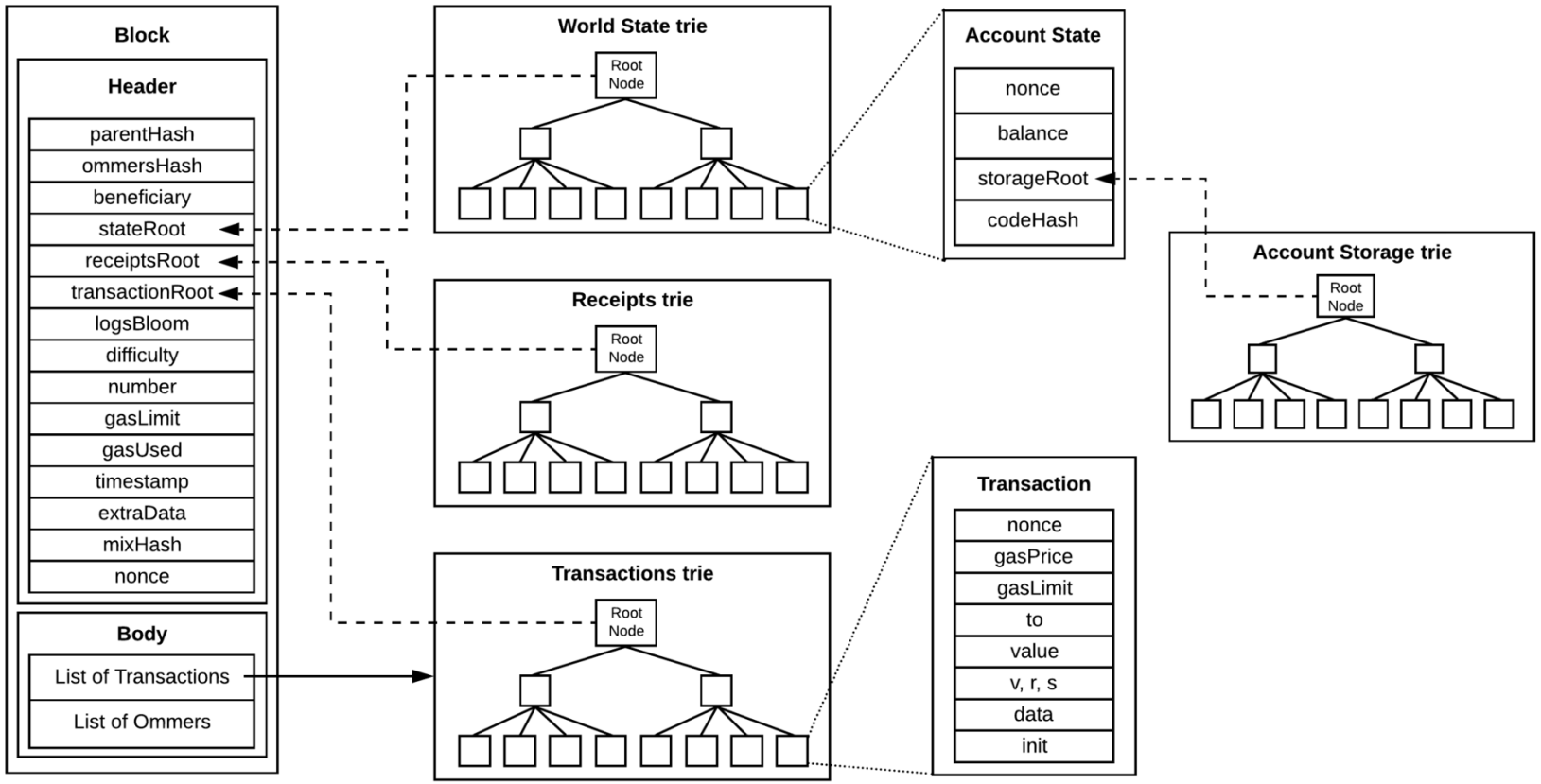
CSE446: Blockchain & Cryptocurrencies

Lecture – 14: Ethereum - 3

Agenda

- Ethereum Block
- Ethereum Blockchain
- Ethereum Consensus

Ethereum block



Block, transaction, account state objects and Ethereum tries

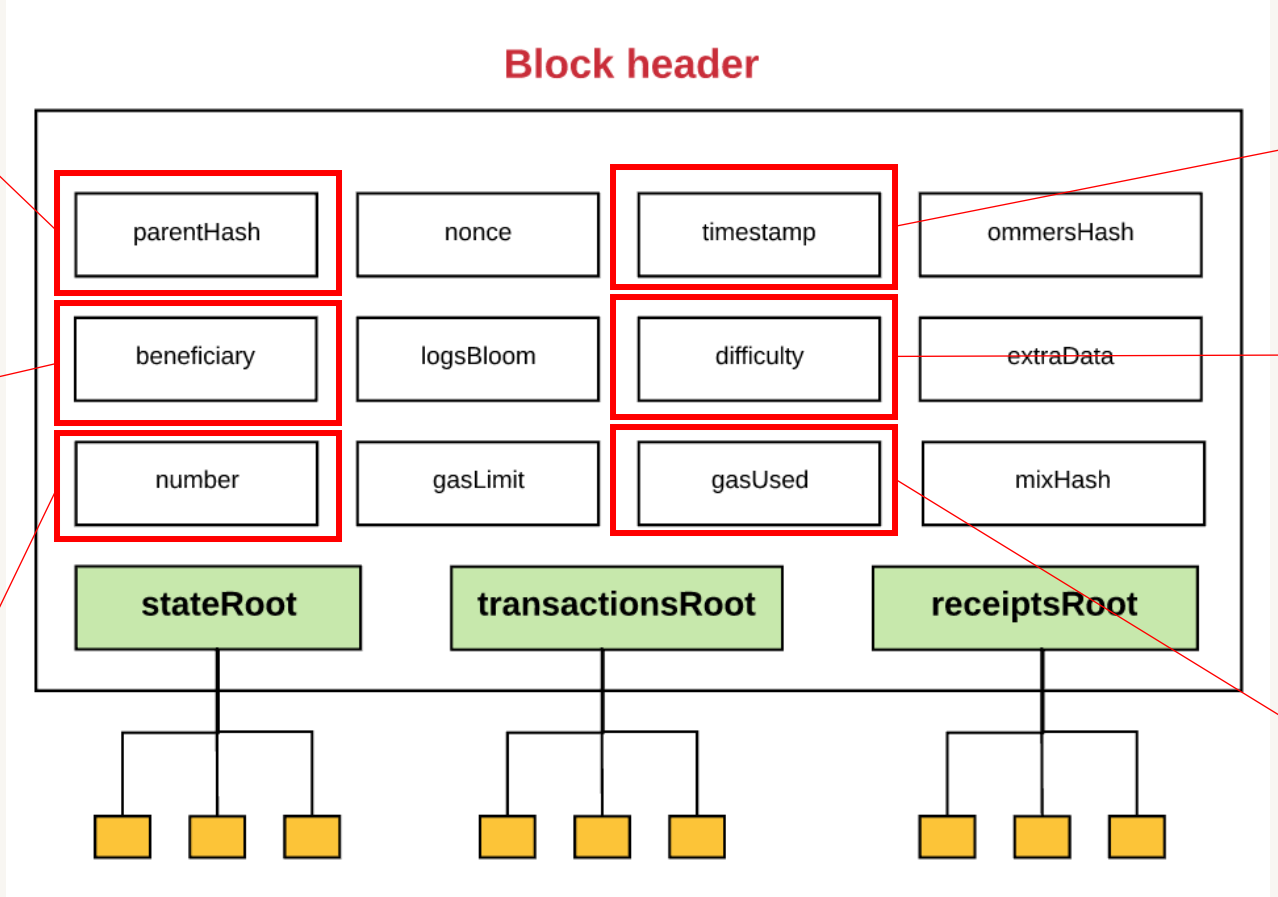
https://miro.medium.com/max/4800/1*BVvVSUIj4A9fmmVBNs4WhQ.png

Ethereum block header

A hash of the parent block's header

The account address that receives the fees for mining this block

The count of the current block



The time stamp of this block's inception

The difficulty level of this block

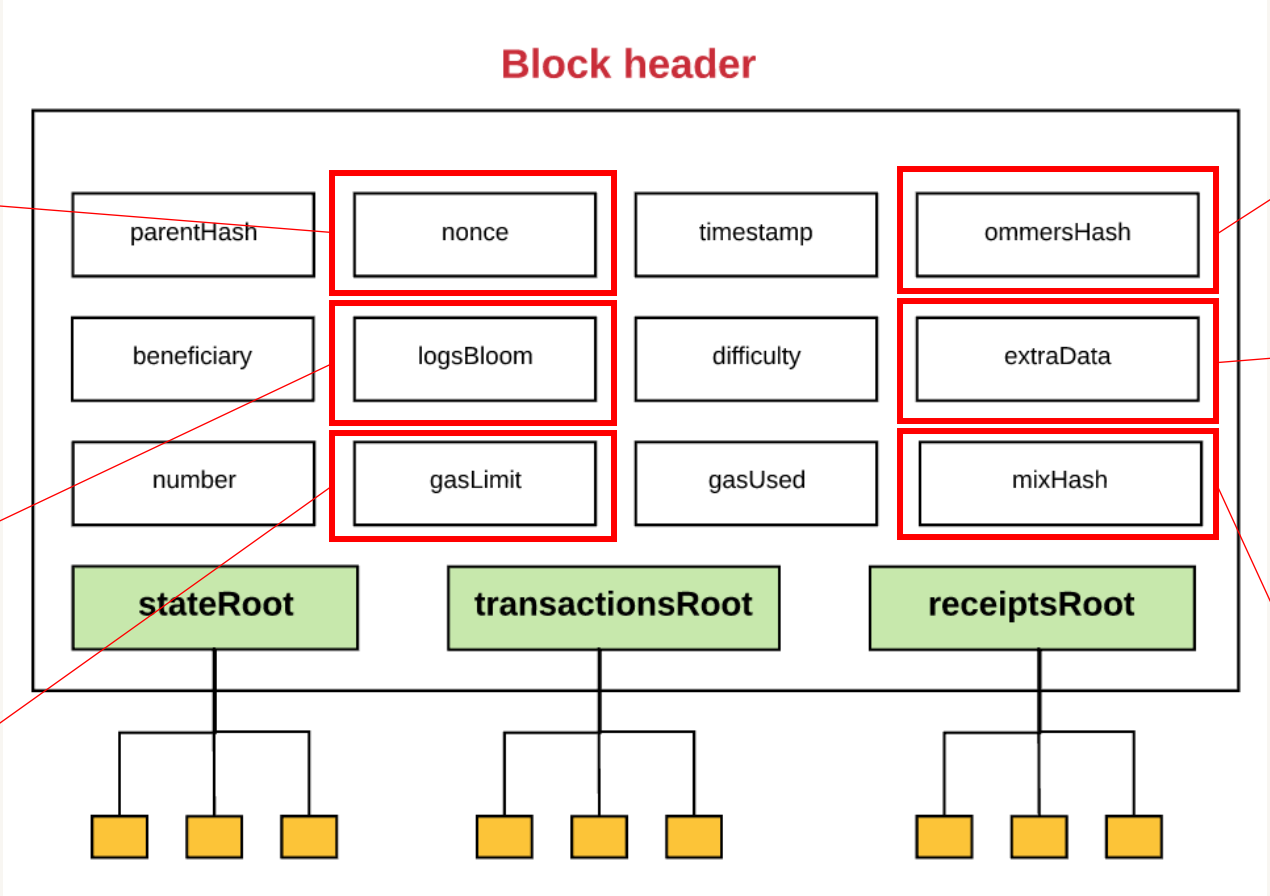
The sum of the total gas used by transactions in this block

Ethereum block header

A value that, when combined with the mixHash, proves that this block has carried out enough computation

A Bloom Filter(data structure) that consists of log information

The current gas limit per block



A hash of the current block's list of ommers

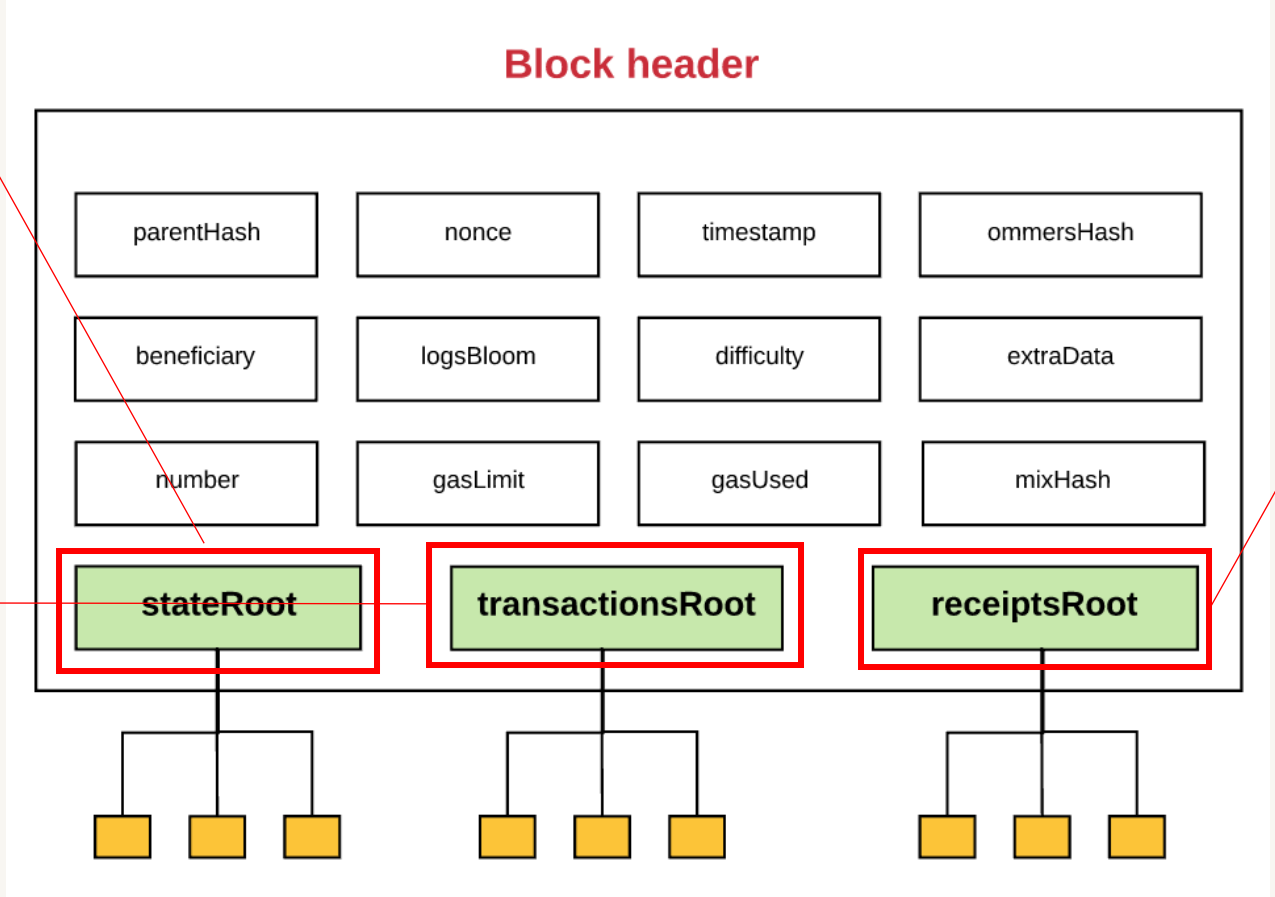
Extra data related to this block

A hash that, when combined with the nonce, proves that this block has carried out enough computation

Ethereum block header

The hash of the root node of the state tree

The hash of the root node of the tree that contains all transactions listed in this block



The hash of the root node of the tree that contains the receipts of all transactions listed in this block

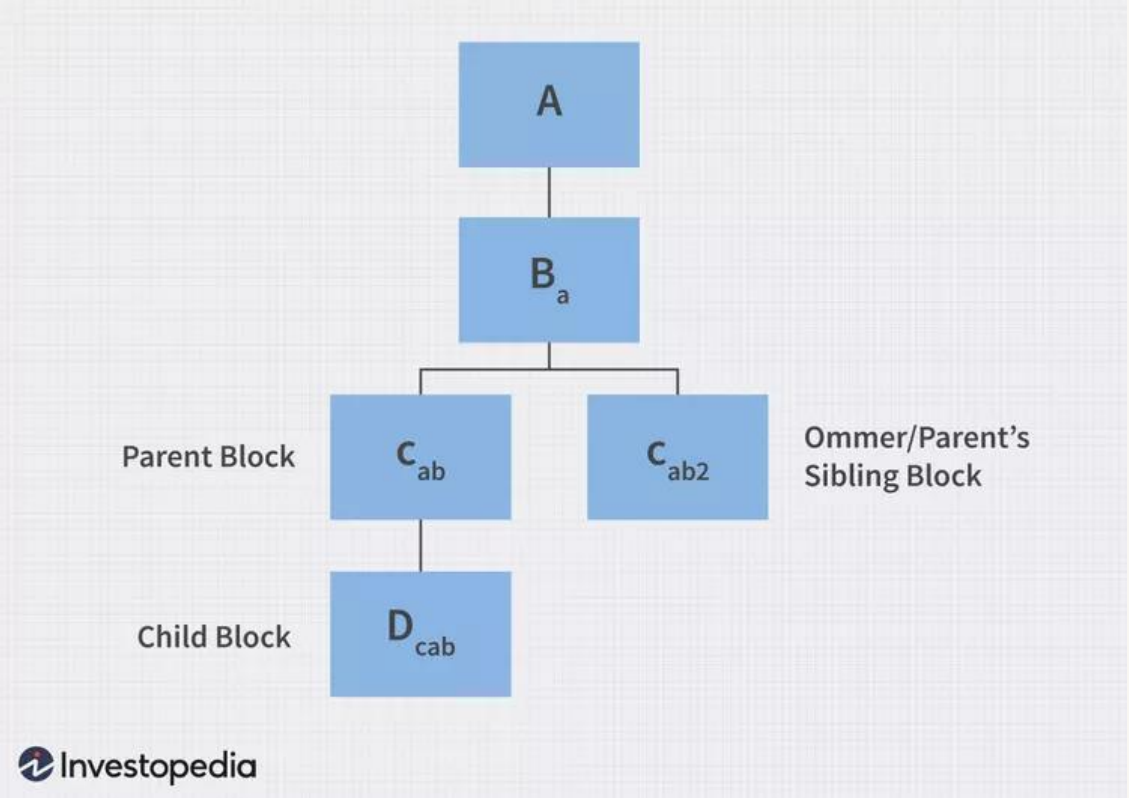
Ethereum ommers

- It is possible for two blocks to be created simultaneously by a network
- When this happens, a fork happens and eventually one block is left out
- This leftover block is called an ommer block
- In the past, they were called uncle blocks
  - referring to the familial relationships used to describe block positions within a blockchain
- In Bitcoin, there is no reward for this omner block
  - Ethereum provides a minimum amount of reward to the omner miner

https://www.investopedia.com/terms/u/uncle-block-cryptocurrency.asp

Ethereum ommers

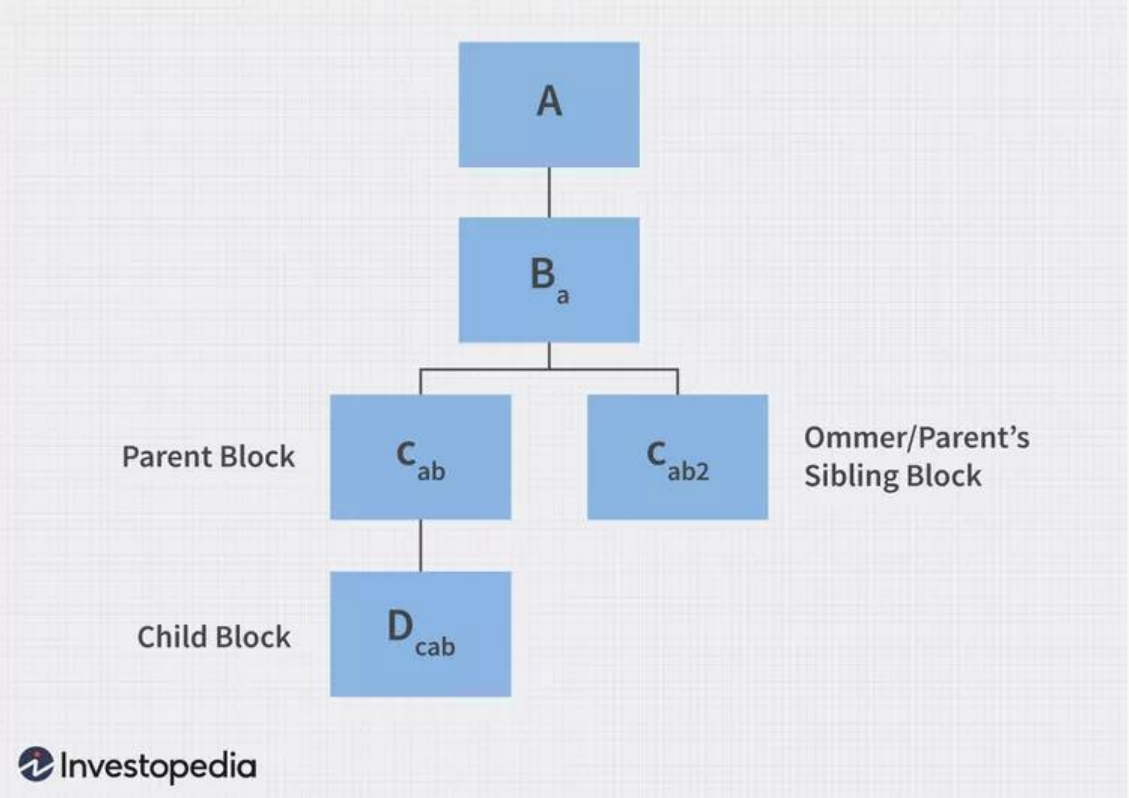
- An ommer is a block whose parent is equal to the current block's parent's parent
- Block times in Ethereum are around 15 sec
  - This is much lower than that in Bitcoin (10 min)
- This enables faster transaction



https://www.investopedia.com/thmb/Bf315VzUb4nwViDSK36Piv55D1Q=/750x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/INV-uncle-block-cryptocurrency-fdc913eee3bb40aebba7e499bfceeada.jpg

Ethereum ommers

- But there are more competing blocks, hence a higher number of orphaned blocks
- The purpose of ommers is to help reward miners for including these orphaned blocks
  - Compensating the miners for their computation

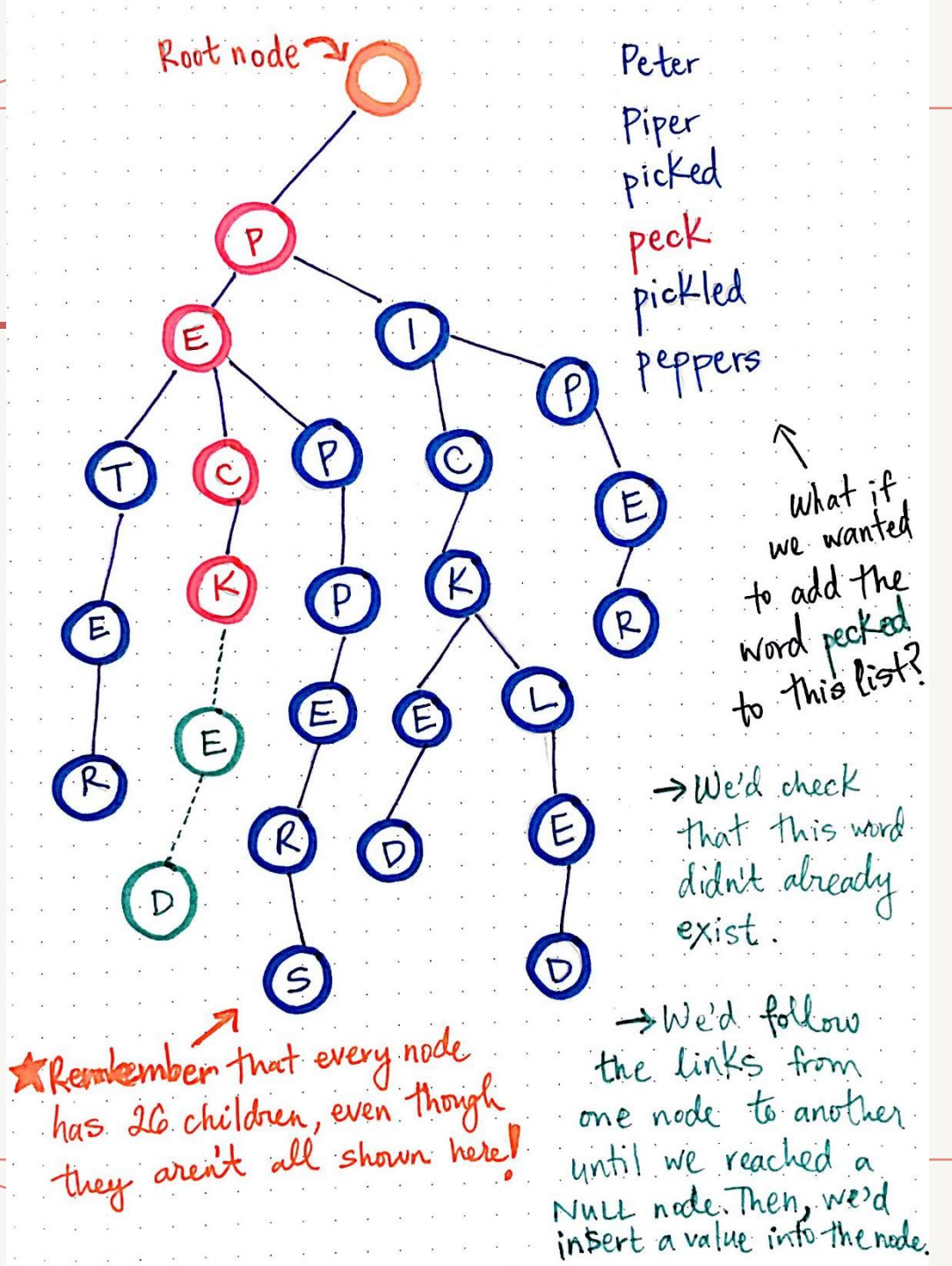


https://www.investopedia.com/thmb/Bf315VzUb4nwViDSK36Piv55D1Q=/750x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/INV-uncle-block-cryptocurrency-fdc913eee3bb40aebba7e499bfceeada.jpg

Trie

A **trie** is a tree-like data structure wherein the nodes of the tree store the entire alphabet, and strings/words can be retrieved by traversing down a branch path of the tree.

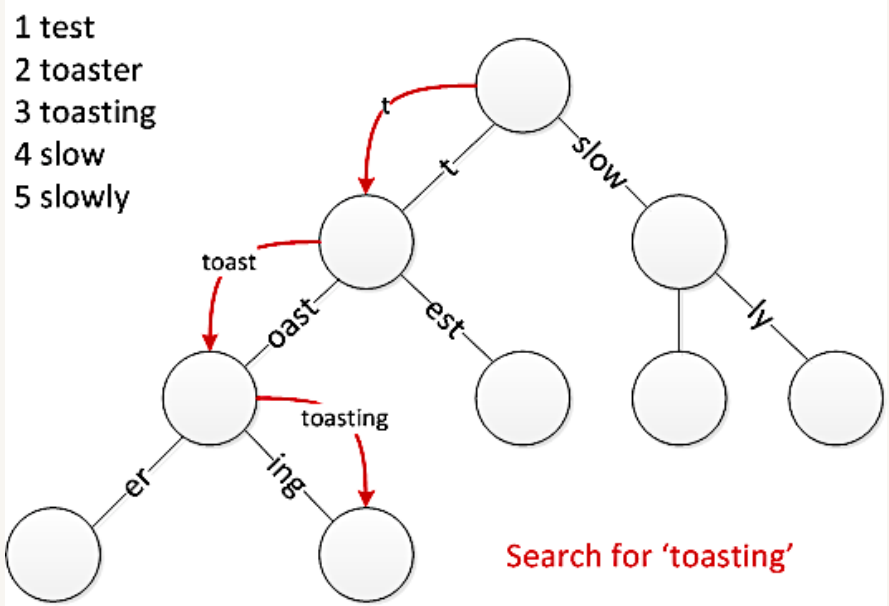
https://cdn-images-1.medium.com/max/1600/1*rkanFIU4G_tmuC939_txhA.jpeg



https://cdn-images-1.medium.com/max/1200/1*sZOrNXzlQICW5ePpav1-g.jpeg

Patricia trie

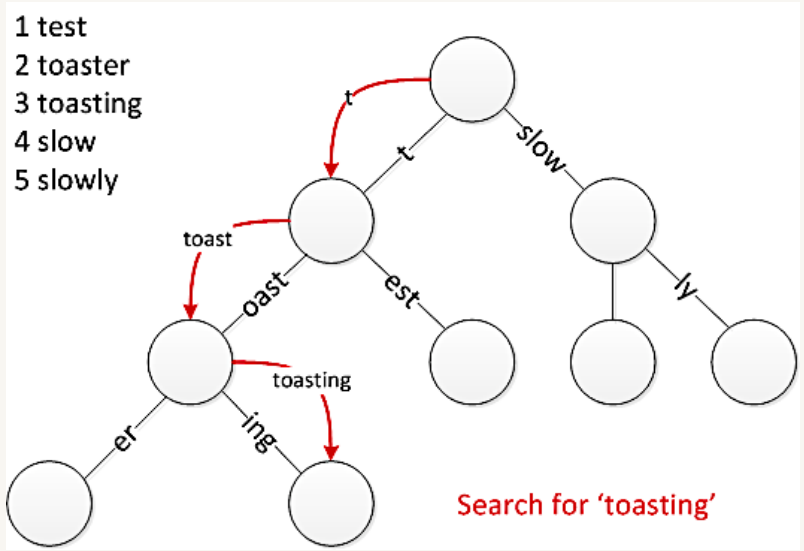
- A Patricia (Practical Algorithm To Retrieve Information Coded In Alphanumeric) trie is a binary radix trie
  - binary choice at each node when traversing the trie
- It is a data structure which uses a key as a path so the nodes that share the same prefix can also share the same path



https://i.stack.imgur.com/d2w07.png

PT

- This structure is fastest at finding common prefixes, simple to implement, and requires small memory
- It is commonly used for implementing routing tables, systems that are used in low specification machines like the router



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

Merkle Patricia Trie (MPT)

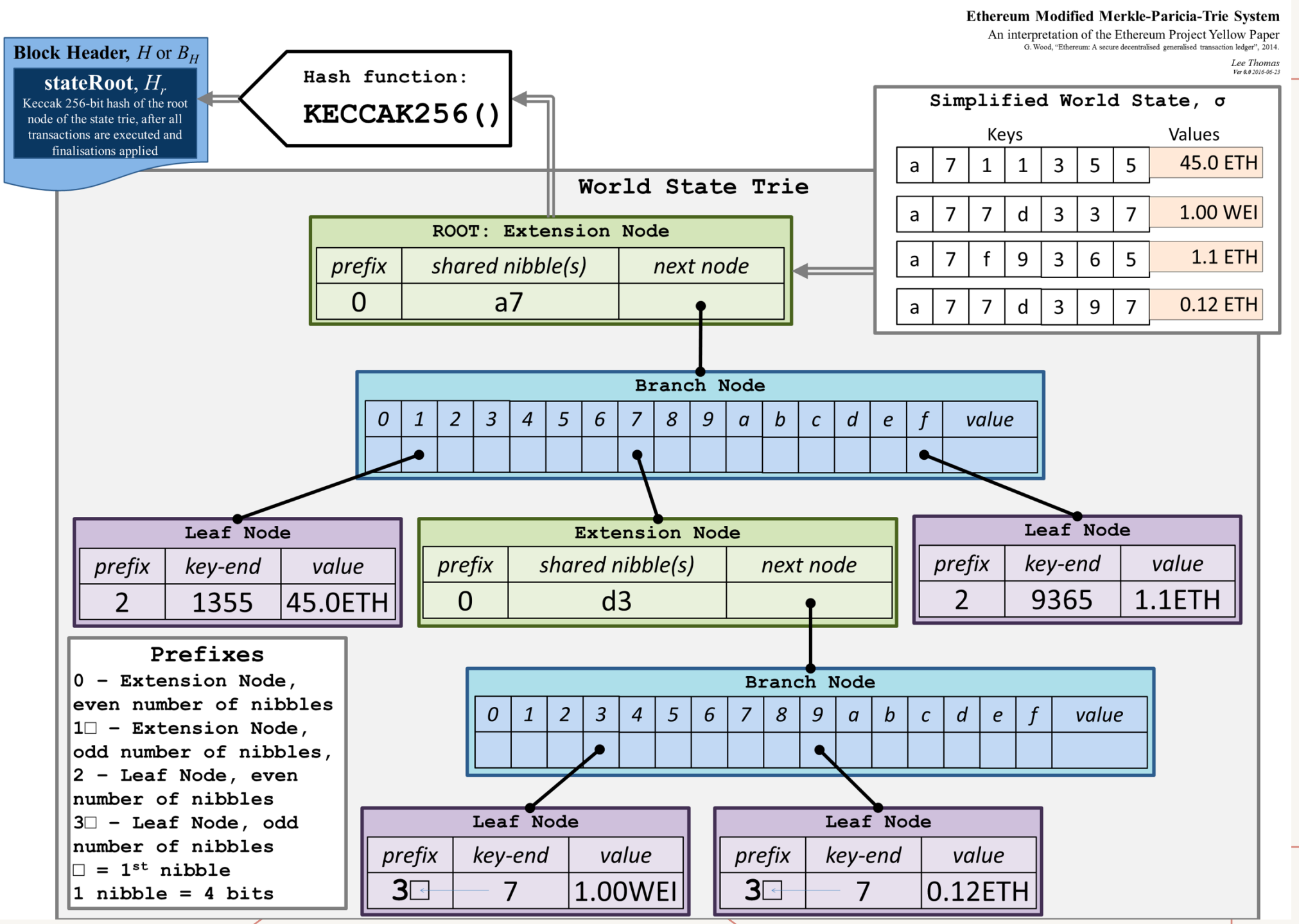
- In Ethereum the concept of PT is modified to Merkle Patricia trie
  - the root node becomes a cryptographic fingerprint of the entire data structure, just like a Merkle tree
- An MPT is a data structure for storing key value pairs in a cryptographically authenticated manner
- In the MPT every node has a hash value
- This hash is also used as the key that refers to the node
- The content of the node is stored in the Ethereum blockchain
- A node that does not have a child node is called a leaf node

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

MPT

- Nodes in MPT can have 16 child nodes
  - Plus it has its value, totalling 17 fields
- In Ethereum, hexadecimal is used – a 16 characters "alphabet”
- Note a hex character is referred to as a "nibble”
- Three different node types: extension, branch and leaf

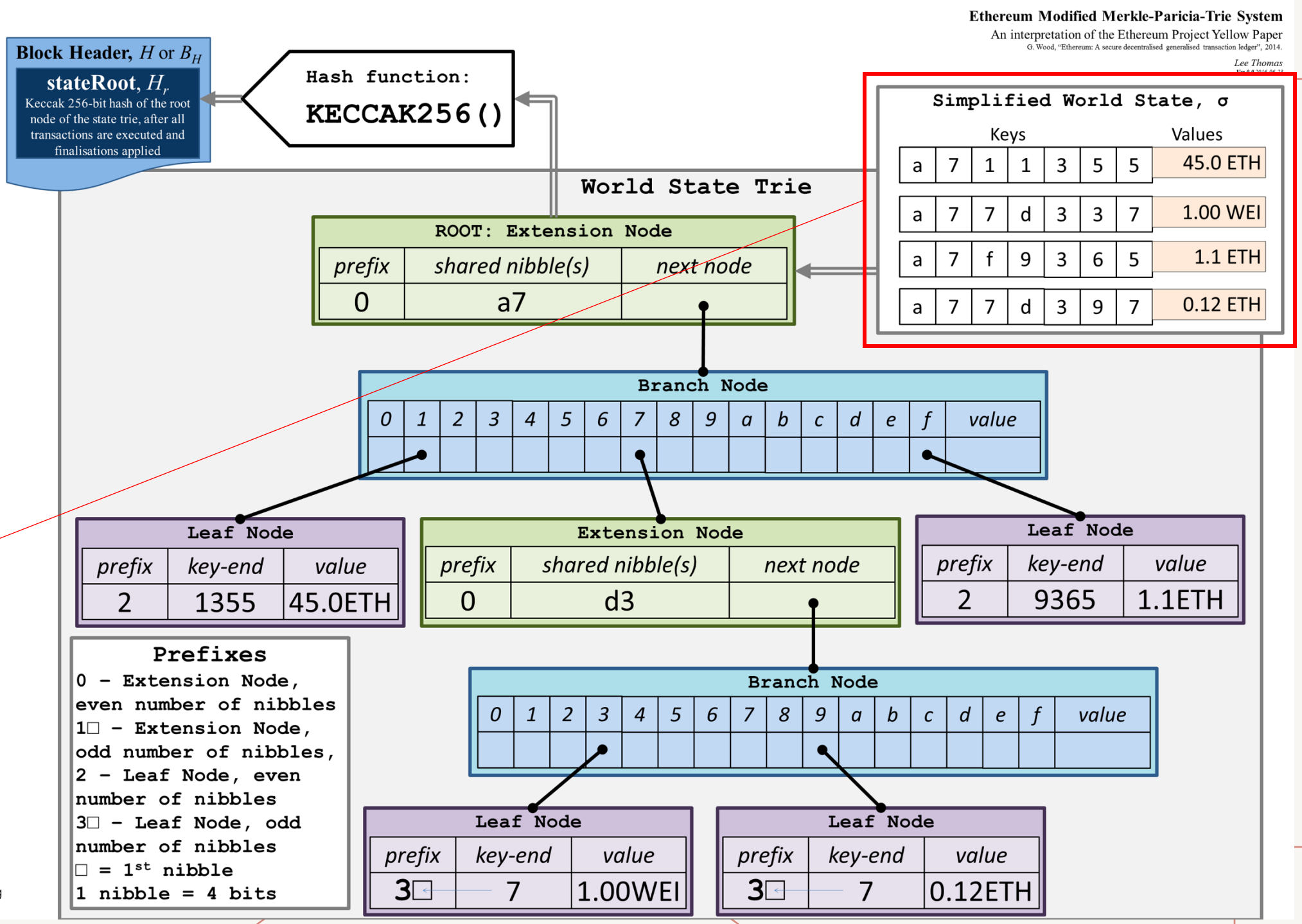
MPT



https://i.stack.imgur.com/YZGxe.png

MPT

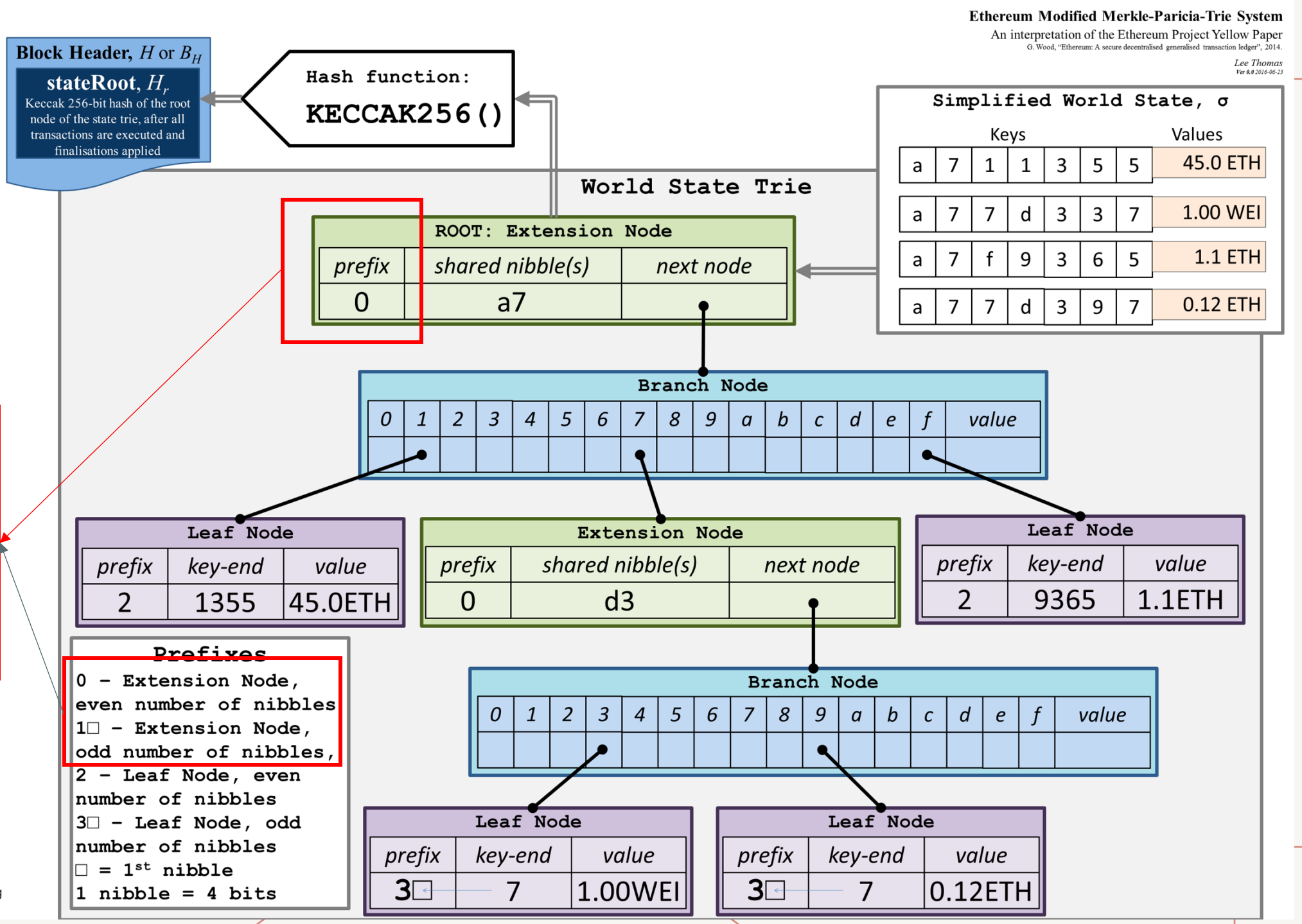
- This represents a simplified world state of accounts
- Instead of storing each account in the blockchain, MPT is used



https://i.stack.imgur.com/YZGxe.png

MPT

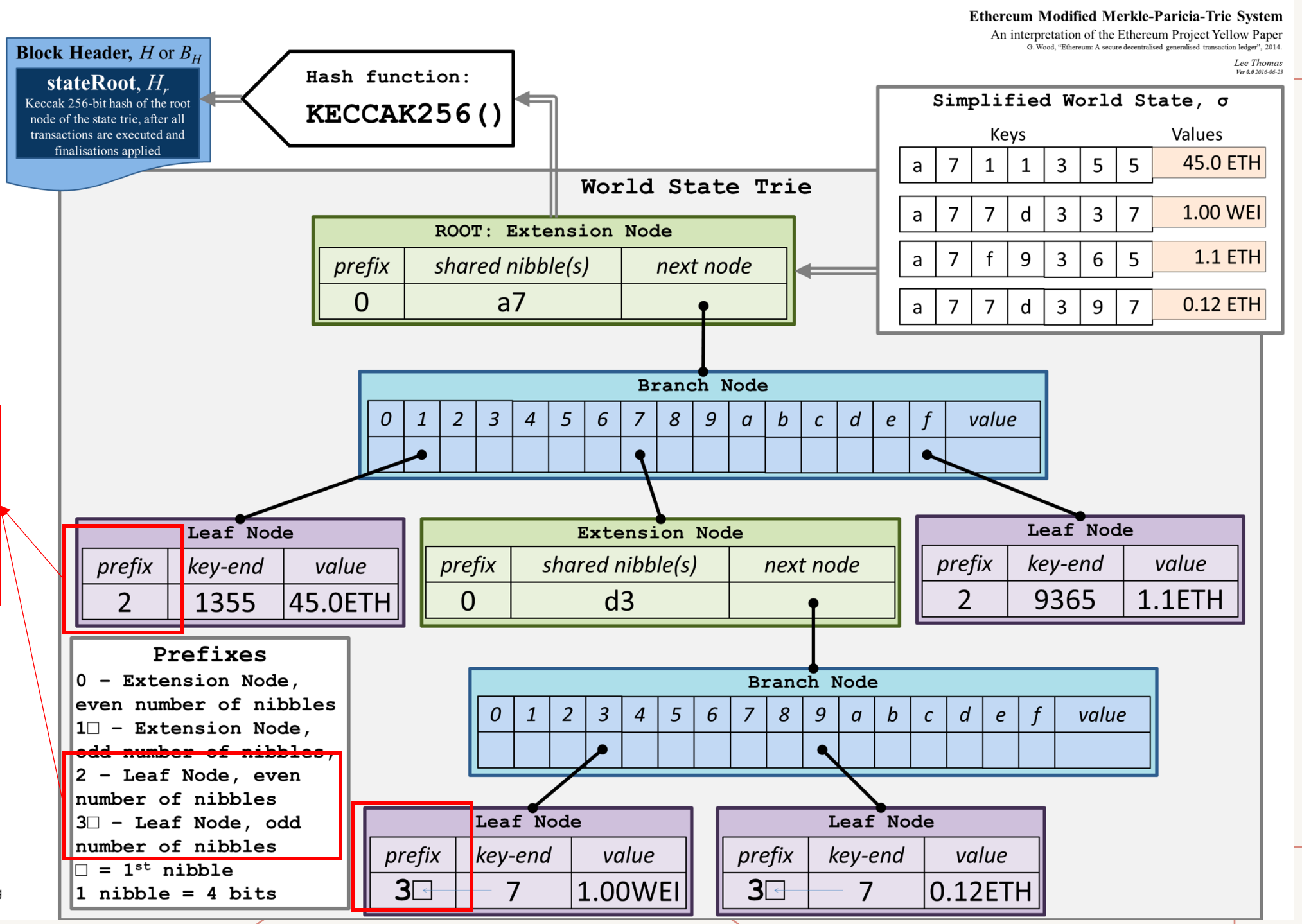
- Prefix determines the type of node
- 0/1 indicates an extension node
- If there are even number of nibbles then 0, otherwise 1



https://i.stack.imgur.com/YZGxe.png

MPT

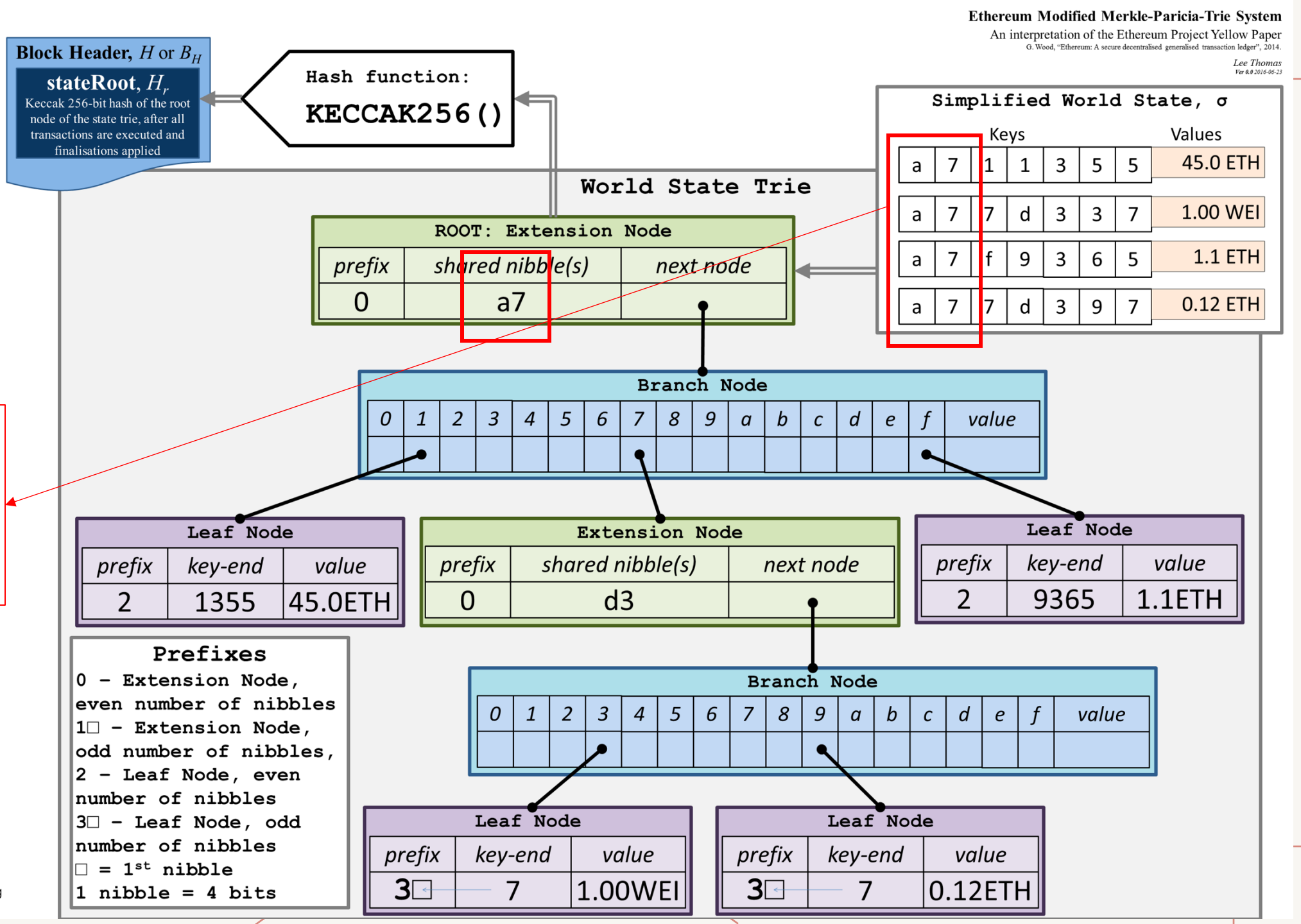
- 2/3 indicates a leaf node
- If there are even number of nibbles then 2, otherwise 3



https://i.stack.imgur.com/YZGxe.png

MPT

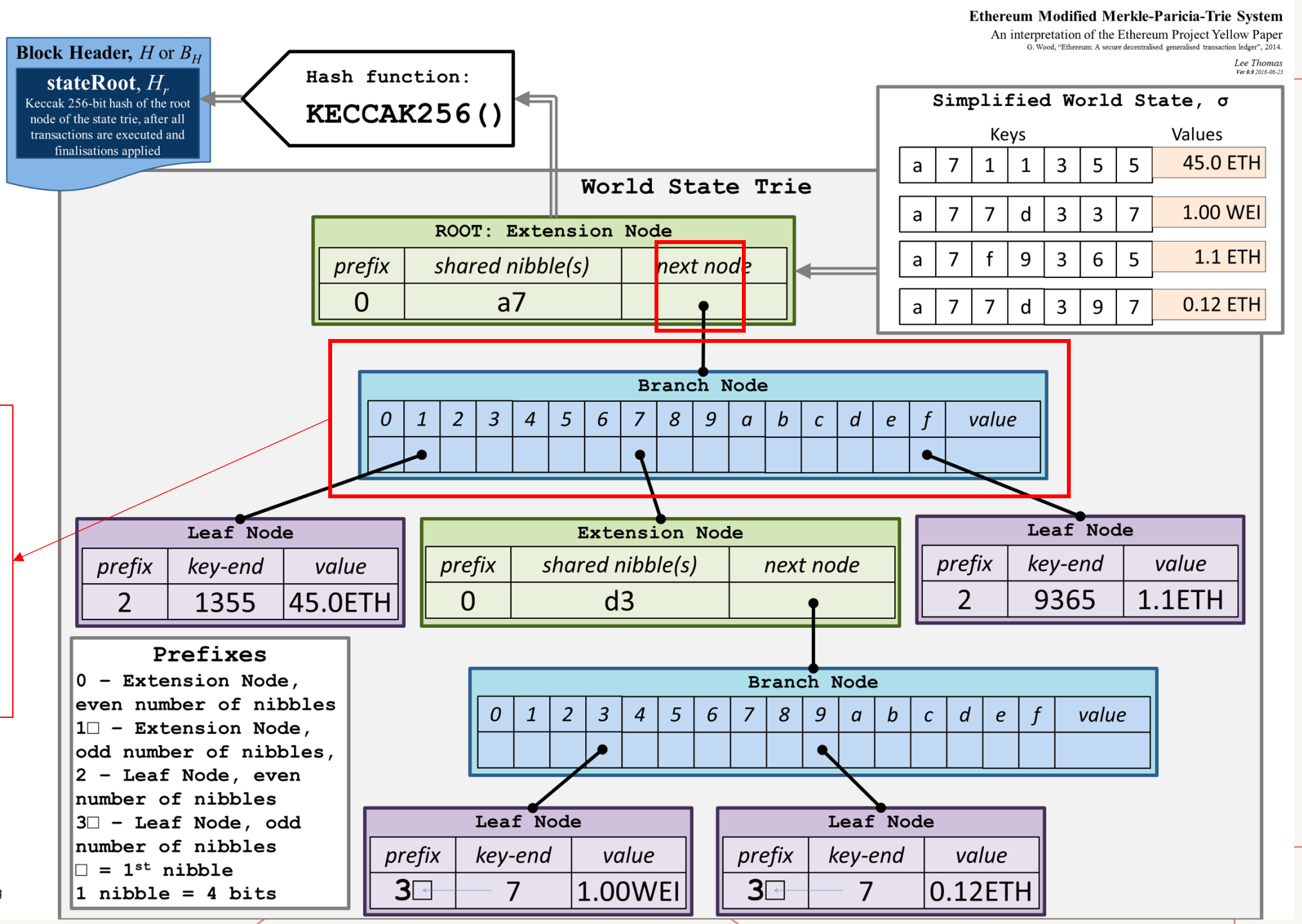
- All these accounts share a common prefix: a7
- That is why a7 remains in the root



https://i.stack.imgur.com/YZGxe.png

MPT

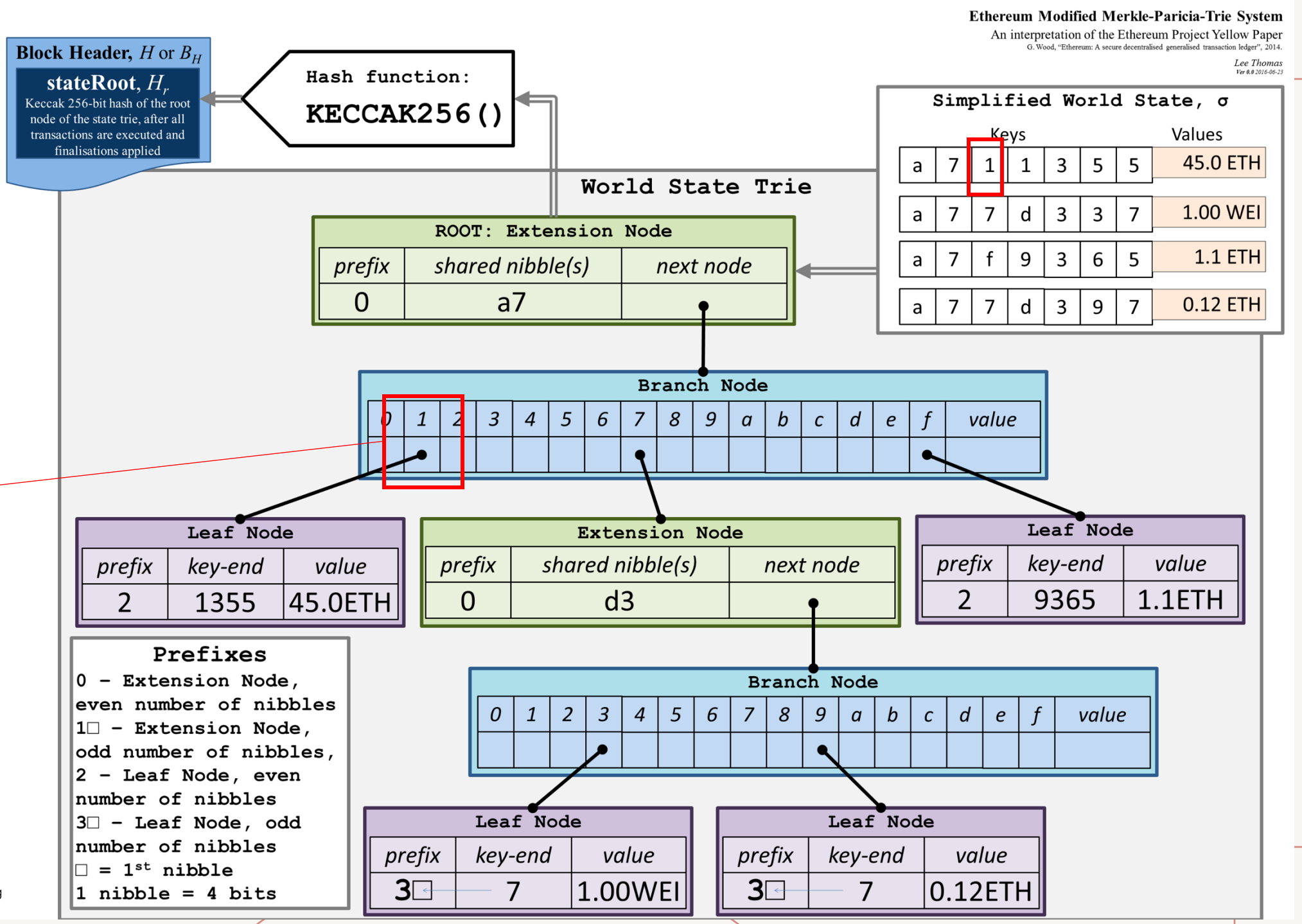
- Next node field in the extension node points to a branch node
- A branch node has 16 hexadecimal characters and a value field



https://i.stack.imgur.com/YZGxe.png

MPT

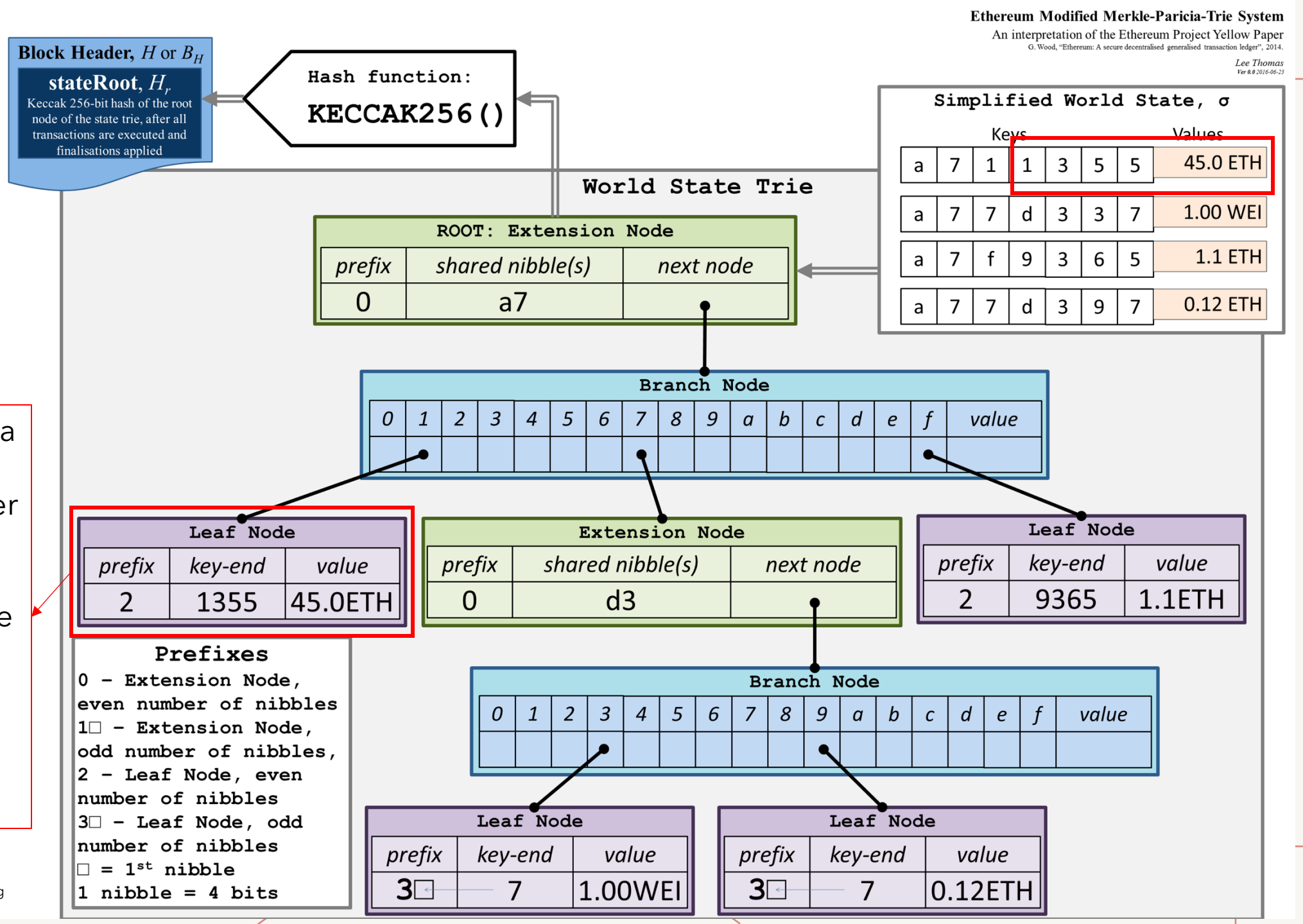
- Each field except the value field represents a key character



https://i.stack.imgur.com/YZGxe.png

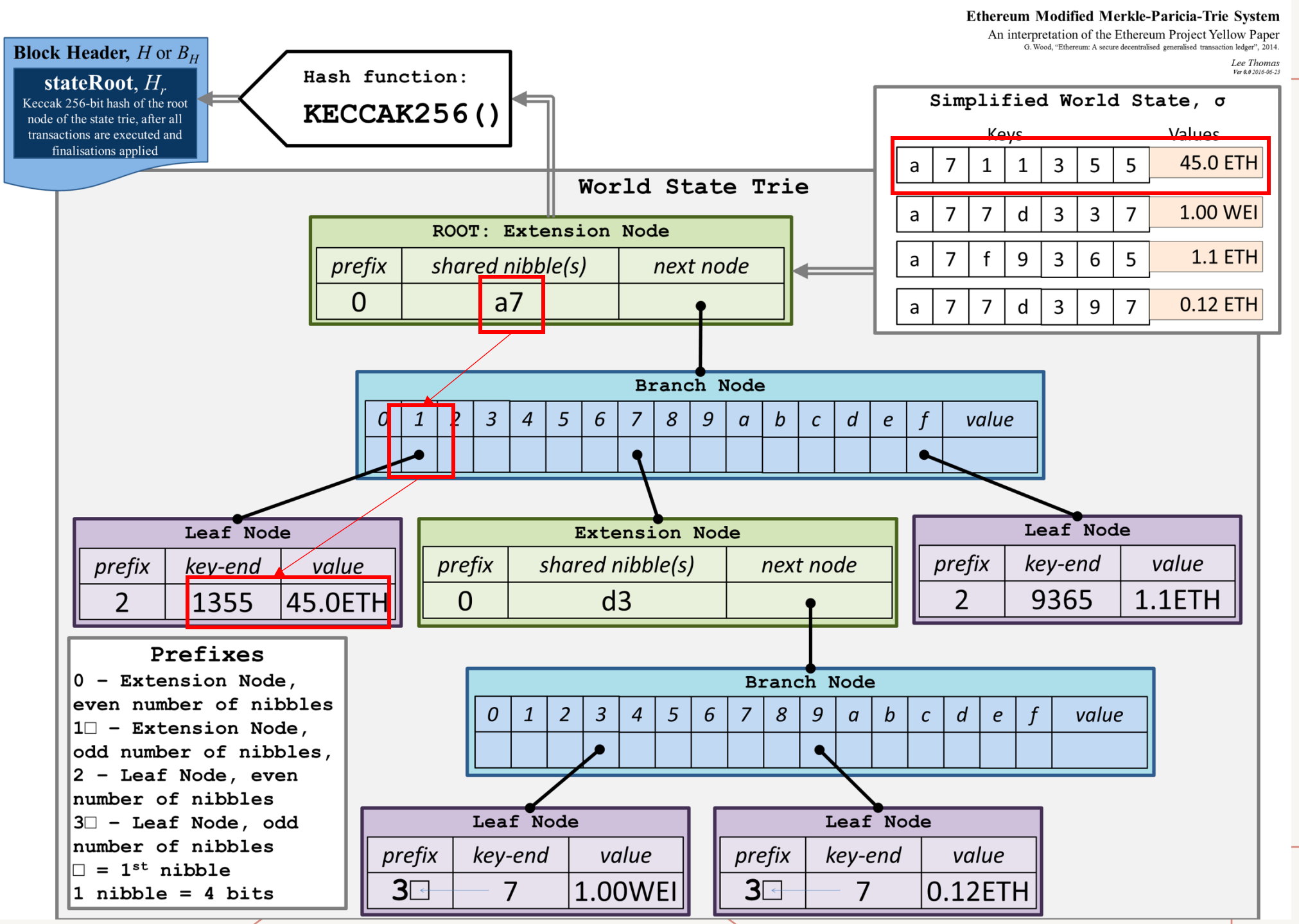
MPT

- Each leaf node has a prefix indicating its even or odd number of nibbles
- A key-end to store the last values of the key
- Finally the corresponding balance for the account



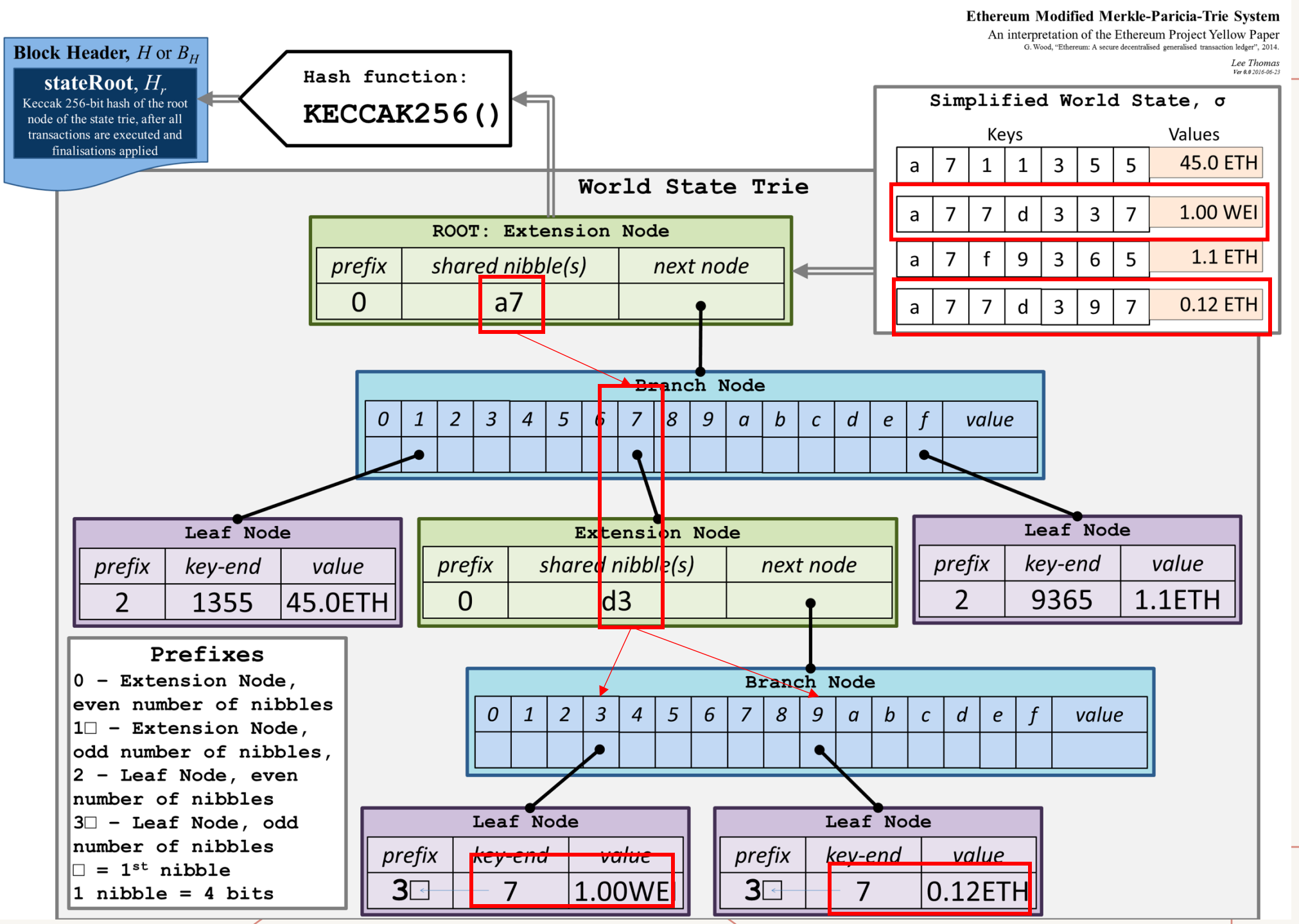
https://i.stack.imgur.com/YZGxe.png

MPT



https://i.stack.imgur.com/YZGxe.png

MPT

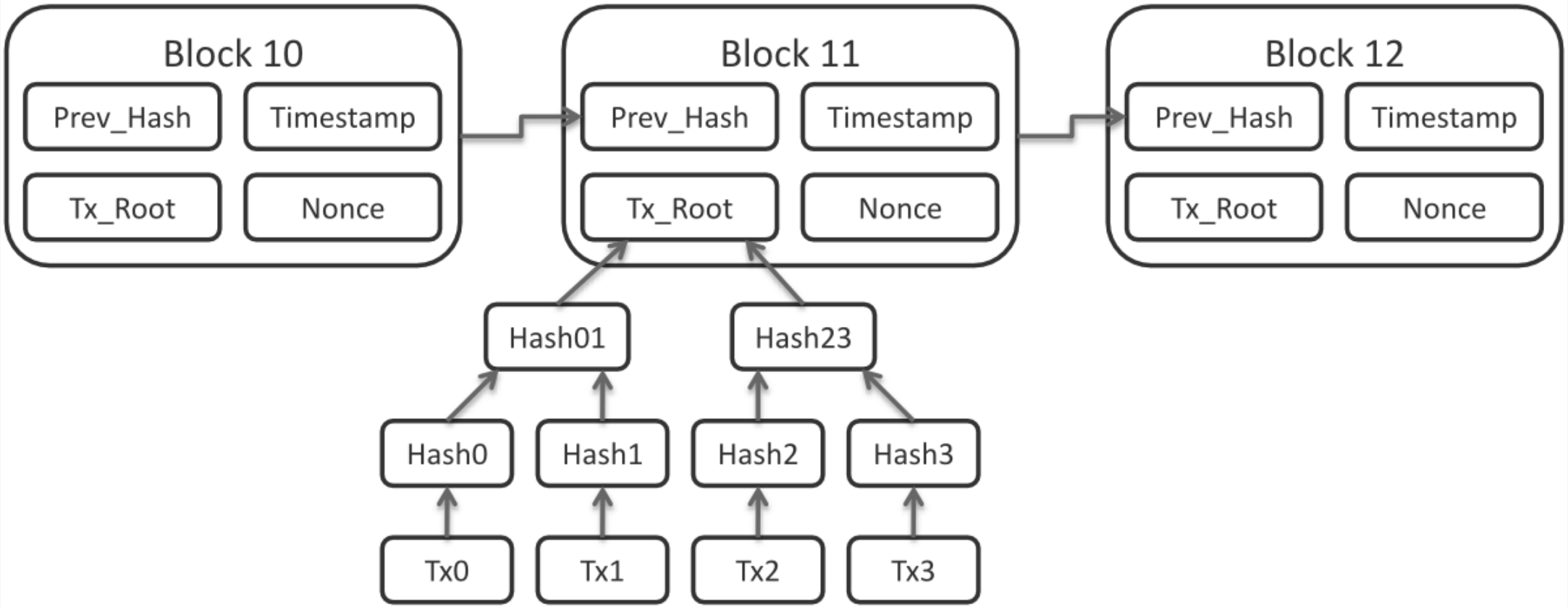


https://i.stack.imgur.com/YZGxe.png

Ethereum tries

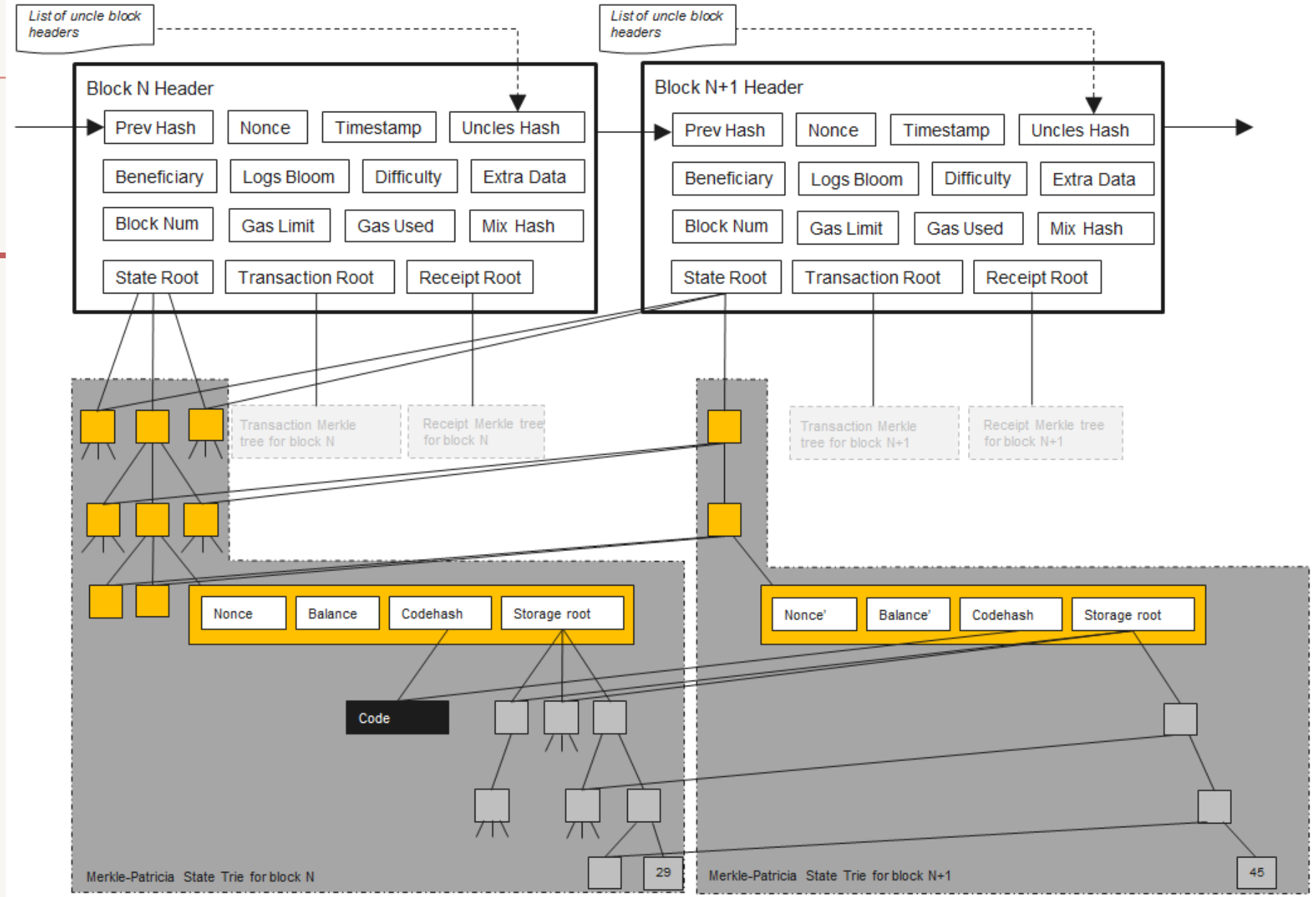
- There are four different tries used in Ethereum
  - State Trie
    - Contains an account information with respect to their address
  - Transaction Trie
    - Contains transaction information
  - Transaction Receipt Trie
    - Contains information regarding transaction receipt
  - Account storage Trie
    - Contains storage information with respect a smart contract

Bitcoin blockchain



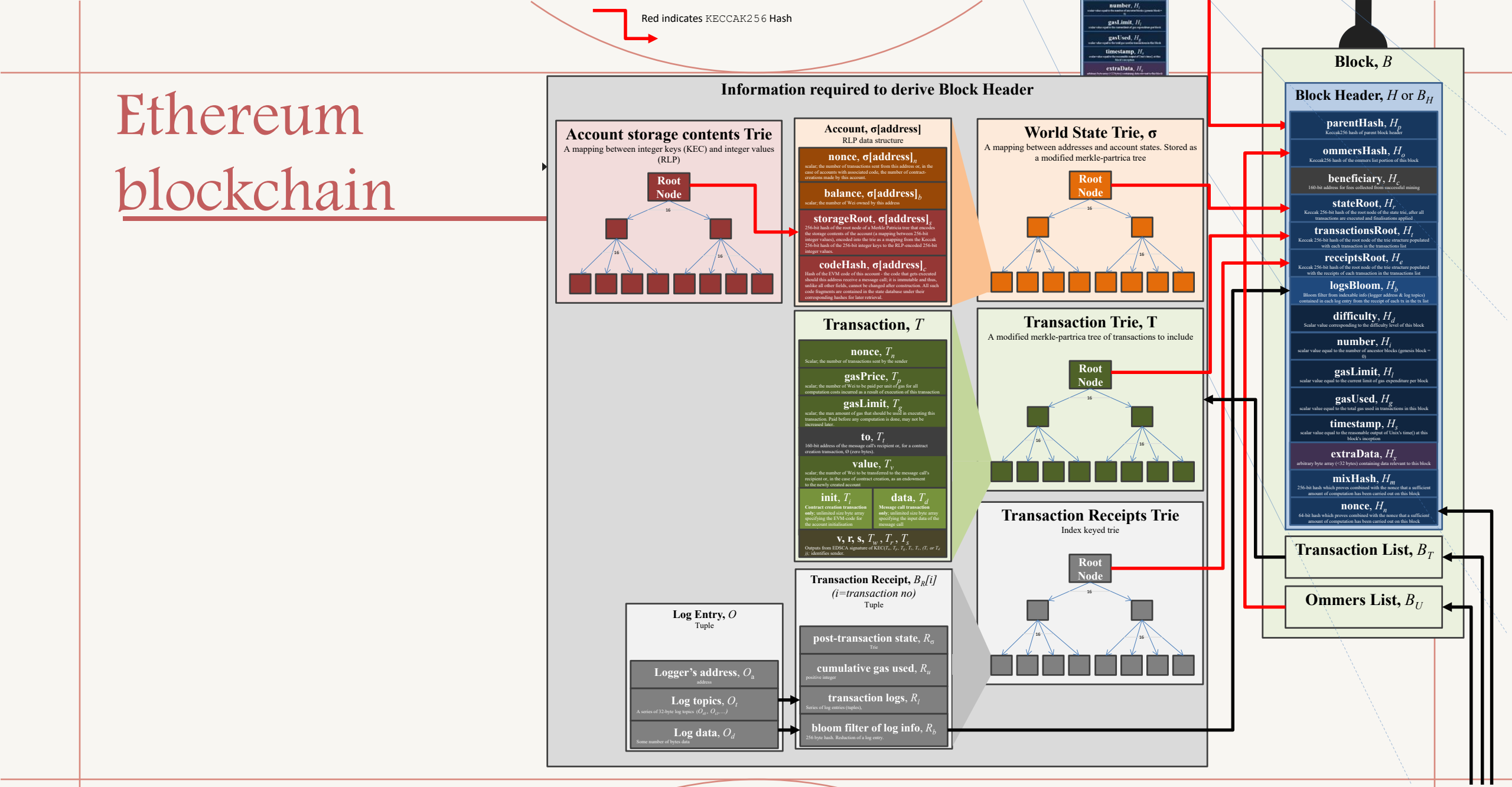
https://i.stack.imgur.com/HrKX0.png

Ethereum blockchain



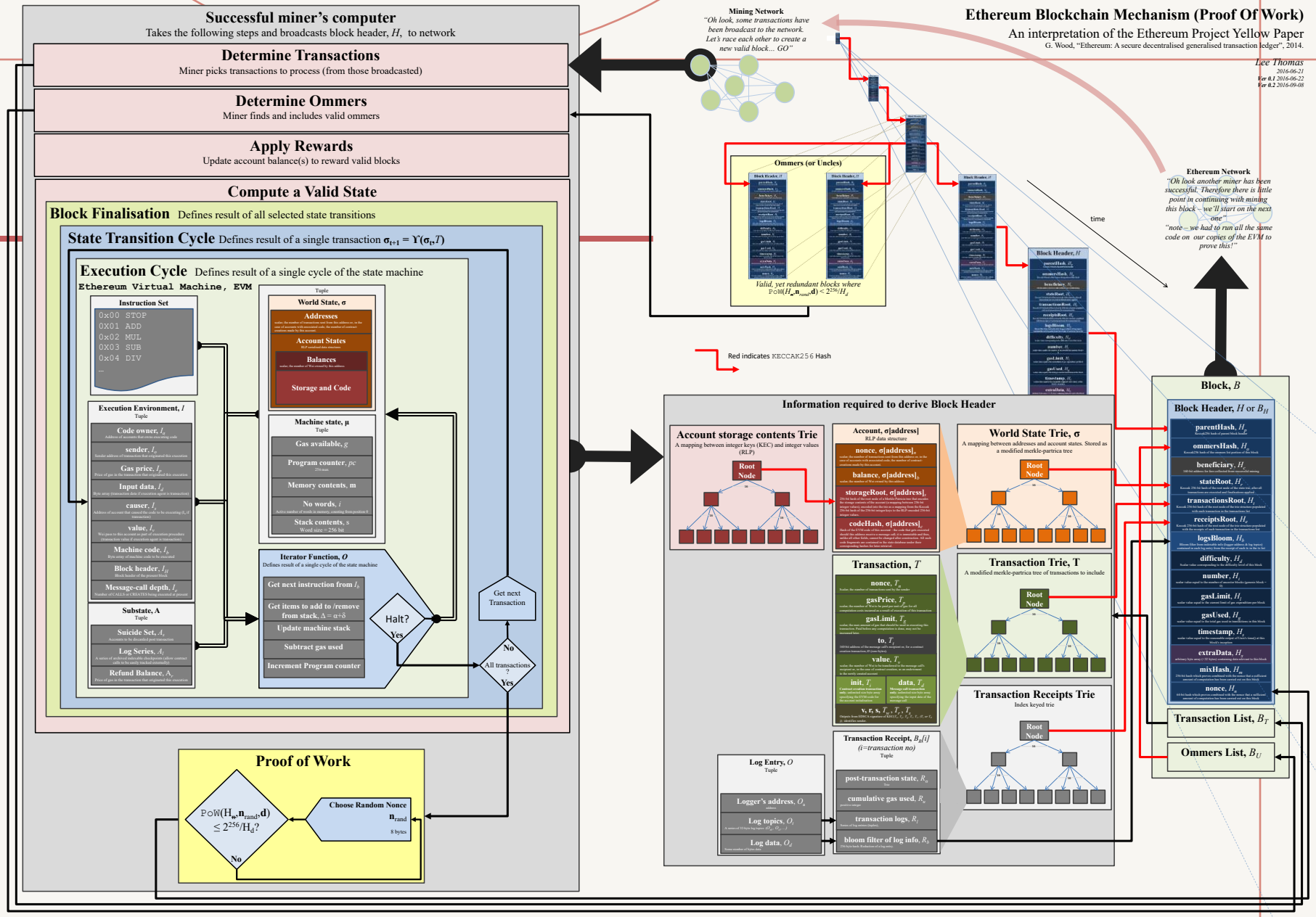
https://i.stack.imgur.com/eOwjD.png

Ethereum blockchain



https://github.com/4c656554/BlockchainIllustrations/blob/master/Ethereum/EthBlockchain5.svg

Ethereum blockchain



https://github.com/4c656554/BlockchainIllustrations/blob/master/Ethereum/EthBlockchain5.svg

Bitcoin consensus

- The PoW algorithm utilised in Bitcoin is called a Compute-bound consensus algorithm
- A Compute-bound PoW, also known as CPU-bound PoW, employs a CPU-intensive function
  - that carries out the required computational task by leveraging the capabilities of the processing units (e.g., CPU/GPU)
  - and it does not rely on the main memory of the system
- These particular characteristics can be massively optimised for faster calculation by using Application-specific Integrated Circuit (ASIC) rigs

Bitcoin consensus

- This is not an ideal scenario as now general people with their general purpose computer cannot participate in the mining process
- The mining process is mostly centralised among a group of mining nodes
- Many crypto-currency enthusiasts suggest that this is not a democratic process and facilitates the “rich getting richer” scenario

Memory-bound consensus algorithm

- To counteract this issue of Bitcoin's CPU-bound PoW algorithm, memory-bound PoWs have been proposed
- A memory-bound PoW requires the algorithm to access the main memory several times
- This ultimately binds the performance of the algorithm within the limit of access latency and/or bandwidth as well as the size of memory
    - The higher the memory the faster the performance

Memory-bound consensus algorithm

- This restricts ASIC rigs based miners not to have manifold performance advantage over CPU/GPU-based mining rigs
- The reason is even though thousands of ASICs could be combined they would have a performance threshold based on the size/bw/latency of the memory
    - Remember that you can’t install unlimited memory within a PC
- This approach also limits the profit margin of the miners who have a mammoth ASIC-based mining rig
- Another motivation of this approach is to de-monopolise the mining concentrations around some central mining nodes

Ethereum consensus algorithm

- Ethash (DAGGER-HASHIMOTO)/DAGGER is the consensus algorithm designed for Ethereum
- Ethash is a memory-bound PoW algorithm with the goal to be ASIC-resistant for a long period of time
- Dagger is one of the earliest proposed memory-bound PoW algorithms which utilises a Directed Acyclic Graph (DAG)
    - a directed acyclic graph (DAG) is a directed graph with no directed cycles
- However it was found be vulnerable
- Ethereum combined Dagger and Hashimoto algorithms to be more secure

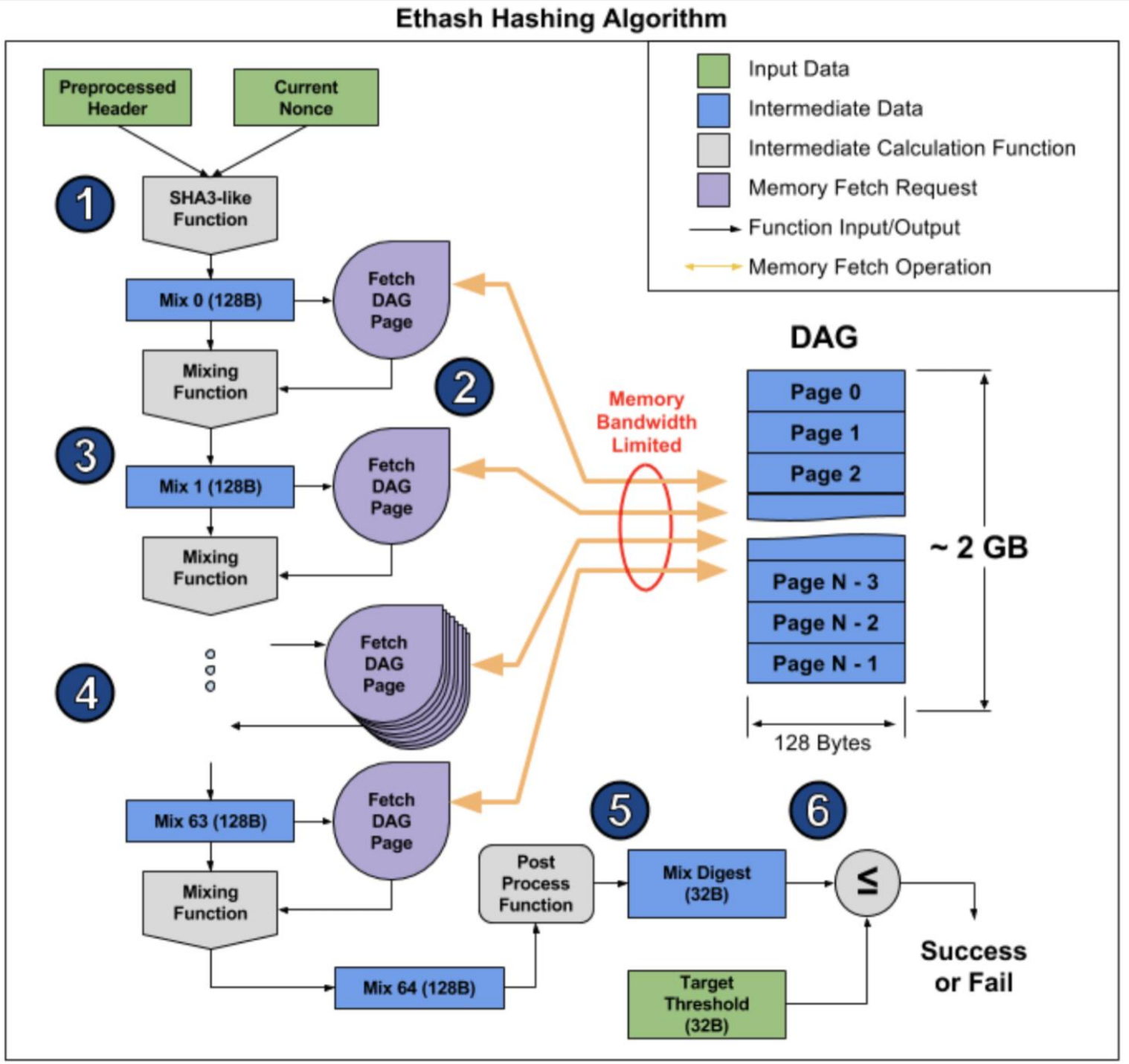
Ethash algorithm

- Ethash depends on a large pseudo-random dataset (the DAG), which is recomputed during each epoch
- Each epoch is determined by the time it takes to generate 30,000 blocks which is approximately five days
- During the DAG generation process, a seed is generated at first, which relies on the length of the chain
- The seed is then used to compute a 16 MB pseudo-random cache
- Then, each item of the DAG is generated by utilising a certain number of items from the pseudo-random cache

Ethash algorithm

- Then, the latest block header and the current candidate nonce are hashed using Keccak (SHA-3) hash function
- The resultant hash is mixed several times with data from the DAG, the mixHash data field
- The final hashed digest is compared to the difficulty target and accepted or discarded accordingly

Ethash algorithm



From Lecture Slides of Blockchains & Cryptocurrencies Course at Johns Hopkins University

Ethereum PoS algorithm

- Ethereum moved to a PoS consensus algorithm in 2022
- Like any PoS algorithm there are a number of validators
- To be a validator, one has to deposit 32 eth to an escrow contract
- On depositing their ETH, the user joins an activation queue that limits the rate of new validators joining the network
- Once activated, validators receive new blocks from peers on the Ethereum network
- The transactions delivered in the block are re-executed, and the block is verified
- The validator then sends a vote (called an attestation) in favour of that block across the network

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

Ethereum PoS algorithm

- Under Ethash algorithm, the timing of blocks is determined by the mining difficulty
- In proof-of-stake, the block generation time is fixed
- Time in proof-of-stake Ethereum is divided into slots (12 seconds) and epochs (32 slots)
- 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

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

Ethereum PoS algorithm

- Also in every slot, a committee of validators is randomly chosen, whose votes are used to determine the validity of the block being proposed
- If a validator is chosen to attest the next block, they are rewarded in ETH as a percentage of their stake
- Conversely, validators who do not perform their duties--if they are offline, for example--receive penalties, or slashes, in the form of small amounts of ETH subtracted from their stakes

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

Question?