

# BLOCKCHAIN

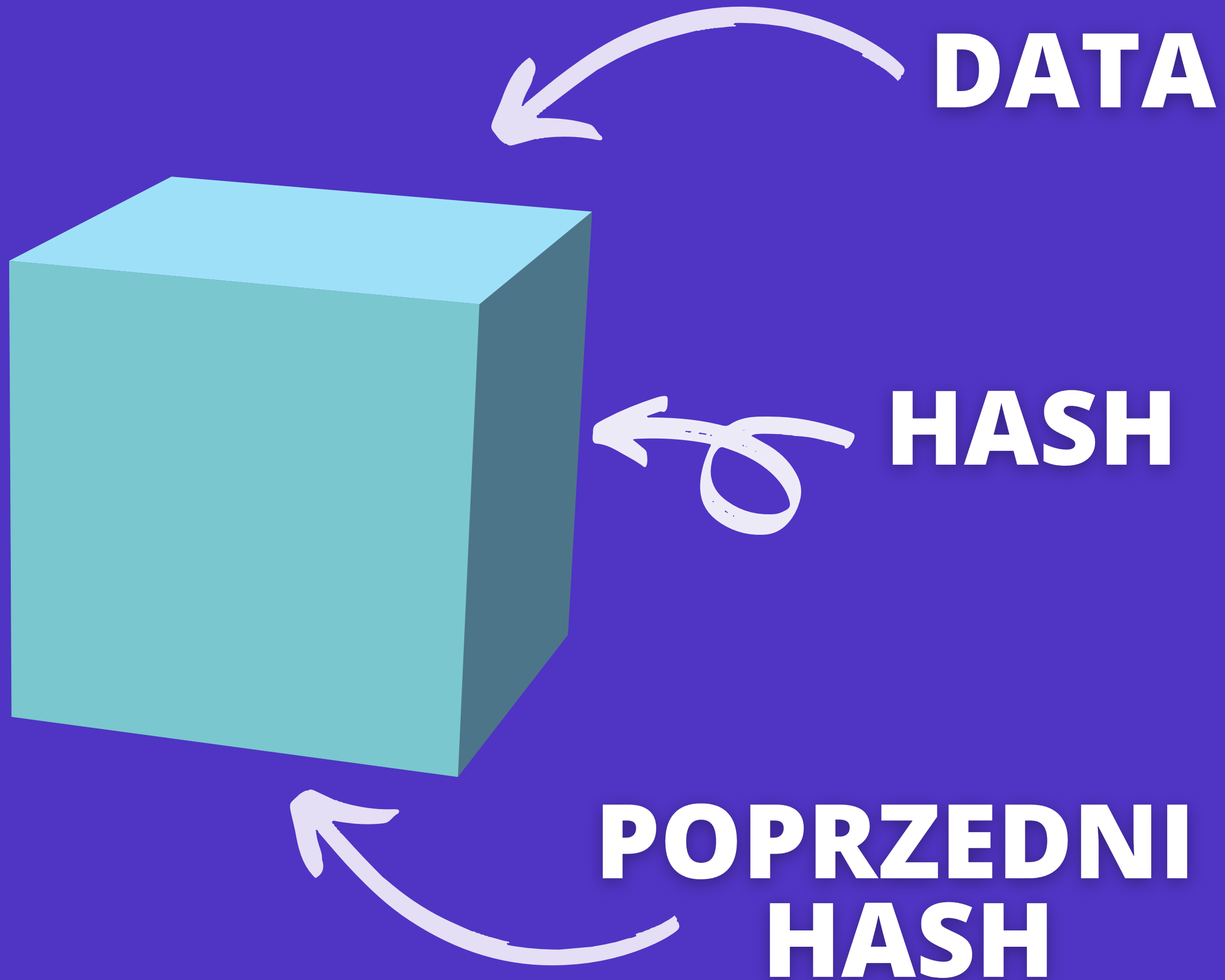
INŻYNIERIA OPROGRAMOWANIA

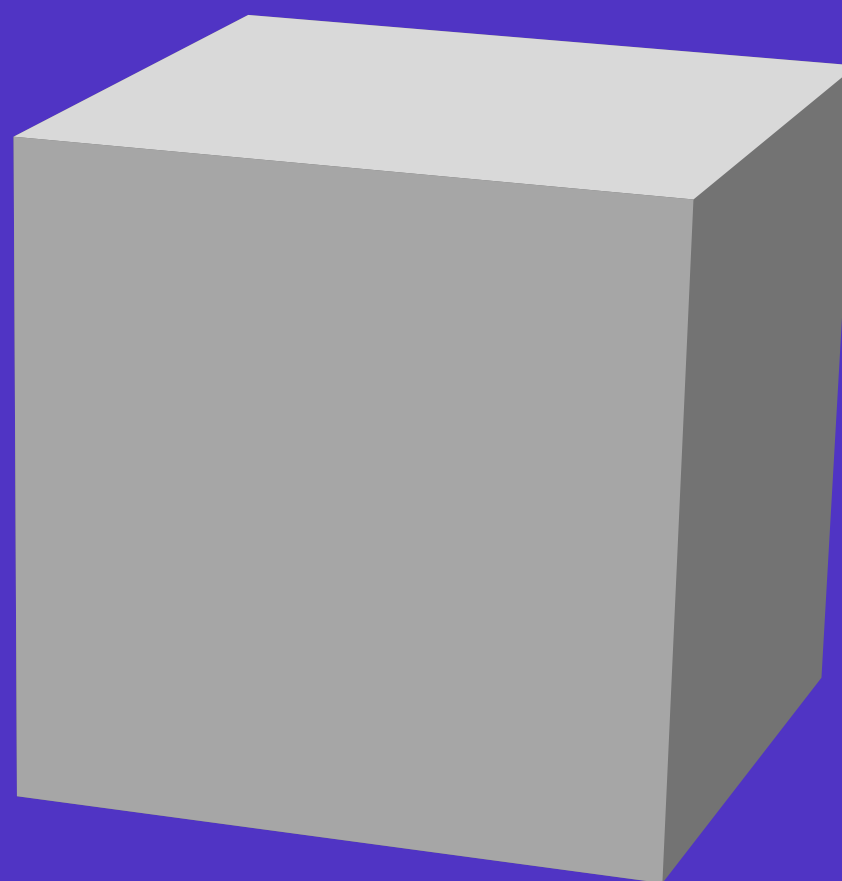
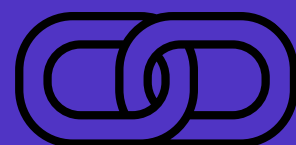
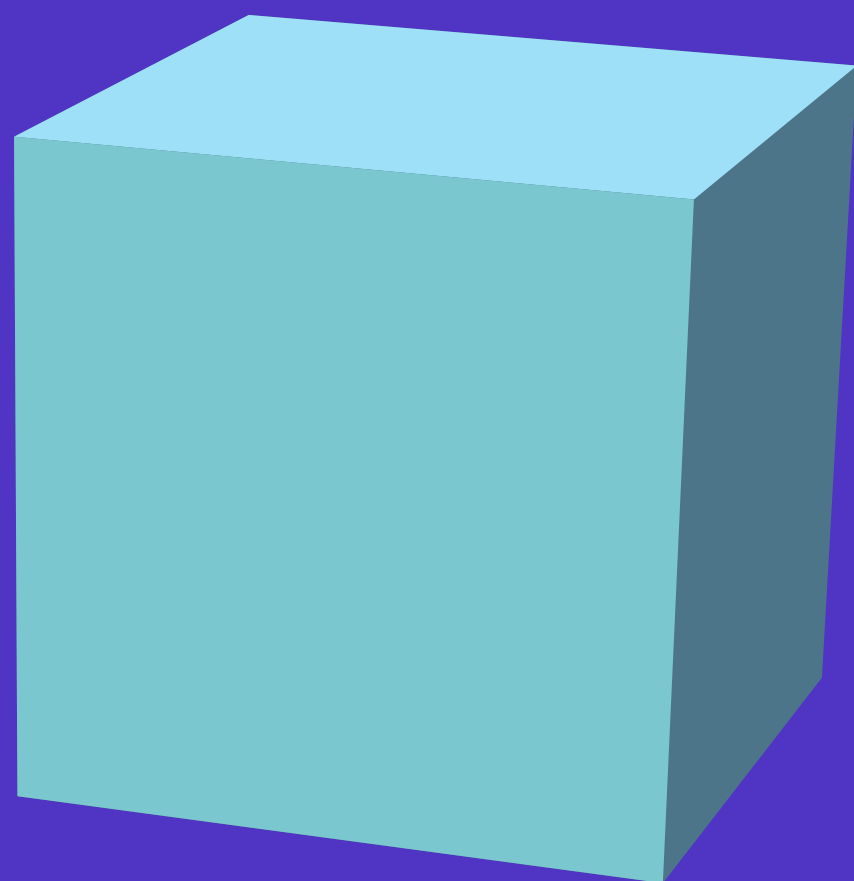


# CZYM JEST I JAK WYGLĄDA BLOCKCHAIN?



Blockchain to rozproszona baza danych, która zawiera stale rosnącą ilość informacji, pogrupowanych w bloki i powiązanych ze sobą w taki sposób, że każdy następny blok zawiera oznaczenie czasu (timestamp) kiedy został utworzony oraz link do poprzedniego bloku będącym hashem





**HASH** 6B1G

**POPRZEDNI  
HASH** 0000

**HASH** 5C1Q

**POPRZEDNI  
HASH** 6B1G

**HASH** 4Z8F

**POPRZEDNI  
HASH** 5C1Q

# HISTORIA

1991

Fizyk Stuart Haber i kryptograf W Scott Stornetta, wydali pracę badawczą „Jak oznaczyć czas cyfrowego dokumentu”

1998

Nick Szabo zaczął pracować nad pomysłem zdecentralizowanej waluty

2000

Stefan Konst zaproponował ideę kryptograficznie zabezpieczonych łańcuchów

# HISTORIA

2004

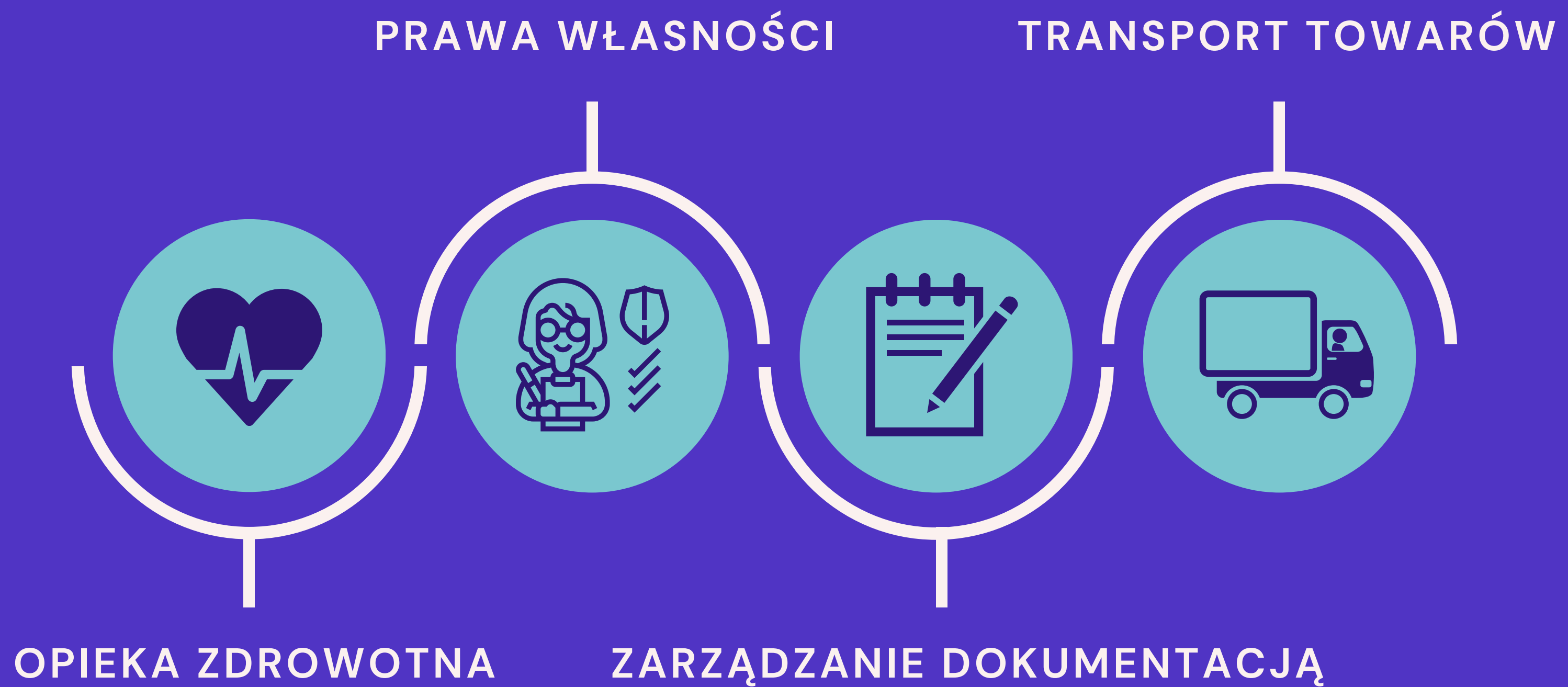
Hal Finney opracował system RPoW, Reusable Proof Of Work

9.11.2008

Fizyk Stuart Haber i kryptograf W Scott Stornetta, wydali pracę badawczą „Jak oznaczyć czas cyfrowego dokumentu”

2009

Nick Szabo zaczął pracować nad pomysłem zdecentralizowanej waluty









JAK WIDZĄ TO  
PROGRAMIŚCI?

```
class Block {
    constructor(timestamp = "", data = []) {
        this.timestamp = timestamp;
        this.data = data;
        this.hash = this.getHash();
        this.prevHash = ""; // previous block's hash
        this.nonce = 0;
    }

    // Our hash function.
    getHash() {
        return SHA256(this.prevHash + this.timestamp + JSON.stringify(this.data) + this.nonce);
    }

    mine(difficulty) {
        // Basically, it loops until our hash starts with
        // the string 0...000 with length of <difficulty>.
        while (!this.hash.startsWith(Array(difficulty + 1).join("0"))) {
            // We increases our nonce so that we can get a whole different hash.
            this.nonce++;
            // Update our new hash with the new nonce value.
            this.hash = this.getHash();
        }
    }
}
```

```
addBlock(block) {  
    block.prevHash = this.getLastBlock().hash;  
    block.hash = block.getHash();  
    block.mine(this.difficulty);  
    this.chain.push(Object.freeze(block));  
}
```

```
const { Block, Blockchain } = require("./your-blockchain-file.js");

const JeChain = new Blockchain();
// Add a new block
JeChain.addBlock(new Block(Date.now().toString(), { from: "John", to: "Bob", amount: 100 }));
// (This is just a fun example, real cryptocurrencies often have some more steps to implement).

// Prints out the updated chain
console.log(JeChain.chain);
```

```
1  import hashlib
2  import json
3  from time import time
4
5  class Blockchain(object):
6      def __init__(self):
7          self.chain = []
8          self.pending_transactions = []
9
10     self.new_block(previous_hash="The Times 03/Jan/2009 Chancellor on  
brink of second bailout for banks.", proof=100)
```

```
15 def new_block(self, proof, previous_hash=None):
16     block = {
17         'index': len(self.chain) + 1,
18         'timestamp': time(),
19         'transactions': self.pending_transactions,
20         'proof': proof,
21         'previous_hash': previous_hash or self.hash(self.chain[-1]),
22     }
23     self.pending_transactions = []
24     self.chain.append(block)
25
26     return block
27
```

```
30 @property
31 def last_block(self):
32
33     return self.chain[-1]
34
35
36 def new_transaction(self, sender, recipient, amount):
37     transaction = {
38         'sender': sender,
39         'recipient': recipient,
40         'amount': amount
41     }
42     self.pending_transactions.append(transaction)
43     return self.last_block['index'] + 1
44
```

```
47 def hash(self, block):
48     string_object = json.dumps(block, sort_keys=True)
49     block_string = string_object.encode()
50
51     raw_hash = hashlib.sha256(block_string)
52     hex_hash = raw_hash.hexdigest()
53
54     return hex_hash
55
```



```
57 blockchain = Blockchain()
58 t1 = blockchain.new_transaction("Satoshi", "Mike", '5 BTC')
59 t2 = blockchain.new_transaction("Mike", "Satoshi", '1 BTC')
60 t3 = blockchain.new_transaction("Satoshi", "Hal Finney", '5 BTC')
61 blockchain.new_block(12345)
62
63 t4 = blockchain.new_transaction("Mike", "Alice", '1 BTC')
64 t5 = blockchain.new_transaction("Alice", "Bob", '0.5 BTC')
65 t6 = blockchain.new_transaction("Bob", "Mike", '0.5 BTC')
66 blockchain.new_block(6789)
67
68 print("Blockchain: ", blockchain.chain)
```



# DZIĘKUJEMY ZA UWAGĘ

Justyna Krotoszyńska, Kajetan Warmbier

# BIBLIOGRAFIA:

<https://cryps.pl/artykul/historia-technologii-blockchain-od-1991-roku/>

<https://tokeny.pl/blockchain/>

<https://www.lazarski.pl/pl/nauka-i-badania/instytuty/wydzial-ekonomii-i-zarzadzania/centrum-technologii-blockchain/blockchain-aspekty-technologiczne-oraz-przyklady-zastosowan/>

<https://www.rp.pl/rachunkowosc/art9272021-jak-wykorzystac-blockchain-w-praktyce>

<https://efl.pl/pl/biznes-i-ty/artykuly/technologia-blockchain>

<https://itwiz.pl/praktyczne>