U LISBOA | UNIVERSIDADE DE LISBOA

Blockchain
FOR
DUMMIES

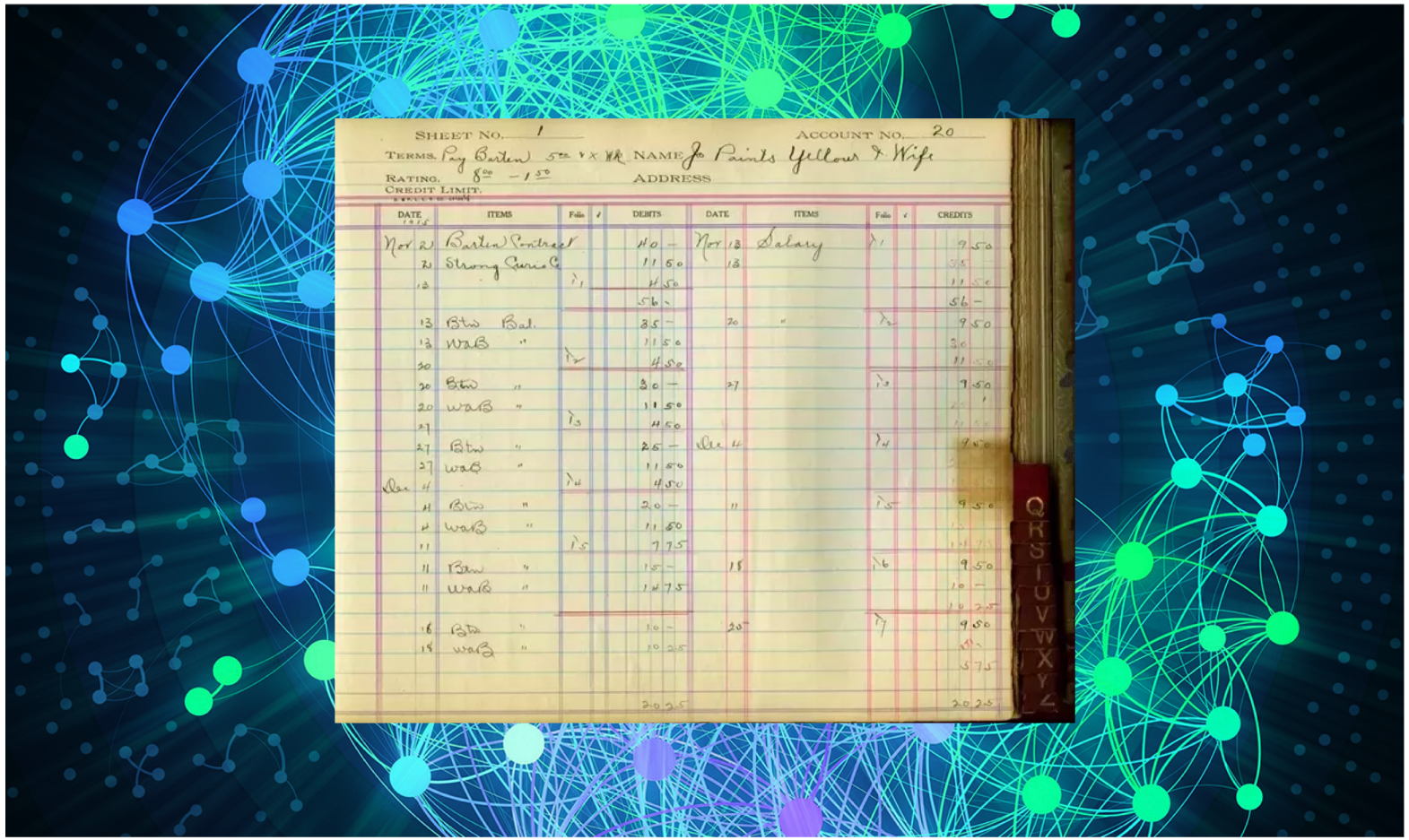
Alysson Bessani

FCiências ULisboa

This talk…

- … is about
  - What is the blockchain
  - How it works
  - How it can be used
- … it's not about
  - Cryptocurrencies
  - How to make money with them ☺

What is a Blockchain?



General ledger on top of a peer-to-peer network

Blockchain

ORIGINS


WHO IS
SATOSHI NAKAMOTO?

The paper/system that started the revolution (2009)

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 hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as

From Satoshi's paper

- 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*.
- *A* ***trusted third party is not 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 hash-based proof-of-work, forming ***a record that cannot be changed*** *without redoing the proof-of-work*.

From Satoshi's paper

(cont.)

- The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as *a* ***majority of CPU power*** *is controlled by nodes that are not cooperating to attack the network*, they'll generate the longest chain and outpace attackers.
- The network itself requires minimal structure. Messages are broadcast on a best-effort basis, and *nodes can leave and rejoin the network at will, accepting the* ***longest proof-of-work chain*** *as proof of what happened while they were gone*.

The Big ideas

- From Satoshi's paper:
  - Peer-to-peer electronic transactions and interactions
  - Without financial institutions
  - Cryptographic proof instead of central trust
  - Put trust in the network instead of in a central institution

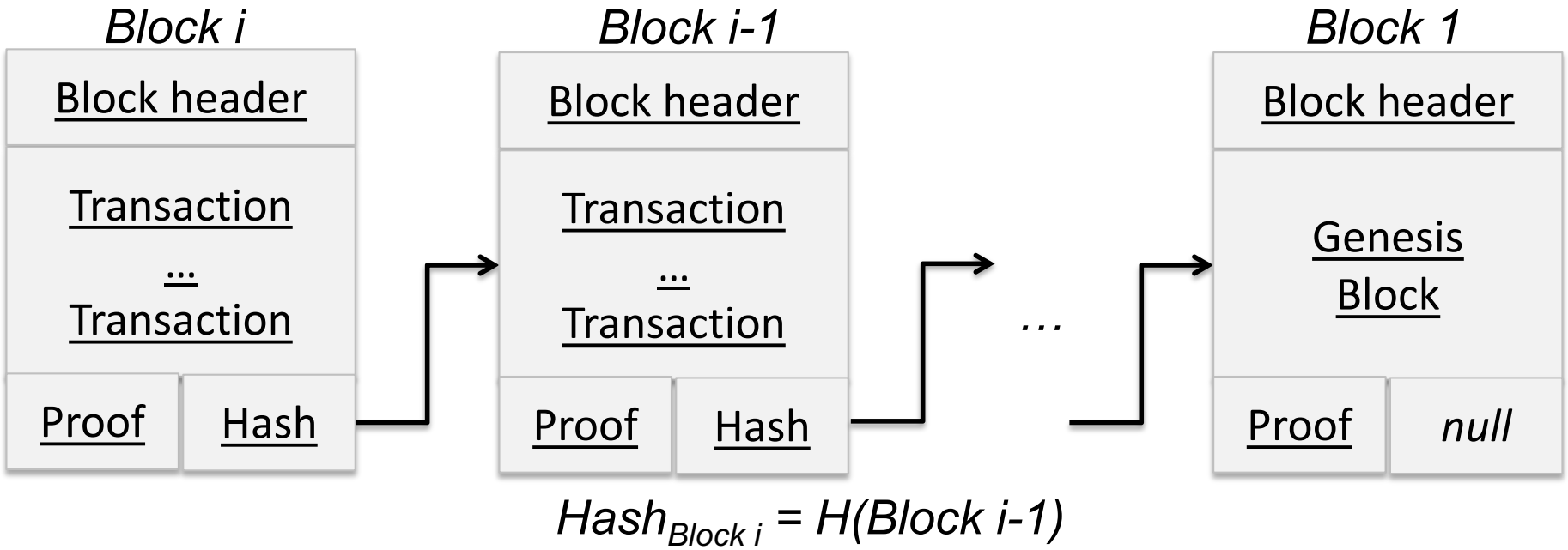
- Three definitions of blockchain:
  - TECHNICAL: Open database that maintains a distributed ledger
  - BUSINESS: Exchange network for moving values between peers
  - LEGAL: A transaction validation mechanism, not requiring intermediate assistance

Blockchain

TECHNICAL OVERVIEW

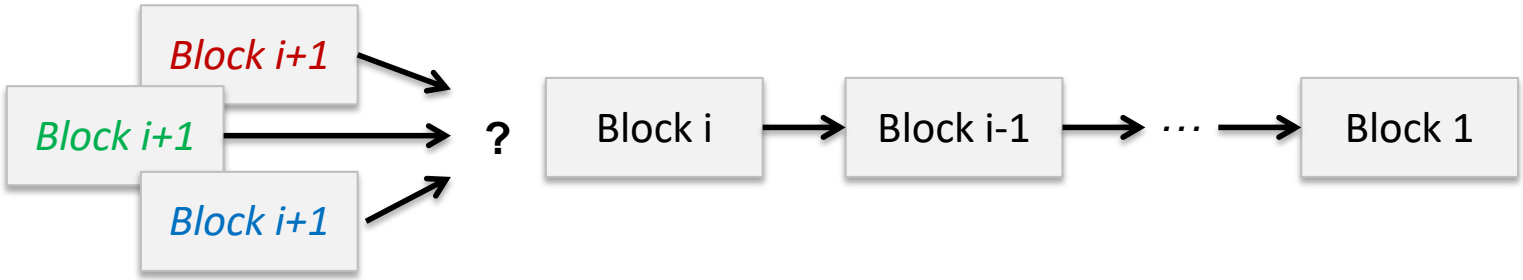
Blockchain

- A blockchain is an open database that maintains a distributed ledger on a peer-to-peer network
- Comprised by a continuously growing list of records called *blocks* that contain *transactions*
- Blocks are protected from tampering by *cryptographic hashes* and a *consensus algorithm*



What is a Blockchain?

- Is it just a secure log?
  - Yes, but now it is called **Secure Ledger**
- What is new?
  - It is distributed, maintained in a P2P way, a **Distributed Secure Ledger**
- It requires solving **distributed consensus** under **Byzantine faults**
  - Given a blockchain of *i* blocks, and several proposals for block *i+1*, how to decide (in a distributed way) which proposal to adopt?



  - **Byzantine faults:** a faulty process can do anything. It can an model hardware defects, software bugs and even **intrusions.**

Two Models

- **Public (Open) Ledgers**
  - Like the ones used in **Bitcoin** and **Ethereum**
  - The parties do not have strong identities
- **Permissioned (Private) Ledgers**
  - Also called **Distributed Ledger Technology (DLT)**
  - Being used in many new “business cases”
  - The parties have verifiable ids and require permission to participate

Public (Open) Ledgers

- Usually implemented through Nakamoto Consensus or its variations

- Key ideas:
  - Anyone can participate in the network
  - A block can be added to the *blockchain* only if a **cryptographic puzzle** is solved
  - New blocks are disseminated in a peer-to-peer network
  - If multiple proposals for extending the chain are received, the longest proposal is used

Nakamoto Consensus

(code on every peer)

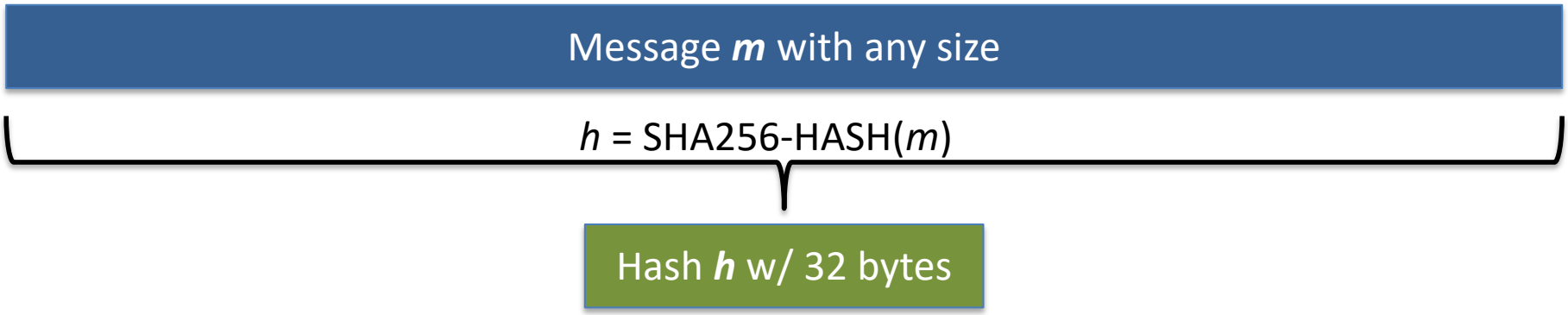
- Local state:
  - *C*: local copy of the blockchain
- Algorithm:
  - When a new chain *C'* is received
    - *C* = ***maxvalid****(C,C')*
  - When a new batch of transaction *txs* is received
    - *C* = ***proof-of-work****(C,txs)*
    - *Broadcast(C)*
  - When a read request is received
    - Return the transactions on *C*

Compares two chains and **chooses the longest one** that is valid, i.e., each block is correctly signed, contains the hash of the previous and solved the proof-of-work puzzle

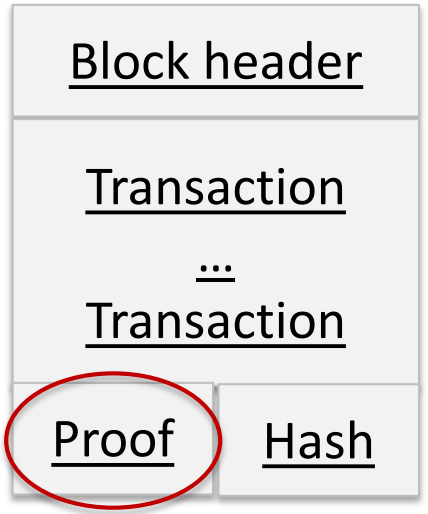
Solves the following **cryptopuzzle**: find a valid block containing the transactions and the hash of the previous block such that the hash of the this block is smaller than D (a difficulty parameter)

Proof of Work

Cryptographic hash function (e.g., SHA256)

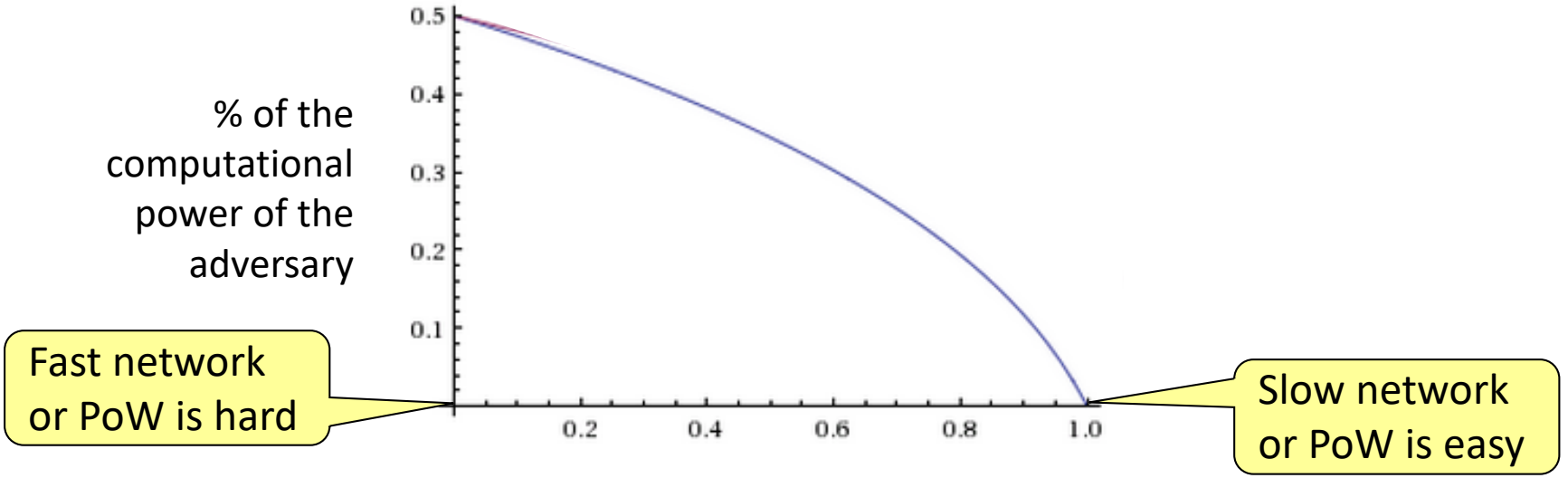


- The Proof-of-Work is generated by changing the block until you find a hash starting with a given number of zeros (difficulty)
- Miners have to try a lot, for example, in Bitcoin it takes 10 min. on average



Why and how is it secure?

- The protocol works **if the adversary controls less than half of the total computing power** and the network disseminate data “fast enough”



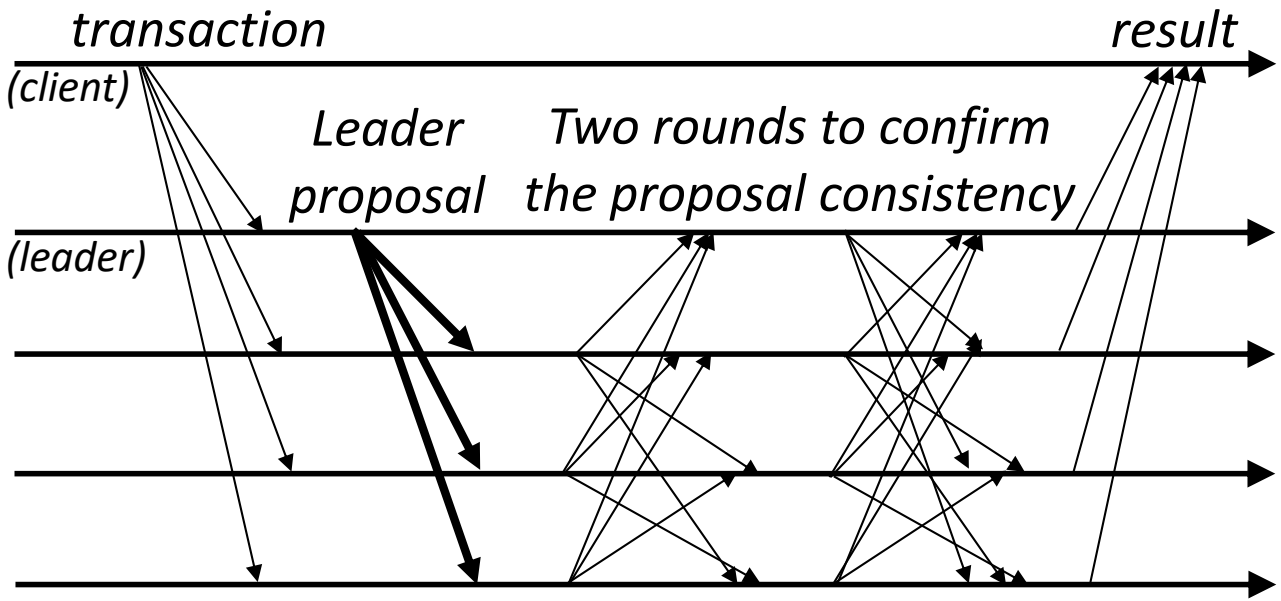
Permissioned Ledgers

- This is the blockchain architecture most companies are interested these days as it removes intermediaries and is not related with dark web or illegal activities

- Key idea:
  - To participate, a party needs to be **accepted by others** (or by the owner of the ledger) and to have an **authorized ID**
  - Employ traditional *Byzantine consensus* protocols:
    - The **adversarial threshold** is not computational, but structural, e.g., the number of faulty nodes is less than 1/3

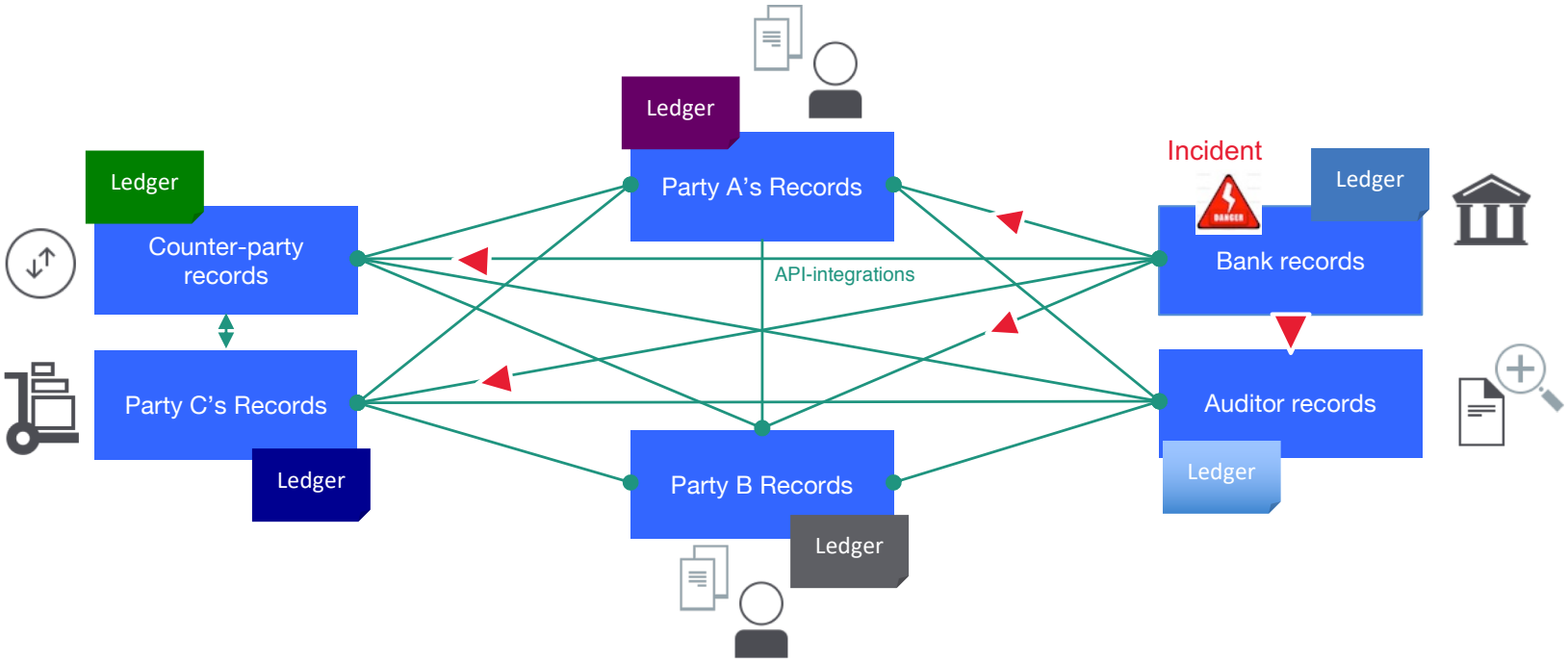
Permissioned Consensus Protocol

(e.g., PBFT, BFT-SMaRt)



BFT-SMaRt: http://bft-smart.github.io/library/

How to monitor asset ownership and transfers in a business network?

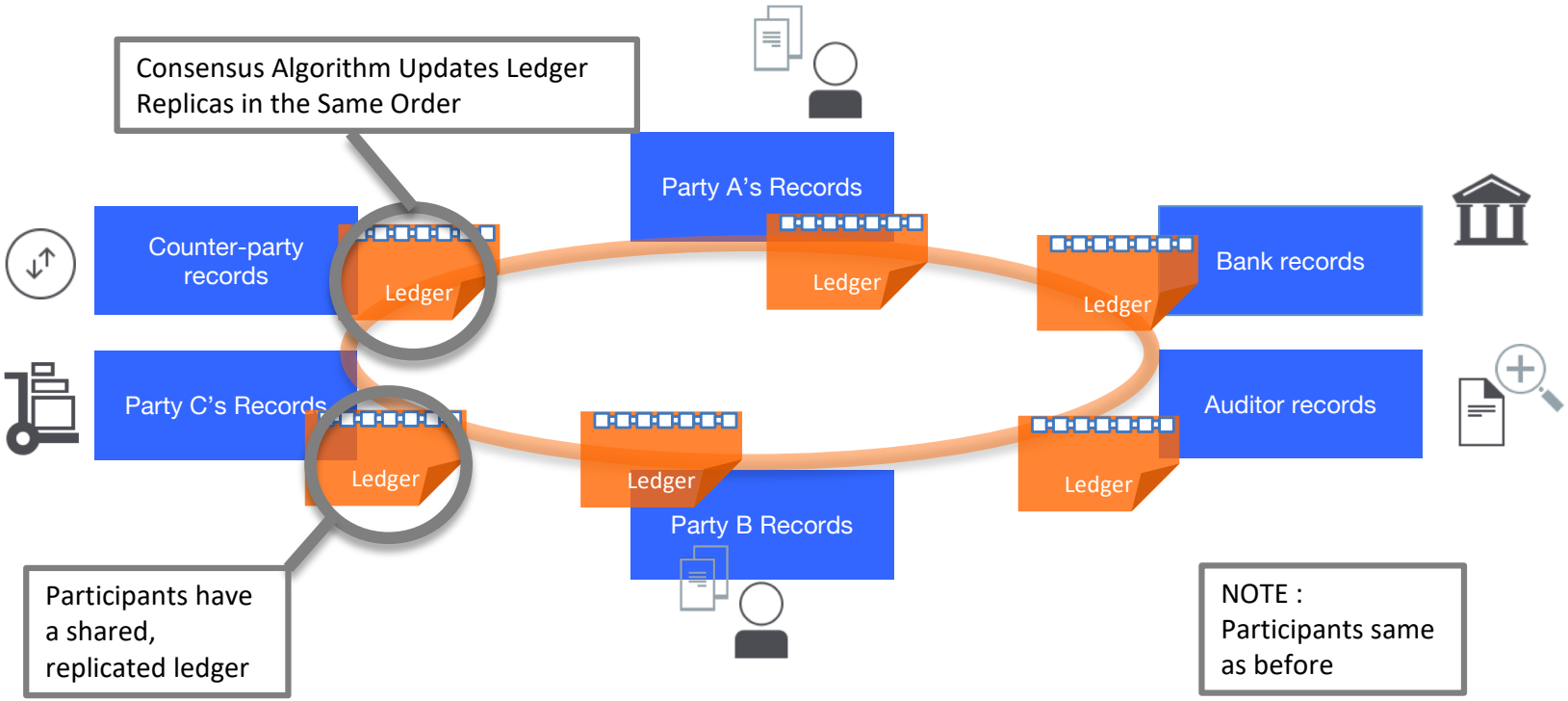


Inefficient, expensive, vulnerable

A Centralized Trusted Solution (trusted third party)?

Issues with Reliability, Scalability, Trust, Censorship, Cost, ...

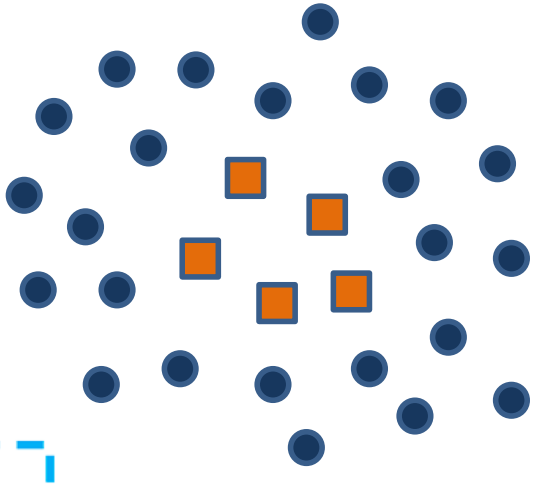
Solution: Blockchain, a permissioned, replicated, shared ledger

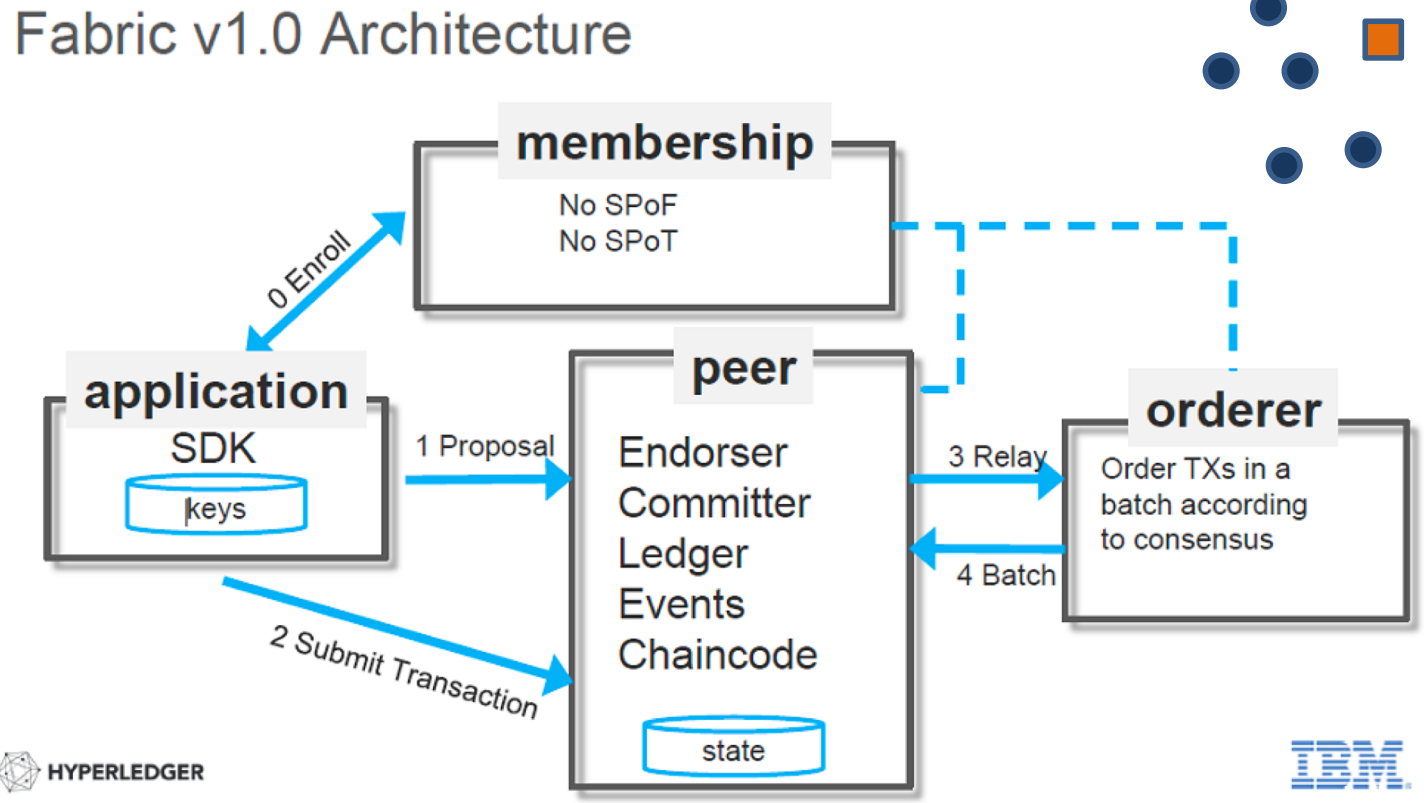


Consensus on transaction order and validity, provenance, immutability, finality

HYPERLEDGER Fabric

Modular, permissioned blockchain
(a Linux foundation project)





Blockchain

BUSINESS AND SOCIETAL ASPECTS

Mostly taken from W. Mougayar's book "*The Business of Blockchain*" (Willey 2016)

The 5th Disruptive Computing Paradigm

- Blockchain as the 5th disruptive computing paradigm

- The blockchain (finally) implements the (missing) **trusted layer of the internet** and enable secure transactions
- Three levels of blockchain adoption:
  - Blockchain 1.0 – Currency
  - Blockchain 2.0 – Smart contracts
  - Blockchain 3.0 – Applications beyond finance

The Many Faces of Blockchain

1. Cryptocurrency
   - Enables the execution of monetary transactions
   - Provide incentives for participation in the network

2. Computing Infrastructure
   - After ordering transactions in the blockchain, peers execute them
   - It implements a kind of *"replicated state machine"*, and thus it can be the ultimate unstoppable computer

3. Transaction Platform
   - A blockchain is a transaction processing platform, as it can validate, record, and execute value-related transactions
   - How does it compare with current financial trans. platforms?

Non-blockchain:	VisaNet: 2000 trans/s	Paypal: 155 trans/s
Permissionless:	Bitcoin: 5 trans/s	Ethereum: 100-1000 trans/s
Permissioned:	BFT-SMaRt: 2000-70000 trans/s	

The Many Faces of Blockchain

4. Decentralized Database
   - Blockchains are like databases, but with parts of the information being public
   - Blockchains are not as efficient as databases
5. Distributed Accounting Ledger
   - A blockchain is a distributed time-stamped ledger that records every transaction processed in the network
   - The ledger can be public, private, or semi-private
6. Development Platform
   - The blockchain defines a new ecosystem for developing new types of applications, e.g., smart contracts, and apps interacting with ledgers
7. Open Source Software

The Many Faces of Blockchain

8. Financial Services Marketplace
   - Having a cryptocurrency in its core makes the blockchain an innovation environment for financial services
9. Peer-to-Peer Network
   - There is nothing centralized about blockchains
   - Architecturally, the base layer of a blockchain is a P2P network
   - There are no intermediates to filter, delay, or block any transaction between users
10. Trust Services Layer
   - It is almost impossible to control or put down a blockchain
   - This creates a new environment for creating new types of services that require trust on its core

Blockchain Applications

- Blockchains enable ATOMIC programmability

  Assets, Trust, Ownership, Money, Identity, Contracts

- More specifically, blockchains enable:
  - Creation and real-time movement of digital assets
  - Embedding trust rules inside transactions and interactions
  - Time-stamping, rights, and ownership proofs
  - Self-execution of business logic with self-enforcement
  - Selective transparency and privacy
  - Resistance to single points of failure and censorship

Questions?