Scalability

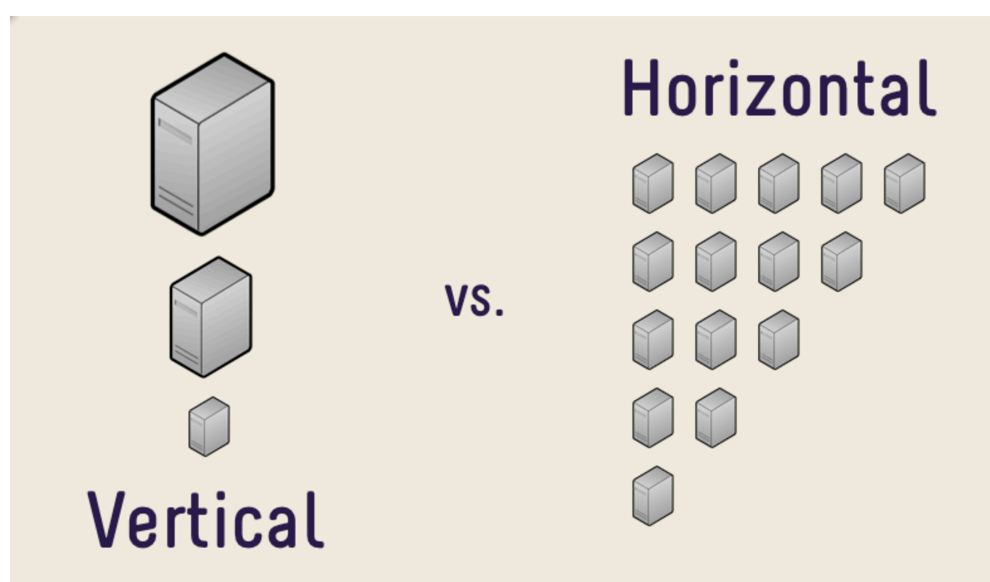
50.037 Blockchain Technology

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



https://pudgylogic.blogspot.com/

Blockchain Stack

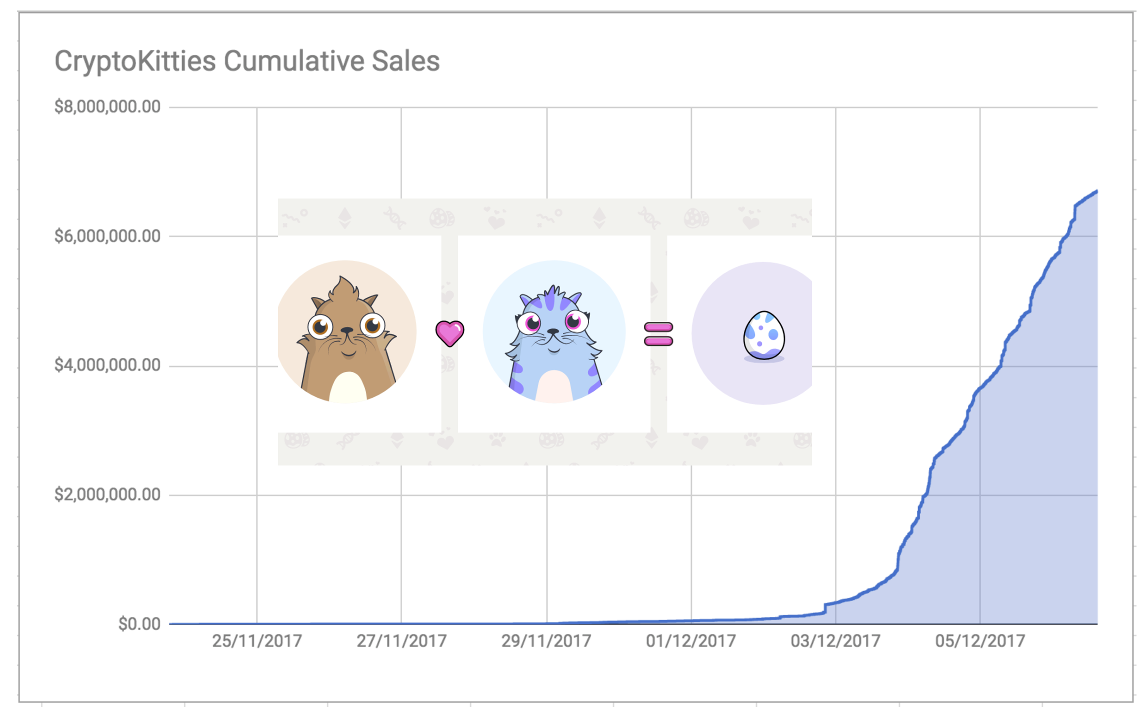
- Network: propagate transactions
  - Latency, bandwidth, # of nodes, …
- **Consensus**: order Txs
  - # of nodes, # of Txs (throughput)
- RSM: Txs validation, contract execution, …
  - State size, execution complexity, …
- Apps: use the current state to implement some logic

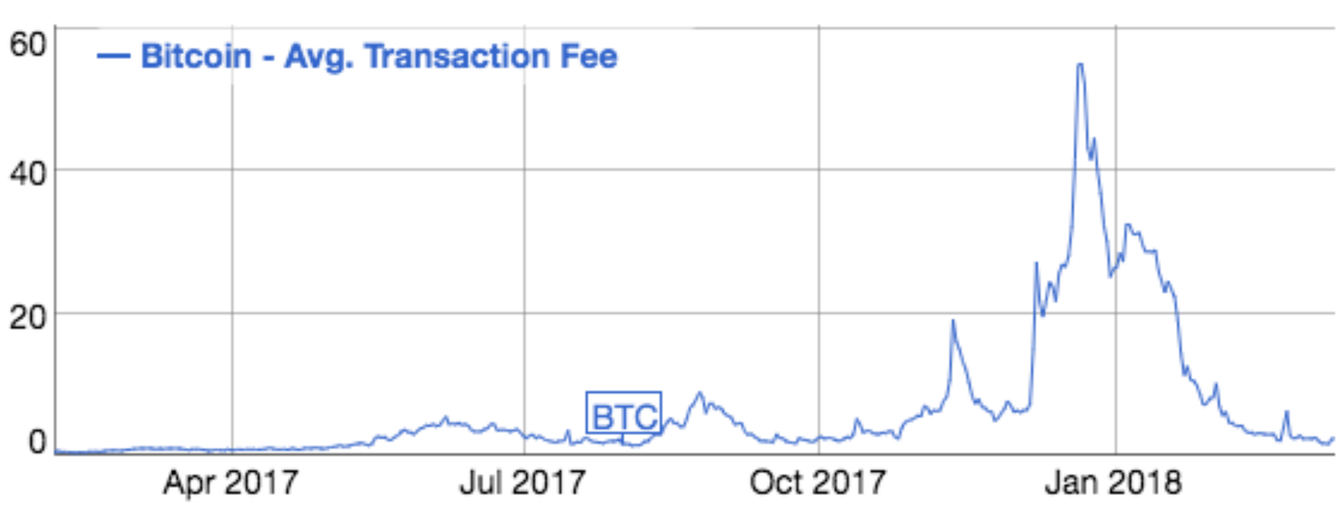
Current State

- 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

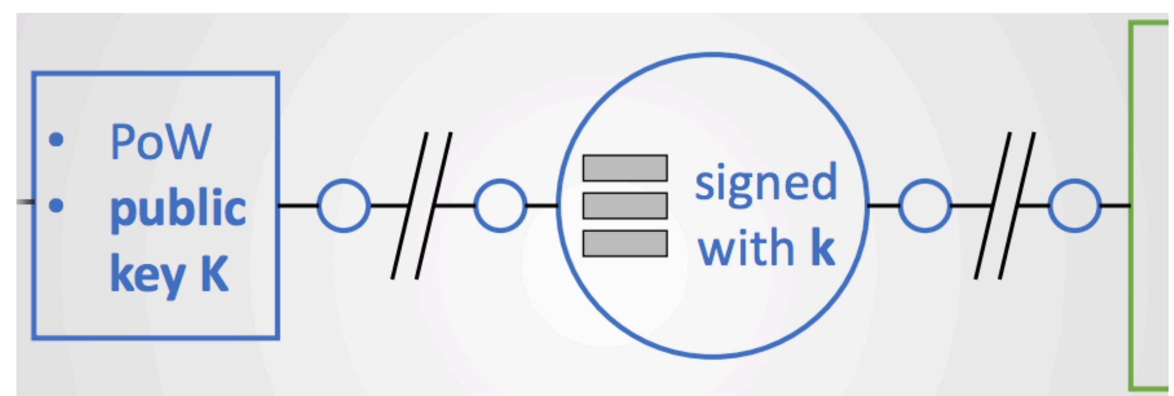
- Seems like Nakomoto consensus has some inherent tradeoffs
  - Security vs performance tradeoff
- Does it have to be like that?
  - We cannot significantly increase block or make them very frequent
- Design space
  - Why we need PoW?
  - Does it have to be combined with transactions propagation?

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)

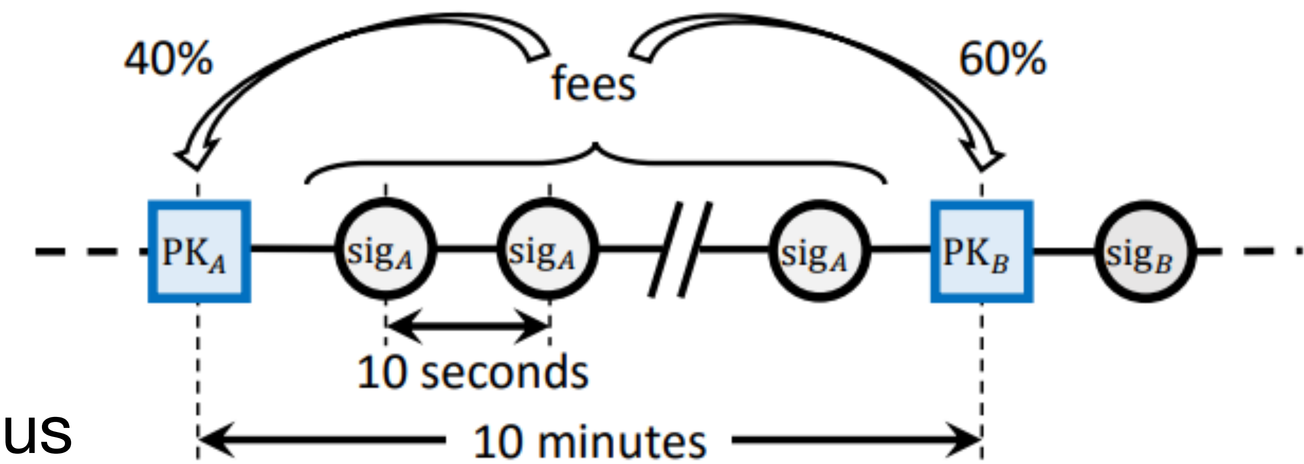
Bitcoin-NG

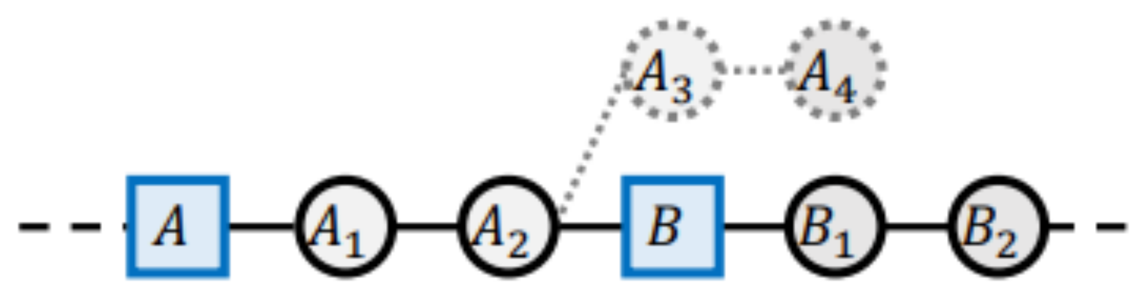
- Key blocks
  - Used for PoW-based leader election, i.e., H(header) < T
  - Point to the previous block (key or microblock)
  - The strongest-chain rule
- Microblocks
  - Generated by leader, at a defined rate
  - Contains header (with PrevHash) and a set of transactions



Bitcoin-NG

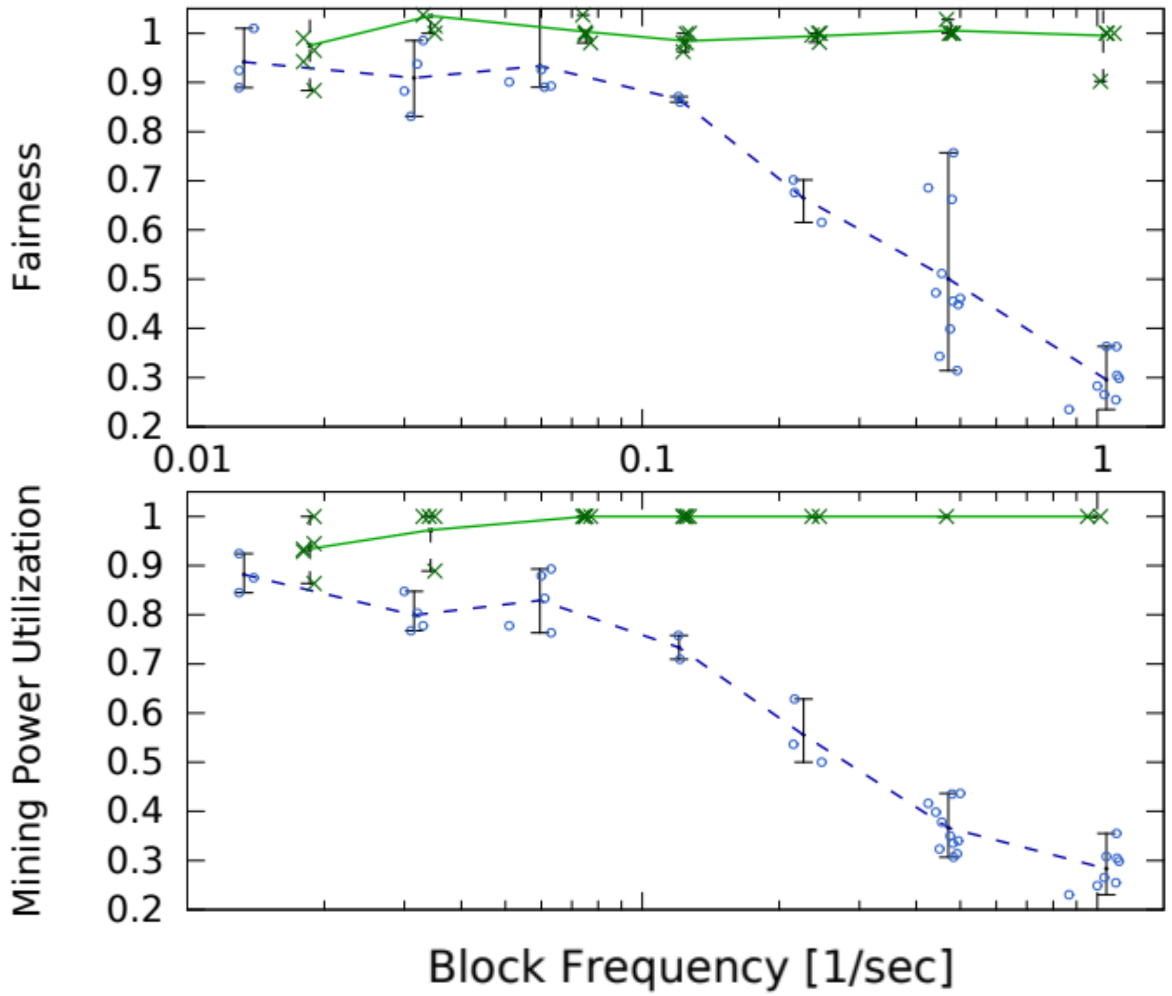
- Incentives
  - Leaders get rewards and tx fees
  - The next leader gets 60% of the previous tx fees (why?)
- Confirmations
  - Short forks will be frequent
- Microblock forks may be malicious
  - Entry with a proof of fraud can invalidate the revenue of malicious leaders





Bitcoin-NG

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



Sharding

Sharding

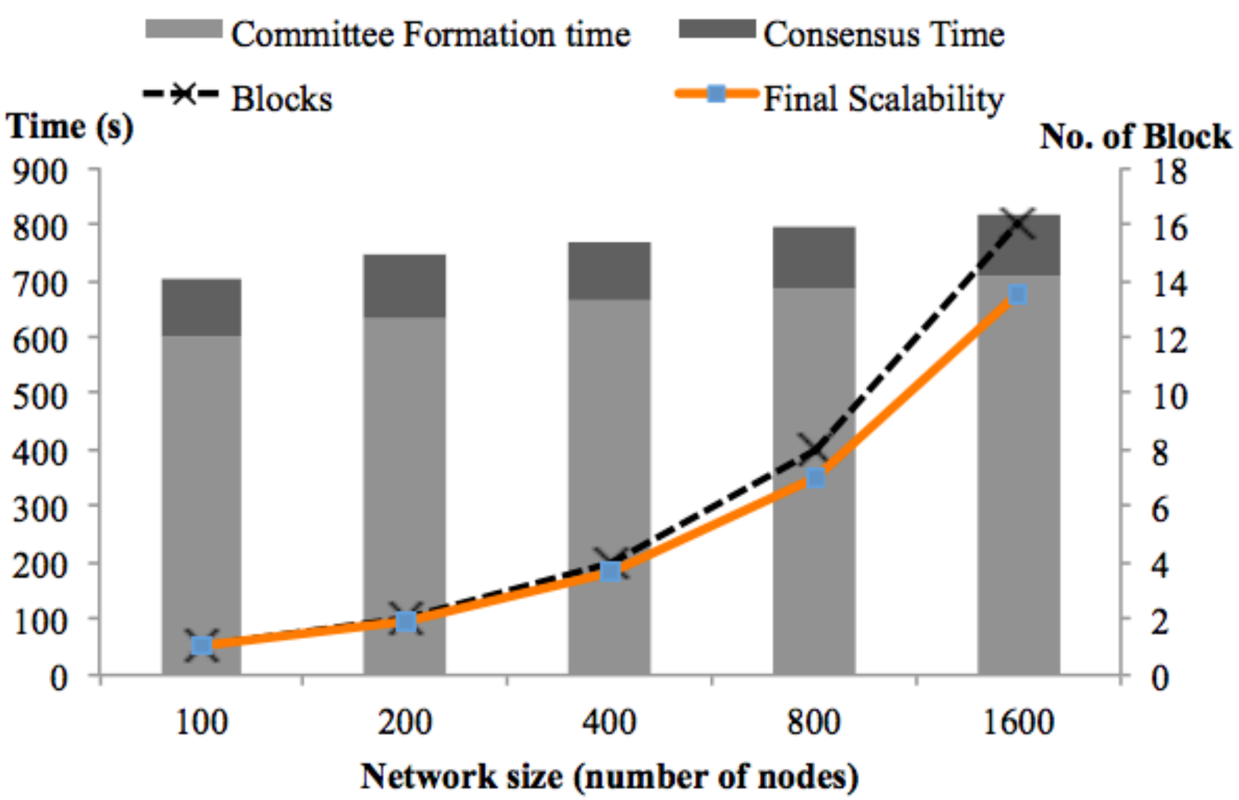
- The concept from database processing
- Divide transactions into groups and let different nodes process them
- Horizontal scaling
    - Throughput increases linearly as the network grows
- Ideas: establish identities via PoW, divide work, run BFT

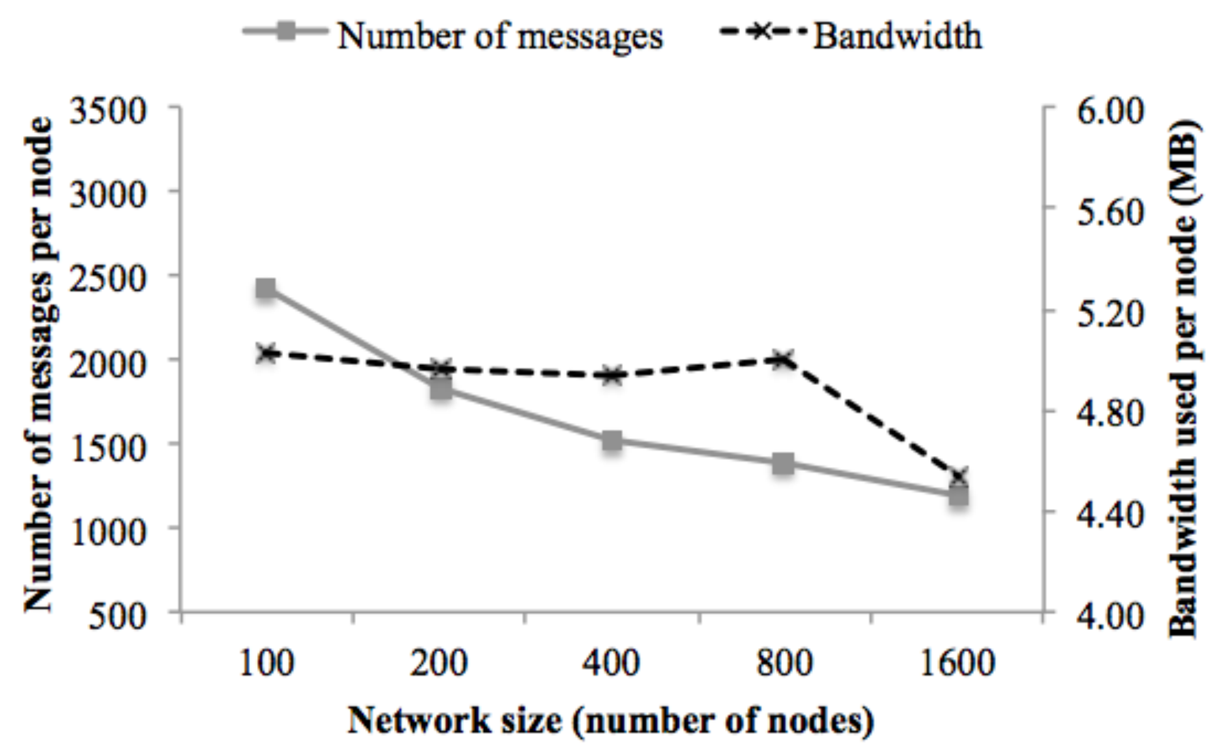
Sharding: Elastico

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, …

Directed Acyclic Graph (DAG)

DAG

- Nakamoto-like consensus is very easy
- The chain structure is simple
  - Can we introduce a more efficient data structure?
- Graphs
  - More powerful, as blocks can encode their worldviews
  - Why acyclic?

SPECTRE

- https://eprint.iacr.org/2016/1159.pdf (payment oriented)
- Intuitions and insights:
  - DAG can be used to encode the longest-chain rule
  - Bitcoin is related to voting
    - Every block votes for its chain
    - Cloning in Bitcoin
  - Vote amplification (miners strengthen the majority decision)

SPECTRE

- Miners create PoW-protected blocks with Txs
  - Blocks point to all know *tips* of the DAG
- Blocks can have conflicting Txs
- Nodes maintain local copies of DAG and accept/reject Tx
  - RobustReject, Pending, RobustAccepted

SPECTRE

- Vote over blocks (pairwise)
  - How many think A<B vs how many think B<A
    - This is just interpretation, does not have to be “*true*”
- Use results to accept/reject Txs
  - Tx of block B if:
    - All inputs are accepted
    - For every conflicting Tx’ in B’, B < B’

SPECTRE

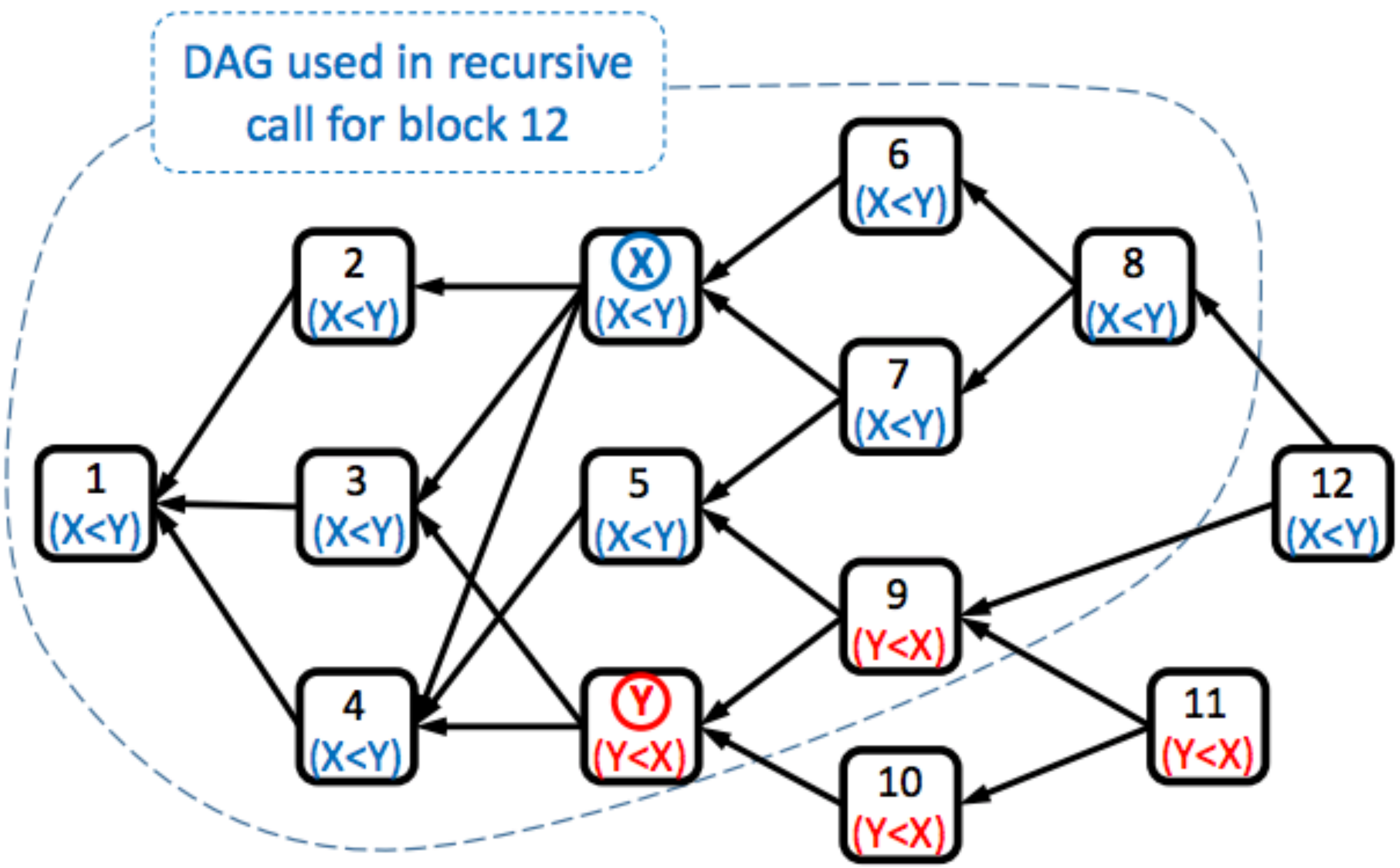


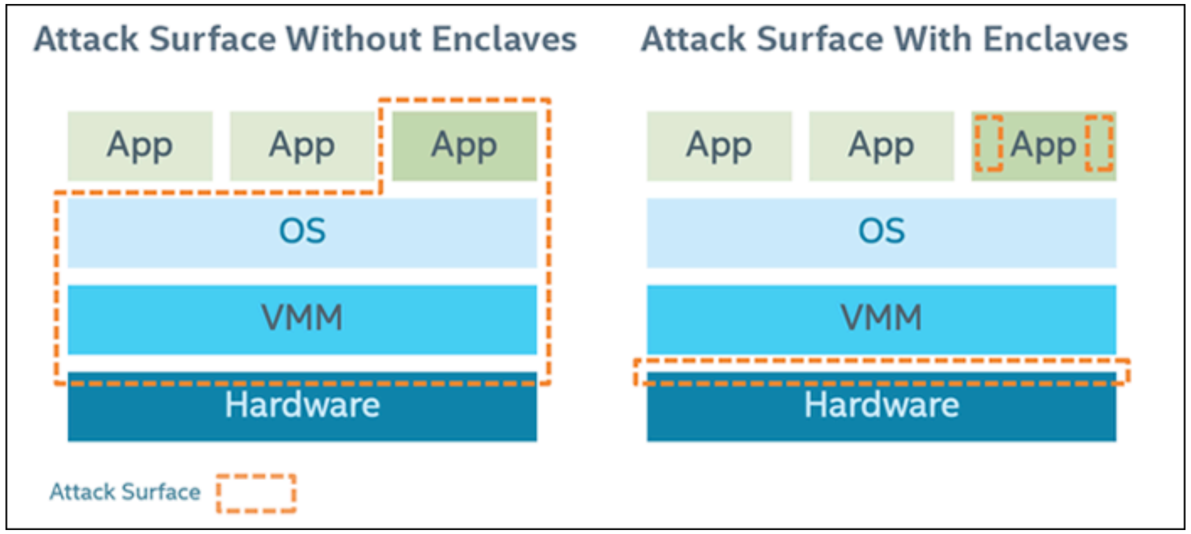
Fig. 1: An example of the voting procedure on a simple DAG. Block x and blocks 6-8 vote $x \prec y$ as they only see x in their past, and not y. Similarly, block y and blocks 9-11 vote $y \prec x$. Block 12 votes according to a recursive call on the DAG that does not contain blocks 10,11,12. Any block from 1-5 votes $x \prec y$, because it sees more $x \prec y$ voters in its future than $y \prec x$ voters.

- Clone-proof and amplified decisions
- Quick confirmations

Intel’s Proof of Elapsed Time (PoET)

Intel Software Guard Extensions (SGX)

- Set of new CPU instructions (Trusted Execution Environment — TEE)
- User code can allocate private regions of memory (**enclaves**), protected from other processes (even those running at higher privilege levels)
  - Running code and its memory is isolated from the rest of system
- Minimize trusted base
  - only CPU is trusted (even DRAM is untrusted, encryption needed)
- Attestation
  - Prove to remote system what code enclave is running



PoET

- Observation: PoW is introduced to mitigate Sybil attacks
  - … but with TEE that could be easier
- Idea: simulate PoW by sleep(…);
  - PoW-like guarantees
  - No energy waste
    - More energy vs more Intel SGX CPUs
  - Intel & SGX are trusted, not fully open + see the recent attacks

PoET

1. A newcomer node downloads the trusted code and sends a *join* message with the signed attestation

2. Nodes verify and accept/reject

3. In each round, every nodes gets a trusted random R and calls

   ```
   sleep(R);
   ```

4. The first awake node sends a signed msg that she is a leader

5. The statement is validated and the blocks can be produced

Permissioned Blockchains

Permissioned Blockchains

- We discuss open/permissionless blockchains so far
  - Great for some use cases, not so great for other
    - Good: Append-only, decentralized, available, robust, transparent…
    - Bad: Slow, expensive storage&computation, volatility, immature technology, publicly available data, difficult to manage/update…
      - Mainly caused by the permissionless setting
  - Why not reuse some of those ideas and run classic BFT consensus?
  - Efficiency could be one of the major benefits

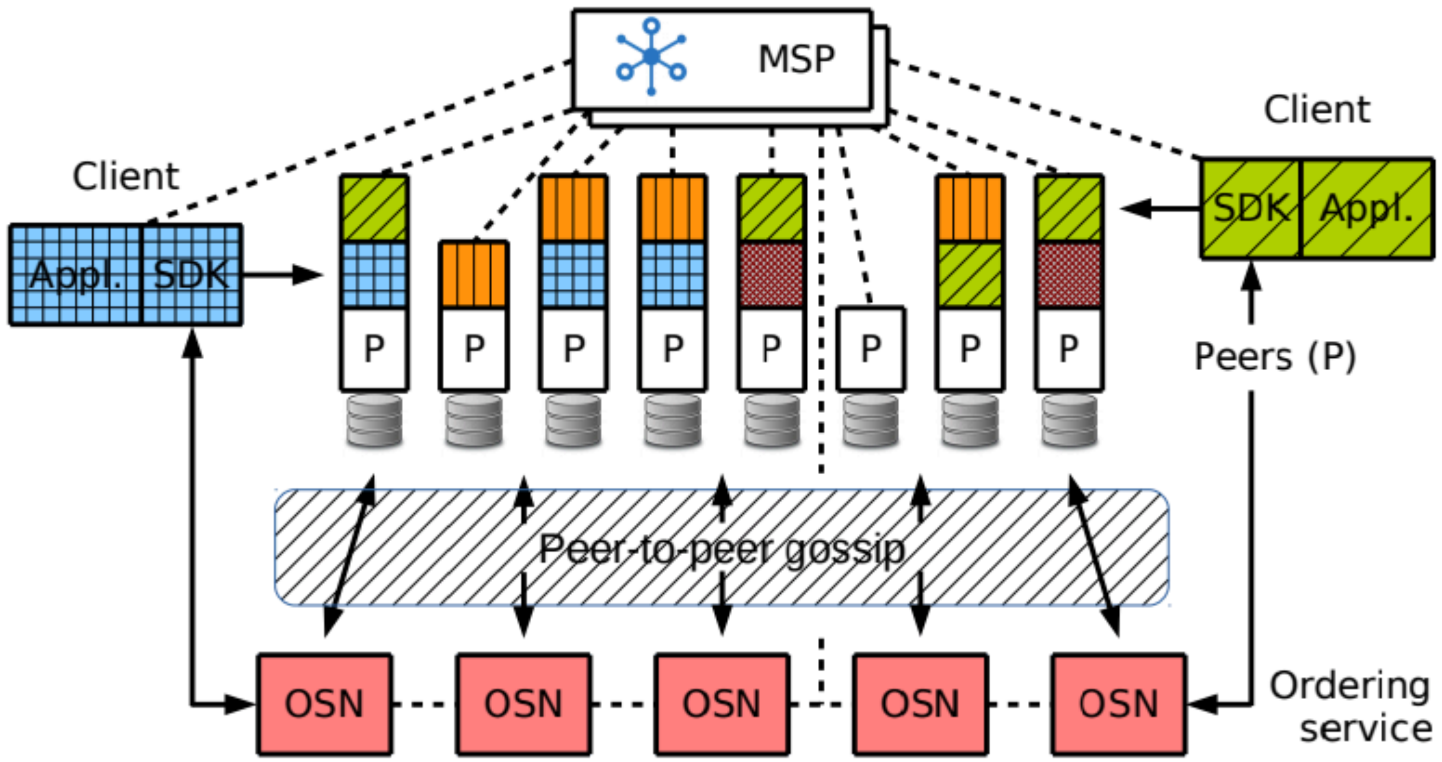
HYPERLEDGER

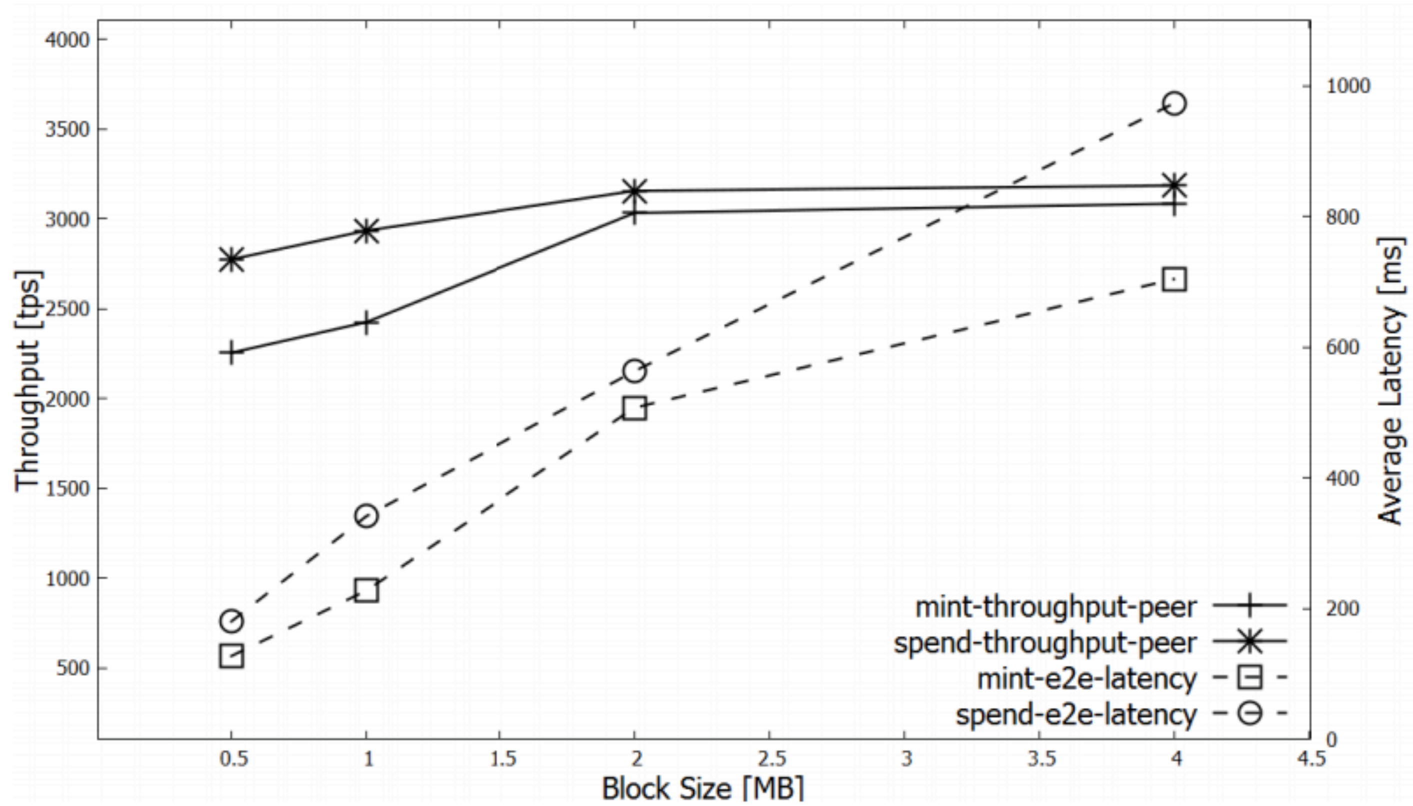
- Consortium
- Hyperledger Fabric (framework)
  - https://arxiv.org/abs/1801.10228
  - Business-oriented
  - Different consensus protocols supported
  - No built-in cryptocurrency
  - Powerful smart contracts

Hyperledger Farbic

- Membership service provider (MSP) provides identities to participants
- Clients submit transaction proposals for execution
- Peers execute transaction proposals and validate transactions
  - Only endorsing peers execute transactions (specified by a policy)
  - All peers maintain the blockchain ledger
- Ordering Service Nodes (OSN) establish the total order of all transactions

Hyperledger Fabric





Reading

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