ethereum

A presentation on Ethereum network upgrades

By:

Parithosh Jayanthi

Disclaimers

About myself

- MSc. Communications Engineering at Technical University of Munich
- Currently working as a DevOps Engineer at the Ethereum Foundation
- Tasks: Maintain, automate and manage testnets and help out with #TestingTheMerge
- Fun fact: I've helped setup a data center deep inside a mountain

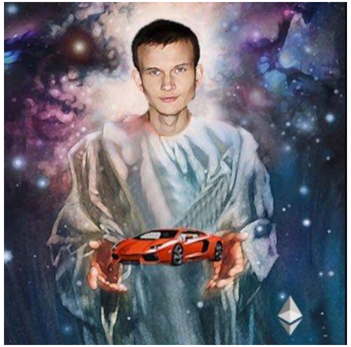
@parithosh_j

parithosh

Content

- Intro to Proof of Work
- Intro to Proof of Stake
- PoW vs PoS
- What is the merge?
- Benefits of the merge
- Future upgrades
- How you can help

What is Ethereum?



- Decentralized, open-sourced blockchain launched in 2015
- The Ethereum blockchain contains a state machine called the Ethereum Virtual Machine (EVM)
- The EVM specifies the execution model for state changes: Enabling programs to be written
- Code execution costs gas, which is paid in Ether -> Prevents individuals from spamming the network

What is Proof of Work?

- Proof of Work is a mechanism that allows nodes to come to consensus
- Mining is the process of creating a block of transactions to be added to the blockchain
- The underlying algorithm sets the difficulty and rules for the miners
- The “work” done by miners is hashing to satisfy the rules of the consensus algorithm

Pro:

- Relatively easy to implement
- Proven technology
- Easier to fork away if you disagree with the main chain

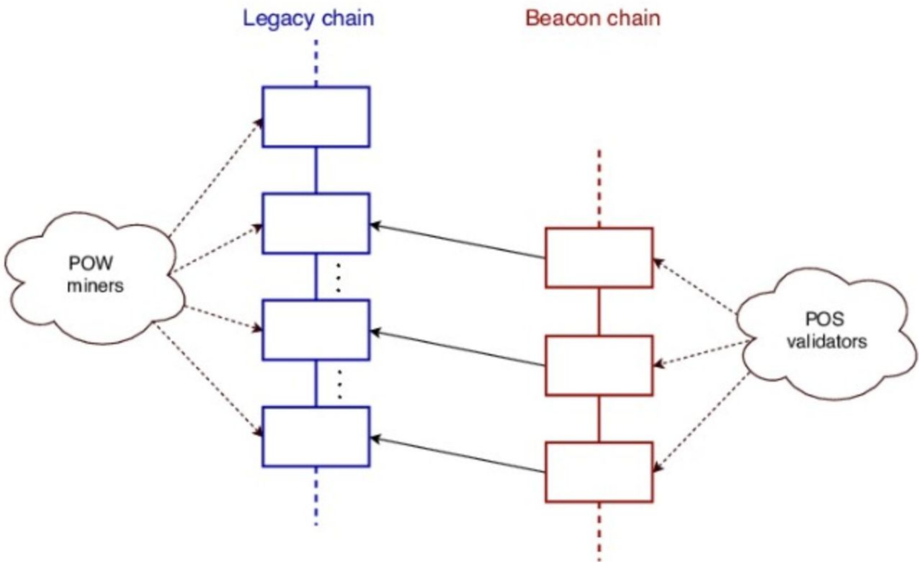
What is Proof of Stake(PoS)?

- PoS is a consensus algorithm, Ethereum PoS is a bit different from others
- PoS replaces the importance of hash power with Ether
- PoS replaces miners with validators
- To become a validator you must stake (lock) Ether in a smart contract
- You cannot delegate your stage in the Eth base layer! => You deposit ether & run your own validator
- Validators receive fees for *correct* participation
- Validators loose their stake for *malicious* actions
- What is *correct* and *malicious* is decided by the 2/3rds majority of validators
- To successfully attack the blockchain, one must control more than 2/3rds of the validators => 2/3rds of all the Ether locked up
- => Harder to attack, uses ~99.95% less energy, higher throughput, faster!

Eth2 phase 0

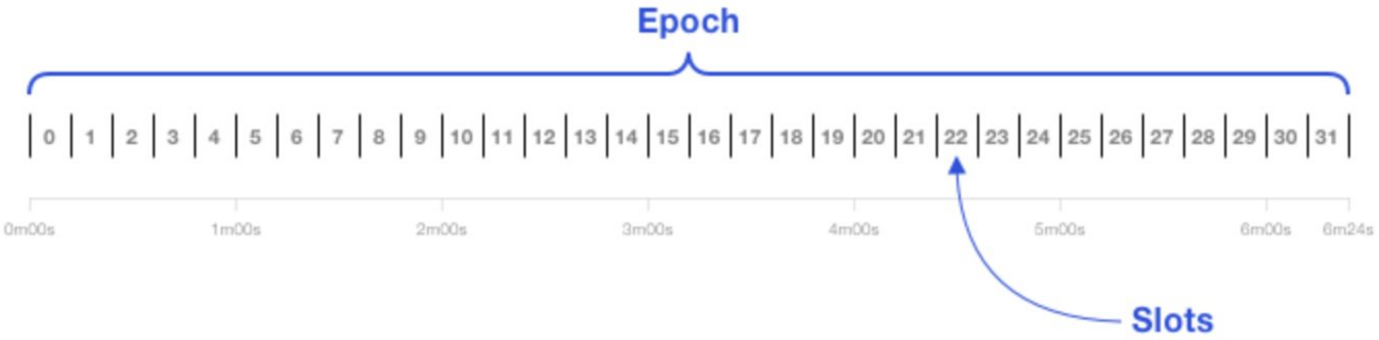
Enter, the Beacon Chain

- Phase 0 introduces the *Beacon chain*
- The *Beacon chain* uses PoS to achieve consensus
- The *Beacon chain* handles:
    - Assigning duties to validators
    - Finalization of the chain
    - Stores attestations
    - State of validators



How do blocks look then?

- *Beacon chain* has Slots and Epochs
- 1 Slot = 12 seconds, 1 Epoch = 32 Slots (6.4 minutes)
- Validators *DO* need to be roughly synced with NTP(network time protocol) servers
- Finality* in most cases can occur in a deterministic amount of time (2 epochs)



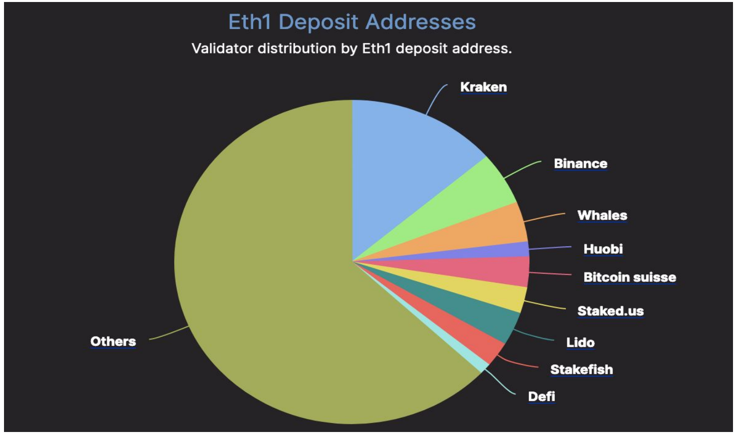
***Finality** is the assurance or guarantee that **cryptocurrency** transactions cannot be altered

What all do the validators do?

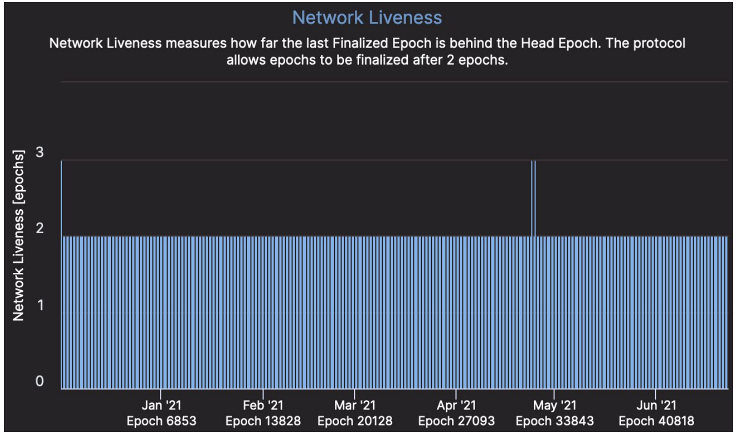
- A validator can:
  - Attest a block: Sign that it agrees with the information in the block
  - Propose a block: Produces the block that is attested by other validators
  - Publish proof of malicious intent (double attestation of conflicting states) to other validators

How well does it work?

- Eth2 will ship in phases, Phase 0 is already live!
- Currently >400,000 validators participating => each with 32 Eth staked
- >12,400,000 Ether ($13B USD) locked in the eth2 staking contract
- >99.5% of all validators are up and validating
- Phase 0 proves PoS consensus can work => But it has no transactions or EVM execution


Eth1 Deposit Addresses

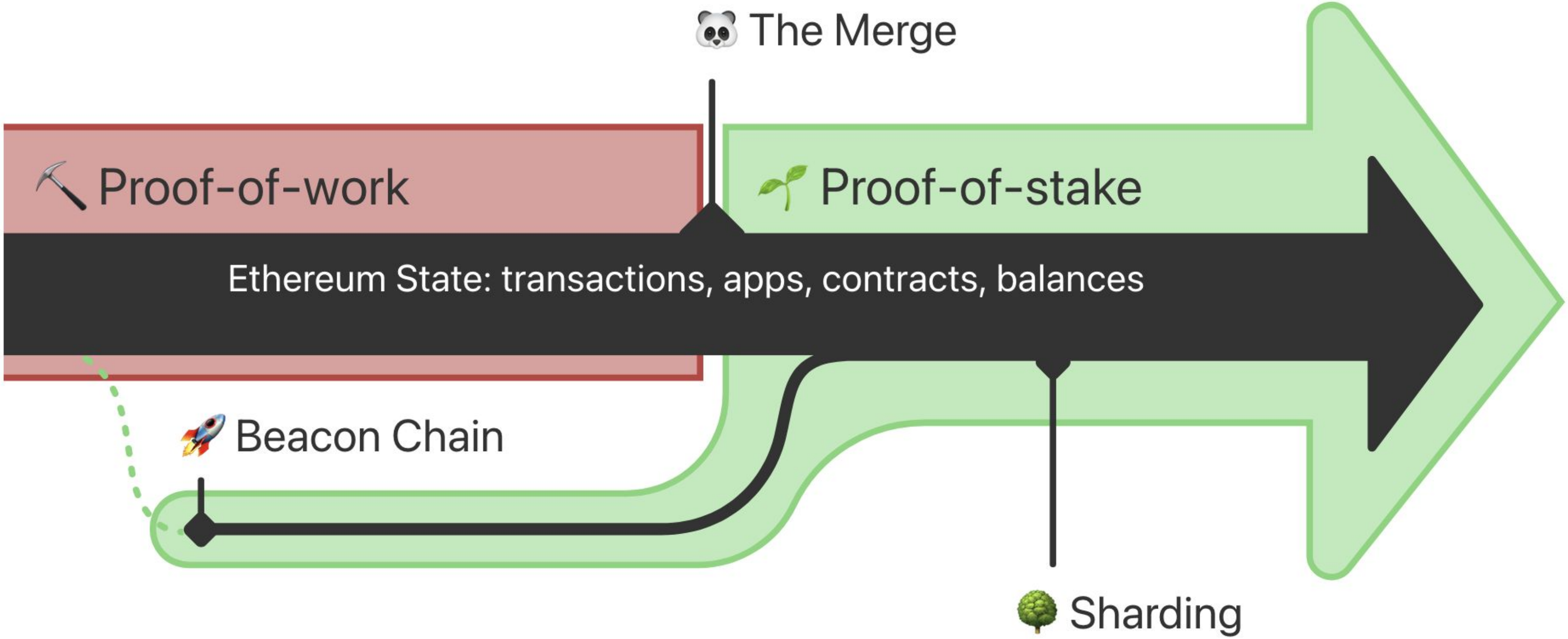
Validator distribution by Eth1 deposit address.


Network Liveness

Network Liveness measures how far the last Finalized Epoch is behind the Head Epoch. The protocol allows epochs to be finalized after 2 epochs.

PoW	PoS
Easier to bootstrap in a decentralized manner	Harder to bootstrap a decentralized network
Large, unbounded energy use	Extremely low energy use
Requires high constant issuance to maintain security	Requires a far lower issuance to maintain security
51% attacks can be retried if they fail	Attackers gets slashed => new attack requires new stake
Realistically, hashrate centralizes into pools	Stake centralizes into staking pools, but solo stakers can deterministically produce blocks
Centralized pools can censor the network	Even a small minority of validators can deterministically propose non-censored blocks
Harder to implement sharding on PoW - Scalability	Lays groundwork for Sharding, Data availability sampling(DAS) - Scalability
Light clients are hard to implement on Ethereum PoW	Ethereum PoS's sync committee's make light clients easier to implement

What is the merge?



🐼 The Merge

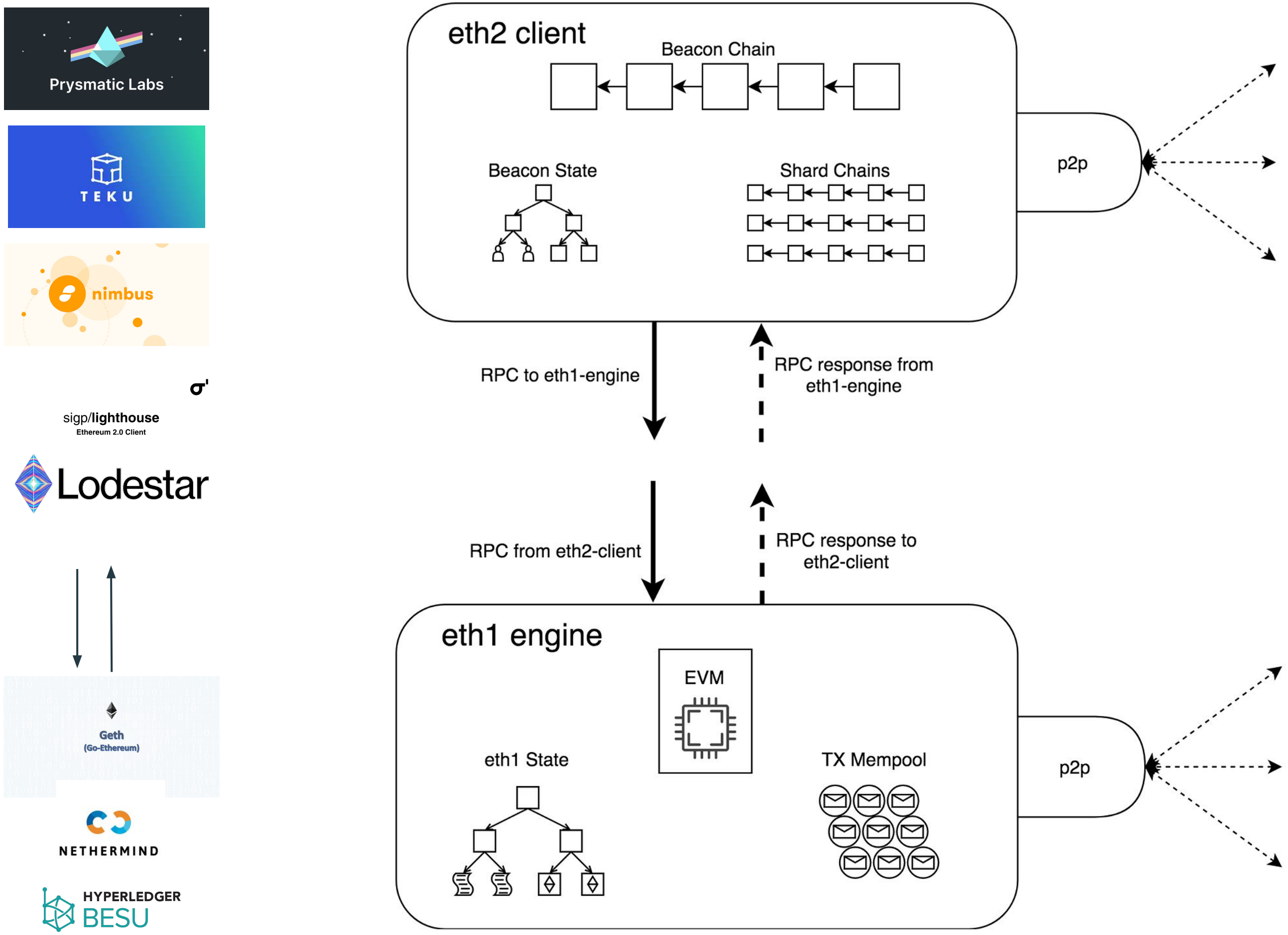
⛏️ Proof-of-work

🌱 Proof-of-stake

Ethereum State: transactions, apps, contracts, balances

🚀 Beacon Chain

🌳 Sharding



eth2 client

Beacon Chain

Beacon State

Shard Chains

p2p

RPC to eth1-engine

RPC response from eth1-engine

RPC from eth2-client

RPC response to eth2-client

eth1 engine

EVM

eth1 State

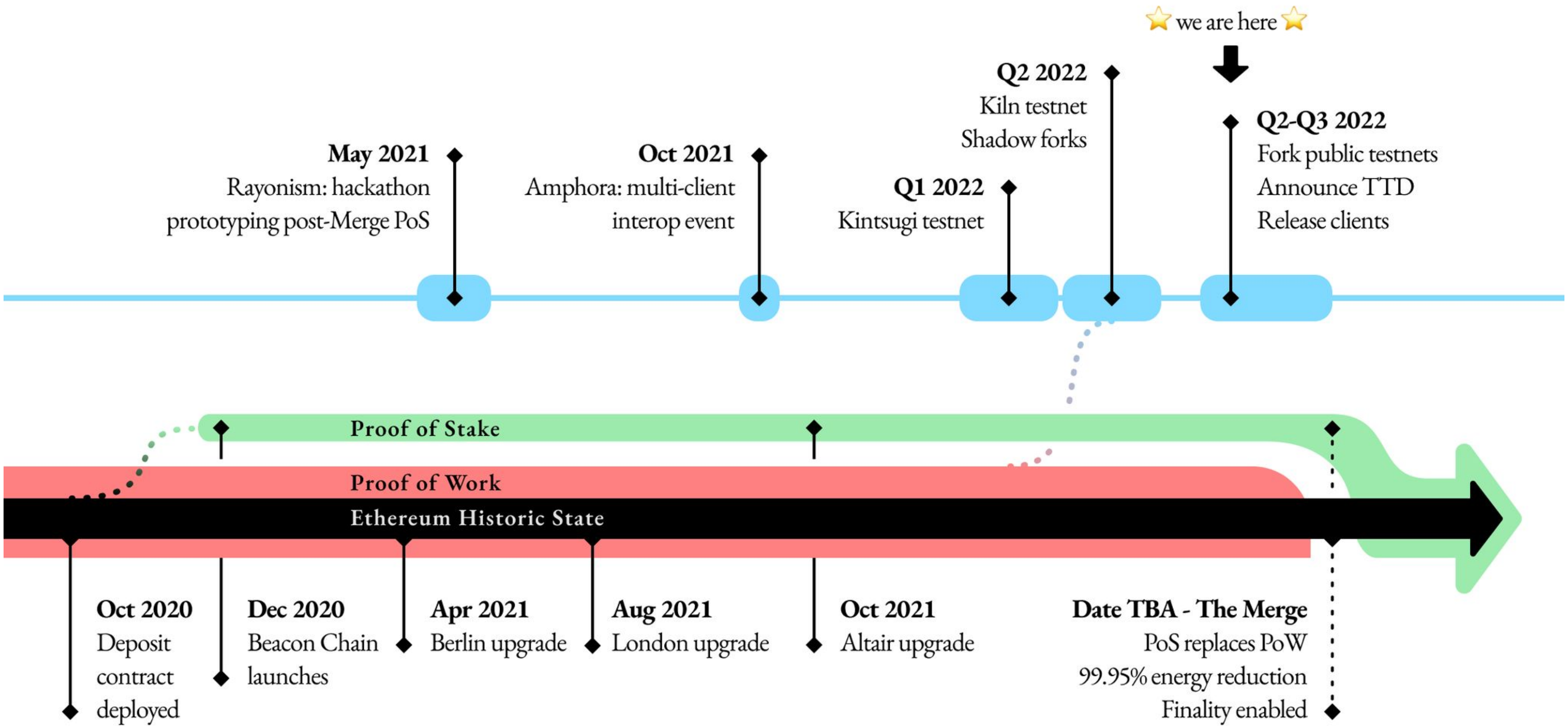
TX Mempool

p2p

The Path to the Merge

🟦 Offchain 🟩🟥 Onchain

May 27 2022 // @trent_vanepps // pixels
between milestones do not scale to reality



May 2021
Rayonism: hackathon
prototyping post-Merge PoS

Oct 2021
Amphora: multi-client
interop event

Q1 2022
Kintsugi testnet

Q2 2022
Kiln testnet
Shadow forks

Q2-Q3 2022
Fork public testnets
Announce TTD
Release clients

Oct 2020
Deposit
contract
deployed

Dec 2020
Beacon Chain
launches

Apr 2021
Berlin upgrade

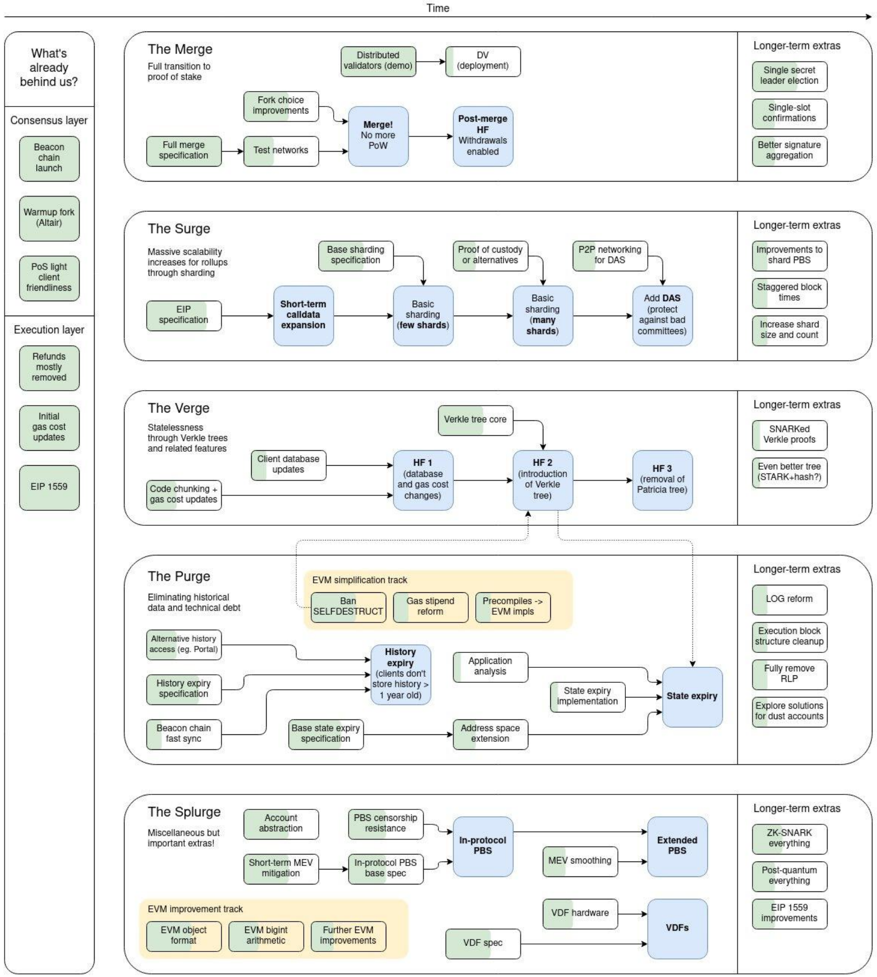
Aug 2021
London upgrade

Oct 2021
Altair upgrade

Date TBA - The Merge
PoS replaces PoW
99.95% energy reduction
Finality enabled

Benefits of post merge Ethereum

- No more PoW energy wastage -> ~99.95% less energy used
- Predictable and faster finality
- Better support for light clients
- Faster ability to detect network issues or attacks
- Groundwork laid for future upgrades such as sharding
- Harder to collude -> more decentralized
- More resilient to attacks
- Higher client diversity -> less reliant on 1 performant client
- People can pool in Ether to setup validators using L1 solutions (sorta Dapps):
  - Rocketpool
  - Lido



Time

What's already behind us?

Consensus layer

Beacon chain launch

Warmup fork (Altair)

PoS light client friendliness

Execution layer

Refunds mostly removed

Initial gas cost updates

EIP 1559

The Merge

Full transition to proof of stake

Distributed validators (demo)

DV (deployment)

Fork choice improvements

Full merge specification

Test networks

Merge! No more PoW

Post-merge HF Withdrawals enabled

Longer-term extras

Single secret leader election

Single-slot confirmations

Better signature aggregation

The Surge

Massive scalability increases for rollups through sharding

Base sharding specification

Proof of custody or alternatives

P2P networking for DAS

EIP specification

Short-term calldata expansion

Basic sharding (**few shards**)

Basic sharding (**many shards**)

Add **DAS** (protect against bad committees)

Longer-term extras

Improvements to shard PBS

Staggered block times

Increase shard size and count

The Verge

Statelessness through Verkle trees and related features

Verkle tree core

Client database updates

Code chunking + gas cost updates

HF 1 (database and gas cost changes)

HF 2 (introduction of Verkle tree)

HF 3 (removal of Patricia tree)

Longer-term extras

SNARKed Verkle proofs

Even better tree (STARK+hash?)

The Purge

Eliminating historical data and technical debt

EVM simplification track

Ban SELFDESTRUCT

Gas stipend reform

Precompiles -> EVM impls

Alternative history access (eg. Portal)

History expiry specification

Beacon chain fast sync

History expiry (clients don't store history > 1 year old)

Application analysis

State expiry implementation

Base state expiry specification

Address space extension

State expiry

Longer-term extras

LOG reform

Execution block structure cleanup

Fully remove RLP

Explore solutions for dust accounts

The Splurge

Miscellaneous but important extras!

Account abstraction

PBS censorship resistance

Short-term MEV mitigation

In-protocol PBS base spec

In-protocol PBS

MEV smoothing

Extended PBS

EVM improvement track

EVM object format

EVM bigint arithmetic

Further EVM improvements

VDF hardware

VDF spec

VDFs

Longer-term extras

ZK-SNARK everything

Post-quantum everything

EIP 1559 improvements

Where to contribute

- https://notes.ethereum.org/@lsankar/security-rfp - call for proposals involving security testing
- Merge data challenge - Coming Soon!
- Core Developer Apprenticeship Program - Coming Soon!
- https://hackmd.io/@poojaranjan/EIP-ERC-Editor-handbook - EIP editor Apprenticeship Program
- https://ethereum.org/en/community/events/ - Events this year
- https://ethresear.ch/ - Long form research discussions
- Eth R&D Discord - Place for quick sync ups and questions
- Youtube - Ethereum Foundation channel for all the public meetings
- https://github.com/OffcierCia/DeFi-Developer-Road-Map - DeFi developer roadmap
- https://github.com/ethereum/EIPs - Ethereum Improvement Proposals
- Ethereum Cat Herders - Non-technical, communication/management related contribution medium

Q&A

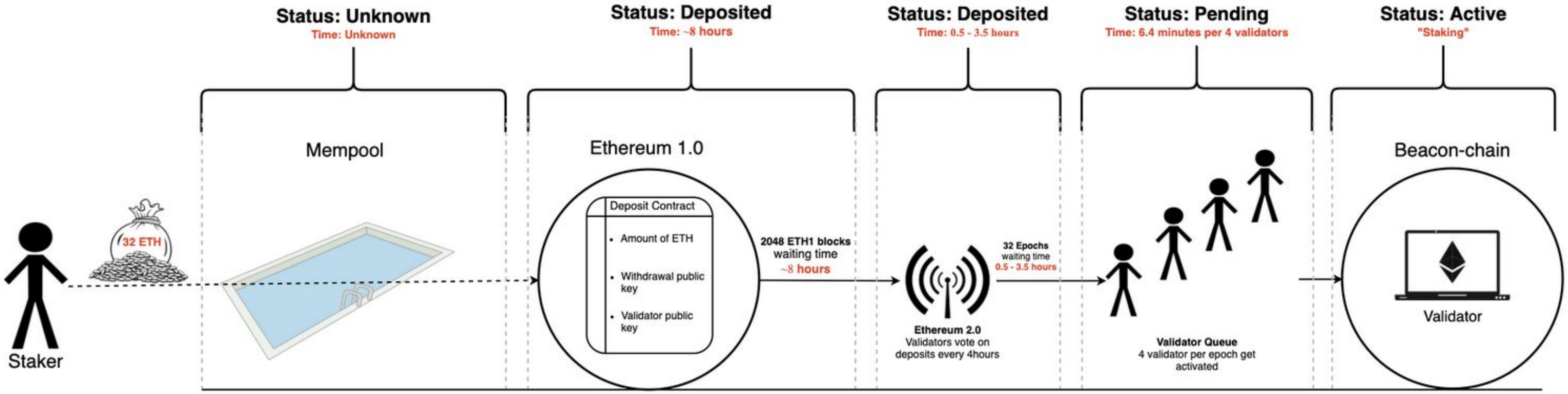
Thank you for your time!

Sources

- https://www.youtube.com/watch?v=N5DdClfLQfw
- https://github.com/ethereum/eth2.0-specs
- https://ethresear.ch/t/pragmatic-signature-aggregation-with-bls/2105
- https://ethos.dev/beacon-chain/
- https://notes.ethereum.org/@vbuterin/SkeyEI3xv
- https://www.adiasg.me/2020/04/09/casper-ffg-in-eth2-0.html

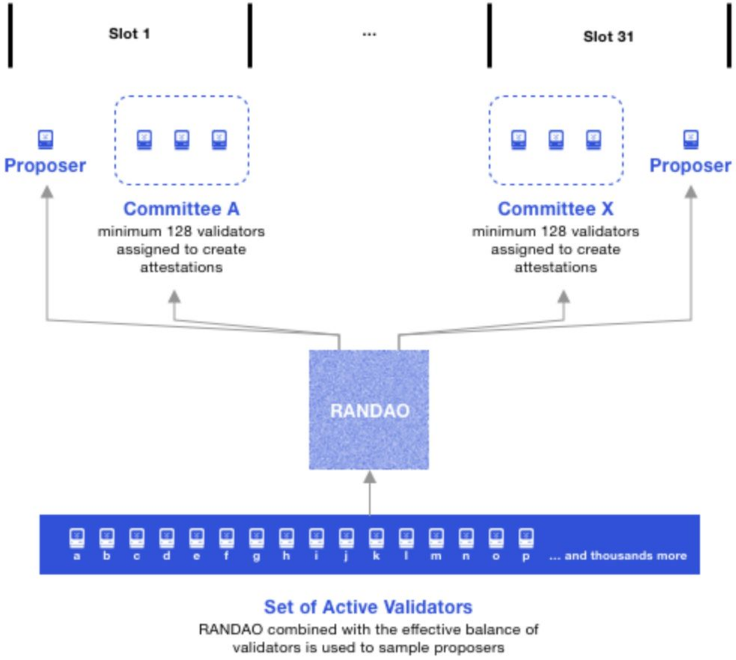
Connection to eth1 chain

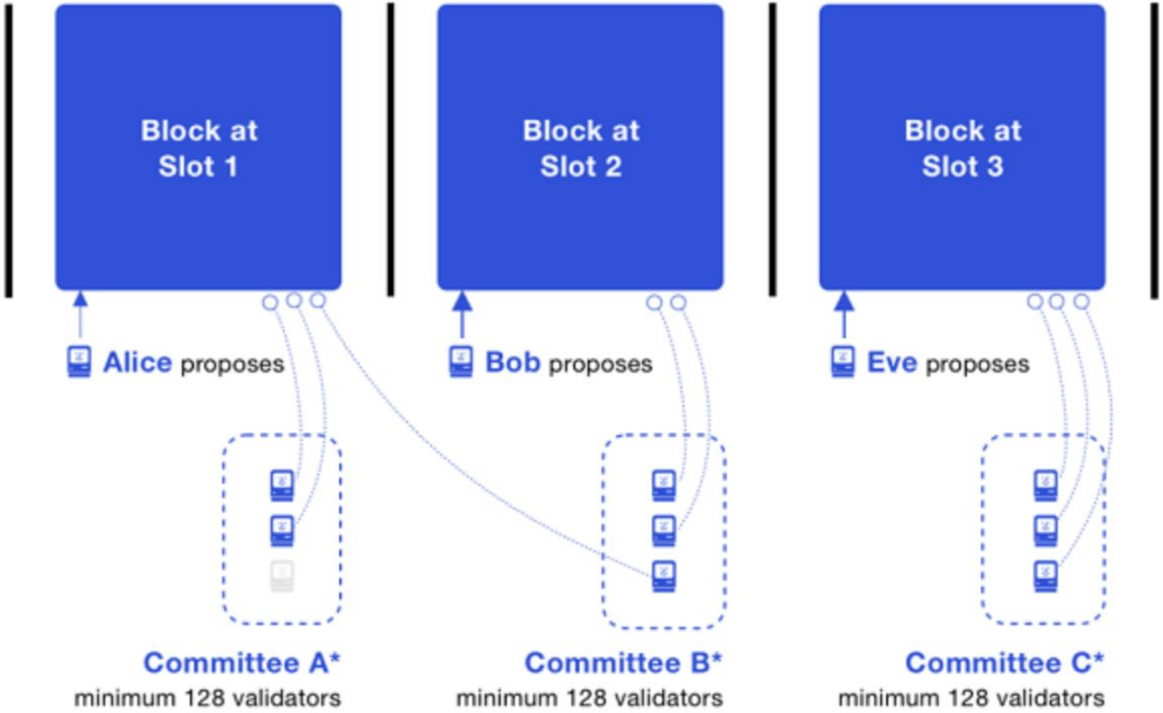
- Deposits are done on the eth1 chain
- Eth2 validator routinely queries an eth1 endpoint for the block
- Eth2 validator then processes the deposit transactions to activate new validators



Who decides what the validator does?

- A committee is a group of validators
- Each slot provides input used to form the committees in the *next* epoch
- Each validator reveals a *random number* after attesting/proposing a block
- This random value from each validator is used by the *Beacon chain* to assign duties for the next epoch
- This random value is also used to decide which validator produces a block


Slot 1
...
Slot 31
Proposer
Committee A
minimum 128 validators assigned to create attestations
Committee X
minimum 128 validators assigned to create attestations
Proposer
RANDAO
a b c d e f g h i j k l m n o p ... and thousands more
Set of Active Validators
RANDAO combined with the effective balance of validators is used to sample proposers



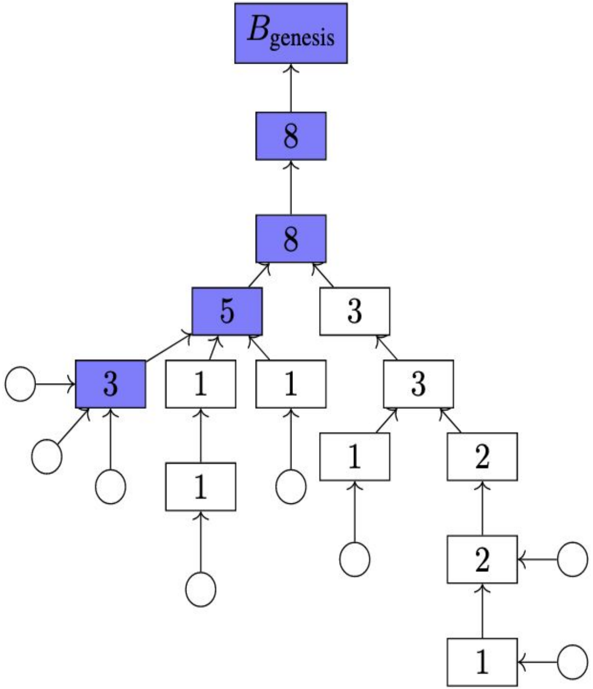
Validators in the committees are supposed to attest to what they believe the head of the blockchain is

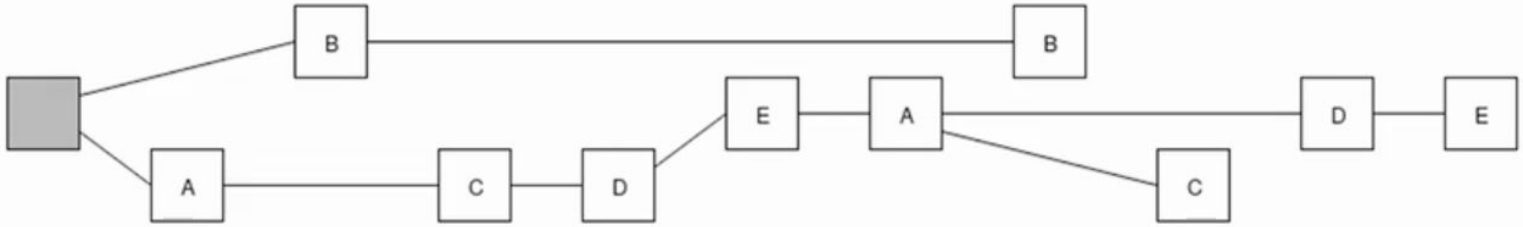
How do we decide which state is correct?

- Validators need to be able to finalize the chain => Decide the valid Head state
- Validators use 2 methods for this:
    a. First the validators find the valid head of the chain by using LMD GHOST
    b. Validators then decide finality based on Casper FFG

LMD GHOST

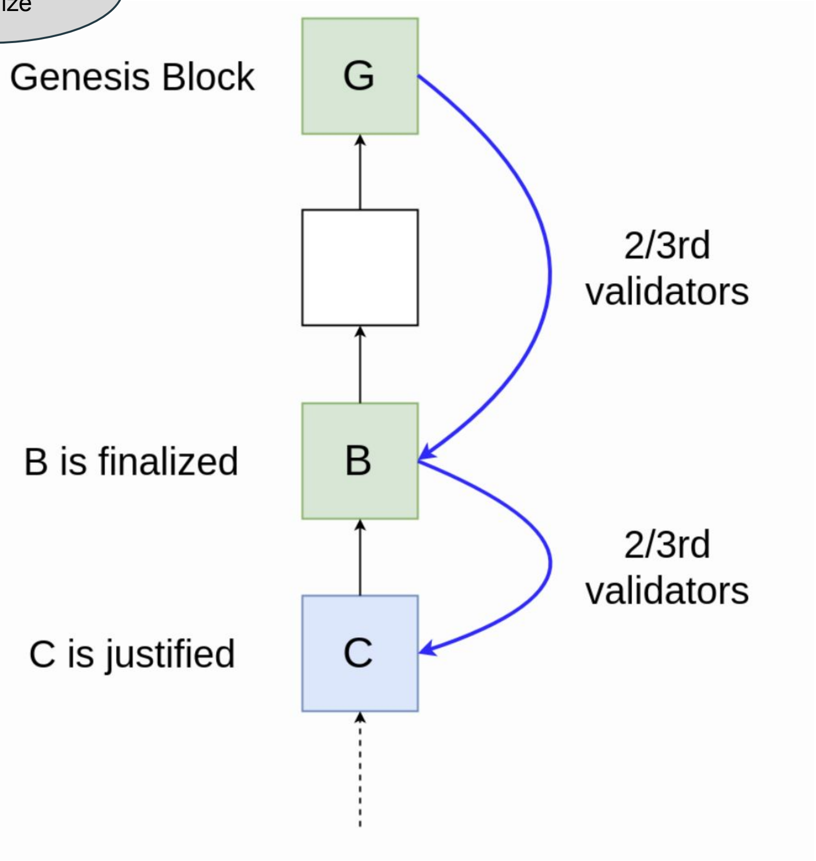
- Latest Message Driven Greedy Heaviest-Observed Sub-Tree
- Chooses the option with the largest weight
- In PoS, weight = number of attestations
- I.e, if there is a fork, choose the fork with the most attestations




B
B
E
A
D
E
A
C
D
C

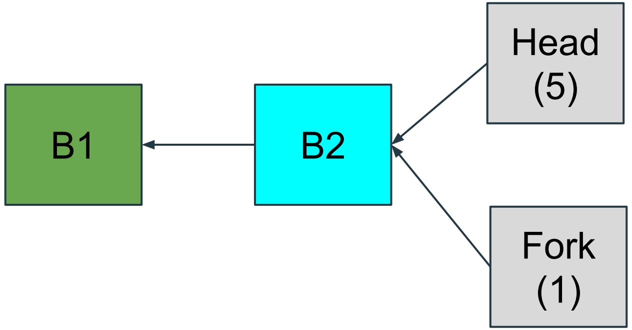
Casper FFG

- Casper the Friendly Finality Gadget
- Block with 2/3rds validator votes become *justified*
- Previous *justified* block becomes *finalized*
- *Finalized* block cannot be changed anymore
- Longest chain holds no value, it’s the chain with highest number of attestations that holds value



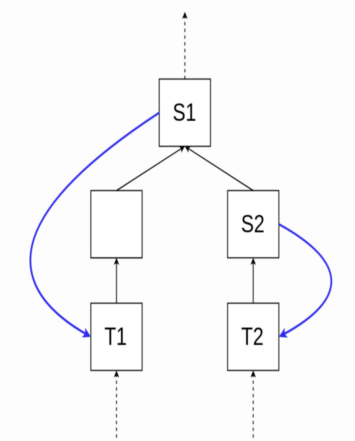
LMD GHOST + Casper FFG

- A validator does the following:
    - Selects the last finalized block its aware of: B1
    - Selects the highest-epoch justified block that is a descendent of B1: B2
    - Uses LMD GHOST starting from B2 to find the head

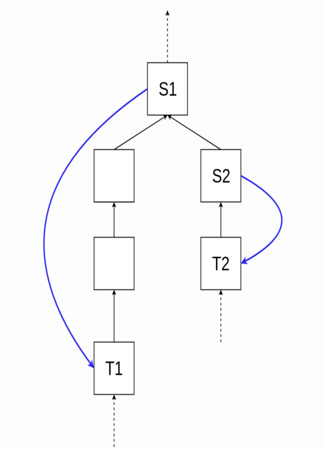


Malicious validation

- A validator attests maliciously if:
  - Validator votes on 2 blocks at the same height
  - Validator votes on 2 blocks at different heights in a fork



Violation of Casper FFG Rule 1: *height(T1) = height(T2)*

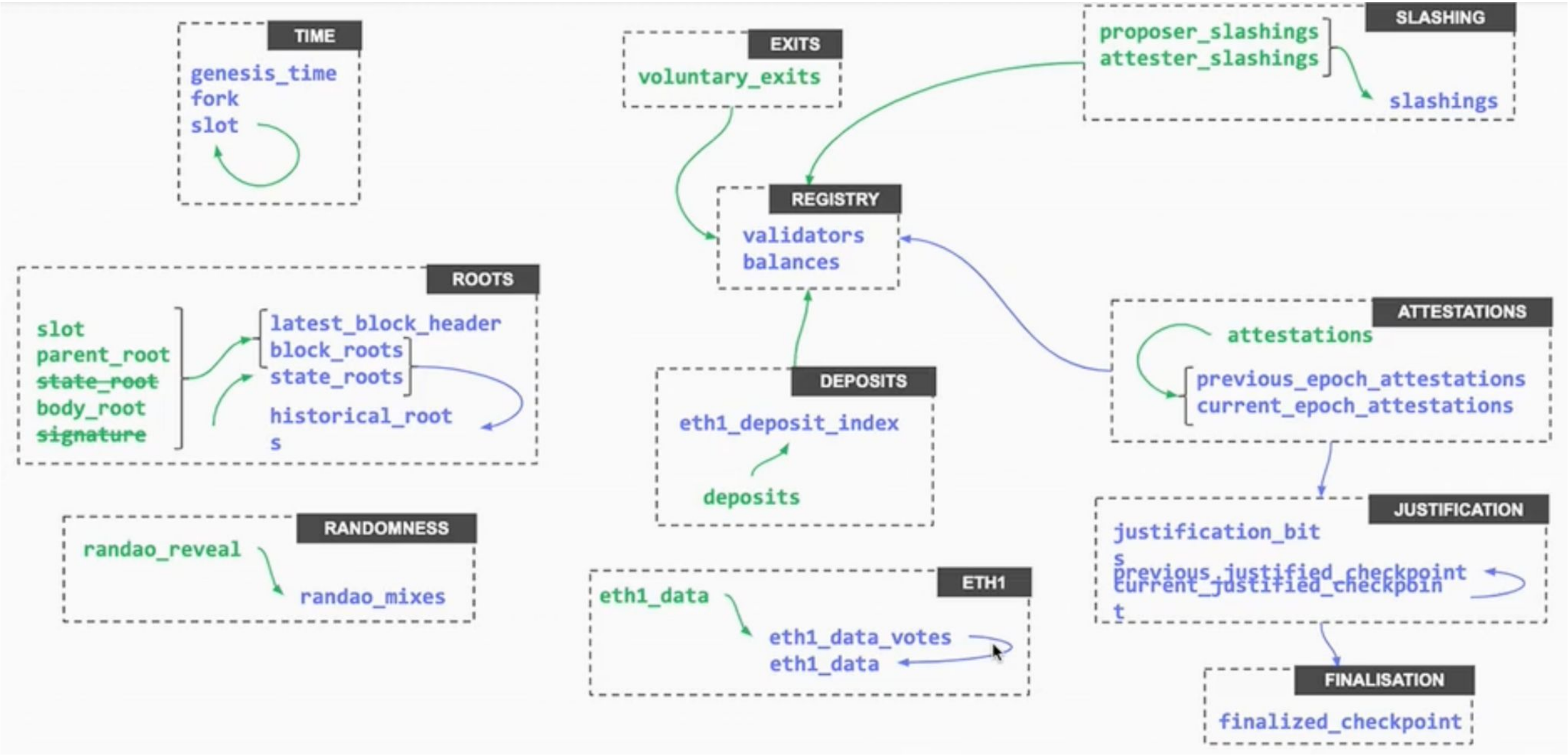


Violation of Casper FFG Rule 2: *height(S1) < height(S2) < height(T2) < height(T1)*

BLS signatures

- If we used ECDSA signatures, verification of all the attestations would take >1day!
- Boneh–Lynn–Shacham (BLS) signature scheme allows a user to verify that a signer is authentic
- Allows for Signature aggregation:
    - $s1 = k1 * H(m)$
    - $s1 + ... sn = kn * H(m)$
- Verification is just: e(Aggregate(public_keys), H(m)) = e(E(1),Aggregate(signatures))
- Much smaller signatures, much less computation involved

Summary so far



TIME

genesis_time
fork
slot

EXITS

voluntary_exits

SLASHING

proposer_slashings
attester_slashings
slashings

REGISTRY

validators
balances

ROOTS

slot
parent_root
~~state_root~~
body_root
~~signature~~

latest_block_header
block_roots
state_roots
historical_roots

DEPOSITS

eth1_deposit_index
deposits

ATTESTATIONS

attestations
previous_epoch_attestations
current_epoch_attestations

RANDOMNESS

randao_reveal
randao_mixes

ETH1

eth1_data
eth1_data_votes
eth1_data

JUSTIFICATION

justification_bits
previous_justified_checkpoint
current_justified_checkpoint

FINALISATION

finalized_checkpoint