Blockchain decrypted

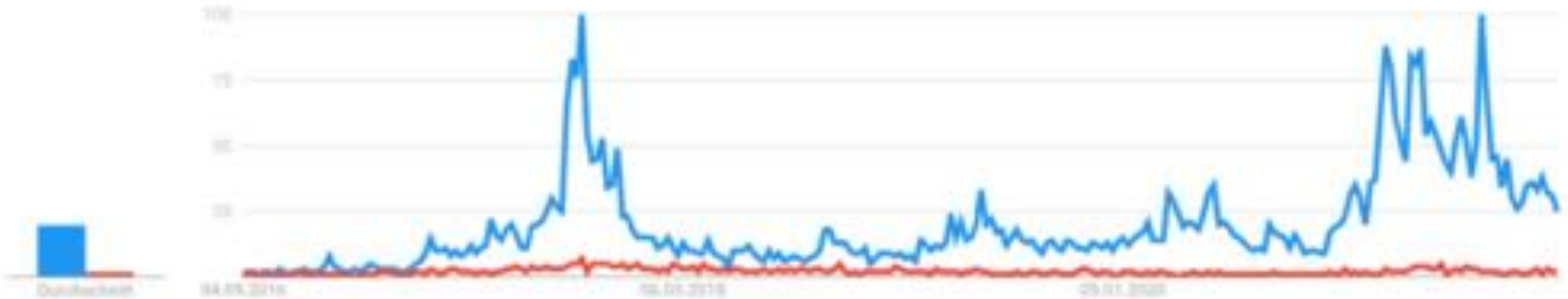
Bitcoin
Suchbegriff

Blockchain
Suchbegriff

+ Vergleich hinzufügen

Schweiz | Letzte 5 Jahre | Alle Kategorien | Websuche

Interesse im zeitlichen Verlauf

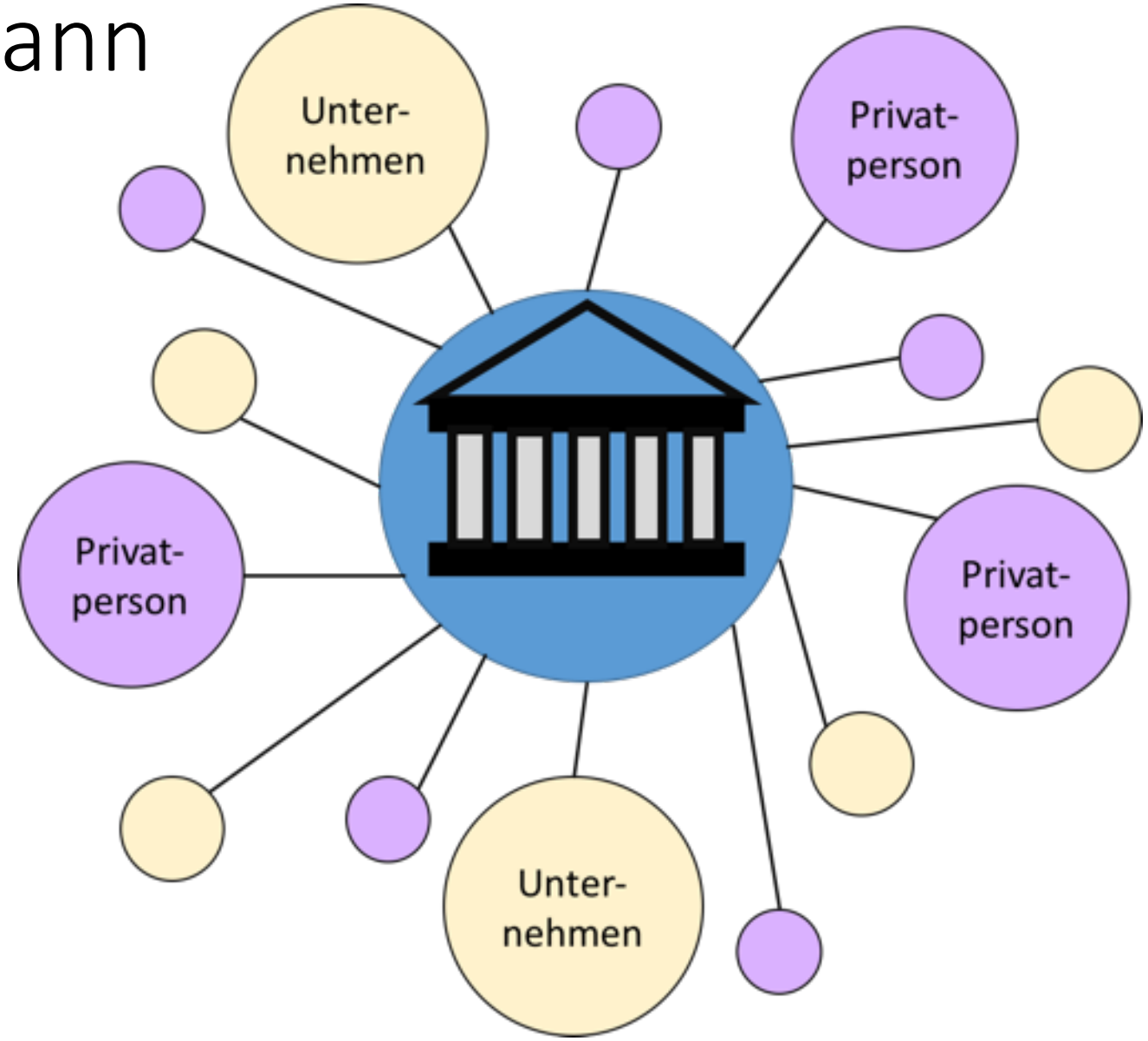



DECENTRALIZED
PEER TO PEER

Inhalt

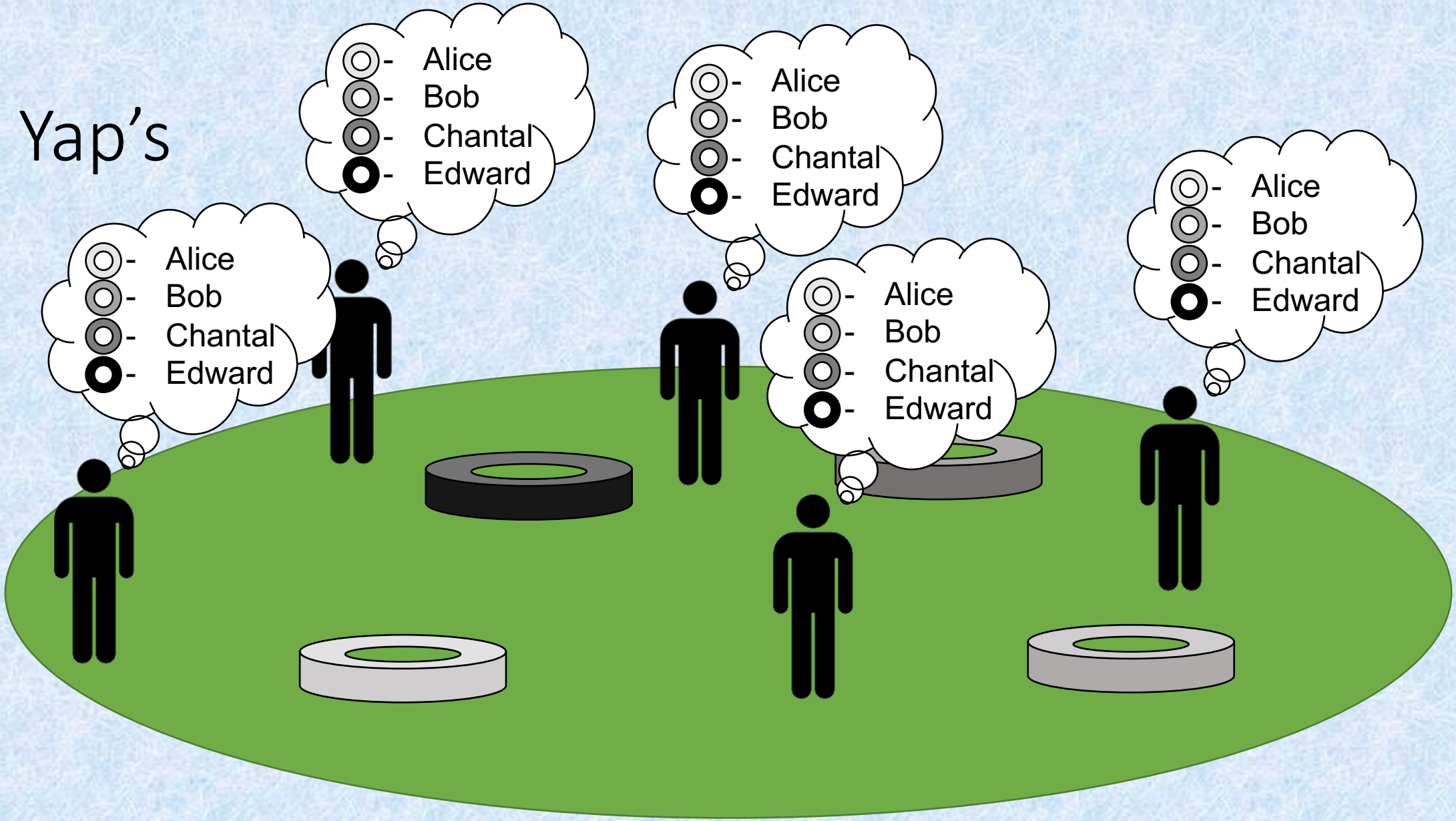
- Dezentrale Netzwerke
- Hash-Funktionen
- Funktionsweise der Blockchain
- Digital Signature Algorithmus
- Smart Contracts
- Anwendungen

Bank als Mittelsmann

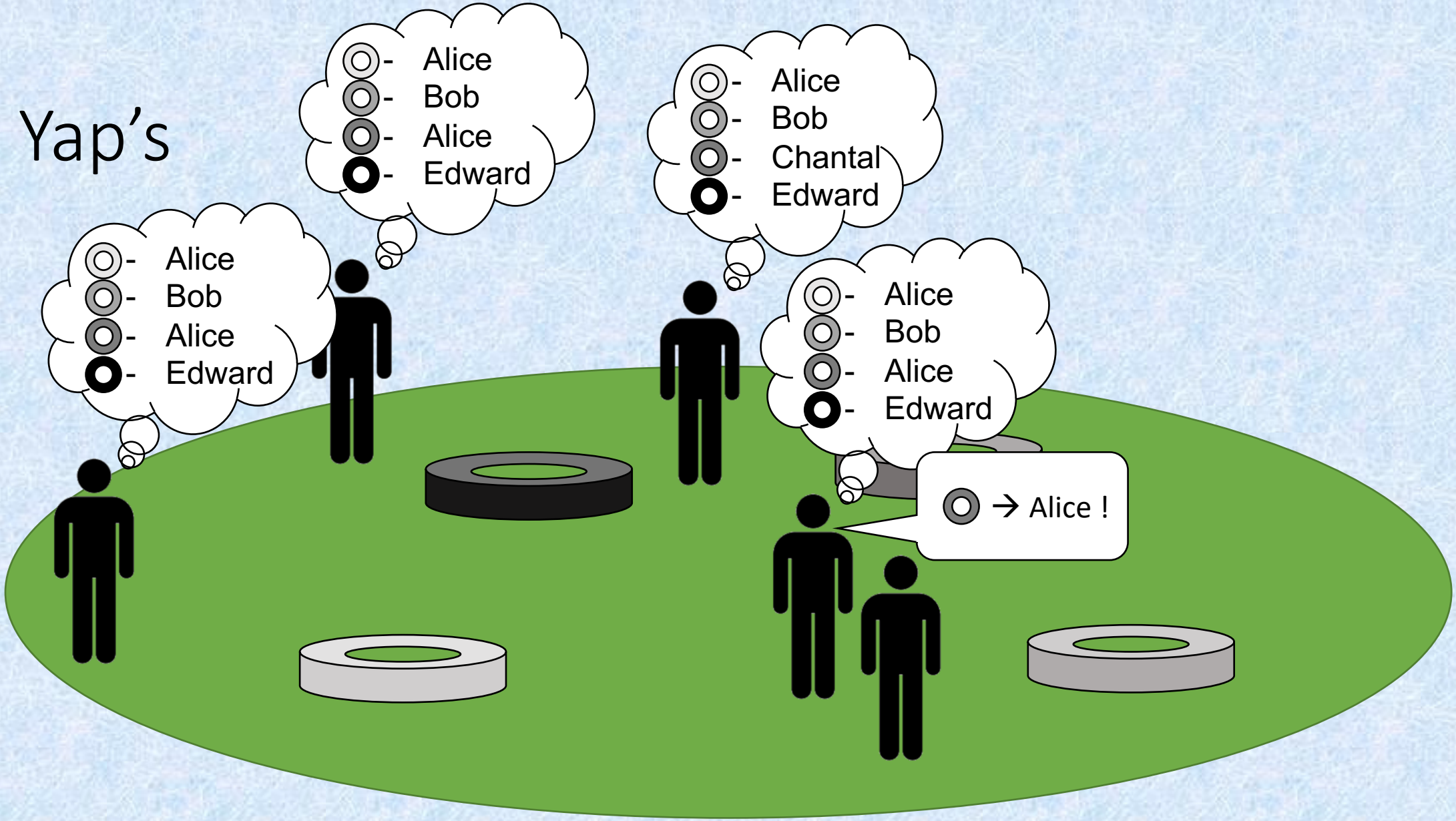


Dezentralisiertes Peer-to-Peer Netzwerk

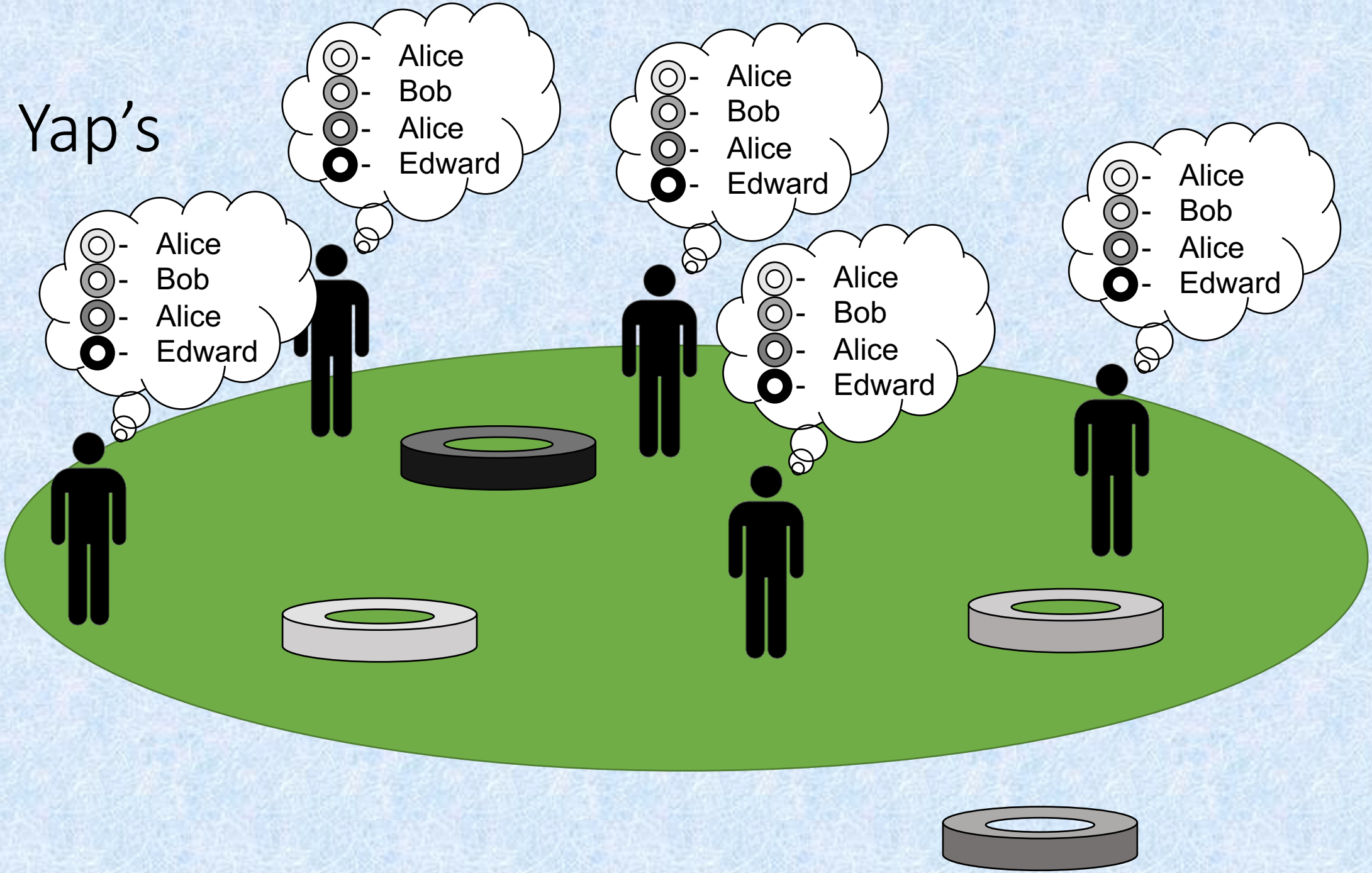
Yap's


- Alice
- Bob
- Chantal
- Edward
- Alice
- Bob
- Chantal
- Edward
- Alice
- Bob
- Chantal
- Edward
- Alice
- Bob
- Chantal
- Edward
- Alice
- Bob
- Chantal
- Edward

Yap's


- Alice
- Bob
- Alice
- Edward
- Alice
- Bob
- Chantal
- Edward
- Alice
- Bob
- Alice
- Edward
- Alice
- Bob
- Alice
- Edward
→ Alice !

Yap's


- Alice
- Bob
- Alice
- Edward
- Alice
- Bob
- Alice
- Edward
- Alice
- Bob
- Alice
- Edward
- Alice
- Bob
- Alice
- Edward
- Alice
- Bob
- Alice
- Edward

Hashfunktionen

Beispiele für Hashfunktionen

- TMU2020:

f6fff59256a3b4f189de4f7fe294379fb7250024f3d3178d73c9755279e3862e

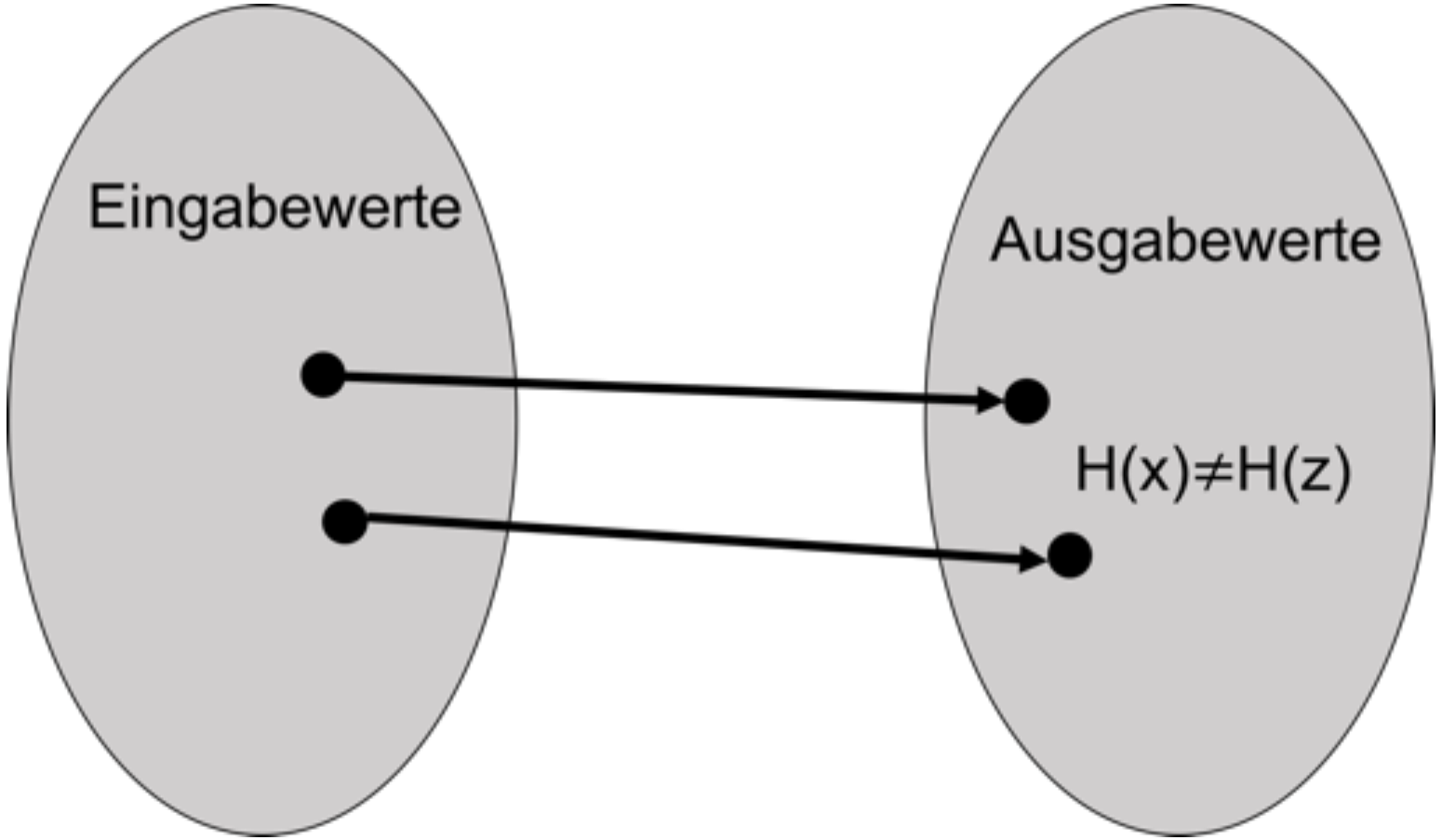
- TMU2021:

af15ff29faa859bb131e4b52b32a7a5f4f4100b57bd1679acc0fe203915a233f

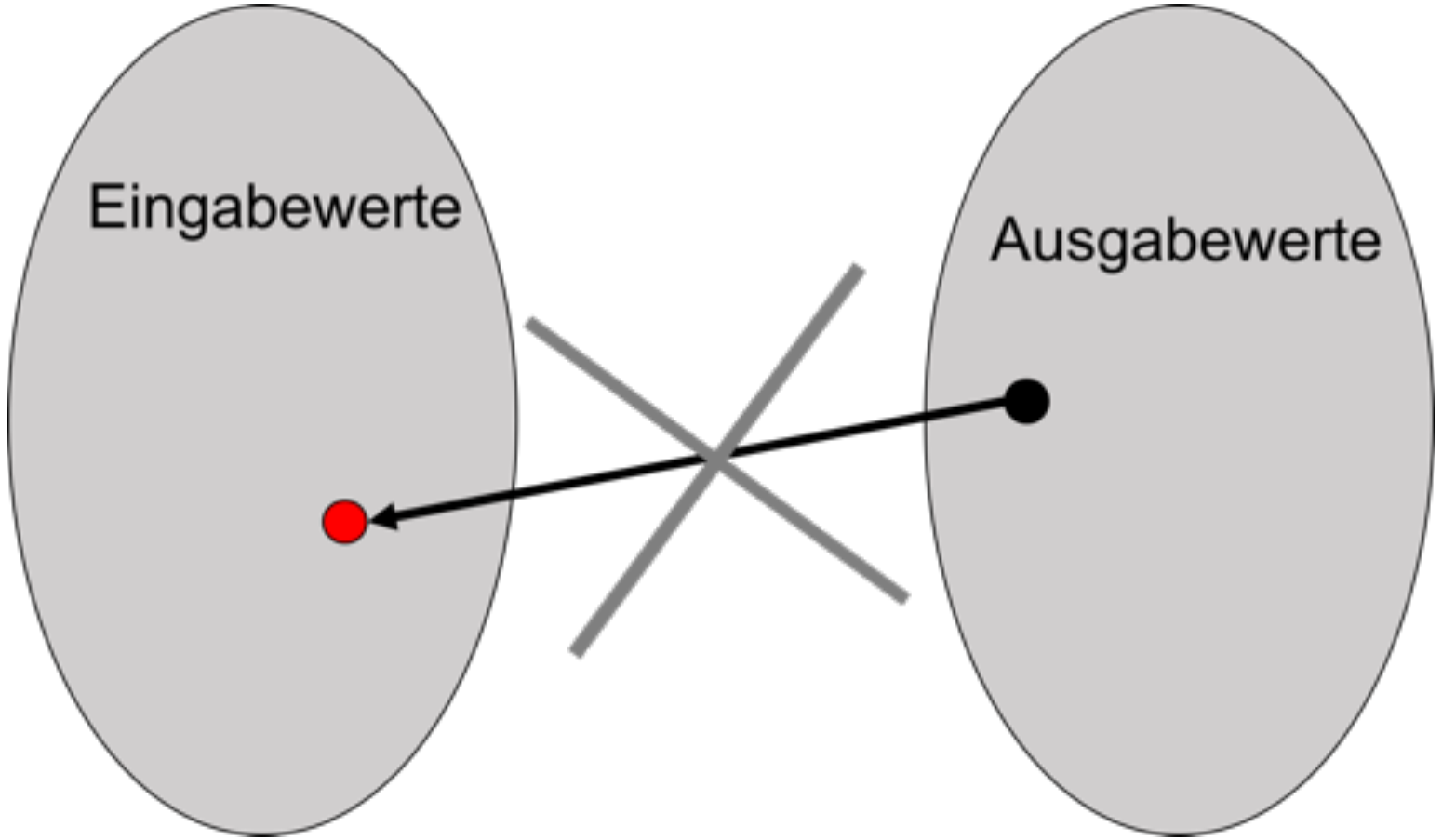
- Paper von Satoshi Nakamoto:

858cf328f81390d931ab296adac94aaf860c759fd54233b5d8eeaf581a866308

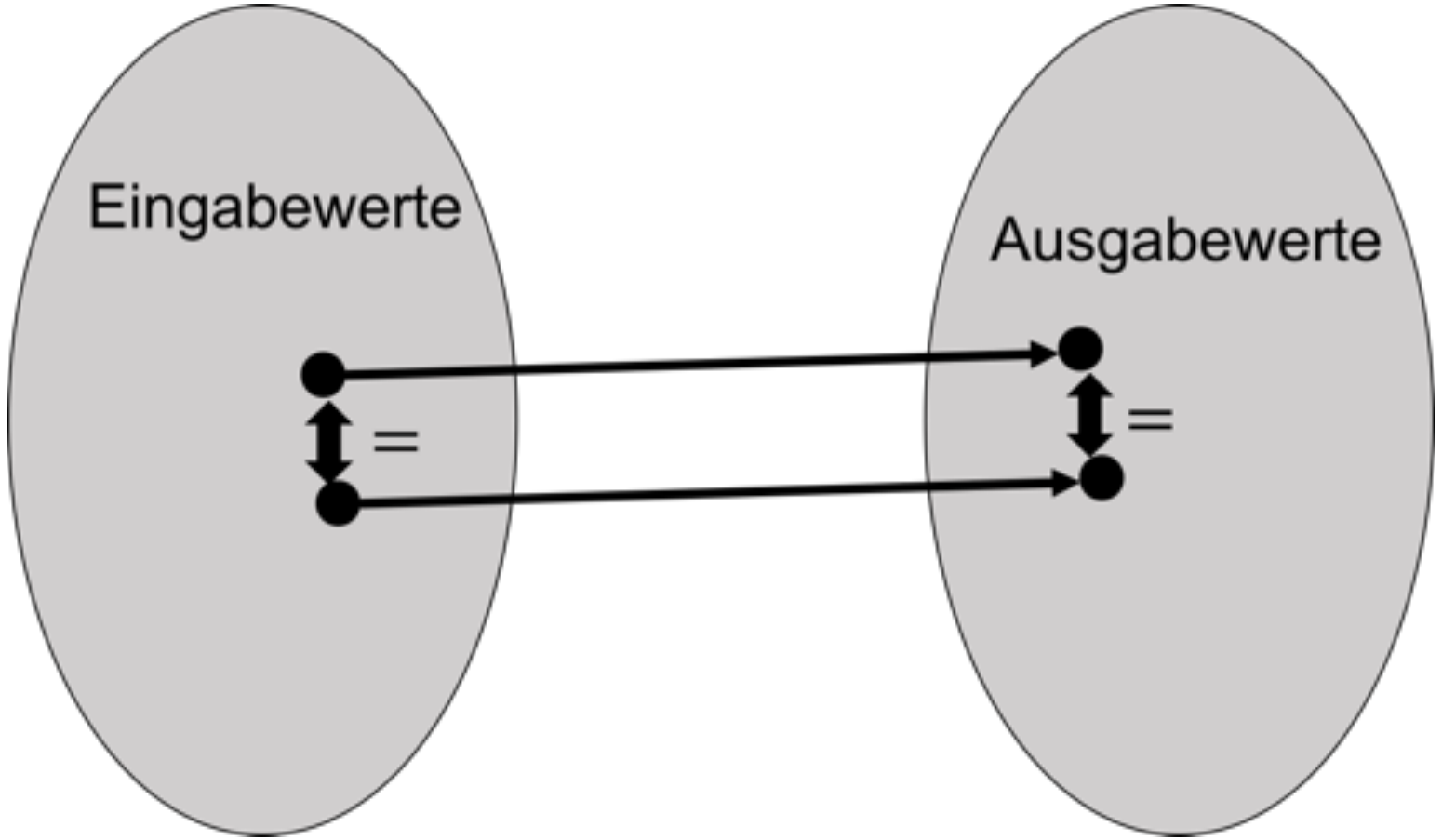
Hashfunktionen: Kollisionsfreiheit



Hashfunktionen: Einwegeigenschaft



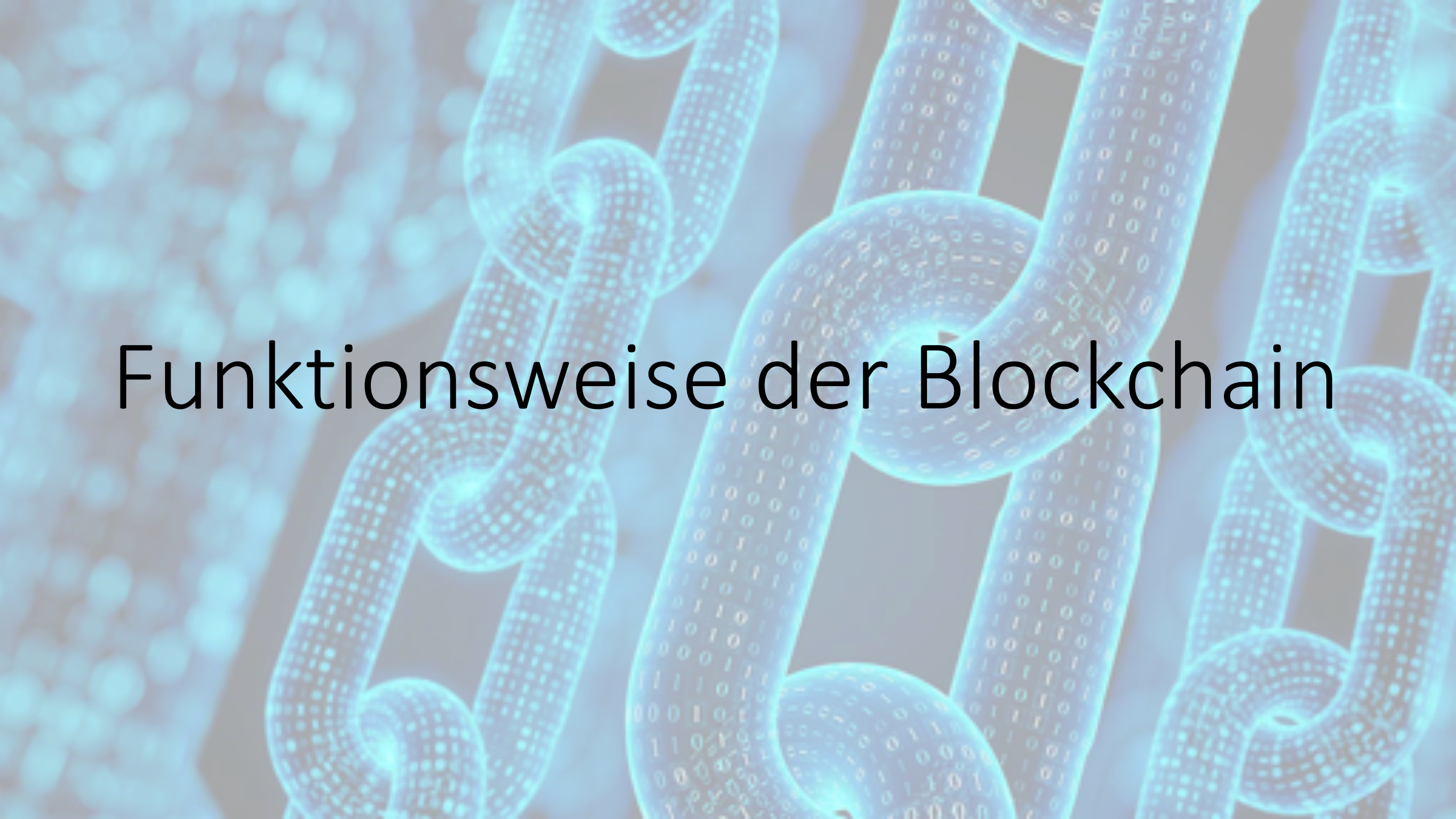
Hashfunktionen: Determiniertheit



Hashfunktionen - Pseudozufallsfunktionen

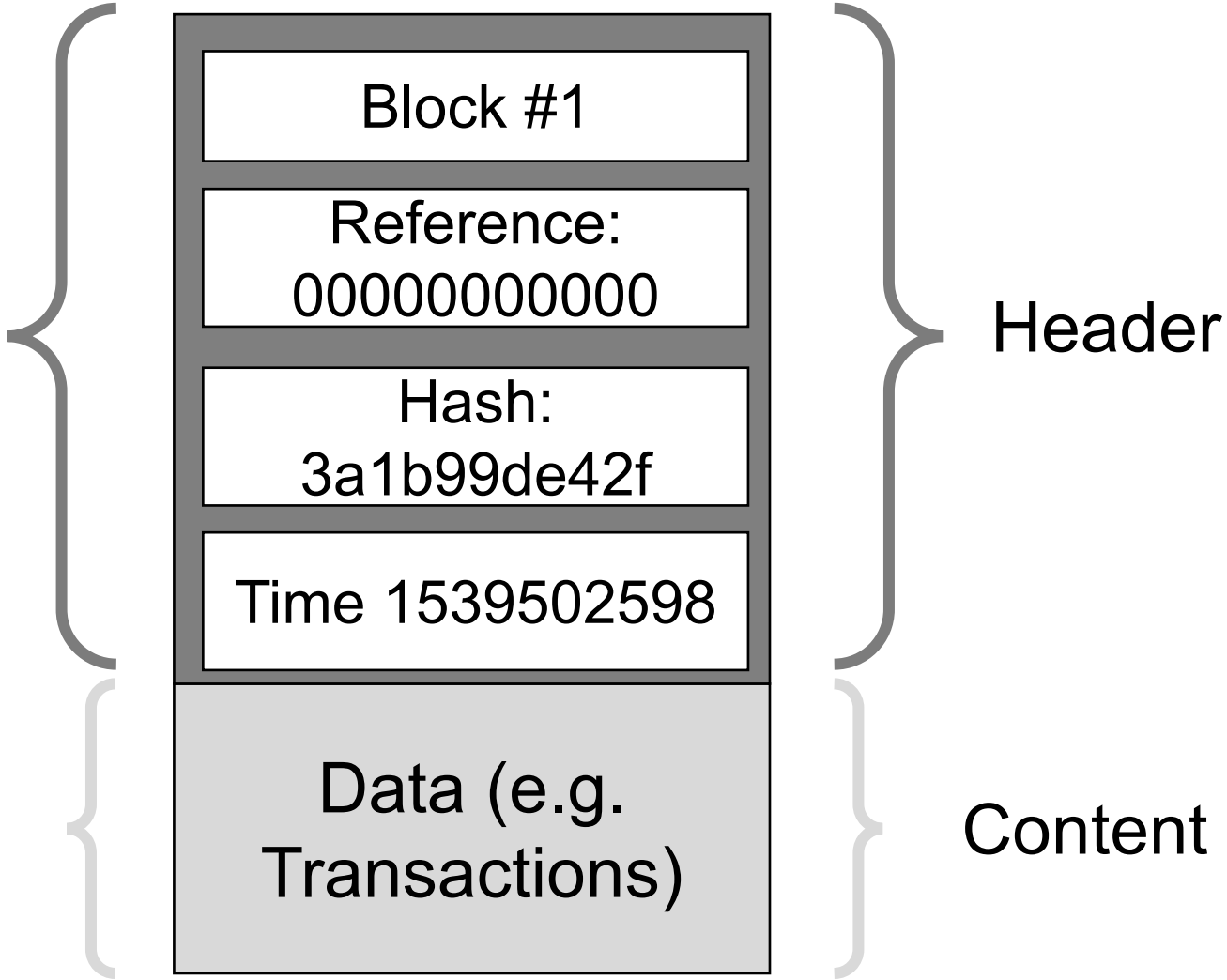
https://andersbrownworth.com/blockchain/hash

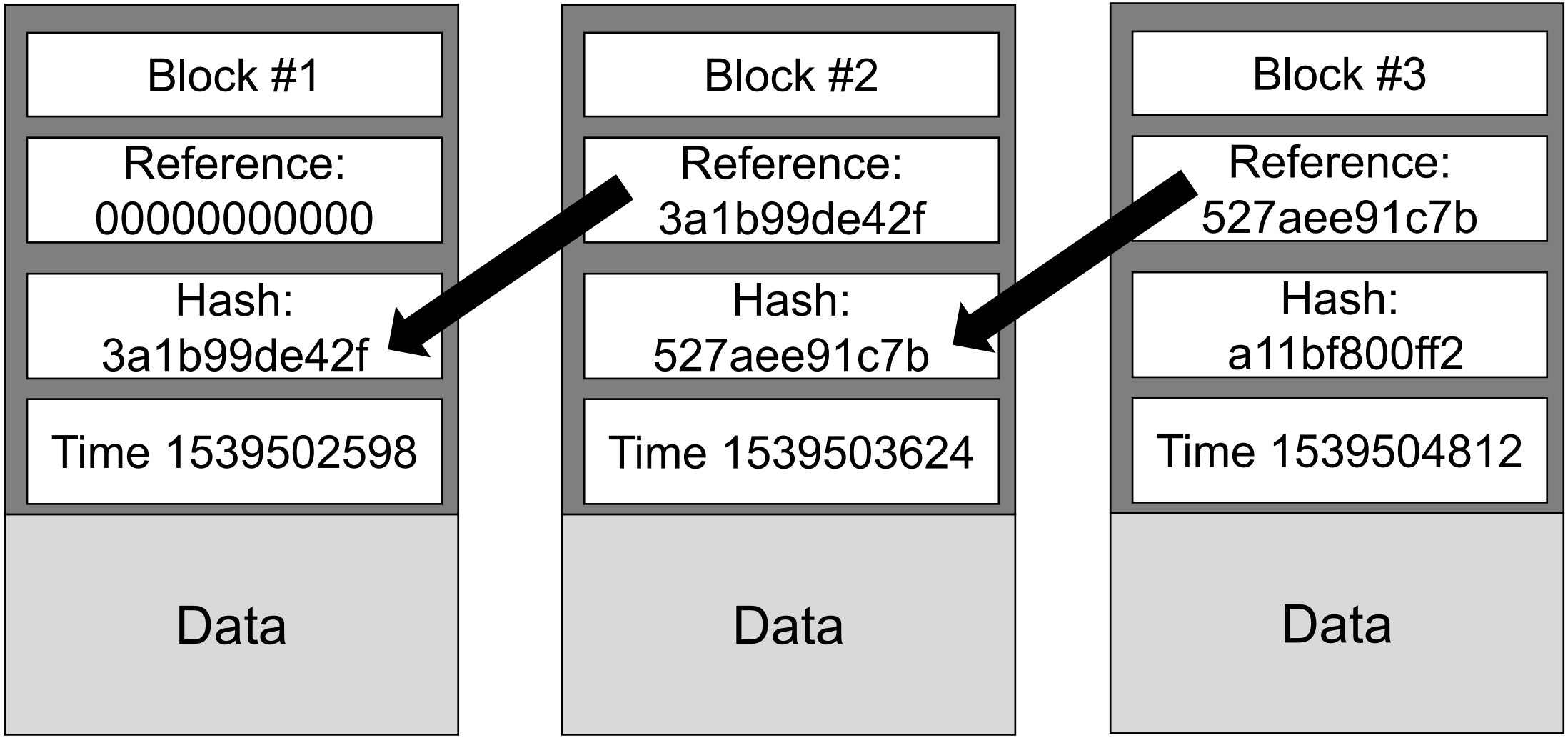
Hash-Rätsel

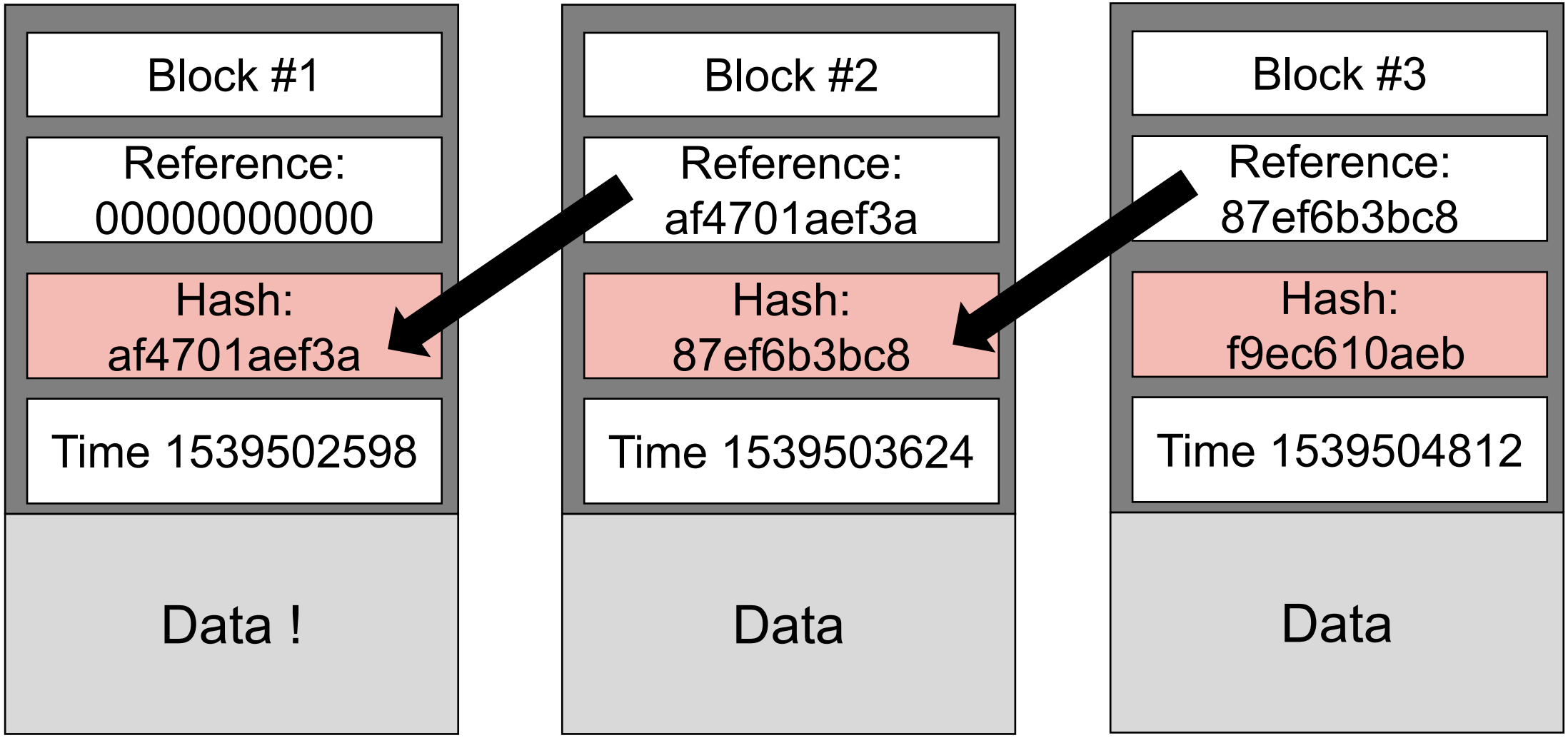


Funktionsweise der Blockchain

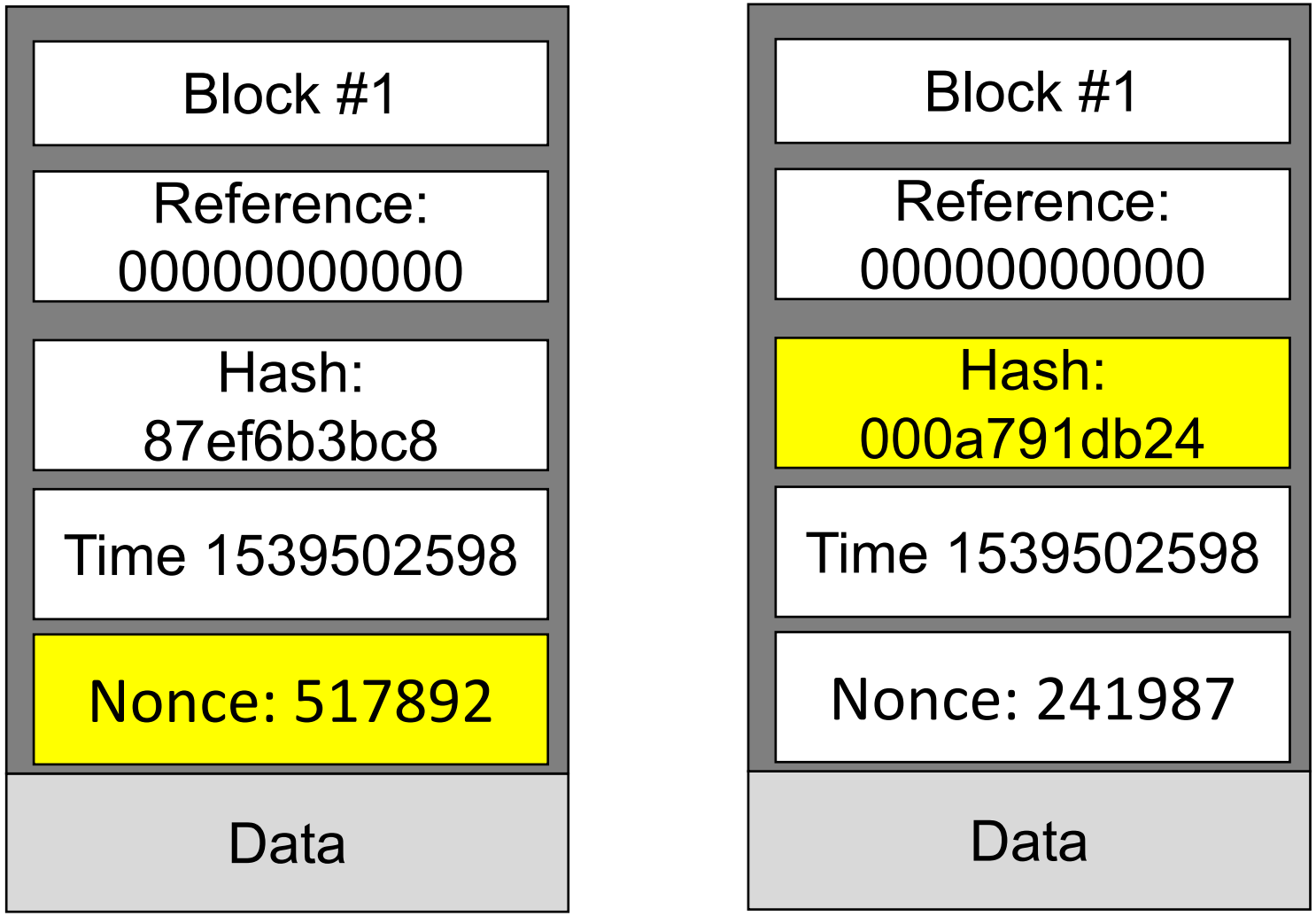
Blocks

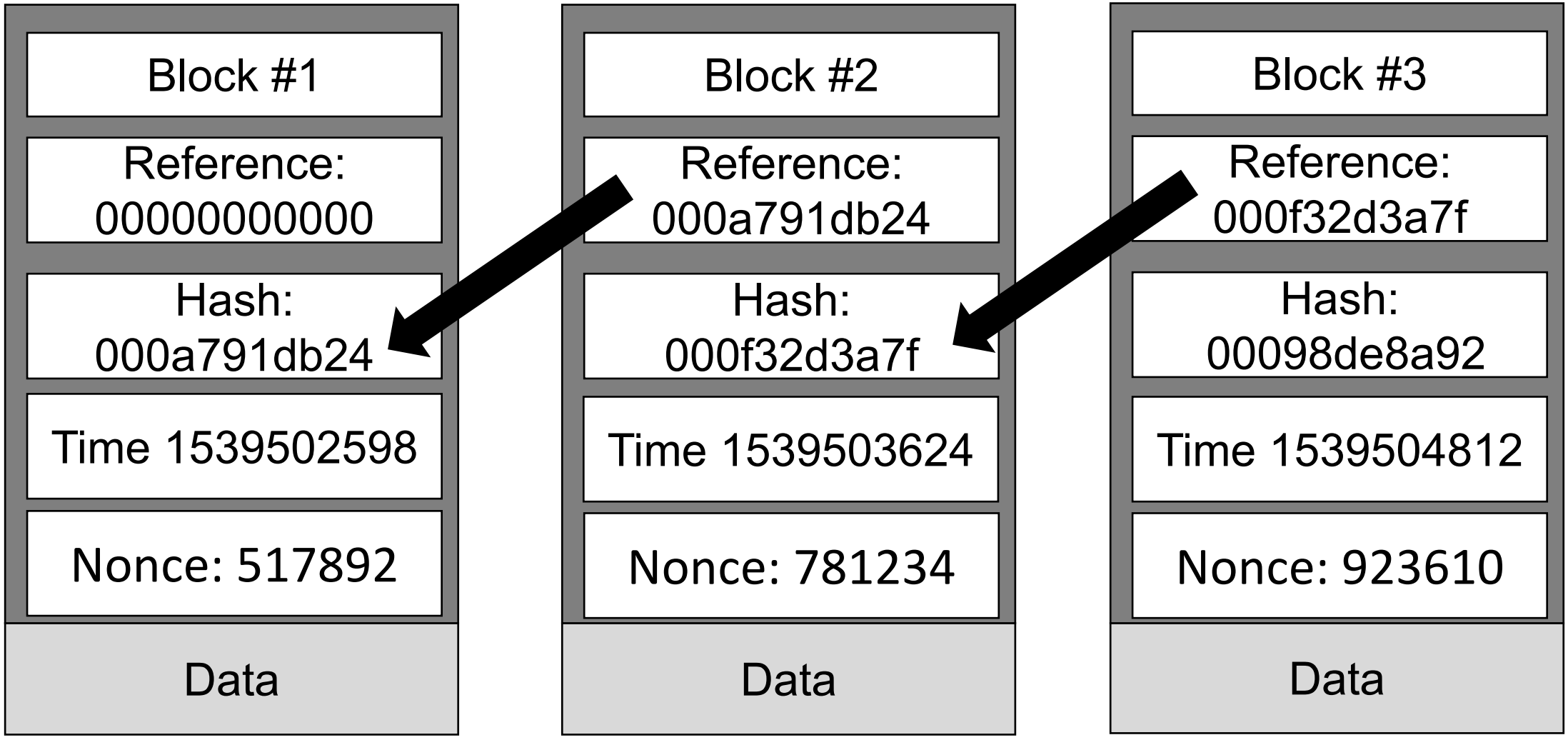

Block #1
Reference:
00000000000
Hash:
3a1b99de42f
Time 1539502598
Data (e.g.
Transactions)
Header
Content

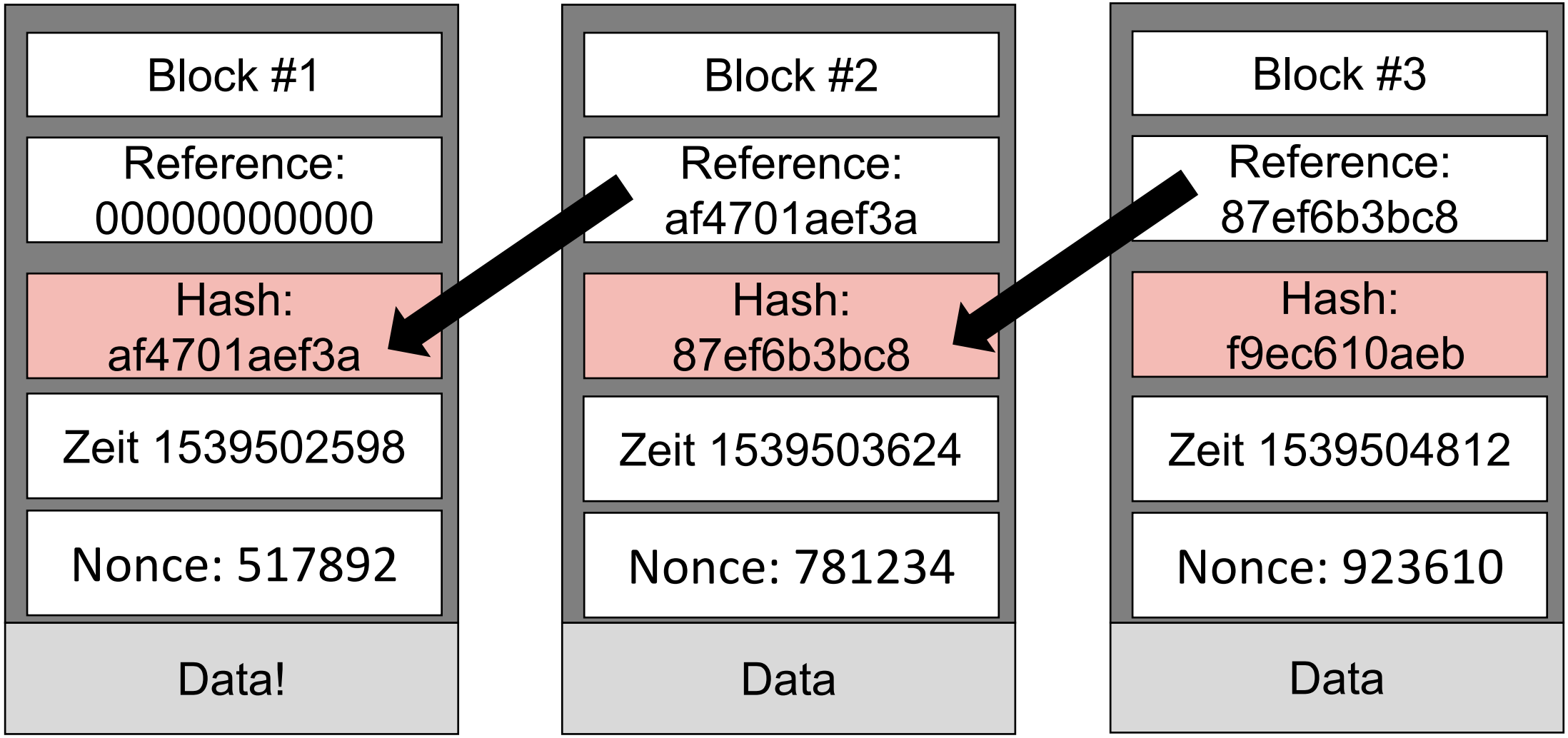

Block #1
Reference:
00000000000
Hash:
3a1b99de42f
Time 1539502598
Data
Block #2
Reference:
3a1b99de42f
Hash:
527aee91c7b
Time 1539503624
Data
Block #3
Reference:
527aee91c7b
Hash:
a11bf800ff2
Time 1539504812
Data


Block #1
Reference:
00000000000
Hash:
af4701aef3a
Time 1539502598
Data !
Block #2
Reference:
af4701aef3a
Hash:
87ef6b3bc8
Time 1539503624
Data
Block #3
Reference:
87ef6b3bc8
Hash:
f9ec610aeb
Time 1539504812
Data

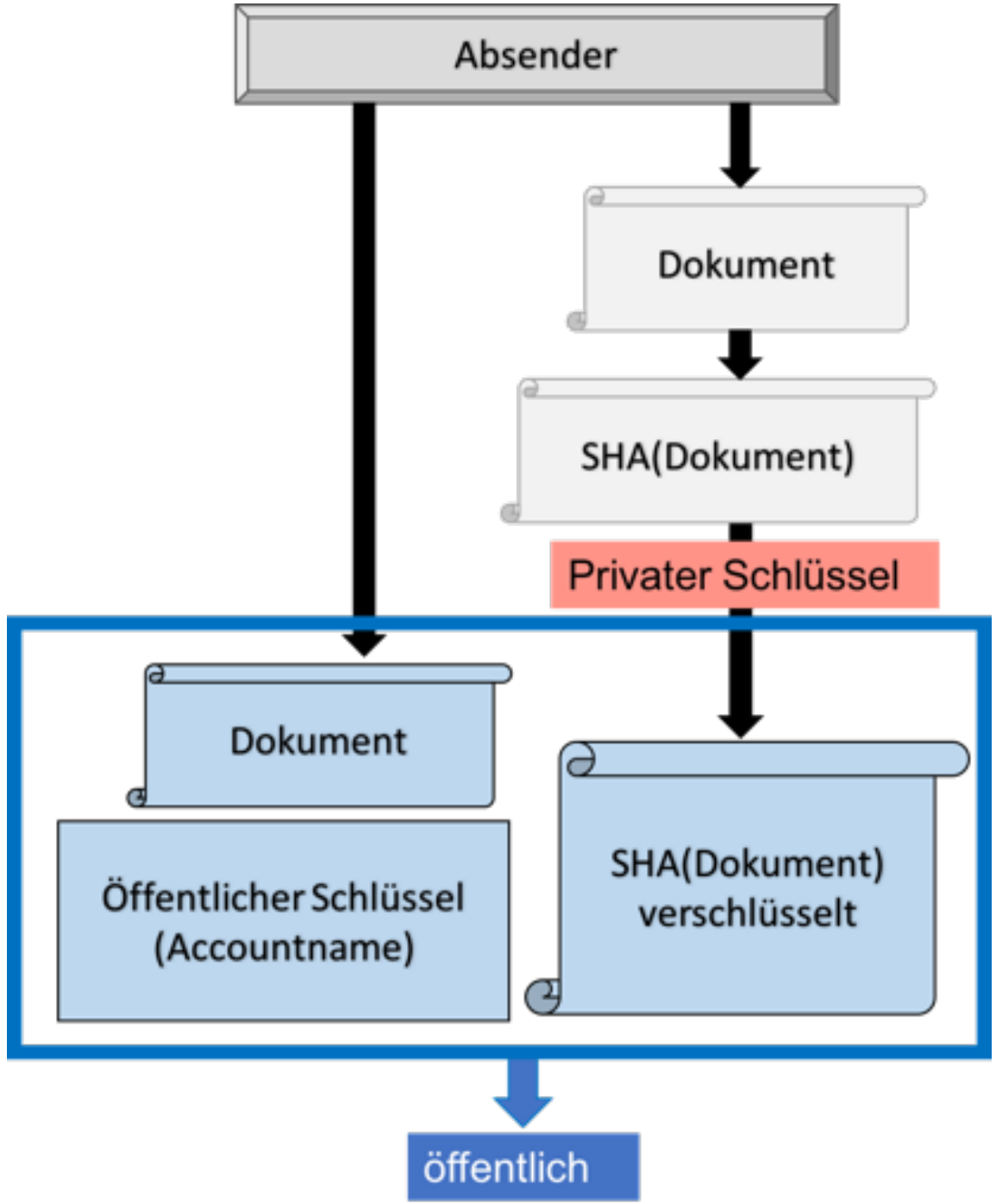
Proof of work

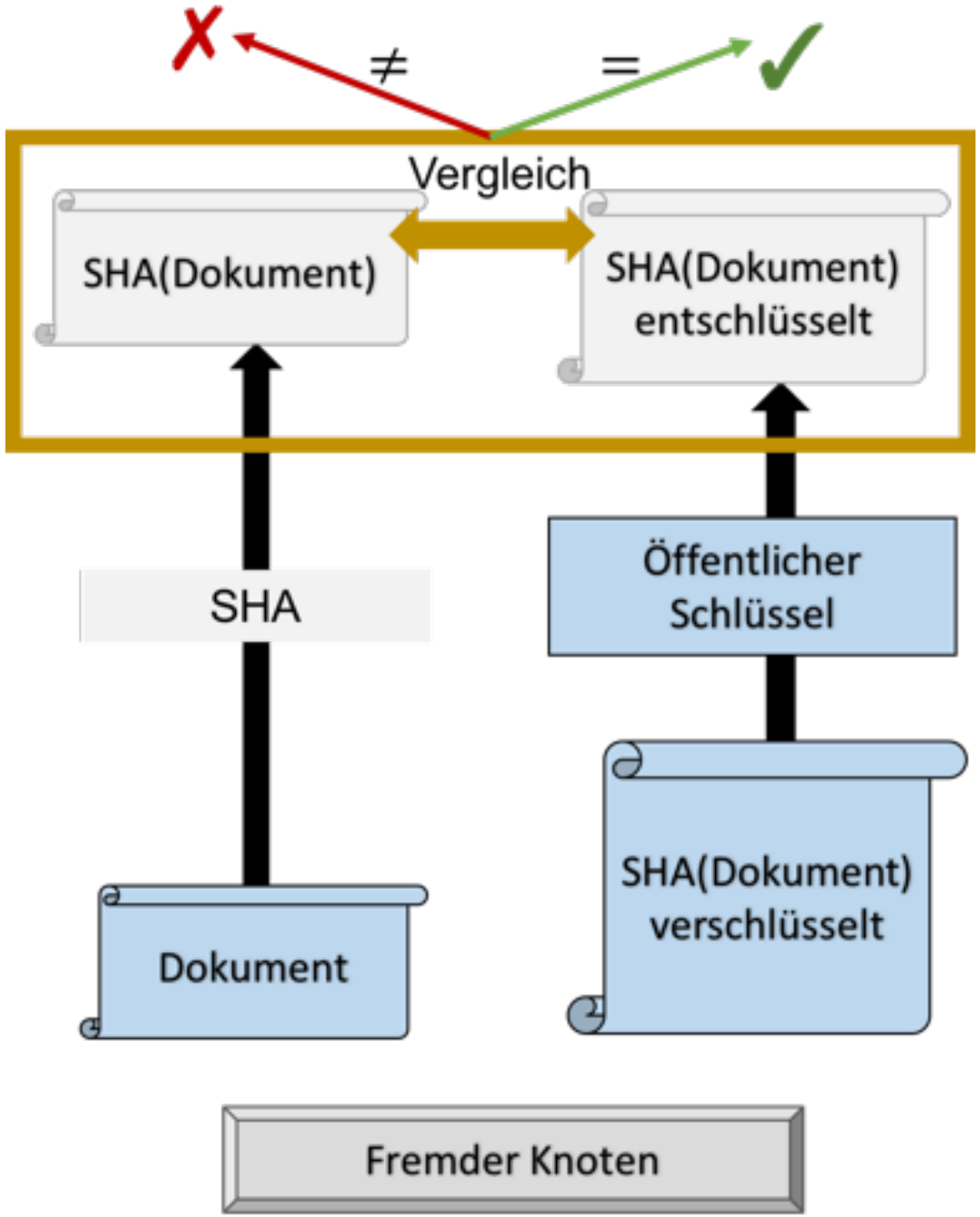



Block #1
Reference:
00000000000
Hash:
000a791db24
Time 1539502598
Nonce: 517892
Data
Block #2
Reference:
000a791db24
Hash:
000f32d3a7f
Time 1539503624
Nonce: 781234
Data
Block #3
Reference:
000f32d3a7f
Hash:
00098de8a92
Time 1539504812
Nonce: 923610
Data


Block #1
Reference: 00000000000
Hash: af4701aef3a
Zeit 1539502598
Nonce: 517892
Data!
Block #2
Reference: af4701aef3a
Hash: 87ef6b3bc8
Zeit 1539503624
Nonce: 781234
Data
Block #3
Reference: 87ef6b3bc8
Hash: f9ec610aeb
Zeit 1539504812
Nonce: 923610
Data

DSA-Algorithmus


Absender
Dokument
SHA(Dokument)
Privater Schlüssel
Dokument
Öffentlicher Schlüssel
(Accountname)
SHA(Dokument)
verschlüsselt
öffentlich

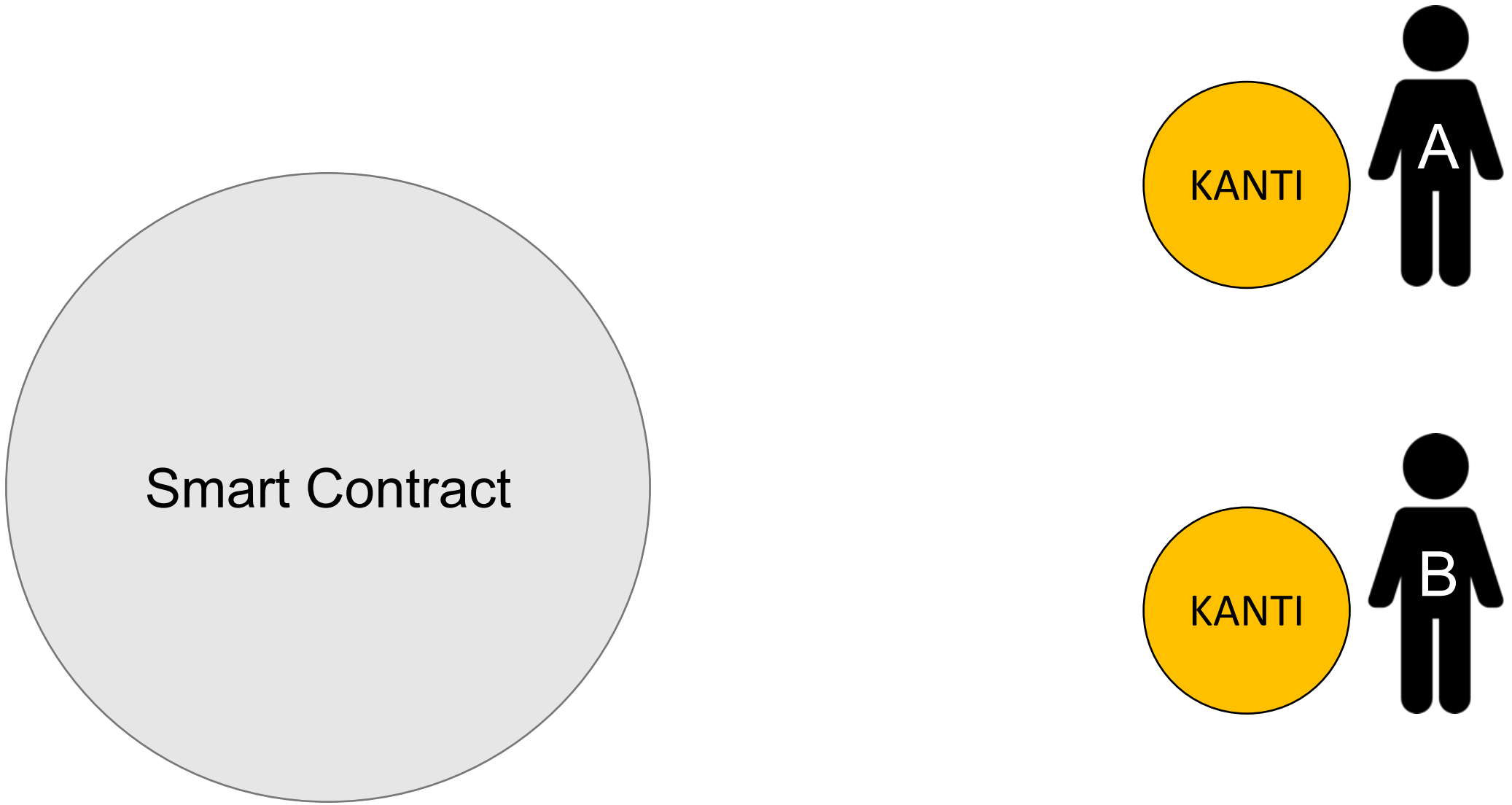

≠
=
Vergleich
SHA(Dokument)
SHA(Dokument)
entschlüsselt
SHA
Öffentlicher
Schlüssel
Dokument
SHA(Dokument)
verschlüsselt
Fremder Knoten

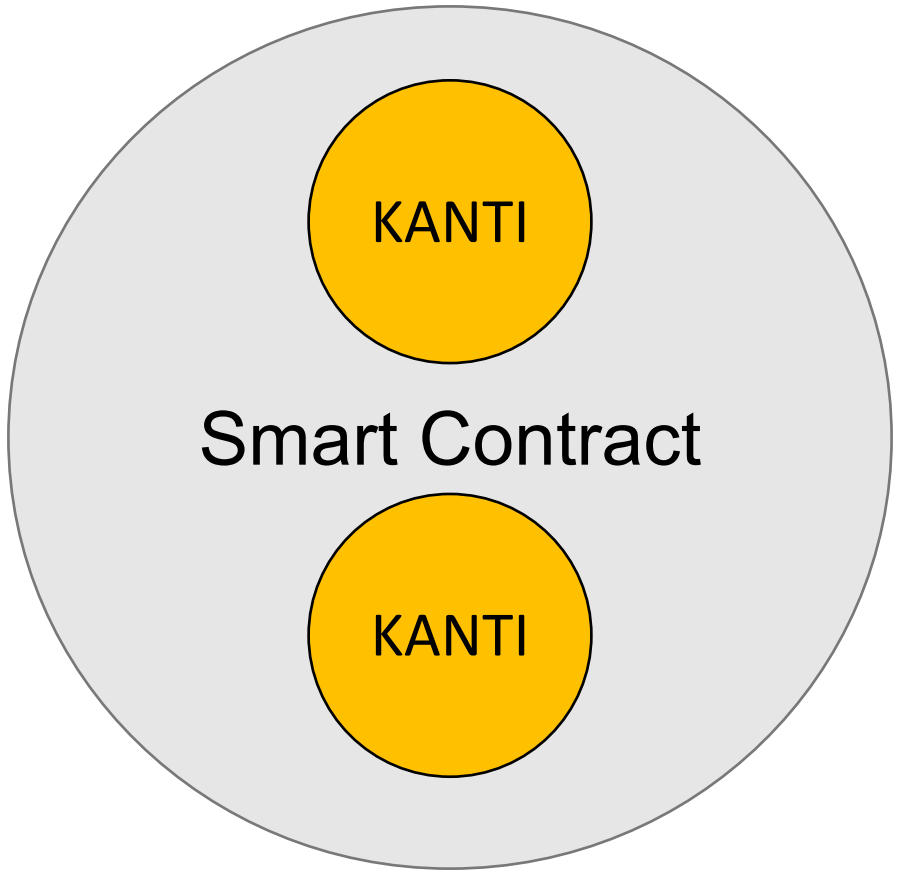
Smart Contracts

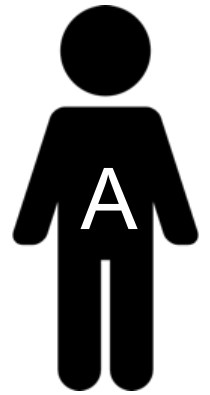
SMART CONTRACT

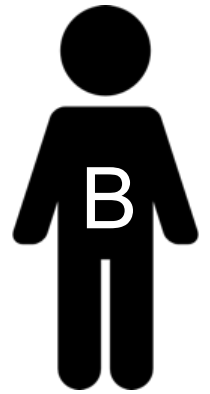
PARTIES SMART CONTRACT EXECUTION

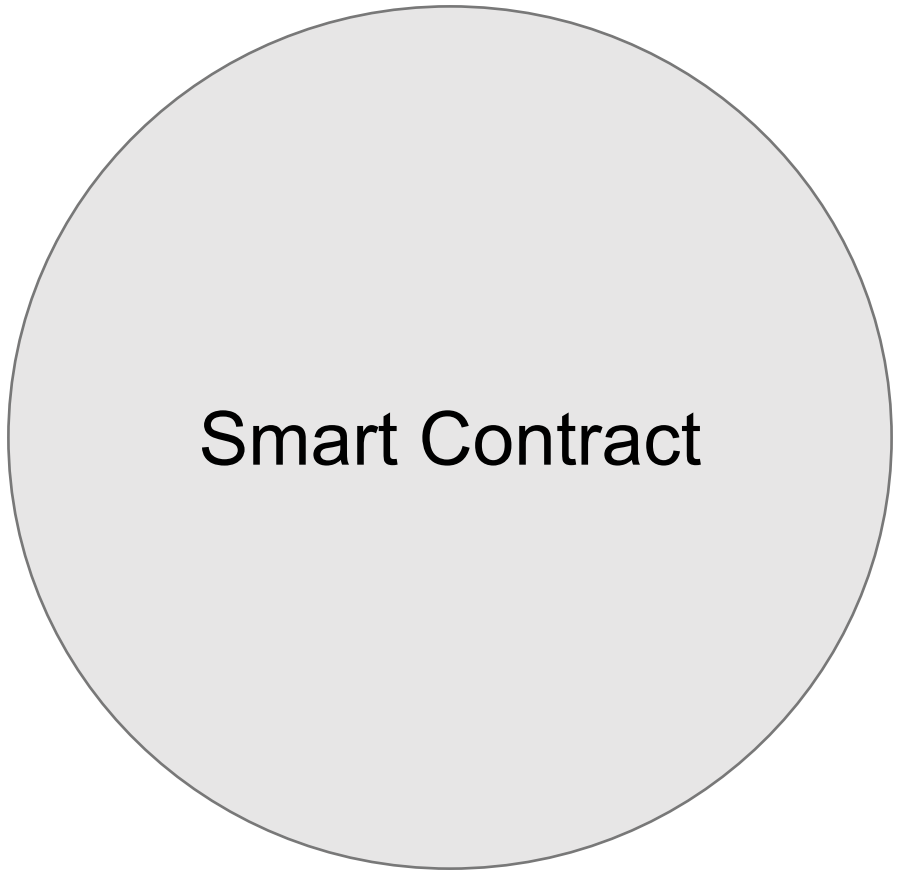
Beispiel:
Tic Tac Toe Contract

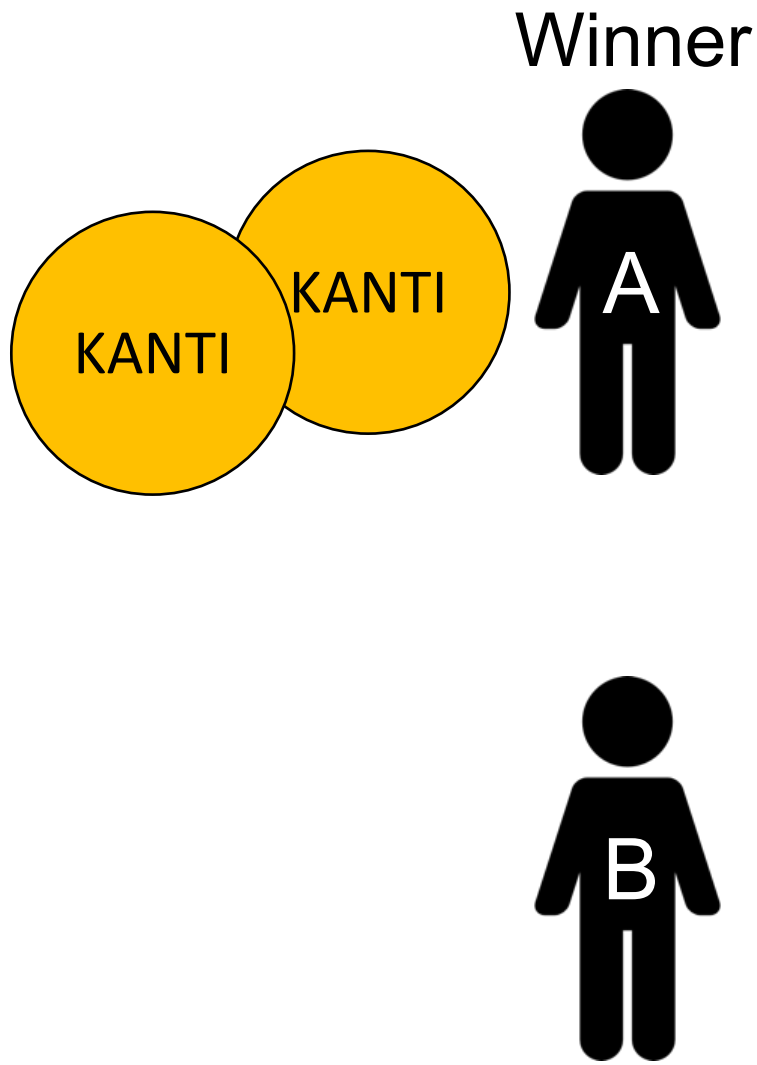

Smart Contract
KANTI
A
KANTI
B

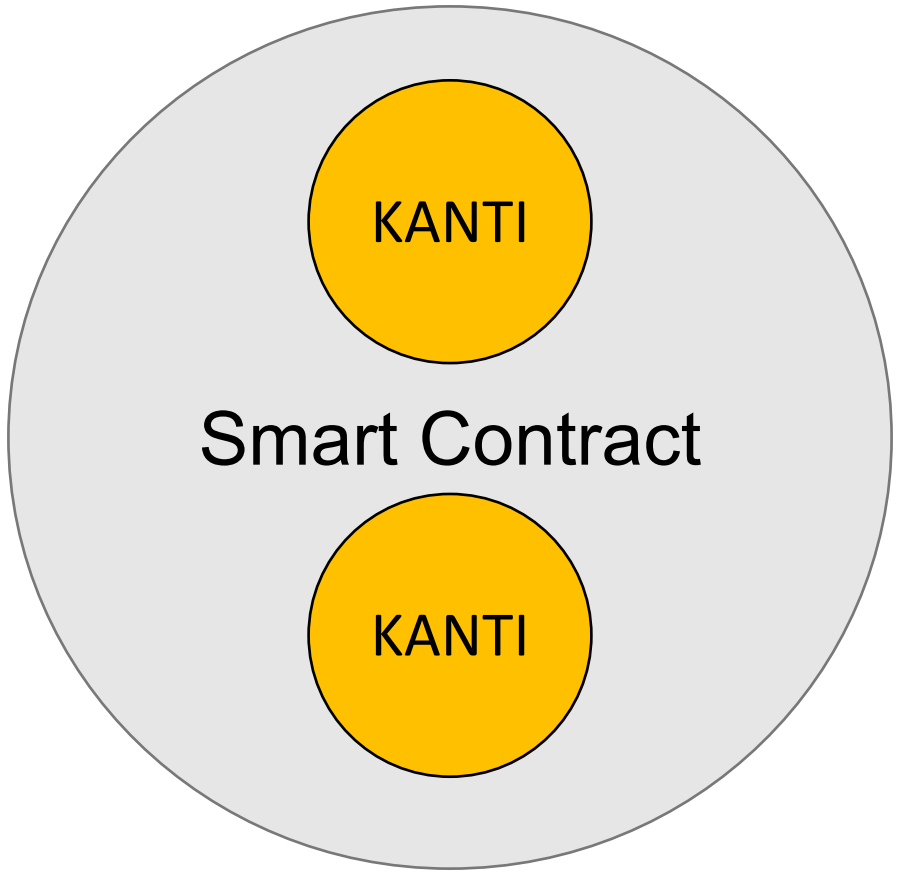

KANTI
Smart Contract
KANTI

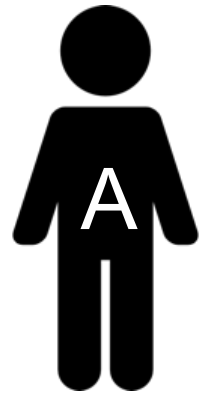

A

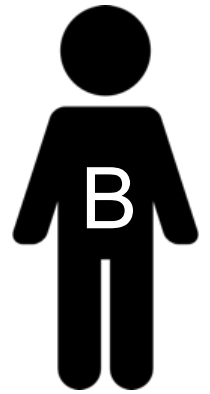

B

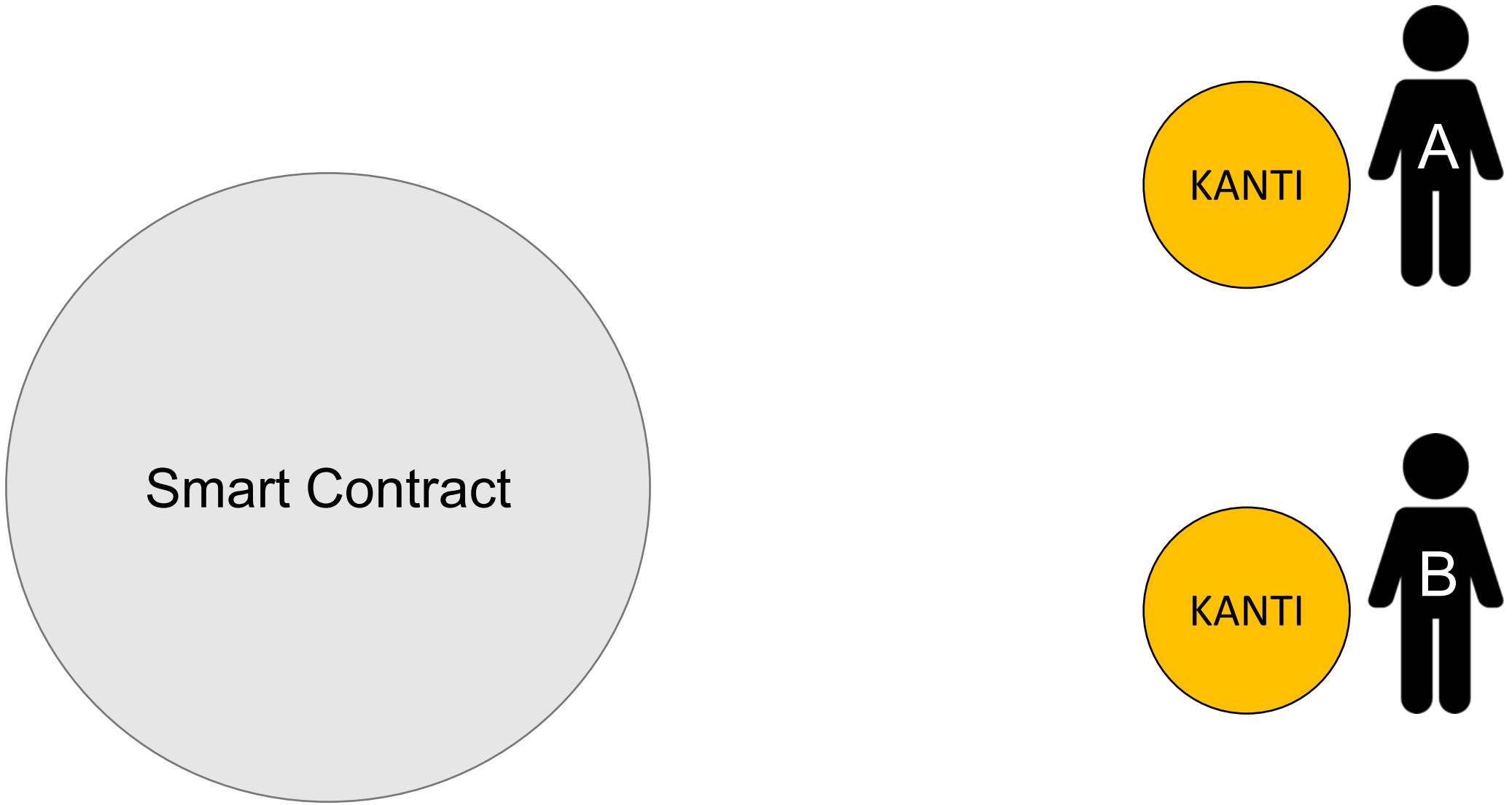

Smart Contract


Winner
KANTI
KANTI
A
B


KANTI
Smart Contract
KANTI

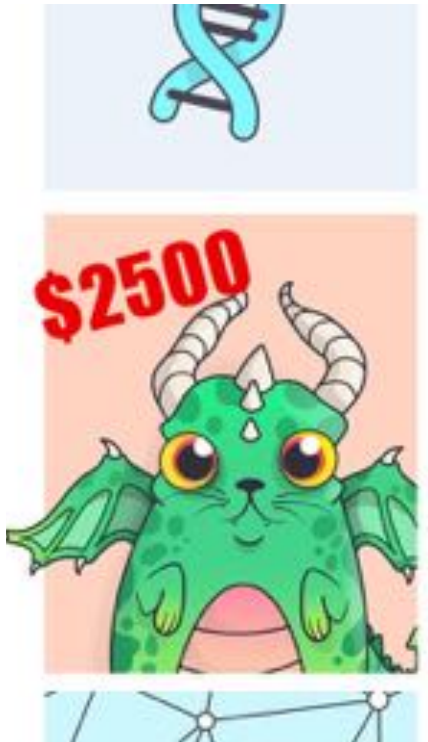

A

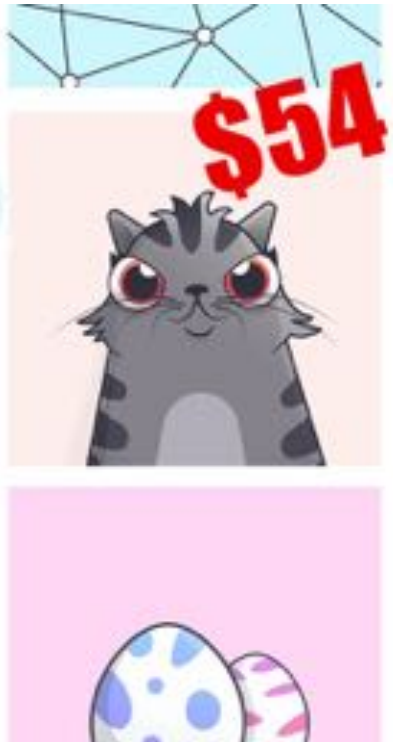

B


Smart Contract
KANTI
A
KANTI
B

ERC20 Token

- Token = Digitaler Wertgegenstand
- ERC20 Token: Kryptowährung („Geld“)
- ERC721 Token “Schmuck“, „Uhren“





Blockchain Tic Tac Toe Game

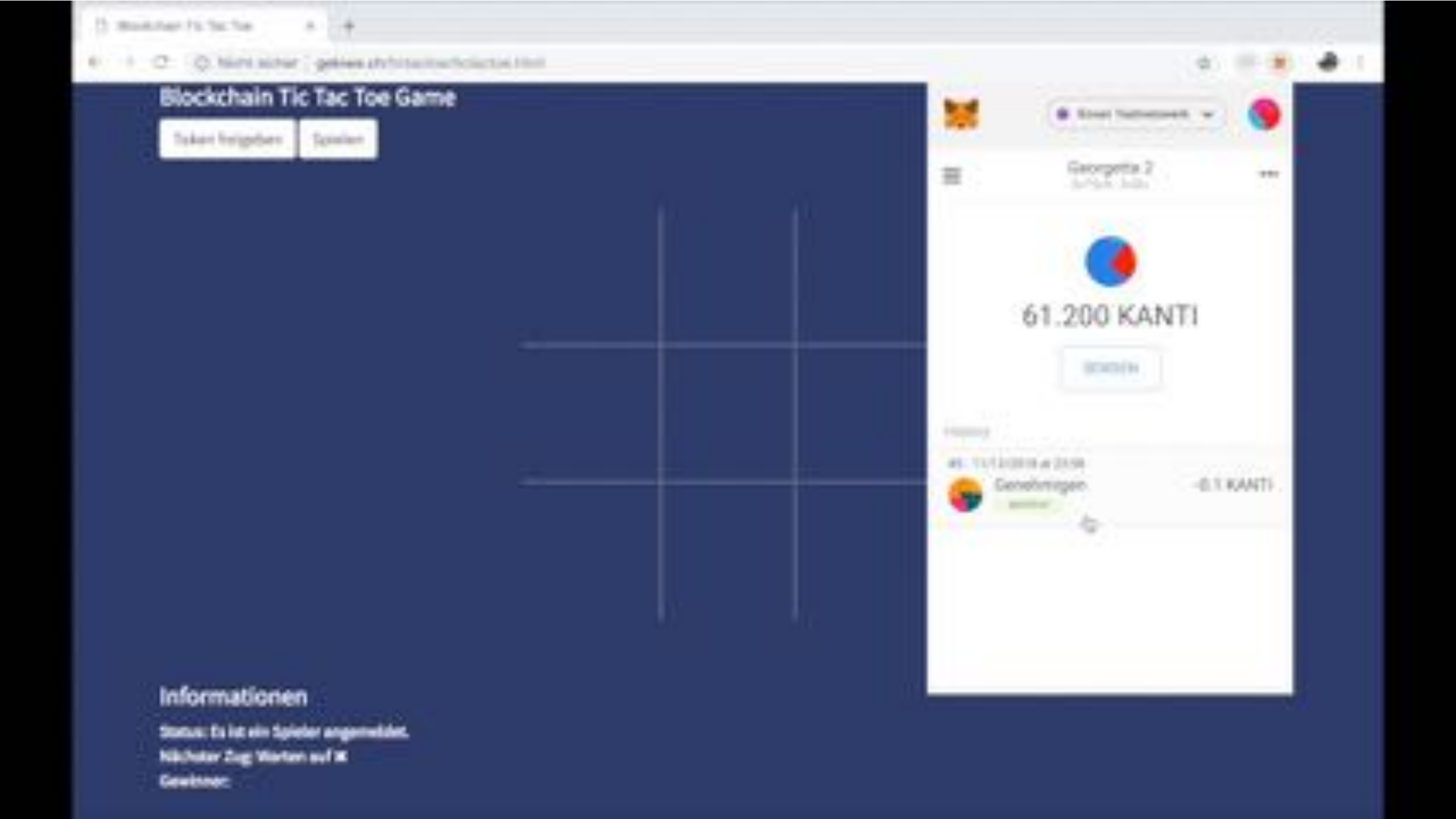
Token freigeben | Spielen

Informationen

Status: Das Spiel ist fertig.

Nächster Zug:

Gewinner: Kein Gewinner!


Blockchain Tic Tac Toe Game
Token freigeben
Spielen
Georgette 2
61.200 KANTI
-0.1 KANTI
Informationen
Status: Es ist ein Spieler angemeldet.
Nächster Zug: Warten auf ✖
Gewinner:

Cryptozombies

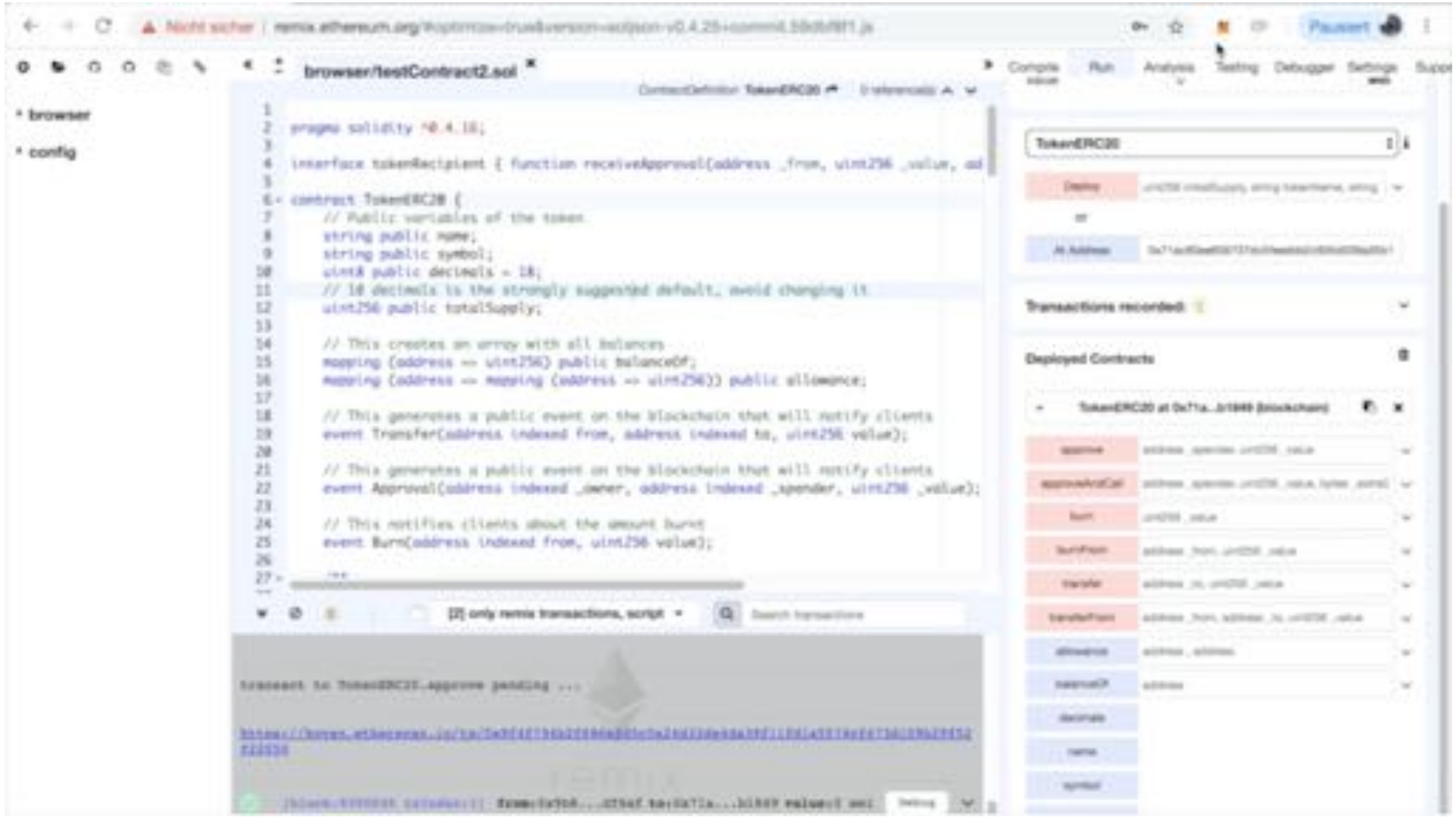
https://cryptozombies.io/

Remix

Beispiel Contract: Counter

remix

ERC20 Token


browser/testContract2.sol
browser
config
pragma solidity ^0.4.16;
interface tokenRecipient { function receiveApproval(address _from, uint256 _value, ad
contract TokenERC20 {
// Public variables of the token
string public name;
string public symbol;
uint8 public decimals = 18;
// 18 decimals is the strongly suggested default, avoid changing it
uint256 public totalSupply;
// This creates an array with all balances
mapping (address => uint256) public balanceOf;
mapping (address => mapping (address => uint256)) public allowance;
// This generates a public event on the blockchain that will notify clients
event Transfer(address indexed from, address indexed to, uint256 value);
// This generates a public event on the blockchain that will notify clients
event Approval(address indexed _owner, address indexed _spender, uint256 _value);
// This notifies clients about the amount burnt
event Burn(address indexed from, uint256 value);
Compile
Run
Analysis
Testing
Debugger
Settings
TokenERC20
Transactions recorded
Deployed Contracts

ERC20 Token


browser/testContract2.sol
browser
config
pragma solidity ^0.4.16;
contract TokenERC20 {
// Public variables of the token
string public name;
string public symbol;
uint8 public decimals = 18;
// 18 decimals is the strongly suggested default, avoid changing it
uint256 public totalSupply;
// This creates an array with all balances
mapping (address => uint256) public balanceOf;
mapping (address => mapping (address => uint256)) public allowance;
// This generates a public event on the blockchain that will notify clients
event Transfer(address indexed from, address indexed to, uint256 value);
// This generates a public event on the blockchain that will notify clients
event Approval(address indexed _owner, address indexed _spender, uint256 _value);
// This notifies clients about the amount burnt
event Burn(address indexed from, uint256 value);
TokenERC20
Transactions recorded:
Deployed Contracts

Maturaarbeit