Communication in Distributed Systems

11 Blockchain

Dr.-Ing. Michael Rademacher

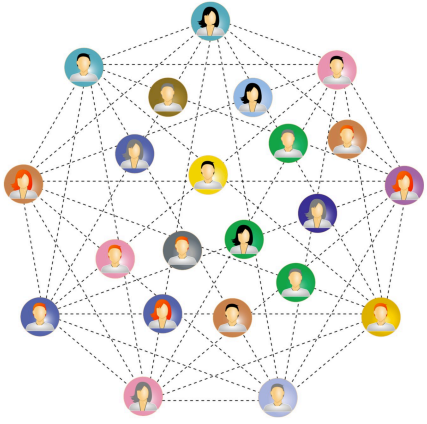
2021-06-15

Blockchain Overview

A major problem with blockchains

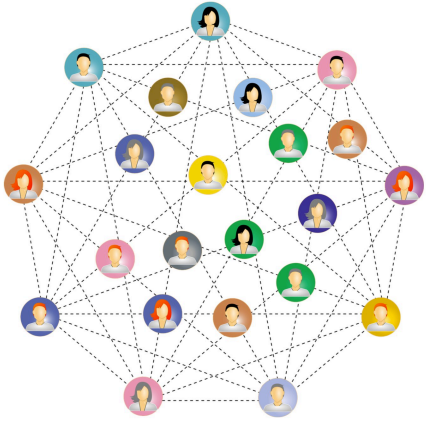
This Lecture: No guide for trading strategies of cryptocurrencies.

Definition



- A **blockchain** is a network based on a distributed transaction database for a tamper-proof storage of linear records
- The stored data is protected against unauthorized manipulation by **cryptography**, **decentralization** and **game theory**
- The innovation lies in the clever combination of these research areas and led for the first time to an efficient solution of the **Double-Spending-Problem**

Definition



- Blockchains make it possible to **dispense intermediate instances** and significantly reduce trust problems for transactions
- They offer very good protection against manipulation and are therefore highly **resistant to censorship**
- These properties lead to different possible **application**
- However, the overhead a blockchain is very high compared to a centralized database. The possible application should be questioned critically

Double-Spending-Problem

Definition:

The possibility to propagate a transaction (digital value transfer) multiple times on a decentralized network.

- Problem: Duplicates can cause a digital token to be transferred more than once
- Solution: Find a chronological **consensus** in a decentralized way so that no participant can manipulate it selfishly
- Central solution: Classic database (SEPA, Paypal)
- Decentralized solution: Blockchain

The blockchain led for the first time to the solution of the double-spending problem in decentralized systems.

Classification

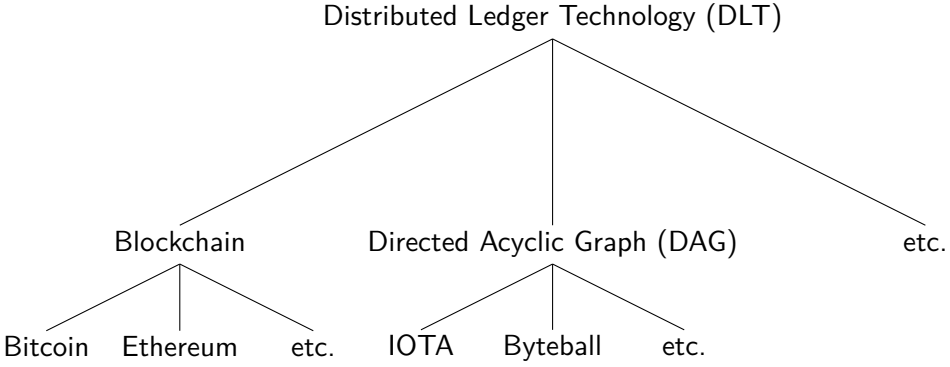
A Blockchain can be classified under the term Distributed Ledger Technology (DLT).

- A ledger is redundantly managed by different parties using multiple copies
- New transactions are transferred to all copies
- **The distribution of copies is a known and solved problem**
- **The challenge and the goal is to establish a consensus of the copies**

A blockchain is an innovative approach to implement a distributed ledger.

Every blockchain is a DLT, but not every DLT is a blockchain.

Classification



Models

Blockchains can be categorized into 3 different models [12]:

- **Public**
- **Private**
- **Federated**

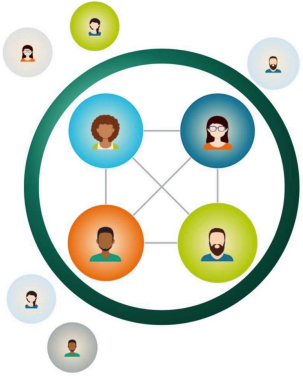
These models differ greatly in their degree of decentralization and the consensus algorithm used.

The term blockchain is highly controversial and debated for private and federated ledgers.

Public Blockchains

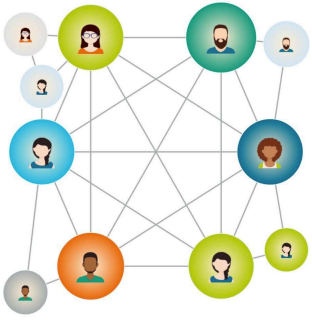
- Publicly accessible network
- Transactions are transparent and anonymous/pseudonymous
- **Anyone** can read, write and validate transactions
- Highest protection against tampering:
  - Cryptography
  - Decentralization
  - Game theory
- No central management
  - Therefore **no trust necessary**
- Complex consensus building with high computing power
  - Therefore mostly slow and poorly scalable
- Dimensions: Internet

Private Blockchains



- Closed network
- Transactions are private
- Special authorization needed to read, write and validate transactions
- Low protection against tampering:
  - Cryptography
  - Hardly any decentralization
  - No game theory necessary since network participants are all known
- Central management
  - Therefore **trust necessary**
- Trivial consensus building
  - Therefore mostly fast and easily scalable
- Dimensions: Intranet (private LAN/WAN)

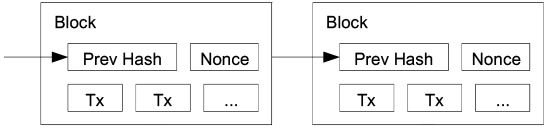
Federated Blockchains



- Hybrid solution
- Semi-public network
- Transactions are mostly transparent
- Special authorization needed to read, write or validate transactions
- Low protection against tampering:
  - Cryptography
  - Low decentralization
  - Low game theory
- Decentralized management by a consortium
  - Therefore **trust necessary**
- Trivial consensus building by a consortium
  - Therefore mostly fast and easily scalable

Important building blocks of blockchains

Cryptographic hash functions



[11]

Usage of cryptographic hash functions in Blockchains:

- Data can be stored in units of blocks
  - Enables the concept of a hashed list
  - Data blocks have fixed chronological order
  - Previous data blocks cannot be manipulated unnoticed
- Consensus mechanism
  - Basis of the so-called Proof of work (PoW) Hashcash Puzzle

Public Key Cryptography

$$f_{oneway}(k) = p$$

Private Key k: 

- Basis for generating a key pair
- Secret, non-public key

Public Key p: 

- Is irreversibly derived from the private Key
- Public, not secret key

Digital signatures

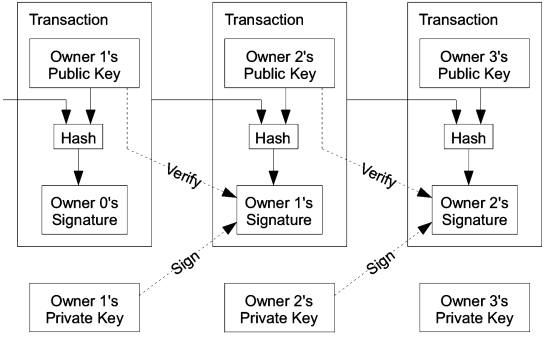
$$f_{sign}(m, k) = sig$$

$$f_{verify}(sig, p)$$

- Are supposed to emulate properties of physical signatures
- Three protection goals are fulfilled:
  1. Integrity
  2. Authenticity
  3. Liability
- Are generated by a signature scheme of the **sender**

Digital signatures

Transaction from owner to owner+1.
owwner signs the public key of owner+1



"We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin."[11]

Usage of digital signatures in Blockchains:

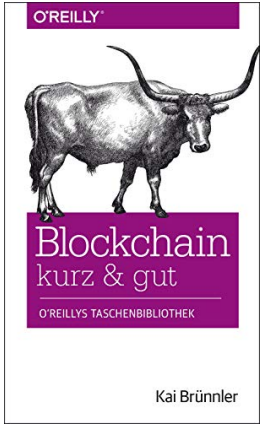
- Full transaction security
- Transactions are:
  - Not manipulated (integrity)
  - Authorized (authenticity)
  - Not deniable (liability)
- **digital ownership through private keys**
- **no private key, no ownership**

From a centralized Bank to a blockchain

Methodology

Methodology for the rest of theses slides: Piece by Piece derivation of a functional Blockchain starting with current online banking.

The methodology is inspired by the following book [6]:



The author's summary of blockchains [10]:

BRSU Communication Report Nr. 1
ISBN: 978-3-96043-081-0, Digital Object Identifier: 10.18418/978-3-96043-081-0

Technical Fundamentals of Blockchain Systems

Oliver Kattwinkel[1], Michael Rademacher[2]

Abstract
This work provides a short but technical introduction to the main building blocks of a blockchain. It argues that a blockchain is not a revolutionary technology but rather a clever combination of three fields: cryptography, decentralization and game theory. In addition, it summaries the differences between a public, private and federate blockchain model and the two prominent consensus mechanism Proof-of-Work (POW) and Proof-of-Stake (POS).

Keywords
Blockchain — Cryptography — Proof-of-Work — Proof-of-Stake

[1] *Kleenecode GmbH, Sankt Augustin, Germany*
[2] *Department of Computer Science, University of Applied Sciences Bonn-Rhein-Sieg, Sankt Augustin, Germany*
oliver.kattwinkel@kleenecode.com, michael.rademacher@inf.h-brs.de

01: Bank-protocol

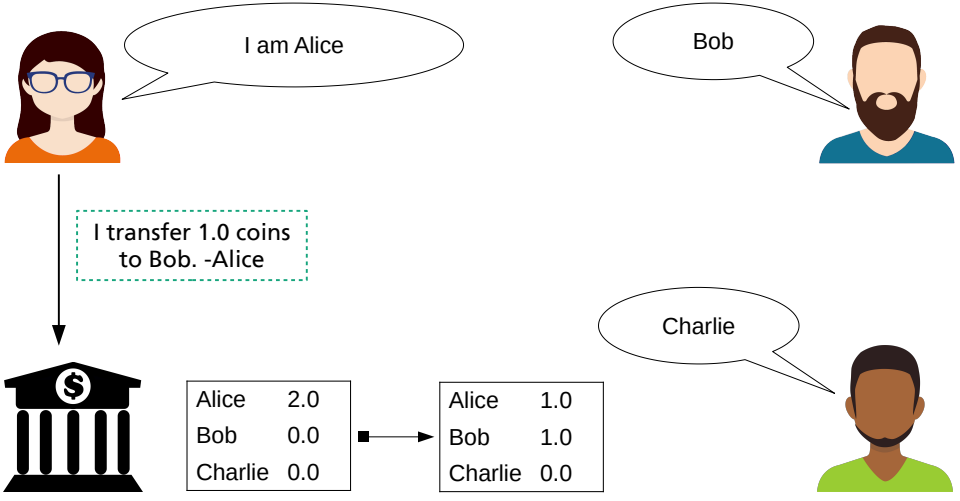
Initial situation: Current online banking

- A bank functions as a central administration
- Digital value transfer of a Fiat currency
- Access through credentials
  - Username and password
  - Digital signatures

Rules for this protocol:

- The bank manages credit balances of all participants in a central credit database
- To transfer credits, participants send signed transactions to the bank
- The bank accepts only valid transactions
- The bank updates balances according to accepted transactions
- All participants request their current credit via the bank

01: Bank-protocol



01: Bank-protocol

Problem:

- By managing all data the bank has great centralized power
- Participants dependent on the security of the bank
- As intermediary, the bank has more rights than any other participant in this system
  - Censorship through rejection of transactions
  - Inflation through generation of new credit
  - Manipulation by changing credit balances

The Five/Nine Hack in the TV Series Mr. Robot [4]. Destroy financial records of the largest bank in the USA.

Solution:

- No bank, no single intermediary, no centralized power
- Who manages the credit of the participants?
- **Each participant manages each credit!**

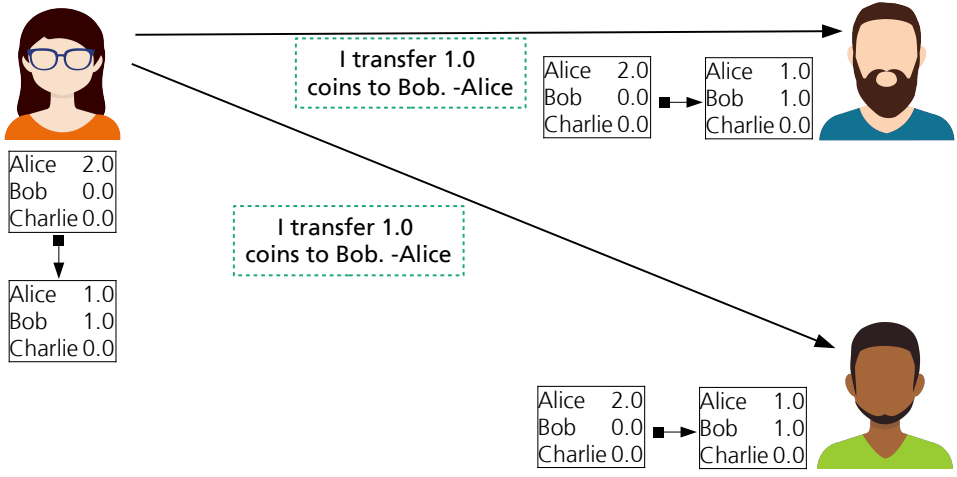
02: P2P-protocol

Transiation from a hierarchical bank protocol to a flat P2P-protocol

Rules:

- Each participant manages each credit of all participants in their own credit database
- To transfer credits, participants send signed transactions to all participants
- Each participant accepts only valid transactions
- Each participant updates all balances according to accepted transactions

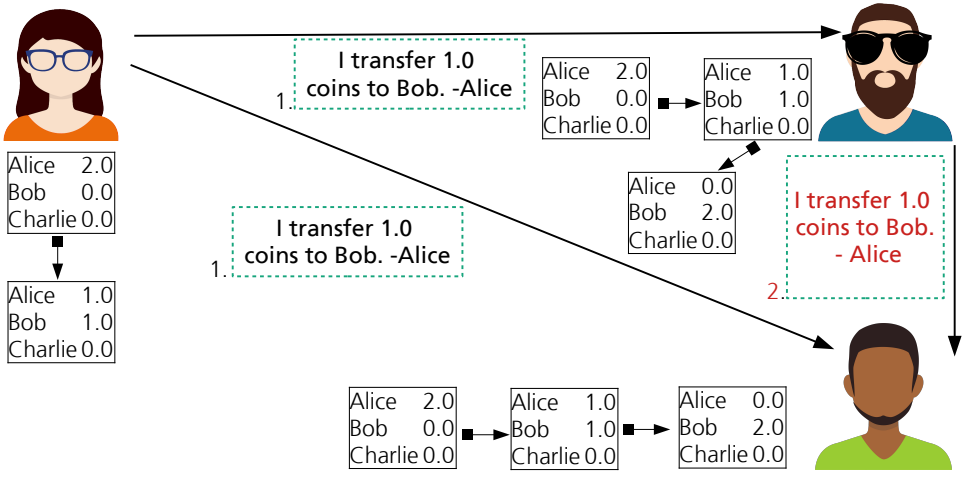
02: P2P-protocol



Problem:

- Each participant can manipulate their own database “as they wish”
- Manipulation of other databases through so-called replay attacks
- Replay attack:
  - Attacker can resend received transactions
  - Recipients verify authenticity only by the signature
  - All participants (except the victim) will accept the transaction
- Consensus (all participants agree on state of the database) is not assured

02: P2P-protocol replay attack



Solution:

- Each participant does not store credit as a mere value
- Credit is stored as a quantity of digital assets (coins) with unique serial-numbers

Coin 57

Coin 58

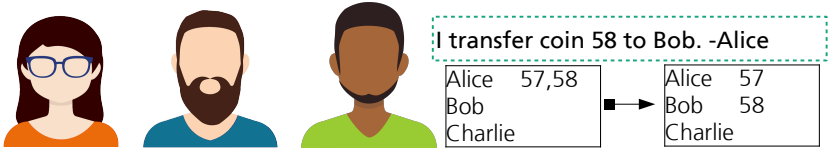
Coin 59

03: Serial-number-protocol

Introduction of coins with serial-numbers to protect against replay attacks

Rules:

- Each participant manages serial-numbers of coins of all participants in their own credit database
- To transfer credits, participants send signed transactions to all participants
- Each participant accepts only valid transactions
- Each participant updates balances according to accepted transactions



Problem:

- Still (but limited) vulnerable to replay attacks
- Attacker can successfully resend received transactions if the victim has regained possession of the coin
- Consensus is not assured

Solution:

- A coin requires not only a serial-number but its entire transaction history

Reinterpretation of credit balances as transaction history

Rules:

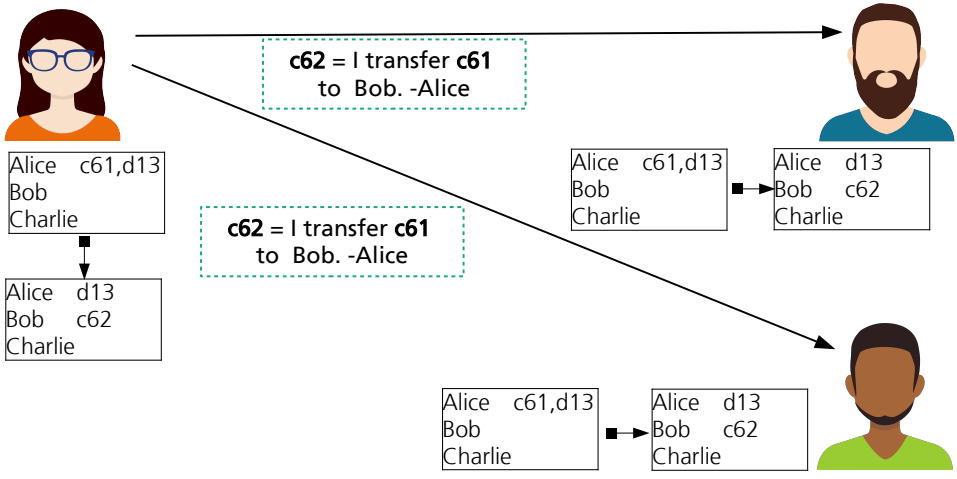
- The Credit is the transaction that transfers the coin
  - i.e. credit is a list of transactions (history)
- The payee (receiver) of a transaction is the owner of the credit
- Each participant manages list of all transactions in his own transaction database
- To transfer credits, participants send signed transactions to all other participants
- Each participant accepts only valid transactions
- Each participant updates its transaction database according to accepted transactions

Transaction history:

1. ...
2. $c_{61} =$ I transfer *transaction* to *receiver*. - *sender*
3. $c_{62} =$ I transfer c_{61} to Bob. - Alice
4. $c_{63} =$ I transfer c_{62} to Charlie. - Bob

   $c_{63} =$ I transfer [I transfer c_{61} to Bob. - Alice] to Charlie. - Bob
5. ...

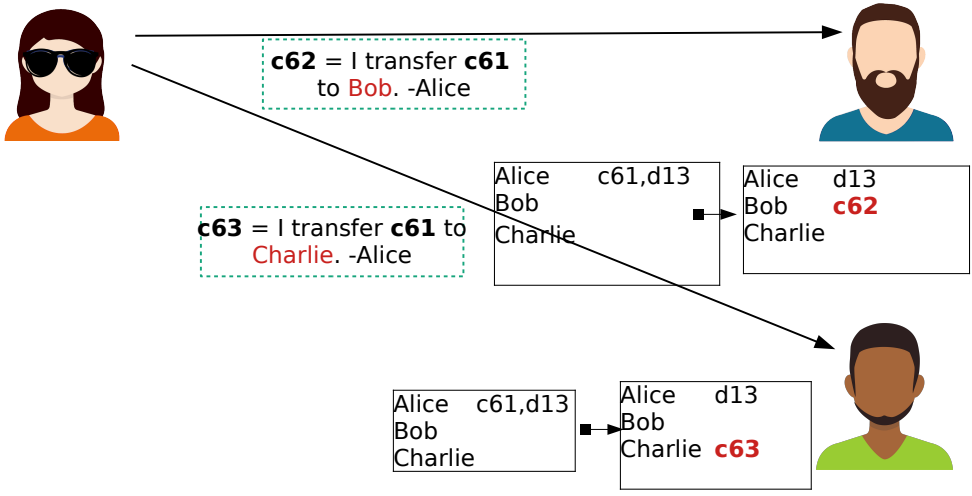
04: Transaction-protocol



Problem:

- Replay attacks are no longer possible due to the transaction history
- But credit balances can still be issued several times
- Attackers can send self-signed transaction multiple times, simultaneously or selectively which is the famous Double-Spending-Problem.
- Consensus is not assured

04: Transaction-protocol: Double-Spending-Problem



Solution:

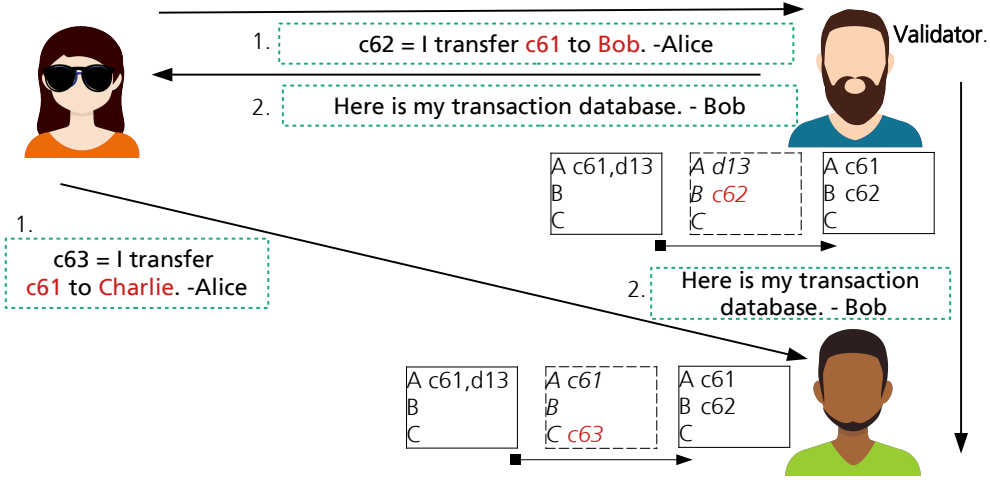
- It must be impossible to send transactions selective to different participants
- Each transactions sent, must reach all participants
- Transactions must be validated in advance
- Randomly selected “protocol validators” perform this task

Random selection of protocol validators

Rules:

- Each participant is a potential validator
- Participants update their transaction database not immediately
- Participants first collect all valid transactions in a so-called transaction pool (intermediate storage)
- After a temporary validator is randomly selected, its transaction pool is signed and sent to all participants
- Each participant then validates the signature and the received transactions of the validator
- Each participant discards its own transaction pool and updates its transaction database

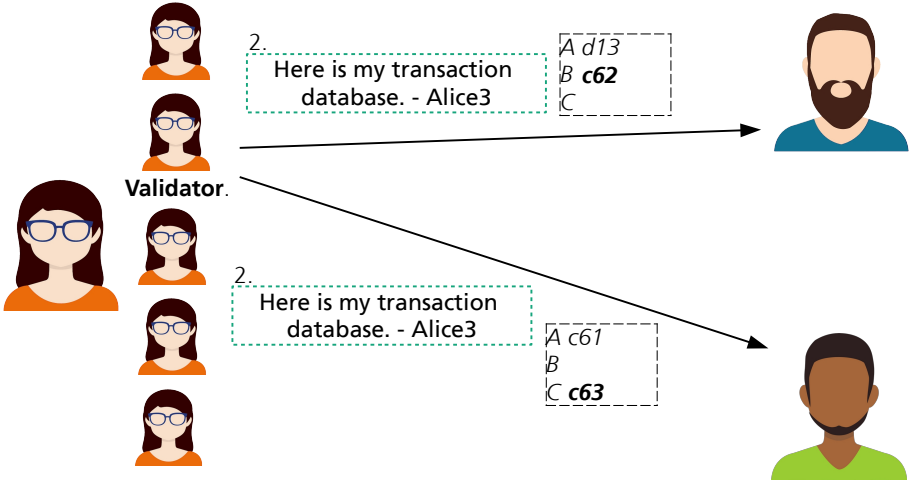
05: Validator-protocol



Problem:

- Restricted to evil validators, but still vulnerable to double-spending
- If the attacker becomes a validator himself, he can perform a double-spending attack
- Can the *random* selection of validators be influenced?
  - Yes, by doing a so-called Sybil-Attack
- Sybil-Attack:
  - Influencing a network by creating several false identities
  - An Attacker can execute the protocol-software on any number of network nodes and clone himself several times
  - Using this approach, the attacker can become the validator nearly all the time
  - Attacker possesses centralized power

05: Validator-protocol: Sybil-Attacke



Solution:

- Periodic selection of validators on the basis of unforgeable characteristics
- Need for identification features that can not be reproduced with little effort

06: Proof of work (PoW)-protocol

Identification using computing time performed: Proof of work (PoW)

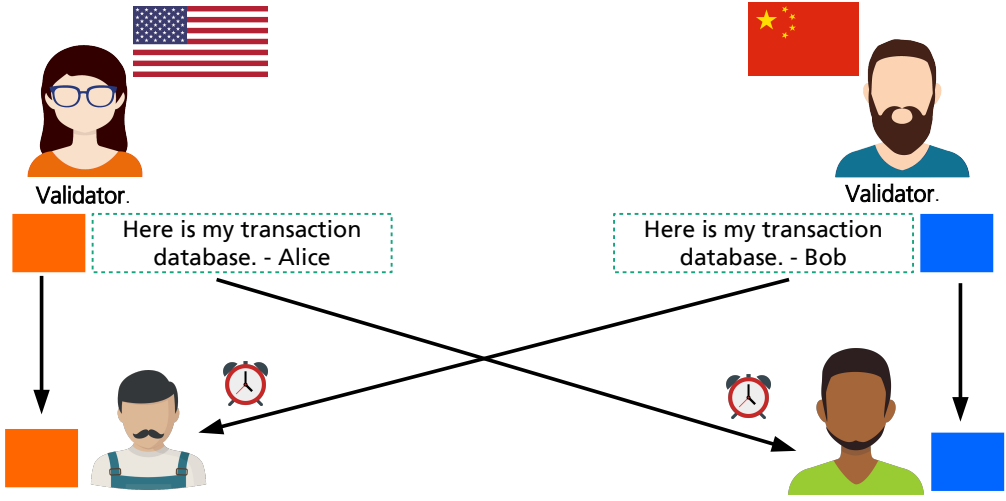
Rules:

- Each participant is a potential validator and collects valid transactions in its transaction pool
- Each participant works continuously on the so-called Hashcash puzzle
- Each participant uses its transaction pool as input value for the calculation
- As soon as a participant solves the puzzle, he becomes a validator and sends his transaction pool together with the solution to all other participants
- Each participant then validates the solution (Nonce) and the transactions received from the validator
- Each participant discards its transaction pool and updates its transaction database

Problem:

- Sybil-attack/Double-spending has become very costly
- In order to influence the selection process favourably, participants have to do more caculations (which requires energy, which requires money)
- The more participants are involved in the Proof of work (PoW) calculation, the less likely individual selection becomes
- A common competition is created - game theory
- What happens if several participants calculate a solution almost simultaneously?
  - No winner can be identified due to network latencies
- The network will split into incompatible sub-networks (Forks)
- Consensus is not assured

06: Proof of work (PoW)-protocol: Forks



Solution:

- Short-term acceptance of inconsistent data by forks
- Network must be able to reorganize consensus independently

07: Blocklist-protocol

Chronological sorting of Proof of work (PoW)-protocol and transaction data in units of blocks using a hash-linked list.

Rules:

- Each participant is a potential validator and collects valid transactions in its transaction pool
- Periodic selection by computing a so-called Proof of work (PoW)
- Each participant uses its transaction pool and the hash value of the previous block as input value for the calculation
- As soon as a participant solves the calculation, he becomes a validator and forms a block consisting of:
  - Transaction pool + hash value of previous block + solution (nonce)
- The validator sends his block to all other participants
- Each participant then validates the block of the validator
- Each participant discards its transaction pool and updates its transaction database

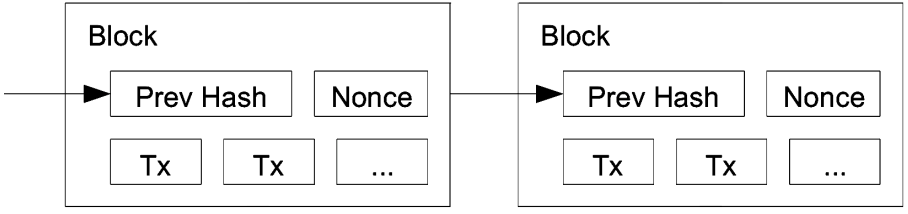


Figure from: Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008

Advantages using these hash-linked block list:

- Blocks have a fixed chronological order
- Previous data inside the blocks cannot be manipulated unnoticed
- Each new block implicitly confirms all previous blocks and thus also all previous transactions
- Each block secures all previous transactions through Proof of work (PoW)
- Therefore each block is considered a transaction confirmation
- The longest block list is the only valid one

07: Blocklist-protocol

1.

2.

3.

4.

5. ...

6. ...

Fork

Reorganisation

...

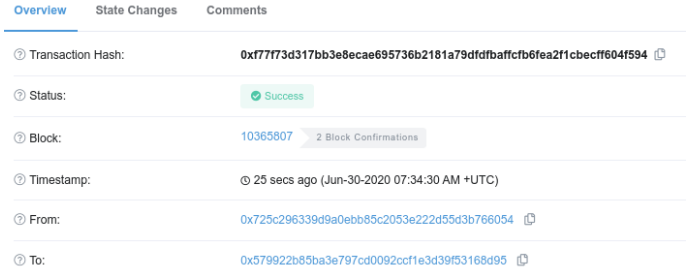
Problem:

- The network can now reorganize itself
- But Forks are still possible and not unlikely
- Therefore indirect (valid for specific amount of time/blocks) double-spends are possible
- What does this mean for the validity of a transaction?
- When should a transaction considered to be final or confirmed?

07: Blocklist-protocol

Solution:

- The probability of Forks must be further reduced
- It must be ensured for a transaction not to be rejected by reorganizations
- To confirm a transaction safely, a certain time must be waited
- Each subsequent block increasingly secures previous transactions
- After a certain number of subsequent blocks, a transaction is considered final or confirmed



[3]

08: Blocktime-protocol

Introduction of further Parameters like block size and interblock time to reduce the probability of Forks.

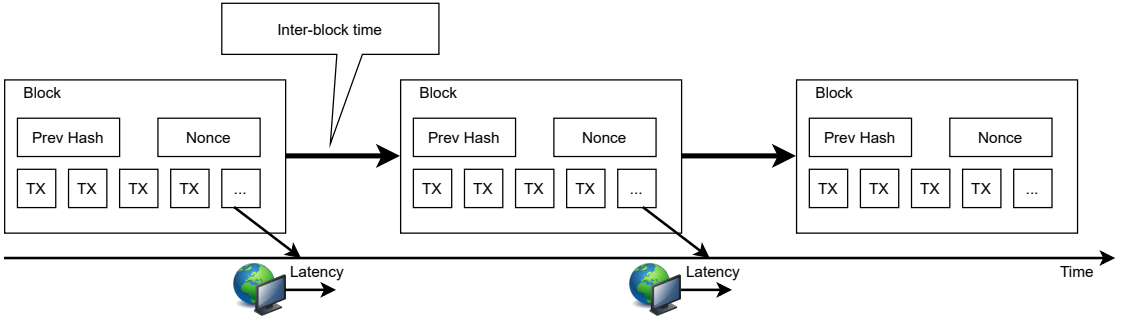
Rules:

- 07: Blocklist-protocol + ...
- Eeach block must reach all participants as quickly as possible
  - Short block propagation time
- Each block must not exceed a certain total data size
  - Large amounts of data propagate more slowly
- Each block must have a comparable Proof of work (PoW) difficulty
  - The more blocks in a particular list, the more computing power has been used for this list

08: Blocktime-protocol

The greater the inter-block time in relation to the block propagation time, the less likely is the repetitive formation of Forks.

If the block chain parameters (block size and inter-block time) are chosen sensibly, network Forks can reorganize itself and finally find a consensus.



Problem:

- The inter-block time is defined by selecting the Proof of work (PoW) difficulty
- For a stable network the inter-block time should be longer than the block propagation time
- How can the inter-block time be guaranteed with dynamic network capacity?
- The more participants try to solve the calculation, the faster it will be found, the shorter will be the inter-block time

Solution:

- Introduction of a flexible and relative proof-of-work difficulty

09: Adaptiv-protocol

Adaptive Proof of work (PoW) difficulty depending on the computing power of the entire network.

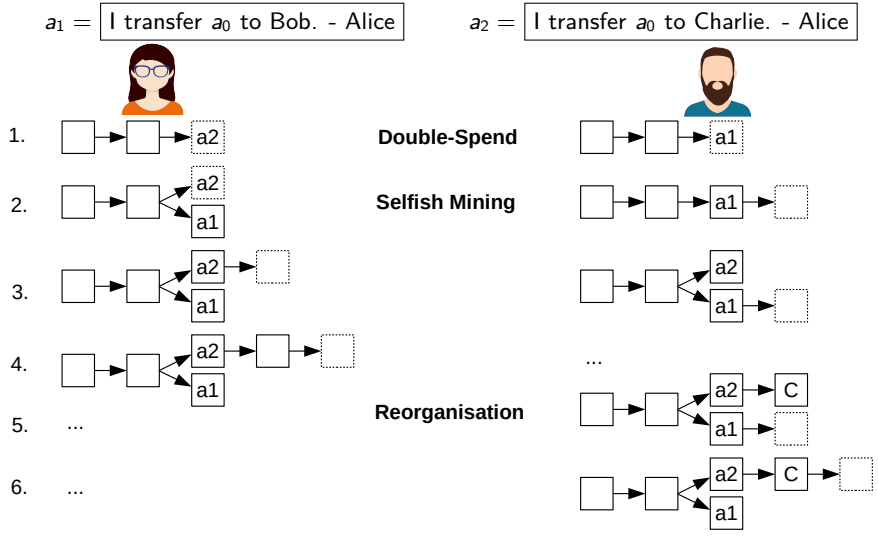
Rules:

- 08: Blocktime-protocol + ...
- The proof-of-work difficulty periodically adjusts to the network capacity
- For each block it is checked whether the protocol requires an adjustment of the difficulty

Problem:

- Low probability of Forks due to guaranteed inter-block time and maximum block size
- Therefore low probability for indirect double-spending
- But double-spending is not 100% impossible
- An attacker can act selfish and deviate from the given protocol
- Through so-called Selfish-Mining, an attacker with enough computing power can still successfully conduct a double-spending
- The attacker works in secret on its own Fork of the block list. As soon as the attacker received the product payed with the transaction the attacker publishes its own Fork (which needs to be longer)
- *Payable* double-spending attacks are possible by a rationally acting attacker
  - Question: How much do the power costs in comparison to the product?

09: Adaptiv-protocol



The probability of a successful double-spending attack depends on exactly two factors:

1. The speed in the competition
   - The computing power of the attacker in relation to the computing power of the entire network
2. The backlog to be made up
   - The number of subsequent blocks that the attacker must calculate to generate a longer block list

If an attacker has more than 50% of the computing power, he will ultimately win every competition and is always a validator if he wants to. Similar to the Sybil-Attacke, the attacker possesses through a so-called 51% attack centralized power.

Solution:

- 51%- and double-spending Attacks cannot be prevented
- However, they can be made disproportionately expensive and therefore uneconomic
- The computing power added to the network is compensated by rewards
- *Failed attackers will not receive this reward´*

High cost potential failed attacks due to loss of the so-called Block Rewards

Rules:

- 09: Adaptiv-protocol + ...
- Introduction of a reward for successfully calculated blocks (Block Reward)
  - Commercial incentive and game theory
- The more computation power a participant provides to the network, the higher is the probability to find a block and receive the Block Reward
  - Through a joint competition the overall computing power of the network grows
  - High overall computing power increases the cost of a double-spending attack
  - Double-spending attacks are no longer "*payable*"

Bitcoin: A Peer-to-Peer Electronic Cash System

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

[11]

Costs of a 51% Attack

PoW 51% Attack Cost

This is a collection of coins and the theoretical cost of a 51% attack on each network.

Learn More | Tip

Name	Symbol	Market Cap	Algorithm	Hash Rate	1h Attack Cost
Bitcoin	BTC	$168.43 B	SHA-256	127,674 PH/s	*$408,251*
Ethereum	ETH	$25.33 B	Ethash	173 TH/s	*$143,002*
BitcoinCashABC	BCH	$4.12 B	SHA-256	2,486 PH/s	*$7,949*
BitcoinSV	BSV	$2.94 B	SHA-256	2,140 PH/s	*$6,844*
Litecoin	LTC	$2.69 B	Scrypt	243 TH/s	*$14,810*

[1]

Selected current blockchain topics

Selected current blockchain topics

- Alternatives and further development of consensus mechanisms
- Smart Contracts
- Layer-2 scalability
- Decentralized finance - DeFi
- Non-fungible Tokens (NFT)

Alternatives and further development of consensus mechanisms

- Proof of Stake (PoS) (stake = share)
  - The creator of the next block is selected by his stake (age AND number of currency)
  - No computing power required as with Proof of work (PoW)
  - Motivation: If you own a lot of the currency, you also want everything in the protocol to run correctly..
  - Main problem: The “nothing-at-stake” problem: A user can participate in all forks for free. It costs him nothing. Whether he harms others by doing so does not matter to **him**

Smart Contracts

- Vision: Enforce contracts **automatically**, **trustless** and **impartial**. No intermediary (e.g. notary) is needed
- Well-known example: ticket vending machine
- Smart contracts on the blockchain are (typically) executed by every node. **There is therefore consensus on the outcome**
  - Supply chain management: goods arrived, amount paid automatically
  - Digital voting and decision making

Digitaler Corona-Impfpass: IBM, Ubirch und fünf Blockchains

Die Bundesregierung hat IBM und der Firma Ubirch den Zuschlag für die Entwicklung eines digitalen Impfpasses gegeben.

Lesezeit: 1 Min. 316

(Bild: FabrikaSimf/Shutterstock.com)

[5]

Layer-2 scalability

- Bitcoin: 7 transactions per second, Ethereum (1.0): 30 transactions per second
- **Alternative method to make a blockchain more scalable**
- Not all transactions are processed directly on the blockchain, but through another protocol
- Only the **results** of the **transactions** are recorded in compressed form on the blockchain
  - Example: Alice offers Bob her internet connection for 0.01 coins per megabyte
  - Alice and Bob agree on the terms in a Smart Contract
  - Both sign the smart contract
  - For every megabyte that Bob uses, he sends Alice a signed message outside the blockchain
  - In the end, Alice sends these messages with a single transaction to the smart contract and receives a payment from the smart contract (from Bob)

Decentralized finance - DeFi

- Typical banking services such as “loans,” and “currency exchanges” are regulated on a decentralized basis through a combination of various smart contracts. (so-called “lending protocols”) via modern Web interfaces [7]
- By eliminating the overhead costs of a bank, **attractive conditions** are possible
- Very common misunderstanding: DeFi is **not person-to-person lending**, but is realized through **“pools”**

Decentralized finance - DeFi

From a lender's perspective:

- Deposit of a cryptocurrency into a "pool"
- The interest rate is highly flexible based on current supply and demand
- Direct "use" of cryptocurrencies possible. Anonymous lending
- Risk diversification through "pool"

From a borrower's perspective:

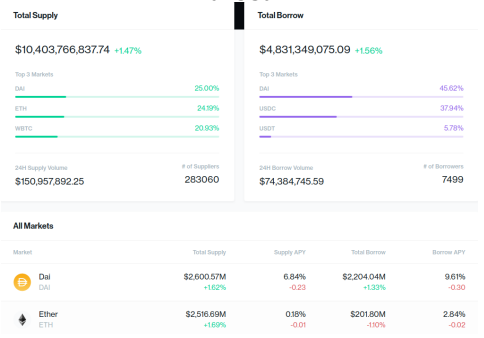
- Direct lending from the "pool"
- Security deposit "collateral" must be **prior** deposited (in the form of a cryptocurrency)
- The security deposit is usually significantly **higher** than the loan to compensate for price fluctuations in the field of cryptocurrency
- Direct payout without KYC or other checks
- If the loan is not repaid, the security deposit is automatically seized by the smart contract

DeFi - Dezentrale Finanzwirtschaft

Risks:

- The entire system/protocol becomes insolvent due to large price fluctuations
- Programming errors are found in the platform or in the smart contract (“hack“)

All a scam?



[8] (11.03.2021)

- 10 billion USD in a single smart contract
- Interest between 9% and 15%

Non-Fungible Tokens

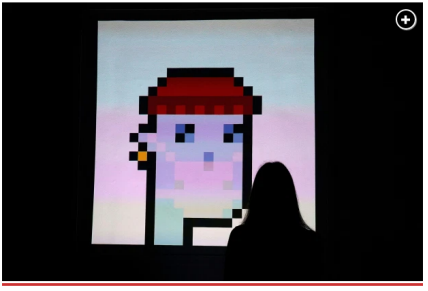
- Digital trading cards, game characters, virtual lands in virtual worlds or the so-called crypto art. The ownership is publicly announced via a smart contract

[2]

BUSINESS

'CryptoPunk' NFT sells for $11.8 million at Sotheby's

By Paula Froelich June 12, 2021 | 3:43pm | Updated



A woman looks at a NFT by Larva Labs titled "CryptoPunk 7523," during a media preview at Sotheby's.
AFP via Getty Images

[9]

Thank you for your attention.
Are there any questions left?

Room K331
Rathausallee 10
Technopark
Sankt Augustin

michael.rademacher@h-brs.de
www.mc-lab.de
https://michael-rademacher.net

References I

[1] Cost of a 51% attack for different cryptocurrencies | crypto51.
https://www.crypto51.app/.
(Accessed on 06/14/2021).

[2] Cryptopunks.
https://www.larvalabs.com/cryptopunks.
(Accessed on 06/14/2021).

[3] Ethereum (eth) blockchain explorer.
https://etherscan.io/.
(Accessed on 06/14/2021).

[4] Mr. robot - wikipedia.
https://en.wikipedia.org/wiki/Mr._Robot.
(Accessed on 06/15/2021).

[5] Borchers, D.
Digitaler corona-impfpass: Ibm, ubirch und fünf blockchains | heise online.
https://www.heise.de/news/Digitaler-Corona-Impfpass-IBM-Ubirch-und-fuenf-Blockchains-5076161.html.
(Accessed on 06/14/2021).

References II

[6] Bruennler, K.
Blockchain kurz und gut.
O'Reilly, Sebastopol, 2018.

[7] Chen, R.
What is defi? an introduction to decentralized finance – openzeppelin blog.
https://blog.openzeppelin.com/what-is-defi/, 2020.
(Accessed on 06/14/2021).

[8] Compound Labs, I.
Compound.
https://compound.finance/, 2021.
(Accessed on 06/14/2021).

[9] Froelich, P.
'cryptopunk' nft sells for $11.8 million at sotheby's.
https://nypost.com/2021/06/12/cryptopunk-nft-sells-for-11-8-million-at-sothebys/.
(Accessed on 06/14/2021).

[10] Kattwinkel, O., and Rademacher, M.
Technical fundamentals of blockchain systems.
BRSU Communication Report (2020).

References III

[11] Nakamoto, S.
Bitcoin: a peer-to-peer electronic cash system, oct. 2008.
URL http://www. bitcoin. org/bitcoin. pdf.(cited on pp. 15 and 87) (2017).

[12] Voshmgir, S.
Types of blockchains & dlts (distributed ledger technologies).
https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/.
(Accessed on 06/14/2021).