

# Blockchain Technologies

@romeokienzler

# IBM Disclaimer

- THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
- WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.
- IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.
- NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:
  - CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
  - ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

We are looking forward to hearing your suggestions and feedback so that we can improve our solutions and products.

By joining this web conference, or clicking the "Join" link to join the ITM vNext Open Beta Community, you agree to the following terms, in addition to those set forth in the developerWorks terms of use:

IBM shall be free to use for any purpose and without restriction any oral or written suggestions or feedback that you provide to IBM. By providing IBM with any information or material, you grant IBM an unrestricted, irrevocable license to copy, reproduce, publish, upload, post, transmit, distribute, publicly display, perform, modify, create derivative works from, and otherwise freely use, those materials or information. You also agree that IBM is free to use any ideas, concepts, know-how, or techniques that you provide us for any purpose.

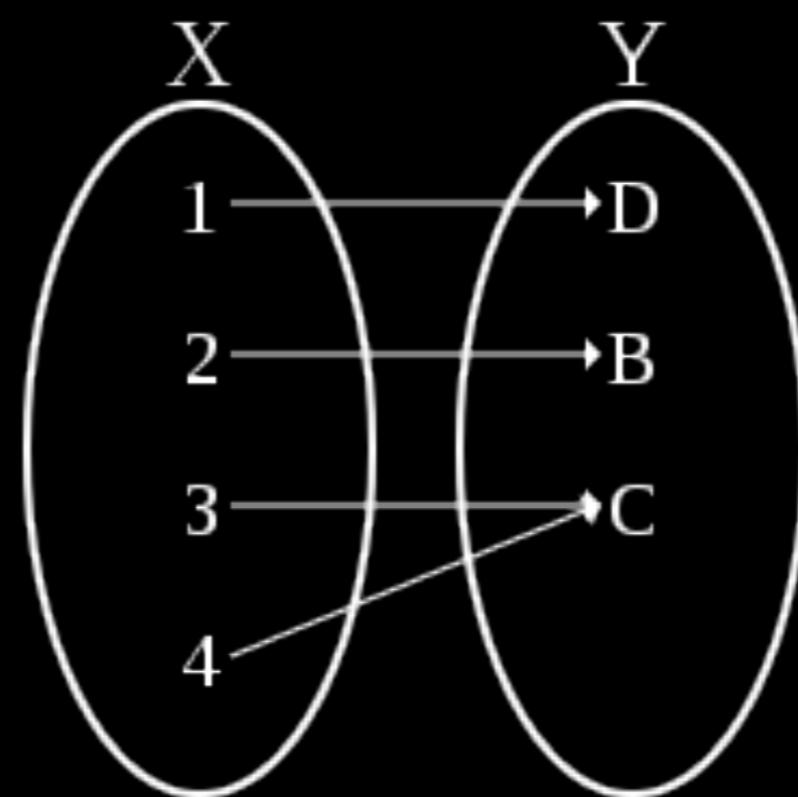
$$\text{hash}(x)=h$$

$\text{hash}(x)=h$   
=> fast

$$\text{hash}^{-1}(h) = x$$

$$\text{hash}^{-1}(h)=x$$

=> slow / brute force / impossible



# “RSA”

*Asymmetric Cryptography*

$p = 29, q = 31$

$$n = p * q = 29 * 31 = 899$$

$$t = (p - 1) * (q - 1) = (29 - 1) * (31 - 1) = 840$$

find  $e$  relatively prime to  $t$

find  $e$  relatively prime to  $t$   
( $t$  cannot be divisible by  $e$ )

find  $e$  relatively prime to  $t$   
( $t$  cannot be divisible by  $e$ )  
e.g.  $e = 11$  ( $t = 840$ )

find d such that  $(d * 11) / t$  give us a remainder of one  
 $d * e \equiv 1 \pmod{t}$

p - 29

q - 31

n - 899

t - 840

e - 11

d - 611

public key n & e  
private key n & d

$$C = M^e \bmod n$$

$$C = M^e \bmod n$$
$$M = 'w'$$

119

$$C = M^e \bmod n$$

'w' => ascii value is 119

$$C = 119^{11} \bmod 899 = 595$$

595

$$M = C^d \bmod n$$

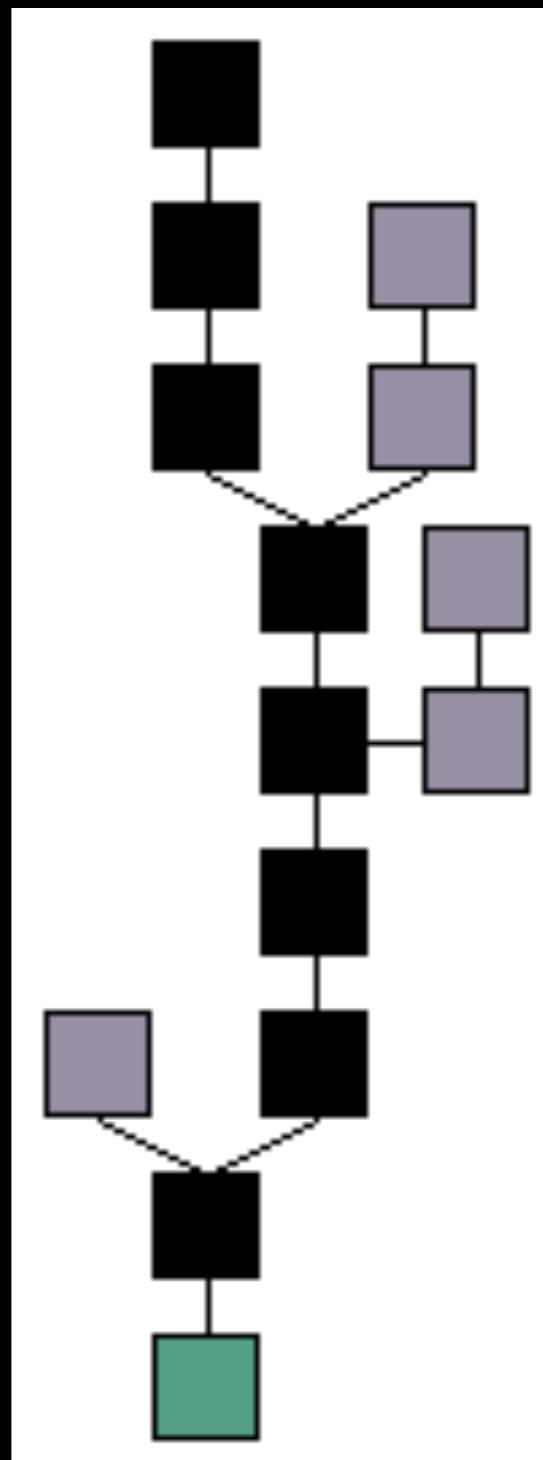
$$M = 595^{611} \bmod 899 = 119$$

119

# What was Bitcoin?

- **oldest cryptocurrency**
- **initiated anonymously (paper + source code)**
- **creates a distributed ledger (blockchain)  
secured by proof of work**
- **proof of work SHA-256 based**
  - **This is a problem! (see later)**

# Bitcoin (Blockchain)



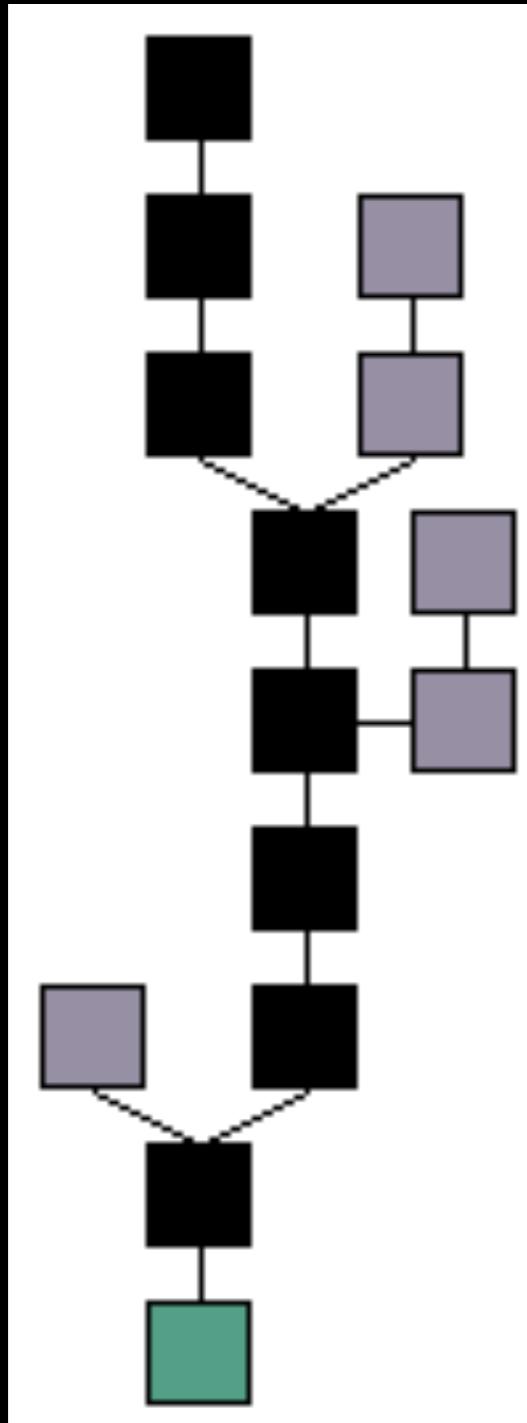
# Bitcoin (Block)

| Field               | Description                                  | Size                                    |
|---------------------|--|---|
| Magic no            | value always 0xD9B4BEF9                      | 4 bytes                                 |
| Blocksize           | number of bytes following up to end of block | 4 bytes                                 |
| Blockheader         | consists of 6 items                          | 80 bytes                                |
| Transaction counter | positive integer <a href="#">VI = VarInt</a> | 1 - 9 bytes                             |
| transactions        | the (non empty) list of transactions         | <Transaction counter>-many transactions |

# Bitcoin (Transaction)

| Field           | Description  | Size                       |
|-----------------|--|----------------------------|
| Version no      | currently 1  | 4 bytes                    |
| In-counter      | positive integer <a href="#">VI = VarInt</a>   | 1 - 9 bytes                |
| list of inputs  | <a href="#">the first input of the first transaction is also called "coinbase" (its content was ignored in earlier versions)</a> | <in-counter>-many inputs   |
| Out-counter     | positive integer <a href="#">VI = VarInt</a>   | 1 - 9 bytes                |
| list of outputs | <a href="#">the outputs of the first transaction spend the mined bitcoins for the block</a>                                      | <out-counter>-many outputs |
| lock_time       | if non-zero and sequence numbers are < 0xFFFFFFFF: block height or timestamp when transaction is final                           | 4 bytes                    |

# Bitcoin (Blockchain)



| Field               | Description                                  | Size                                    |
|---------------------|--|---|
| Magic no            | value always 0xD9B4BEF9                      | 4 bytes                                 |
| Blocksize           | number of bytes following up to end of block | 4 bytes                                 |
| Blockheader         | consists of 6 items                          | 80 bytes                                |
| Transaction counter | positive integer VI = VarInt                 | 1 - 9 bytes                             |
| transactions        | the (non empty) list of transactions         | <Transaction counter>-many transactions |

| Field               | Description                                  | Size                                    |
|---------------------|--|---|
| Magic no            | value always 0xD9B4BEF9                      | 4 bytes                                 |
| Blocksize           | number of bytes following up to end of block | 4 bytes                                 |
| Blockheader         | consists of 6 items                          | 80 bytes                                |
| Transaction counter | positive integer <b>VI = VarInt</b>          | 1 - 9 bytes                             |
| transactions        | the (non empty) list of transactions         | <Transaction counter>-many transactions |

| Field               | Description                                  | Size                                    |
|---------------------|--|---|
| Magic no            | value always 0xD9B4BEF9                      | 4 bytes                                 |
| Blocksize           | number of bytes following up to end of block | 4 bytes                                 |
| Blockheader         | consists of 6 items                          | 80 bytes                                |
| Transaction counter | positive integer VI = VarInt                 | 1 - 9 bytes                             |
| transactions        | the (non empty) list of transactions         | <Transaction counter>-many transactions |

| Field               | Description                                  | Size                                      |
|---------------------|--|---|
| Magic no            | value always 0xD9B4BEF9                      | 4 bytes                                   |
| Blocksize           | number of bytes following up to end of block | 4 bytes                                   |
| Blockheader         | consists of 6 items                          | 80 bytes                                  |
| Transaction counter | positive integer <b>VI = VarInt</b>          | 1 - 9 bytes                               |
| transactions        | the (non empty) list of transactions         | < Transaction counter > many transactions |

| Field               | Description                                  | Size                            |
|---------------------|--|---------------------------------|
| Magic no            | value always 0xD9B4BEF9                      | 4 bytes                         |
| Blocksize           | number of bytes following up to end of block | 4 bytes                         |
| Blockheader         | consists of 6 items                          | 80 bytes                        |
| Transaction counter | positive integer VI = VarInt                 | 1 - 9 bytes                     |
| Transactions        | the following is a list of transactions      | Total transaction size + 1 byte |

| Field             | Description  | Size                        |
|-------------------|--|-----------------------------|
| version_no        | currently 1  | 4 bytes                     |
| position_id       | positive int = Variant   | 1 - 9 bytes                 |
| in-out-count      | the input of this field is also called "variant". (its content was ignored in earlier versions)  | 1 - 9 bytes                 |
| out-count         | the output of this field is also called "variant". (its content was ignored in earlier versions) | 1 - 9 bytes                 |
| out-count-mapping | the output of the sub-fields of the first number spans the min-max range for the book            | >1 - out-count-many outputs |
| out-count-time    | if non-zero and the first number spans the min-max range for the book                            | >1 - out-count-many outputs |
| book_time         | the output of the first number spans the min-max range for the book                              | >1 - out-count-many outputs |
| book_time_i       | the output of the first number spans the min-max range for the book                              | >1 - out-count-many outputs |

| Field       | Description   | Size                        | Field       | Description   | Size                        |
|-------------|---|-----------------------------|-------------|---|-----------------------------|
| version.no  | currently 1   | 4 bytes                     | version.no  | currently 1   | 4 bytes                     |
| in-counter  | positive integer 0 .. Variant   | 1 .. 9 bytes                | in-counter  | positive integer 0 .. Variant   | 1 .. 9 bytes                |
| in-inputs   | The first input of the first transaction is also called "content". (its content was stored in earlier version)  | <in-counters-many inputs>   | in-inputs   | The first input of the first transaction is also called "content". (its content was stored in earlier version)  | <in-counters-many inputs>   |
| out-counter | positive integer 0 .. Variant   | 1 .. 9 bytes                | out-counter | positive integer 0 .. Variant   | 1 .. 9 bytes                |
| out-inputs  | The last output of the first transaction opened the mixed blocks for the block<br>last time .. If no open and sequence numbers are < C#FFFFFFFFFF .. block height or timestamp when transaction is final .. 4 bytes | <out-counters-many outputs> | out-inputs  | The last output of the first transaction opened the mixed blocks for the block<br>last time .. If no open and sequence numbers are < C#FFFFFFFFFF .. block height or timestamp when transaction is final .. 4 bytes | <out-counters-many outputs> |

| Field         | Description   | Size                  |
|---------------|---|-----------------------|
| version_no    | currently 1   | 4 bytes               |
| in-counter    | positive integer 1..Value   | 1..9 bytes            |
| id_of_inputs  | The last input of this transaction is also called "combine".(its content was ginned in earlier version)           | >counter-many inputs  |
| out-counter   | positive integer 1..Value   | 1..9 bytes            |
| id_of_outputs | The last output of this transaction is also called "combine".(its content was ginned in earlier version)          | >counter-many outputs |
| last_time     | If non-zero and sequence numbers are <FFFFFFFFFF 256 blocks height or timestamp when transaction is final 4 bytes | 4 bytes               |
| Field         | Description   | Size                  |
| version_no    | currently 1   | 4 bytes               |
| in-counter    | positive integer 1..Value   | 1..9 bytes            |
| id_of_inputs  | The last input of this transaction is also called "combine".(its content was ginned in earlier version)           | >counter-many inputs  |
| out-counter   | positive integer 1..Value   | 1..9 bytes            |
| id_of_outputs | The last output of this transaction is also called "combine".(its content was ginned in earlier version)          | >counter-many outputs |
| last_time     | If non-zero and sequence numbers are <FFFFFFFFFF 256 blocks height or timestamp when transaction is final 4 bytes | 4 bytes               |

| Field       | Description   | Size        | Field       | Description   | Size        |
|-------------|---|-------------|-------------|---|-------------|
| Version     | currently 1   | 4 bytes     | Version     | currently 1   | 4 bytes     |
| Out-counter | postpone message ID - Variant   | 1 - 8 bytes | Out-counter | postpone message ID - Variant   | 1 - 8 bytes |
| Out-inputs  | list of inputs. The last input of the first transaction is also called "lastbase". (its content was spinned in earlier version) | >0 inputs   | Out-inputs  | list of inputs. The last input of the first transaction is also called "lastbase". (its content was spinned in earlier version) | >0 inputs   |
| Out-counter | postpone message ID - Variant   | 1 - 8 bytes | Out-counter | postpone message ID - Variant   | 1 - 8 bytes |
| Out-outputs | list of outputs. The last output of the first transaction spend the unique UTXO for the block.                                  | >0 outputs  | Out-outputs | list of outputs. The last output of the first transaction spend the unique UTXO for the block.                                  | >0 outputs  |
| Out-time    | (the number of microseconds from the first transaction until the time when transaction is final)                                | 4 bytes     | Out-time    | (the number of microseconds from the first transaction until the time when transaction is final)                                | 4 bytes     |
| Out-flags   | 0x00000000 - 0xffffffff. State flags or timestamps when transaction is final  | 4 bytes     | Out-flags   | 0x00000000 - 0xffffffff. State flags or timestamps when transaction is final  | 4 bytes     |

| Field       | Description   | Size       | Field       | Description   | Size       |
|-------------|---|------------|-------------|---|------------|
| Version_no  | currently 1   | 4 bytes    | Version_no  | currently 1   | 4 bytes    |
| In-counter  | positive integer < N_max  | 4-9 bytes  | In-counter  | positive integer < N_max  | 4-9 bytes  |
| Out-counter | positive integer < N_max  | 4-9 bytes  | Out-counter | positive integer < N_max  | 4-9 bytes  |
| lock        | the output of the first transaction is also called "constraint". Its content was ignored in earlier versions. | > 10 bytes | lock        | the output of the first transaction is also called "constraint". Its content was ignored in earlier versions. | > 10 bytes |
| lock_out    | the output of the first transaction spent the min-balance for the block                                       | > 10 bytes | lock_out    | the output of the first transaction spent the min-balance for the block                                       | > 10 bytes |
| lock_time   | If non-zero and sequence numbers are < DxFFFFFFF, block height or timestamp when transaction is final         | 4 bytes    | lock_time   | If non-zero and sequence numbers are < DxFFFFFFF, block height or timestamp when transaction is final         | 4 bytes    |

| Field       | Description  | Size                     | Field       | Description  | Size                     |
|-------------|--|--------------------------|-------------|--|--------------------------|
| Version.no  | currently 1  | 4 bytes                  | Version.no  | currently 1  | 4 bytes                  |
| In-counter  | positive integer $\leq V_{\text{limit}}$   | 1 - 9 bytes              | In-counter  | positive integer $\leq V_{\text{limit}}$   | 1 - 9 bytes              |
| Out-inputs  | The first input of the first transaction is also called "counter". (Its content was ignored in earlier versions) | out-counter-many inputs  | Out-inputs  | The first input of the first transaction is also called "counter". (Its content was ignored in earlier versions) | out-counter-many inputs  |
| Out-counter | positive integer $\leq V_{\text{limit}}$   | 1 - 9 bytes              | Out-counter | positive integer $\leq V_{\text{limit}}$   | 1 - 9 bytes              |
| Out-outputs | all outputs of the subgraph spent the most bitcoins for the block.   | out-counter-many outputs | Out-outputs | all outputs of the subgraph spent the most bitcoins for the block.   | out-counter-many outputs |
| tx_id       | tx_id + previous transaction hash = CDF1FFFF...00  | 32 bytes                 | tx_id       | tx_id + previous transaction hash = CDF1FFFF...00  | 32 bytes                 |
| tx_in       | tx_in + previous transaction hash = CDF1FFFF...00  | 32 bytes                 | tx_in       | tx_in + previous transaction hash = CDF1FFFF...00  | 32 bytes                 |

| Field          | Description   | Size                      | Field          | Description   | Size                      |
|----------------|---|---------------------------|----------------|---|---------------------------|
| Version        | currently 1   | 4 bytes                   | Version        | currently 1   | 4 bytes                   |
| In-counter     | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 | In-counter     | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 |
| Out-of-order   | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 | Out-of-order   | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 |
| Int of inputs  | The int of the first transaction is also called "combine".(its content was ignored in earlier version)    | >out-counter-many inputs  | Int of inputs  | The int of the first transaction is also called "combine".(its content was ignored in earlier version)    | >out-counter-many inputs  |
| Out of order   | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 | Out of order   | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 |
| Int of outputs | The int of the first transaction spent the min-bits for the block   | >out-counter-many outputs | Int of outputs | The int of the first transaction spent the min-bits for the block   | >out-counter-many outputs |
| Int, Int       | If non-zero and sequence numbers are <= DFFFFFFFFFFF, block height or timestamp when transaction is final | 4 bytes                   | Int, Int       | If non-zero and sequence numbers are <= DFFFFFFFFFFF, block height or timestamp when transaction is final | 4 bytes                   |
| Field          | Description   | Size                      | Field          | Description   | Size                      |
| Version        | currently 1   | 4 bytes                   | Version        | currently 1   | 4 bytes                   |
| In-counter     | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 | In-counter     | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 |
| Out-of-order   | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 | Out-of-order   | positive integer $\leq V_{\text{limit}}$  | 1-9 bytes                 |
| Int of outputs | The int of the first transaction is also called "combine".(its content was ignored in earlier version)    | >out-counter-many outputs | Int of outputs | The int of the first transaction is also called "combine".(its content was ignored in earlier version)    | >out-counter-many outputs |
| Int of outputs | The int of the first transaction spent the min-bits for the block   | >out-counter-many outputs | Int of outputs | The int of the first transaction spent the min-bits for the block   | >out-counter-many outputs |
| Int, Int       | If non-zero and sequence numbers are <= DFFFFFFFFFFF, block height or timestamp when transaction is final | 4 bytes                   | Int, Int       | If non-zero and sequence numbers are <= DFFFFFFFFFFF, block height or timestamp when transaction is final | 4 bytes                   |

# Proof of Work

- $\text{sha256}(\text{sha256}(\text{Block} + \text{Nonce})) = \text{Hash}$
- choose Nonce so that Hash starts with d leading zeros
- d depends on “difficulty”
  - continuously updated by the bitcoin network
- Finding correct Nonce is "proof of work"
  - rewarded with bitcoins (bitcoin mining)

# Proof of Work

```
Hello, world#0 => fa2881e9b47b8e1535df08f1d6d47b71854aa0706c959a2726fc964fc90ff15
Hello, world#1 => 795544e740045733b4713381f8e3e47bfff379e059aca1971b711b0ab8b54fb4
Hello, world#2 => d4d47d6600c4b5f6e2aa6936925741758714a5f8dd8d69eda0ccf3a0287a2c0e
Hello, world#3 => 32192c79cd80b64e57808745bbbaafc4aebab6bec25f5df5435a439323930833
...
...
...
Hello, world#69159 => 46201a335a85608d349fec758409160b40c612fdb51a6c91e65d3b4fddb2f06a
Hello, world#69160 => fecc199b038da2d577745fb05ea85227eb823b2489bb8070e2f42f38d60155f3
Hello, world#69161 => 35ff61c959bec6a6c66ff2cc602a84d02823c46e9ca2e70f03f6ea9c9212842b
Hello, world#69162 => 0000c5a9a24161e58868c858fc2700eeabf21a86862cbfa3bbd18a4d63e5b010
```

# The Problem with SHA256

- Easy to speed up using
  - GPU
  - FPGA
  - ASIC
- Currently without the latest ASIC you spend more on energy than earning bitcoins

```
for (i=0,j=0; i < 16; ++i, j += 4)
    m[i] = (data[j] << 24) | (data[j+1] << 16) | (data[j+2] << 8) | (data[j+3]);
```

# What is an ASIC?

*Very low cost ultra high performance chip  
specialised on only one task*



# What is an ASIC?

*Very low cost ultra high performance chip  
specialised on only one task*



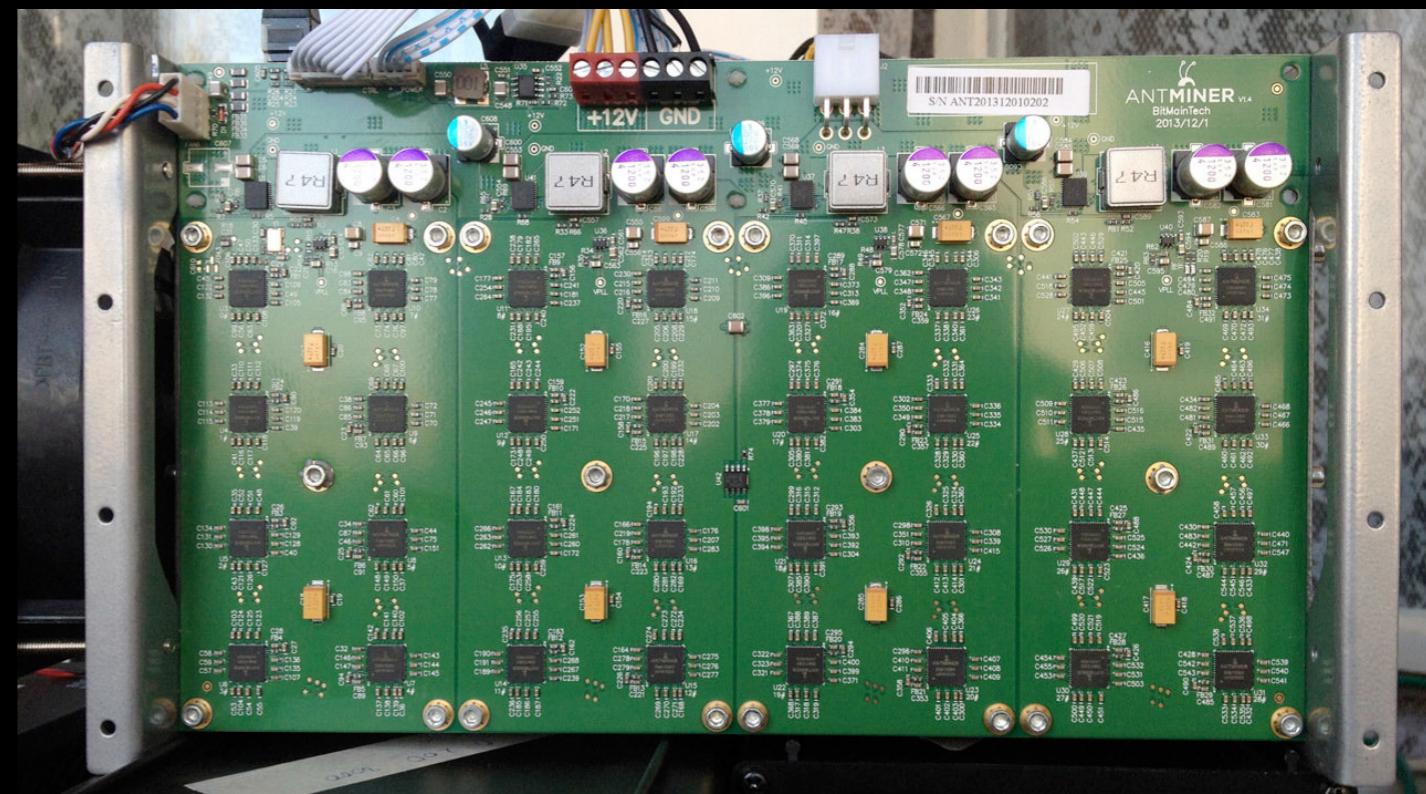
# What is an ASIC?

*Very low cost ultra high performance chip  
specialised on only one task*



# What is an ASIC?

*Very low cost ultra high performance chip  
specialised on only one task*



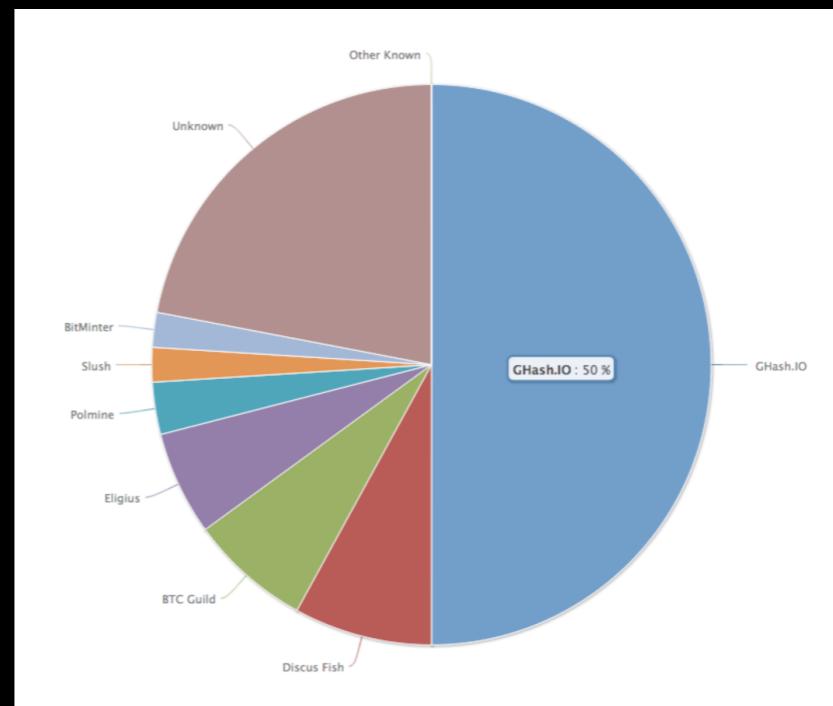
# What is an ASIC?

*Very low cost ultra high performance chip  
specialised on only one task*



# The Problem with SHA256

- *Original Idea is gone (everybody can mine bitcoins using his spare compute resources at home)*
- *system vulnerable to 51% attack*
- *recently 50% of mining power achieved by a single pool*



# Alternative CryptoCurrencies

- *There are many alternatives*
  - *Auroracoin, BlackCoin, Dash, Dogecoin, DigitalNote, Ethereum, Litecoin, Mastercoin, MazaCoin, Monero, Namecoin, Nxt, Peercoin, Emercoin, PotCoin, Primecoin, Ripple, ShadowCash, Titcoin*
- *Alternative hash algorithms*
  - *scrypt, X11, cryptonight, hashimoto*
- *Alternative consensus methods*
  - *proof of work, proof of stake, proof of elapsed time*



# beyond payments

- *Idea: BlockChain as distributed ledger to ensure non-centralized authority on validation of any type of transaction and business logic*
- *two promising candidates*
  - *Ethereum (Swiss Non-Profit Organization)*
  - *Hyperledger Project (The Linux Foundation, members: IBM, Deutsche Börse Group, Intel, JP Morgan, Cisco, RedHat, ...)*

# What is a Blockchain?

*A distributed, peer-to-peer  
replicated, integrity protected  
linked-list of data blocks*

# Hyperledger

- IBM Research prototype “OpenBlockchain”
- OpenSourced by IBM and donated to the Linux Foundation
- Any member can contribute and steer development
- 1st production ready release "Hyerledger Fabric V1.0" was announced at IBM Interconnect 2017

# What is a Hyperledger Fabric?

*"Hyperledger Fabric is a enterprise grade, distributed based on blockchain technology that use smart contracts that enforce trust between entities" @gatakka Ivan Vankov*

# Misconceptions

- Hyperledger is a Blockchain
- Hyperledger is not a Cryptocurrency
- Hyperledger is not using Mining/PoW(Proof of Work)
- But preserves important properties of a crypto blockchain

# Main benefit

- throughput of the system
- Ethereum 1000 transactions per minute
- Hyperledger 500 000 transactions per minute
- No loss of money through mining (electricity)

# Architecture Intro

- distributed by design
- no single point of failure

# Fabric-CA

- User management through X.509 certificates
- Attributes inside certificates are used to define roles and rights
- Can be attached to LDAP / Active Directory
- "only" a tool, can also use standalone OpenSSL

# Peer

- place where ledger is stored

# Orderer

- coordinator for transactions

# Membership Service Provider

- Certificate Management for Hyperledger Fabric Components

# Channel

- **data isolation / multi-tenancy**
- **every party must accept an additional party to join**
- **peers take part in channel**
- **add / remove possible during runtime**

# Chaincode

- Smart-contract / Business-logic over data in the ledger
- Only way to interact with the ledger
- NodeJS, GoLang, Java
- no limit of what you can use, external libraries, external network calls
- Chaincode runs inside channel

# Chaincode

- Needs to be installed and instantiated on every peer (can be automated)

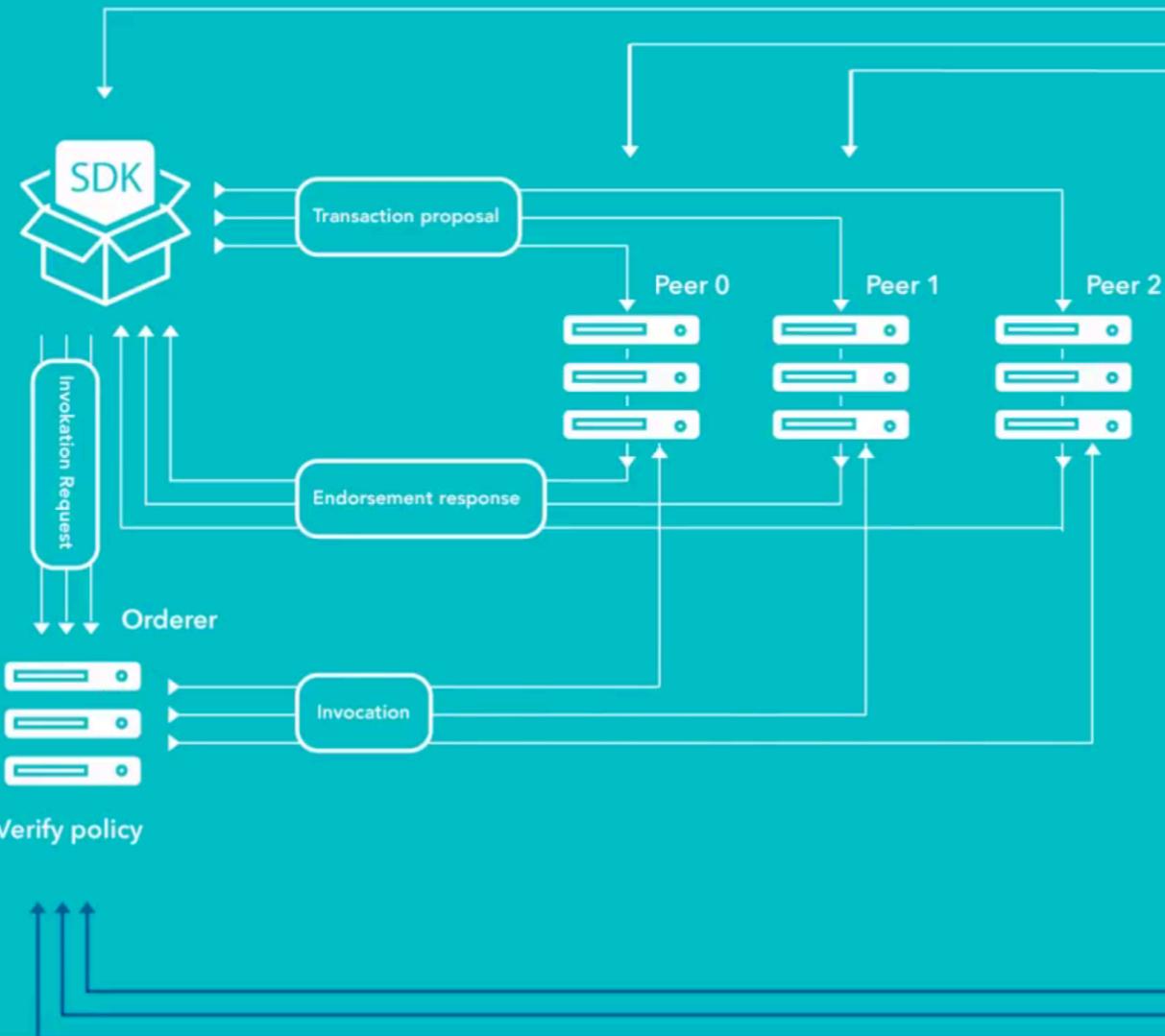
# Policies

- Define level of security
- No chaincode without policy file

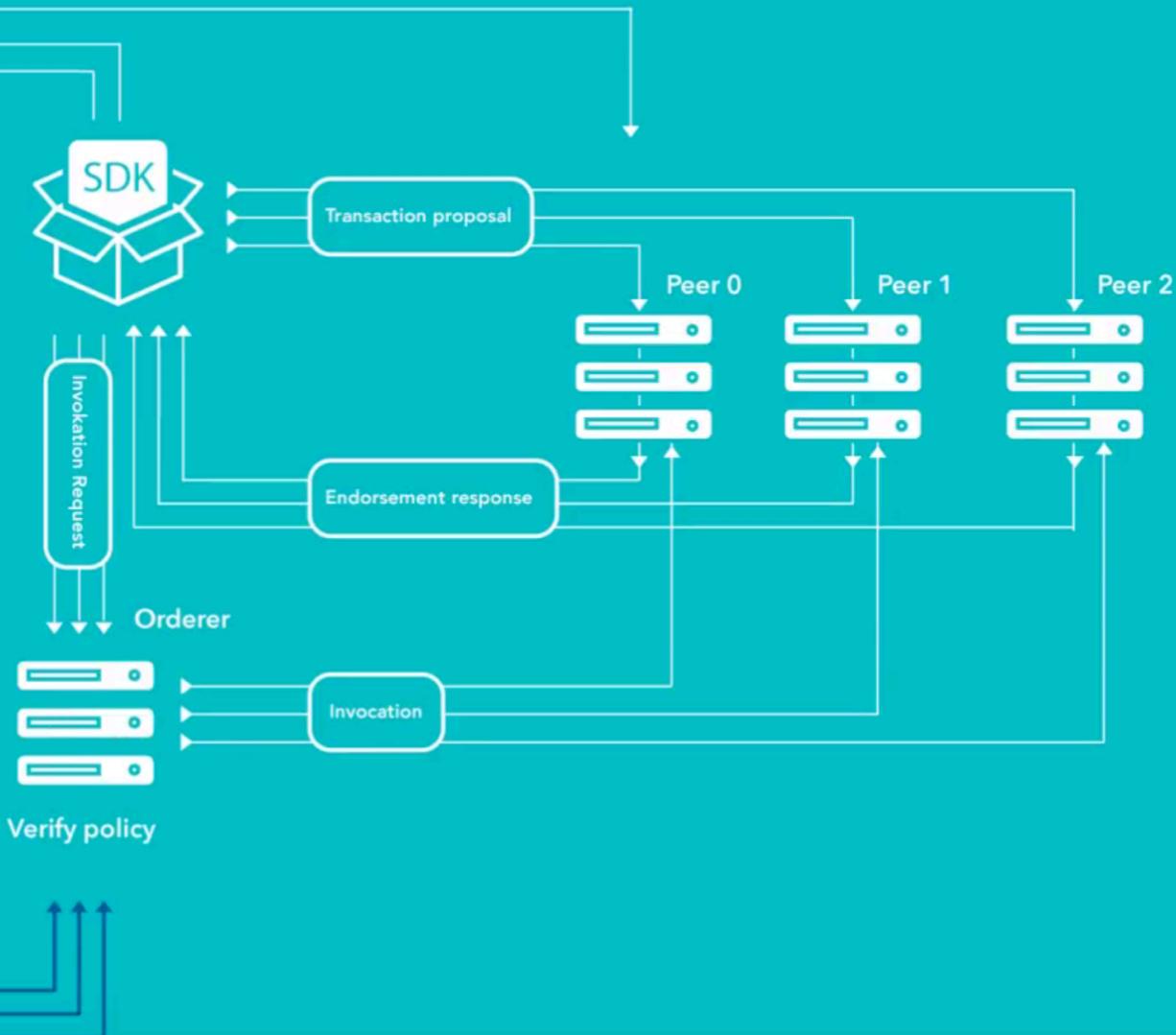
# RECAP

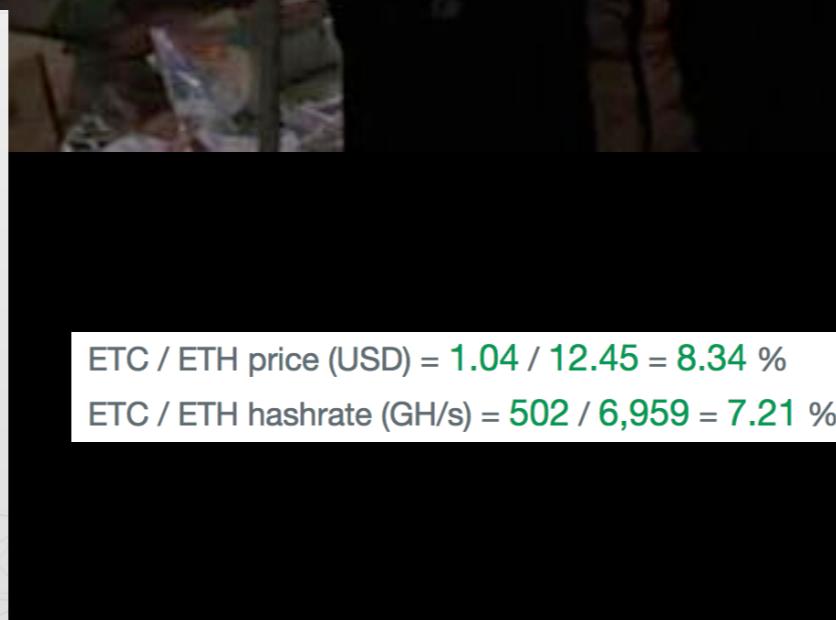
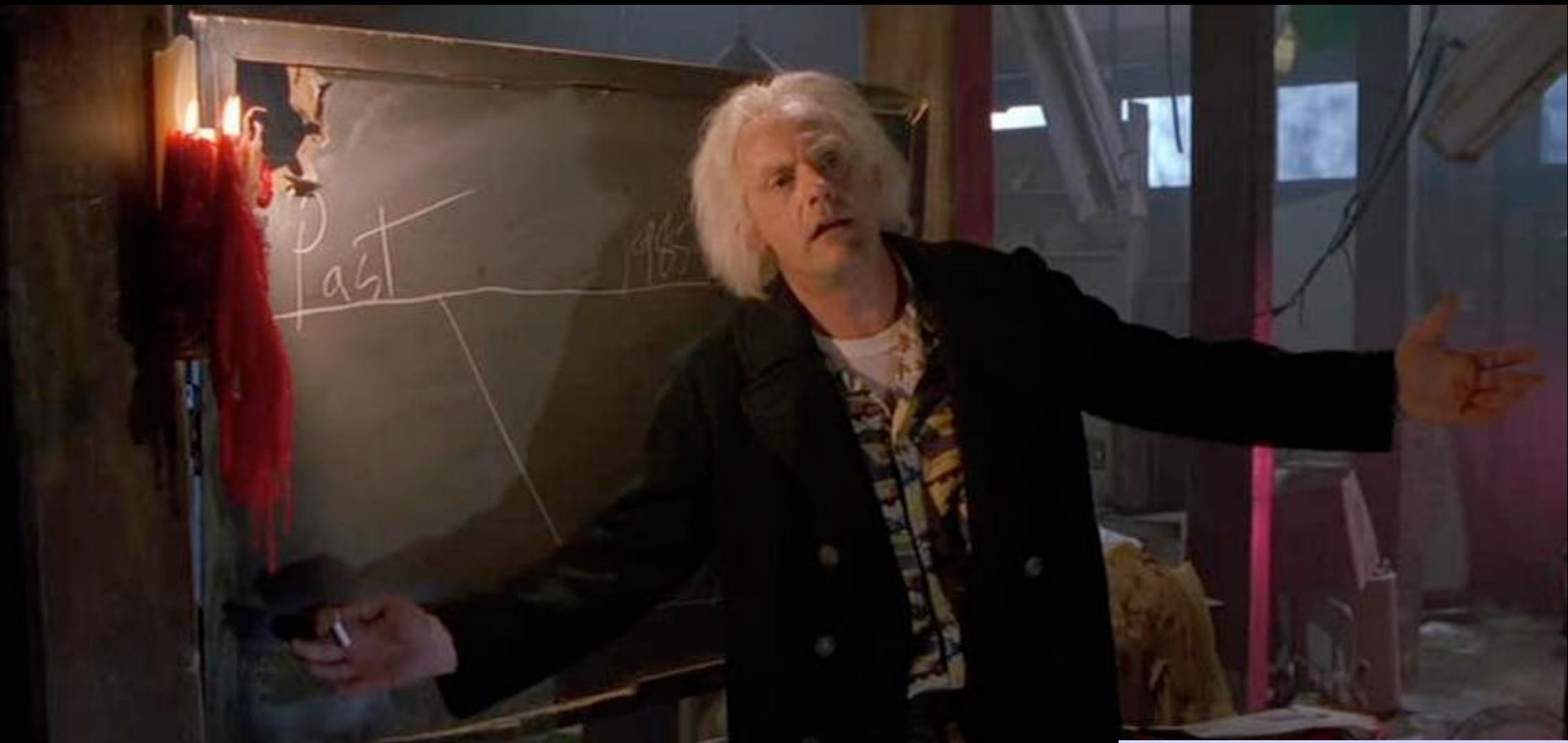
- Peer - may be part of one or many channels
- Every single channel has a separate ledger
- Every channel has one or many chaincodes
- Every chain code has a different policy

## 1 ORGANISATION

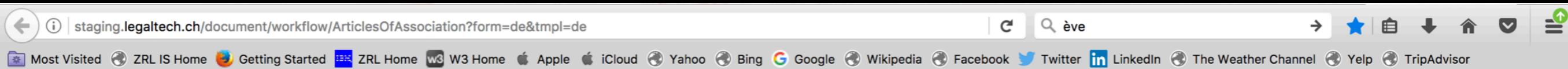


## 2 ORGANISATION





# digital switzerland challenge



## Gesellschaftsrecht-Statuten (Aktiengesellschaft)

Sign in Sign up

- Angaben zur Aktiengesellschaft
- Aktien und Aktienkapital
- Generalversammlung
- Datum und Unterschrift
- Beglaubigung

### Aktien und Aktienkapital

Bitte geben Sie das Aktienkapital (CHF) der Gesellschaft ein.

\*  
50000

In welche Anzahl Namenaktien ist das Aktienkapital der Gesellschaft eingeteilt?

\*  
5000

Bitte geben Sie den Nominalwert (CHF) pro Namenaktie ein.

\*  
10

Back

Next

Statuten  
der  
Gesellschaft  
(the „Company“)

Art. 1 Firma, Sitz, Dauer und Zweck  
Name, Domicile, Duration and  
Purpose  
Art. 1 Firma, Sitz, Dauer  
Under the name of  
bestellt eine Aktiengesellschaft,  
welche den Vorschriften des 26. Titels des  
Buches IV des Eidgenössischen Organisationsrechts  
(„FOR“) untersteht.  
Der Sitz der Gesellschaft ist in  
Die Gesellschaft besteht  
auf  
Art. 2 Zweck  
Die Gesellschaft kann im In- und  
Ausland Zweigniederlassungen er-  
richten, sich an anderen  
Unternehmungen beteiligen, an  
beteiligen, gemeinsames Übernehmen  
oder Veräußern von Anteilen am  
Vertrag abschließen, um die  
Geschäfte der Gesellschaft zu fördern; aber die direkt  
oder indirekt damit im Zusammenhang  
stehenden Geschäfte müssen mit dem  
Zweck der Gesellschaft übereinstimmen. Diese  
Klausur für eigene oder fremde Rechnung  
oder auf eigene oder fremde Rechnung  
Handelsgeschäfte für verbündete  
Unternehmungen und Dritte eingehen.

The Company may establish  
subsidiaries in Switzerland and abroad;  
participate in other companies without  
restriction; enter into joint ventures and  
enter into contracts for the promotion of  
the Company's business objects; the  
Company may enter into contracts for  
its own or third parties' account and  
enter into business transactions with  
affiliated companies or for third parties. The  
Company may enter into contracts for  
the promotion of its business objects;  
enter into contracts for the promotion of  
the Company's business objects; the  
Company may enter into contracts for  
its own or third parties' account and  
enter into business transactions with  
affiliated companies or for third parties. The  
Company may enter into contracts for  
the promotion of its business objects;

Legal Entity Establishment 4w - 6w => 48h

<https://ibm-blockchain.github.io/>