

APPLICATIONS OF BLOCKCHAIN IN THE PPDR SECTOR

dr. Urban Sedlar
Faculty of Electrical Engineering
University of Ljubljana, Slovenia

Things to discuss

- What blockchain is
- How it works
- When to use it
- What kind of problems it can solve in PPDR

HOW BLOCKCHAIN WORKS

The birth of blockchain



- Bitcoin – the first (working) example of a decentralized electronic currency
 - Uses cryptography and peer-to-peer communication
- Author: Satoshi Nakamoto
 - (Identity still unknown)
 - Paper published in 2008
- *Blockchain* is the underlying technology for storing transactions

Bitcoin: A Peer-to-Peer Electronic Cash System

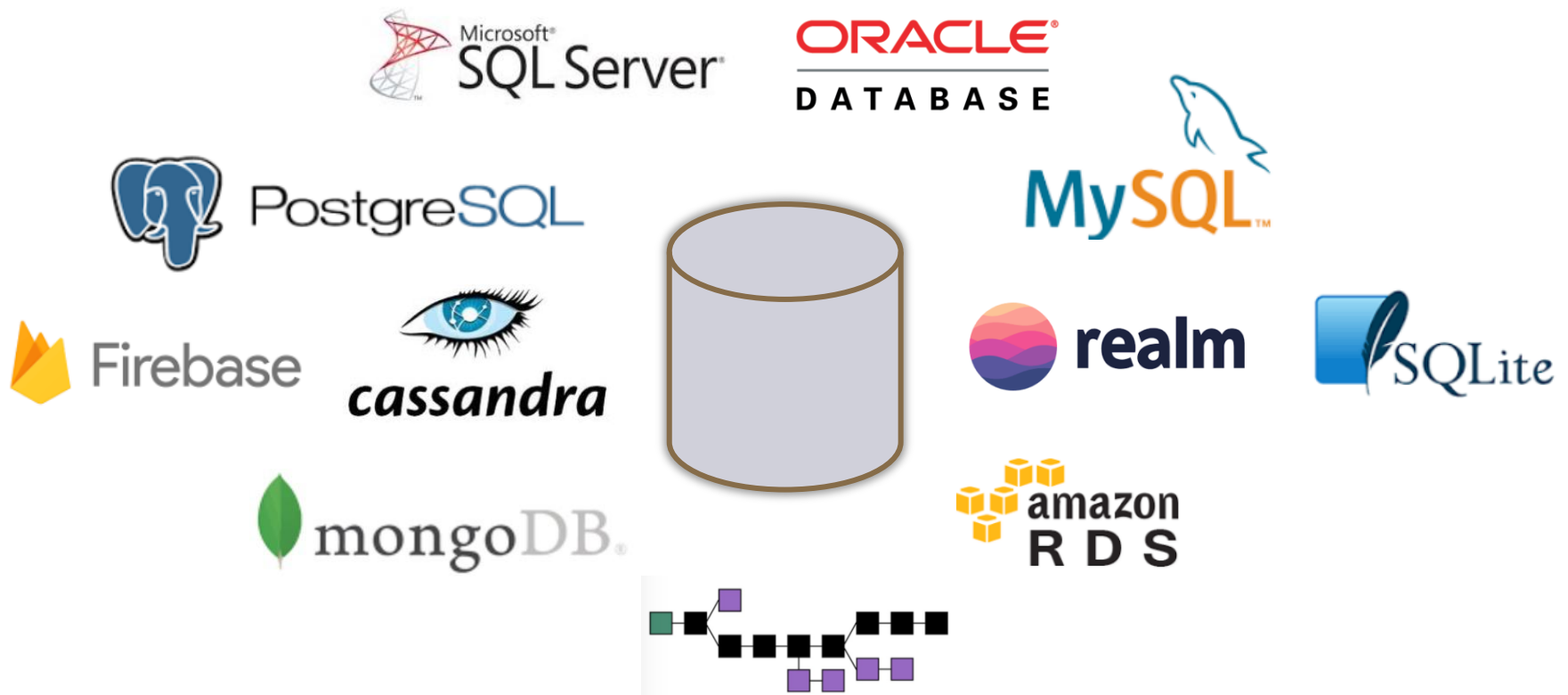
Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

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.

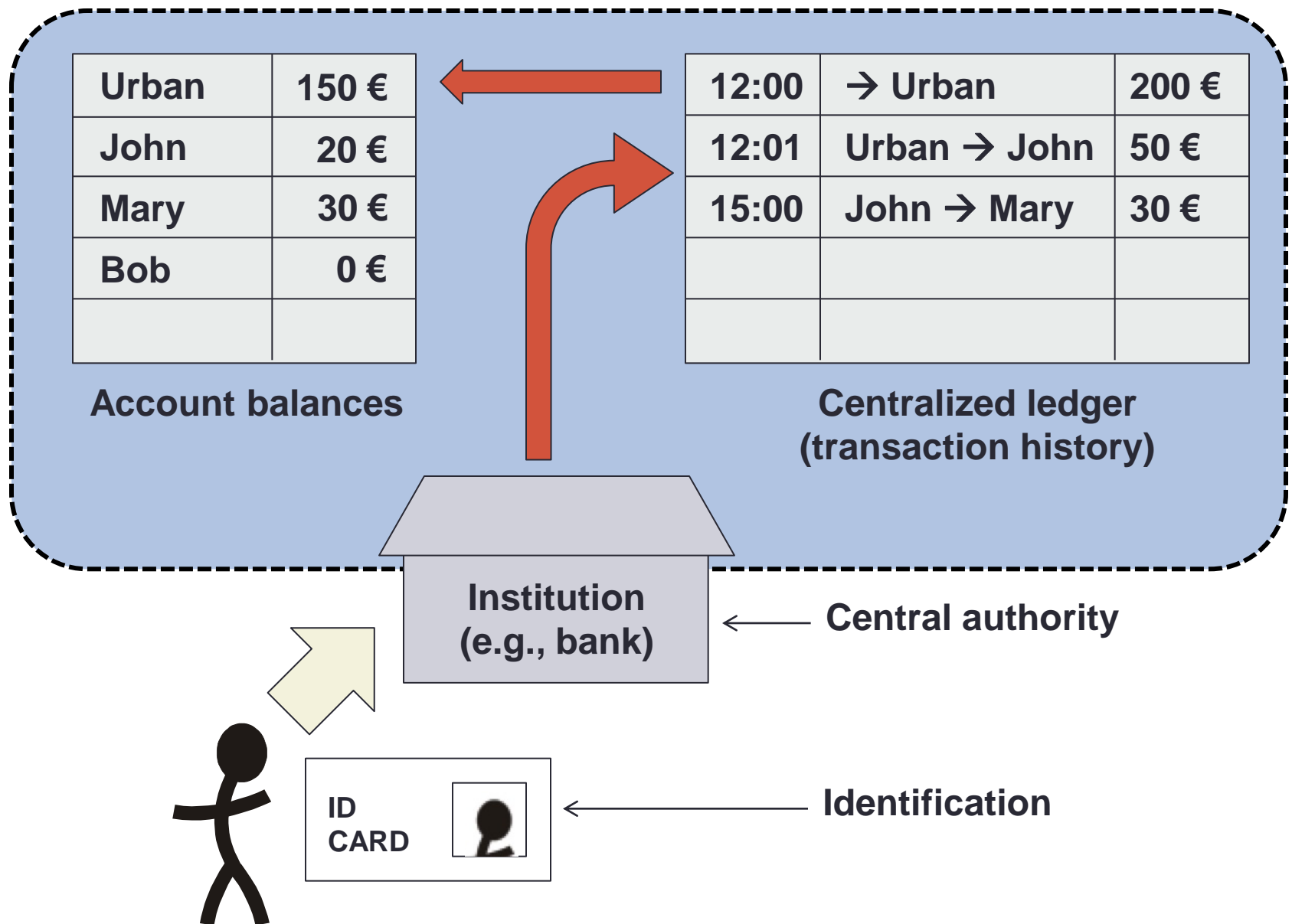
<https://bitcoin.org/bitcoin.pdf>

Another way to look at it

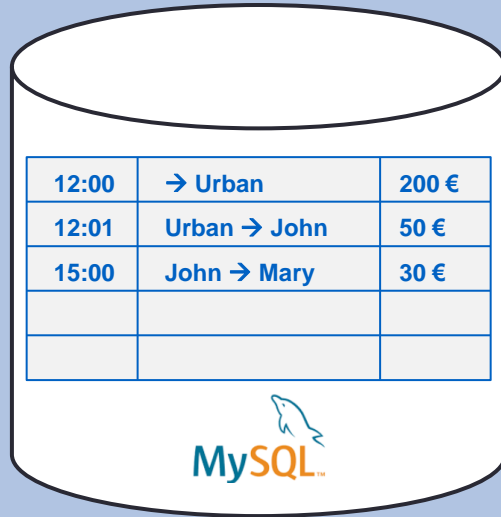
- Blockchain is a database
 - It solves similar problems as the technologies below
 - Database is a useful building block for many projects



A simple database problem: money



Relational database (e.g., MySQL)



Centralized ledger (transaction history)

- + Guaranteed consensus
- + Atomic transactions
- + High throughputs
- + Extremely cheap
- + Simple to maintain
- No redundancy → single point of failure

Institution
(e.g., bank)

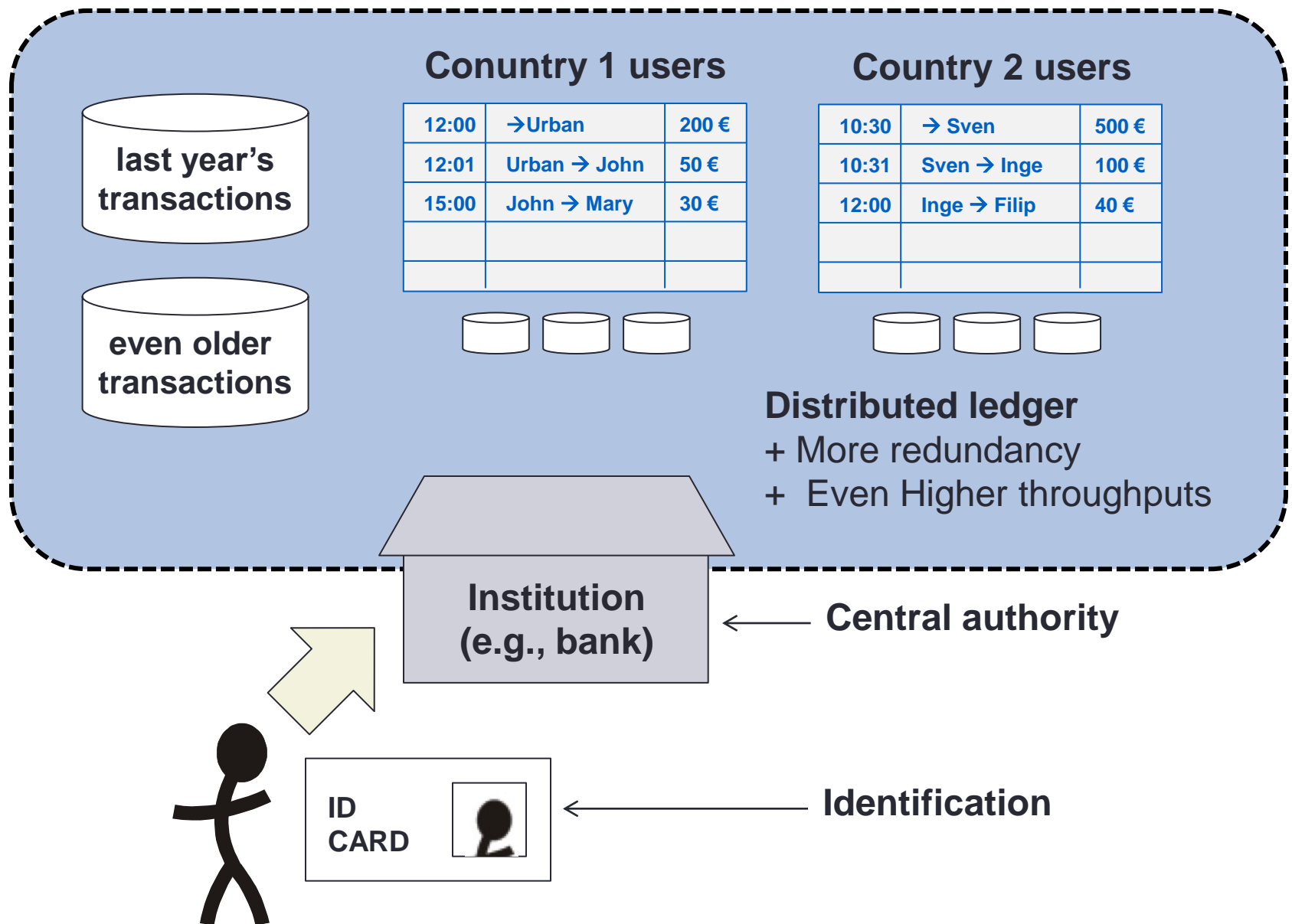
← Central authority

ID
CARD

← Identification



Distributed database

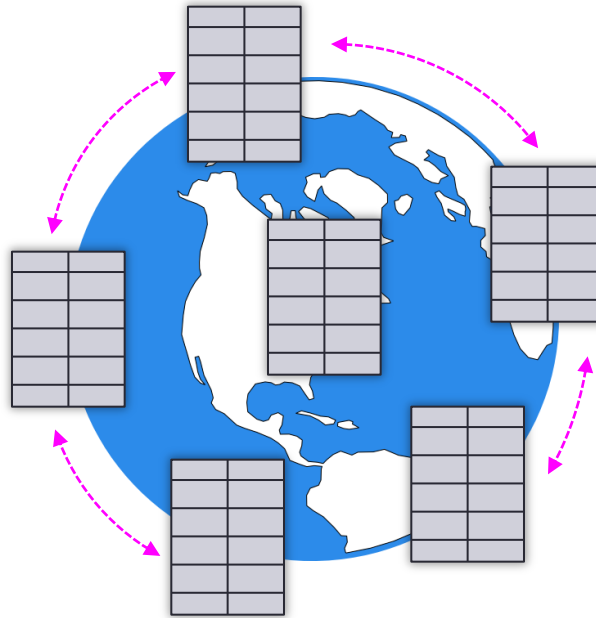


Centralization problems

- In both cases, the system is *organizationally* centralized
 - The organization can forbid you to make the transaction
 - Or could require 3 working days for completion
 - Or could enforce working hours
 - Or could even declare bankruptcy
- We still have a single point of failure
- To get around this, blockchain makes the database public
 - This can be truly public (like Bitcoin, Ethereum, and other currencies)
 - a.k.a. public ledger
 - Or only public to the members of the system (like Hyperledger Fabric)
 - a.k.a. permissioned ledger

Public ledger

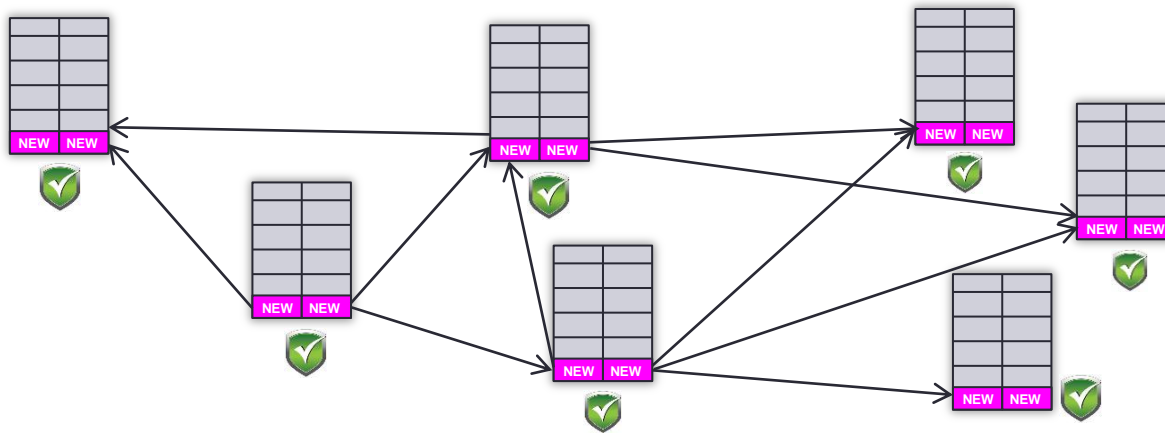
- Every *full* node stores the entire history of the system (all transactions ever made)




- Nodes also synchronize all the changes
 - This is done in a peer-to-peer manner (like BitTorrent)
- The entire history can be quite large
 - Example: Bitcoin (May 2018): ~168 GB and growing

How transactions are made

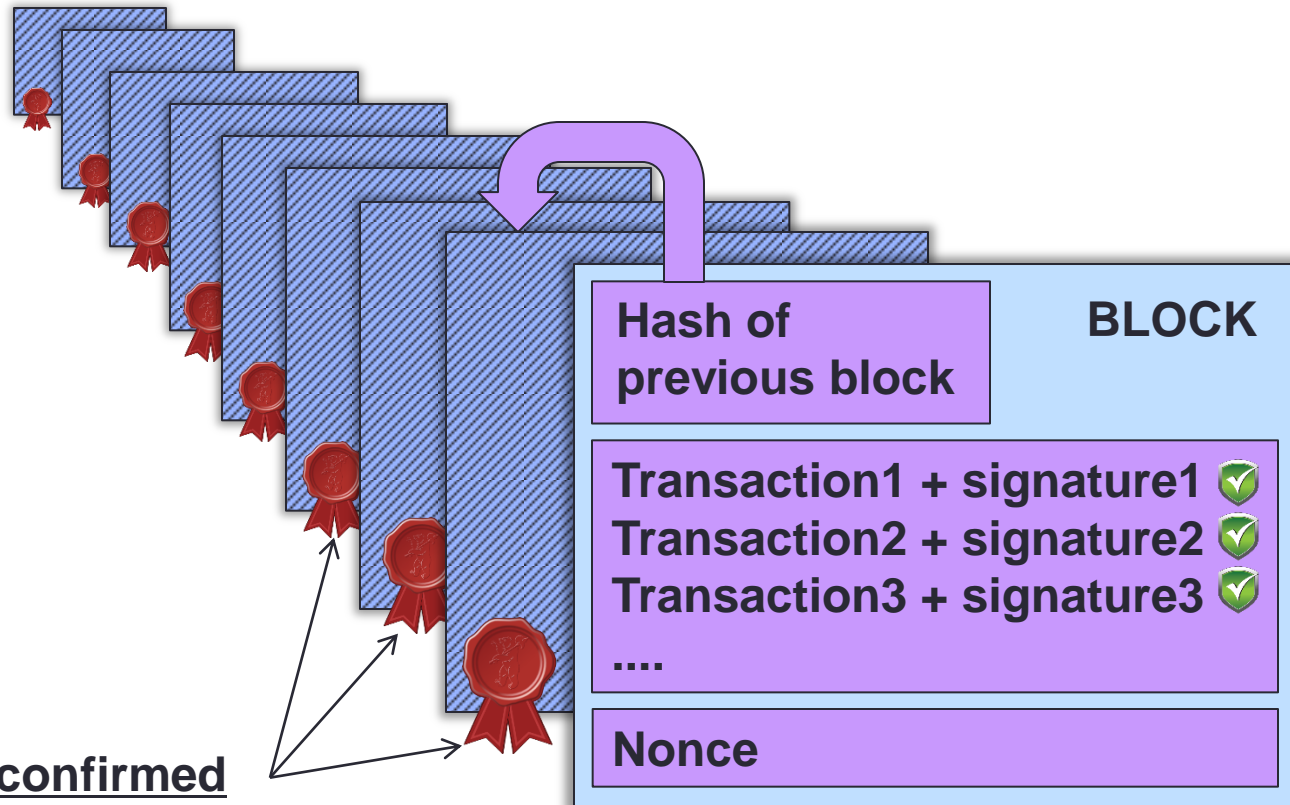
- Transactions are authorized with digital signatures
- And are shared among all of the peers



- Each peer acts as its own “auditor”, and checks all transactions
 - Validity of signature, availability of funds 
- But before confirmation, this is just an intention

Transactions are grouped into blocks

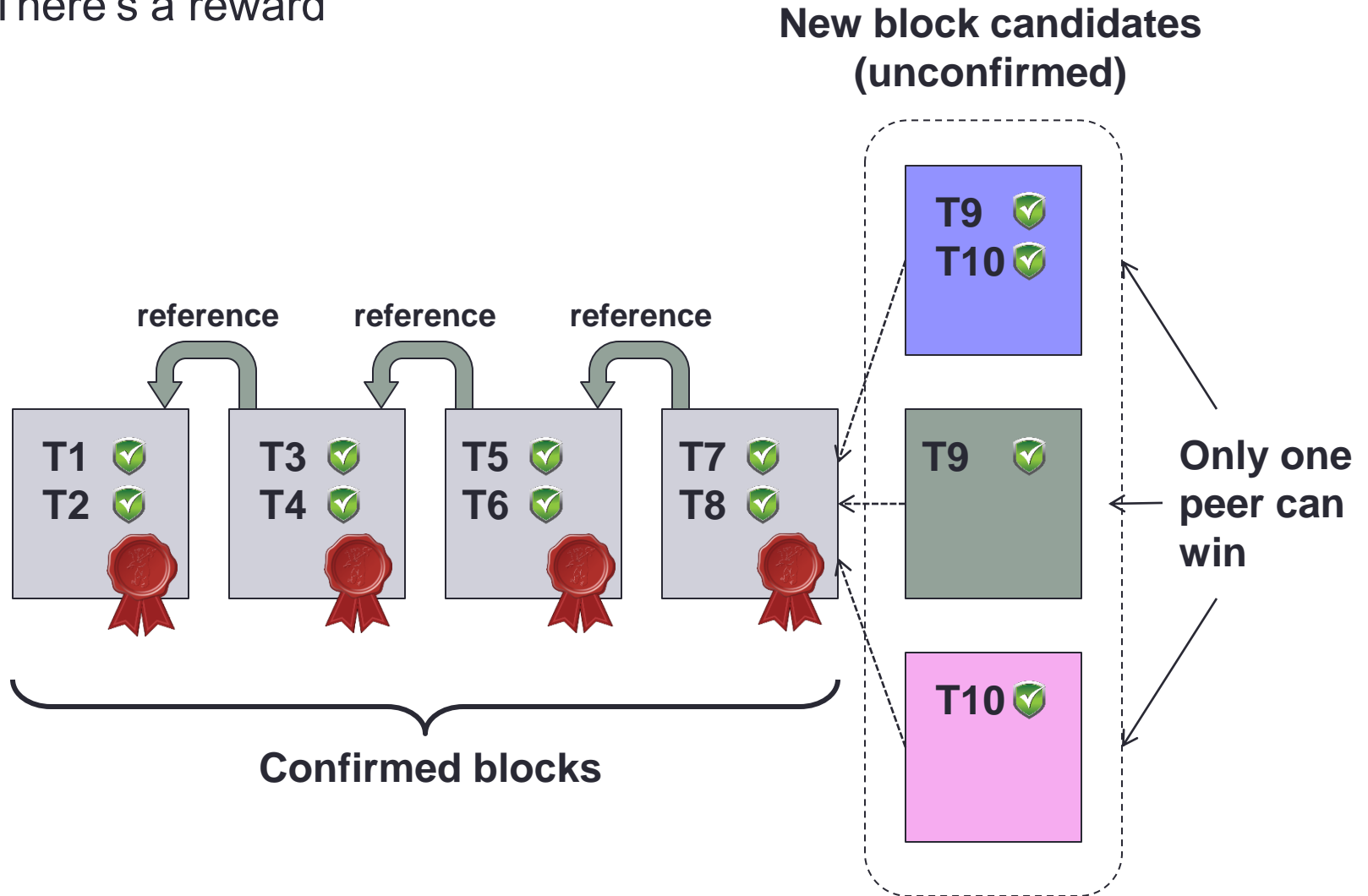
- Transaction history forms a chain of blocks
 - Blocks are periodically confirmed (this is expensive).
- The blockchain:



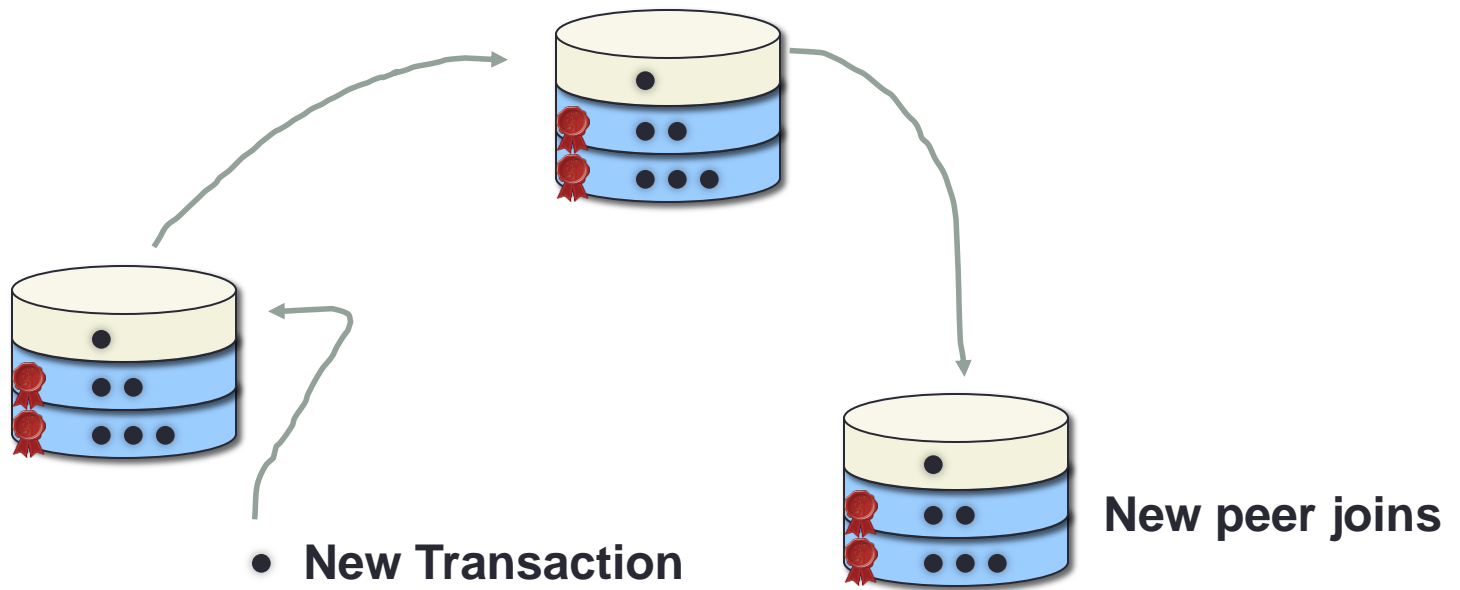
Each block must be confirmed to become a valid part of history

Peers have a competition

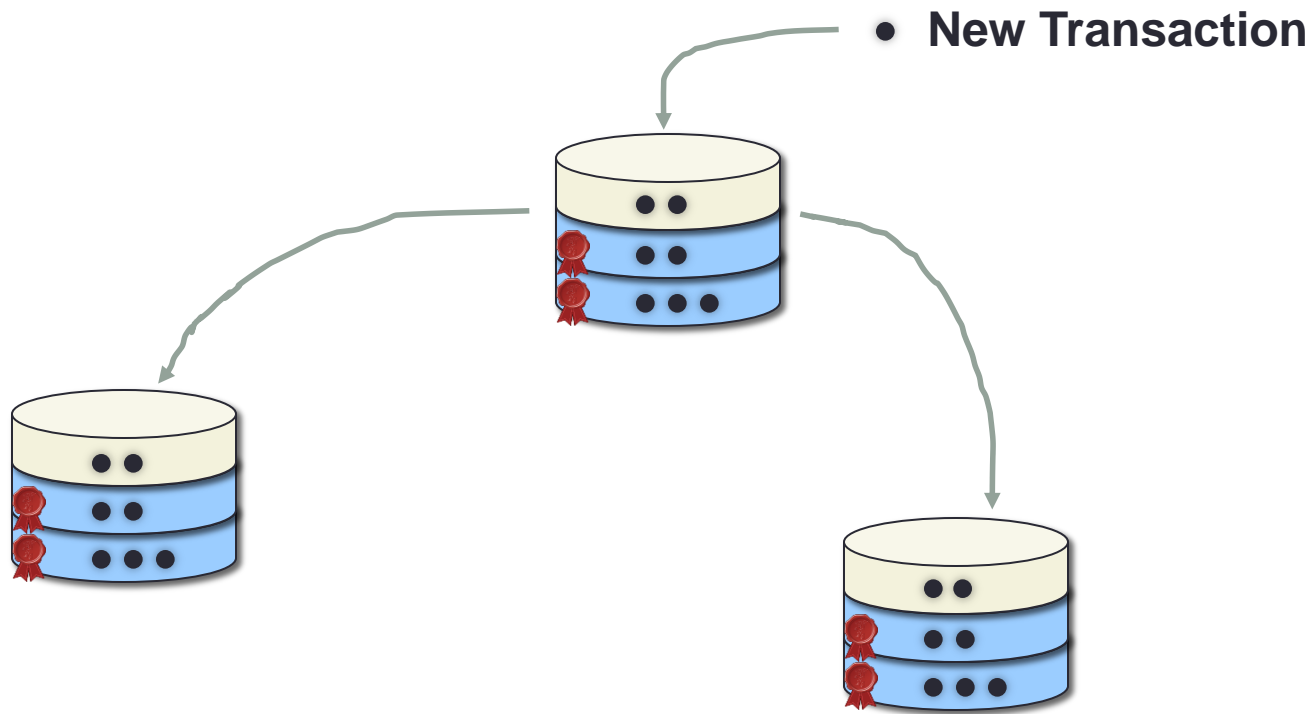
- Peers compete to confirm the next block
 - There's a reward



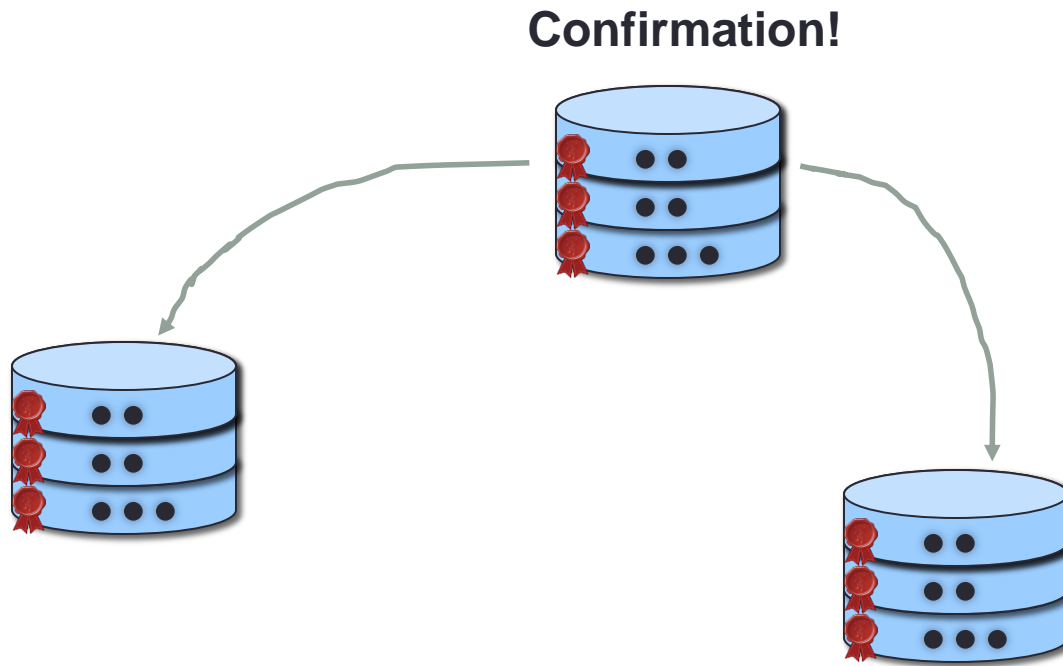
How it all fits together



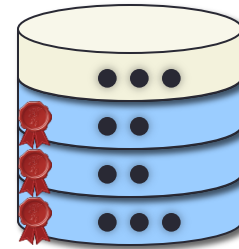
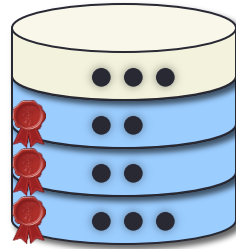
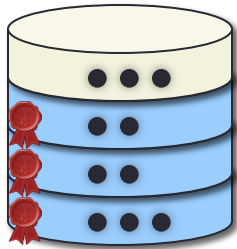
How it all fits together



How it all fits together

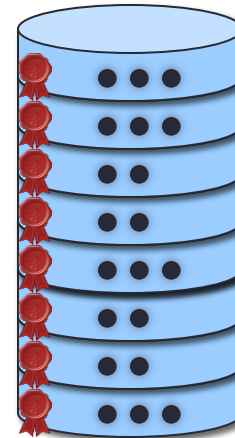
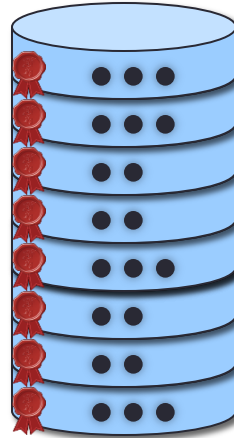
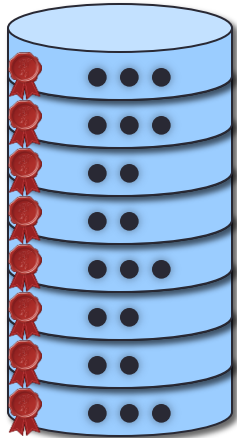


How it all fits together



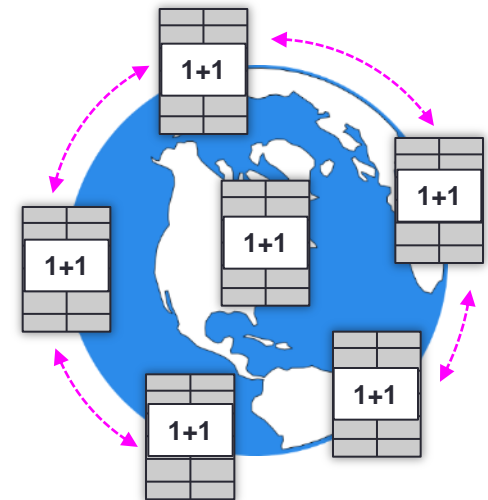
How it all fits together

- The database can only grow
 - History is unchangeable!



Beyond currencies

- We can publish *computer programs* in the shared database
 - This is called a Smart contract
- Every node will execute the instructions
 - And save the results, for everyone to see
- With this, a blockchain has become a computer
 - Extremely slow and inefficient
- But it has some advantages
 - Can't be powered off
 - Can't be reset,
 - destroyed,
 - censored,
 - and anyone can see what it's doing, and what are the results

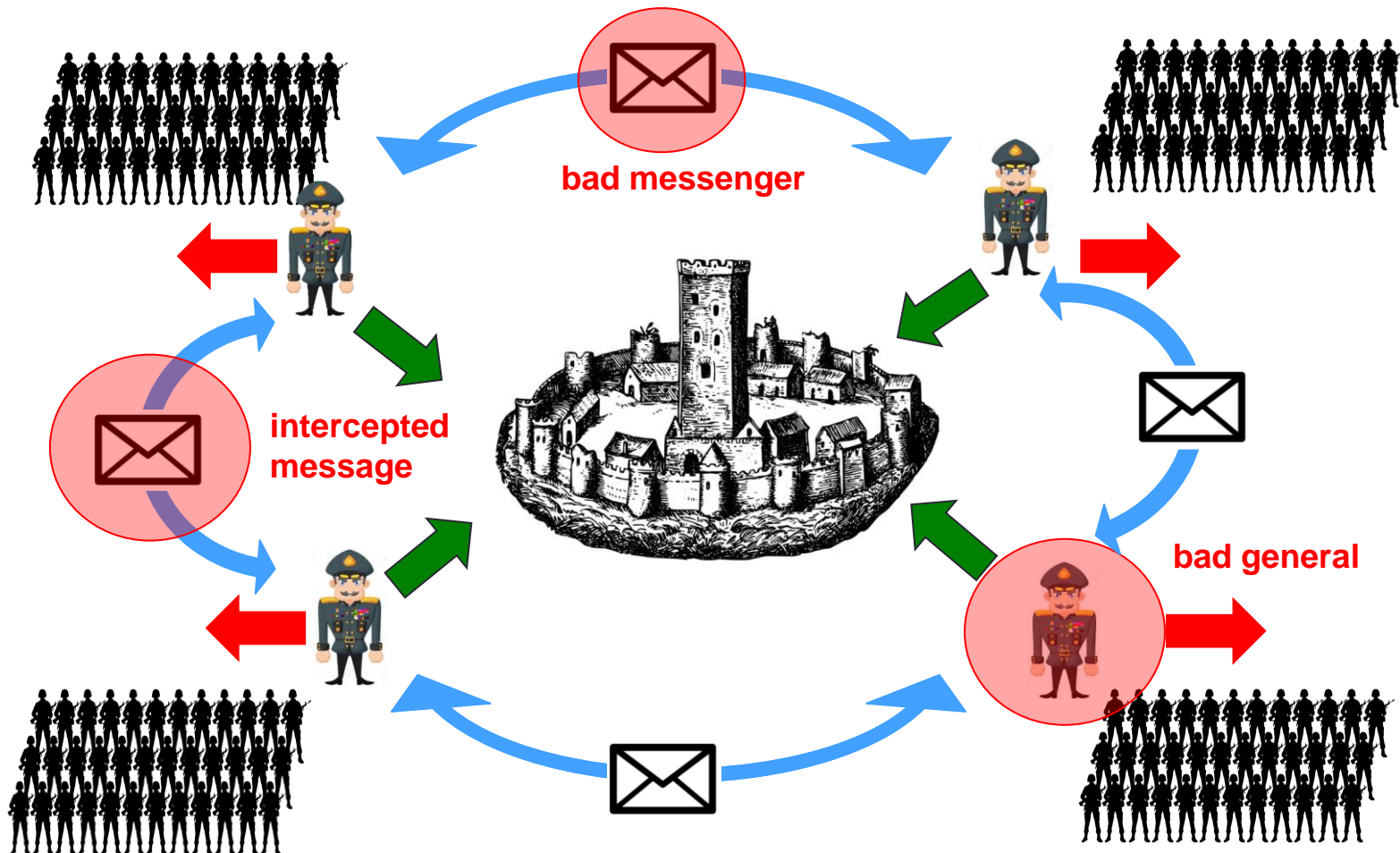


"The World Computer"

WHEN TO USE BLOCKCHAINS

When we need consensus

- Byzantine generals problem
 - Generals are trying to attack a city
 - If all attack or all retreat, there won't be any casualties



Rule of thumb – when to use blockchain

- When you have actors that don't trust each other
 - Or have a conflict of interest
- When you don't want central authority
- When you need to see full audit trail
 - and also have an assurance it is immutable
- When you're afraid someone could
 - censor historical data
 - destroy the database
 - do a denial of service attack on the database

Application domains

- Person-to-person trust / organization-to-organization trust
 - Payments and tokens
 - Health record validation, data access authorization
 - Supply chain management
- Device-to-device trust
 - IoT scenarios & smart appliances
 - Proof of ownership
- Proof-of-existence and proof-of-integrity
 - Certifications, notary service
 - Passports, Birth/death/wedding certificates
- A public (or private) shared database for any kind of data, with single source of truth
 - Viewable and auditable by anyone

Example PPDR use cases

- Timestamping & proof-of-existence
 - Publishing event hashes and timestamps to the blockchain
 - Ensures there's always a clear trail of events
- Managing delegation, ownership and access to resources
 - Linking resources (equipment) to operational units or people
 - Authentication to access network resources / slices, etc.
- Serving as a common database to connect stakeholders from multiple domains → multi-agency collaboration
 - Different EROs (police, fire brigade, medics)
 - Communication infrastructure providers
 - Medical services (doctors, patients and their data)
 - Real estate ownership data
 - Vehicle registry
 - Insurance companies

A more futuristic use case

- Participatory emergency response
 - blockchain identity tied to a person / organization
 - tokens can be traded for services (similarly as “insurance coupons”)
- Examples:
 - a countryman with a truck takes an injured person to the city
 - he earns a token that can be redeemed at the insurance company
 - similarly, different services could be “bought” by different stakeholders
 - an opinion of an independent expert / doctor
 - a local car towing service
 - voluntary firefighter’s help
 - any nearby qualified person’s “service” to bring an AED to the site of emergency
 - All of this is completely verifiable/auditable

GDPR issues

- General Data Protection Regulation
 - Is incompatible with blockchain
- Blockchain is an immutable database
 - Nothing can be deleted from it, not even upon request
 - Data can only be *invalidated* in a later transaction
 - But that doesn't remove old copies
- Don't store sensitive data in cleartext
 - Or you will have problems
 - Make sure you use encryption
 - Or better: store data elsewhere and only use blockchain for proof of existence / proof of integrity

Conclusion

- Blockchain is just a different kind of database
 - Especially useful when stakeholders don't trust each other
 - And when we want an immutable trail
- But today there's a tradeoff
 - Low transaction throughput
 - Highly redundant storage
 - Proof-of-Work schemes burn a lot of energy
- It's in active development
 - Coming soon: sharding, channels, proof-of-stake on many platforms
 - The best applications are yet to come

Thank you!