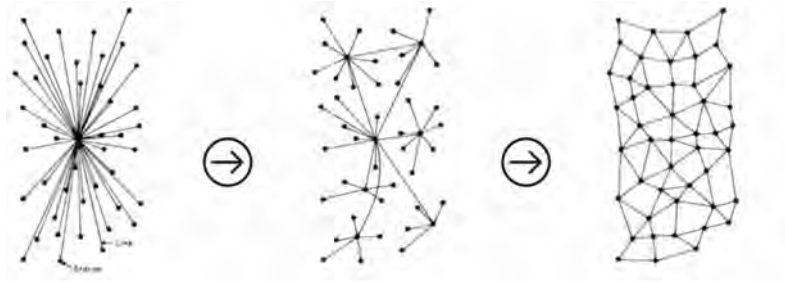
BLOCKCHAIN SHORT INTRO

ROBERTO GARCÍA
ASSOCIATE PROFESSOR,
UNIVERSITAT DE LLEIDA

Frankfurt, June 19th, 2019

DISTRIBUTED LEDGERS

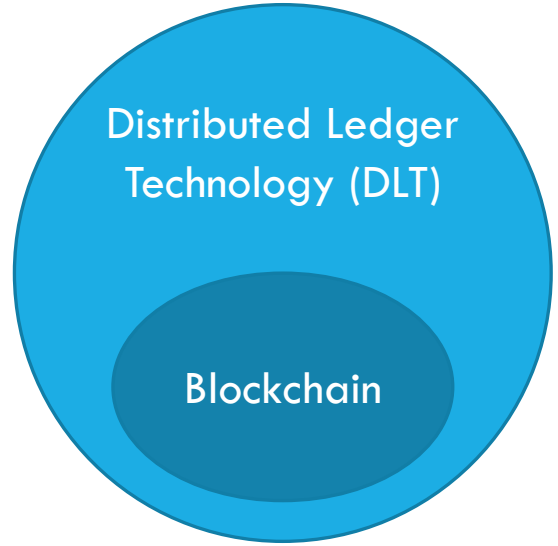


Ledgers: foundation of accounting, currently mostly centralized control

- Facilitates data integrity, faster access and control
- Updated through transactions (atomic, durable and consistent)

Distributed Ledgers (DL): no central authority controlling data

- Transactions: should be also atomic, durable and consistent
- Less performant as agreement mechanisms are required
- Replicated securely across geographic locations, to do so:
  - **Consensus** formation mechanisms, p2p protocols, crypto infrastructure

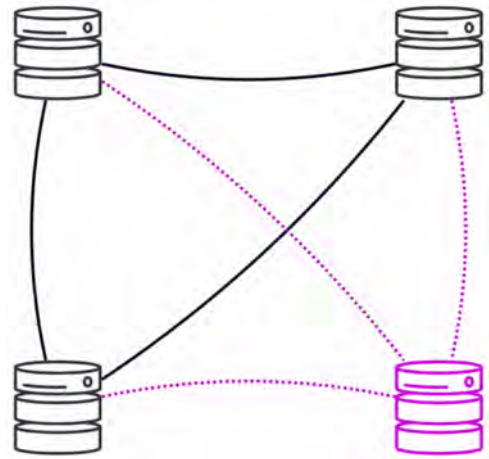


Blockchain: a type of DL which uses blocks to maintain shared state

CONSENSUS MECHANISMS

Allow secure data state update according to state transition rules

Facilitate agreement for data consistency, incentivize honesty



Approaches (all rule what is a valid block):

- Byzantine Fault Tolerance
- Proof of Work / Stake

Bizantine Fault Tolerance:

- Mathematical verification of messages
- Tolerates ~ 1/3 dishonest or absent participants
- Ex.: functioning network with 1 out of 4 faulty nodes
- Ensure a minimum number of nodes reach consensus about the sequence and result of transactions before appending them to the shared ledger

Bitcoin: consensus based on Proof of Work

How a ₿itcoin transaction works

Bob, an online merchant, decides to begin accepting bitcoins as payment. Alice, a buyer, has bitcoins and wants to purchase merchandise from Bob.

WALLETS AND ADDRESSES

Bob and Alice both have Bitcoin "wallets" on their computers.

Wallets are files that provide access to multiple Bitcoin addresses.

An address is a string of letters and numbers, such as 1HULMwZEPkjEPeCh43BeKJL1ybLCWrfDpN.

CREATING A NEW ADDRESS

Bob creates a new Bitcoin address for Alice to send her payment to.

Each address has its own balance of bitcoins.

SUBMITTING A PAYMENT

Alice tells her Bitcoin client that she'd like to transfer the purchase amount to Bob's address.

Public Key Cryptography 101

When Bob creates a new address, what he's really doing is generating a "cryptographic key pair," composed of a private key and a public key. If you sign a message with a private key (which only you know), it can be verified by using the matching public key (which is known to anyone). Bob's new Bitcoin address represents a unique public key, and the corresponding private key is stored in his wallet. The public key allows anyone to verify that a message signed with the private key is valid.

It's tempting to think of addresses as bank accounts, but they work a bit differently. Bitcoin users can create as many addresses as they wish and in fact are encouraged to create a new one for every new transaction to increase privacy. So long as no one knows which addresses are Alice's, her anonymity is protected.

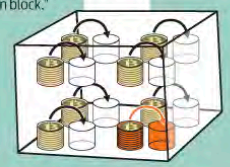
Alice's wallet holds the private key for each of her addresses. The Bitcoin client signs her transaction request with the private key of the address she's transferring bitcoins from.

Anyone on the network can now use the public key to verify that the transaction request is actually coming from the legitimate account owner.

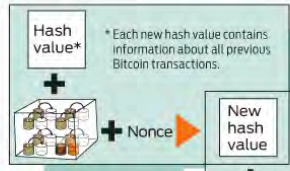
VERIFYING THE TRANSACTION

Gary, Garth, and Glenn are Bitcoin miners.

Their computers bundle the transactions of the past 10 minutes into a new "transaction block."



The miners' computers are set up to calculate cryptographic hash functions.



The mining computers calculate new hash values based on a combination of the previous hash value, the new transaction block, and a nonce.

Cryptographic Hashes

Cryptographic hash functions transform a collection of data into an alphanumeric string with a fixed length, called a hash value. Even tiny changes in the original data drastically change the resulting hash value. And it's essentially impossible to predict which initial data set will create a specific hash value.

The root of all evil	6d0a 1899 086a... (56 more characters)
The root of all evil	486c 6be4 6dde...
The root of all evil	b8db 7ee9 8392...

Nonces

To create different hash values from the same data, Bitcoin uses "nonces." A nonce is just a random number that's added to data prior to hashing. Changing the nonce results in a wildly different hash value.

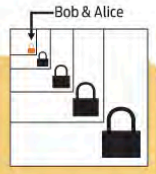
The root of all evil ???	0000 0000 0000 ...

Creating hashes is computationally trivial, but the Bitcoin system requires that the new hash value have a particular form—specifically, it must start with a certain number of zeros.

The miners have no way to predict which nonce will produce a hash value with the required number of leading zeros. So they're forced to generate many hashes with different nonces until they happen upon one that works.

Each block includes a "coinbase" transaction that pays out 50 bitcoins to the winning miner—in this case, Gary. A new address is created in Gary's wallet with a balance of newly minted bitcoins.

TRANSACTION VERIFIED

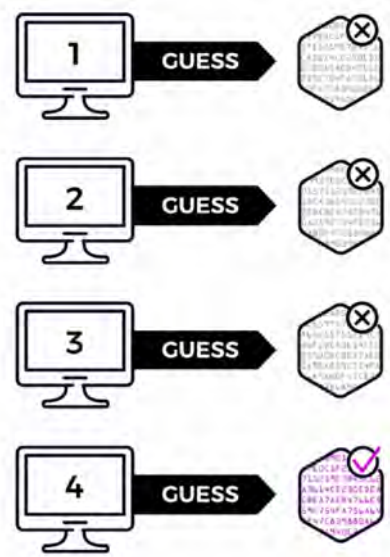


As time goes on, Alice's transfer to Bob gets buried beneath other, more recent transactions. For anyone to modify the details, he would have to redo the work that Gary did—because any changes require a completely different winning nonce—and then redo the work of all the subsequent miners. Such a feat is nearly impossible.

JOSHUA J. ROMERO, BRANDON PALACIO & KARISSA WILKER, INC.

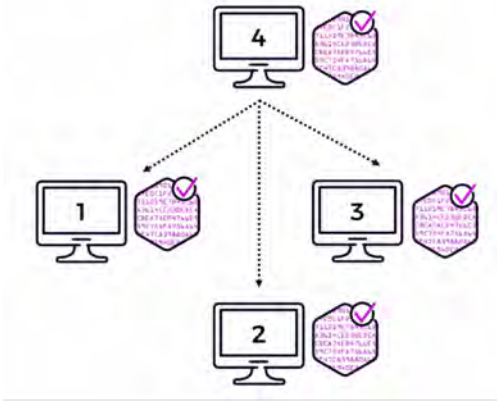
https://visual.ly/community/infographic/technology/bitcoin-infographic

CONSENSUS MECHANISM



Proof of Work (Mining)

- Participants only accept valid block if its hash is less than target number (difficulty)
- To find valid hash, miners guess and check hashes
- When found, broadcast valid block to network
- Solution includes reward for miner, incentivize honest behaviour
- Other participants verify the solution before accepting it
- Difficulty adjusted to desired block frequency
- Computationally expensive (high energy use)
  - Discourage cheating
  - Problem: low transaction volume

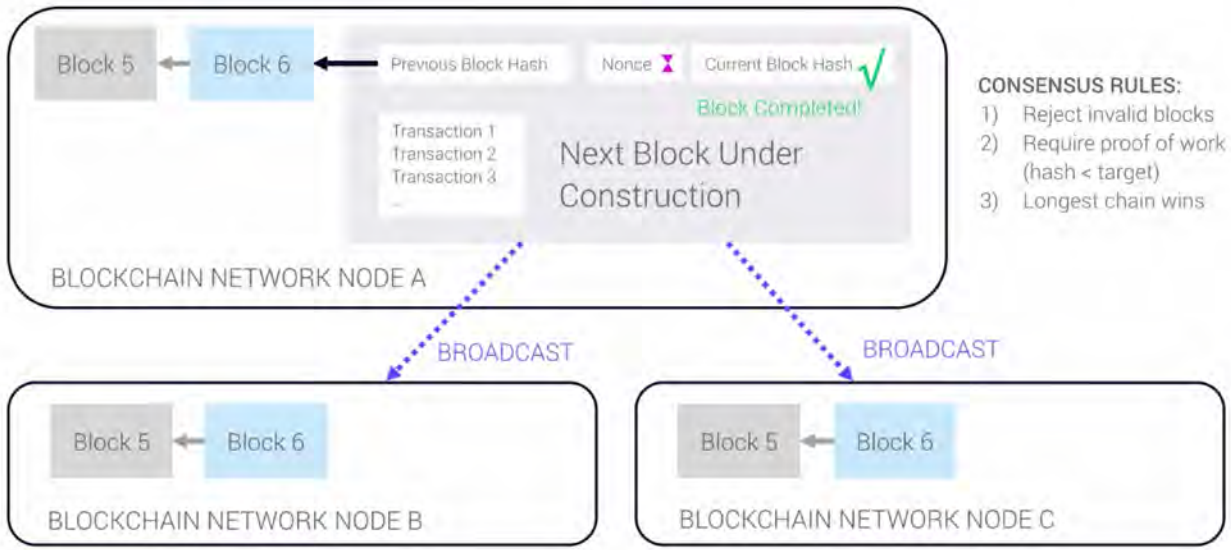


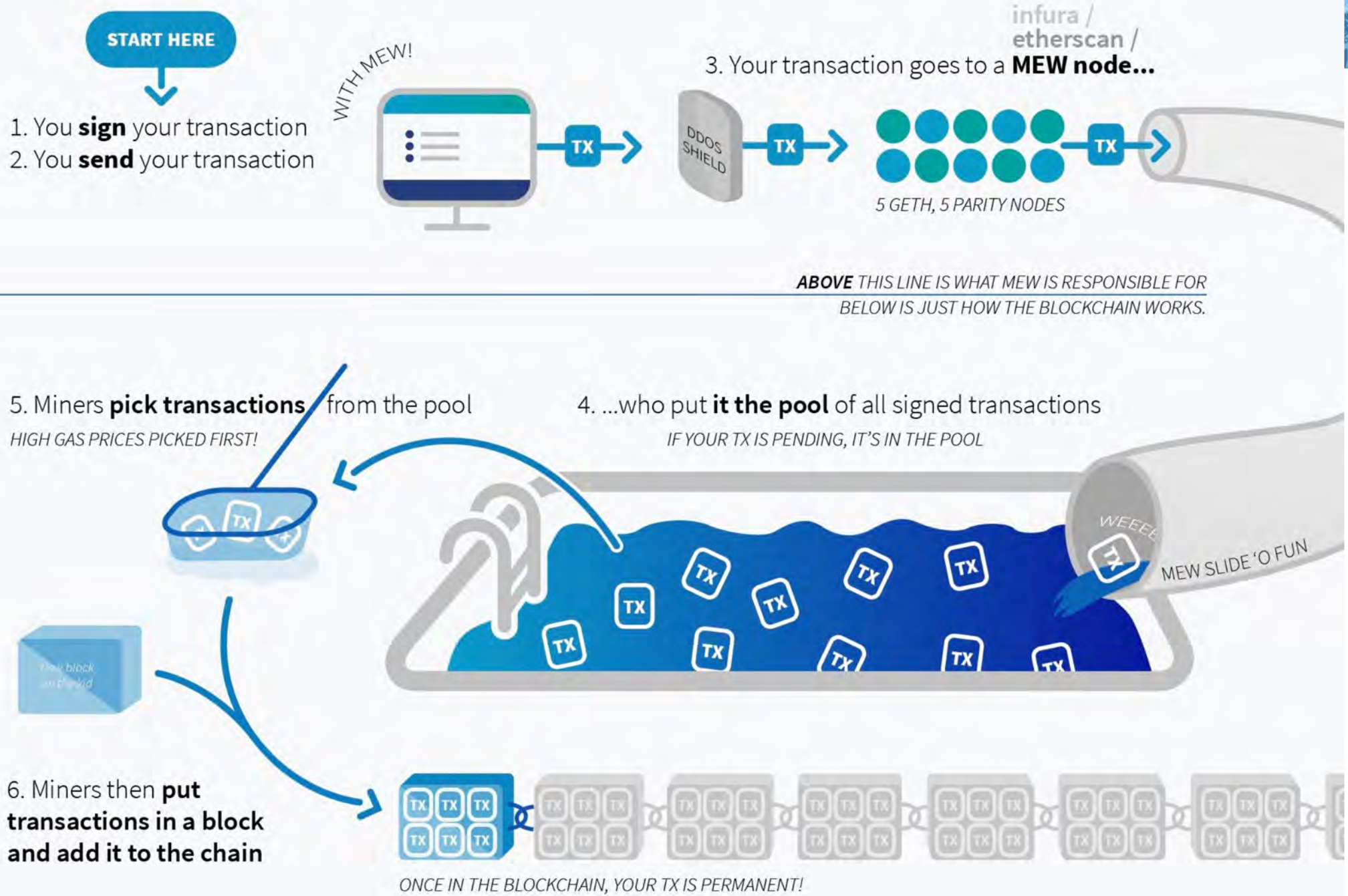
MINING (IN ETHEREUM BLOCKCHAIN)

The answer to the guess is a block, which contains:

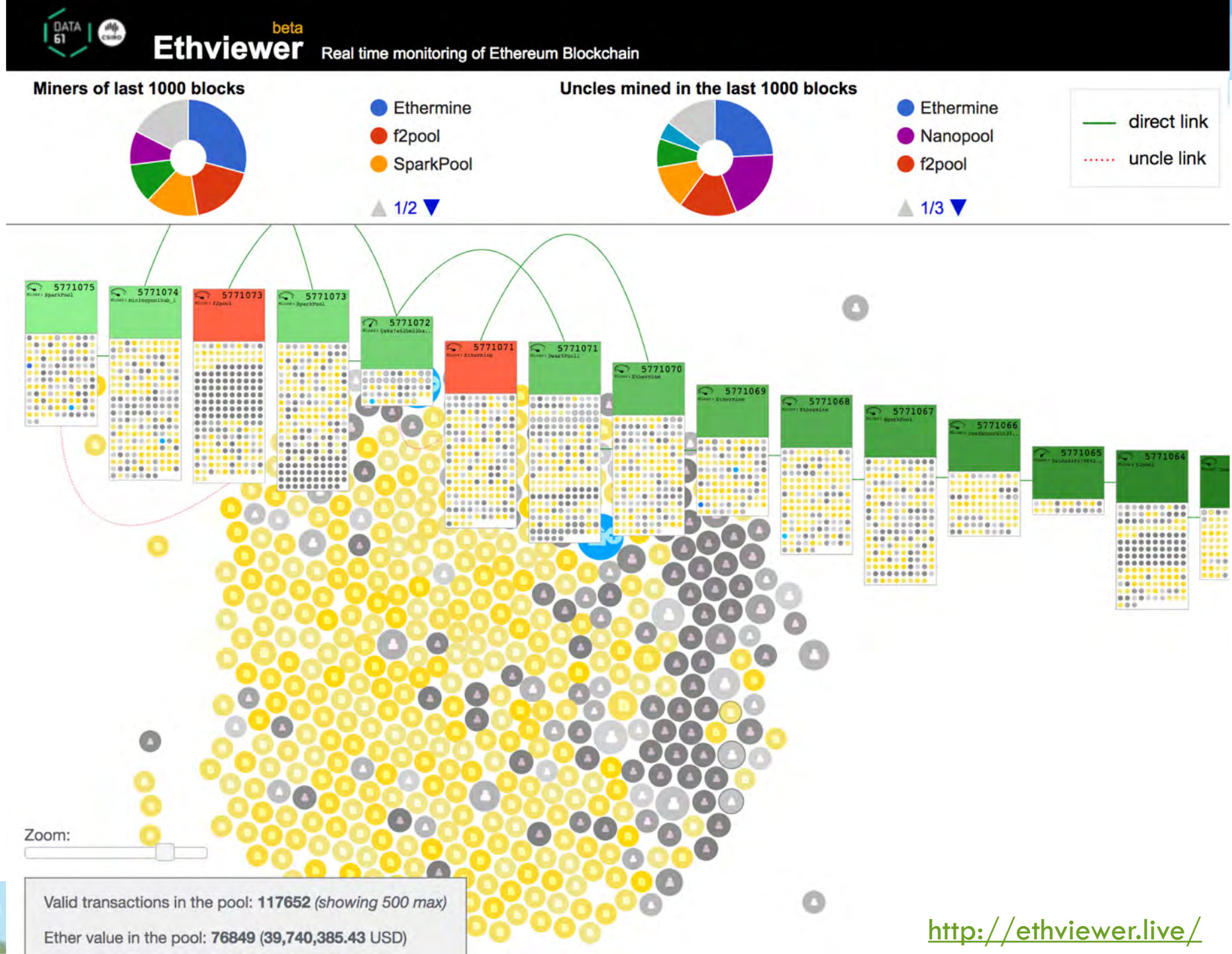
- List of transactions (updates to the system)
- Hash of the most recent block
- Random number nonce (number used once to do hash guessing game)
- Miner reward (transaction of reward to winner node, only valid if block accepted by the network)

The longest block chain is acknowledge as the correct one, and the way to secure rewards, thus incentivising coordination





https://steemit.com/ethereum/@n-ur/behind-the-scene-on-how-myetherwallet-works-simple-illustration



http://ethviewer.live/

PUBLIC AND PRIVATE BLOCKCHAINS

Classification depending on **permission levels**

Public or permission-less: allow anyone to join as a trust-less participant, anyone can read/write data

- Pseudo-anonymous transactions, addresses of participants are public but not necessary linked to an identity
- Transaction processors must invest financially to prevent fraud
  - For instance through Proof of Work
  - Incentivised by direct economic reward (cryptocurrency)
- Cost digital currency to process transactions
- Censorship resistant (anyone can access and record transactions)
- Examples: Ethereum (public) or Bitcoin

PUBLIC AND PRIVATE BLOCKCHAINS

Permissioned or Consortium: only verified participants are allowed

- Block validator identities are known allowing punishment of dishonest actors
- Different consensus mechanisms are possible that achieve higher transaction throughput than in public ones
- Example: **Alastria** (https://alastria.io)

ALASTRIA

Private or sandboxes: for rapid application development, instant deployment and single-enterprise deployment solutions

- Prototyping and learning

BLOCKCHAIN PLATFORMS

	ETHEREUM	HYPERLEDGER	BITCOIN	CORDA	RIPPLE
Consensus Algorithm	Proof of Work (Proof of Stake)	PBFT	Proof of Work	BFT or RAFT	Ripple Protocol
Network Size	Global	Limited	Global	Limited	Limited
Protocol Implementation(s)	Go, C++, Python, Haskell, Java, Rust, Ruby, JavaScript	Go, Java	C++, Java	Kotlin, Java	N/A
Consortia	Enterprise Ethereum Alliance	Hyperledger Foundation	N/A	R3	N/A
Native Digital Currency	Yes	No	Yes	No	Yes
Built-in Smart Contracts	Yes	Yes	No	Yes	No
Blocks	Yes	Yes	Yes	No	No
Mining	Yes	No	Yes	No	No
Public / Private Interoperability	Public, Private / Permissioned, Permissionless	Private / Permissioned	Public / Permissionless	Private / Permissioned	Private / Permissioned
Developer Community Strength	30X	X	N/A	N/A	N/A

Source: Consensys Academy

THANK YOU FOR YOUR ATTENTION

QUESTIONS?

ROBERTO GARCÍA
http://rhizomik.net/~roberto/

Frankfurt, June 19th, 2019