Ethereum 2.0

Moving towards a sharded, proof-of-stake Ethereum.

August 21, 2018

Overview

Introduction

Ethereum 1.0

Ethereum 2.0 Overview

Sharding

Proof-of-Stake

Validating

Sharding

Introduction

Personal Introduction

Paul Hauner

@paulhauner | paul@sigmaprime.io

- Co-founder of Sigma Prime
- Software developer
- Contributor to ethereum/casper
- Building an Eth 2.0 client
- Built a binary Casper TFG simulator

Sigma Prime Introduction

Basics

- Founded in late-2016.
- Team of six
  - Cyber-sec experts
  - PhDs.
  - Blockchain PhD candidiate.
  - Me — cat-owing nerd.
- Based in Newtown w/ Europe presence.

Activities

- Cyber Security
  - Penetration Testing
  - Smart contract audits
  - Consultancy
- Blockchain
  - Research
  - Technical specification
  - Niche development
  - Technical consultancy

Ethereum 1.0

What is Ethereum 1.0?

The chain we all know and love.

- Went live in 2015.
- Proof-of-Work.
- “EVM” virtual-machine.
- DEVp2p network.
- Worth billions of USD.
- Has lots of ICOs on it.

Ethereum 1.0 Issues

- Proof-of-work:
  - Burns tons of power.
  - Questionable security model.
- Runs 15 transactions per second.
- Non-parallelizable:
  - Bound by $O(c)$ — the capacity of a single computer.
- "Clunky" EVM:
  - Non-standard.
  - Low cross-compilation support.
  - Obsessed with 256 bit integers.

Ethereum 2.0 Overview

What it is and isn't

Ethereum 2.0 is:

- Being specified by Vitalik Buterin (et al.).
- A core focus of the Ethereum Foundation Research team.
- Under development by several teams (including us!).
- Pretty cool (IMO).

Ethereum 2.0 is not:

- Not completely specified (70% presently).
- Not finalized.
- Not implemented (though the reference implementation is getting close).
- Not widely accepted by "the community".

Specification:

https://notes.ethereum.org/SCIg8AH5SA-O4C1G1LYZHQ?view

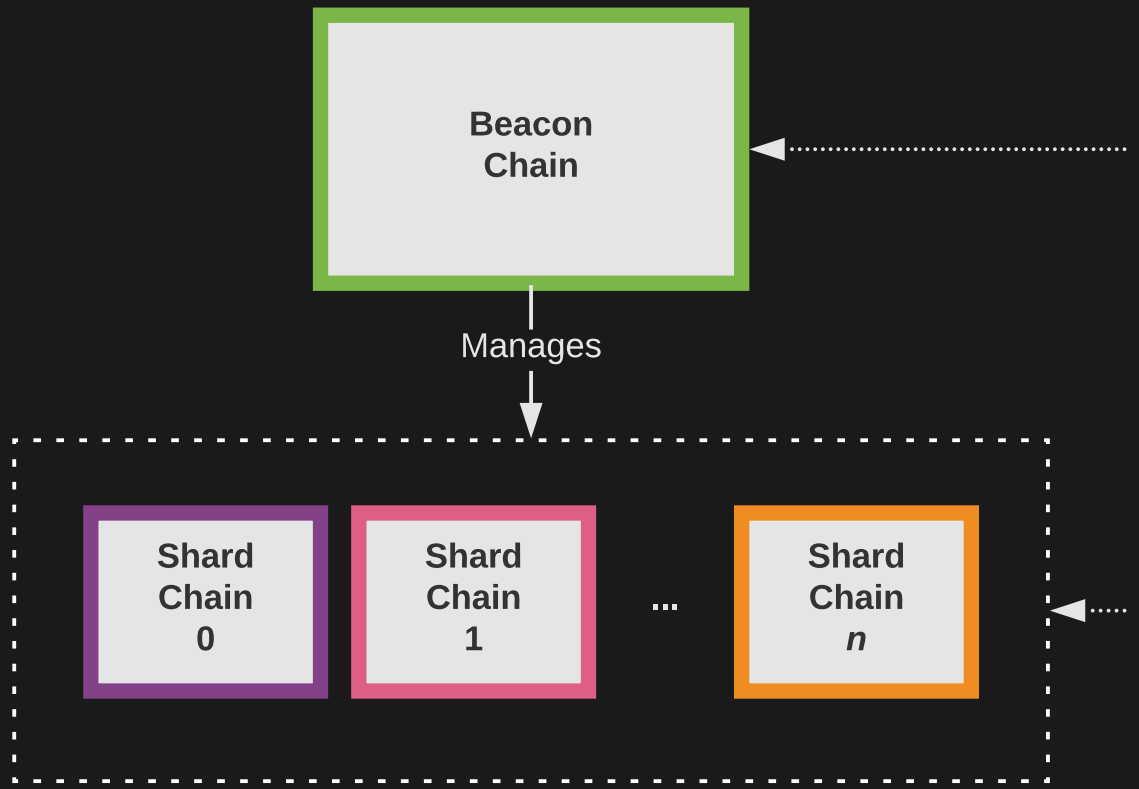
Recent History

Ethereum 2.0 deprecates EIP1011 and combines PoS (Casper) and Sharding into a single project: **Shasper.**

Ethereum 2.0 Features

Features:

- Full Proof-of-Stake with 32 ETH staking requirement.
- Sharding with 1024 initial shards.
- New p2p network — likely libp2p.
- New EVM — likely WASM.
- New blocks!
- New hashes!
- New signatures!
- **Same ETH!** <– No need to reach for your hardware-wallet.



Maintains a validator set, manages consensus (voting) on its own blocks and shard block hashes.

Provides *n* distinct (but inter-operable) chains for general transaction processing.

Defers consensus to Beacon Chain.

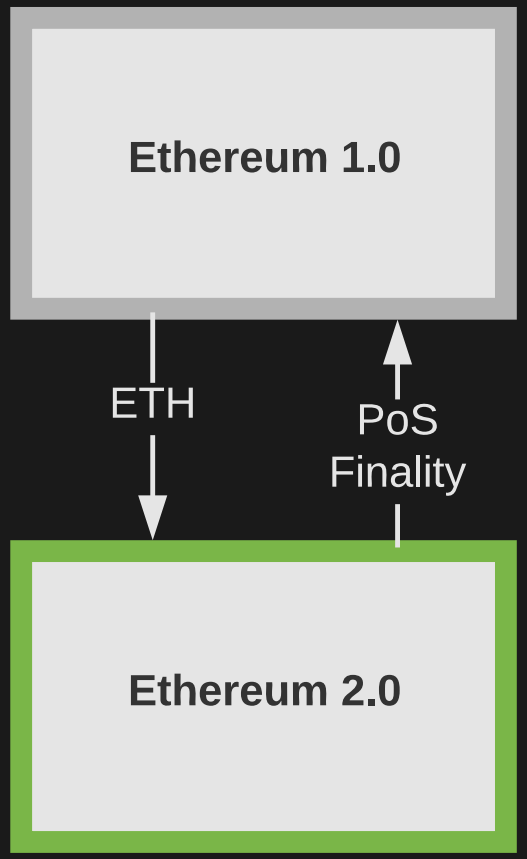
Why add the Beacon Chain?

Sharding and PoS have new requirements.

- Managing shards in the EVM was restrictive.
- Sharding has different network messaging requirements — DEVp2p is not suitable.
- It's sensible to prioritise PoS voting messages over ordinary transactions — this primitive does not exist in the current chain.
- PoS requires a significantly different block structure.

1.0 & 2.0

- Ethereum 1.0 and 2.0 will co-exist.
- 2.0 will initially rely upon 1.0 for a valuable token (Ethereum).
- 1.0 may choose to rely upon 2.0 for finality.



Both can fundamentally exist without each other.

Ethereum 2.0 Timeline

- **Late 2018:** PoC Beacon Chain test-net
- **Early 2019:** Sharding PoCs using Beacon Chain consensus.
- **Late 2019:** Beacon Chain staking real ETH.

These are my personal predictions.
Very limited accuracy, no guarantees.

Present Implementations

- Reference Implementation (Python):
  https://github.com/ethereum/beacon_chain
- ChainSafe Systems (Javascript):
  https://github.com/ChainSafeSystems/lodestar_chain
- Prysm (Go):
  https://github.com/prysmaticlabs/beacon-chain
- Lighthouse (Rust):
  https://github.com/sigp/rust_beacon_chain
- Ether-Camp Harmony (Java):
  https://github.com/ether-camp/ethereum-harmony

Note: none of these are complete.

Sharding

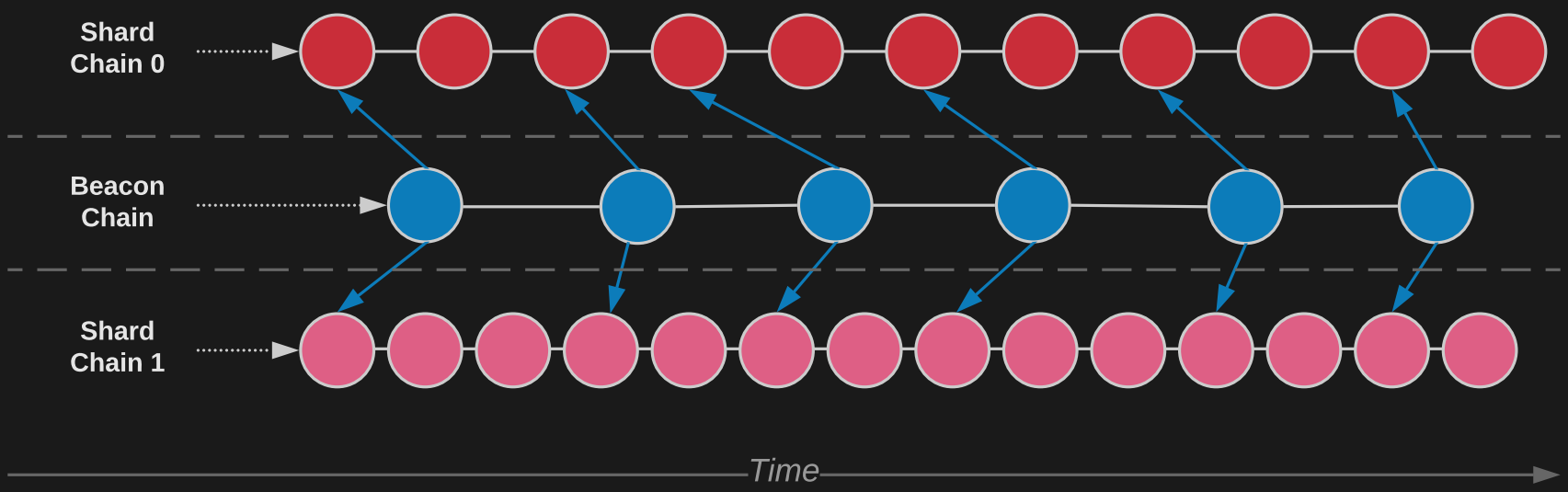
Sharding Intro

Sharding involves running multiple, separate chains as “children” to some other chain.

- Allows for parallel transaction processing.
- Everyone doesn’t have to store *everything* — only store locally the chain you use.
- Helps isolate network congestion.
- Multiple shards is less over-head than multiple blockchains.

Sharding

The beacon chain co-ordinates multiple *shard chains*, where each can have a distinct state and block-time.



Blue arrows are *cross-links*:
simply a reference to the hash of a shard block.

Sharding Progress

Sharding is fairly well defined from a protocol perspective, however the user-experience details of sharding are still undecided.

- "Transparent" vs. "opaque" UX: do users actively pick shards, or is that behind-the-scenes?
- Contract yanking: maybe we can "yank" a contract from one shard to another.
- Delegation of contracts to shards: randomly or by some other means?

See ethresear.ch for discussion.

Proof-of-Stake

PoS: In a Nutshell

Proof-of-Stake involves putting up some "stake" (e.g., ETH) and then participating in a voting process to select valid blocks.

The protocol is designed in such a way that *most* bad behaviour is easily detectable and punishable. If you do bad things, you get "slashed" and lose stake.

Likewise, the protocol rewards you for doing the things that help the chain grow and remain stable.

PoS Benefits

Ethereum is presently proof-of-work but has always intended to move to proof-of-stake.

PoS has the following benefits over PoW:

- It doesn't use insane amounts of energy.
- It is much easier to stop and punish certain types of attacks.
- Theoretically, it should lead to faster blocktimes.
- Resists centralization around hardware manufacturers.

Casper

Casper is a "family" of proof-of-stake consensus protocols

Friendly Finailty Gadget (FFG)

- "Vitalik's Casper".
- Simple.
- Achievable.
- Ready-to-deploy.

The Friendly GHOST (TFG)

- "Vlad's Casper".
- Advanced.
- Complex (in a good way).
- Still under-research.

Ethereum 2.0 will be FFG first, with TFG in mind for the future.

Validating

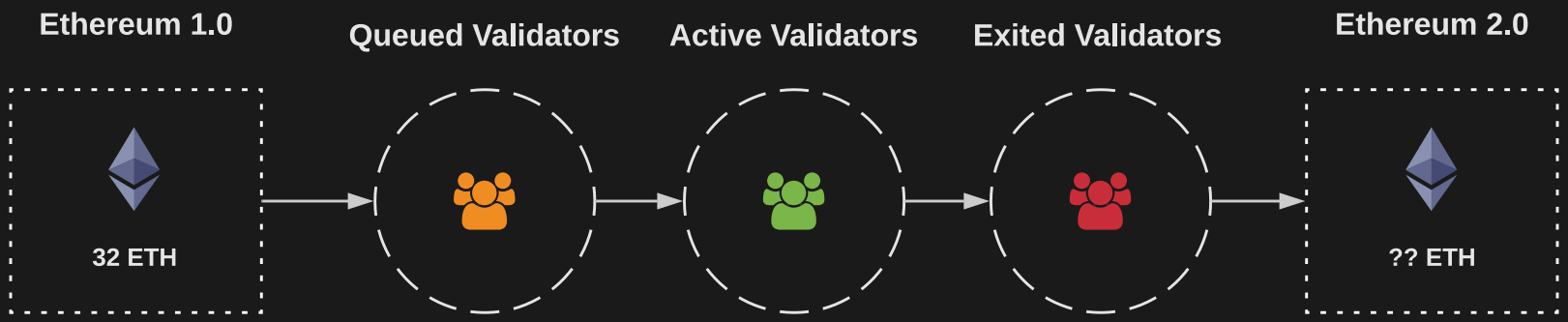
How to become a Validator

Deposits

Along with a fixed-size amount of 32 ETH, deposits to the "deposit" contract specify the following:

- **pubkey**: a BLS public key for signing attestations.
- **bls_proof_of_possession**: required to prevent rouge-key attacks.
- **withdrawal_shard_id**: where withdrawn ETH will be deposited.
- **withdrawal_addr**: the address on the withdrawal shard.
- **randao_commitment**: the address on the withdrawal shard.

Validator Set Management



Presently, validators can only deposit via an Eth 1.0 smart-contract.

Queued Validators are inducted as slots open due to growth or exiting Active Validators.

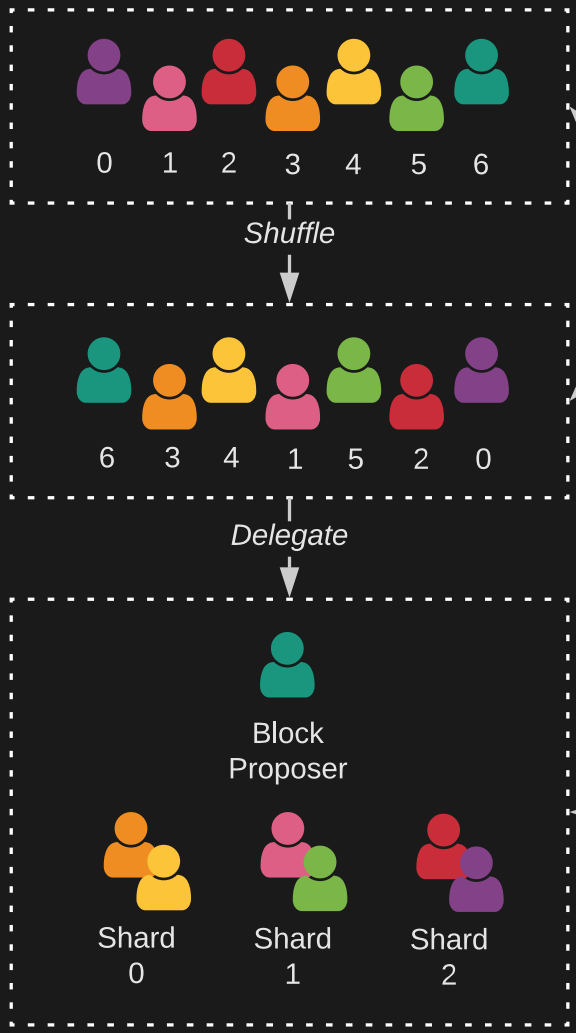
The size of the Active Validator pool grows over time.

Validators exit by request or if they are slashed to some minimum deposit.

After significant delay, validator ETH is deposited to the shard chain specified on deposit.

Validator Delegation

Validators are shuffled and re-allocated each 512 seconds.



The Active Validator set is deterministically shuffled.

Shuffling entropy is derived from RANDAO.

The shuffled Active Validator set is assigned to shards and block production.

This diagram is simplified. In reality, tasks are also assigned with respect to slots and the block proposer is also assigned to shards.

Active Validator Duties

- Stay online.
- Maintain a beacon-chain full node.
- Maintain a full node(?) of assigned shards.
- Produce blocks and cross-links.
- Attest to blocks and cross-links.

Sharding

Resources

- "The spec": https://notes.ethereum.org/SCIg8AH5SA-O4C1G1LYZHQ?view
- Reference implementation: https://github.com/ethereum/beacon_chain
- BLS aggregate signatures: https://ethresear.ch/t/pragmatic-signature-aggregation-with-bls/2105/18
- RPJ fork-choice: https://ethresear.ch/t/beacon-chain-casper-ffg-rpj-mini-spec/2760
- Vitalik "history of casper" tweet-storm: https://twitter.com/VitalikButerin/status/1029900695925706753