# What's New in Eth2 - 22 October 2018

Edition 4. Archive.

Ben Edgington (PegaSys, ConsenSys)

---

*I'm consolidating and tracking spec updates, research progress, design decisions and other significant discussions from the previous 7 days in the rapidly moving world of Ethereum 2.0[1]. I'm not following individual client implementation progress, and I'm only covering things that I think are particularly interesting or useful for implementers. Sorry if I missed your thing.*

---

It's been almost a month already! Timing of updates over the next weeks might be erratic due to Devcon IV and subsequent PegaSys commitments. But there should be lots of good Ethereum 2.0 info to report, so I'll try to consolidate it and get it out there at some point.

## Specification updates

- Validator exiting has been re-worked. Specifically, there is now a function `exit_validator()` to take care of housekeeping. The concept of a minimum online deposit size has been introduced: *if a validator's stake falls below 16 Ether then it will be exited*. (Presumably this can happen only as a result of the quadratic leak, as penalised validators are exited immediately.)
- RANDAO changes. Validators are required to provide a "RANDAO commitment" when they are inducted. This is an initial random value hashed many, many times: a hash onion. When the validator is called upon to propose a block, it provides the pre-image of this commitment as the RANDAO contribution for the block, which is then mixed into the existing random state. However, if the proposed block is orphaned for some reason (not on the canonical chain), then the validator's next RANDAO reveal is known to all ahead of time (Justin has promised an ethresear.ch post on this). This change gradually increases with time the number of layers of the hash onion that must be unpeeled a reveal: every 4096 slots/18 hours, the depth of pre-image to be provided increases by one, greatly reducing the chance that this early reveal can have an impact (presumably as there is little chance of the validator being proposer again within that number of slots).
- *@paulhauner* made some fixes to the `shuffle()` helper function.
- Integers are now unsigned throughout. (Pity the poor Java developers...)

Some RFCs from Vitalik for discussion:

---

[1] Ethereum 2.0 is a planned upgrade to the existing Ethereum protocol, introducing proof-of-stake and scalability via sharding among other things. It is a work-in-progress, decentralised, distributed R&D programme, and things can change rapidly. Some background.

- A proposal for chain initialization and updating, main chain block inclusion, and deposit processing. The Beacon Chain starts when 16384 validators have placed a deposit in a PoW Mainnet contract.
- A proposal for shard blocks, including structure, validation, fork choice, and crosslinks.
- A proposal for an alternative withdrawal mechanism when validators want to exit. See the ethresear.ch post discussed below.

A spec for a general test format has been created. Good discussion, including how to handle forks.

## Implementers' call

- Excellent notes from the last implementers' call. This really is a labour of love by @pggallagher.
- No call during the last week. Next call date TBD - it was left hanging at the end of the last call whether we would do the 25th or not. Keep an eye on the the PM repo and Gitter, I guess.

## Gitter

- Discussion about whether the Milagro crypto library is suitable for use in Eth2.0. There are some concerns about it.
    - Danny Ryan made a start on a summary of the status of the various available BLS12-381 curve and aggregation libraries, including language, licence and audit information.
- Early on, there were ideas about different shards running different protocols, maybe some test shards. Confirmation that this is no longer the plan: shards will be homogeneous.
- Some discussion about what could happen in a chain-split scenario, as recently happened on the Ropsten testnet during Constantinople hard fork testing.

## Ethresear.ch

- Suggested average-case improvements to reduce capital costs of being a Casper validator. Instead of fixing the *length of time* for a withdrawal (the time from a validator signalling to exit to receiving its deposit back), instead fix the *rate* of withdrawals. This approach may have some interesting properties.

## In other news...

- My Ethereum 2.0 slides from the London Ethereum Meetup on 17th October. This was a shorter version than previous presentations, but with some updates and re-working.

---

## Main sources:

- Updates to the specification document
    - Pull requests
    - Commits
- Fortnightly Eth2.0 implementers' calls
    - https://github.com/ethereum/eth2.0-pm
- https://ethresear.ch/
- https://gitter.im/ethereum/sharding and https://gitter.im/ethereum/casper
- Updates to the Eth2.0 Handbook.
- Updates to https://github.com/ethereum/wiki/wiki/Sharding-roadmap
- Issues at https://github.com/ethereum/beacon_chain/issues