

P2P Systems and Blockchain

Spring 2025,

Instructor: Laura Ricci

laura.ricci@unipi.it

Lesson 1:

Introduction

17/2/2025

GENERAL INFORMATION

- **Instructor:** Laura Ricci
- **Didactic Support:** Domenico Tortola
- **Credits:** 6
- **Moodle Link:** <https://elearning.di.unipi.it/course/view.php?id=1044>
- **Timetable:**
 - 11:00- 13:00 Monday, Room C1
 - 9:00 - 11:00 Friday, Room L1
- **Available for:**
 - master in Computer Science
 - master in Computer Science and Networking
 - free choice course for other masters
- **Prerequisites:**
 - Computer Networks
 - Algorithm design
 - Basic concepts of applied cryptography, but we will review them in the course

EXAMINATION METHOD

- final project or presentation + oral exam
- final project
 - developing a blockchain-based application: smart contract, Solidity
 - project proposals from the students are accepted, if positively evaluated
- presentation
 - read a set of paper assigned by the teacher
 - prepare a 20 minutes presentation with slides
- oral exam
 - discussion of the project + questions on topics not covered by the project
 - presentation + questions on topics not covered by the presentation

FINAL PROJECT

- languages for developing the project
 - Solidity for the development of the smart contracts on Ethereum
 - JavaScript/TypeScript: used for test script and to automate the development process of Hardhat.
- environment for the project: Hardhat
 - an open source environment to build and test smart contracts
 - includes a Solidity compiler, a testing framework, a debug framework, functionalities for the smart contract deployment
- some examples of past projects
 - playing MasterMind or BattleShip on the blockchain
 - smart auctions
 - smart lotteries
 - content trade and reward on the blockchain.

OTHER PROJECT PROPOSALS

- writing smart contracts on Ethereum for
 - games
 - tokens
 - voting
 - gambling
 - auctions
 -
- can be run on the local computer, but also exploiting the Ethereum test nets

FINAL PRESENTATION

- some examples
 - conducting an analysis of security and privacy threats associated with the use of Bitcoin, getting inspiration from a set of scientific paper
 - examine one Distributed Hash Table and how it is used in real case scenarios.
 - review the main analysis techniques for detecting scams on the Bitcoin blockchain
- show the connections between the selected papers and the topics covered in the course

COURSE STRUCTURE

- focus on:
 - basic principles of P2P systems (DHT, IPFS)
 - block-chains
 - Bitcoin, Ethereum: in depth
 - smart contract programming
 - scaling blockchain: layer-2 technologies
 - applications: DeFi, token (ERC-20, NFT), supply chains, SSI: Self Sovereign Identity
- to define these systems, we will exploit
 - distributed algorithms
 - cryptographic methods
 - probabilistic data structures

- **overlays:** application level virtual networks
 - **unstructured overlays:** Flooding, Random Walks, epidemic protocols
 - **structured overlays:**
 - Distributed Hash Tables (DHTs)
 - Kademlia, applications
 - IPFS (Internet Planetary File System): a distributed file storage protocol that allows computers all over the globe to store and serve files as part of a huge peer-to-peer network.
 - Ethereum P2P Network

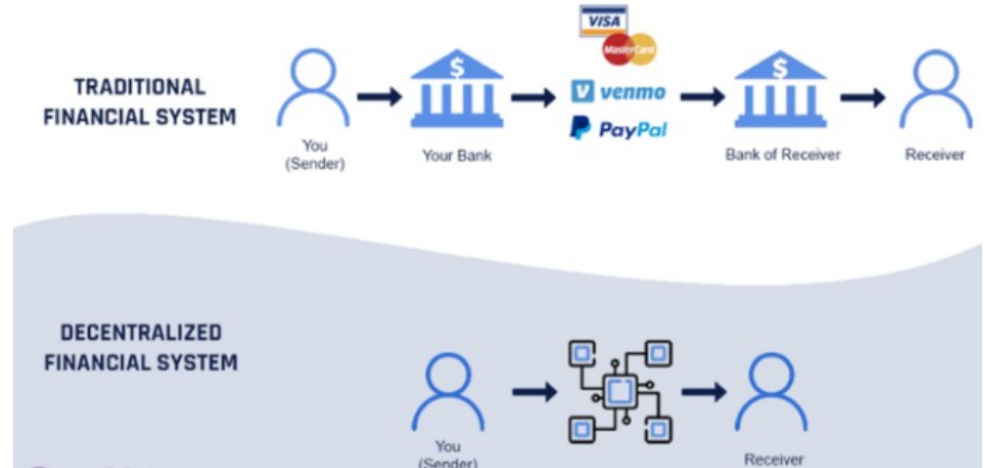
PROGRAM: BLOCKCHAIN FUNDAMENTALS

- cryptographic tools and data structures (from an applicative point of view)
 - digital signatures
 - cryptographic hash
 - Bloom filters (review of basic concepts)
 - authenticated Data Structures
 - Merkle trees
 - Merkle Patricia tries
 - other techniques: Zero-knowledge, ZK-Snarks: Zokrates,
- Consensus Protocols
 - Nakamoto Consensus: Proof of Work
 - Proof of Stake
 - Byzantine agreement, Practical Byzantine Fault Tolerance

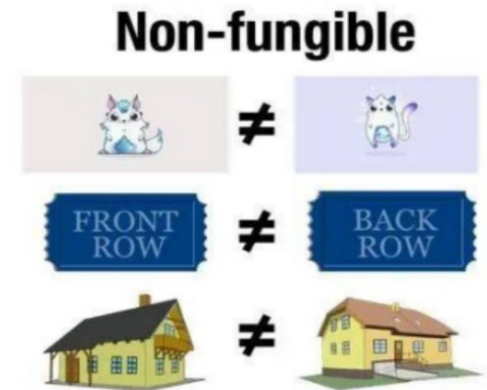
- Bitcoin
 - structure of transactions and blocks
 - Nakamoto consensus: PoW and Mining
 - double spending
 - the Bitcoin P2P Network
 - the Bitcoin ecosystem: wallet, light weight clients
 - pseudo-anonymity: traceability and mixing
- Ethereum
 - the blockchain: differences with respect to Bitcoin
 - smart contract programming in Solidity
 - gas, smart contract vulnerabilities
- Blockchain scalability
 - Off-chain Payment Channels: Lightning Network
 - Layer-2 solutions: cross-chain protocols

PROGRAM: BLOCKCHAINS APPLICATIONS

- Cryptocurrencies
- DeFi: Decentralized Finance



- tokens
- supply chain
- Self sovereign identity (SSI)



FURTHER TOPICS (IF THERE WILL BE TIME)

- Permissioned Blockchain
 - HyperLedger
- Oracles and cross chain bridges
- Exchangers: centralized and decentralized
- Stablecoins

Mandatory

- lesson slides (download new slides, slides of the previous years are obsolete)
- tutorial and material published on the course page

Reference books:

- *Andreas M. Antonopoulos and Gavin Wood, [Mastering Ethereum](#), Implementing Digital Contracts, O'Really*
- *Andreas M. Antonopoulos, [Mastering Bitcoin](#), Unlocking Digital Cryptocurrencies, O'Really*
- *Andreas M. Antonopoulos, [Mastering Lightning Network](#), Unlocking Digital Cryptocurrencies, O'Really*
- *Kalle Rosenbaum, [Grokking Bitcoin](#), April 2019, Manning*

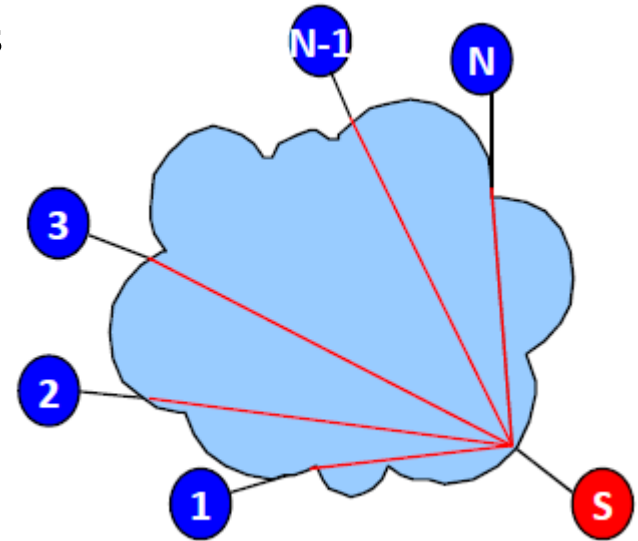
OTHER REFERENCES

- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, [Bitcoin and Cryptocurrency Technologies](#), Princeton University Press
- Saravanan Vijayakumaran, [An Introduction to Bitcoin](#), Course Notes

THE CLIENT SERVER PARADIGM



- | | |
|--|---------------------------------|
| • runs on end-hosts | • runs on dedicated hosts |
| • on/off behavior | • always on |
| • service consumer | • service provider |
| • issue requests | • receive requests |
| • do not communicate directly among them | • satisfies all client requests |
| • need to know the server IP address | • need a fixed IP (or DNS name) |



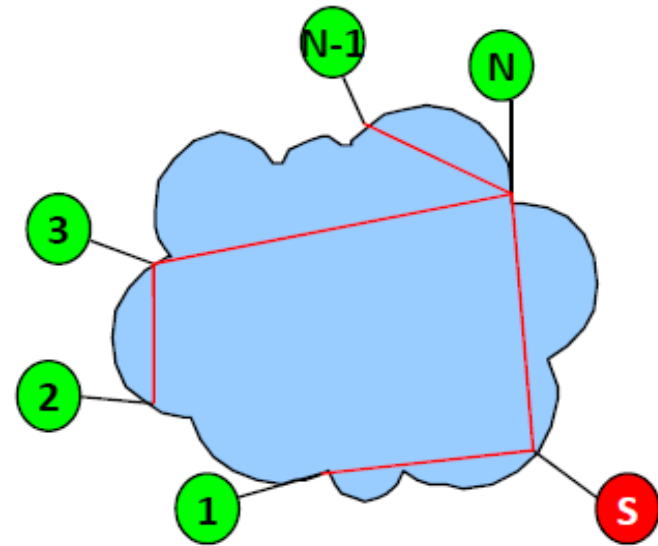
THE PEER TO PEER PARADIGM



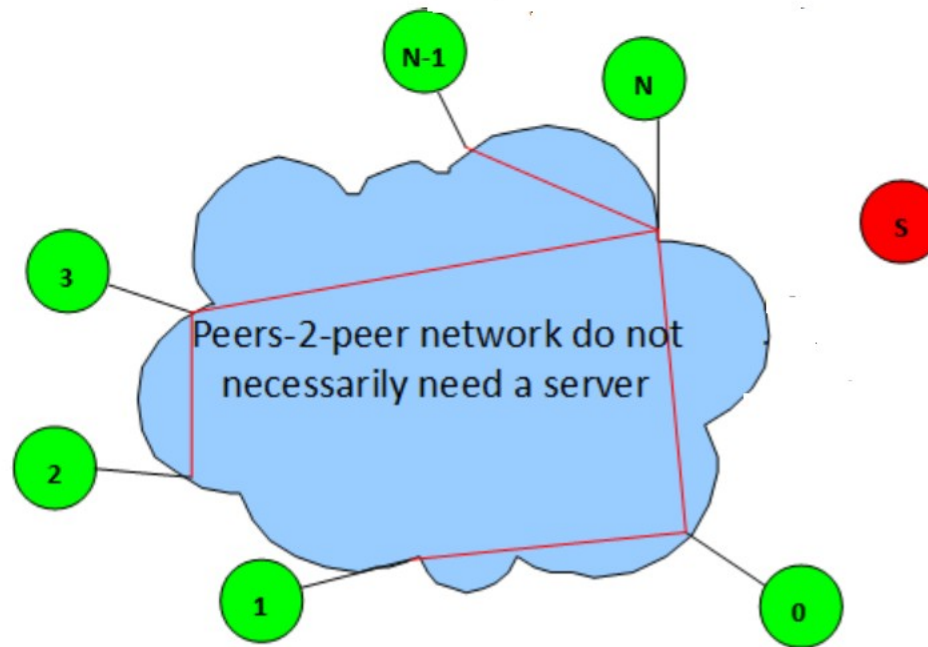
runs on end-hosts

- on/off behavior, handle **churn**
- need to join
- need to discover other peers
- service providers and consumers
- communicate directly among them
- need to define communication rules
 - prevent free riding
 - incentivate participation ... and

reciprocation



THE PEER TO PEER PARADIGM



notice that:

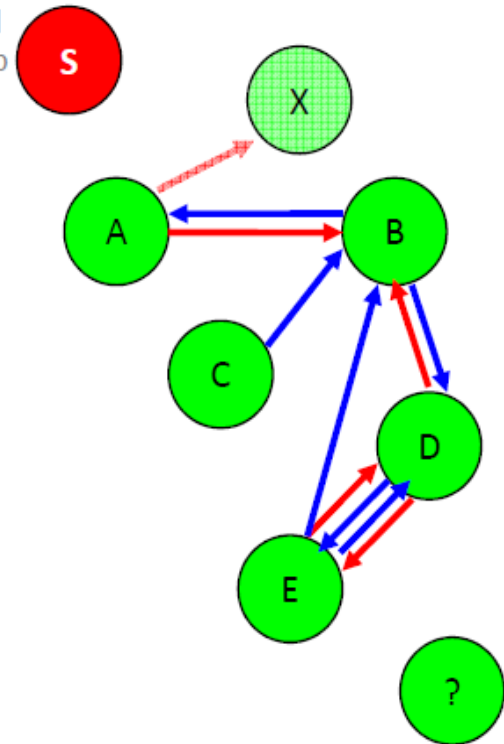
- servers are still typically needed for bootstrap
- but servers aren't needed for resource sharing

THE PEER TO PEER PARADIGM



- runs on end-hosts
- On/off behavior, handle churn
- Need to join 
- Need to discover other peers 
- Service providers and consumers 
- Communicate directly among them 
- Need to define communication rules
- Prevent free riding 
- Incentivate participation and reciprocation , 

Servers used
for bootstrap



- Definition 1:

A peer to peer system is a set of **autonomous entities** (peers) able to **auto-organize** and sharing a set of distributed resources in a computer network. The system exploits such resources to give a service in a **complete or partial decentralized way**

- Shared Resources:

- Ledgers
- Read/Write storage space (Distributed File System)
- Computing power
- Bandwidth

Definition 2:

A P2P system is a distributed system defined by a set of nodes interconnected able to **auto-organize** and to build different topologies with the goal of **sharing resources** like CPU cycles, memory, bandwidth. The system is able to adapt to a **continuous churn** of the nodes maintaining connectivity and reasonable performances without a centralized entity (like a server)

RESOURCE SHARING

- P2P: is relative to **give** and **receive** from a community. Each peer shares a set of resources and obtains, in return, a set of resources/services
 - one of the first scenarios:
 - share musics (audio files) and obtain music, in return (Napster, Gnutella, Bittorrent, ...)
 - a peer like a client and like a server (symmetric functionality = **Servent**)
- a peer can **offer for free** a resource, for instance to participate to a project
 - searching extra-terrestrial life
 - cancer therapies research
 - contributing to the maintenance of a **distributed ledger**
- a peer can be rewarded for contributing to the management of the network
 - Bitcoin miners
- the shared resources are at “the border” of Internet, they are directly shared by the peers, there are no “special purpose nodes” for their management.

RESOURCE SHARING

- the peers' connection is **transient**: the connections and disconnections to the network are very frequent
- the resources offered by the peers are **dynamically** added and removed
- each peer is paired with a different IP address for each connection to the system
 - a resource cannot be located by a static IP
 - new addressing mechanisms have to be defined, at the application level, not at the IP level

P2P file sharing

- file sharing: light weight/ best effort
- persistence and security are not the main goal
- anonymity is important
- examples:
 - Napster
 - Gnutella, KaZaa
 - eMule
 - BitTorrent
- Cyptocurrencies and Blockchains
- Distributed Social Networks
- Distributed file System: Internet Planetary File System

FILE SHARING: THE FIRST P2P 'KILLER APPLICATION'

- a user U has a P2P client on its notebook
- the connection to Internet is intermittent : each time the user obtains a new IP address for each new connection
- the user stores the shared files in a directory and pairs each file with a set of keys able to identify it (for a song: title, author, publication date,...)
- U is interested in finding a song and sends a query to the system
- U finds and shows the information about the other peers which own the required song
- U chooses a peer P among these (according some criteria, we will see in the following)
- the file is copied from the notebook of P to that of U
- while U is performing the download, other users can upload the parts of the file already downloaded by U and put in the shared directory

FILE SHARING: THE FIRST P2P 'KILLER APPLICATION'

- **File Sharing:** begins with the rapid success of Napster, at the end of nineties, about 10 years later than Wide Web
- **First generation: Napster**
 - introduces a set of servers where the users register the **descriptors** of the files they are going to share
 - only the content transmission (download/upload) exploits a P2P protocol, content search is centralized
 - the presence of a centralized directory has been the 'Achille's heel' of this application
 - it was convicted because it did not respect the law on the copyright
 - it could have detected the content exchanged between the users through the analysis of the centralized directory

FILE SHARING: THE FIRST P2P 'KILLER APPLICATION'

- File sharing: second generation
 - no centralization point
 - both file search and content transfer are **completely distributed**
 - Gnutella, FastTrack/Kazaa, BitTorrent
- Side effects of the diffusion of file sharing applications: change of the way users music is accessed by the
 - from CD to **online music**
 - iTunes

FILE SHARING: THE FIRST P2P 'KILLER APPLICATION'

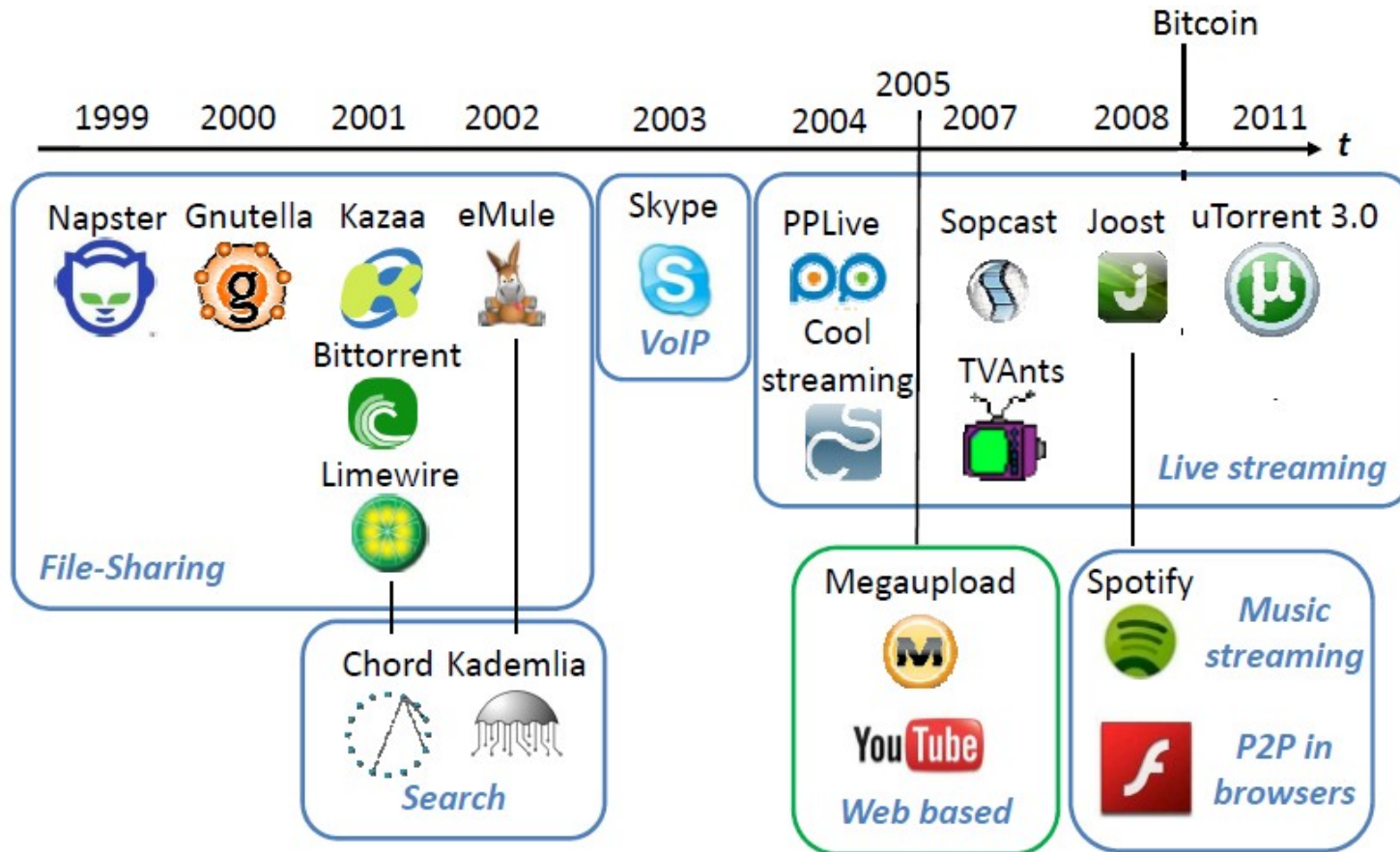
The P2P client behaves like a **servlet** and allows:

- the user to define a directory, in its file system, where the file it is going to share are stored.

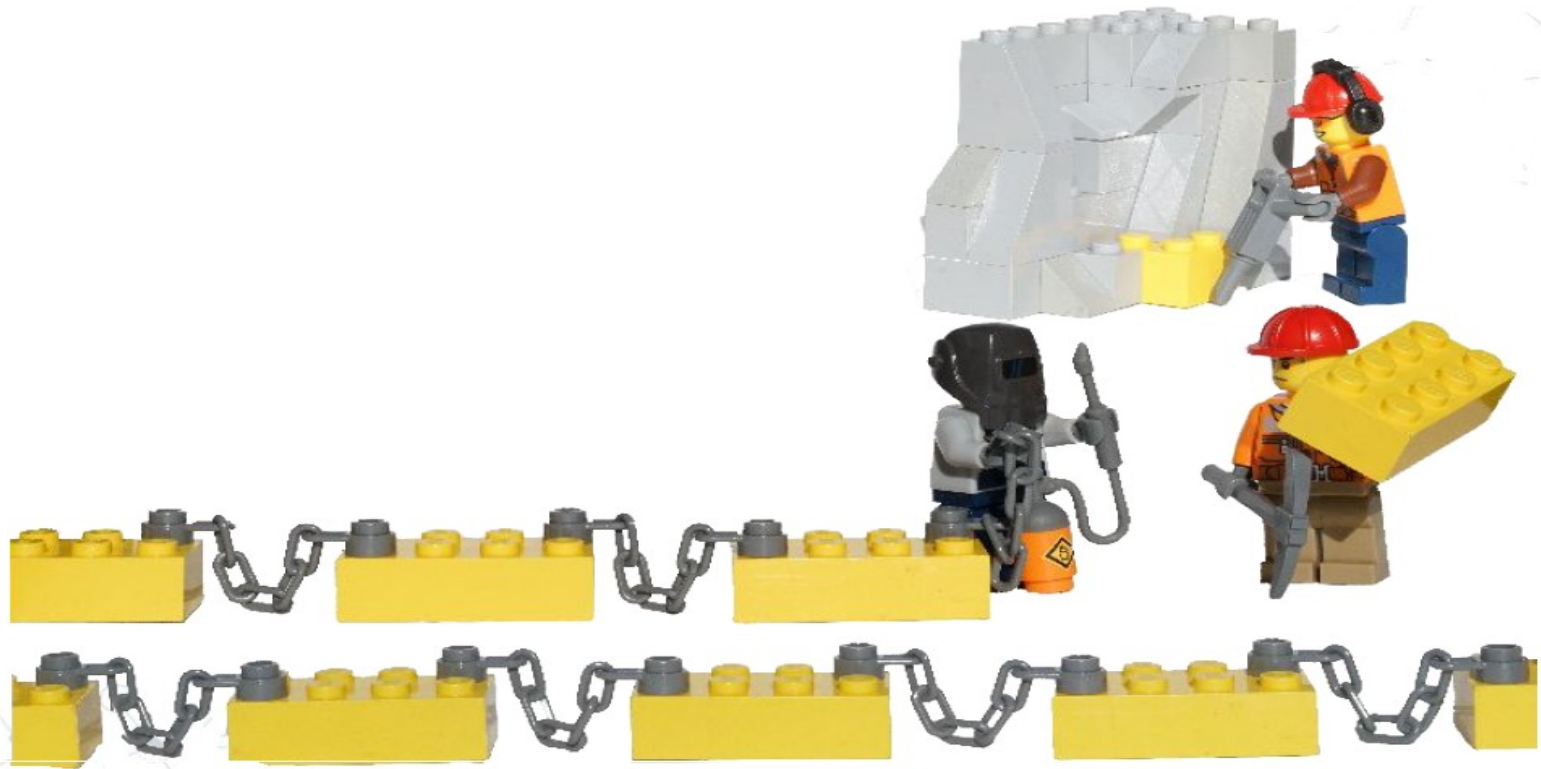
Each other peer of the P2P network may download files from this directory

- a peer behaves like a **web server**
- the user to download files from the shared directory
 - a peer behaves like a **client**
- the user to find the content it is interested in, through queries submitted to the system
 - this functionality is similar to that of **Google**

FROM FILE SHARING TO BITCOIN



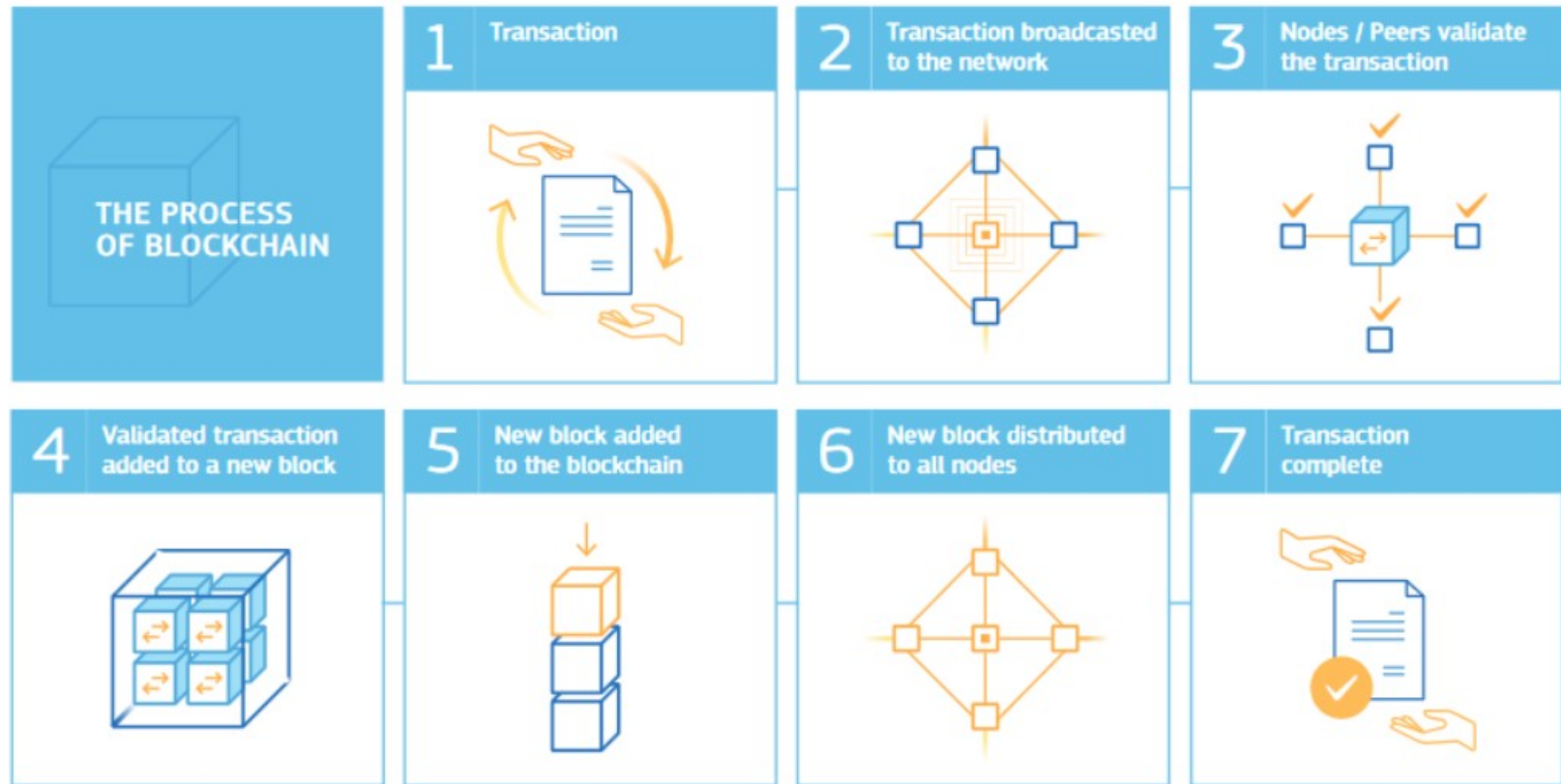
BLOCKCHAIN: THE SECOND KILLER APPLICATION



BLOCKCHAIN 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 IN A NUTSHELL



THE BLOCKCHAIN BASIC TECHNOLOGIES

- Digital signatures (e.g. public-key cryptography)
 - provide authentication
- Cryptographic hash functions (e.g. hash chains of data transactions)
 - provide tamper-resistant immutability
- Replication (e.g. full copies stored everywhere)
 - provides availability
- Distributed consensus amongst mutually trusting or distrusting replica
 - provides integrity and decentralized control

13 YEARS OF A DISRUPTIVE TECHNOLOGY



“THE MOTHER OF ALL BLOCKCHAIN”: BITCOIN



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

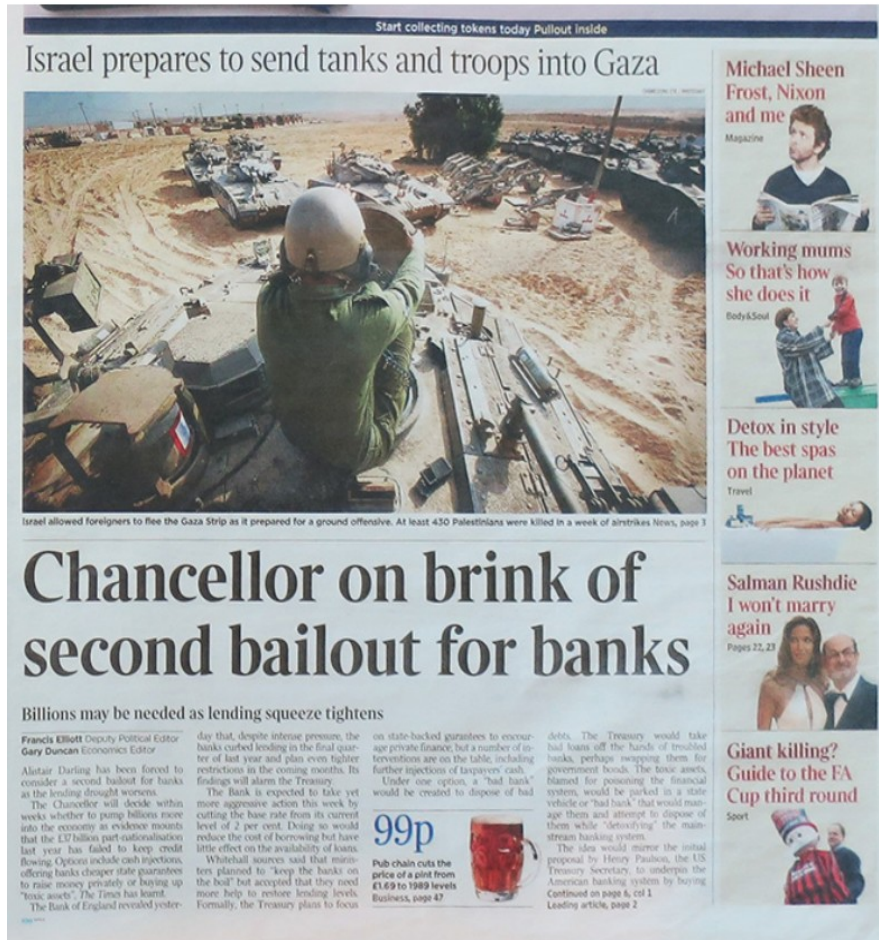
Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of

```
bitcoin-0.1.0.rar  
bitcoin-0.1.0.tgz
```

Paper published in October 2008: more than ten years of Bitcoin!

BITCOIN: P2P CRYPTOCURRENCIES

The Times 03/Jan/2009 Chancellor on brink of second bailout for banks.










































- written in the first coinbase transaction by Satoshi
- in italian: “Il Cancelliere dello scacchiere Alistair Darling ipotizza un secondo salvataggio per le banche”
- ending the control of banks and government on your money
- witness the date of the first transaction

- payments directly done between the users, no a centralized financial entity which guarantees the electronic payment, lower costs
- the cypherpunk vision: “*we can revolutionize our world by building secure protocols*”
- new motivation and tools for learning traditional concepts in computer security
- “Blockchain technology,” a related and more general concept
 - a new technology for developing secure applications in an untrusted environment
 - Ethereum and many others
 - affects many processes, companies and societies

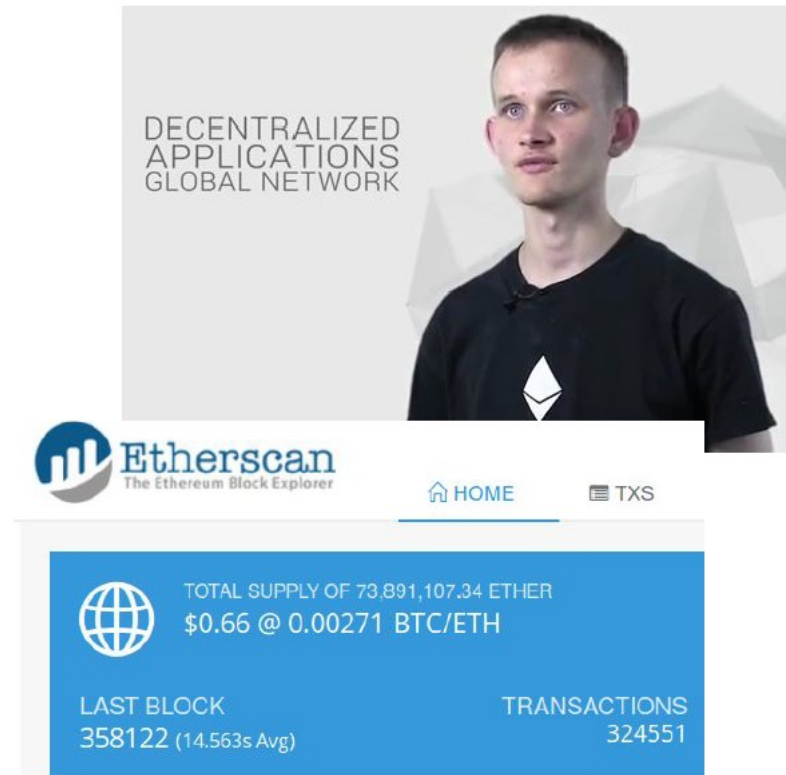
CRYPTOCURRENCES: A COMPLEX ECOSYSTEM

- different actors are involved

Cryptocurrencies	Infrastructure Providers	NFT Marketplaces	Exchanges	Risk Marketplaces
  ETH BTC   USDT LTC   BNB TRX   XRP NIM	 Tezos  ethereum  CARDANO  Polkadot  HYPERLEDGER CØSMOS  TRON  Chainlink  EOS™  ripple  Stellar	          asy ^{nc} .	coinbase  BINANCE  Blocktrade  kraken  BitGo.  Coincheck BITFINEX   GEMINI  CURVE	 ETHERISC  bridge.  NAYMS Nexus  Mutual

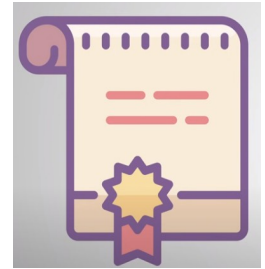
ETHEREUM

- 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
 - executed by all nodes:
 - consensus as agreement on the results of computation



SMART CONTRACT USE CASE: FLIGHT DELAYS

- Bob is at the airport and his flights is delayed
- he has an insurance that guarantees a reimbursement if his flight is delayed
- the insurance company has deployed a smart contract on Ethereum. The smart contract
 - is connected to the database
 - monitors the flight delays
 - as soon it verifies a delay of at least X minutes (or hours)
 - it automatically generates the reimbursement
 - the reimbursement (in crypto) is moved to Bob's wallet



THE ETHEREUM BLOCKCHAIN

- introduces smart contracts to be executed by blockchain nodes
 - Turing-complete : can solve any computational problem
 - in Bitcoin scripts have only limited computational power
 - gas to avoid denial of service
- treats blockchain and its nodes as a single, global, replicated, consistent computer
- entire state machine, its code, and its input/output replicated and executed in a consistent manner
- but many other blockchain have been recently proposed...

FINDING THE WAY IN THE BLOCKCHAIN JUNGLE

Trade-off: Open vs. Transaction volume		Read Access	
		Everyone Public <i>Medium overhead</i>	Restricted Private <i>Low overhead</i>
Write Access	Everyone Permissionless <i>Large overhead</i>	1) Public & Permissionless Low Scalability Bitcoin	2) Private & Permissionless <i>Medium Scalability</i>
	Restricted Permissioned <i>Medium overhead</i>	3) Public & Permissioned <i>High Scalability</i>	4) Private & Permissioned Very High Scalability Industrial Blockchain

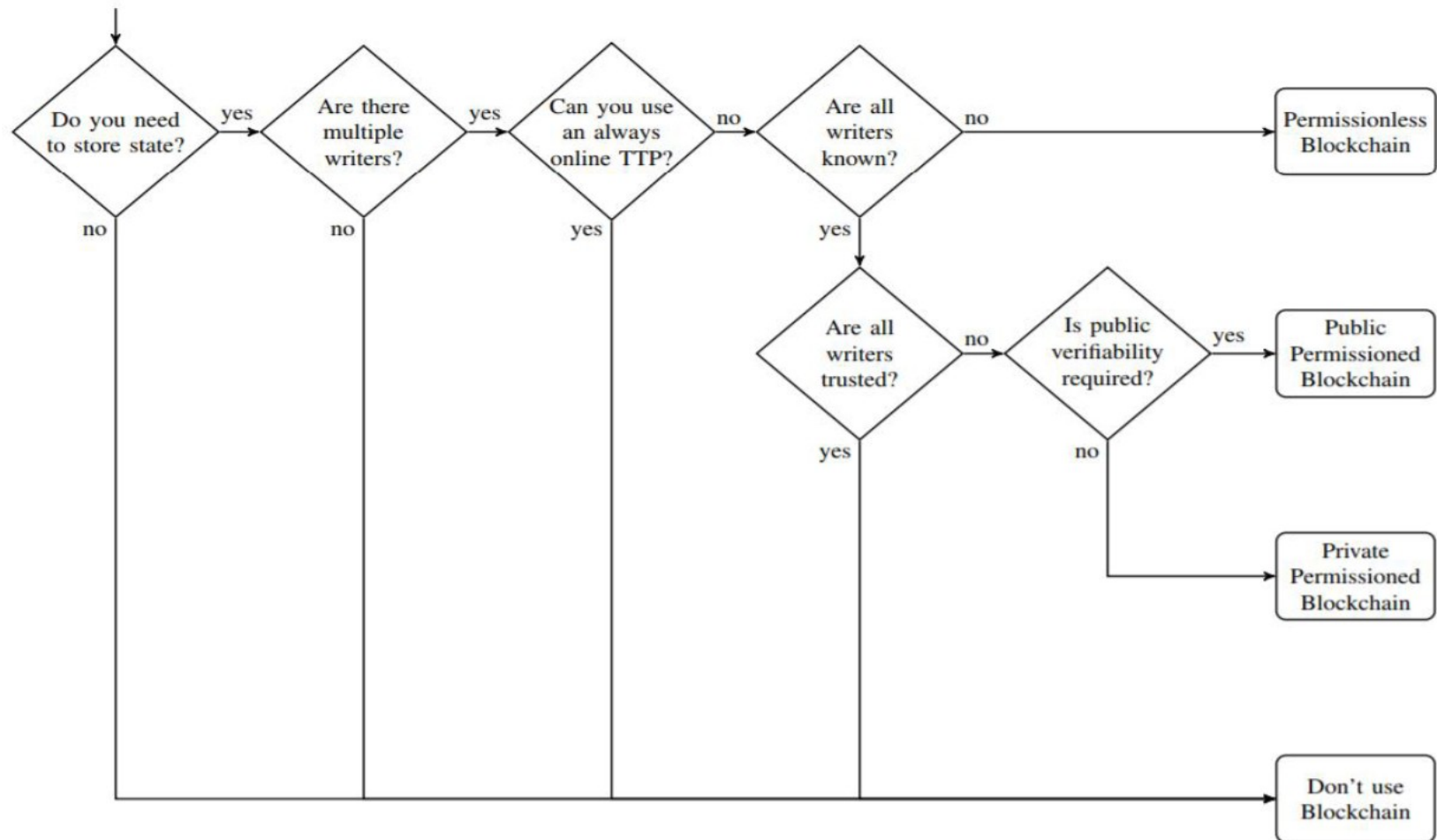
Blockchain Variants and Scalability

FINDING THE WAY IN THE BLOCKCHAIN JUNGLE



WHEN DO YOU NEED A BLOCKCHAIN?

a classical question: instead of a blockchain, why you do not use a database?



Do you need a Blockchain? <https://eprint.iacr.org/2017/375.pdf>

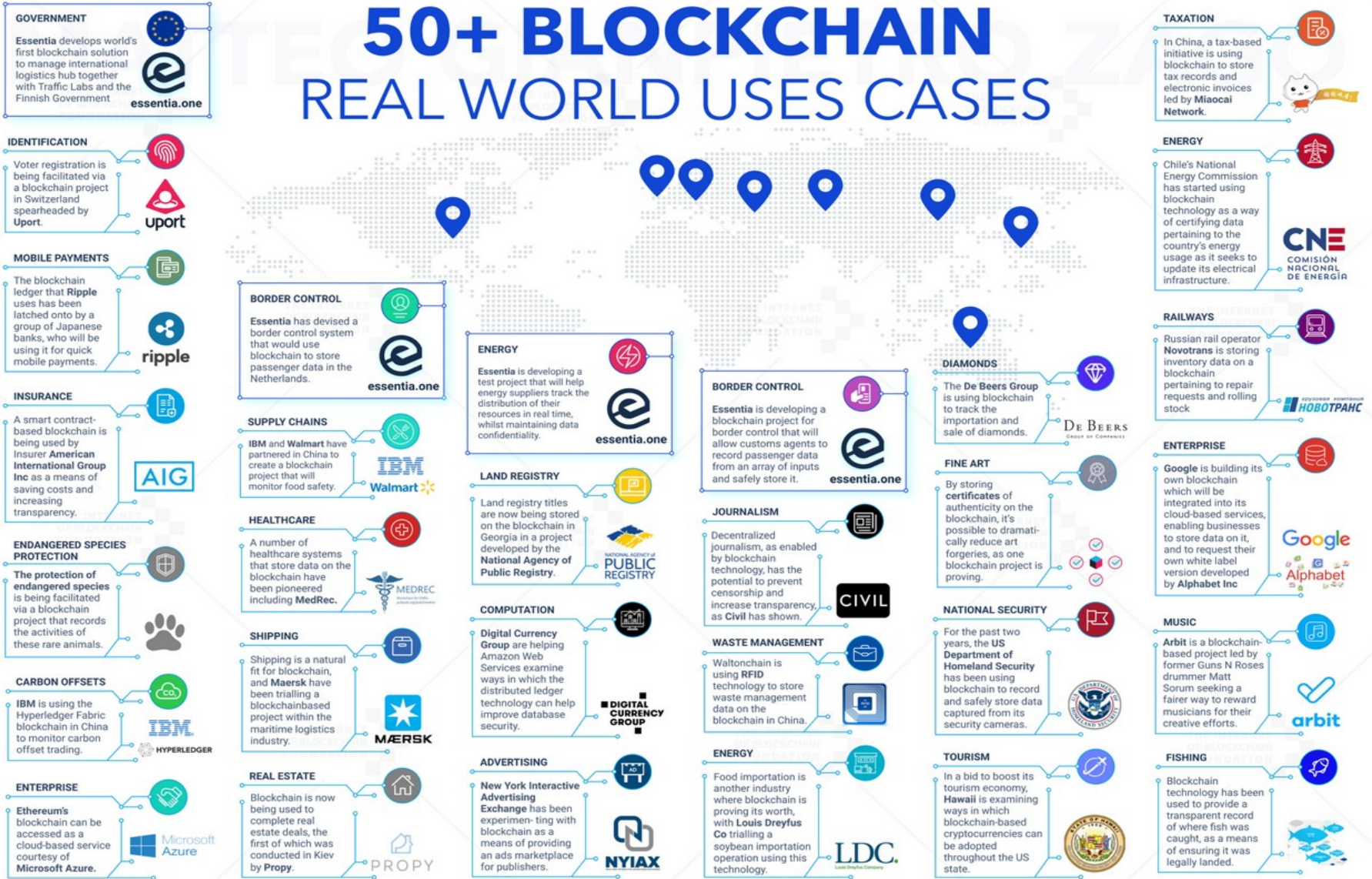
YOU DO NOT NEED A BLOCKCHAIN WHEN....

- if all parties are **known** and **trusted**, **DO NOT** use a blockchain
 - use any number of databases
 - many proposed uses of blockchains for business applications fall in this category!
- if all parties are **known** and **trusted**, but you also need immutability **DO NOT** use a blockchain
 - use databases augmented with cryptographic checksums (e.g. AWS QLDB, Kafka)

TARGETS FOR BLOCKCHAIN

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

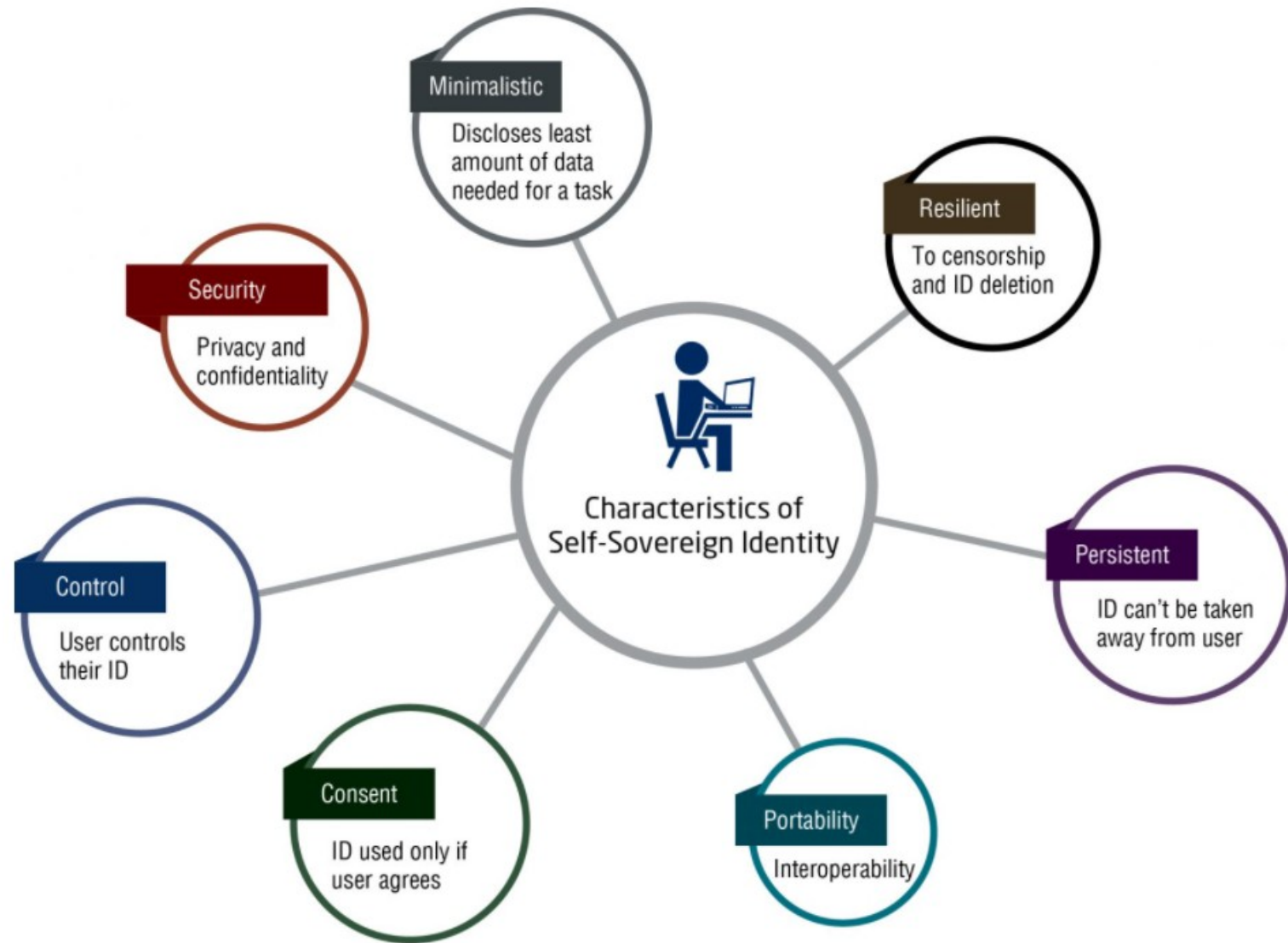
50+ BLOCKCHAIN REAL WORLD USE CASES



Alternative to fiat currencies:

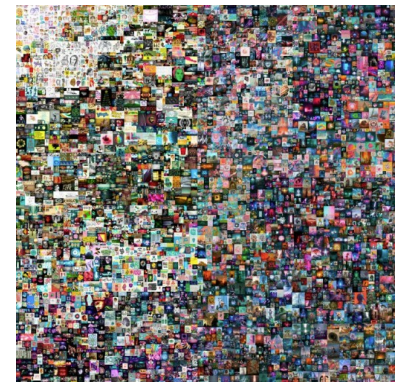
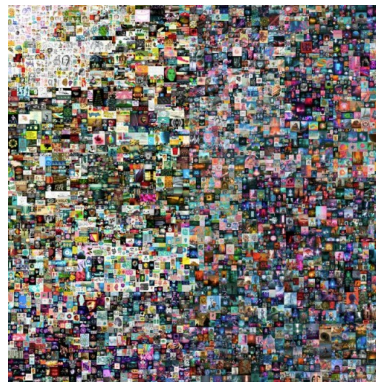
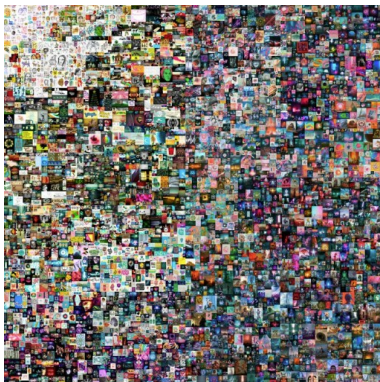
- breaks status-quo where:
 - only government issues money, defines issuing procedures
 - central authorities (banks) decide which transactions are valid and which are not
- fiat currencies decouple supply from a physical good (i.e. gold)
- block-chain typically ties supply to a bounded, virtual good
 - cryptographic bounded
- blockchain records and verifies transfers
- blockchain solves the problem of double spending
- Fungible and non fungible tokens

SELF SOVEREIGN IDENTITY



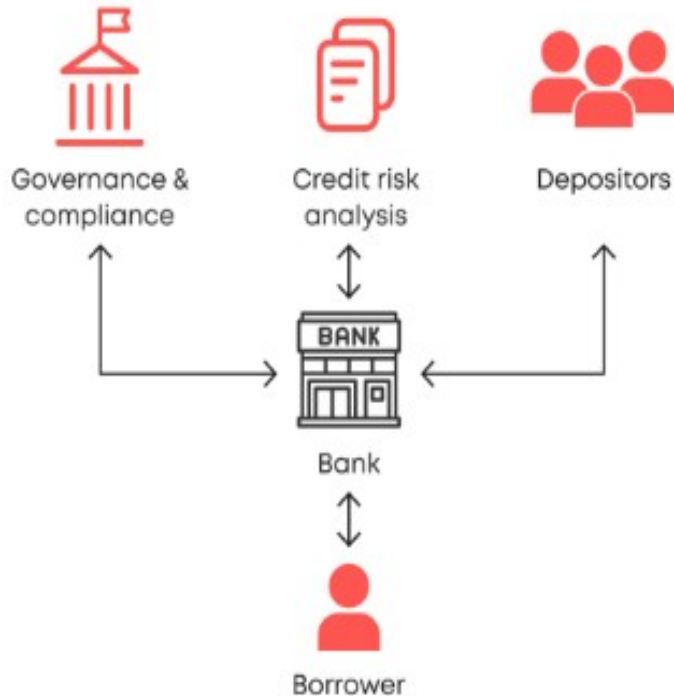
BEEPLE'S NFT: "EVERYDAYS: THE FIRST 5000 DAYS"

- an artwork in a .jpeg file: may be copied and distributed easily
- easy to copy if digital: difficult to find a business model for a art-work, even if for very good artworks!
- how to prove who is the real owner?
 - NFT: non fungible tokens

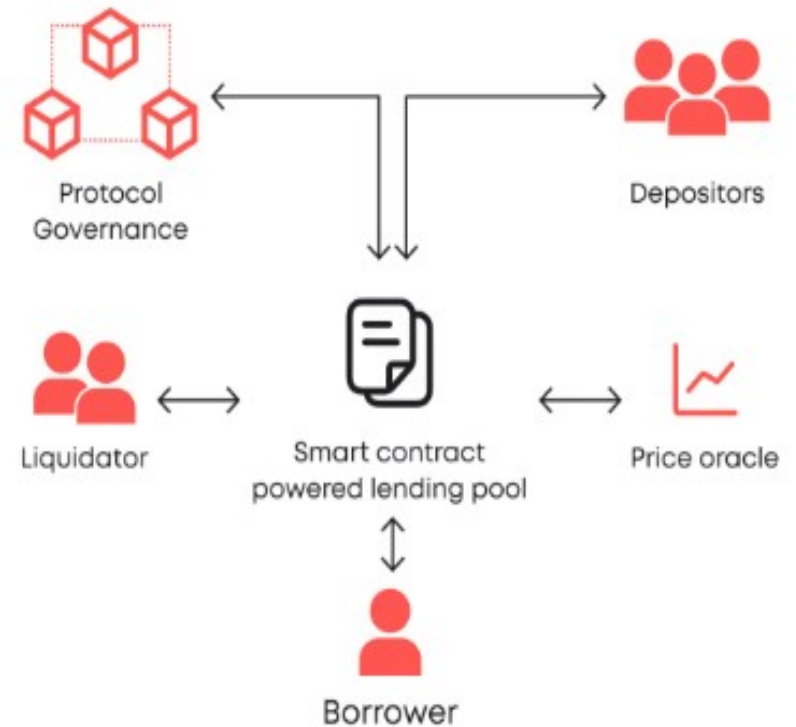


DECENTRALIZED FINANCE

Traditional Lending



DeFi Lending



BLOCKCHAIN APPLICATIONS: SUPPLY CHAIN

- Monitoring and certification of a supply chain

What does Bob say?

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



Carol's factory



Bob's truck



PROVENANCE AND SUPPLY CHAIN: WALLMART

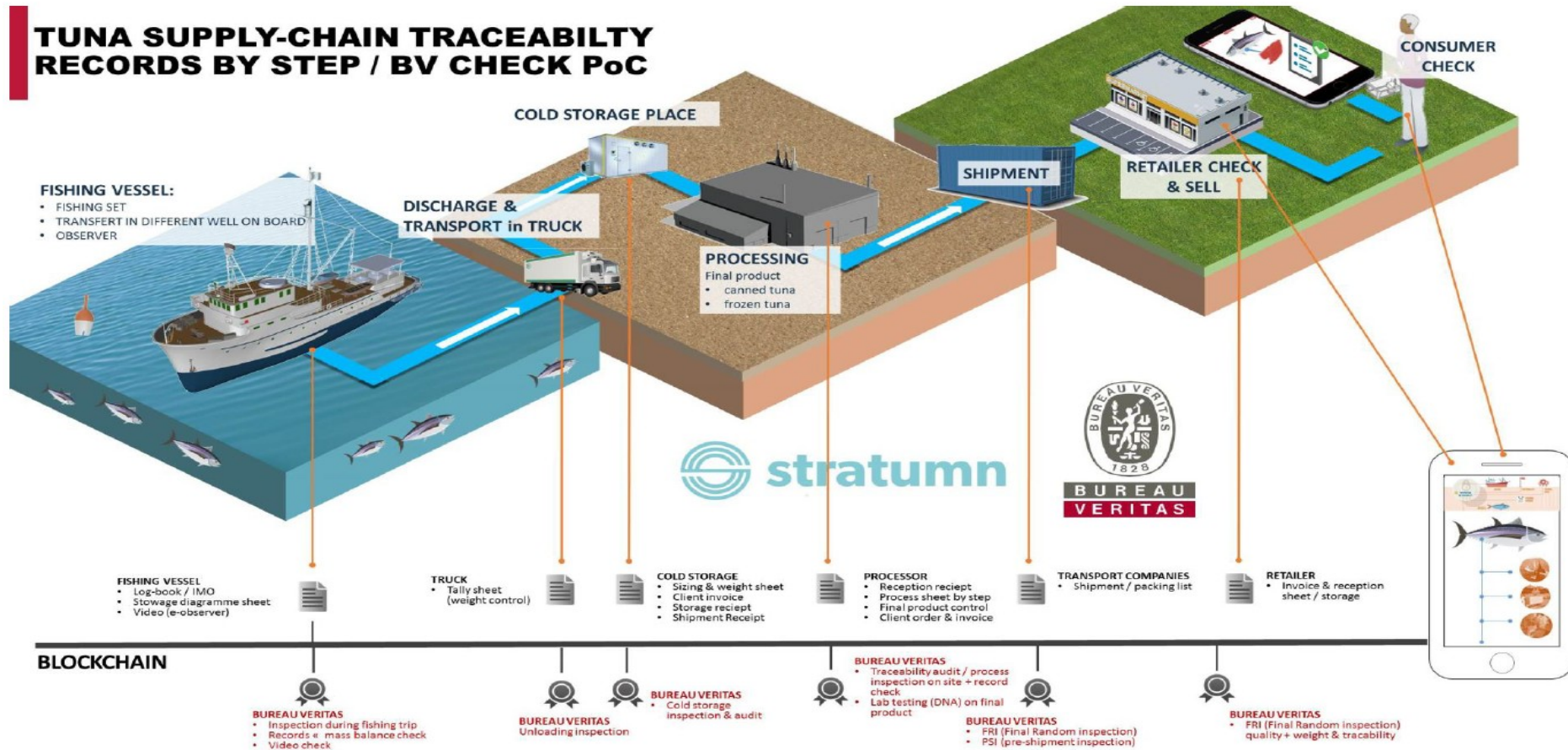
- devoped using Hyperledger technology
- a collaboration between Walmart and IBM
- track farm origin, expiration dates, storage temperature shipping details, parameters taken from sensors,....



PROVENANCE AND SUPPLY CHAIN: FISHING

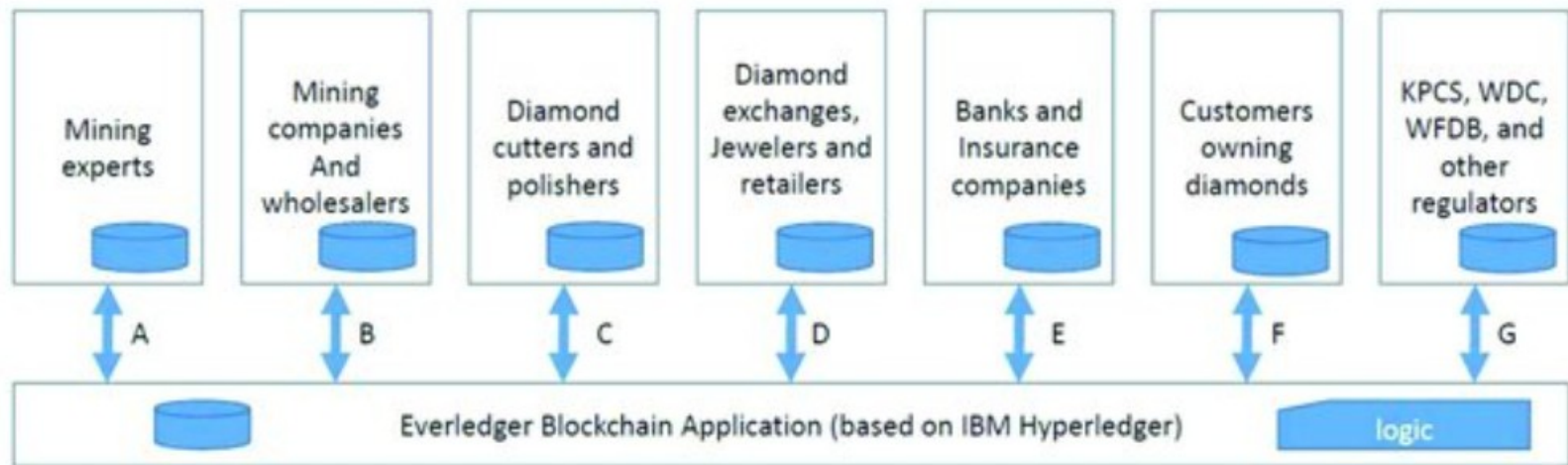
- restaurants can view and verify chain of custody for fish
 - sensors attached to fish can log location/temperature/humidity
- <https://youtu.be/Buw3g8oNG74>

TUNA SUPPLY-CHAIN TRACEABILITY RECORDS BY STEP / BV CHECK PoC



PROVENANCE AND SUPPLY CHAIN: DIAMONDS

- auditing to track provenance and chain of custody for materials and products
- conflict diamonds (e.g. blood diamonds)
- a distributed ledger where all the transaction regarding a diamond are recorded.



INTELLECTUAL PROPERTY

- Digital content owner hashes content together with their identity and commits to the blockchain.
 - if nobody else can prove they published it prior to that commitment, this is evidence that they own it.
 - more convenient than a patent office and allows for you to not have to disclose details of the digital object.



Sign in

Get started

Using Blockchain to Protect Artists and Manage Intellectual Property Law



Marie Gonzalez [Follow](#)

Jun 24 • 5 min read

GoChain offers the use of blockchain technology as a tool to manage and store Intellectual Property rights on a decentralized ledger.

- Bitcoin ransom (2019) using the blockchain-based social network Steemit
 - group attempting to get paid to release damaging papers on 9/11 attacks
 - payment mileposts in BTC determine which documents are released
 - banned from mainstream social media platforms
 - messaging via Steemit to prevent censorship. Banned by Steemit, but track remains on Steem blockchain

EDITOR'S PICKS JANUARY 04, 2019 19:28 EST

Bitcoin Ransom: Hacker Group Releases Layer 1 Of "Damaging" 9/11 Papers

Twitter has suspended their account. They moved to Steemit, a blockchain-based censorship-resistant social media platform. Since their initial announcement, they have received more than 3 bitcoins from the public. The first "level" and a few "checkpoints" are now publicly available.



How does Steemit work?

Steemit.com is one of the many websites (including [Busy.org](#), [DTube](#), and [Utopian.io](#)) that are powered by the Steem blockchain and STEEM cryptocurrency. All of these websites read and write content to the Steem blockchain, which stores the content in an immutable blockchain ledger, and rewards users for their contributions with digital tokens called STEEM.

CONCLUSIONS

- advantages of P2P:
 - to exploit computational resources 'in excess' (idle CPU cycles unused storage space, unused band width,...) to have in return resources/services/social networks participation,..
 - to use resources to guarantee trust in a trustless environment
- advantages for the community
 - self scaling property: the participation of a larger number of users/host naturally increases the system resources and its capacity to serve a larger number of requests
- advantages for the seller: decrease of the cost to set up a new application
 - client/server: requires the use of a server farm characterized by high connectivity so that the request of millions of users can be satisfied
 - fault tolerance: the server farm must be replicated in different locations
 - the server must be managed so to offer a service 24*7

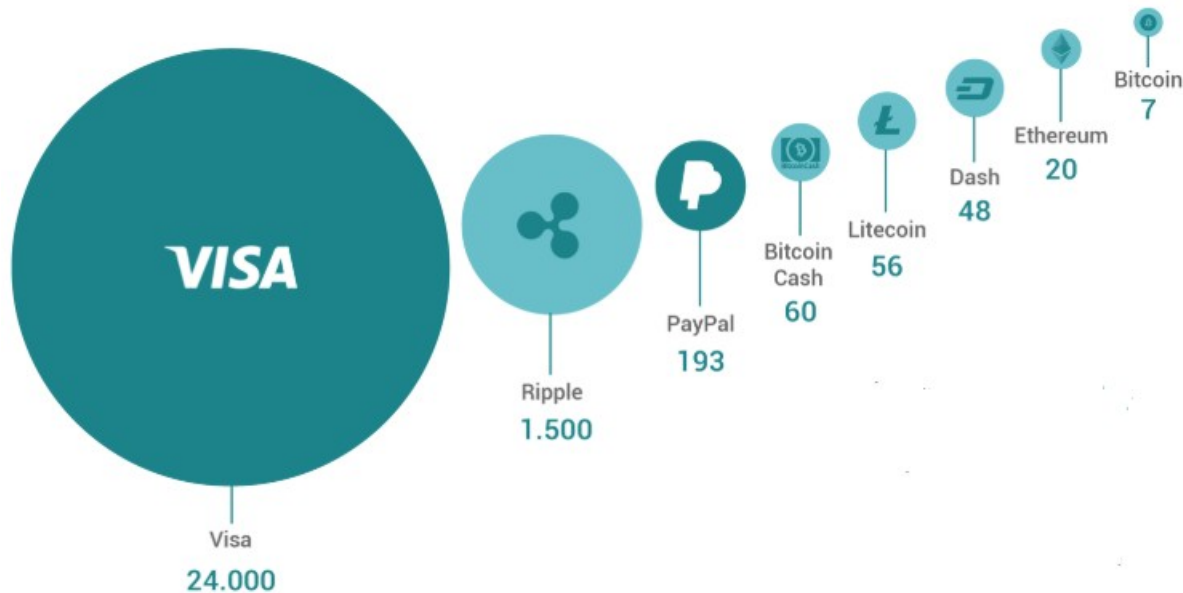
CONCLUSION: THE FUTURE

- The success of a P2P application greatly depends from the creation of a “critical mass” of users: a level of participation to the P2P network which enables the application to self sustain
- in the first P2P systems the novelty of the application has been fundamental for reaching a critical mass
 - file sharing: free content, wide content selection
 - cryptocurrencies as a profitable asset
 - tokens as a simple way to exchange assets
- the success of new P2P applications will greatly depend besides a good engineering of the application
 - from the new application appealing
 - from the definition of new 'business model'

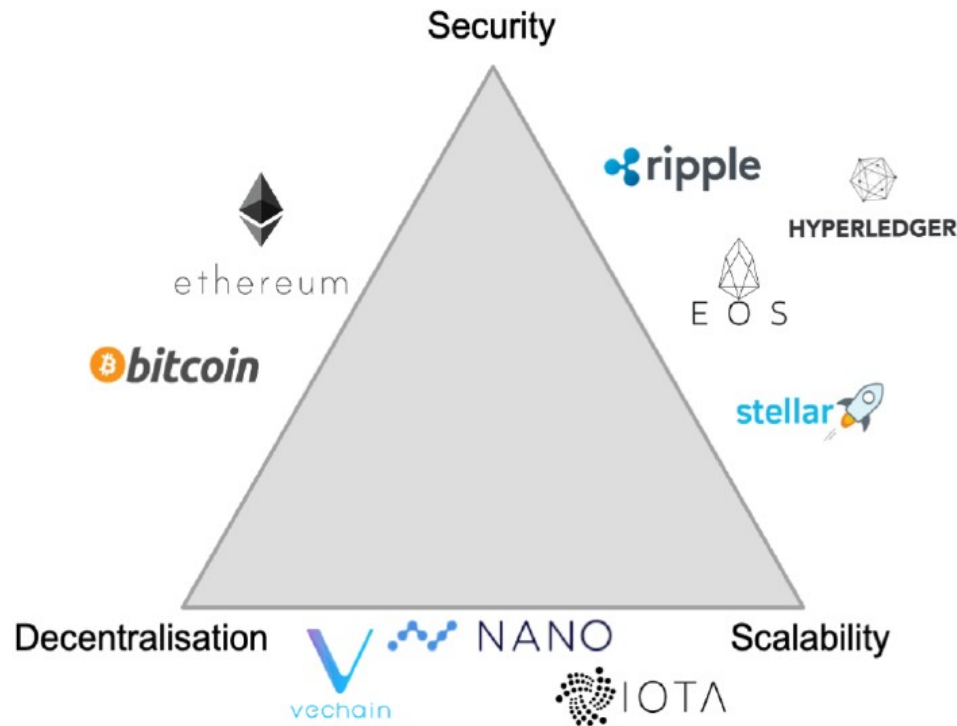
CONCLUSION: SCIENTIFIC CHALLENGES

- the development of P2P applications is a challenge and requires the solution of several novel problems
- the classic methodologies for the development of distributed systems of the “old generation” cannot be exploited:
 - the order of magnitude of a P2P system is different with respect to that of classical systems (million of nodes with respect to hundreds of nodes)
 - classical algorithms/techniques classiche “do not scale” on network of this size
 - the failure/leave of one of the nodes is a normal event
- Systems with dimension and with this dynamicity level require new tools
 - strategies for the peer cooperation/Nash equilibrium: game theory
 - novel cryptographic techniques
 - novel consensus algorithms
 - complex system analysis tools

CONCLUSION: SCIENTIFIC CHALLENGES

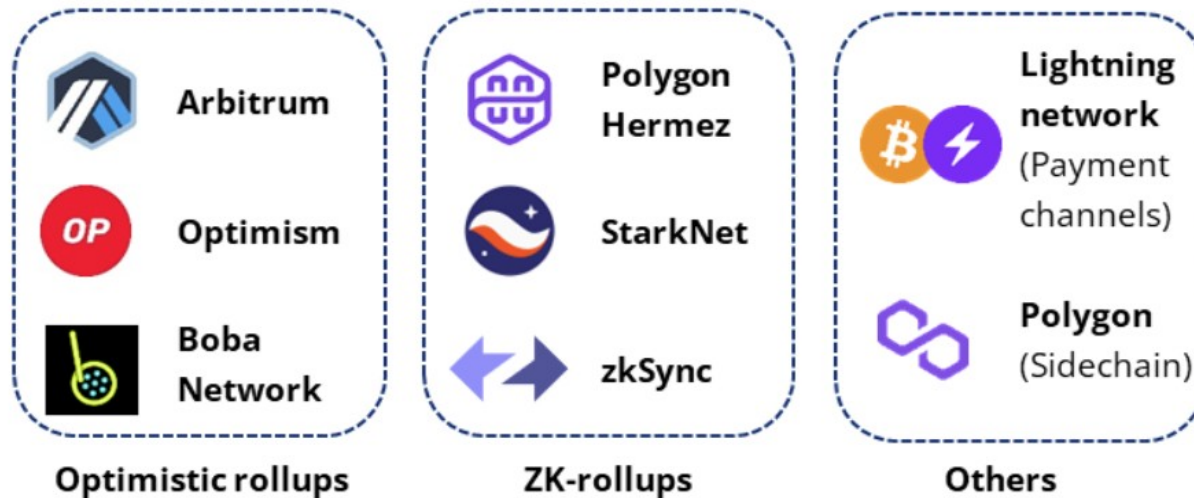


THE BLOCKCHAIN TRILEMMA



- a big scientific challenge:
- how can we improve scalability without reducing the security level and maintaining a high level of decentralization?

THE BLOCKCHAIN TRILEMMA: SOLUTIONS?



general idea:

move heavy computation/data outside the blockchain

use the blockchain as a “trust anchor”