# Blockchain: what it is and why it matters

## Progetto di Ricerca di Rilevante Interesse Nazionale - PRIN 2020 "Nirvana - Noninterference and Reversibility Analysis In Private Blockchains" - N. 20202FCJMH

Laura Ricci

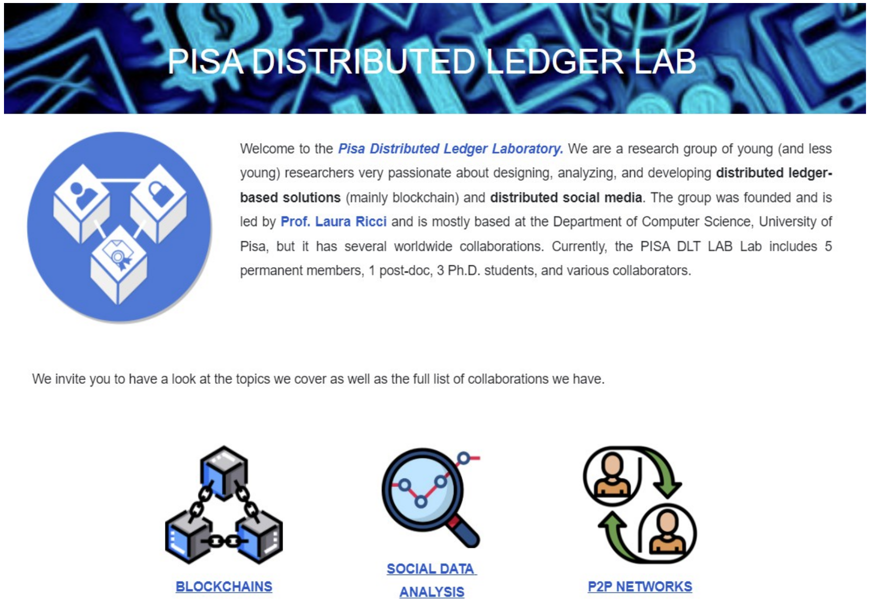
Dipartimento di Informatica
Università degli Studi di Pisa

28th of April 2022

# THE PISA DISTRIBUTED LEDGER LAB

- Permanent/semi-permanent position
  - *Laura Ricci* associate professor
  - Fabrizio Baiardi, full professor
  - *Paolo Mori*, IIT CNR, Pisa
  - *Barbara Guidi*, RTD-B
  - *Damiano Di Francesco Maesa*, RTD-A
  - *Andrea Michienzi, RTD-A*
  - *Andrea De Salve*, ISASI, Lecce
- PhD
  - *Andrea Lisi*
  - *Matteo Loporchio*
  - *Domenico Tortola*
- Collaboration
  - *Andrea Marino*, University of Florence
  - *Anna Bernasconi,* University of Pisa
  - *Roberto Di Pietro*, Hamad Bin Kalifa University, Quatar
  - *Nishanth Sastry*, University of Surrey

https://sites.google.com/unipi.it/pisadltlaboratory



Welcome to the *Pisa Distributed Ledger Laboratory.* We are a research group of young (and less young) researchers very passionate about designing, analyzing, and developing **distributed ledger-based solutions** (mainly blockchain) and **distributed social media**. The group was founded and is led by **Prof. Laura Ricci** and is mostly based at the Department of Computer Science, University of Pisa, but it has several worldwide collaborations. Currently, the PISA DLT LAB Lab includes 5 permanent members, 1 post-doc, 3 Ph.D. students, and various collaborators.

We invite you to have a look at the topics we cover as well as the full list of collaborations we have.

# PISA DISTRIBUTED LEDGER LABORATORY

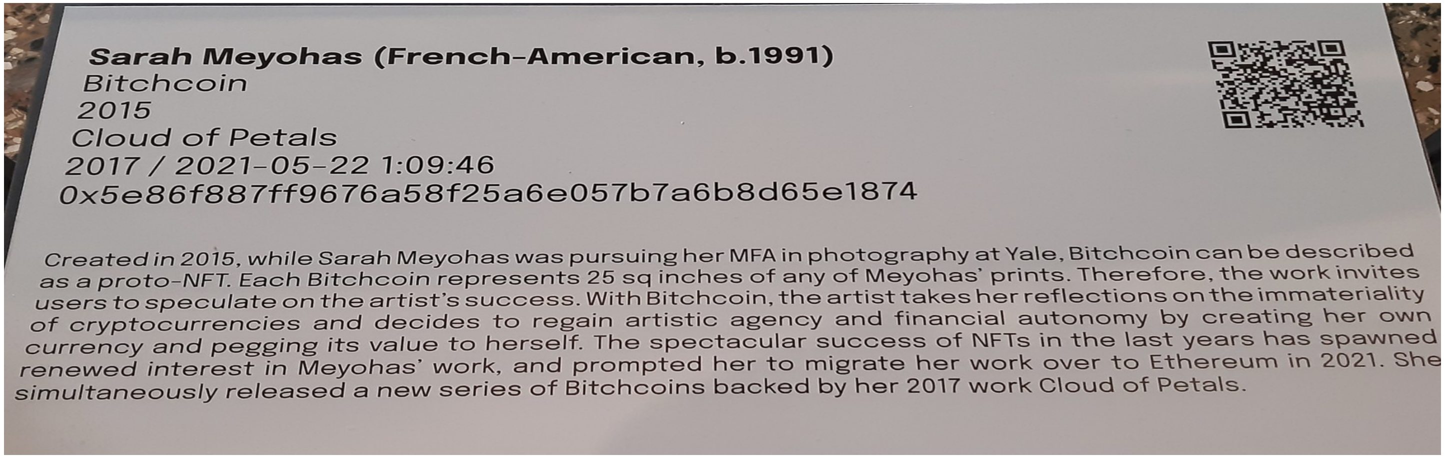
- what are we doing? We work on different aspects of blockchain
  - privacy: Zero Knowledge Proofs
  - data reliability: oracles
  - scalability: off-chain computation, light weight channels, side chains
  - applications
    - Self Sovereign identity (SSI)
    - access control systems
  - transaction analysis
    - scam detection
- references
  - https://sites.google.com/unipi.it/pisadltlaboratory
  - e-mail: laura.ricci@unipi.it

# BLOCKCHAIN : HYPE OR REALITY?

*Singularity*: un tuffo nell'arte decentralizzata e nelle opere d'arte NFT

Venezia, Biennale Arte 2022

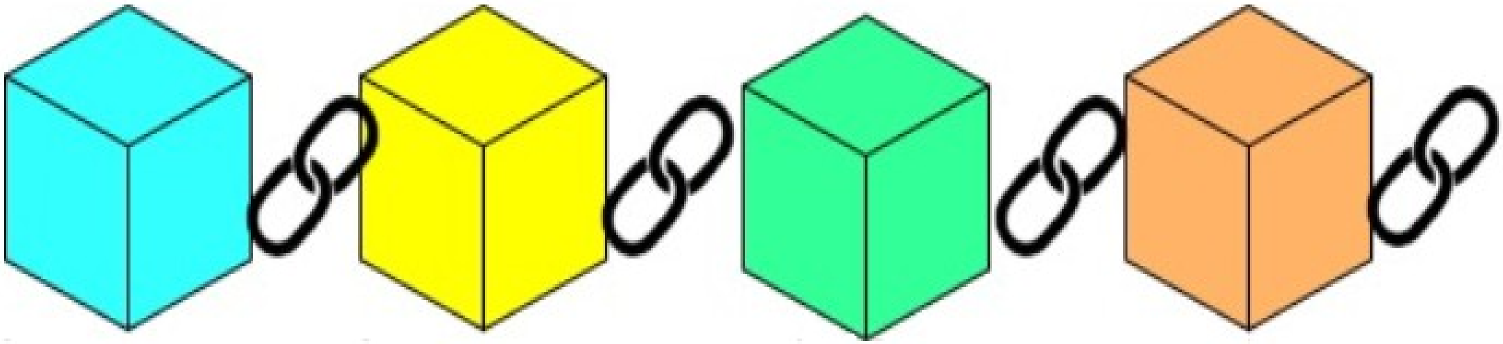
27 Aprile 2022



**Sarah Meyohas (French-American, b.1991)**
Bitchcoin
2015
Cloud of Petals
2017 / 2021-05-22 1:09:46
0x5e86f887ff9676a58f25a6e057b7a6b8d65e1874

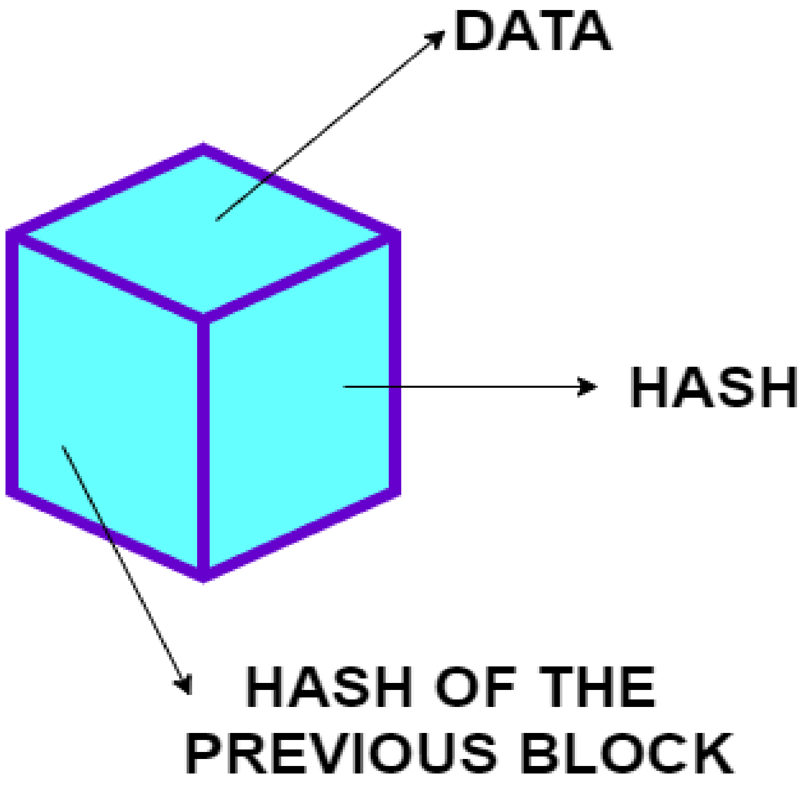
Created in 2015, while Sarah Meyohas was pursuing her MFA in photography at Yale, Bitchcoin can be described as a proto-NFT. Each Bitchcoin represents 25 sq inches of any of Meyohas' prints. Therefore, the work invites users to speculate on the artist's success. With Bitchcoin, the artist takes her reflections on the immateriality of cryptocurrencies and decides to regain artistic agency and financial autonomy by creating her own currency and pegging its value to herself. The spectacular success of NFTs in the last years has spawned renewed interest in Meyohas' work, and prompted her to migrate her work over to Ethereum in 2021. She simultaneously released a new series of Bitchcoins backed by her 2017 work Cloud of Petals.

# BLOCKCHAIN "AT A GLANCE"

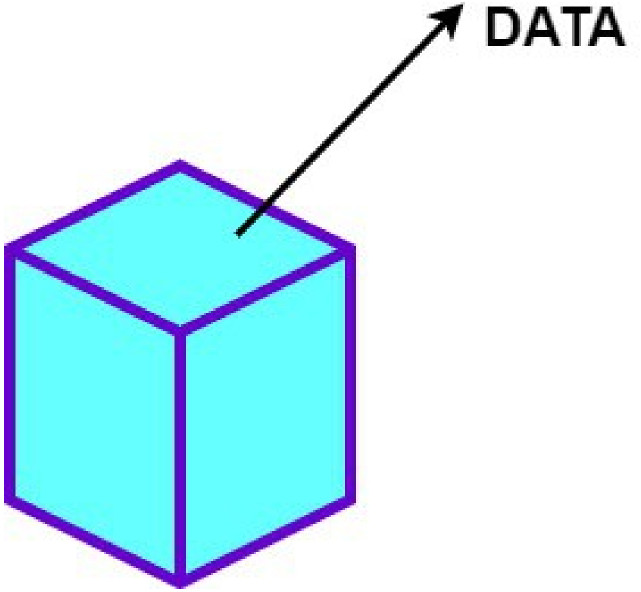


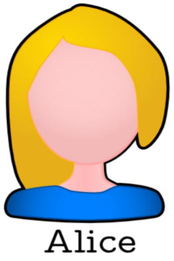
- a ledger which is replicated among the nodes of a peer-to-peer network
- all the nodes have the same replica of the ledger
- is immutable
  - benefits of the tamper freeness property
- may act like a notary

# LOOKING INSIDE A BLOCK



# LOOKING INSIDE A BLOCK: WHICH DATA?





FROM

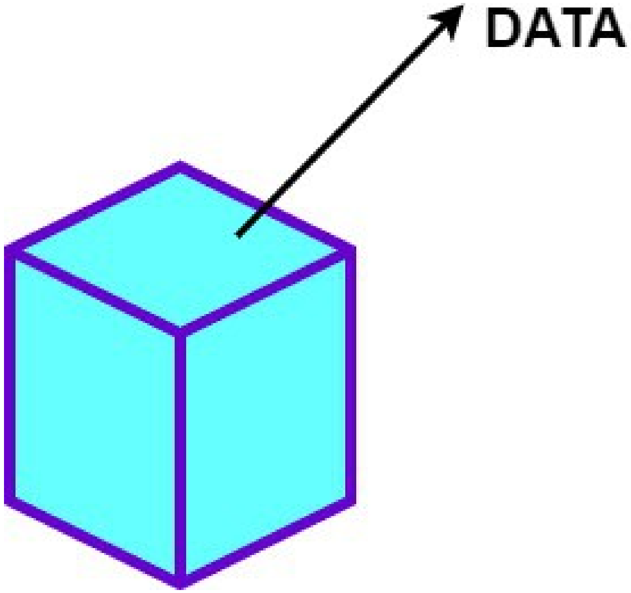
TO

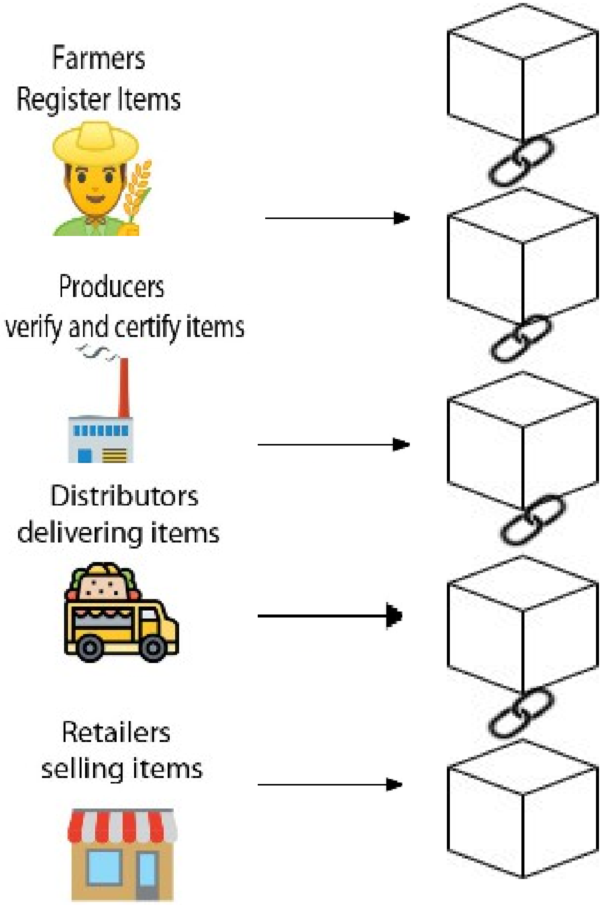
AMOUNT

for cryptocurrencies

- data are transactions which transfer an amount of money between two entities

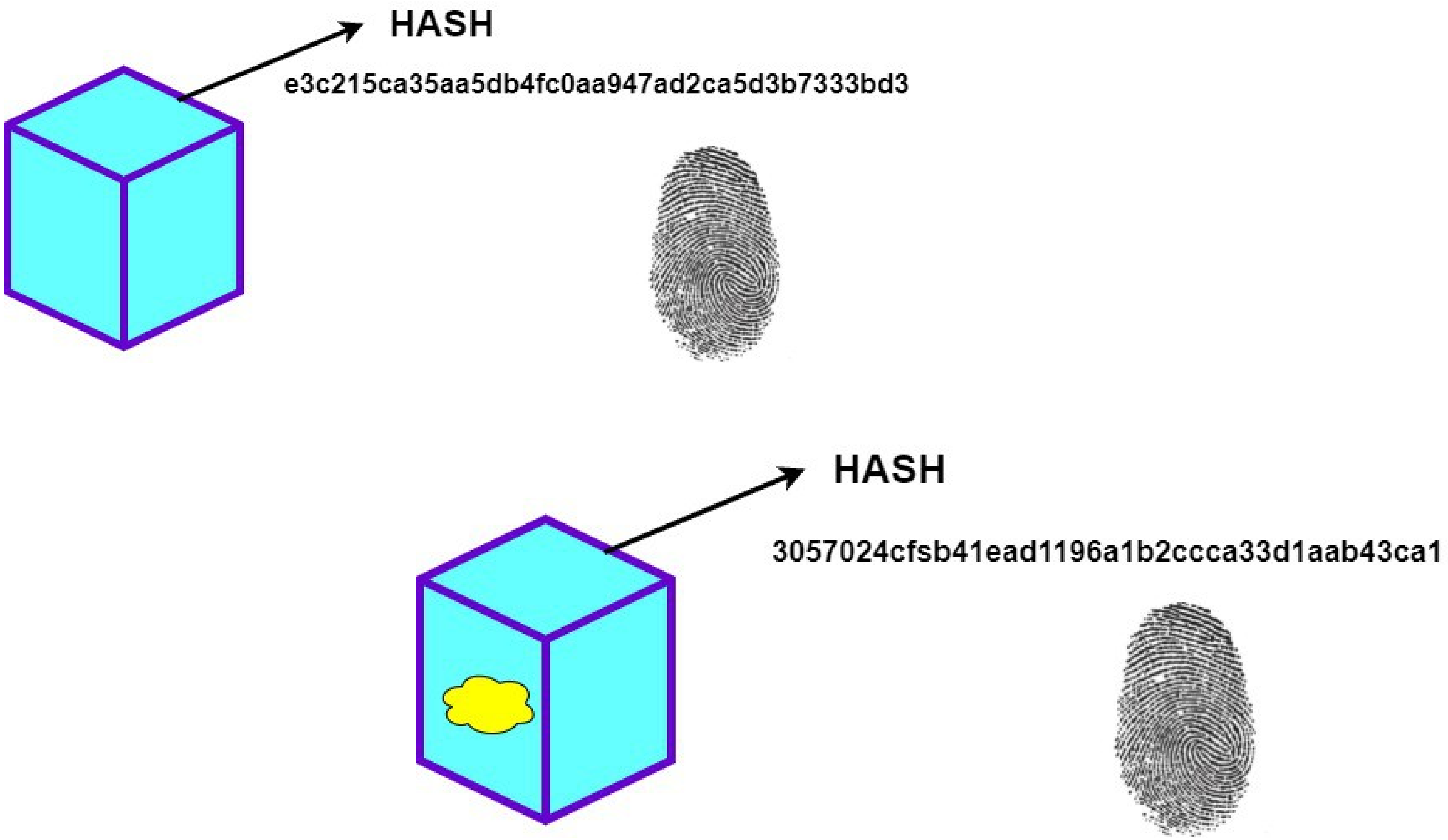
# LOOKING INSIDE A BLOCK: WHICH DATA?





- many further scenarios where data may be
  - data collected in a supply chain
  - a contract
  - an intellectual property licence
  - the temperature detected by a sensor inside a truck carrying drugs,....

# LOOKING INSIDE A BLOCK: HASH



HASH

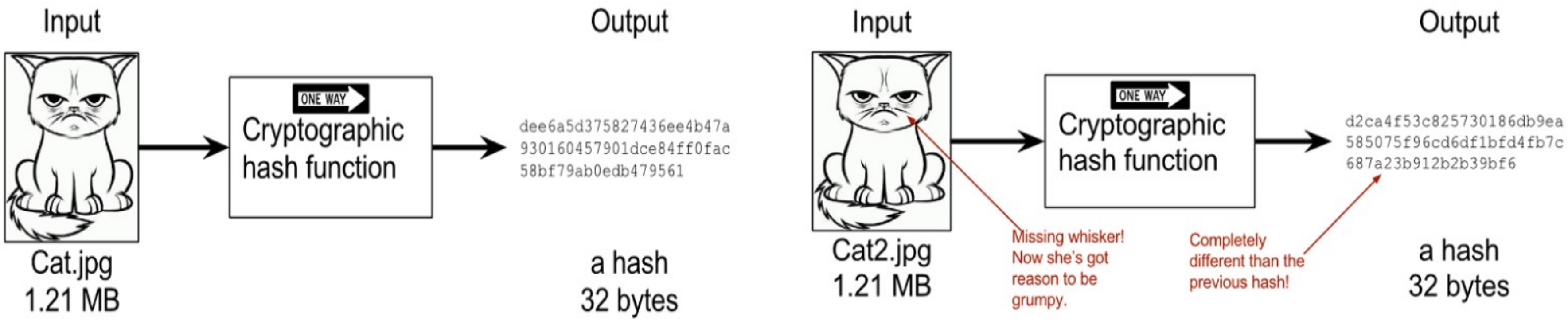
e3c215ca35aa5db4fc0aa947ad2ca5d3b7333bd3

HASH

3057024cfsb41ead1196a1b2ccca33d1aab43ca1

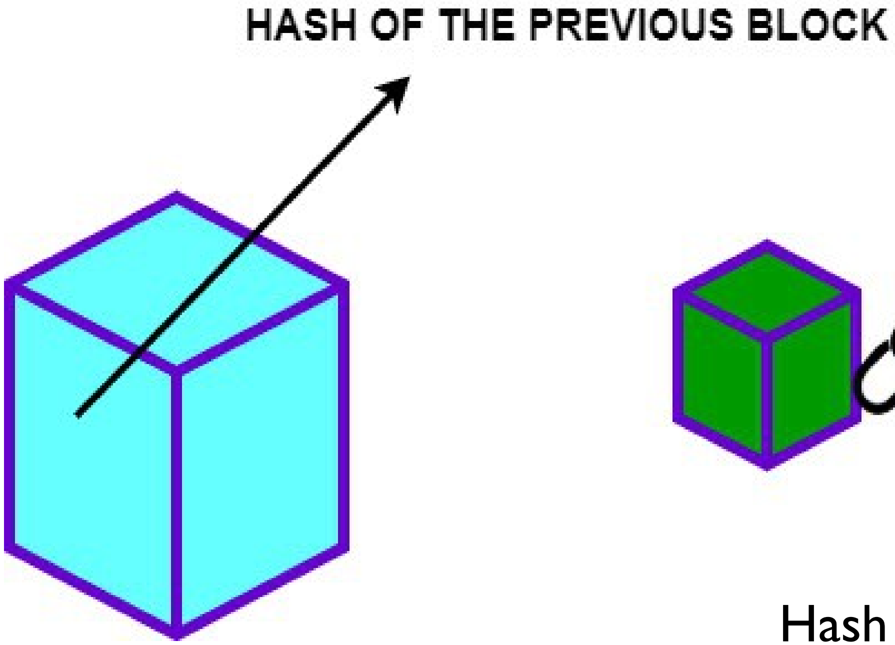
# CRYPTOGRAPHIC HASH FUNCTIONS

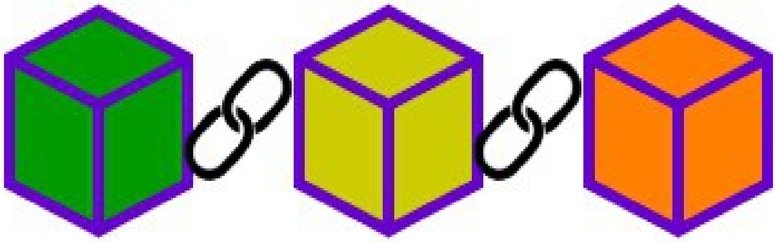
- a mathematical function pairing to each input data a “fingerprint” of fixed length



- input data : any length, any type
- output data:
  - fixed-length sequence of characters
  - if input is slightly changed, output is completely changed
- one-way: it is computationally hard to go from the hash to the input
- collision freeness
- and other properties

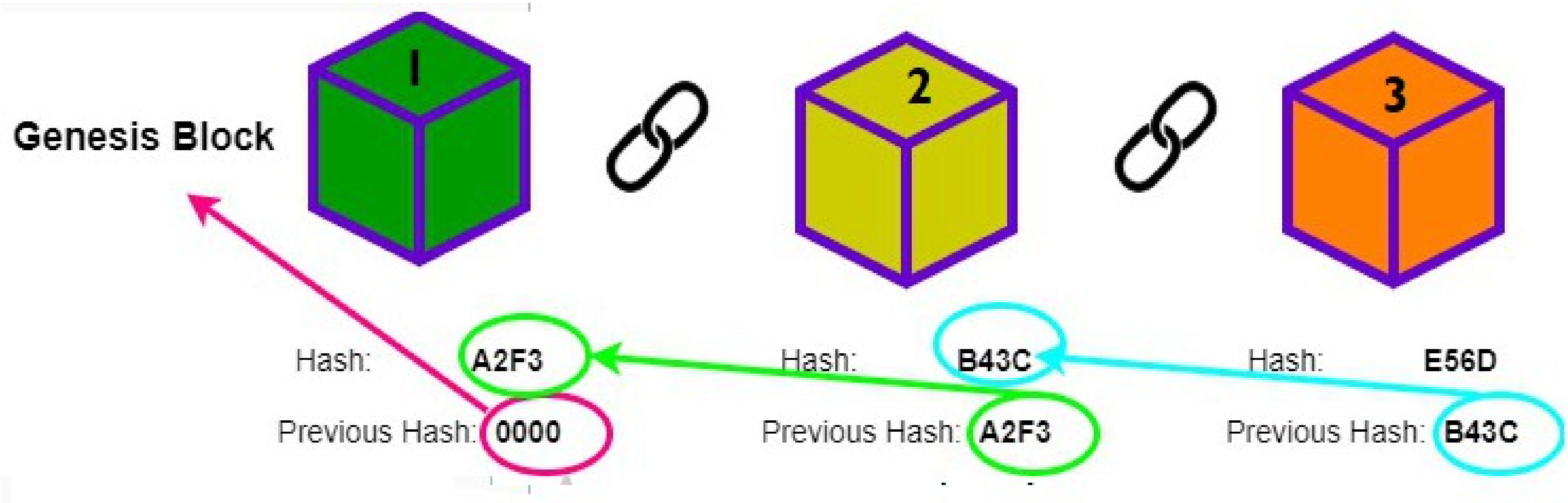
# LOOKING INSIDE A BLOCK: HASH POINTES



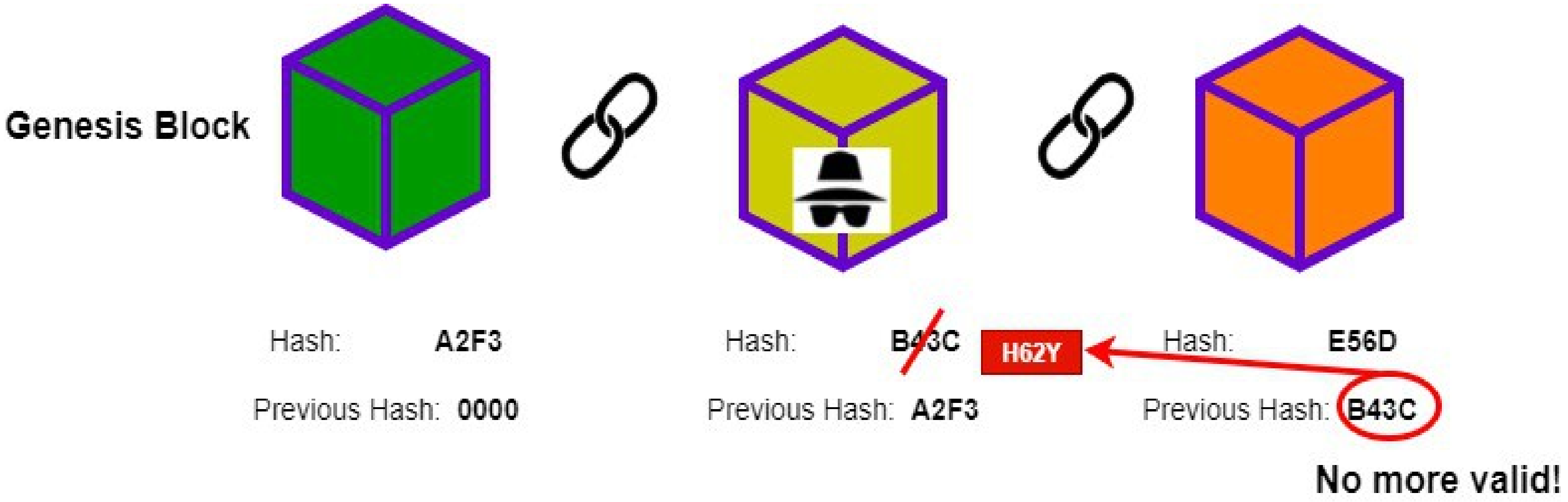


Hash pointers create the chain

# THE BLOCKCHAIN

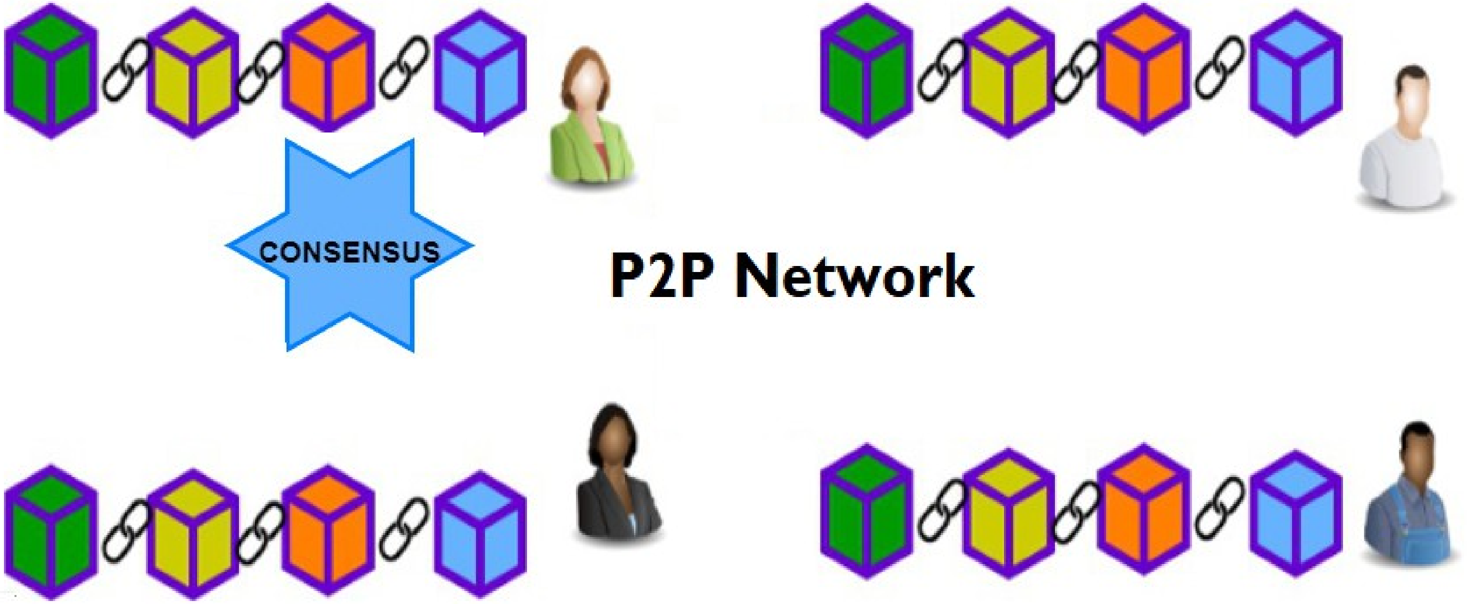


# THE BLOCKCHAIN: TAMPER FREENESS



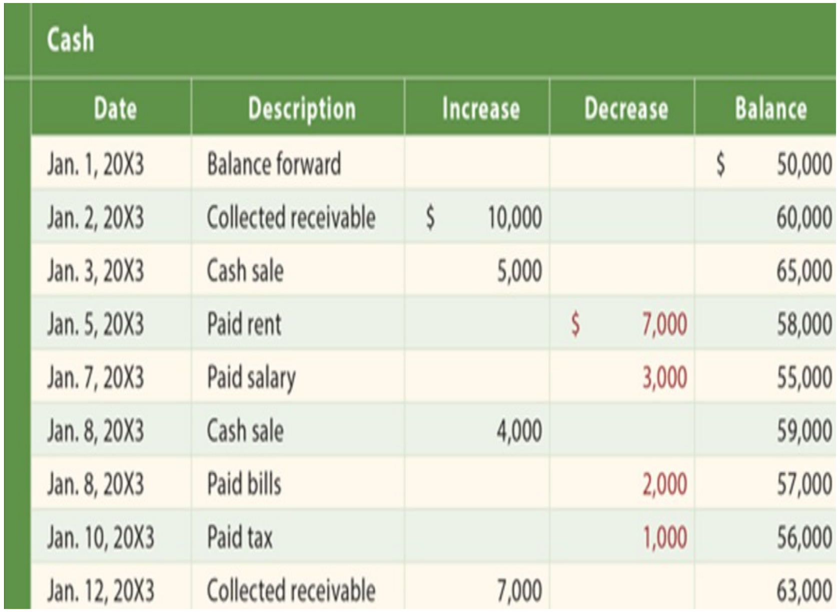
- changing one hash caused changing the hash of the following blocks
- this does not only imply to recompute some hashes, but to find a value, that combined with the new hash solves the Proof of Work
    - other blockchains may use different mechanisms

# THE BLOCKCHAIN: CONSENSUS



# ABSTRACTING THE BLOCKCHAIN: THE LEDGER

- a ledger
  - like a bulletin storing operations consistently replicated on the nodes of a P2P network
- which properties needed for a ledger?
  - *append-only* list of events
  - *tamper-proof*
    - immutability, auditability
  - *everyone agrees on content*
    - consensus
- not just financial!
  - any application which needs a log of events

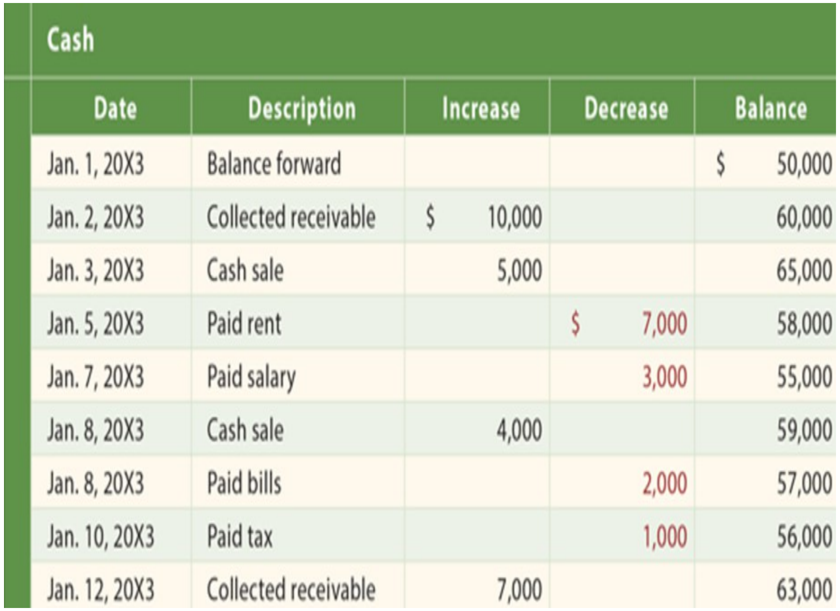


Cash

| Date | Description | Increase | Decrease | Balance |
|---|---|---|---|---|
| Jan. 1, 20X3 | Balance forward | | | $ 50,000 |
| Jan. 2, 20X3 | Collected receivable | $ 10,000 | | 60,000 |
| Jan. 3, 20X3 | Cash sale | 5,000 | | 65,000 |
| Jan. 5, 20X3 | Paid rent | | $ 7,000 | 58,000 |
| Jan. 7, 20X3 | Paid salary | | 3,000 | 55,000 |
| Jan. 8, 20X3 | Cash sale | 4,000 | | 59,000 |
| Jan. 8, 20X3 | Paid bills | | 2,000 | 57,000 |
| Jan. 10, 20X3 | Paid tax | | 1,000 | 56,000 |
| Jan. 12, 20X3 | Collected receivable | 7,000 | | 63,000 |

# ABSTRACTING THE BLOCKCHAIN: THE LEDGER

- a write-only, decentralized, state machine that is maintained by untrusted actors, secured by economic incentive
- cannot delete data
- cannot be shut down or censored
- supports defined operations agreed upon by participants
- participants may not know each other (public)
- in actors best interest is to play by the rules



Cash

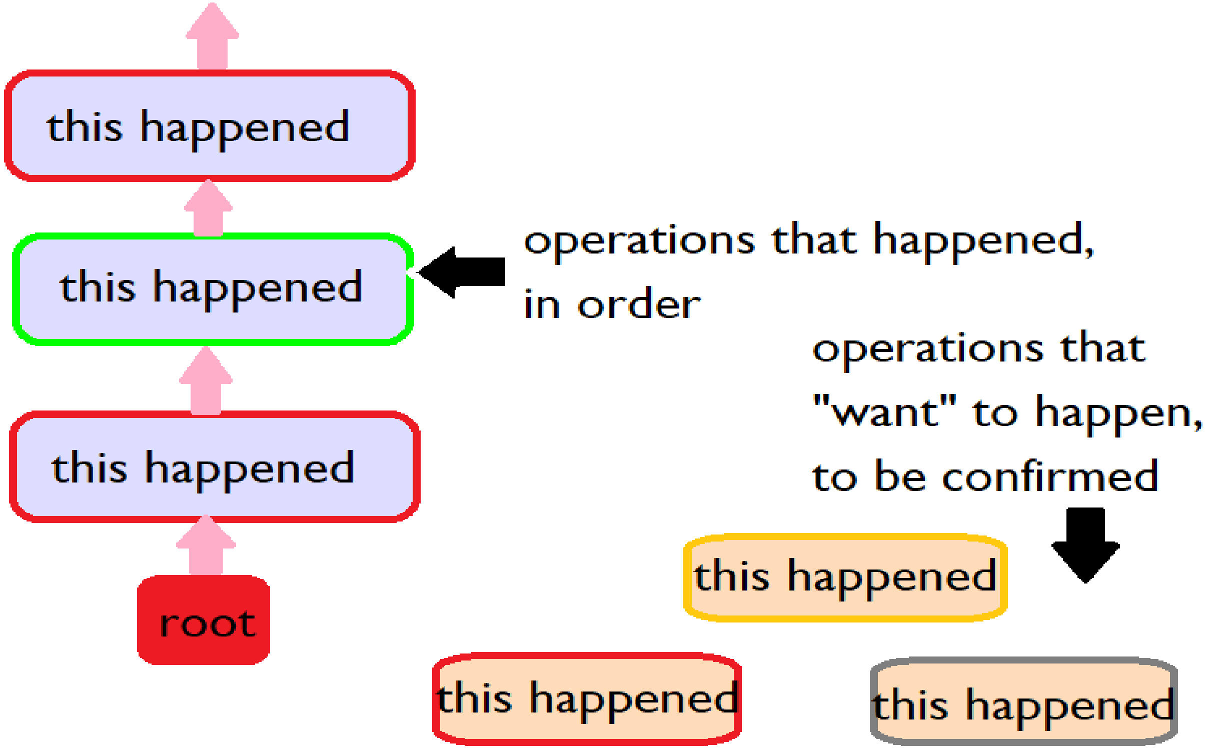
| Date | Description | Increase | Decrease | Balance |
|---|---|---|---|---|
| Jan. 1, 20X3 | Balance forward | | | $ 50,000 |
| Jan. 2, 20X3 | Collected receivable | $ 10,000 | | 60,000 |
| Jan. 3, 20X3 | Cash sale | 5,000 | | 65,000 |
| Jan. 5, 20X3 | Paid rent | | $ 7,000 | 58,000 |
| Jan. 7, 20X3 | Paid salary | | 3,000 | 55,000 |
| Jan. 8, 20X3 | Cash sale | 4,000 | | 59,000 |
| Jan. 8, 20X3 | Paid bills | | 2,000 | 57,000 |
| Jan. 10, 20X3 | Paid tax | | 1,000 | 56,000 |
| Jan. 12, 20X3 | Collected receivable | 7,000 | | 63,000 |

# BLOCKCHAIN: BASIC TECHOLOGICAL TOOLS

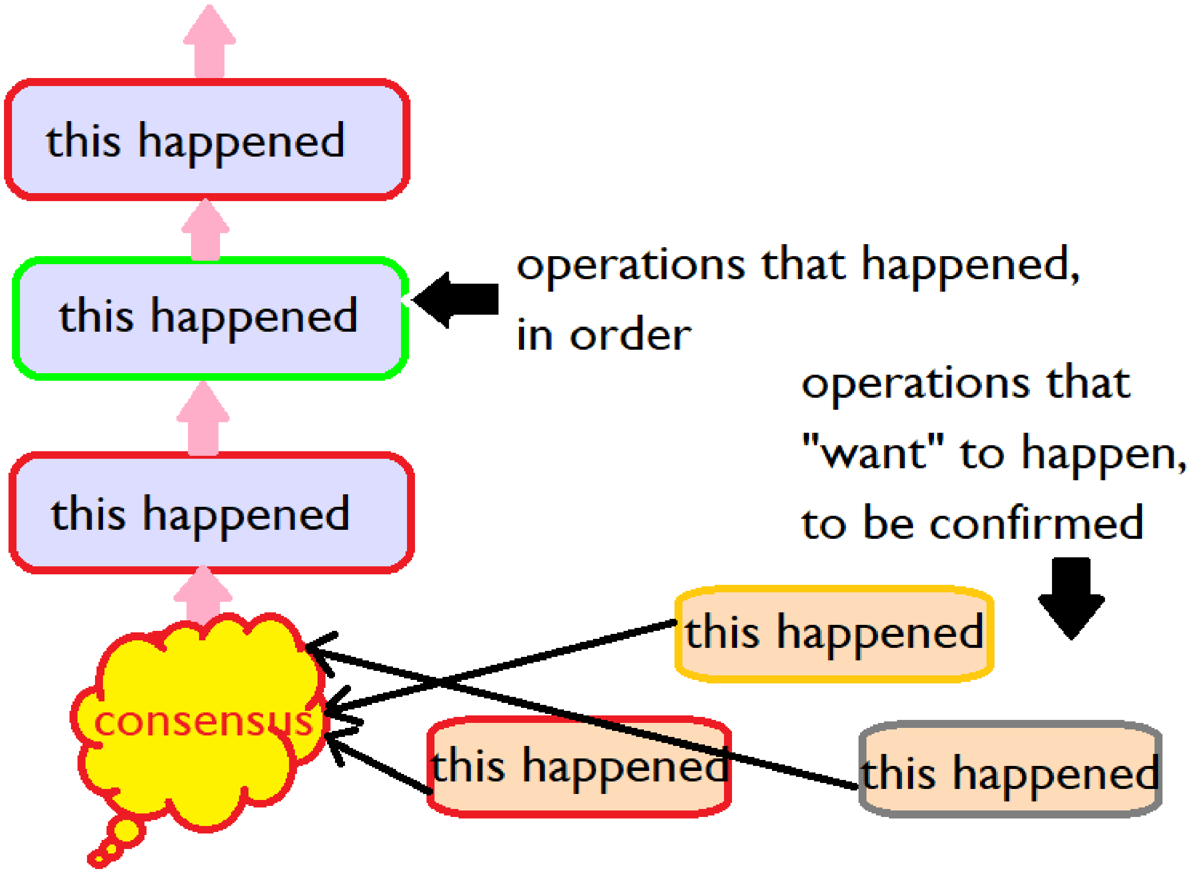
- *Cryptographic hash functions* (e.g. hash chains of data transactions)
  - provide tamper-resistant immutability
- *Distributed consensus* amongst mutually trusting or distrusting replica
  - provides integrity and decentralized control
- *Replication* (e.g. full copies stored everywhere)
  - provides availability
- *Digital signatures* (e.g. public-key cryptography)
  - provide ownership
- these are the basic tools, but other tools are needed and deserve interest
  - *cryptography:* zero-knowledge, multi party, verifiable random functions, authenticated data structures
  - *formal verification techniques:* smart contracts security
  - *performance models*

# THE LEDGER AS A BLOCKCHAIN

- if the ledger is organized as a list of blocks
  - call it a blockchain
  - but other structures are possibles! for instance, graphs...
- let us do a simplification: blocks contain single operations (not true for Bitcoin or Ethereum)



# ADDING ENTRIES TO THE LEDGER

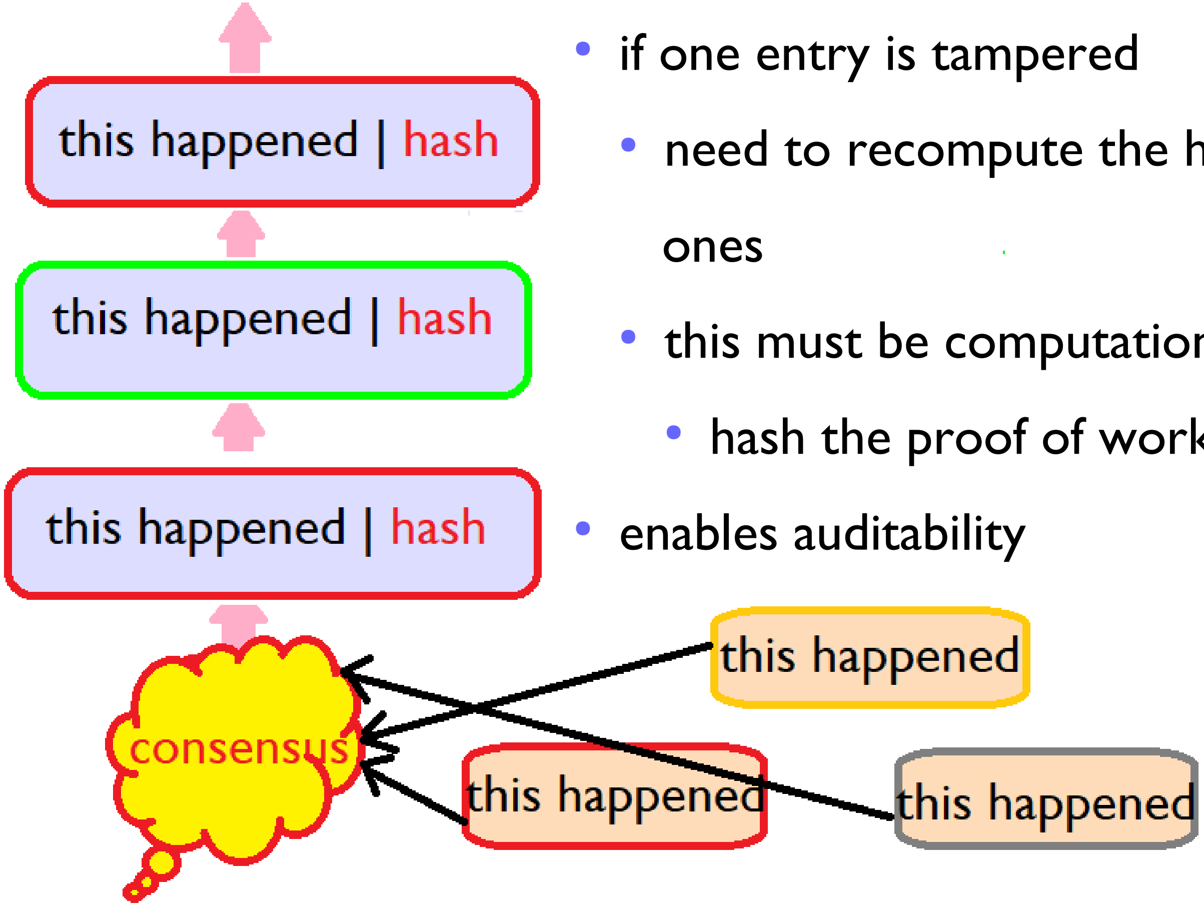


consensus is the mechanism which defines

- who decides which operation will be added to the blockchain
- which operation among those to be confirmed, will be added

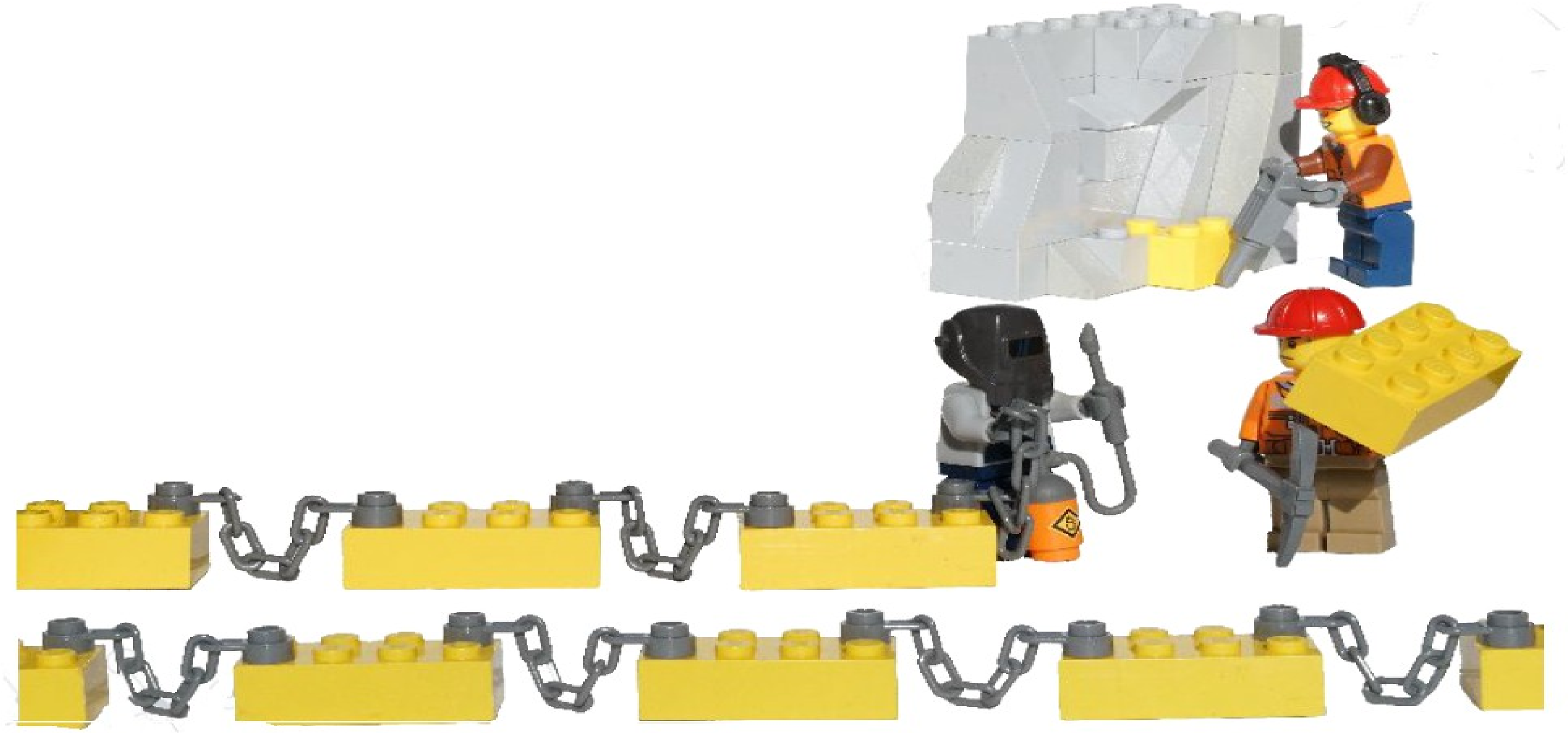
# TAMPER FREENESS THROUGH HASH

- compute the hash of each entry (block)
- store in each entry the predecessor's hash
- if one entry is tampered
  - need to recompute the hash of all the following ones
  - this must be computationally hard
    - hash the proof of work with the block
- enables auditability



# CONSENSUS IN BITCOIN: MINING

- Proof-of-work is an implementation of consensus realized in Bitcoin
  - a lottery
  - only who wins (ad to win is complex), can append the next block to the blockchain
  - called mining because the winner is rewarded



# BITCOIN CONSENSUS FROM NAKAMOTO

- let us suppose, for the moment, that:
  - it is possible to pick a random node in the network
    - like picking a random token in a lottery
  - at least 51% of the time, this process will pick an honest node.
- the consensus protocol:
  - at each round: select a node at random
  - that node unilaterally proposes, without contacting other nodes, the next block of transactions to be inserted in the ledger (from the unconfirmed transactions)
  - that node broadcasts it in the peer-to-peer network
  - all the nodes check the validity of the block and update their blockchain with the new chosen block

# RANDOM NODE SELECTION

- how to select a random node at each round?
- the key idea: the probability to select a node must be proportional to the amount of resource has, a resource which  is hard to monopolize
- in Bitcoin the probability to be selected is proportional to the  computational power and selection is done on the basis of the Proof of work

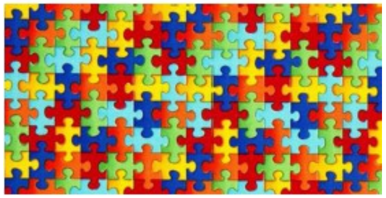
Proof of work

- nodes which try to solve the proof of work are called miners and the whole validation process is called mining

# PROOF OF WORK

- based on cryptographic puzzles that
  - can be solved
  - require a considerable effort which cannot be short-circuited
  - it must be possible to verify the effort made to solve a PoW in a easy way
    - verification requires less time with respect to the time needed to conduct the PoW

Hard to find solution

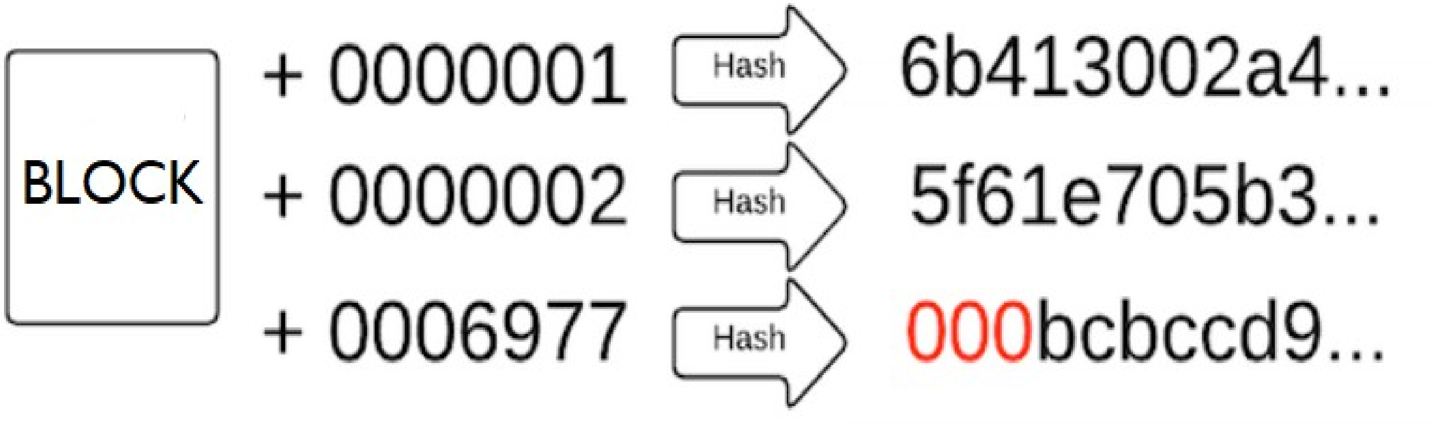


Easy to verify

- winner of the lottery decides which is the next node of the blockchain
- like a  lottery to choose which node will decide the next block
  - tickets of the lottery are very expensive (proof of work)
  - winner of the lottery is paid when other nodes endorse validity
  - give incentives for well behaviour

# PROOF OF WORK: THE CRYPTOGRAPHIS PUZZLE

- find a value X and hash (block || X) : the result must be less than a threshold fixed value
- X is said a *nounce*

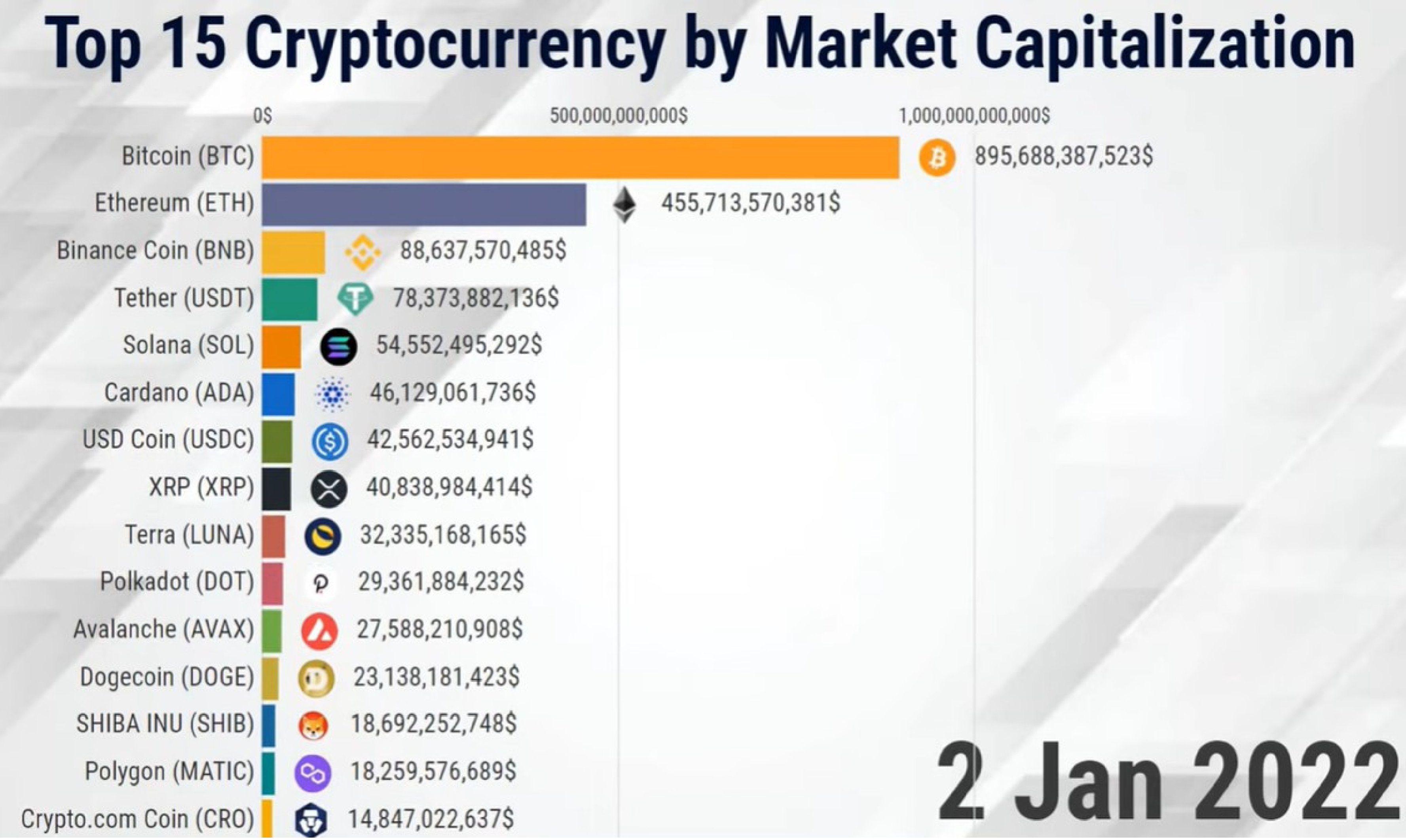


- actually only the header of the block is hashed
- needs large computational resources

# POW DISADVANTAGES

- the cost of Bitcoin mining is too high
  - energy waste
  - mining pools control large portion of the Bitcoin blockchain
    - make blockchain not fully distributed
- other solutions
  - employ energy than cannot be stoked
  - alternative consensus algorithm
    - *proof of stake* (Algorand, Cardano (Ourboros), Solana, ..)
    - *delegated Proof of Stake* (Steemit, EOS,...)
    - *byzantine consensus* (Hypeledger,..)

# NOT ONLY BITCOIN AND NOT ONLY POW!!


Top 15 Cryptocurrency by Market Capitalization

2 Jan 2022

# OTHER CONSENSUS MECHANISMS:PROOF OF STAKE

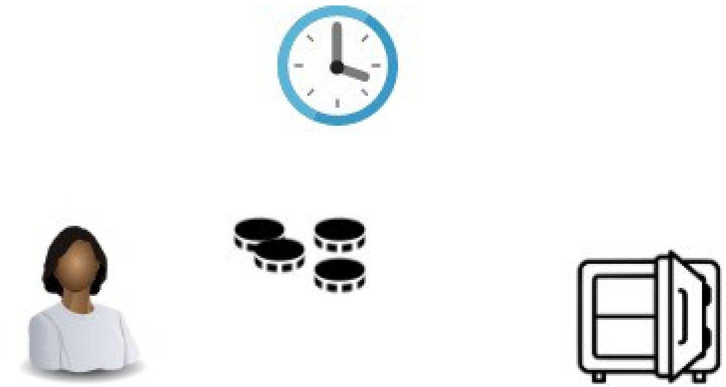
- an election process in which one node is randomly chosen to validate the next block
- no miners, no mining                    instead  validators, minting or forging

- but choose is not completely random
  - to become a validator a node has to deposit an amount of coins as stake
    - a security deposit
  - the size of the stake defines the probability to be chosen as a validator to forge next block

# OTHER CONSENSUS MECHANISMS:PROOF OF STAKE

- the chosen node has to check if all the transactions within the block are valid
  - as a reward, the node receives the fees associated to each transaction
- validator lose a part of their stake if they validate fraudlent transactions
  - if the stake is higher than the total obtained by the fees, the validator is not incentivized to cheat
- if a node stop doing the validator, receives the stake + the transaction fees it got, but only after a certain period
- less energy
- no expensive equipments, more people

  are encouraged to participate
  - incentive for the decentralization

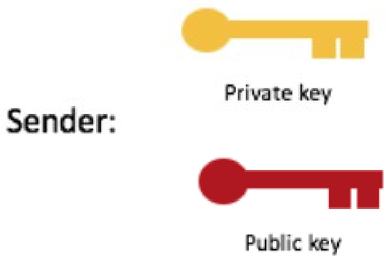


# IS IT ALL? NO, PROOF OF OWNERSHIP IS ALSO NEEDED

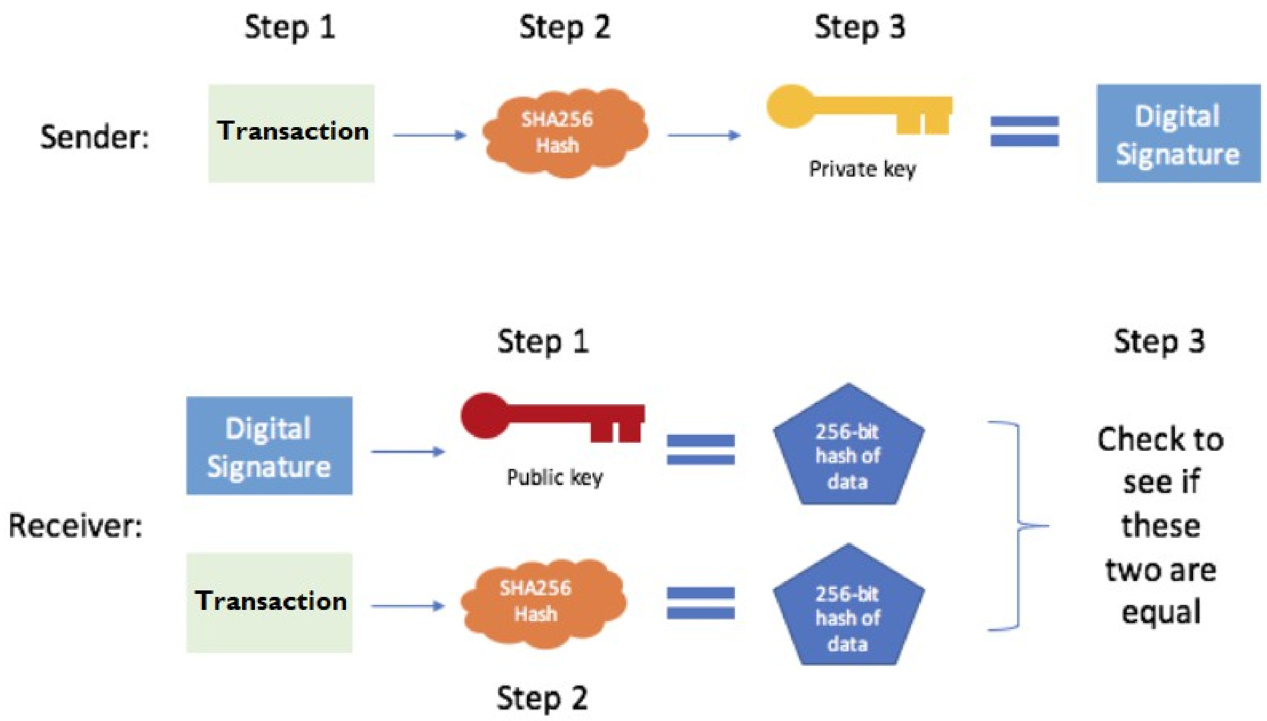
Alice opens a restaurant

- rental is high, venture capitalists are greedy
- Alice uses an `ICO` (`Initial Coin Offering`)
  - proposes a project that will be implemented on a blockchain
  - get funding from people proposing to participate to the project
  - create tokens to be given to the funders, as a compensation
    - discount meals for the restaurant
- how can Alice prove that tokens are really released by herself?

# PROOF OF OWNERSHIP



- use public key signatures
  - the second basic cryptographic tools for blockchains



# PROOF OF OWNERSHIP

- Alice generates a pair (public key, private key)
- private key gives ownership
  - possibility to sign the transfer operation
- public key gives the proof of ownership
  - prove that the emitter of the transfer is really the owner of the coupon
- when she releases tokens, she registers on the ledger a signed transactions
  - can be verified by the receiver

# PERMISSIONLESS BLOCKCHAIN

Alice's cryptotokens are permissionless

- anyone can participate
- anyone can be a miner
- no central authority
- based on reward
- may have some problems
  - blockchain forks
  - $$54M  DAO Ethereum Attack

# A PERMISSIONED BLOCKCHAIN

- Alice sells her restaurant and opens a frozen yogurt business
- but her business is in trouble
  - shipments arrive melted
  - where is the problem?

# THE FROZEN YOGURT SUPPLY CHAIN

Carol's factory

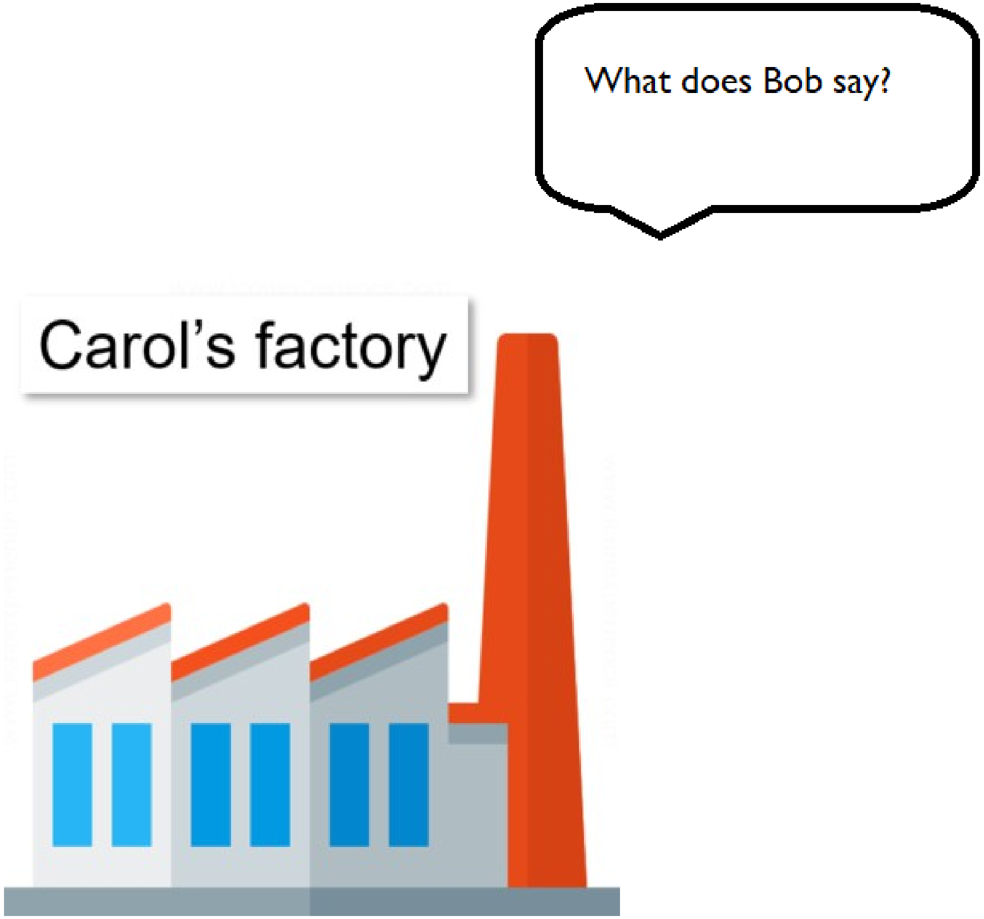
Bob's truck

# ALICE SUPPLY CHAIN

1. I never trasported that yougurt
2. It was meletd when I got it from Carol
3. It was OK when I delivered it to Alice

Bob's truck

# ALICE SUPPLY CHAIN



# USE A BLOCKCHAIN

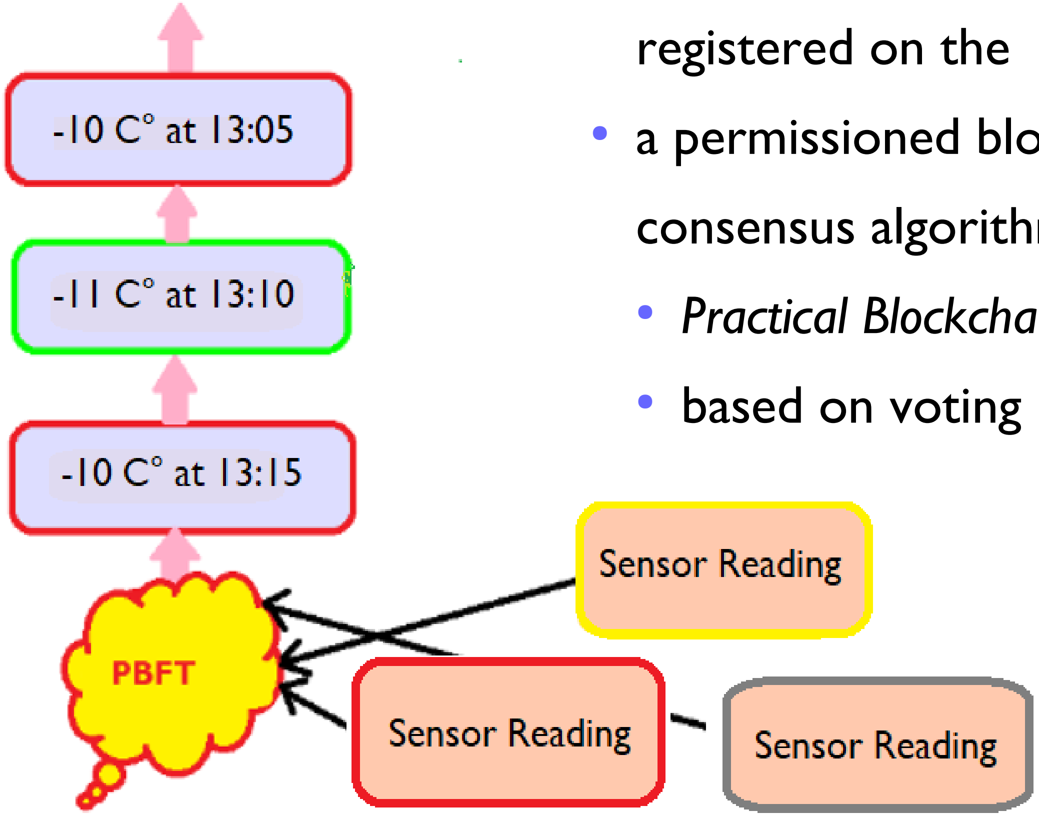
- record the events ina  blockchain
  - temperature, humidity
  - in the truck, in the factory
- put the ledger in the cloud
- events are registered: auditability

Bob and Carol

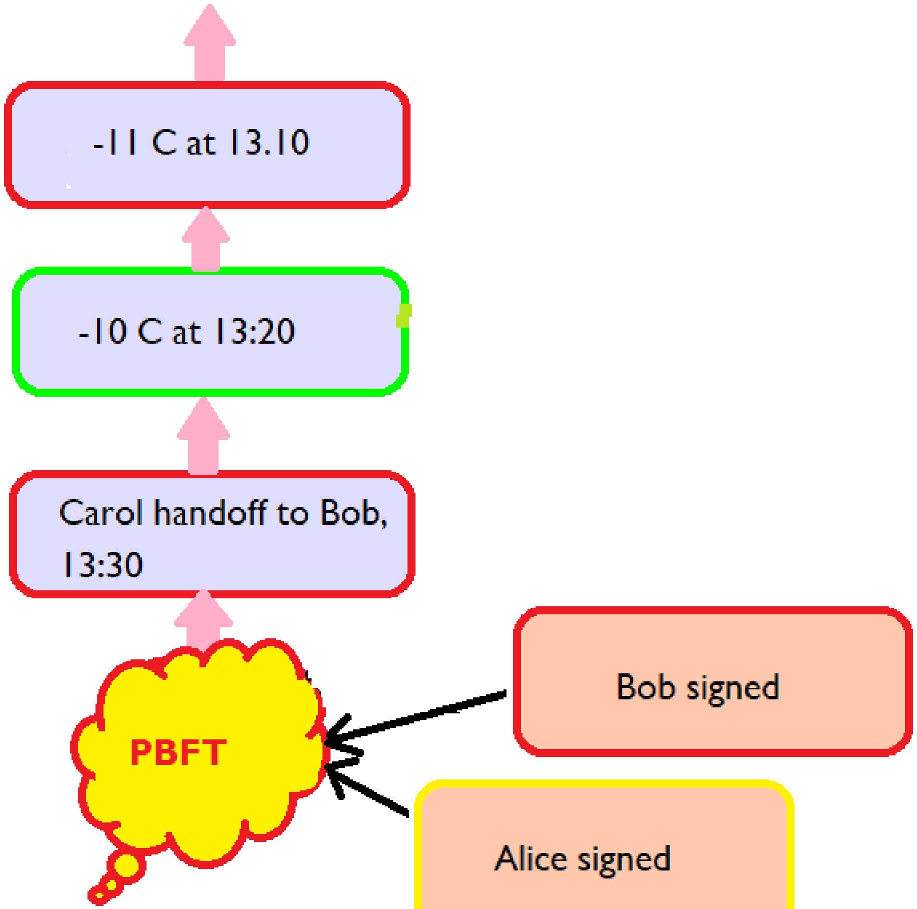
Sensors

# PERMISSIONED BLOCKCHAIN FOR SUPPLY CHAIN

- put sensors in the truck and in the factory
- not transactions, but sensor readings are registered on the  blockchain
- a permissioned blockchain with a new consensus algorithm
  - *Practical Blockchain Fault Tolerance (PBFT)*
  - based on voting



# PERMISSIONED BLOCKCHAIN FOR SUPPLY CHAIN



-11 C at 13.10

-10 C at 13:20

Carol handoff to Bob, 13:30

PBFT

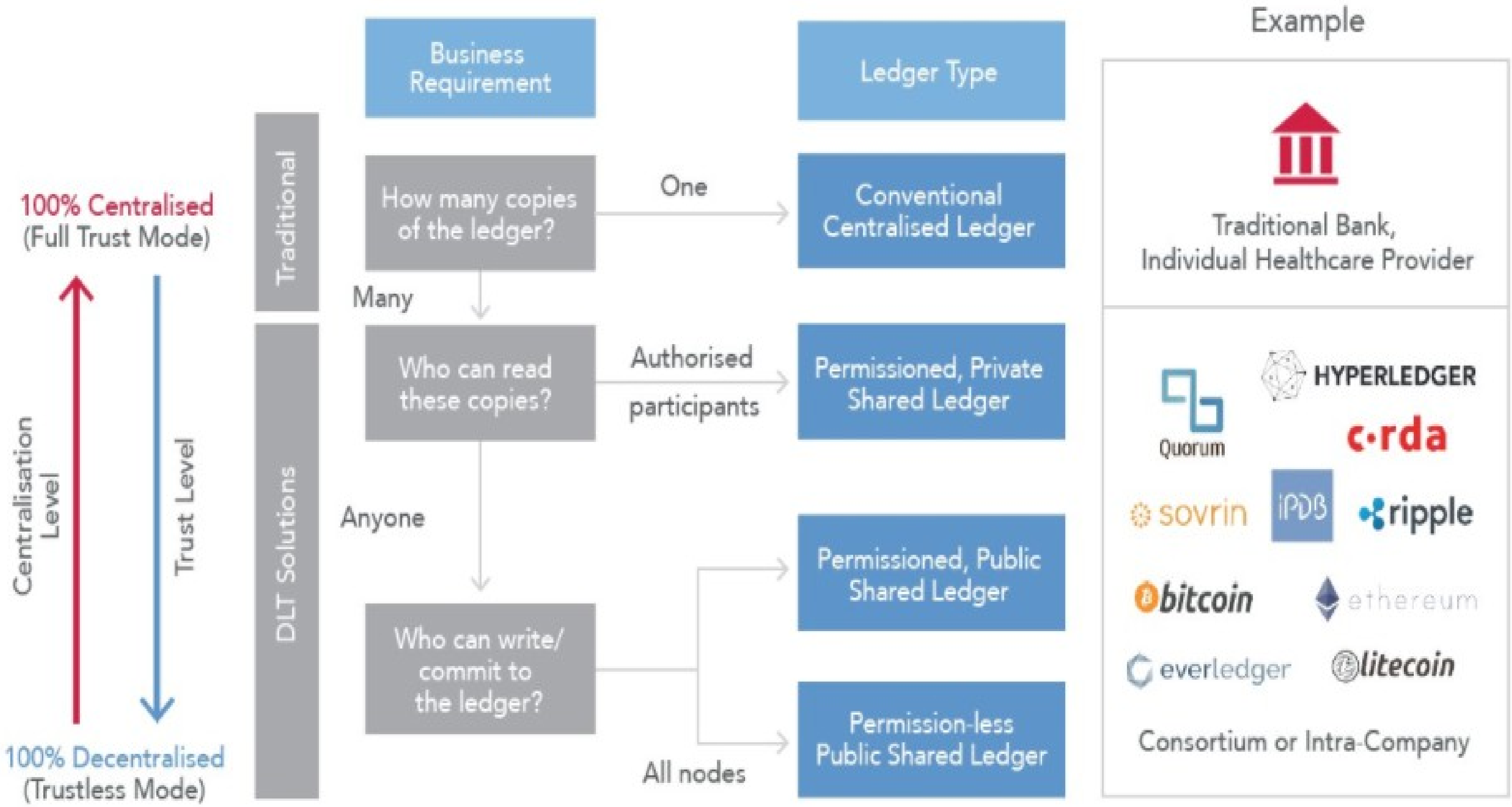
Bob signed

Alice signed

# PERMISSIONED BLOCKCHAIN: WHAT IS DIFFERENT?

- parties are identified
- humans have passwords, keys
- sensors have keys
  - both humans and sensors are authenticated
- different consensus mechanisms
- accountability if caught cheating

# FINDING THE WAY IN THE BLOCKCHAIN JUNGLE


Business Requirement
Ledger Type
Example
Traditional
How many copies of the ledger?
One
Conventional Centralised Ledger
Traditional Bank, Individual Healthcare Provider
100% Centralised
(Full Trust Mode)
Many
DLT Solutions
Who can read these copies?
Authorised participants
Permissioned, Private Shared Ledger
Quorum
HYPERLEDGER
c•rda
sovrin
IPDB
ripple
Centralisation Level
Trust Level
Anyone
Permissioned, Public Shared Ledger
bitcoin
ethereum
Who can write/ commit to the ledger?
everledger
litecoin
Permission-less Public Shared Ledger
All nodes
Consortium or Intra-Company
100% Decentralised
(Trustless Mode)

# SMART CONTRACTS FOR DUMMIES

- what is a smart contract?
- use cases

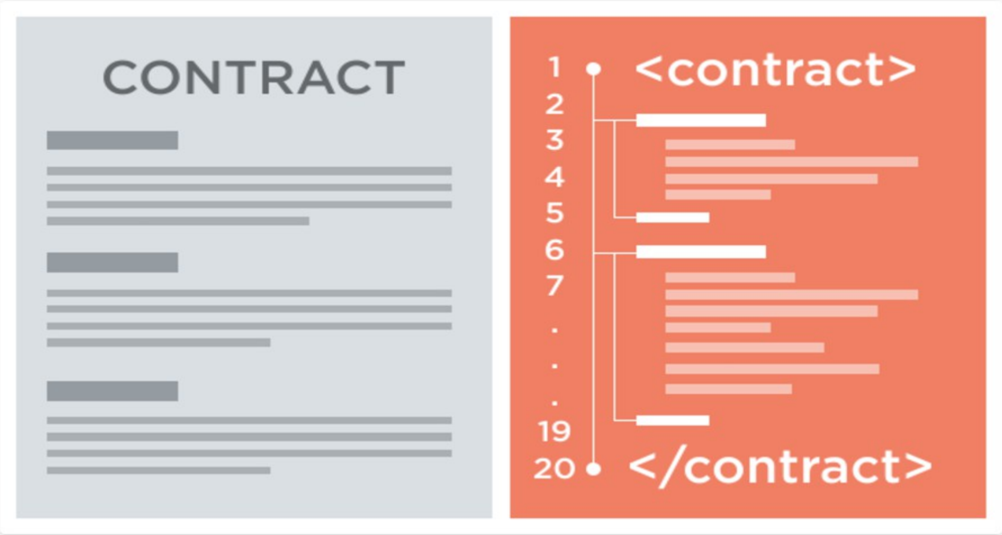
# THE FIRST IDEA OF SMART CONTRACTS

- the term "smart contract" was first introduced by Nick Szabo, computer scientist, law scholar, and cryptographer, in the nineties, long before Bitcoin

- *"a smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitrations and enforcement costs, and other transaction costs"*

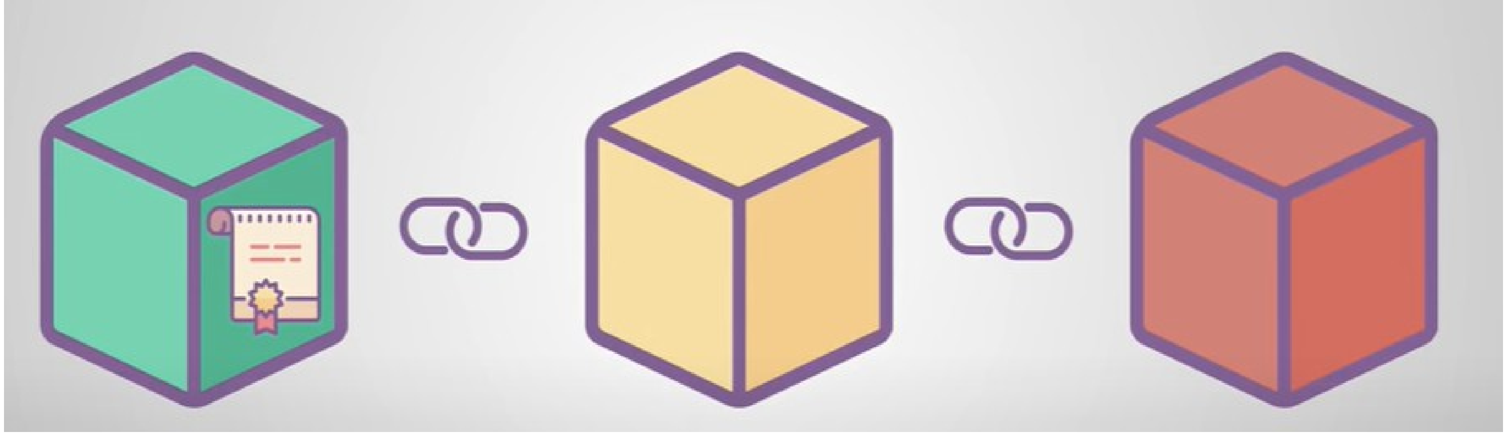
  *[Nick Szabo "The Idea of Smart Contracts" ]*

# SMART CONTRACTS AND BLOCKCHAIN



- just like contracts in the real world
- but they are completely digital
  - a tiny computer program stored inside the blockchain
  - code automates the "*if this happens then do that*" part of traditional contracts
- better with respect to normal contract: computer code behaves in expected ways and doesn't have the linguistic nuances of human languages.

# SMART CONTRACTS: RECAP



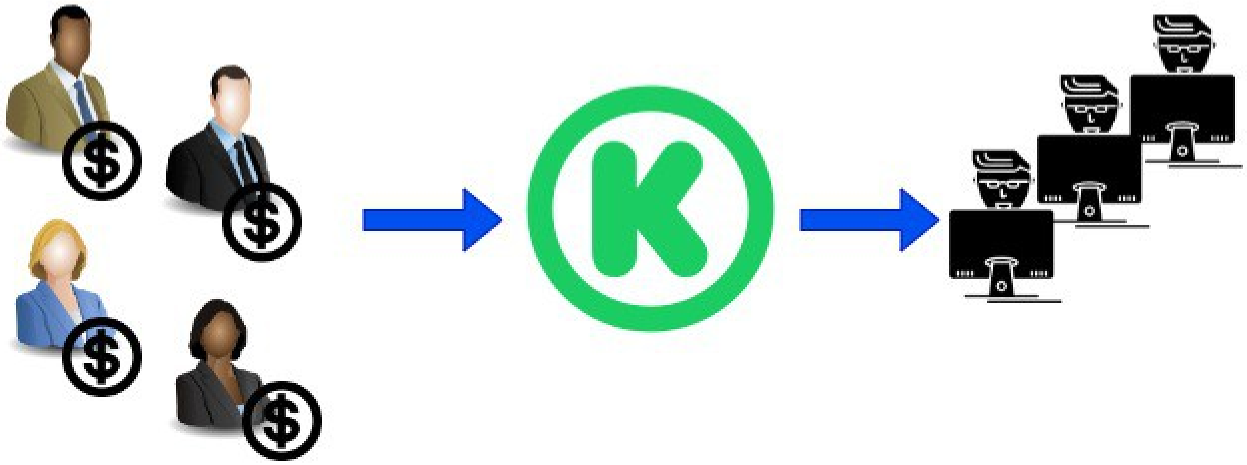
- a piece of software (program) written on a blockchain.
- all parties can view the contract, but it is not possible to change the contract (the code) after it has been deployed on the blockchain.
- as a result, the parties do not necessarily have to trust each other
  - they can rely on the contract and trust underlying blockchain technology
  - disintermediation: smart contracts ensure that an intermediary (Airbnb, broker, notary...) is not needed

# SMART CONTRACTS AND DISTRIBUTED LEDGERS

- contract
  - formalizes a relationship and contains promises made between principals.
- smart contract
  - based on the translation of contractual clauses into code
    - a digital agreement: two or more parties specify agreements with conditions.
  - more functional compared to paper-based: can reduce costs
  - aim to remove the need for trusted intermediaries
    - make it more difficult for malicious parties to undermine compliance with the contract terms
  - uses cryptography and other security mechanisms
    - secure algorithmically specifiable relationships from being breached and ensure the agreed upon terms are satisfied.

# CENTRALIZED CRODWFUNDING

- product teams can go to *Kickstarter*
  - essentially a third party that sits between start-up and supporters
  - operating on the web
- create a project and start collecting funds from other supporters who do believe in their idea

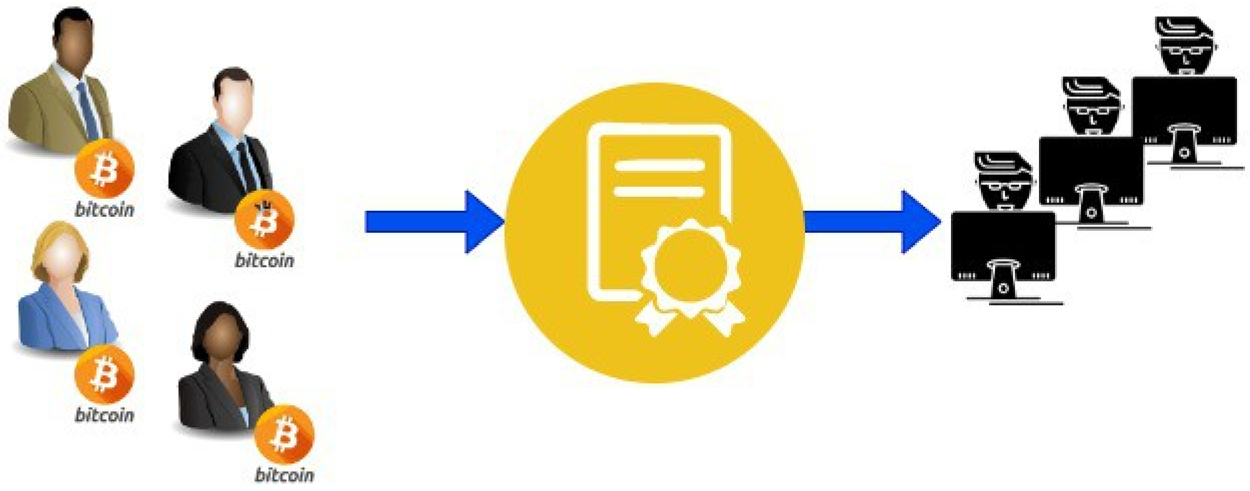


# CENTRALIZED CROWDFUNDING: TRUSTING A THIRD PARTY

- both supporters and producers need to trust *Kickstarter* to handle their money correctly
- if the project gets successfully funded
  - the team
    - expects *Kickstarter* to give them the collected money
  - supporters
    - want some rewards if the project is successfull
- if the project has not been funded supporters gets refunded

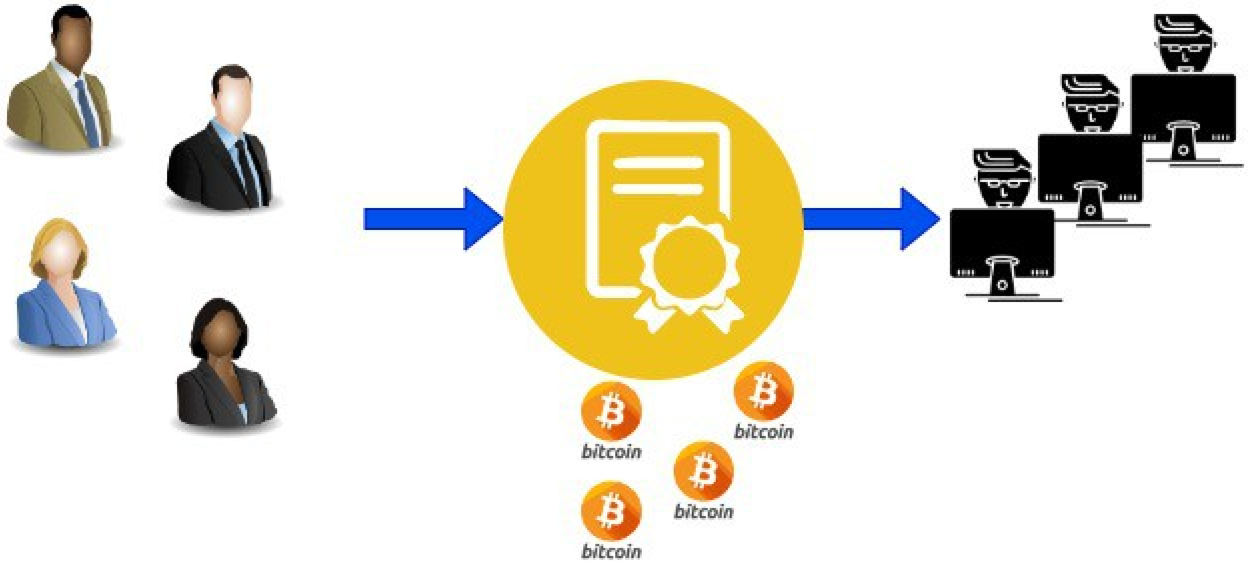
# DECENTRALIZED FINANCE (DEFI)

- no third party: instead program a smart contract between supporters and investors



# DECENTRALIZED FINANCE (DEFI)

- the supporters can transfer their money to the smart contract
  - it holds all the received funds until a certain goal is reached
- computation and money transfer inside the smart contract



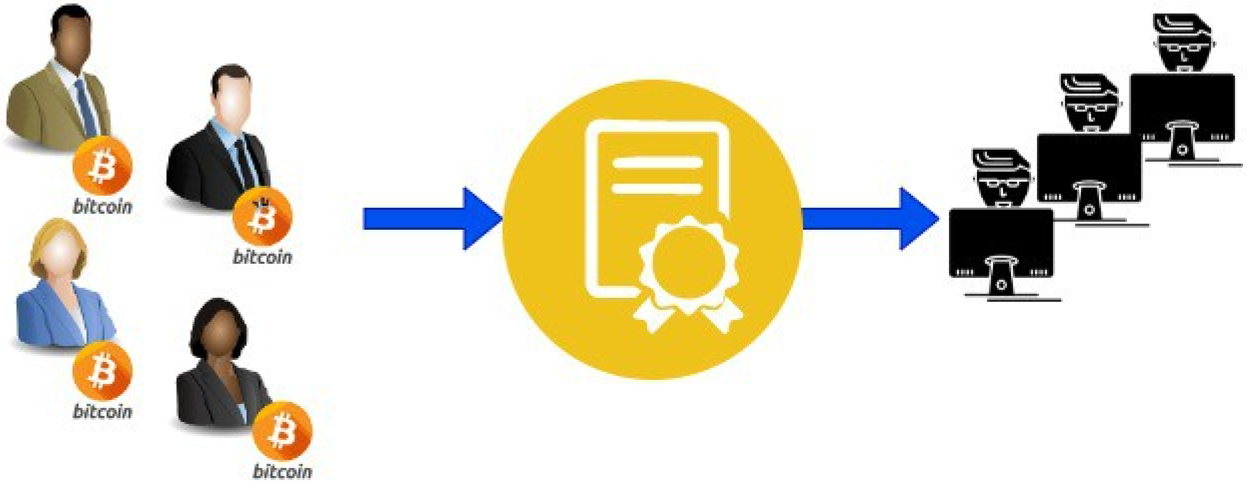
# DECENTRALIZED FINANCE (DEFI)

- no third party: instead program a smart contract
- if the project gets fully funded
  - the contract automatically passes the money to the creators

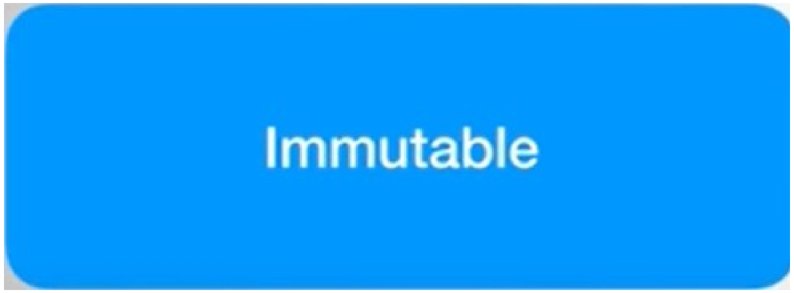
bitcoin
bitcoin
bitcoin
bitcoin

# DECENTRALIZED FINANCE (DEFI)

- if the project fails to meet its goal
  - the money automatically goes back to the supporters



# DEFI: A SMART CONTRACT BASED SOLUTION



Immutable

Distributed

- smart contract are stored on the blockchain, everything completely distributed
- but why do we trust a smart contract? Because they inherit some properties of the blockchain
  - *immutable*: once a smart contract is created, it can never be changed. No one can tamper with the code of the smart contract
  - *distributed*: the program is executed by all the nodes of the blockchain and the output of the contract is validated by everyone in the network
    - no one can control the money

# DECENTRALIZED FINANCE (DEFI)

Releasebthe funds
even if the goal
is not reached
No, the goal is not
reached!
Do not release
the funds!

- a single person (or adversary) cannot force the contract to release funds
- because other nodes on the network will spot this attempt and consider it invalid
  - the smart contract is executed by all the nodes and a consensus on its results is reached
  - this hold if and only if the 51% of the nodes are honest!

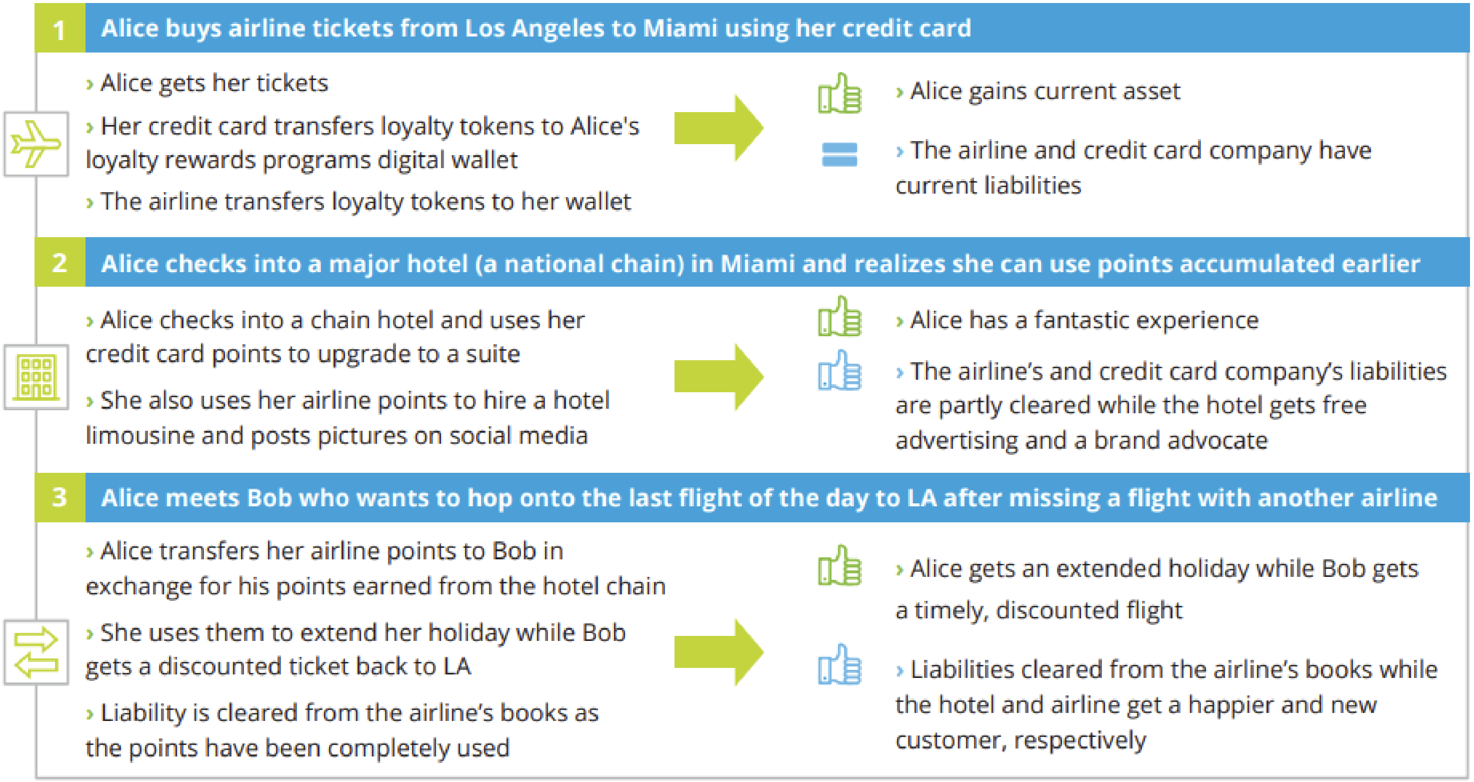
# ETHEREUM VERSUS BITCOIN

Bitcoin

Ethereum

# FUNGIBLE TOKENS AND NFT

- a “killer application” for blockchain
- an example: customer loyalty rewards tokens
  - several flight companies deliver coupons connected to rental cars, airport parking, hotels, and massage services



1 Alice buys airline tickets from Los Angeles to Miami using her credit card

- › Alice gets her tickets
- › Her credit card transfers loyalty tokens to Alice's loyalty rewards programs digital wallet
- › The airline transfers loyalty tokens to her wallet

- › Alice gains current asset
- › The airline and credit card company have current liabilities

2 Alice checks into a major hotel (a national chain) in Miami and realizes she can use points accumulated earlier

- › Alice checks into a chain hotel and uses her credit card points to upgrade to a suite
- › She also uses her airline points to hire a hotel limousine and posts pictures on social media

- › Alice has a fantastic experience
- › The airline’s and credit card company’s liabilities are partly cleared while the hotel gets free advertising and a brand advocate

3 Alice meets Bob who wants to hop onto the last flight of the day to LA after missing a flight with another airline

- › Alice transfers her airline points to Bob in exchange for his points earned from the hotel chain
- › She uses them to extend her holiday while Bob gets a discounted ticket back to LA
- › Liability is cleared from the airline’s books as the points have been completely used

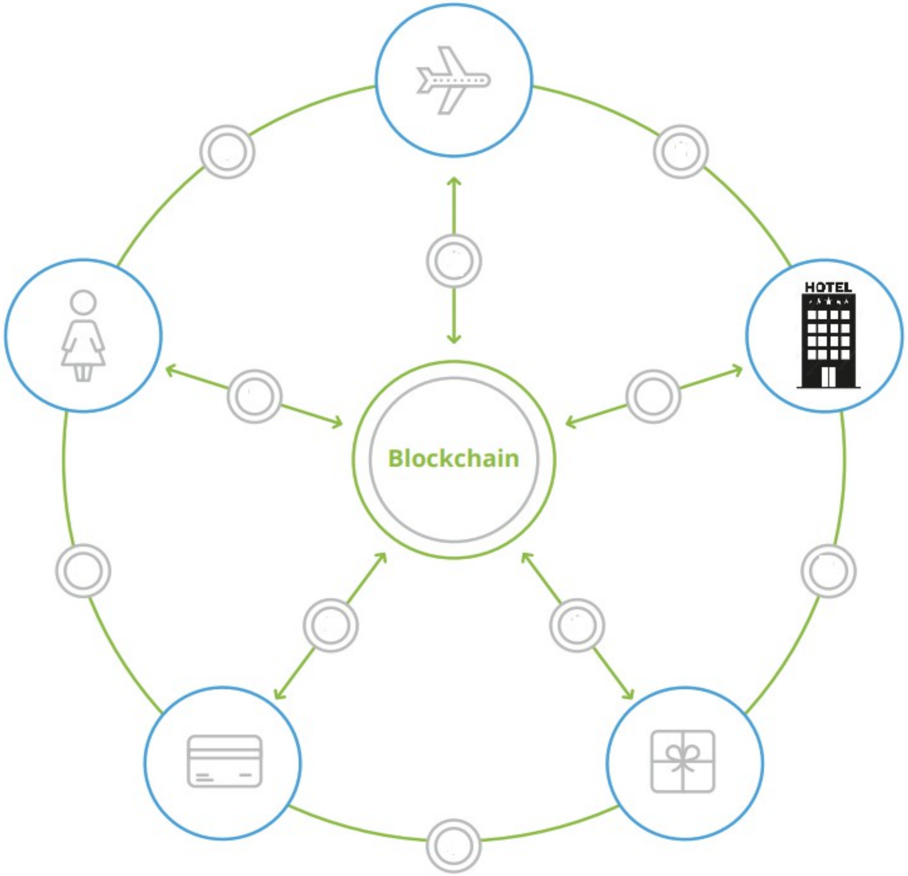
- › Alice gets an extended holiday while Bob gets a timely, discounted flight
- › Liabilities cleared from the airline’s books while the hotel and airline get a happier and new customer, respectively

Deloitte report “Making blockchain Real for customer Loyalty rewards programs”

# LOYALTY PROGRAMS AND BLOCKCHAINS

blockchain allows instantaneous and secure creation, redemption, exchange of loyalty reward points

- across programs, vendors, and industries
  - using different software systems
- in a trustless environment
  - no trusted third parties and administrators
  - trust given by the platform


HOTEL
Blockchain