INFO8002

Large-Scale Data Systems

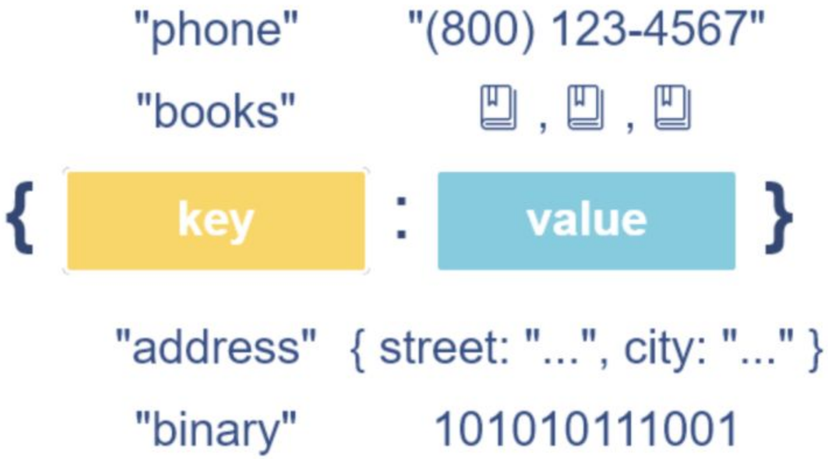
Project Presentation

Academic year 2021-2022

LIÈGE université

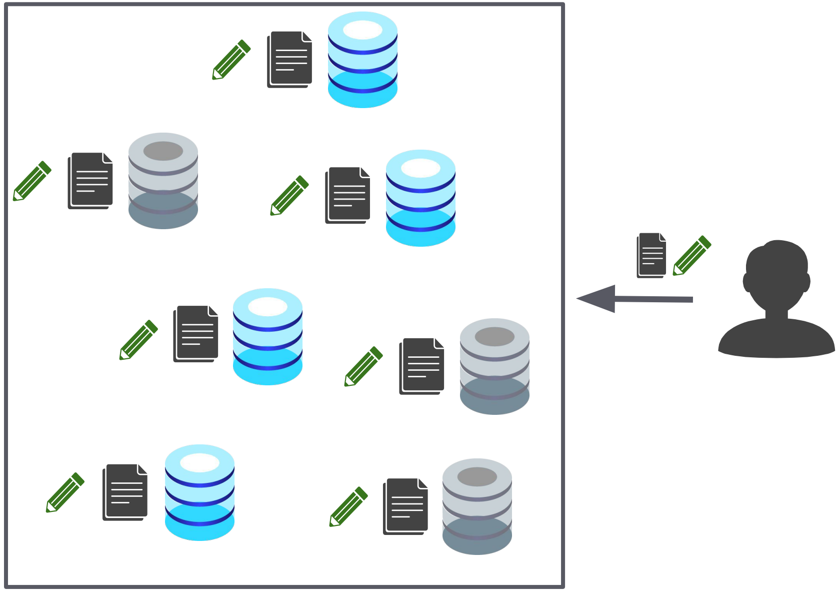
Project: Key-Value Storage

Your task is to implement a byzantine fault-tolerant, decentralized and immutable **distributed Key-Value Storage**.

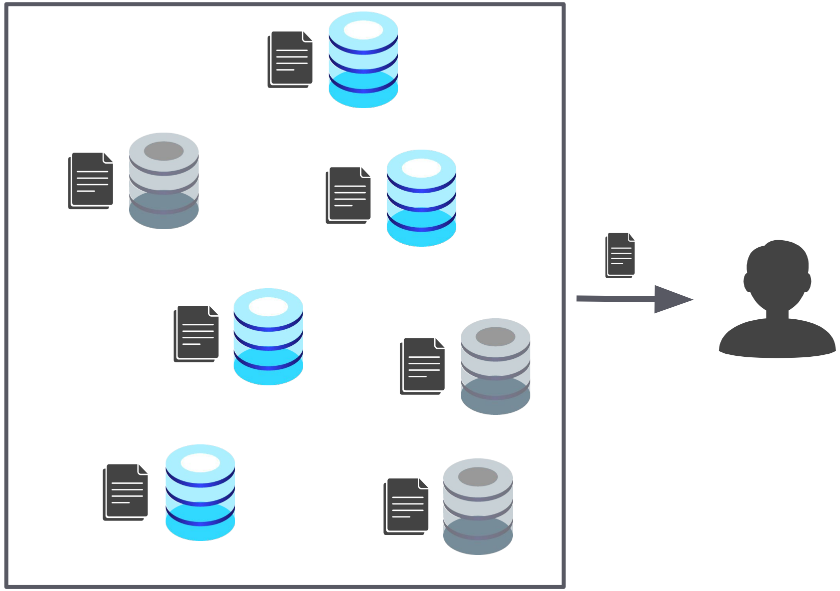


Project: Key-Value Storage

Put query



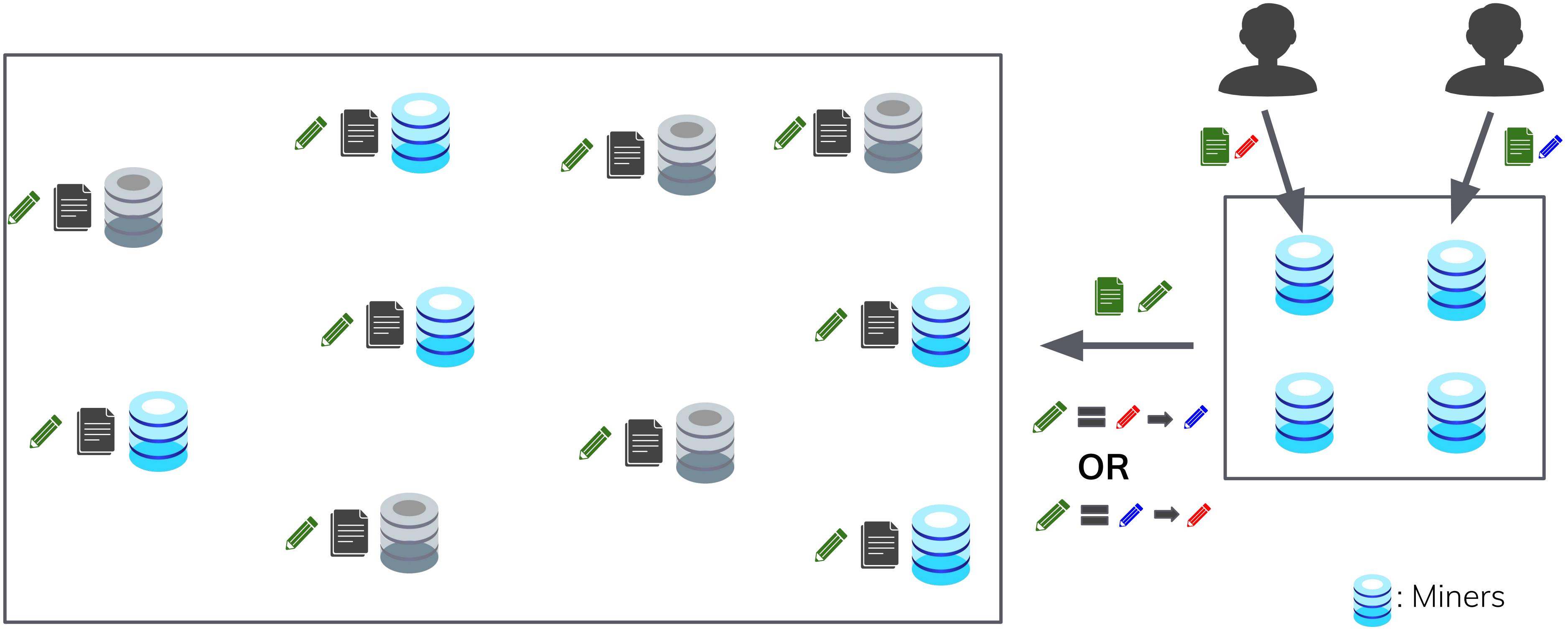
Retrieve query



: Miners

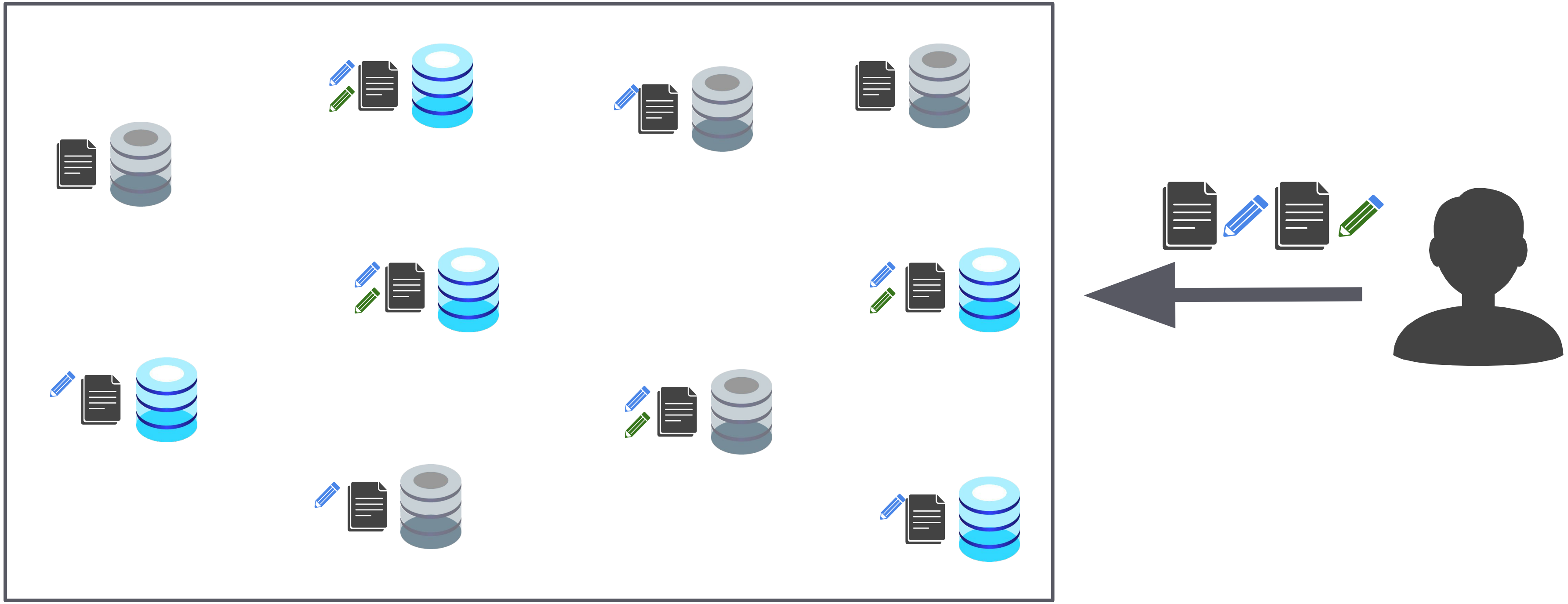
Project: Key-Value Storage

Consensus Problem



Project: Key-Value Storage

Consistency Problem



Project: Blockchain Implementation

Blockchain is a decentralised distributed ledger

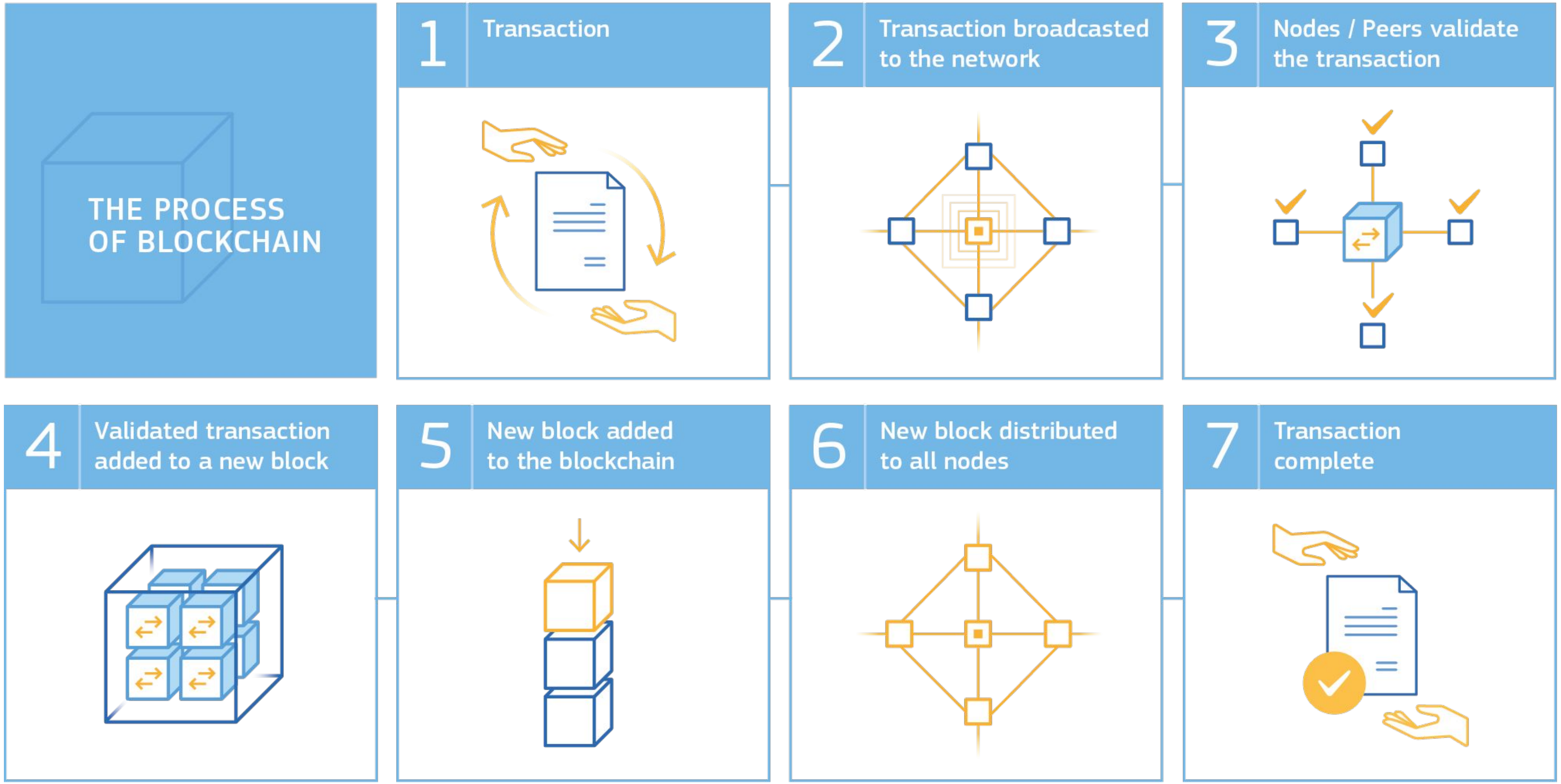
Project: Blockchain Implementation

Blockchain is a decentralised distributed ledger

- Everyone can take part in the consensus process and becomes a **miner**.

- Immutable
- Fully Replicated

Project: Blockchain Implementation



THE PROCESS OF BLOCKCHAIN

1 Transaction

2 Transaction broadcasted to the network

3 Nodes / Peers validate the transaction

4 Validated transaction added to a new block

5 New block added to the blockchain

6 New block distributed to all nodes

7 Transaction complete

Project: REST API & Flask

REST is an acronym for **RE**presentational **S**tate **T**ransfer and an architectural style for distributed hypermedia systems.

Flask is a web framework that provides you with tools, libraries and technologies for building RESTfull APIs for web applications.

- Here, we can use it to create ***endpoints*** for the “*simulation*” of our peer-to-peer network and perform *real-time operations* on our nodes.

```
@app.route ('/new_transaction', methods = ['POST'])
def new_transaction ():
      ...
```

Project: Report

Architecture & Implementation

- Specifications & Pseudo-code (cfr. *Practical Sessions* and *Lectures*)
- Mapping between the pseudo-code and your implementation should be **clear**.

Experiments

- Show the resilience of your implementation against faulty processes.
    - Crash-stop? Omission? Crash-recovery? Byzantine?
    - What happens if one or several miners fails? During the consensus?

- Show the efficiency of your implementation
    - How fast transactions can be added? Worst-Case?
    - How fast transactions can be retrieved? Worst-Case?

- Show the impact on your implementation of the 51% attack?

Project: Report

Discussions

- **Applicability**
    - Why is the blockchain relevant to this problem?
    - How does the blockchain address the **CAP** trade-off? Is it fitting our goals?
    - Is your architecture truly decentralised? Why?

- **Scalability**
    - Is your blockchain implementation scalable? Why (not)?
        - Broadcast? **Put/Retrieve** methods? Consensus?
    - Why is scalability important? In which contexts?

- **Anything else that is relevant.**

Project: Key-Value Storage