The Blockchain

bonus content by Edan Sneh

Vocabulary

Transaction - an atomic unit of data on the blockchain

Block - Object in chain containing multiple transactions and prev and current hash

Blockchain - A chain of blocks corresponding to a non-modifiable database

Node - Process that holds the blockchain

Miner - Process that runs PoW until 000x…xxx hash is found

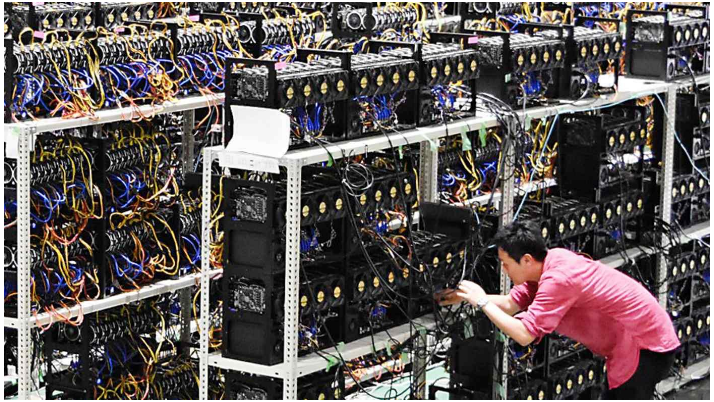
Nodes

- Validate transactions **(No double spending)**
- Keep a historic record of transactions (**Store blockchain**)
- Dictate and enforce the rules of the network. (**No bulls**t!**)

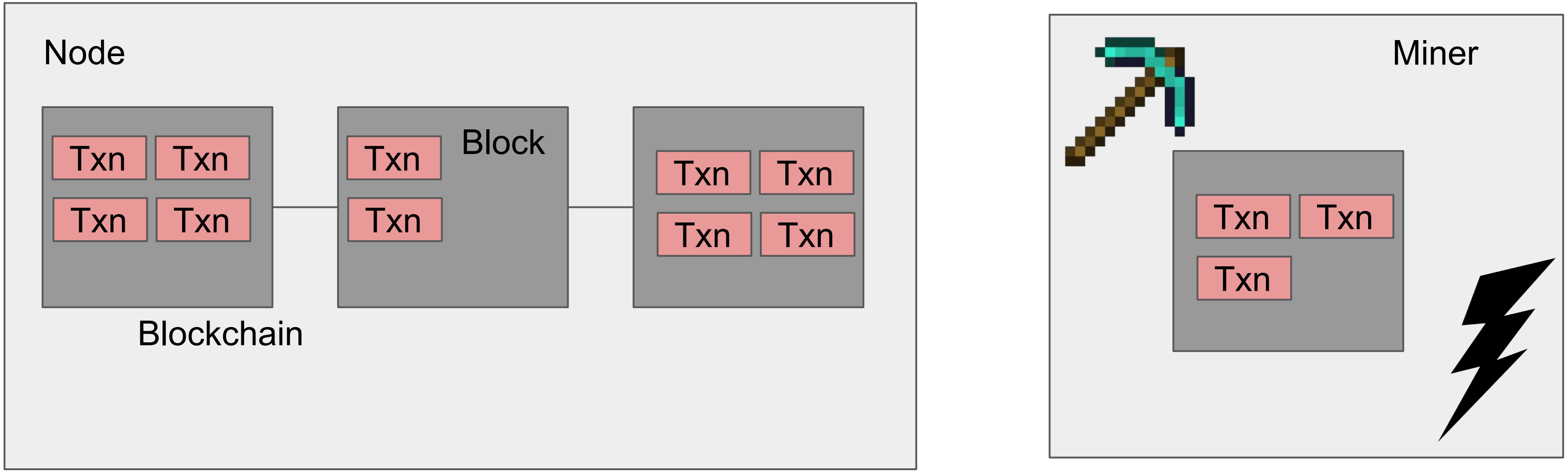
```
RaspiBlitz v1.5  RecklessBlitz
bitcoin Fullnode + Lightning Network + TOR
CPU load 0.40, 0.45, 0.43, temp 51°C 124°F
Free Mem 1322M / 1939M  HDDuse 308G (36%)
ssh admin@192.168.178.95 d22.6MiB u15.4MiB
web admin --> http://192.168.178.95:3000
bitcoin v0.19.1 mainnet Sync OK 100.00%
Public rvxlan6p2guj2ren.onion:8333 11 peers

LND 0.10.0-beta wallet 61148 sat
2/2 Channels 371841 sat 4 peers
Fee Report: 0-0-0 sat (D-W-M)

033edbdfc7f382ab0858b9e4382f171e2ba4c12cd09b8243ad172339cfb2fb
f7b8@b4tnjucd3osmdqddaujn75zf4qbh6lpcztyfvswfmhtkfdxtxp3tq7id.
onion:9735
```

Miners

- Confirm transactions (put transactions into blocks with PoW)
- Secure the blockchain (Keep track of largest chain and continue building it)
- Gain $$$ reward (often transaction fee for solver)



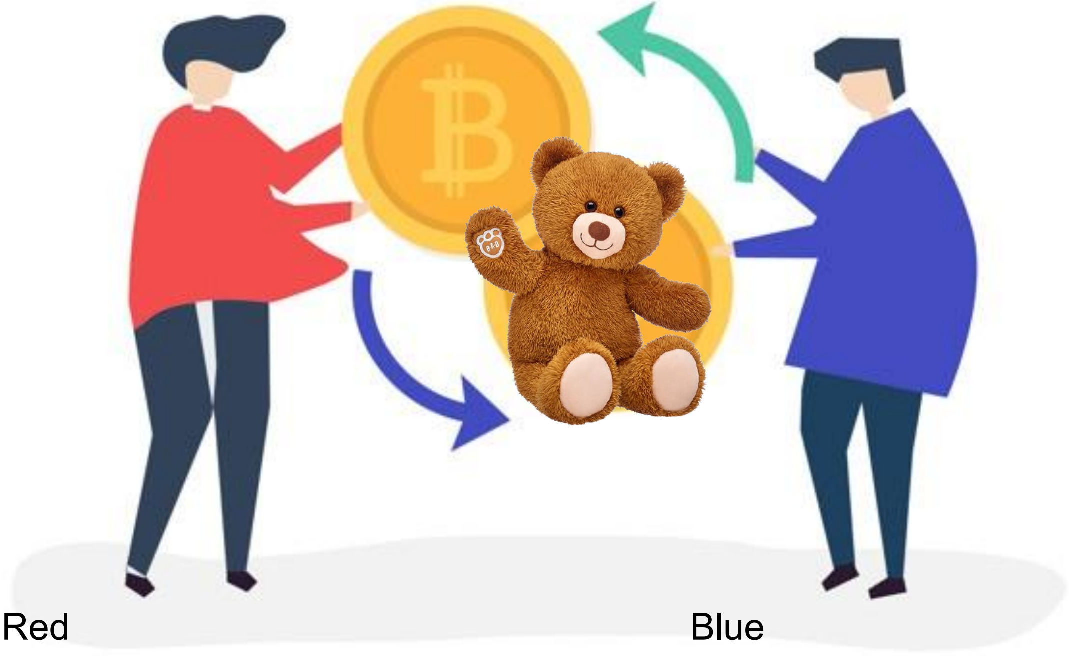
Diagram


Node
Txn
Txn
Txn
Txn
Txn
Block
Txn
Txn
Txn
Txn
Txn
Blockchain
Miner
Txn
Txn
Txn

Walkthrough

I want to buy this teddy bear with my bitcoin!

Red’s acc: c766227e7af569848...286e6ef5



Red

Blue

Tx1:
Log - Gave red 1 bc
Hash: **37df**...aef
Prev hash: ???

Tx2:
Log - red gave blue 1 bc
Hash: **ad80**...2e2
Prev hash: **37df**

Blue shouldn’t give away his precious teddy bear yet!!

Hash contains red’s public key

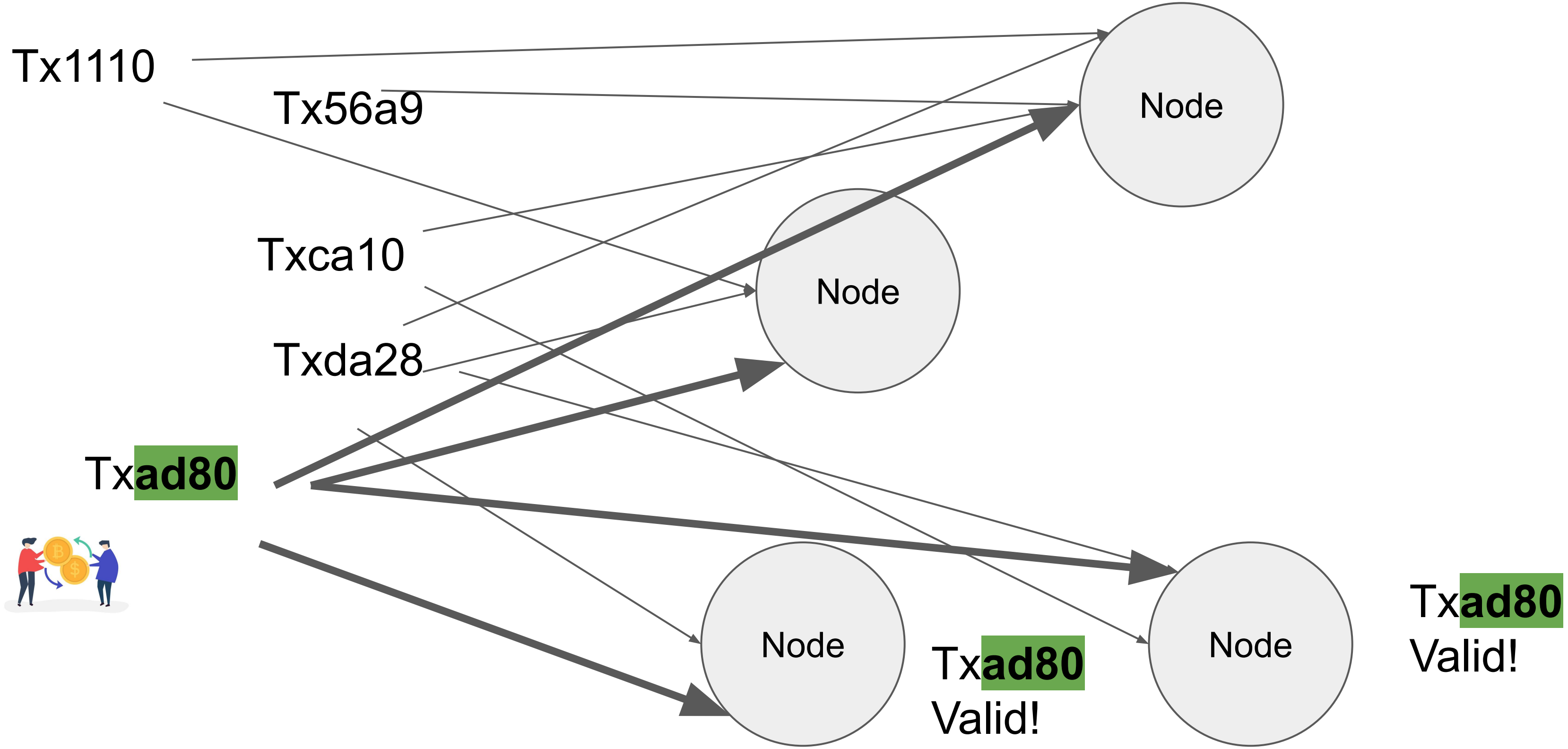
Tx1:
Log - Gave red 1 bc
Hash: **37df**...aef
Prev hash: ???

Hash signed with reds private key
Proving red owns coin in Tx1

Tx2:
Log - red gave blue 1 bc
Hash: **ad80**...2e2
Prev hash: **37df**

Hash contains blue’s public key

Transaction Validation



Tx1110

Tx56a9

Txca10

Txda28

Tx**ad80**

Node

Node

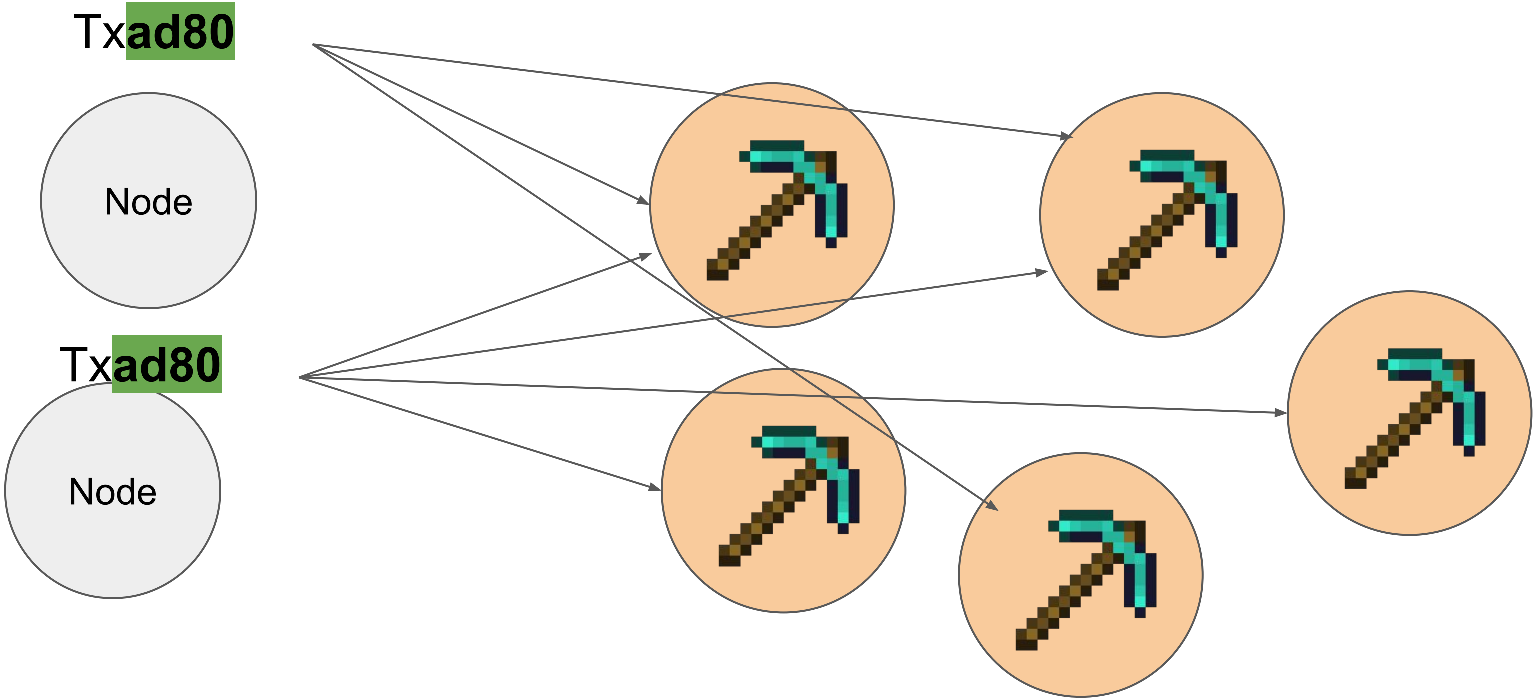
Node

Node

Tx**ad80**
Valid!

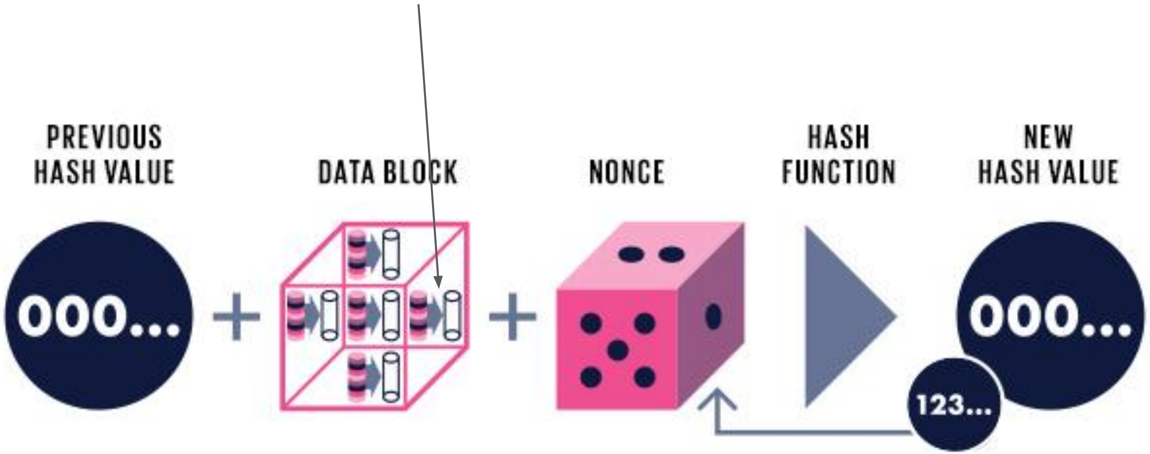
Tx**ad80**
Valid!

Mining time: P2P Network on top of internet

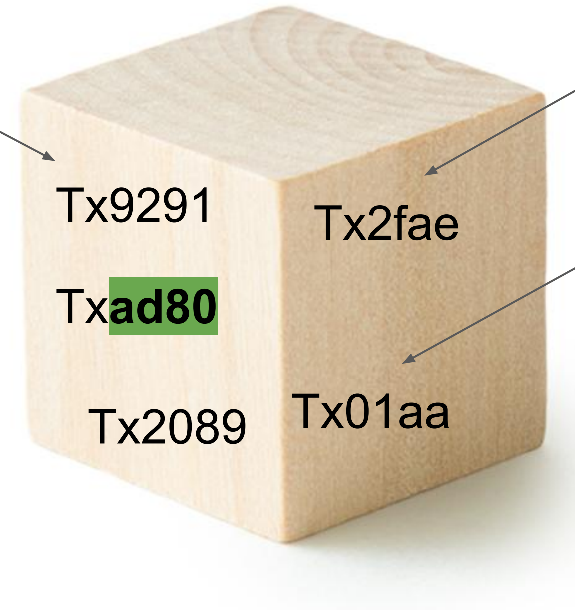


Proof of Work (PoW)

Txad80



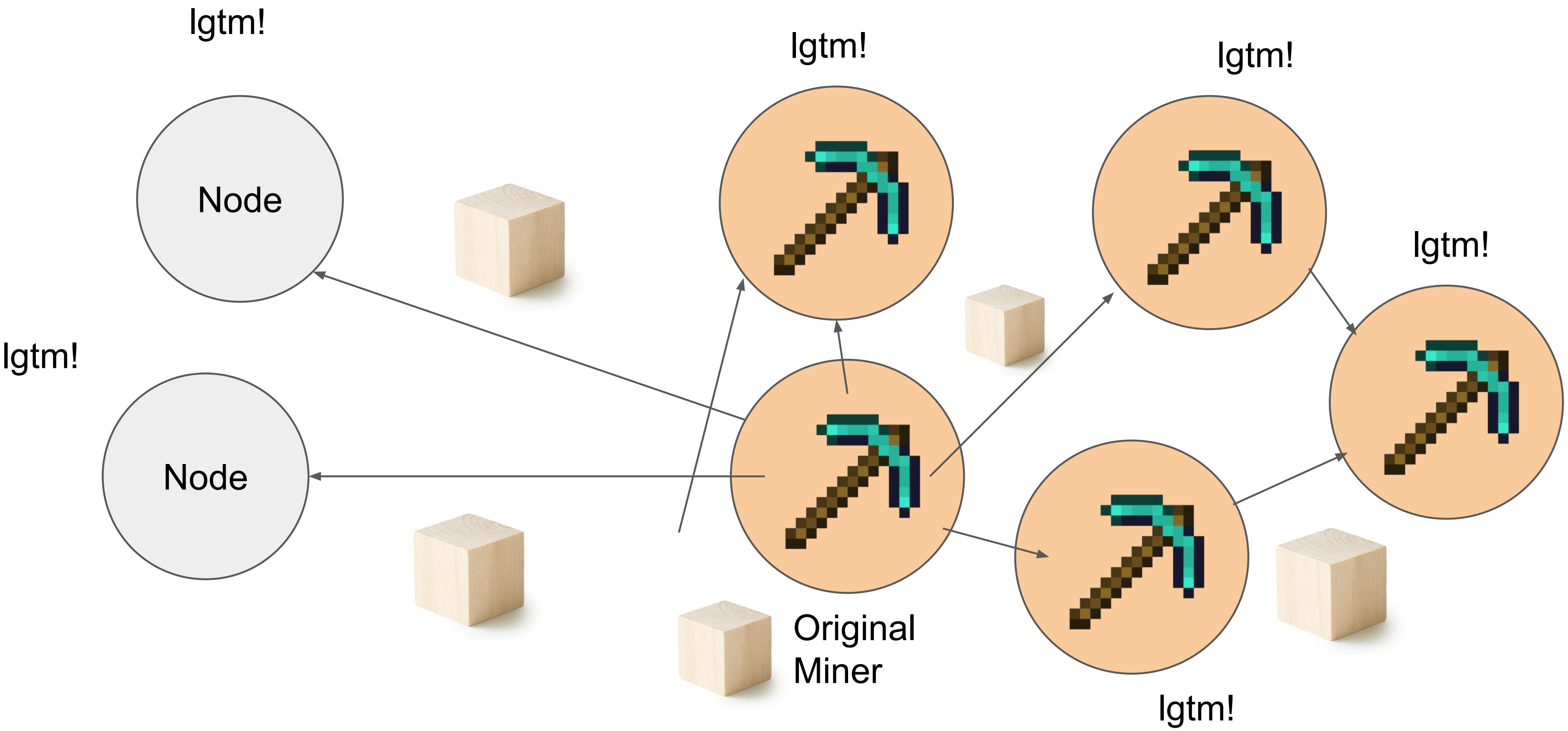
Yay! Red's transaction has made it into a block



Miner's cut!

Or empty space
in transactions for
miners address

Block Verification - Nodes add block to blockchain!



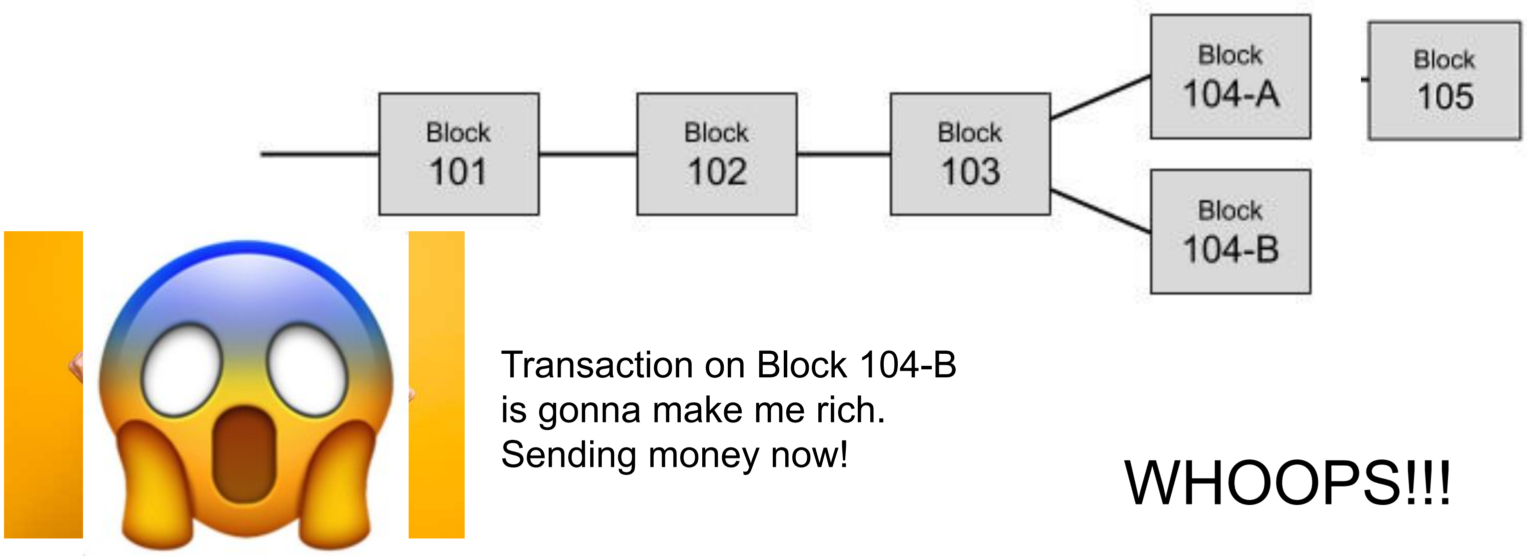
Discuss:

- Should blue hand over their Teddy bear now? Why?
- What are some weaknesses of blockchain?
- Why is decentralization important?
- What are some applications of blockchain?

https://tinyurl.com/btcblk

Transaction validity (Race Attack)

Common practice is to wait until block is 3 deep into chain before accepting. Since top block can change



Transaction on Block 104-B
is gonna make me rich.
Sending money now!

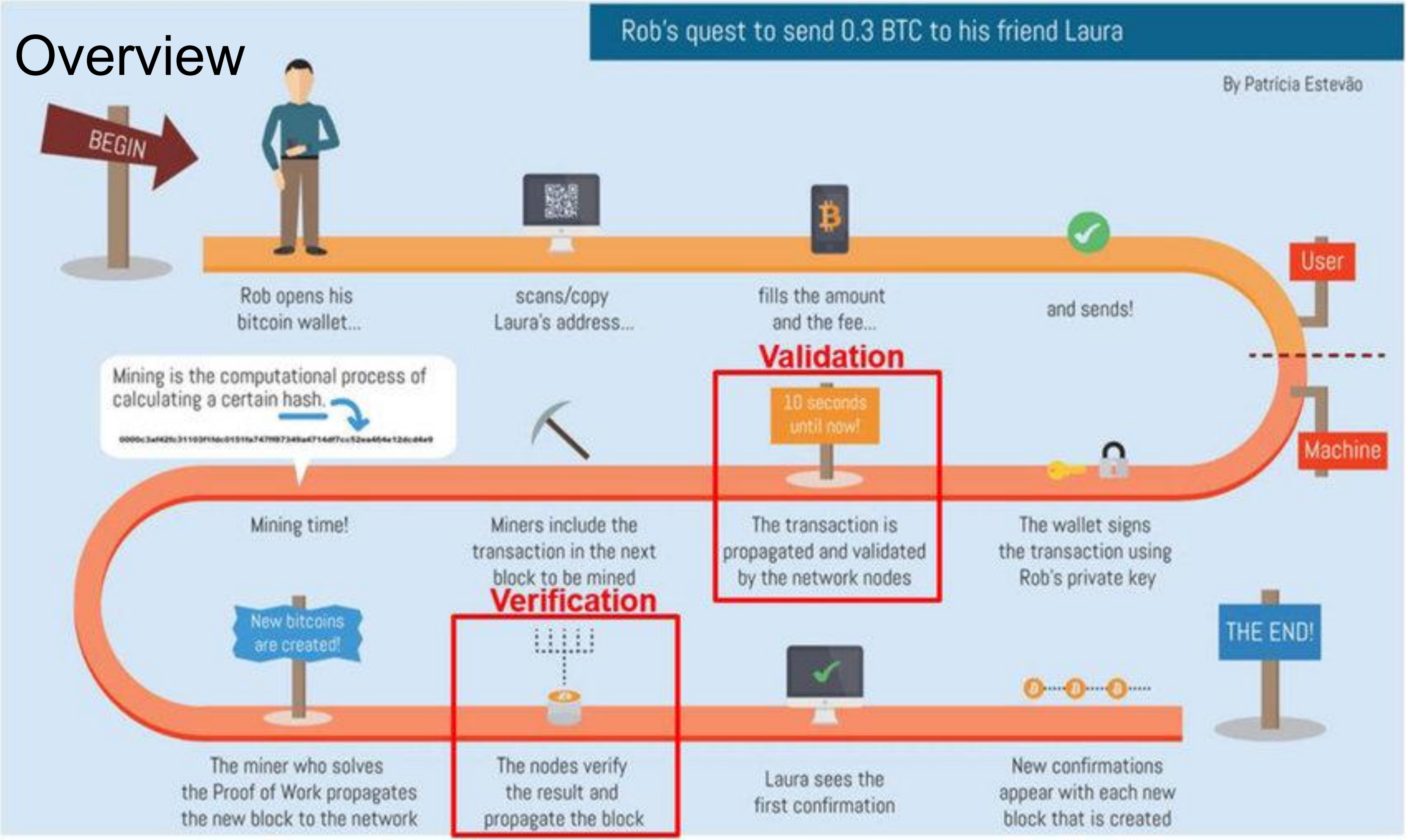
WHOOPS!!!

Blue can now give red teddy bear

Red

Blue

Overview



Rob opens his bitcoin wallet...

scans/copy Laura's address...

fills the amount and the fee...

and sends!

The wallet signs the transaction using Rob's private key

Validation

The transaction is propagated and validated by the network nodes

Miners include the transaction in the next block to be mined

Mining time!

Mining is the computational process of calculating a certain hash.

0000c3af42fc31103f1dc0151fa747f87349a4714df7cc52ea464e12dcd4e9
New bitcoins are created!

The miner who solves the Proof of Work propagates the new block to the network

Verification

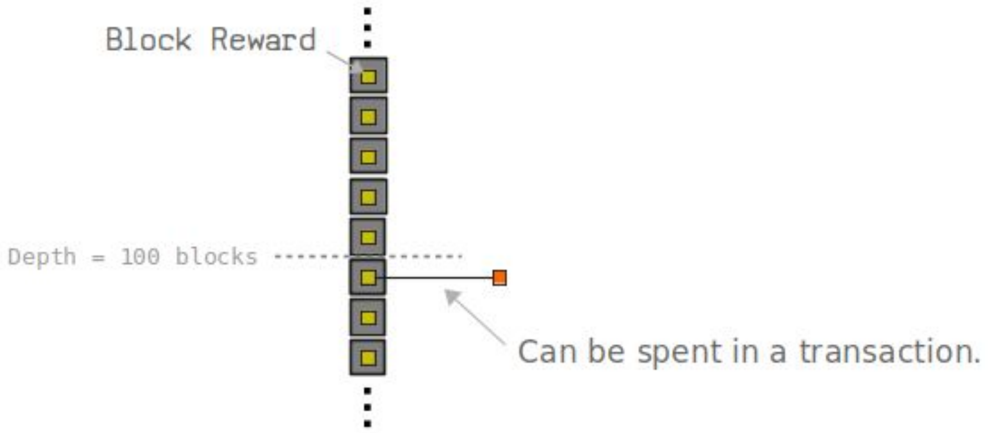
The nodes verify the result and propagate the block

Laura sees the first confirmation

New confirmations appear with each new block that is created

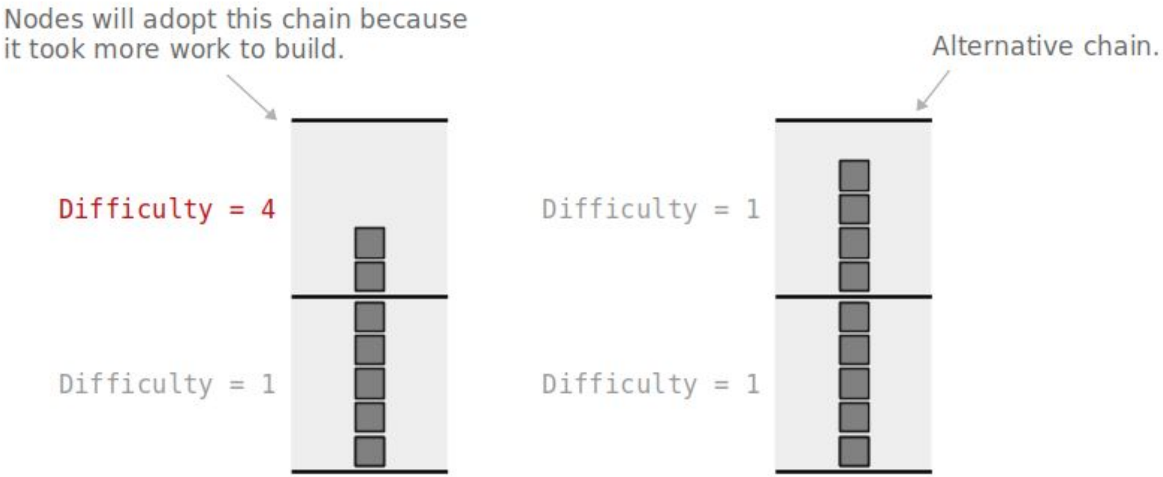
Miner Reward

Only achieved if block is 100 blocks deep in the chain.

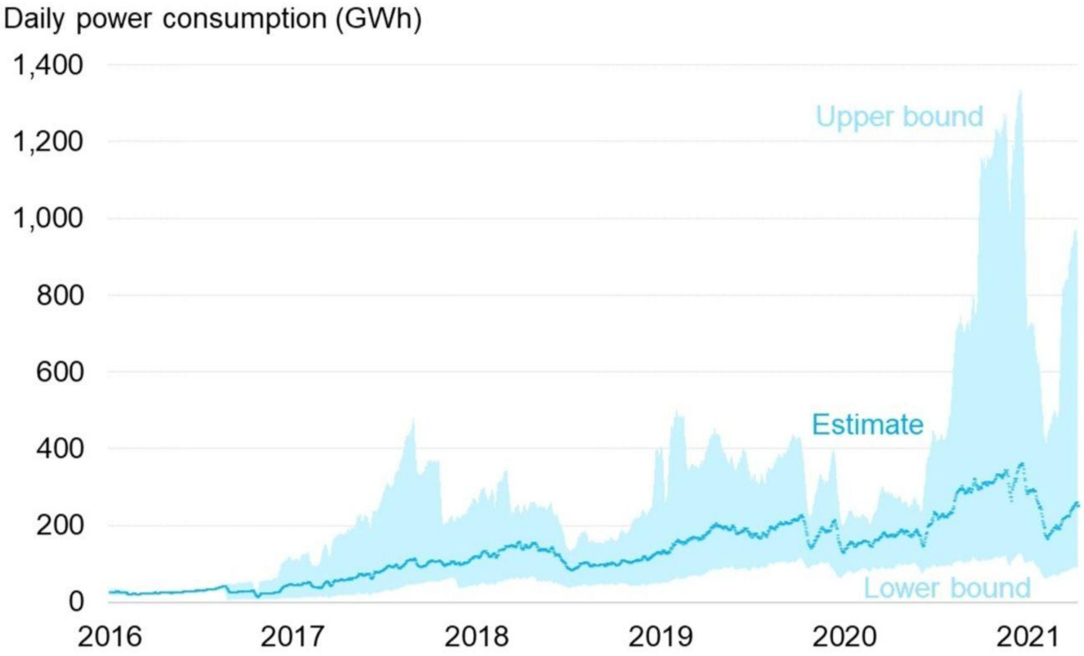


Chainwork

For bitcoin - Longest chain doesn’t necessarily mean literal longest, it means chain with the most “chainwork”



Issue - Energy use increases with Moore's law!



On par with a small country

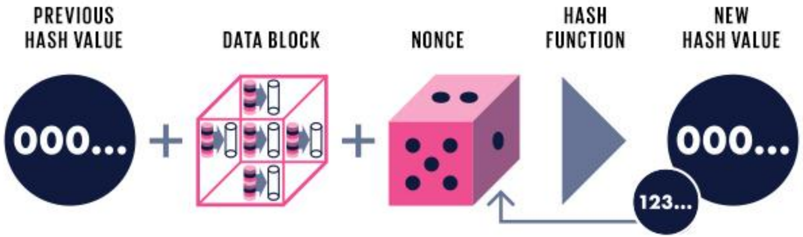
Why bother with PoW?

Solves

- Blockchain conflict
- Node creation and creation time
- Coin generation and distribution
- Incentive

Problems

- Energy
- 51% attack
- Mining pool

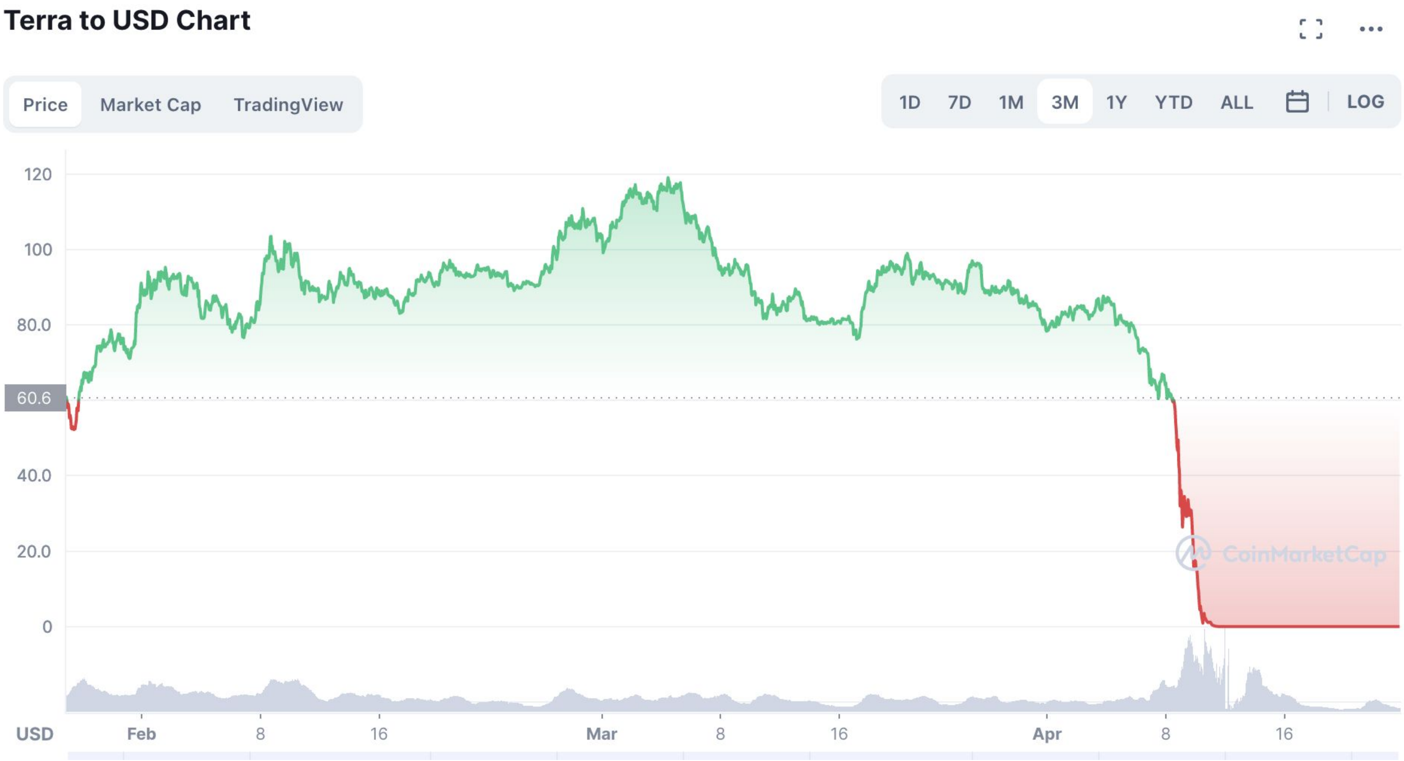


Consensus Mechanism

- Proof of Stake
  - Validators put “collateral” in blockchain. Validators picked at random based on collateral size
  - If validator enters faulty transaction a fraction of collateral is lost.
- Proof of Capacity
  - Instead of cpu power PoC relies on disk space
- Proof of Authority
  - Moderators: block validators
- Practical Byzantine Fault Tolerance
  - f faulty replicas, n-f>f. But f faulty in n-f, so n - 2f > f, n > 3f replicas.
  - Not as decentralized as PoW, performance drop with more replicas.

Backing up

Terra - A case study



Lesson?

Just because the algorithm is cool doesn’t mean you should invest your life savings

Not everything blockchain related is good. There are a lot of scams

Etherium

Crypto isn’t everything blockchain can do:

A blockchain with new types of transactions:

- Regular transactions - “What we just learned”
- Contract deployment transactions - “classes”
- Execution of a contract - “calls”

Pay “gas” to execute code

Smart contracts

A Transaction creates a programmable contract (aka class)

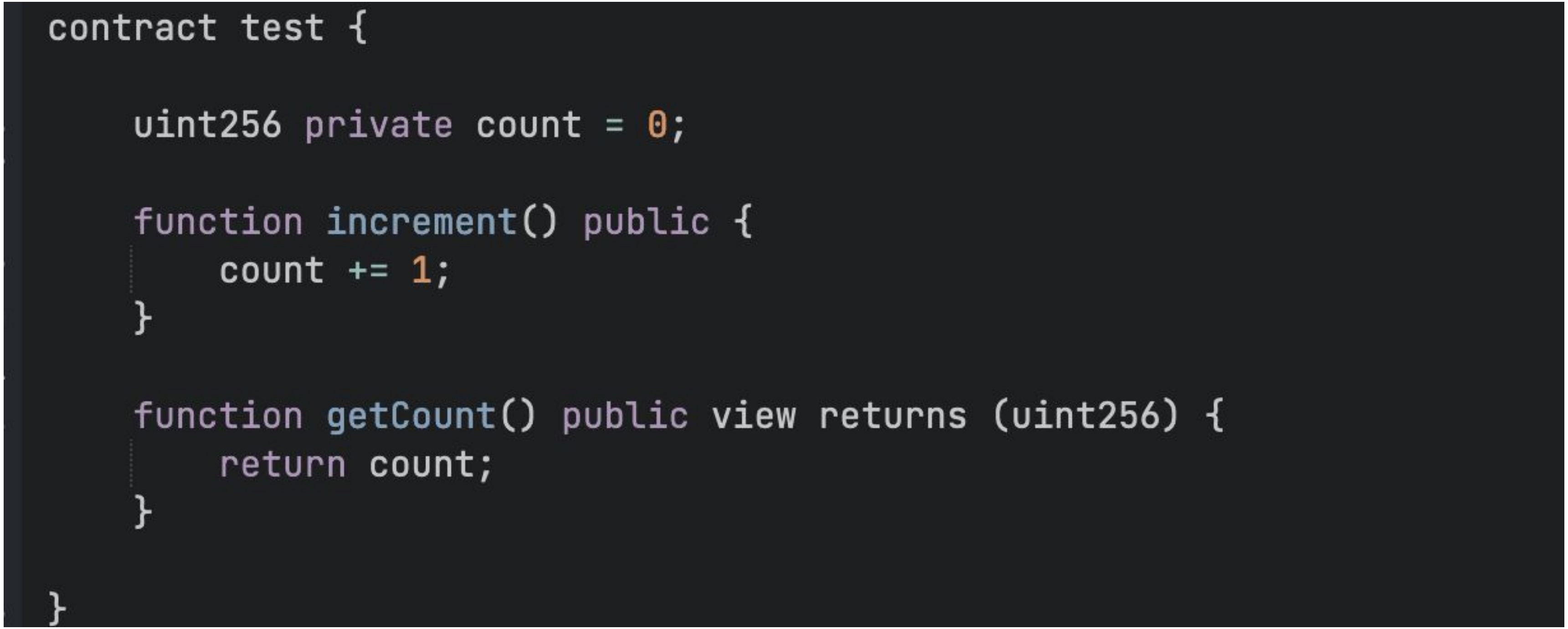
Contract is run in Ethereum VM on all nodes

Contract can never be modified

```
pragma solidity >=0.5.0 <0.7.0;

contract Store {
    function greet() public view returns (string) {
        Return “Welcome to teh store”
    }
```

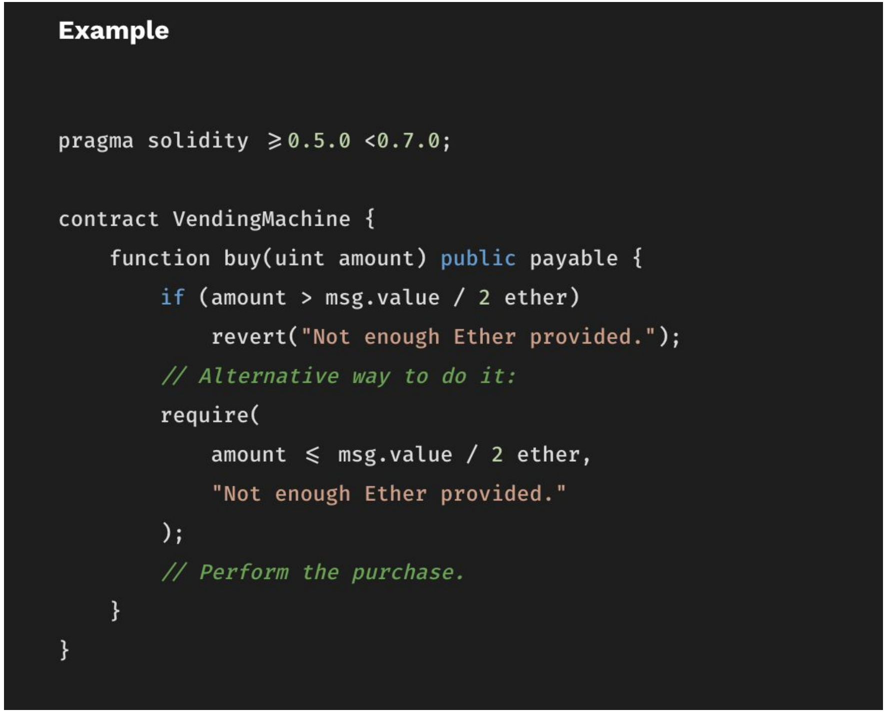
SOLIDITY

Smart Contract Example



```
contract test {

    uint256 private count = 0;

    function increment() public {
        count += 1;
    }

    function getCount() public view returns (uint256) {
        return count;
    }

}
```

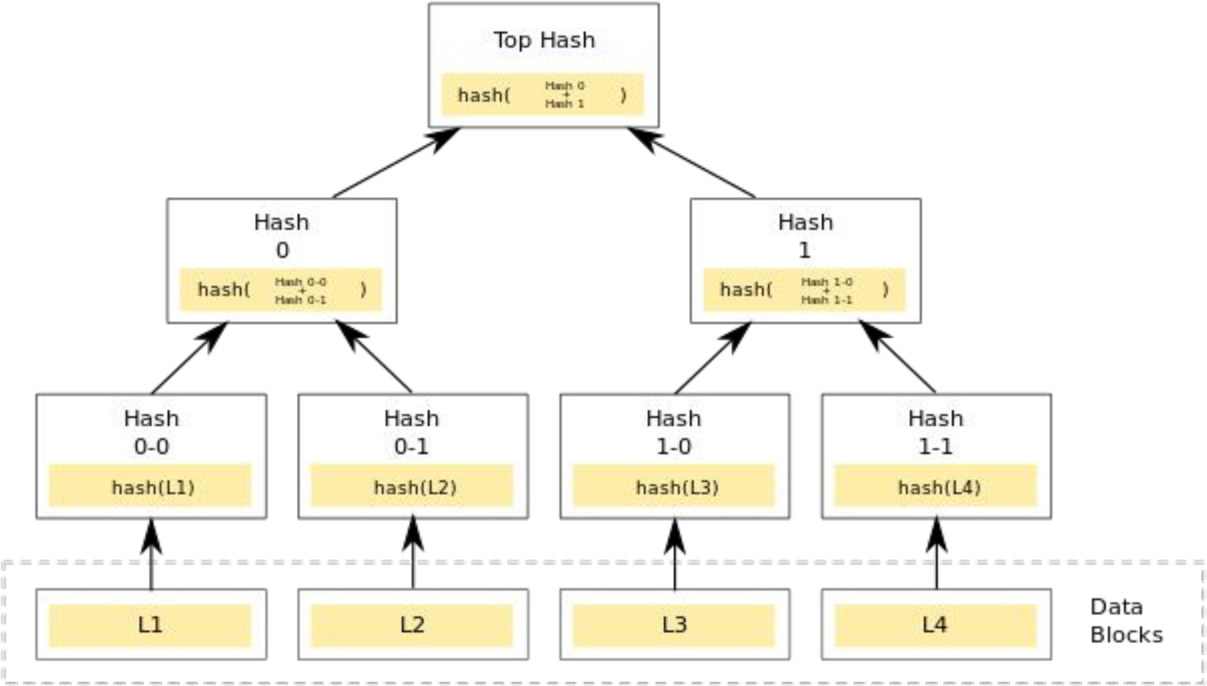
Smart Contract Cont. Require



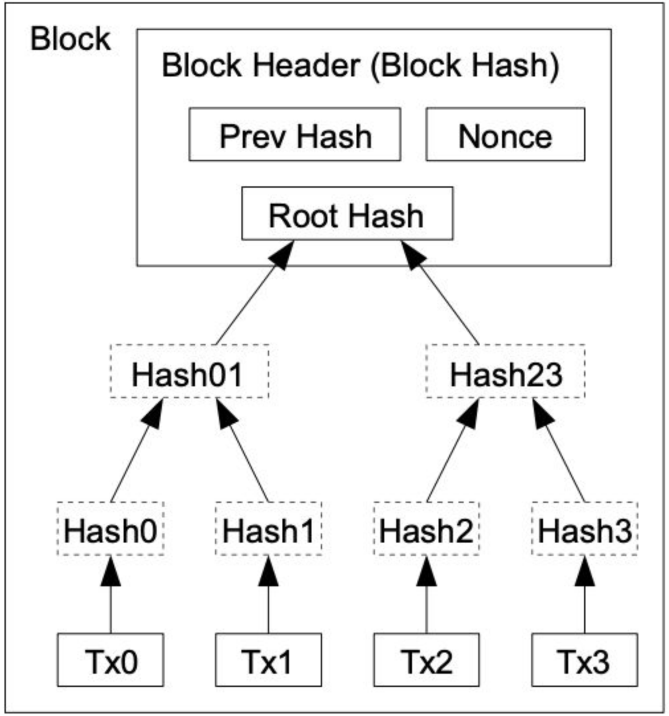
Example

```
pragma solidity >=0.5.0 <0.7.0;

contract VendingMachine {
    function buy(uint amount) public payable {
        if (amount > msg.value / 2 ether)
            revert("Not enough Ether provided.");
        // Alternative way to do it:
        require(
            amount <= msg.value / 2 ether,
            "Not enough Ether provided."
        );
        // Perform the purchase.
    }
}
```

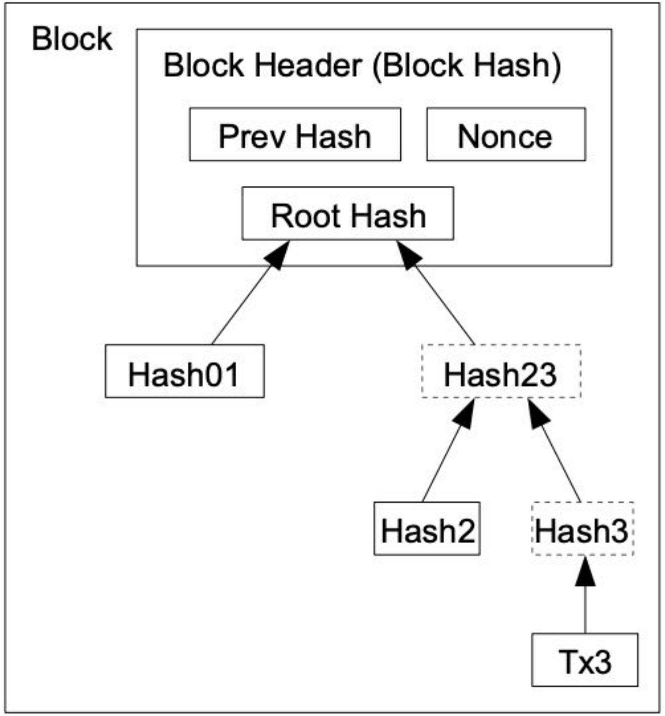
Merkle Tree


Top Hash
hash(Hash 0 + Hash 1)
Hash 0
hash(Hash 0-0 + Hash 0-1)
Hash 1
hash(Hash 1-0 + Hash 1-1)
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4
Data Blocks

Merkle Tree: Pruning

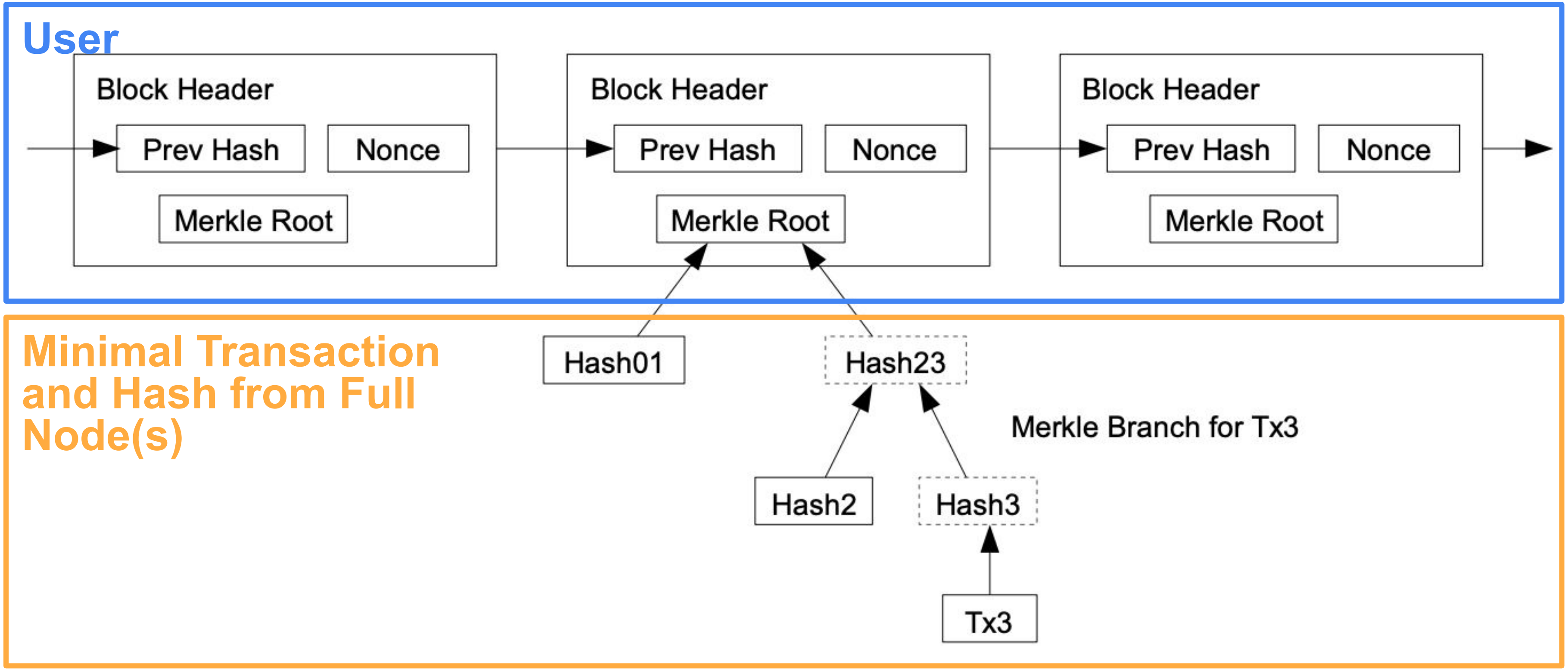


Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

Merkle Tree: Simplified Payment Verification



User

Minimal Transaction and Hash from Full Node(s)

Hard, Soft Forks and Chain splits

What happens when things go wrong

Soft fork

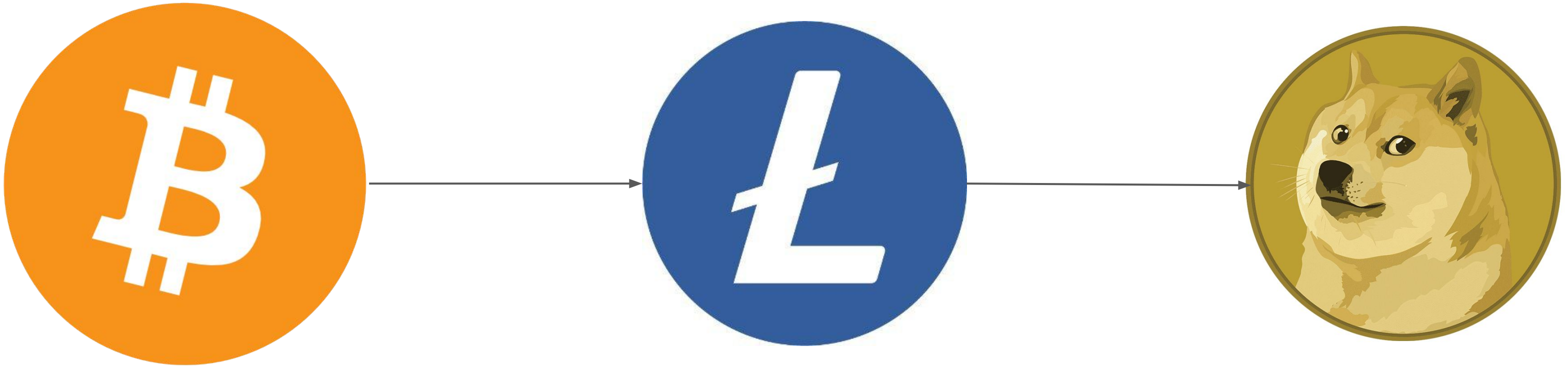
- Backwards compatible
- Previously valid blocks are made invalid.
- Old nodes recognize new block as valid.
- Ex: Decrease **max** block size from 1 MB to 0.5 MB

Only 1 blockchain!

Hard Fork

- Not backwards compatible
- Blocks previously invalid are now valid and previously valid blocks are invalid
- Ex: Change block size from 1MB to a strict 2MB

Multiple Blockchains!



More applications, more concepts

- Decentralized Finance (DeFi)
- Non-fungible token (NFT)
  - Opensea.io
- Privacy-Preserving Compute Network
- ...

Quantum Computers!

https://crypto.stackexchange.com/questions/59375/are-hash-functions-strong-against-quantum-cryptanalysis-and-or-independent-enoug

Beware Shor’s algorithm!