Proof of Authority Consensus Algorithms

Wrocław Blockchain Meetup #18
April 1st, 2019

Hernando Castano
Core Developer @ Parity Technologies Ltd.

hernando@parity.io | @hcastano

Outline

- Proof of * Algorithms
- Proof of Authority
  - Why PoA?
  - PoA in Ethereum
- Authority Round (AuRa)
- Clique

Icon made by Freepik from www.flaticon.com

parity

Consensus Algorithms

Intro to Consensus Algorithms

- A method of ensuring everybody agrees on the current state of the blockchain
- Want to create a single source of truth
  - Ensure everyone knows I have X amount of Eth
- Challenges include:
  - Bad actors
  - Network latency
- Lots of different algorithms exist for establishing consensus

Proof of Work

Icon made by Darius Dan from www.flaticon.com

parity

Proof of Steak

Icon made by Freepik from www.flaticon.com

parity

Proof of Stake*

Icon made by Freepik from www.flaticon.com

Proof of Authority

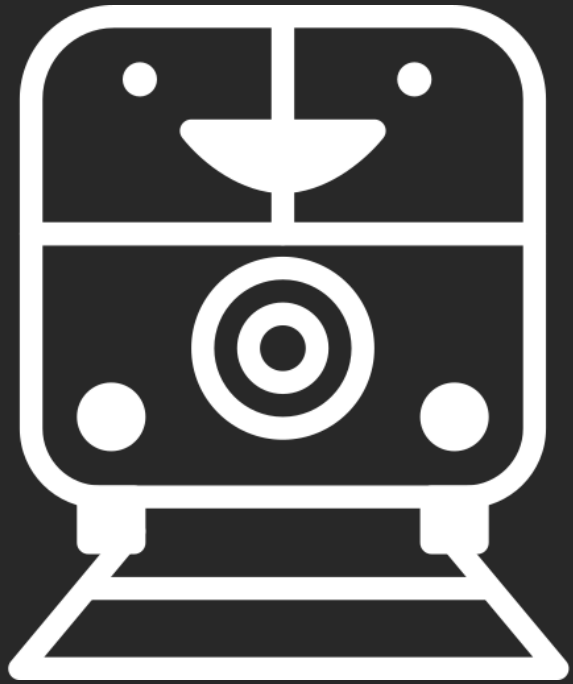
parity

Proof of Authority

- Trusted set of nodes acting as validators
- Want to produce blocks at a reliable, fixed interval
- Don’t have adversarial conditions on network
- Potentially higher performance

PoA in Ethereum

- Ropsten is a PoW testnet
- No incentive to mine meant low hash rate
  - Lead to spam attacks on Ropsten back in early 2017
- Two different PoA networks were spun up in the aftermath of the Ropsten attacks



Icon made by Darius Dan from www.flaticon.com

parity

Kovan

Authority Round

Rinkeby

Clique

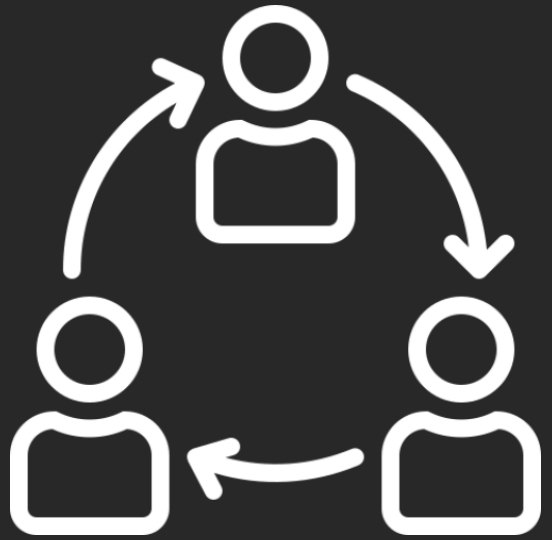
parity

Authority Round

parity

Authority Round (Aura)

- Round robin based algorithm
- Time is divided into discrete steps
- Every validator has an assigned slot to propose a block
- Configurable through a smart contract



Icon made by mynamepong from www.flaticon.com

parity

Authority Round: Idea

- Time is divided into discrete steps, where:
  - Step = Unix Time / Length of Step
    - Step 1 = 5 / 5
    - Step 20 = 100 / 5
- Each step has an assigned validator, chosen through:
  - Index = s *mod* n
    - Validators [s % n]
    - Validators [1 % 5] = Validators [1]

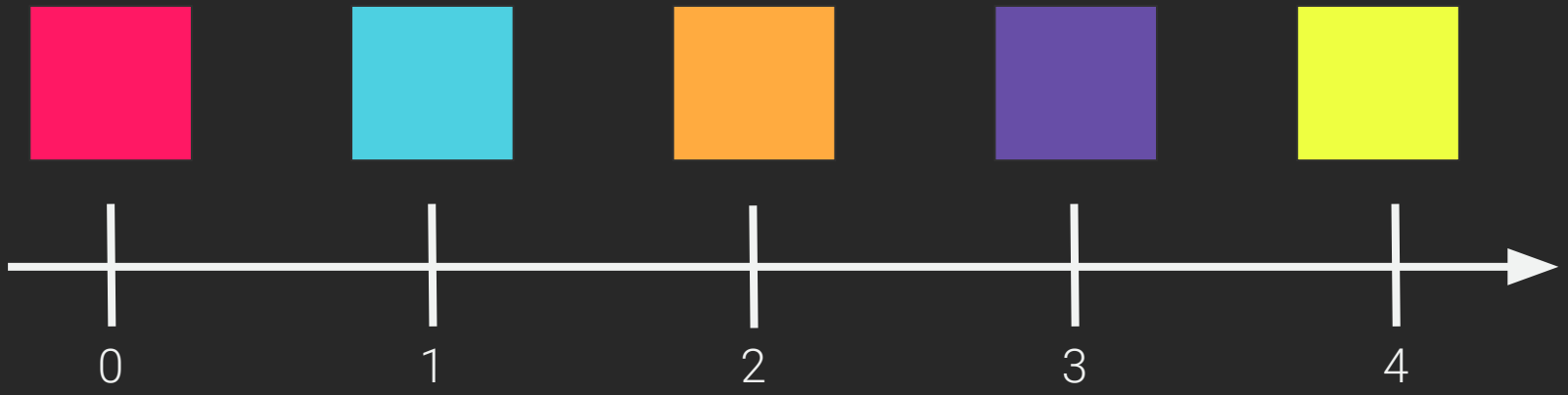
parity

Finality in Aura

- Aura provides the concept of finality
- Bitcoin and Ethereum only have probabilistic finality
  - “Might not be reverted, but let’s wait a few blocks just in case…”
- A block is finalized when more than half of the validators have built on top of it

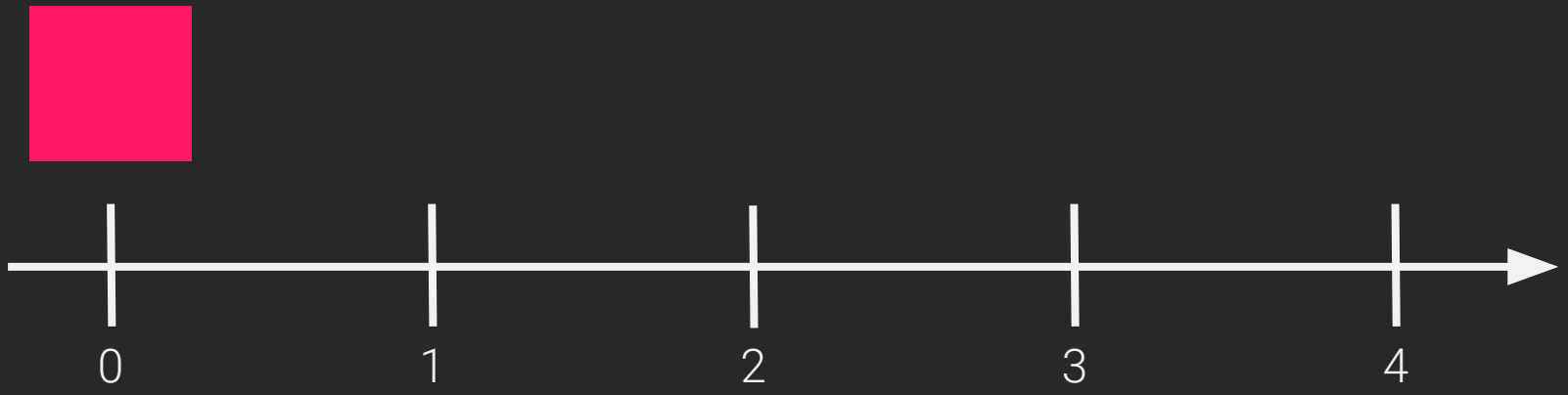
Icon made by Freepik from www.flaticon.com

parity



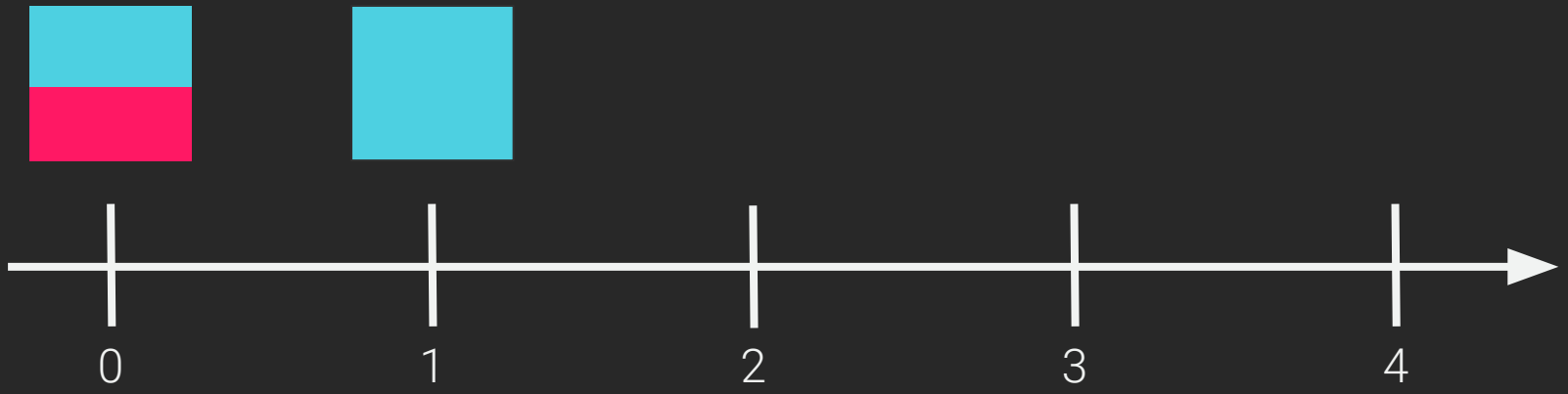
Finality in Aura

parity



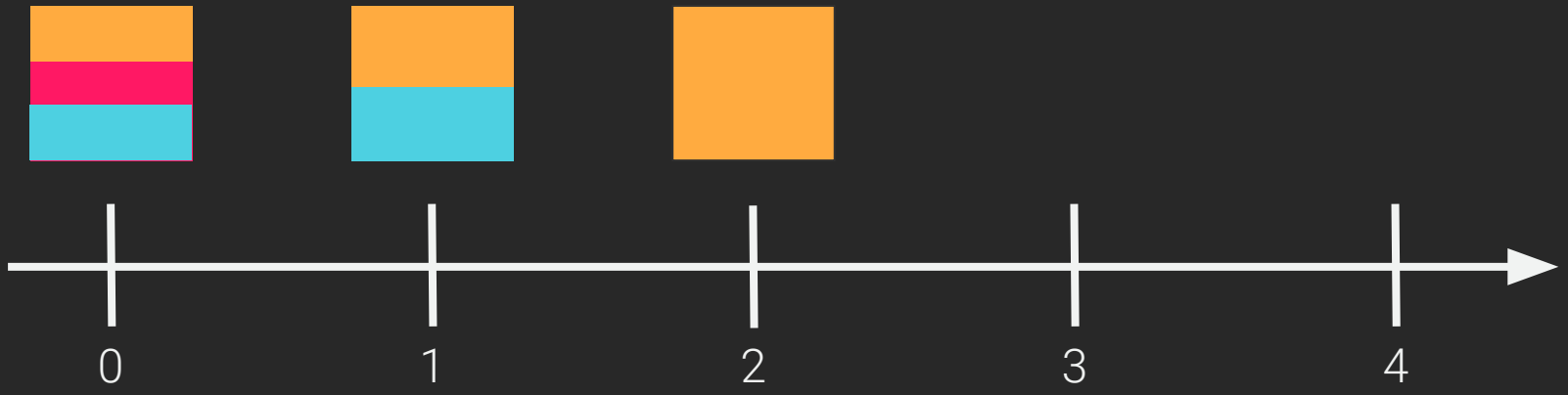
Finality in Aura

parity



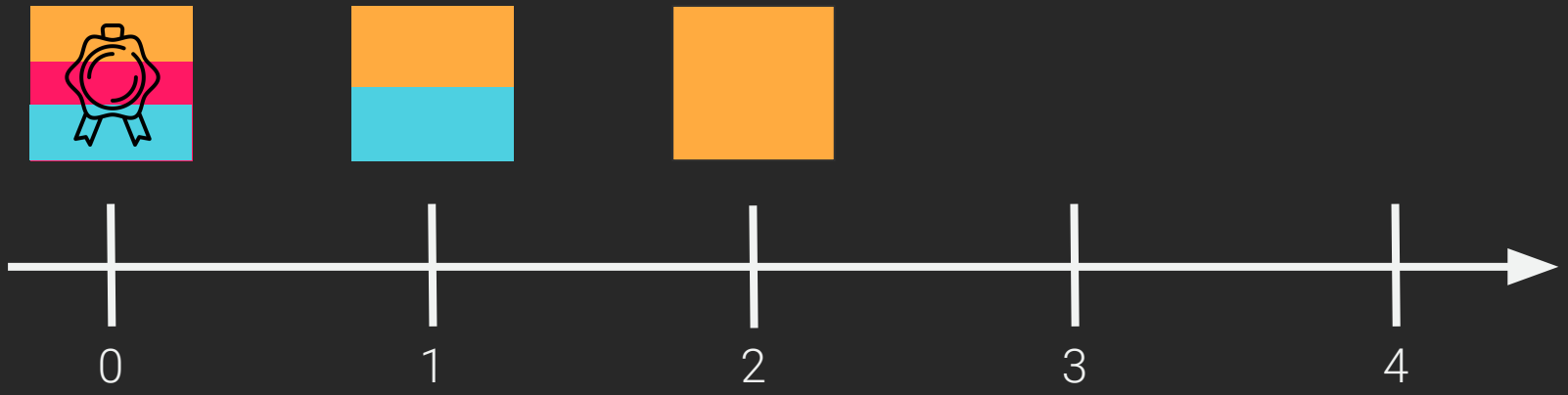
Finality in Aura

parity



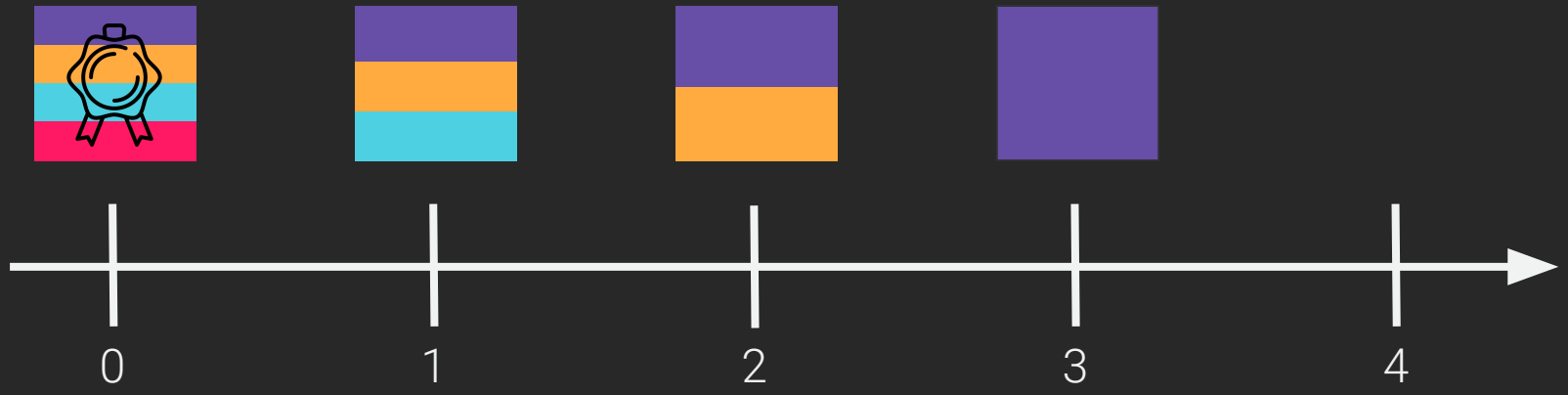
Finality in Aura

parity



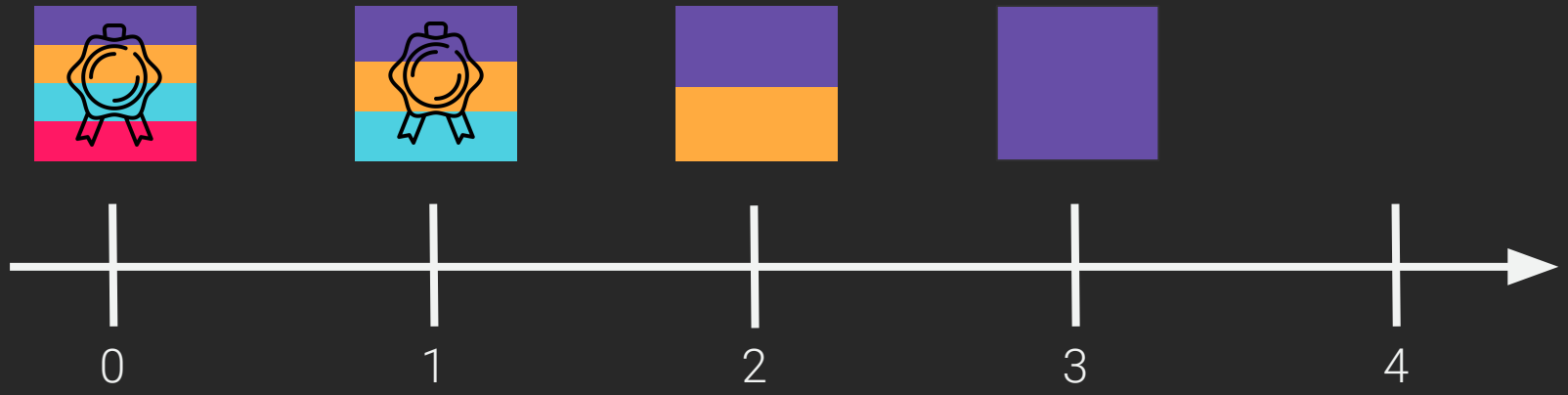
Finality in Aura

parity



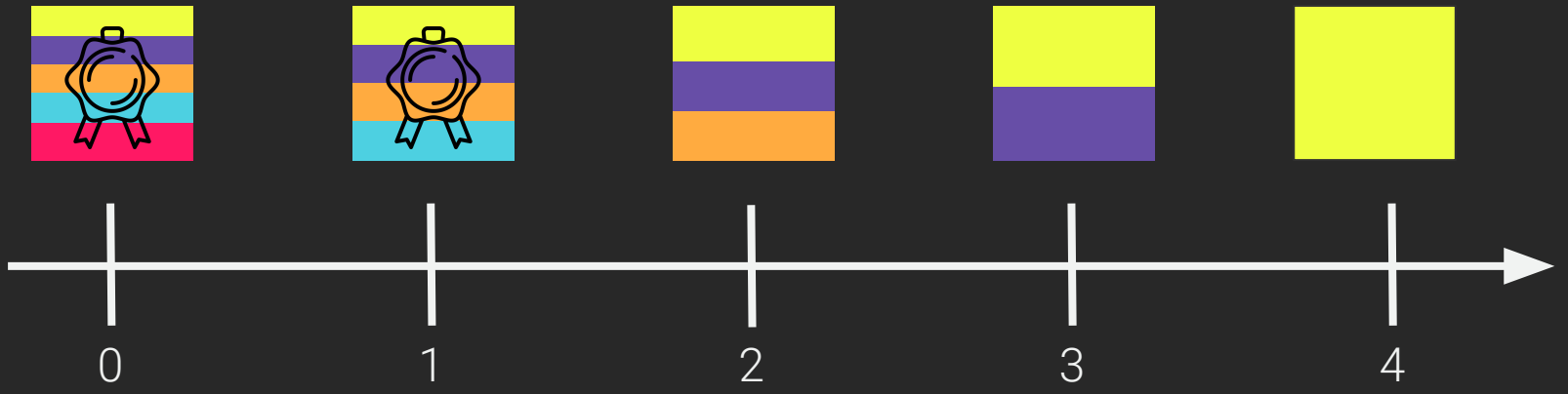
Finality in Aura

parity



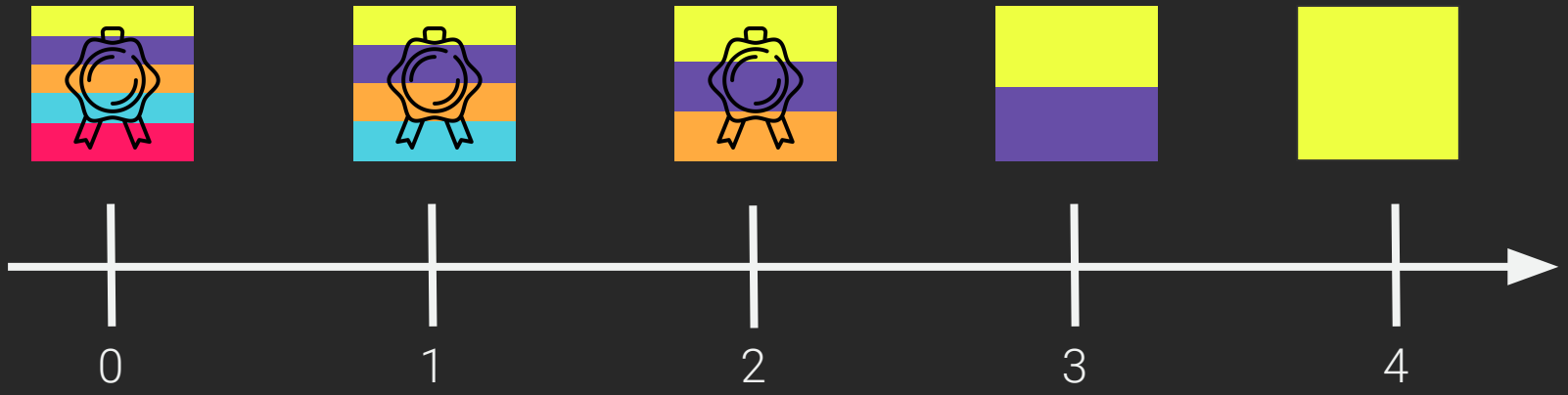
Finality in Aura

parity



Finality in Aura

parity



Finality in Aura

parity

Validator Set

- The set of validators can be defined in two ways:
  1. Chain Specification (*chain.json*)
  2. Smart Contract
- Epoch = Period of time with same validators

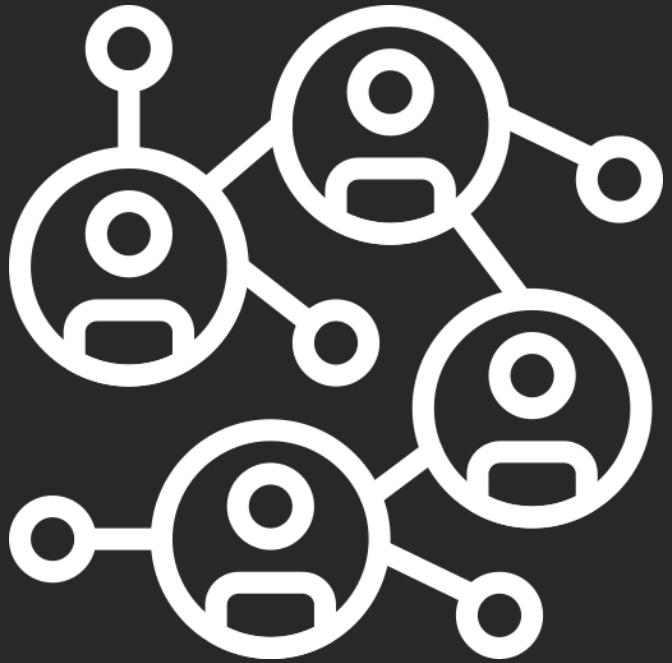
```
"validators" : {
    "multi": {
        "0": { "list": ["0xc6d9d2cd449a754c494264e1809c50e34d64562b"] },
        "10": { "list": ["0xd6d9d2cd449a754c494264e1809c50e34d64562b"] },
        "20": { "contract": "0xc6d9d2cd449a754c494264e1809c50e34d64562b" }
    }
}
```

parity

Clique

Background

- PoA consensus algorithm
- Defined in EIP-225 by Péter Szilágyi of Geth
- Also developed after Ropsten attacks
- Goal is to standardize PoA for Eth clients
- Used in Rinkeby and Goerli testnets



Icon made by prettycons from www.flaticon.com

parity

Block Proposals

- Block proposals are a little less strict than they are with Aura
  - Have concept of INTURN and NOTURN validators
- After signing a block validators are not allowed to sign the next floor(SIGNER_COUNT / 2) + 1 blocks

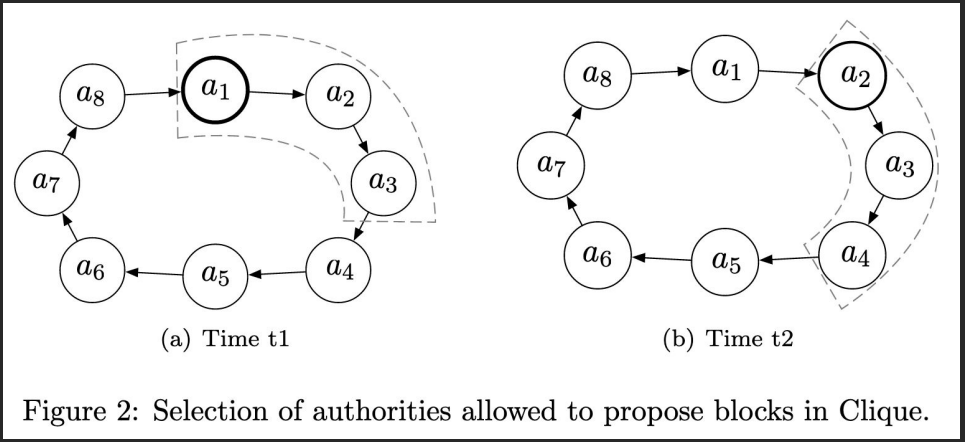


Figure 2: Selection of authorities allowed to propose blocks in Clique.

http://ceur-ws.org/Vol-2058/paper-06.pdf

parity

Choosing the Longer Chain

- Blocks have a higher difficulty depending on when they were made
  - DIFF_INTURN = 2
  - DIFF_NOTURN = 1
- Gives preference to chain built with INTURN validators

Icon made by Freepik from www.flaticon.com

parity

Validator Set Changes

- Authorities vote for validator set changes
- For non-epoch transition blocks (from the spec)
  - Signers may cast one vote per own block to propose a change to the authorization list.
  - Only the latest proposal per target beneficiary is kept from a single signer.
  - Votes are tallied live as the chain progresses (concurrent proposals allowed).
  - Proposals reaching majority consensus SIGNER_LIMIT come into effect immediately.
  - Invalid proposals are not to be penalized for client implementation simplicity.

Epochs

- Smart contracts require state access
  - Need way to verify authorities without access to state
- An epoch block is a stateless transition
  - Contain no votes
  - Contain list of current authorities
  - All non-settled votes are discarded
- Can be used as a checkpoint for clients syncing the network
- Default is every 30,000 blocks

Pop Quiz!

```
signers: []string{"A", "B", "C"},
blocks:  []block{
  {signer: "A", voted: "C", auth: false},
  {signer: "B", voted: "C", auth: false},
},
```

Pop Quiz: Consensus Tests!

parity

```
// Three signers, two of them deciding to drop the third
signers: []string{"A", "B", "C"},
blocks:  []block{
  {signer: "A", voted: "C", auth: false},
  {signer: "B", voted: "C", auth: false},
},
results: []string{"A", "B"},
```

Pop Quiz: Consensus Tests!

parity

```
signers: []string{"A"},
blocks:  []block{
  {signer: "A", voted: "B", auth: true},
  {signer: "B"},
  {signer: "A", voted: "C", auth: true},
},
```

Pop Quiz: Consensus Tests!

parity

```
    // Single signer, voting to add two others (only
accept first, second needs 2 votes)
    signers: []string{"A"},
    blocks:  []block{
      {signer: "A", voted: "B", auth: true},
      {signer: "B"},
      {signer: "A", voted: "C", auth: true},
    },
    results: []string{"A", "B"},
```

Pop Quiz: Consensus Tests!

parity

```
signers: []string{"A", "B", "C", "D"},
blocks:  []block{
  {signer: "A", voted: "C", auth: false},
  {signer: "B", voted: "C", auth: false},
},
```

Pop Quiz: Consensus Tests!

parity

```
    // Four signers, consensus of two not being
enough to drop anyone
    signers: []string{"A", "B", "C", "D"},
    blocks:  []block{
      {signer: "A", voted: "C", auth: false},
      {signer: "B", voted: "C", auth: false},
    },
    results: []string{"A", "B", "C", "D"},
```

Pop Quiz: Consensus Tests!

parity

```
epoch:   3,
signers: []string{"A", "B"},
blocks:  []block{
  {signer: "A", voted: "C", auth: true},
  {signer: "B"},
  {signer: "A", checkpoint: []string{"A", "B"}},
  {signer: "B", voted: "C", auth: true},
},
```

Pop Quiz: Consensus Tests!

parity

```
    // Epoch transitions reset all votes to allow chain
checkpointing
    epoch:   3,
    signers: []string{"A", "B"},
    blocks:  []block{
      {signer: "A", voted: "C", auth: true},
      {signer: "B"},
      {signer: "A", checkpoint: []string{"A", "B"}},
      {signer: "B", voted: "C", auth: true},
    },
    results: []string{"A", "B"},
```

Pop Quiz: Consensus Tests!

parity

Questions?

hernando@parity.io

 @hcastano

parity