MEETUP #1

ETHEREUM SMART CONTRACTS 101

SUMMARY

- About us
- Cryptocurrencies - Some Context
- Bitcoin
- Ethereum
  - Applications
  - Smart Contracts and Gas
- Create your own contract
- ERC20 tokens
- Tools
- Recap

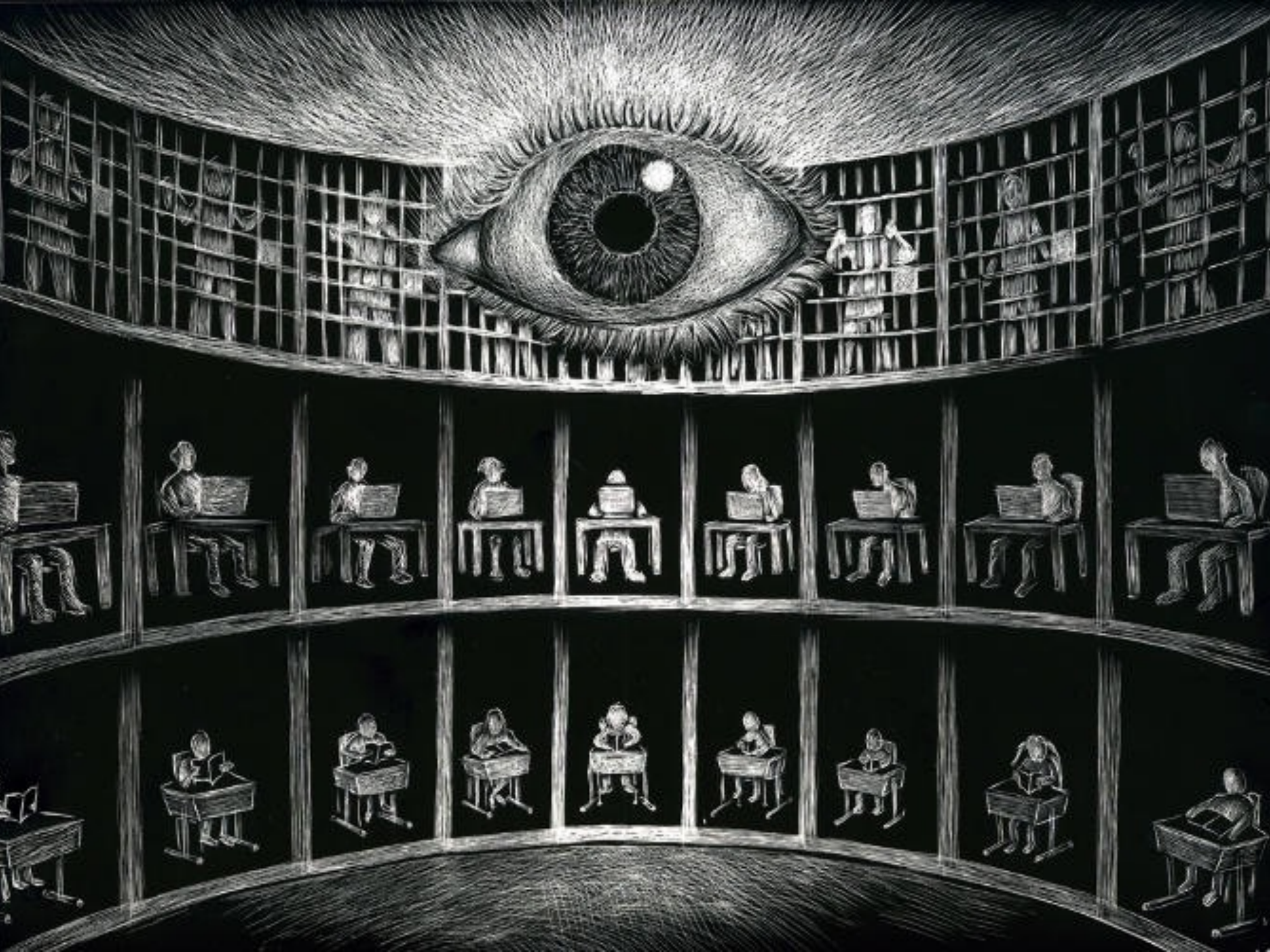

Blockchain
Dev

BLOCKCHAINDEV MEETUPS

- Focused on blockchain **technologies**
- Knowledge-sharing and learning
- Beginner friendly

SOME CONTEXT

CRYPTOCURRENCIES



CRYPTOCURRENCIES - SOME CONTEXT

- Cypherpunk Movement (started in the 80s)
- Activists, programmers, hackers, researchers, mathematicians, ...
- Advocated the use of strong cryptography and privacy enhancing technologies as a route to **social and political change**.

CRYPTOCURRENCIES - SOME CONTEXT

- Financial Crisis of 2007-2008
- During the last quarter of 2008, central banks purchased US$2.5 trillion of government debt and troubled private assets from banks

A P2P ELECTRONIC CASH SYSTEM

BITCOIN

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

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

October, 2008

BITCOIN

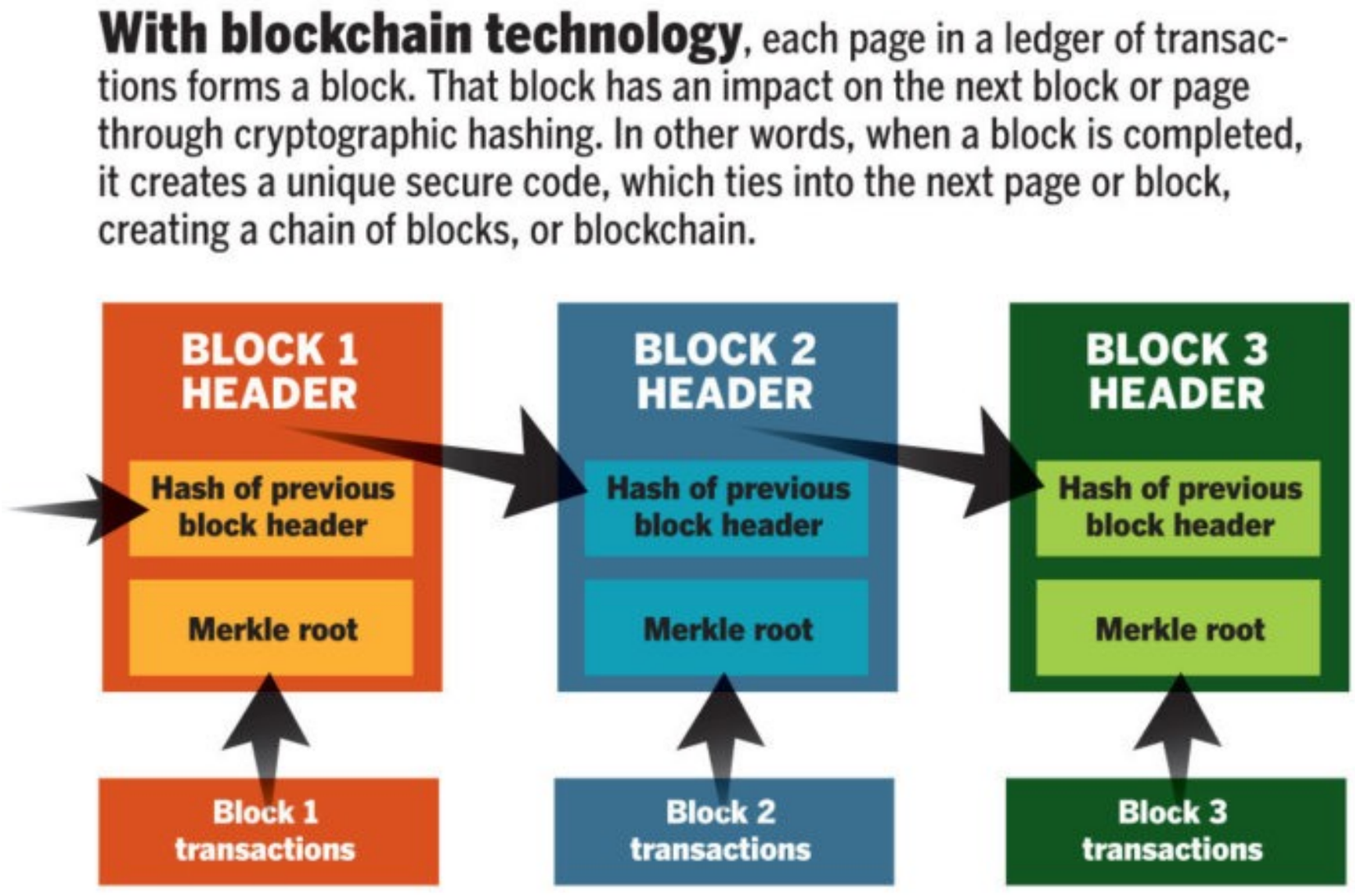
- Decentralized P2P
- No need for banks
- Limited supply
- Strong protection of your digital assets

BITCOIN

- Sign transactions with a digital signature
- Transactions are stored in a blockchain
- A consensus algorithm run on the network to verify the validity of the transactions (proof-of-work)
- Transactions are non-reversible

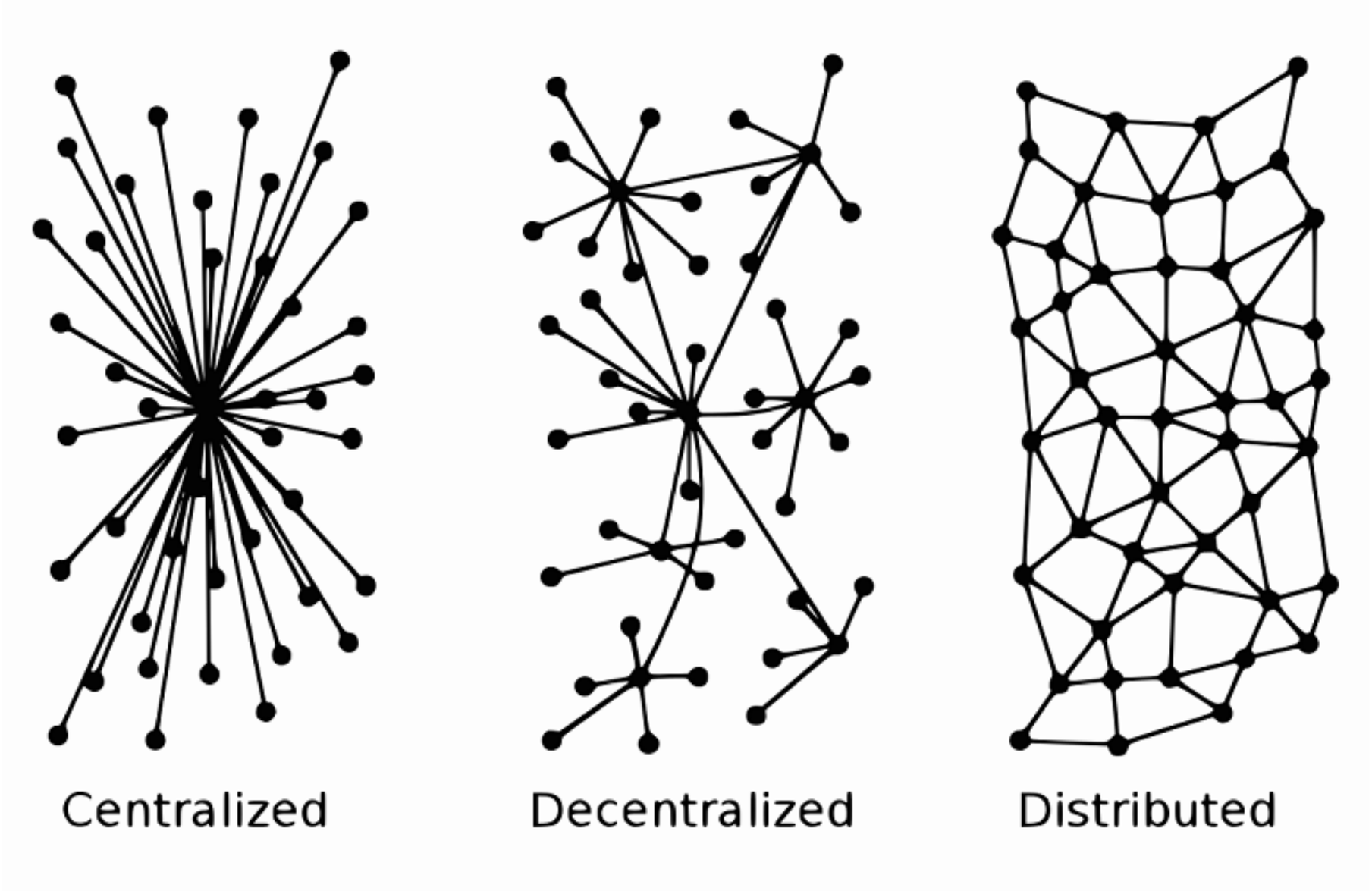
BLOCKCHAIN

- Database
- Distributed
- Decentralized
- Immutable (write once, append only)
- Public (or private)
- Stores transactions



A blockchain (https://www.computerworld.com/article/3191077/security/what-is-blockchain-the-most-disruptive-tech-in-decades.html)

DECENTRALIZED AND DISTRIBUTED NETWORK



NETWORK

- Executes code
- Validates transactions (consensus)
- Mines new blocks
- Stores the blockchain
- Application infrastructure / supercomputer

DECENTRALIZED
APPLICATION PLATFORM

ETHEREUM

A Next-Generation Smart Contract and Decentralized Application Platform

In the last few months, there has been a great amount of interest into the area of using Bitcoin-like blockchains - the mechanism that allows for the entire world to agree on the state of a public ownership database - for more than just money. Commonly cited applications include using on-blockchain digital assets to represent custom currencies and financial instruments ("colored coins"), "smart property" devices such as cars which track a colored coin on a blockchain to determine their present legitimate owner, as well as more advanced applications such as decentralized exchange, financial derivatives, peer-to-peer gambling and on-blockchain identity and reputation systems. Perhaps the most ambitious of all cited applications is the concept of autonomous agents or decentralized autonomous organizations (DAOs) - autonomous entities that operate on the blockchain without any central control whatsoever, eschewing all dependence on legal contracts and organizational bylaws in favor of having resources and funds autonomously managed by a self-enforcing smart contract on a cryptographic blockchain.

Ethereum White Paper (2013)

ETHEREUM SMART CONTRACTS 101

Vitalik Buterin

WHAT IS ETHEREUM ANYWAY?

- A cryptocurrency (ETH), but more!
- A platform for developing and running decentralized applications

WHAT IS ETHEREUM ANYWAY?

- Virtual Machine (EVM)
- A programming language
- A blockchain for state transitions
- A “world computer”

ETHEREUM: A WORLD COMPUTER

- Arbitrary code is executed on the network
- You can run your code by creating a Smart Contract
- Contracts alter the state of the blockchain

APPLICATIONS

- Financial
- Semi-financial
- Non-financial

FINANCIAL APPLICATIONS

- Payments
- Sub-currencies and tokens
- Initial Coin Offering (ICOs)
- Smart Property
- Banking and Trading (derivatives, hedging contracts, currency, futures, digital assets)

SEMI-FINANCIAL APPLICATIONS

- Bounties
- Coupons, any type of miles
- Gambling
- Marketplaces and Monetization

NON-FINANCIAL APPLICATIONS

- Reputation and identity
- Decentralized Autonomous Organizations (DAOs)
- Voting
- Games (cryptokitties)
- Decentralized file storage

ALSO, MEMES . . .

HOW TO RUN YOUR CODE ON THE EVM

SMART CONTRACTS

SMART CONTRACTS

- Computerized transaction protocol that executes the terms of a contract
- A contract is defined as a piece of code
- Self-executing contract
- You can create use it as a backend for your dapps

SMART CONTRACTS

- Your smart contract will contain instructions
- The code is also stored on the blockchain
- The execution of the code can create transactions and a new state of the blockchain

ETHEREUM AND CRYPTO-LAW: USES BLOCKCHAIN TO IMPLEMENT ARBITRARY SOCIAL CONTRACTS WITHOUT A CENTRAL SERVER

Dr. Gavin Wood

GAS

- Execution fee or cost
- Reading information is free
- Changing data/creating transactions costs gas
- You pay gas to the network

GAS

- Incentive for nodes to run your code and store your data
- Code must be efficient (time + space complexity) to optimize the costs
- Different instructions have different costs
- You can add more gas to your transactions to have a higher priority

EXAMPLES OF INSTRUCTIONS AND THEIR GAS PRICES

- Calculating a hash
- Multiplying two numbers
- See link

GETTING STARTED

CREATING A SMART CONTRACT

- Install Metamask and create a wallet
- Use the Rinkeby test network
- Get some fake ether @ rinkeby faucet
- Open remix and create your contract
- Deploy on (injected web3) and check etherscan.io!
- Also check web3.js

```
pragma solidity ^0.4.22;

contract HelloWorld {

    string message;

    function setName(string _message) public {
        message = _message;
    }

    function hello() public constant returns (string) {
        return (message);
    }
}
```

ERC20 TOKEN STANDARD

- Read the ERC20 Token Standard
- Implement the EIP20Interface to create your own token

```
pragma solidity ^0.4.21;

contract EIP20Interface {
    uint256 public totalSupply;

    function balanceOf(address _owner) public view returns (uint256 balance);

    function transfer(address _to, uint256 _value) public returns (bool success);

    function transferFrom(address _from, address _to, uint256 _value) public returns (bool
success);

    function approve(address _spender, uint256 _value) public returns (bool success);

    function allowance(address _owner, address _spender) public view returns (uint256
remaining);

    event Transfer(address indexed _from, address indexed _to, uint256 _value);
    event Approval(address indexed _owner, address indexed _spender, uint256 _value);
}
```

OTHER TOOLS YOU CAN USE

- GETH: command line interface + ethereum node
- Drizzle: front-end libs for dapps
- Truffle: development environment for ethereum
- Ganache: run your personal blockchain

RECAP

- Decentralization is good!
- You can create your own Smart contracts and dapps to run on the ethereum network
- Create your own tokens by implementing the EIP20 interface
- Use web3js on the frontend (ethereum js api)

KTHXBYE

@thdaraujo (twitter)

thd.araujo@gmail.com

aleph.uno (blog)

Thiago Araujo