Ethereum

Course IN5420 / IN9420 «**Blockchain Technologies**»

26.02.2024, University of Oslo

By Alina Lapina

Overview

Consensus Algorithm

Proof-of-Stake, GHOST, The Merge (2022)

Smart Contracts

Smart contracts, gas, EVM

Data Structure

Merkle Patricia Tries, Verkle trees

Compare to Bitcoin

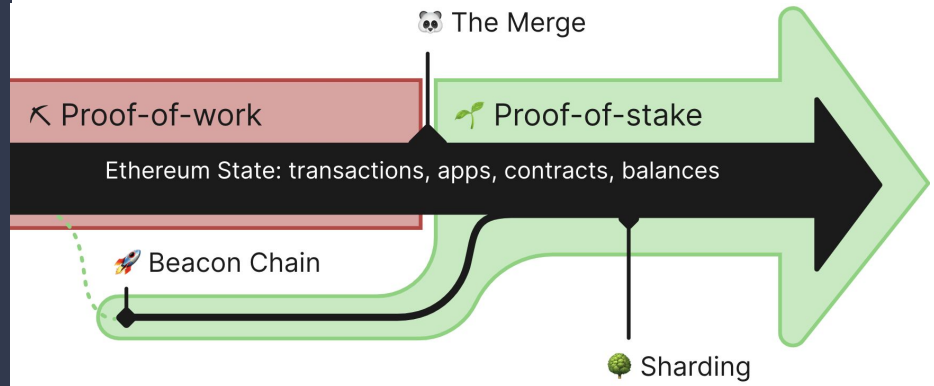
Performance, consensus, future

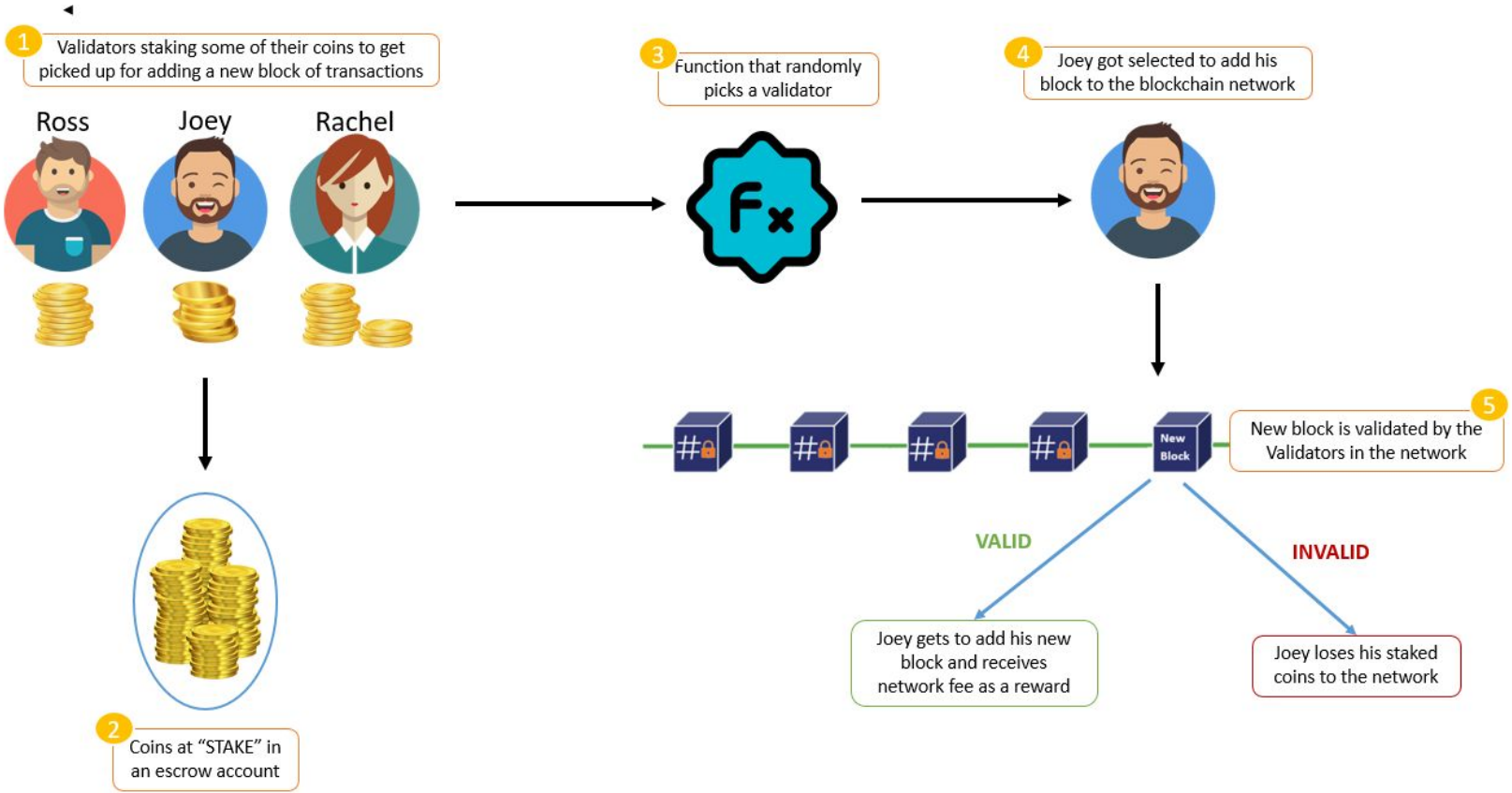
Consensus Algorithm: PoS

What Is The Merge?

https://ethereum.org/roadmap/merge

The Merge was the moment when Ethereum switched off its proof-of-work-based consensus mechanism and switched on its proof-of-stake-based consensus mechanism. The Merge happened on **September 15, 2022**.





Credit: https://www.helius.dev/blog/proof-of-history-proof-of-stake-proof-of-work-explained

Validators

Nodes and committees

To participate as a validator, a user must

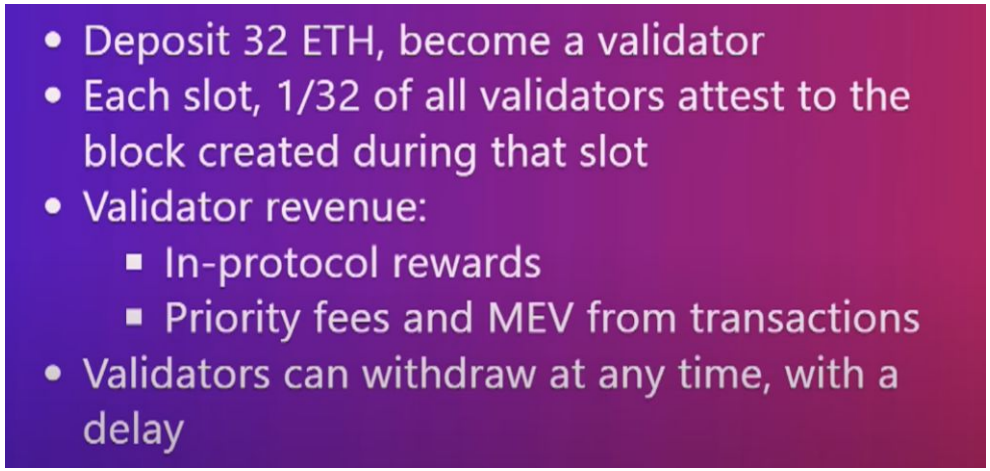
- deposit **32 ETH** into the deposit contract (activation queue) and
- **run** three separate pieces of **software**: an execution client, a consensus client, and a validator client.
- The transactions delivered in the block are re-executed to check that the proposed changes to Ethereum's state are valid, and the block signature is checked. The validator then sends a vote (called an **attestation**) in favor of that block across the network.

Dividing the validator set up into **committees** is important for keeping the network load manageable.

Deposit and reward

Withdrawals are promised in the next versions (2023)

https://www.youtube.com/watch?v=UihMqcj-cgc&list=PLVB9Cbj6p2QhhQz84pNqTulrWzi3NUdYi&index=19 , Vitalik Buterin, «Ethereum in 30 minutes»



Time

Slots and epoches

Proof-of-stake, the tempo is fixed. Time in proof-of-stake Ethereum is divided into **slots** (12 seconds) and **epochs** (32 slots).

Every slot

One validator is randomly selected to be a **block proposer** in **every slot**. This validator is responsible for creating a new block and sending it out to other nodes on the network. A **committee of validators** is randomly chosen, whose votes are used to determine the validity of the block being proposed.

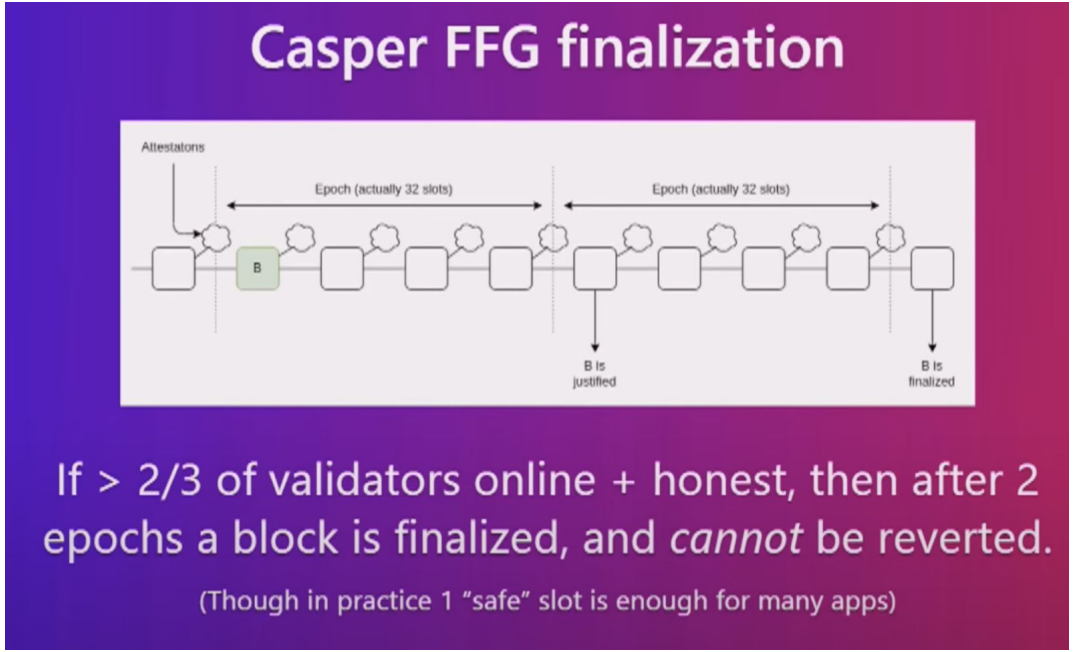
Every epoch

Committees divide up the validator set so that every active validator attests in **every epoch**, but not in every slot.

Finalisation and checkpoints

For high value use cases

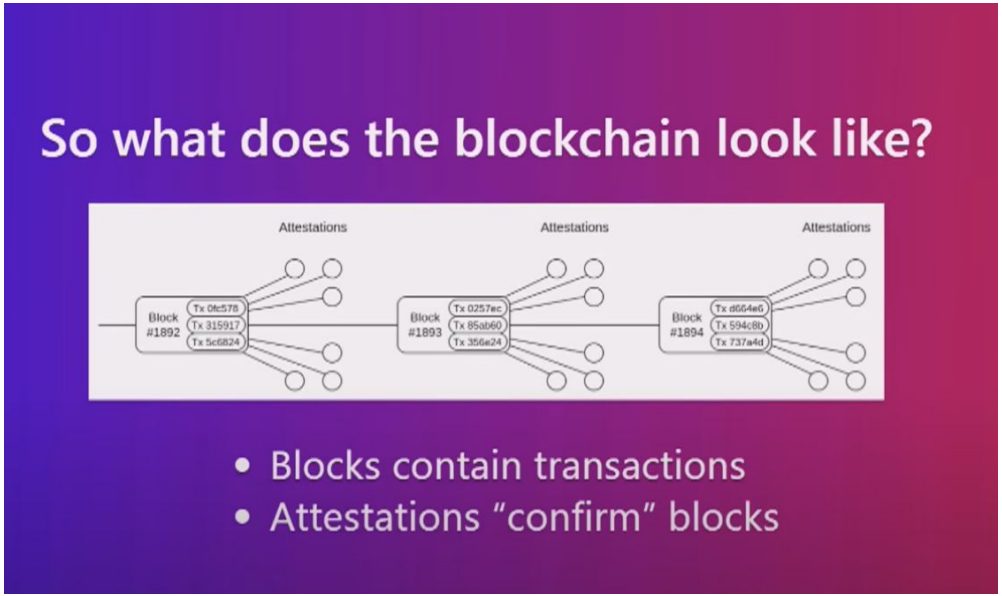
https://www.youtube.com/watch?v=UihMqcj-cgc&list=PLVB9Cbj6p2QhhQz84pNqTuIrWzi3NUdYi&index=19 , Vitalik Buterin, «Ethereum in 30 minutes»



The transaction can be considered **"finalized"** if it has become part of a chain with a "supermajority link" between two checkpoints.
Checkpoints occur at the start of each epoch and they exist to account for the fact that only a subset of active validators attest in each slot, but all active validators attest across each epoch. Therefore, it is only between epochs that a 'supermajority link' can be demonstrated (this is where 66% of the total staked ETH on the network agrees on two checkpoints).

Attestations, reorgs, 2023

https://www.youtube.com/watch?v=UihMqcj-cqc&list=PLVB9Cbj6p2QhhOz84pNqTulrWzi3NUdYi&index=19 , Vitalik Buterin, «Ethereum in 30 minutes»



Greedy Heaviest Observed Subtree (GHOST)* protocol

The miner is incentivized to do so by increased block creation reward for each referenced stale block (up to 2).

A stale block that is not included in the main chain but referenced by a main chain's block is called **uncle** or **ommer block** in Ethereum.

* Ethereum's version of GHOST protocol **does NOT consider transactions in uncle blocks** when computing the blockchain state.

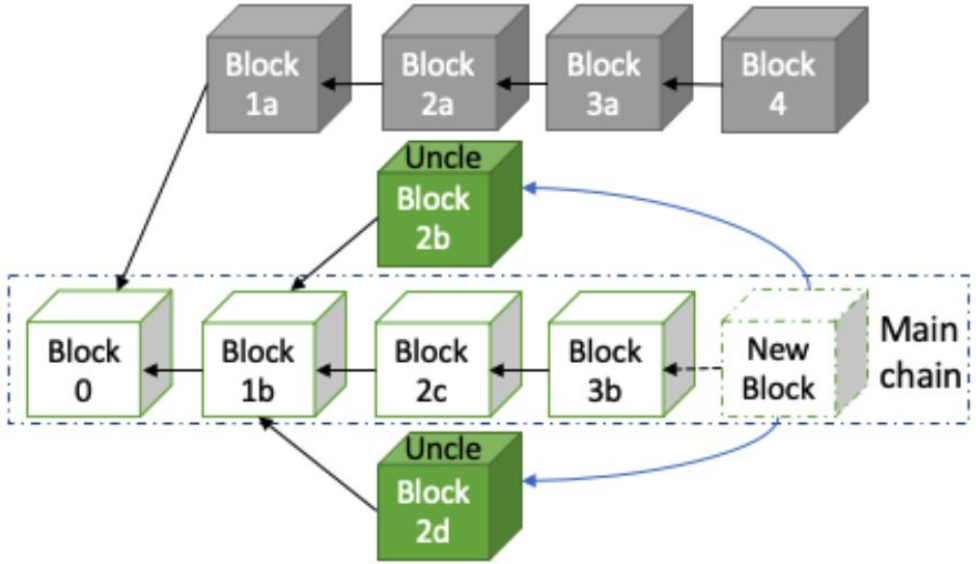


Figure 8. Ethereum chooses the main chain based on the GHOST protocol. Uncle blocks are considered when choosing the main chain (the heaviest chain rule) and their miners are rewarded in Ethereum.

Crypto-Economic Security

https://ethereum.org/en/developers/docs/consensus-mechanisms/pos

Participating as a validator also opens new avenues for users to **attack the network** for personal gain or sabotage.

To prevent this, validators miss out on ETH rewards if they fail to participate when called upon, and their existing stake can be destroyed if they behave dishonestly.

Two primary behaviors can be considered **dishonest**:

- proposing multiple blocks in a single slot (equivocating) and
- submitting contradictory attestations.

The amount of ETH slashed depends on **how many validators are also being slashed** at around the same time. This is known as the **"correlation penalty"**, and it can be minor (~1% stake for a single validator slashed on their own) or can result in 100% of the validator's stake getting destroyed (mass slashing event). The punishment getting worse by day.

Let’s discuss!

Attack price tag: **Is economic loss a good protection?**

Attacks on proposers: **Is it a good idea to punish nodes for non-availability?**

Why 32 ETH?

Smart Contracts

EVM: transaction flow

https://www.youtube.com/watch?v=UihMqcj-cqc&list=PLVB9Cbj6p2QhhOz84pNqTulrWzi3NUdYi&index=19 , Vitalik Buterin, «Ethereum in 30 minutes» 13:45

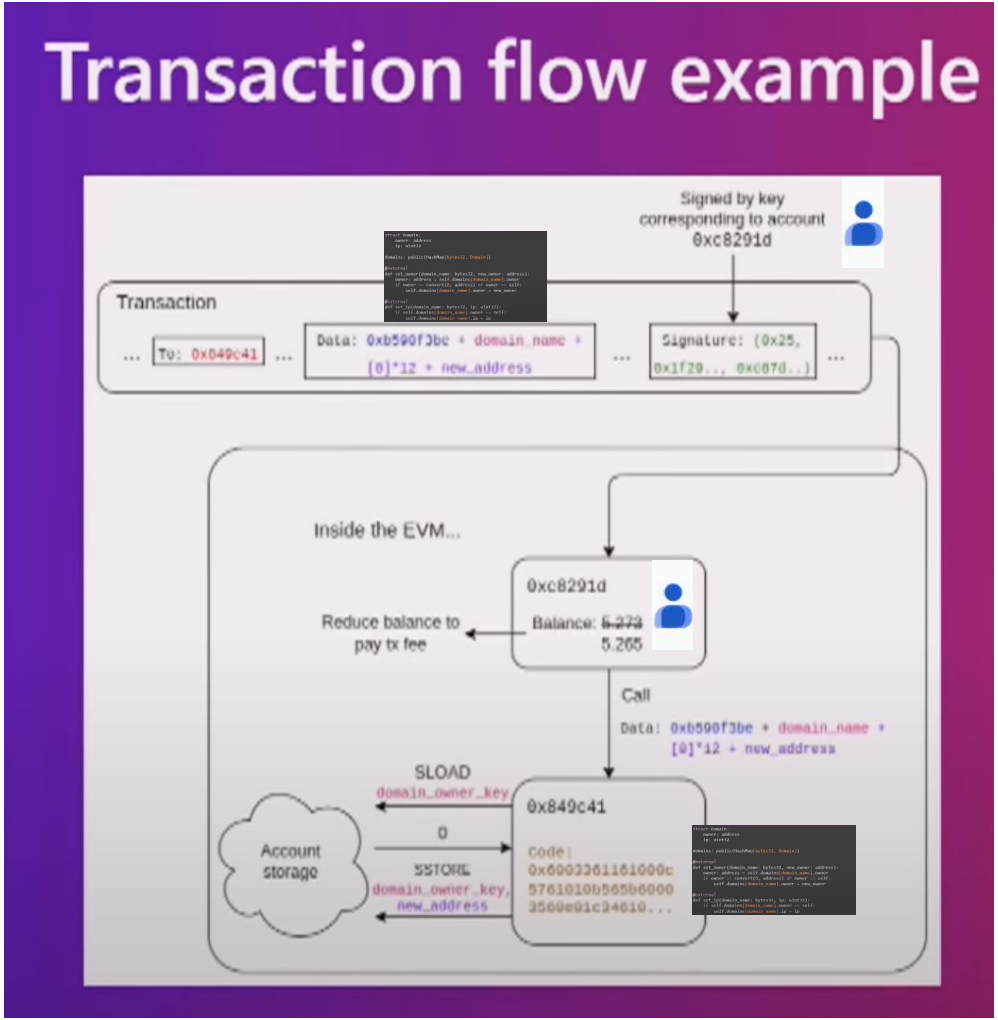
https://ethereum.org/developers/docs/evm
EVM Explained

```
struct Domain:
    owner: address
    ip: uint32

domains: public(HashMap[bytes32, Domain])

@external
def set_owner(domain_name: bytes32, new_owner: address):
    owner: address = self.domains[domain_name].owner
    if owner == convert(0, address) or owner == self:
        self.domains[domain_name].owner = new_owner

@external
def set_ip(domain_name: bytes32, ip: uint32):
    if self.domains[domain_name].owner == self:
        self.domains[domain_name].ip = ip
```



Gas

https://www.youtube.com/watch?v=UihMqcj-cqc&list=PLVB9Cbj6p2QhhOz84pNqTulrWzi3NUdYi&index=19 , Vitalik Buterin, «Ethereum in 30 minutes»

https://ethereum.org/developers/docs/gas Gas and Fees Explained

- Gas is the "unit" of resource consumption within Ethereum
- Gas cost examples:
  - A transaction costs a "base" 21,000 gas
  - Each computational step costs ~2-10 gas (usually)
  - Editing a storage slot costs 5,000 gas (20,000 if slot not filled yet)
  - Each byte of data costs 16 gas (4 if zero byte)

- If you send a transaction which gets included in a block, you must pay a fee proportional to how much gas the transaction consumed
  - `fee = (base_fee + priority_fee) * gas_used`
  - Base fee: burned, value determined by protocol
  - Priority fee: paid to block proposer
- A maximum of 30,000,000 gas can be spent in each block

Let’s discuss!

Who pays the gas and defines the budget?

Is it possible to attack EVM by publishing a malicious smart contract code?

Will the smart contract be executed at the time of publishing?

If Ethereum only stores the bytecode of a smart contract how it’s original code version available at Etherscan.io?

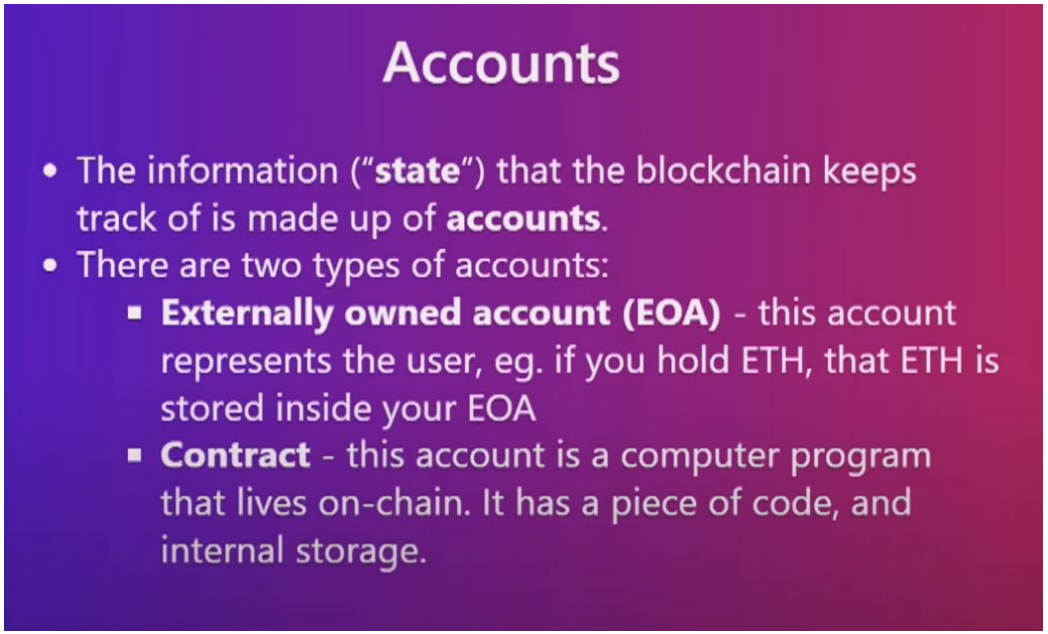
How a call to other blockchains are done from a smart contract?

Data Structure

Accounts

User balance and Contract

https://www.youtube.com/watch?v=UihMqcj-cgc&list=PLVB9Cbj6p2QhhQz84pNqTulrWzi3NUdYi&index=19 , Vitalik Buterin, «Ethereum in 30 minutes»



Data storage

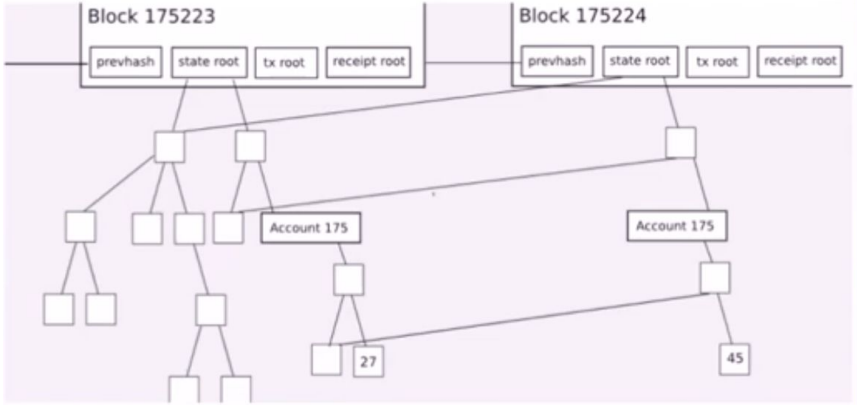
Ethereum constructs the following four tries:

State trie represents the global state that efficiently stores the mapping between all the addresses and accounts.

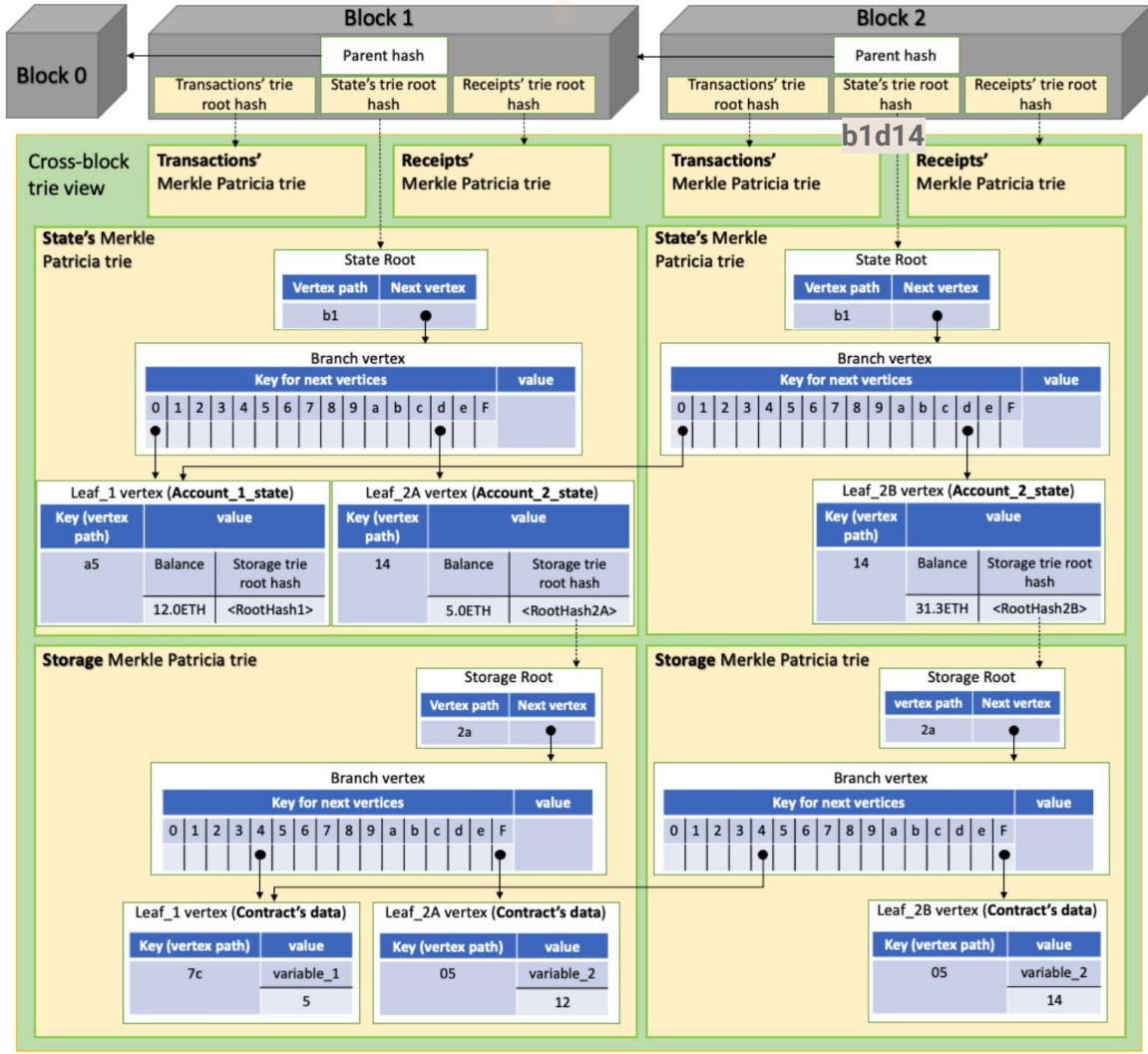
Storage trie is responsible for maintaining the relationship between the account and the corresponding balance. State trie is linked to the storage trie.

Transactions trie represents the transactions that change the state of the Ethereum.

Receipts trie represents the outcome of the successful execution of transactions.



Constructed tries across multiple blocks in Ethereum



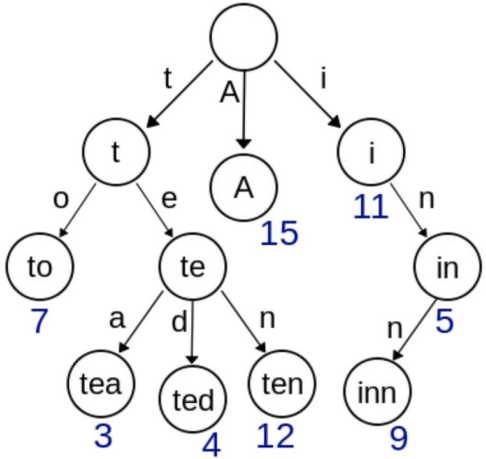
Merkle Patricia Trie

Practical Algorithm to Retrieve Info Coded in Alphanumeric

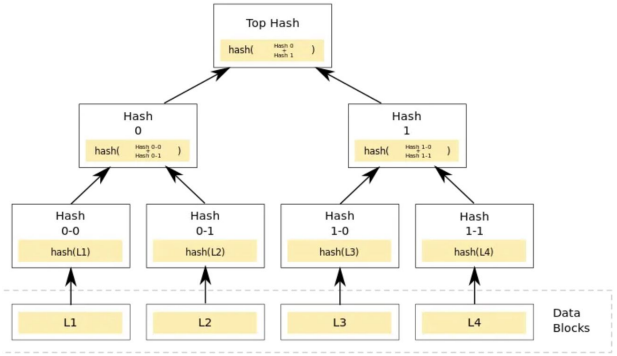
https://medium.com/codechain/modified-merkle-patricia-trie-how-ethereum-saves-a-state-e6d7555078dd

https://ethereum.org/developers/docs/data-structures-and-encoding/patricia-merkle-trie

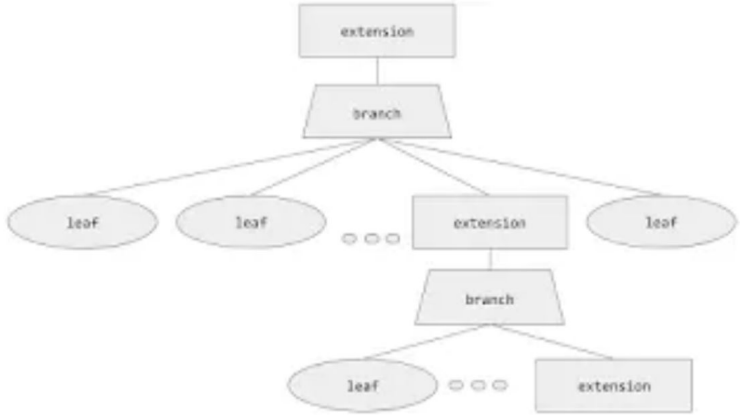
https://www.youtube.com/watch?v=QlawpoK4g5A&list=PLVB9Cbj6p2QhhQz84pNqTulrWzi3NUdYi&index=17 14:40



Example of Patricia Trie



Example of Merkle Tree

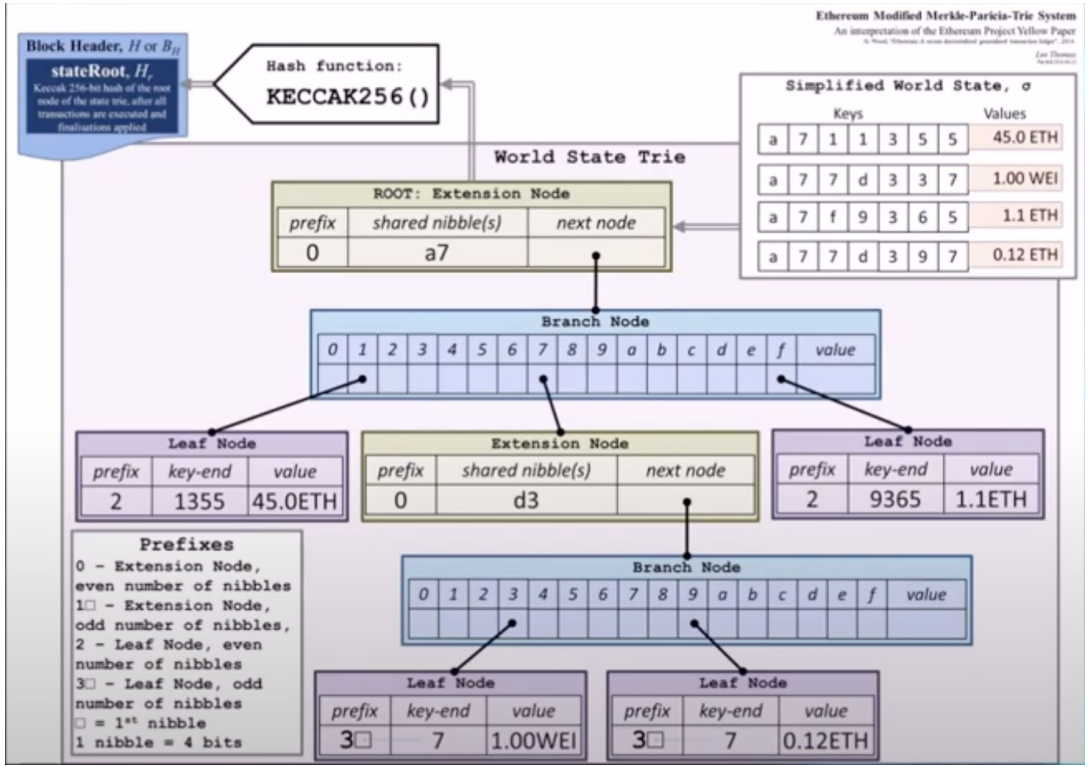


Example of Merkle Patricia Trie

Merkle Patricia Trie

Practical Algorithm to Retrieve Info Coded in Alphanumeric

Ethereum Project Yellow Paper



Let’s discuss!

Overall complexity: pros and cons?

Does storage access and search affect throughput?

What kind of search is covered by the indexes in Ethereum? Who uses it, in what circumstances?

Verkle trees and statelessness

https://ethereum.org/roadmap/verkle-trees

Verkle trees (a portmanteau of "Vector commitment" and "Merkle Trees") are a data structure that can be used to upgrade Ethereum nodes so that they can stop storing large amounts of state data without losing the ability to validate blocks. Verkle trees are a critical step on the path to stateless Ethereum clients.

Statelessness

Stateless clients are ones that **do not have to store the entire state database in order to validate incoming blocks**.

Instead of using their own local copy of Ethereum's state to verify blocks, stateless clients **use a "witness" to the state data** that arrives with the block. A **witness** is a collection of individual pieces of the state data that are required to execute a particular set of transactions, and a cryptographic proof that the witness is really part of the full data.

The witness is used instead of the state database. For this to work, the **witnesses need to be very small**, so that they can be safely broadcast across the network in time for validators to process them within a 12 second slot.

Comparison to Bitcoin

Hardware layer

Features	Bitcoin	Ethereum
Limiting resource	Processor	Memory bandwidth
Cryptopuzzle solving device	ASIC	GPU
Additional hardware for security (Research initiatives)	Hardware-based trusted execution environment (e.g. Intel SGX processor)	Hardware-based trusted execution environment

Communication layer

Features	Bitcoin	Ethereum
Granularity of dissemination	Whole network	Whole network
Entities forming the network	Full nodes	Full nodes
Communication protocol	Inventory push-gossiped & blocks pulled by full nodes	Blocks or block hashes push-gossiped & block headers and bodies pulled
Mean time to receive a block	About 12.6 seconds	About 109 milliseconds
Ordering guarantees	No guarantees	No guarantees
Privacy & security guarantees	No guarantees	Proprietary deployment can enable encrypted & authenticated messages
Initial peer discovery	Through a set of DNS seeds or direct conn. to a known full node	Through a set of bootnodes or direct conn. to a known full node
Geo-proximity in the network	135 ms (avg.) peer-to-peer latency	171 ms (avg.) peer-to-peer latency

Data manipulation layer

Features	Bitcoin	Ethereum
Consensus protocol	Chain convergence using longest-chain rule	Chain convergence using GHOST protocol
Mining difficulty	About 10 minutes to mine a block	About 12-15 seconds to mine a block
Throughput (tps)	3-7	About 15
Mining power utilization	Above 99%	Below 97%
Transaction confirmation	Probabilistic based on the number of blocks	Probabilistic based on the number of blocks
Mitigating DoS attacks	PoW	PoW + Gas price
Rich search functionality	No	No

Performance and security metrics in Ethereum

HOW DIFFERENCES FROM BITCOIN AFFECT VARIOUS PERFORMANCE AND SECURITY METRICS IN ETHEREUM

Factor	Throughput	Confirmation time	Energy consumption	Decentralization and Mining power utilization
General transactions	worsens	worsens	worsens	worsens
Complex storage	worsens	worsens	worsens	unknown
Considering uncles	unknown	unknown	unknown	improves
Less complex cryptopuzzles	improves	improves	improves	worsens
Deployment characteristics	unknown	unknown	unknown	unknown

Contract layer

Features	Bitcoin	Ethereum
Type	Scripting system	Smart contracts
Programming language	Forth-like	Several languages, Solidity as the most popular one
Turing completeness	No	Yes
Executing computing devices	Full nodes	Full nodes
Execution language	Bitcoin script	Ethereum Bytecode
Execution environment	Script interpreter	Ethereum Virtual Machine

Future development

Conservatism vs. constant evolvement

Let’s discuss!

Conservatism vs. constant evolvement: pros and cons?