

Digital wallet

Save money and send it later. Transfer between wallets faster.

EDS: balance transfer operations / 1 bill TPS / Correctness / Reproducibility /
availability 99.99% / Reliability / Transactions ↗ reconstruct historical balance

1 DB support 1,000 TPS, need 2 mill TPS (deducting / depositing money)
→ 2,000 DB nodes Relational DB.

API. POST /v1/wallet/balance-transfer

Req:

from_acc stz
to_acc stz
amount stz
currency stz
transaction_id stz

Resp:

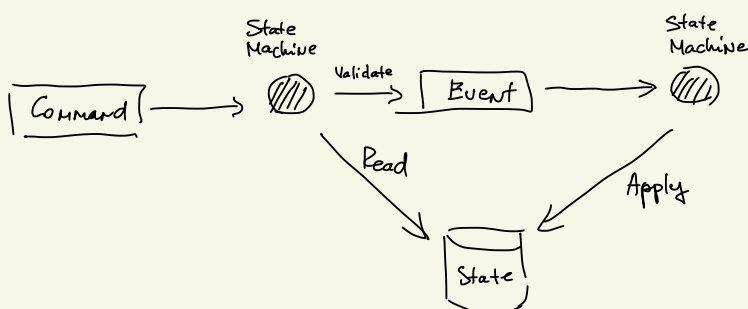
Status: success
Transact_id: XXXX

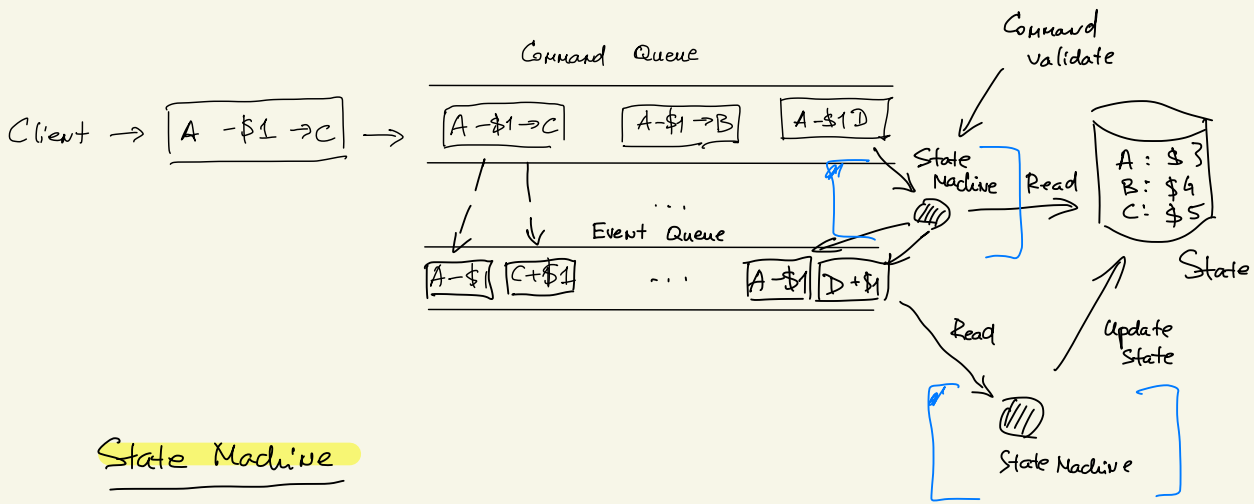
← to avoid potential risk of losing precision

(in-memory)
Redis not enough for storing 1 mill TPS, Cluster with evenly distributed.
account ID, hash Code () %, partition Number;
+ Zookeeper as HA configuration storage.
Wallet → stateless. Ready to scale horizontally.

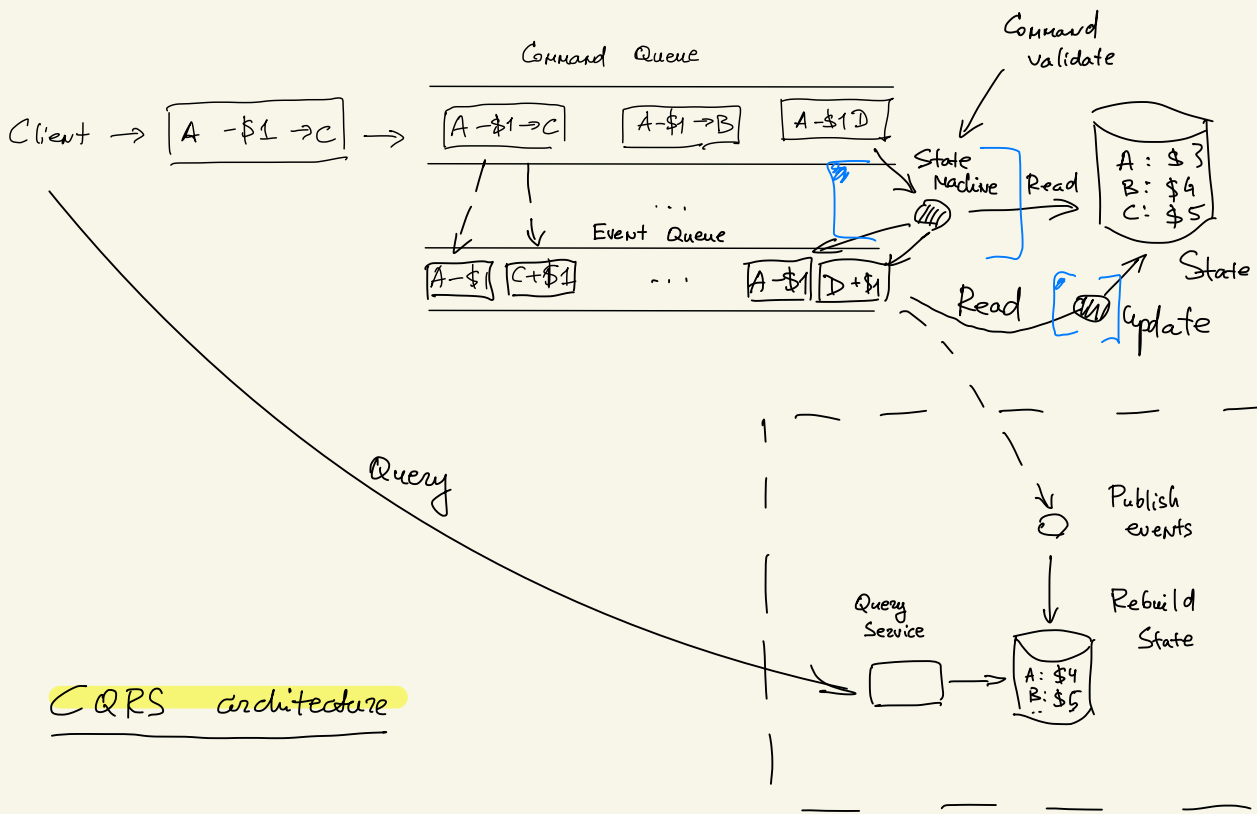
Distributed transactions: 1. 2PC
2. TC/C ("undo"), Parallel execution
3. Saga. Sequence. Linear execution
- Choreography (ASYNC). Maintain "State Machine"
- Orchestration

Event sourcing:





State Machine



CQRS architecture

DDD: SQLite - local
 Rock DB - file-based local } RDBMS
 \rightarrow log-structured merge-tree (LSM)
 write +
 • to improve read \rightarrow data is cached

File-based data approach allow to make snapshots.
 giant binary file \rightarrow HDFS

high reliability \rightarrow replicate event list across multiple nodes.

Raft algorithm \rightarrow consensus-based replication

