CSE446: Blockchain & Cryptocurrencies

Lecture – 17: Ethereum - 5

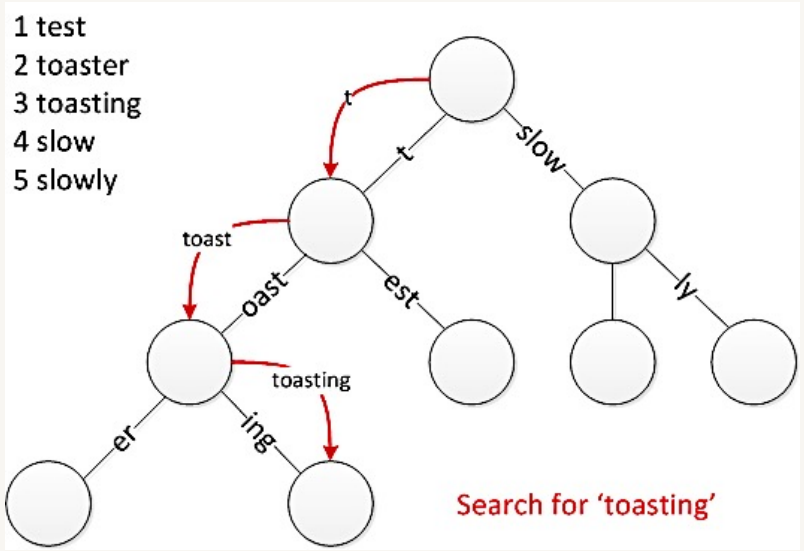
Inspiring Excellence

Agenda

- Ethereum Tries

MPT

- PT 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
- 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)

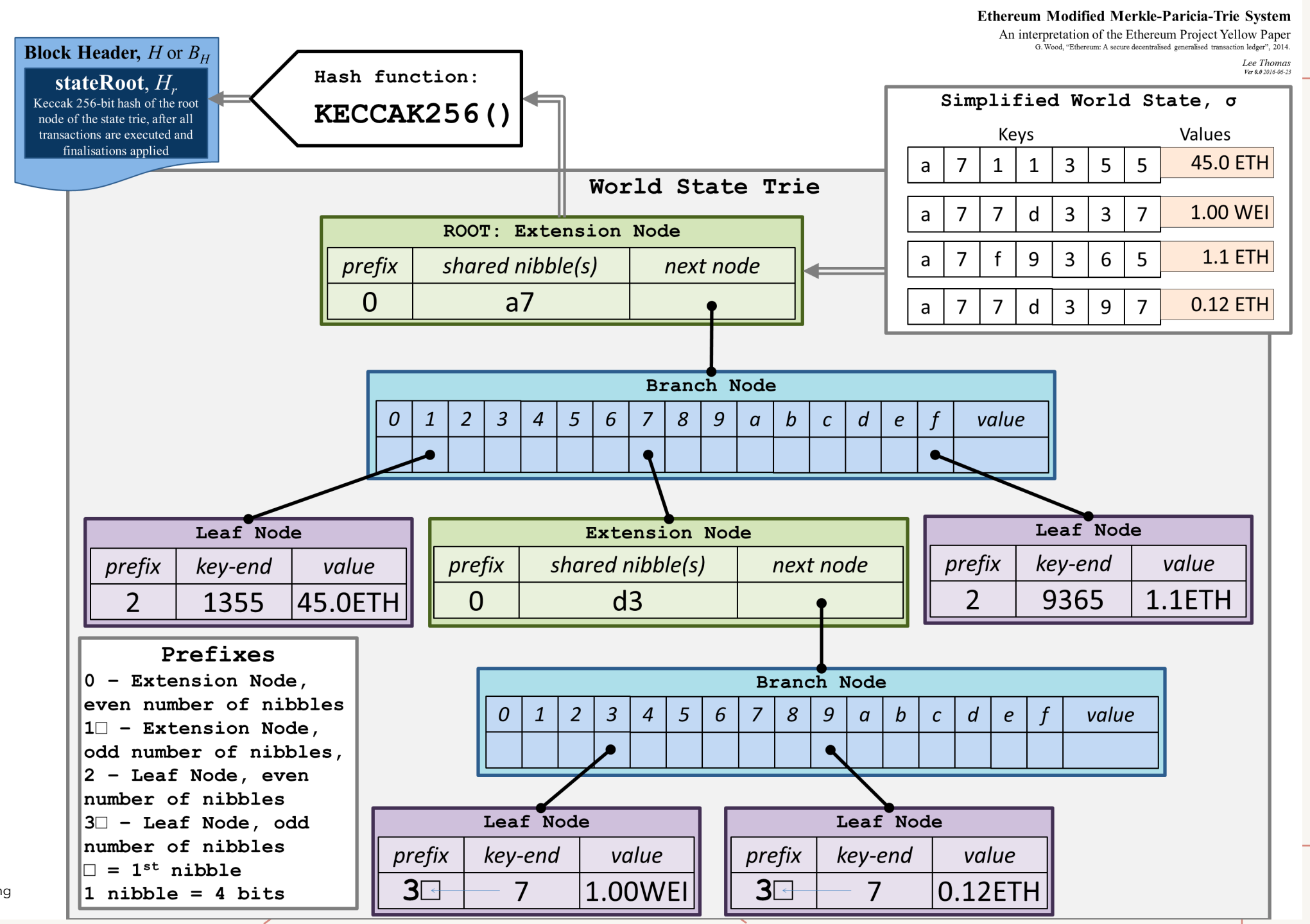
- An MPT is a data structure for storing key value pairs in a cryptographically authenticated manner
- In the MPT every node has a hash value
  - Each node's hash is decided by the sha3 hash value of its contents
- This hash is also used as the key that refers to the node
- The content of the node is 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

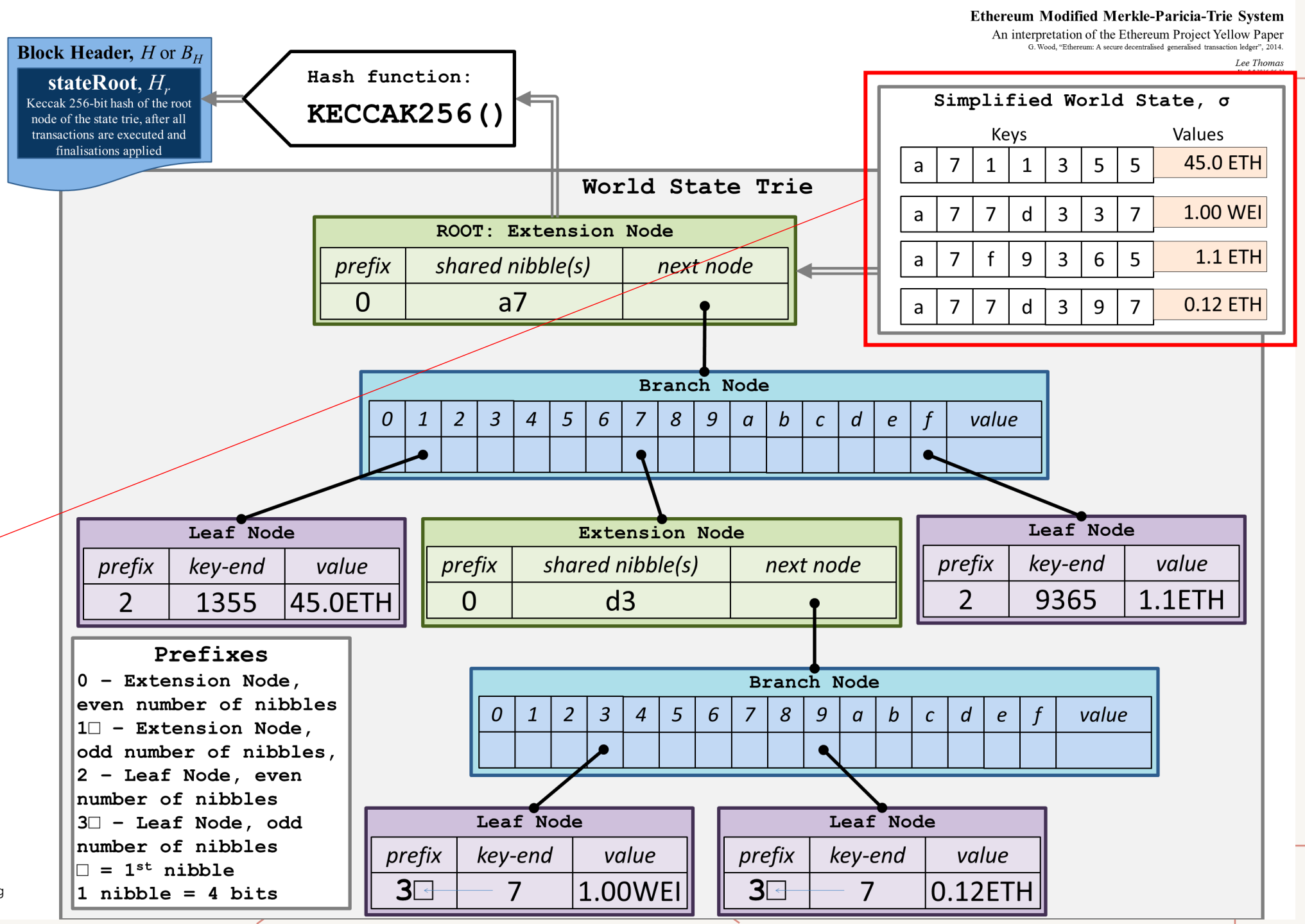
MPT



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

MPT

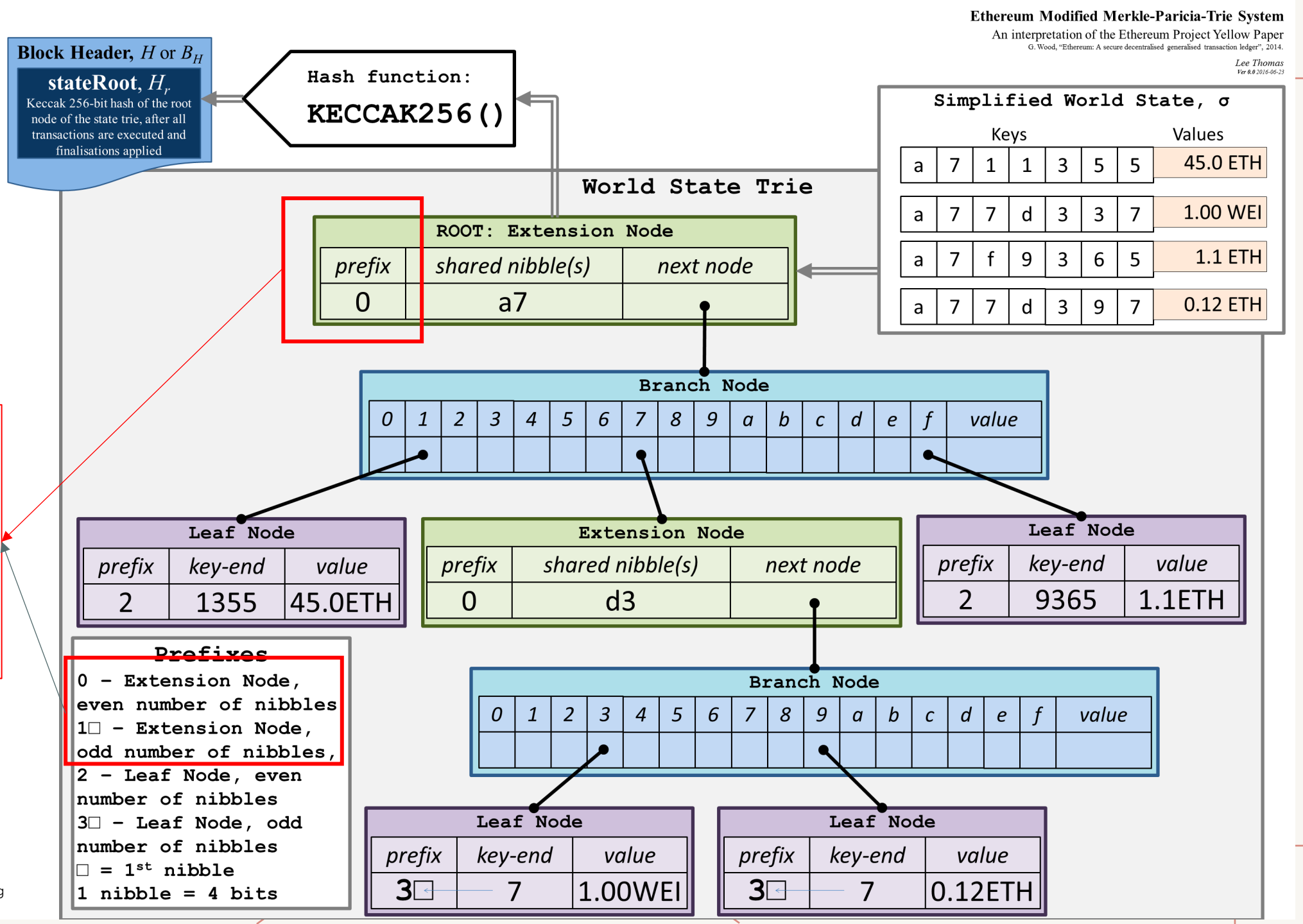
- This represents a simplified world state of accounts
- Instead of storing each account in the blockchain, MPT is used



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

MPT

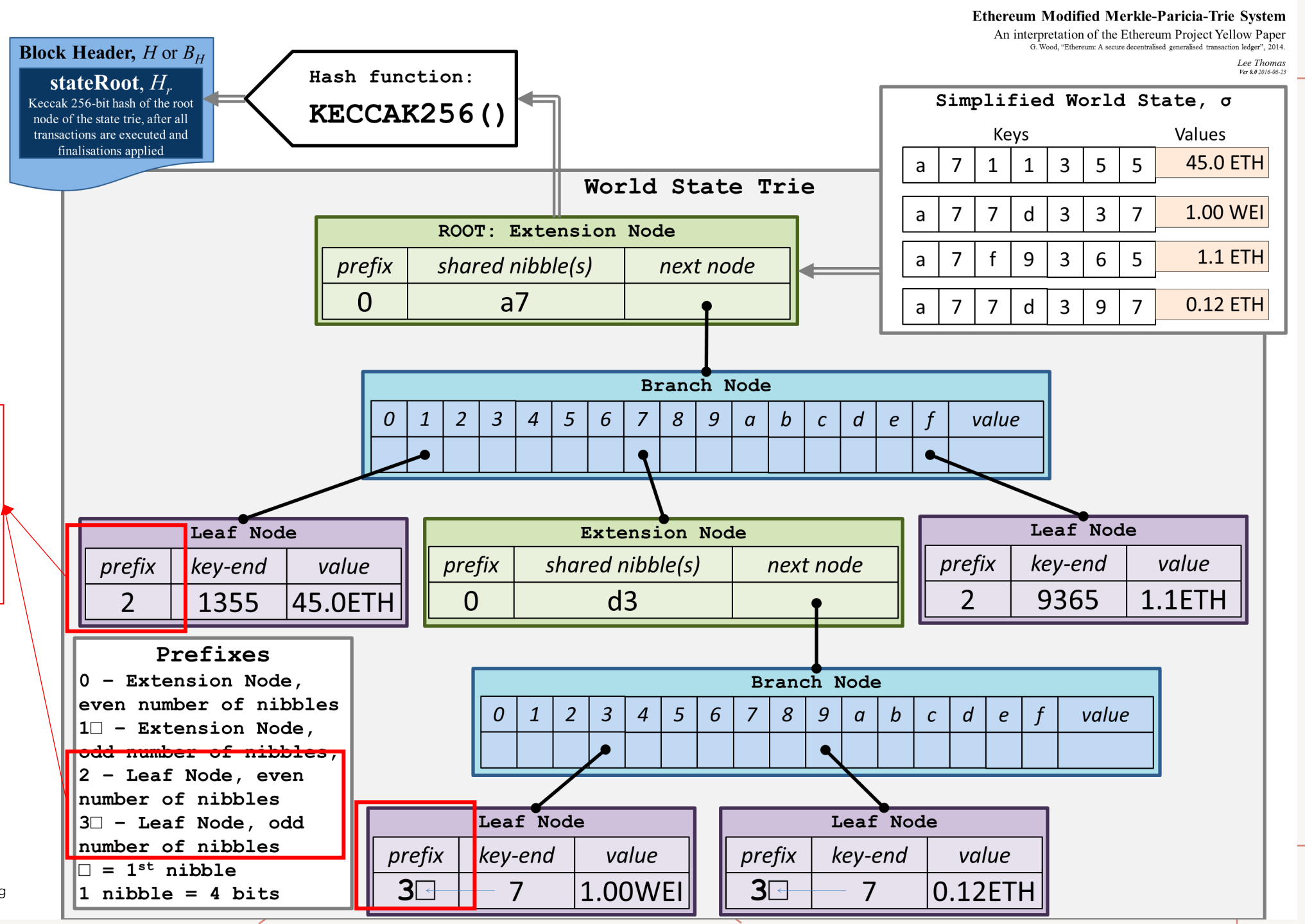
- Prefix determines the type of node
- 0/1 indicates an extension node
- If there are even number of nibbles then 0, otherwise 1



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

MPT

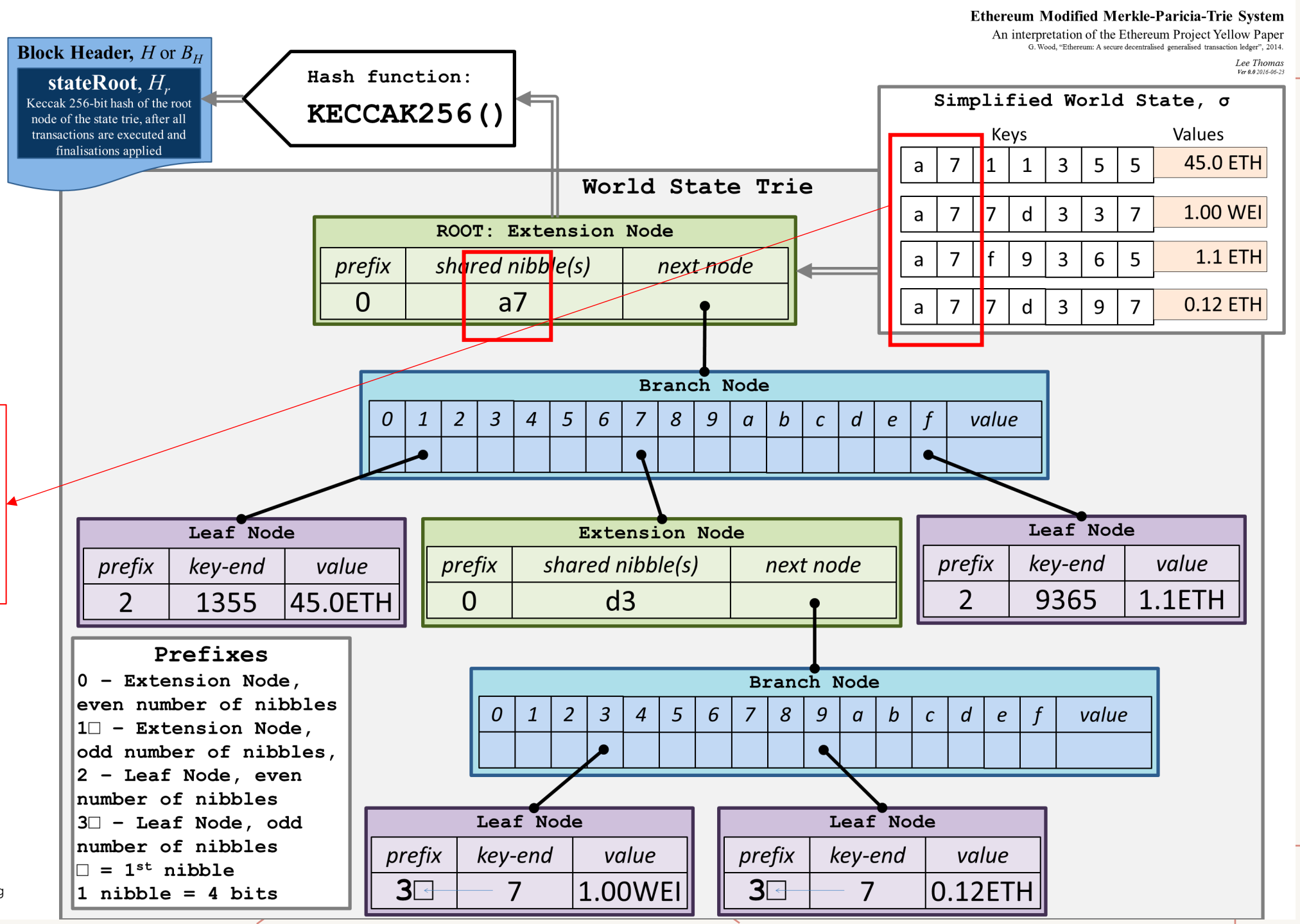
- 2/3 indicates a leaf node
- If there are even number of nibbles then 2, otherwise 3



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

MPT

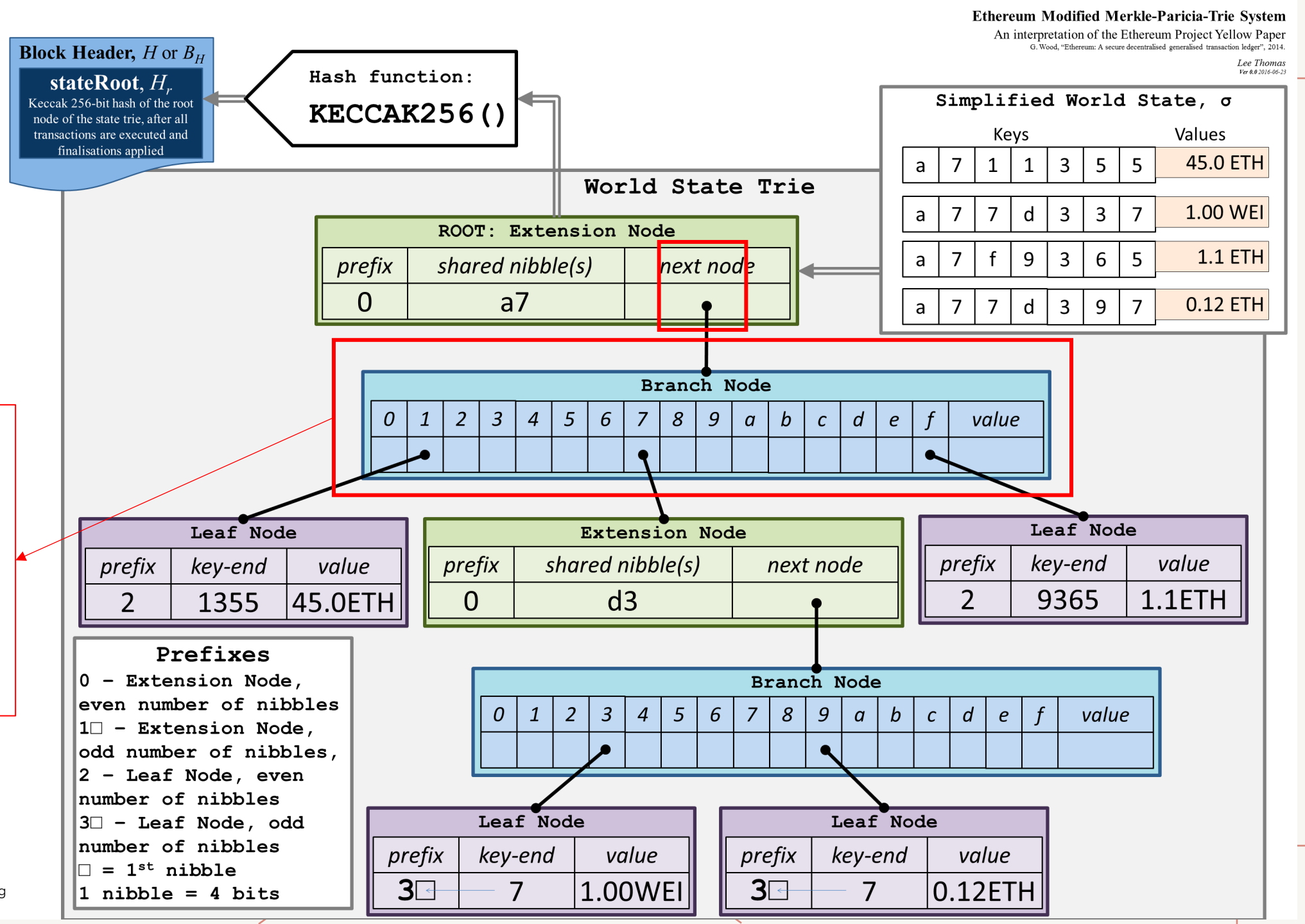
- All these accounts share a common prefix: a7
- That is why a7 remains in the root



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

MPT

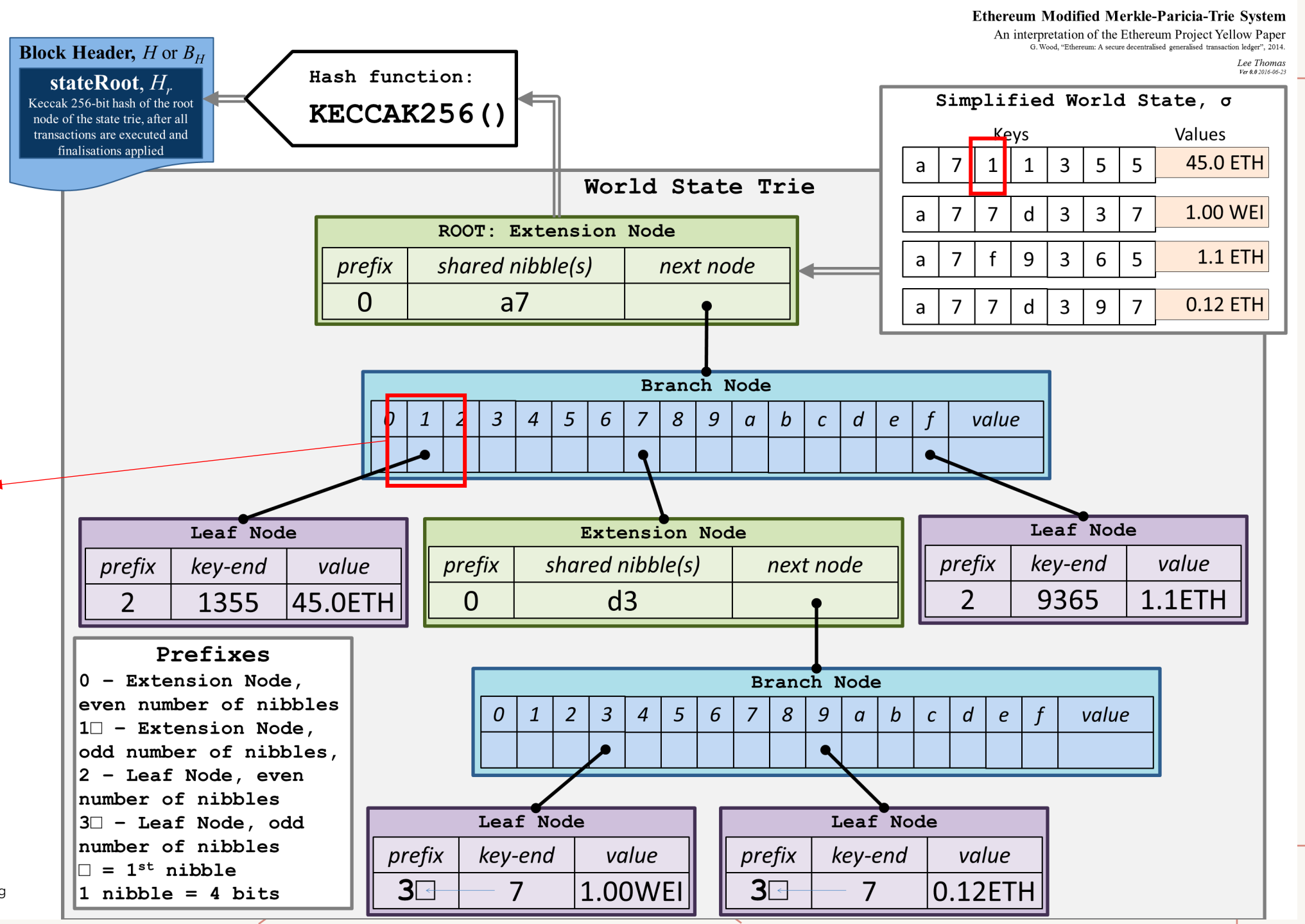
- Next node field in the extension node points to a branch node
- A branch node has 16 hexadecimal characters and a value field



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

MPT

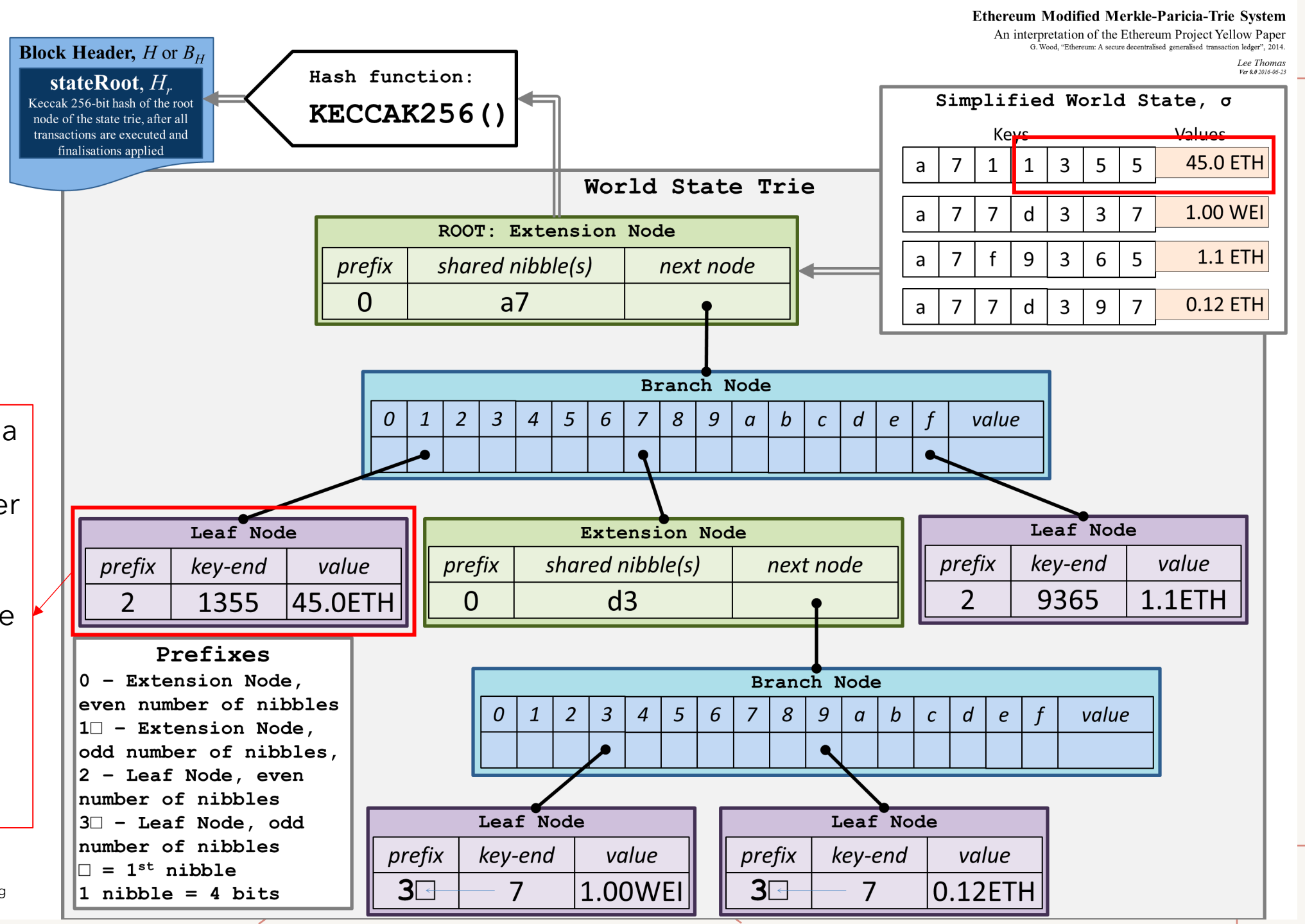
- Each field except the value field represents a key character



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

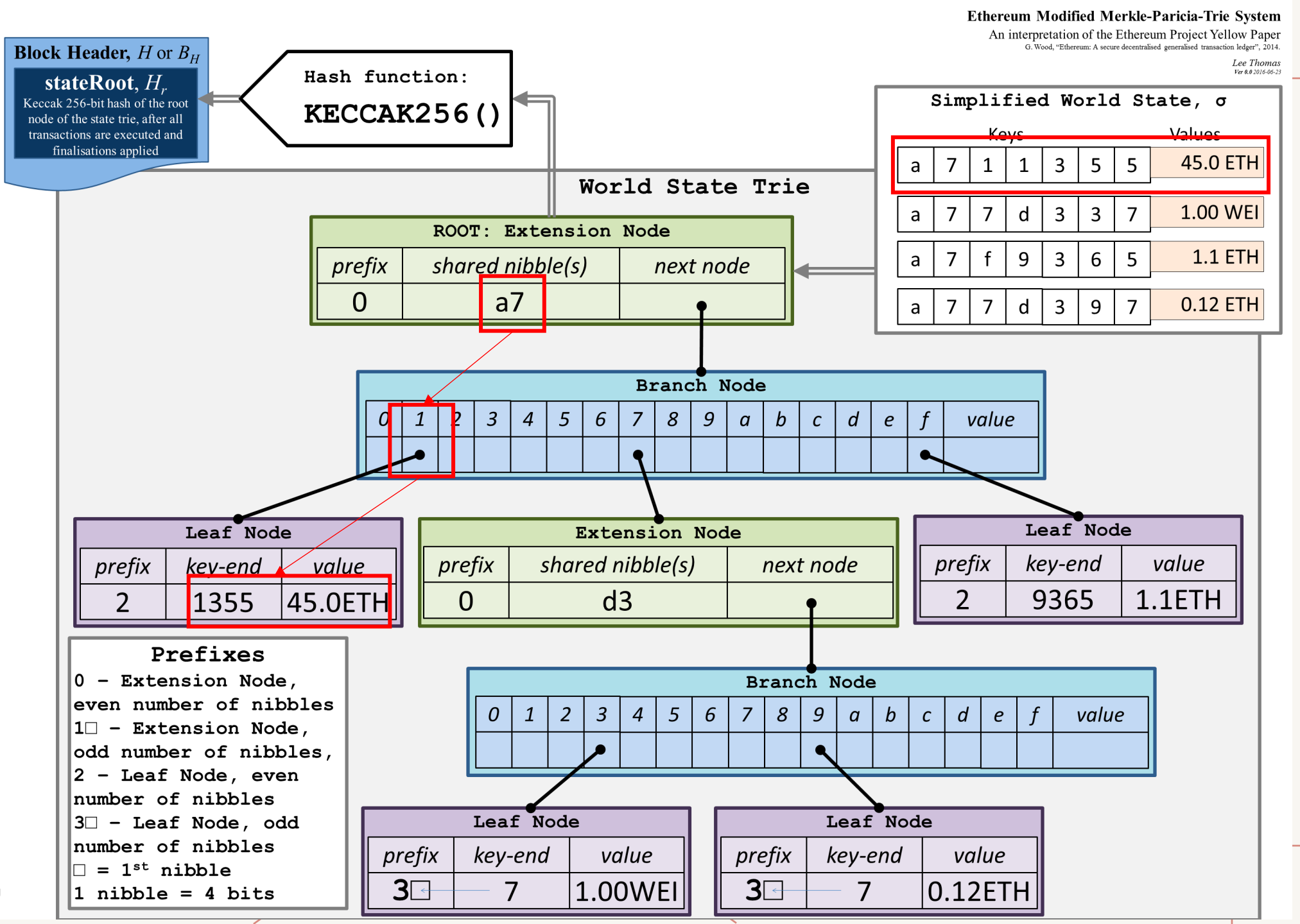
MPT

- 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



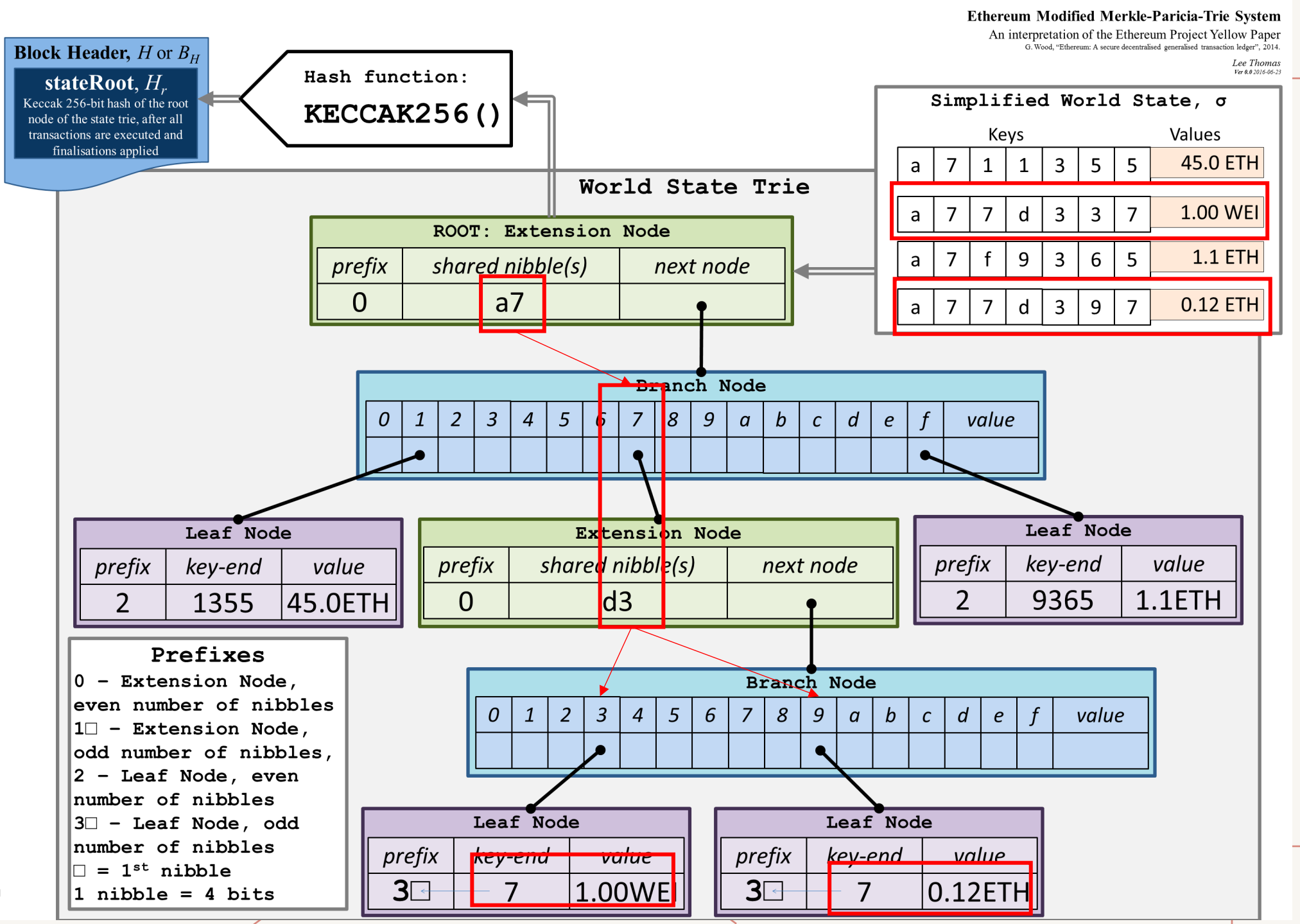
https://i.stack.imgur.com/YZGxe.png

MPT



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

MPT

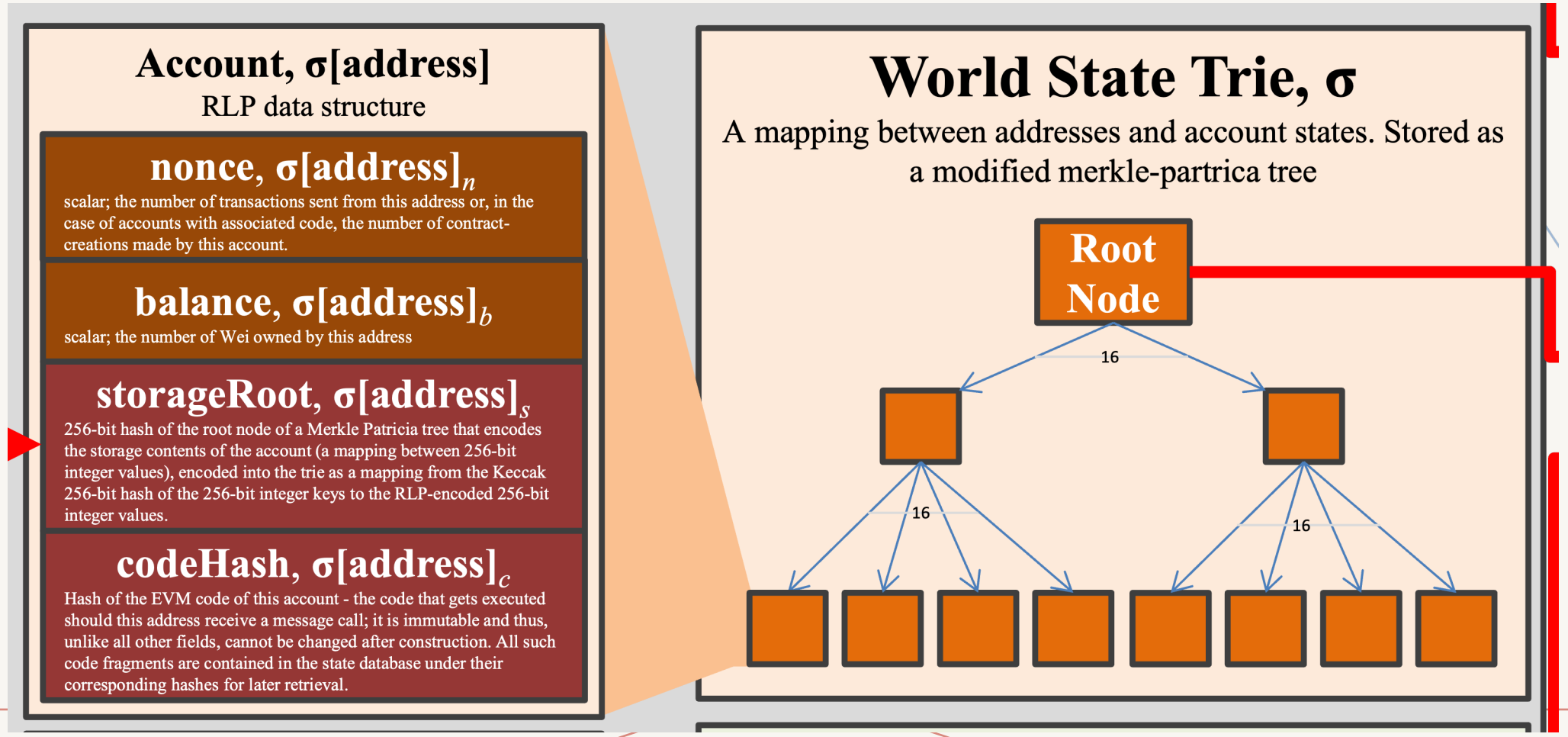


https://i.stack.imgur.com/YZGxe.png

Ethereum tries

- There are four different tries used in Ethereum
  - State Trie
  - Transaction Trie
  - Transaction Receipt Trie
  - Account storage Trie

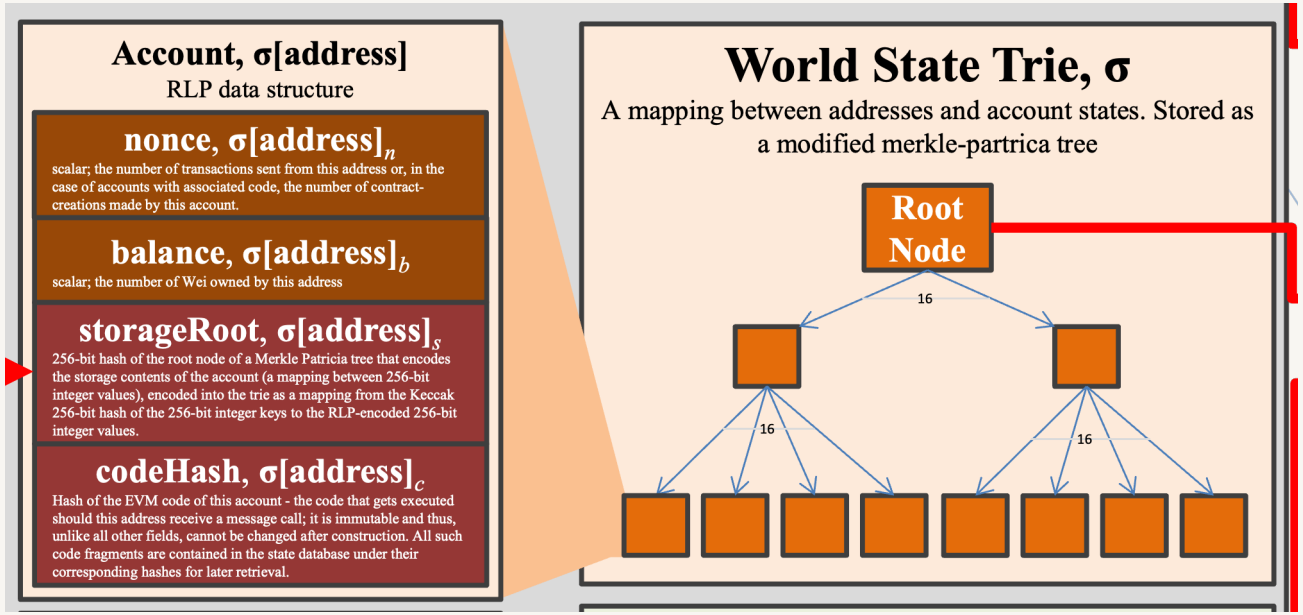
State trie



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

State trie

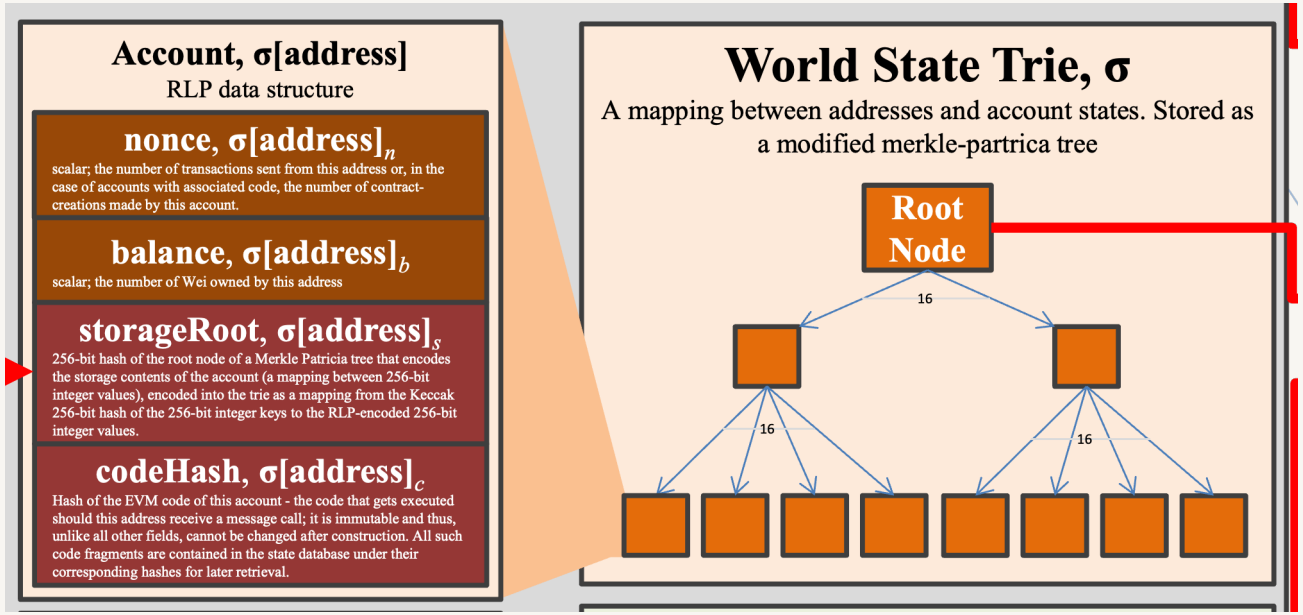
- The state trie contains a key and value pair for every account which exists on the Ethereum network
- The “key” is the address of an Ethereum account
- The “value” created by encoding the following account details of an Ethereum account
  - nonce, balance, storageRoot, codeHash



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

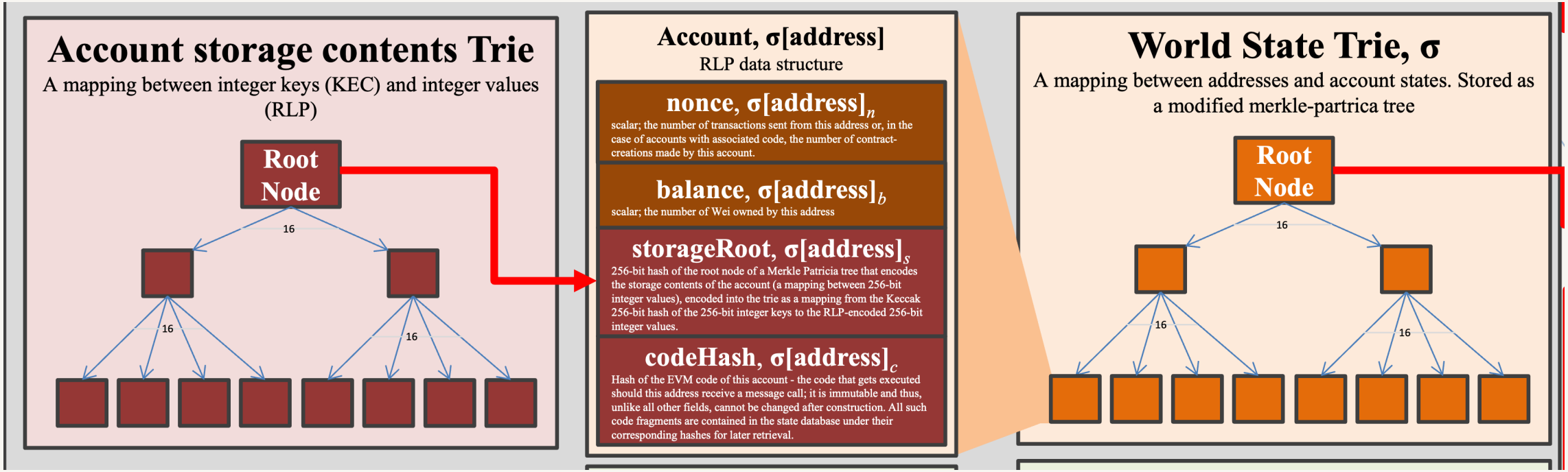
State trie

- Nonce – the number of transactions sent from this account
- Balance – balance of this account
- codeHash – hash of the EVM code for this account, applicable for contract account
- storageRoot:
  - Root of a storage trie where all of the contract data lives for this particular account



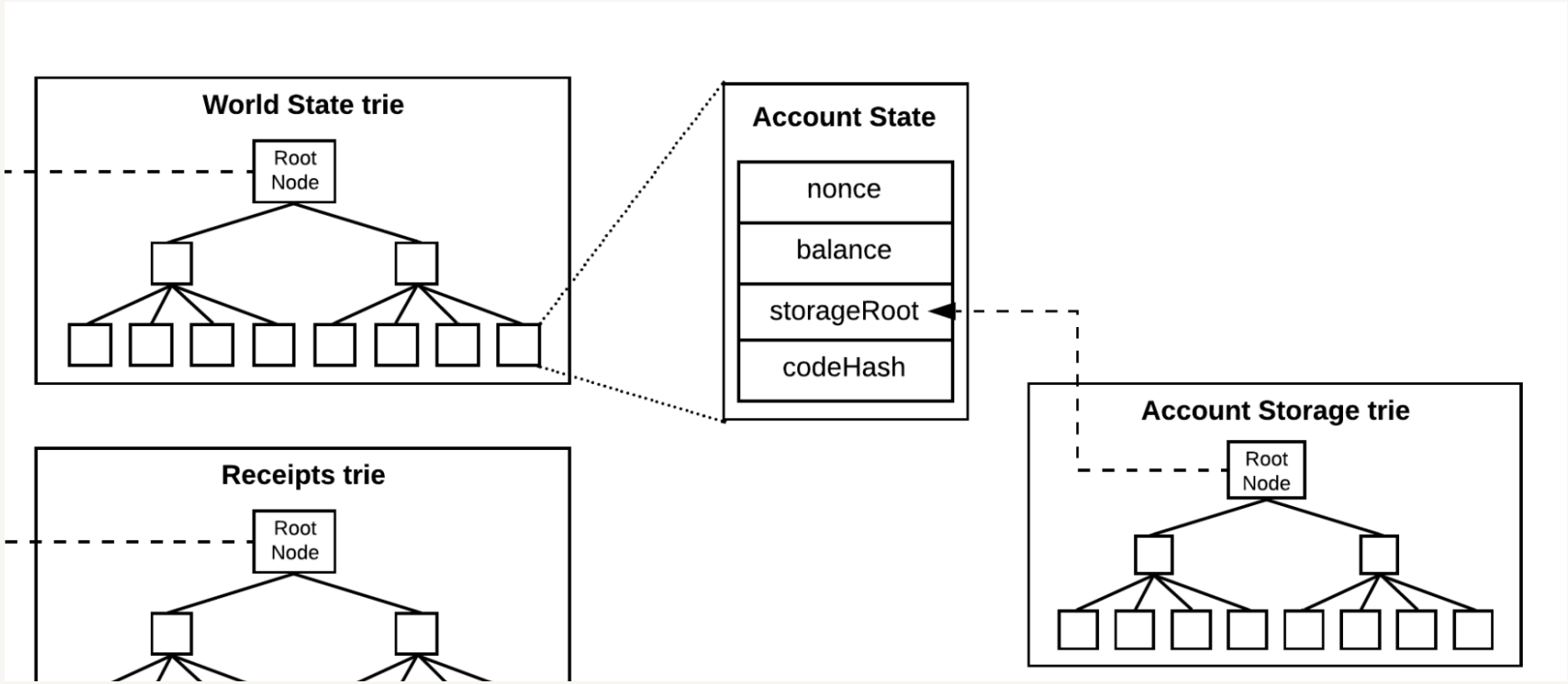
https://github.com/4c656554/BlockchainIllustrations/blob/master/Ethereum/EthBlockchain5.svg

Storage trie



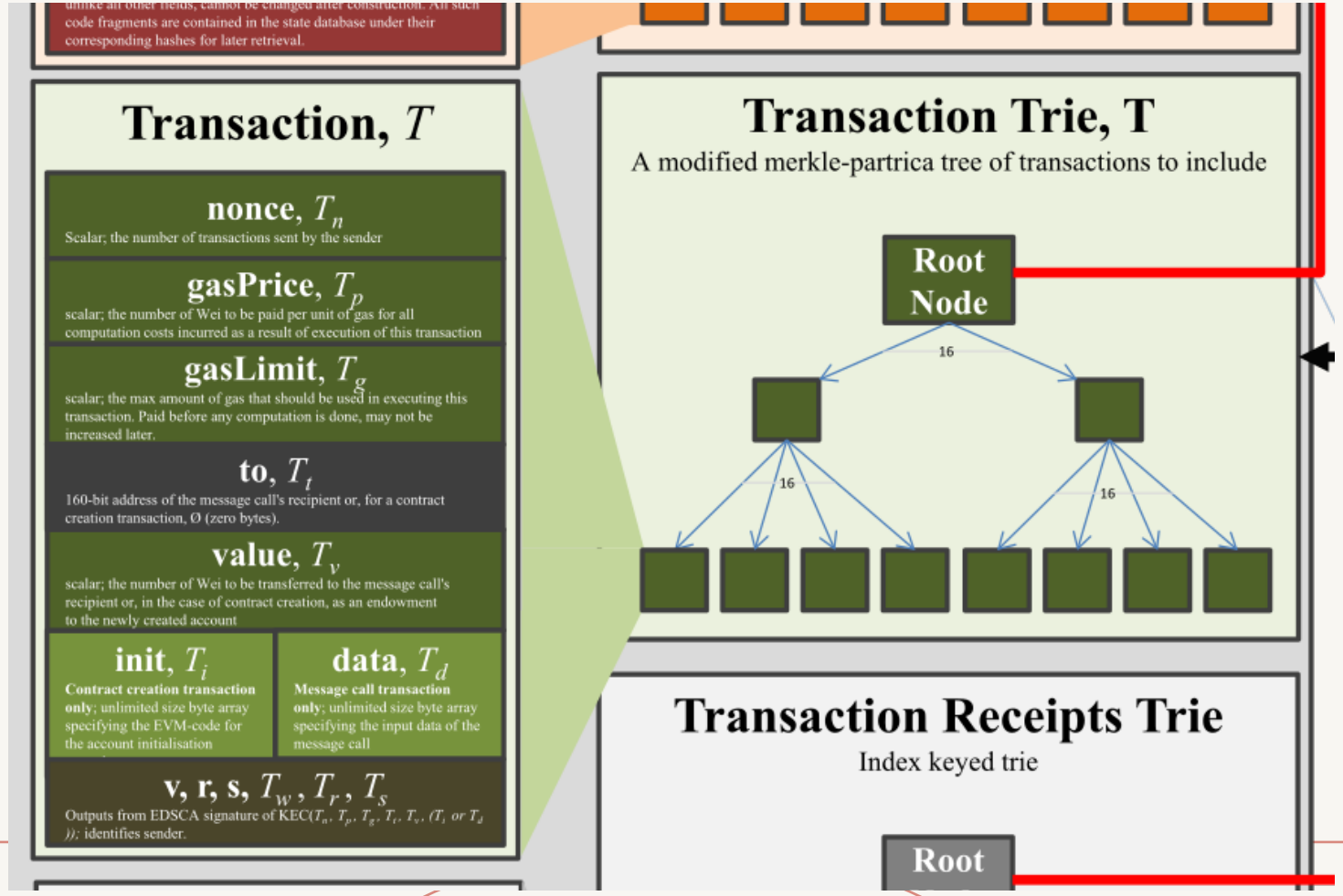
https://github.com/4c656554/BlockchainIllustrations/blob/master/Ethereum/EthBlockchain5.svg

Storage trie



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

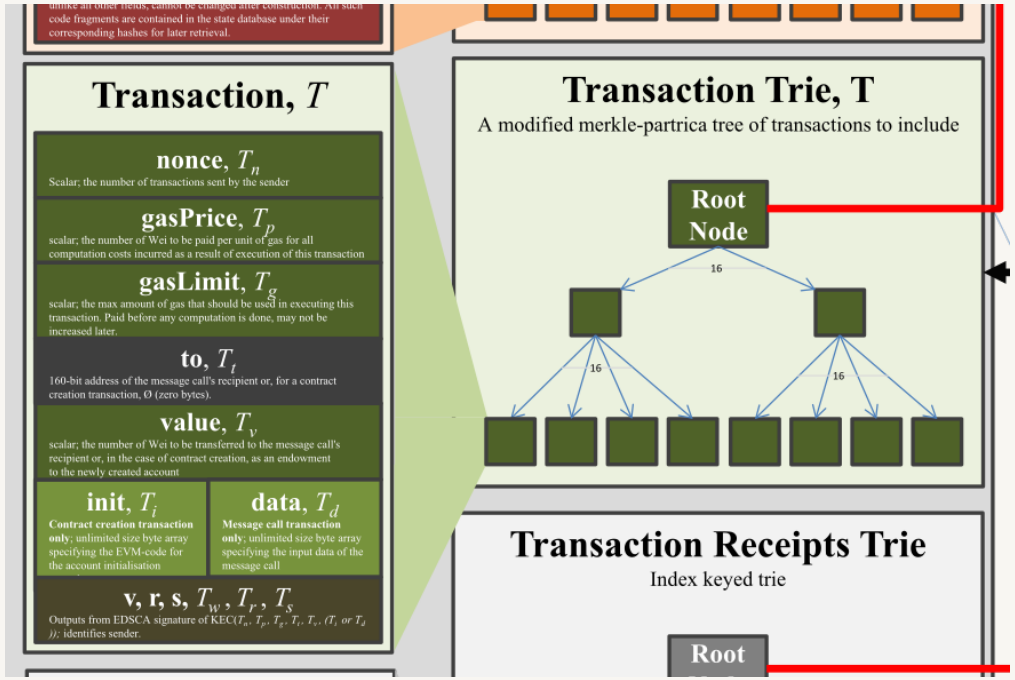
Transaction trie



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

Transaction trie

- Account nonce -> the number of transactions sent by the account
- Gas price & Gas limit -> as explained before
- Recipient -> the receiving account
- Transfer value -> the value to be transferred
- Transaction signature values -> digital signature to prove ownership
- Account initialization (if transaction is of contract creation type), or transaction data (if transaction is a contract (message) call)

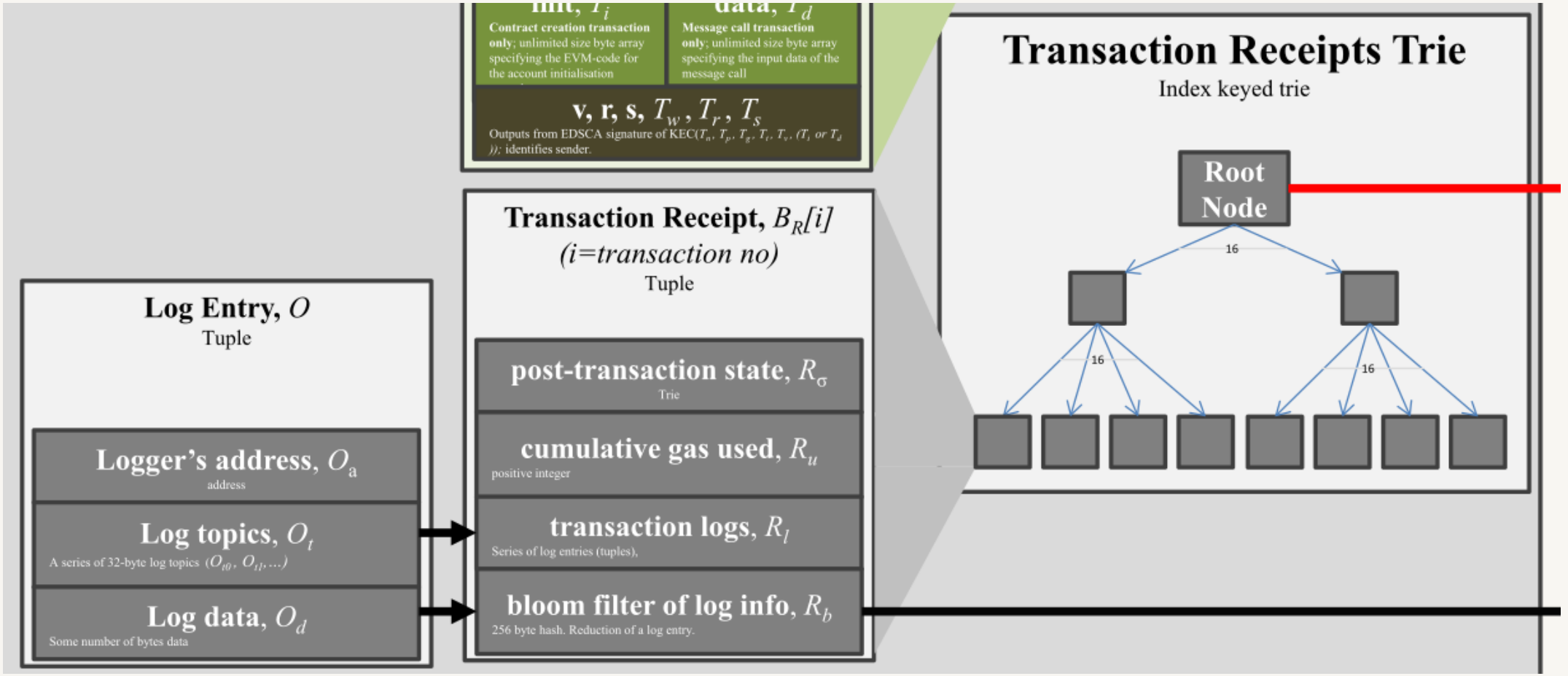


https://github.com/4c656554/BlockchainIllustrations/blob/master/Ethereum/EthBlockchain5.svg

Ethereum events and logs

- EVM supports generating of events and logs in the smart-contract and propagating them to outside the chain
- These events and logs are stored in transaction receipts (in Receipt Trie) within each block
- These are very useful
  - to track certain execution has been completed within a function of a smart-contract or
  - to trigger callback functions in the front end of the DApp

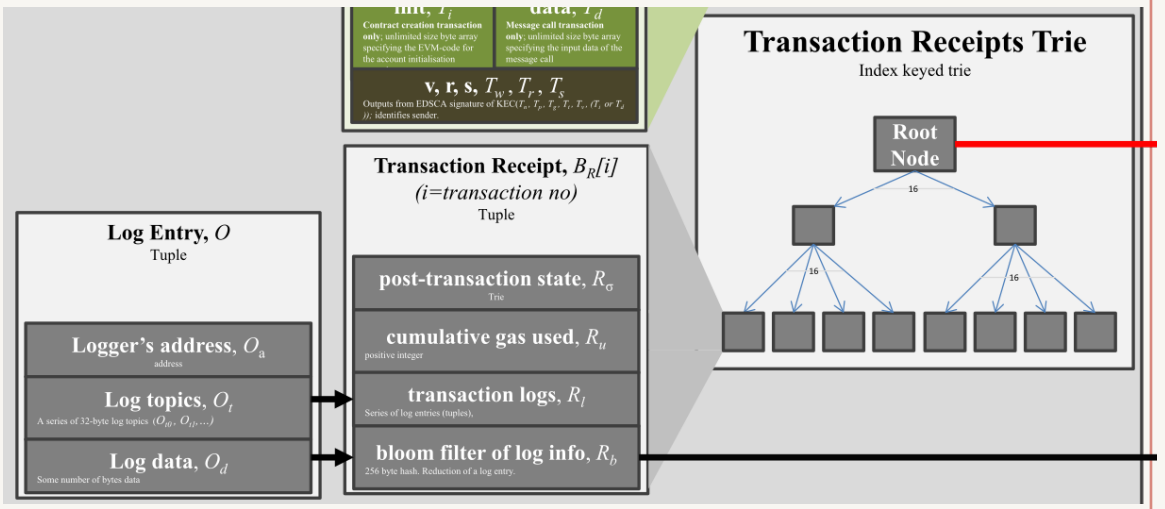
Transaction receipts trie



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

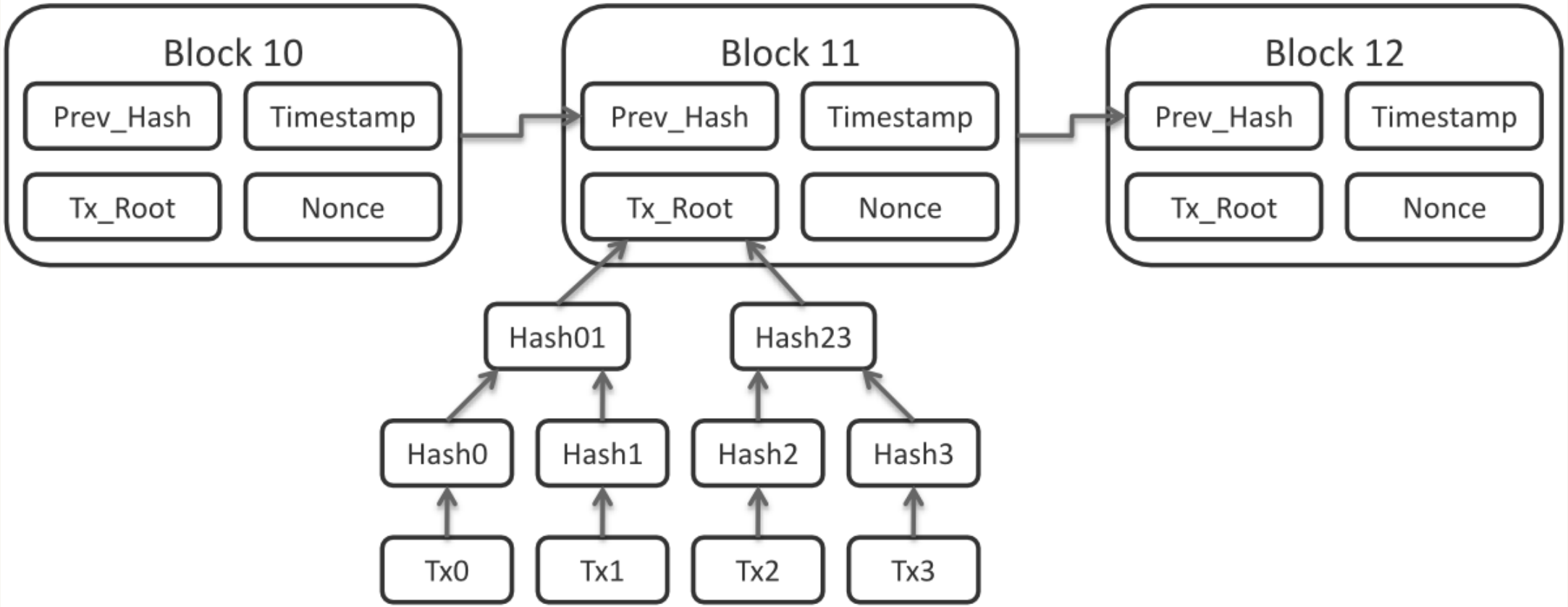
Transaction receipts trie

- The receipt root lives within the block header and its purpose is to record the outcome of a transaction
- A receipt trie is one per block
- Parameters used in composing a transaction Receipt Trie:
  - post-transaction state (1 for a successful transaction and 0 for unsuccessful),
  - the cumulative gas used,
  - the set of logs created through execution of the transaction (for event propagation), and
  - the Bloom filter composed from information in those logs



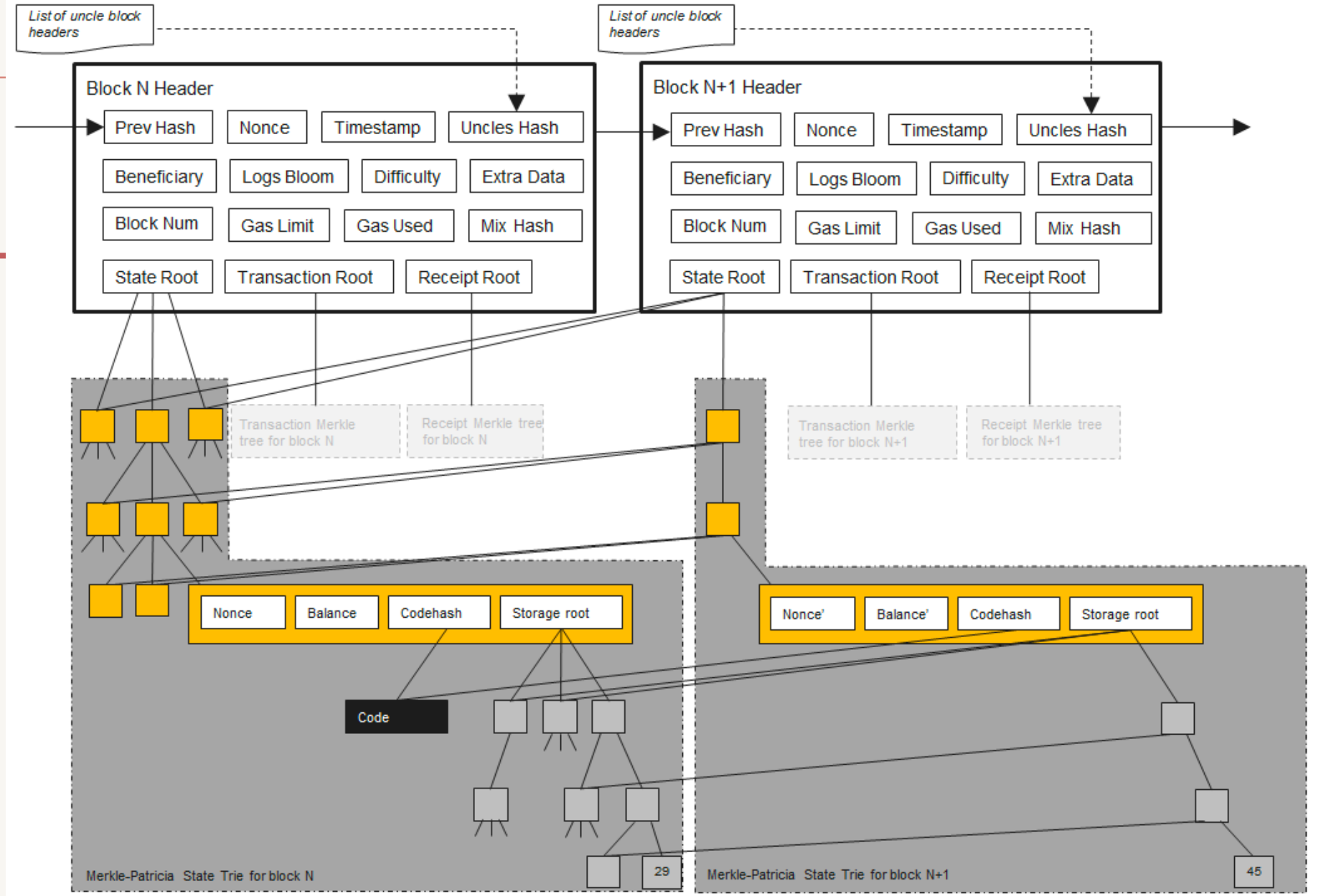
https://github.com/4c656554/BlockchainIllustrations/blob/master/Ethereum/EthBlockchain5.svg

Bitcoin blockchain



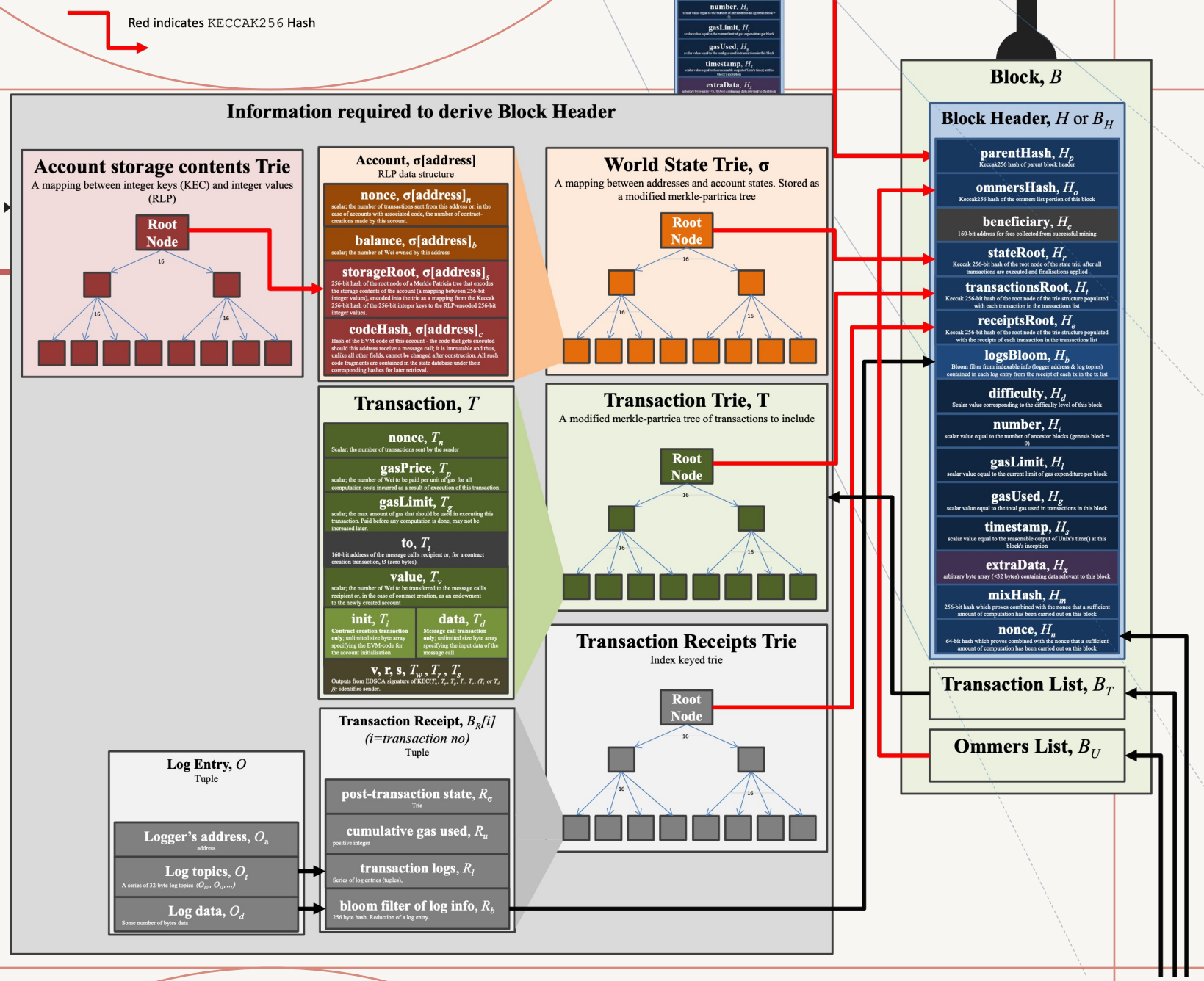
https://i.stack.imgur.com/HrKX0.png

Ethereum blockchain



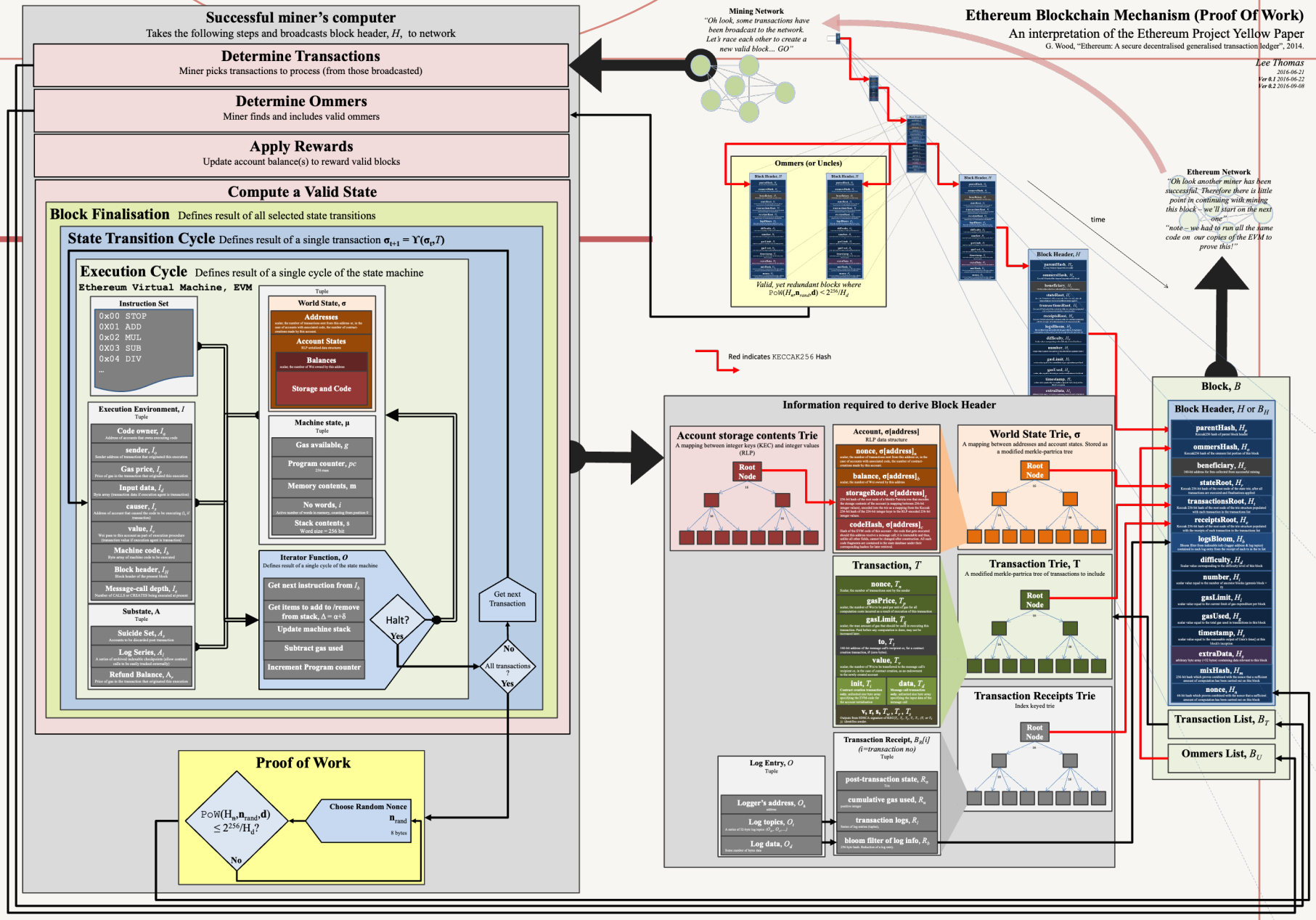
https://i.stack.imgur.com/eOwjD.png

Ethereum blockchain



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

Ethereum blockchain



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

Question?