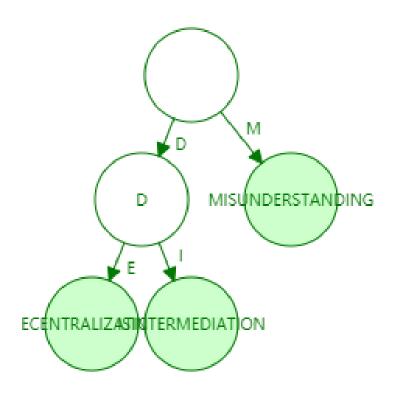
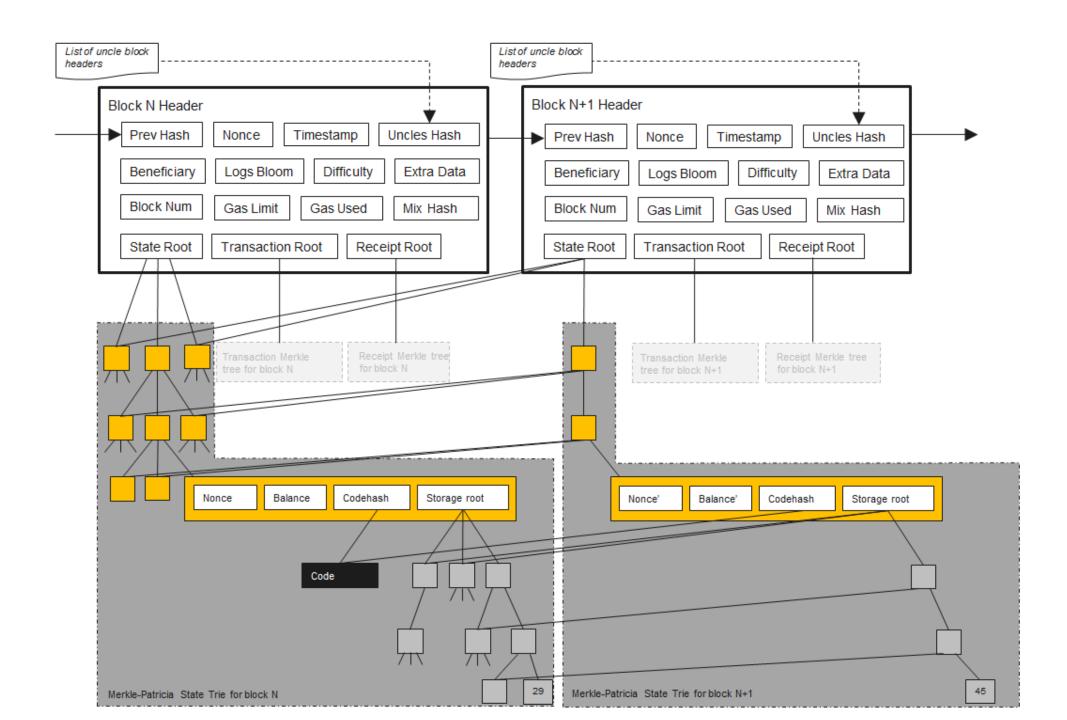


Misunderstanding Decentralization Disintermediation



Misunderstanding Decentralization Disintermediation

Ethereum Modified Merkle-Paricia-Trie System An interpretation of the Ethereum Project Yellow Paper Block Header, H or B_H Hash function: stateRoot, H. Simplified World State, σ Keccak 256-bit hash of the root KECCAK256() node of the state trie, after all Values Keys transactions are executed and finalisations applied 45.0 ETH 3 5 a World State Trie 1.00 WEI 3 3 a ROOT: Extension Node 1.1 ETH 9 3 6 a prefix shared nibble(s) next node a7 0 0.12 ETH d 3 9 a Branch Node 8 3 5 6 9 0 4 а b d value С е Leaf Node Leaf Node Extension Node prefix shared nibble(s) key-end value value prefix next node prefix key-end d3 9365 1.1ETH 1355 45.0ETH **Prefixes** 0 - Extension Node, Branch Node even number of nibbles 2 3 4 5 6 8 9 value b d а С е $1\Box$ - Extension Node, odd number of nibbles, 2 - Leaf Node, even number of nibbles Leaf Node Leaf Node 3□ - Leaf Node, odd number of nibbles prefix key-end value prefix key-end value \Box = 1st nibble **3** 1.00WEI 0.12ETH 1 nibble = 4 bits



```
// Header represents a block header in the Ethereum blockchain.
69
70
     type Header struct {
71
             ParentHash
                         common.Hash
                                         `ison:"parentHash"
                                                                   gencodec: "required"
72
             UncleHash
                         common.Hash
                                         `ison:"sha3Uncles"
                                                                   gencodec: "required"
             Coinbase
                         common.Address `json:"miner"
                                                                   gencodec:"required"`
73
74
                         common.Hash
                                         `ison:"stateRoot"
                                                                   gencodec: "required"
             Root
                                         `json:"transactionsRoot"
                                                                  gencodec:"required"`
75
             TxHash
                         common.Hash
             ReceiptHash common.Hash
76
                                         `json:"receiptsRoot"
                                                                   gencodec: "required"
77
             Bloom
                         Bloom
                                         `json:"logsBloom"
                                                                   gencodec: "required"
                                         `json:"difficulty"
             Difficulty
                         *big.Int
                                                                   gencodec: "required"
78
                                         `ison:"number"
                                                                   gencodec: "required"
79
             Number
                         *big.Int
                                         `json:"gasLimit"
80
             GasLimit
                         uint64
                                                                   gencodec: "required"
                         uint64
                                         `json:"gasUsed"
                                                                   gencodec:"required"`
81
             GasUsed
                                         `json:"timestamp"
82
             Time
                         *big.Int
                                                                   gencodec: "required"
                                         `json:"extraData"
                                                                   gencodec: "required"
83
             Extra
                         []byte
             MixDigest
                         common.Hash
                                         `json:"mixHash"
                                                                   gencodec: "required"
84
                                         `json:"nonce"
                         BlockNonce
                                                                   gencodec: "required"
             Nonce
```

```
// Block represents an entire block in the Ethereum blockchain.
144
      type Block struct {
145
146
              header
                           *Header
              uncles
                           []*Header
147
              transactions Transactions
148
149
             // caches
150
              hash atomic.Value
151
              size atomic.Value
152
153
             // Td is used by package core to store the total difficulty
154
             // of the chain up to and including the block.
155
              td *big.Int
156
157
              // These fields are used by package eth to track
158
              // inter-peer block relay.
159
              ReceivedAt time.Time
160
              ReceivedFrom interface{}
161
162
```

```
// "external" block encoding. used for eth protocol, etc.
type extblock struct {
Header *Header

Txs []*Transaction
Uncles []*Header
}
```