19. APRIL 2018

KÖLN / INNOQ HANDS-ON-EVENT

MARC JANSING, ROBERT GLASER

Blockchain

INNOQ

1. **Was ist das?**
2. **Anwendungsfälle**
3. **Aufgabe**

Was ist das?

Was ist eine Blockchain?

Eine verteilte Datenbank,
die ohne vertrauenswürdige Instanz
auskommt.

Was ist ein Block?

Block 2

```
index: 2
timestamp: 1523443235636
value: "foobar"
```

Warum keine Tabelle?

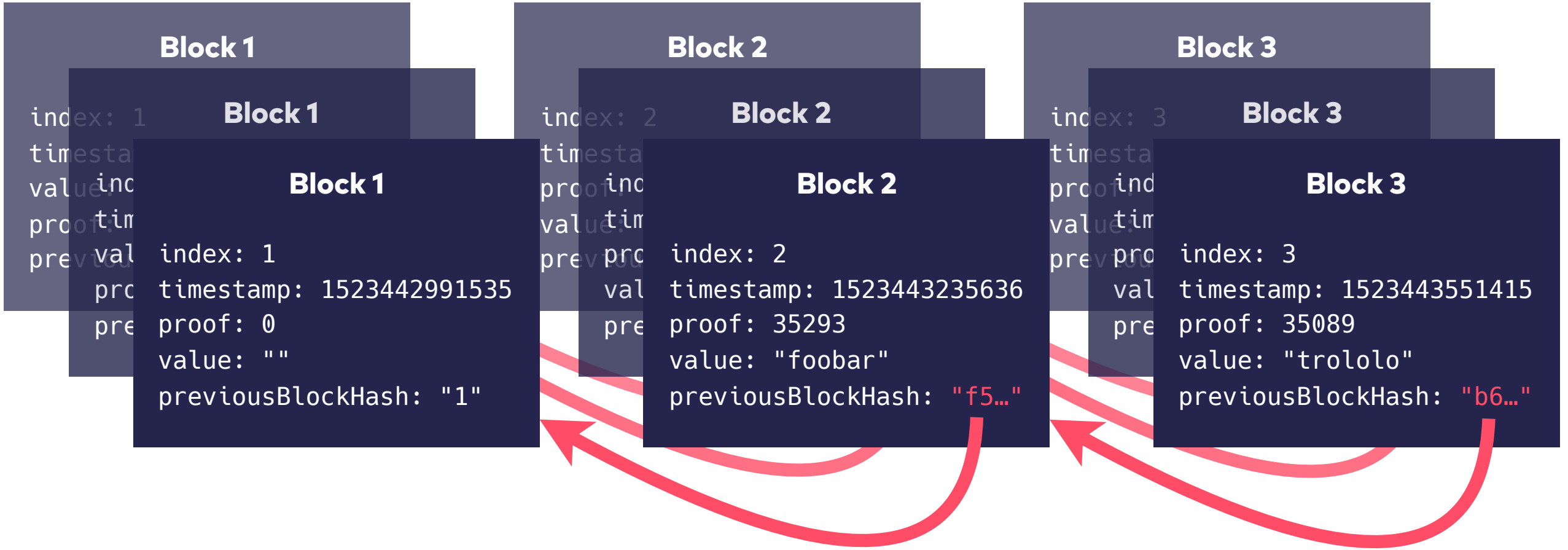
Block 2

```
index: 2
timestamp: 1523443235636
value: "foobar"
proof: 35089
previousBlockHash: "f521a…"
```

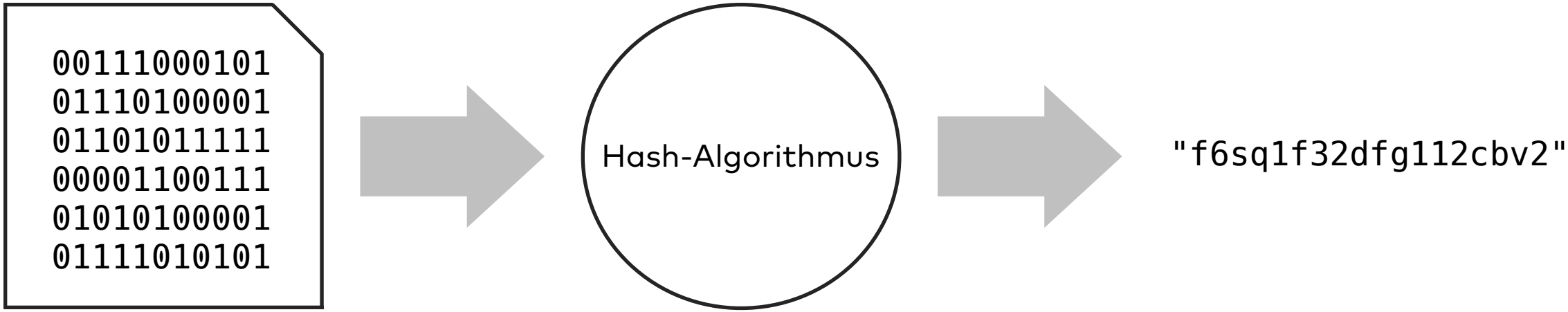
Verkettung


Block 1
index: 1
timestamp: 1523442991535
value: ""
proof: 0
previousBlockHash: "1"
Block 2
index: 2
timestamp: 1523443235636
proof: 35293
value: "foobar"
previousBlockHash: "f5…"
Block 3
index: 3
timestamp: 1523443551415
proof: 35089
value: "trololo"
previousBlockHash: "b6…"

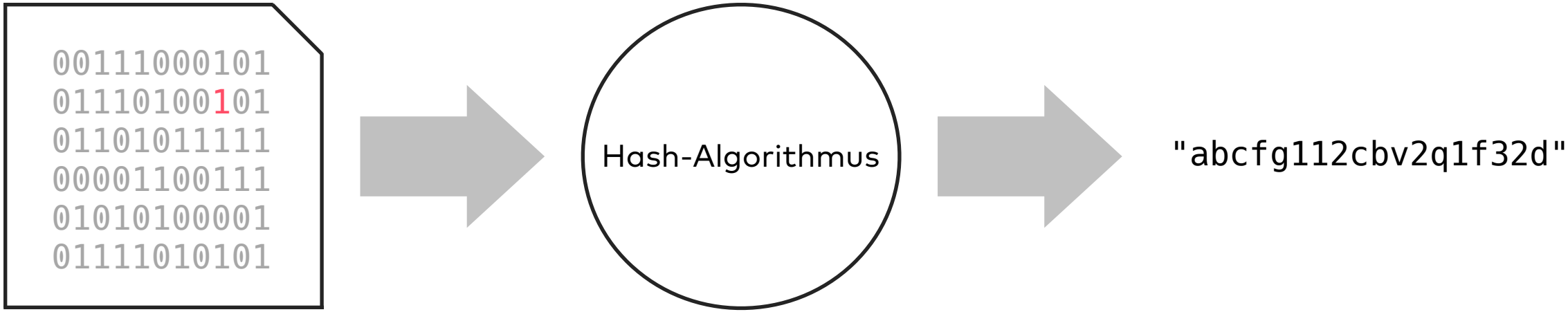
Verteilung


Block 1
Block 1
Block 1
index: 1
timestamp: 1523442991535
proof: 0
value: ""
previousBlockHash: "1"
Block 2
Block 2
Block 2
index: 2
timestamp: 1523443235636
proof: 35293
value: "foobar"
previousBlockHash: "f5…"
Block 3
Block 3
Block 3
index: 3
timestamp: 1523443551415
proof: 35089
value: "trololo"
previousBlockHash: "b6…"

Hashing


00111000101
01110100001
01101011111
00001100111
01010100001
01111010101
Hash-Algorithmus
"f6sq1f32dfg112cbv2"

Hashing


00111000101
01110100101
01101011111
00001100111
01010100001
01111010101
Hash-Algorithmus
"abcfg112cbv2q1f32d"

Hashing

```
00111000101
01110100001
01101011111
00001100111
01010100001
01111010101
```

"abcfg112cbv2q1f32d"

Wie entsteht ein Block?

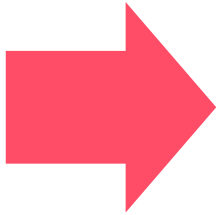
Mining

- Ein Block muss „erarbeitet“ werden
- Meist: Lösung einer kryptographischen Aufgabe
- Rätsel ist schwer zu lösen, aber einfach zu überprüfen
- Anreiz?

Mining – eine Metapher

Aufgabe: Löse dieses Sudoku-Feld.

4	1			6	5			7
		6			7	4	8	
2		7	4	9				6
	6			7		1		
3		1	5				7	2
	9			4	2	3		8
1		8	6				2	9
	2			1	8	6	4	
6			3				1	



4	1	3	8	6	5	2	9	7
9	5	6	2	3	7	4	8	1
2	8	7	4	9	1	5	3	6
8	6	2	9	7	3	1	5	4
3	4	1	5	8	6	9	7	2
7	9	5	1	4	2	3	6	8
1	3	8	6	5	4	7	2	9
5	2	9	7	1	8	6	4	3
6	7	4	3	2	9	8	1	5

Mining – ein weiteres Beispiel

Aufgabe: Finde einen Hash, der mit 6 Nullen beginnt.

```
basis = "I am Heribert Innoq"
zähler = 0

zähler + 1 until
  Hash("basis+zähler").starts_with "000000"
```

Warum nicht einfach einen Block fälschen?

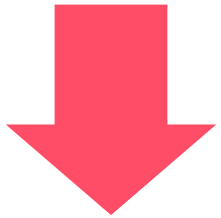
!Vertrauen

Proof of Work

- Erstelle eine Block, dessen Hash einem vorgegebenen Muster entspricht
- Ich kenne Basis und Zähler
- Dann kann ich schnell prüfen, ob das Ergebnis ein Hash ist, der mit 6 Nullen beginnt

Beweis

```
Hash("I am Heribert Innoq3926606")
```



```
0000002 15ac8b015c1e8046cba23b99a5ff5f6ef8651
5a191065d840cd0c5dc8
```

Anwendungsfälle

Digitale Währungen

Grundbucheinträge

Patientendaten

Fahrtenbücher

Verträge

...

„MISSION E"

So setzt Porsche die Blockchain beim neuen Elektroflitzer ein

Porsche will mit dem Elektro-Sportwagen „Mission E" bei der Digitalisierung des Autos ganz vorne mitmischen. Ein Blick hinter die Kulissen.

Aufgabe

Baue eine Blockchain!

github.com/innoq/innoq-blockchain-SPRACHE

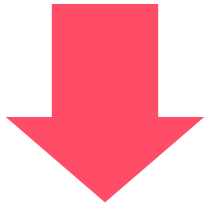
#IamHeribertInnoq

Genesis-Block


```
{
  "index": 1,
  "timestamp": 0,
  "proof": 955977,
  "transactions": [{
    "id": "b3c973e2-db05-4eb5-9668-3e81c7389a6d",
    "timestamp": 0,
    "payload": "I am Heribert Innoq"
  }],
  "previousBlockHash": "0"
}
```

Block-Kandidat

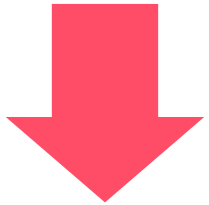
```
{
  "index": 2,
  "timestamp": 1523525426,
  "proof": 0,  // initial proof
  "transactions": [],
  "previousBlockHash": "0000008793d0a9…"
}
```



123456215ac8b015c1e8046cba23b99a5ff5f6ef86515a191065d840cd0c5dc8

Gefundener Block

```
{
  "index": 2,
  "timestamp": 1523525426,
  "proof": 7803399,  // valid proof
  "transactions": [],
  "previousBlockHash": "0000008793d0a9…"
}
```



000000ab4766ccea2cf59302480956b3667eebc5634b0dff285ff52edf88eca5

previousBlockHash

- SHA256 über den ganzen Vorgänger-Block (JSON)
- Vor dem Hashen: Nach Keys sortieren

Blockchain-API

GET /

GET /blocks

GET /mine

GET /


```
{
  "nodeId": "bcfeb8c5-c9a6-4731-9a17-e0fedd7aa073",
  "currentBlockHeight": 69
}
```

GET /blocks


```
{
  "blocks": [
   {
    "index": 1,
    "timestamp": 955977,
    "proof": 0,
    "transactions": [
       {
         "id": "b3c973e2-db05-4eb5-9668-3e81c7389a6d",
         "timestamp": 0,
         "payload": "I am Heribert Innoq",
       }
     ],
    "previousBlockHash": "0"
  }, {
    "index": 2,
    "timestamp": 1524086823469,
    "proof": 3288718,
    "transactions": []
    "previousBlockHash": „0000008793d0a9aa…"
  }],
  "blockHeight": 2,
}
```

GET /mine


```
{
  "message": "Mined a new block in 11.214s. Hashing power: 58854 hashes/s.",
  "block": {
    "index": 5,
    "timestamp": 1524087328713,
    "proof": 659987,
    "transactions": [],
    "previousBlockHash": "000000555398faa74ff…"
  }
}
```

Bonus 1:
Transaktionen

Transaktionen

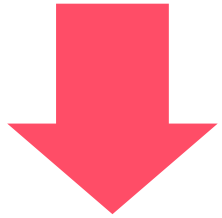
- Eine neue Transaktion wird in dem nächsten, neu gemineten Block persistiert
- Fließt damit automatisch in das Hashing ein
- Block darf max. 5 Transaktionen beinhalten
  - nicht valide wenn >5 Transaktionen

Transaktionen


```
{
  "index": 1,
  "timestamp": 1523525426,
  "proof": 124451,
  "previousBlockHash": "dsafsavew4d1as0adf001…",
  "transactions": [
    {
      "id": "ff09d94d-dd51-4df6-809f-ee50b2df3eff",
      "payload": "Arnulf Beckenbauer",
      "timestamp": 1523525426
    }
  ]
}
```

POST /transactions

```
curl -X POST http://localhost:8333/transactions \
 -H 'Content-Type: application/json' \
 -d '{
  "payload": "Arnulf Beckenbauer",
}'
```



```
{
  "id": "ff09d94d-dd51-4df6-809f-ee50b2df3eff",
  "payload": "Arnulf Beckenbauer",
  "timestamp": 1523525426,
  "confirmed": false
}
```

GET /transactions/:id


```
{
  "id": "ff09d94d-dd51-4df6-809f-ee50b2df3eff",
  "payload": "Arnulf Beckenbauer",
  "timestamp": 1523525426,
  "confirmed": true
}
```

Bonus 2:
Chain-Distribution

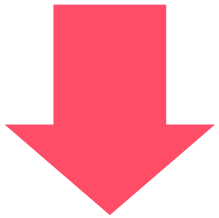
Happy hacking

Aufgabe

- Verteilung der gesamten Chain auf alle registrierten Nodes
- Billig-Konsens: Übernahme der jeweils längsten, validen Chain
- Verteilung des Transaktions-Pools (unbestätigte Transaktionen)

POST /nodes/register

```
curl -X POST http://localhost:8333/nodes/register \
 -H 'Content-Type: application/json' \
 -d '{
    "host": "http://localhost:8334"
}'
```



```
{
  "message": "New node added",
  "node": {
    "nodeId": "f788cd02-c4fd-4135-bd6d-d054b98983c9",
    "host": "http://localhost:8334"
  }
}
```

GET /events

- new_block
- new_transaction
- new_node

GET /events

```
id: 6
event: new_block
data: {
  "index": 7,
  "timestamp": 1524087602774,
  "proof": 8684026,
  "transactions": []
  "previousBlockHash": "0000004fe026ead871ce…"
}

id: 7
event: new_transaction
data: {
  "id": "ff09d94d-dd51-4df6-809f-ee50b2df3eff",
  "payload": "Arnulf Beckenbauer",
  "timestamp": 1523525426
}
```