Blockchain 101

Kamer Kaya
Sabancı Üniversitesi

What do people think about Bitcoin/Blockchain?

"[Blockchain] is the biggest opportunity set we can think of over the next decade." – Bob Greifeld, (former) NASDAQ CEO

"We'll all look back in 20 years and conclude that Bitcoin was as influential a platform for innovation as the Internet itself was." – Marc Andreessen, WWW Pioneer

"Bitcoin is the most important invention in the history of the world since the Internet." – Roger Ver, Bitcoin investor and evangelist

What do people think about Bitcoin/Blockchain?

Bitcoin is the mother of all scams (N. Roubini, 2018).

It may be a partial store of value because, unlike thousands of other what I call shitcoins, it cannot be so easily debased because there is at least an algorithm that decides how much the supply of bitcoin raises over time (N. Roubini, 2020).

The big revolution we're going to see in the next three years is going to be central bank digital currencies. They're going to be crowding out digital payment systems (N. Roubini, 2020).

NFT is The Mother of All Tulip-Mania! And it will crash similarly. Every piece of crappy or junk video or pseudo-art is put on NFTs these days. Soon enough even good art such as Piero Manzoni's "Merde d'Artiste" so that even cow manure will be turned into NFTs! (N. Roubini, 2022).

(https://news.bitcoin.com/dr-doom-nouriel-roubini-bitcoin-store-of-value/)

Blockchain: Definition

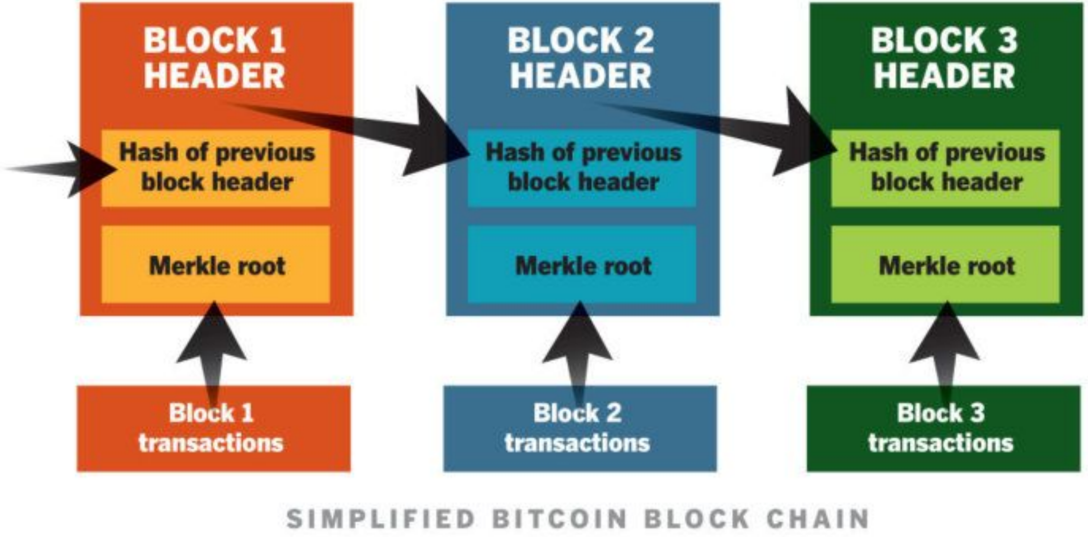
(Blockchain ≠ Bitcoin)

A Blockchain (originally Block Chain) is a

- Continuously growing list of records, called blocks, which are linked and secured using cryptography.
- Each block typically contains a cryptographic hash of the previous block, a timestamp and transaction data.
- By design, a Blockchain is inherently resistant to modification of the data.
- It is "an open, distributed ledger that can record transactions between multiple parties efficiently and in a verifiable and permanent way”.

(https://en.wikipedia.org/wiki/Blockchain.)

Blockchain



(https://www.cio.com/article/3055847/security/what-is-blockchain-and-how-does-it-work.html)

Blockchain/Bitcoin

version	02000000
previous block hash (reversed)	17975b97c18ed1f7e255adf297599b55 330edab87803c8170100000000000000
Merkle root (reversed)	8a97295a2747b4f1a0b3948df3990344 c0e19fa6b2b92b3a19c8e6badc141787
timestamp	358b0553
bits	535f0119
nonce	48750833
transaction count	63
coinbase transaction	
transaction	
...	

Block hash

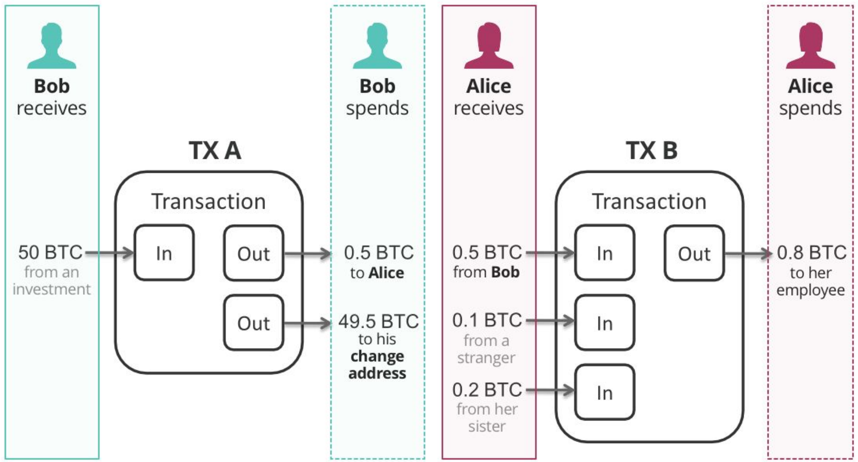
0000000000000000
e067a478024addfe
cdc93628978aa52d
91fabd42992982a50

(http://static.righto.com/images/bitcoin/block_diagram_ghash.png)

Blockchain/Bitcoin

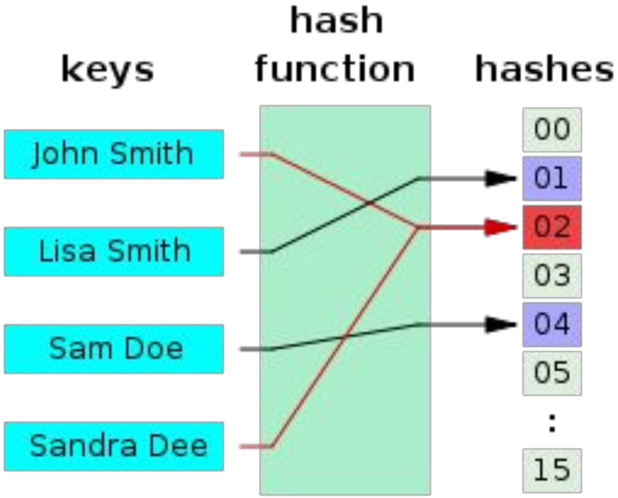
Blocks contain transactions; but what kind of info do we store in them?
Lets see some transactions within the Bitcoin blockchain:

https://live.blockcypher.com/btc/



The First Aid: Cryptographic Hash Function

A hash function is any function that maps an arbitrary size input to a fixed size output.



(https://en.wikipedia.org/wiki/Hash_function)

The First Aid: Cryptographic Hash Function

The ideal cryptographic hash function has four main properties:

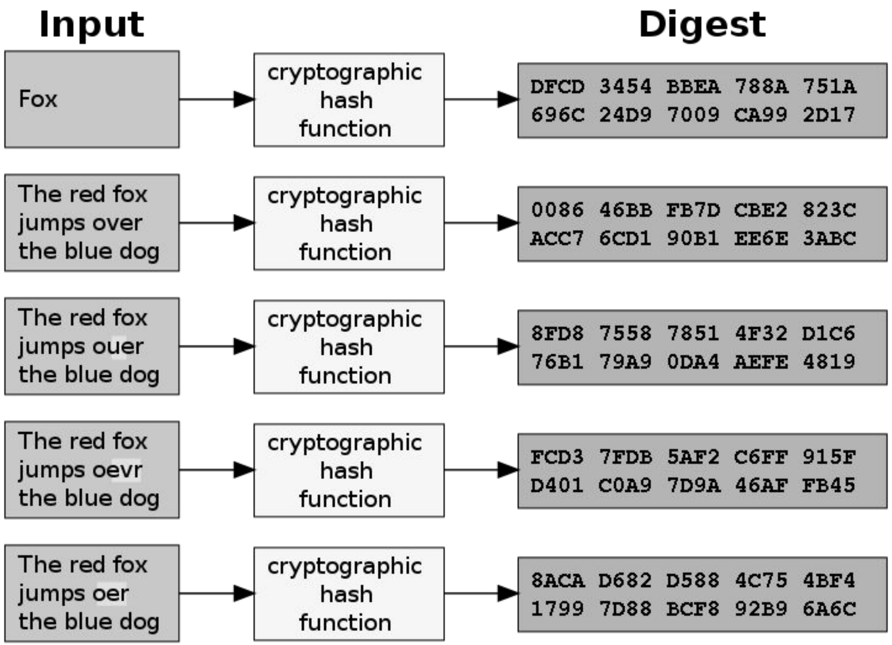
- it is quick to compute the hash value for any given message.
- it is infeasible to generate a message from its hash value except by trying all possible messages (pre-image resistance).
- It is unpredictable in the sense that a small change to a message should change the hash value so extensively that the new hash value appears uncorrelated with the old hash value.
- it is infeasible to find two different messages with the same hash value (collusion resistance).

The First Aid: Cryptographic Hash Function

The need for cryptographic hash functions are stated by Diffie and Hellman in 1976.

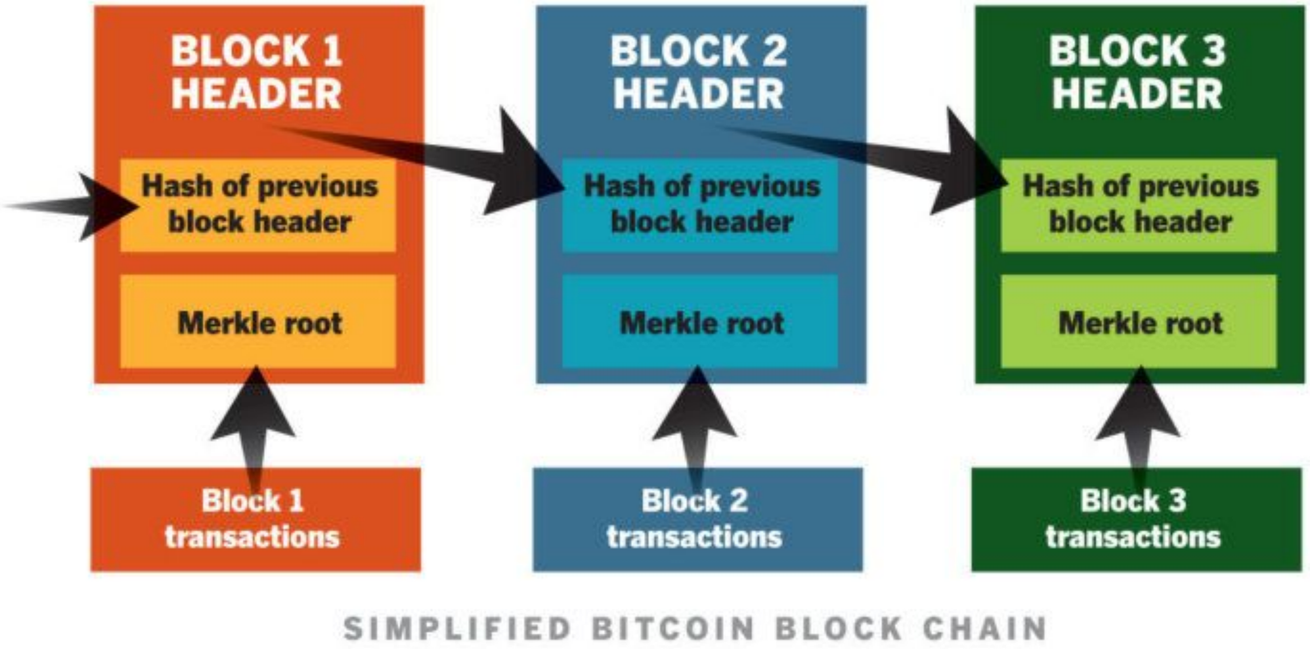
New directions in cryptography
https://dl.acm.org/citation.cfm?id=2269104
by W Diffie - 1976 - Cited by 17440 - Related articles
Sep 1, 2017 - Two kinds of contemporary developments in **cryptography**
applications of teleprocessing have given rise to a need for ...

The first designs and proposals are there since late 1970s.



(https://en.wikipedia.org/wiki/Cryptographic_hash_function)

Blockchain



(https://www.cio.com/article/3055847/security/what-is-blockchain-and-how-does-it-work.html)

Cryptocurrencies

Numbers Can Be a Better Form of Cash than Paper
dl.acm.org/citation.cfm?id=723552
by D Chaum - 1991 - Cited by 11 - Related articles
Numbers Can Be a Better Form of Cash than Paper, 1991 Article. Bibl
Citation Count: 0 · Downloads (cumulative): n/a · Downloads ...

- Why cryptocurrencies / digital cash / e-cash?
    - Freedom in payment, control and security, transparent information (except personal information).
- Why not conventional cash?
    - Inflation, inflation, inflation...
    - Counterfeitable, slow, costly, vulnerable to stealing/thieves, frauds, little privacy, higher transaction fees.

Bitcoin does not fully solve the problems above.

We have other types of cryptocurrencies and we are still looking for better solutions. And until now, Blockchain is the main arsenal we have.

Cryptocurrencies: Lessons from the Past

- When big players want to adopt - do not be paranoid.
  - Being smart must be hard... You must be smarter.
- Centralized solutions can easily be shut down by the governments.
  - Or by the centralized entity; DigiCash for instance needs a bank.
- Security breaches in digital currencies are rampant
  - For instance, fraud probably did accelerate Flooz's demise.
- Hackers are making fabulous money through digital cash.
  - If we include the loss from Mt. Gox's hacking with those from thefts that occurred between December 2017 - January 2018, the total loss would amount to more than $1 billion.
- Cryptocurrency can also be and are being exploited to commit fraud.
  - The problem is not only hacking but also exploitation of cryptocurrency for illicit purposes.

(https://medium.com/@ParolaAnalytics/bitcoin-etc-what-can-we-learn-from-flooz-and-beenz-73566f140932)

STEVEN LEVY SECURITY 02.01.93 12:00 PM

CRYPTO REBELS

Cypherpunks

"... like barbed wire made possible the fencing-off of vast ranches and farms, thus altering forever the concepts of land and property rights in the frontier West, so too will the seemingly minor discovery out of an arcane branch of mathematics come to be the wire clippers which dismantle the barbed wire around intellectual property.

Timothy C. May 1988 The Crypto Anarchist Manifesto)

"Governments of the Industrial World, you weary giants of flesh and steel, I come from Cyberspace, the new home of Mind. On behalf of the future, I ask you of the past to leave us alone. You are not welcome among us. You have no sovereignty where we gather." (John Perry Barlow A Declaration of the Independence of Cyberspace 1996

Cypherpunks: The Road to Bitcoin

(https://www.delphitools.info/DWSH)

Cypherpunks were in a quest for the currency of the cyberspace. The problem is that trust among the anonymous peers is harder to obtain.

Hence, it is harder to collaboratively form a consensus on something:

- Everyone can join and leave.
- They may be malicious or not.

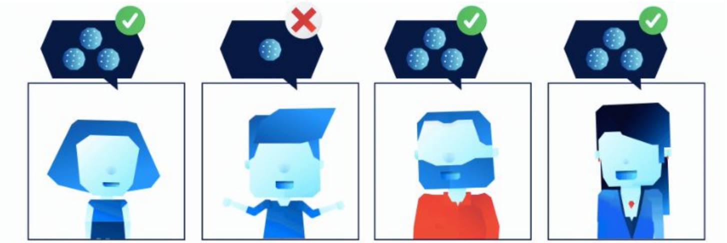
Unfortunately, a distributed network is the crypto-anarchy dream of cypherpunks for cyberspace, digital cash and other applications.

Untraceable electronic mail, return addresses, and digital pseudonyms
https://dl.acm.org/citation.cfm?id=358563
by DL Chaum - 1981 - Cited by 5415 - Related articles
Untraceable electronic mail, return addresses, and digital pseudonyms, Published by ACM 1981
Article. Bibliometrics Data Bibliometrics. · Citation Count: 1,017

Cypherpunks: The Road to Bitcoin

- In a distributed identity management system, you can choose whatever name you want without informing anyone in the system.
- The notion of pseudonyms, untraceable addresses are around since 80s by Chaum. They have also been used in pre-Bitcoin proposals.
- The problem is we do not know which transaction must be forwarded to which machine.
- In Bitcoin, the nodes store all the transactions, hence, using pseudonyms is already free. There are 2^160 (around 1.46e+48) possible addresses.

The Road to Bitcoin



One of the main problems for a decentralized system is consistency. Inconsistencies/forks can cause due to asynchronous actions/network latencies or malicious behaviour.

The solution to invalidate these inconsistencies - consensus algorithms - form the most crucial part in Bitcoin and in all other cryptocurrencies.

In Bitcoin, an inconsistency happens when some nodes will think block A is the latest block, while other nodes will think it is block B.

The Road to Bitcoin

The consensus problem has been studied in different names - state machine replication is a general technique to obtain a fault tolerant system. The faults can be in any form such as stop/fail.

Byzantine-fault tolerance is a related line of study where the faults are deviations from the protocol.

Time, clocks, and the ordering of events in a ...
https://dl.acm.org/citation.cfm?id=359563
by L Lamport - 1978 - Cited by 11163 - Related articles
Sep 1, 2017 - The concept of one **event** happening before anothe
and is shown to define a partial **ordering** of the **events**.

[PDF] The Part-Time Parliament - Leslie Lamport
https://lamport.azurewebsites.net/pubs/lamport-paxos.pdf
by L Lamport - Cited by 2560 - Related articles
spite the peripatetic propensity of its part-time legislators. The legi
with **a part-time parliament** bears a remarkable corre-.

[PDF] The Byzantine Generals Problem - EECS a
https://people.eecs.berkeley.edu/~luca/cs174/byzantine.pdf
by L LAMPORT - 1982 - Cited by 5342 - Related articles
IC1 follows from IC2 if the commander is loyal, so we need only
commander is a traitor. As we saw from the scenario of Figures i
makes **the Byzantine Generals Problem** so difficult.

Cypherpunks: The Road to Bitcoin


PROOFS OF WORK AND BREAD PUDDING PROTOCOLS
(EXTENDED ABSTRACT)

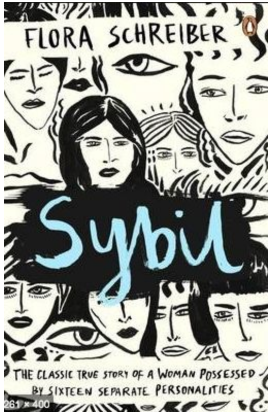
Markus Jakobsson
Information Sciences Research Center, Bell Labs, Murray Hill, New Jersey 07974
www.bell-labs.com/user/markusj

Ari Juels
RSA Laboratories, 20 Crosby Drive, Bedford, MA 01730
ari@rsa.com

Obtaining a consensus is not easy: A Sybil attack (or pseudo-spoofing as previously coined by L. Detweiler) destabilizes a system by forging identities in a peer-to-peer network.

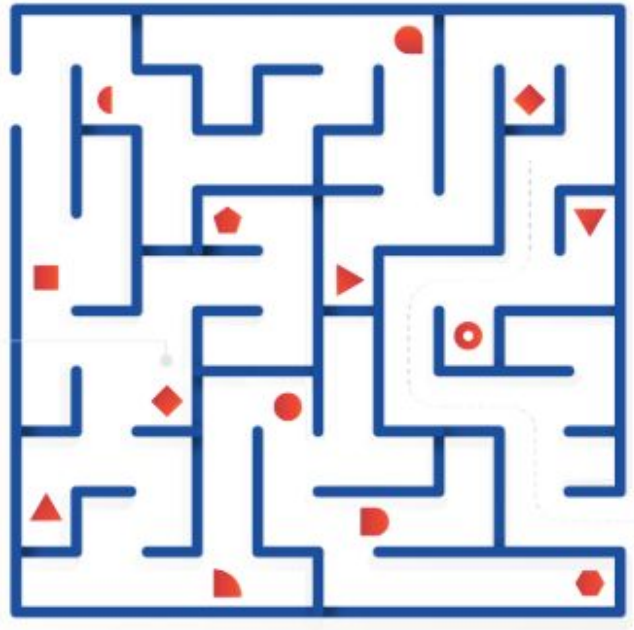
The Sybil attack was formalized in 2002 by John Douceur, who turned to a cryptographic construction called proof-of-work to mitigate it.

The term proof-of-work is coined by Markus Jacobsson and Ari Juels (1999).



(https://en.wikipedia.org/wiki/Sybil_attack)

Cypherpunks: The Road to Bitcoin



Cypherpunks: The Road to Bitcoin

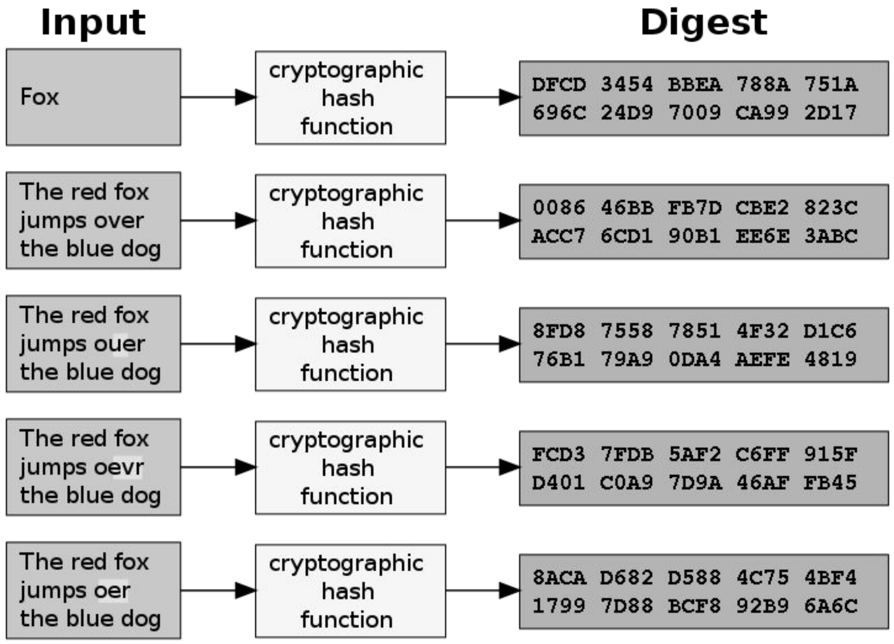
The PoW concept (under the name pricing function) was proposed by Cynthia Dwork and Moni Naor in 1992 to make the e-mail spammers slower.

Back simplified the algorithm.

Pricing via Processing or Combatting Junk Mail
https://dl.acm.org/citation.cfm?id=705669
by C Dwork - 1992 - Cited by 1014 - Related articles
We present a computational technique for **combatting junk mail** i
a shared resource in general. The main idea is to require a ...

- **Index:** A prime p of length depending on the difference parameter; a reasonable length would be 1024 bits.
- **Definition of f_p:** The domain of f_p is Z_p. $f_p(x) = \sqrt{x} \bmod p$.
- **Verification:** Given x, y, check that $y^2 \equiv x \bmod p$.

Reminder: Cryptographic Hash Function



(https://en.wikipedia.org/wiki/Cryptographic_hash_function)

The Road to Bitcoin

How to time-stamp a digital document | Spı
https://link.springer.com/article/10.1007/BF00196791
by S Haber - 1991 - Cited by 902 - Related articles
The problem is to **time-stamp** the data, not the medium. W
procedures for **digital** time-stamping of such documents so

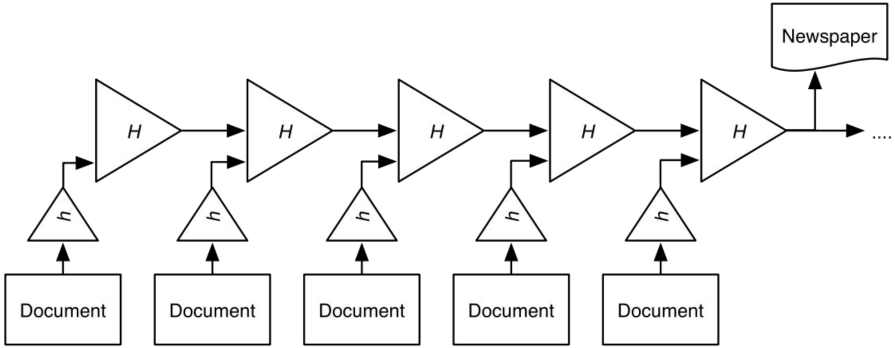
Linked timestamping creates time-stamp tokens dependent on each other.

Later modification of the issued time-stamps would invalidate this structure.

The temporal order is protected by this data structure, making backdating of the issued time-stamps impossible, even by the issuing server itself.

Haber and Stornetta (and Bader in one work) proposed document timestamping and also various improvements (1991-1997).

Independently, Benaloh and DeMare proposed the same set of optimizations in 1991.

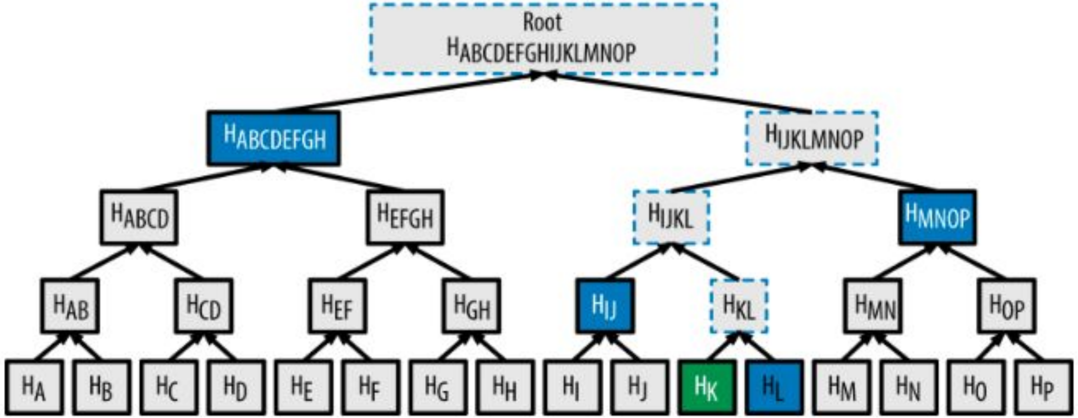


The Road to Bitcoin


A certified digital signature
RC Merkle - Conference on the Theory and
A practical digital signature system based o
secure as the conventional encryption funct
systems are available it can be implemente
required for certification of an untested syst
Cited by 1799 Related articles

Two main optimizations to linked timestamping are

- Store multiple documents in a single block instead of one.
- Use a Merkle tree to summarize (and efficiently verify) all the data and hashes inside a block.



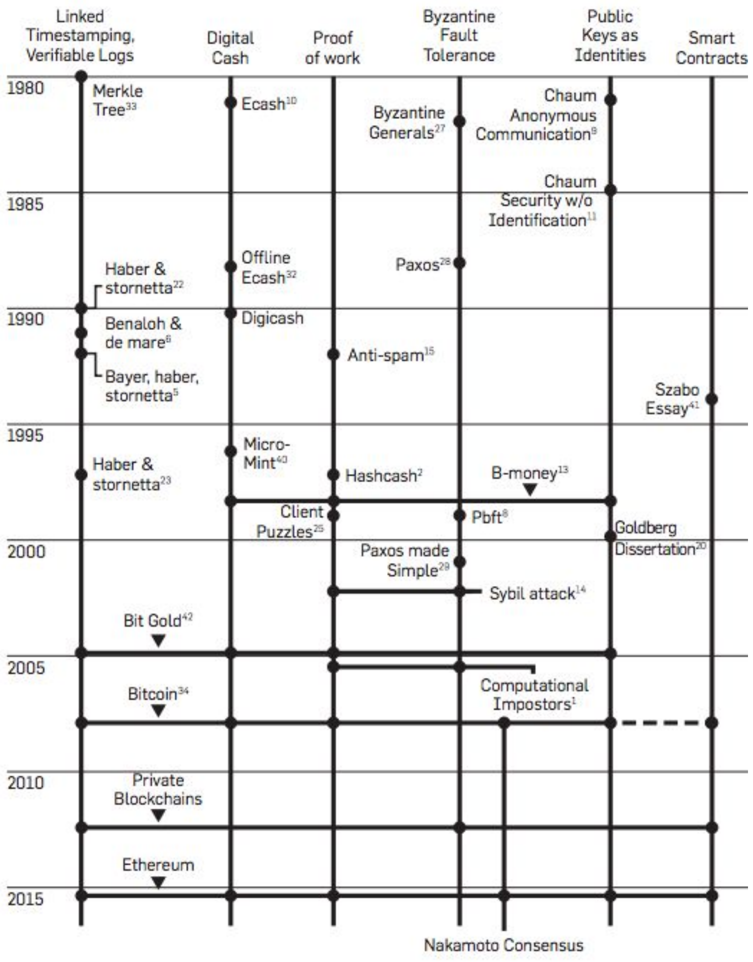
(https://hackernoon.com/merkle-trees-181cb4bc30b4)

The Road to Bitcoin

As explained by Naranayan and Clark 2017 , Bitcoin/Blockchain is not full of novel ideas.

Instead, almost all of the CS- and math-based tool it employs, had been proposed by cypherpunks and other people.

(Bitcoin’s Academic Pedigree, Narayanan and Clark)



The Road to Bitcoin

Bitcoin uses PoW, a hashcash-like computation, to allow a peer to add a new block to the chain.

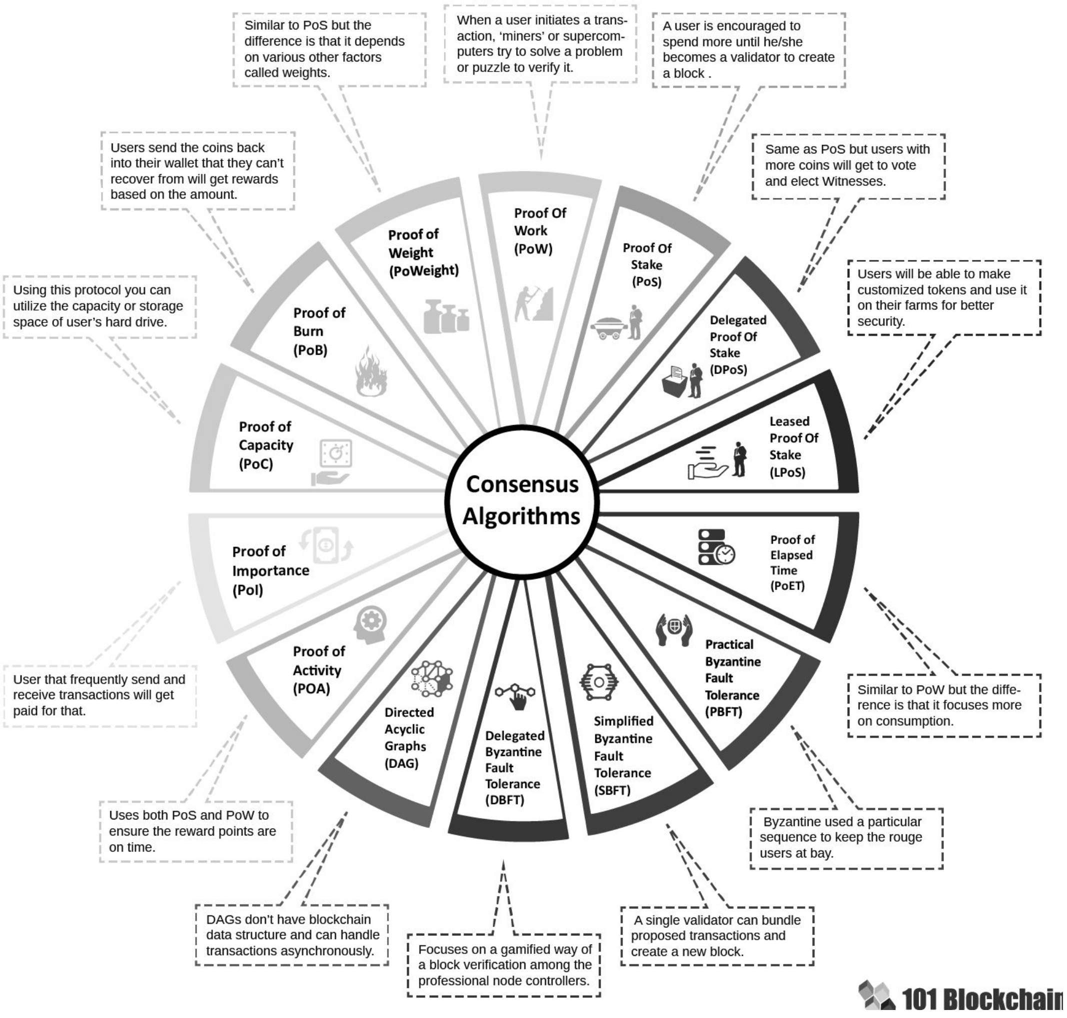
Hence, each CPU-cycle has one vote.

```
nonce = 0
while nonce < 0x100000000:
    header = ( struct.pack("<L", ver) +     prev_block.decode('hex')[::-1]
             + mrkl_root.decode('hex')[::-1] + struct.pack("<LLL", time_,
               bits, nonce))
    hash = hashlib.sha256(hashlib.sha256(header).digest()).digest()
    print nonce, hash[::-1].encode('hex')
    if hash[::-1] < target_str:
        print 'success'
        break
    nonce += 1
```

The Road to Bitcoin

Currently, we have many algorithms available to be used.

A few of them are proposed with a minor or major difference each year.



What Did Nakamoto Do?

Bitcoin P2P e-cash paper

Satoshi Nakamoto | Sat, 01 Nov 2008 16:16:33 -0700

I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.

The paper is available at:
http://www.bitcoin.org/bitcoin.pdf

The main properties:
 Double-spending is prevented with a peer-to-peer network.
 No mint or other trusted parties.
 Participants can be anonymous.
 New coins are made from Hashcash style proof-of-work.
 The proof-of-work for new coin generation also powers the
    network to prevent double-spending.

Bitcoin: A Peer-to-Peer Electronic Cash System

In 2008, Nakamoto created a synergistic cycle among the parties contributing to the Bitcoin network.

What Did Nakamoto Do?

1. A secure ledger is necessary to prevent double spending and thus ensure that the currency has a value. -> 2
2. A valuable currency is necessary to reward miners. -> 3
3. In turn, strength of mining power is necessary to secure the ledger. -> 1

Nakamoto's challenge was not just the design, but also convincing the initial community of users and miners to take a leap together into the unknown.

(http://delivery.acm.org/10.1145/3140000/3132259/p36-narayanan.pdf)

What Happened After?

Block #0

BlockHash 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

Summary

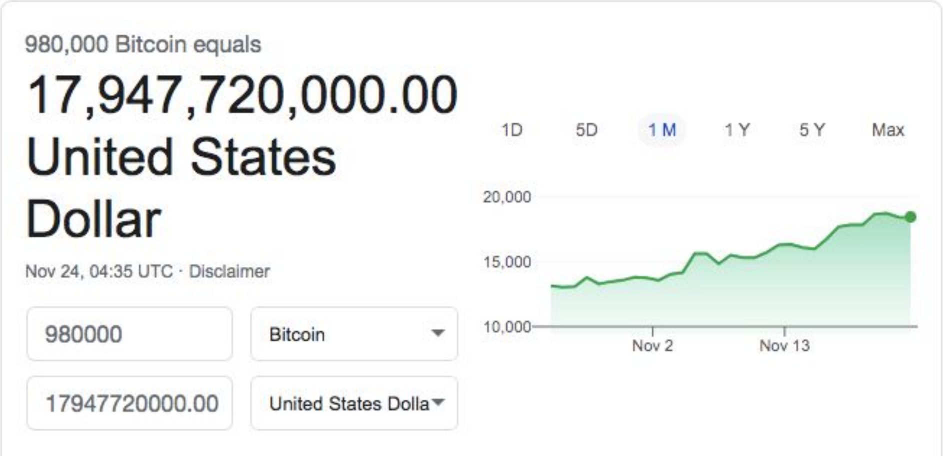
Number Of Transactions	1	**Difficulty**	1
Height	0 (Mainchain)	**Bits**	1d00ffff
Block Reward	50 BTC	**Size (bytes)**	285
Timestamp	Jan 3, 2009 8:15:05 PM	**Version**	1
Mined by		**Nonce**	2083236893
Merkle Root	4a5e1e4baab89f3a32518a88c31bc...	**Next Block**	1

What Happened After?

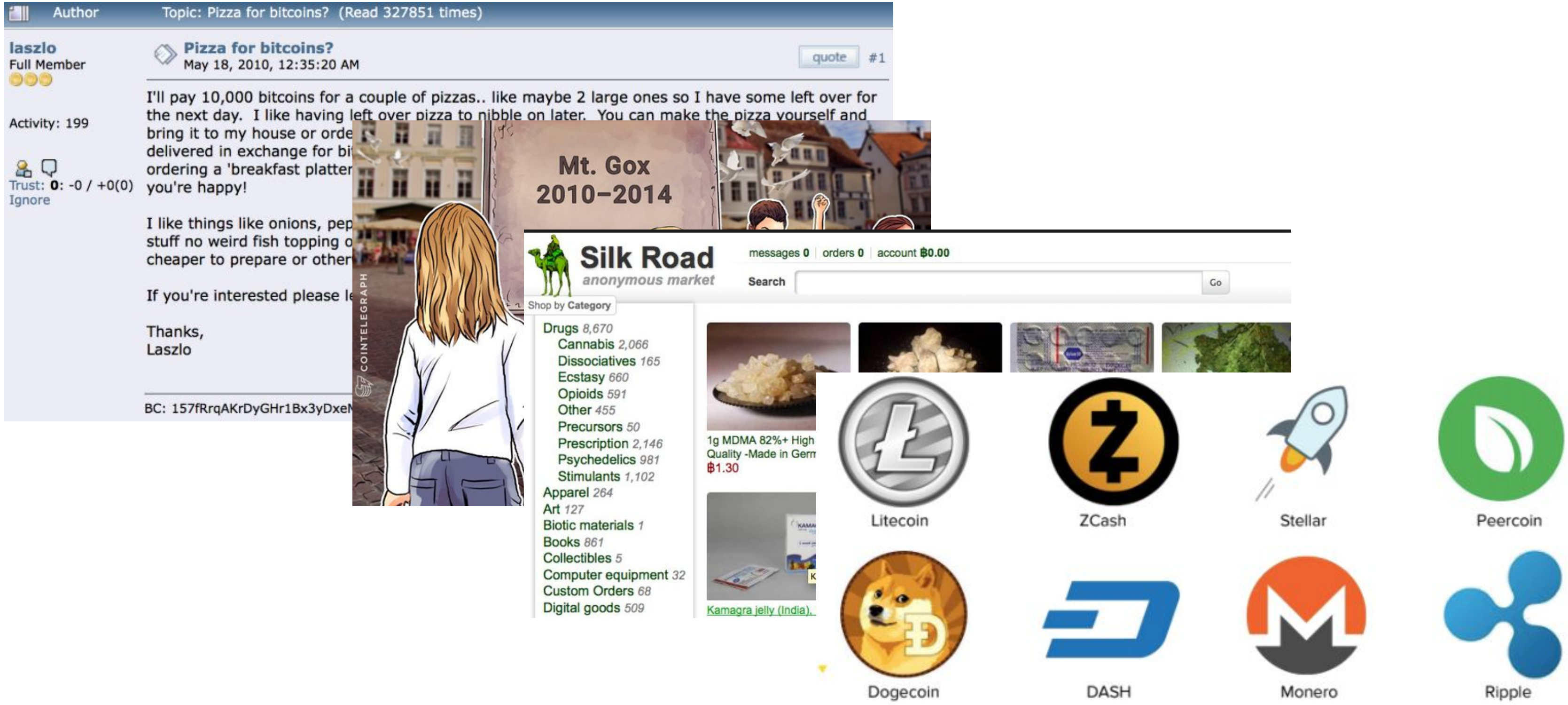
Overall, Nakamoto mined 19600 blocks. But does not Bitcoin provide anonymity, how do we know that?

For each block, he earned 50 BTC.

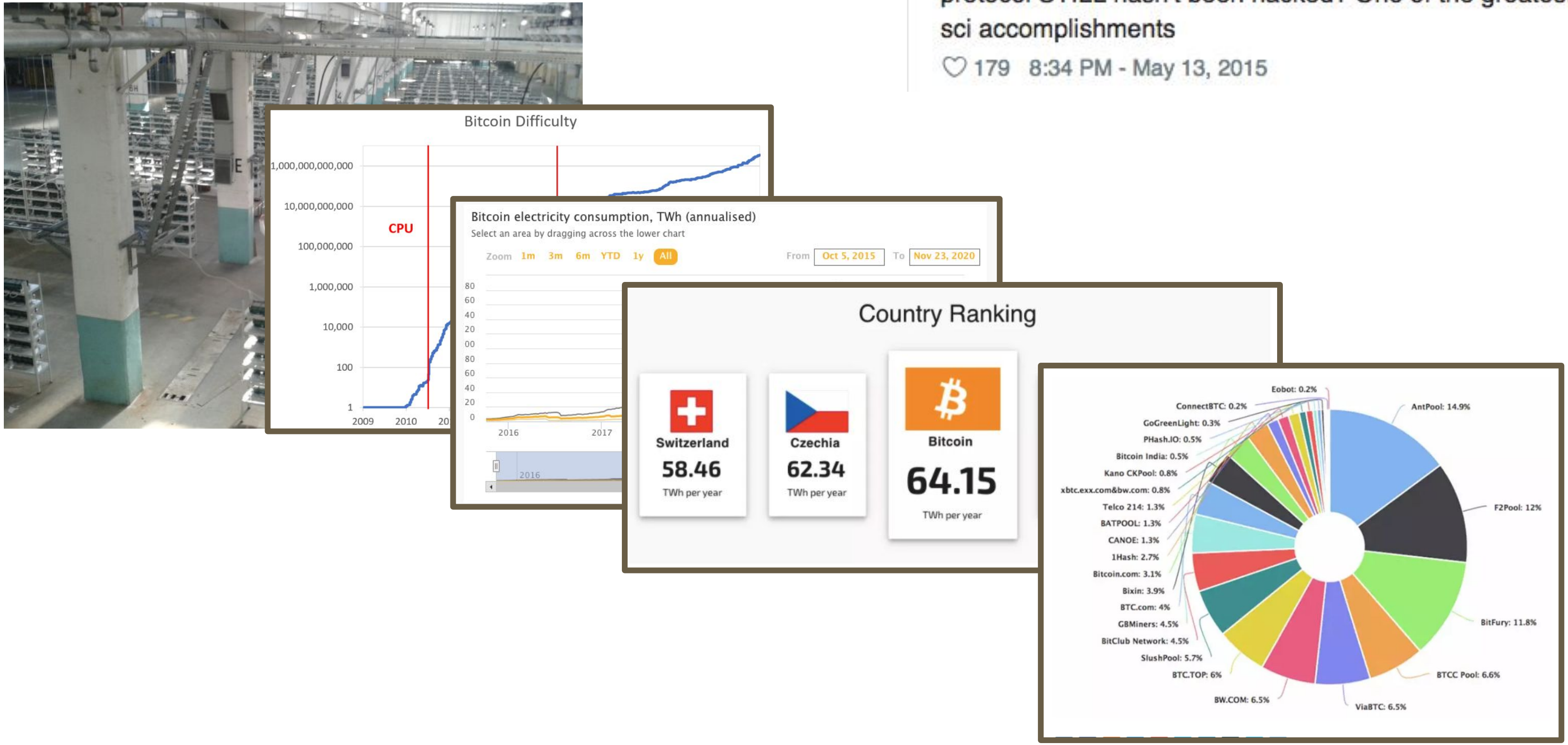
The current guess is that he owns 980,000 BTC.



What Happened After?



What Happened After?



Types of Blockchains

In the literature, there are three blockchain types:

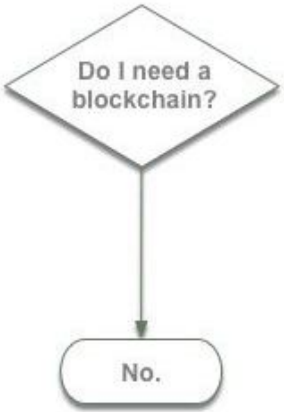
- Public blockchains - for the people, by the people and of the people
- Private blockchains - an evil company is stealing the tech of Cypherpunks :)
- Federated/consortium blockchains - multiple evil companies come together and stealing the tech of Cypherpunks :)

Types of Blockchains

Public and private blockchains are both

- run on decentralized network where each participant maintains a replica of a shared append-only ledger of digitally signed transactions,
- maintain the replicas in sync through a protocol referred to as consensus,
- provide certain guarantees on the immutability of the ledger, even when some participants are faulty or malicious.

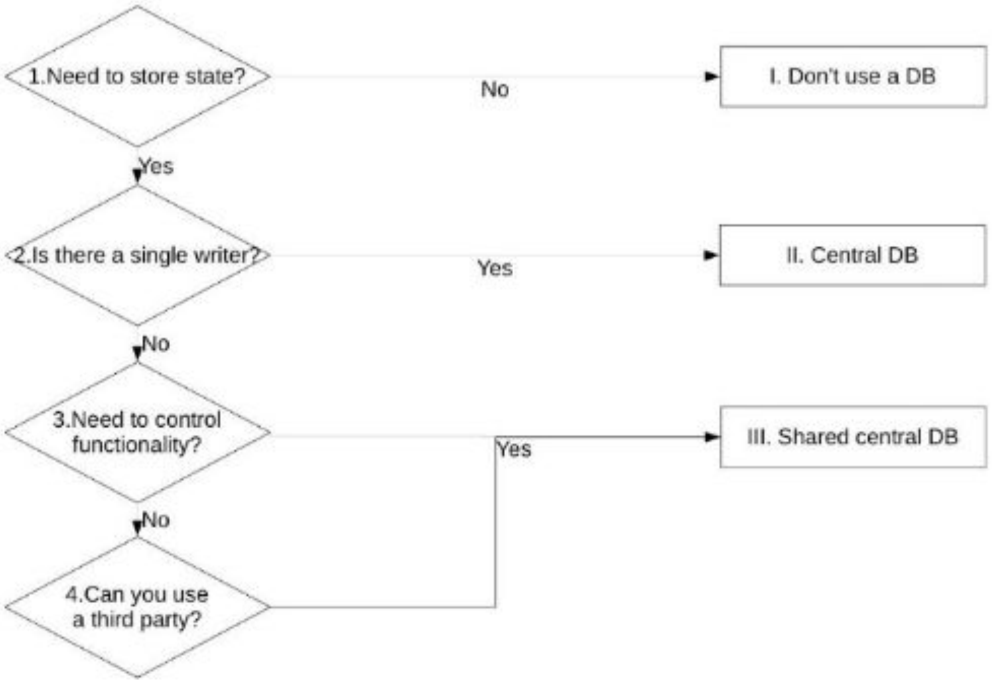
Types of Blockchains



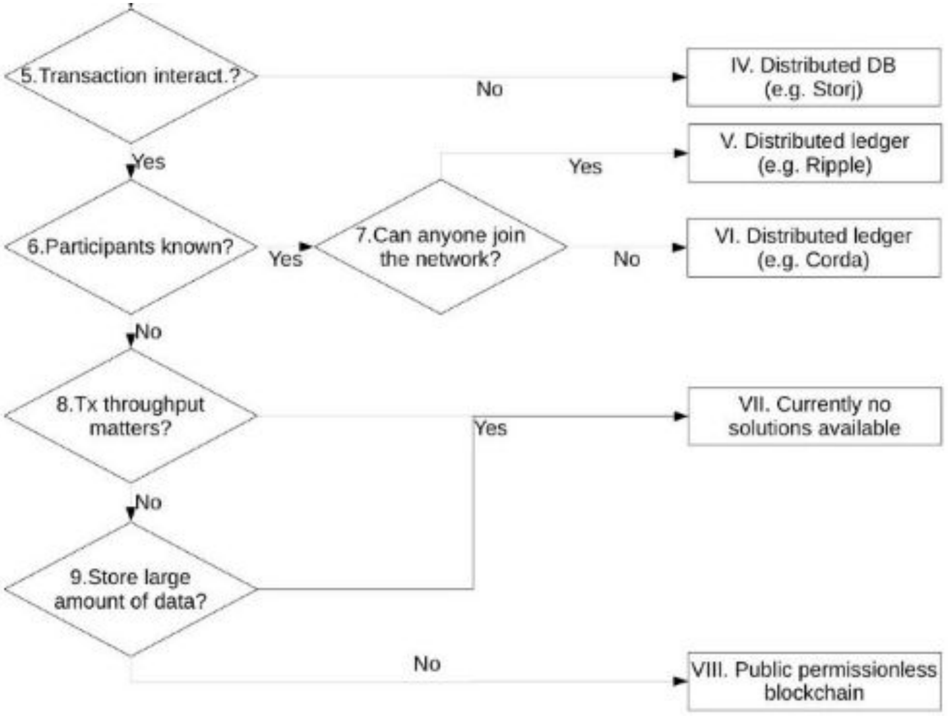
	Permissionless Blockchain	Permissioned Blockchain	Central Database
Throughput	Low	High	Very High
Latency	Slow	Medium	Fast
Number of readers	High	High	High
Number of writers	High	Low	High
Number of untrusted writers	High	Low	0
Consensus mechanism	Mainly PoW, some PoS	BFT protocols (e.g. PBFT [5])	None
Centrally managed	No	Yes	Yes

(Do you need a Blockchain?, Karl Wüst, Arthur Gervais, https://eprint.iacr.org/2017/375.pdf)

Do you need a blockchain? (Koens & Poll, 1/2)

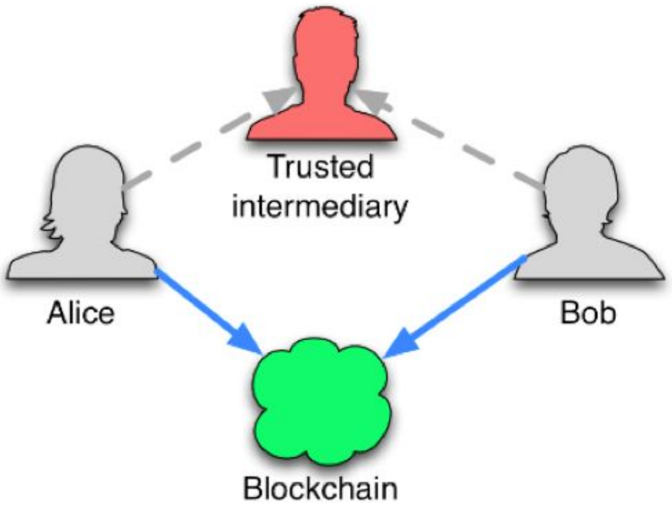


Do you need a blockchain? (Koens & Poll, 2/2)



Possible Applications

- Digital Assets/Currencies
- Internet of Things
- Globally Verifiable IDs/Certificates
- Smart Grids
- Digital Single Market
- Smart Appliances
- Supply Chains with Sensors
- Smart Contracts
- Data Sharing



References

Bitcoin's Academic Pedigree, Arvind Narayanan, Jeremy Clark, vol. 60, no. 12, December 2017 Pages 36-45 ACM New York, NY,

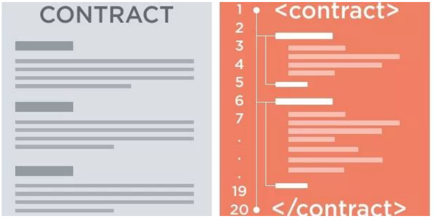
Blockchain at Berkeley Courses, https://blockchain.berkeley.edu/decal/fa18/fund/

Do you need a Blockchain?, Karl Wüst, Arthur Gervais, https://eprint.iacr.org/2017/375.pdf

Smart Contracts (and Ethereum)

Kamer Kaya
Sabancı Üniversitesi

Smart Contracts



con·tract

(noun) /ˈkäntrakt/

1. a written or spoken agreement ... that is intended to be enforceable by law.

smart con·tract

(noun) /smärt ˈkäntrakt/

1. code that **facilitates**, **verifies**, or **enforces** the negotiation or execution of a digital contract.
   a. **Trusted entity** must run this code

Smart Contracts

”A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract.

Smart contracts allow the performance of credible transactions without third parties.

These transactions are trackable and irreversible.”

(Wikipedia)

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.10;

contract Counter {
    uint public count;

    // Function to get the current count
    function get() public view returns (uint) {
        return count;
    }

    // Function to increment count by 1
    function inc() public {
        count += 1;
    }

    // Function to decrement count by 1
    function dec() public {
        count -= 1;
    }
}
```

https://solidity-by-example.org/first-app/

Smart Contracts - Digital Assets

- A smart contract is a computer program executed in a secure environment that directly controls digital assets
  - Domain name
  - Website
  - Money
  - Anything tokenisable (e.g. gold, silver, stock share etc)
  - Game items
  - Network bandwidth
  - Computation cycles

Smart Contracts - Possible Applications

Digital Identity

Healthcare (including Clinical Trials)

Cross Border Payments

Financial Data Recording (Loans and Mortgages etc.)

Government (or Governing)

Supply Chain Management

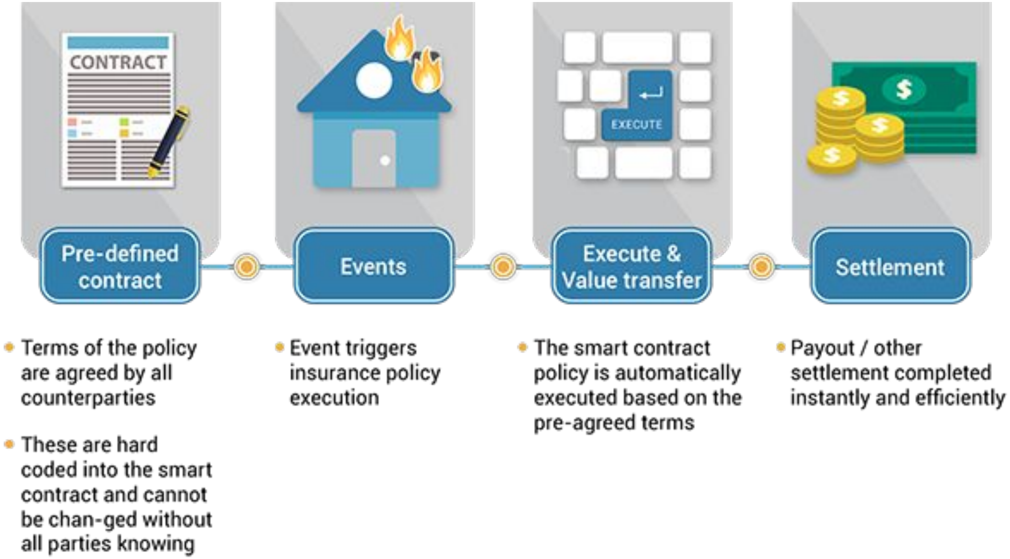
Insurance

Smart Contracts - Insurance

Measurable parameters of the event, such as wind speed, location of a hurricane or magnitude of an earthquake can be recorded onto the blockchain.

The rest is easy.

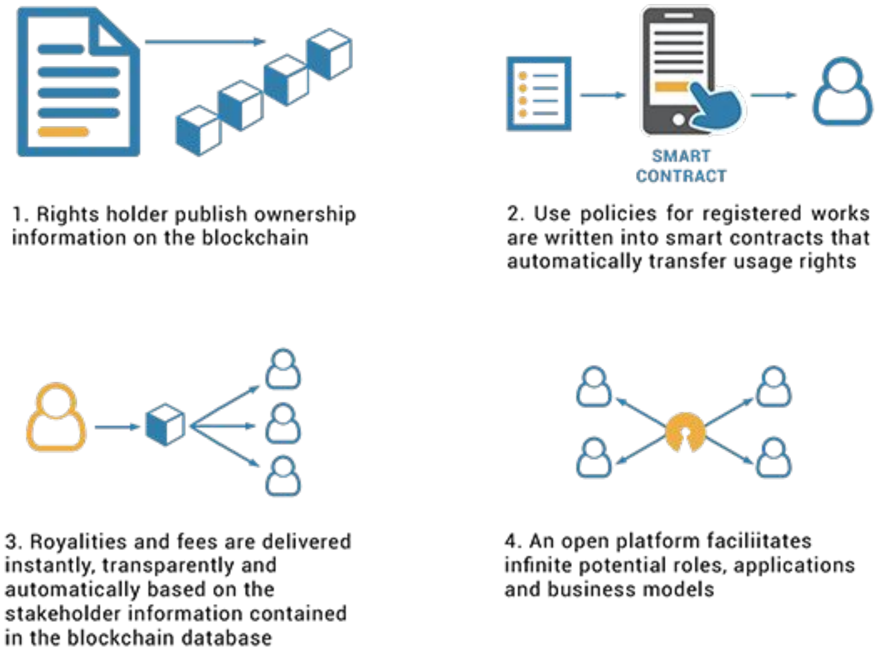
Use Case: Insurance Policies



Smart Contracts - Royalties

Use Case: Copyrighted Content

The issue with the current system is knowing who owns these rights and then to ensure that royalty payments are distributed to all who are legally obliged to receive payment.



Smart Contracts - Supply Chain

Use Case: Internet of Things

Each sensor forms its own node on a blockchain and smart contracts can record “possession” of the device to each individual sensor (and subsequent location).

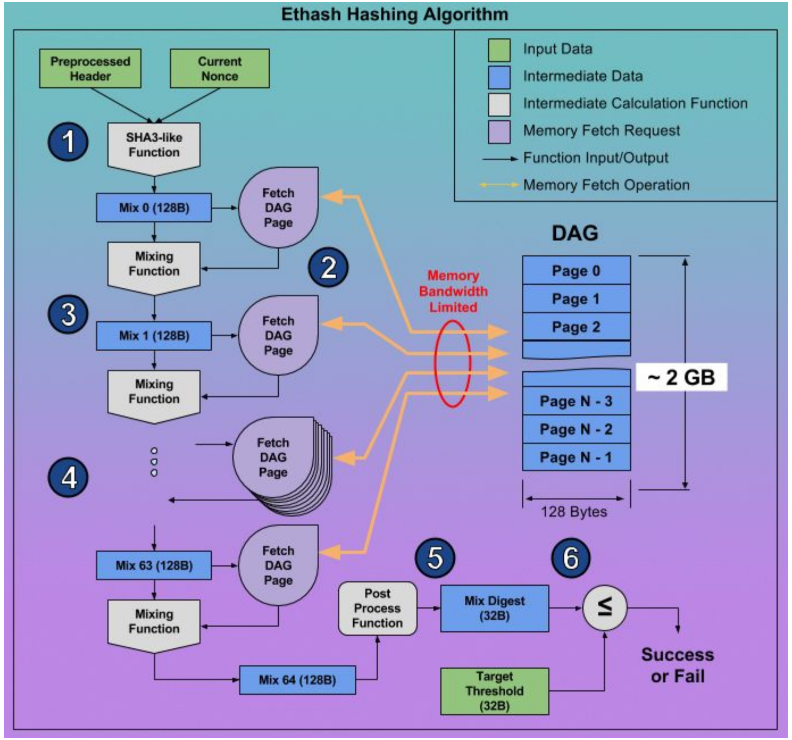
Ethereum

- Ethereum builds on Blockchain technology:
  - Bitcoin validates, stores and replicates transactions;
  - Ethereum adds computer code to this distributed framework.
- It is a platform running on several computers in a distributed network.
- It replicates and process data and small programs, i.e., smart contracts, without a central coordinator

Ethereum

As Bitcoin, Ethereum uses Proof of Work but a different hash function Ethash

The idea is almost the same but better ASIC resistance; GPUs are used.



Ethereum

As Bitcoin, Ethereum has its own token - Ether or ETH

Ethereum

As Bitcoin, Ethereum has its own token - Ether or ETH

Ethereum too has its own unit naming convention. The smallest unit is a wei and there are 1,000,000,000,000,000,000 of them per ETH.

Unit	Wei Value	Wei
wei	1 wei	1
Kwei (babbage)	1e3 wei	1,000
Mwei (lovelace)	1e6 wei	1,000,000
Gwei (shannon)	1e9 wei	1,000,000,000
microether (szabo)	1e12 wei	1,000,000,000,000
milliether (finney)	1e15 wei	1,000,000,000,000,000
ether	1e18 wei	1,000,000,000,000,000,000

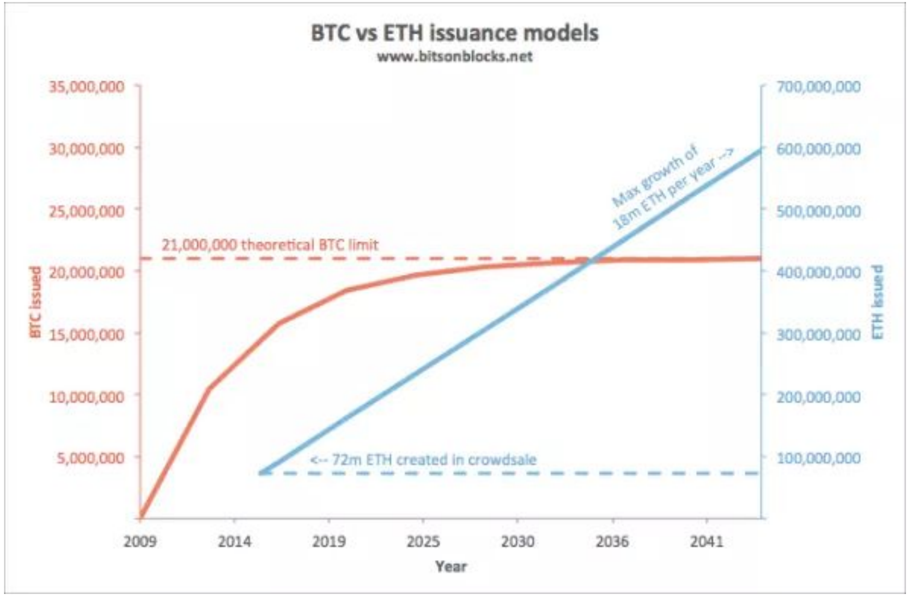
Ethereum

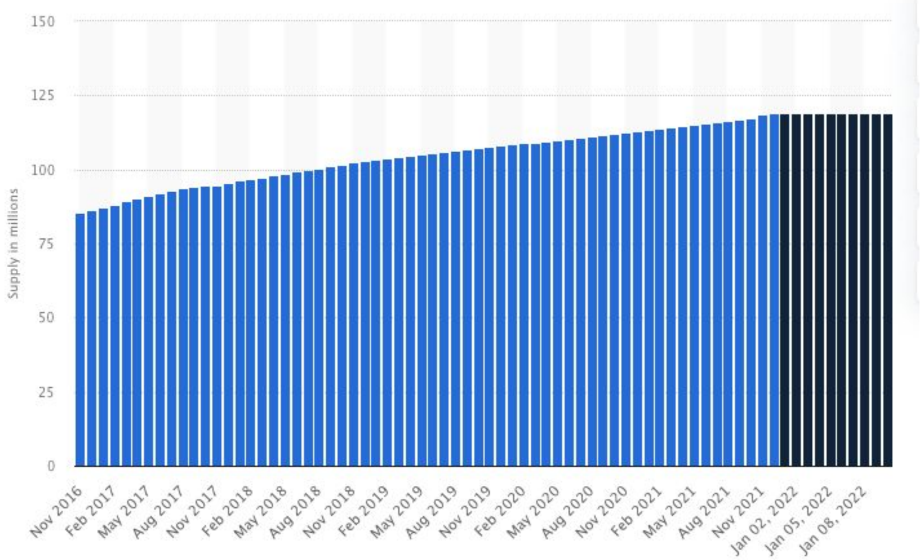
When compared to Bitcoin:

- Ethereum has a shorter block time (10 minutes vs 15-17 seconds)
    - More suitable for complex transactions (within smart contracts)
- Ethereum has smaller blocks
    - The number of transactions - the block size - is not limited in terms of byte size
    - It is limited by Gas - currently the maximum Gas in a single Ethereum block is 30,000,000
    - Even a simple transaction uses 21,000 Gas
    - Datawise, most Ethereum blocks are around 80KBs

Ethereum

Token generation has a constant rate.





Ethereum

The vision of Ethereum is being a world computer, or a distributed state machine.

Ethereum - State Model

Ethereum blockchain is essentially a transaction based state machine

A state machine refers to something that will read a series of inputs and based on those inputs, will transition to new state

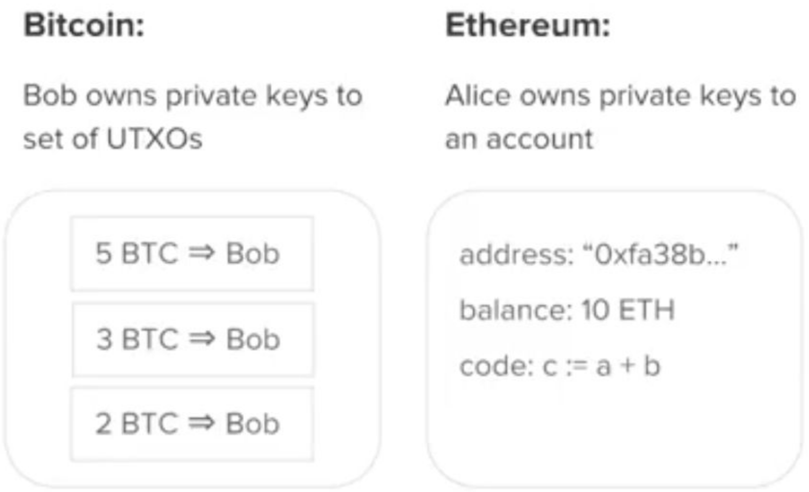
This state machine begins with genesis state - state at which no transactions have occurred on the network

Once transactions are executed, the current (genesis at the beginning) state transitions to another state, where the final state is the current state of the Ethereum network.

Ethereum - State Model

Ethereum does not follow the UTXO model as Bitcoin does - which was a nice approach to prevent double spending.

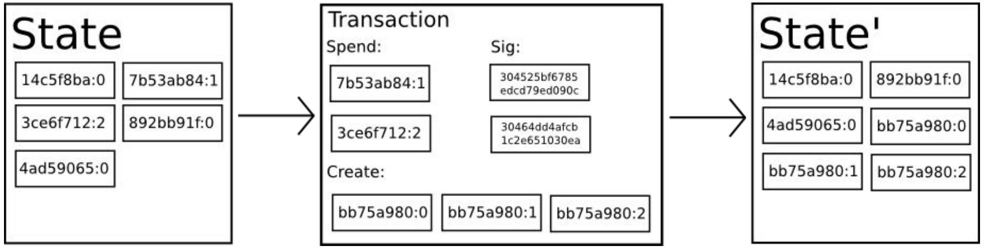
In accounts, we have balances.

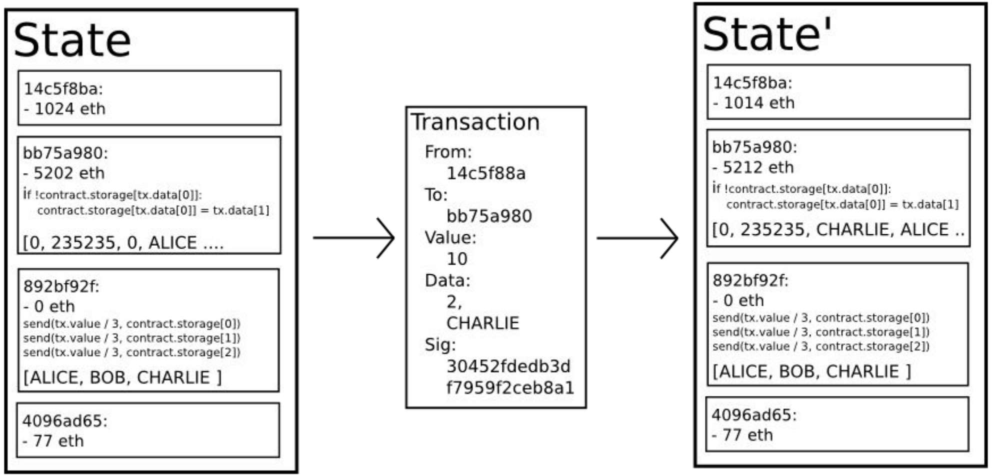


Ethereum - State Model

Ethereum does not follow the UTXO model as Bitcoin does - which was a nice approach to prevent double spending.

In accounts, we have balances.





Ethereum - State Model

For Ethereum - a blockchain is a “cryptographically secure transactional singleton computer/machine with shared-state.”

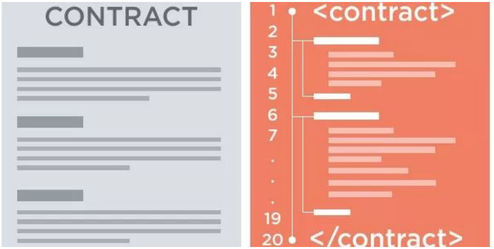
- Cryptographically secure - Can’t create fake transactions, erase transactions because of complex mathematical algorithms.
- Singleton machine - single instance of the machine for all the transactions being created in the system (“global truth”)
- Shared-state - state stored on this machine is shared and open to everyone

Smart Contracts in Ethereum (and in other chains)

In the context of blockchains and cryptocurrencies, smart contracts are:

- pre-written logic (computer code),
- stored and replicated on a distributed storage platform (eg a blockchain),
- executed/run by a network of computers (usually the same ones running the blockchain),
- and can result in ledger updates (cryptocurrency payments, etc).

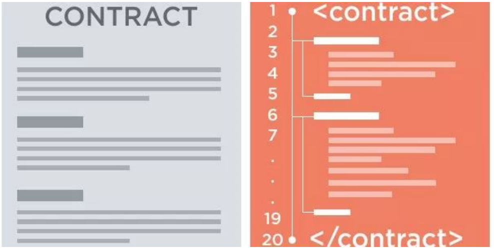
Smart Contracts in Ethereum (and ...)



There are four main functionalities of a contract:

- Store and maintain data; e.g., a token currency or organization’s membership
- Manage contract or relationship between untrusting users; e.g., financial contracts, escrow, insurance.
- Provide functions to other contracts.
- Complex authentication; e.g., **m-out-of-n** multisignature access

Smart Contracts in Ethereum



Smart contracts are little computer programs that are stored on Ethereum's blockchain.

They can be activated, or run, by funding them with some ETH.

In Ethereum, you set up a smart contract by creating a new account with some code in it, and uploading it to the Ethereum blockchain in a transaction.

Ethereum - Accounts

The address of an external account is determined from the public key while the address of a contract is determined at the time the contract is created (it is derived from the creator address and the number of transactions sent from that address, the so-called “nonce”).

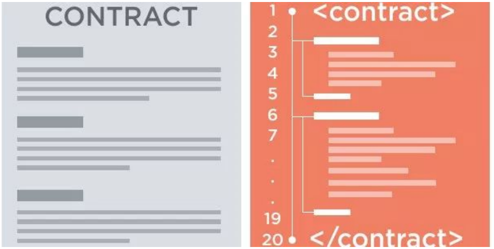
Regardless of whether or not the account stores code, the two types are treated equally by the EVM.

Ethereum - Accounts

In an account, we store

- nonce: The number of transactions sent (for external accounts), number of contracts created (for contract accounts)
- balance: The number of Wei owned by this address. 1e18 Wei per 1 Ether.
- storageRoot: A hash of the root node of a Merkle Patricia tree. This tree encodes the hash of the storage contents of this account, and is empty by default.
- codeHash: The hash of the EVM (Ethereum Virtual Machine—more on this later) code of contract account. Hash(“”) for external accounts.

Smart Contracts in Ethereum



When you want to use the contract, you can send a transaction to this account.

The nodes run the contract on their machines and agree on the output.

If no one is behaving badly consensus will be reached since all of the nodes are running the same code, with the same internal ether balance, internal contract state and the same input.

Ethereum - Transactions

The term "transaction" is used in Ethereum to refer to the signed data package that stores a message to be sent from an externally owned account. Transactions contain:

- The recipient of the message
- A signature identifying the sender
- The amount of ether to transfer from the sender to the recipient
- An optional data/payload field
- A `STARTGAS` value, representing the maximum number of computational steps the transaction execution is allowed to take
- A `GASPRICE` value, representing the fee the sender pays per computational step

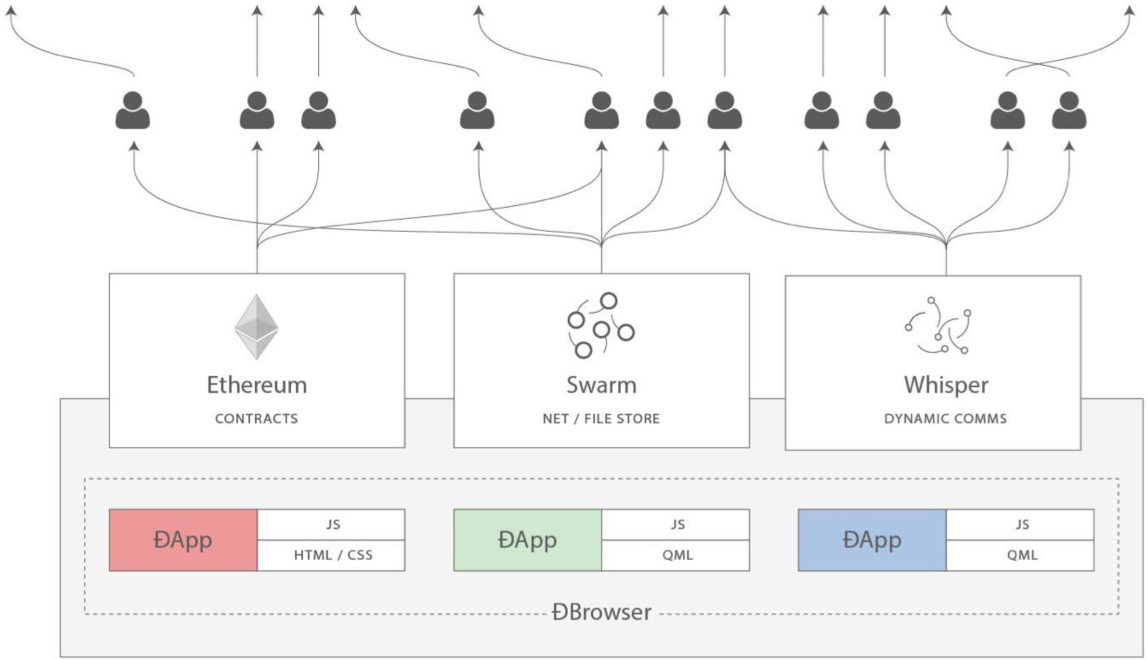
Ethereum - Transactions

A transaction

- is simply a message that is sent from one account to another account (which might be the same or empty - has a special meaning - see below).
- can include binary data (which is called “payload”) and Ether.
  - If the target account contains code, that code is executed and the payload is provided as input data.
  - If the target account is empty (the transaction does not have a recipient or the recipient is set to null), the transaction creates a new contract. The payload of contract creation is taken to be EVM bytecode and executed. The output of this execution is permanently stored as the code of the contract.

SCs in Ethereum - Blockchain Components

The vision of Ethereum is being a world computer, or a distributed state machine.



Smart Contracts in Ethereum

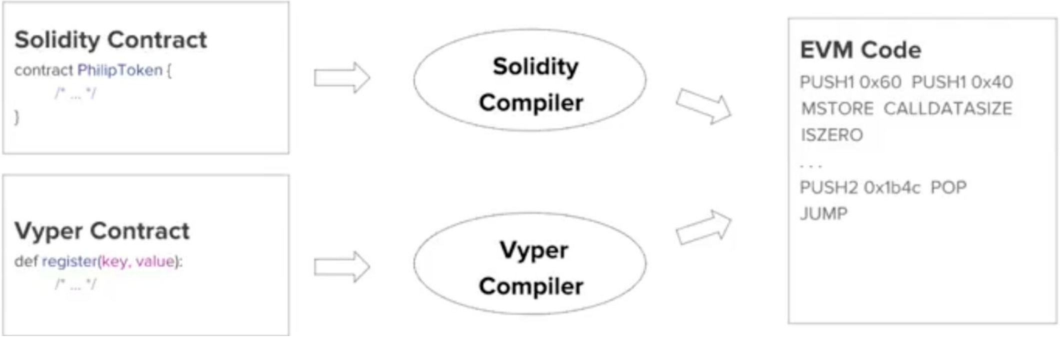
There were four common languages to write smart contracts, which can be compiled to run on Ethereum Virtual Machines.

- **Solidity** – similar to the language Javascript. This is currently the most popular and functional smart contract scripting language.
- **Vyper**: A python like (newer) language, simpler and claimed to be take over the Ethereum development.
- Serpent – similar to the language Python, and was popular in the early history of Ethereum.
- LLL (Lisp Like Language) – similar to Lisp and was only really used in the very early days. It is probably the hardest to write in.

Smart Contracts in Ethereum

To remove architectural differences which may be a risk for consistency the contracts are compiled down to a deterministic code and run by EVM.

So everyone will get the same exact result.



Smart Contracts in Ethereum

- The EVM is the ‘calculate’ element that can run contract logic
    - This is computation without relying on a central server.
- Swarm is P2P file sharing, incentivised with micropayments of ETH. Files are split into chunks, distributed and stored with participating volunteers. The storage nodes are compensated with ETH.
    - This is file storage without relying on a central server.
- Whisper is an encrypted messaging protocol that allows nodes to send messages directly to each other in a secure way and that also hides the sender and receiver from third party snoopers.
    - This is communications without relying on a central server.

Smart Contracts in Ethereum (and in other chains)

- Do not forget! when we say that the vision is “world computer” the main motivation is not efficiency
  - There is indeed parallelism but it is totally redundant
  - That is the codes are running on different machines in the same way (in a replicated manner)
- The main motivation is being a world computer without needing trusted third parties.

Smart Contracts in Ethereum (and in some other chains)

In Ethereum, there are two types of accounts:

1) Accounts that only store ETH – these are similar to Bitcoin addresses and are sometimes known as Externally Owned Accounts (EOAs, owned by people, organizations, etc.).

   You make payments from these accounts by signing transactions with the appropriate private key.

   (e.g., https://etherscan.io/address/0xb6103b35c1dfcfbc66aa6aa59e5dec376b79dd6a)

Smart Contracts in Ethereum (and in some other chains)

In Ethereum, there are two types of accounts:

2) Accounts that store ETH and have code (**smart contracts**) that can be run

   These accounts are owned by the contract.

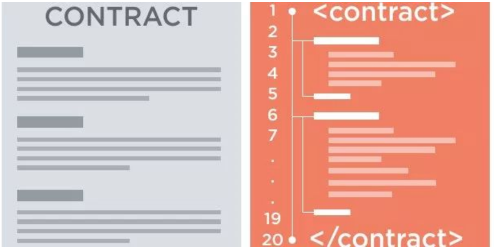
   A contract is activated by a transaction sending ETH into it. Once the smart contract has been uploaded, it sits there waiting to be activated/poked.

   (e.g., https://etherscan.io/address/0xcbe1060ee68bc0fed3c00f13d6f110b7eb6434f6#code)

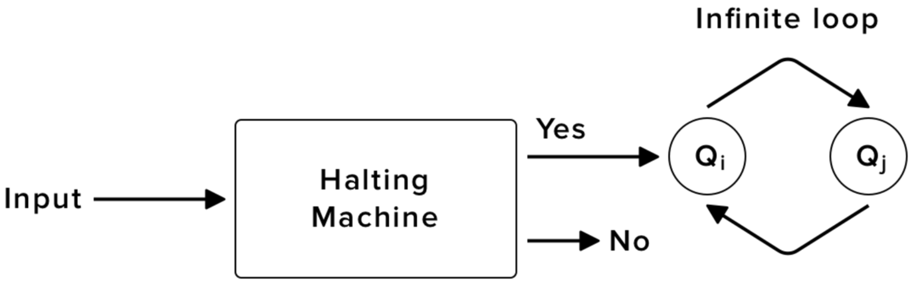
Smart Contracts in Ethereum/Solidity

```
contract token {
    mapping (address => uint) public coinBalanceOf;
    event CoinTransfer(address sender, address receiver, uint amount);

  /* Initializes contract with initial supply tokens to the creator of the contract */
  function token(uint supply) {
        if (supply == 0) supply = 10000;
        coinBalanceOf[msg.sender] = supply;
    }

  /* Very simple trade function */
    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
        if (coinBalanceOf[msg.sender] < amount) return false;
        coinBalanceOf[msg.sender] -= amount;
        coinBalanceOf[receiver] += amount;
        CoinTransfer(msg.sender, receiver, amount);
        return true;
    }
}
```

SCs in Ethereum (and in other chains)



Q: What happens when there is an infinite loop (or a similar code pattern) that makes the contract run forever?



Smart Contracts in Ethereum

Ethereum's solution is setting a cost for each op-code execution

When you activate/poke a smart contract, you ask the miners in the whole network to each individually perform the calculations within it.

This costs them time and energy, and Gas is the mechanism by which you pay them for that service.

Smart Contracts in Ethereum

Ethereum's solution is setting a cost for each op-code execution

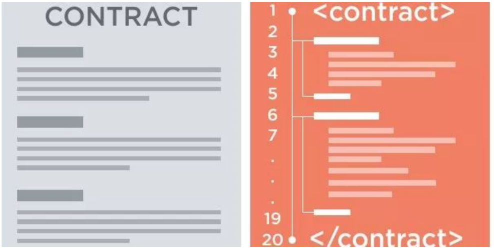
- Every contract requires Gas, which fuels contract executions
- Every EVM op-code requires some gas in order to execute.
  - Hence, the more complex the smart contract (the number and type of computational steps, memory used for storage, etc), the more Gas the contract requires to run and complete.
- Every transaction specifies
  - The startgas/gasamount, or the maximum quantity of gas it is willing to consume
    - this is fixed for a contract
  - The gasprice, or the fee in ether it is willing to pay per unit gas
    - this is similar to transaction fee in Bitcoin

Smart Contracts in Ethereum

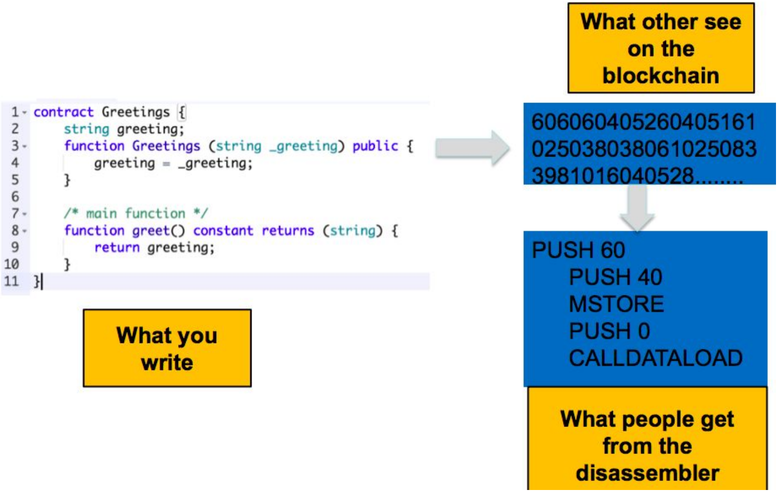
- fee = startgas x gasprice is subtracted from the sender’s account
  - **sender**: who poke the contract
- If the contract succeeds: the remaining Gas is refunded
- If the contract execution runs out of Gas before it finishes:
  - the execution is reverted
  - no refund
- purchasing gas = purchasing distributed, trustless computational power

Gas is for miners who run the contract, not for the people who wrote the contract.

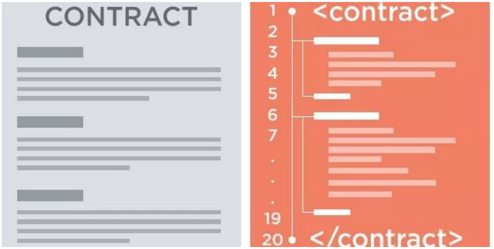
Smart Contracts in Ethereum



The EVM has three areas where it can store data- storage, memory and the stack, which are explained in the following slides.



Smart Contracts in Ethereum

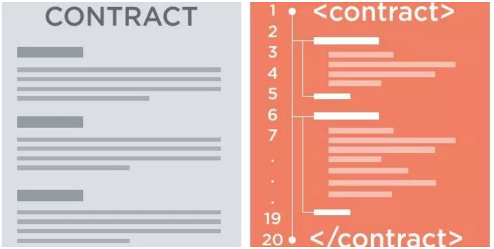


Each account has a data area called storage, which is persistent between function calls and transactions (stored in the blockchain, consider this like a hard disk).

Storage is a key-value store that maps 256-bit words to 256-bit words. It is not possible to enumerate storage from within a contract and it is comparatively costly to read, and even more to modify storage.

A contract can neither read nor write to any storage apart from its own.

Smart Contracts in Ethereum



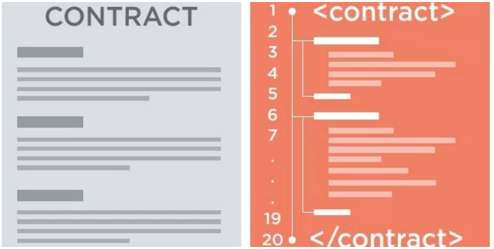
A contract obtains a freshly cleared memory for each message call.

Memory is linear and can be addressed at byte level, but reads are limited to a width of 256 bits, while writes can be either 8 bits or 256 bits wide.

Memory is expanded by a word (256-bit), when accessing (either reading or writing) a previously untouched memory word (i.e., any offset within a word).

At the time of expansion, the cost in Gas must be paid. Memory is more costly the larger it grows (it scales quadratically).

Smart Contracts in Ethereum



All computations in EVM are performed on a data area called the stack.

It has a maximum size of 1024 elements and contains words of 256 bits. Access to the stack is limited to the top end.

Of course it is possible to move stack elements to storage or memory in order to get deeper access to the stack, but it is not possible to just access arbitrary elements deeper in the stack without first removing the top of the stack.

Smart Contracts in Ethereum

How you write the contract has an impact on the cost.

Each operation spends a different amount of Gas.

- ADD/SUB 3 Arithmetic operation
- MUL/DIV 5 Arithmetic operation
- ADDMOD/MULMOD 8 Arithmetic operation
- AND/OR/XOR 3 Bitwise logic operation
- LT/GT/SLT/SGT/EQ 3 Comparison operation
- POP 2 Stack operation
- PUSH/DUP/SWAP 3 Stack operation
- MLOAD/MSTORE 3 Memory operation
- JUMP 8 Unconditional jump
- JUMPI 10 Conditional jump
- SLOAD 200 Storage operation
- SSTORE 5,000/20,000 Storage operation
- BALANCE 400 Get balance of an account
- CREATE 32,000 Create a new account using CREATE
- CALL 25,000 Create a new account using CALL

Smart Contracts in Ethereum

- The fee for expanding memory is determined as follows:
  - fee(x) = x * 3 + floor(x**2 / 512) is the total fee for expanding the memory to **x** 32-byte (256-bit) chunks and if a particular operation expands memory from size **x** to **y**, the additional gas cost is fee(**y**) - fee(**x**)s

- Storage gas costs are a little simpler but much more expensive:
  - 20,000 gas when a value is set to non-zero from zero;
  - 5,000 gas when writing to existing storage
  - and a 15,000 gas refund when a non-zero value is set to zero.

Smart Contracts in Ethereum (and some other chains)

Hence, it is a good idea to:

1. Removing useless code
2. Reduce expensive operations (especially in a loop)
3. Use short circuit evaluations
4. Combine the loops

Smart Contracts in Ethereum

- Contracts can even create other contracts using a special opcode (i.e. they do not simply call the zero address as a transaction would).
- As before, the only difference between these create calls and normal message calls is that the payload data is executed and the result stored as code and the caller / creator receives the address of the new contract on the stack.

Smart Contracts in Ethereum

- The only way to remove code from the blockchain is when a contract at that address performs the selfdestruct operation.
- Removing the contract in theory sounds like a good idea, but it is potentially dangerous, as if someone sends Ether to removed contracts, the Ether is forever lost.
- Even if a contract is removed by “selfdestruct”, it is still part of the history of the blockchain and probably retained by most Ethereum nodes.
- So using *selfdestruct* is not the same as deleting data from a hard disk.

Smart Contracts in Ethereum - Dapps

Some References

- https://blockchain.berkeley.edu/spring-2022-developers-decal/
- https://bitsonblocks.net/2016/10/02/gentle-introduction-ethereum/