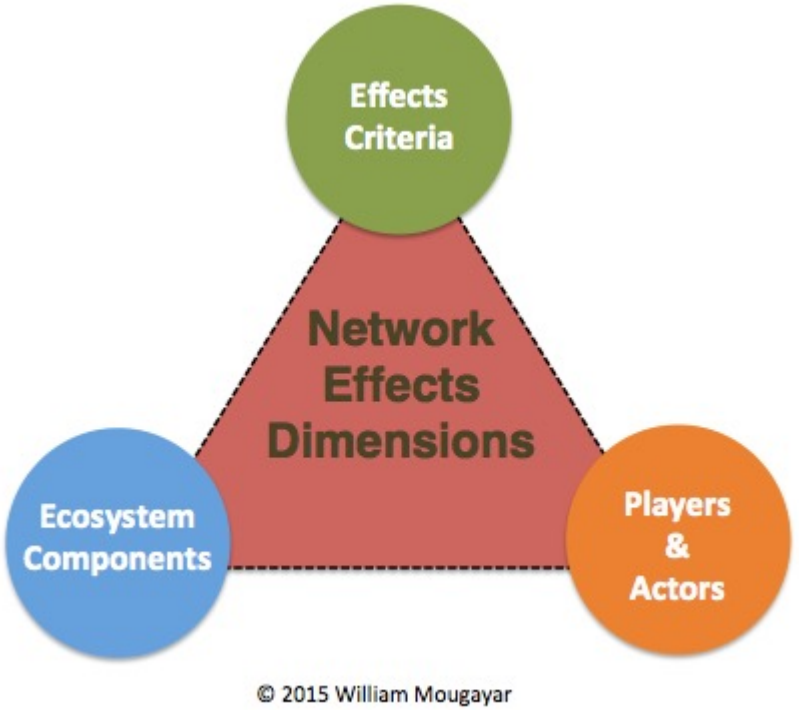
- Another case is that some improvements are simply too unconventional or extreme to be deemed feasible or safe by the actors in the ecosystem, be that the general community, developers, miners or others. Take for instance MimbleWimble or Bitcoin-NG.

- There is only a limited number of researchers and developers that are on top of the nascent ecosystem and fortunately there is more activity coming from academic environment each year (for example, the IC3 group has provided many contributions to Bitcoin and Ethereum ecosystems, there is also a very productive MIT’s Digital Currency Initiative), but there are many others, see this list:
  - https://cdecker.github.io/btcresearch/

Network effects

Which brings us to network effects. Arguably, Bitcoin‘s network effects are one of its strongest pillars. Let‘s take a closer look at them by studying three dimensions as described by William Mougayar.

- We assess network effects of **players** (that is, products or companies) through a set of **criteria** for each of ecosystem‘s **components**.



http://startupmanagement.org/2015/01/04/its-too-early-to-judge-network-effects-in-bitcoin-and-the-blockchain/

Network effects, continued

- We are evaluating a model through effects criteria:
  - **Size:** its size should be significant and scalable
  - **Inter-connectivity:** must exist between parties inside the network
  - **Engaged users:** what percentage of active users uses it often
  - **User experience**: must be original and add new value with usage
  - **Network effects:** the value of the service increases as more people use it
  - **Defensibility:** high switching costs
  - **Monetization:** atomic value units emerge and become the basis for economic activity

- While some alt-coins compete against Bitcoin, Bitcoin itself competes with fiat currencies as well. Hence, while Bitcoin is much larger than Litecoin and US dollar is much larger than Bitcoin, the USD has comparatively more engaged users and its defensibility is a lot higher. On the other hand, not being able to scale in line with increased usage means the size criteria would pose negative network effect.

Network effects, continued

- We test criteria on these ecosystem components:
  - Currency liquidity
  - Consensus engine
  - Blockchain platform services
  - End-user Applications

- Above components are specific to digital currencies and wouldn‘t cover some cases of blockchain use that we will discuss later (like Corda, Juno, Kadena, even Ripple). Still, we can think of how each effect criteria purports to these components.

Network effects, continued

- Players and actors should fulfill every criteria for all components.

- Do they do that already? Most of alt-coins can‘t be used in real life, many are only speculatively traded, some don‘t even have a stand-alone wallet. Many are still in the inception or beta phase, especially consortium distributed ledger technologies. The liquidity on exchanges is getting better, but it is still no match for high stake players.
- Consensus engines come in all shapes and sizes, yet the one that is still humming the least interrupted in the open field is the comparatively primitive Bitcoin’s Proof-of-Work. And the ones that are gaining the most traction in permissioned blockchains, are iterations of years/decades old consensus algorithms, well researched in the distributed computing field.

- Hence, there are still a lot of opportunities for new products.

Conclusions

Conclusions

- We have positions blockchains in the space of distributed systems and replicated state machines.
- We are witnessing a maturation of many consensus approaches and only real-life stress situations will gradually funnel secure and performant winners. There are already few services relying on the security model of the brute Proof-of-Work mechanism, yet there are also many participants in the ecosystem that feel the time for more sophisticated, virtual mining has come.
- An important improvement over Proof-of-Work was presented with GHOST and while it will not be included in Bitcoin, at least not soon, it is already implemented in Ethereum.
- At least initial Proof-of-Stake protocols suffered from nothing at stake problem, which was reduced by partial centralization of developers/founders to achieve immutability at certain chain depths.
- Delegated Proof-of-Stake addresses „nothing at stake“ problem by delegating block generation to perceived trustful participants, that need to maintain their reputation.
- Modifications to GHOST and Delegated Proof-of-Stake are basis for the DECOR protocol, that will be used in Rootstock, an Ethereum smart contract competitor.

Glossary

Glossary

Blockchain consensus: an algorithm that ensures that all peers of a Blockchain network reach to a common acceptance or consensus about the real-time state of our shared distributed ledgers.
Verifiable computation: provides a means to prove the correctness of a computation, such that verifying this proof is computationally less complex that performing the computation itself

Self-Assessment Exercises and Further Readings

Self-Assessment Exercises

- Consider how reaching a consensus might differ when transaction validators are known or unknown. Think about the effort of preserving a reputation.

- With the fairly recent emergence of new distributed consensus mechanisms, there are also attempts to isolate consensus code from other parts of the blockchain platforms – in order to achieve „pluggable“ consensus services which are optimally suited for particular applications. How integral is consensus protocol for a blockchain platform?

You are welcome to share your thoughts on the forums!

Further Reading

- [Casey et al., 2018] The Truth Machine: The Blockchain and the Future of Everything
- [Werbach, 2018] The Blockchain and the New Architecture of Trust. MIT Press
- [Attaran, 2019] Attaran M., Gunasekaran A. (2019) The Evolution of Blockchain. In: Applications of Blockchain Technology in Business. SpringerBriefs in Operations Management. Springer, Cham
- [Juan et al., 2020] Sok: A consensus taxonomy in the blockchain era. In Cryptographers' Track at the RSA Conference, pp. 284-318. Springer, Cham, 2020.
- [Zheng et al., 2017] Zheng, Zibin, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. "An overview of blockchain technology: Architecture, consensus, and future trends." In 2017 IEEE International Congress on Big Data (BigData Congress), pp. 557-564. IEEE, 2017.
- [Mohan, 2018] Mohan, C. "Blockchains and databases: A new era in distributed computing." In 2018 IEEE 34th International Conference on Data Engineering (ICDE), pp. 1739-1740. IEEE, 2018.
- Meneghetti, A., Sala, M., & Taufer, D. (2020). A survey on pow-based consensus. Annals of Emerging Technologies in Computing (AETiC), Print ISSN, 2516-0281.

Further Reading

- SPECTRE: Serialization of Proof-of-Work Events

https://medium.com/@avivzohar/the-spectre-protocol-7dbbebb707b5 (by Aviv Zohar, 2016)

- Confirmation Times in SPECTRE

https://medium.com/@ancapalex/confirmation-times-in-spectre-7f68fec0d997

(by Alexandra Tran, 2018)

- Bach, L. M., Branko Mihaljevic, and Mario Zagar. "Comparative analysis of blockchain consensus algorithms." In 2018 41st
- International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1545-1550. IEEE, 2018. https://ieeexplore.ieee.org/iel7/8392484/8399814/08400278.pdf
- Bach, L. MXiao, Yang, Ning Zhang, Wenjing Lou, and Y. Thomas Hou. "A Survey of Distributed Consensus Protocols for Blockchain Networks." arXiv preprint arXiv:1904.04098 (2019). https://arxiv.org/pdf/1904.04098
- (This is for more tech people - ***optional***) Cachin, C., Guerraoui, R. and Rodrigues, L., 2011. Introduction to reliable and secure distributed programming. Springer Science & Business Media. https://www.springer.com/gp/book/9783642152597

UNIVERSITY *of* NICOSIA

Questions?

Contact Us:

Twitter: @mscdigital
Instructor's Email: christodoulou.kl@unic.ac.cy

Course Support:
Mark Wigmans - wigmans.m@unic.ac.cy
Marios Touloupos - touloupos.m@unic.ac.cy

IT & live session support: dl.it@unic.ac.cy