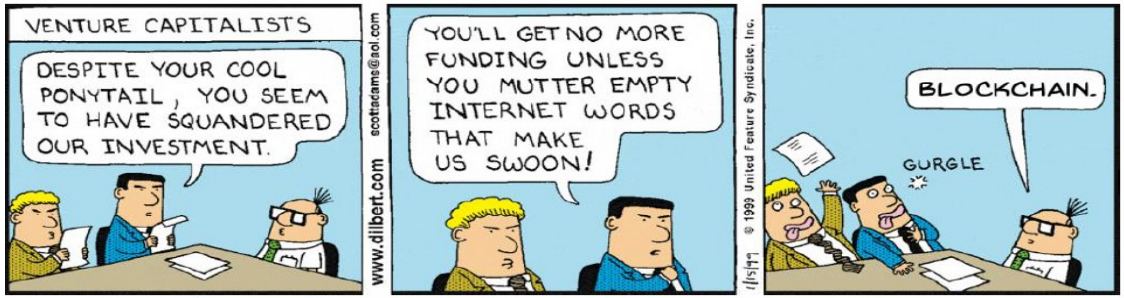
Blockchain: what it is and why it matters

Laura Ricci,
Damiano Di Francesco Maesa

Dipartimento di Informatica
Università degli Studi di Pisa

6 Aprile 2022

BLOCKCHAIN : HYPE OR REALITY?

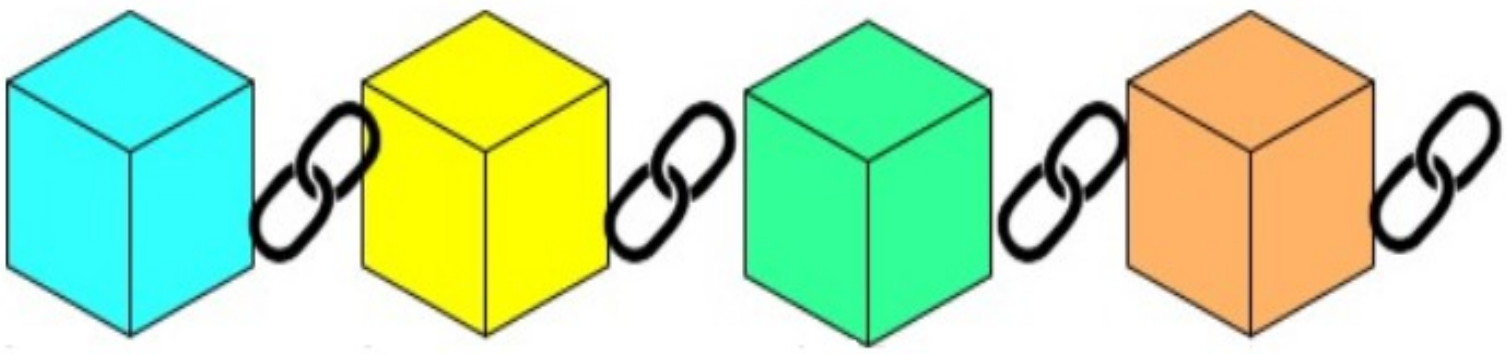


MENU CNBC MARKETS BUSINESS NEWS INVESTING TECH

Salaries for blockchain engineers are skyrocketing, now on par with AI experts

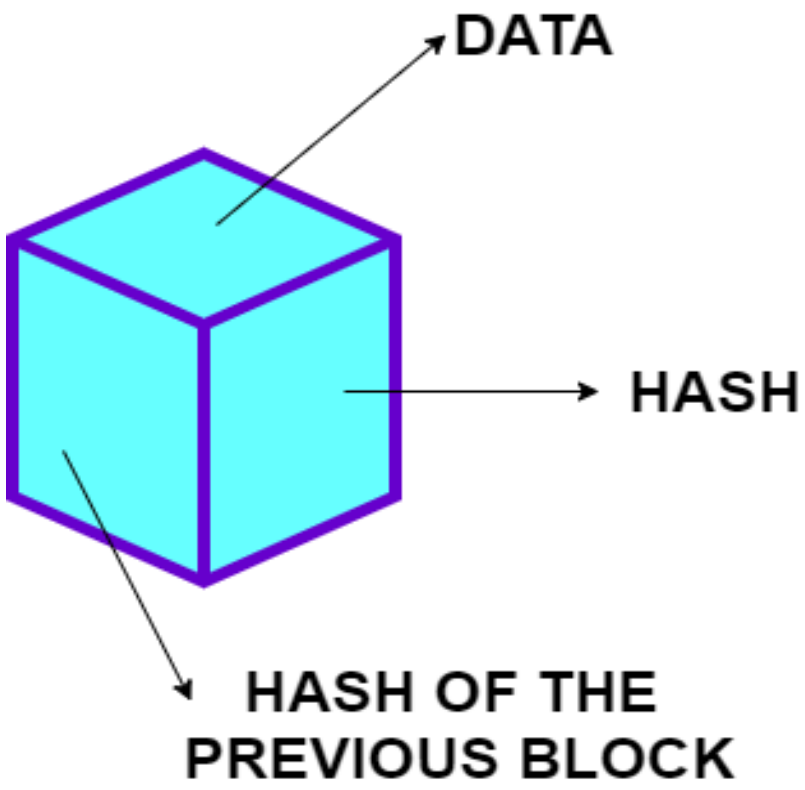
- Blockchain engineers are making between $150,000 and $175,000 in annual salaries on average.
- Blockchain engineers are the top paid roles in software development, on par with specialists focused on artificial intelligence.
- Demand for blockchain engineers has increased by 400 percent since late 2017 on Hired, a firm that helps clients recruit tech candidates.

BLOCKCHAIN "AT A GLANCE"

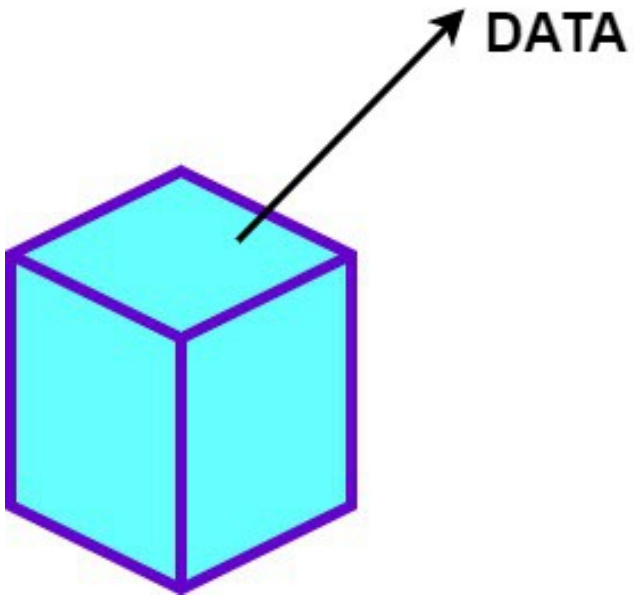


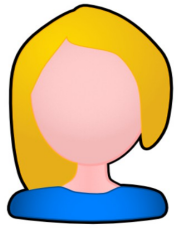
- a ledger which is replicated among the nodes of a peer-to-peer network
- all the nodes have the same replica
- benefits of the tamper freeness property
- may act like a notary

LOOKING INSIDE A BLOCK

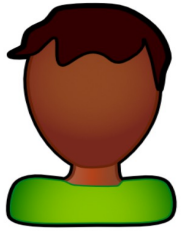


LOOKING INSIDE A BLOCK: DATA





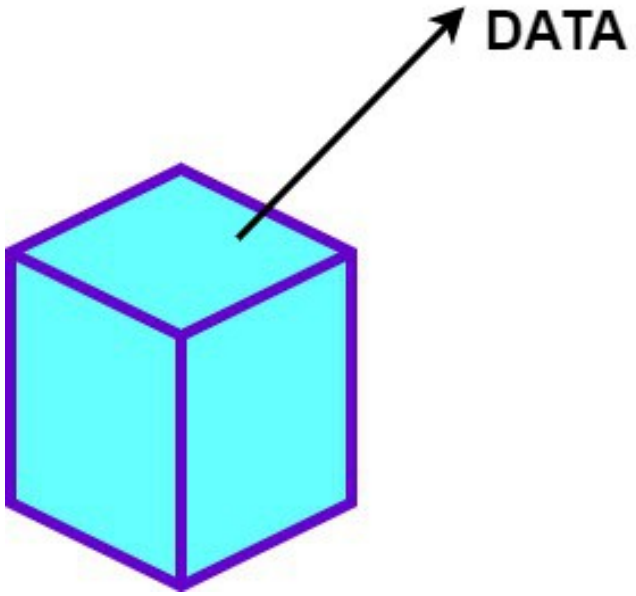
FROM

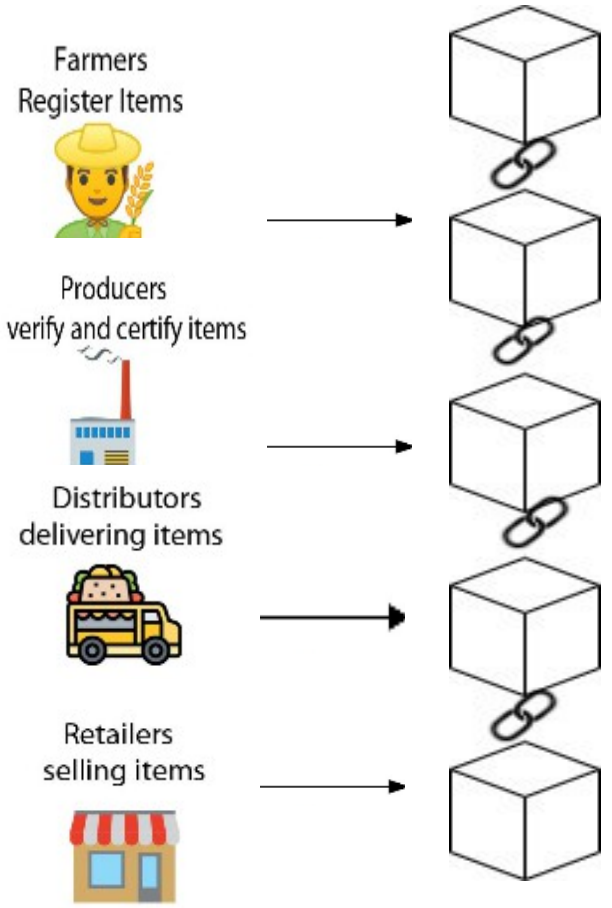


TO

AMOUNT

LOOKING INSIDE A BLOCK: DATA

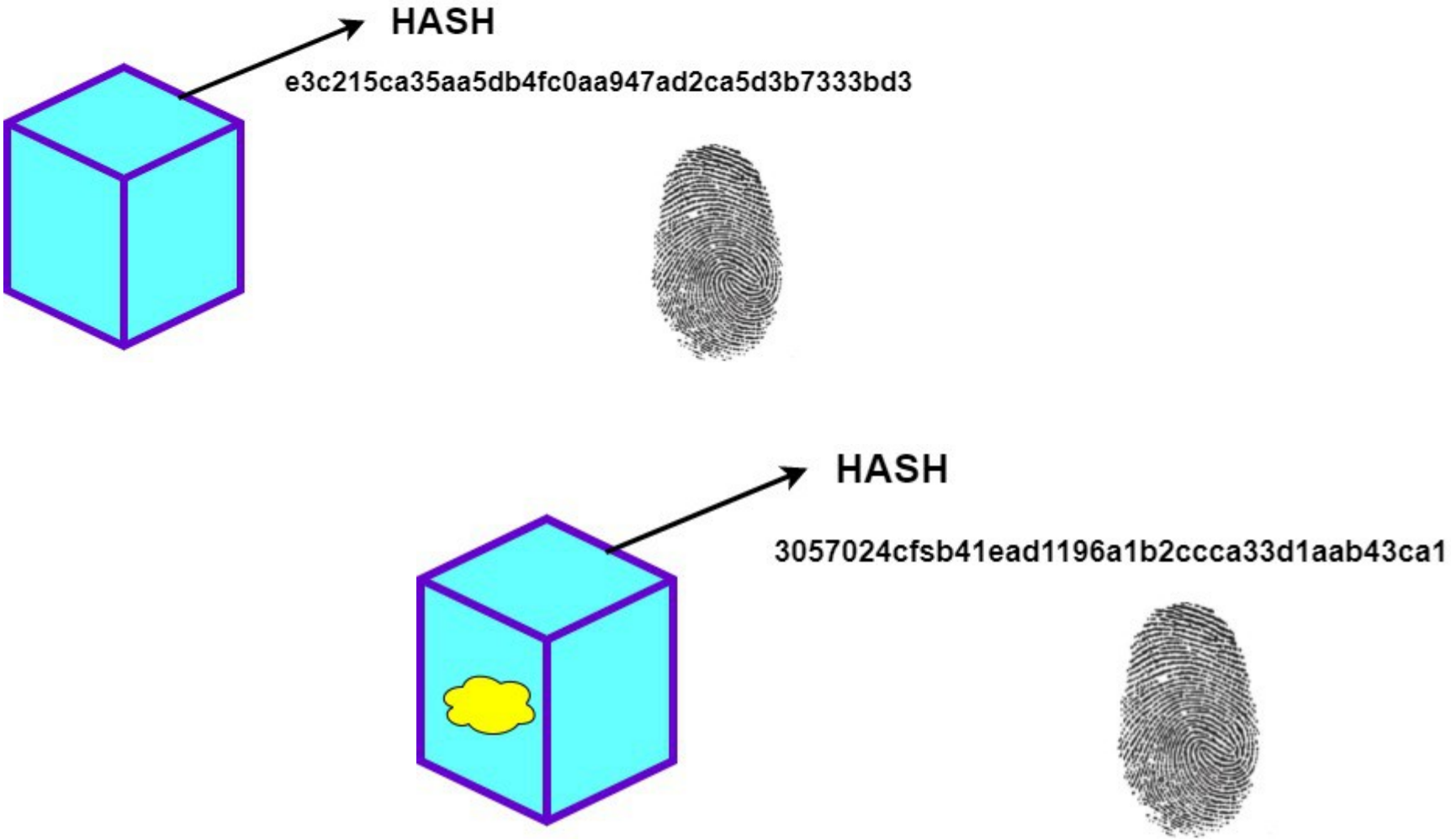




BLOCKCHAIN TRANSACTIONS

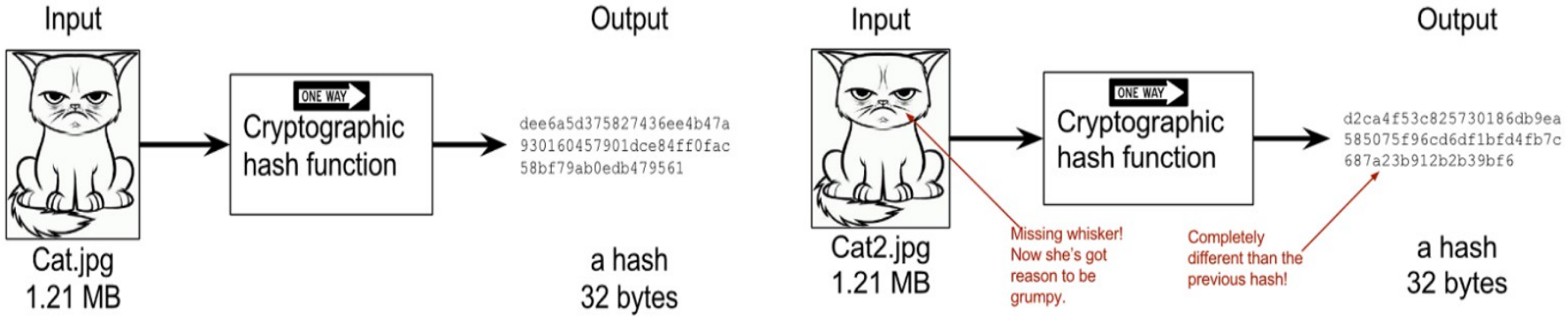
- what do we mean, when we speak of transactions?
  - what is registered in the replicated ledger?
- the simplest scenario: Bitcoin, in general cryptocurrencies
  - transfer of an amount of money between two entities
- but we may imagine many further scenarios....
  - a contract
  - an intellectual property licence
  - the temperature detected by a sensor inside a truck carrying drugs
  - the registration of fever detection to access the Department, in COVID time,...

LOOKING INSIDE A BLOCK: HASH



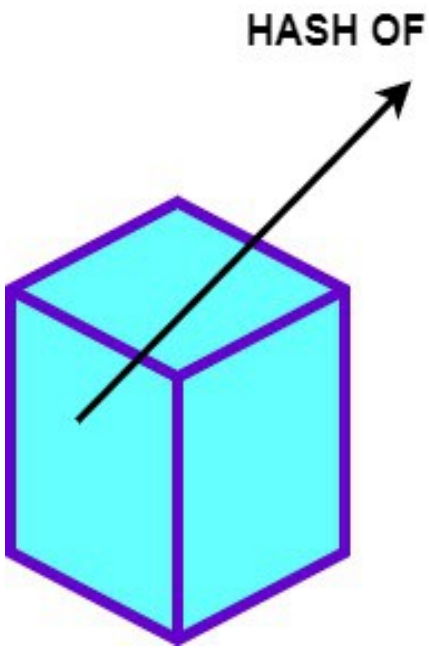
CRYPTOGRAPHIC HASH FUNCTIONS

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



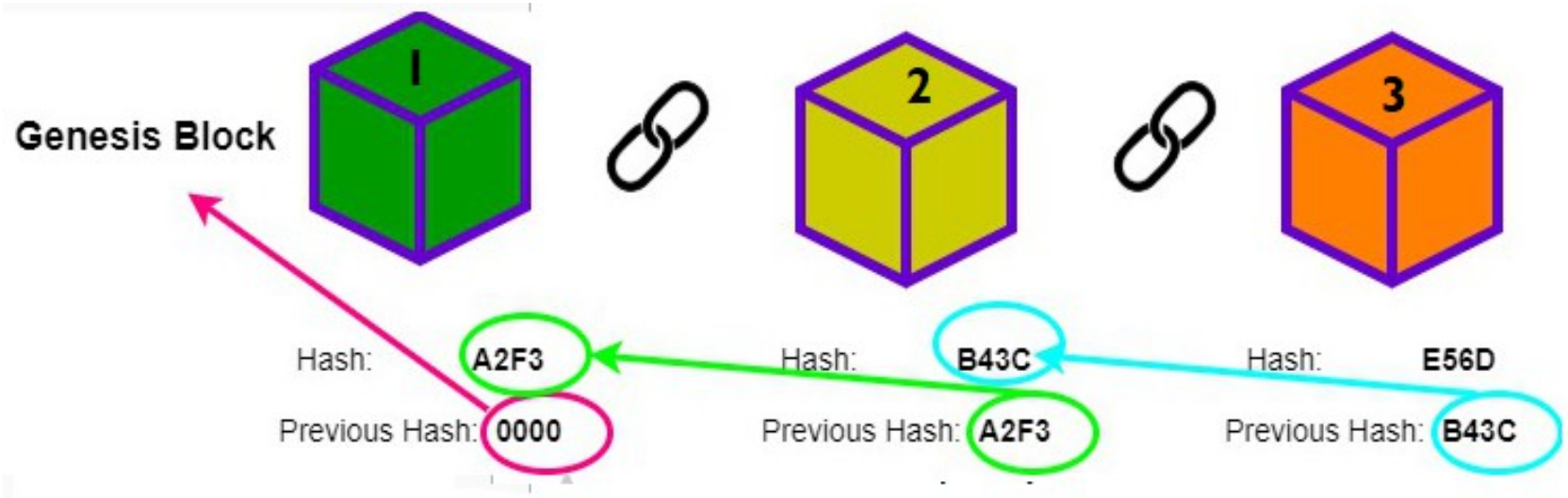
- input data : any length, any type
- output data:
  - fixed-length sequence of characters
  - if input is slightly changed, output is completely changed
- one-way: it is computationally hard to go from the hash to the input
- other nice properties, collision freeness, an dothers

LOOKING INSIDE A BLOCK: HASH POINTES



Hash pointers create the chain

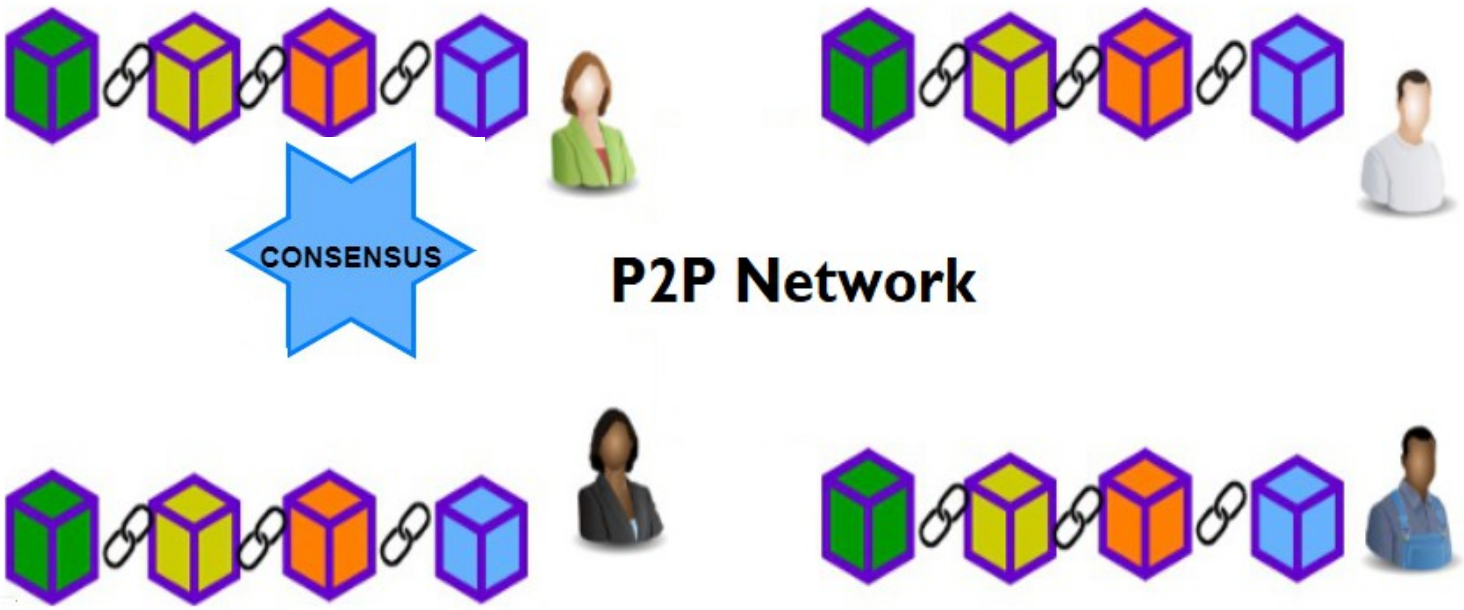
THE BLOCKCHAIN



THE BLOCKCHAIN: TAMPER FREENESS

- changing one hash caused changing the hash of the following blocks
- it must be very difficult to recompute the hash of the blocks
  - this is guaranteed in many blockchains by the Proof of Work
  - other blockchains may use different mechanisms

THE BLOCKCHAIN: CONSENSUS



BLOCKCHAIN: POSSIBLE DEFINITIONS

- Definition #1
  - a shared database stored in multiple copies on computers throughout the world
  - maintained without the need for a central authority (e.g. a bank, a government, Google, etc.)
- Definition #2
  - replicated and consistent, immutable, append-only data storage system resistant to tampering
- Definition #3
  - 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

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
- more complex cryptographic tools are under study but are not strictly required by basic blockchain
  - zero-knowledge, multi party, verifiable random functions, authenticated data structures

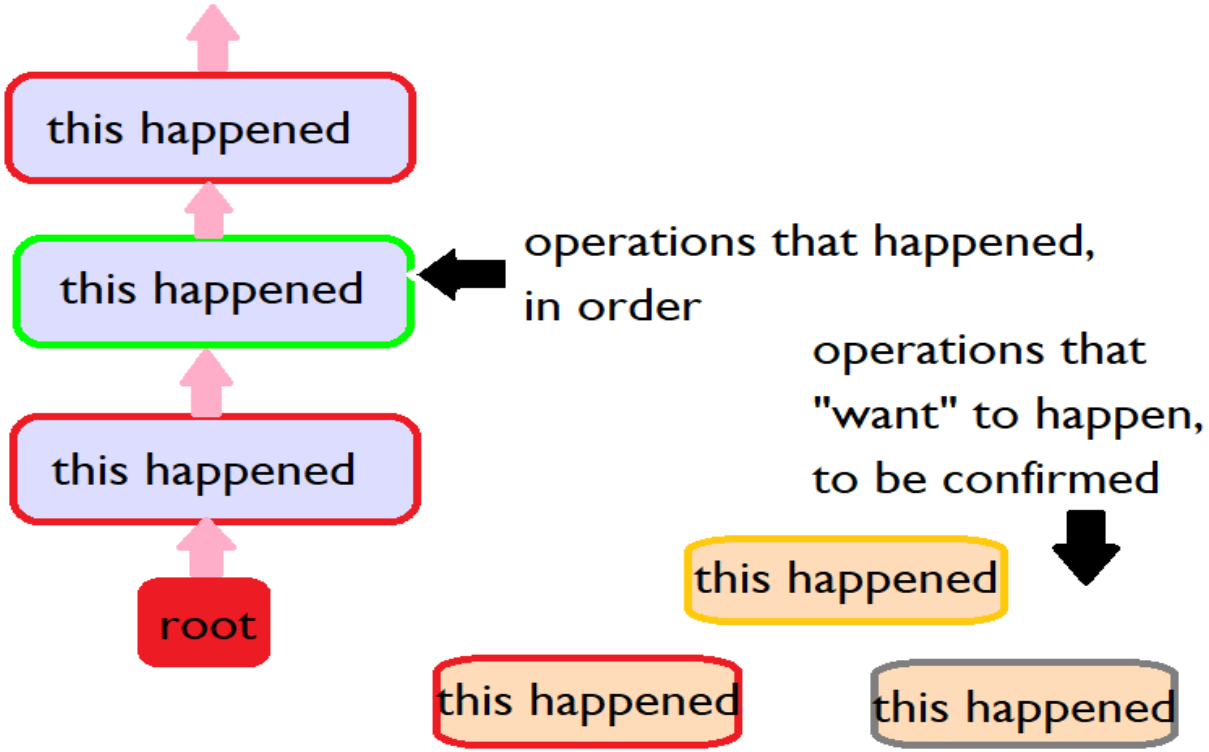
ABSTRACTING THE BLOCKCHAIN: THE LEDGER

- a ledger
  - like a bulletin storing operations
  - maintains the order of operations
- which properties needed for a ledger?
  - *append-only* list of events
  - *tamper-proof*
    - auditability
  - *everyone agrees on content*
    - consensus
- not just financial!
  - any application which needs a log of events

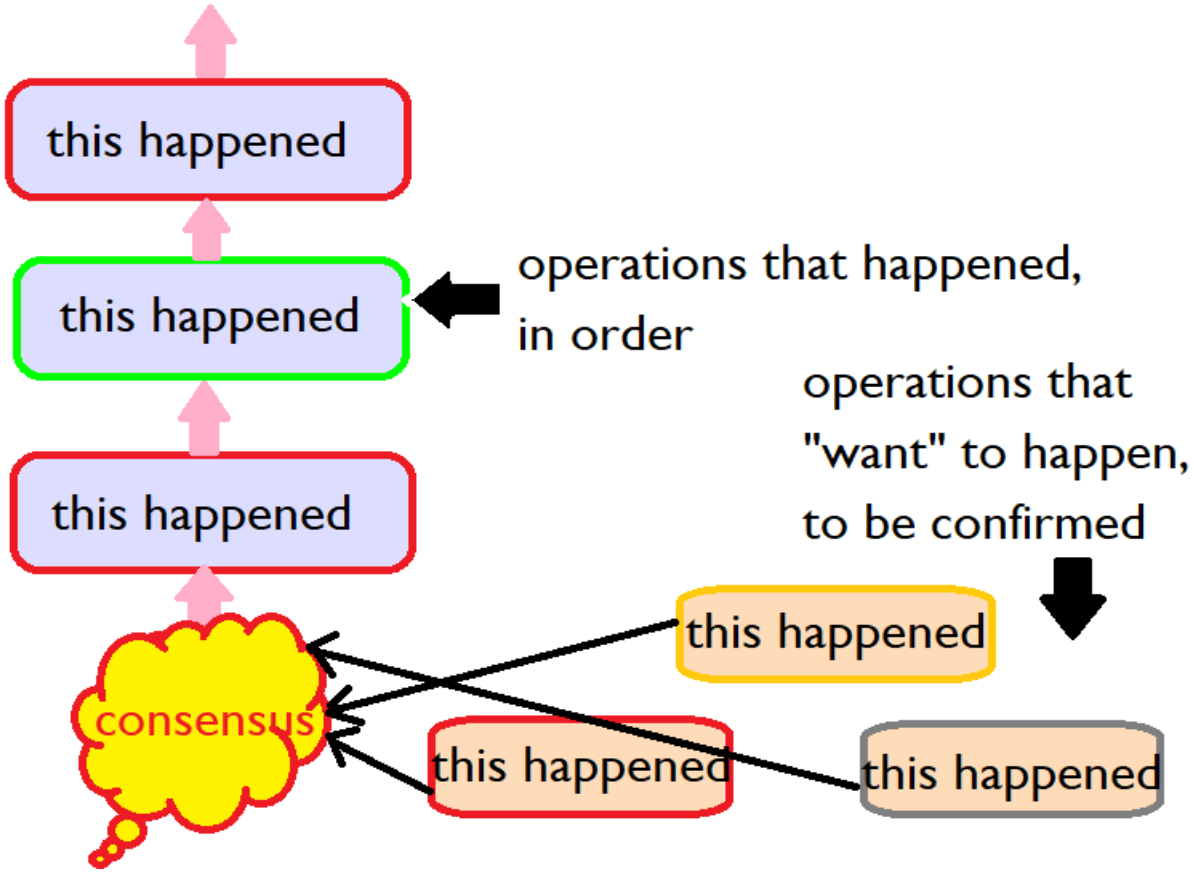
Cash				
Date	Description	Increase	Decrease	Balance
Jan. 1, 20X3	Balance forward			$ 50,000
Jan. 2, 20X3	Collected receivable	$ 10,000		60,000
Jan. 3, 20X3	Cash sale	5,000		65,000
Jan. 5, 20X3	Paid rent		$ 7,000	58,000
Jan. 7, 20X3	Paid salary		3,000	55,000
Jan. 8, 20X3	Cash sale	4,000		59,000
Jan. 8,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

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)



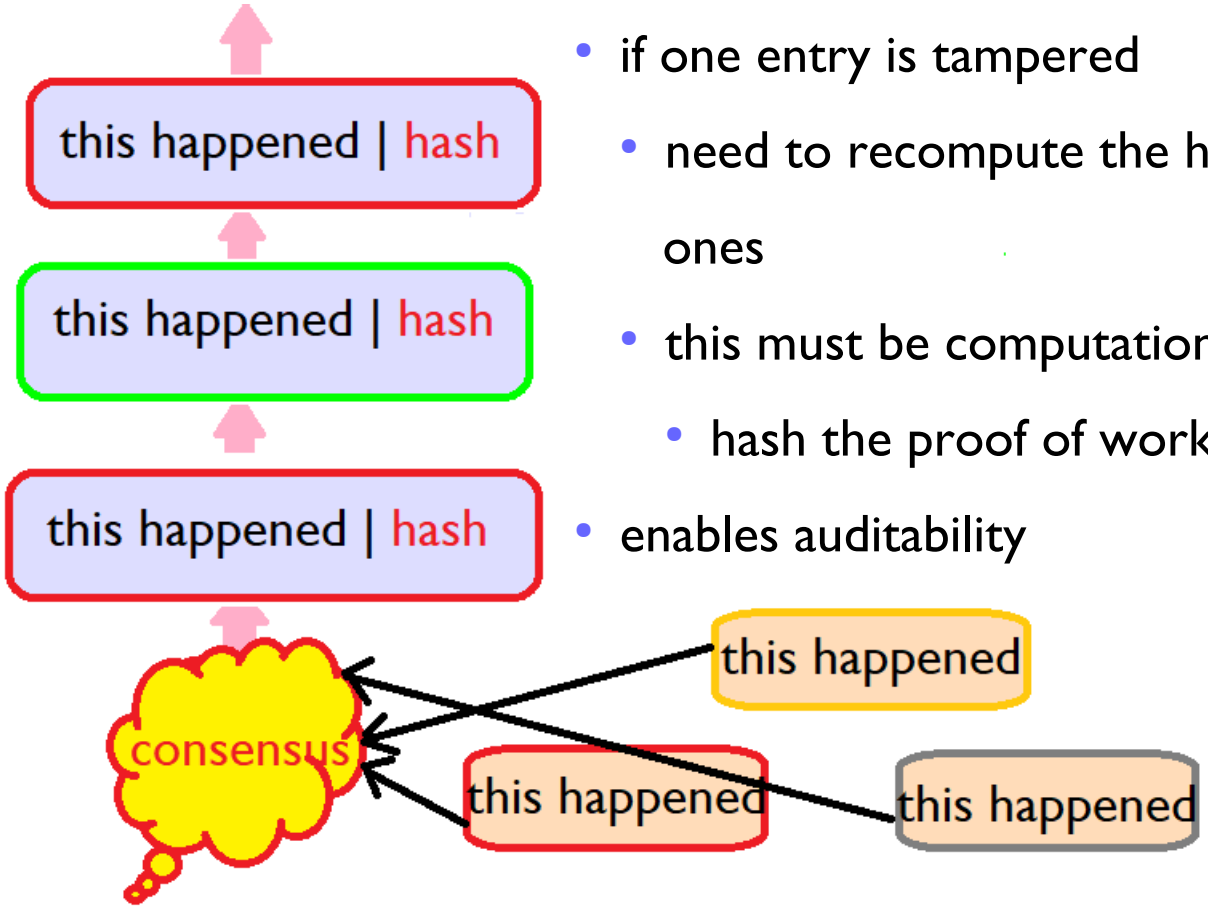
ADDING ENTRIES TO THE LEDGER



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

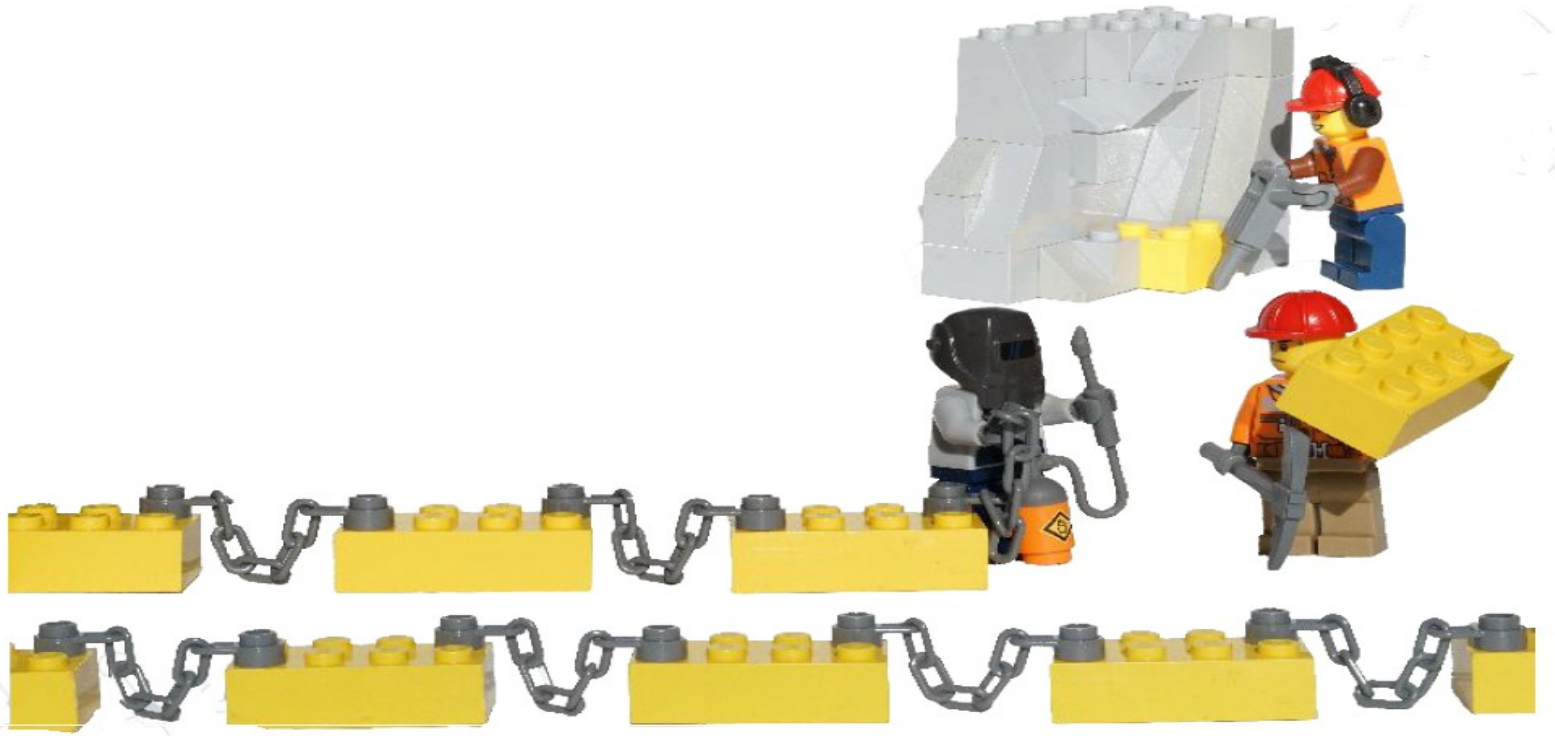
TAMPER FREENESS THROUGH HASH



- 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

- first implementation of consensus realized in Bitcoin:
  - a lottery
  - only who wins (ad to win is complex), can append the next block to the blockchain
  - 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

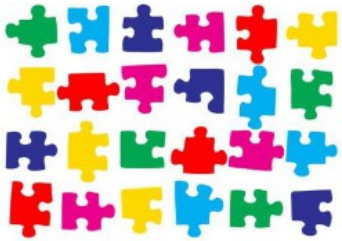
- how to select a random node at each round?
- the key idea: the probability to select a node must be proportional to the amount of resource has, a resource which is hard to monopolize
- in Bitcoin the probability to be selected is proportional to the computational power and selection is done on the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

Proof of work

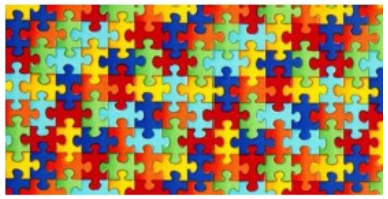
- nodes which try to solve the proof of work are called miners and the whole validation process is called mining

PROOF OF WORK

- based on cryptographic puzzles that
  - can be solved
  - require a considerable effort which cannot be short-circuited



Hard to find solution

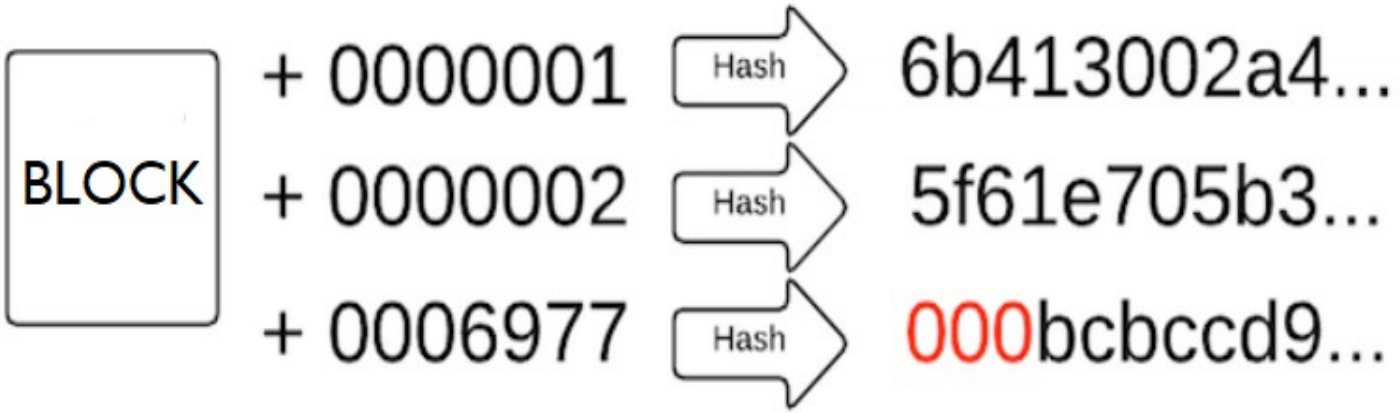


Easy to verify

  - 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
- 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: WHAT IS THE PROBLEM?

- 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

THE COST OF CONSENSUS

- the cost of Bitcoin mining is too high
  - energy waste
  - mining pools
- other solutions
  - employ energy than cannot be
  - alternative consensus algorithm
    - *proof of stake* (Algorand, Ethereum,...)
    - *delegated Proof of Stake* (Steemit, EOS,...)
    - *byzantine consensus* (Hypeledger,..)

IS IT ALL? NO PROOF OF OWNERSHIP §IS ALSO NEEDED

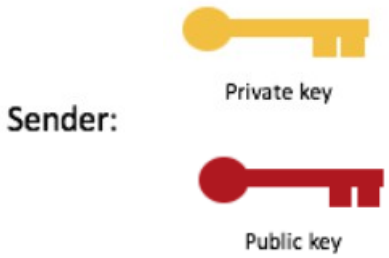
Alice opens a restaurant

- 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
  - crypto-coupon: discount meals when the restaurant opens
- Alice has to prove she is the real owner of the bitcoin.

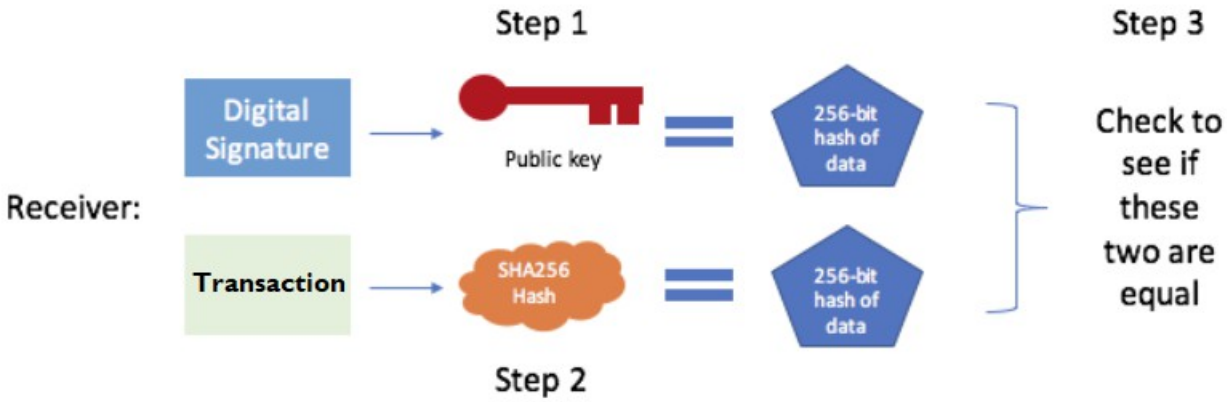
PROOF OF OWNERSHIP

- Alice generates a pair (public key, private key)
- anyone who knows the private key matching the public key owns the cryptocoupons associated to the private key. May be:
  - Alice themself
  - everyone Alice gives the private key to
  - everyone who steal Alice's 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
- register on the ledger the signed transactions
  - can be verified by the receiver

PROOF OF OWNERSHIP



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



PERMISSIONLESS BLOCKCHAIN

Alice's cryptocoupons are permissionless

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

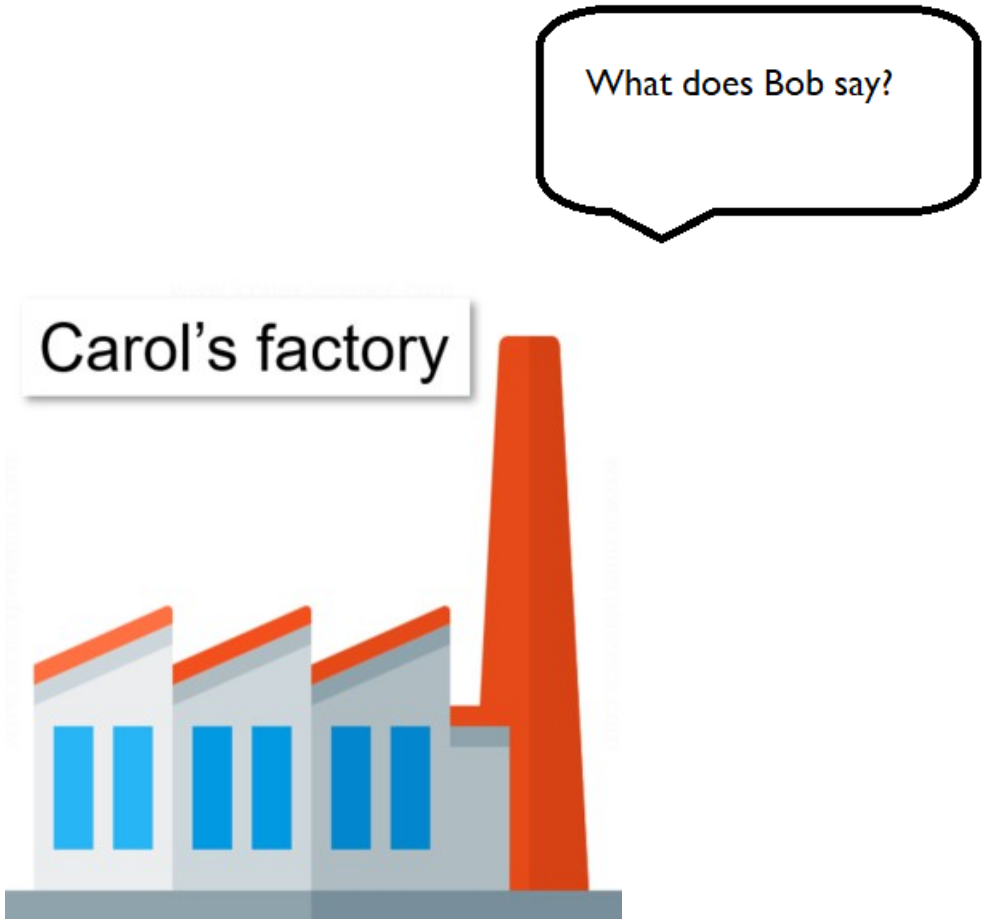
THE FROZEN YOGURT SUPPLY CHAIN

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



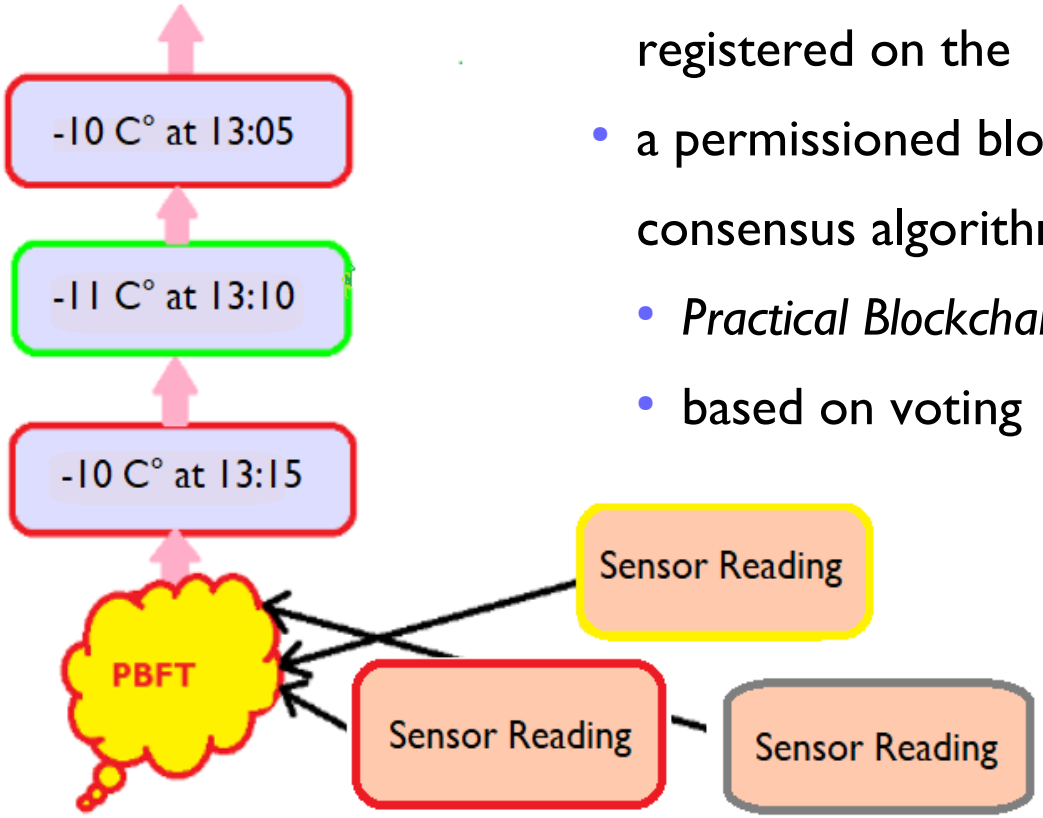
USE A BLOCKCHAIN

Bob and Carol

Sensors

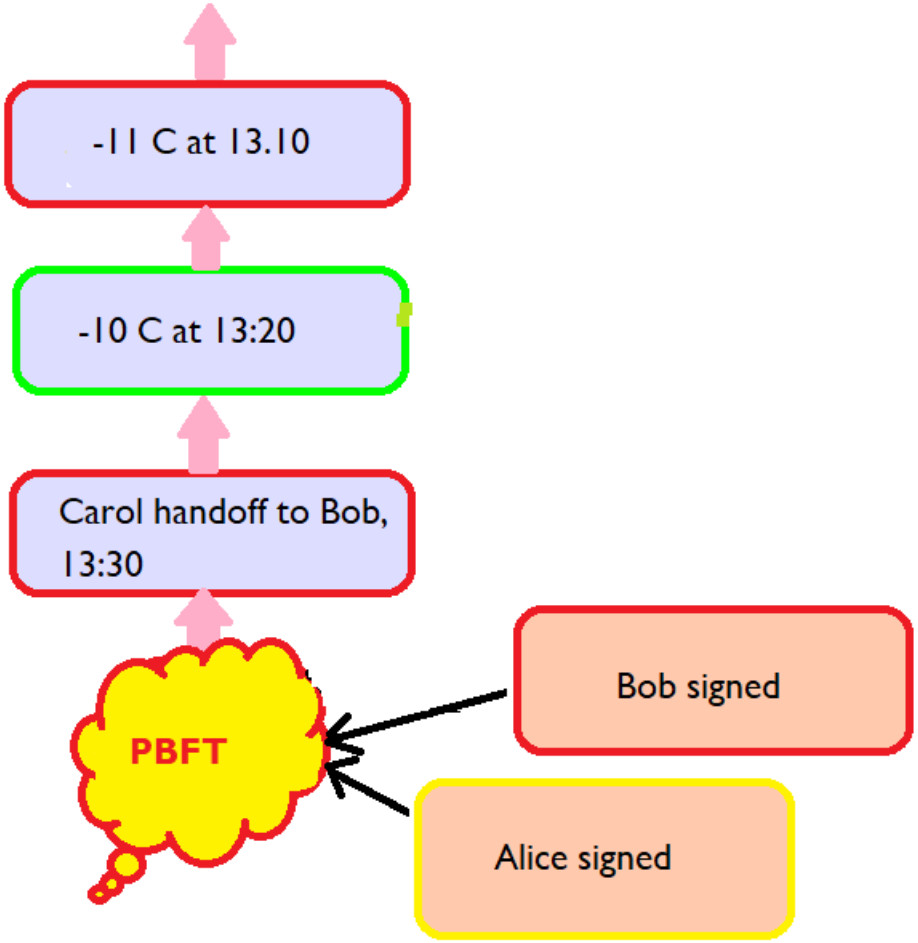
- record the events ina blockchain
  - temperature, humidity
  - in the truck, in the factory
- put the ledger in the cloud
- events are registered: auditability

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

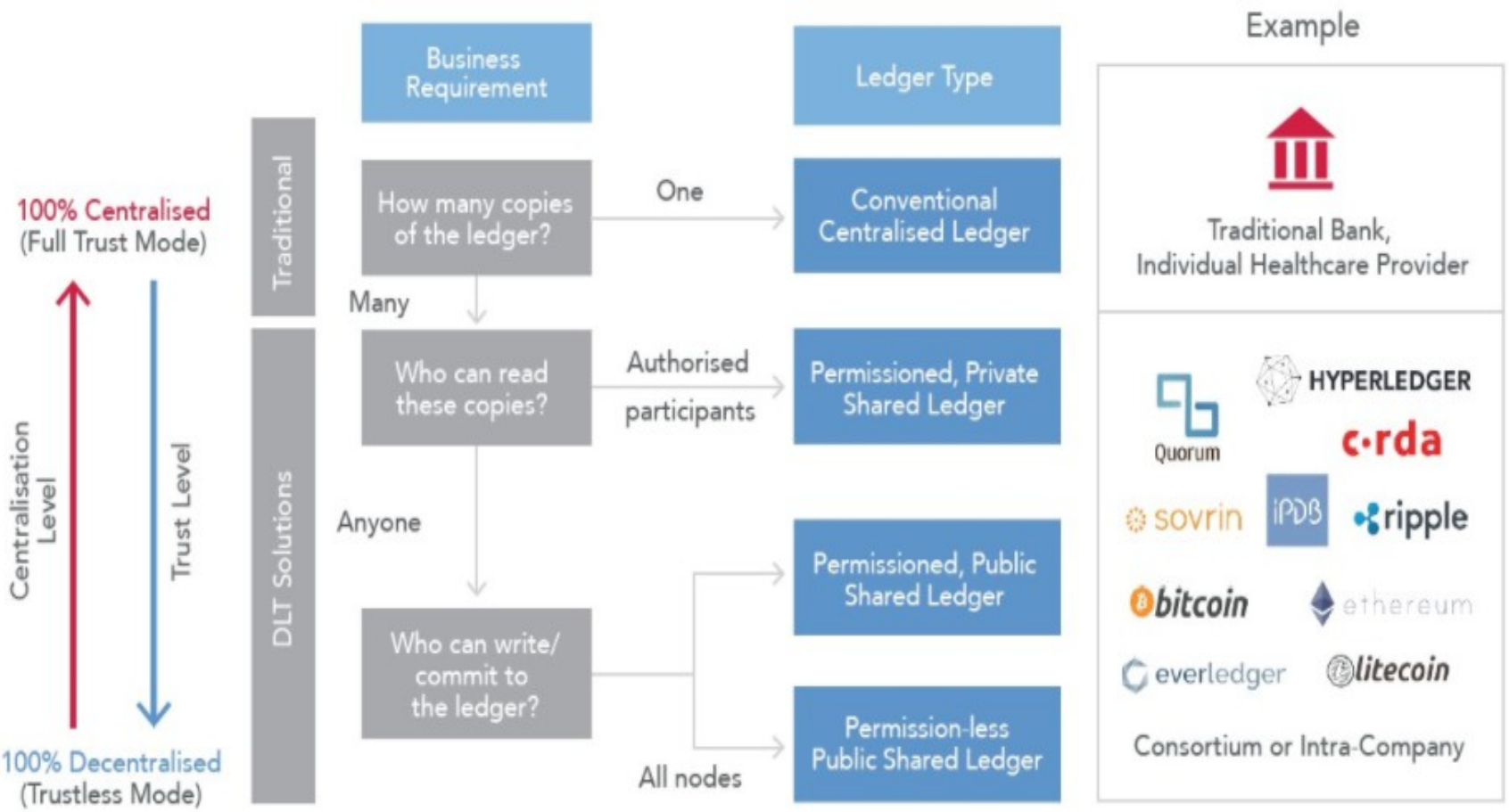
PERMISSIONED BLOCKCHAIN FOR SUPPLY CHAIN



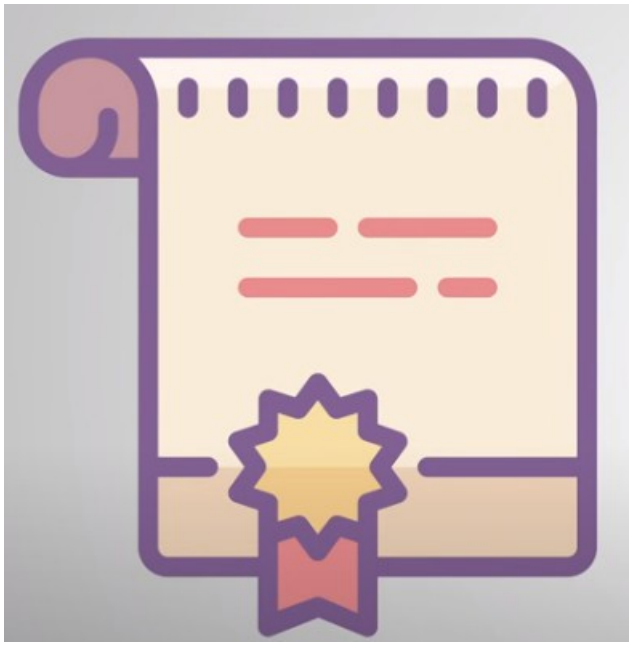
PERMISSIONED BLOCKCHAIN: WHAT IS DIFFERENT?

- 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



SMART CONTRACTS FOR DUMMIES



- what is a smart contract?
- use cases

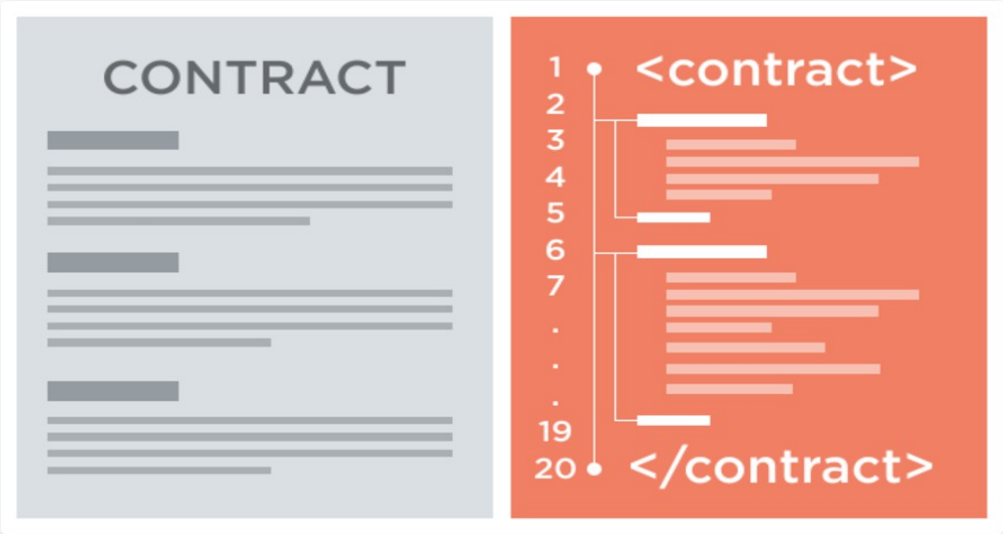
THE FIRST IDEA OF SMART CONTRACTS

- 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

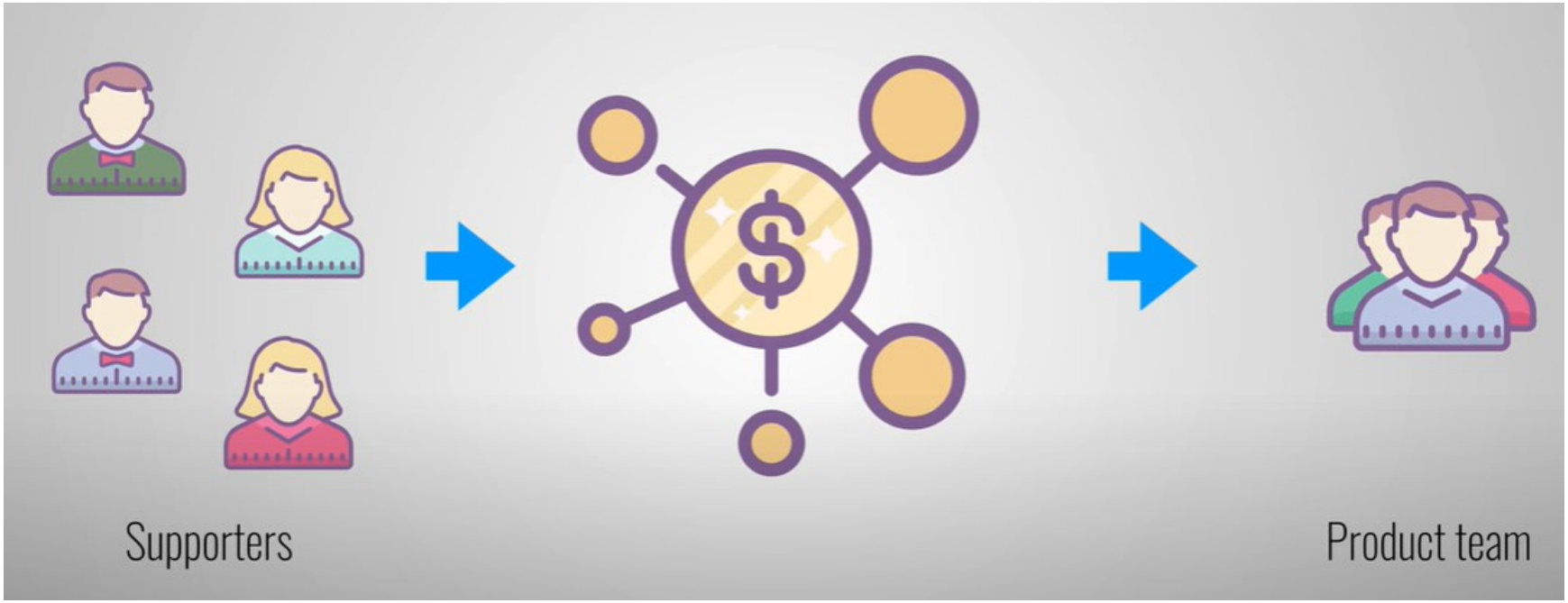
- a piece of software (program) written on a blockchain.
- all parties can view the contract, but it is not possible to change the contract (the code) after it has been deployed on the blockchain.
- as a result, the parties do not necessarily have to trust each other
  - they can rely on the contract 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

- *Kickstarter*: a large crowdfunding centralized platform
- product teams can go to *Kickstarter*, create a project and start collecting funds from other supporters who do believe in their idea
- essentially a third party that sits between start-up and supporters

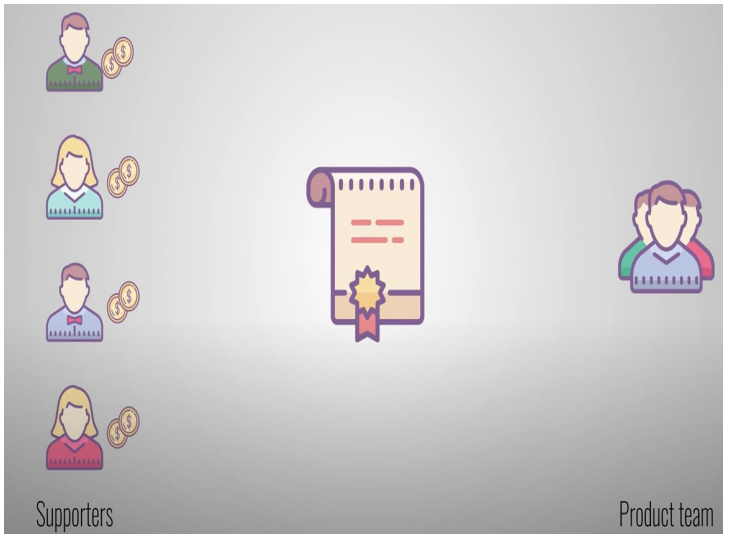


CENTRALIZED CROWDFUNDING: TRUSTING A THIRD PARTY

- 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 their money if the project is succesfull
- if the project has not been funded supporters gets a refund

EXAMPLE 1: DECENTRALIZED FINANCE

- no third party: instead program a smart contract



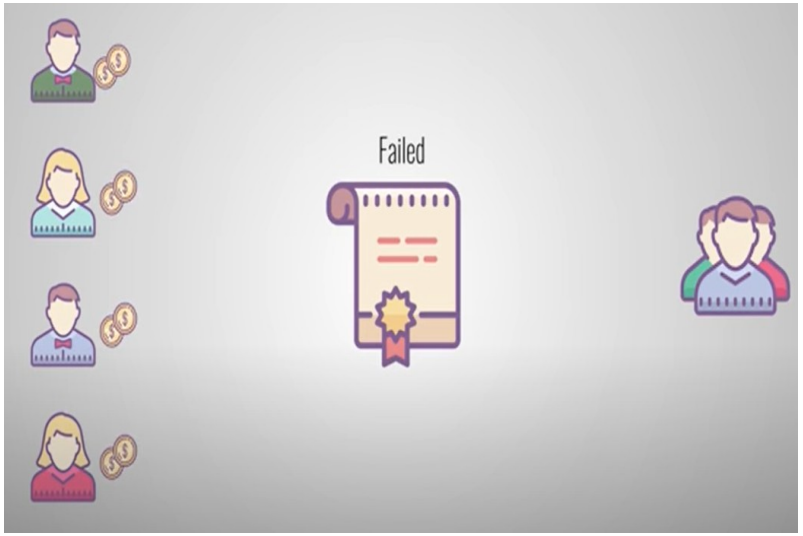
- the supporters can transfer their money to the smart contract

- it holds all the received funds until a certain goal is reached

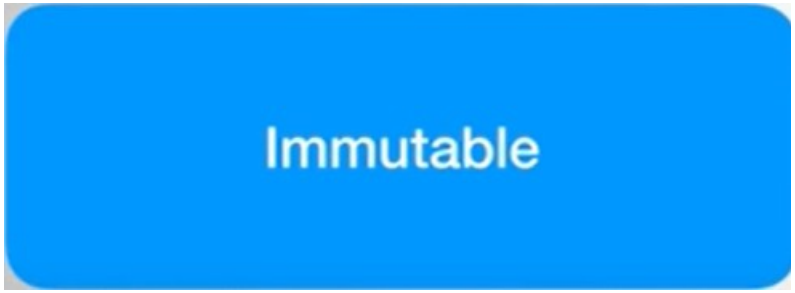
EXAMPLE 1: DECENTRALIZED FINANCE

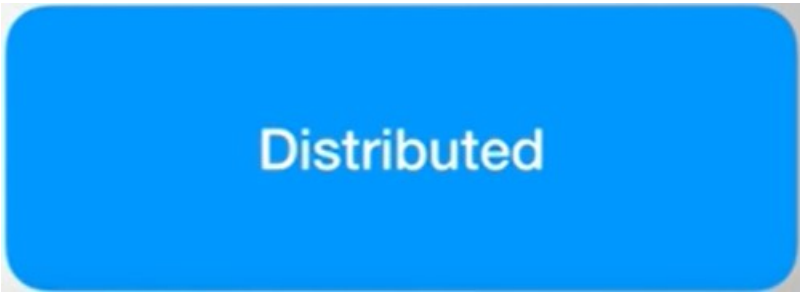
- no third party: instead program a smart contract
- if the project gets fully funded
  - the contract automatically passes the money to the creators

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



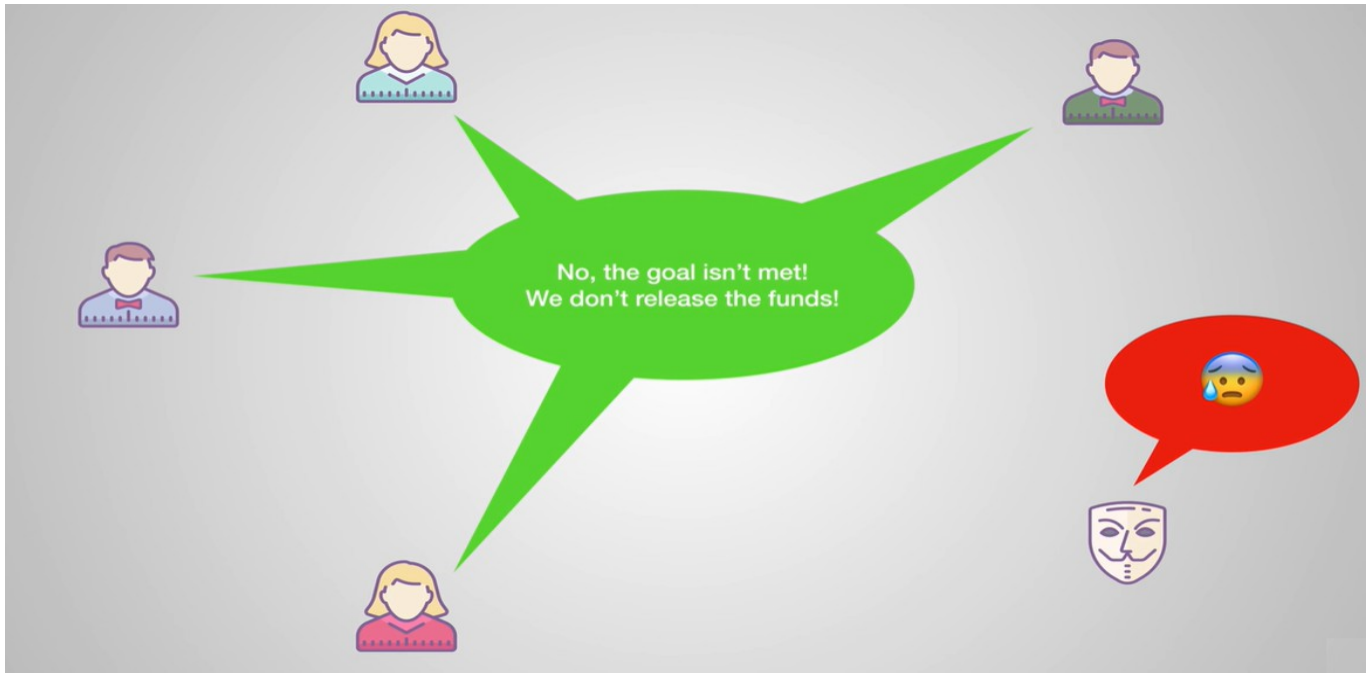
DEFI: A SMART CONTRACT BASED SOLUTION





- 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

DEFI: A SMART CONTRACT BASED SOLUTION

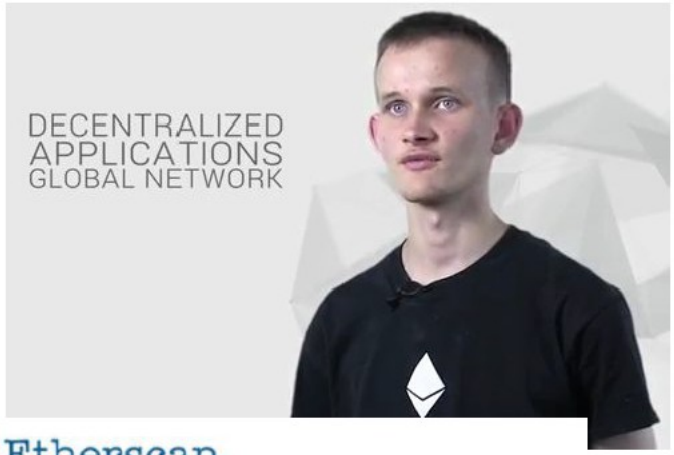


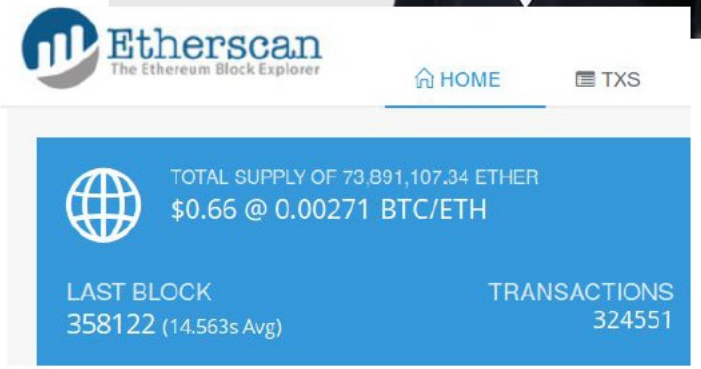
- a single person (or adversary) cannot force the contract to release funds

  because other nodes on the network will spot this attempt and consider it invalid
  - this hold if and only if the 51% of the nodes are honest!
  - in this case tampering with smart contract is almost impossibile!

ETHEREUM

- crowdfunded ~$20M in ~ a month
- popularized a grand vision of "generalized" cryptocurrency
- smart contracts:
  - implement a protocol that uses a block-chain
  - programmable through Turing complete language
    - `Solidity`
    - `Serpent`,...
  - executed by all nodes: consensus as agreement on the results of computation





ETHEREUM VERSUS BITCOIN

Bitcoin

Ethereum

CONCLUSIONS

- target for blockchains are applications that
  - require shared common, append-only database with limited capacity
  - with multiple participants with varying degrees of trust amongst them
  - that must run in a distributed manner
  - that would require a complex settlement process with a trusted third party
  - needing integrity, authentication, and non-repudiation
  - governed by precise rules that do not change and are simple to encoding
  - requiring transparency (as opposed to privacy)

THE PISA DISTRIBUTED LEDGER LAB

- Permanent or semi-permanent position
  - *Laura Ricci* full professor
  - *Paolo Mori*, IIT CNR, Pisa
  - *Barbara Guidi*, RTD-B
  - *Damiano Di Francesco Maesa*, RTD-A
  - *Andrea De Salve*, ISASI, Lecce
- Post-doc
  - *Andrea Michienzi*
- 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