U LISBOA | UNIVERSIDADE DE LISBOA

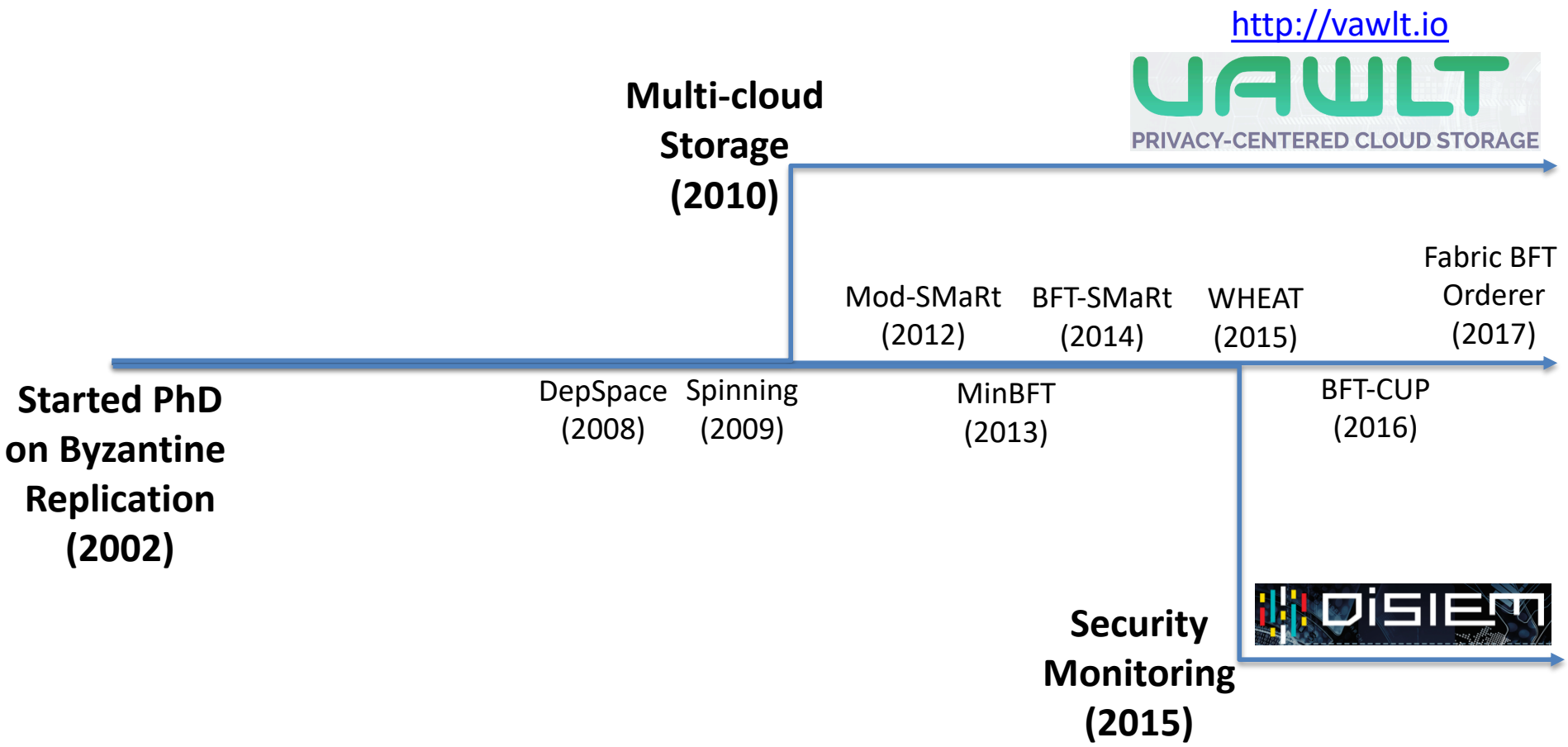
- Permissioned Ledgers -
Consensus is Only the Beginning

Alysson Bessani

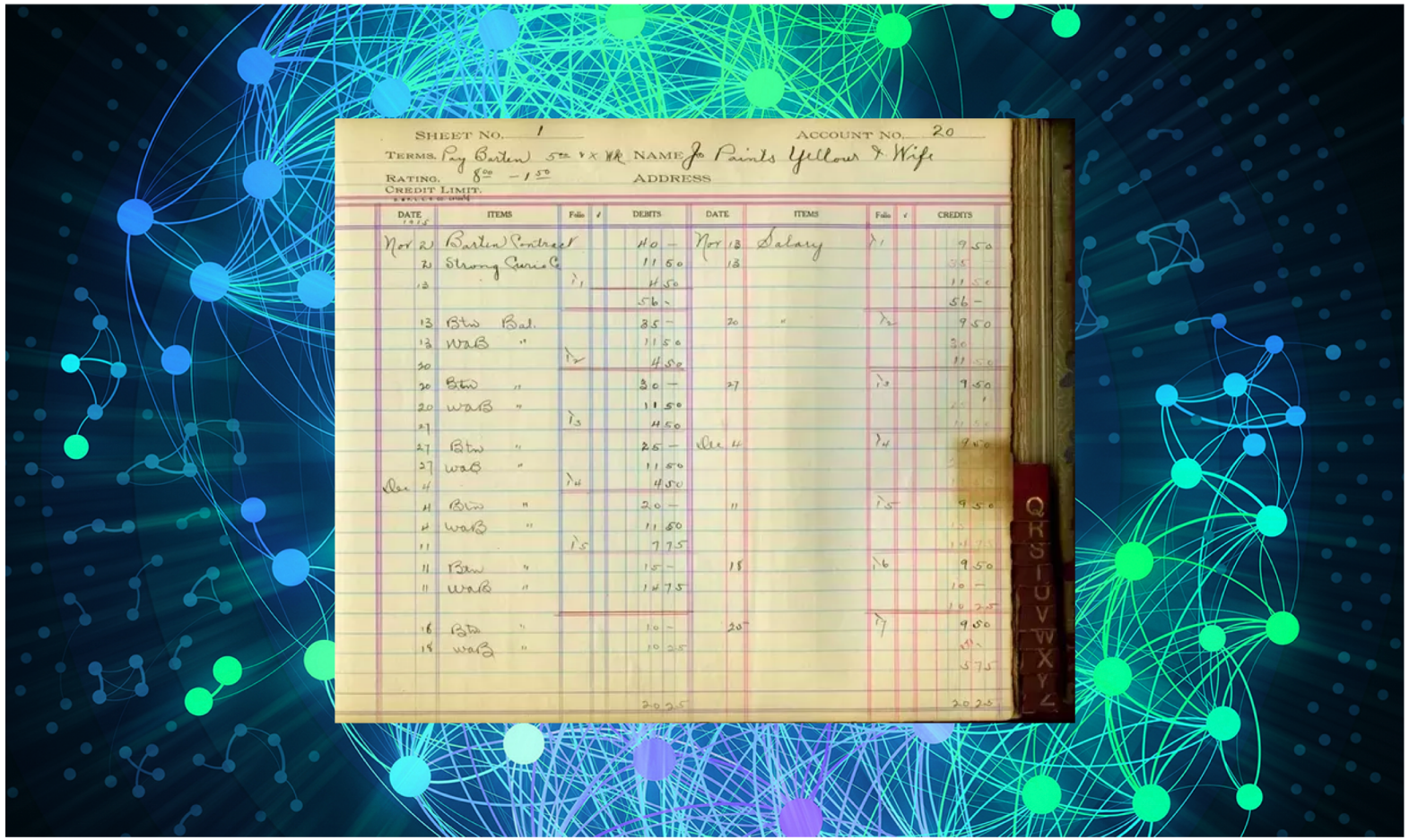
Ciências ULisboa

LASIGE

Who am I?


http://vawlt.io
VAWLT
PRIVACY-CENTERED CLOUD STORAGE
Multi-cloud Storage (2010)
Started PhD on Byzantine Replication (2002)
DepSpace (2008)
Spinning (2009)
Mod-SMaRt (2012)
MinBFT (2013)
BFT-SMaRt (2014)
WHEAT (2015)
BFT-CUP (2016)
Fabric BFT Orderer (2017)
Security Monitoring (2015)
DISIEM

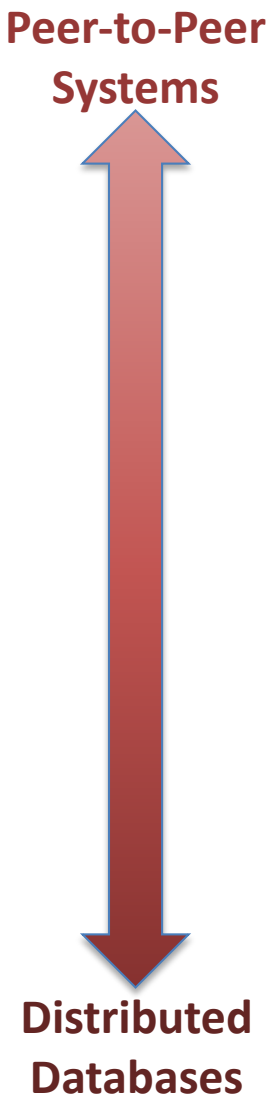
What is a Blockchain?



General ledger on top of a peer-to-peer network

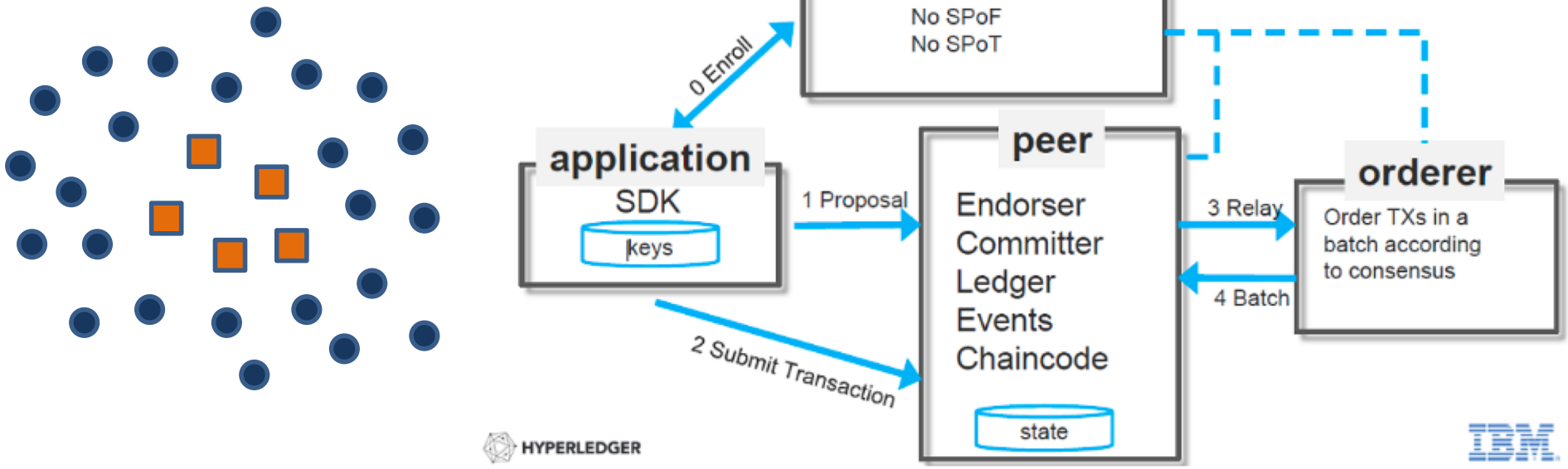
Blockchain Models

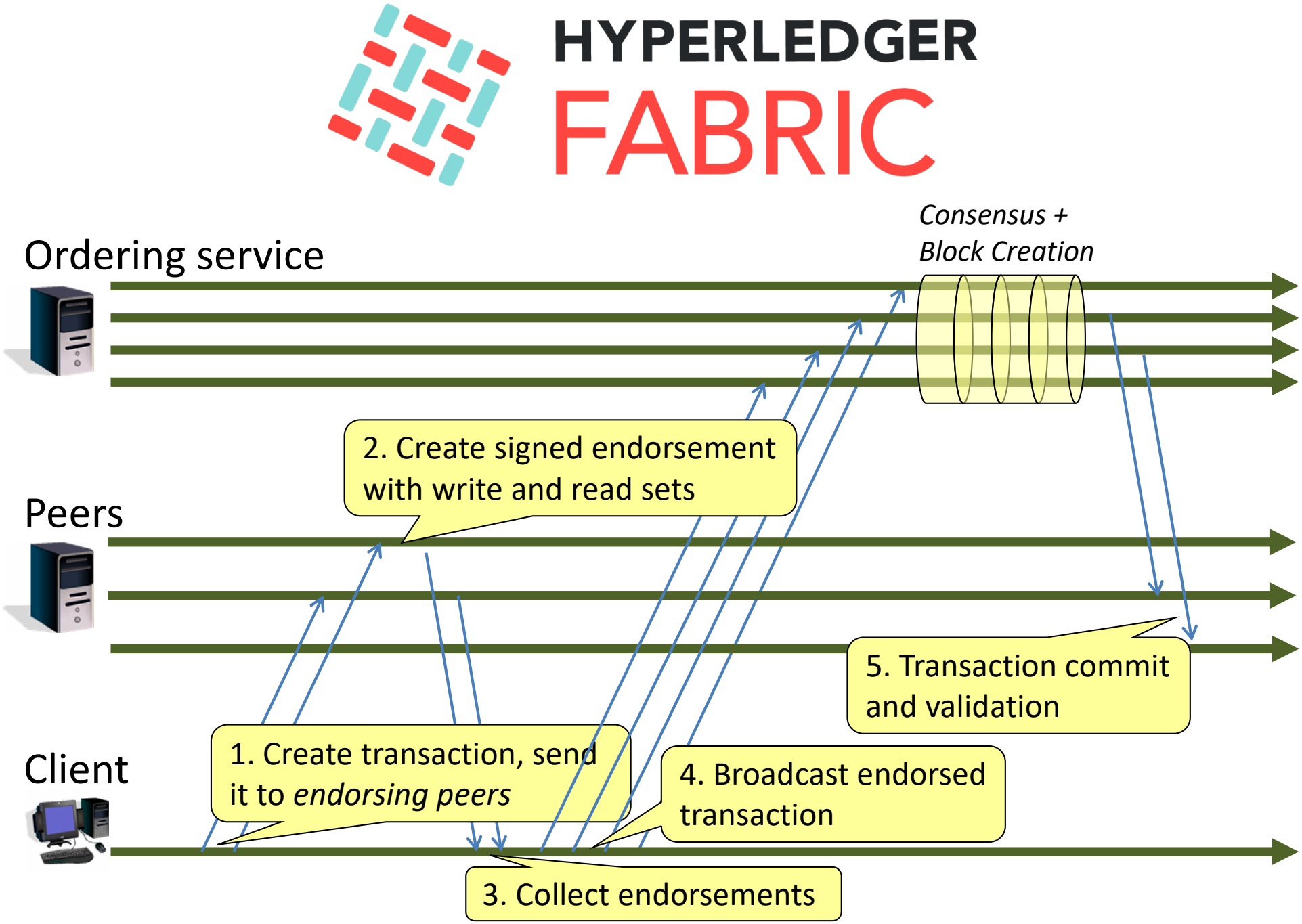
- **Public (Open) Ledgers**
  - Like the ones used in **Bitcoin** and **Ethereum**
  - Peers don't need strong identities
  - PoW/PoS/… consensus
- **Permissioned (Private) Ledgers**
  - a.k.a. **Distributed Ledger Technology (DLT)**
  - Peers have strong/verifiable identities
  - Classical Byzantine consensus



HYPERLEDGER FABRIC

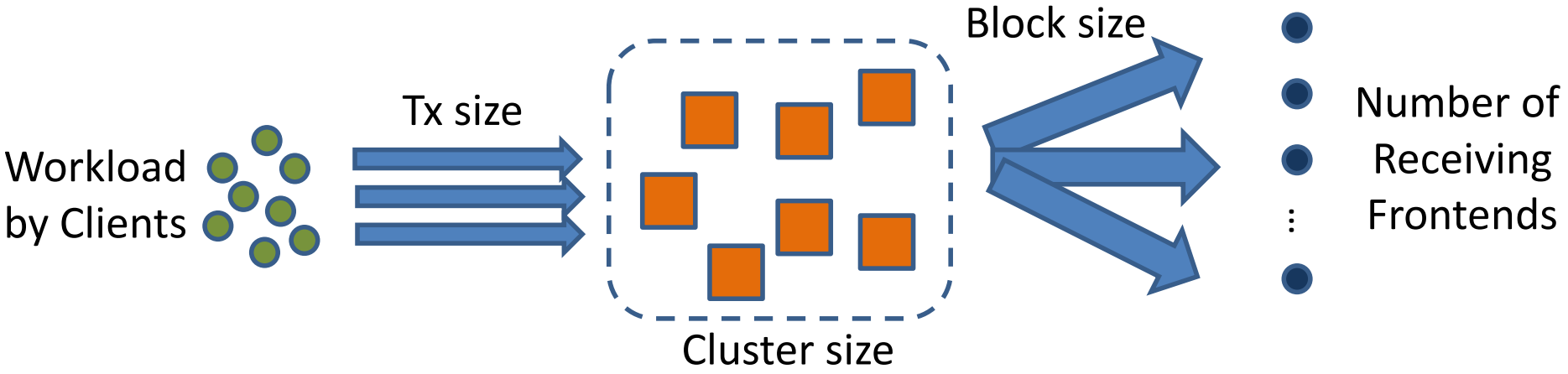
- Open-source, modular, permissioned
- Architecture: not all “peers” are equal




HYPERLEDGER
FABRIC
Ordering service
Consensus +
Block Creation
Peers
2. Create signed endorsement with write and read sets
5. Transaction commit and validation
Client
1. Create transaction, send it to *endorsing peers*
4. Broadcast endorsed transaction
3. Collect endorsements

HYPERLEDGER
FABRIC

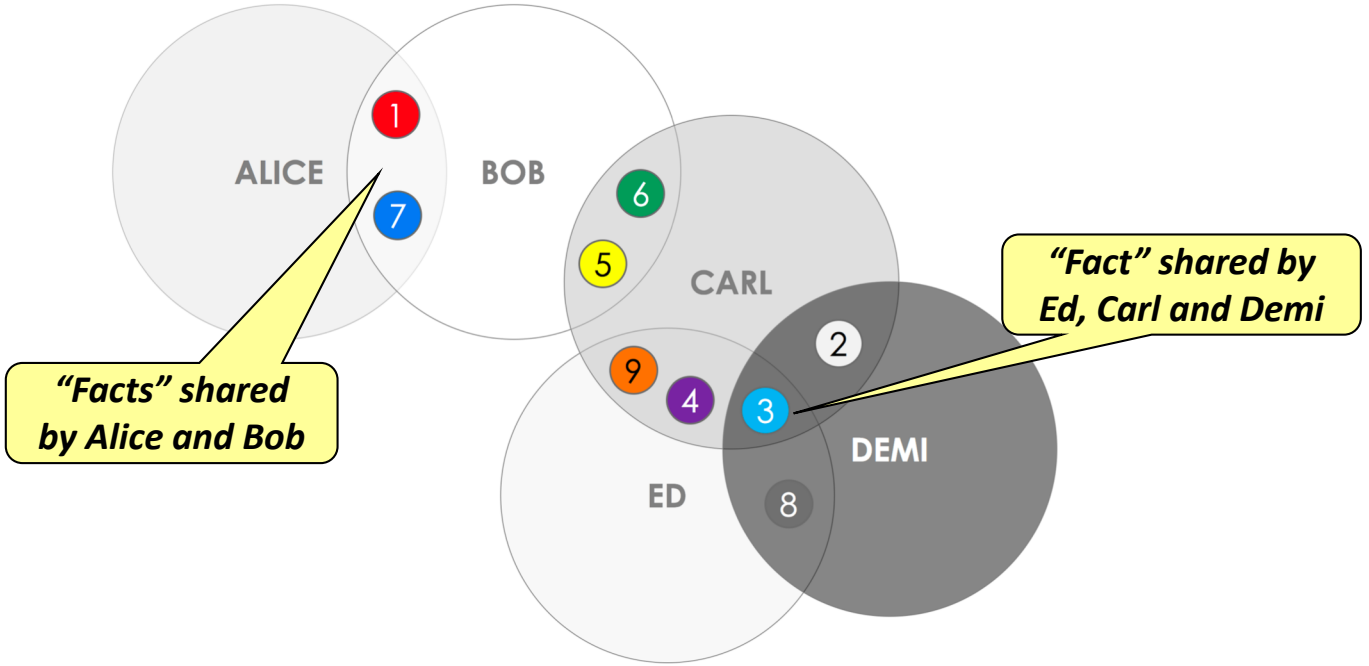
Ordering Cluster



- Ordering node state:
  - the ordered transactions not yet in a block,
  - header of the last generated block, and
  - latest configuration block

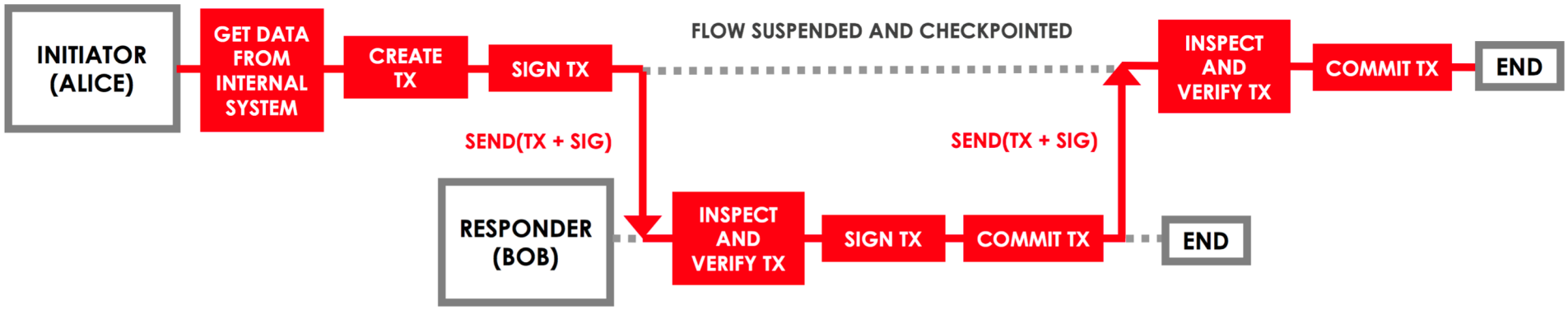
corda

- Open-source blockchain project targeting (at least initially) the financial market
- Key idea: **there is no shared global ledger**
  - Instead, **there are many distributed ledgers**



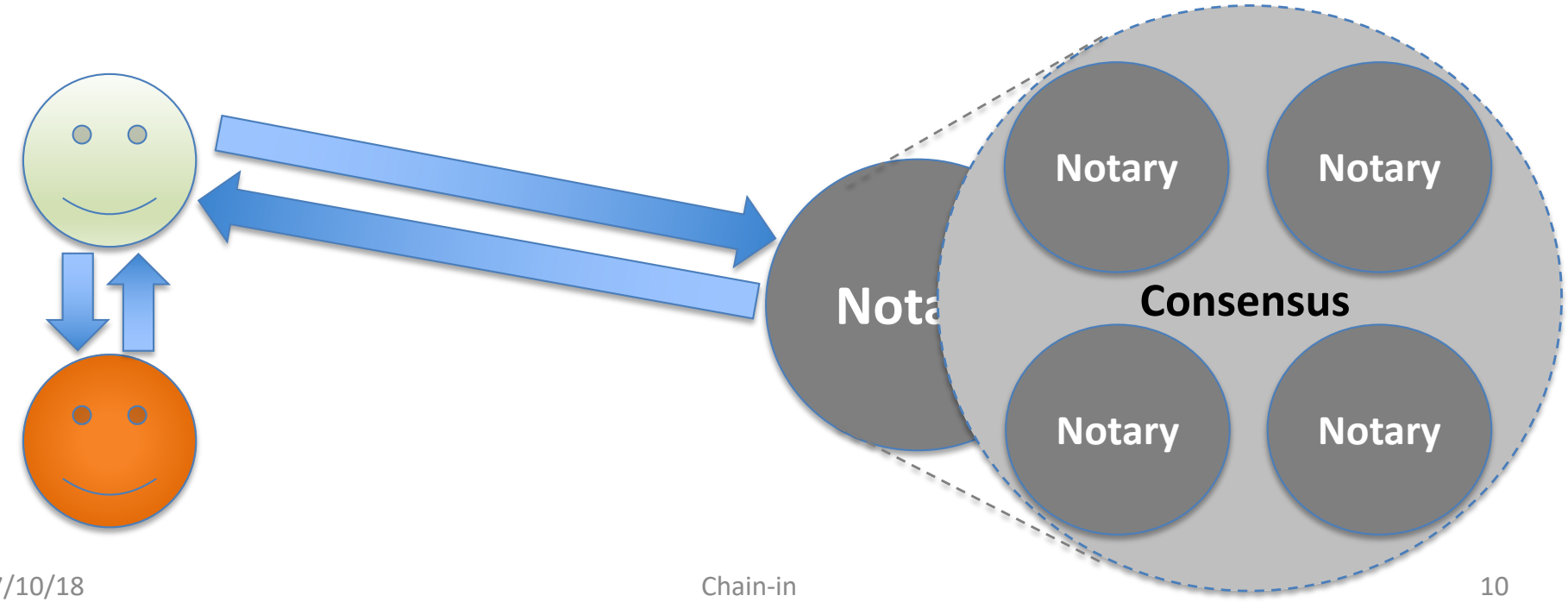
corda

- Only involved participants have to *execute* and *validate* the transaction
- A transaction is *committed* only if it achieve
  - **Validity consensus**: all involved participants need to validate and sign the transaction
  - **Uniqueness consensus**: requires a notary service

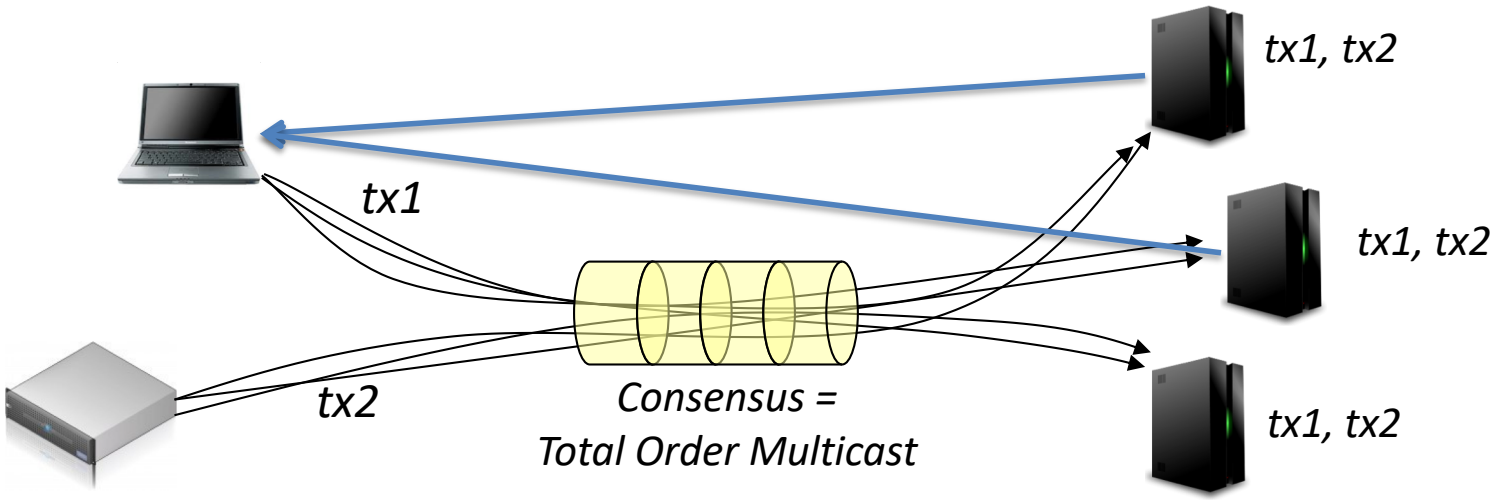


corda

- Notary implements an insert-only key-value store that register all state “consumptions”
- Some specific transaction validation might be executed
- Multiple notaries might be used



State Machine Replication



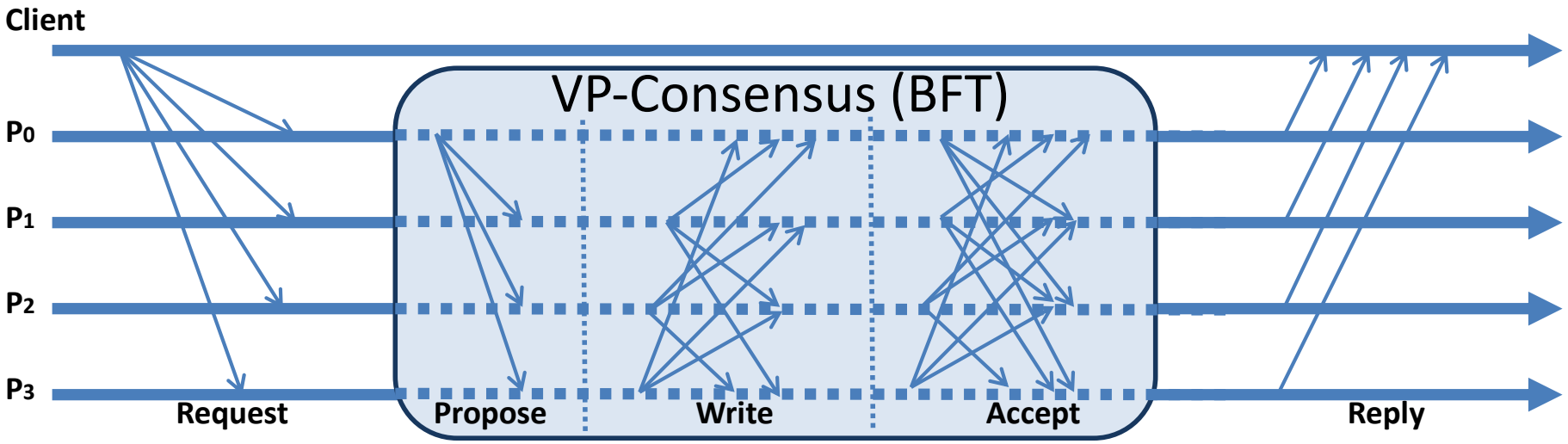
Safety: all replicas execute the same sequence of transactions

Liveness: transactions issued by correct clients are answered

BFT-SMaRt

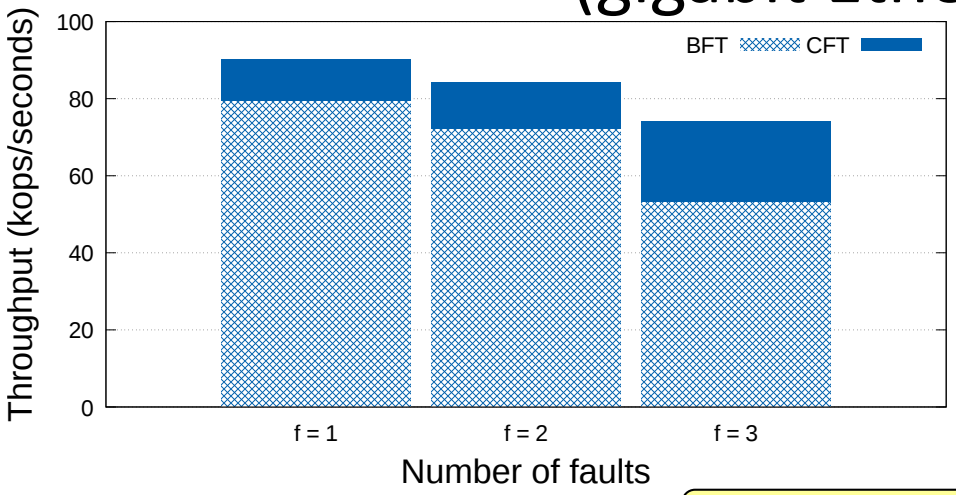
http://bft-smart.github.io/library/

- Byzantine Fault tolerant state machine replication library written in Java (under development since 2010)
- Tolerates either crash (*2f+1* replicas) or Byzantine faults (*3f+1* replicas)
- Available under Apache license

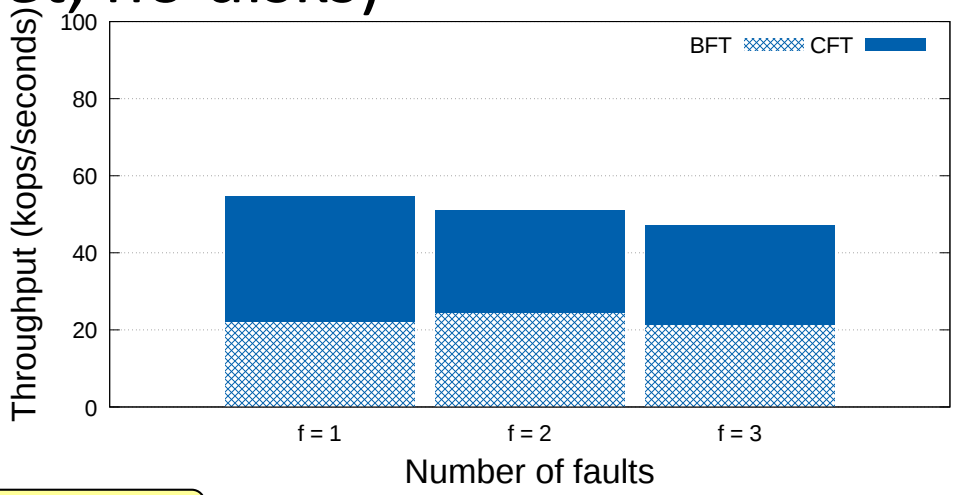


BFT-SMaRt Performance

(gigabit Ethernet, no disks)

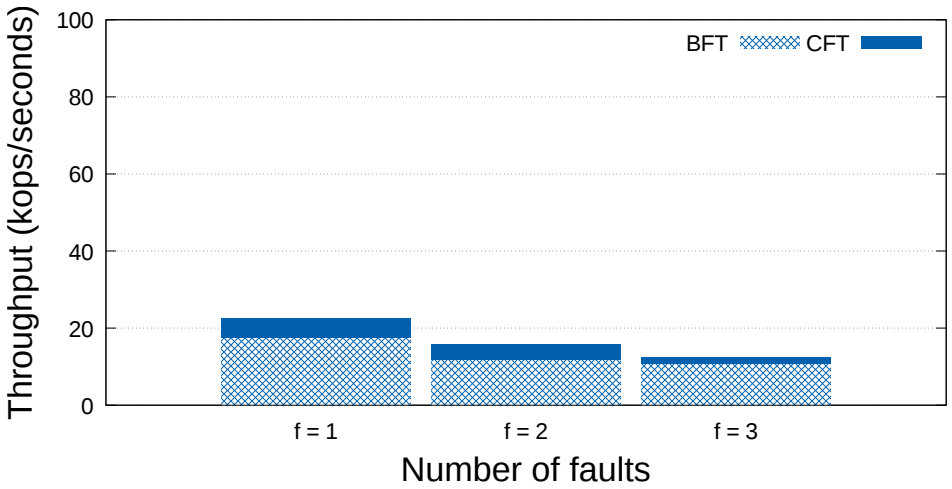


(a) 0/0

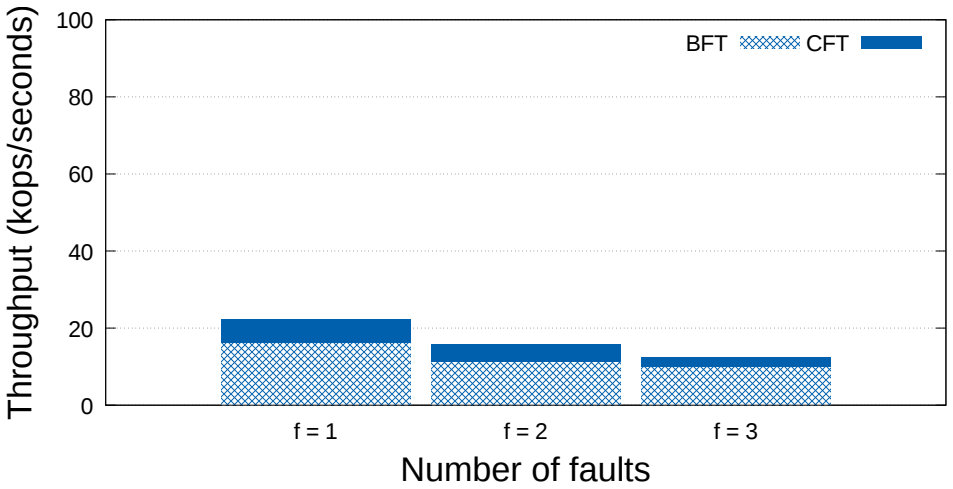


(b) 0/1024

<request size>/<reply size>



(c) 1024/0

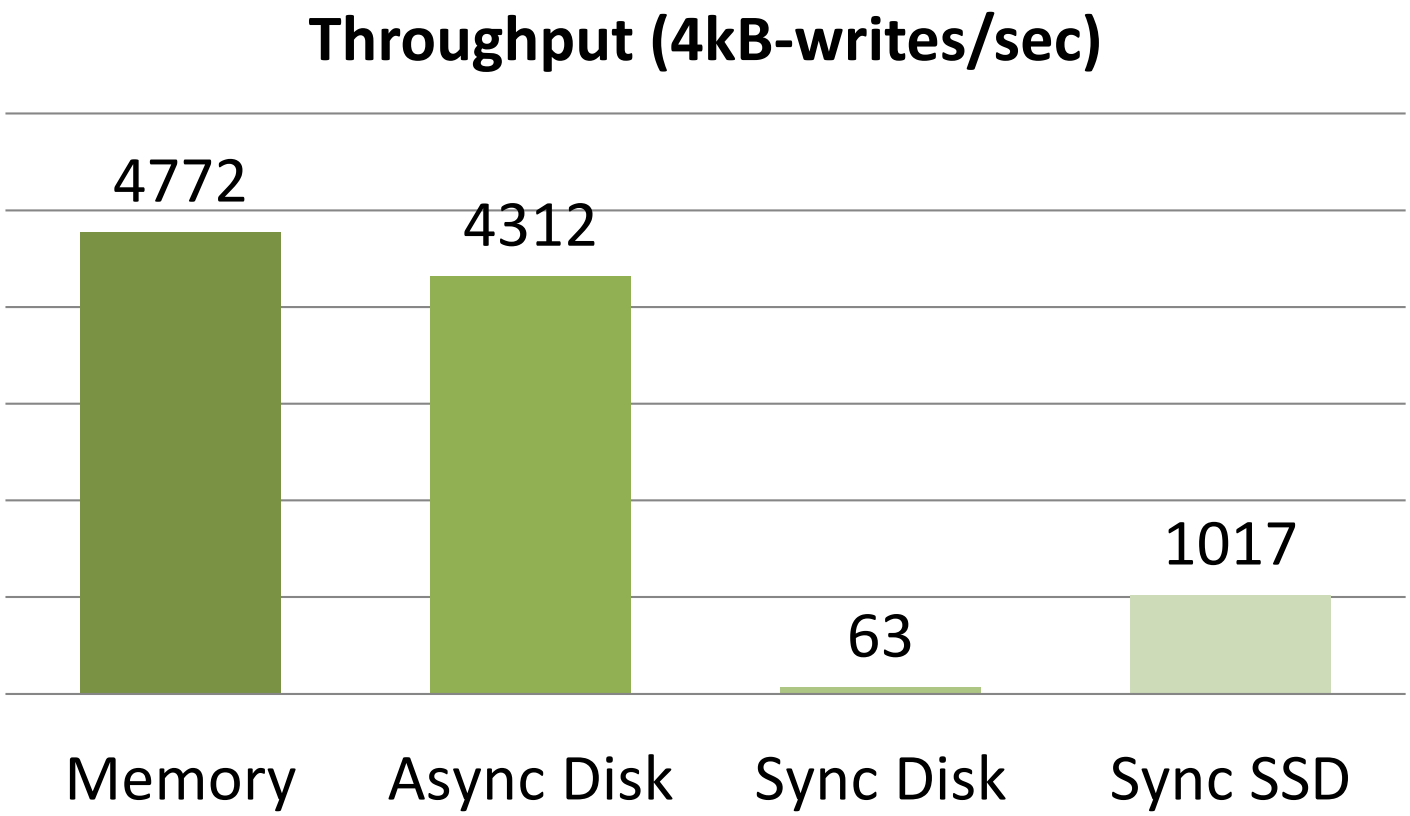


(d) 1024/1024

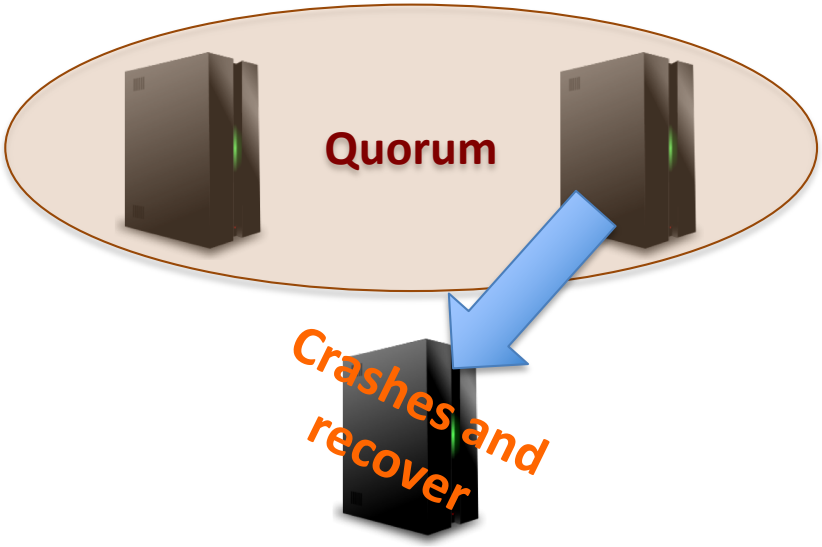
Consensus is not enough

- A consensus engine also needs:
  - **Durability**: transactions on stable storage
  - **Recovery**: recovered replicas need to be synched
  - **Reconfiguration**: replica group changes

Durability = Stable Logging

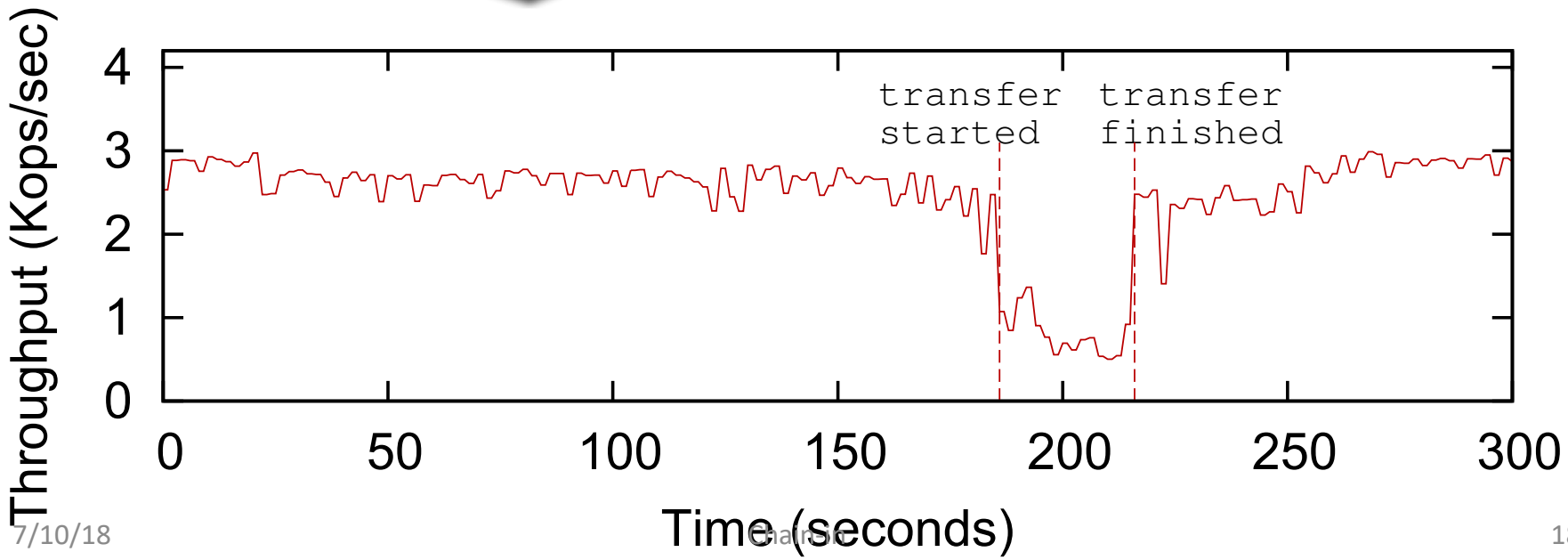


Recovery = State Transfer

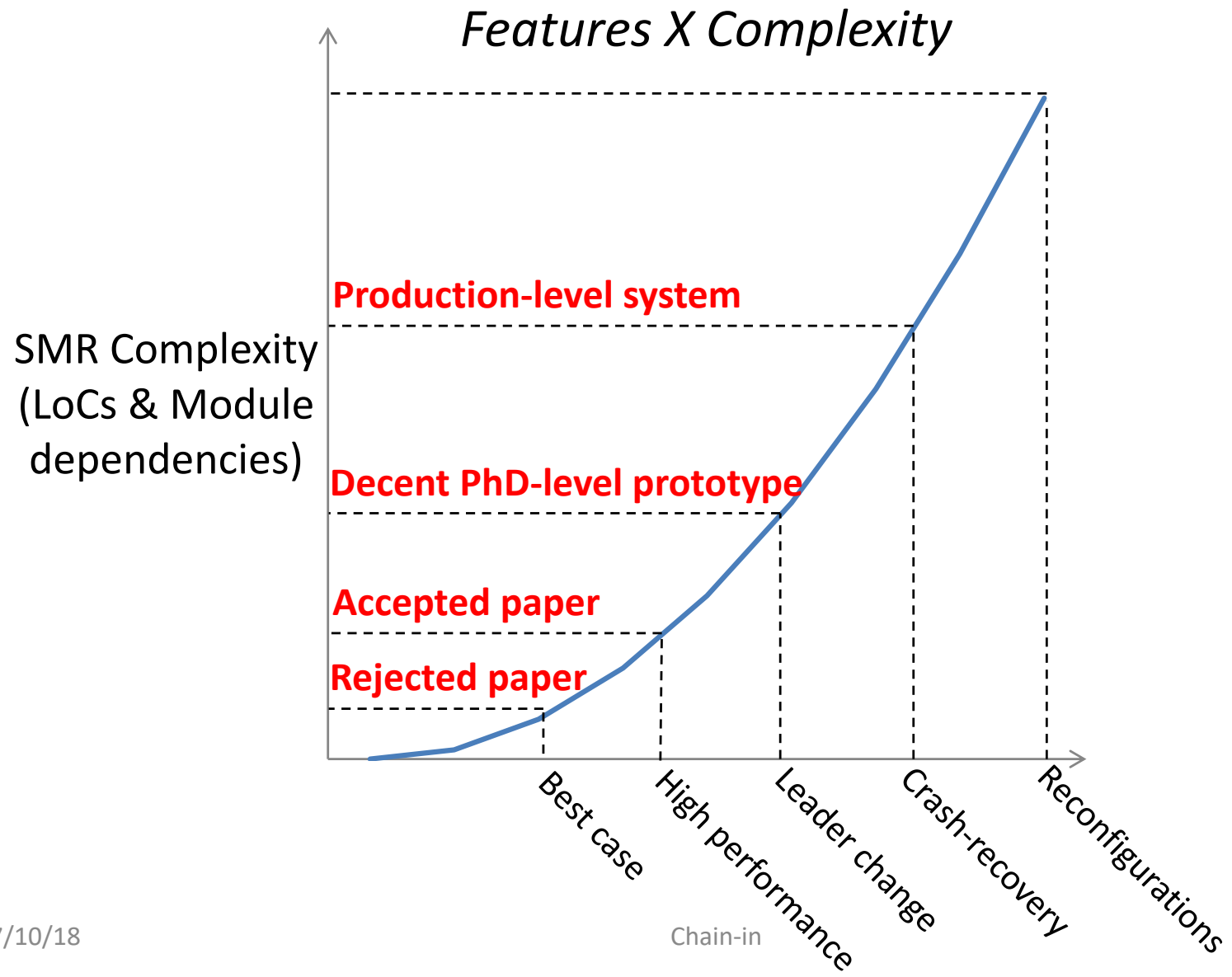


PROBLEM:

- Need 2 replicas to order requests
- 1 stopped and 1 transmitting state

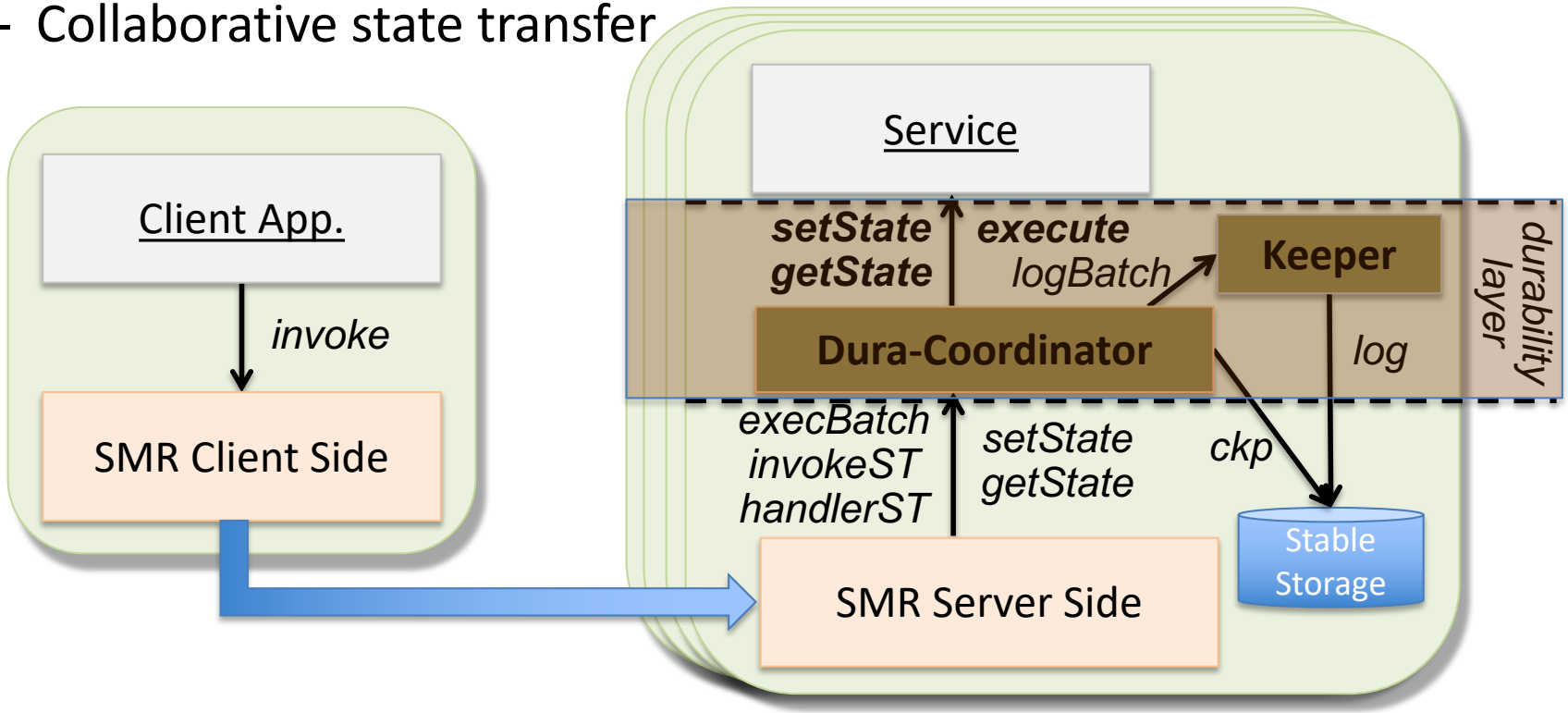


Reconfiguration = Complexity

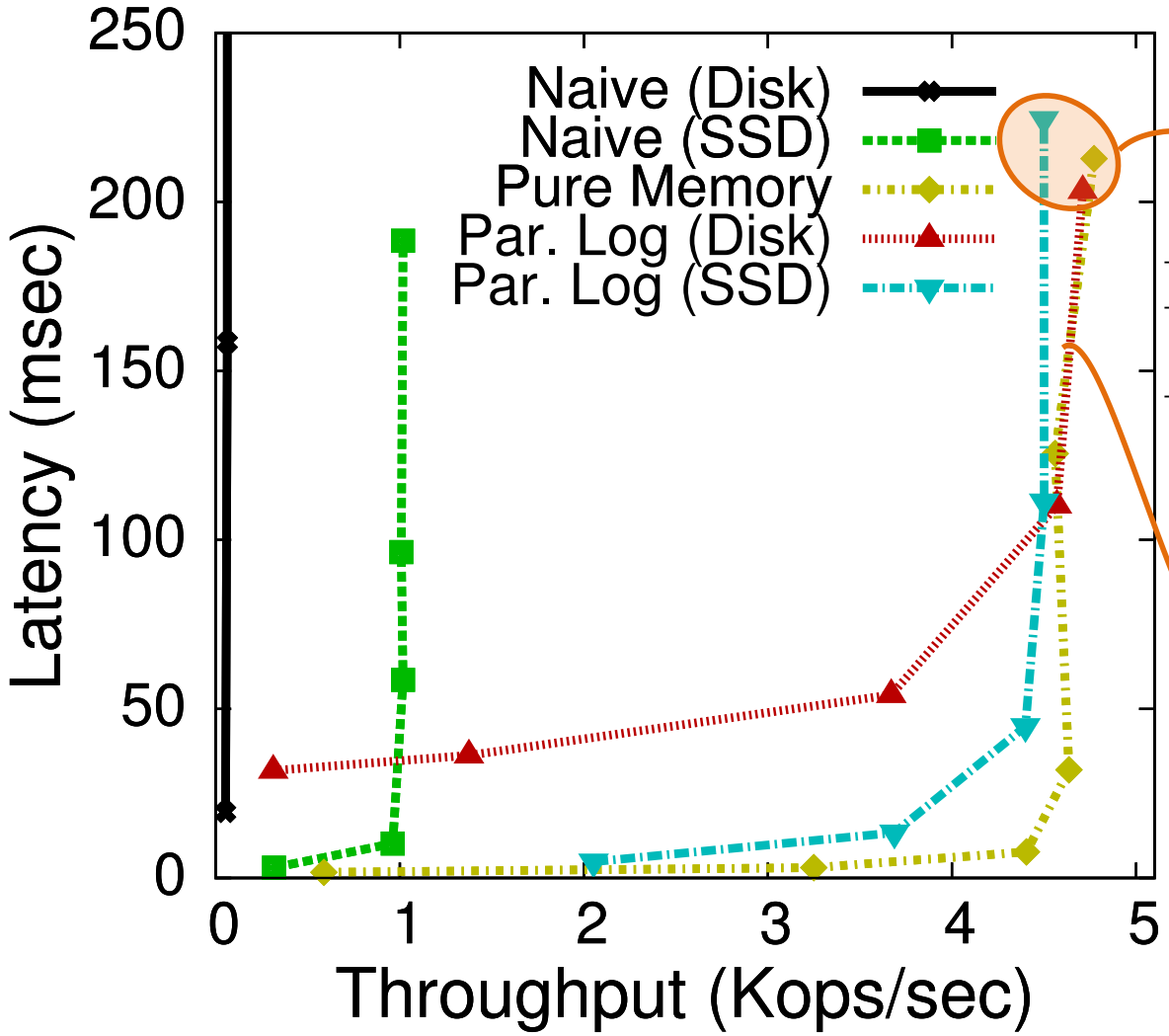


Durability in BFT-SMaRt

- Techniques for efficient durability
  - Parallel Logging
  - Sequential checkpoints
  - Collaborative state transfer



Durable 4kB-write Throughput

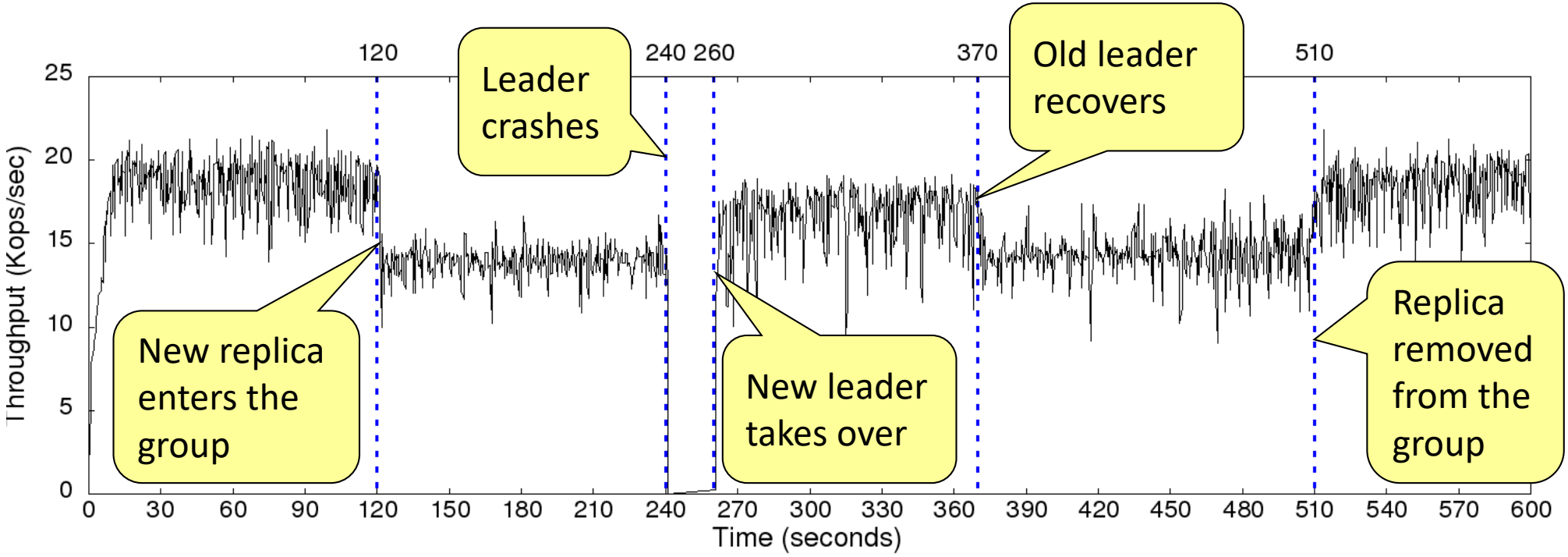


For large writes, our disks absorb more data than (our) SSDs

Record length	Disk	SSD
1MB	96.1 MB/s	128.3 MB/s
4MB	135.6 MB/s	130.7 MB/s

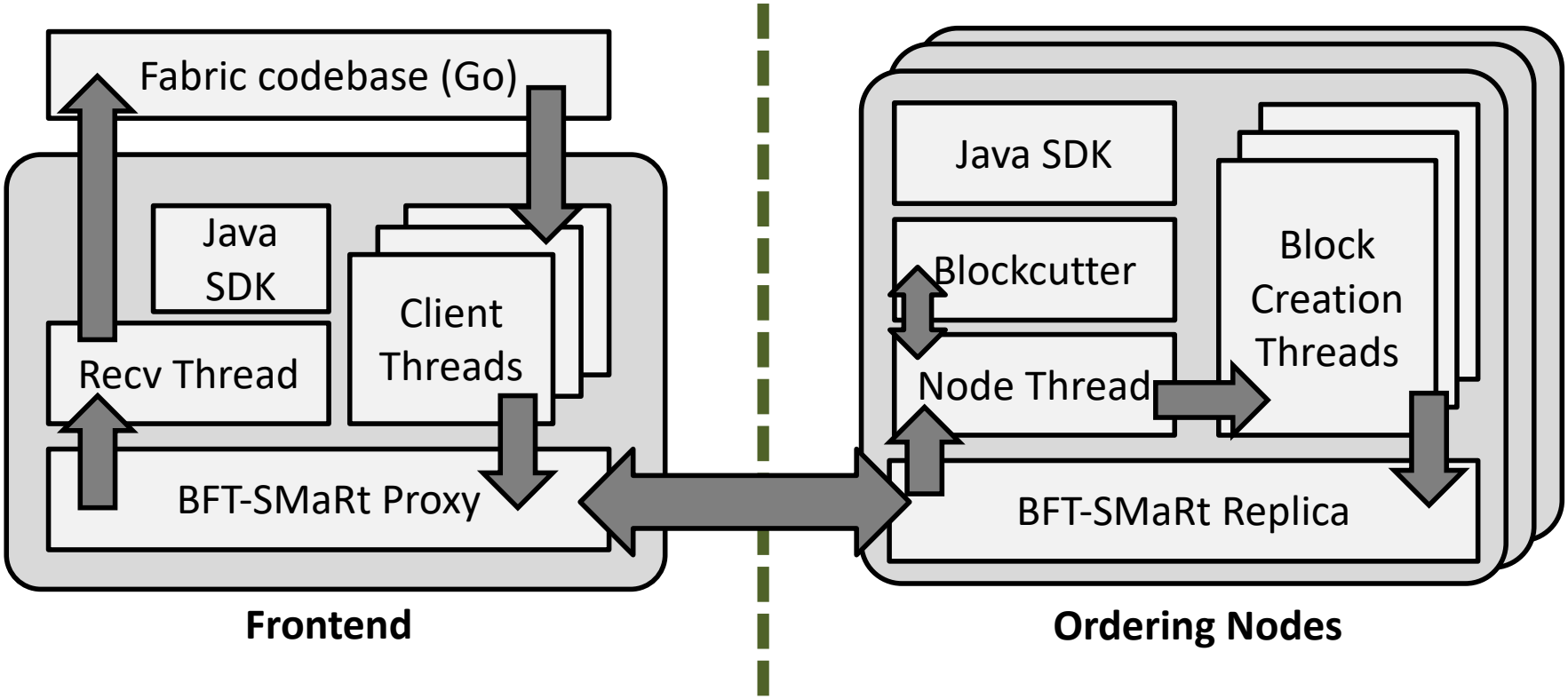
Disks can reach pure memory throughput

BFT-SMaRt Performance under "sporadic" events

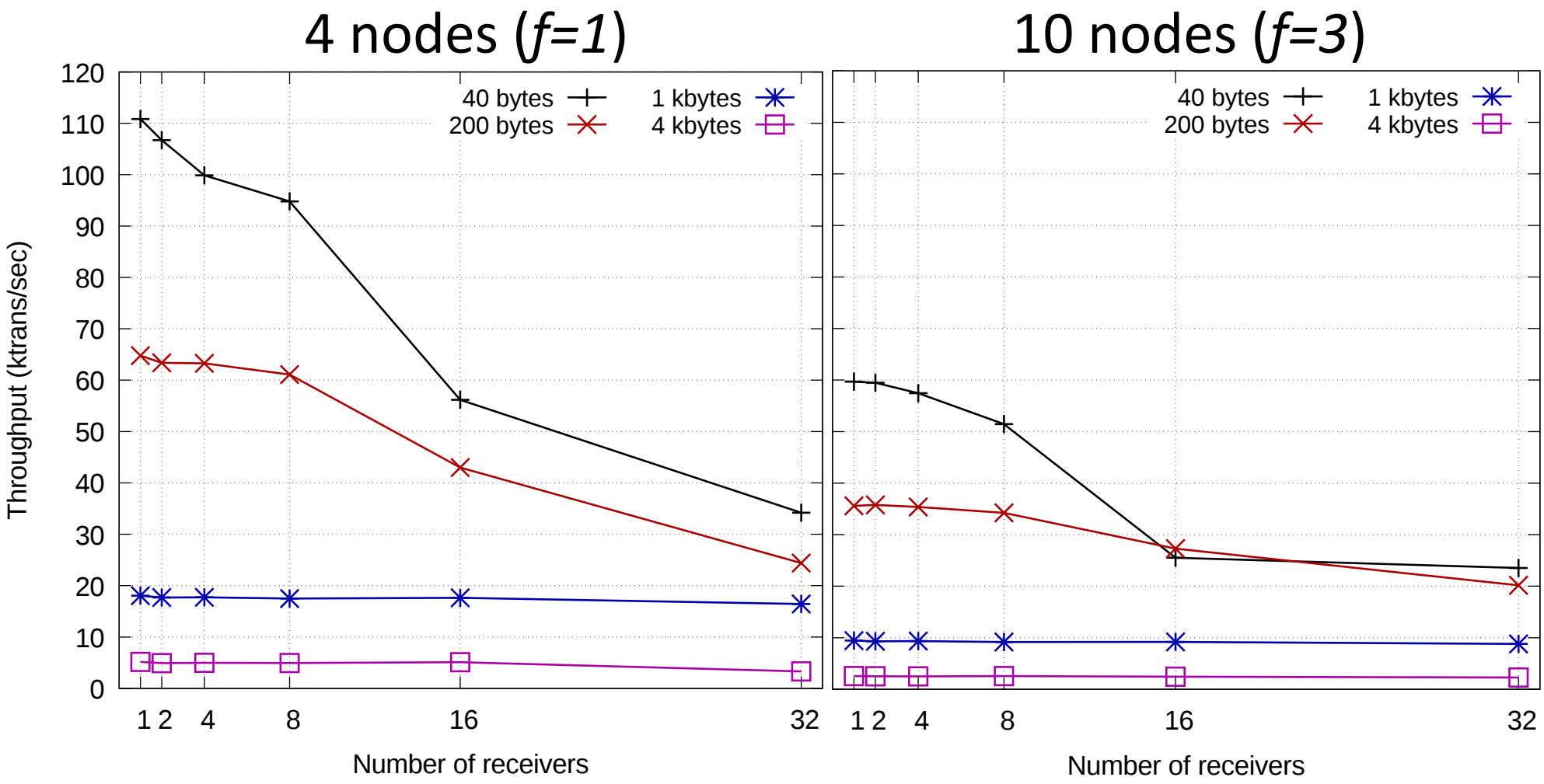


HYPERLEDGER
FABRIC

BFT-SMaRt Ordering



BFT-SMaRt Ordering Evaluation



A R&D Agenda

- **Robust BFT replication library**
  - Maintain a good basic implementation
- **Geo-replication**
  - Key BFT application: distributed trust
- **Scalability & Elasticity**
  - Increase performance dynamically w/ additional replicas
- **Diversity and Fault Independence**
  - How to withstand *f* malicious faults
- **Design a simple blockchain "platform"**
  - How to go from BFT SMR to a Blockchain

Questions?

- Alysson Bessani
  - anbessani@fc.ul.pt
  - www.di.fc.ul.pt/~bessani

Ciências ULisboa

- To know more:
  - BFT-SMaRt: http://bft-smart.github.io/library/
  - **Bessani et al.** ***State Machine Replication for the Masses with BFT-SMaRt.*** **IEEE/IFIP DSN'14.**
  - **Bessani et al.** ***On the Efficiency of Durable State Machine Replication.*** **USENIX ATC'13.**
  - **Sousa, Bessani.** ***Separating the WHEAT from the Chaff: An Empirical Design for Geo-replicated State Machines.*** **IEEE SRDS'15.**
  - **Sousa et al.** ***A Byzantine Fault-Tolerant Ordering Service for Hyperledger Fabric Blockchain Platform.*** **IEEE/IFIP DSN'18.**