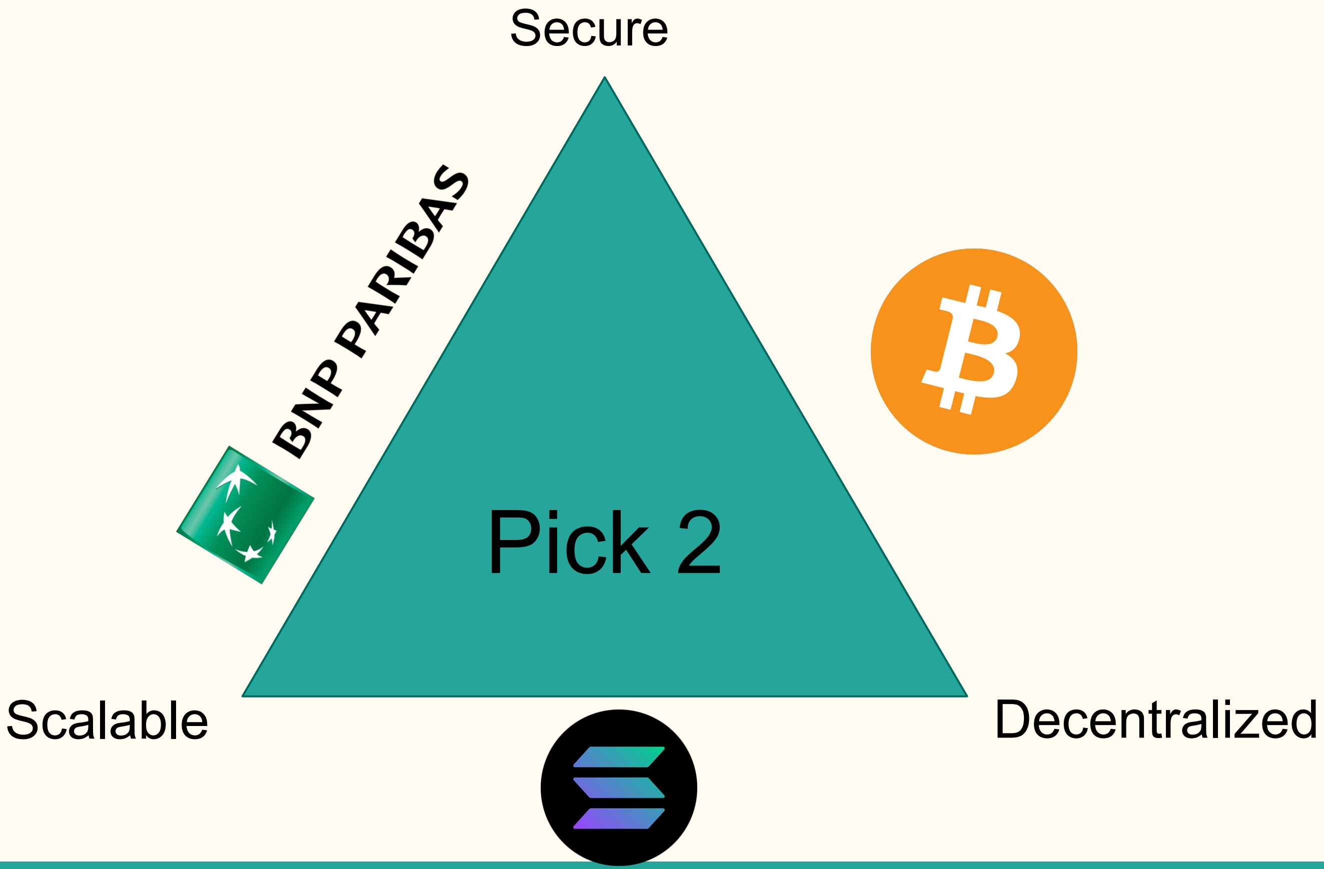
Lecture 12

Ethereum Scaling

The Blockchain Trilemma



Secure

Pick 2

Scalable

Decentralized

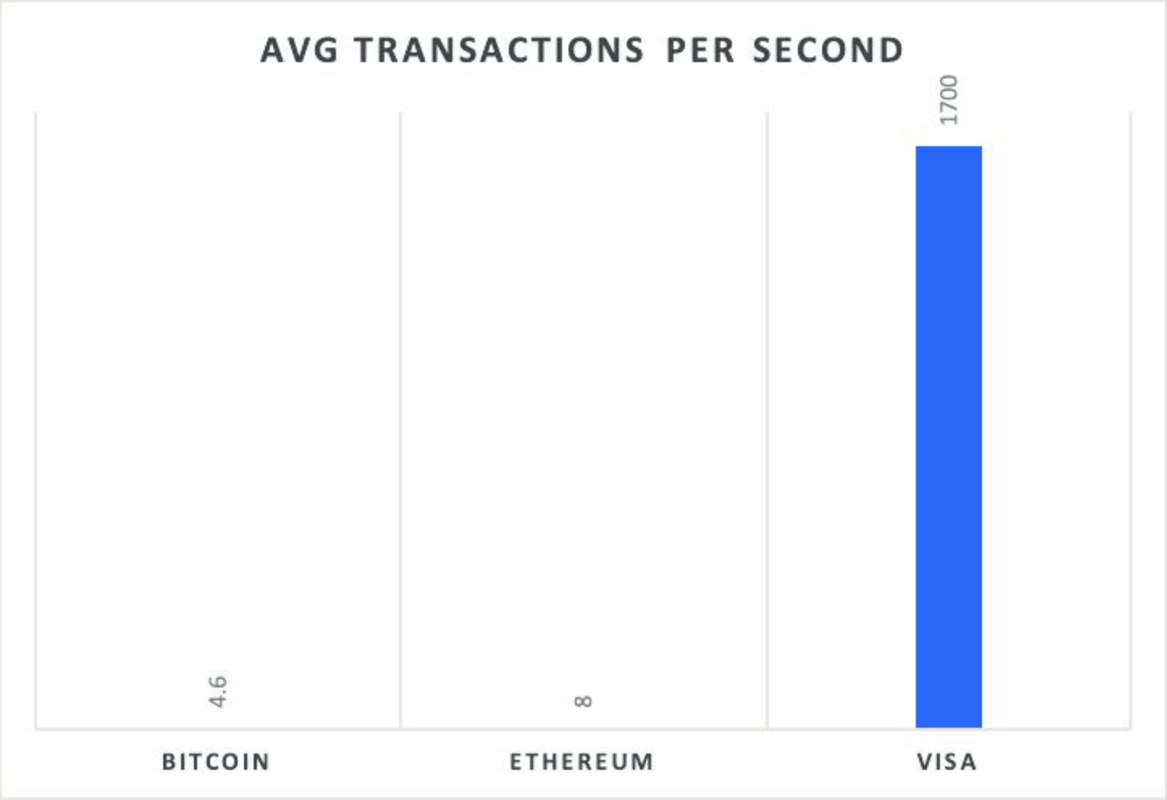
Transactions per Second

Financial Services generating transactions:

- ATMs
- POS devices
- Bank transfers
- Mobile banking
- Online payments

Speed is key to real world adoption.

24 hour SEPA mandate
How long would you wait for online payment confirmation?



Types of scaling

Layer 1 Scaling

Improvements that are made directly to the blockchain itself.

These improvements involve speed and utility increases to the chain itself. Eg. proof and consensus mechanisms. Block architecture.

Data processing and storage improvements of the chain.

Layer 2 Scaling

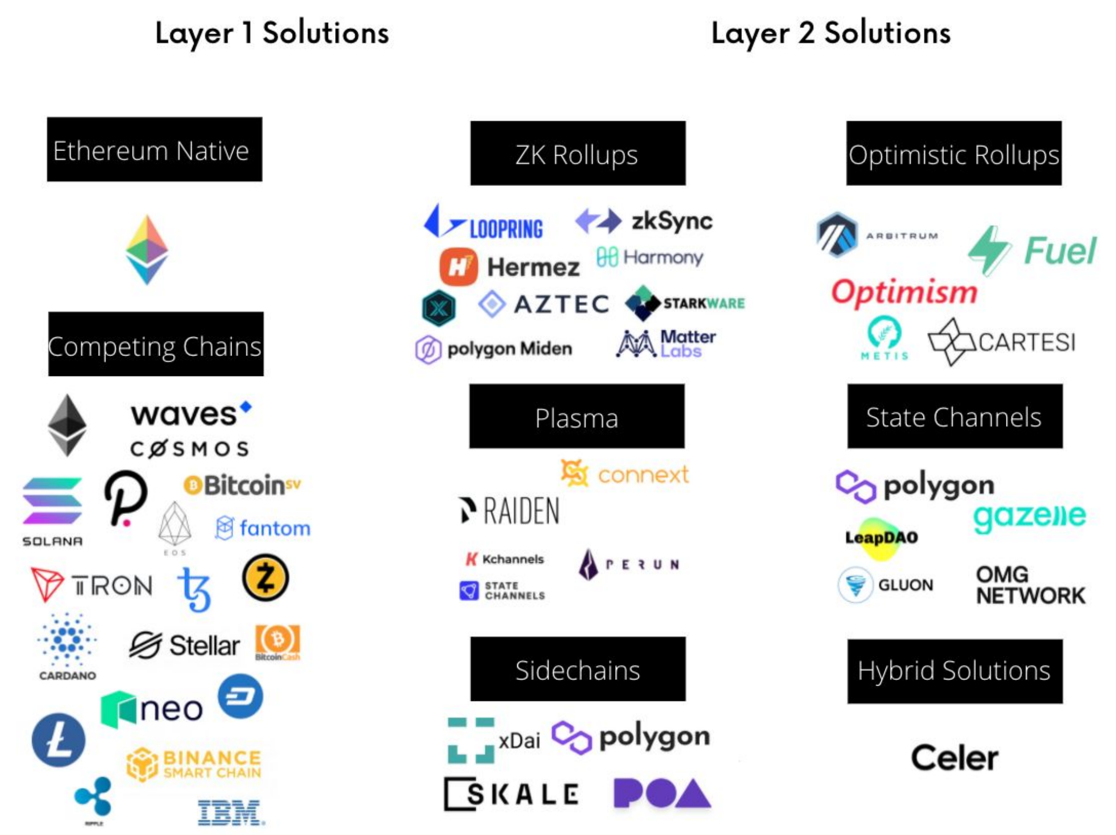
Improvements that are made on top of the L1 chain

L2 creation and operations are defined on L1 (Ethereum) but move the computation and storage demands off of L1.

The validity of L2 information is of concern when it is posted back onto L1 chains.

Layer 3 Hyperscaling

Blockchain Scaling Landscape



Layer 1 Solutions

Layer 2 Solutions

Ethereum Native

Competing Chains

ZK Rollups

Optimistic Rollups

Plasma

State Channels

Sidechains

Hybrid Solutions

Layer 1 Scaling

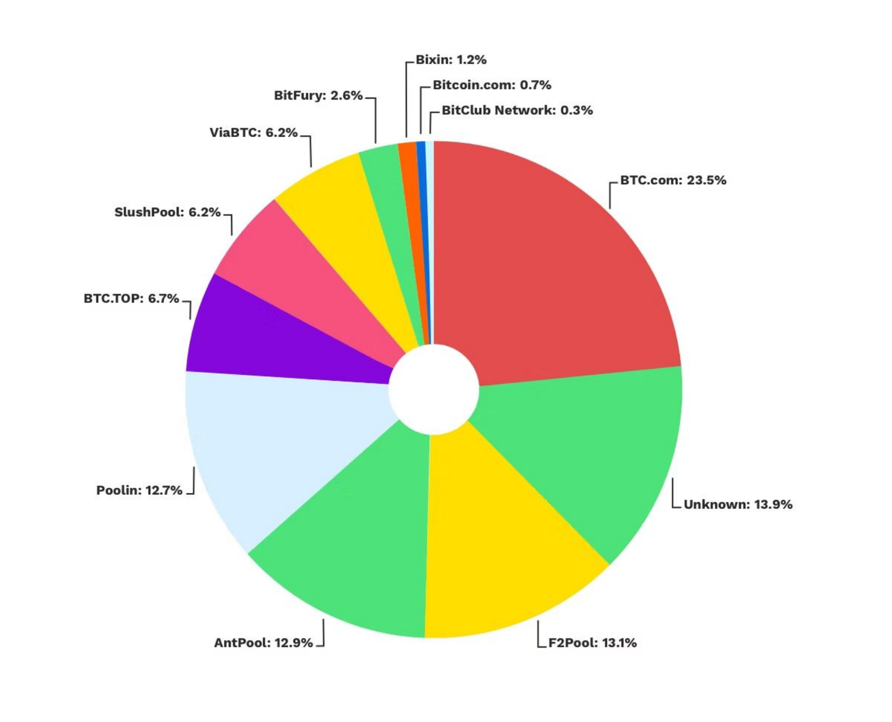
Consensus improvements & Sharding

Consensus mechanism improvements - Bitcoin

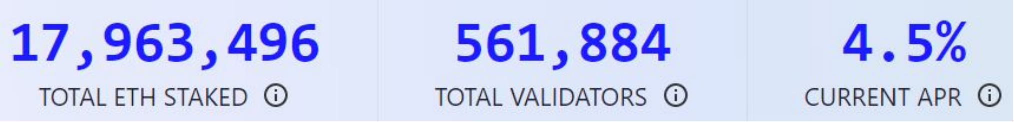
Proof of Work

Earn rewards by solving hashing puzzles of new blocks. The hash is usually required to produce a certain number of zeros at the beginning of the block. Once a new block is created and added to the chain, it is mined.

- Bitcoin rewards halving and puzzle difficulty increase reduces profitability.
- Currently puzzle difficulty means only specialised hardware (ASIC) can solve these puzzles.
- Individual mining is now impossible. Only mining pools profiting



Consensus mechanism improvements - Ethereum



17,963,496	561,884	4.5%
TOTAL ETH STAKED ⓘ	TOTAL VALIDATORS ⓘ	CURRENT APR ⓘ

Proof of Stake

Validators stake ETH vs Miners stake computation power. Validators can **propose** new blocks or **attest** blocks being propagated.

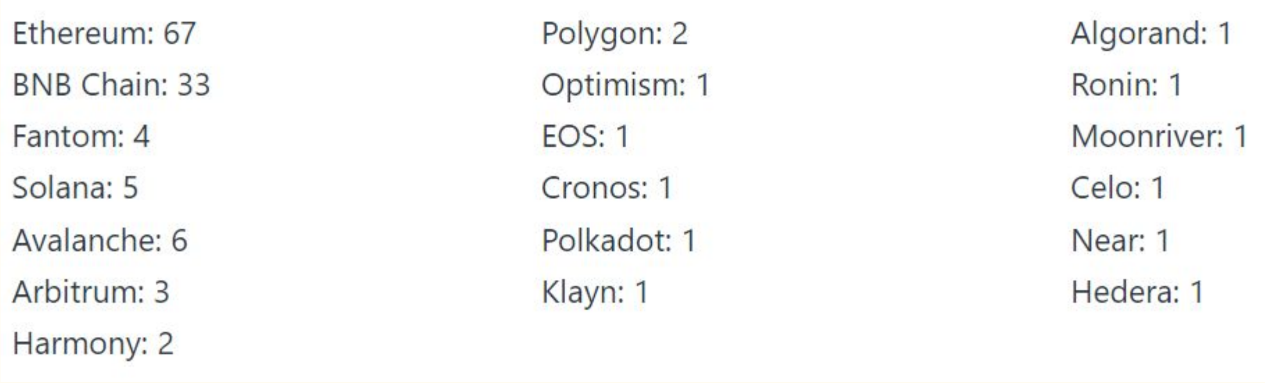
- Ethereum Classic (ETC) used PoW. Switched to PoS in 2022 (ETH).
- Every 12s, a Validator is chosen to be the proposer and a group is selected to be attestors. Availability is key!
- Beware of slashing, check rules.
  - Attesting two competing blocks
  - Proposer spams blocks or proposes malformed blocks (Gasper)
  - ETH penalty and ban period. Or permanent expulsion.
  - Whistleblower rewards
- Staking methods
  - Solo staking requires 32ETH (58k USD) + hardware (computation + redundancy)
  - Staking as a Service (32 ETH min)
  - Staking pools

Consensus mechanism improvements - Solana

Proof of History

A cryptographic clock that is implemented on top of PoW or PoS. A different take on Byzantine fault tolerance (51% attack, Sybil attack). Ethereum requires ⅔ attester confidence. PoH server generates a Verified Delay Function (VDF) and stamps each transaction. Validators check timestamp and send vote to PoH server. **No need for validators to come to consensus about a block.**

- PoH server a huge source of centrality.
- Added complexity increases network outages
- Decreased security checks means a lot hacks!

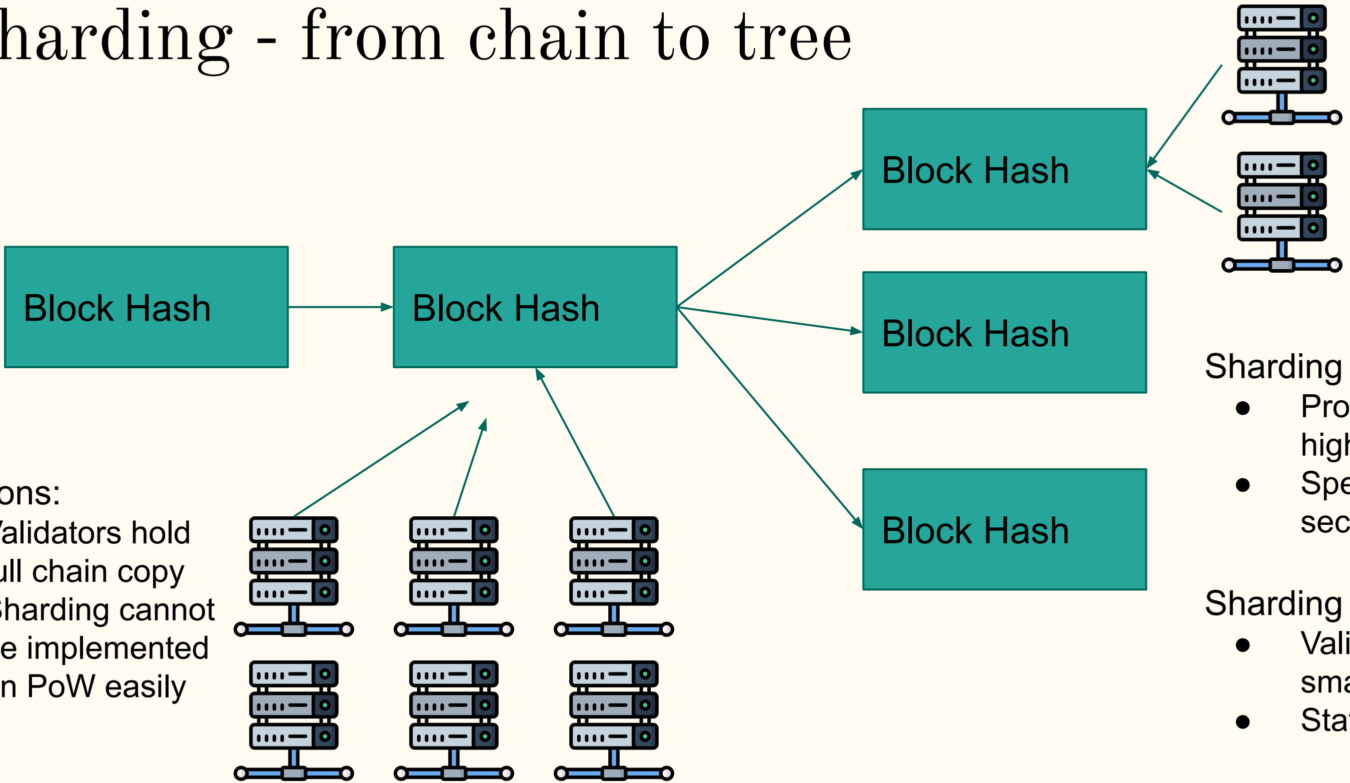


Ethereum: 67
BNB Chain: 33
Fantom: 4
Solana: 5
Avalanche: 6
Arbitrum: 3
Harmony: 2

Polygon: 2
Optimism: 1
EOS: 1
Cronos: 1
Polkadot: 1
Klayn: 1

Algorand: 1
Ronin: 1
Moonriver: 1
Celo: 1
Near: 1
Hedera: 1

Sharding - from chain to tree



Limitations:

- Validators hold full chain copy
- Sharding cannot be implemented on PoW easily

Sharding advantages:

- Process transactions in parallel, higher TPS!
- Speed improvement without security loss unlike PoH.

Sharding challenges:

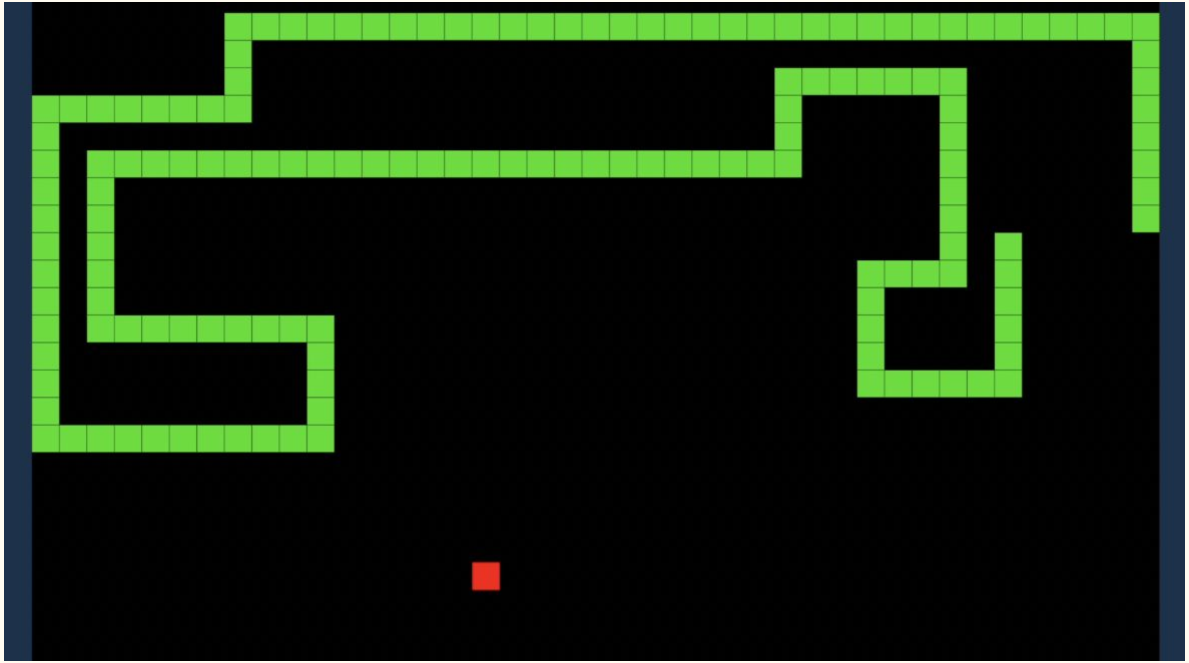
- Validators are now diluted into smaller shard pools (still BFT?)
- State collisions must be resolved

Sharding, ultimately, was never implemented on Ethereum even though it was discussed for a long time. With EIP-4844 (30 March 2023), Ethereum is considering **Danksharding** to work with Layer 2 solutions

Layer 2 Scaling

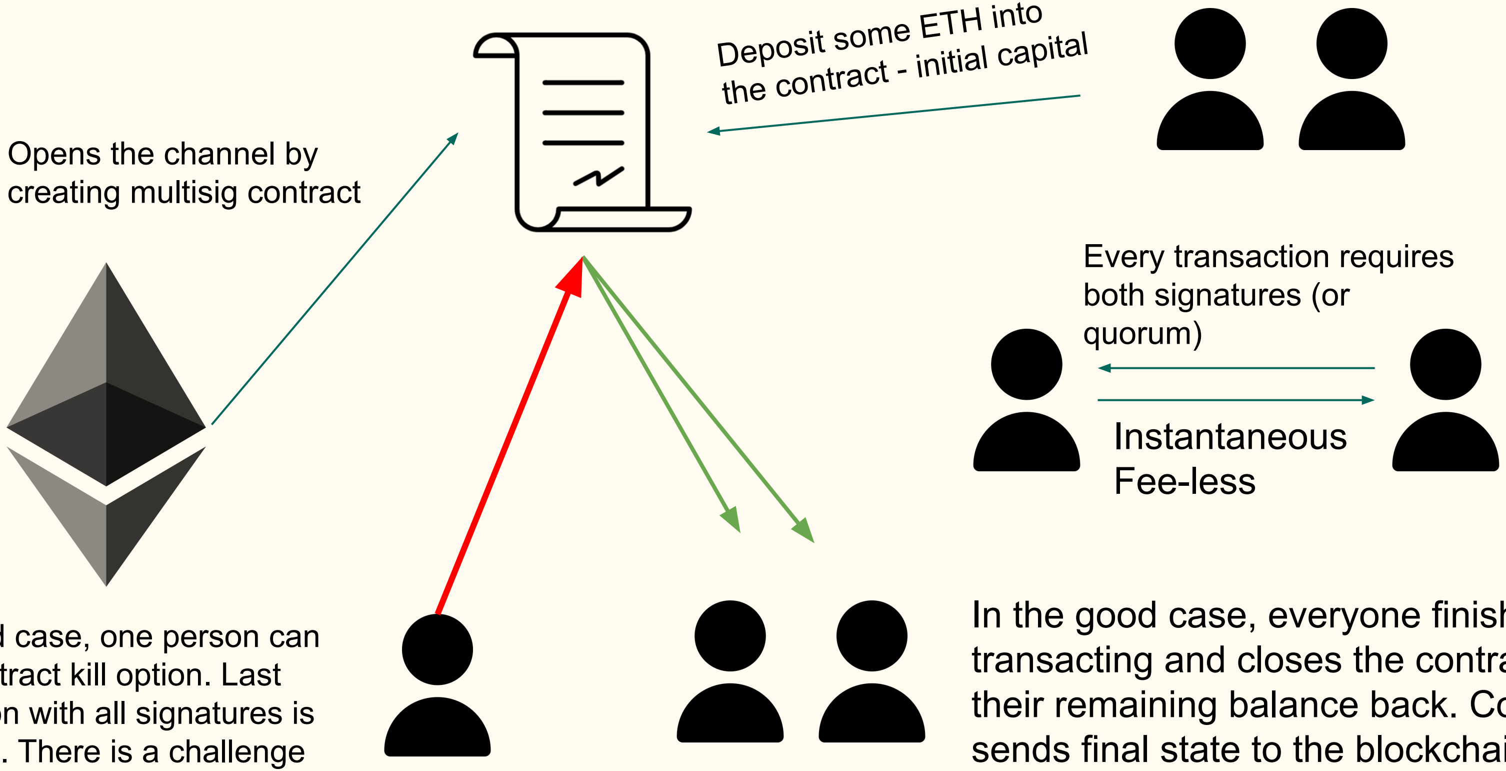
State Channels, Side Chains, Rollups

State Channels - Moving transactions off chain



Only 2 transactions: channel open state and channel close state

State Channels - multisig smart contracts



Opens the channel by creating multisig contract

Deposit some ETH into the contract - initial capital

Every transaction requires both signatures (or quorum)

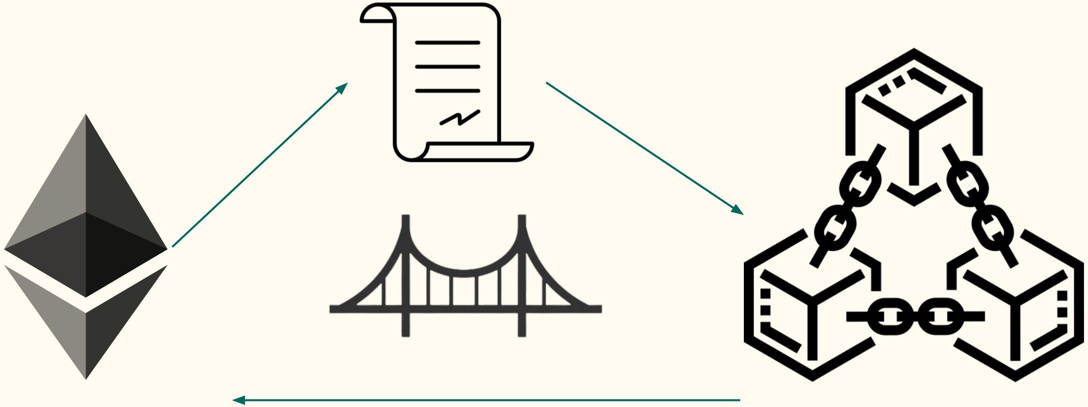
Instantaneous
Fee-less

In the bad case, one person can call a contract kill option. Last transaction with all signatures is final state. There is a challenge period before resolution.

In the good case, everyone finishes transacting and closes the contract, getting their remaining balance back. Contract sends final state to the blockchain.

Side Chains (Plasma)

The root contract governs the creation of the side chain - its consensus mechanisms and block design



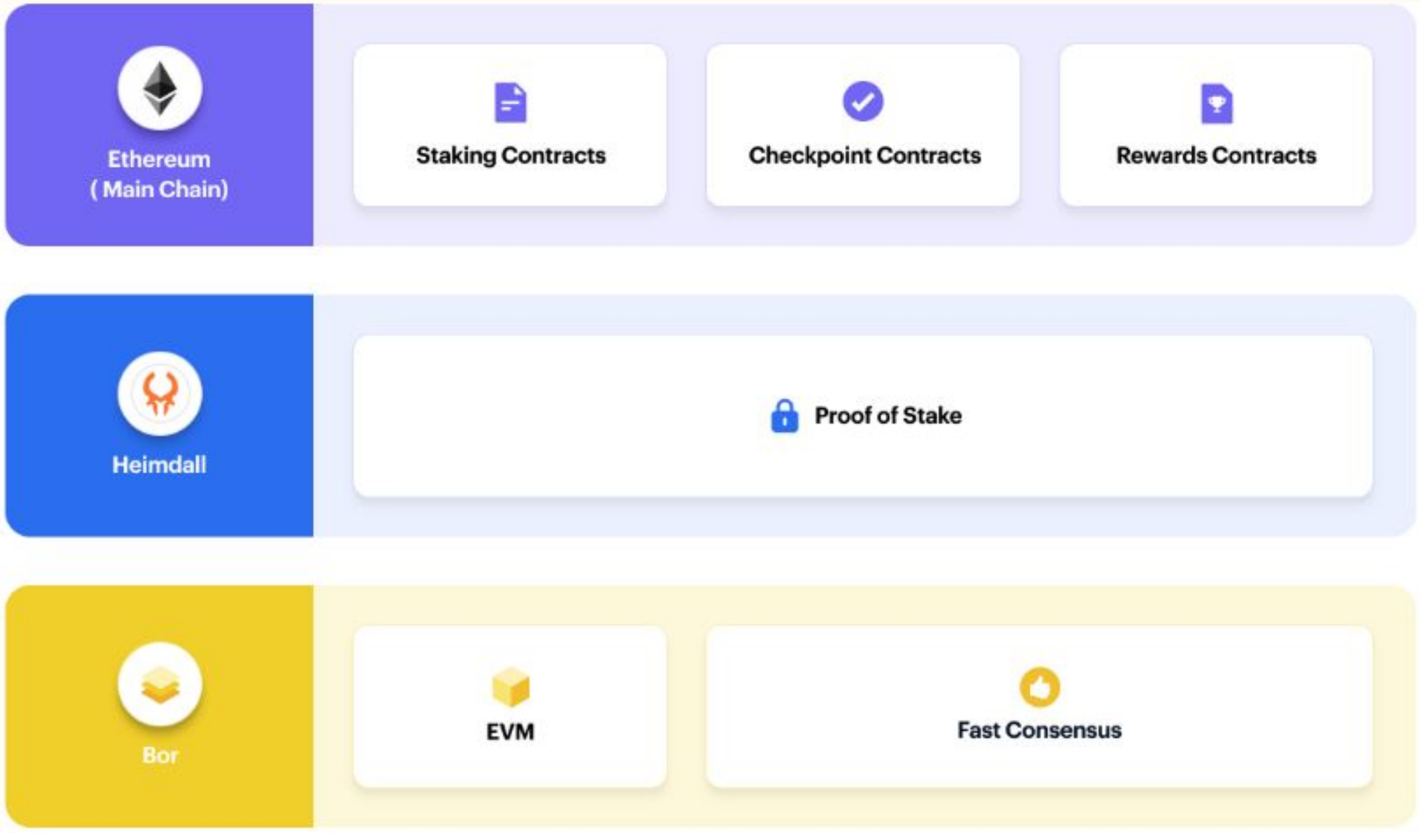
Side chains are the child of the main chain (Ethereum). They are similar to State channels but instead of a contract, a full chain is used.

Tradeoff: security vs gas fees and transaction speed.

Arbitration of a side chain can take a week vs almost instantaneous in a State Channel (challenge period).

Plasma chains differ from side chains by posting root hashes of blocks.
These blocks become checkpoints in the child chain that are verified on the parent chain

Side Chain Example - Polygon



Ethereum
(Main Chain)

Staking Contracts

Checkpoint Contracts

Rewards Contracts

Heimdall

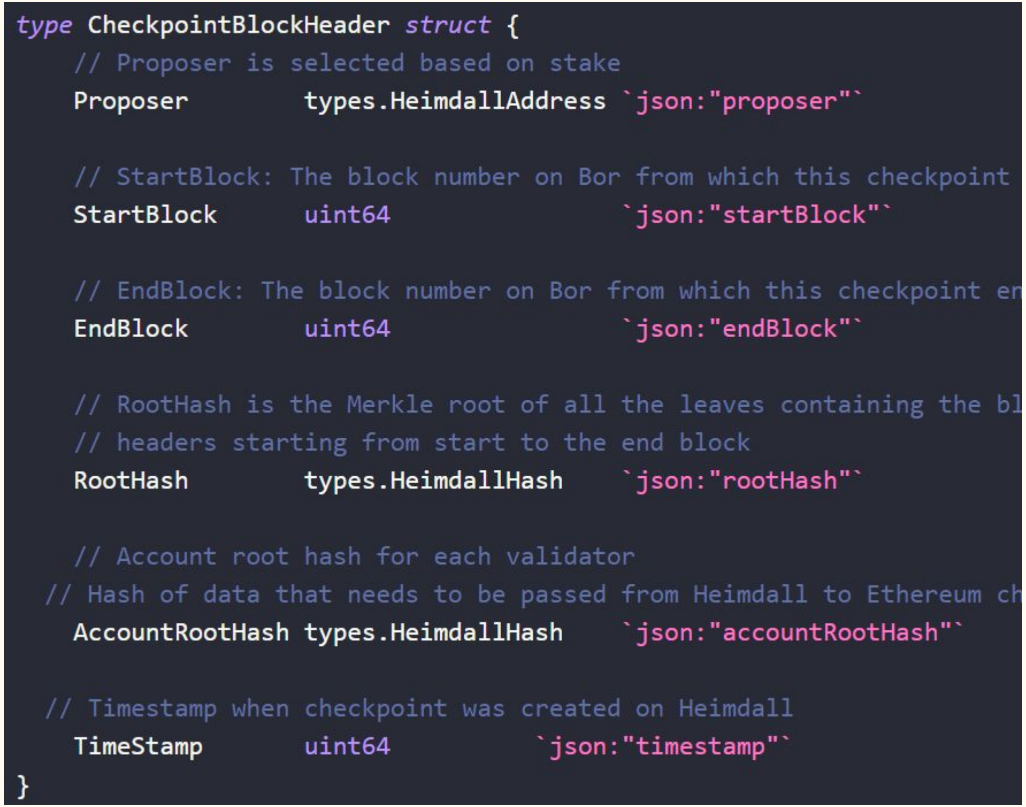
Proof of Stake

Bor

EVM

Fast Consensus

Polygon Checkpoints



```
type CheckpointBlockHeader struct {
    // Proposer is selected based on stake
    Proposer        types.HeimdallAddress `json:"proposer"`

    // StartBlock: The block number on Bor from which this checkpoint
    StartBlock      uint64                `json:"startBlock"`

    // EndBlock: The block number on Bor from which this checkpoint en
    EndBlock        uint64                `json:"endBlock"`

    // RootHash is the Merkle root of all the leaves containing the bl
    // headers starting from start to the end block
    RootHash        types.HeimdallHash    `json:"rootHash"`

    // Account root hash for each validator
  // Hash of data that needs to be passed from Heimdall to Ethereum ch
    AccountRootHash types.HeimdallHash    `json:"accountRootHash"`

  // Timestamp when checkpoint was created on Heimdall
    TimeStamp       uint64          `json:"timestamp"`
}
```

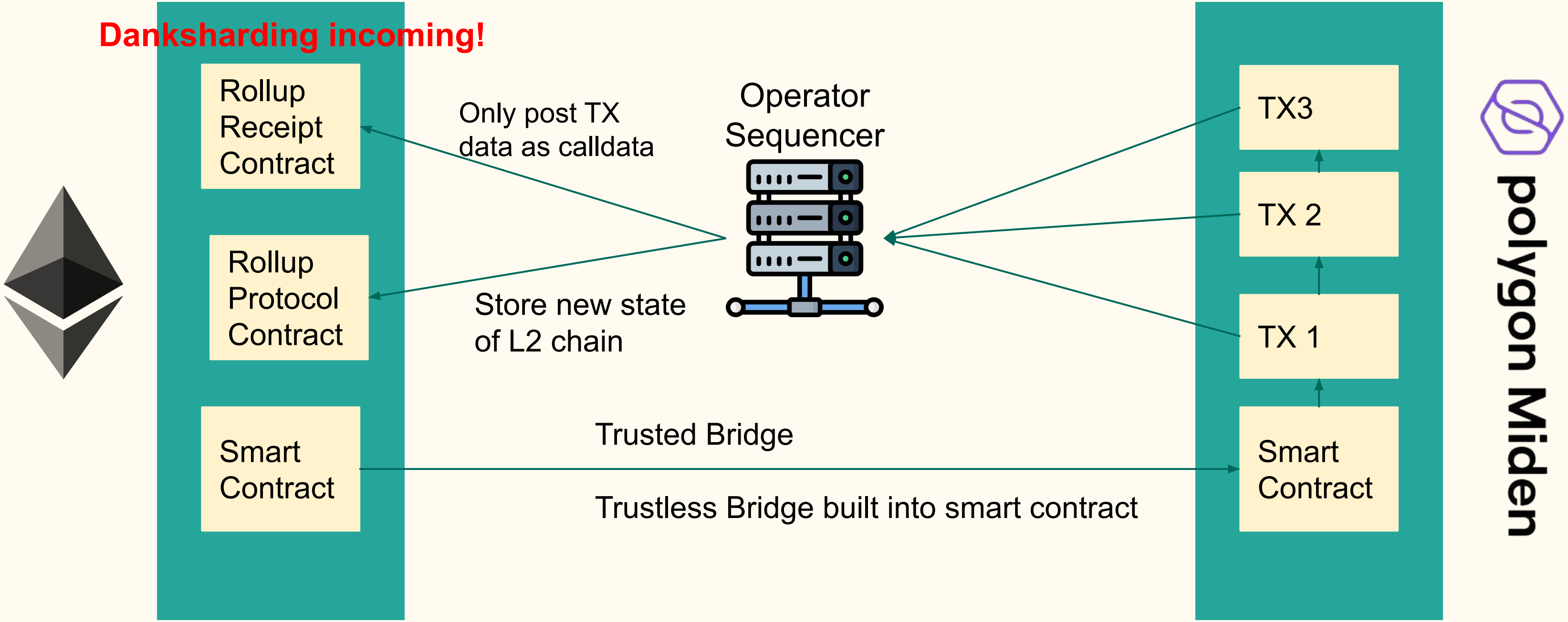
Heimdall Nodes must produce checkpoints which conform to the struct shown in code.

The RootHash hashes all blocks from start to end.

Validator group must vote on the validity of this checkpoint before it is submitted to the Checkpoint contract on Ethereum.

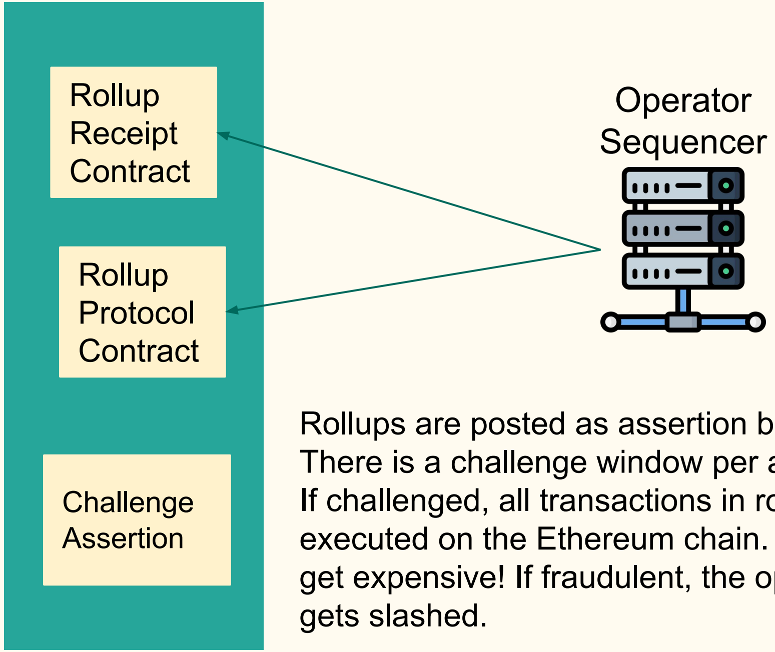
Rollups - best of both world

State Channels are only useful for limited use cases.
Side Chains sacrifice security for speed



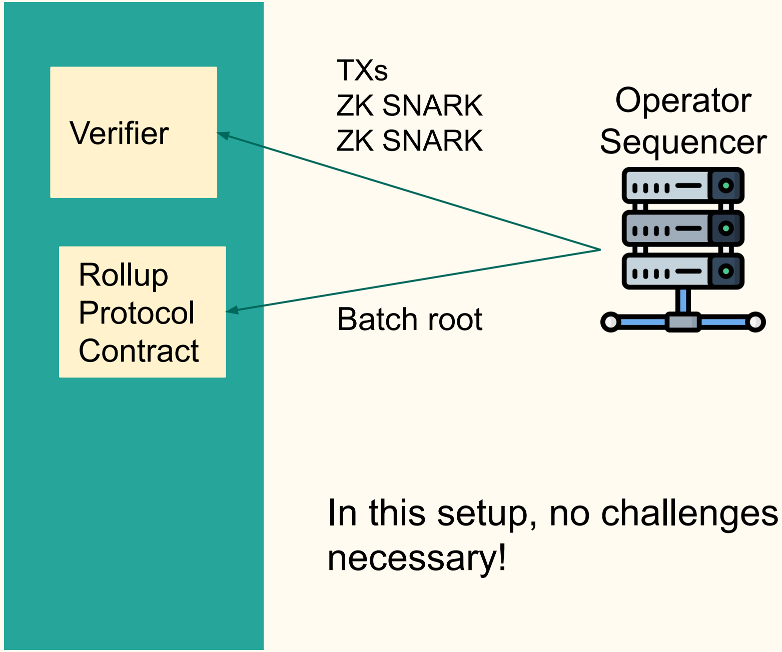
Rollups - Optimistic vs Zero Knowledge

Optimistic - Fraud Proof



Rollups are posted as assertion blocks. There is a challenge window per assertion. If challenged, all transactions in rollup gets executed on the Ethereum chain. This can get expensive! If fraudulent, the operator gets slashed.

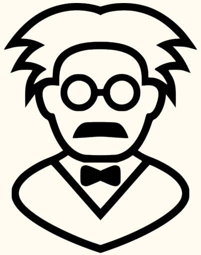
ZK- Validity Proof



In this setup, no challenges are necessary!

Zero Knowledge Proofs

ZK proof - Fun Example



We know everything about Blockchain and want an A in the class

Prove to me you have knowledge of Blockchain

We don't want to reveal what we know about Blockchain

Here is my private key to my wallet. I expect that you will hack my wallet and your wallet increases in money

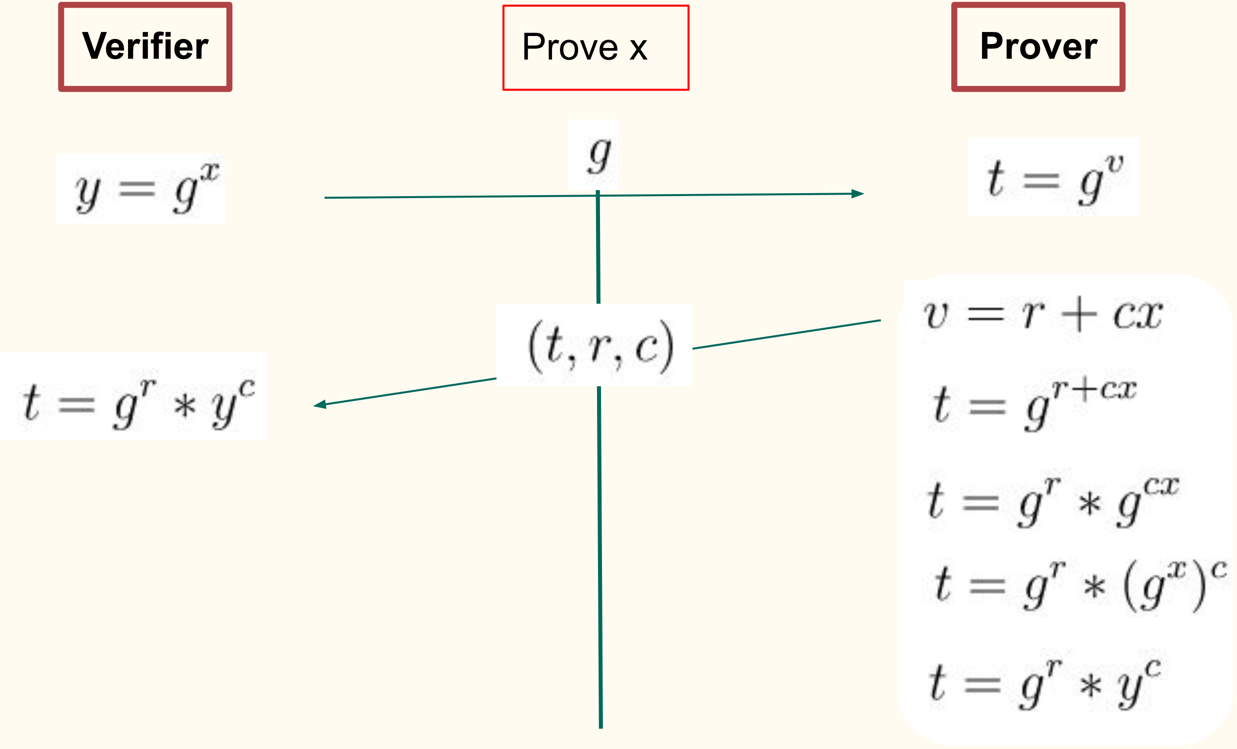
Here is the block explorer where all money from your wallet goes into mine

Okay you proved to me you know about Blockchain. You get an A.

ZK proof - Real World Examples

- Non Proliferation of Nuclear Weapons Treaty
    - US and Russia both has large nuclear stockpiles
    - How to prove they are dismantling their weapons without giving away military secrets?
- Vote Transparency
    - Everyone wants to know their vote was accurately accounted for
    - How to prove each vote without revealing identity and political affiliation?
- Confirm Transactions Off-chain
    - Confirm a bundle (rollup) of transactions and verify that they are true
    - Recursively confirm the blockchain piece by piece
    - Verify the truthfulness of the entire chain by adding pieces of little proofs

ZK Proof - Mathematical Example



Verifier

Prove x

Prover

$y = g^x$

g

$t = g^v$

(t, r, c)

$t = g^r * y^c$

$v = r + cx$

$t = g^{r+cx}$

$t = g^r * g^{cx}$

$t = g^r * (g^x)^c$

$t = g^r * y^c$