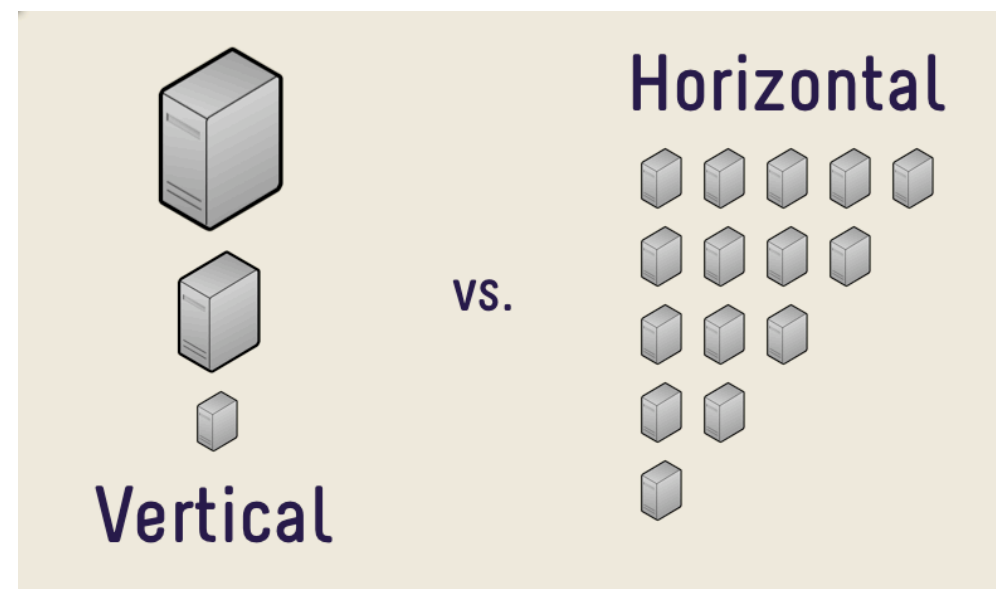


Scalability

50.037 Blockchain Technology
Paweł Szałachowski

Scalability

- “*Scalability is the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged to accommodate that growth.*”
- Horizontal and vertical scaling



Blockchain Stack

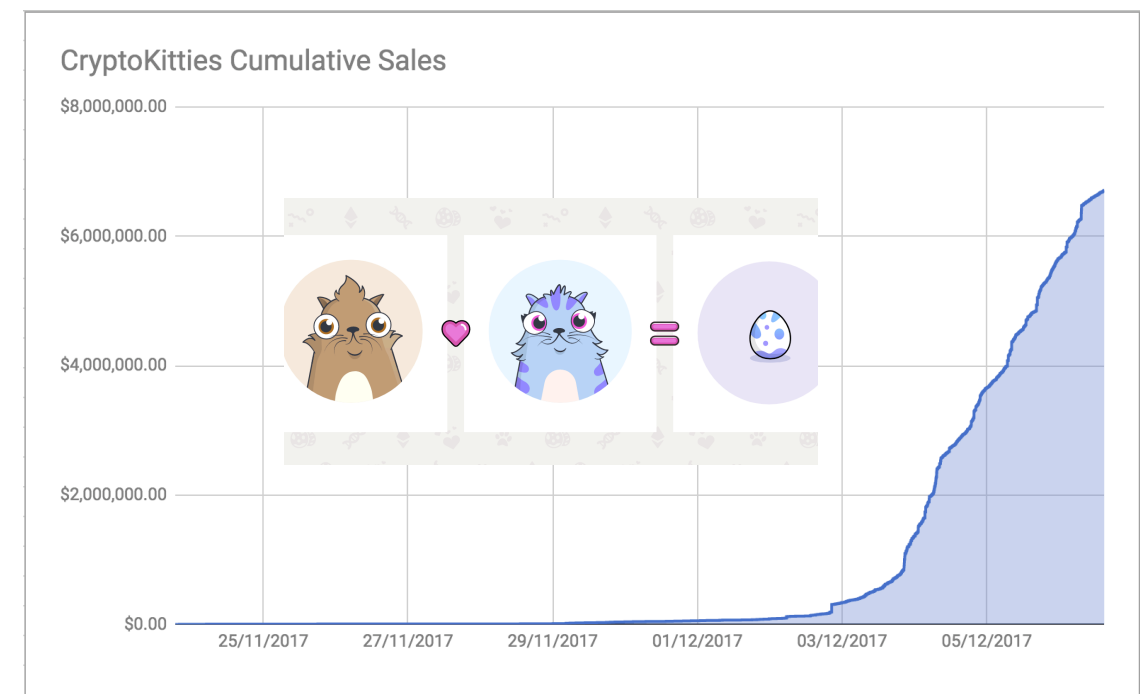
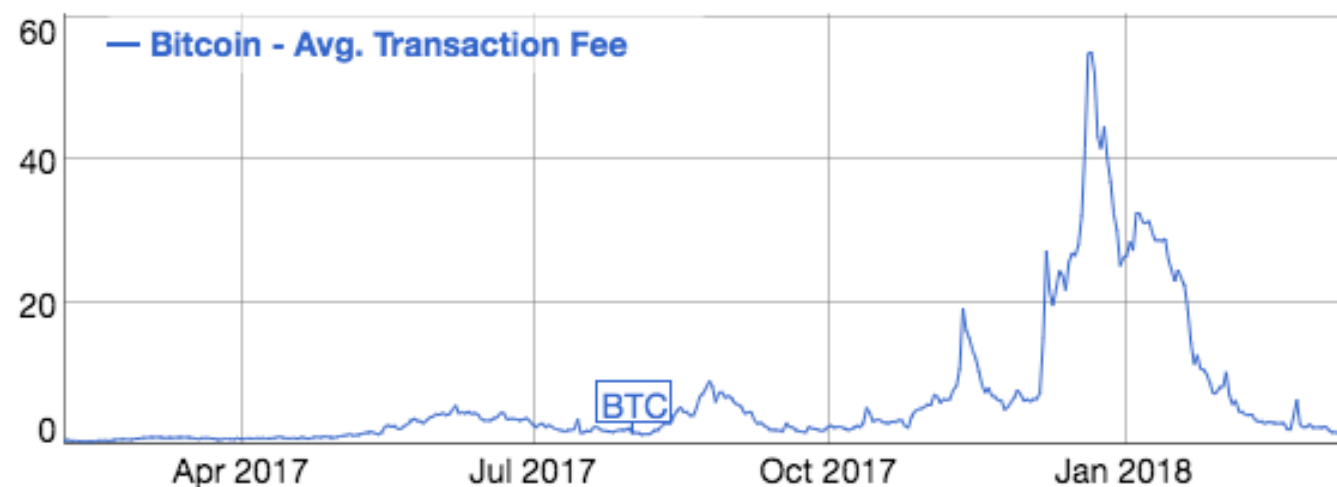
- Network: propagate transactions
 - Latency, bandwidth, # of nodes, ...
- **Consensus:** order Tx's
 - # of nodes, # of Tx's (throughput)
- RSM: Tx's validation, contract execution, ...
 - State size, execution complexity, ...
- Apps: use the current state to implement some logic

Current State

- Throughput
 - Bitcoin: 7 tx/s
 - Ethereum: 10 tx/s
 - Visa: 50k tx/s
- Strategy
 - Faster tx processing
 - Faster consensus
 - Parallel execution

Why needed?

- Adoption
 - Real-world apps require high throughput and low latency
- Inverse scale effect
- Fees



Blockchain Performance

- Bandwidth:
 - How many Txs can be processed?
- Latency
 - What is the consensus delay?
- Mining power utilization
 - The ratio between the mining power of the current chain and the mining power of the entire blockchain (describes stale block rate too), describes *security*
- Fairness
 - A miner should benefit from rewards proportionally to its mining power

Naive Improvements

- Blocks not every 10 minutes, but e.g., every 10 seconds
 - More forks => less mining power utilization => weaker security
- Larger blocks (very controversial topic BTW)
 - It takes longer to propagate
 - More forks =>
- Bitcoin -> Bitcoin Cash -> Bitcoin ABC vs Bitcoin SV

Security vs Performance

- Seems like Nakamoto consensus has some inherent tradeoffs
 - Security vs performance tradeoff
- Does it have to be like that?
 - We cannot significantly increase block or make them very frequent
- Design space
 - Why we need PoW?
 - Does it have to be combined with transactions propagation?

Bitcoin-NG

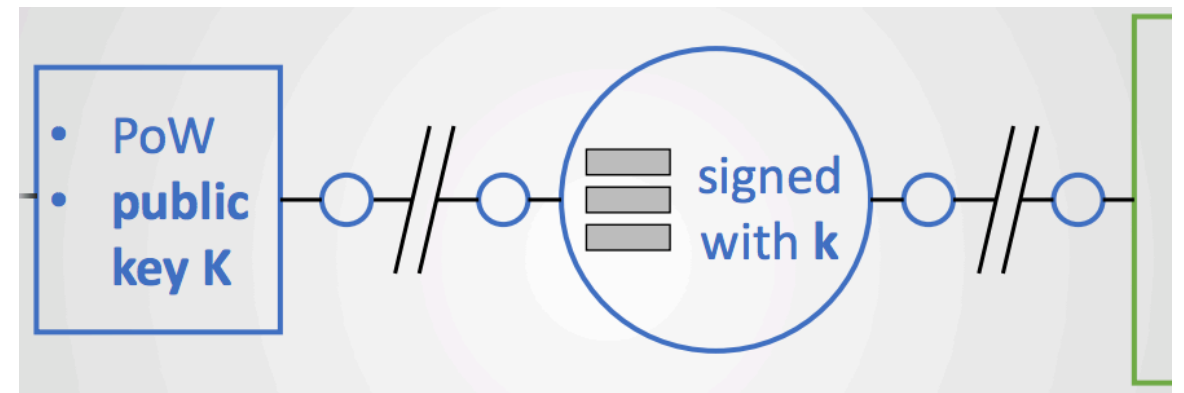
- <https://www.usenix.org/node/194907> (paper, slides, talk)
- Insights
 - In Bitcoin, leader election and transaction serialization is combined
 - Why do not try to decouple it?
 - Elect leader via PoW and let her commit transactions
 - (Different order than in Bitcoin)

Bitcoin-NG

- Key blocks
 - Used for PoW-based leader election, i.e., $H(\text{header}) < T$
 - Point to the previous block (key or microblock)

- The strongest-chain rule

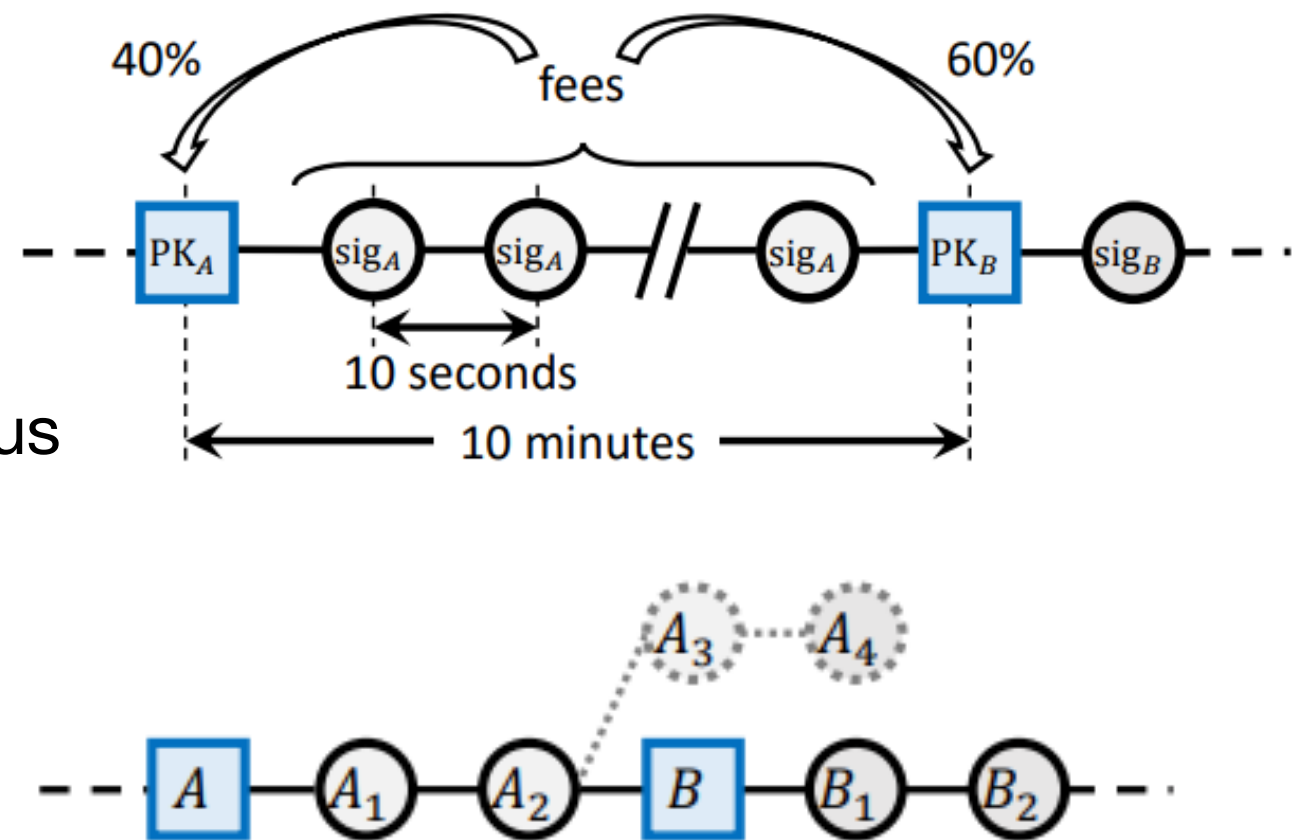
- Microblocks



- Generated by leader, at a defined rate
- Contains header (with PrevHash) and a set of transactions

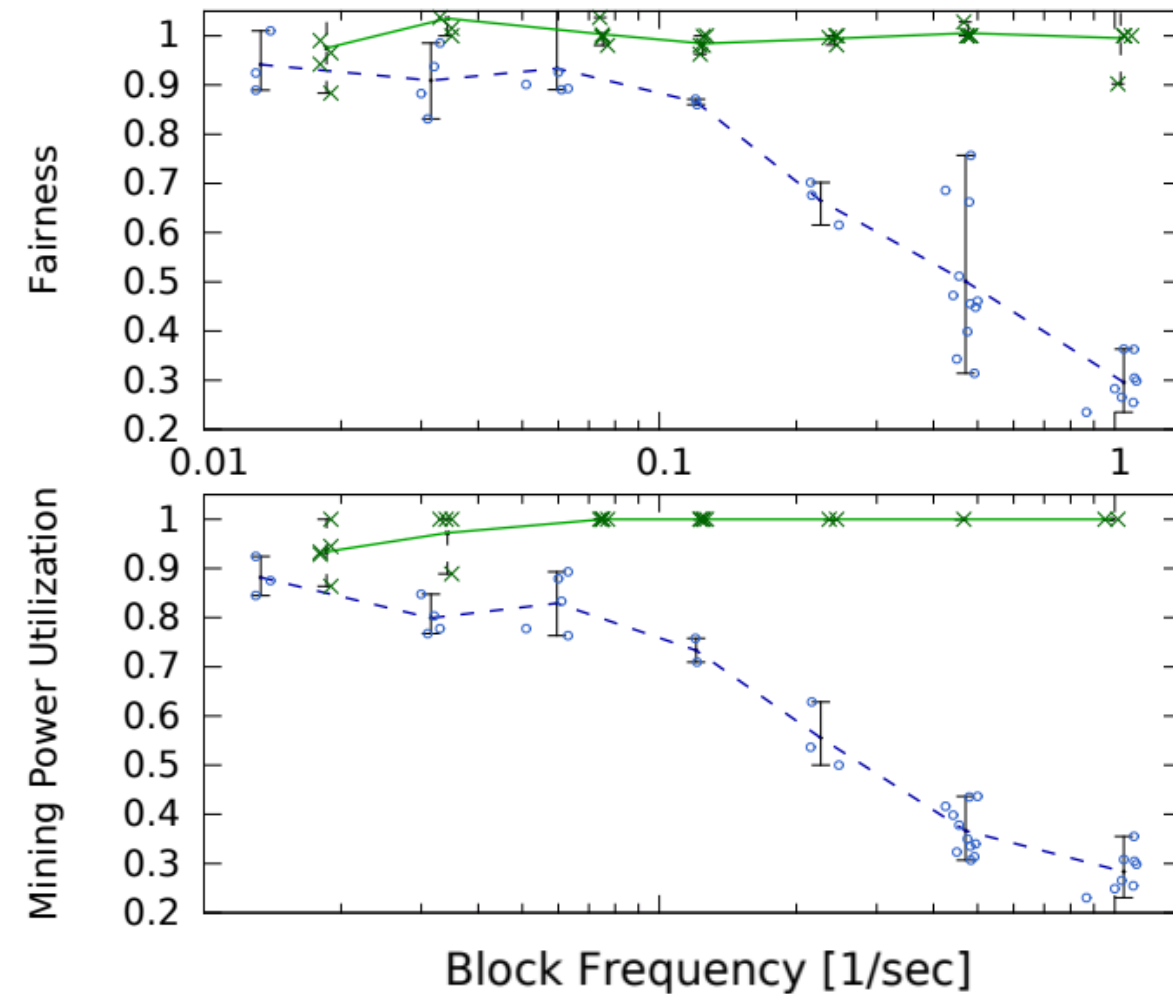
Bitcoin-NG

- Incentives
 - Leaders get rewards and tx fees
 - The next leader gets 60% of the previous tx fees (why?)
- Confirmations
 - Short forks will be frequent
- Microblock forks may be malicious
 - Entry with a proof of fraud can invalidate the revenue of malicious leaders



Bitcoin-NG

- Much better scalability and performance than Bitcoin
- Many systems build on this or similar ideas
- Everyone validates Tx's
 - Throughput limited by a single machine
- Can we do better?



Sharding

Sharding

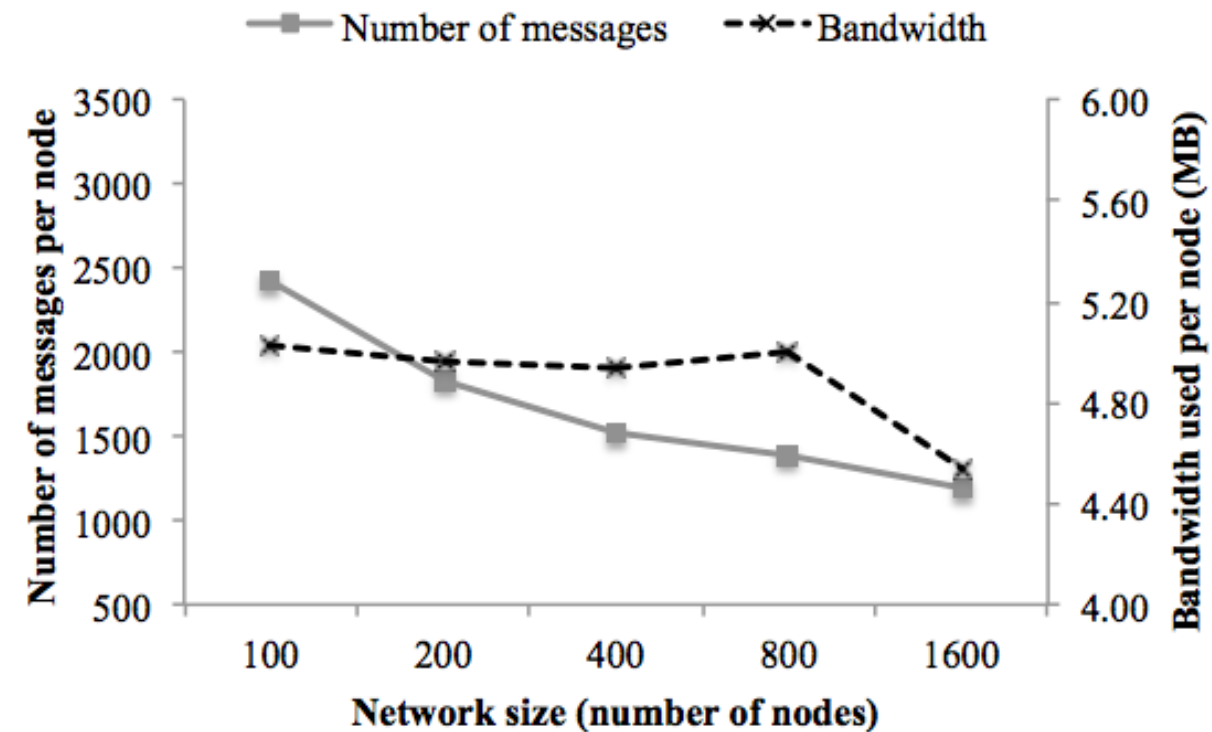
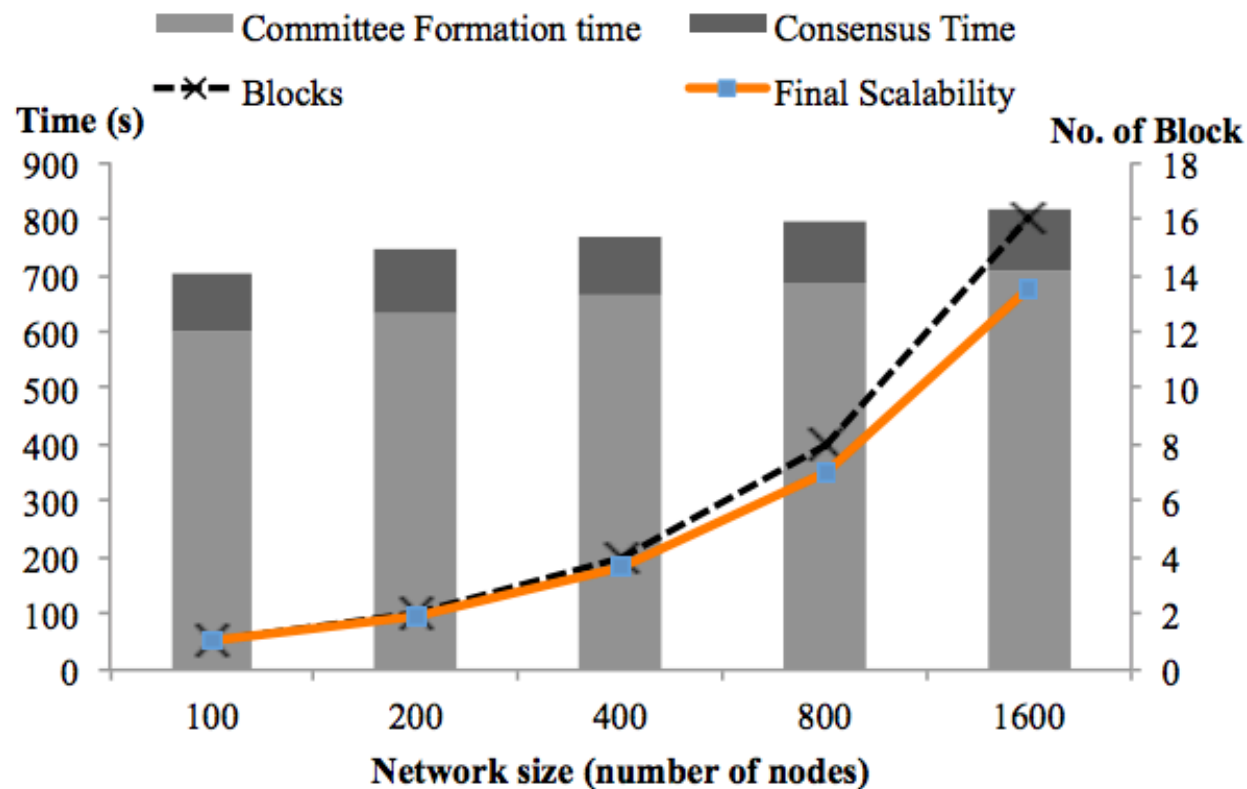
- The concept from database processing
- Divide transactions into groups and let different nodes process them
- Horizontal scaling
 - Throughput increases linearly as the network grows
- Ideas: establish identities via PoW, divide work, run BFT

Sharding: Elastico

<https://dl.acm.org/citation.cfm?id=2978389>

1. Use PoW to establish identities
 - $ID = H(R, IP, PubKey, Nonce) < T$
 - R is security-critical, see below
2. Assign committees (use randomness of IDs)
 - Each committee has C members and a directory server (w/ members)
3. Propose a block within a committee
 - Run BFT agreement, valid blocks have $2C/3 + 1$ signatures
4. Final committee to union all data blocks
 - Run BFT to produce a final block, that is then broadcast to everyone
 - R generated using R_x of final committee members

Sharding: Elastico



- Multiple improvements (ongoing research)
- ZILLIQA, OmniLedger, Chainspace, Saber, ...

Reading

- <https://fc16.ifca.ai/bitcoin/papers/CDE+16.pdf>
- <https://eprint.iacr.org/2016/555.pdf>
- + inline references