

BLOCKCHAIN

CADEC 2018 - Pär Wenåker & Peter Larsson



BITCOIN PAPER

Posted 31/10 2008

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model.

Bitcoin v0.1 released

Satoshi Nakamoto [satoshi at vistomail.com](mailto:satoshi@vistomail.com)

Thu Jan 8 14:27:40 EST 2009

- Previous message: [\[tmoore at seas.harvard.edu: \[fc-announce\] Financial Crypto February 23-26 in Barbados, Early Registration Deadline Approaching\]](#)
- Next message: [MD5 considered harmful today, SHA-1 considered harmful tomorrow](#)
- Messages sorted by: [\[date\]](#) [\[thread\]](#) [\[subject\]](#) [\[author\]](#)

Announcing the first release of Bitcoin, a new electronic cash system that uses a peer-to-peer network to prevent double-spending.
It's completely decentralized with no server or central authority.

See bitcoin.org for screenshots.

Download link:

<http://downloads.sourceforge.net/bitcoin/bitcoin-0.1.0.rar>

Windows only for now. Open source C++ code is included.

- Unpack the files into a directory
- Run BITCOIN.EXE
- It automatically connects to other nodes

If you can keep a node running that accepts incoming connections, you'll really be helping the network a lot. Port 8333 on your firewall needs to be open to receive incoming connections.

The software is still alpha and experimental. There's no guarantee the system's state won't have to be restarted at some point if it becomes necessary, although I've done everything I can to build in extensibility and versioning.

You can get coins by getting someone to send you some, or turn on Options->Generate Coins to run a node and generate blocks. I made the proof-of-work difficulty ridiculously easy to start with, so

WHY EVEN BOTHER



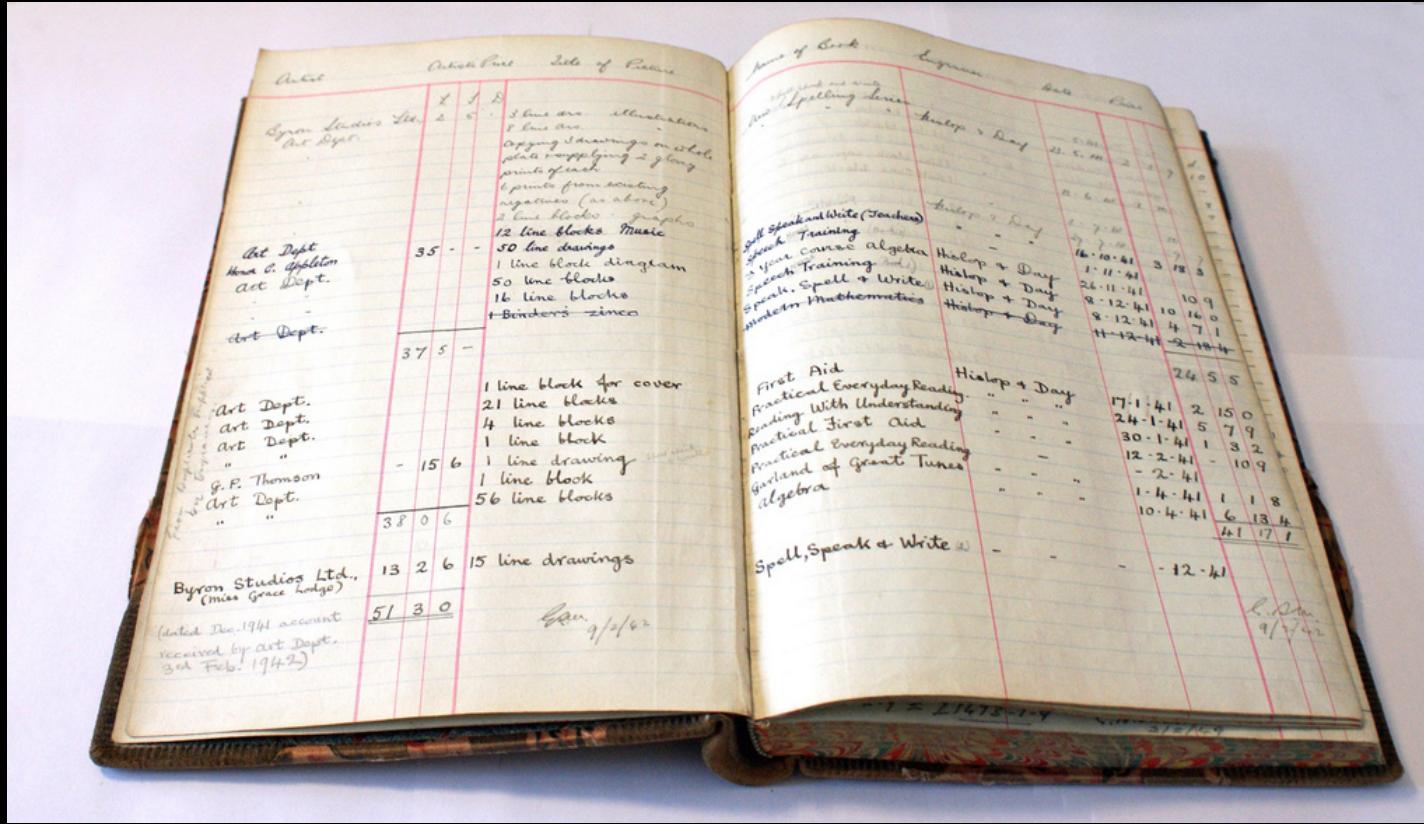
BLOCKCHAIN

Blockchain is a platform for exchange of value

Digital assets can be moved but not copied

No trust in a third party is needed

BLOCKCHAIN LEDGER



<https://www.flickr.com/people/30239838@N04/>

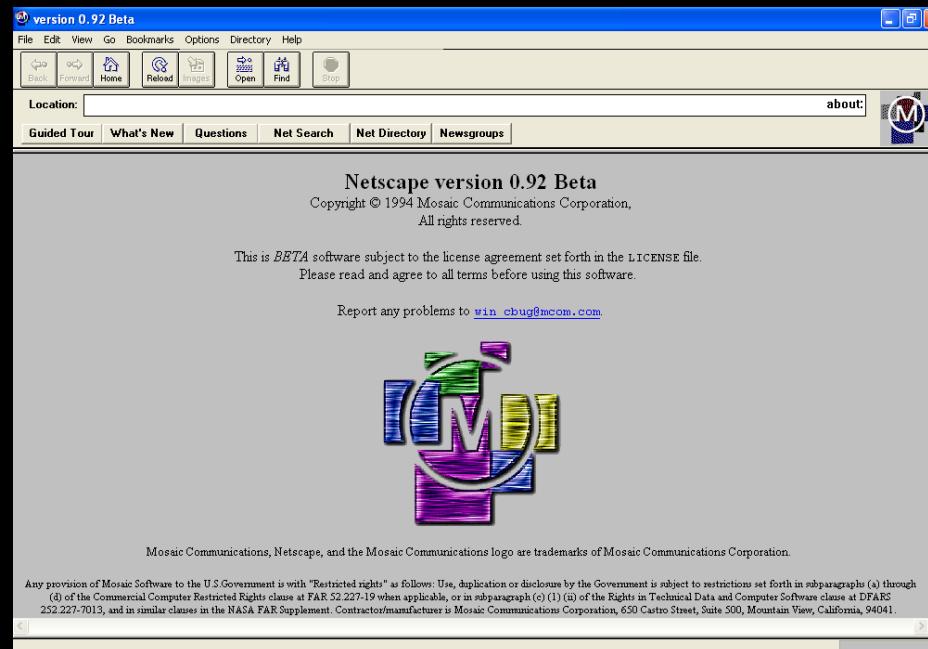
TRUST

The need for trust is removed from exchange of value

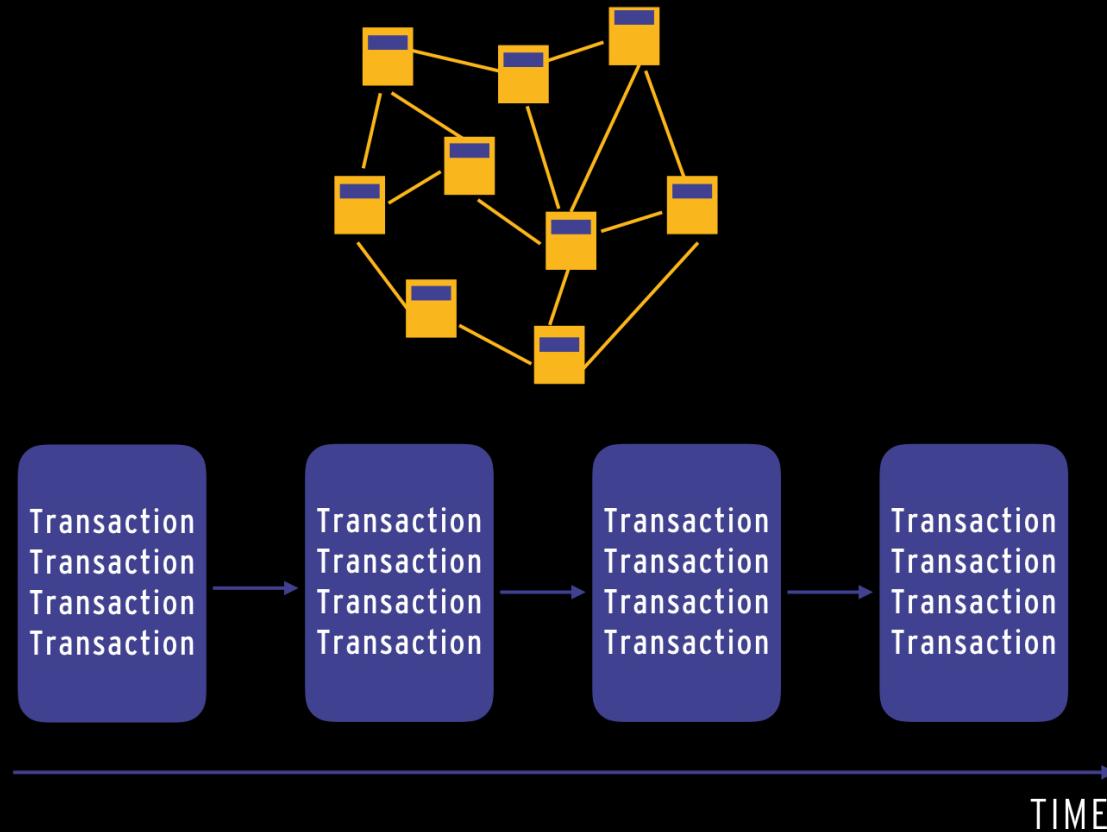
WORLD WIDE WEB OF VALUE

EARLY STAGES

"Like the Internet in the middle of the 1990-ies"



BLOCKCHAIN



BLOCKCHAIN TECHNICAL

peer-to-peer network

decentralized read-only & append-only event database

extremely hard to change

updated via consensus

cryptographically secure

eventually consistent

permissionless

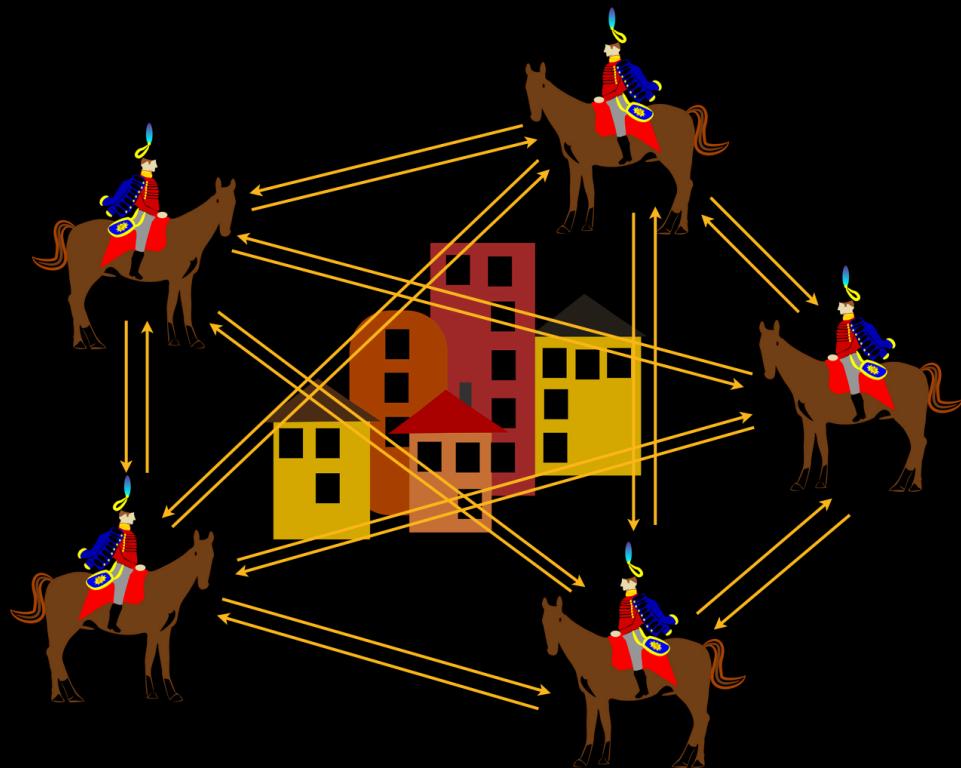
visibility and transparency

BYZANTINE GENERALS PROBLEM



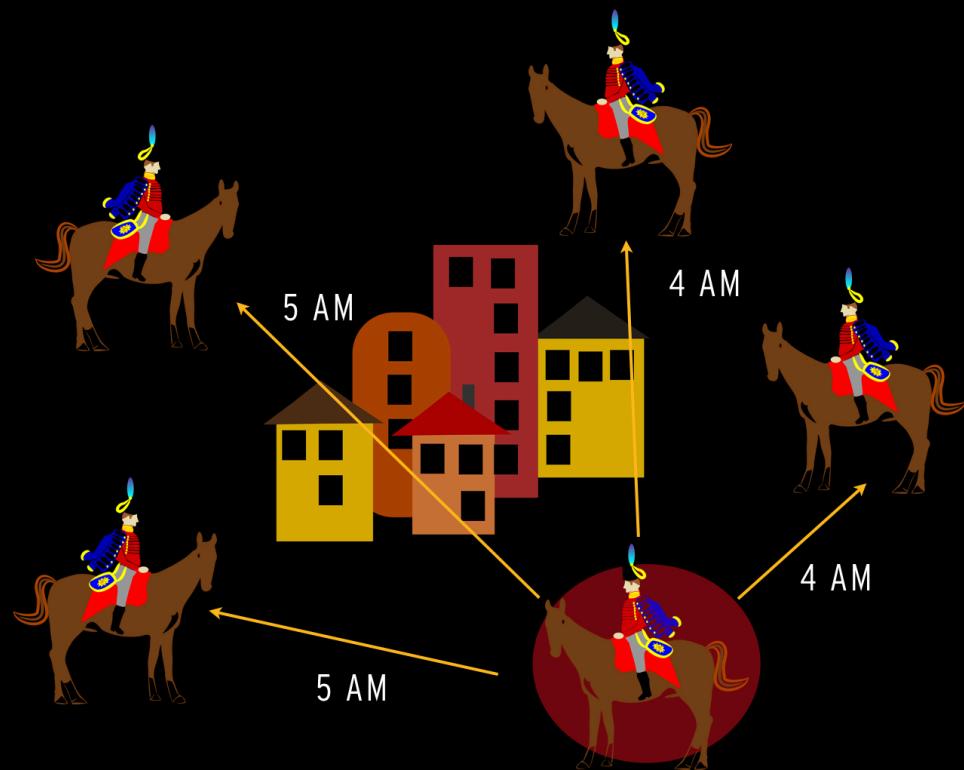
<https://bitcointalk.org.oldSiteFiles/byzantine.html>

BYZANTINE GENERALS PROBLEM



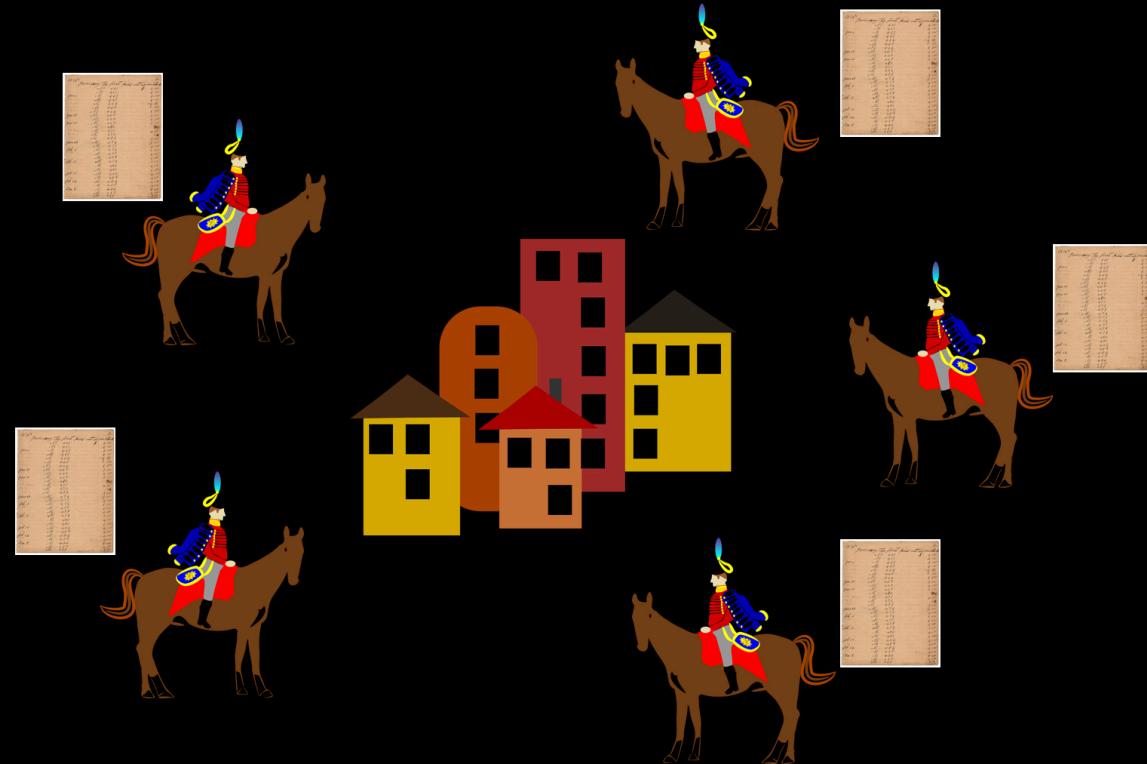
<https://bitcointalk.org.oldSiteFiles/byzantine.html>

BYZANTINE GENERALS PROBLEM



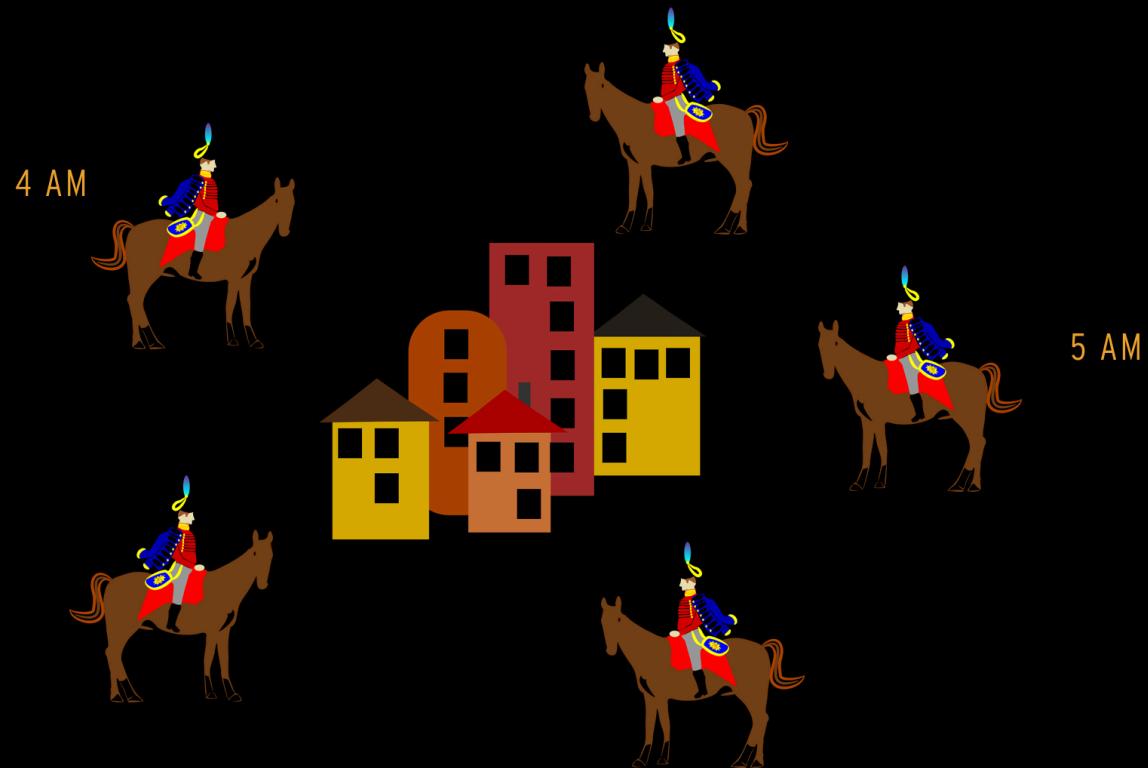
<https://bitcointalk.org.oldSiteFiles/byzantine.html>

BYZANTINE GENERALS PROBLEM



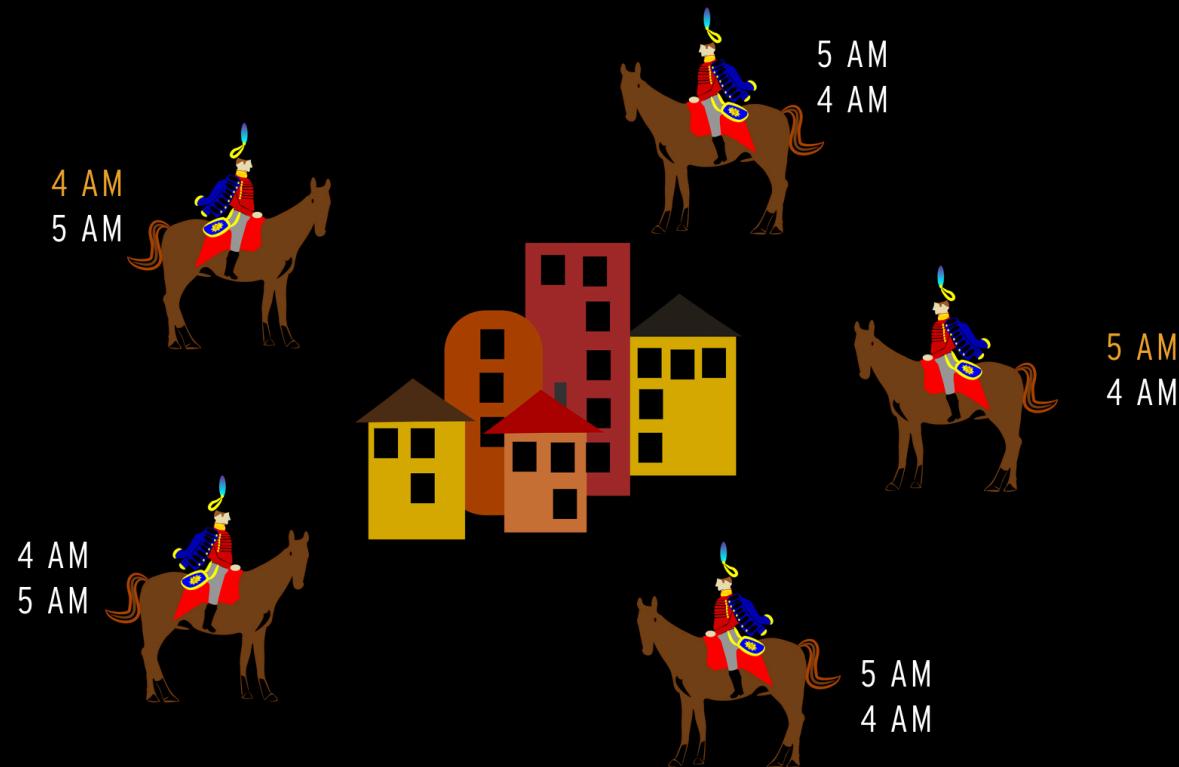
<https://bitcointalk.org.oldSiteFiles/byzantine.html>

BYZANTINE GENERALS PROBLEM



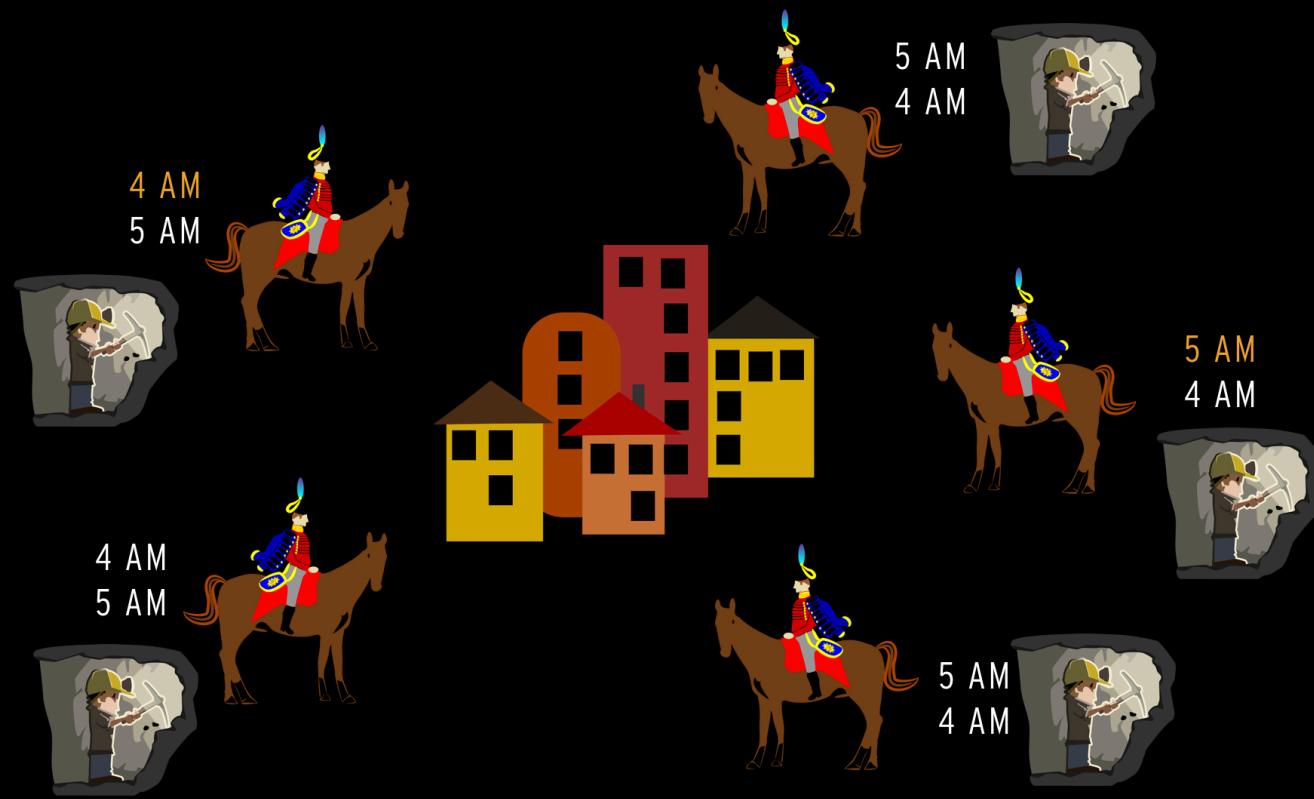
<https://bitcointalk.org.oldSiteFiles/byzantine.html>

BYZANTINE GENERALS PROBLEM



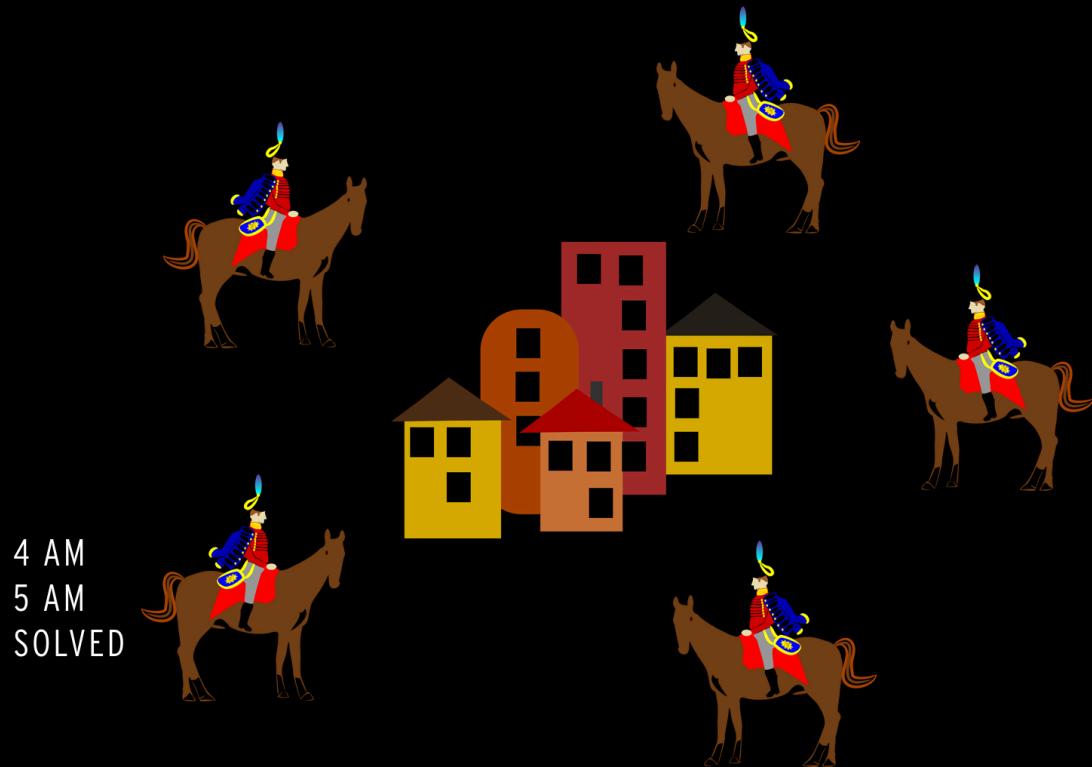
<https://bitcointalk.org.oldSiteFiles/byzantine.html>

BYZANTINE GENERALS PROBLEM



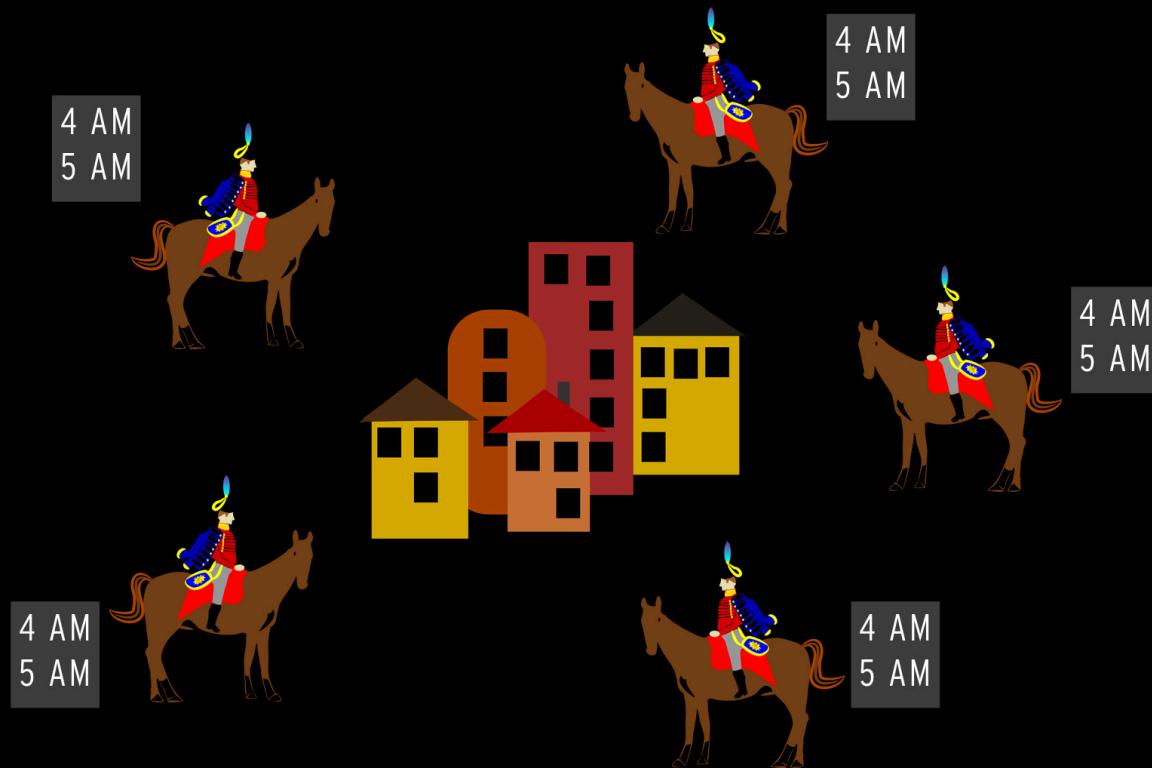
<https://bitcointalk.org.oldSiteFiles/byzantine.html>

BYZANTINE GENERALS PROBLEM



<https://bitcointalk.org.oldSiteFiles/byzantine.html>

BYZANTINE GENERALS PROBLEM



<https://bitcointalk.org.oldSiteFiles/byzantine.html>

PROOF OF WORK

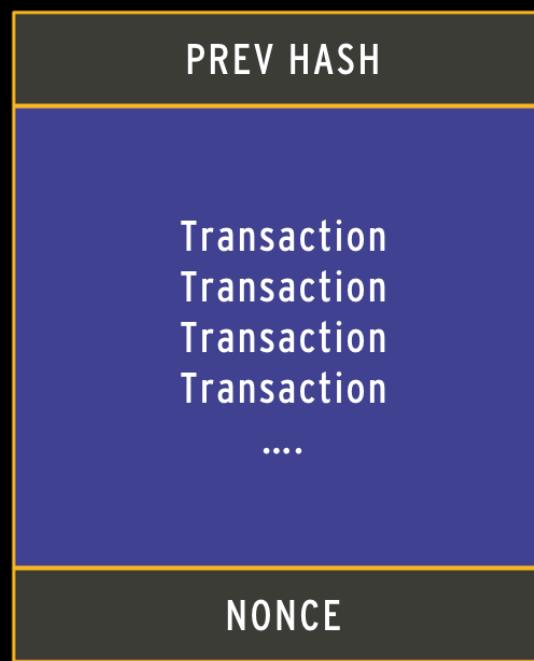
CRYPTOGRAPHIC HASH FUNCTION

```
> SHA256('cadec2017')  
> 7c764c8edff10d307b464a3625b085bd89e6c4f2e4ff9abe137d45e75f  
ff1a3  
> SHA256('cadec2018')  
> c8d681ef9574b02a945fba49f25c62ff12e2966291e2d5a1fc1fea7ab06  
caac1
```

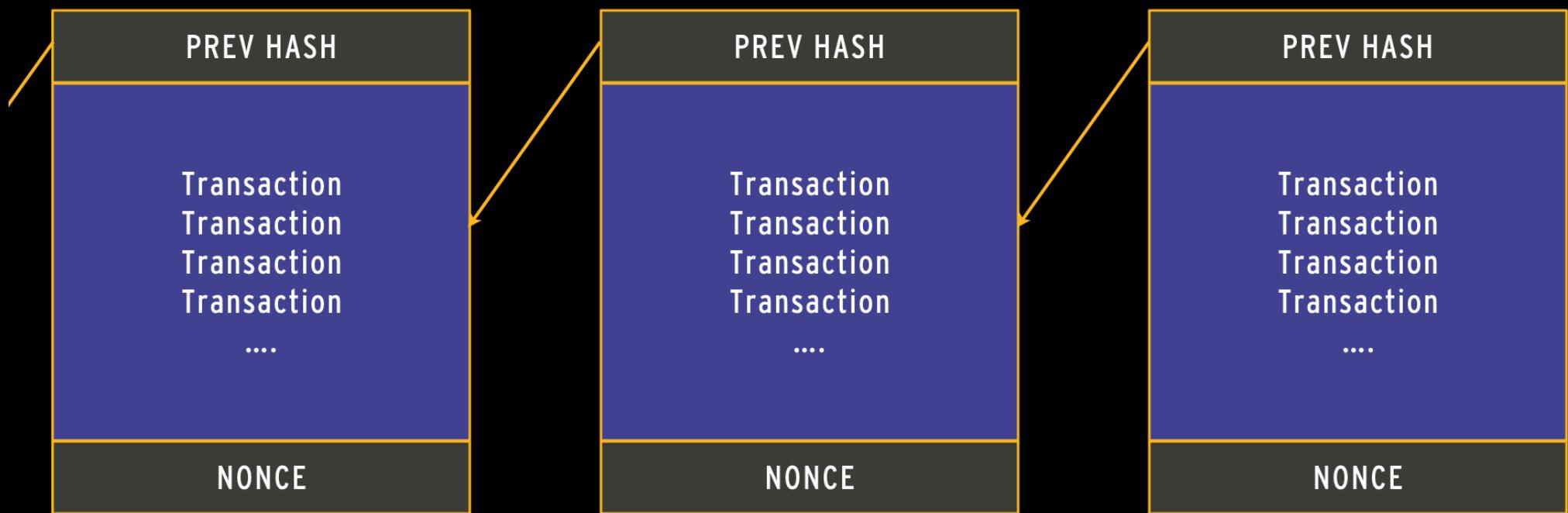
CRYPTOGRAPHIC NONCE

An arbitrary number that can only be used once.

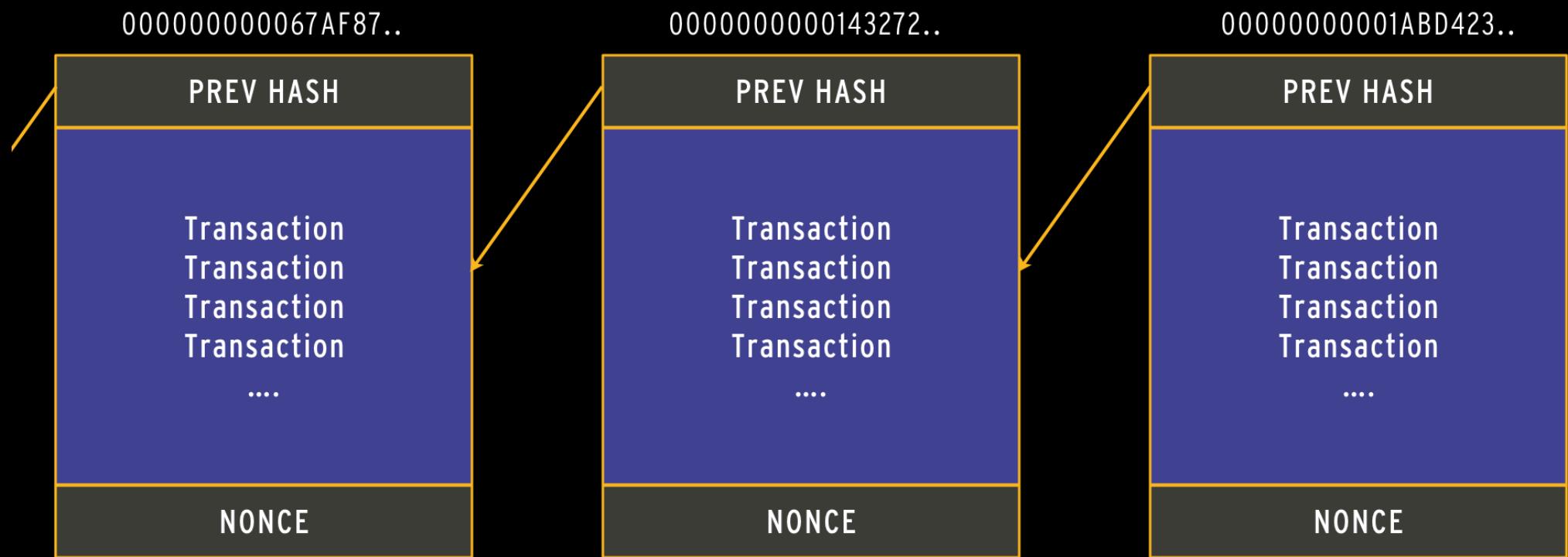
BLOCK



CHAIN OF BLOCKS



CHAIN OF BLOCKS



THE NODE THAT FINDS THE SOLUTION:

- Gets a reward (12.5B)
- Gets transaction fees

HASH POWER

Bitcoin	16 EH/s	16.000.000.000.000.000.000 H/s	38 TWh/år
	24 EH/s	24.000.000.000.000.000.000 H/s	53 TWh/år
Ethereum	150 TH/s	150.000.000.000.000 H/s	11 TWh/år
	250 TH/s	250.000.000.000.000 H/s	15 TWh/år

CHALLENGES

SCALABILITY & PRIVACY

SCALABILITY

Bitcoin	4 transactions/s
Ethereum	20 transactions/s
Paypal	400 transactions/s
Visa	2000 transactions/s

SMART CONTRACTS

"A smart contract is a secure and unstoppable computer program representing an agreement that is automatically executable and enforceable"

- Imran Bashir



Vitalik Buterin



"I happily played World of Warcraft during 2007-2010, but one day Blizzard removed the damage component from my beloved warlock's Siphon Life spell. I cried myself to sleep, and on that day I realized what horrors centralized services can bring. I soon decided to quit."

https://about.me/vitalik_buterin

ETHEREUM & SMART CONTRACTS IN SOLIDITY

APPLICATION PLATFORM

Ethereum is a decentralized platform that runs smart contracts (Dapps): Ethereum Virtual Machine (EVM) applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference.

Operations in the EVM have gas cost. Gas itself also has a gas price measured in terms of Ether. Every transaction specifies the gas price it is willing to pay in ether for each unit of gas, allowing the market to decide the relationship between the price of Ether and the cost of computing operations (as measured in gas).

Network number 1

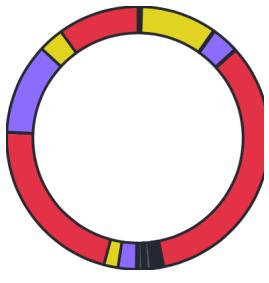
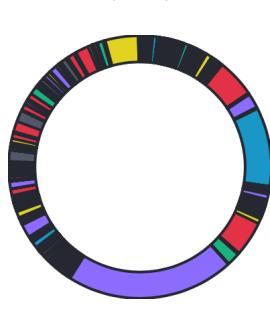
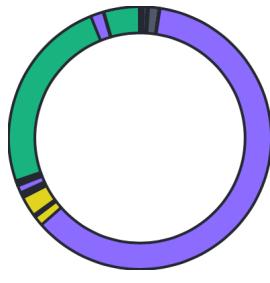
Last updated a few seconds ago

Search... 

ETHEREUM

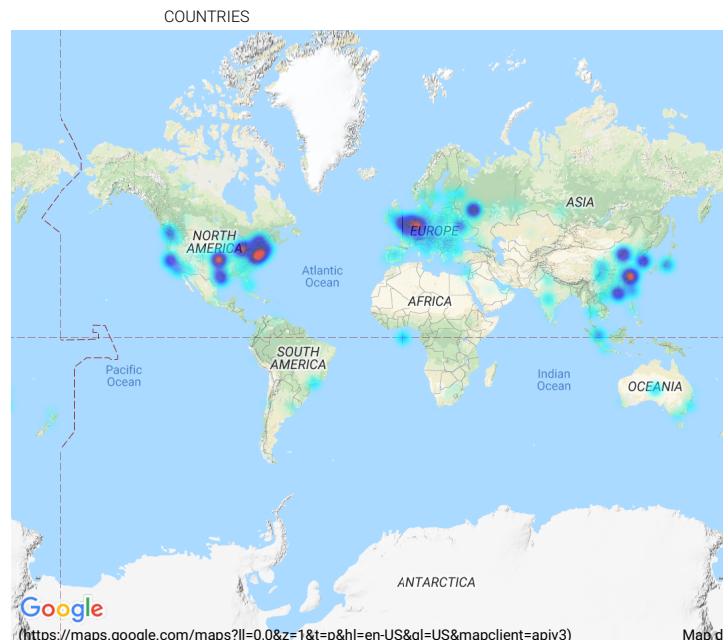
Mainnet (/network/1)

Testnet (/network/2)



Like what you see? Support the node explorer! (/donate)

Total	17672 (100%)
United States	5780 (32.71%)
China	2293 (12.98%)
Russian Federation	1009 (5.71%)
Germany	969 (5.48%)
Canada	898 (5.08%)
United Kingdom	608 (3.44%)
Korea, Republic of	462 (2.61%)
Netherlands	424 (2.40%)
France	409 (2.31%)
Ukraine	333 (1.88%)



[View all \(1/nodes\)](#)

EVOLUTION

LAST 24H

+2.5% 

LAST WEEK

-3.7% 

LAST MONTH

-15.9% 





Estimates over last 1,500 blocks - Last update: Block 5289133

Change Currency ▾

►Std Cost for Transfer

\$0.022

Gas Price SafeLow (Gwei)

2

Gas Price Std (Gwei)

2

Median Wait (s)

45

SafeLow Cost for Transfer

\$0.022

Median Wait (blocks)

3**Gas-Time-Price Estimator:** For transactions sent at block: 5289159

Adjust confirmation time

Avg Time (min)

5.29

Gas Used*

21000

95% Time (min)

13.23

Avg Time (blocks)

21.16

Gas Price (Gwei)*

2

95% Time (blocks)

52.9

Tx Fee (Fiat)

\$0.022

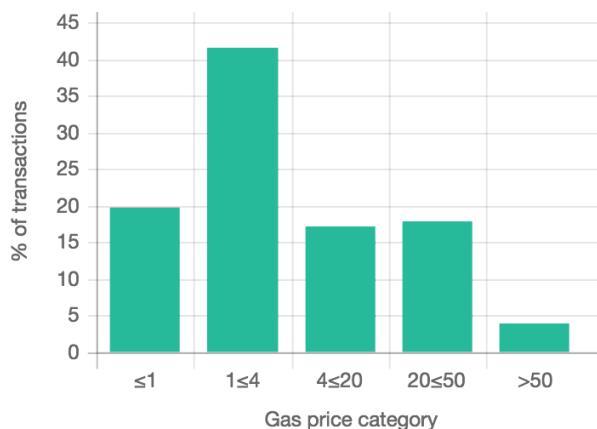
Tx Fee (ETH)

0.00004

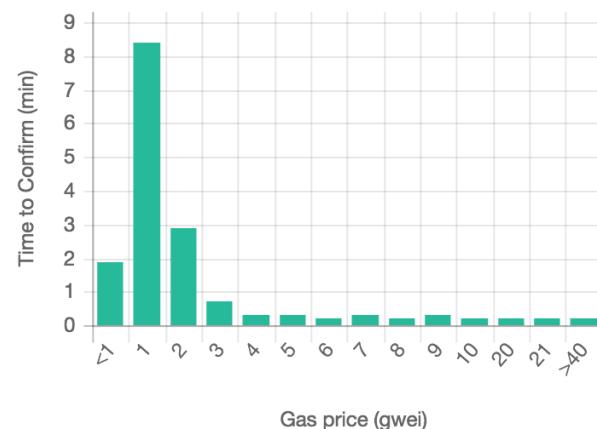
Real Time Gas Use: % Block Limit (last 10)

Last Block: 5289159

Transaction Count by Gas Price



Confirmation Time by Gas Price



Recommended Gas Prices

(based on current network conditions)

Speed	Gas Price (gwei)
SafeLow (<30m)	2
Standard (<5m)	2
Fast (<2m)	14

Note: Estimates not valid when multiple transactions are batched from the same address or for transactions sent to addresses with many (e.g. > 100) pending transactions

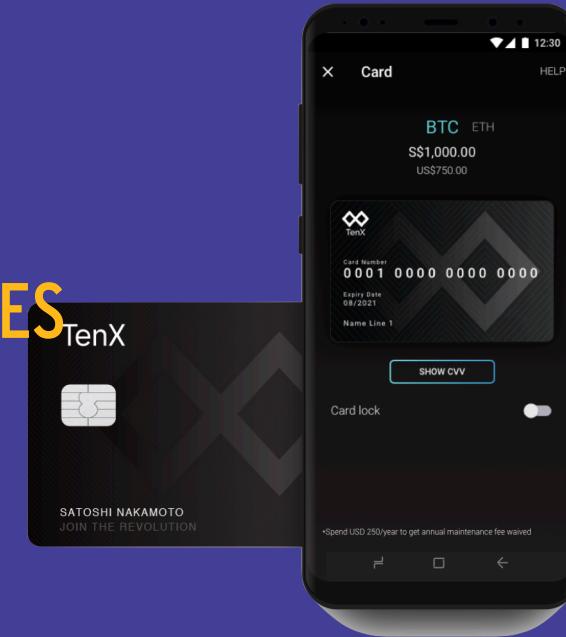
Top 10 Miners by Blocks Mined: Support for user transactions

Miner	Lowest gas price (gwei)	Weighted avg gas price (gwei)	% of total blocks
0x5a0b54d5dc17e0adc383d2db43b0a0d3e029c4c	0.1	7	14
miningpoolhub	0.2	11	11
f2pool	0.2	16	19
0x180ba8f73897c0cb26d76265fc7868cf936e617	1	8	1
Nanopool	1	12	15
Dwarfpool	1	14	3
Ethermine	1	18	27
Ethpool	4	24	1
0xf3b9d2c81f2b24b0fa0acaaa865b7d9ced5fc2fb	12	40	2
0x84990f5d2e09f56cabdabf6409ad31bdd8363b50	20	43	3

Tokenization
Initial Coin Offering
Identity Applications
Gaming
Payments
Charity
Voting
...



APP EXAMPLES



SelfKey Identity Network



Blockchain for good

SOLIDITY LANGUAGE

CONTRACTS, STATE VARIABLES, FUNCTIONS, MODIFIERS & EVENTS

CONTRACTS

```
// Interface
contract ERC20Interface {
    function totalSupply() public view returns (uint);
    function balanceOf(address tokenOwner) public view returns (uint balance);
    //...
}

// Contract
contract MyToken is AbstractToken, ERC20Interface {
    // Constructor
    function MyToken(uint256 initialSupply) AbstractToken(initialSupply, 'My Token', 'MT') public {}
    // ...
}
```

STATE VARIABLES

```
contract Ballot {  
    // Primitive Type  
    uint256 numVoters;  
  
    // Enum Type  
    enum State { Open, Closed, Delegated }  
  
    // Struct Type  
    struct Voter {  
        uint weight;  
        State state;  
        address delegate;  
        uint vote;  
    }  
  
    // Mapping Type  
    mapping(address => Voter) voters;
```

FUNCTIONS

```
contract MyToken {  
    mapping(address => uint256) balances;  
    // ...  
    function balance() public view returns(uint256 amount) {  
        return balances[msg.sender];  
    }  
}
```

MODIFIERS

```
contract Purchase {
    address public seller;

    // Modifier
    modifier onlySeller() {
        require(msg.sender == seller);
        _; // inlines function code here
    }

    // Modifier usage
    function abort() onlySeller {
        // ...
    }
}
```

EVENTS

```
contract SimpleAuction {
    // Event
    event HighestBidIncreased(address bidder, uint256 amount);

    function bid() payable {
        // ...
        // Emits event
        HighestBidIncreased(msg.sender, msg.value);
    }
}
```

THE BEER EXPERIENCE APP

Demonstrating development tools and APIs

CONTRACT DEVELOPMENT TOOLS

Remix

Ethereum Solidity IDE and tools for the web.

testrpc/ganache-cli

A node (npm) based Ethereum client for testing and development.

Truffle

Truffle is a node (npm) based development environment, testing framework and asset pipeline.

Geth

Command line interface for running a full ethereum node implemented in Go.

Mist/Ethereum Wallet

To browse and use accounts and Dapps.

```
pragma solidity ^0.4.18;

// Simple BeerCoin (CADEC)
contract BeerCoin {
    string public constant name = "Beer Experience Token";
    string public constant symbol = "BEET";

    uint256 price;
    address owner;
    mapping (address => uint256) balances;
    mapping (address => bool) banned;

    event Buy(address from, uint256 units);
    event Intoxicated(address target, bool on);

    modifier ownerOnly {
        require(msg.sender == owner);
        ;
    }

    function BeerCoin() public {
        owner = msg.sender;
        price = 10;
    }

    function setPrice(uint256 _price) ownerOnly public {
        price = _price;
    }

    function getPrice() public constant returns(uint256 beets) {
        return price;
    }

    // Receive ether and change to beets
    function() public payable {
        uint256 amount = (msg.value / 1000000000000000000);
        require(amount > 0);
        balances[msg.sender] += amount;
    }
}
```

APP. DEVELOPMENT TOOLS

- MetaMask Browser plugin to run Ethereum Dapps right in your browser without running a full Ethereum node.
- Web3 Ethereum compatible JavaScript API (also supporting Python, Haskell, Java and Scala).

```
// initialize web3 contract (injected by metamask)
web3 = (typeof web3 !== 'undefined') ? new Web3(web3.currentProvider)
  : new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

ctrl = new Controller(new BeerCoinWrapper(web3));
ctrl.listenForEvents(localStorage.getItem('fromBlock'));
```

```
// Application/contract binary interface and contract address
BeerCoinWrapper.config = {
  abi: [...],
  address: "0x3391d32023d97427ef2a5d48b97ce422bca29bba"
};

// beercoin contract wrapper
function BeerCoinWrapper(web3) {
  // the actual contract
  const contract = web3.eth.contract(BeerCoinWrapper.config.abi).at(BeerCoinWrapper.config.address);

  // adds price to data (promise)
  this.addPrice = (data) => {
    return new Promise((resolve, reject) => {
      contract.getPrice((error, price) => {
        if (error) {
          reject(error);
        } else {
          data.price = price;
          resolve(data);
        }
      });
    });
  }
}
```

FINALLY, SOME STUFF TO CONSIDER

Make the business model comprehensible (UX)

One-off releases

Contract execution and transaction model

Immaturity, pace of change

Public vs. private (Ethereum, Hyperledger, ...) networks

That's all, Thanks!