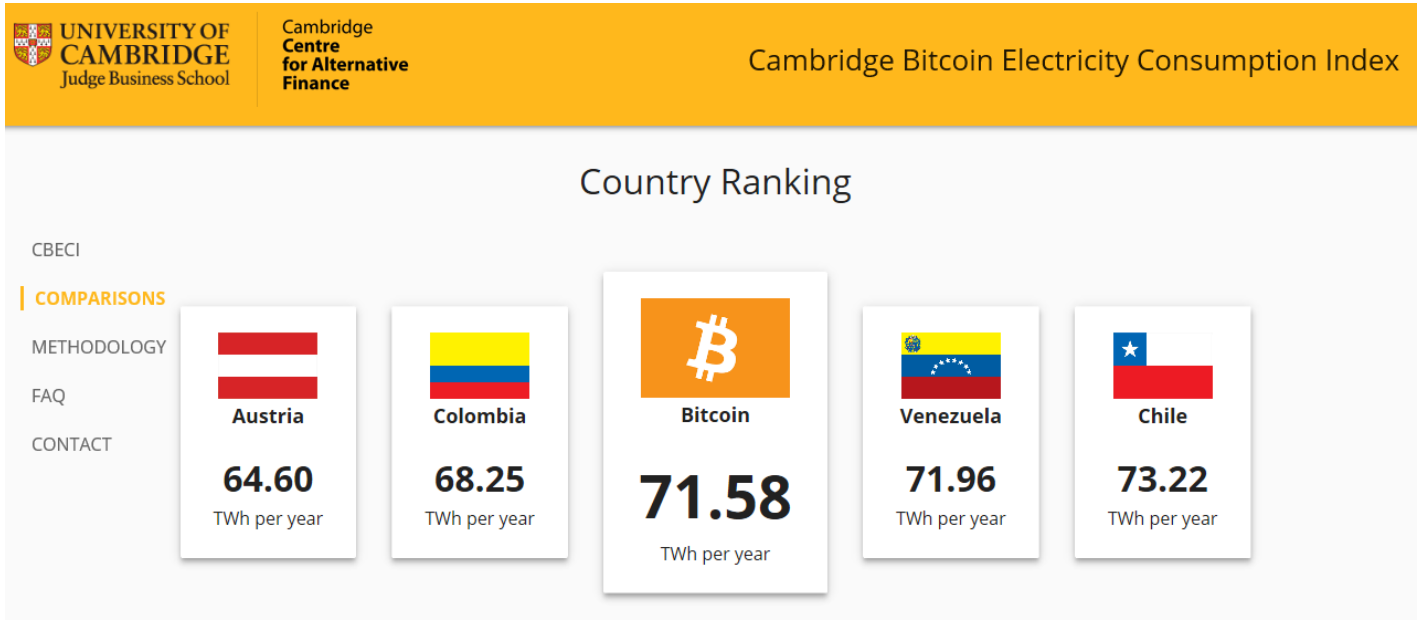
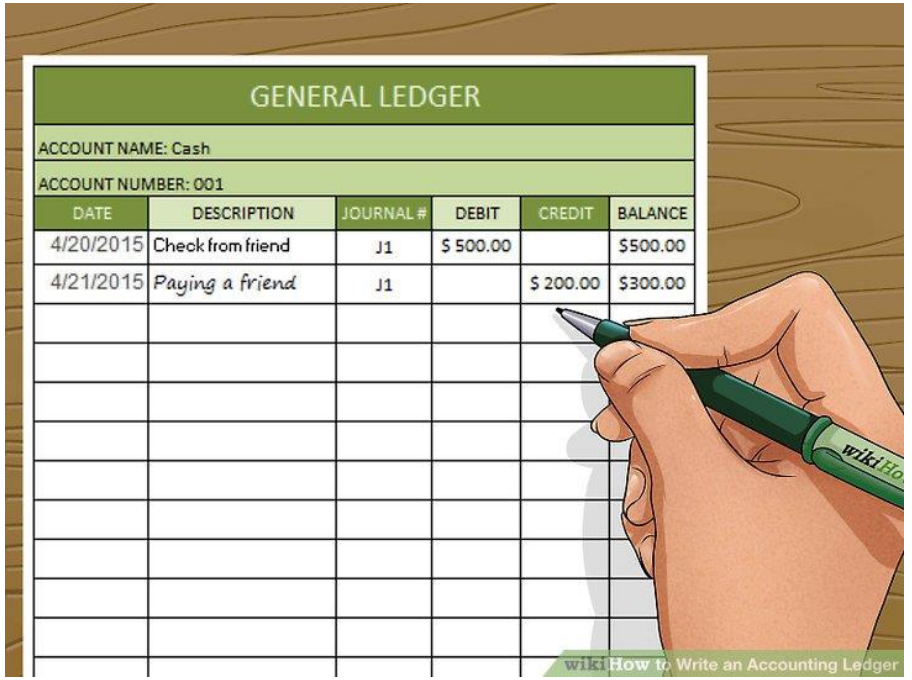

Blockchain 101

- Bitcoin ≠ Blockchain

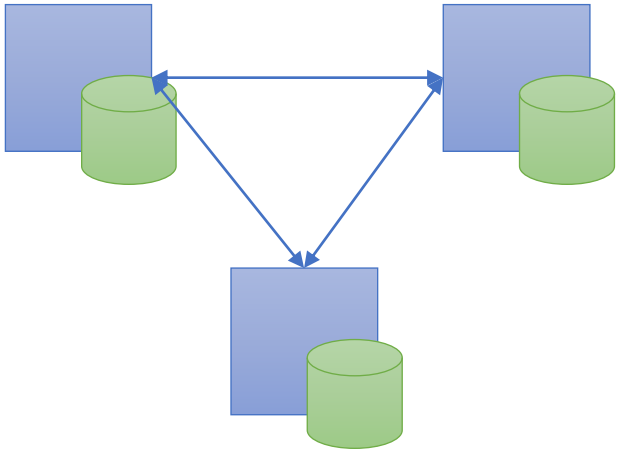


- Blockchain = Distributed ledger (data structure) + consensus algorithm

Distributed ledger



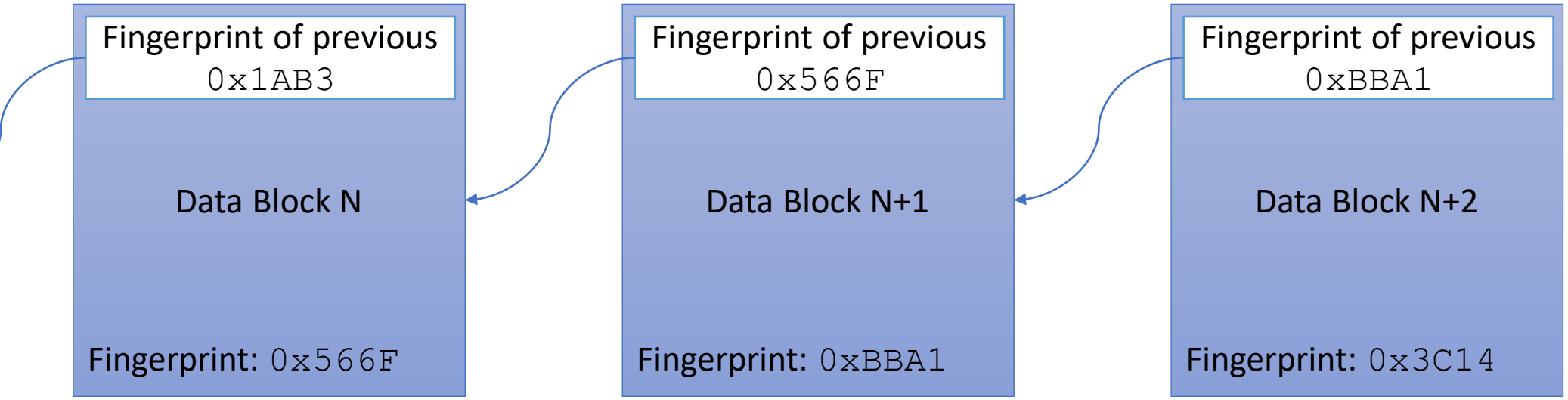
- Append-only log of operations
  - Money transfers
  - Database modifications
  - Medical history
  - …



- Data resides on several distributed nodes
  - Durability
  - High availability
  - Shared data

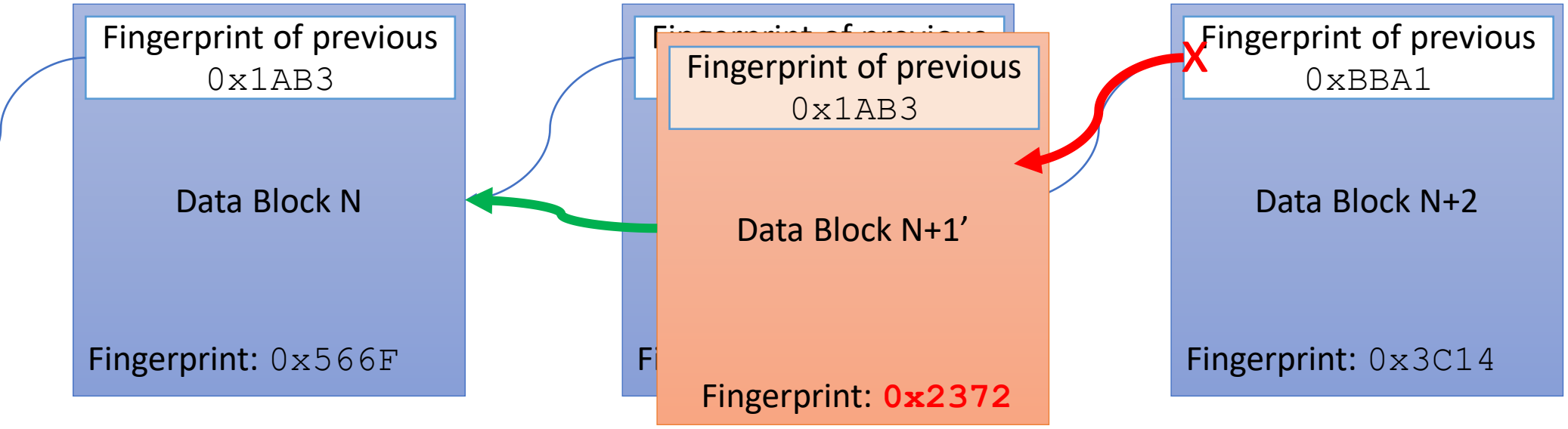
How do we know that the data is correct?

- Form a “chain” of data (blocks)
- Fingerprint of data block (including the pointer to previous)



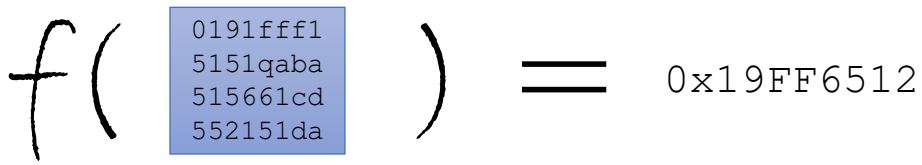
How do we know that the data is correct?

- Form a “chain” of data (blocks)
- Fingerprint of data block (including the pointer to previous)
  - E.g., SHA256 hash of all bytes
  - Cannot change value afterwards – fingerprint will not match



Digression: Cryptographic primitives

- Cryptographic hash function
  - Cannot be reversed
  - Collision resistant

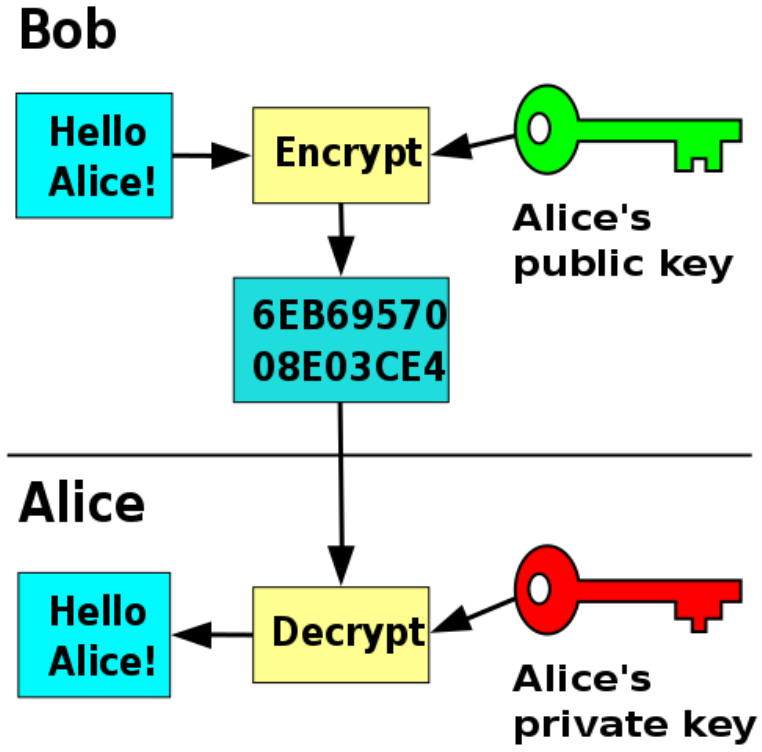


- Examples
  - SHA256, SHA-3, bcrypt
  - ~~MD5~~

Digression: Cryptographic primitives

- Public key cryptography
  - Each actor has a public and a private key
  - Derived from a source of randomness

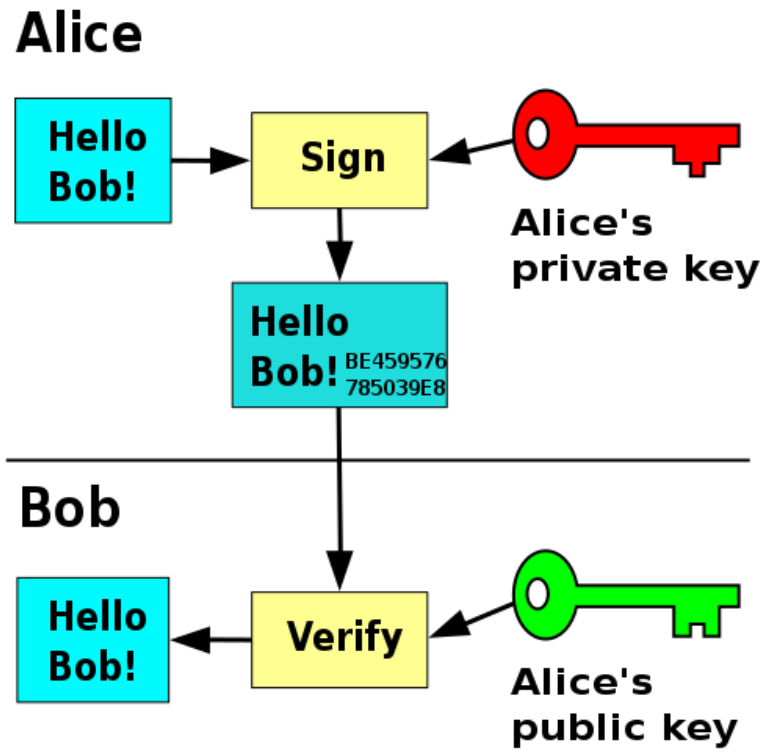
1. Alice creates a key-pair
2. Publishes the public key
3. Bob can send Alice a message that only she can decrypt (not even Bob)



Digression: Cryptographic primitives

- Public key cryptography
  - A key-pair is not an identity!
  - Key authentication authority – match a key to an identity

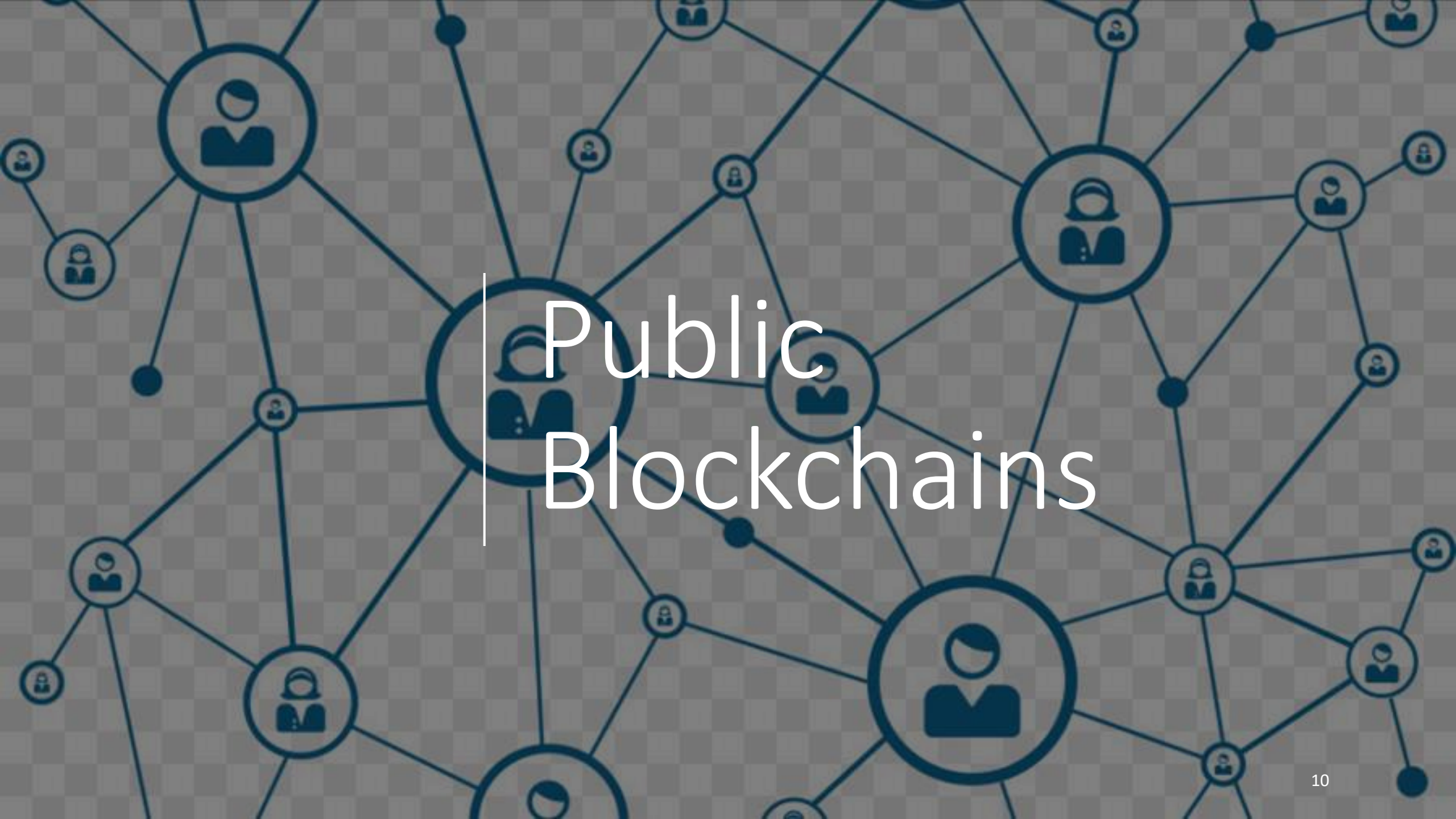
1. Alice creates a key-pair
2. Registers its public key with a Certification Authority (CA)
3. Alice signs a message with her private key (encrypt a hash value)
4. Bob checks with CA what is Alice's public key
5. Bob and anyone else can verify that a message comes from Alice by decrypting the hash with the public key (no other key could decrypt it!)



We can verify the chain, how do we add to it?

Public system – Everyone and anyone can add to it

Permissioned system – A consortium or third party authenticates nodes



Public Blockchains

How do we ensure that everyone can “speak”?

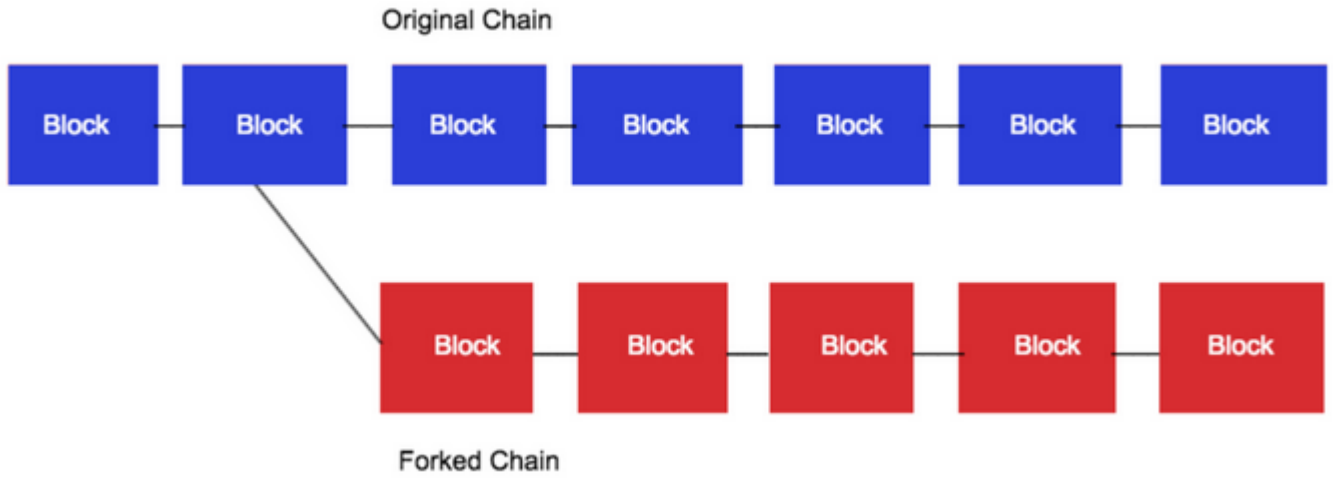
- Imagine we go for “majority”
  - What node is online/offline?
  - Easy to create fake nodes (IP address? MAC?, etc.) – **Sybil attack**

- Proof of work
  - Relies on past block, current block and a source of randomness
    - Hash(“previous-seq:previous-hash:this-seq:this-hash:RANDOM”)=0x0000556
    - Compute hash with N leading zeros – needs 2^N tries
  - Reward to the one that discovers the new hash (x Bitcoins)
    - Miners collect transactions, compute block and start “guessing”

Issues with proof of work

- High compute cost
  - Continuously computing
  - Competing for same block

- Transaction finality
  - Consider the longest chain as “real”
  - Can take minutes to decide



Can we replace Proof of Work?

PROOF OF WORK

The probability of mining a block is determined by how much computational work is done by the miner.

A reward is given to the first miner to solve the cryptographic puzzle of each block.

PROOF OF STAKE

The probability of validating a new block is determined by how large of a stake a person holds (how many coins they possess).

The validators do not receive a block reward, instead they collect network fees as their reward.

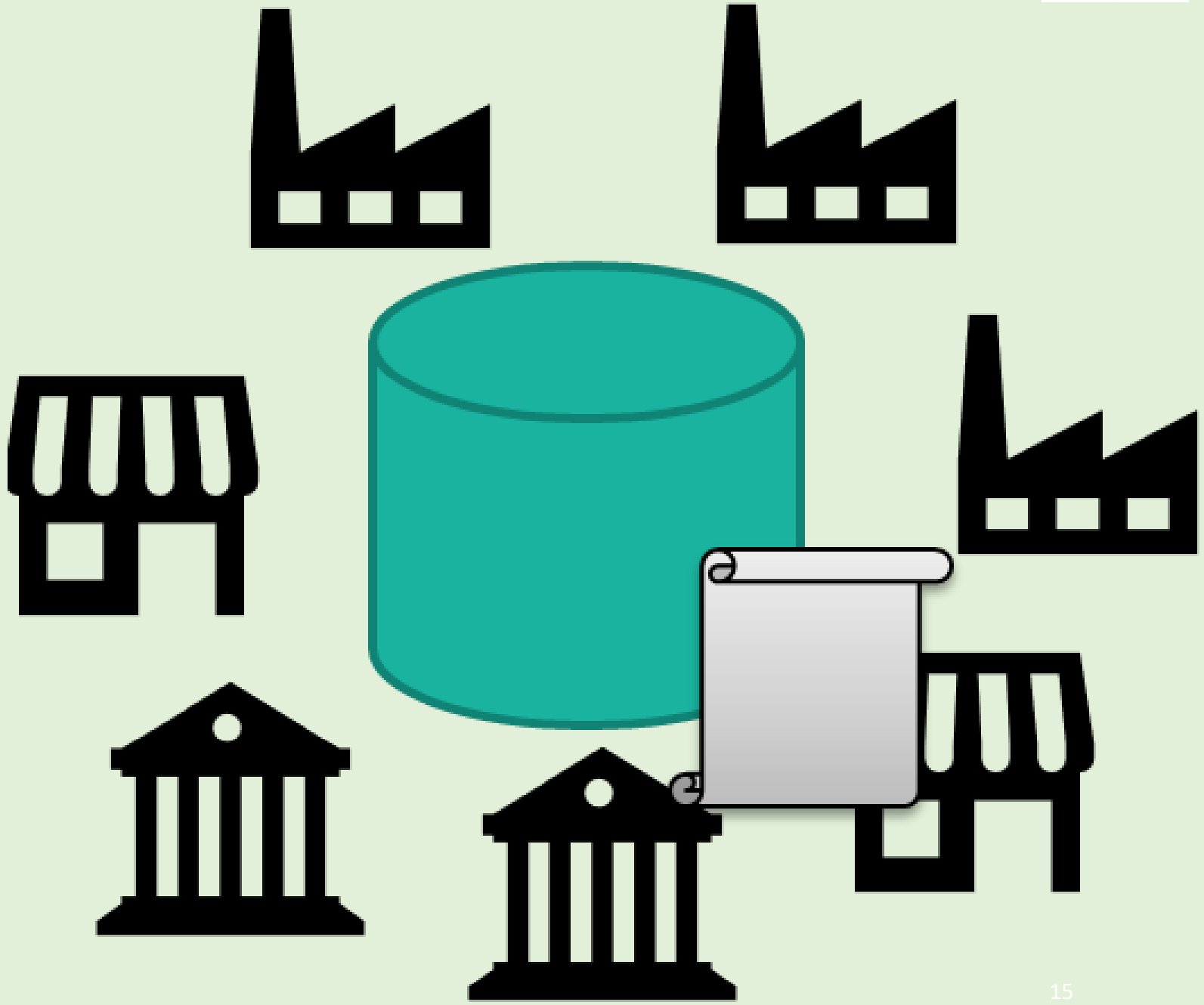
Public Blockchain use-cases

- No “onboarding” process – anyone can join
- Pseudonymous
- Privacy not guaranteed (unless encrypted data)

- E.g., cryptocurrencies

- Example platforms
  - Bitcoin (PoW)
  - Ethereum (PoW, will move to PoS)
  - Tezos (PoS)
  - Tendermint* (PoS)

Permissioned Blockchains

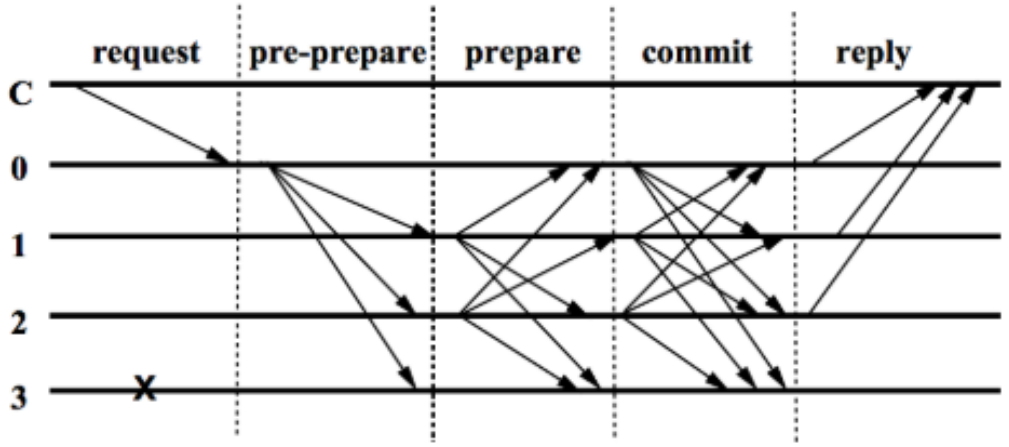


Consensus Algorithms

- Regular BFT Consensus
  - Relies on majority votes – we know wh the members are

  ✓Finality → Low latency

  ✓No “mining” → High throughput



- Can have issues with scalability
  - Not necessarily every node takes part in consensus (think of it like PoS)

Permissioned Blockchain use-cases

- Onboarding process – Identities
- Data is not public, but can be still seen by all members – example of partial secrecy in tutorial

- E.g., Companies sharing datasets, E-governance, Supply chain management, etc.

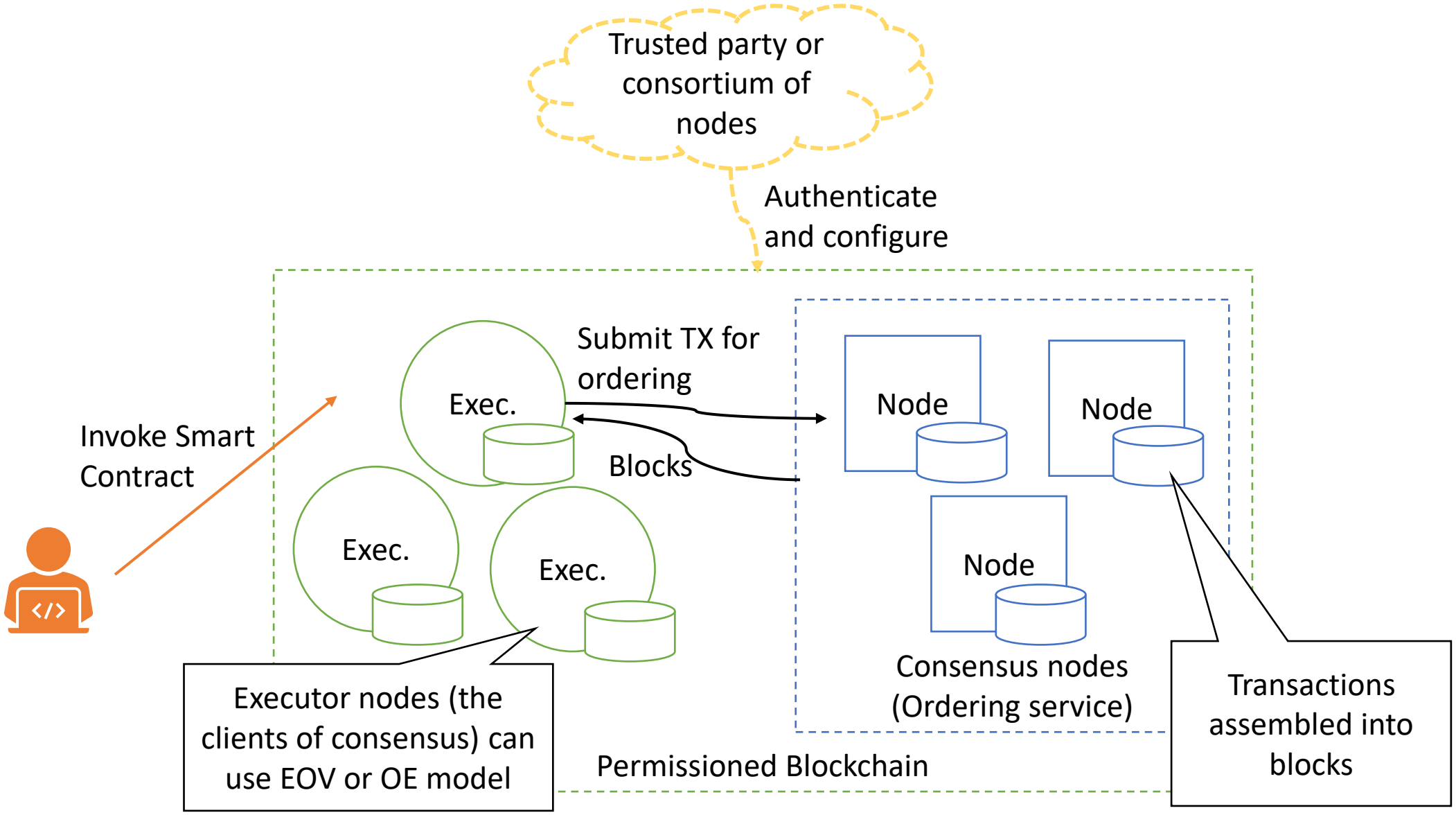
- Example systems: Corda, R3, Hyperledger Fabric

What are Smart Contracts?

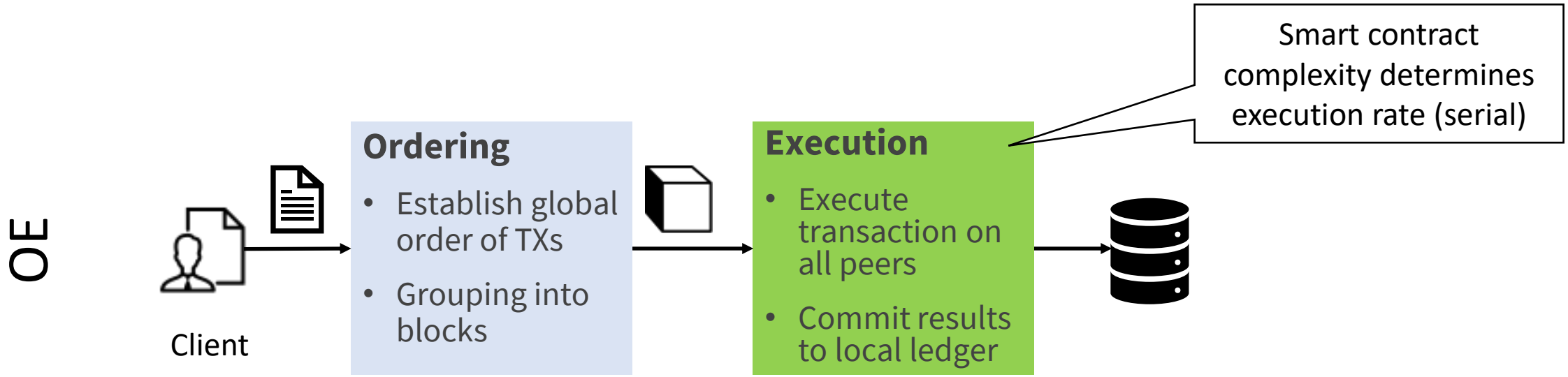
e.g., “Pull images from satellite, analyze them and record state onto ledger”

- Transaction submitted to ledger can be:
  - “Statement”, e.g., “X->Y 1 Euro”
  - A program – executed and its result recorded into the ledger
- Smart Contracts typically manipulate key/value pairs
- Stored on the Ledger
  - The code of the programs are also subject to consensus, cannot be tampered with
- Important aspects
  - Cost of executing the contract matters because it is replicated!
  - Contracts need to be deterministic (same input → same output)

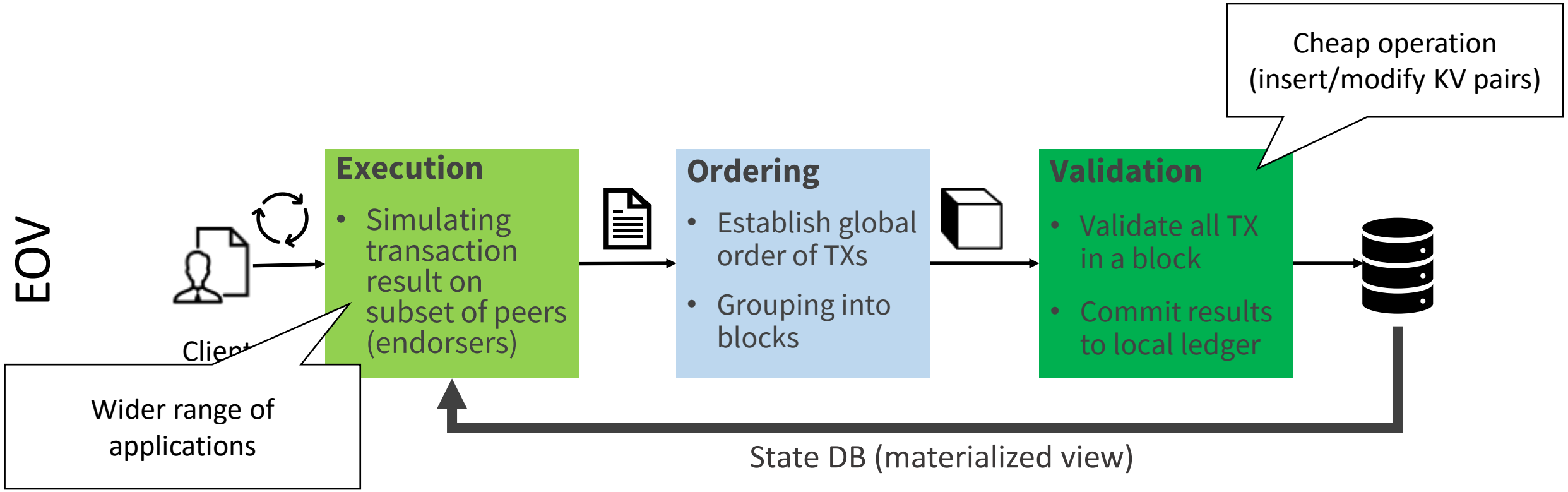
Anatomy of a Blockchain System



Order-Execute



Execute-Order-Validate



Summary (energy efficiency)

- Blockchain is not fundamentally wasteful
  - PoW is wasteful!

- Still, should be used over existing DB technology only if, e.g.:
  a) Sharing dataset across several entities and no one is trusted to “own” the data
  b) It is important to publicly record data without the fear of being censored
  c) It is important to provide access to the service without having to “ask” permission