Ethereum blockchain concept

Hypermine
TECHNOLOGIES

Overview : Bitcoin and existing concept

Bitcoin - A state transition system

APPLY(S, Tx) -> S'

S (State) : UTXO i.e unspent transaction o/p => [value, address]

Tx (Transaction) : One or more i/p (each,reference to existing UTXO and signature)

S' (New State) : state with all i/p UTXO removed and new o/p UTXO added

Hypermine
TECHNOLOGIES

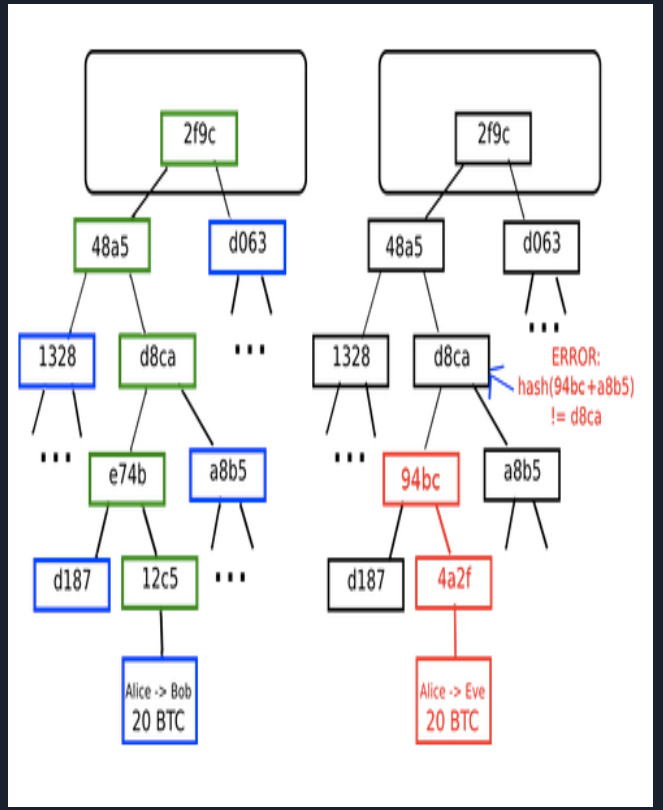
Centralize to Decentralize

- State Transition System + **Consensys System**
- Consensus required, in order to ensure everyone in the network **agrees** on order of transaction.
- No single entity can take decision
- Network has to come to an agreement

Hypermine
TECHNOLOGIES

Merkle Tree

- Block is stored in multilevel DS
- Hash of block is the hash of ***block header*** (*previous hash + nounce + timestamp +* ***Root hash***)
- **Merkle Tree** is a type of binary tree in which each leaf node is the hash the data and each non-leaf is hash of its children and which ultimately results to a single parent node called Root hash.
- Helpful because it allows miners to verify Tx without downloading whole



Hypermine
TECHNOLOGIES

Ethereum

- Distributed - Decentralized permission less public blockchain.
- Platform to build Dapps
- Blockchain with Turing Complete language

Bitcoin Vs Ethereum

1. Distributed data storage
2. Non turing complete
3. Blocktime of 10 mins
4. Block rewards decreases every 4 years
5. Tracking ownership of digital currency.
6. Tx. fee based on size

1. Distributed data storage + Computation
2. Turing complete
3. Blocktime 14-15 sec
4. Same block rewards every year
5. Focused on running prog. Code on of any Dapp
6. Tx. fee based on contract code complexity

Hypermine
TECHNOLOGIES

Accounts in Ethereum

- What are accounts why they needed?
  - To maintain state
  - 20 bytes
- Types of accounts?
  - EOA and CA
- Why there are two types ? why not only EOA

EOA

1. Externally owned account
2. Controlled by private key
3. Have balance

CA

1. Contract account
2. Controlled by code
3. Have both balance and storage for code

Hypermine
TECHNOLOGIES

Ether and Gas

- **Ether :** Digital currency in Ethereum network
- **Wei :**
  - Smallest using of ether.
  - 1 ether = 10^18 Wei
- Gas :
  - Cost of network resource / utilization.
  - Every operation (OPCODE) is associated with a number called Gas
  - Principle behind Gas is to have stable value for how much a transaction.

Hypermine
TECHNOLOGIES

Gas - Why?

Look at this piece of code:

```
while (i++ < 1000) {
    j = j + i;
}
```

- If i=0 then it will execute for 1000 times
- It has to be executed in all the miners machines
- They will spend their **resource and time**
- Hence Gas, execution cost paid by caller to miner
- **Why in Gas not in Ether?**

Hypermine
TECHNOLOGIES

Gas Terms

- **Gas Limit :** Maximum amount of Gas willing to spend on Tx.
- **Gas cost :** Static value associated with OPCODES
- **Gas Price :** Amount of ether, user pays per gas - dynamic
- **Gas fee:** (Gas Cost * Gas Price)

Hypermine
TECHNOLOGIES

EVM - Ethereum Virtual Machine

- Stack Based virtual machine
- Ships with Ethereum clients like Geth
- Consists of **Stack** and **Store**/Memory

Implementation

- **go-ethereum**
- parity
- cpp-ethereum
- py-EVM

Hypermine
TECHNOLOGIES

Compiler

- **solc** is a compiler which compiles .sol files, written in c++
- Install : http://solidity.readthedocs.io/en/v0.4.21/installing-solidity.html
- Usage : solc [option] [input files]
- Ex. solc --bin test.sol

Hypermine
TECHNOLOGIES

Wallets & Remix

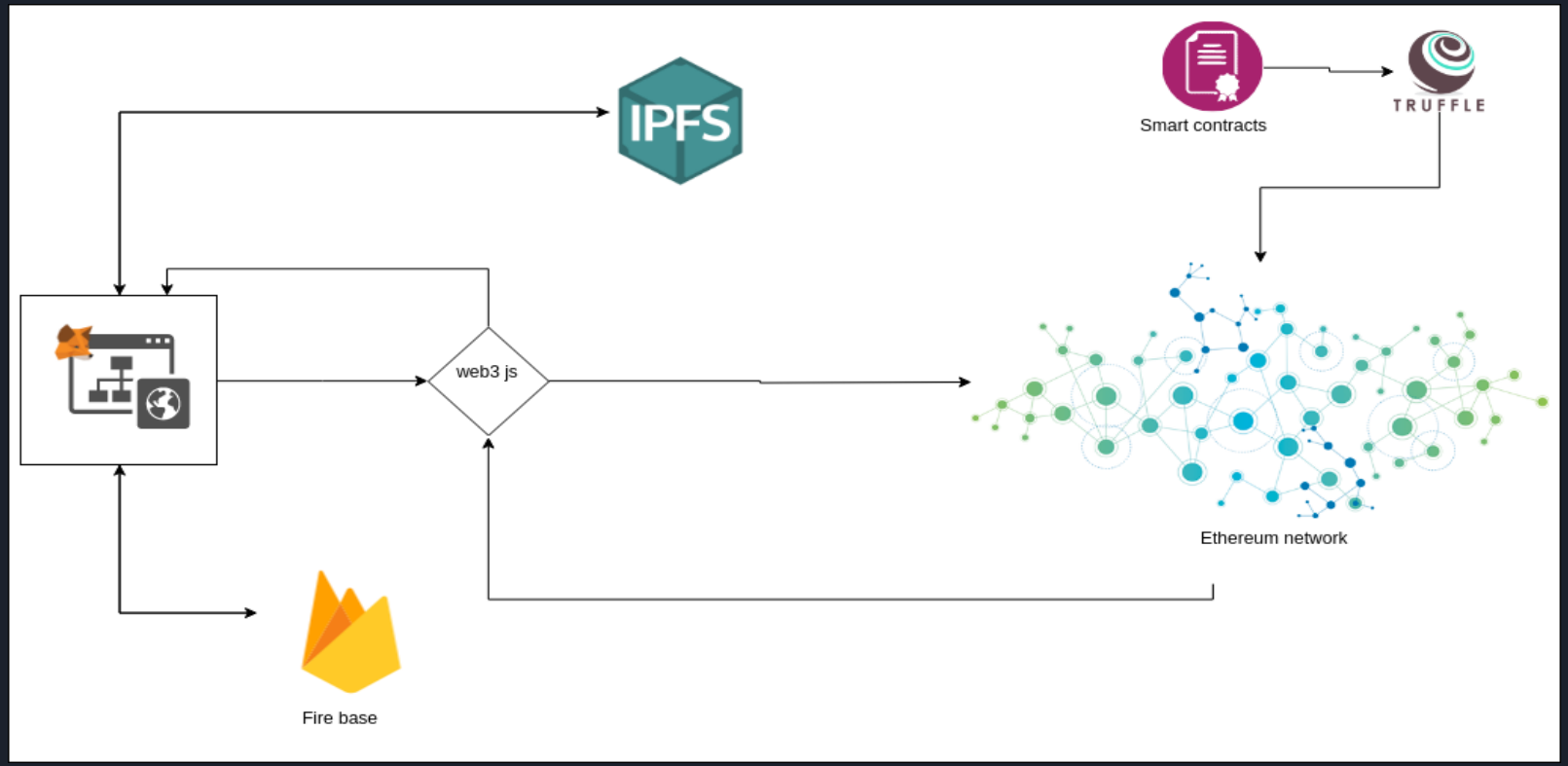
Wallets

- Metamask - Extension
- Mist - Desktop Wallet
- MyEthereumWallet - Web based

Remix

Hypermine
TECHNOLOGIES

DApp - Decentralized applications



Hypermine
TECHNOLOGIES

References

Ethereum White Paper

Ethereum yellow paper

Centralize, Decentralize, Distributed

EVM

Ethereum architecture

How to use Metamask

Mist Wallet Demo

Remix

Hypermine
TECHNOLOGIES