



UCDAVIS

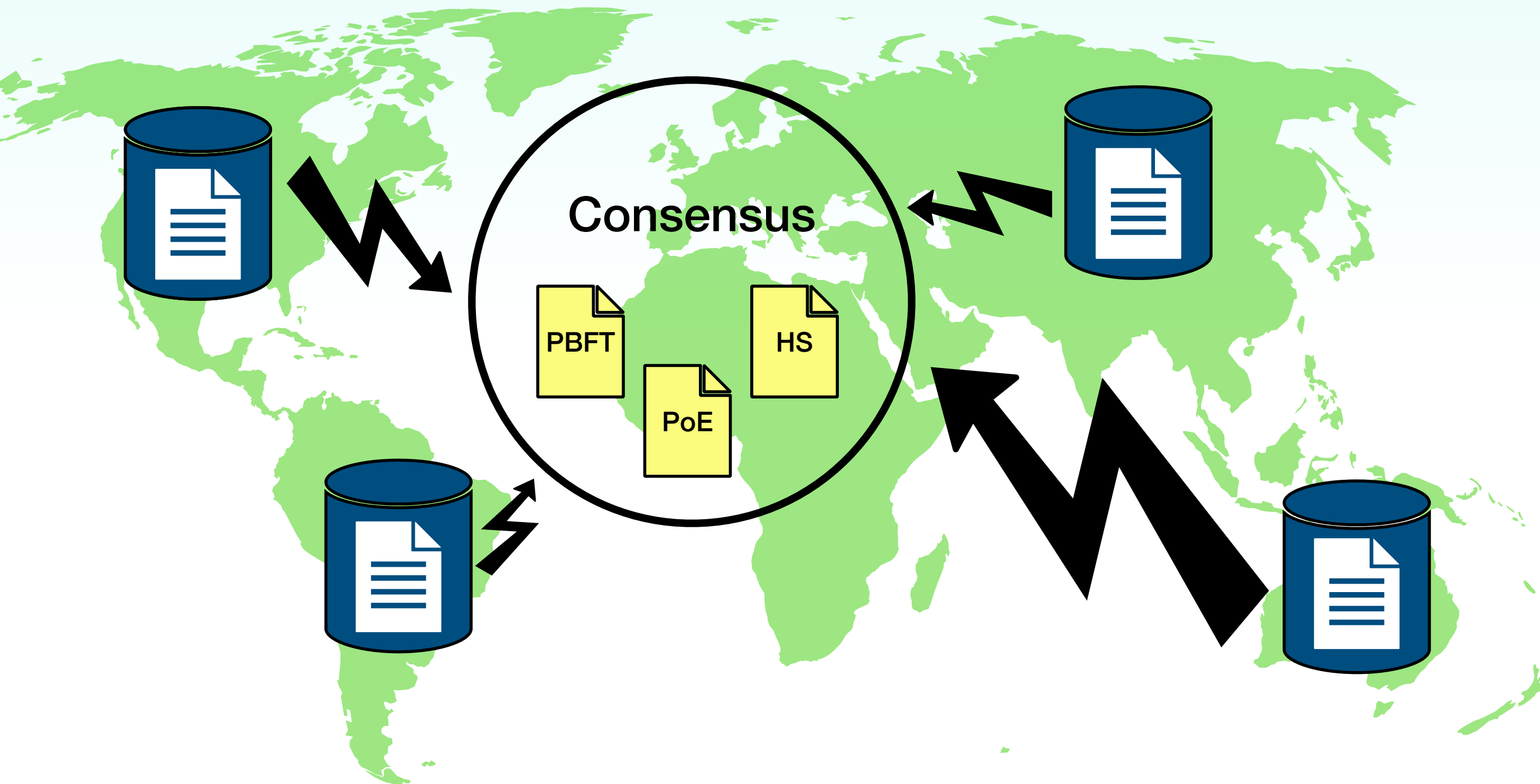
ResilientDB Overview

An Open-Source High-Performance Permissioned Blockchain Platform

Sep 25, 2024



What is Permissioned Blockchain?



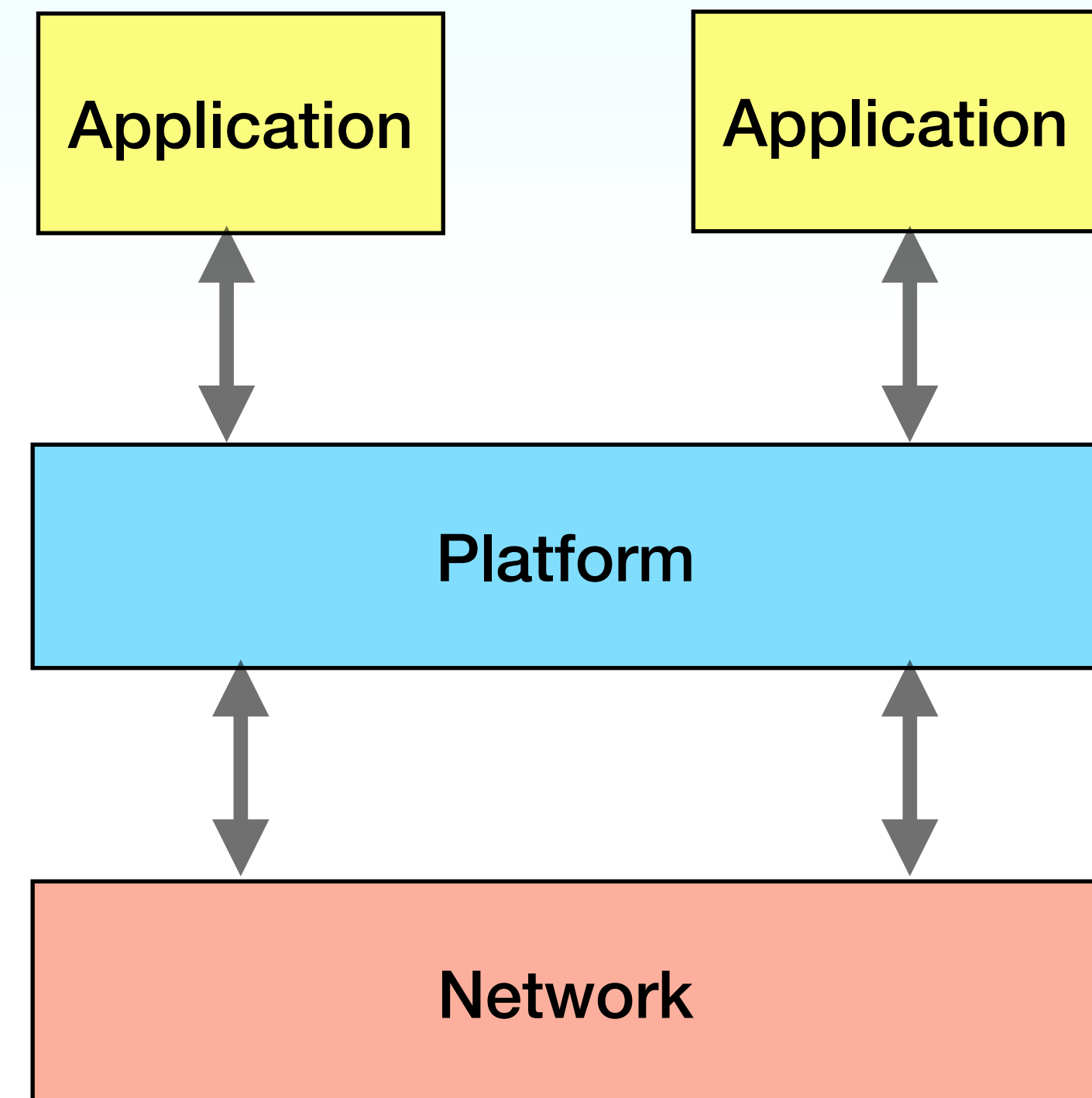
- **Distributed database consists of a set of replicas (participants).**
- **Each replica holds a copy of the ledger, which is a chain of blocks containing transactions.**
- **Fault Model: Byzantine replicas may behave arbitrarily.**
- **Consensus Guarantees: Safety; Liveness.**
- **Consensus Protocols: PBFT, PoE, HotStuff, etc**

ResilientDB Transaction Workflow

Application: Submit Transactions

Platform: Commit Transactions

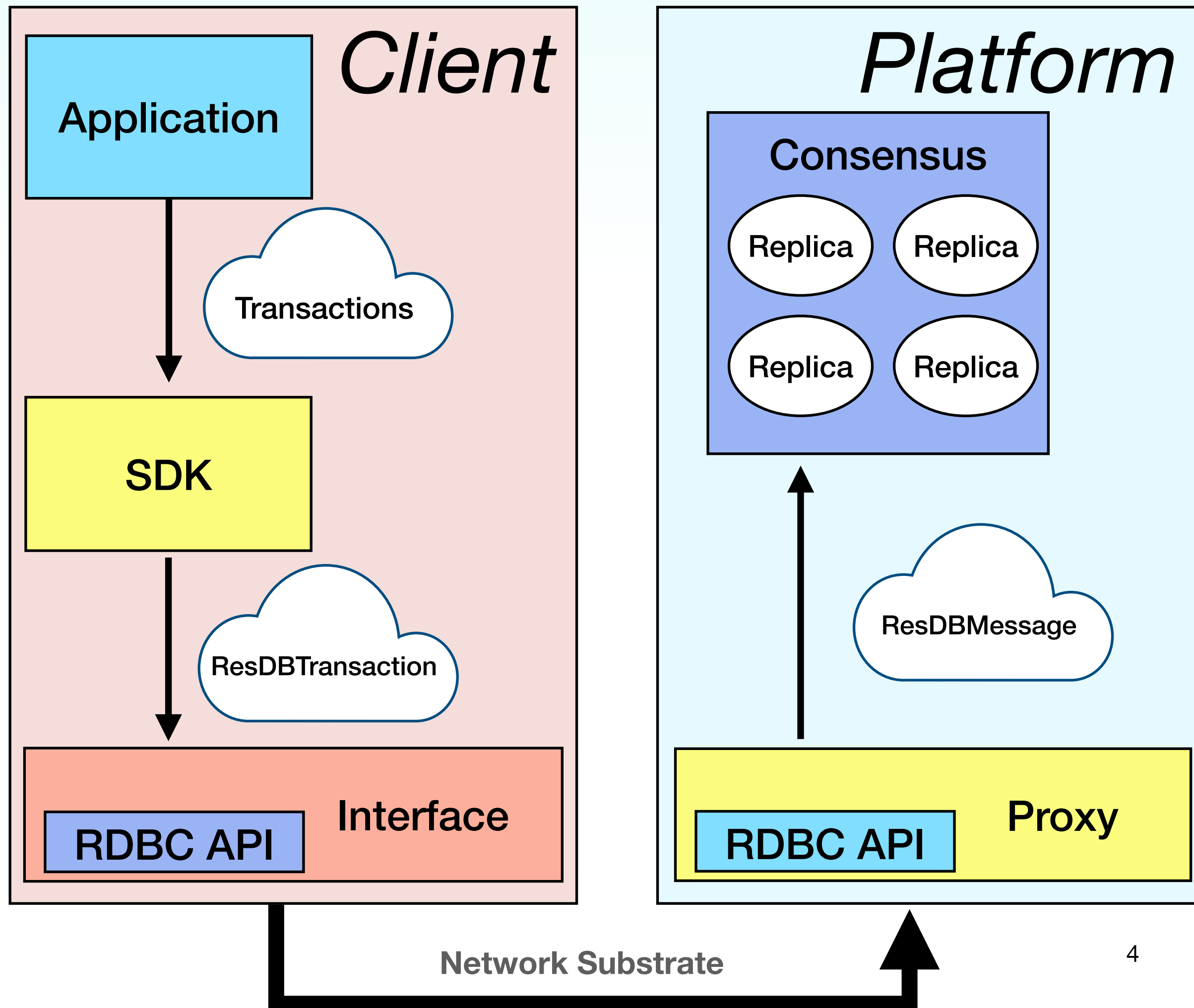
Network: Exchange Messages



ResilientDB Transaction Workflow

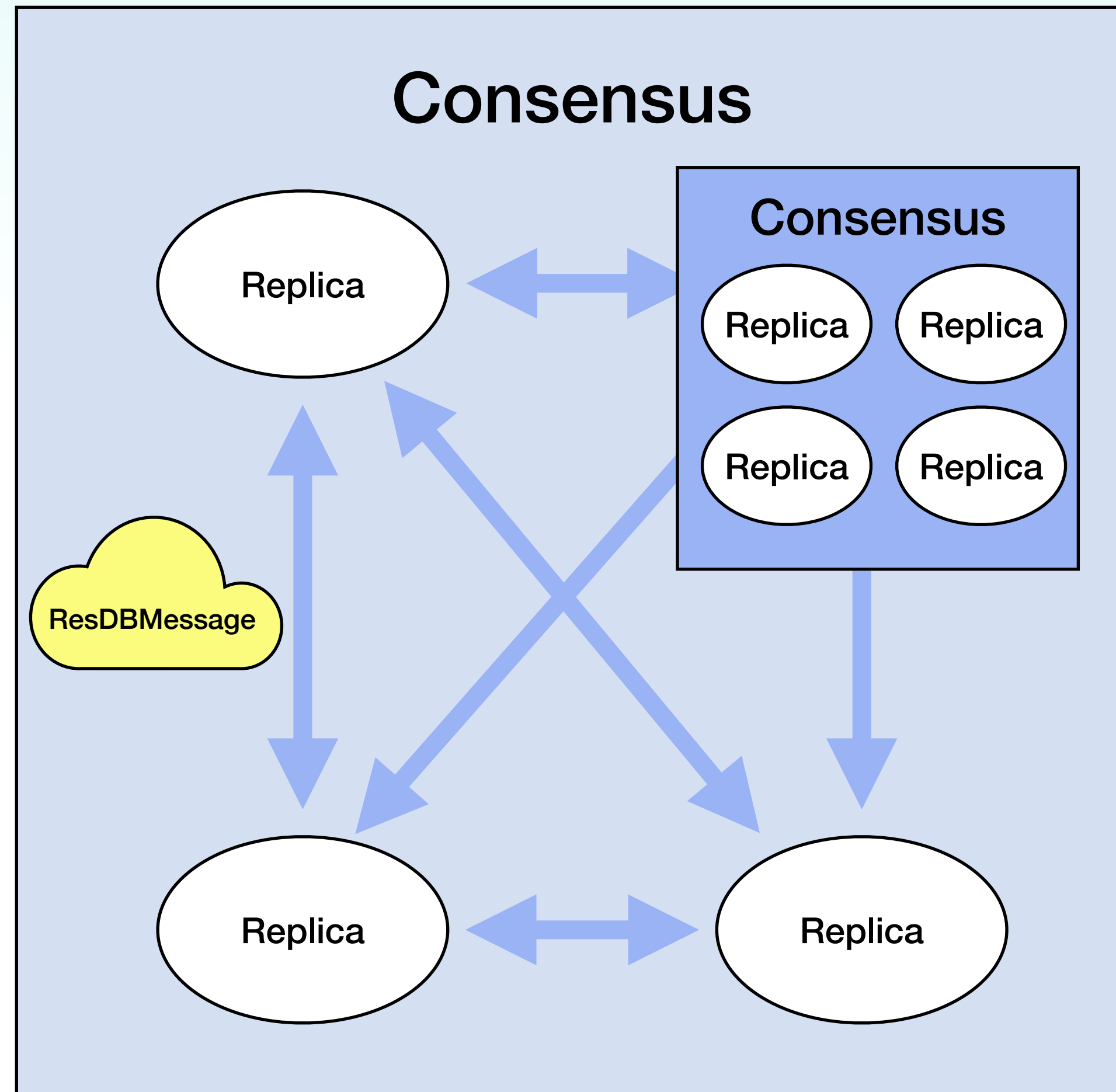
Construct and Send a ResDBTransaction to the Platform

1. **Applications** submit **client transactions** to SDK;
2. **SDK** transforms the client transactions into **ResDBTransaction** objects;
3. Sends the ResDBTransaction to **Proxy** by invoking the **RDBC API**;
4. The **ResDBTransaction** is delivered from the client to the **Proxy** via the **Network Substrate**;
5. The Proxy packs the ResDBTransaction into **ResDBMessage** and forwards it to **Replicas**



ResilientDB Transaction Workflow

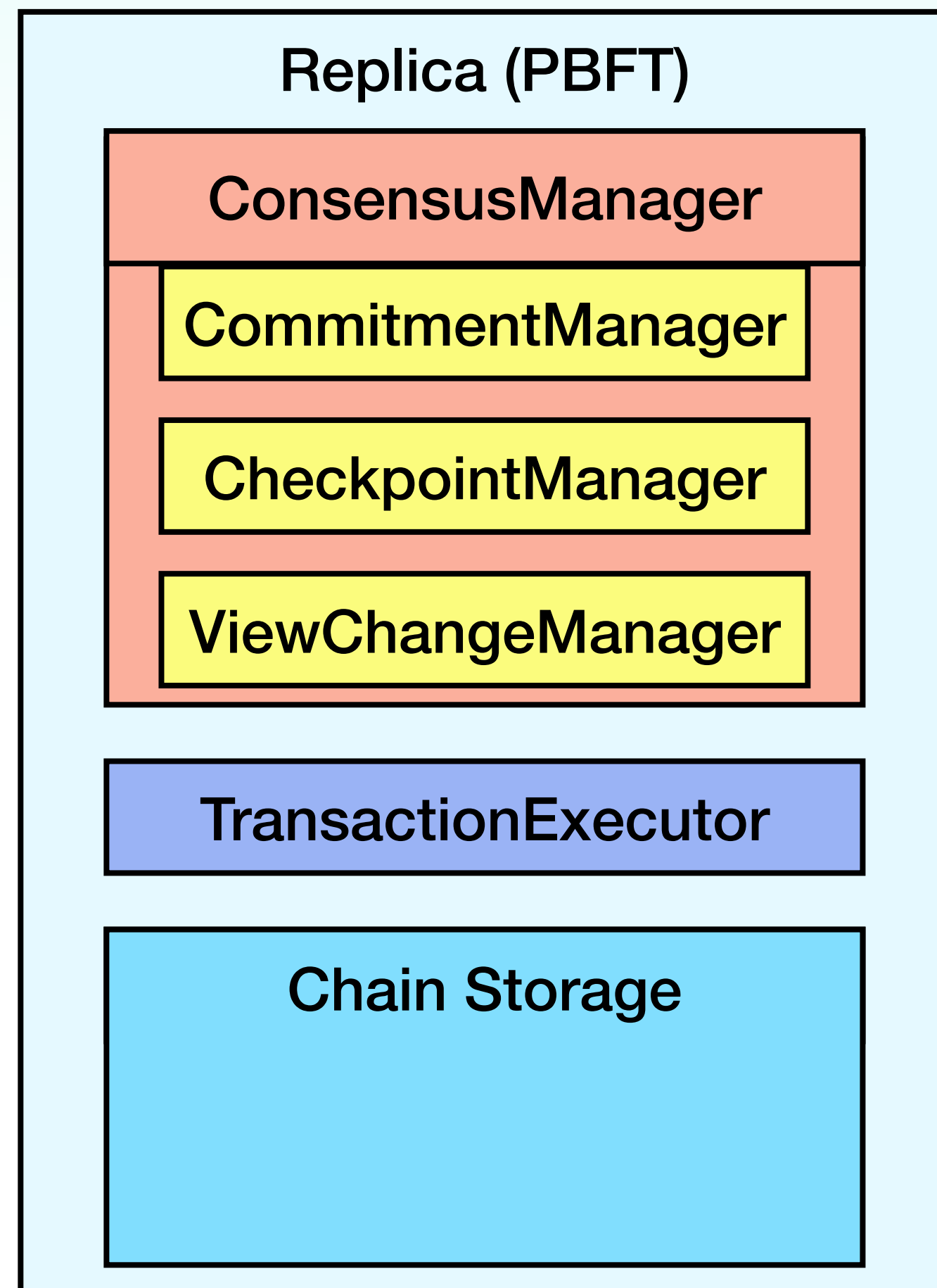
Reach Agreement on the ResDBTransaction



1. Applications send **Operation and Data** to SDK;
2. **SDK** transforms the Operation and Data into a **ResDBTransaction** object;
3. Sends the ResDBTransaction to **Proxy** by **Calling the RDBC API**;
4. The **ResDBTransaction** is delivered from the User to the **Proxy** via the **Network Substrate**;
5. The Proxy packs the ResDBTransaction into **ResDBMessage** and forwards it to **Replicas**
6. Replicas exchange **consensus messages** with each other via the Network Substrate.

ResilientDB Transaction Workflow

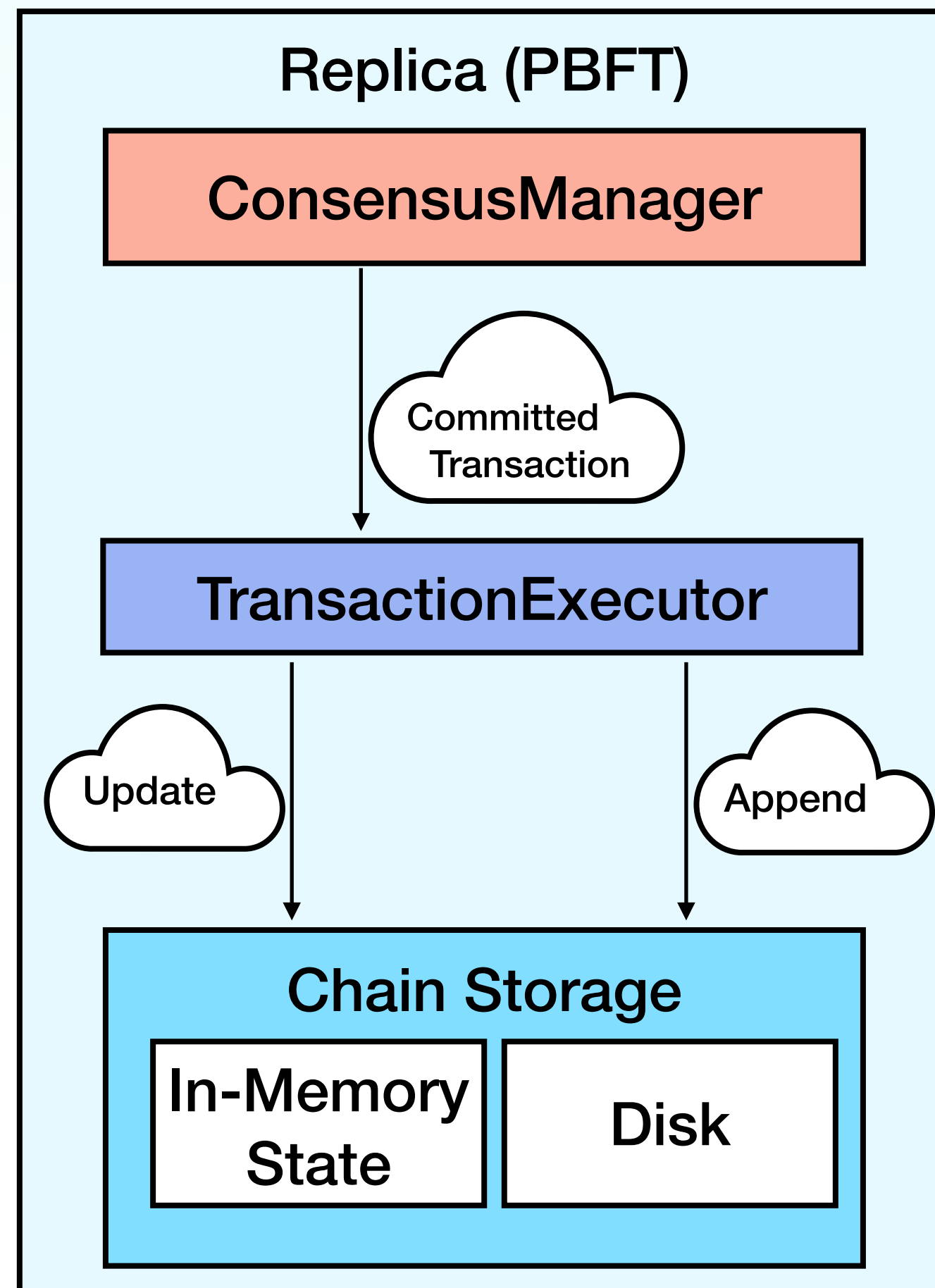
Internal Structure of a PBFT replica



- **ConsensusManager:** Reaching Consensus on the order of Transactions
 - **CommitmentManager**
 - **CheckpointManager**
 - **ViewChangeManager**
- **TransactionExecutor:** Execute the committed transactions
- **Chain Storage:** In-memory and on disk

ResilientDB Transaction Workflow

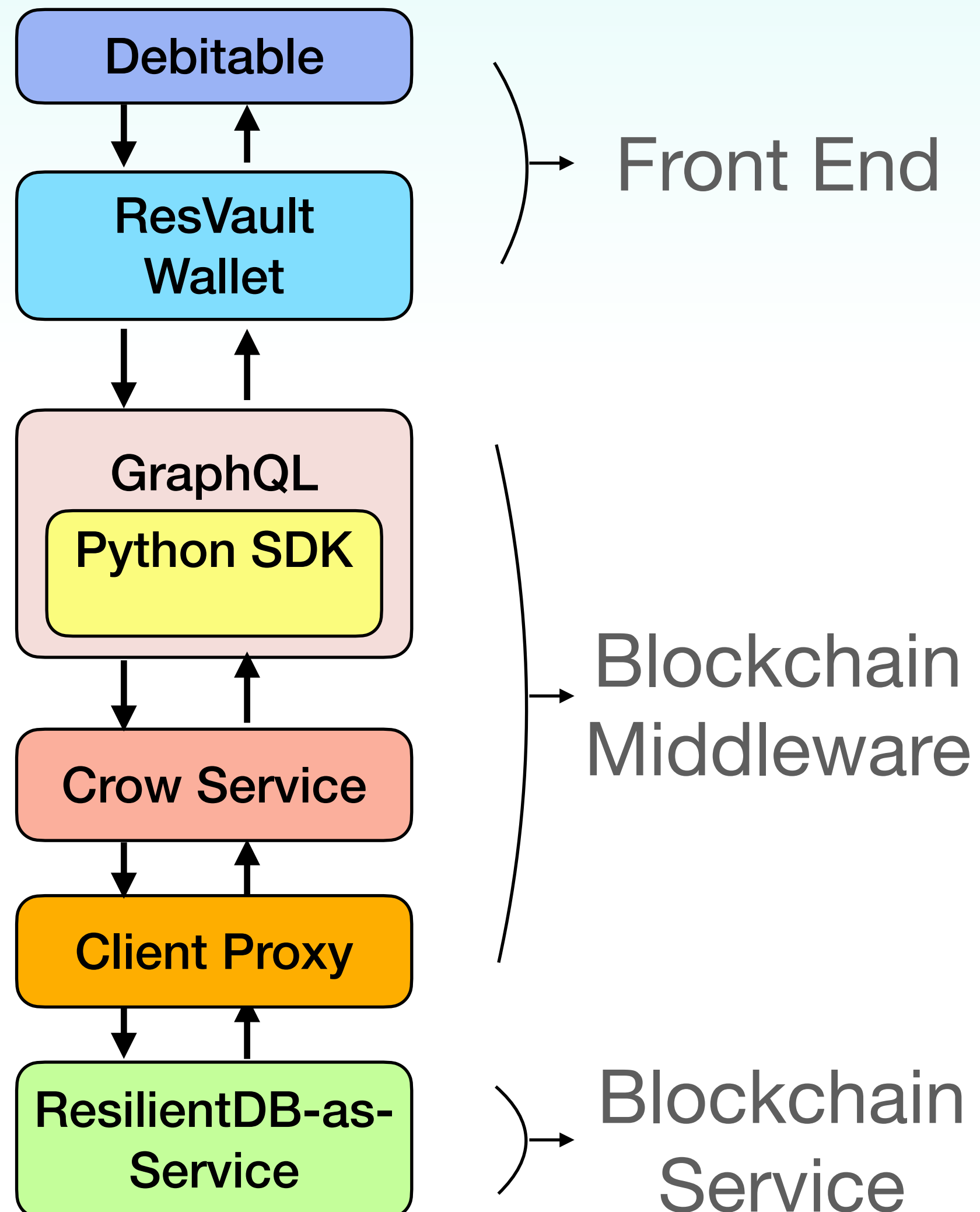
Internal Structure of a PBFT replica



- **ConsensusManager**: Reaching Consensus on the order of Transactions
 - **CommitmentManager**
 - **CheckpointManage**
 - **ViewChangeManager**
- **TransactionExecutor**: Execute the committed transactions
- **Chain Storage**: In-memory and on disk
 - Committed transactions are sent to **TransactionExecutor**
 - Update the **In-Memory State** based on transaction content
 - Append the transaction to ledger stored on **Disk**

Building DApp in ResilientDB

Debitable: An Example DApp Built on Top of ResilientDB



1. Deploy a ResilientDB Blockchain Service
2. Deploy Client Proxy to batch client transactions
3. Start Crow HTTP Service which provides HTTP Interface to ResilientDB Service
4. Using Python SDK to send HTTP requests, storing and retrieving data
5. Deploy a GraphQL Server that wraps the Python SDK, providing more efficient data retrieval and flexible requests
6. Build and install ResVault Wallet which generates and stores tokens securely on the chain
7. Enable DApp development by utilizing ResVault for token management

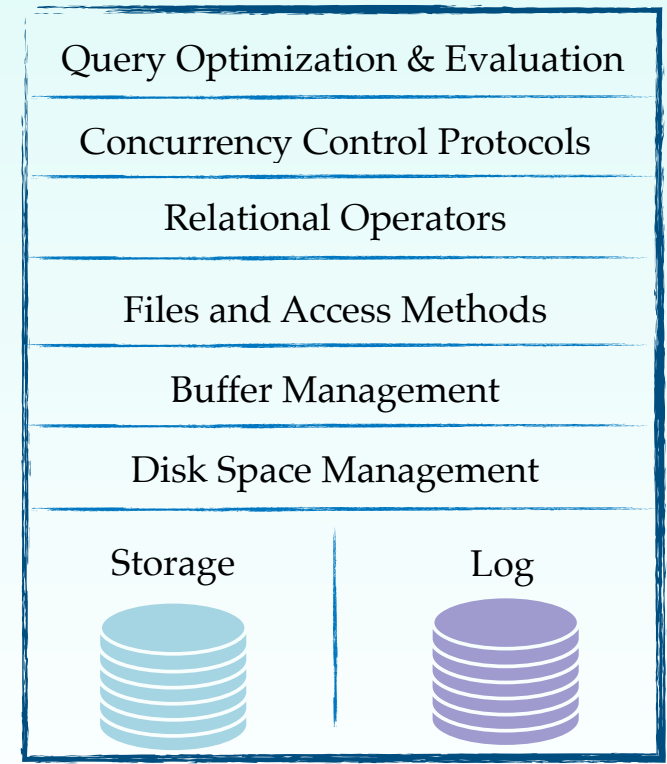
Layer 1 (e.g., Proof-of-Work)

Layer 2 (e.g., PBFT, Po*)

Power-of-Collaboration [IEEE Data Eng. Bulletin'22]

Chain Management (off-chain, on-chain)

Database Stack



ML Analytics (Read-only)

Permissioned Blockchain Through the Looking Glass [ICDCS'20]

Delayed Replication [ICDT'20]

Waif-free BFT [DISC'19]

Resilient Concurrent Consensus [ICDE'21]

RingBFT [EDBT'22]

Cerberus [JSys'23]

Cluster Sending Primitive [DISC'19]

Probabilistic Communication [JSys'23]

SGX-Accelerated Consensus [EuroSys'23]

Resilient Replication / Consensus

Sharding (Isolation Semantics, Consistency Levels)

Cross-chain Network

Global Distribution

Trusted Component

Recovery (View-change)

Bedrock of BFT [NSDI'24]

Proof-of-Execution [EDBT'21]

Chemistry of BFT [CIDR'23]

ByShard [VLDB'21]

ByShard [VLDBJ'23]

GeoBFT [VLDB'20]

SpotLess [ICDE'24]

Identity Management

Permissioned

Permissionless

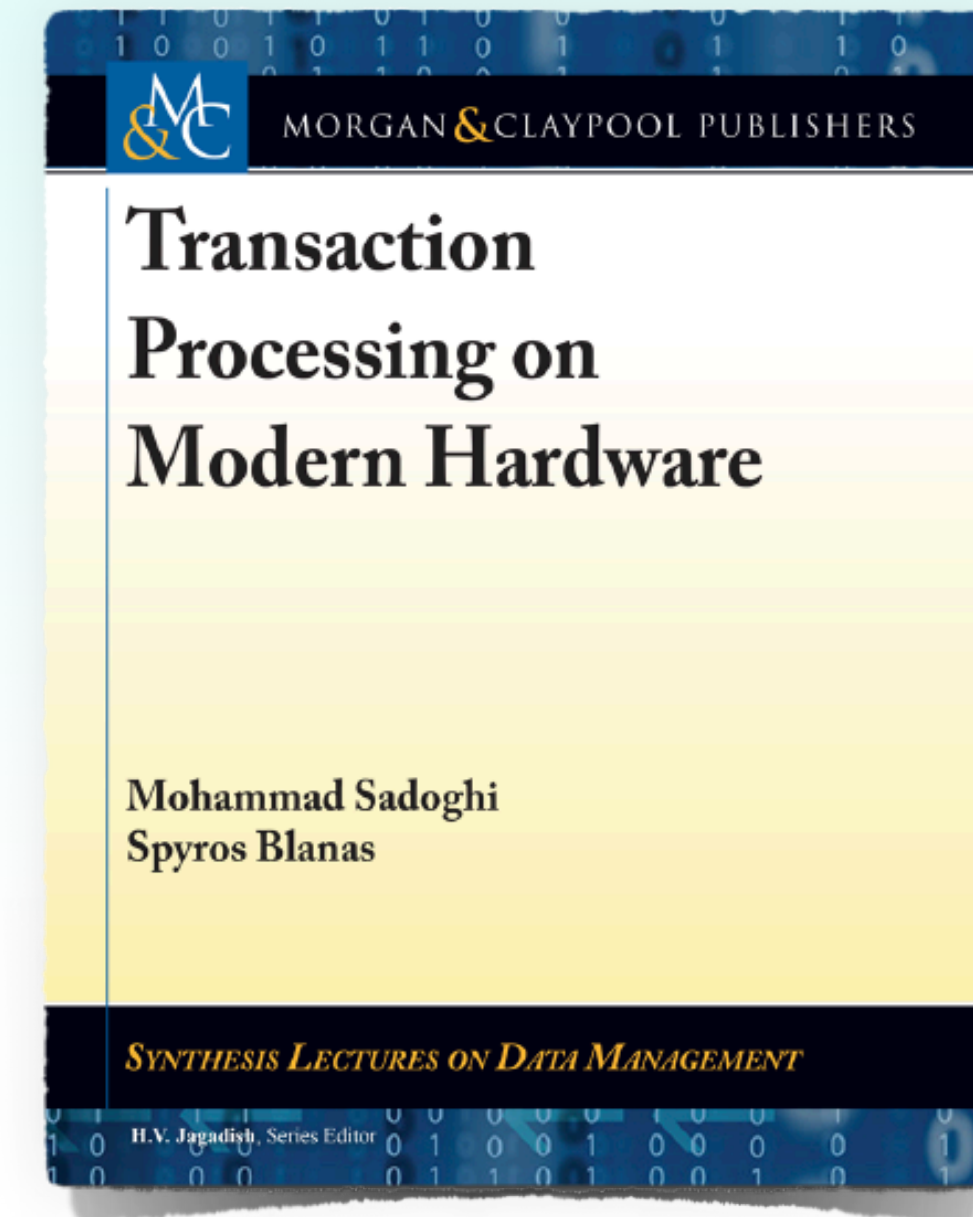
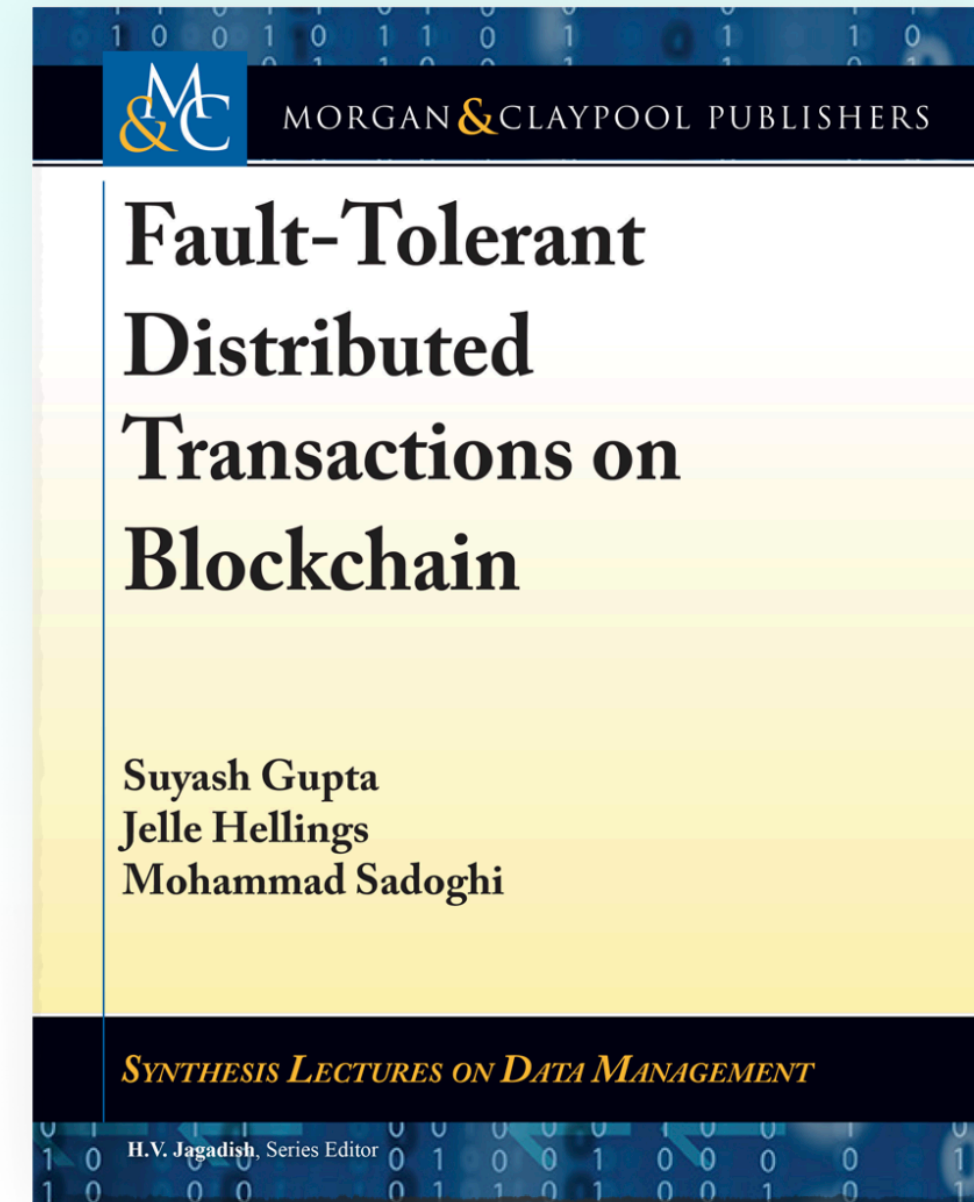
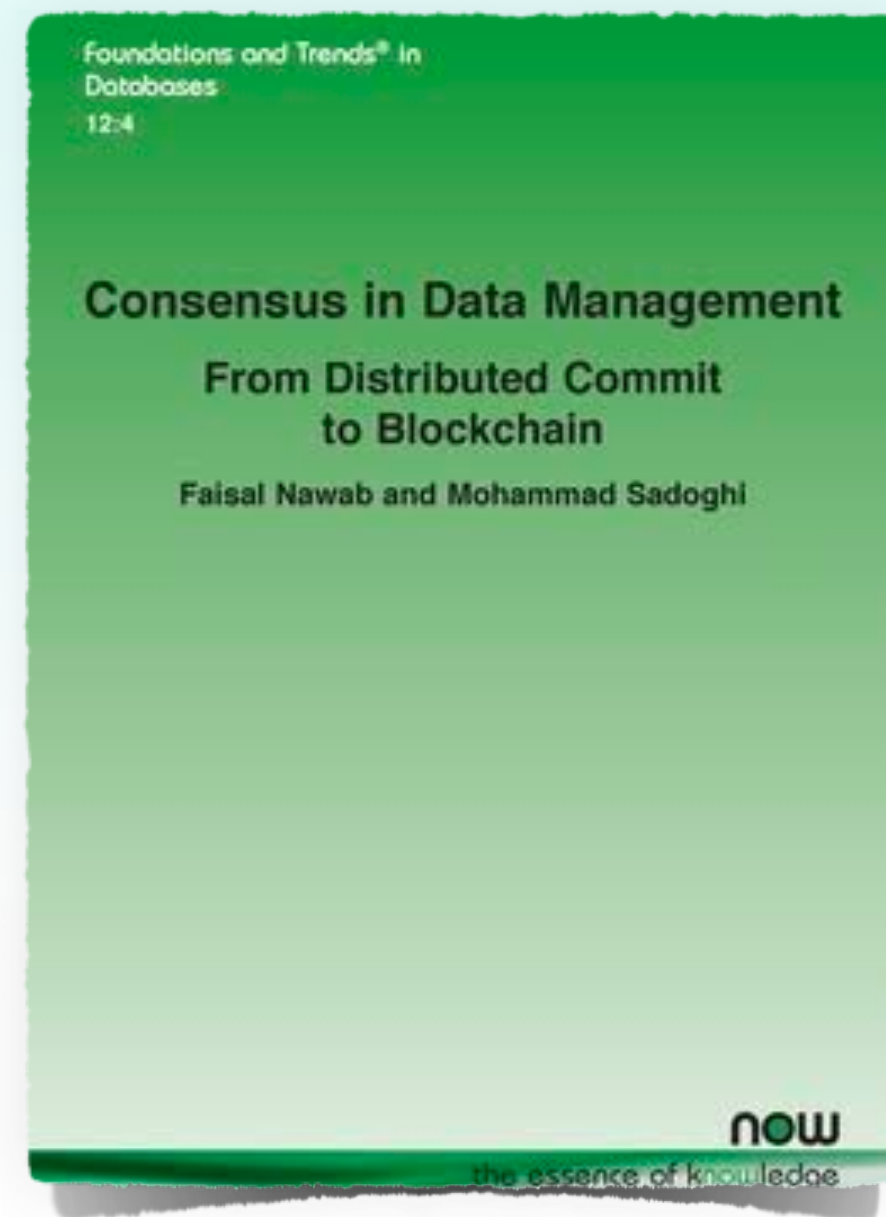
ServerlessBFT [ICDE'23]

GDPR Compliance [EDBT'24]

Applications: DeFi, Smart Contracts, IoT, Serverless

Blockplane [ICDE'19]





Books

Consensus in Data Management: From Distributed Commit to Blockchain.
Foundations and Trends® in Databases. Now Publisher. 2023

Fault-Tolerant Distributed Transactions on Blockchain.
Synthesis Lectures on Data Management, Morgan & Claypool / Springer Publisher. 2021

Transaction Processing on Modern Hardware.
Synthesis Lectures on Data Management, Morgan & Claypool / Springer Publishers. 2019



THANK YOU



<https://resilientdb.incubator.apache.org>



PUBLICATIONS:

