INFINITY
BLOCKCHAIN LABS

Ethereum

for noobs

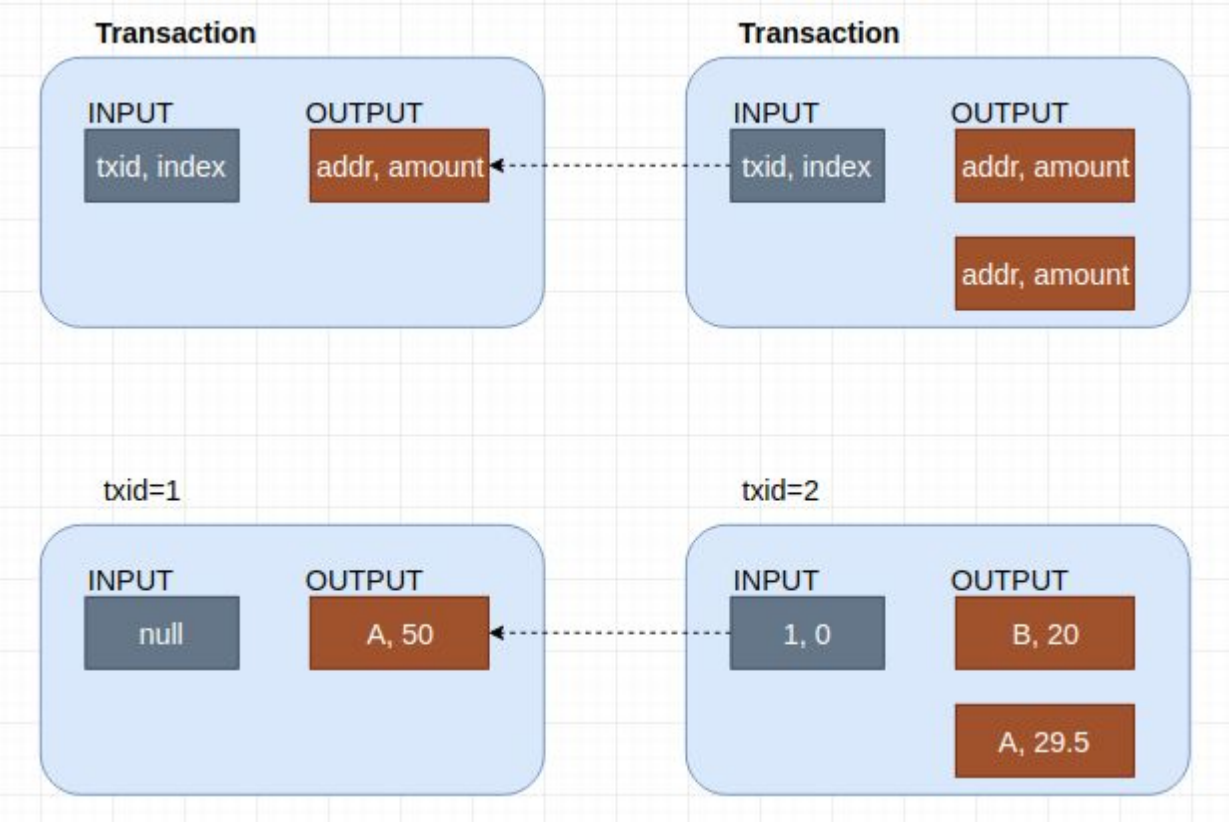
Ethereum state

Hội anh em blockchain "thiện lành"

Contents

- Transaction-based vs Account-based
- Merkle Patricia tree
- Ethereum state
- Demo code

Transaction-based



Spent state

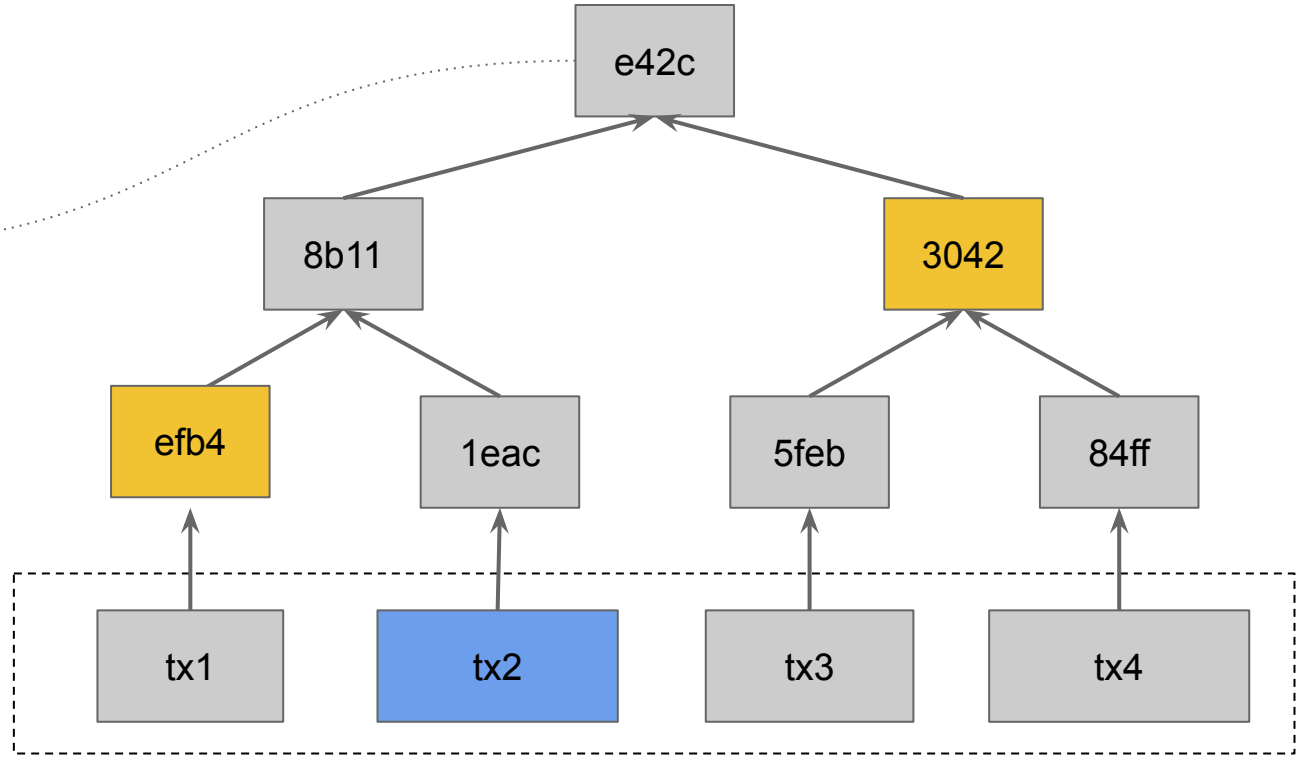
1,0: true 2,0: false 2,1: false

SPV, Simplified Payment Verification: verifying if particular transactions are included in a block without downloading the entire block

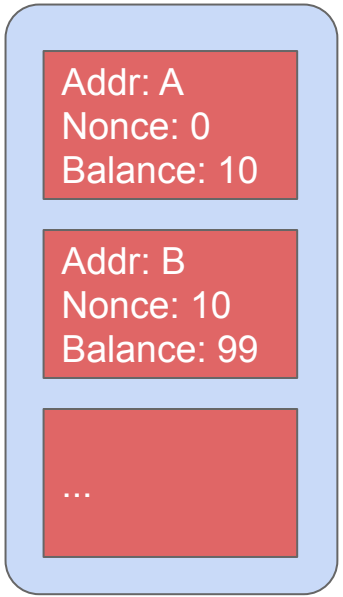
Merkle tree

Block header

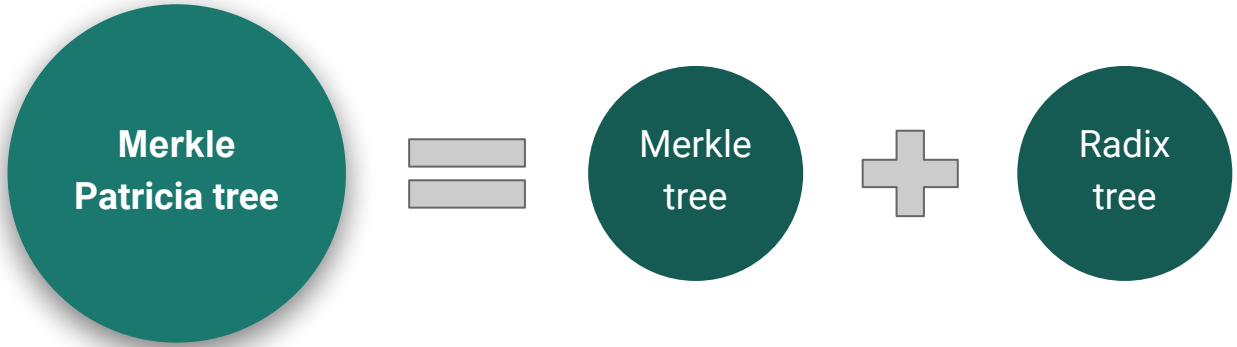
Prev hash
Merkle root
Time
...



Account-based



- Get information of an account quickly
- Smart contract data
- Revert when a fork occurs
- Simplified payment verification

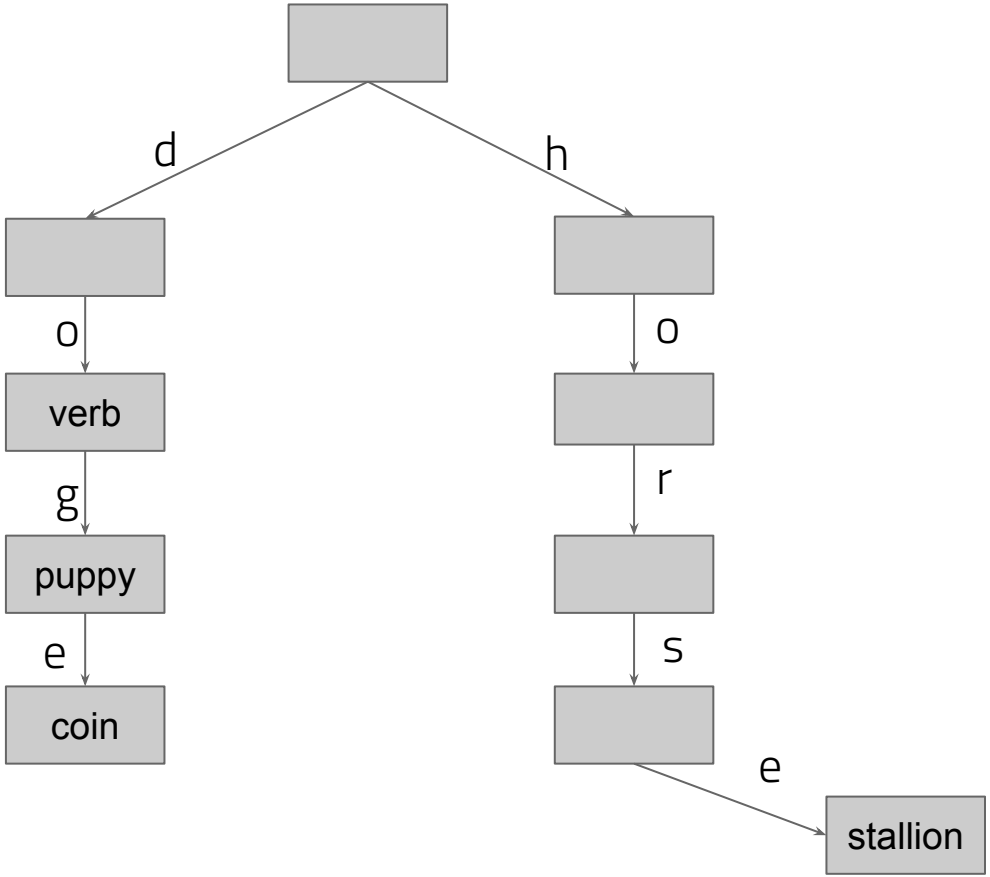


Radix tree

- Key is a string
- Edge stores part of the key
- Node stores data

Data

do: verb
dog: puppy
doge: coin
horse: stallion



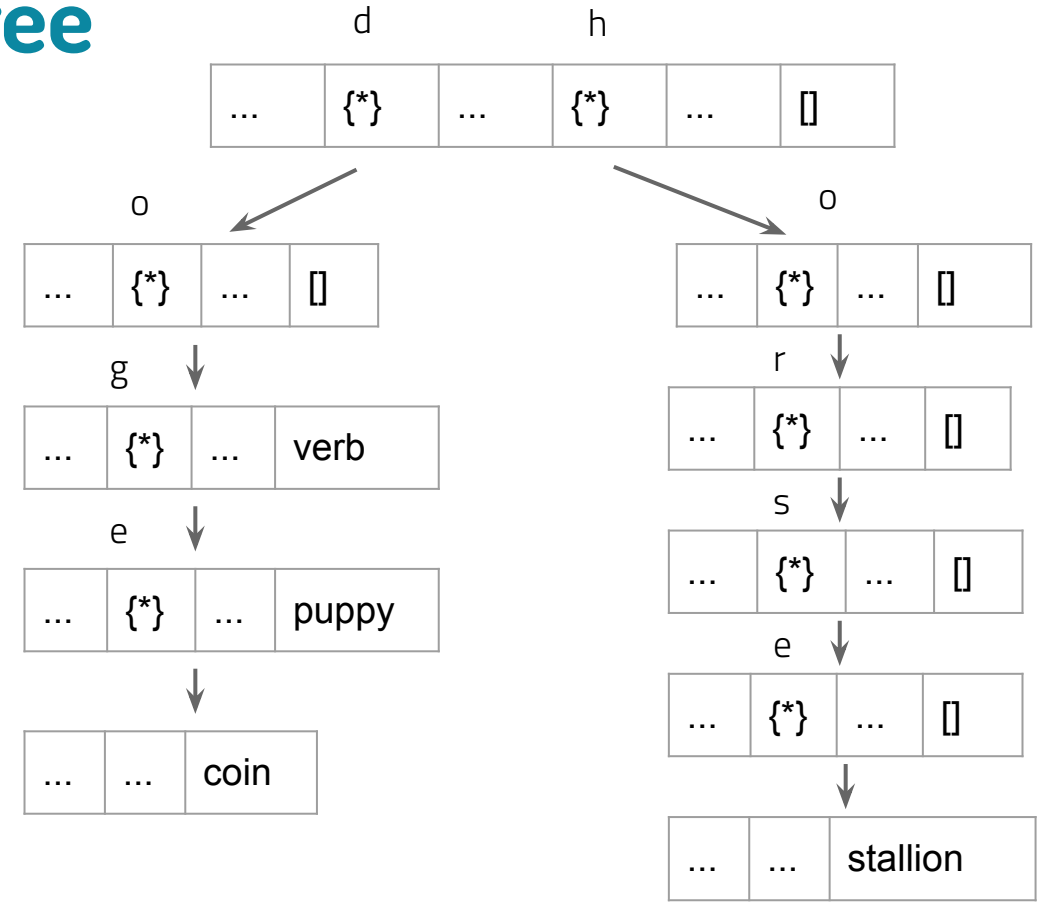
Merkle Patricia Tree

Example of an English word tree

- Node is an array of 27 items.
- The last item is the value.
- The others are the hash pointer to next node.
- Index of the array is a part of the key(key is a combination of the English alphabet character).

Data

do: verb
dog: puppy
doge: coin
horse: stallion



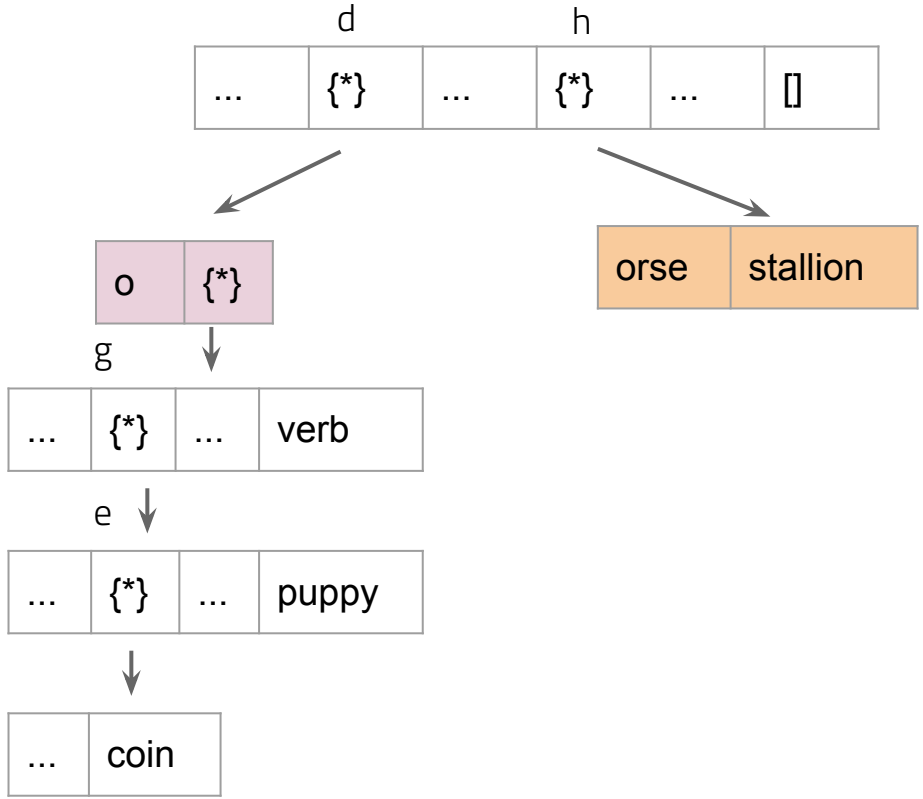
Merkle Patricia Tree

Three kinds of node:

- branch node: 27 items **[v0, ..., v26, vt]**
- extension node: 2 items **[path, key]**
- leaf node: 2 items **[path, value]**

Data

do: verb
dog: puppy
doge: coin
horse: stallion



Merkle Patricia Tree

Convert key to hex string:

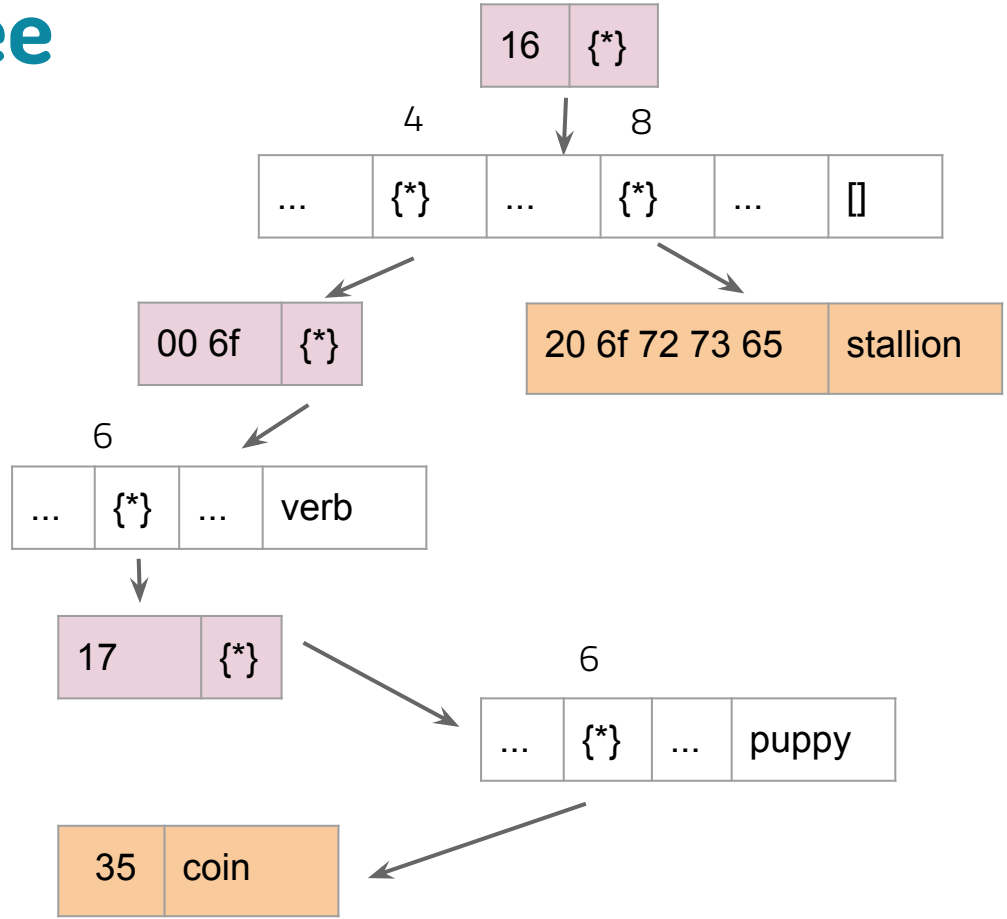
- <64 6f> : 'verb'
- <64 6f 67> : 'puppy'
- <64 6f 67 65> : 'coin'
- <68 6f 72 73 65> : 'stallion'

branch node: 17 items

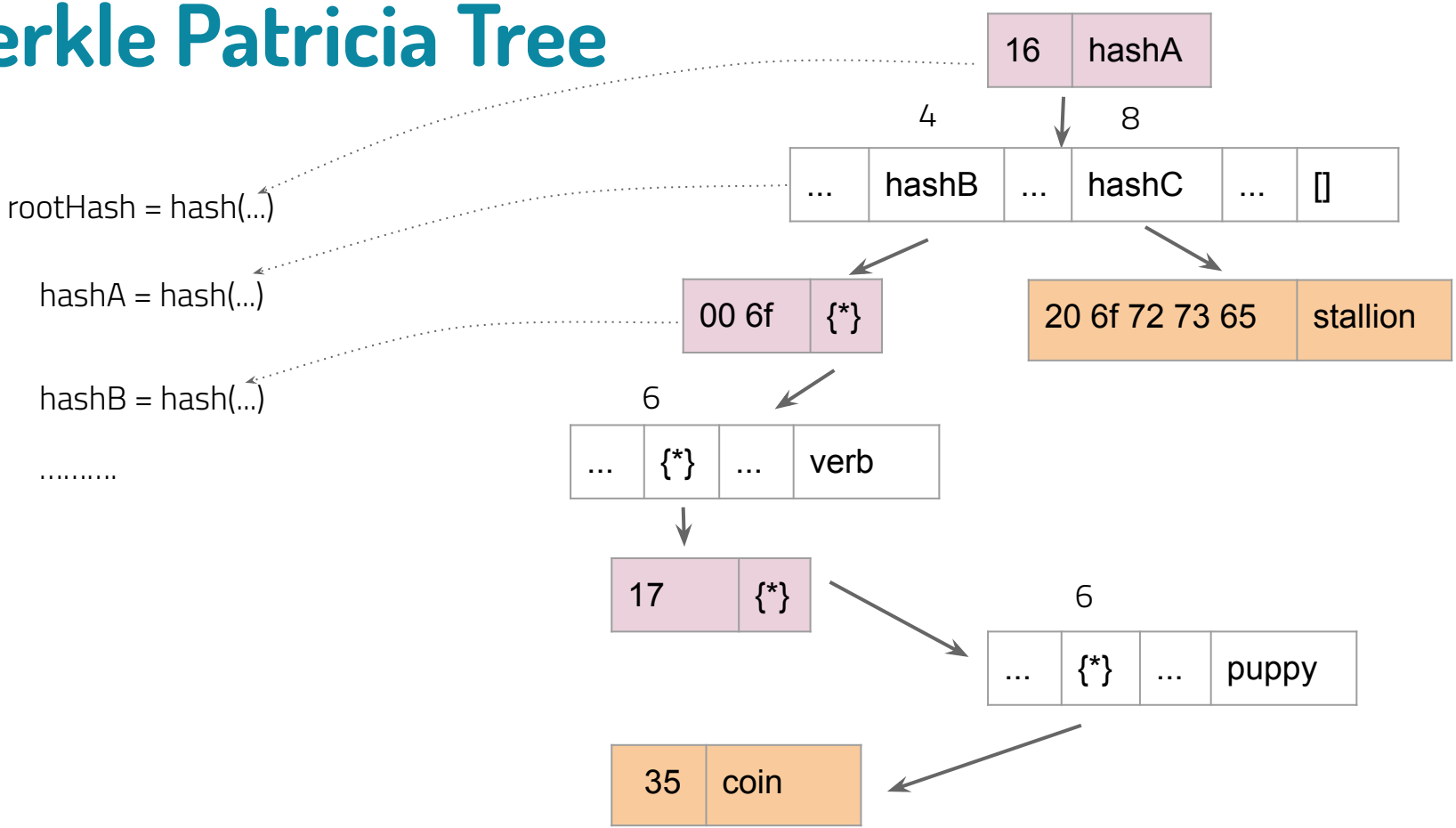
add prefixed flag key path

Data

do: verb
dog: puppy
doge: coin
horse: stallion



Merkle Patricia Tree



Merkle Patricia Tree

Tree is stored in LevelDB

rootHash: [<16>, hashA]

hashA: [<>, <>, <>, <>, hashB, <>, <>, <>, hashC, <>, <>, <>, <>, <>, <>, <>, <>]

hashC: [<20 6f 72 73 65>, 'stallion']

hashB: [<00 6f>, hashD]

hashD: [<>, <>, <>, <>, <>, <>, hashE, <>, <>, <>, <>, <>, <>, <>, <>, <>, 'verb']

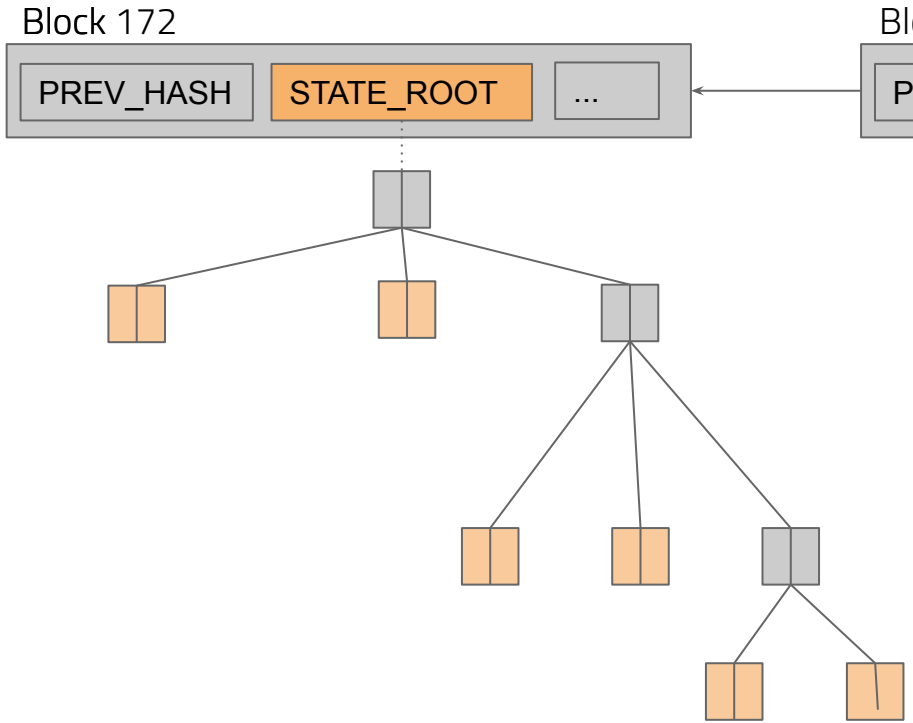
hashE: [<17>, hashF]

hashF: [<>, <>, <>, <>, <>, <>, hashG, <>, <>, <>, <>, <>, <>, <>, <>, <>, 'puppy']

hashG: [<35>, 'coin']

https://github.com/ethereum/wiki/wiki/Patricia-Tree

Ethereum state



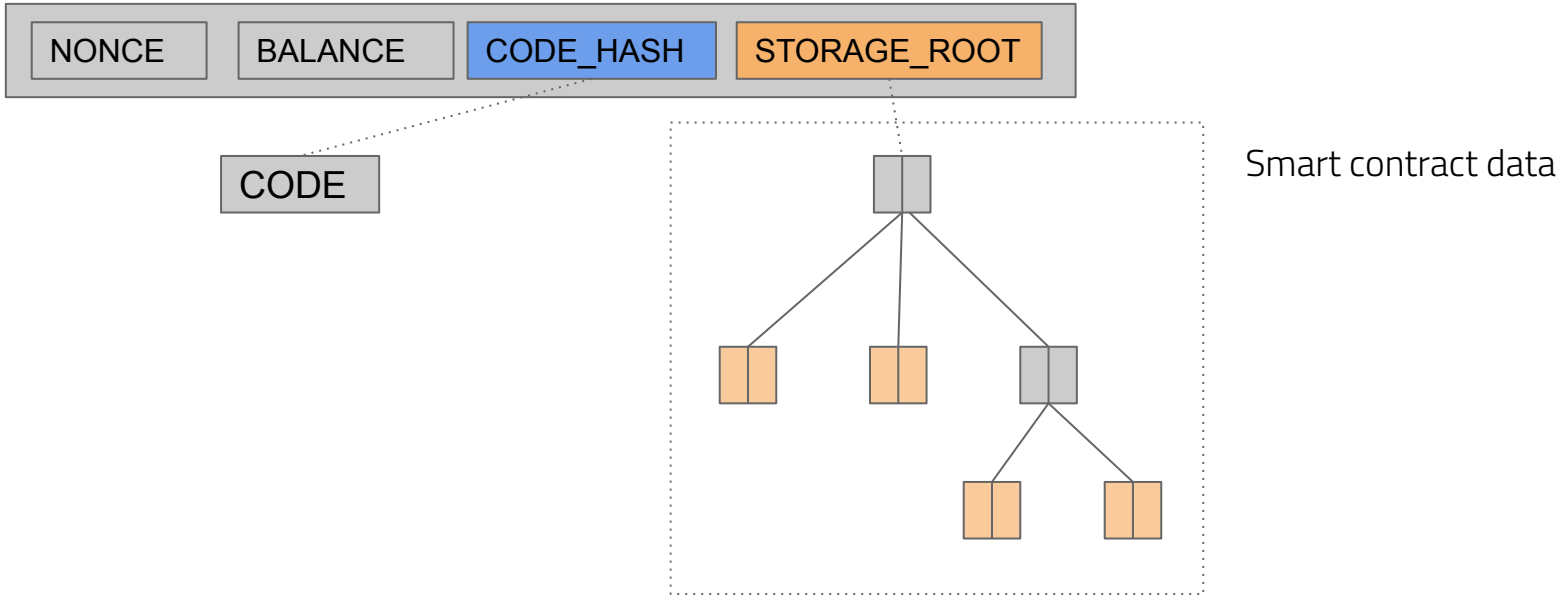
The state is made up of objects called "accounts", with each account having a 20-byte address

Address example:
5a0b54d5dc17e0aadc383d2db43b0a0d3e029c4c

https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial

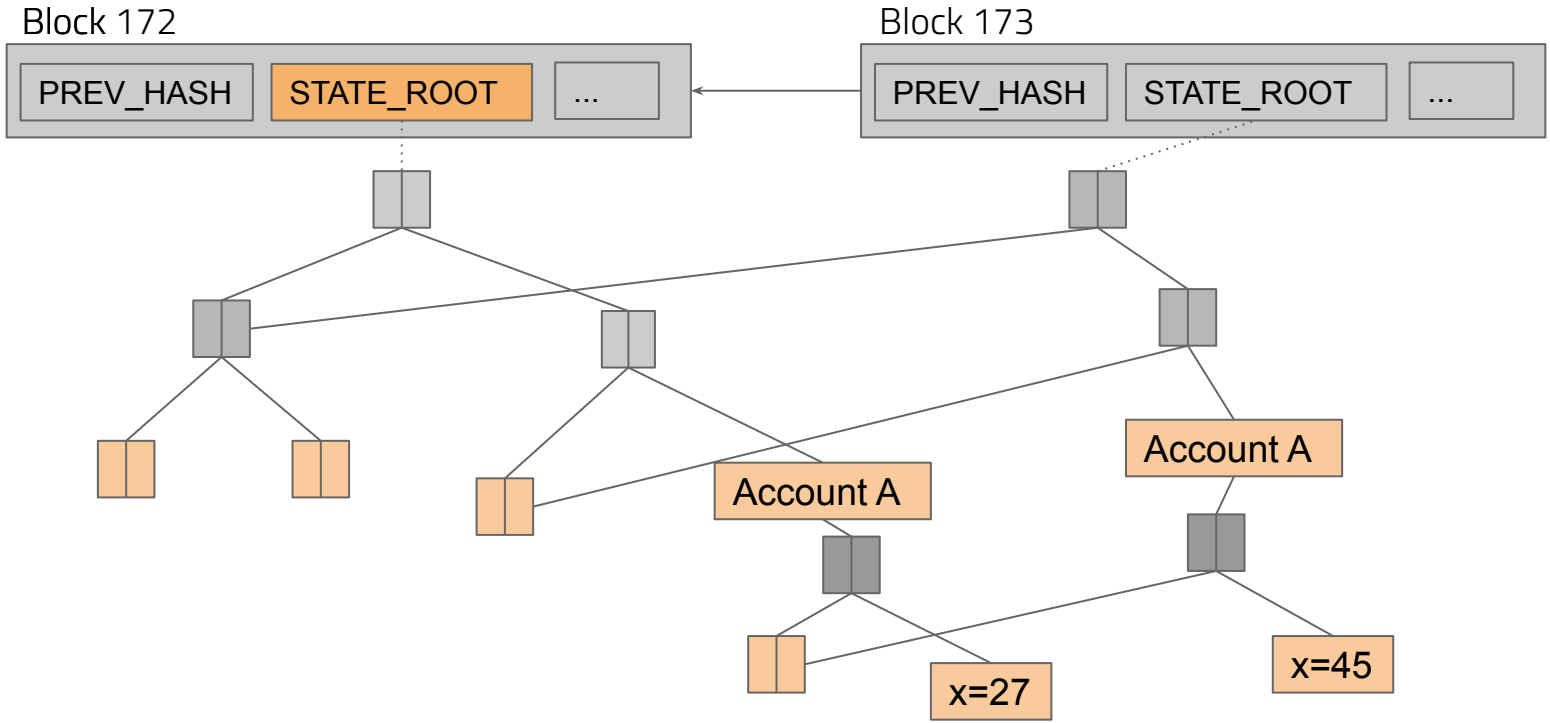
Ethereum state

Account



https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial

Ethereum state



https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial

Code demo

Creating a trie with ethereum-trie library

https://github.com/phukq/ethnoob-geth-example/tree/master/state