An Introduction to Blockchain

my two bits
arun sharma

How did it come about

How did it come about

Roots in the **Cypherpunks** movement (1980s) - Privacy/Anonymity using Cryptography

How did it come about

Roots in the **Cypherpunks** movement (1980s) - Privacy/Anonymity using Cryptography

Digital Money experiments - ecash (David Chaum), eGold, Liberty Reserve, bit gold (a concept close to bitcoin, by Nick Szabo)

How did it come about

Roots in the **Cypherpunks** movement (1980s) - Privacy/Anonymity using Cryptography

Digital Money experiments - ecash (David Chaum), eGold, Liberty Reserve, bit gold (a concept close to bitcoin, by Nick Szabo)

Financial crisis - 2007-08 - major erosion of trust

How did it come about

Roots in the **Cypherpunks** movement (1980s) - Privacy/Anonymity using Cryptography

Digital Money experiments - ecash (David Chaum), eGold, Liberty Reserve, bit gold (a concept close to bitcoin, by Nick Szabo)

Financial crisis - 2007-08 - major erosion of trust

Oct 31, 2008 - The whitepaper - **Bitcoin: A Peer-to-Peer Electronic Cash System** - announced on metzdowd.com, a cypherpunks mailing list, by **Satoshi Nakamoto (Who?)**

How did it come about

Roots in the **Cypherpunks** movement (1980s) - Privacy/Anonymity using Cryptography

Digital Money experiments - ecash (David Chaum), eGold, Liberty Reserve, bit gold (a concept close to bitcoin, by Nick Szabo)

Financial crisis - 2007-08 - major erosion of trust

Oct 31, 2008 - The whitepaper - **Bitcoin: A Peer-to-Peer Electronic Cash System** - announced on metzdowd.com, a cypherpunks mailing list, by **Satoshi Nakamoto (Who?)**

Jan 3, 2009 - first bitcoin created with the message - "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"

How did it come about

Roots in the **Cypherpunks** movement (1980s) - Privacy/Anonymity using Cryptography

Digital Money experiments - ecash (David Chaum), eGold, Liberty Reserve, bit gold (a concept close to bitcoin, by Nick Szabo)

Financial crisis - 2007-08 - major erosion of trust

Oct 31, 2008 - The whitepaper - **Bitcoin: A Peer-to-Peer Electronic Cash System** - announced on metzdowd.com, a cypherpunks mailing list, by **Satoshi Nakamoto (Who?)**

Jan 3, 2009 - first bitcoin created with the message - "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"

It's all about

TRUST

How did it come about

Roots in the **Cypherpunks** movement (1980s) - Privacy/Anonymity using Cryptography

Digital Money experiments - ecash (David Chaum), eGold, Liberty Reserve, bit gold (a concept close to bitcoin, by Nick Szabo)

Financial crisis - 2007-08 - major erosion of trust

Oct 31, 2008 - The whitepaper - **Bitcoin: A Peer-to-Peer Electronic Cash System** - announced on metzdowd.com, a cypherpunks mailing list, by **Satoshi Nakamoto (Who?)**

Jan 3, 2009 - first bitcoin created with the message - "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"

It's all about

TRUST

or the lack of

How did it come about

Roots in the **Cypherpunks** movement (1980s) - Privacy/Anonymity using Cryptography

Digital Money experiments - ecash (David Chaum), eGold, Liberty Reserve, bit gold (a concept close to bitcoin, by Nick Szabo)

Financial crisis - 2007-08 - major erosion of trust

Oct 31, 2008 - The whitepaper - **Bitcoin: A Peer-to-Peer Electronic Cash System** - announced on metzdowd.com, a cypherpunks mailing list, by **Satoshi Nakamoto (Who?)**

Jan 3, 2009 - first bitcoin created with the message - "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"

It's all about

TRUST

or the lack of

or the need to

The realization

The realization

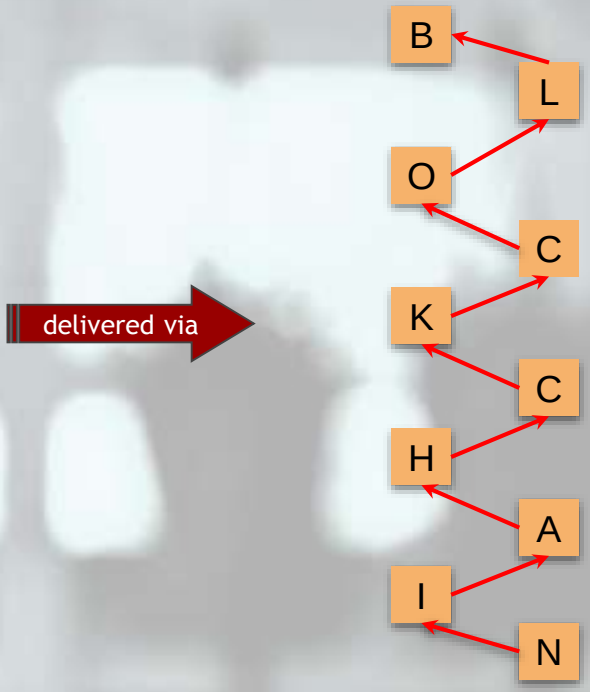
What had the Bitcoin protocol delivered?

- An open, distributed database
- Disintermediation - P2P
- Consensus based, hack proof
- Trustless transfer of value - transactions
- Definitive representation of value
- Insert only, permanent

The realization

What had the Bitcoin protocol delivered?

- An open, distributed database
- Disintermediation - P2P
- Consensus based, hack proof
- Trustless transfer of value - transactions
- Definitive representation of value
- Insert only, permanent

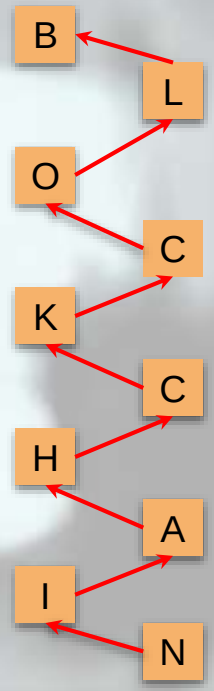


The realization

What had the Bitcoin protocol delivered?

- An open, distributed database
- Disintermediation - P2P
- Consensus based, hack proof
- Trustless transfer of value - transactions
- Definitive representation of value
- Insert only, permanent

delivered via



The Internet of Value

The technology promise

The technology promise

Distributed

Failure resistant
Shared Infrastructure

Decentralized

Censorship resistant
No middlemen
True P2P

Trust(less)

Encrypted
Cryptography based
Cheat proof

Open Source

Hack resistant
Clear rules

Immutable

Auditable
Open ledger
Traceability

Programmable

Platform creation
Business apps
Governance

Accessibility

Permissionless
Permissioned
Pseudonymity
Anonymity

Shared

Incentivized
Game Theory based

The technology promise

Disruptive ?
Transformative ?
Revolutionary ?
Sharing economy 2.0 ?
Internet 3.0 ?

Distributed

Failure resistant
Shared Infrastructure

Decentralized

Censorship resistant
No middlemen
True P2P

Trust(less)

Encrypted
Cryptography based
Cheat proof

Open Source

Hack resistant
Clear rules

Immutable

Auditable
Open ledger
Traceability

Programmable

Platform creation
Business apps
Governance

Accessibility

Permissionless
Permissioned
Pseudonymity
Anonymity

Shared

Incentivized
Game Theory based

The technology promise

Disruptive ?
Transformative ?
Revolutionary ?
Sharing economy 2.0 ?
Internet 3.0 ?

Distributed

Failure resistant
Shared Infrastructure

Decentralized

Censorship resistant
No middlemen
True P2P

Trust(less)

Encrypted
Cryptography based
Cheat proof

Open Source

Hack resistant
Clear rules

Immutable

Auditable
Open ledger
Traceability

Programmable

Platform creation
Business apps
Governance

Accessibility

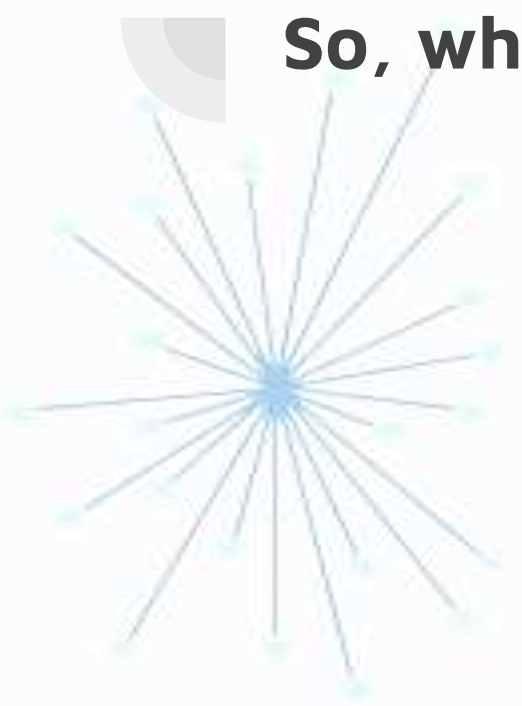
Permissionless
Permissioned
Pseudonymity
Anonymity

Shared

Incentivized
Game Theory based

The Internet of Participation

So, what exactly is it?



Centralized

Decentralized

Distributed

So, what exactly is it?

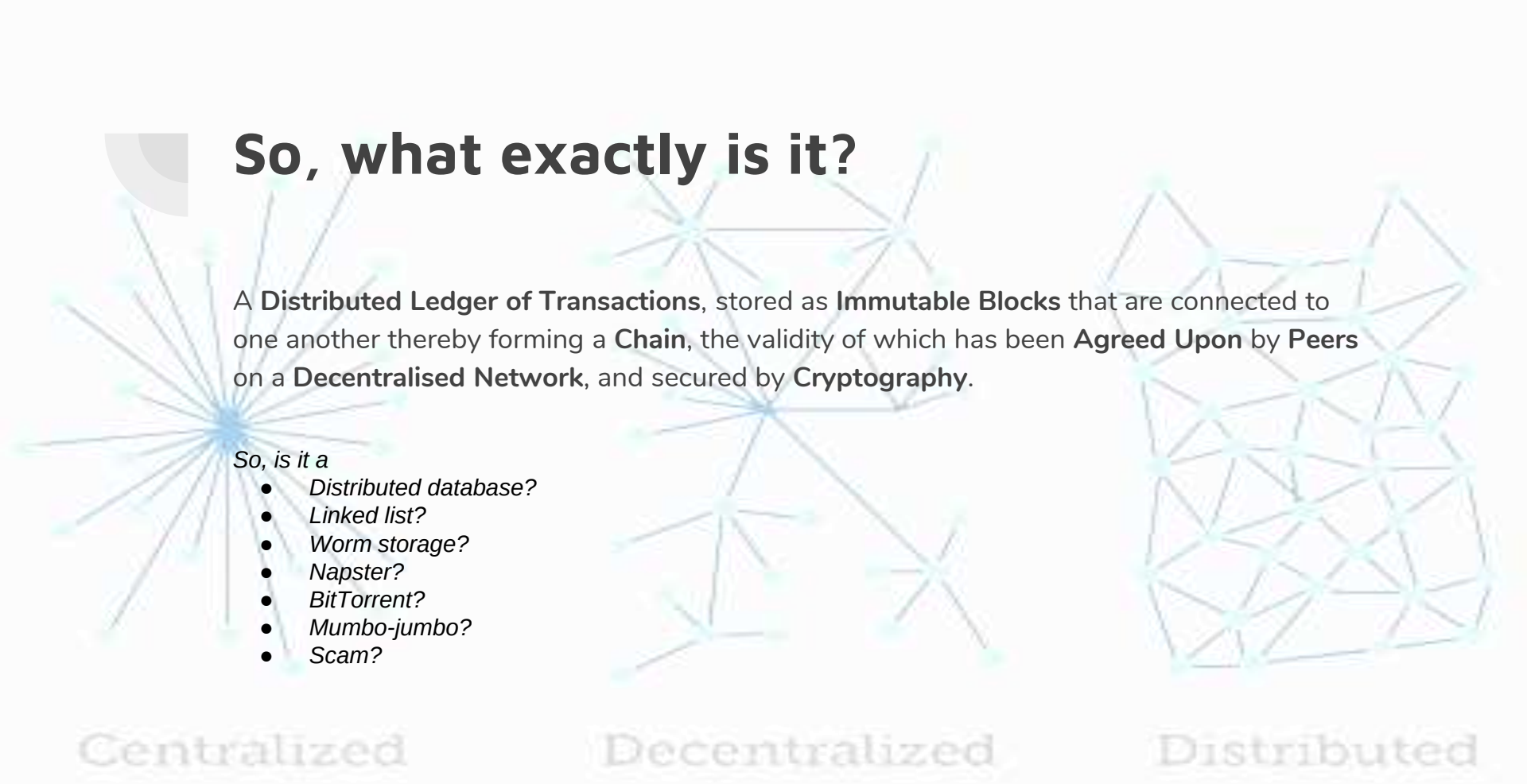
A **Distributed Ledger of Transactions**, stored as **Immutable Blocks** that are connected to one another thereby forming a **Chain**, the validity of which has been **Agreed Upon** by **Peers** on a **Decentralised Network**, and secured by **Cryptography**.

So, what exactly is it?

A **Distributed Ledger of Transactions**, stored as **Immutable Blocks** that are connected to one another thereby forming a **Chain**, the validity of which has been **Agreed Upon** by **Peers** on a **Decentralised Network**, and secured by **Cryptography**.

So, is it a

- *Distributed database?*
- *Linked list?*
- *Worm storage?*
- *Napster?*
- *BitTorrent?*
- *Mumbo-jumbo?*
- *Scam?*



So, what exactly is it?

A **Distributed Ledger of Transactions**, stored as **Immutable Blocks** that are connected to one another thereby forming a **Chain**, the validity of which has been **Agreed Upon** by **Peers** on a **Decentralised Network**, and secured by **Cryptography**.

So, is it a

- *Distributed database?*
- *Linked list?*
- *Worm storage?*
- *Napster?*
- *BitTorrent?*
- *Mumbo-jumbo?*
- *Scam?*

Combines concepts of

- ➔ Distributed Computing +
- ➔ Cryptography (Mathematics) +
- ➔ Database +
- ➔ Application Code +
- ➔ Network +
- ➔ Game Theory +
- ➔ Economics

So, what exactly is it?

A **Distributed Ledger of Transactions**, stored as **Immutable Blocks** that are connected to one another thereby forming a **Chain**, the validity of which has been **Agreed Upon** by **Peers** on a **Decentralised Network**, and secured by **Cryptography**.

So, is it a

- *Distributed database?*
- *Linked list?*
- *Worm storage?*
- *Napster?*
- *BitTorrent?*
- *Mumbo-jumbo?*
- *Scam?*

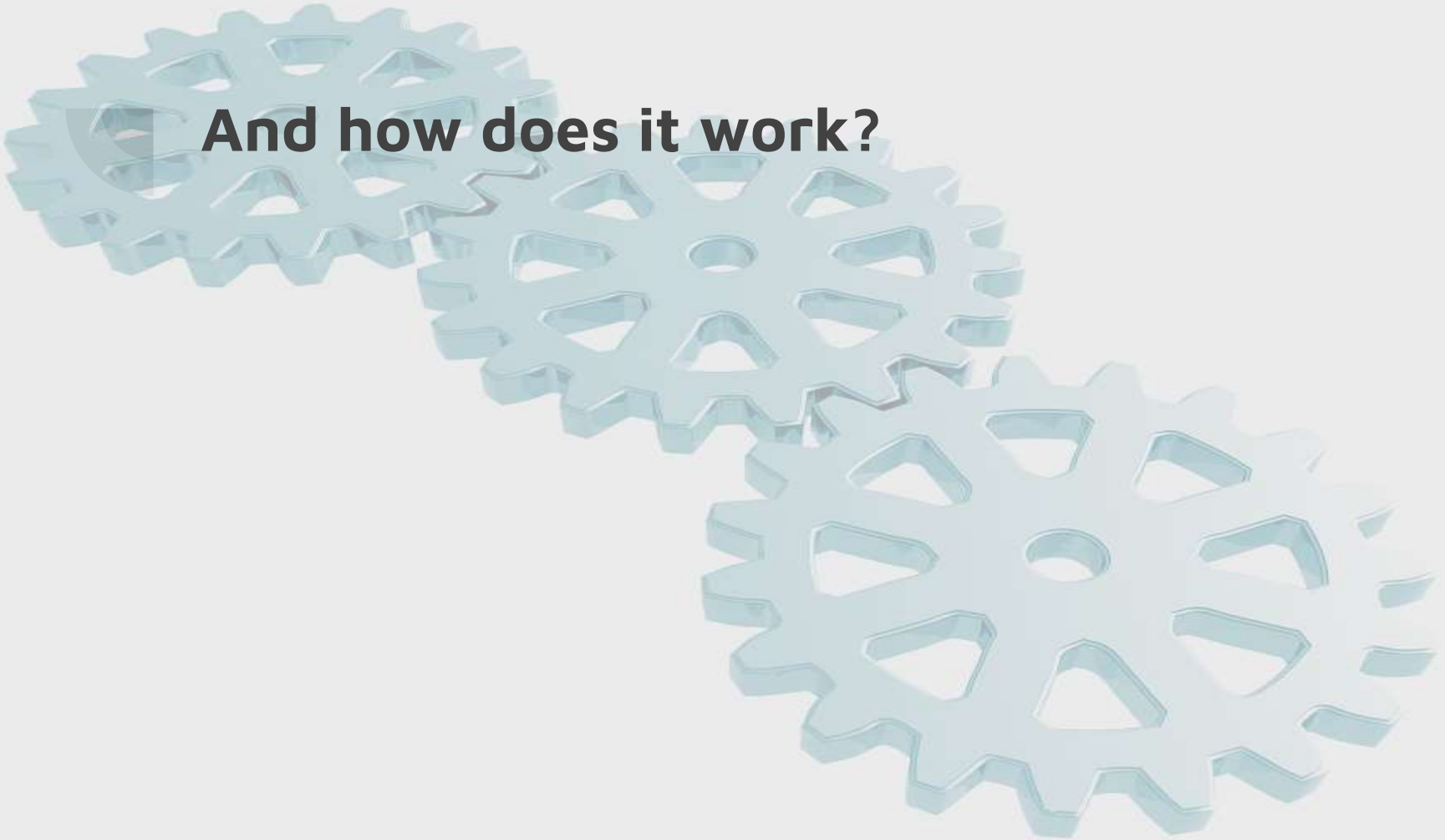
Combines concepts of

- ➔ Distributed Computing +
- ➔ Cryptography (Mathematics) +
- ➔ Database +
- ➔ Application Code +
- ➔ Network +
- ➔ Game Theory +
- ➔ Economics

Centralized Decentralized Distributed

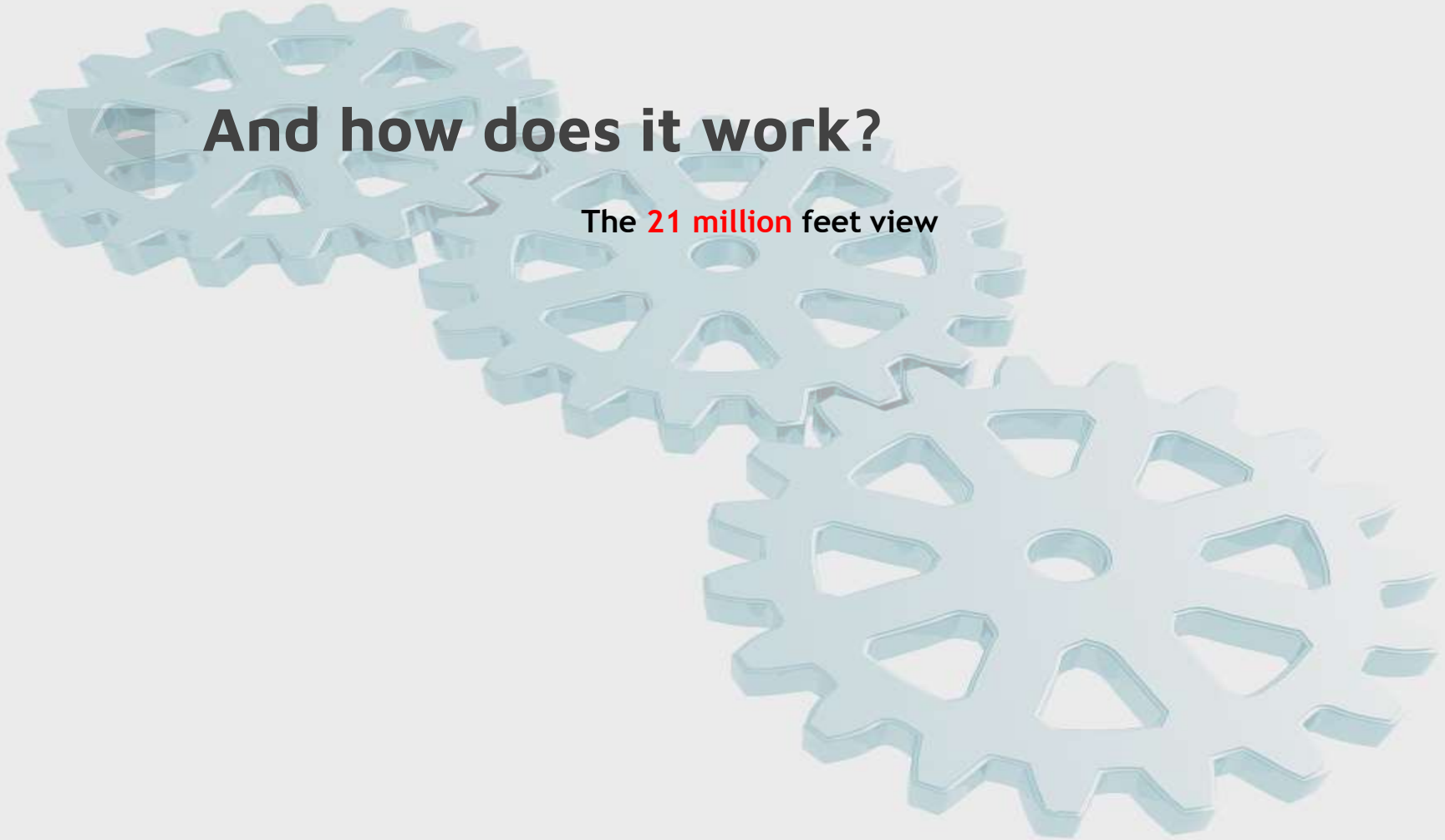
Based on sound principles

And how does it work?



And how does it work?

The 21 million feet view



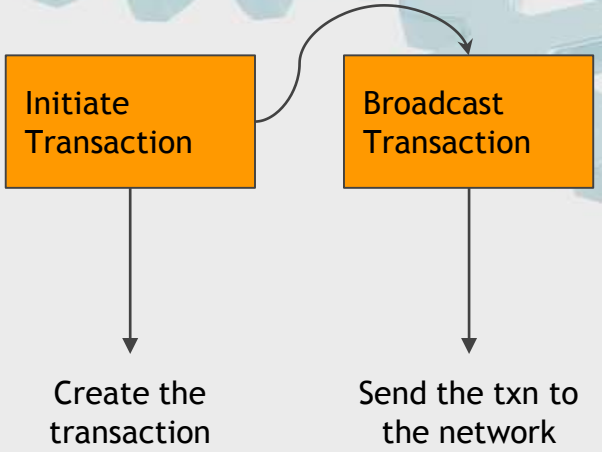
And how does it work?

The 21 million feet view

Create the
transaction

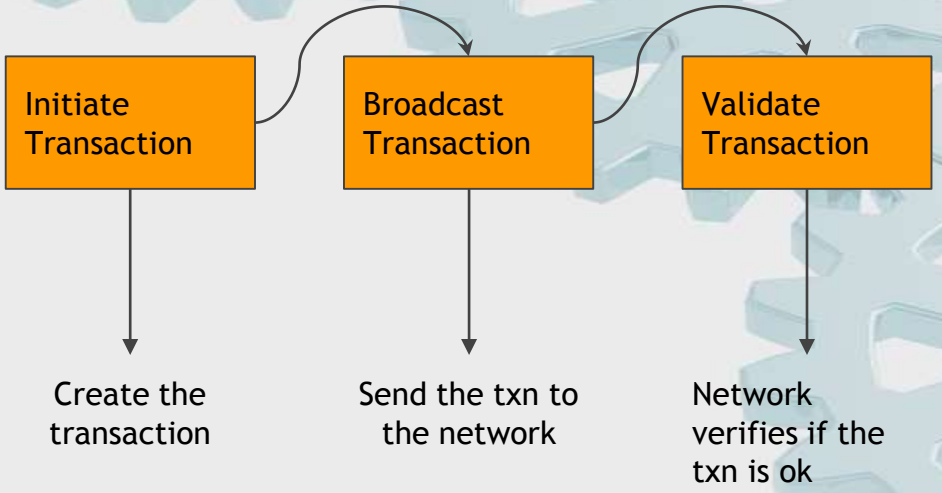
And how does it work?

The 21 million feet view



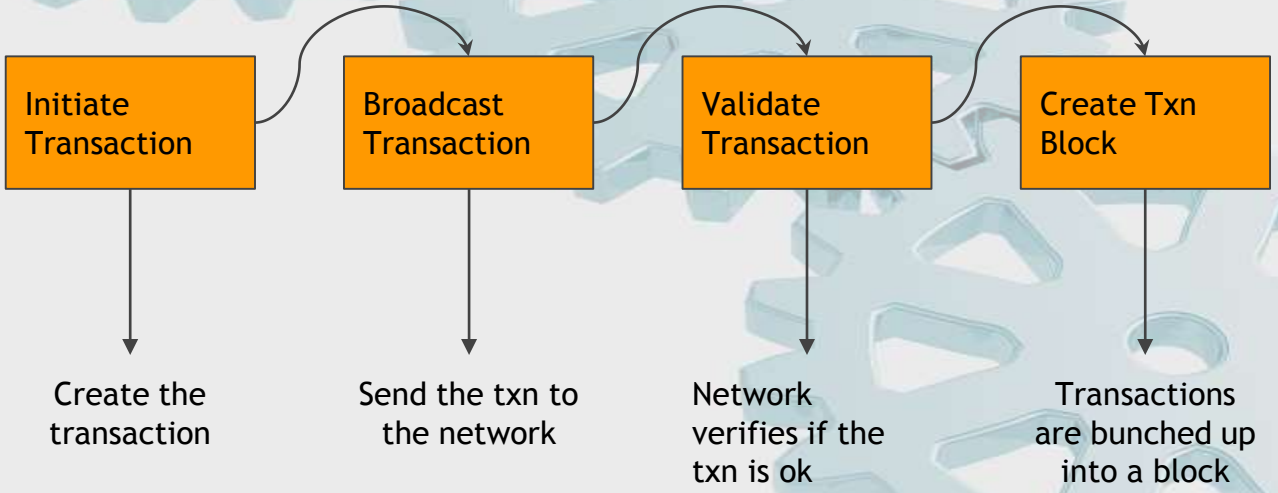
And how does it work?

The 21 million feet view



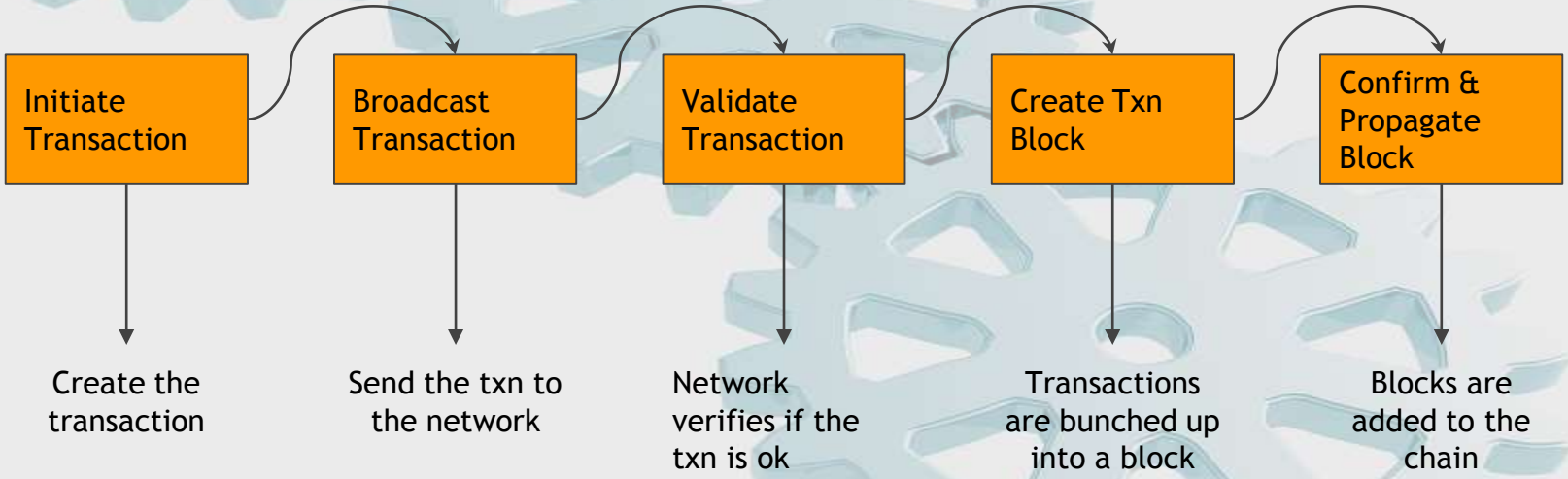
And how does it work?

The 21 million feet view



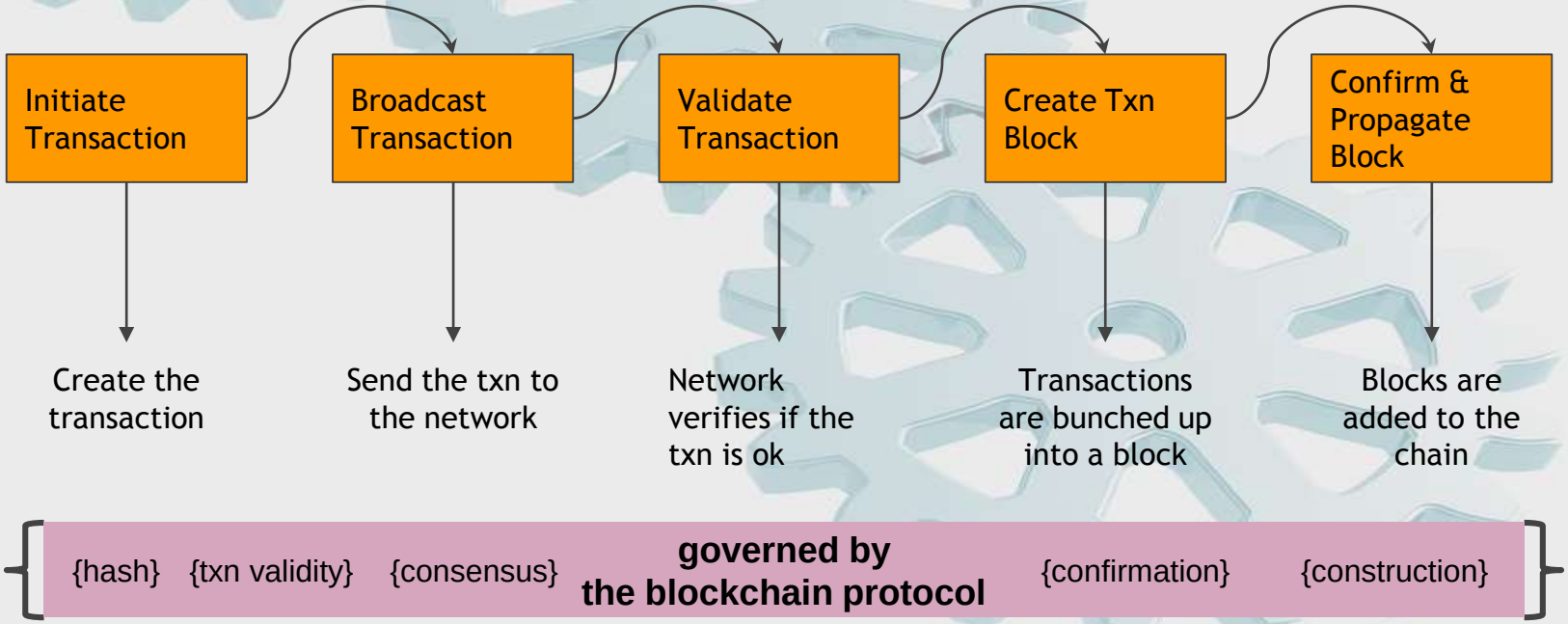
And how does it work?

The 21 million feet view



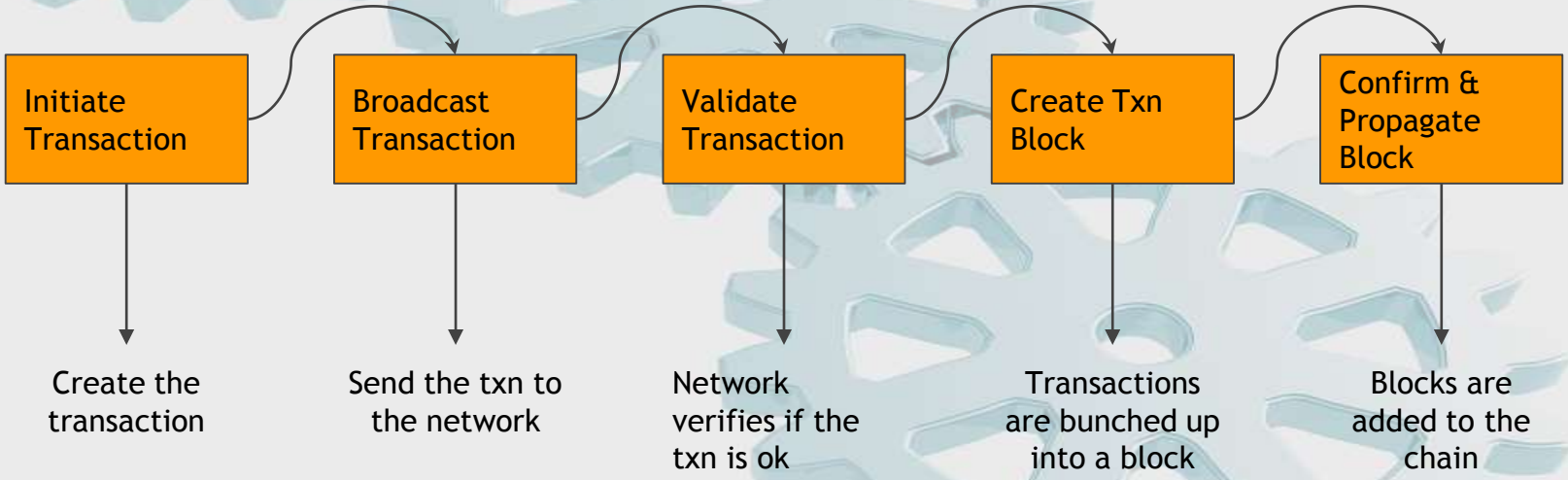
And how does it work?

The 21 million feet view



And how does it work?

The 21 million feet view



{hash} {txn validity} {consensus} **governed by the blockchain protocol** {confirmation} {construction}

Thanks, BBC

Breaking it down, bitcoin by bitcoin

Breaking it down, bitcoin by bitcoin

Alice wants
to transfer 10
BTC to Bob

Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

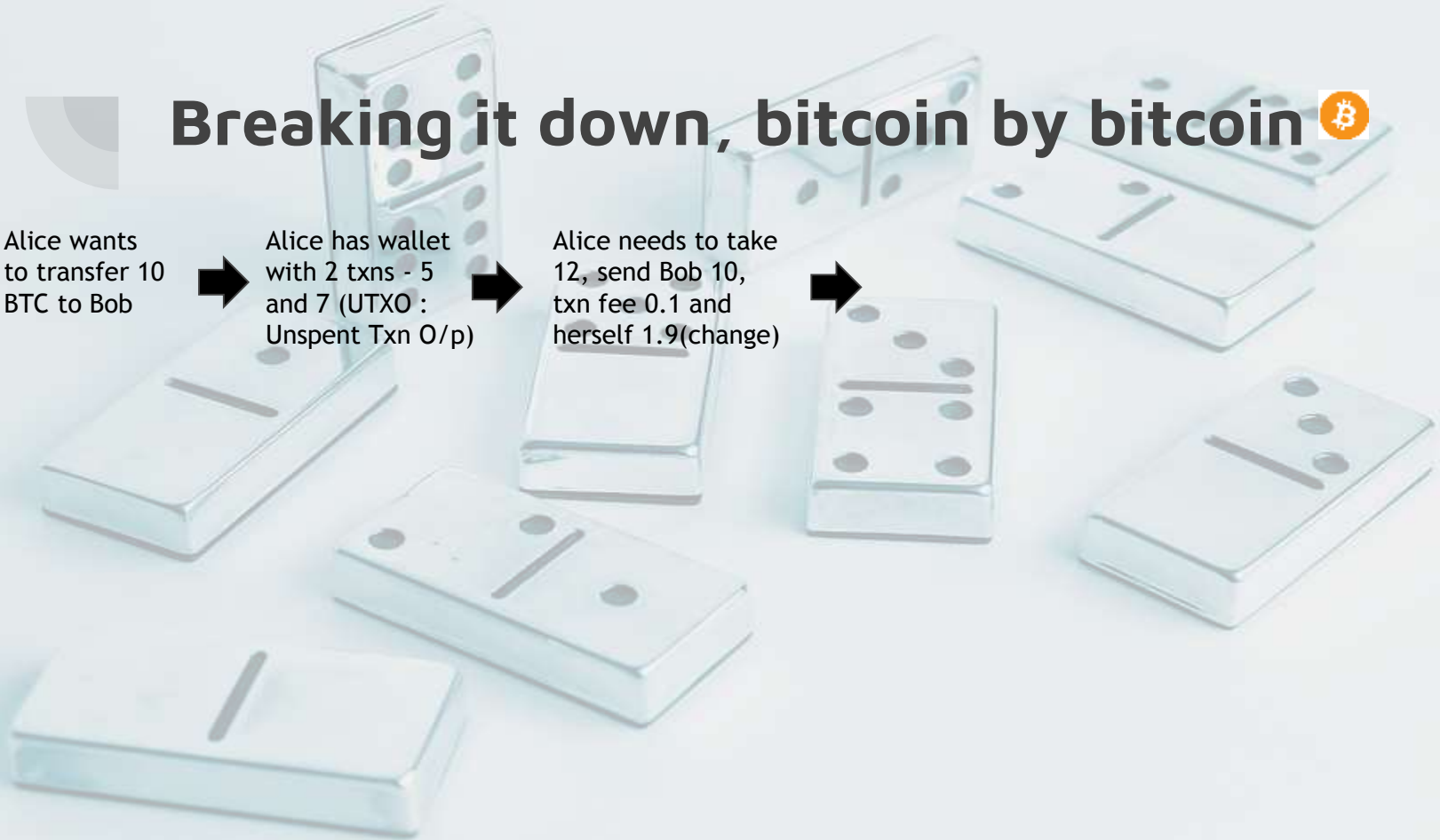
Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)



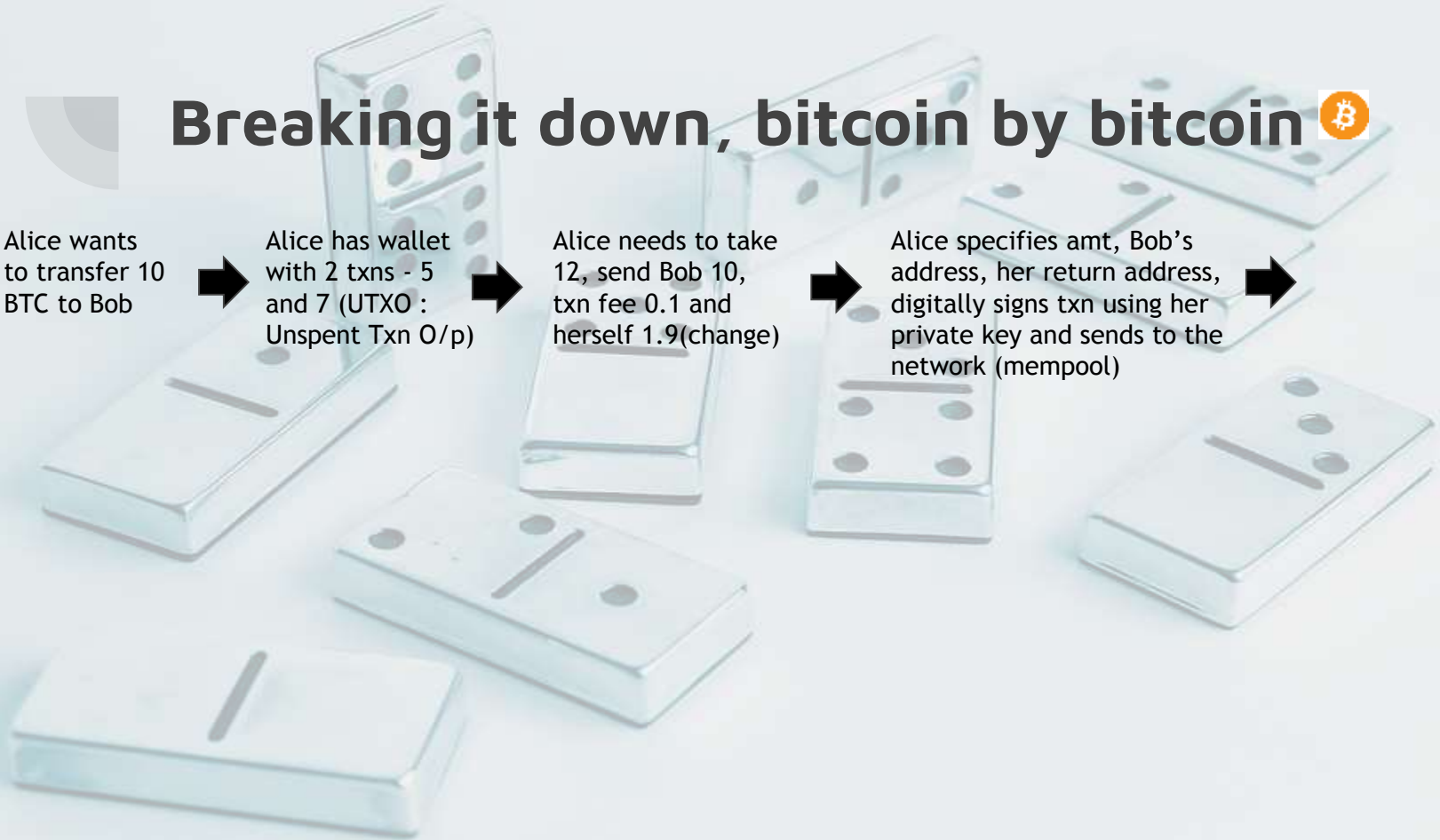
Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)



Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)

Nodes validate-
a)Does Alice have the money?
b)Has she spent it elsewhere?
Remember, they have the history !!!

Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)

Nodes validate-
a)Does Alice have the money?
b)Has she spent it elsewhere?
Remember, they have the history !!!

Special nodes called Miners collect valid transactions, bunch them up into a block and start 'working'. Miners add a 'reward' txn to themselves - this creates new bitcoin

Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)

Nodes validate-
a)Does Alice have the money?
b)Has she spent it elsewhere?
Remember, they have the history !!!

Special nodes called Miners collect valid transactions, bunch them up into a block and start 'working'. Miners add a 'reward' txn to themselves - this creates new bitcoin

Work is to solve a math puzzle - start hashing the info in the blocks until a 'particular' type of hash is got (hash starting with 'x' number of zeroes)
Hash(fixed,variable)=000xyz

Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)

Nodes validate-
a)Does Alice have the money?
b)Has she spent it elsewhere?
Remember, they have the history !!!

Special nodes called Miners collect valid transactions, bunch them up into a block and start 'working'. Miners add a 'reward' txn to themselves - this creates new bitcoin

Work is to solve a math puzzle - start hashing the info in the blocks until a 'particular' type of hash is got (hash starting with 'x' number of zeroes)
Hash(fixed,variable)=000xyz

On success, add the block to own chain, announce the block to the network with the txns and the variable which solved the puzzle

Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)

Nodes validate-
a)Does Alice have the money?
b)Has she spent it elsewhere?
Remember, they have the history !!!

Special nodes called Miners collect valid transactions, bunch them up into a block and start 'working'. Miners add a 'reward' txn to themselves - this creates new bitcoin

Work is to solve a math puzzle - start hashing the info in the blocks until a 'particular' type of hash is got (hash starting with 'x' number of zeroes)
Hash(fixed,variable)=000xyz

On success, add the block to own chain, announce the block to the network with the txns and the variable which solved the puzzle

Others confirm if puzzle is indeed solved, and if satisfied with the work, update their own chains with the new block

Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)

Nodes validate-
a)Does Alice have the money?
b)Has she spent it elsewhere?
Remember, they have the history !!!

Special nodes called Miners collect valid transactions, bunch them up into a block and start 'working'. Miners add a 'reward' txn to themselves - this creates new bitcoin

Work is to solve a math puzzle - start hashing the info in the blocks until a 'particular' type of hash is got (hash starting with 'x' number of zeroes)
Hash(fixed,variable)=000xyz

On success, add the block to own chain, announce the block to the network with the txns and the variable which solved the puzzle

Others confirm if puzzle is indeed solved, and if satisfied with the work, update their own chains with the new block

Once 'x' amount of confirmations have been provided (6), the txn is deemed irreversible (New UTXO: Bob-10, Alice-1.9, Miner claims reward & txn fees)

Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)

Nodes validate-
a)Does Alice have the money?
b)Has she spent it elsewhere?
Remember, they have the history !!!

Create the transaction

Special nodes called Miners collect valid transactions, bunch them up into a block and start 'working'. Miners add a 'reward' txn to themselves - this creates new bitcoin

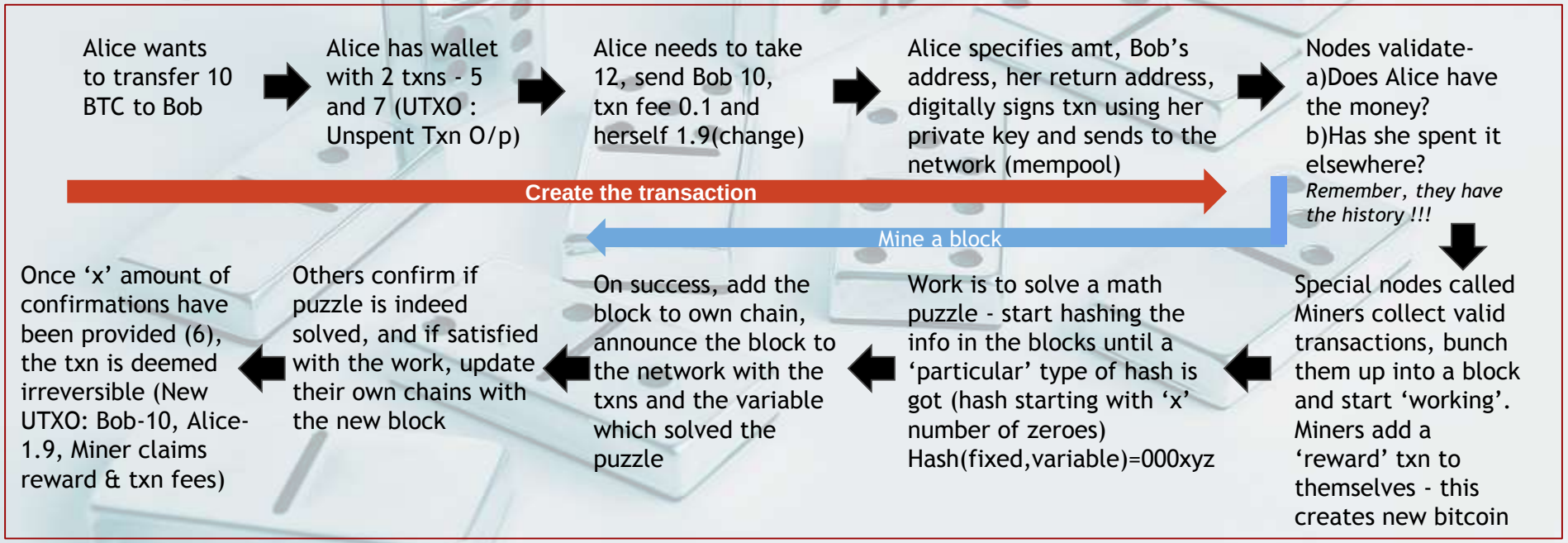
Work is to solve a math puzzle - start hashing the info in the blocks until a 'particular' type of hash is got (hash starting with 'x' number of zeroes)
Hash(fixed,variable)=000xyz

On success, add the block to own chain, announce the block to the network with the txns and the variable which solved the puzzle

Others confirm if puzzle is indeed solved, and if satisfied with the work, update their own chains with the new block

Once 'x' amount of confirmations have been provided (6), the txn is deemed irreversible (New UTXO: Bob-10, Alice-1.9, Miner claims reward & txn fees)

Breaking it down, bitcoin by bitcoin



Breaking it down, bitcoin by bitcoin

Alice wants to transfer 10 BTC to Bob

Alice has wallet with 2 txns - 5 and 7 (UTXO : Unspent Txn O/p)

Alice needs to take 12, send Bob 10, txn fee 0.1 and herself 1.9(change)

Alice specifies amt, Bob's address, her return address, digitally signs txn using her private key and sends to the network (mempool)

Nodes validate-
a)Does Alice have the money?
b)Has she spent it elsewhere?
Remember, they have the history !!!

Create the transaction

Special nodes called Miners collect valid transactions, bunch them up into a block and start 'working'. Miners add a 'reward' txn to themselves - this creates new bitcoin

Mine a block

Work is to solve a math puzzle - start hashing the info in the blocks until a 'particular' type of hash is got (hash starting with 'x' number of zeroes)
Hash(fixed,variable)=000xyz

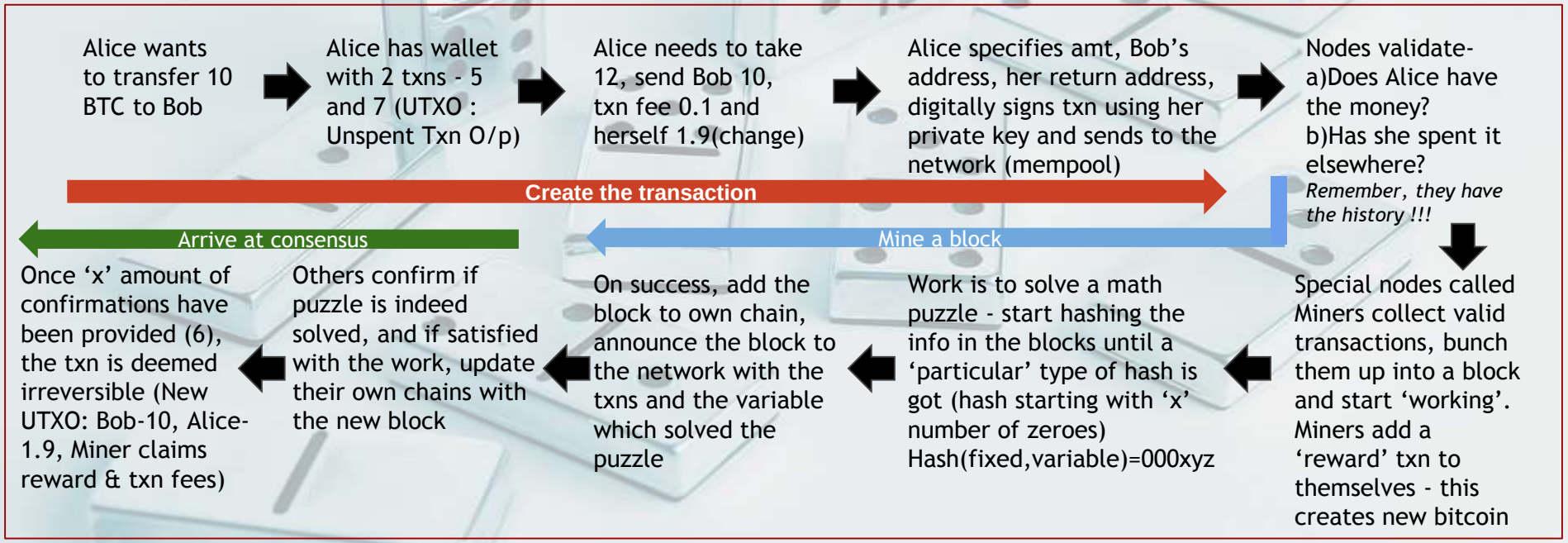
On success, add the block to own chain, announce the block to the network with the txns and the variable which solved the puzzle

Others confirm if puzzle is indeed solved, and if satisfied with the work, update their own chains with the new block

Arrive at consensus

Once 'x' amount of confirmations have been provided (6), the txn is deemed irreversible (New UTXO: Bob-10, Alice-1.9, Miner claims reward & txn fees)

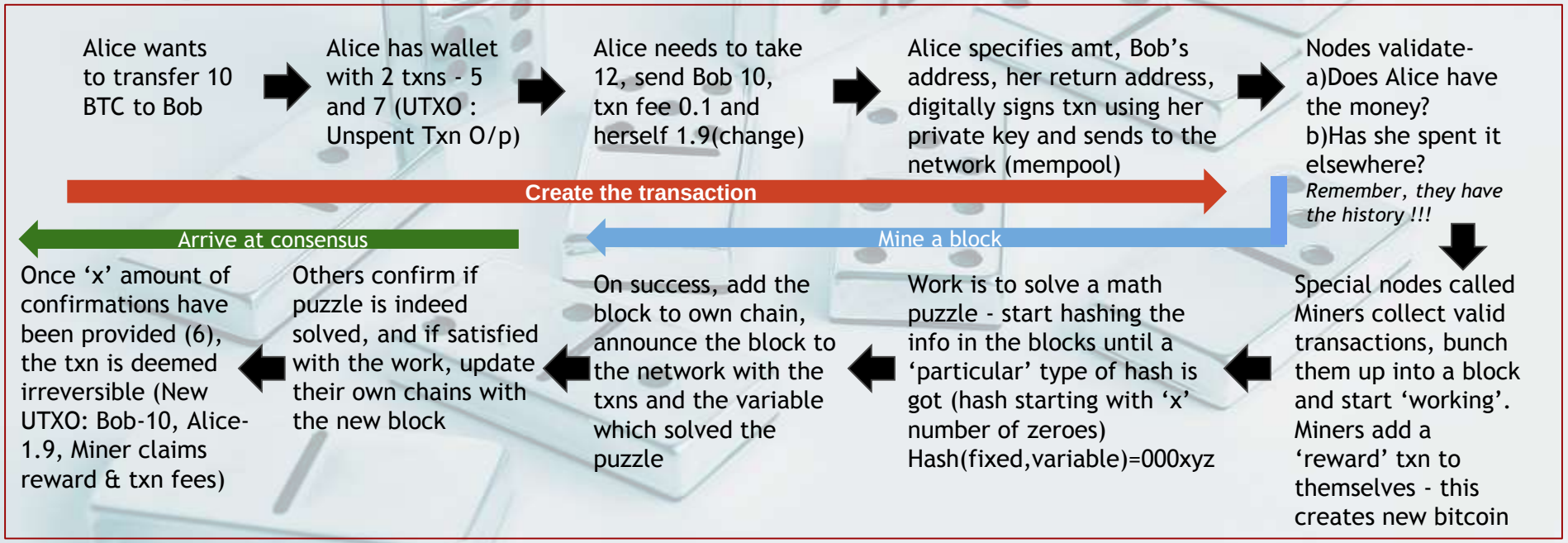
Breaking it down, bitcoin by bitcoin



By the way - Bitcoin = the network, bitcoin = the currency (BTC) and 1 BTC = 100 million satoshi

Names used in Crypto examples - A for Alice, B for Bob, C…, D…,E for Eve,....., O for Olivia,.....,S for Sybil.....

Breaking it down, bitcoin by bitcoin

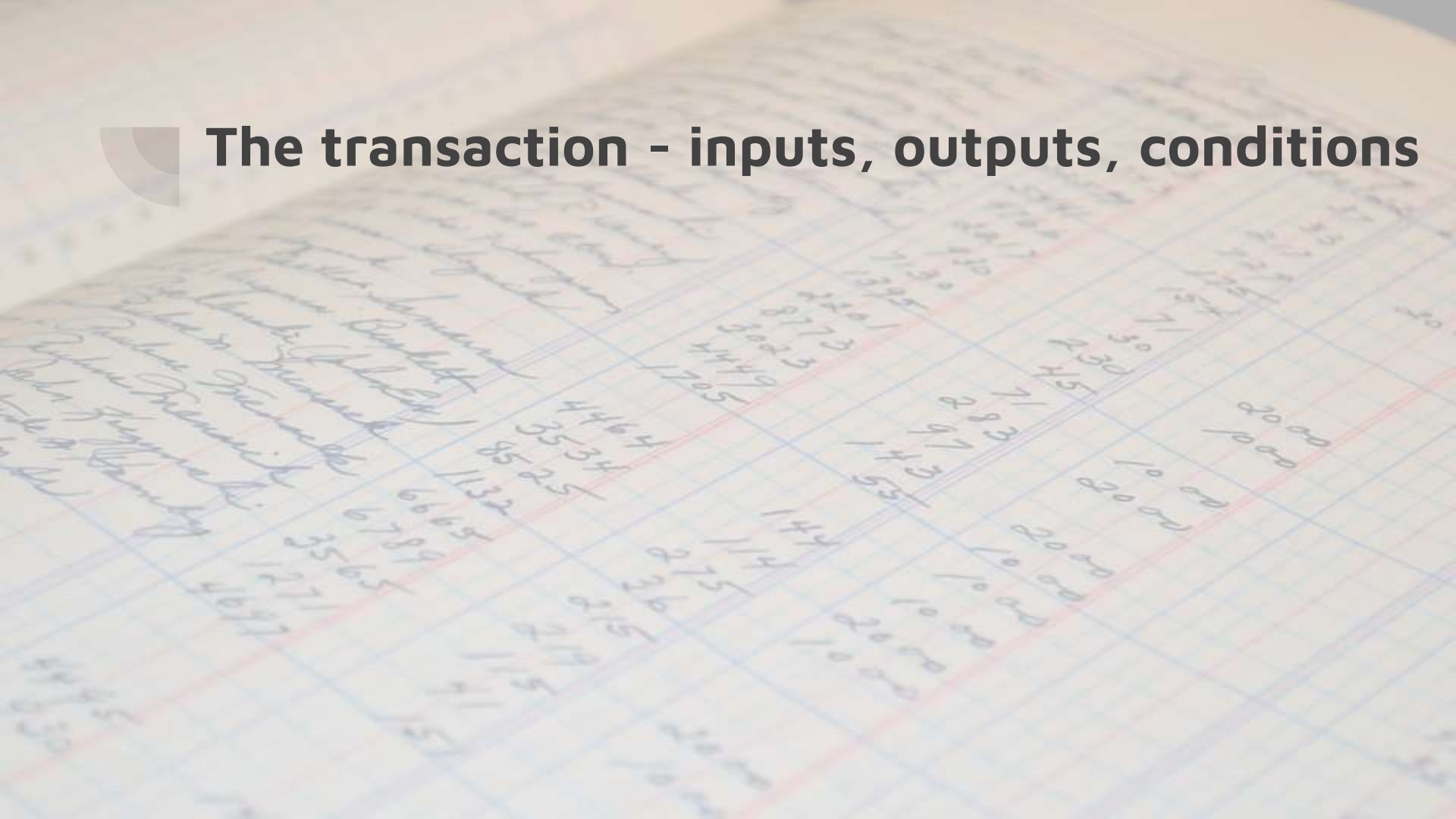


By the way - Bitcoin = the network, bitcoin = the currency (BTC) and 1 BTC = 100 million satoshi

Names used in Crypto examples - A for Alice, B for Bob, C…, D…,E for Eve,....., O for Olivia,.....,S for Sybil.....

https://bitbonkers.com/

The transaction - inputs, outputs, conditions



The transaction - inputs, outputs, conditions

bitcoins are owned by addresses - the person who has the key to the address is the owner

The transaction - inputs, outputs, conditions

bitcoins are owned by addresses - the person who has the key to the address is the owner

It is a ledger of transactions and not balances - only sent and received details stored

The transaction - inputs, outputs, conditions

bitcoins are owned by addresses - the person who has the key to the address is the owner

It is a ledger of transactions and not balances - only sent and received details stored

You can only send what you have previously received (or mined, which is a txn to yourself)

The transaction - inputs, outputs, conditions

bitcoins are owned by addresses - the person who has the key to the address is the owner

It is a ledger of transactions and not balances - only sent and received details stored

You can only send what you have previously received (or mined, which is a txn to yourself)

Inputs are where the money is coming from and Outputs are where the money is going to

The transaction - inputs, outputs, conditions

bitcoins are owned by addresses - the person who has the key to the address is the owner

It is a ledger of transactions and not balances - only sent and received details stored

You can only send what you have previously received (or mined, which is a txn to yourself)

Inputs are where the money is coming from and Outputs are where the money is going to

If you want to send 5 BTC, you have to collect all transactions which you have received that add up to 5 BTC or more, and not already spent - these become your Inputs

The transaction - inputs, outputs, conditions

bitcoins are owned by addresses - the person who has the key to the address is the owner

It is a ledger of transactions and not balances - only sent and received details stored

You can only send what you have previously received (or mined, which is a txn to yourself)

Inputs are where the money is coming from and Outputs are where the money is going to

If you want to send 5 BTC, you have to collect all transactions which you have received that add up to 5 BTC or more, and not already spent - these become your Inputs

Outputs are the amounts to be sent and the addresses of where they need to be sent to

The transaction - inputs, outputs, conditions

bitcoins are owned by addresses - the person who has the key to the address is the owner

It is a ledger of transactions and not balances - only sent and received details stored

You can only send what you have previously received (or mined, which is a txn to yourself)

Inputs are where the money is coming from and Outputs are where the money is going to

If you want to send 5 BTC, you have to collect all transactions which you have received that add up to 5 BTC or more, and not already spent - these become your Inputs

Outputs are the amounts to be sent and the addresses of where they need to be sent to

In the Input, you have to prove that you own the bitcoins - 'unlock' the bitcoins received earlier

In the Output, you set conditions for the recipient to prove their ownership when they want to spend

This is done through Public Key Cryptography using Script - the programming language of Bitcoin

The transaction - inputs, outputs, conditions

bitcoins are owned by addresses - the person who has the key to the address is the owner

It is a ledger of transactions and not balances - only sent and received details stored

You can only send what you have previously received (or mined, which is a txn to yourself)

Inputs are where the money is coming from and Outputs are where the money is going to

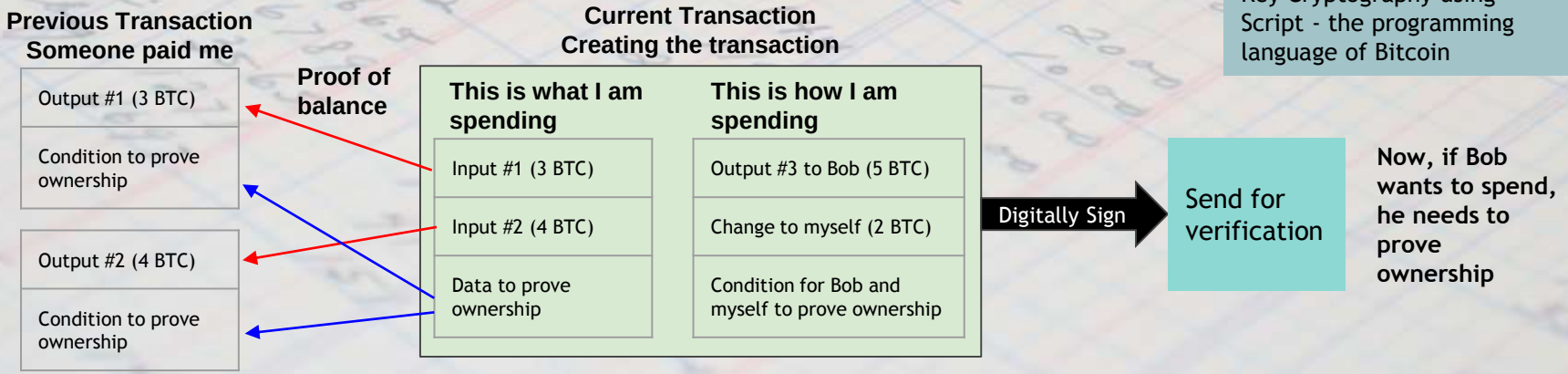
If you want to send 5 BTC, you have to collect all transactions which you have received that add up to 5 BTC or more, and not already spent - these become your Inputs

Outputs are the amounts to be sent and the addresses of where they need to be sent to

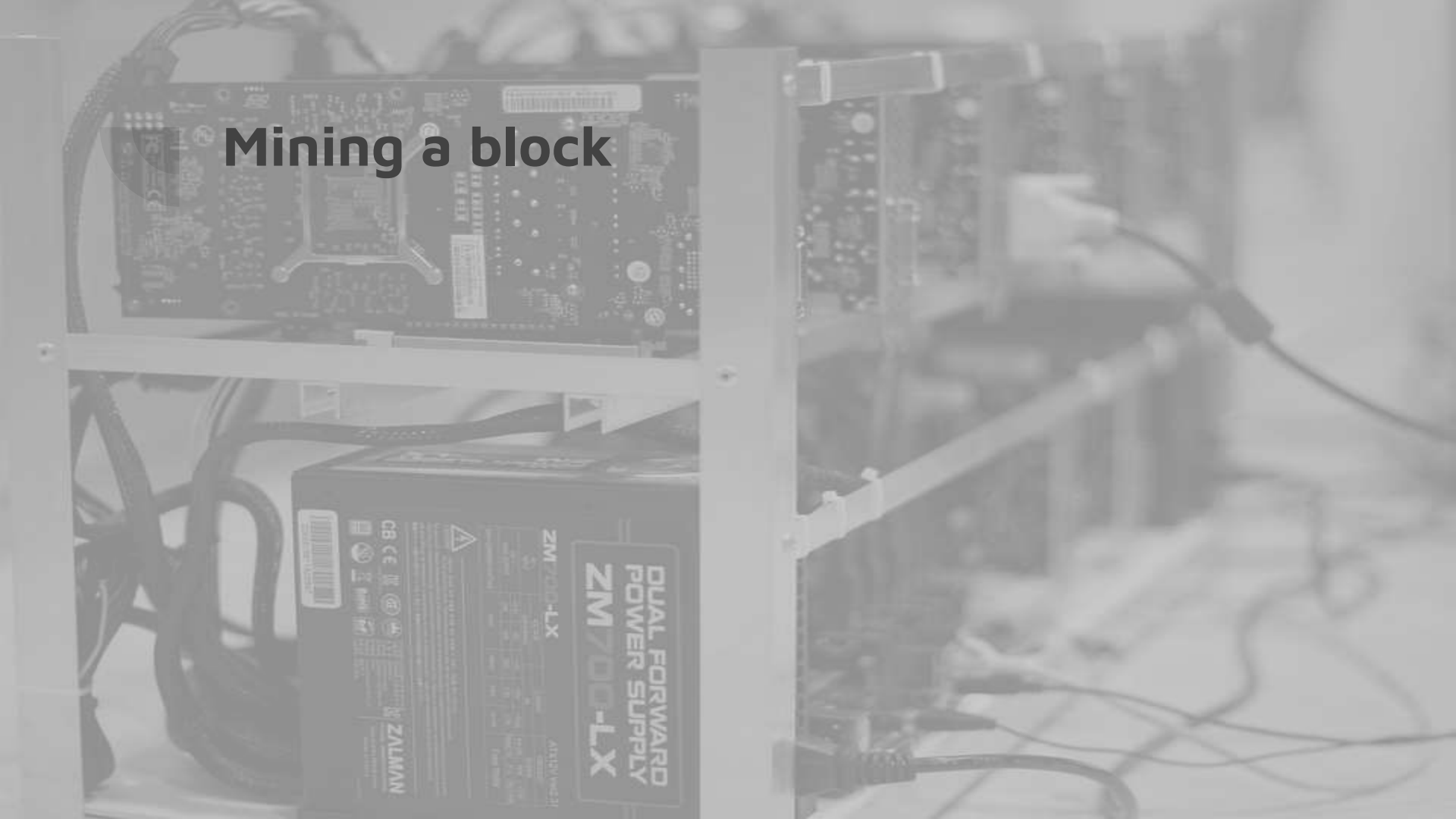
In the Input, you have to prove that you own the bitcoins - 'unlock' the bitcoins received earlier

In the Output, you set conditions for the recipient to prove their ownership when they want to spend

This is done through Public Key Cryptography using Script - the programming language of Bitcoin

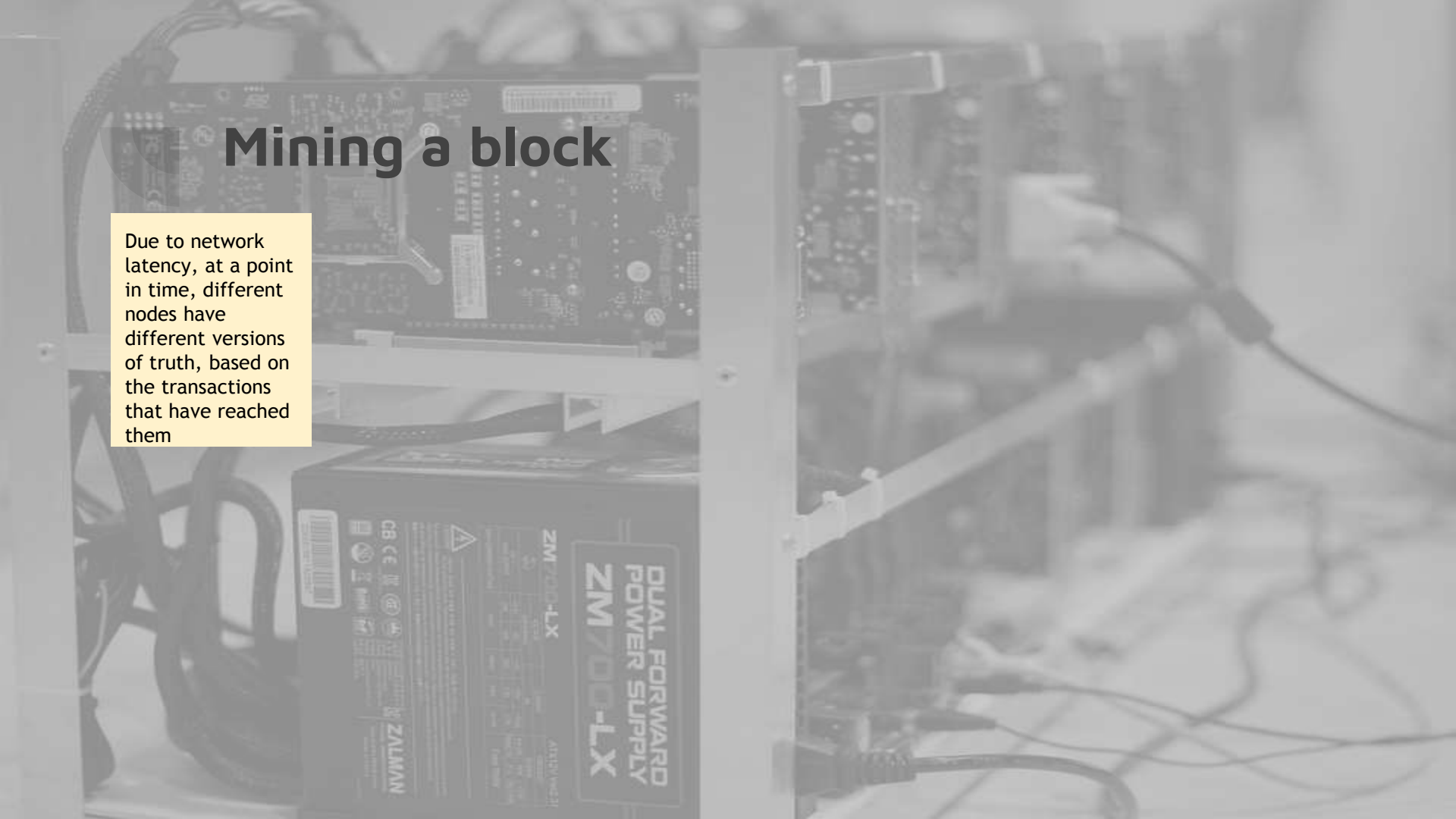


Mining a block



Mining a block

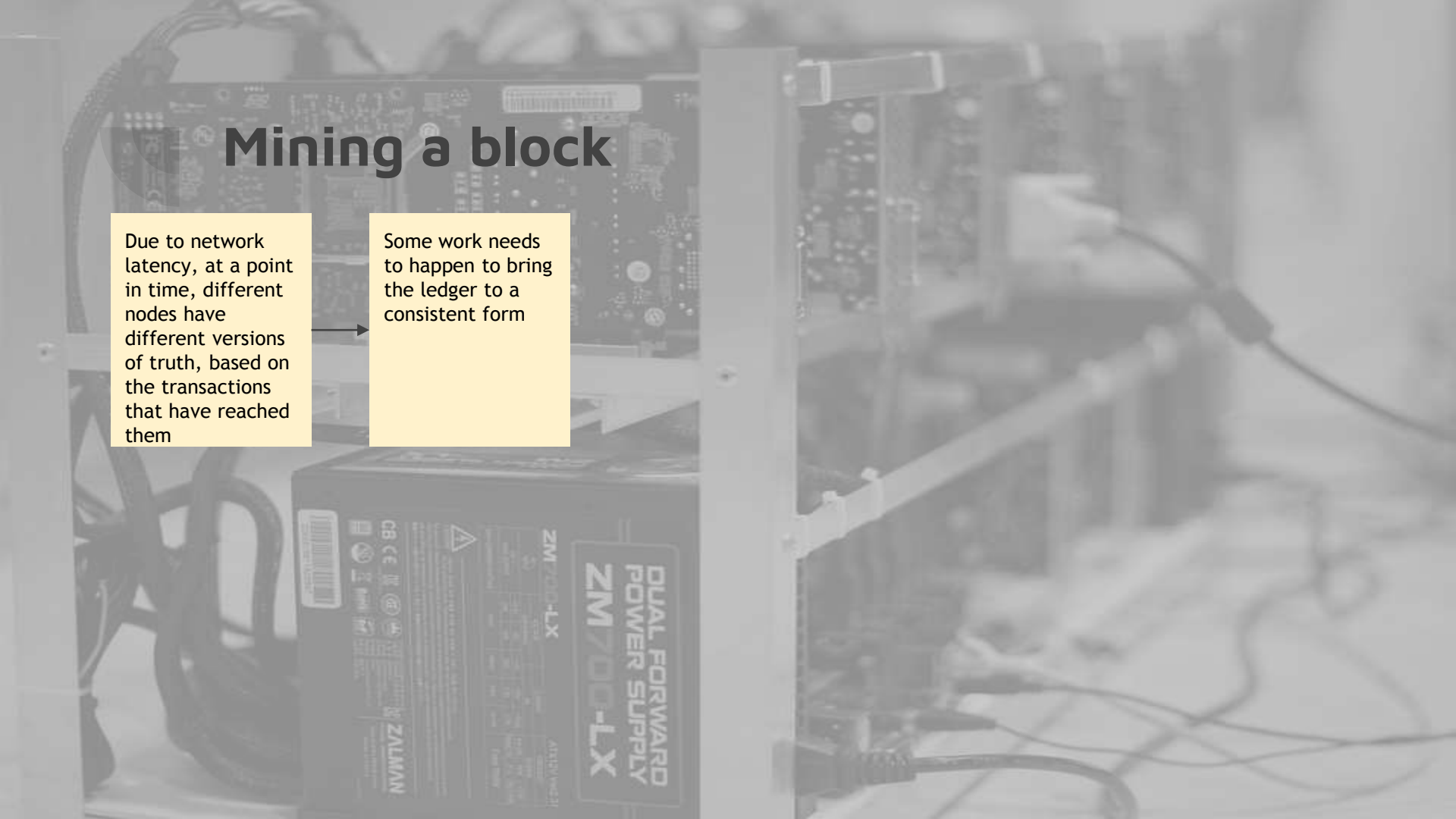
Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them



Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

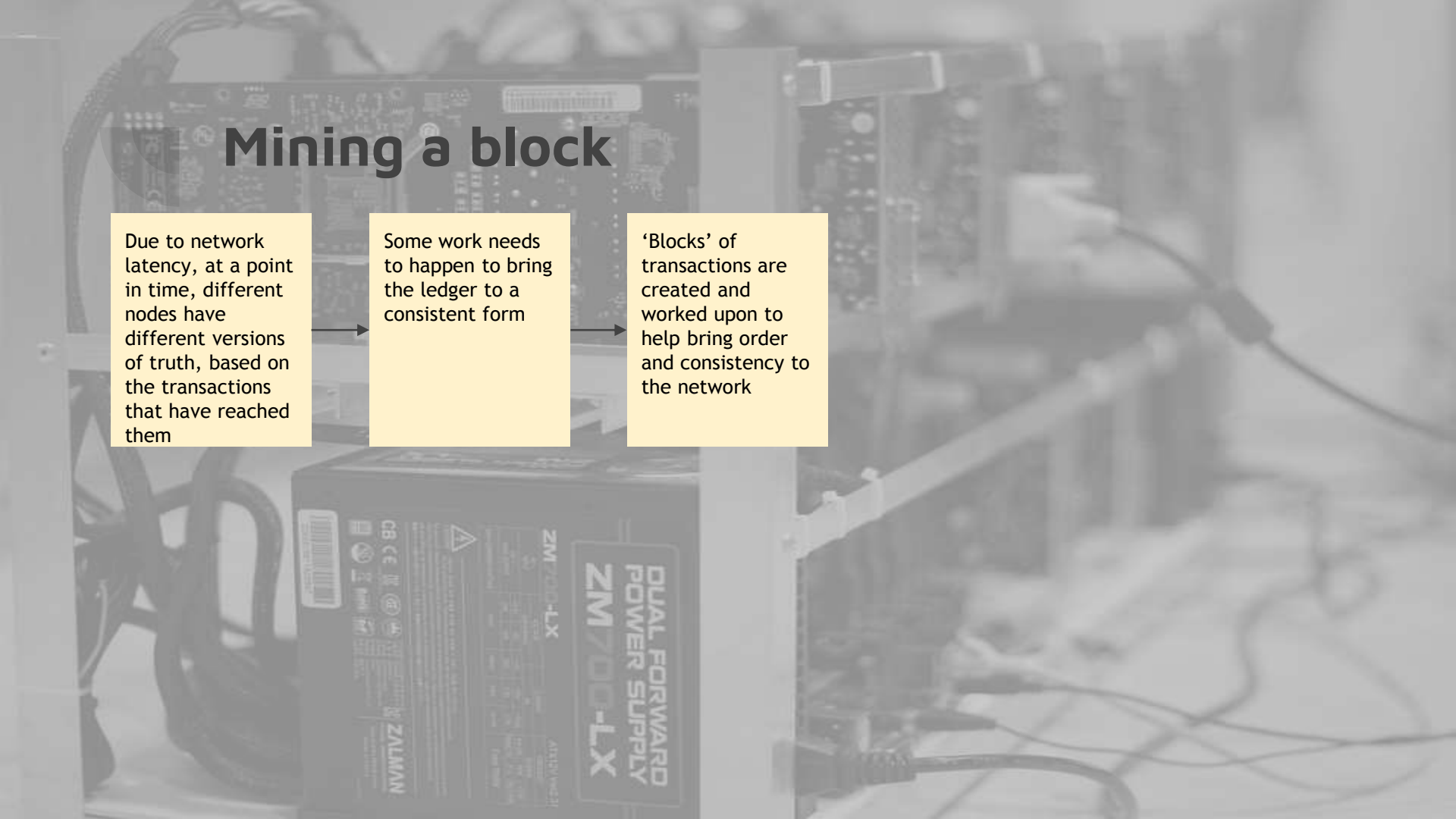


Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

‘Blocks’ of transactions are created and worked upon to help bring order and consistency to the network



Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

‘Blocks’ of transactions are created and worked upon to help bring order and consistency to the network

Fundamentally, a block is an arrangement of transactions

Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

'Blocks' of transactions are created and worked upon to help bring order and consistency to the network

Fundamentally, a block is an arrangement of transactions

Miners are responsible to collect unconfirmed transactions, perform validations and create a block

Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

'Blocks' of transactions are created and worked upon to help bring order and consistency to the network

Fundamentally, a block is an arrangement of transactions

Miners are responsible to collect unconfirmed transactions, perform validations and create a block

To prevent fraudulent transactions and blocks, a safeguard has been created using **Proof of Work (PoW)**

Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

'Blocks' of transactions are created and worked upon to help bring order and consistency to the network

Fundamentally, a block is an arrangement of transactions

Miners are responsible to collect unconfirmed transactions, perform validations and create a block

To prevent fraudulent transactions and blocks, a safeguard has been created using **Proof of Work (PoW)**

To get the block accepted, miners must race each other to solve a computationally difficult problem, thereby providing Proof of Work for the block

Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

'Blocks' of transactions are created and worked upon to help bring order and consistency to the network

Fundamentally, a block is an arrangement of transactions

Miners are responsible to collect unconfirmed transactions, perform validations and create a block

To prevent fraudulent transactions and blocks, a safeguard has been created using **Proof of Work (PoW)**

To get the block accepted, miners must race each other to solve a computationally difficult problem, thereby providing Proof of Work for the block

The Proof of Work is difficult to solve, but easy to verify

Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

'Blocks' of transactions are created and worked upon to help bring order and consistency to the network

Fundamentally, a block is an arrangement of transactions

Miners are responsible to collect unconfirmed transactions, perform validations and create a block

To prevent fraudulent transactions and blocks, a safeguard has been created using **Proof of Work (PoW)**

To get the block accepted, miners must race each other to solve a computationally difficult problem, thereby providing Proof of Work for the block

The Proof of Work is difficult to solve, but easy to verify

The miner who solves the Proof of Work problem propagates the block with the answer to the nodes in the network, who verify the solution

Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

'Blocks' of transactions are created and worked upon to help bring order and consistency to the network

Fundamentally, a block is an arrangement of transactions

Miners are responsible to collect unconfirmed transactions, perform validations and create a block

To prevent fraudulent transactions and blocks, a safeguard has been created using **Proof of Work (PoW)**

To get the block accepted, miners must race each other to solve a computationally difficult problem, thereby providing Proof of Work for the block

The Proof of Work is difficult to solve, but easy to verify

The miner who solves the Proof of Work problem propagates the block with the answer to the nodes in the network, who verify the solution

The successful miner receives a reward in bitcoin and collects the transaction fees available in the block

Mining a block

Due to network latency, at a point in time, different nodes have different versions of truth, based on the transactions that have reached them

Some work needs to happen to bring the ledger to a consistent form

'Blocks' of transactions are created and worked upon to help bring order and consistency to the network

Fundamentally, a block is an arrangement of transactions

Miners are responsible to collect unconfirmed transactions, perform validations and create a block

To prevent fraudulent transactions and blocks, a safeguard has been created using **Proof of Work (PoW)**

To get the block accepted, miners must race each other to solve a computationally difficult problem, thereby providing Proof of Work for the block

The Proof of Work is difficult to solve, but easy to verify

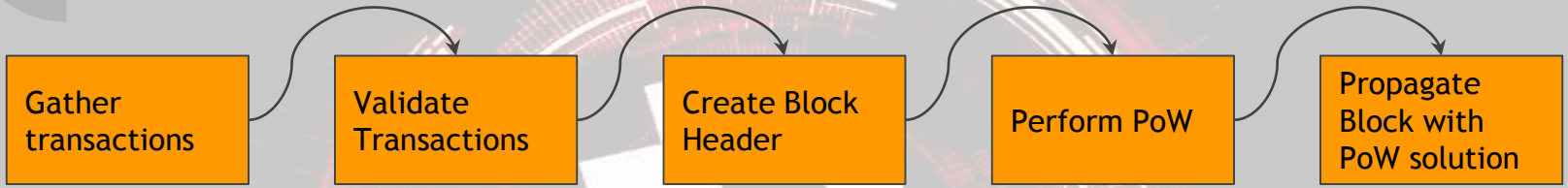
The miner who solves the Proof of Work problem propagates the block with the answer to the nodes in the network, who verify the solution

The successful miner receives a reward in bitcoin and collects the transaction fees available in the block

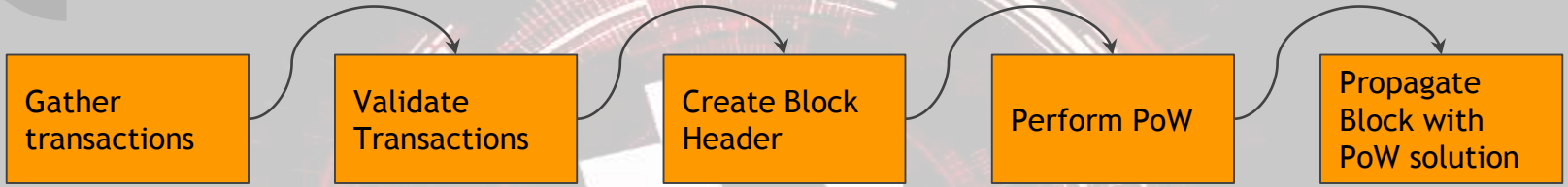
- The transactions are confirmed in a trustless, distributed manner
- Money has been generated through the reward

The mining process

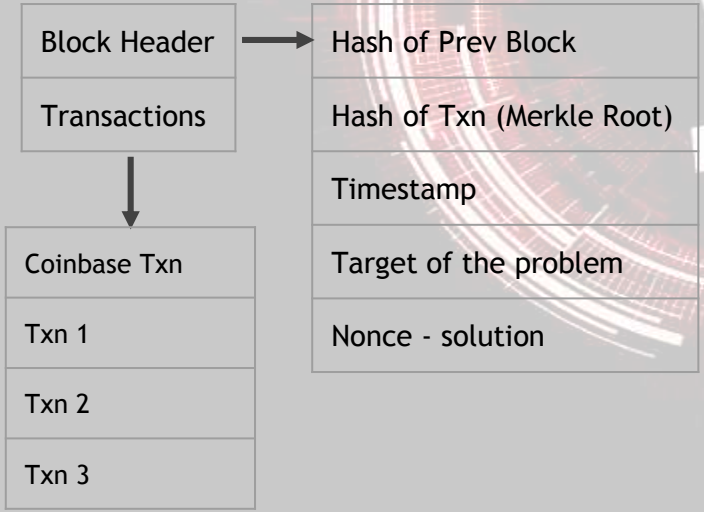
The mining process



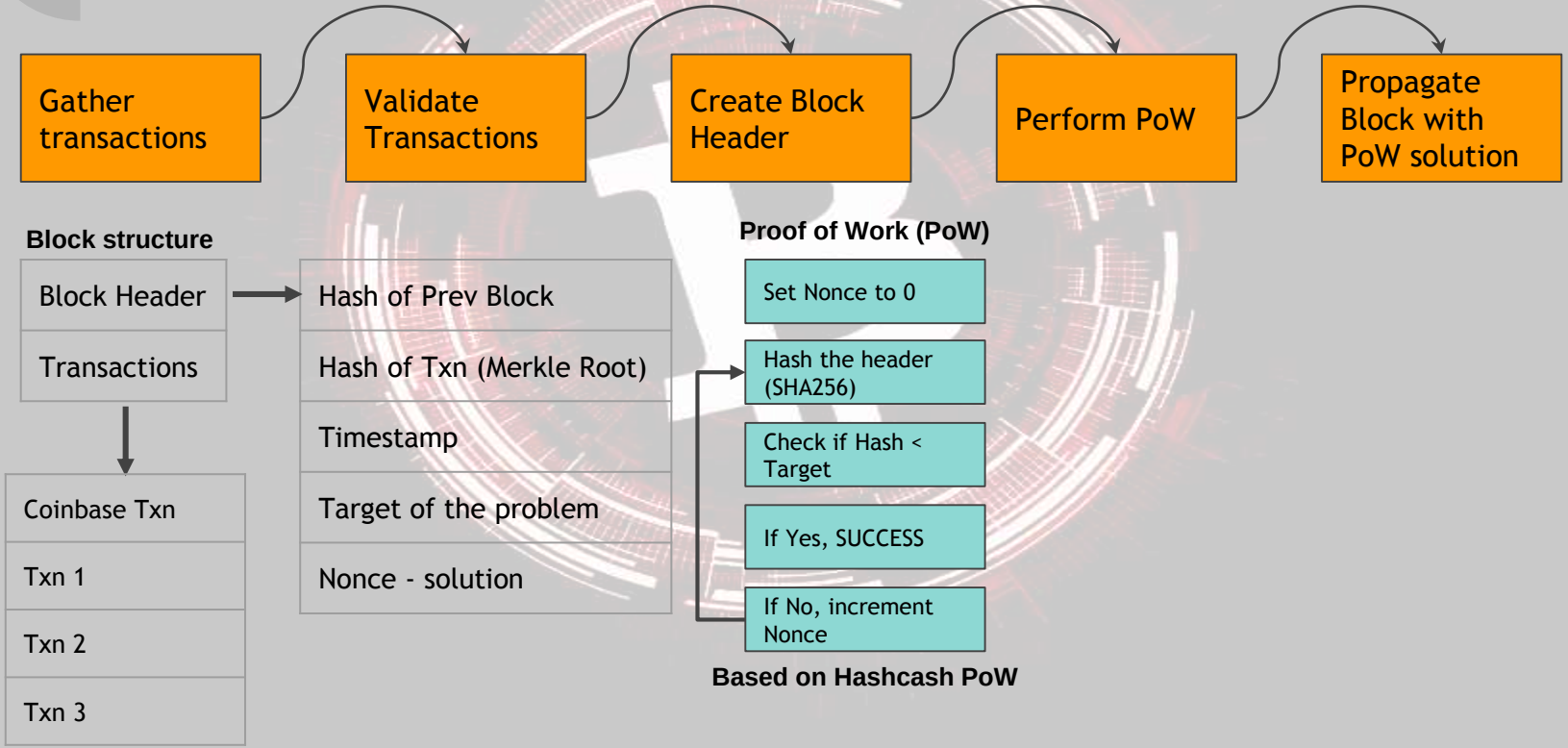
The mining process



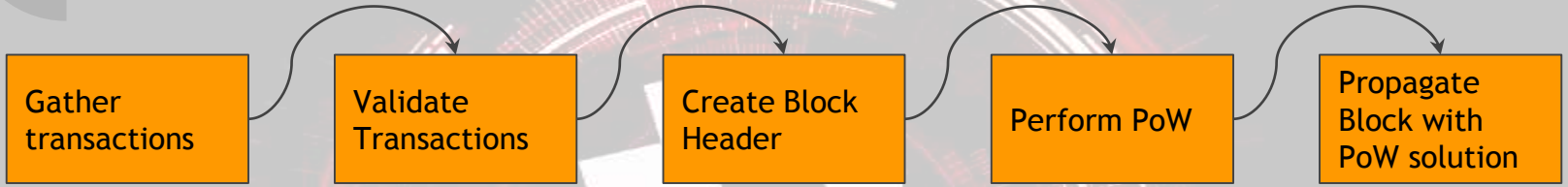
Block structure



The mining process



The mining process



Block structure

- Block Header
  - Hash of Prev Block
  - Hash of Txn (Merkle Root)
  - Timestamp
  - Target of the problem
  - Nonce - solution
- Transactions
  - Coinbase Txn
  - Txn 1
  - Txn 2
  - Txn 3

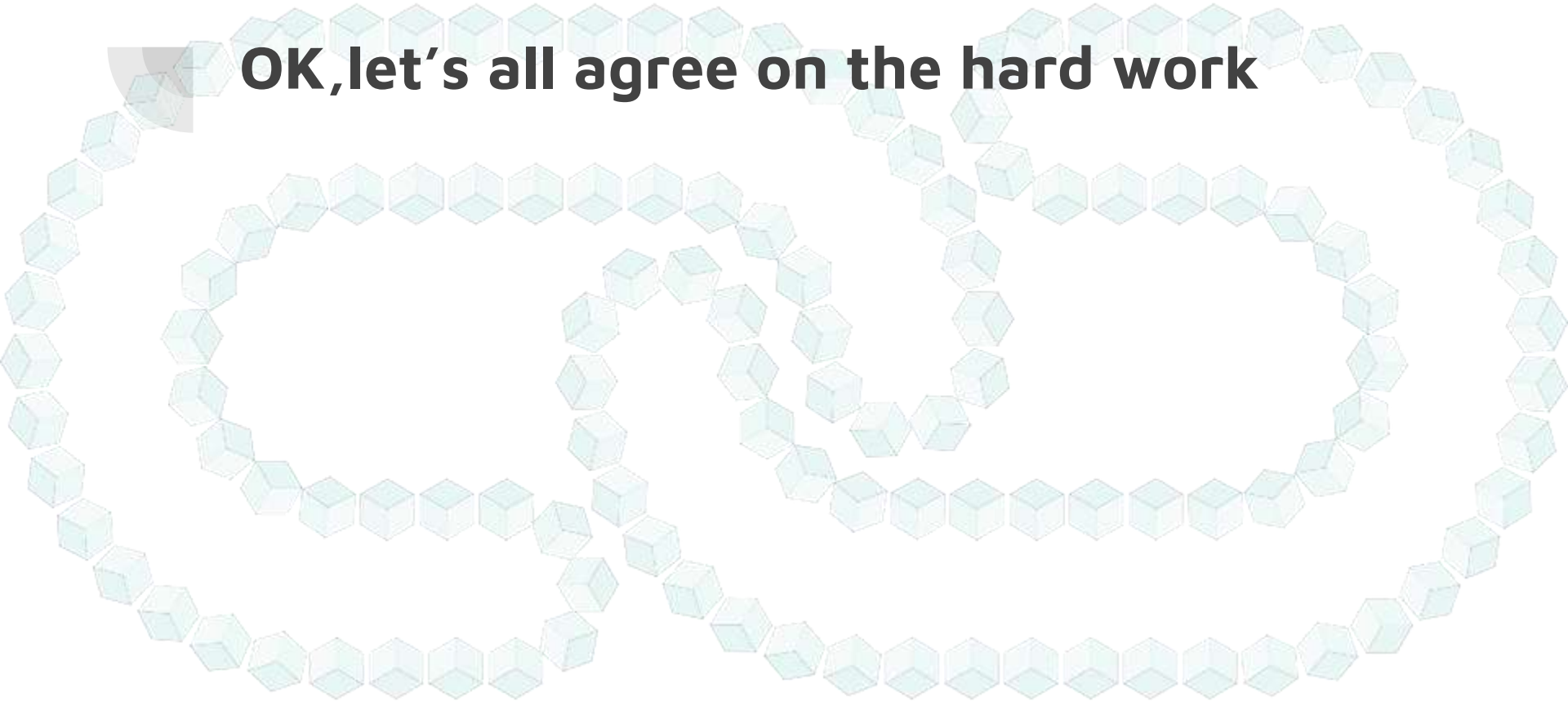
Proof of Work (PoW)

1. Set Nonce to 0
2. Hash the header (SHA256)
3. Check if Hash < Target
4. If Yes, SUCCESS
5. If No, increment Nonce

Based on Hashcash PoW

- Blocks are created every 10 mins on average
- If compute power (hashrate) increases, target changes to bring block creation time back to 10 mins
- Nonce - Number used Once
- Reward for a block started with 50 BTC, halves every 210000 blocks, currently 12.5
- At around 21 million, there will be almost Zero reward (c. 2140)
- Miners will then get txn fee for mining blocks
- The total reward (block reward + txn fees) is a ‘coinbase transaction’ inserted by miners, as a claim, in each block
- Since hash of previous block is included in hash of new block, changing prev block must ensure changing all subsequent blocks

OK,let's all agree on the hard work



OK,let's all agree on the hard work

Building the chain

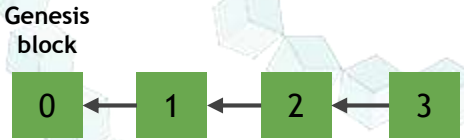
- ❑ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❑ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❑ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?

OK,let's all agree on the hard work

Building the chain

- ❏ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❏ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❏ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?

Txn Set A

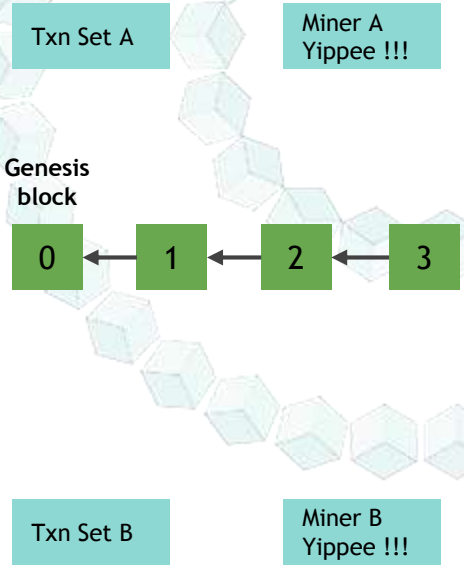


Txn Set B

OK,let's all agree on the hard work

Building the chain

- ❏ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❏ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❏ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?



OK,let's all agree on the hard work

Building the chain

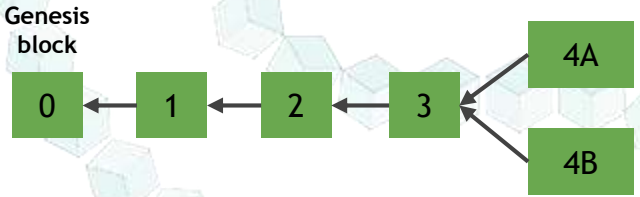
- ❏ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❏ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❏ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?

Txn Set A

Miner A
Yippee !!!

Nodes 1-10

Add Block 4A,Reject 4B
as its pointing to 3



Txn Set B

Miner B
Yippee !!!

Nodes 11-20

Add Block 4B,Reject 4A
as its pointing to 3

OK,let's all agree on the hard work

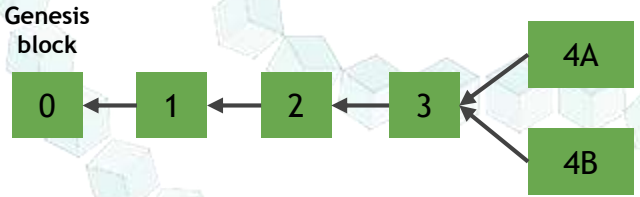
Building the chain

- ❏ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❏ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❏ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?

Txn Set A

Miner A
Yippee !!!

Nodes 1-10
Add Block 4A,Reject 4B
as its pointing to 3



Txn Set B

Miner B
Yippee !!!

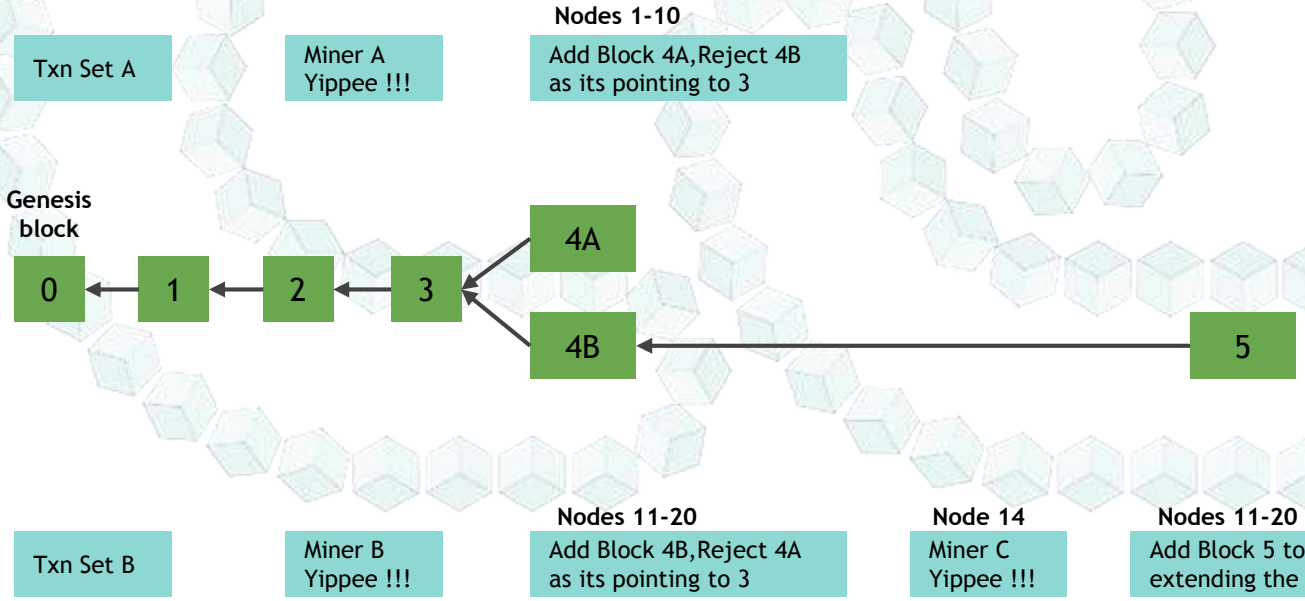
Nodes 11-20
Add Block 4B,Reject 4A
as its pointing to 3

Node 14
Miner C
Yippee !!!

OK,let's all agree on the hard work

Building the chain

- ❏ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❏ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❏ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?



OK,let's all agree on the hard work

Building the chain

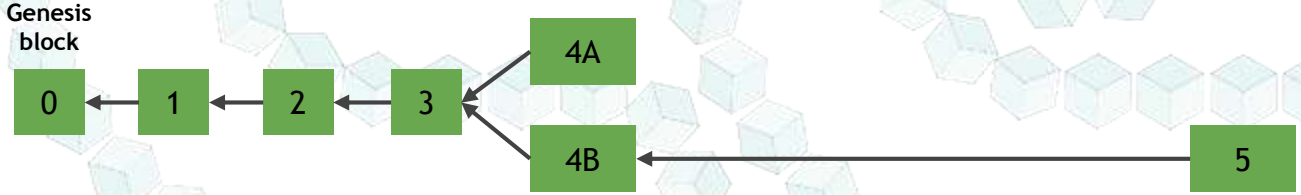
- ❏ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❏ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❏ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?

Txn Set A

Miner A
Yippee !!!

Nodes 1-10
Add Block 4A,Reject 4B
as its pointing to 3

Nodes 1-10
See 2 Chains
a)3-4A and b)3,4B,5



Txn Set B

Miner B
Yippee !!!

Nodes 11-20
Add Block 4B,Reject 4A
as its pointing to 3

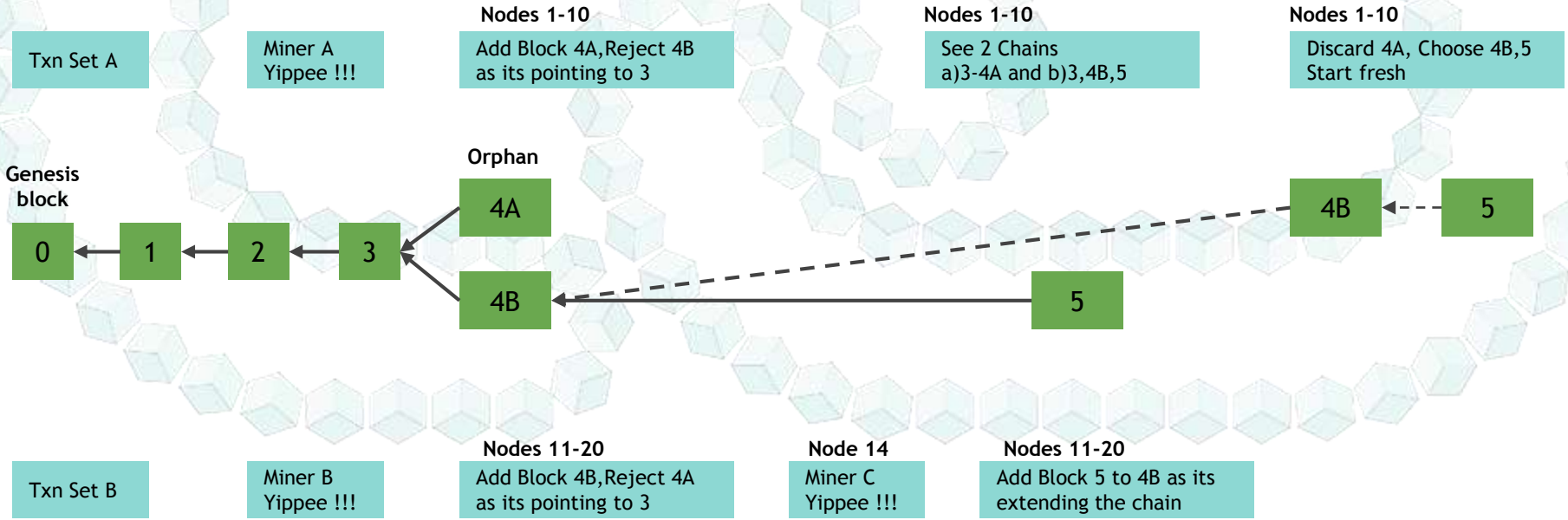
Node 14
Miner C
Yippee !!!

Nodes 11-20
Add Block 5 to 4B as its
extending the chain

OK,let's all agree on the hard work

Building the chain

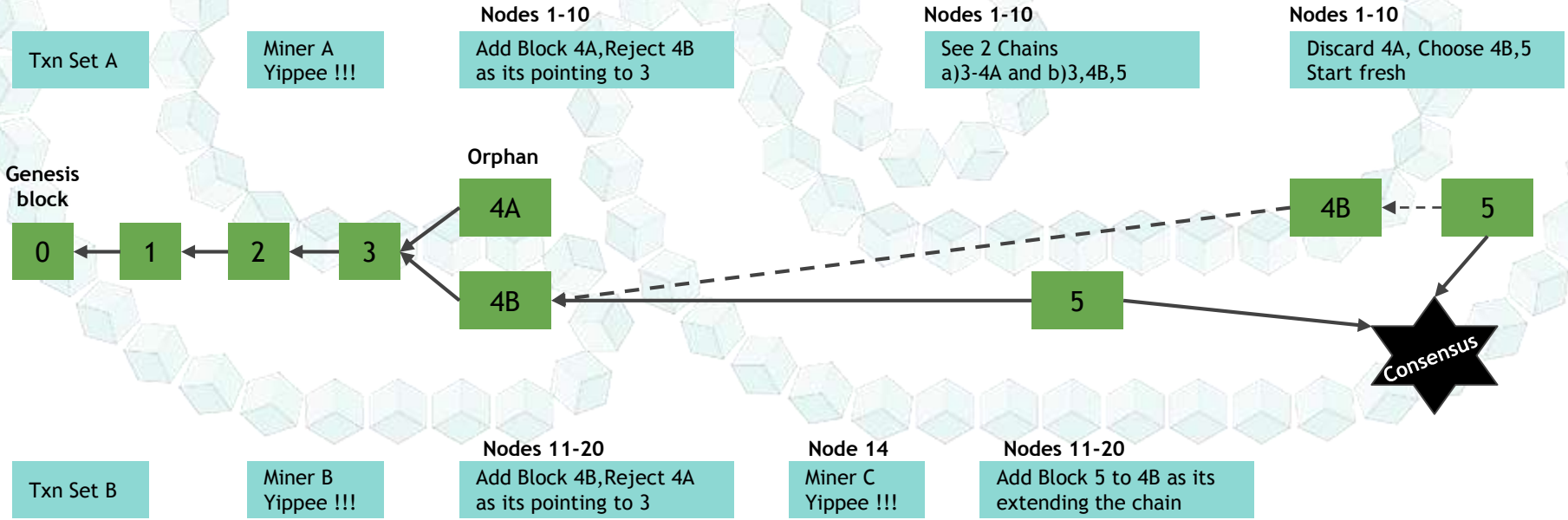
- ❏ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❏ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❏ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?



OK,let's all agree on the hard work

Building the chain

- ❏ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❏ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❏ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?



OK,let's all agree on the hard work

Building the chain

- ❑ Loads of transactions, lots of miners racing to find the solution, highly distributed
- ❑ Not a linear blockchain - possible that two miners find solutions at the same time unknown to each other
- ❑ What then? How will the nodes decide which one to add to the chain? How long to wait for the decision?

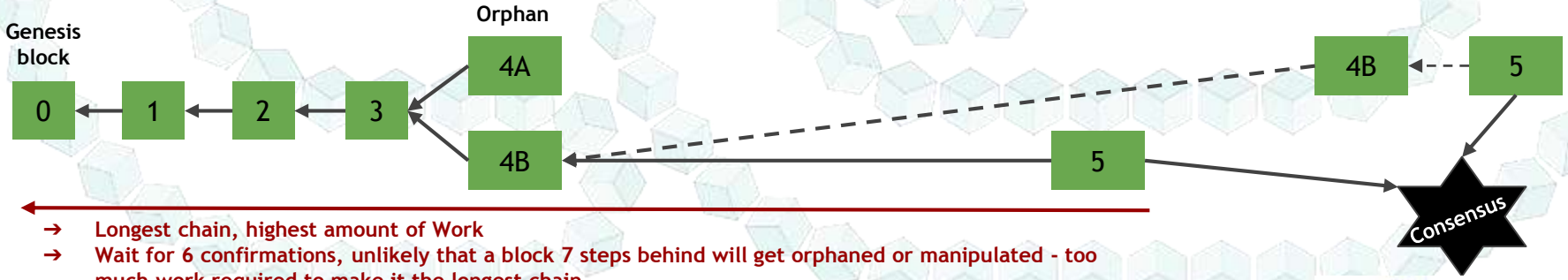
Txn Set A

Miner A
Yippee !!!

Nodes 1-10
Add Block 4A,Reject 4B
as its pointing to 3

Nodes 1-10
See 2 Chains
a)3-4A and b)3,4B,5

Nodes 1-10
Discard 4A, Choose 4B,5
Start fresh



- ➔ Longest chain, highest amount of Work
- ➔ Wait for 6 confirmations, unlikely that a block 7 steps behind will get orphaned or manipulated - too much work required to make it the longest chain

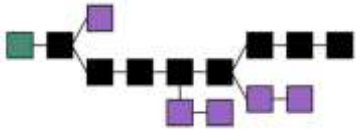
Txn Set B

Miner B
Yippee !!!

Nodes 11-20
Add Block 4B,Reject 4A
as its pointing to 3

Node 14
Miner C
Yippee !!!

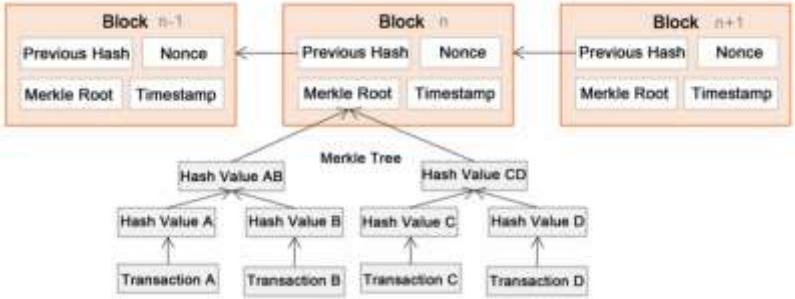
Nodes 11-20
Add Block 5 to 4B as its
extending the chain



In summary

In summary

ON EVERY NODE



In summary

Cheaper

Who to hack?

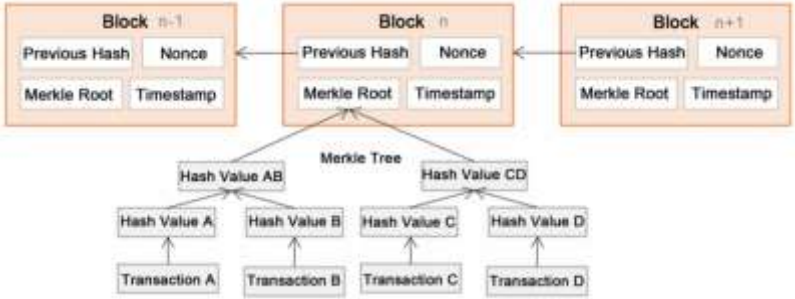
Can't say no, you signed it

Censorship proof

I am in control

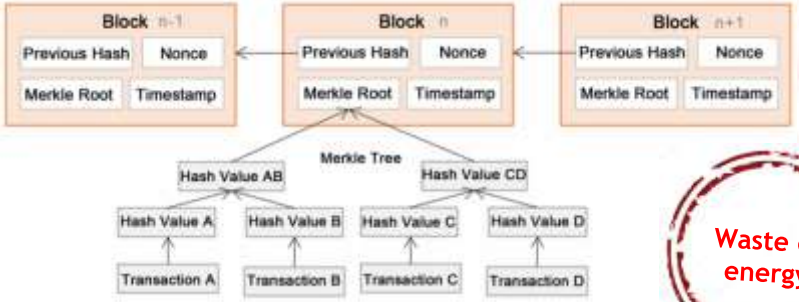
Change 1 txn, change the chain

ON EVERY NODE



In summary

ON EVERY NODE



Cheaper

Who to hack?

Can't say no, you signed it

Censorship proof

I am in control

Change 1 txn, change the chain

Waste of energy

Lost my pvt. key

Scaling SCALing SCALING

51% Attack

Who understands code?

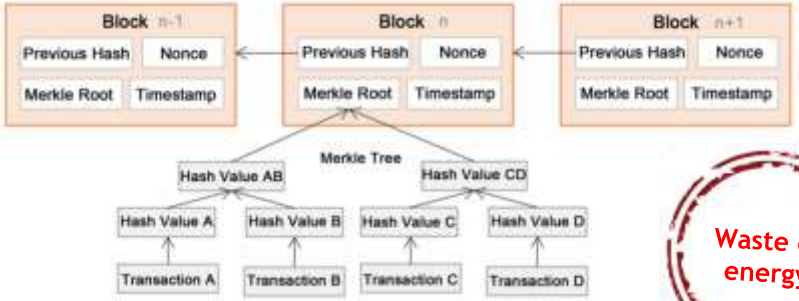
Reversal?

Hard forks

In summary

Cheaper

ON EVERY NODE



Lost my pvt. key

Who to hack?

Can't say no, you signed it

Waste of energy

Who understands code?

Censorship proof

I am in control

Mining Mining

Scaling SCALing SCALING

51% Attack

Change 1 txn, change the chain

Reversal?

Hard forks

Blockchain 2.0 - Smart Contracts

Blockchain 2.0 - Smart Contracts

- What if we wanted to do more than just transfer digital money around? What about other industries? Can they benefit? What's required for that?
- We need to be able to run complex code / computer programs on the blockchain.
- Remember - Bitcoin has a scripting language, but it is not powerful enough. It does what it was built for (pretty well !!!)
- So, we need something that supports running these 'programs' in a decentralized manner i.e., on each node of the blockchain
- We need a blockchain that can store 'programs' and 'execute' them based on instructions

Blockchain 2.0 - Smart Contracts

- What if we wanted to do more than just transfer digital money around? What about other industries? Can they benefit? What's required for that?
- We need to be able to run complex code / computer programs on the blockchain.
- Remember - Bitcoin has a scripting language, but it is not powerful enough. It does what it was built for (pretty well !!!)
- So, we need something that supports running these 'programs' in a decentralized manner i.e., on each node of the blockchain
- We need a blockchain that can store 'programs' and 'execute' them based on instructions

- Enter **ETHEREUM**, a platform built to host and execute these programs in a decentralized manner
- These programs are called '**smart contracts**'

Ethereum - programmable blockchain

Ethereum - programmable blockchain

- Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Ethereum – programmable blockchain

- Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum - programmable blockchain

- Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum Virtual Machine (EVM) - Each node runs an EVM which executes smart contract code

Ethereum - programmable blockchain

- Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum Virtual Machine (EVM) - Each node runs an EVM which executes smart contract code

Solidity - The main programming language, modelled on JavaScript (Serpent - Python, LLP - Lisp)

Ethereum - programmable blockchain

- Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum Virtual Machine (EVM) - Each node runs an EVM which executes smart contract code

Solidity - The main programming language, modelled on JavaScript (Serpent - Python, LLP - Lisp)

Accounts - 2 types -

- Externally Owned Accounts (EOA) - user accounts, with private keys
- Contract Accounts - refer to the programs, invoked/triggered by user accounts to execute via transactions
- Maintains **Account State** - balance of the account, no need to track UTXO

Ethereum - programmable blockchain

- Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum Virtual Machine (EVM) - Each node runs an EVM which executes smart contract code

Solidity - The main programming language, modelled on JavaScript (Serpent - Python, LLP - Lisp)

Accounts - 2 types -

- Externally Owned Accounts (EOA) - user accounts, with private keys
- Contract Accounts - refer to the programs, invoked/triggered by user accounts to execute via transactions
- Maintains **Account State** - balance of the account, no need to track UTXO

Gas - the transaction fee, in Ether, payable to the network to run the program. Every computation costs Gas. Prevents malicious or badly written code from running continuously because Gas runs out and program aborts.

Ethereum - programmable blockchain

- Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum Virtual Machine (EVM) - Each node runs an EVM which executes smart contract code

Solidity - The main programming language, modelled on JavaScript (Serpent - Python, LLP - Lisp)

Accounts - 2 types -

- Externally Owned Accounts (EOA) - user accounts, with private keys
- Contract Accounts - refer to the programs, invoked/triggered by user accounts to execute via transactions
- Maintains **Account State** - balance of the account, no need to track UTXO

Gas - the transaction fee, in Ether, payable to the network to run the program. Every computation costs Gas. Prevents malicious or badly written code from running continuously because Gas runs out and program aborts.

Gas limit - amount of gas you are willing to pay for your transaction. If less than what is required, you lose all. If more, you get refunded.

Ethereum - programmable blockchain

- ❏ Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- ❏ Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- ❏ Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum Virtual Machine (EVM) - Each node runs an EVM which executes smart contract code

Solidity - The main programming language, modelled on JavaScript (Serpent - Python, LLP - Lisp)

Accounts - 2 types -

- ❏ Externally Owned Accounts (EOA) - user accounts, with private keys
- ❏ Contract Accounts - refer to the programs, invoked/triggered by user accounts to execute via transactions
- ❏ Maintains **Account State** - balance of the account, no need to track UTXO

Gas - the transaction fee, in Ether, payable to the network to run the program. Every computation costs Gas. Prevents malicious or badly written code from running continuously because Gas runs out and program aborts.

Gas limit - amount of gas you are willing to pay for your transaction. If less than what is required, you lose all. If more, you get refunded.

Block Gas Limit - determination of the block size, currently around 7.9m - transaction cost cannot exceed this

Ethereum - programmable blockchain

- ❏ Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- ❏ Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- ❏ Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum Virtual Machine (EVM) - Each node runs an EVM which executes smart contract code

Solidity - The main programming language, modelled on JavaScript (Serpent - Python, LLP - Lisp)

Accounts - 2 types -

- ❏ Externally Owned Accounts (EOA) - user accounts, with private keys
- ❏ Contract Accounts - refer to the programs, invoked/triggered by user accounts to execute via transactions
- ❏ Maintains **Account State** - balance of the account, no need to track UTXO

Gas - the transaction fee, in Ether, payable to the network to run the program. Every computation costs Gas. Prevents malicious or badly written code from running continuously because Gas runs out and program aborts.

Gas limit - amount of gas you are willing to pay for your transaction. If less than what is required, you lose all. If more, you get refunded.

Block Gas Limit - determination of the block size, currently around 7.9m - transaction cost cannot exceed this

Ethereum Clients - the main software that runs nodes and connects to the blockchain - geth (Go), Parity (Rust)

Ethereum - programmable blockchain

- ❏ Proposed by Vitalik Buterin in 2013. Now in 2nd live release ‘Homestead’ (1st live release - Frontier)
- ❏ Uses the same blockchain consensus (Proof of Work) as in Bitcoin, but a different PoW - Ethash (ASIC resistant)
- ❏ Uses ‘Ether’ (ETH) as currency (‘wei’ is the smallest unit - 10^18) - Miners earn Ether to secure the network

Key Innovations

Ethereum Virtual Machine (EVM) - Each node runs an EVM which executes smart contract code

Solidity - The main programming language, modelled on JavaScript (Serpent - Python, LLP - Lisp)

Accounts - 2 types -

- ❏ Externally Owned Accounts (EOA) - user accounts, with private keys
- ❏ Contract Accounts - refer to the programs, invoked/triggered by user accounts to execute via transactions
- ❏ Maintains **Account State** - balance of the account, no need to track UTXO

Gas - the transaction fee, in Ether, payable to the network to run the program. Every computation costs Gas. Prevents malicious or badly written code from running continuously because Gas runs out and program aborts.

Gas limit - amount of gas you are willing to pay for your transaction. If less than what is required, you lose all. If more, you get refunded.

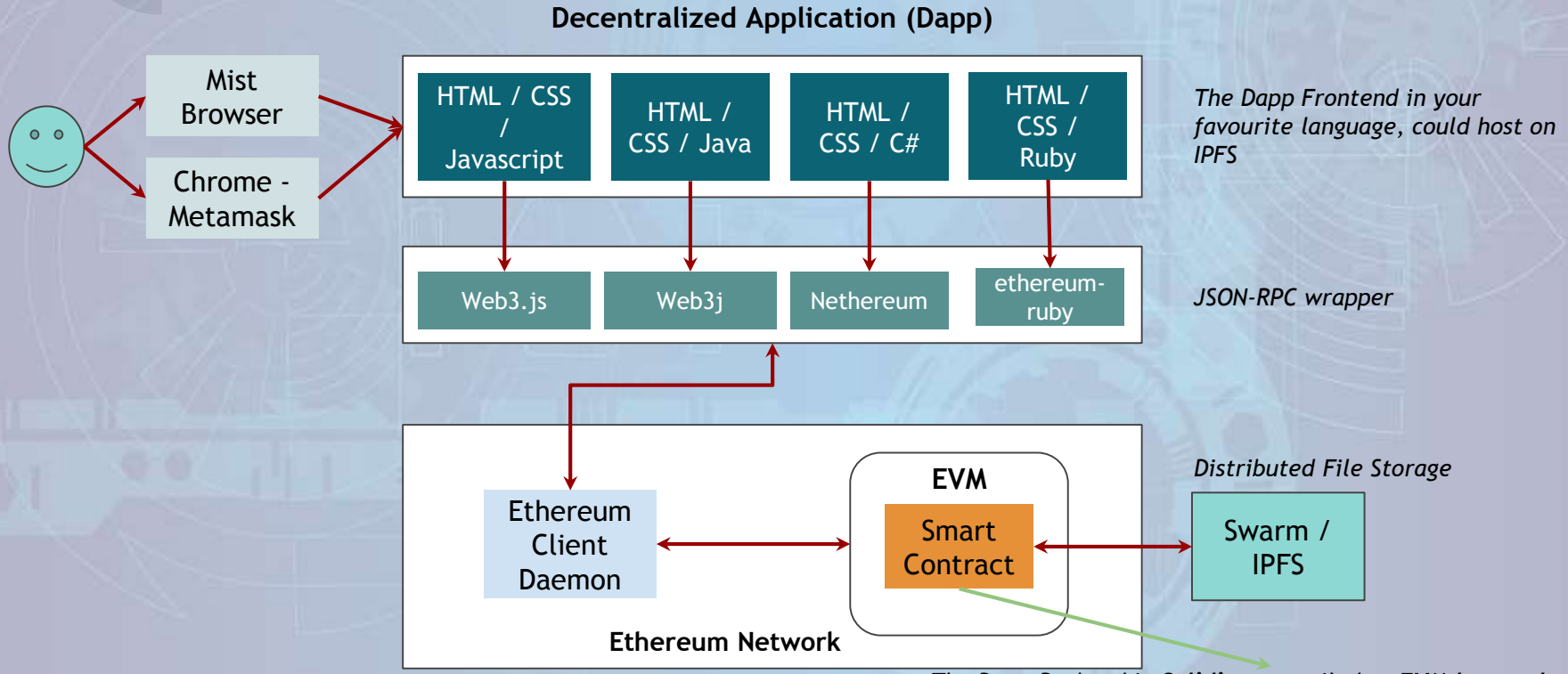
Block Gas Limit - determination of the block size, currently around 7.9m - transaction cost cannot exceed this

Ethereum Clients - the main software that runs nodes and connects to the blockchain - geth (Go), Parity (Rust)

Consensus - Blocks get created every 14 secs or so. Miners earn 3 ETH / block mined. Orphan blocks (Uncles) are also rewarded to promote decentralization.

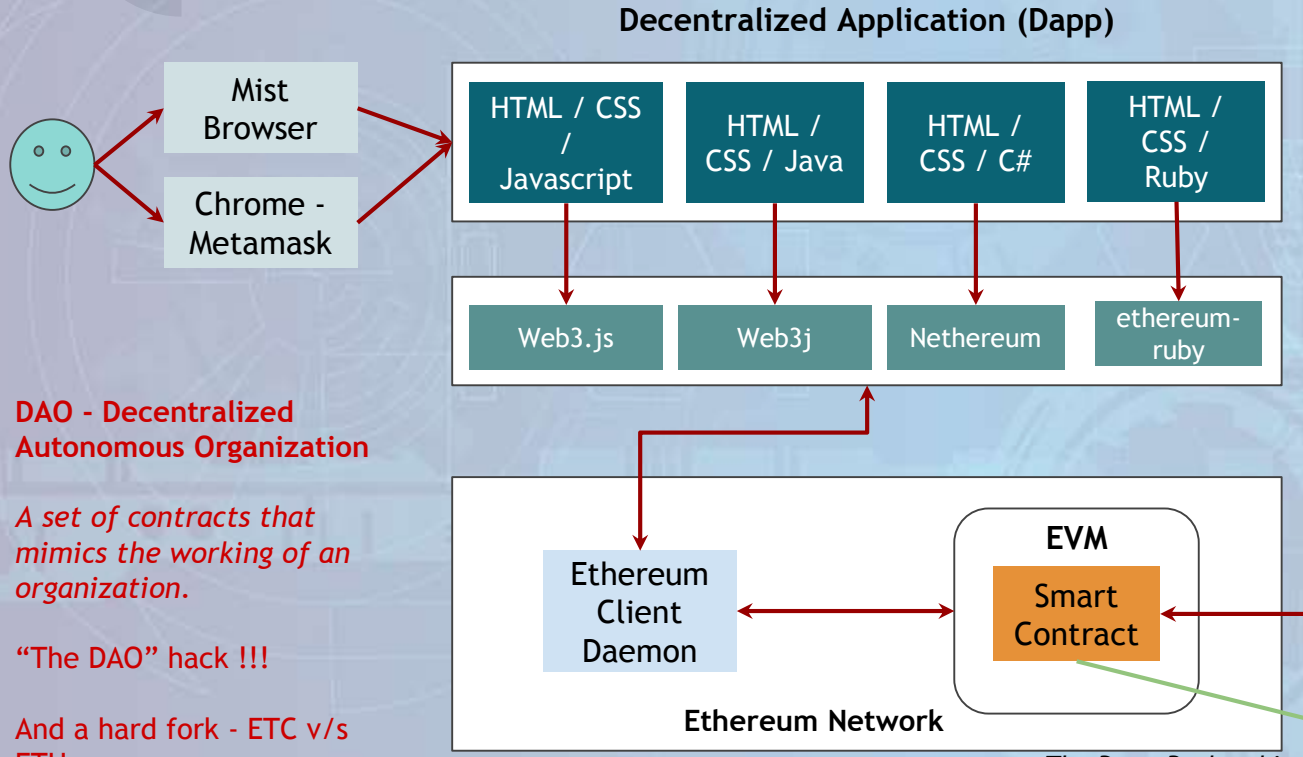
Ethereum ecosystem

Ethereum ecosystem



https://www.stateofthedapps.com

Ethereum ecosystem



The Dapp Frontend in your favourite language, could host on IPFS

JSON-RPC wrapper

Distributed File Storage

DAO - Decentralized Autonomous Organization

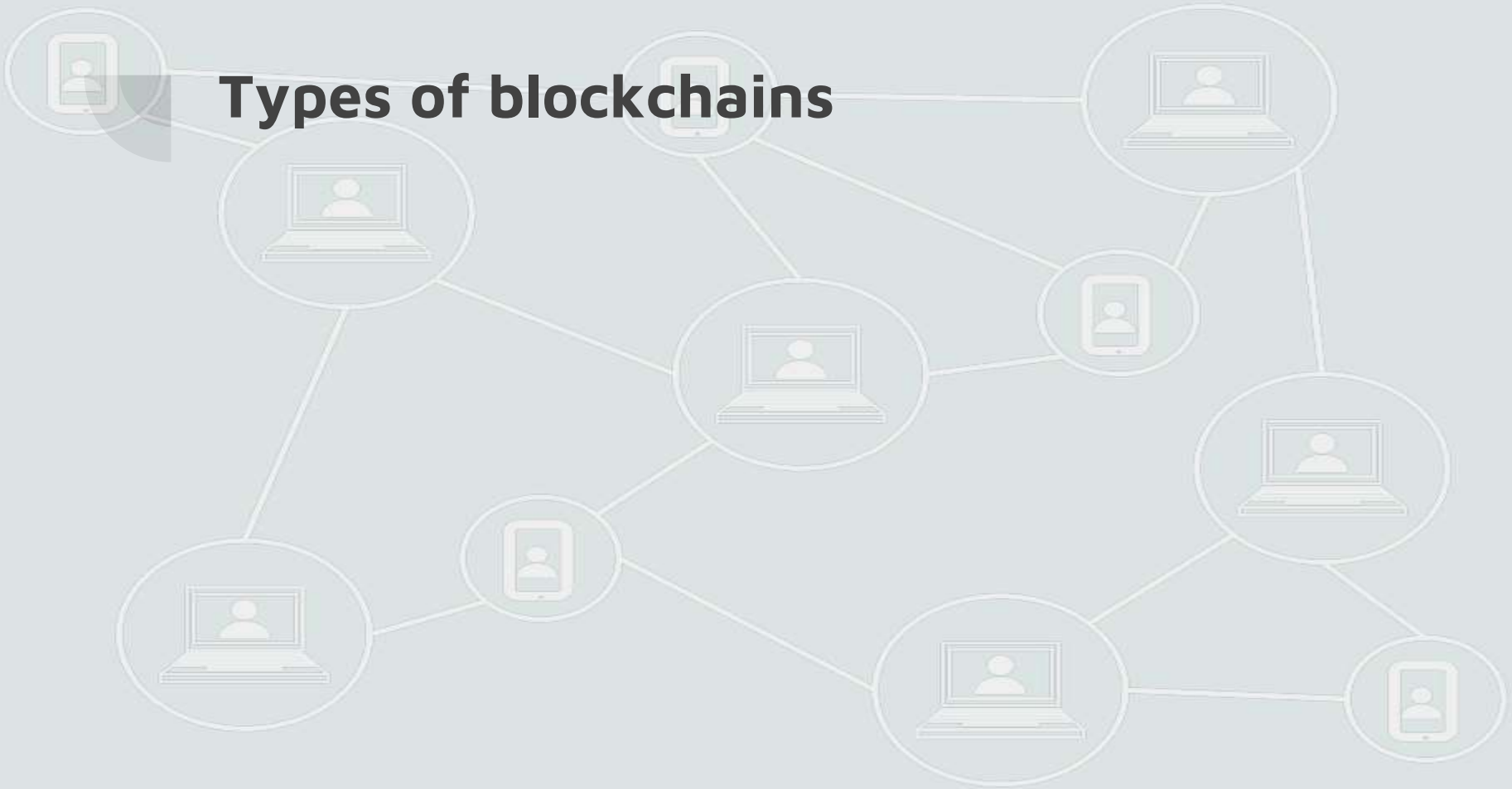
A set of contracts that mimics the working of an organization.

"The DAO" hack !!!

And a hard fork - ETC v/s ETH

*The Dapp Backend in **Solidity**, compiled to **EVM bytecode***
***Truffle** is a popular development framework, **Remix** is the IDE*

Types of blockchains



Types of blockchains

Permissionless	❏ No control ❏ Public, open for all (like Internet ?)	Bitcoin, Ethereum
Permissioned	❏ Controlled ❏ Completely private, where no public access is allowed - within the organization (like Intranet ?) ❏ Consortium or federated, for a set of organizations (like Extranet ?) ❏ Hybrid - some data is public, some private	❏ Platforms - Multichain, Hyperledger Fabric, Corda (R3), Monax ❏ Ripple ❏ Enterprise Ethereum Alliance (EEA) - Quorum (JP Morgan)

Types of blockchains

Permissionless	❏ No control ❏ Public, open for all (like Internet ?)	Bitcoin, Ethereum
Permissioned	❏ Controlled ❏ Completely private, where no public access is allowed - within the organization (like Intranet ?) ❏ Consortium or federated, for a set of organizations (like Extranet ?) ❏ Hybrid - some data is public, some private	❏ Platforms - Multichain, Hyperledger Fabric, Corda (R3), Monax ❏ Ripple ❏ Enterprise Ethereum Alliance (EEA) - Quorum (JP Morgan)

Blockchain as a Service - BaaS

- ❏ Microsoft (Azure) - Corda, Ethereum, Hyperledger Fabric, Quorum, Chain Core
- ❏ IBM (Bluemix) - Hyperledger
- ❏ Amazon (AWS) - Sawtooth, Corda, PokitDok, Samsung Nexledger
- ❏ Oracle (coming soon) - Hyperledger

Blockchain++

Blockchain++

Solving the Scalability issue

- Storage
  - Segwit, Sharding
- Alternate consensus mechanisms
  - Proof of Stake (PoS), Asynchronous Byzantine Fault Tolerant
- Layer 2
  - Off-chain (Lightning, Raiden), Sidechain
- **Non blockchain architecture?**
  - **Tangle (IOTA), Lattice (Nano), Hashgraph (Swirlds)**

Blockchain++

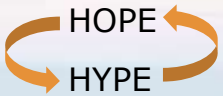
Solving the Scalability issue

- Storage
  - Segwit, Sharding
- Alternate consensus mechanisms
  - Proof of Stake (PoS), Asynchronous Byzantine Fault Tolerant
- Layer 2
  - Off-chain (Lightning, Raiden), Sidechain
- **Non blockchain architecture?**
  - **Tangle (IOTA), Lattice (Nano), Hashgraph (Swirlds)**

Platforms other than Bitcoin & Ethereum

- Cardano
- EOS
- Neo
- IOTA
- Nano
- Tezos
- Qtum
- Vechain
- Lisk
- Stellar
- Nem
- Icon

Blockchain marketplace



Blockchain marketplace

HOPE
HYPE

IDC Market Glance: Blockchain

Source: IDC, 4Q17

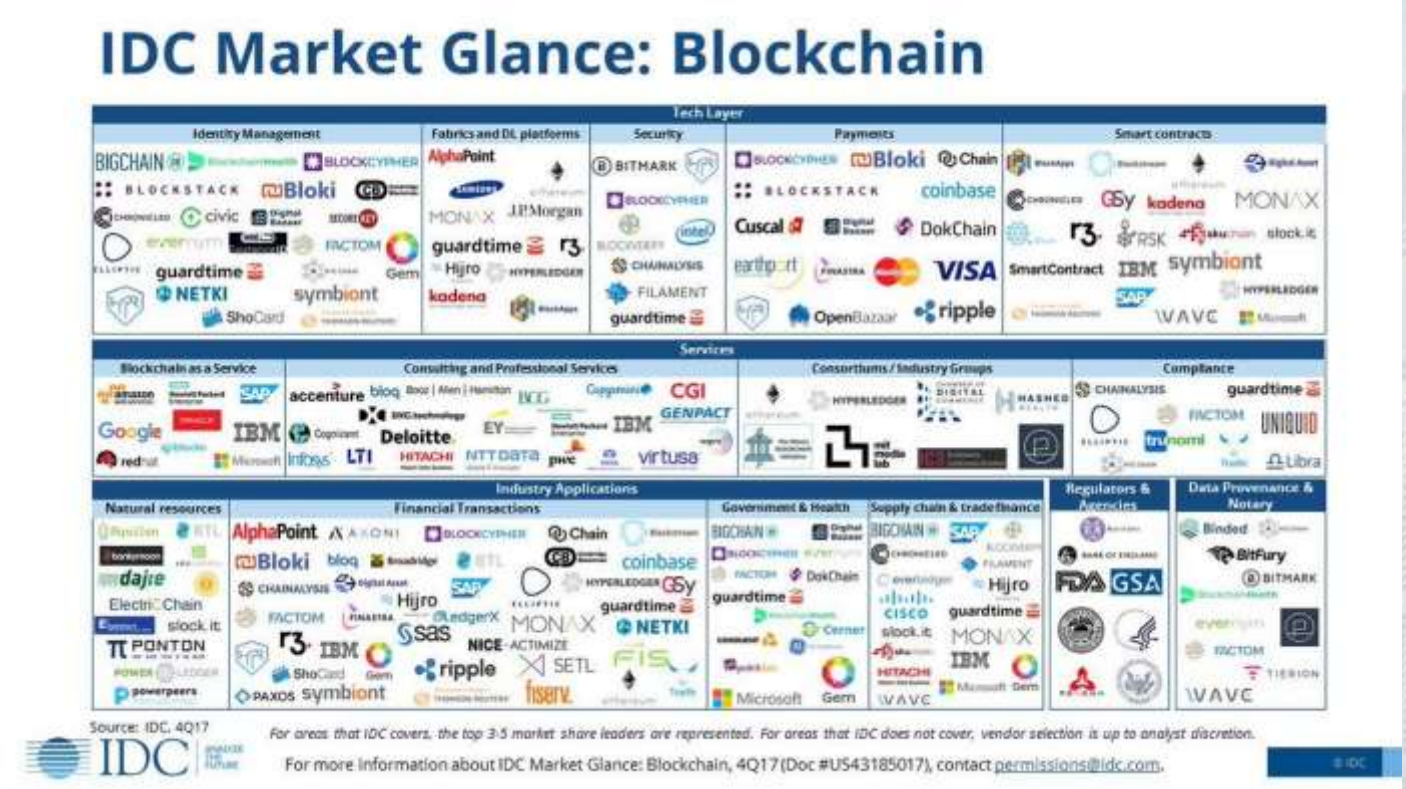
For areas that IDC covers, the top 3-5 market share leaders are represented. For areas that IDC does not cover, vendor selection is up to analyst discretion.

For more information about IDC Market Glance: Blockchain, 4Q17 (Doc #US43185017), contact permissions@idc.com.

Coindesk - State of Blockchain Q4, 2017
https://media.coindesk.com/uploads/research/state-of-blockchain/2017/q4/sob2017q4-2018.pdf

Blockchain marketplace

HOPE
HYPE



Further exploring

Further exploring

- The Bitcoin Whitepaper
- Public Key Cryptography
- Digital Signatures
- Byzantine Generals' Problem
- Byzantine Fault Tolerance
- Merkle, Patricia trees
- Mining Pools
- Full nodes, Light nodes
- Exchanges and buying
- Wallets, storing and "HODL"

Further exploring

- ❏ The Bitcoin Whitepaper
- ❏ Public Key Cryptography
- ❏ Digital Signatures
- ❏ Byzantine Generals' Problem
- ❏ Byzantine Fault Tolerance
- ❏ Merkle, Patricia trees
- ❏ Mining Pools
- ❏ Full nodes, Light nodes
- ❏ Exchanges and buying
- ❏ Wallets, storing and "HODL"
- ❏ Proof of Stake
- ❏ ICOs & Altcoins
- ❏ Lightning Network
- ❏ Ethereum roadmap
- ❏ Hyperledger
- ❏ Use cases of Blockchain
- ❏ Decentralized DNS and IPFS
- ❏ https://steemit.com
- ❏ https://www.openbazaar.org

Further exploring

- ❏ The Bitcoin Whitepaper
- ❏ Public Key Cryptography
- ❏ Digital Signatures
- ❏ Byzantine Generals' Problem
- ❏ Byzantine Fault Tolerance
- ❏ Merkle, Patricia trees
- ❏ Mining Pools
- ❏ Full nodes, Light nodes
- ❏ Exchanges and buying
- ❏ Wallets, storing and "HODL"
- ❏ Proof of Stake
- ❏ ICOs & Altcoins
- ❏ Lightning Network
- ❏ Ethereum roadmap
- ❏ Hyperledger
- ❏ Use cases of Blockchain
- ❏ Decentralized DNS and IPFS
- ❏ https://steemit.com
- ❏ https://www.openbazaar.org

https://coinmarketcap.com https://blockchain.info https://etherscan.io https://lnmainnet.gaben.win

Further exploring

TO THE 

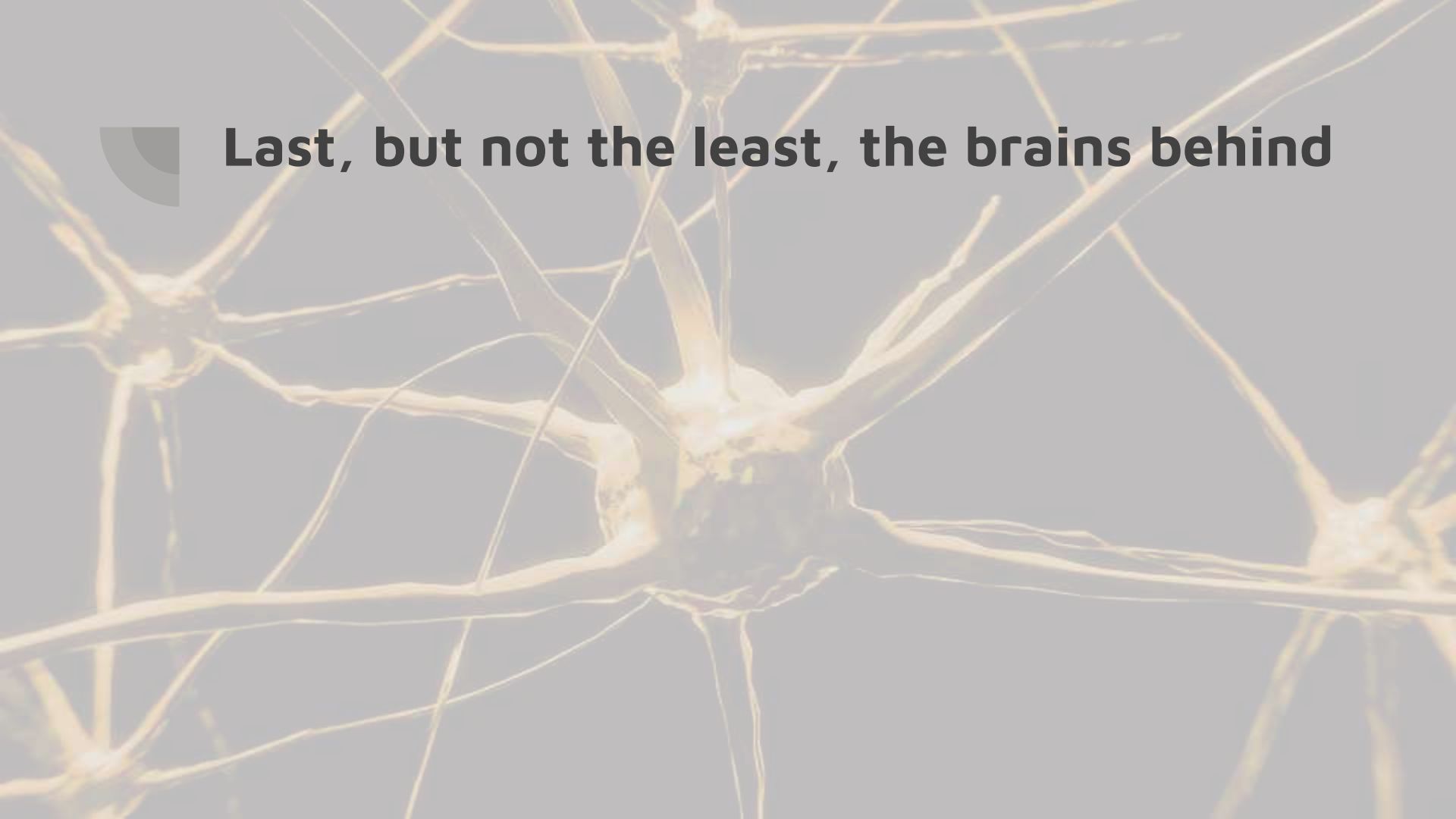
- ❏ The Bitcoin Whitepaper
- ❏ Public Key Cryptography
- ❏ Digital Signatures
- ❏ Byzantine Generals' Problem
- ❏ Byzantine Fault Tolerance
- ❏ Merkle, Patricia trees
- ❏ Mining Pools
- ❏ Full nodes, Light nodes
- ❏ Exchanges and buying
- ❏ Wallets, storing and "HODL"
- ❏ Proof of Stake
- ❏ ICOs & Altcoins
- ❏ Lightning Network
- ❏ Ethereum roadmap
- ❏ Hyperledger
- ❏ Use cases of Blockchain
- ❏ Decentralized DNS and IPFS
- ❏ https://steemit.com
- ❏ https://www.openbazaar.org

https://coinmarketcap.com https://blockchain.info https://etherscan.io https://lnmainnet.gaben.win

Further exploring

- ❏ The Bitcoin Whitepaper
- ❏ Public Key Cryptography
- ❏ Digital Signatures
- ❏ Byzantine Generals' Problem
- ❏ Byzantine Fault Tolerance
- ❏ Merkle, Patricia trees
- ❏ Mining Pools
- ❏ Full nodes, Light nodes
- ❏ Exchanges and buying
- ❏ Wallets, storing and "HODL"
- ❏ Proof of Stake
- ❏ ICOs & Altcoins
- ❏ Lightning Network
- ❏ Ethereum roadmap
- ❏ Hyperledger
- ❏ Use cases of Blockchain
- ❏ Decentralized DNS and IPFS
- ❏ https://steemit.com
- ❏ https://www.openbazaar.org

https://coinmarketcap.com https://blockchain.info https://etherscan.io https://lnmainnet.gaben.win



Last, but not the least, the brains behind

Last, but not the least, the brains behind

David Chaum	Cryptographer, created eCash, a digital cash system. Author of “Security Without Identification: Transaction Systems to Make Big Brother Obsolete.”
Timothy May, Eric Huges, John Gilmore	Cypherpunk founders. The Crypto Anarchist Manifesto https://www.activism.net/cypherpunk/crypto-anarchy.html by Tim May
Adam Back	Cypherpunk, created Hashcash, the PoW algorithm
Hal Finney	Cypherpunk, contributed to Proof of Work, received the first bitcoin from Satoshi
Nick Szabo	Conceptualized ‘bit gold’ before Bitcoin, first to speak about Smart Contracts
Wen Dei	Computer engineer, author of “b-money, an anonymous, distributed electronic cash system”, described principles of cryptocurrency

https://richtopia.com/inspirational-people/blockchain-top-100

Vitalik Buterin	Created Ethereum
Gavin Andresen	Took over as Bitcoin Core Dev Lead after Satoshi
Thanks to the below for helping me in my learning journey	
Andreas Antonopoulos	Blockchain expert, public speaker, author of must-read books - Mastering Bitcoin (technical) and The Internet of Money. Explains stuff really well !!!
Jackson Palmer	Dogecoin creator, Youtuber, awesome content
Ivan on Tech	Youtuber -made things simple to start with

Last, but not the least, the brains behind

David Chaum	Cryptographer, created eCash, a digital cash system. Author of “Security Without Identification: Transaction Systems to Make Big Brother Obsolete.”
Timothy May, Eric Huges, John Gilmore	Cypherpunk founders. The Crypto Anarchist Manifesto https://www.activism.net/cypherpunk/crypto-anarchy.html by Tim May
Adam Back	Cypherpunk, created Hashcash, the PoW algorithm
Hal Finney	Cypherpunk, contributed to Proof of Work, received the first bitcoin from Satoshi
Nick Szabo	Conceptualized ‘bit gold’ before Bitcoin, first to speak about Smart Contracts
Wen Dei	Computer engineer, author of “b-money, an anonymous, distributed electronic cash system”, described principles of cryptocurrency

https://richtopia.com/inspirational-people/blockchain-top-100

Vitalik Buterin	Created Ethereum
Gavin Andresen	Took over as Bitcoin Core Dev Lead after Satoshi
Thanks to the below for helping me in my learning journey	
Andreas Antonopoulos	Blockchain expert, public speaker, author of must-read books - Mastering Bitcoin (technical) and The Internet of Money. Explains stuff really well !!!
Jackson Palmer	Dogecoin creator, Youtuber, awesome content
Ivan on Tech	Youtuber -made things simple to start with