Blockchain-based Database for Multi-party Systems

Dr. Leonardo Aniello
l.aniello@soton.ac.uk

Cyber Security Research Group,
Electronics and Computer Science,
University of Southampton

Distributed Ledger Technology Workshop - February 1st 2018 - Università degli Studi di Perugia

Multi-party Systems

- Main properties
  - Peer parties need to interact with each other
  - Little trust among parties
- Problems
  - High cost for transactions verification and clearance
  - Entrust to a third party
    - Trustworthiness
    - Single-point-off-failure
    - Performance bottleneck

Multi-party Systems

- Cloud federation
  - Interactions among parties regarding governance operations and inter-party transactions
- Transactive energy
  - Interactions among consumers/prosumers to trade and exchange energy within a smart grid
- Supply chain
  - Interactions among involved organizations to certify the stages a product has gone through along the chain

Multi-party Systems

Need for a
well-performing, reliable and trustworthy
mechanism to support inter-party transactions

Multi-party Systems

- Basic idea
  a decentralised database for inter-party transactions, able to provide
  - *high availability*
  - *strong data integrity*
  - *good performances*

What about using a blockchain?

Blockchain-based Systems

- On top of public permissionless blockchains based on PoW
  - Bitcoin 
  - Ethereum 

  ➢ Consensus based on Proof-of-Work (PoW)

Public/Private blockchains:
refer to access restrictions to data

Permissionless/Permissioned blockchains:
refer to restrictions on miner identities

- Relevant limitation: performances

  [T.Bocek, B.Stiller: Smart Contracts - Blockchains in the Wings. In Digital Marketplaces Unleashed. Springer]
  - High latency (10 minutes in Bitcoin, 12 seconds in Ethereum)
  - Low throughput (3-7 tx/s in Bitcoin, 23-25 tx/s in Ethereum)

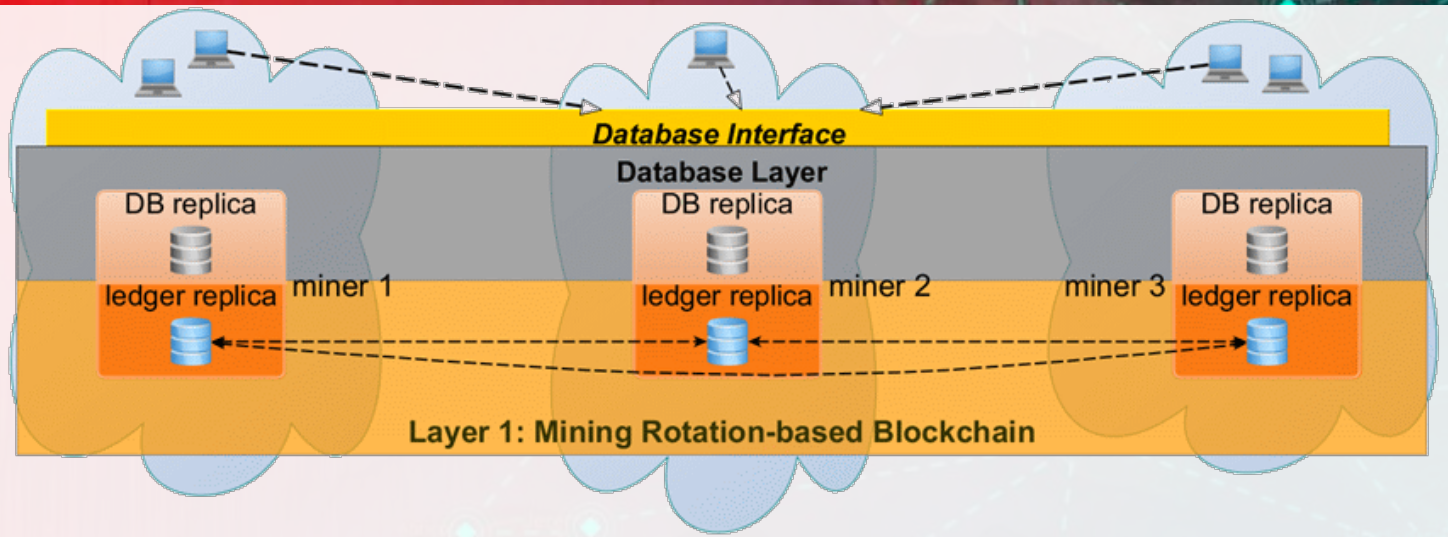
Blockchain-based DB on Two Layers

1. A private permissioned blockchain in the 1st layer
   - To ensure low latency and high throughput
2. A public permissionless blockchain in the 2nd layer
   - To ensure strong integrity guarantees

We consider a key-value store

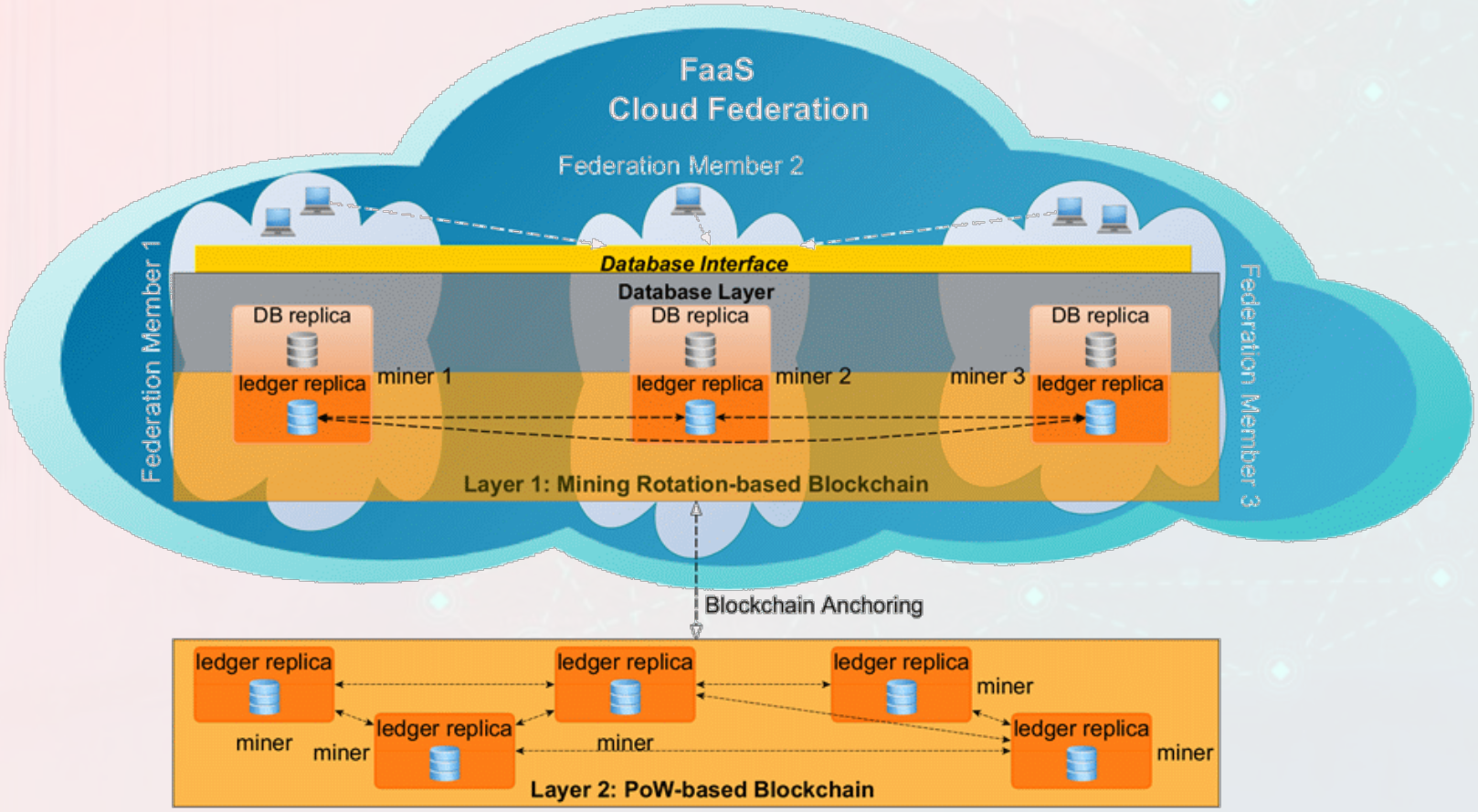
E.Gaetani, L.Aniello, R.Baldoni, F.Lombardi, A.Margheri, V.Sassone
Blockchain-based Database to Ensure Data Integrity in Cloud Computing Environments
ITASEC 2017

Blockchain-based DB: First Layer



- Transactions are set operations
  - The blockchain is the DB redo log
- Leader rotation algorithm for consensus
  - Much faster than PoW
  - Exchanged messages signed by parties

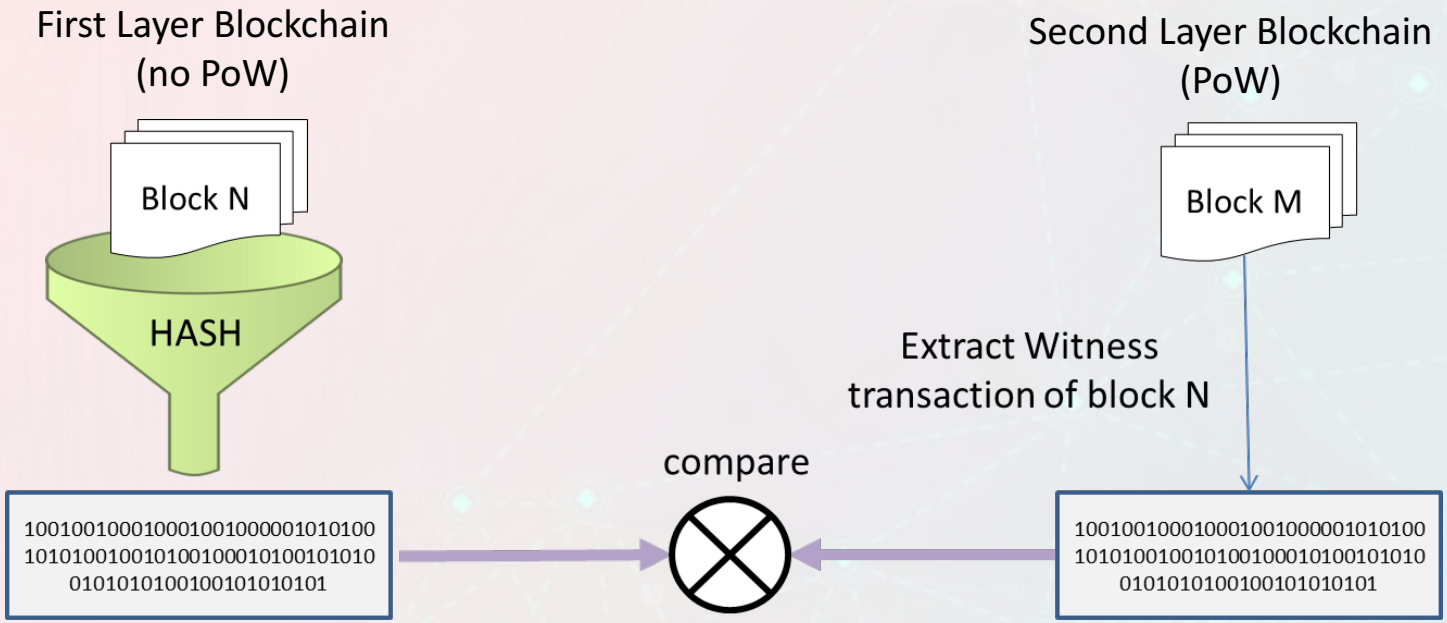
Blockchain-based DB Overall Architecture



Anchoring between the Layers

- ***Anchoring:*** link the first layer blockchain with a PoW blockchain
- ***Witness Transaction*****:** periodically the hash of the first layer blockchain up to the current txn is sent to the second layer

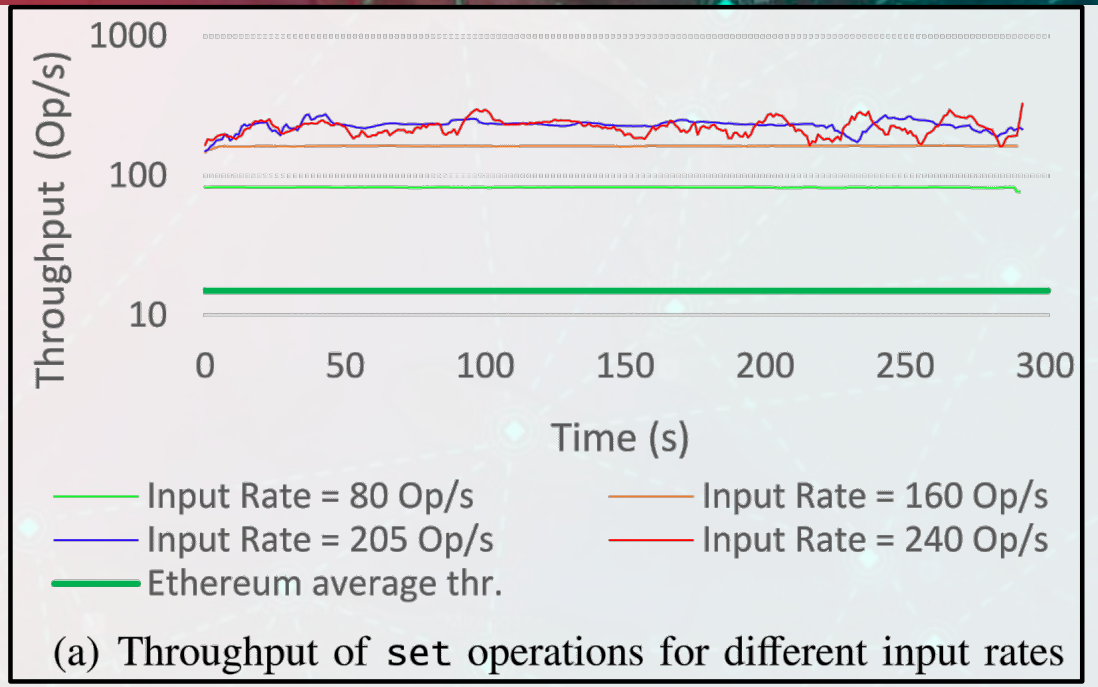
Integrity Verification



Hashes act as forensics evidence for proving and validating the integrity of the data stored in the first-layer blockchain

Prototype Evaluation - Throughput

- Implemented in Java
- Communication through JavaRMI and Jgroup
- 6 nodes



(a) Throughput of `set` operations for different input rates

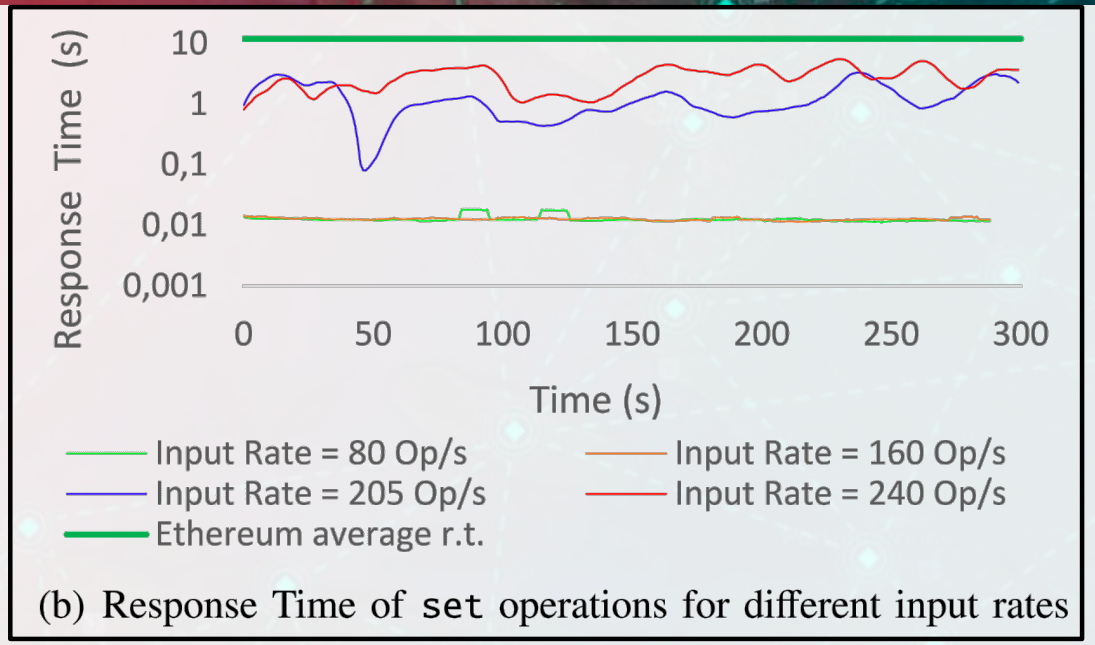
L.Aniello, R.Baldoni, E.Gaetani, F.Lombardi, A.Margheri, V.Sassone

A Prototype Evaluation of a Tamper-resistant High Performance Blockchain-based Transaction Log for a Distributed Database

EDCC 2017

Prototype Evaluation – Response Time

- Implemented in Java
- Communication through JavaRMI and Jgroup
- 6 nodes



(b) Response Time of `set` operations for different input rates

L.Aniello, R.Baldoni, E.Gaetani, F.Lombardi, A.Margheri, V.Sassone
A Prototype Evaluation of a Tamper-resistant High Performance Blockchain-based Transaction Log for a Distributed Database
EDCC 2017

What Integrity Guarantees in a blockchain?

- Bitcoin and Ethereum: what happens between transaction submission and its inclusion in a block?
- Our solution: what happens when the witness transaction is stored in the second layer?

How to quantify data integrity guarantees,
to enable comparison among
different blockchain-based database solutions?

How to Measure Integrity Guarantees?

We propose to measure the integrity as the effort required for an attacker to tamper with data in the blockchain

E.Gaetani, L.Aniello, R.Baldoni, F.Lombardi, A.Margheri, V.Sassone

Blockchain-based Database to Ensure Data Integrity in Cloud Computing Environments

ITASEC 2017

How to Measure Integrity Guarantees?

- In our proposed solution, integrity grows over time
  - Operation stored in the first layer only
    - Effort to subvert the PBFT consensus protocol: with N nodes, $f \geq N/3$ have to be compromised [*]
  - Corresponding witness transaction stored in the 2nd layer
    - Additional effort to subvert PoW-based consensus: hold the majority of computational power [**] [***]

[*] M.Castro, B.Liskov, et al. Practical byzantine fault tolerance. In OSDI, volume 99, 1999
[**] J.Garay, A.Kiayias, and N.Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications. Cryptology ePrint Archive, Report 2014/765, 2014
[***] A.Miller and J.J.LaViola Jr. Anonymous Byzantine Consensus from Moderately-Hard Puzzles: A Model for Bitcoin, 2014

Ongoing/future Work on Blockchain-based Systems

- Formalisation and analysis of the integrity metric
- Benchmark to assess security properties and performance of permissioned blockchain
- Extend architecture to support Smart Contracts
- Towards (H2020 calls) a blockchain-based middleware
  - Data management (data life-cycle)
  - Programmability (function-as-a-service)
  - Access control, privacy preservation, anonymity, …

THANKS FOR THE ATTENTION

BLOCKCHAIN-BASED DATABASE FOR MULTI-PARTY SYSTEMS

DR. LEONARDO ANIELLO
l.aniello@soton.ac.uk

CYBER SECURITY RESEARCH GROUP,
ELECTRONICS AND COMPUTER SCIENCE,
UNIVERSITY OF SOUTHAMPTON

DISTRIBUTED LEDGER TECHNOLOGY WORKSHOP - FEBRUARY 1ST 2018 - UNIVERSITÀ DEGLI STUDI DI PERUGIA

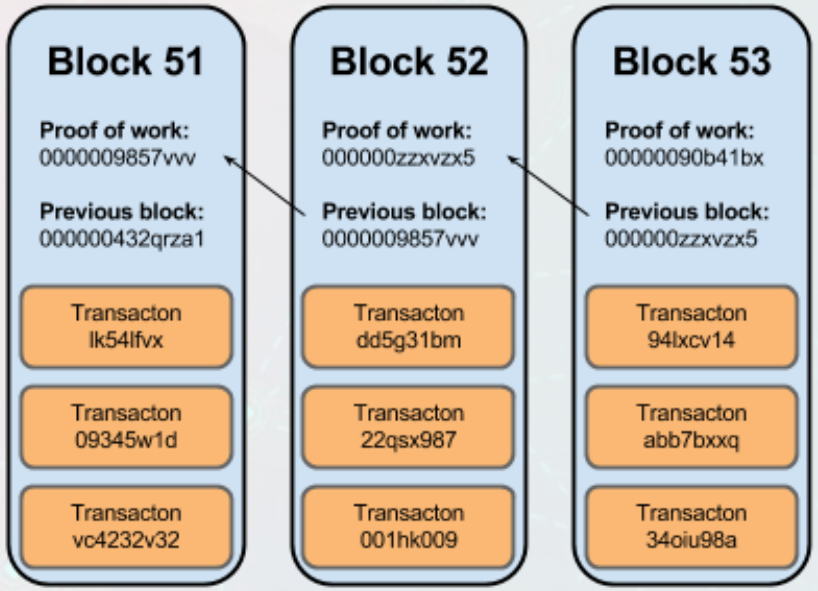
Backup Slides

Blockchain-based DB for Federated Clouds

- Context: *federation of clouds, multi-party environment*
- Need: *any governance action has to be carried out with the consensus of all the member of the federation*
  - No centralised federation control, democratic approach
  - Ensure strong integrity of the contracts regulating inter-cloud interactions
  - Ensure reliable tracking of inter-cloud interactions
- Proposed solution: *blockchain-database where to store contracts and inter-cloud interaction logs*

What is a blockchain?

- Transaction ledger replicated over a trustless p2p network of miners
- Transactions organized in blocks
  - Changing a block requires the update of all the following blocks

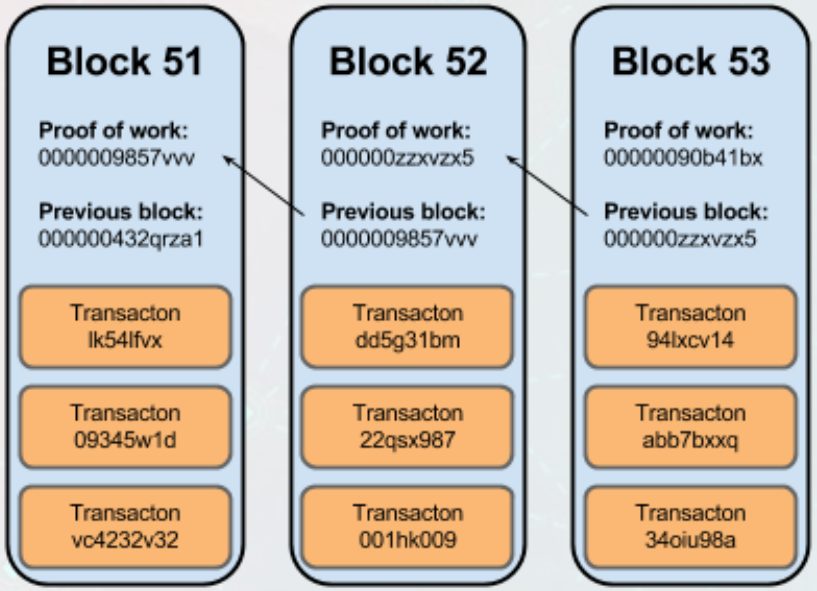


- Consensus among miners on transactions and their order
  - Achieved through mining
  - Proof-of-Work (PoW): time-consuming mathematical challenge

What about forks?

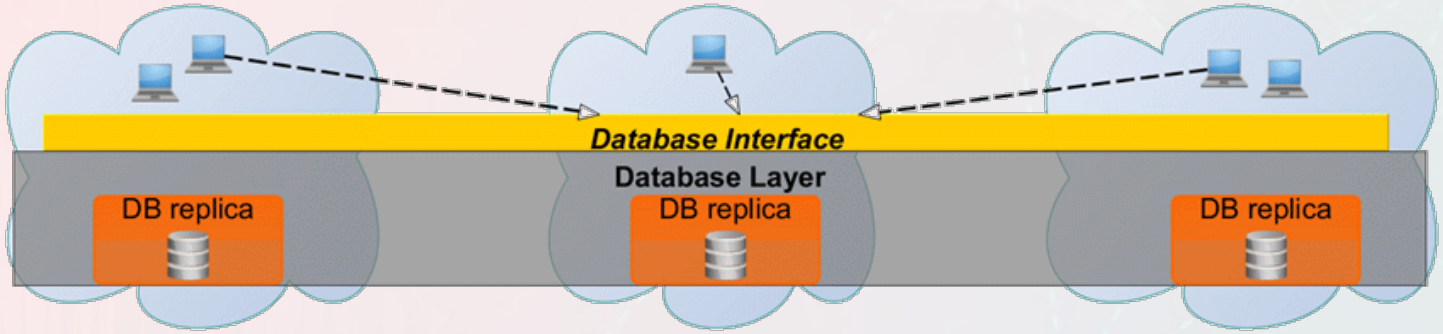
- They can occur
  - On purpose by malicious miners
  - By chance by honest miners
- Honest miners mine on top of the **longest chain** they know about

➢ *With honest miners having the majority of computational power, the longest chain is expected to include honest blocks only*



- ***Strong integrity*** *derives from the difficulty for an attacker to get the majority of the total computation power*
- ***High availability*** *derives from pure p2p architecture*

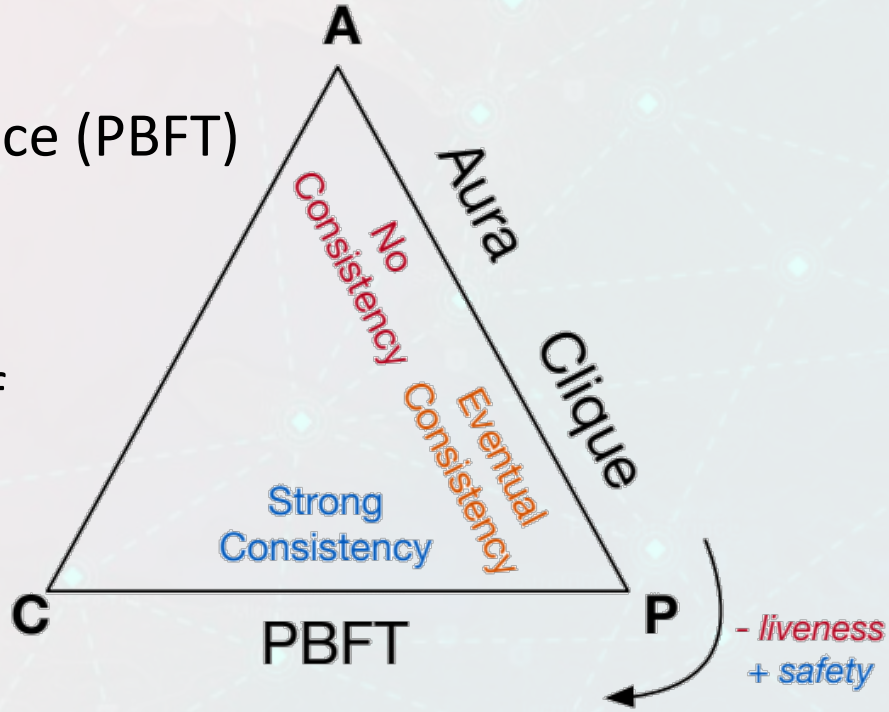
Blockchain-based DB for Federated Clouds



- Key-value store database
- One replica for each federation member
- Client operations:
  - `set(key, value)` → `ack/nack`
  - `get(key)` → `value`

What consensus algorithm for the 1st layer?

- Comparison between
  - Practical Byzantine Fault Tolerance (PBFT)
  - Proof-of-Authority (PoA)
- In terms of
  - Consistency/Availability tradeoff (CAP Theorem)
  - Performance
- In a byzantine scenario



S.De Angelis, L.Aniello, R.Baldoni, F.Lombardi, A.Margheri, V.Sassone
PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain
to be presented at ITASEC 2018