BLOCKCHAIN
ARCHITECTURE, DESIGN AND USE CASES

Dr. Sudeep Tanwar
Professor, CSE

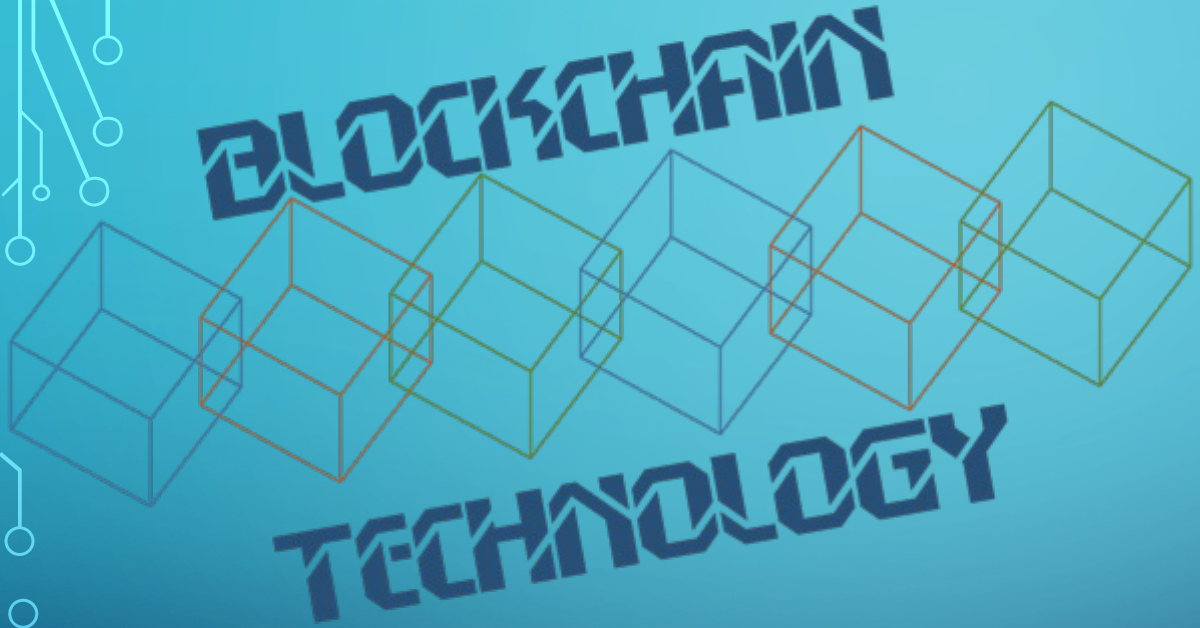


Image courtesy: http://beetfusion.com/

THE ARCHITECTURAL PRINCIPLES

THE BLOCKCHAIN

- A Platform for executing **transactional** services.
- Blockchain is a **public ledger** of a timestamped, ordered, and immutable list of all transactions on the Bitcoin network. Each block is identified by a hash in the chain and is linked to its previous block by referencing the previous block's hash.
- Spanned over multiple organizations or individuals who may not **trust** each other.
- An append-only shared ledger of **digitally signed and encrypted** transactions replicated across a network of peer nodes.

THE BLOCK IN A BLOCKCHAIN – SECURING DATA CRYPTOGRAPHICALLY

- **What is there inside the Block. How individual Blocks are getting connected.**
- Digitally signed and encrypted transactions **verified by the peers**, which means the participants can only view what is inside the Block.
- **Cryptographic security** – Ensures that participants can only view information on the ledger that they are authorized to see.

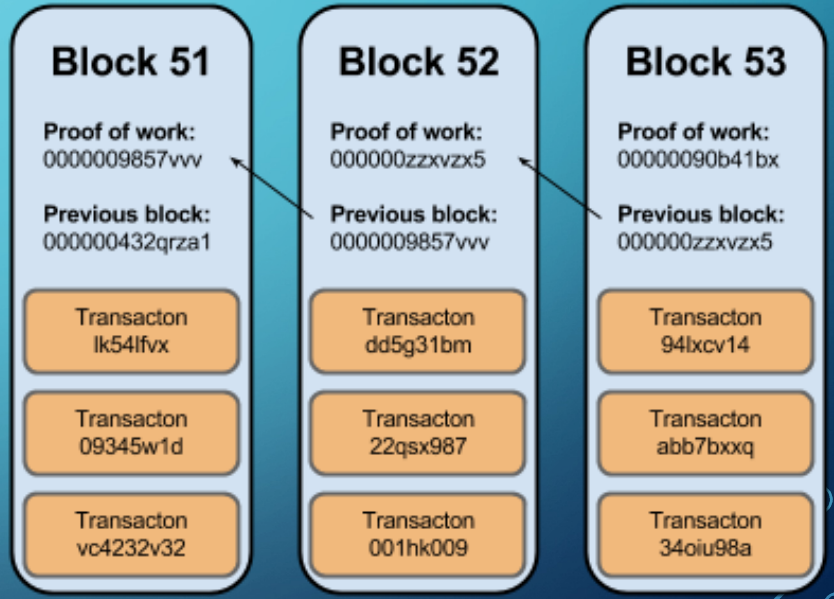


Image source: http://dataconomy.com/

STRUCTURE OF A BLOCK

- A block is a **container data structure** that contains a series of transactions.
- **In Bitcoin:** A block may contain more than **2000** transactions on average, the average size of a block is around **1 MB** (*an upper bound proposed by Satoshi Nakamoto in 2010*)
  - May grow up to **8 MB** or sometime higher
  - Larger blocks can help in processing large number of transactions in one go, which is one of the advantage of having larger size block, but it has some challenges also-We will discuss these when we discuss consensus mechanism

STRUCTURE OF A BLOCK

Size	Field	Description
4 bytes	Block Size	The size of the block, in bytes, following this field
80 bytes	Block Header	Several fields form the block header
1-9 bytes (VarInt)	Transaction Counter	How many transactions follow
Variable	Transactions	The transactions recorded in this block

STRUCTURE OF A BLOCK HEADER

Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The proof-of-work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the proof-of-work algorithm

STRUCTURE OF A BLOCK HEADER

- As shown in Tables, blockchain is a chain of blocks where each block is linked to its previous block by referencing the previous block header's hash.
- This linking makes sure that no transaction can be modified unless the block that records it and all blocks that follow it are also modified.
- The first block is not linked to any previous block and is known as the genesis block.

GENESIS BLOCK

- First block in the Bitcoin blockchain.
- Hardcoded in the bitcoin core software of the applications that utilize its blockchain.
- In 2009, Bitcoin's named developer, Satoshi Nakamoto, created the Genesis Block, which launched the process of Bitcoin trading that's in place today.
- Also, known as Block 0.

MORE ABOUT BLOCK

- Stale blocks are created when a block is solved and every other miner who is still working to find a solution to the hash puzzle is working on that block.
- Orphan blocks, often referred to as stale blocks, are blocks that are not accepted into the blockchain network due to a time lag in the acceptance of the block. Orphan blocks are valid and verified blocks but have been rejected by the chain.
- Orphan blocks are also called detached blocks and were accepted at one point in time by the network as valid blocks but were rejected when a proven longer chain was created that did not include this initially accepted block.

NETWORK FORKS

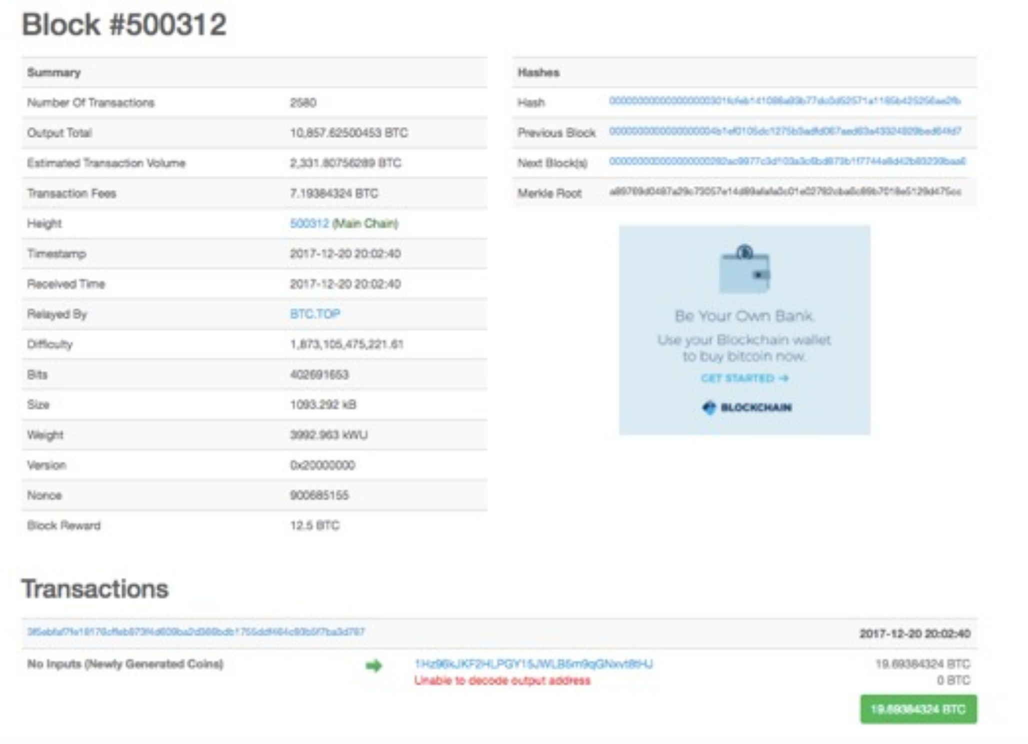
- Because of the distributed nature of blockchain, network forks.
- In cases, where two nodes simultaneously announce a valid block can result in a situation where there are two blockchains with different transactions.
- This is an undesirable situation but can be addressed only by accepting the longest chain.
- In this case, the smaller chain will be considered orphaned.
- If an adversary manages to gain 51% control of the network hash rate (computational power), then they can impose their own version of transaction history.

NETWORK FORKS: **SOFT AND HARD FORKS**

- In soft fork, a client which chooses **not to upgrade to the latest version** supporting the updated protocol will still be able to work and operate normally.
- In this case, previous and new blocks are both acceptable, thus making soft fork backwards compatible.
- In soft fork, **only miners** are required to upgrade to the new client software in order to make use of the new protocol rules.
- Hard fork invalidates previously valid blocks and **requires all users to upgrade.**
- For example, new transaction types are sometimes added as a soft fork, and any changes such as block structure change or major protocol changes results in a hard fork.

STRUCTURE OF A BLOCK (REFERENCE: BITCOIN)

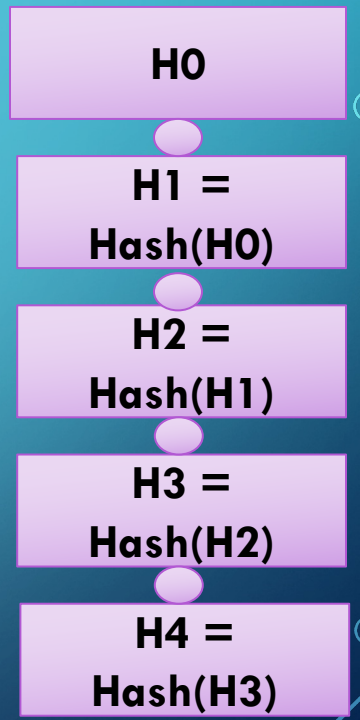
- Two components:
  - **Block Header**
  - **List of Transactions**
- Block height is the number of blocks before a particular block in the blockchain.
- The current height of the blockchain is 766,171 blocks.
- The current size of the bitcoin blockchain as of July 2023, stands at approx. 451 GB.



Block Source: https://blockchain.info/

BLOCK HEADER (REFERENCE: BITCOIN)

- Metadata about a block – (1) Previous block hash, (2) Mining statistics used to construct the block, (3) Merkle tree root (which store all the transactions or hash values)
- **Previous block hash:** Every block inherits from the previous block – we use previous block's hash to create the new block's hash – make the blockchain **tamper proof.**

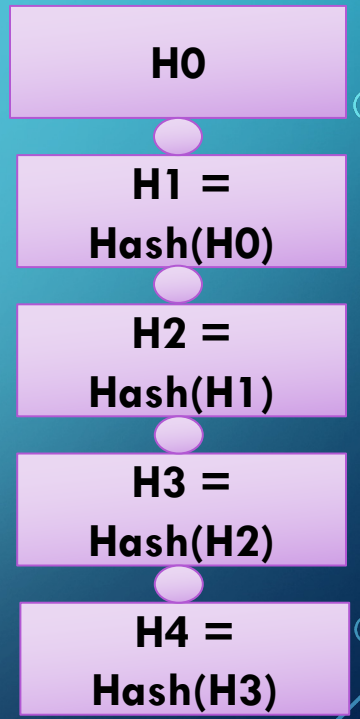


BLOCK HEADER (REFERENCE: BITCOIN)

- **Mining** – the mechanism to generate the hash
  - The mechanism needs to be complicated enough, to make the blockchain **tamper proof**
  - **In Bitcoin Mining, the hash function looks like:**

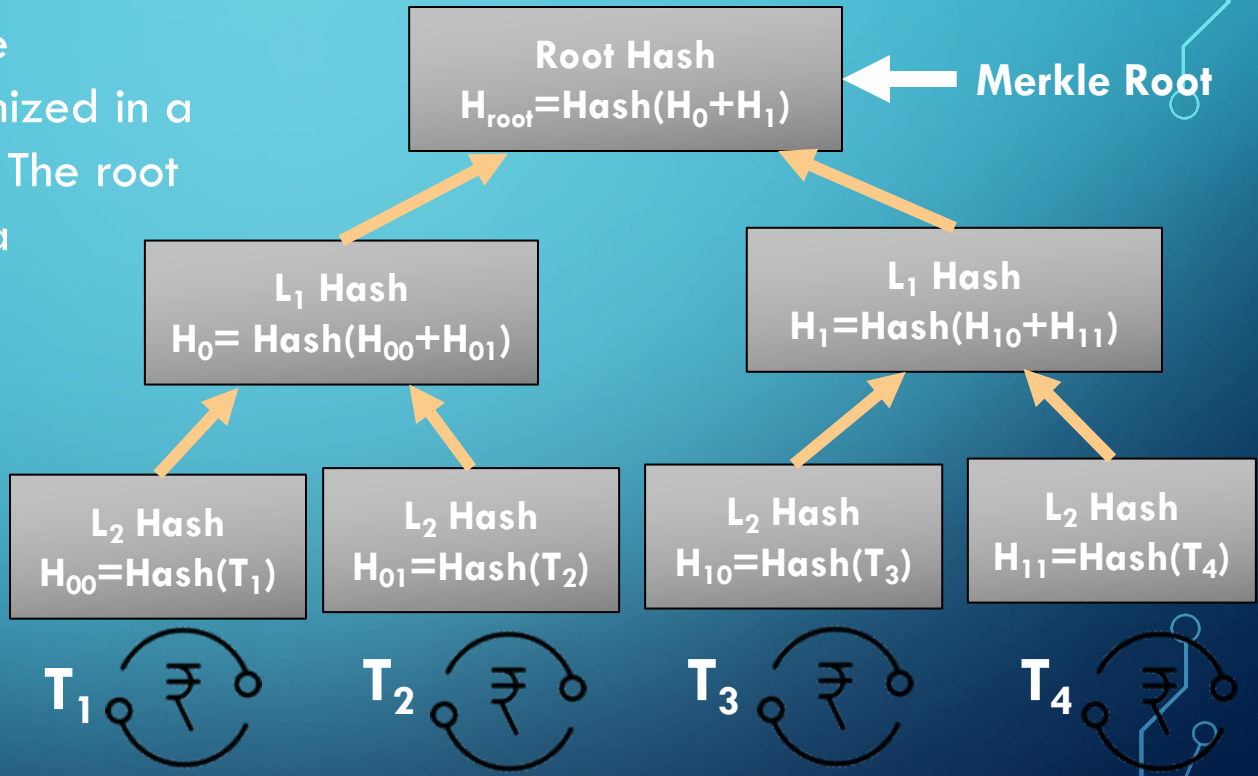
  $H_k = Hash(H_{k-1} \,||\, T \,||\, Nonce)$

  - T is set of transactions
  - So, the task of the miner is to find the nonce (random value) such that H_k has certain predefined **complexity** (number of zeros at the prefix, means complexity of the mining algorithms)
  - If H_k is known, then we can not find out message but if message is known then we can find out H_k
- The header contains mining statistics – timestamp, nonce and difficulty



BLOCK HEADER (REFERENCE: BITCOIN)

- **Merkle Tree Root:** The transactions are organized in a Merkle Tree structure. The root of the Merkle tree is a verification of all the transactions.
- **L-Leaf nodes**



BLOCK HEADER (REFERENCE: BITCOIN)

Summary	
Number Of Transactions	2580
Output Total	10,857.62500453 BTC
Estimated Transaction Volume	2,331.80756289 BTC
Transaction Fees	7.19384324 BTC
Height	500312 (Main Chain)
Timestamp	2017-12-20 20:02:40
Received Time	2017-12-20 20:02:40
Relayed By	BTC.TOP

Difficulty	1,873,105,475,221.61
Bits	402691653
Size	1093.292 kB
Weight	3992.963 kWU
Version	0x20000000
Nonce	900685155
Block Reward	12.5 BTC

Block Source: https://blockchain.info/

- BTC.TOP is the miner who has generated this block
- **Difficulty** level based on the hash finding/mining algorithm, **Bits:** how many bits are there in the Block, **Weight**: balances size among different blocks, **Nonce:** which is used to generate the Hash value, **Block Reward:** means the total bitcoin which the miner will get after generating this block.

THE HASHES IN A BLOCK HEADER (REFERENCE: BITCOIN)

Hashes	
Hash	00000000000000000301fcfeb141088a93b77dc0d52571a1185b425256ae2fb
Previous Block	00000000000000000004b1ef0105dc1275b3adfd067aed63a43324929bed64fd7
Next Block(s)	00000000000000000282ac9977c3d103a3c6bd873b1f7744e8d42b83239baa6
Merkle Root	a89769d0487a29c73057e14d89afafa0c01e02782cba6c89b7018e5129d475cc

- **Block identifier** – the hash of the current block header (Hash algorithm: Double SHA256)
- **Previous block hash** is used to compute the current block hash
- If attacker making any change in the transaction, then Merkle root gets changed, then corresponding hash value gets changed and finally the next block value gets changed.
- **Ultimately, attacker need to change the entire blockchain, which is not possible**

TRANSACTIONS IN A BLOCK (REFERENCE: BITCOIN)

- Transactions are organized as a Merkle Tree. The Merkle Root is used to construct the block hash
- If you change a transaction, you need to change all the subsequent block hash
- The **difficulty** of the mining algorithm determines the **toughness** of tampering with a block in a blockchain
- If the **difficulty is not very high** then it many happen that by the time a miner will accepts a block, **an attacker can change the hash** of all the blocks (means attacker may be successful)
- So, in this situation, one must ensure that finding the **hash value is as tough as possible** so that attacker can not change the hash value of the entire blockchain.

TRANSACTIONS IN A BLOCK (REFERENCE: BITCOIN)

Block Source: https://blockchain.info/

THE BLOCK IN A BLOCKCHAIN - SUMMARY

- The Block contains two parts – **the header** and **the data (the transactions)**
- The header of a block connects the transactions – any change in any transaction will result in a change at the block header
- The headers of subsequent blocks are connected in a **chain – the entire blockchain needs to be updated if you want to make any change anywhere**

THE BLOCKCHAIN REPLICAS

- Next part is How you will manage the Replica.
- Every peer in a Blockchain network maintains a local copy of the Blockchain.
- **Requirements**
  - All the replicas need to be **updated** with the last mined block
  - All the replicas need to be **consistent** – the copies of the Blockchain at different peers need to be **exactly similar**

Here the notion of Consensus comes into the picture???

THE NOTION OF DISTRIBUTED CONSENSUS

- Ensure that different nodes in the network see the same data nearly at the same point of time.
- All nodes in the network need to agree or give their **consent** on a regular basis, that the data stored by them is the same. **(This algorithm, we call it as Consensus Algorithm (CA))**
- **CA** ensures that there is no single point of failure – the data is decentralized
- The system can provide service even in the presence of failures **until and unless the network get disconnected.**

THE NOTION OF DISTRIBUTED CONSENSUS

- Starting from early 90's a large number of works have been devoted on the development of consensus algorithms over a network
- The basic philosophy is based on message passing – **inform your current state to others so that everyone can match their current state with others in the network and they validate their local data.**
- **You can check that you have the latest data that your peers have.**
- However, this philosophy requires that the participants in the consensus algorithm knows each other. Because you need to find out with which node you need to validate or match your data.

THE NOTION OF DISTRIBUTED CONSENSUS

- Can we achieve consensus **even when the network is arbitrarily large ?,** and **no participant in the network really know all other participants?**
- **This we call as:** An **open network scenario or** the **permission-less protocol** – you do not record your identity while participating in the consensus system
- **For that:** A **challenge-response** based system, where the network **would pose a challenge,** and each node in the network **would attempt to solve the challenge**
- **So here nodes need not to reveal their identity and network giving them the challenge.**
- Traditional message passing algorithms not work directly here because you don't know from which node you validate your data

CHALLENGE-RESPONSE TO PERMISSION-LESS CONSENSUS

- Now interesting part is **"The challenge-response protocol"**: The nodes in the network tries to solve the challenge posed by the network
  - The nodes or the participants do not need to reveal their identity
- The node that can solve the challenge first, would get to dictate what the next set of data or state elements to be added
- This will continue iteratively at different rounds

CHALLENGE-RESPONSE TO PERMISSION-LESS CONSENSUS

- **Design of a good challenge** – ensures that different nodes will win the challenge at different runs.
- This ensures that no node would be able to control the network
- So, in one round one node will solve the challenge and, in another node, will say I am ready to solve this challenge and this block is valid block, add it in the BC.
- This Idea is known as: The Bitcoin **Proof of Work** (PoW) algorithm – **ensures consensus over a permission-less setting based on challenge-response**

THE ECONOMICS BEHIND BLOCKCHAIN CONSENSUS

- The challenge-response requires that every node spend large amount of computational power/time to solve a mathematical challenge in each iteration of consensus.
- **What is the incentive/benefit for nodes/Why they participate in this challenge response mechanism?** Only one (or sometime a very few of them) will win in each round
- **What would be the incentive for others?**
- Then, Digital money comes into the picture

THE ECONOMICS BEHIND BLOCKCHAIN CONSENSUS

- The **Digital Money**
  - Ensures operational efficiency
  - More levels of controlling monetary policy
- 1998: Wei Dai published 'b-money' – an anonymous distributed cash system **(Mother of the concept of Cryptocurrency or BITCOIN)**
- **There is no physical currency and network will generate the currency**
- **Cryptocurrency** – a currency beyond the control of banks and governments
- Network will generate the currency.

THE ECONOMICS BEHIND BLOCKCHAIN CONSENSUS

- The mining ensures that no node has the power to sabotage the network and gain control **(good part of the BTC n/w)**
  - No one can hold the control of the cryptocurrency
- The computational effort expended by the nodes in achieving consensus would be paid for by cryptocurrency generated and managed by the network
- So there is a kind of participating benefits to the miners that if the participate then they will get certain amount of money because they spent some of their computational resources.
- Blockchain ensures that the currency is secure and tamper-proof.

IN SUMMARY

- The Technology behind Blockchain
  - **The Data Structure** – Distributed Ledger
  - **Cryptography and Digital Signatures** – Ensure security and tamper-proof architecture
  - **The Consensus** over a Permission-less Environment
  - **The Economy of the Revenue Model** – Encourages participants to join in the mining procedure

WHY IS BLOCK SIZE IMPORTANT

- *The size of individual blocks on a blockchain can have a potentially large impact on the speed and capacity of the network, but there are always trade-offs.*
- Blocks themselves are batches of transaction data, and the amount of data contained in each block combined with the chain's block generation speed determines the number of transactions per second, or TPS, that the network can handle. Obviously, having a high rate of TPS is more attractive, so developers are always looking for ways to improve this metric.
- **Actual rates vary based upon network conditions**, but Bitcoin currently maxes out around seven TPS, and Ethereum isn't much better at 15 TPS. **For comparison**, Visa can process something around 1,700 TPS, so it is imperative that improvements be made if these networks want to compete as global payment solutions. Because the **TPS rate of a blockchain is deeply tied to the size of each block**, this becomes a major factor in finding a path to mainstream adoption.
- However, simply increasing the size indefinitely is only one way to approach the issue, and there are many different philosophies as how to move forward.

WHAT ARE SOME WAYS BLOCKCHAINS CAN SCALE

- ***Scaling solutions come in two forms****: on-chain and off-chain. Both come with pros and cons, but as of now, there is no agreement as to which is more promising for future growth.*
- **On-chain scaling:** refers to the philosophy of changing something about the blockchain itself to make it faster. **For example**, one approach to scaling includes shrinking the amount of data used in each transaction so that more transactions fit into a block. By altering how the transaction data is handled, this patch to Bitcoin allowed a notable improvement to overall network capacity.
- **Another way to potentially boost the TPS of a network** is to increase the rate of block generation. While this can be helpful up to a point, there are limitations to this method relating to the time it takes to propagate a new block through the network. Basically, you don't want new blocks being created before the previous block was communicated to all (or virtually all) of the nodes on the network, as it can cause issues with consensus.

WHAT ARE SOME WAYS BLOCKCHAINS CAN SCALE?

On-chain scaling

- Then there's a technique called sharding, in which transactions are broken up into "shards," and different nodes only confirm certain shards, effectively performing parallel processing to speed up the system. This can be applied to proof-of-work or proof-of-stake systems and is going to form a major component of Ethereum 2.0. This offers the potential to improve the capacity and speed of the network, and developers are hoping that they will see upward of 100,000 TPS become a reality.
- Sharding increases the chances of a "double-spend" occurring as a result of an attack. The issue here is that it takes notably fewer resources to take over individual shards than it does to perform a traditional 51% attack. This can lead to transactions being confirmed that would otherwise be seen as invalid, such as the same Ether (ETH) being sent to two different addresses.
- Some projects have attempted to improve network speeds by limiting the amount of validating nodes — a very different philosophy from Ethereum's. One example is EOS, which has limited its validators to just 21. These 21 validators are then voted on by token holders in an attempt to keep a fair, distributed form of governance — with mixed results. This has given the network a reported 4,000 TPS,
- To scale a blockchain there is need to increase the size of individual blocks. For example,, the average block on the Bitcoin Cash network is still under 1 MB, the debate on this is as of yet unsettled, and we will explore the issue more thoroughly below.

OFF-CHAIN SCALING

Off-chain scaling

- There are also ways to improve network throughput that don't directly change anything about the blockchain. These are often called "second-layer solutions," as they sit "on top of" the blockchain. One of the most well known of these projects is the Lightning Network for Bitcoin. Basically, Lightning Network nodes can open up "channels" between each other and transact back and forth directly, and only when the channel is closed does the Lightning Network transmit the final tally to be recorded on-chain. These nodes can also be strung together, making a much faster, cheaper payment system that only interacts with the main network a fraction of the time.
- Ethereum, of course, also has solutions along these lines. For one, there is the Raiden Network, designed to be Ethereum's version of the Lightning Network, as well as a more general blockchain product called the Celer Network. These projects implement not only off-chain transactions but also state changes, which allow for the processing of smart contracts. Currently, the biggest drawback with these systems is that they are a work in progress, and there are still bugs and other technical issues that can arise if channels aren't created or closed correctly.
- A similar idea is something called "sidechains." These are basically blockchains that are "branched off" of the main chain, with the ability to move the native asset between them. This means sidechains can be created for specific purposes, which will keep that transaction activity off of the primary network, freeing up the overall bandwidth for things that need to be settled on the main chain. This is implemented for Bitcoin through the Liquid sidechain, and Ethereum's version is known as Plasma. One downside here is that each sidechain itself needs to be secured by nodes, which can lead to issues with trust and security if a user is unaware of who is running them behind the scenes.

WHAT ARE THE ARGUMENTS FOR AND AGAINST INCREASING BLOCK SIZE

- **Larger blocks not only improve capacity and speed but also push down fees**. Detractors are concerned that larger blocks will lead to greater centralization.
- As block size increases, not only can more transactions be confirmed in each block, but also the average transaction fee will drop.
- **Increasing the size of blocks does have some consequences.** As simply buying time and not solving the real issue, and that more sophisticated solutions are necessary. The reason they give for why larger blocks are such a problem is that node operators need to download each new block as it is propagated, which with current hardware is of no major issue if blocks are 1 MB, 4 MB or even 32 MB in size. However, if a blockchain is to be adopted globally, then even this is not enough. Before long, blocks would need to be on the scale of gigabytes, and this could be a roadblock for many. If most average users cannot afford hardware or internet connections capable of handling this, then, presumably, fewer and fewer would do so, leading to increased centralization. As Bitcoin Core developer Gregory Maxwell has stated:

WHAT ARE THE ARGUMENTS FOR AND AGAINST INCREASING BLOCK SIZE

- “There’s an inherent tradeoff between scale and decentralization when you talk about transactions on the network. […] You’d need a lot of bandwidth, on the order of a gigabit connection. It would work. The problem is that it wouldn’t be very decentralized, because who is going to run a node?”
- Ultimately, the ones who decide on these changes to a network are the miners, who can “signal” that they support an upgrade to the network’s protocol. Because many miners are grouped into large pools, which ultimately all signal together, this can potentially be another form of centralization, as these conglomerates have far more say than lone miners ever could. Fortunately, there is more than one way to approach this issue, and not all projects want to see open-ended block sizes. Other developers negate this problem in other, clever ways in the hopes of putting scaling to rest once and for all.

HOW HAVE DIFFERENT PROJECTS APPROACHED THE ISSUE?

- *No single solution has emerged as the best one, and projects are still actively exploring creative versions of all these philosophies in an attempt to make scalable networks.*
- At the time of writing, Bitcoin hasn't natively upgraded the nature of its blocks since the implementation of SegWit. That being said, Lightning Network and sidechain research is still going strong, and many expect some form of it to be what enables everyday purchasing with Bitcoin to become the norm. As mentioned, projects such as Bitcoin Cash have embraced the creation of bigger blocks, and BitcoinSV has taken this further with an upper limit on its blocks of a whopping 2 GB. This has, admittedly, led to an increase in the cost of maintaining a node as well as more frequent issues with orphaned blocks.
- Not all projects are taking the larger block approach, of course. While networks such as Zilliqa have joined Ethereum in looking to sharding as their primary means of creating a scalable platform, Ethereum itself is looking to migrate over to a new proof-of-stake system that is being labeled Casper. On the other hand, the project Cardano has developed a new approach called Hydra, which sees each user generating 10 "heads" and each head acting as a new channel for throughput on the network.