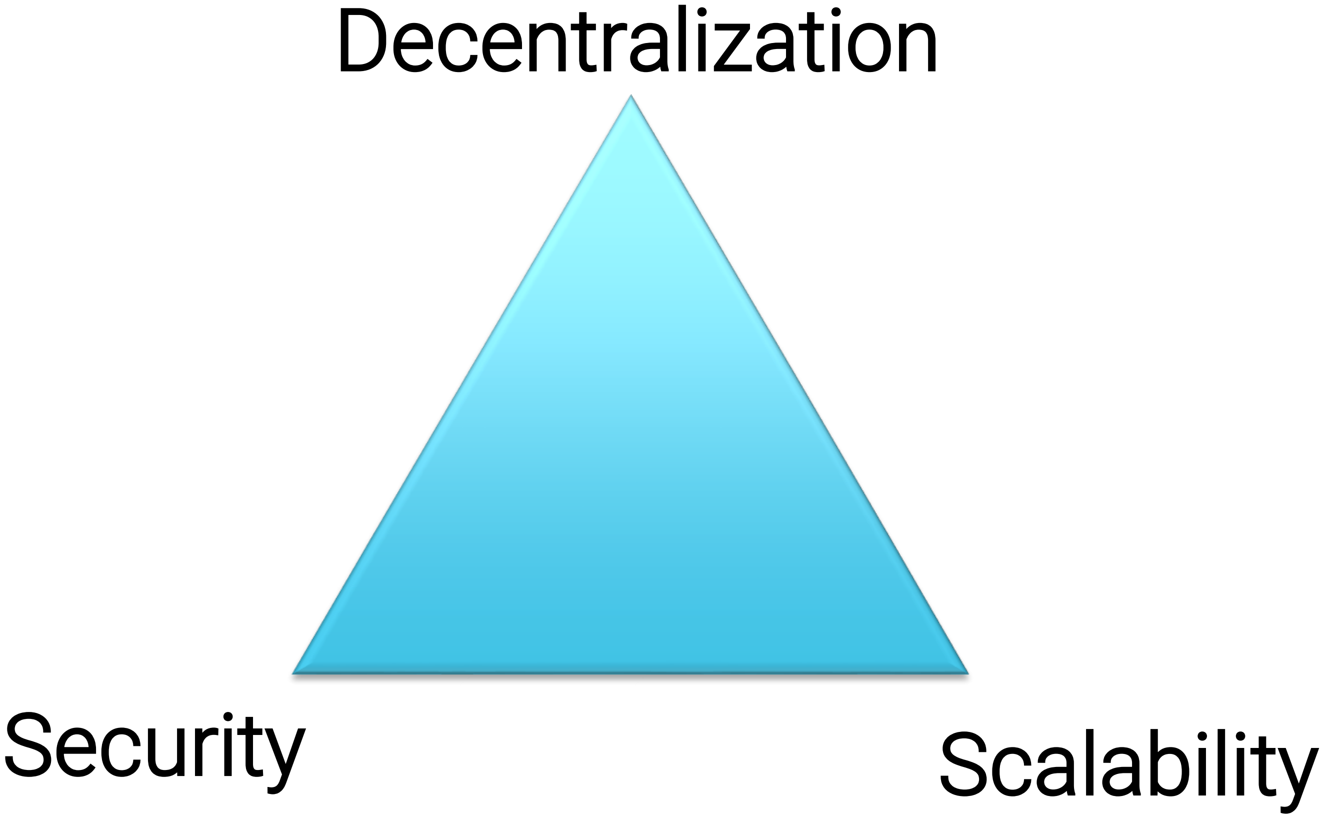
Advanced Concepts in Ethereum

# As of April 11, 2019

Blockchain Trilemma


Decentralization

Security

Scalability

Any Blockchain system can have at most 2 of these 3 properties

Ethereum Constantinople Upgrade

Ethereum Constantinople Upgrade

- After the community decides upgrades on the network, changes to *protocols* are written on different ethereum clients (geth, parity, etc.)
- The protocol changes are *activated* at a specific block number
- Latest upgrade at **block number 72,80,000**, occurred on **Thursday, February 28, 2019** .
- Ethereum *wallets* (like metamask) will *auto-configure*.
- Ethereum clients will offer Constantinople from latest versions. (geth v1.8.23., Parity v2.2.10-stable etc.)

Ethereum Constantinople : New EIPS

- EIP 145: Bitwise shifting instructions in EVM : For less gas cost

  (Alex Beregszaszi, Paweł Bylica)

  - Native *bitwise shifting* along with arithmetic operators.
  - Earlier versions use arithmetic operators for shift, which is expensive
  - Reduce gas cost from 35 gas to 3 gas. i.e. 10x less gas..

Ethereum Constantinople : New EIPS

- EIP 1014: Skinny CREATE2 (Vitalik Buterin) : For off-chain transactions
  - New Opcode for contract creation : CREATE2 (earlier CREATE)
  - CREATE : if destination address already has nonempty code, then the creation throws immediately.
  - CREATE2 : similar to CREATE, but creates a contract at a targeted address that can be determined ahead of time.
  - Adds a new opcode at 0xf5, which takes 4 stack arguments: endowment, memory start, memory length, salt.
  - Milestone for off-chain contract. This EIP makes it so you can interact with addresses that are *yet to be created*.

Ethereum Constantinople : New EIPS

- EIP 1052: EXTCODEHASH opcode : For fast smart contract verification

  (Nick Johnson, Paweł Bylica)

  - specifies a new opcode EXTCODEHASH, which returns the *keccak256 hash* of a contract's code.
  - Many contracts need to perform *checks* on a contract's bytecode, but do not necessarily need the bytecode itself.
  - a contract may want to check if another contract's bytecode is one of a set of permitted implementations, or it may perform analyses on code and whitelist any contract with matching bytecode if the analysis passes.
  - Currently uses EXTCODECOPY which is expensive

Ethereum Constantinople : New EIPS

- **EIP 1234: Constantinople Difficulty Bomb Delay and Block Reward Adjustment** (Afri Schoedon) : Reduce block reward to 2 ETH.
  - The average block times are increasing due to the difficulty bomb (also known as the “ice age”) slowly accelerating.
  - This EIP proposes to delay the difficulty bomb for approximately 12 months and to reduce the block rewards to adjust for the ice age delay.
  - This EIP make sure we don’t freeze the blockchain before proof of stake is ready & implemented.

Ethereum Constantinople : New EIPS

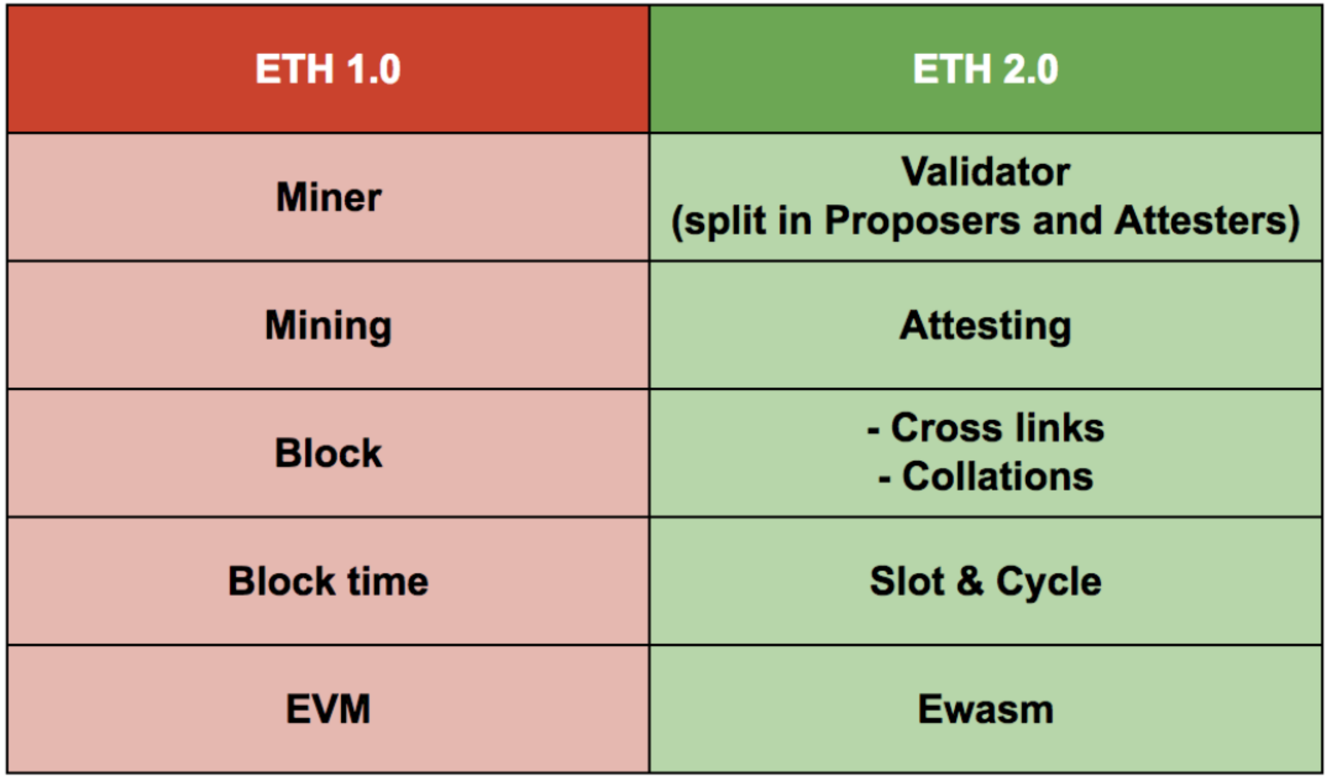
- EIP 1283: Net gas metering for SSTORE without dirty maps

  (Removed from test networks using the St. Petersburg network upgrade)

  - SSTORE : Save a word (256 bits or 32 bytes) to storage. Approx 20,000 gas cost.
  - At least 5000 gas should be spent for every trigger of store operation and gas stipend for Transfer and Send operation is 2300 approx.
  - This EIP proposed to reduce the gas cost of store to only 200 gas.
  - Created chances of re-entrancy attack, if attacker can use gas stipend to manipulate the contract.
  - As of now, this EIP is removed for better replacements.

Ethereum 2.0

Ethereum 2.0



ETH 1.0	ETH 2.0
Miner	Validator (split in Proposers and Attesters)
Mining	Attesting
Block	- Cross links - Collations
Block time	Slot & Cycle
EVM	Ewasm

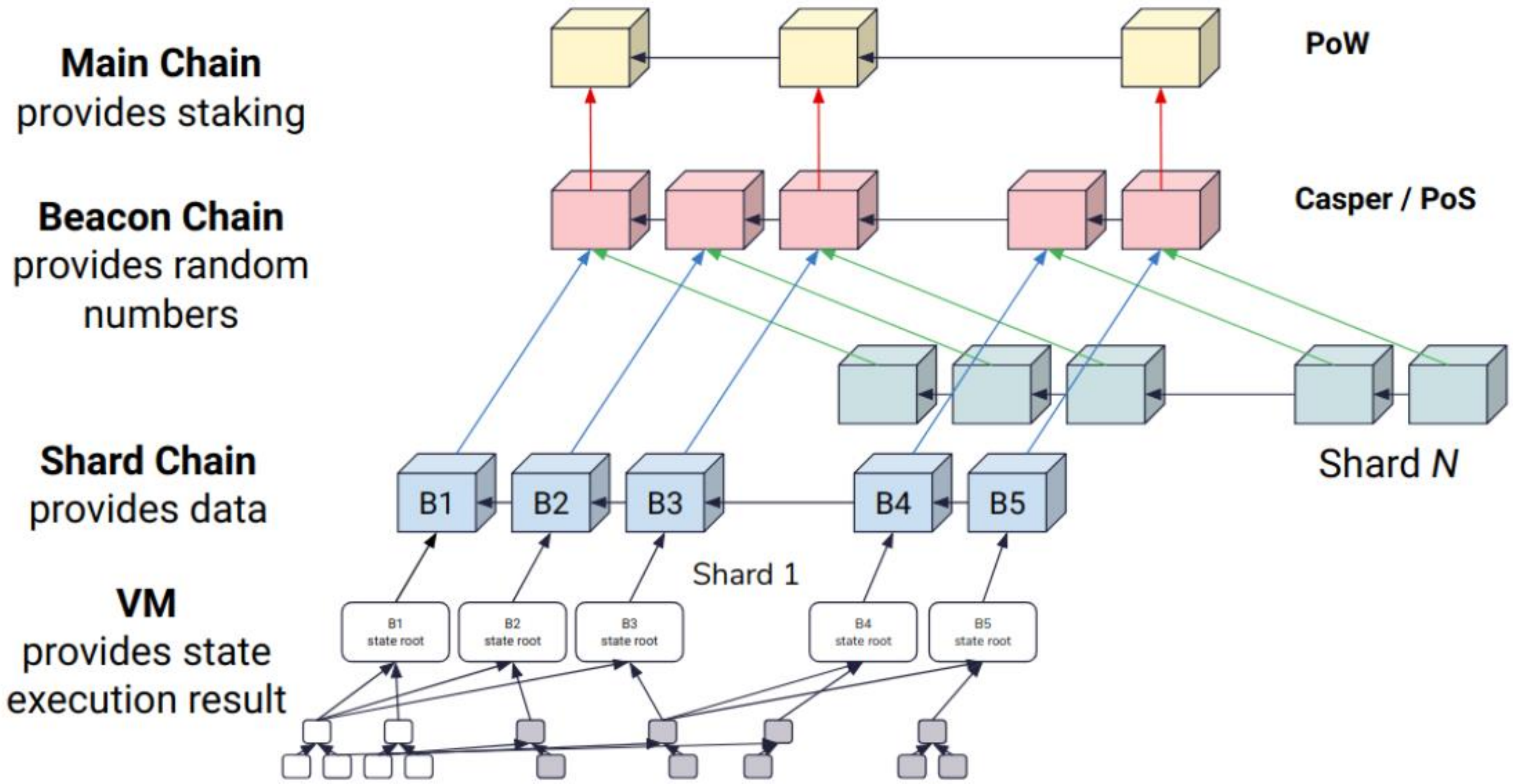
Source : Coinmarketcap.com

Ethereum 2.0

Overview of Ethereum 2.0

a coordination and validation layer

where the smart contracts will be run and where their data will be stored.



Source: Hsiao-Wei Wang

Ethereum 2.0

- Phase 0 -- The Beacon Chain (For Casper / PoS)
- Phase 1 -- Custody game
- Phase 1 -- Shard Data Chains

Vitalik Buterin stated that ETH 2.0 will be able to process 15,000 transactions per second, instead of the current 15 transactions per second.

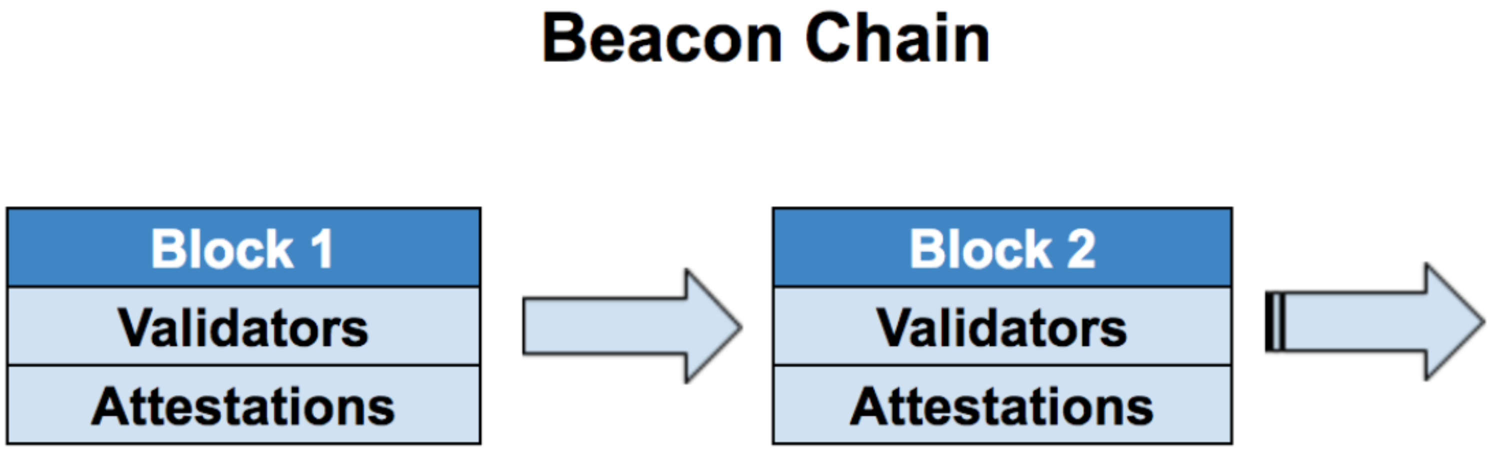
Validator

- Validators are split into **block proposers** and **attesters** who respectively create and validate new blocks.
- Validators **stake** some Ether in a smart contract, and vote to **attest** that new blocks are valid.
- If they don't participate in the vote or if they validate incorrect blocks, their Ether stake will be either **reduced*** or **destroyed*** entirely, if they are found out.
- Idea of **slashing conditions*** and the mechanism that keeps the blockchain secure.

The Beacon Chain

- The core of Ethereum 2.0 is a system chain called the "beacon chain"
- The beacon chain stores and manages the registry of **validators**.
- The only mechanism to become a validator is to make a one-way ETH **transaction** (32 ETH*) to a Validator Main Contract (VMC) on Ethereum 1.0.
- Activation as a validator happens when Ethereum 1.0 deposit receipts are processed by the beacon chain, the activation balance is reached, and a queuing process is completed.
- Primary source of load on the beacon chain is "attestations".

The Beacon Chain



Beacon Chain

Block 1

Validators

Attestations

Block 2

Validators

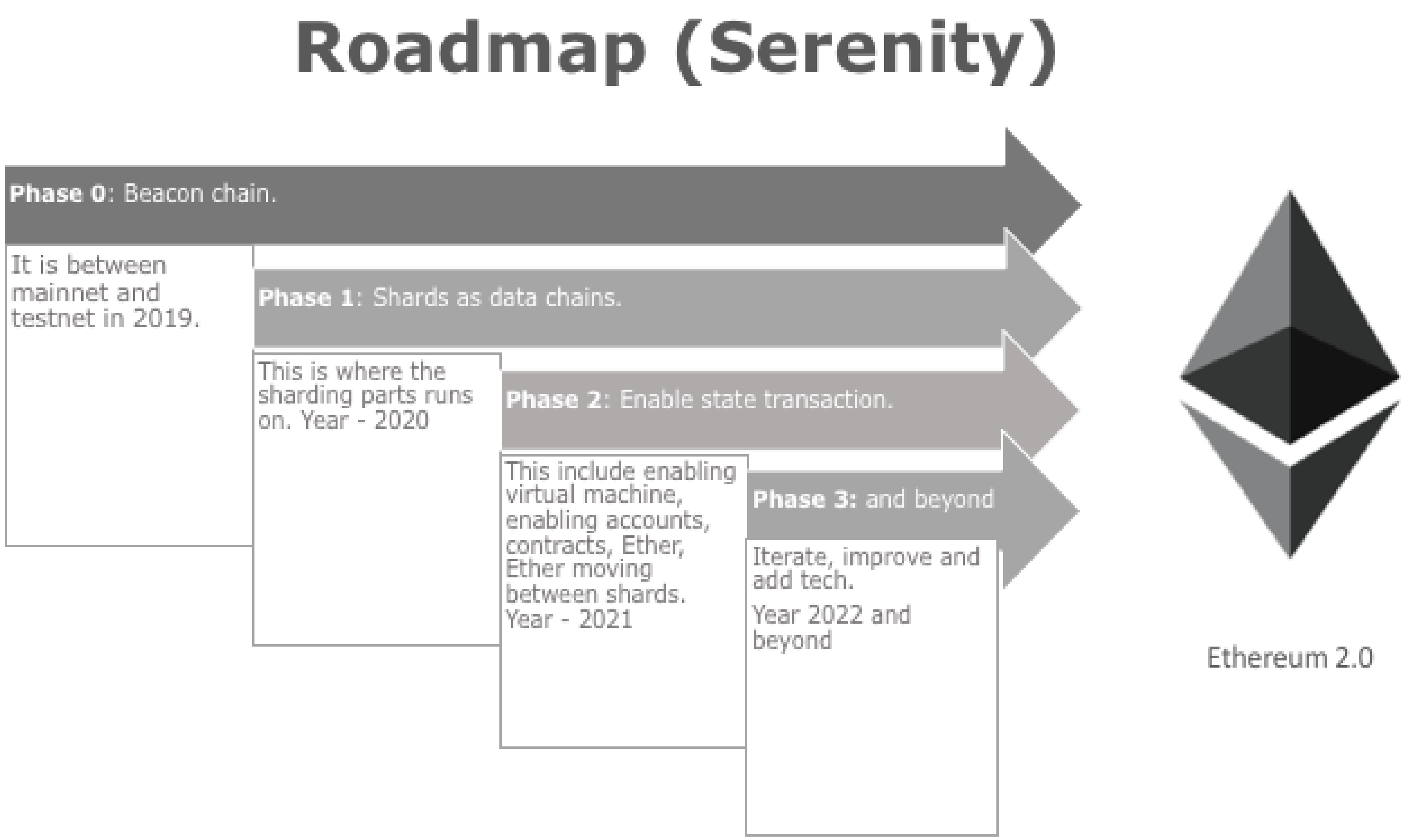
Attestations

Serenity

- Proof of Stake (Casper)
- Scalability (Sharding)
- VM improvement (EWASM)
- Improvements to cross-contract logic

Beacon Chain

Serenity



Roadmap (Serenity)

Phase 0: Beacon chain.

It is between mainnet and testnet in 2019.

Phase 1: Shards as data chains.

This is where the sharding parts runs on. Year - 2020

Phase 2: Enable state transaction.

This include enabling virtual machine, enabling accounts, contracts, Ether, Ether moving between shards. Year - 2021

Phase 3: and beyond

Iterate, improve and add tech.

Year 2022 and beyond

Ethereum 2.0

eWASM - Ethereum Web Assembly

- Smart-contracts are the life-blood of Ethereum.
- eWASM is being developed as a replacement to the EVM
- eWASM is based on the WASM (WebAssembly) instruction-set which is designed as an open standard by a W3C Community Group and is actively being developed by engineers from Mozilla, Google, Microsoft, and Apple.
- eWASM will make a significant difference to how many transactions can be processed and subsequently added to a block—further increasing transaction throughput.

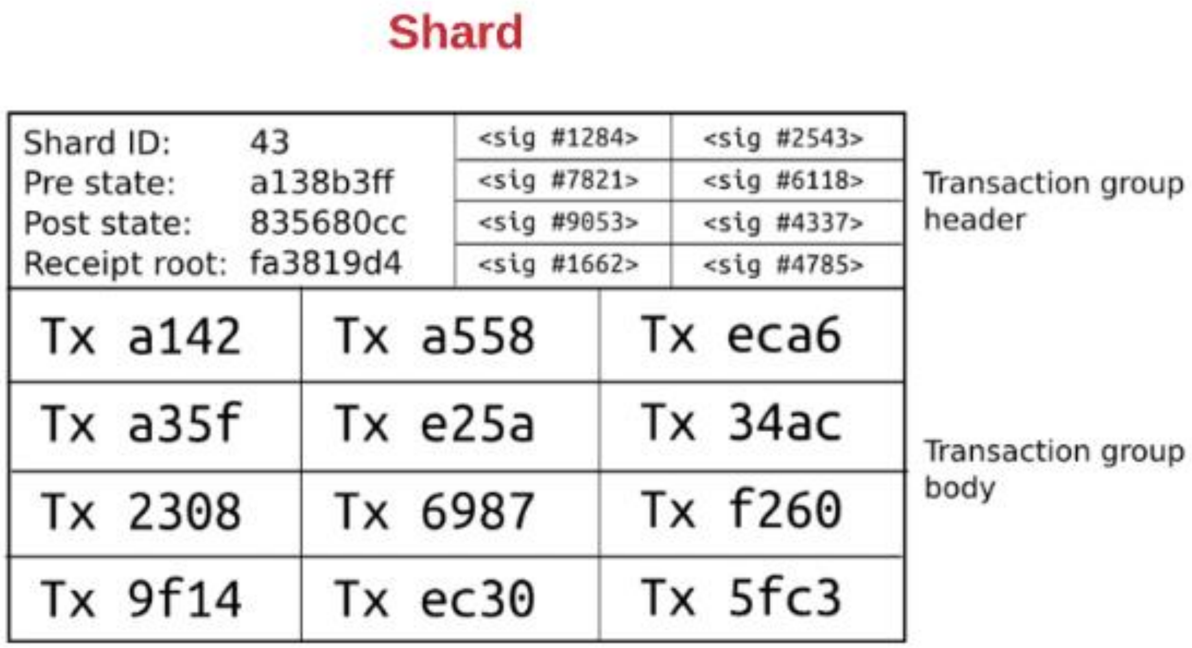
Ethereum Sharding

Concept of Sharding in Ethereum

- As proposed by Vitalik buterin in Devcon
  - Imagine that Ethereum has been split into thousands of islands.
  - Each island can do its own thing.
  - Each of the island has its own unique features and everyone belonging on that island i.e. the accounts, can interact with each other AND they can freely indulge in all its features.
  - If they want to contact with other islands, they will have to use some sort of protocol.

Sharding in Ethereum

- Shard means “ A small part of a whole”
- At base, there is a transaction group and every shard will have its own group



Shard

Shard ID:	43	<sig #1284>	<sig #2543>
Pre state:	a138b3ff	<sig #7821>	<sig #6118>
Post state:	835680cc	<sig #9053>	<sig #4337>
Receipt root:	fa3819d4	<sig #1662>	<sig #4785>

Transaction group header

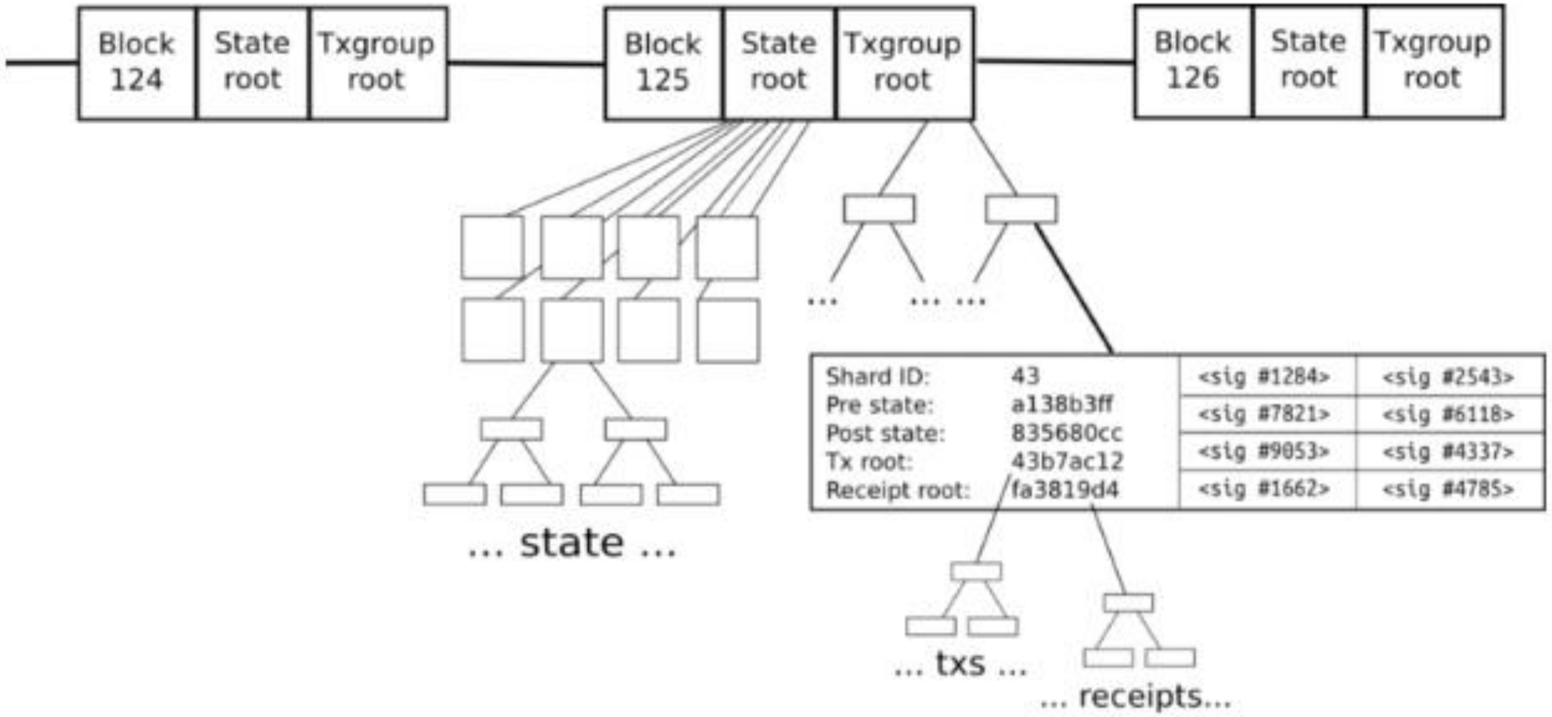
Tx a142	Tx a558	Tx eca6
Tx a35f	Tx e25a	Tx 34ac
Tx 2308	Tx 6987	Tx f260
Tx 9f14	Tx ec30	Tx 5fc3

Transaction group body

Image courtesy: Hackernoon

Sharding in Ethereum

- As a complete blockchain (maximum of 1024 shards)



Shard ID:	43	<sig #1284>	<sig #2543>
Pre state:	a138b3ff	<sig #7821>	<sig #6118>
Post state:	835680cc	<sig #9053>	<sig #4337>
Tx root:	43b7ac12	<sig #1662>	<sig #4785>
Receipt root:	fa3819d4		

Image courtesy: Hackernoon.

THANK YOU