ETH

较BTC做的改进： 1. 出块时间变成10几秒。 以及为此设计的ghost协议。 2. mining puzzle。BTC的puzzle是基于算力设计的，这样导致了挖矿芯片的专业化。不符合去中心化。对此，ETH设计的puzzle对内存要求很高，memory hard。 3. 未来的POS（stake）代替POW 4. 智能合约

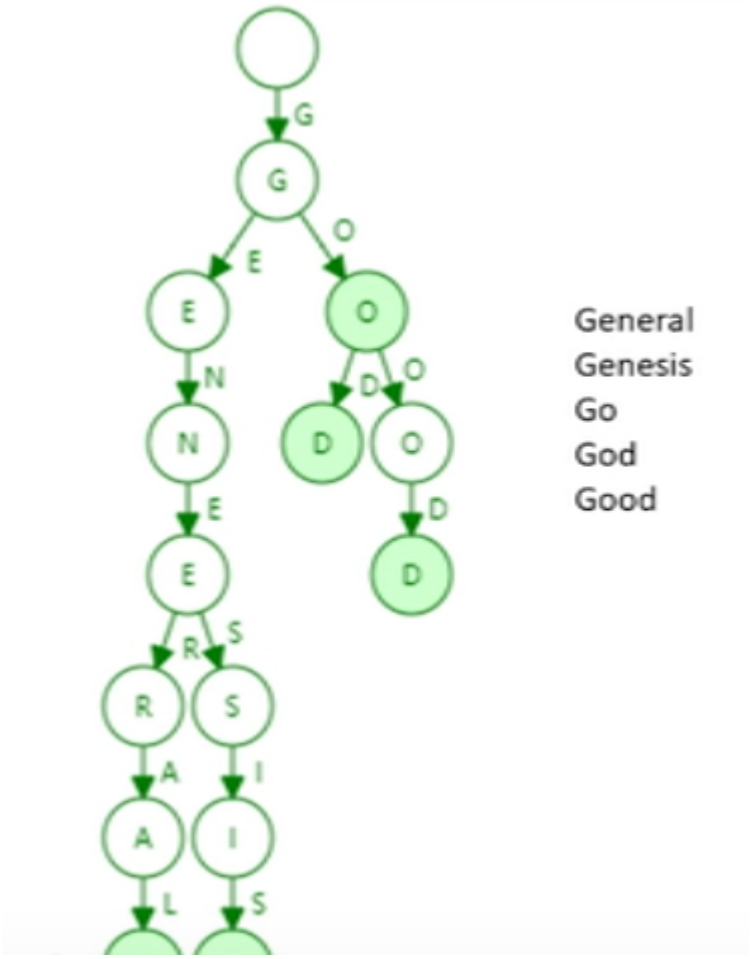
账户模式：外部账户（包括余额，nonce），合约账户（包括代码，存储）

replay attack：收款人不诚实，replay一次交易，想收两笔钱
解决办法：nonce（交易计数器），此处不用于BTC里的nonce随机数

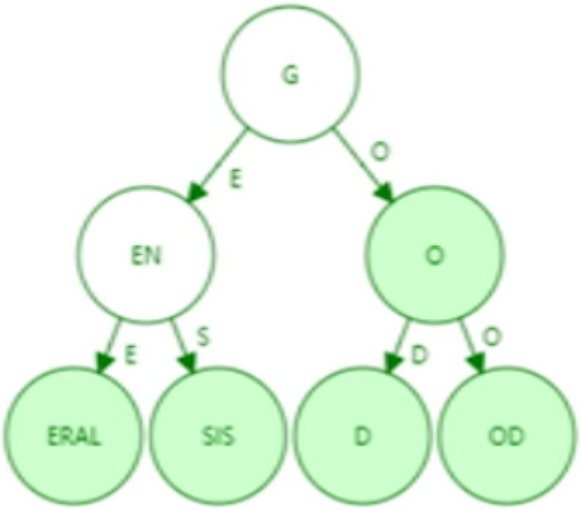
数据结构——状态树

要设计什么样的数据结构来实现addr->state映射（账户地址->账户状态）？
地址160bit，40个16进制字符表示，20个字节。

trie结构：每个分支数最大17个，0~f。地址长度一致。地址空间在tri上不会发生碰撞。自动排好序。
缺点：浪费存储



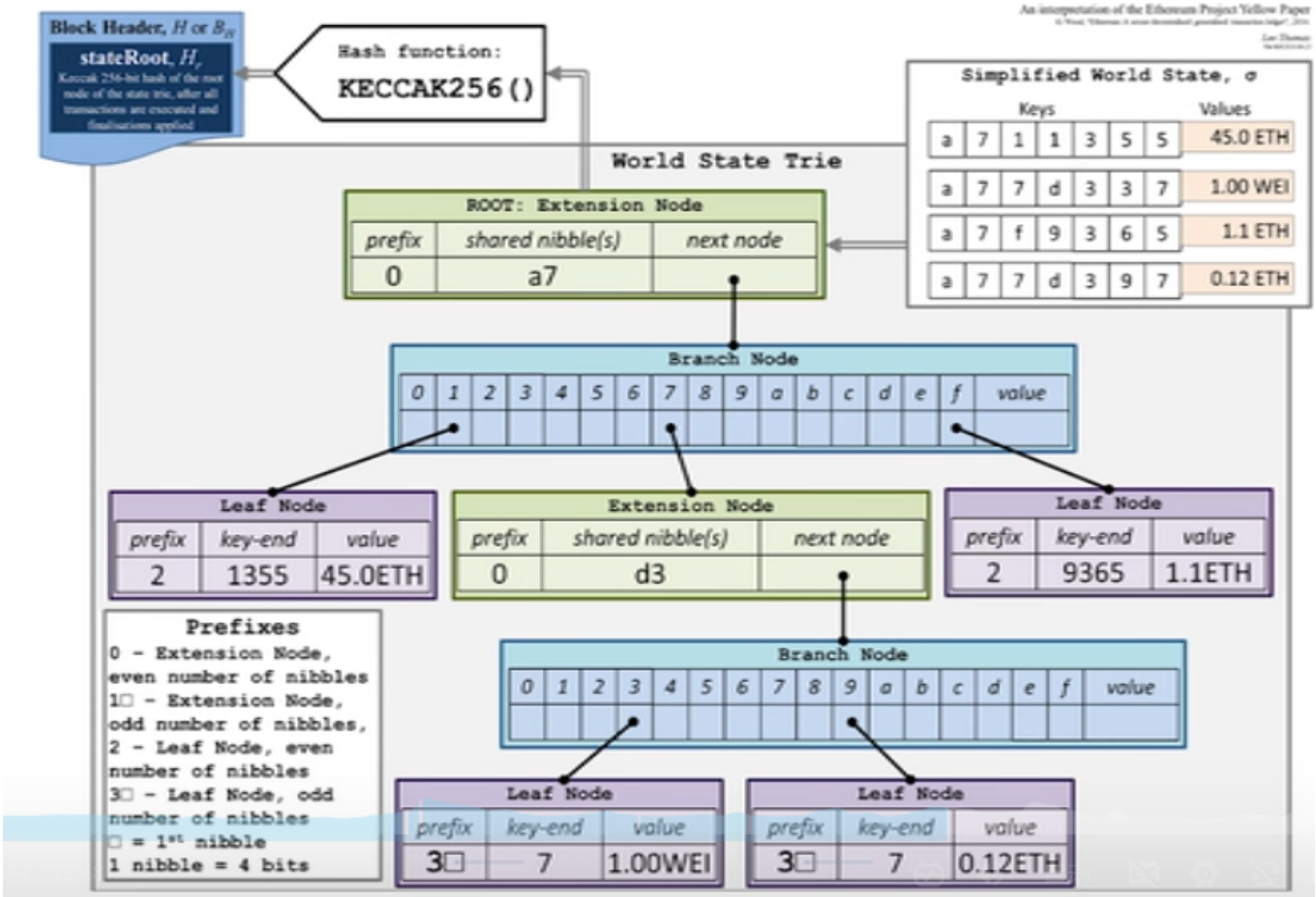
改进：patricia trie 路径压缩前缀树,树高度变短。当时value分布稀疏时，路径压缩效果明显

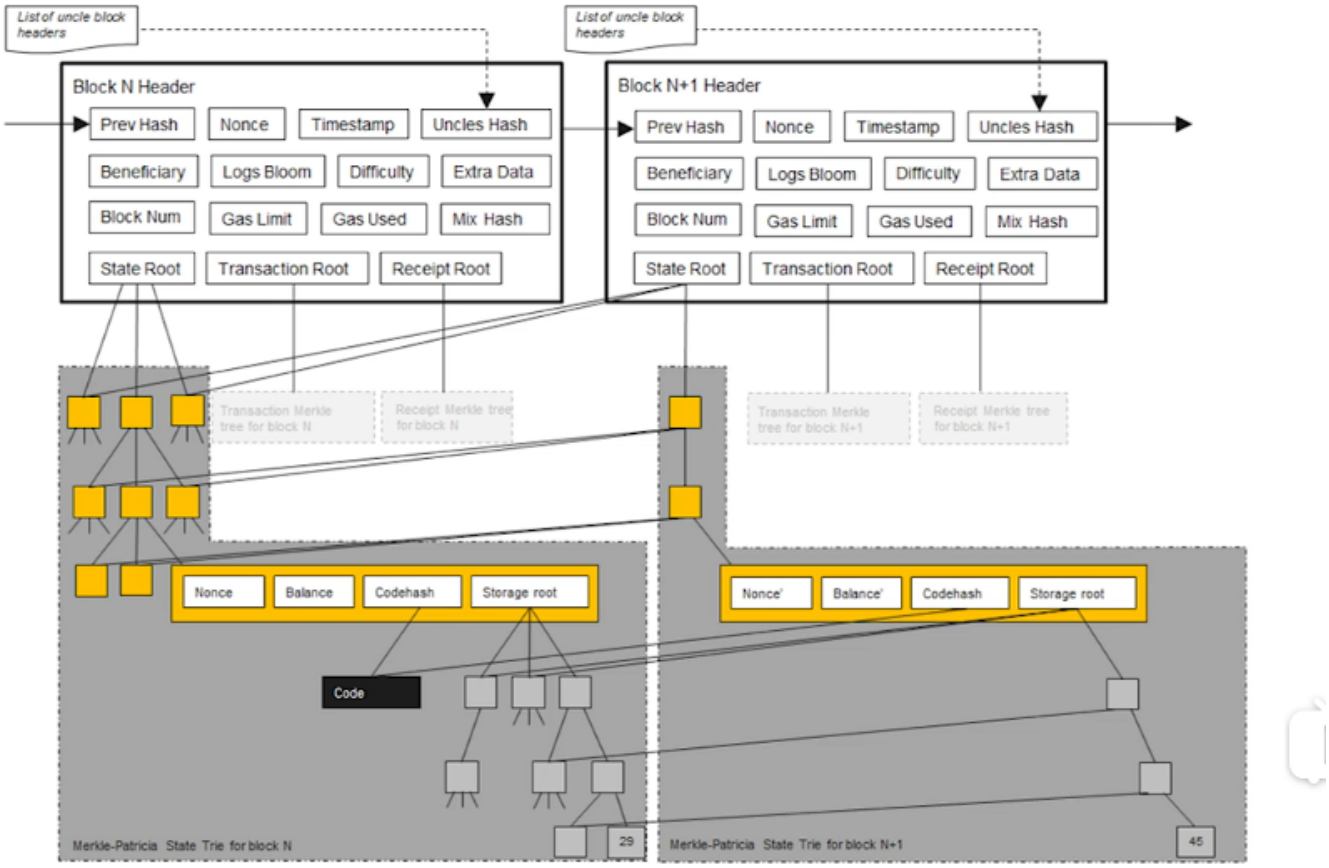


General
Genesis
Go
God
Good

MPT: merkle patricia trie,哈希指针取代传统指针，保证每个地址的状态内容不可篡改

Modified MPT：以太坊中使用的状态树数据结构





每个区块中，状态树大部分结点是不变的，可以共享，只改变少量结点即可。

```
// Header represents a block header in the Ethereum blockchain.
type Header struct {
        ParentHash  common.Hash    `json:"parentHash"       gencodec:"required"`
        UncleHash   common.Hash    `json:"sha3Uncles"       gencodec:"required"`
        Coinbase    common.Address `json:"miner"            gencodec:"required"`
        Root        common.Hash    `json:"stateRoot"        gencodec:"required"`
        TxHash      common.Hash    `json:"transactionsRoot" gencodec:"required"`
        ReceiptHash common.Hash    `json:"receiptsRoot"     gencodec:"required"`
        Bloom       Bloom          `json:"logsBloom"        gencodec:"required"`
        Difficulty  *big.Int       `json:"difficulty"       gencodec:"required"`
        Number      *big.Int       `json:"number"           gencodec:"required"`
        GasLimit    uint64         `json:"gasLimit"         gencodec:"required"`
        GasUsed     uint64         `json:"gasUsed"          gencodec:"required"`
        Time        *big.Int       `json:"timestamp"        gencodec:"required"`
        Extra       []byte         `json:"extraData"        gencodec:"required"`
        MixDigest   common.Hash    `json:"mixHash"          gencodec:"required"`
        Nonce       BlockNonce     `json:"nonce"            gencodec:"required"`
}
```

数据结构——交易树，凭证树

MPT结构

bloom filter

以太坊的运行过程：交易驱动的状态机

交易：每一区块中包含的交易 状态：每个账户的状态，状态树中的内容