



PROOF OF AUTHORITY

...

Gokul G
CB.EN.P2CYS20017

Agenda

1. What is PoA
2. Consensus Comparison(why PoA)
3. History of PoA
4. AuRa
5. Clique
6. Examples

What is PoA

- Consensus mechanism where network participants stake their identity and reputation
- Suitable for permissioned blockchains

Consensus Comparison



PoW:

- Computers compete against each other to find solution of digital puzzle
- Incentive: Upkeep >> profit of forged reward
- Disadvantage:
 - Huge upkeep cost
 - GPU, ASICS needed
 - Environmental pollution
 - Electricity, water costs
 - Probabilistic finality
 - Depends on difficulty level
 - Whether current transaction maybe in a fork/bad block



PoS:

- Stake some funds, larger the fund higher the chance of becoming validator
- If you lie then you lose stake
- Incentive: stake >> profit of forged reward
- Comparing to PoW:
 - Not much computation required
 - Less pollution
 - More democratic
- Disadvantage:
 - Assumes that bigger the numerical value, less chance staker will misbehave
 - Ignores what percentage of his money has the staker put

Consensus Comparison

PoA:

- Better than PoW as
 - Not much computation costs
 - No need of mining blocks
 - Not much environmental problems
 - Deterministic calculation of when next block is created
- Better than PoS as
 - Removes monetary incentives for block creation
 - Removes concept of gas fees
 - Deterministic calculation of when next block is created



PoA is:

- Used in permissioned blockchains
 - Offers consensus with few stakeholder trusted validators
 - Stringent KYC
 - Identities registered in public notary databases
 - Limits possibilities of bad actors
 - High speed of consensus
 - More scalable than PoW, PoS
 - Validators independent of other participants
-
- Centralized
 - More geared towards enterprise
 - By design exposing identity of validators
 - By design capable of censorship/blacklisting
 - Vulnerable to validator compromise leading to loss of integrity of Blockchain.
 - Implicit trust has to be maintained
 - Ex: CoCo framework from microsoft uses TEE



History of PoA

- Ropsten DoS attack (2017)
 - Is PoW network, with low hash rate
 - Attacker
 - Amassed huge number of test ethers
 - Send flood of transactions with very high gas price
 - Caused DoS on Ropsten
- Reaction
 - Creation of Kovan
 - Developed by parity
 - Uses Authority Round (AuRa)
 - Creation of Rinkeby
 - Developed by Geth devs
 - Uses clique

Ethereum Testnet Has Suffered A Spam-Attack



Ethereum network has suffered a spam attack. As noted, the failures in the Ropsten network began when testing new software. The Ethereum developers plan to activate it during the upcoming hardfork called Byzantium. At the same time, they argue that attackers are unlikely to be able to influence the timing of the activation of hardfork and there are no serious reasons for concern.

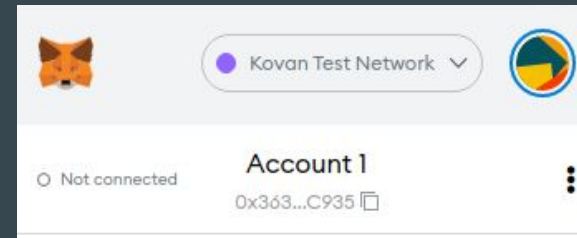
The spam-attack has pushed developers to move into a more secure network Rinkeby. But according to the developer of Raiden, Lefteris Karapetsas, network Ropsten is the closest analogue of the original Ethereum blockchain.

Spam-attack the Ropsten

The attackers have filled the Ropsten network with thousands of automatically generated spam transactions. The last such attack happened in March, and then the Ethereum team has forced to re-write a test version of the blockchain.

The team countered the attacks due to the increase in the cost of transactions, but this step adversely affected the possibilities of further testing the software. Subsequently, the developers stopped for a time to operate testnet Ropsten.

Aura



- Block proposal
 - Validator chosen by round robin algorithm
 - Real time(hence needs to be synced by PTP or similar) is divided into slots
 - Each validator is given slot to propose block.
 - If current validator not responding move to next one

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

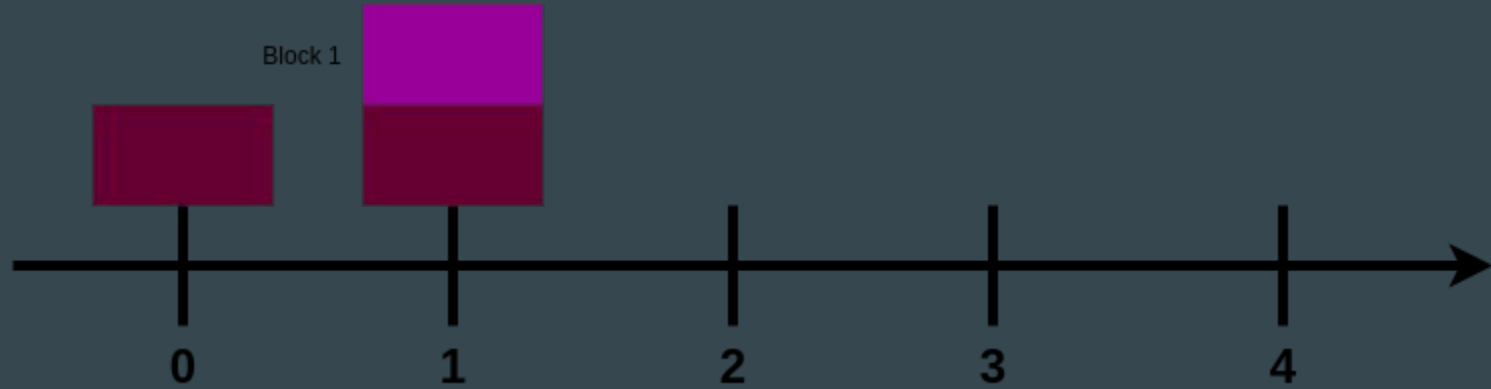
Aura

- Finality in AuRa
 - Block is finalized when more than 50% of validators have confirmed blocks on top of it



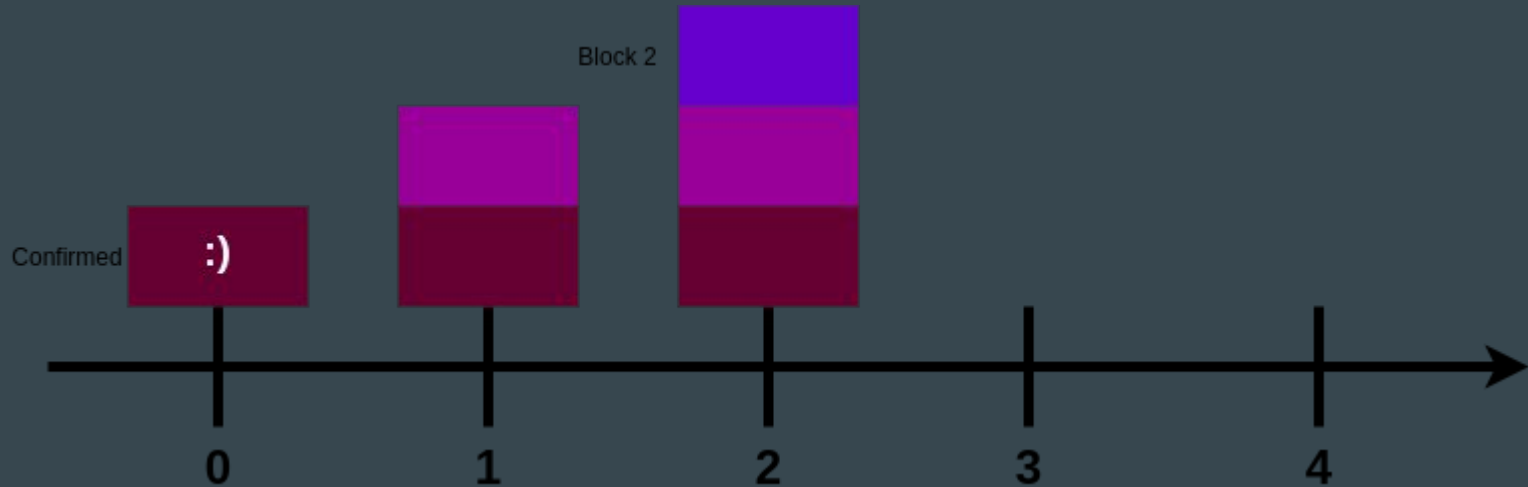
Aura

- Finality in AuRa
 - Block is finalized when more than 50% of validators have confirmed blocks on top of it



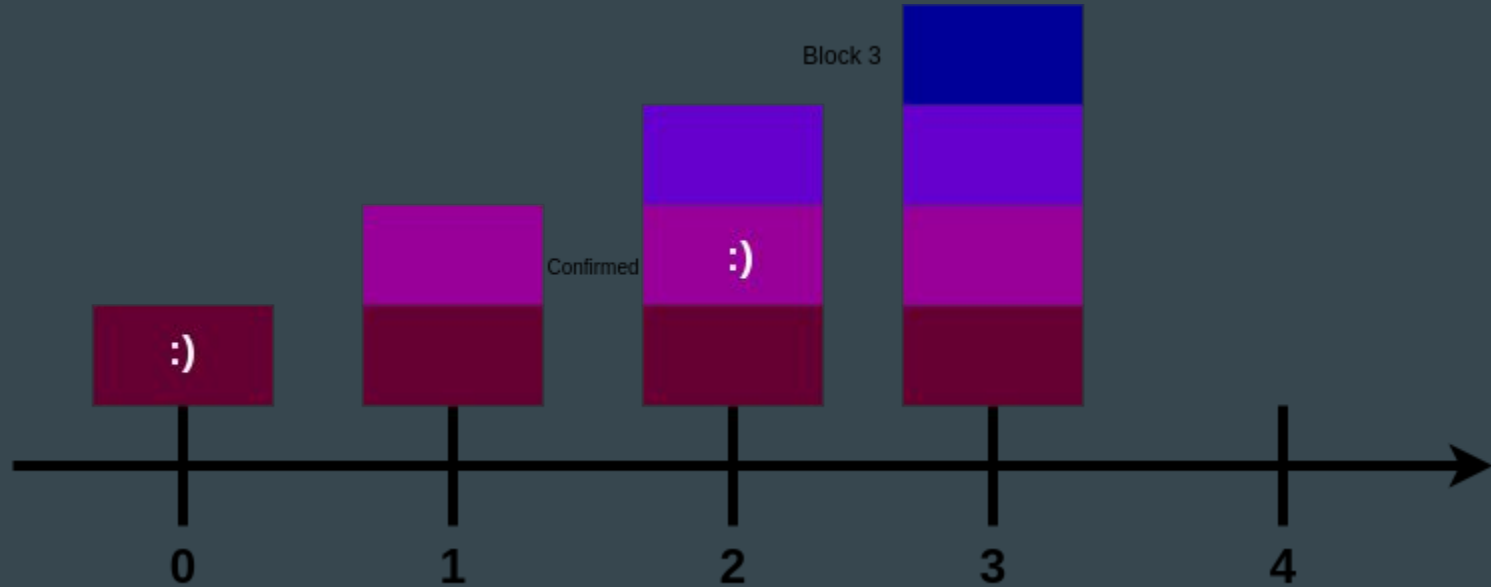
Aura

- Finality in AuRa
 - Block is finalized when more than 50% of validators have confirmed blocks on top of it



Aura

- Finality in AuRa
 - Block is finalized when more than 50% of validators have confirmed blocks on top of it



Aura

- Finality in Bitcoin blockchain
 - to avoid dropped transaction due to bitcoin softforks
 - wait for 6 blocks on average is 60 mins
- Finality in ETH
 - Wait is 1 min on average for one confirmation
 - Need 50 confirmations for larger transactions
- Whereas finality in PoA
 - **5 secs per block** on average



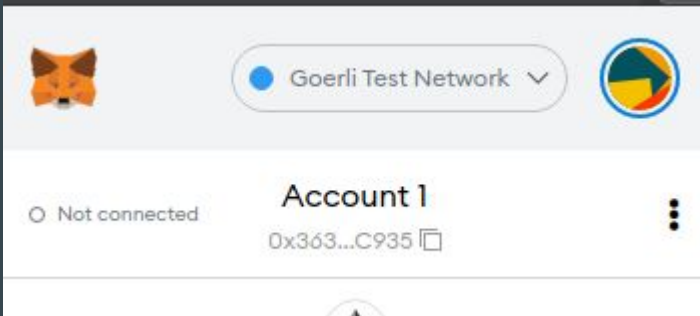
Aura

- Validator set changes
 - Hard code at chain level
 - In chain.json
 - If need to change validator set, do hard fork
 - Automate changes using smart contract
 - Decide easily who gets to be next validator without forking
 - Block 0-10, one set of static validator, 10-20 another set of validator, from 20 let the SC chose.

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

Clique

- Defined in EIP-225
- Used in Rinkeby and Goerli testnets
- Block proposals
 - Validators are either IN_TURN or NO_TURN
 - IN_TURN -> more weight or priority when it comes to proposing a block
 - NO_TURN -> less weight but still can propose block
 - After proposal, validator have to a predetermined time until it allowed to propose again to prevent dominance of one validator
 - Resolving softforks
 - Prefer chain with more net difficulty score
 - gives preference to chains built with IN_TURN validators
 - Difficulty of IN_TURN = 2
 - Difficulty of NO_TURN = 1



Clique

- Validator set changes
 - Smart contract
 - Let the other validators decide
 - Per block, validator can vote who can approve the block
 - Only one vote per validator

- Used in testnets to deter DoS attacks
- Used in Azure platform

Cody [\(Update\)](#)

Current Ethereum Account: 0x07239c2ab5f3640aca9d976a97dc1b6e5b57b75c

Administrators	Validators	Node Status
Current Admins		
	Alias	
0x07239c2ab5f3640aca9d976a97dc1b6e5b57b75c	Cody (You)	<button>VOTE AGAINST</button>
0xeb64d1efc1fbc4e94a8cfd3e50b8e5c0c1d7d	Stephen	<button>VOTE AGAINST</button>
0x55c01bdae5e25863252bd0353363d9fcb1bd8f7	Estifanos	<button>VOTE AGAINST</button>
0xt22210cd5930f0d194bf38e52d80f3c02d5c4743	Daehee	<button>VOTE AGAINST</button>
<input type="text" value="Ex: 0x17Bf5e7b3CE6779DBaeDEB907010601A8c1e3118"/>	<input type="text" value="Ex: Admin 2"/>	<button>PROPOSE</button>

The screenshot displays the 'Overview' page of a blockchain dashboard, specifically focusing on 'Network Stats'. The top navigation bar includes links for 'Est' and 'Clone'. A 'Last 4 hours' filter is active.

Block Overview

Block	Time (Sec)
6837	2
6834	2
6833	2
6832	2
6831	2
6830	2
6829	2
6828	2
6827	2
6826	2

[See all...](#)

Nodes

Node	Blocks Reported Last Hour
10.0.173.8545	1.7K
10.0.174.8545	1.7K

[See all...](#)

Validator Overview

- RPC Requests/sec:** Line graph showing average RPC requests per second from 2:00 PM to 4:00 PM. The y-axis ranges from 30 to 250. There is a significant spike around 2:30 PM reaching approximately 250 requests/sec.
- Transactions:** Line graph showing transactions per minute over the same period. The y-axis ranges from 0.5 to 2. There are several sharp peaks, notably one near 2:00 PM and another around 3:30 PM.
- Gas Used:** Line graph showing gas used every minute. The y-axis ranges from 0M to 3M. It shows very high initial usage peaking at nearly 3M gas units around 2:00 PM, followed by lower, more sporadic usage throughout the afternoon.

Examples

- In hyperledger Besu

