
GDG Rzeszów


CONCISE SOFTWARE
Building an integrated world.

Ethereum Smart Contracts.

Nowe doświadczenie dla developera?

Piotr Pawlak

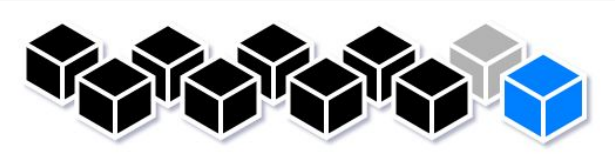
Blockchain / dApp Developer at

CONCISE SOFTWARE

@rhoqx

Czym jest Blockchain?

- Niemodyfikowalny łańcuch bloków występujący jeden po drugim



- Rozproszona baza danych zawierająca wszystkie bloki od czasu powstania
- Umożliwia bezpieczną wymianę danych przy pomocy sieci peer-to-peer
- Sieć oraz dane są dostępne publicznie - dla każdego

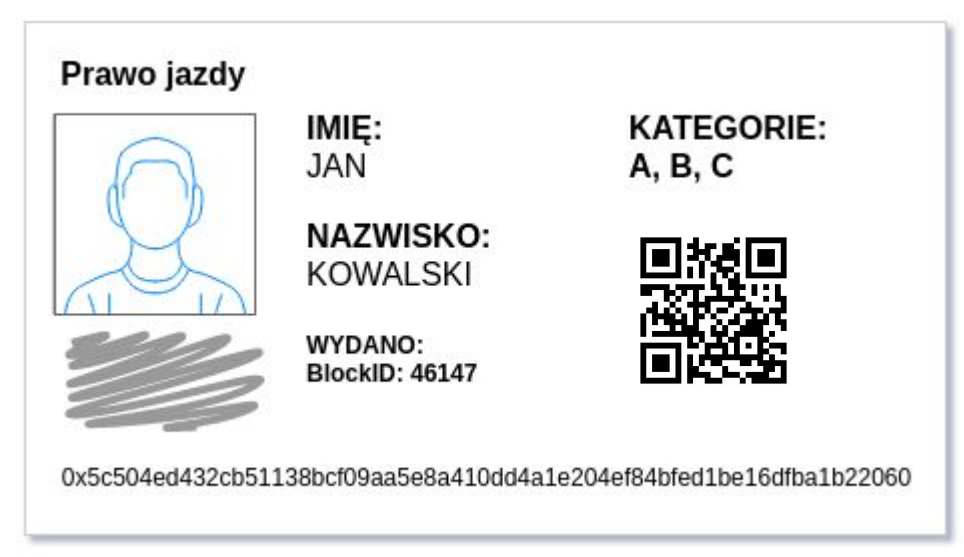
Co możemy zmieścić w bloku?

- Informacje techniczne o bloku poprzednim (hash)
- Informacje na temat aktualnego bloku
- Dane o transakcjach, tokenach
- Inne (dane o paszportach, prawach jazdy, itp.)

Co możemy zmieścić w bloku? Przykład.

Prawo jazdy - dostępne do wglądu publicznie i dla wszystkich.

Data wydania to data wydobycia bloku, w którym byłyby zapisane dane o uprawnieniach.

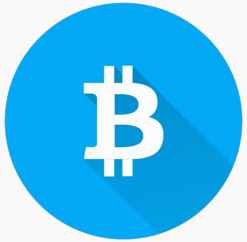


Bitcoin oraz Ethereum

- Elektroniczne waluty oparte o łańcuch bloków (blockchain)
- Wykorzystują do działania algorytmy skrótów (np. sha256) oraz kryptografię
- Transakcje potwierdzane przy pomocy Proof of Work, Proof of Stake (w testach)
- Każdy z klientów posiada taką samą bazę danych

  Bitcoin: 135 GB [2017-10-06]

  Ethereum: 300 GB (38 GB w trybie fast-sync)



Ethereum - klienci sieci oraz Solidity

- Go-ethereum (geth) - klient napisany w języku golang
- Parity - napisany w języku rust
- Cpp-ethereum - klient napisany C++

Solidity - język wysokiego poziomu stworzony do pisania kontraktów uruchamianych na Ethereum Virtual Machine (EVM). Posiada składnię podobną do JavaScript.

Ethereum - Smart Contracts

- Ethereum posiada Ethereum Virtual Machine - EVM
- EVM jest to silnik wykonujący kod skompilowanego kontraktu
- EVM posiada język skryptowy zgodny w sensie Turinga
- Umożliwia napisanie rozproszonych aplikacjach dowolnego przeznaczenia
- Wykonanie transakcji na Smart Kontrakcie kosztuje - Gaz (zapis)
- Odczyt danych z Smart Kontraktu jest darmowy
- Blok Ethereum posiada limit gazu (obecnie około 6 milionów)
- Każda operacja spala daną ilość gazu - np: wyliczenie sha256 może spalić 30 gazu

Stos technologiczny

Czego użyć? Co jest potrzebne?

- NodeJS, JavaScript
- Solidity
- Ethereum Web3 Javascript API - komunikacja z klientem przy pomocy RPC

  Inne implementacje Web3: Python Web3.py, Haskell hs-web3, Java web3j
- Klient sieci (np. geth)
- Kompilator Solidity (solc)

Narzędzia, które pomogą przy pracy z Ethereum

- Truffle Framework - ETHEREUM SWISS ARMY KNIFE
- REMIX - Przeglądarkowy edytor kodu smart kontraktów
- Prywatna sieć Ethereum
- Testowy klient sieci Ethereum symulujący pełną komunikację (testRPC)

Pierwszy kontrakt

Własna sieć - do jej stworzenia potrzebujemy:

- Geth & Tools (1.7.1)
- $ bootnode --genkey node.key
- Konto Ethereum (na prywatnej sieci)
- $ ./geth --datadir=./ethereum account new
- Konfigurację naszej sieci (plik genesis.json)
- $ ./geth --datadir=./ethereum --networkid 2456 --nodekey node.key
  Writing default main-net genesis block
- $ ./geth --datadir=./ethereum --networkid 2456 --nodekey node.key init genesis.json
  Writing custom genesis block
- NodeKey:
  self=enode://979d6e5410410581a0930dbd2bfdd31e1e4fb7425f724721fe46ca482b7462a3179b7
  9dfc816bb6ea3613925841d283645f3e13813a5c1b1e24304e672f7b46e@[::]:30303

Genesis.json

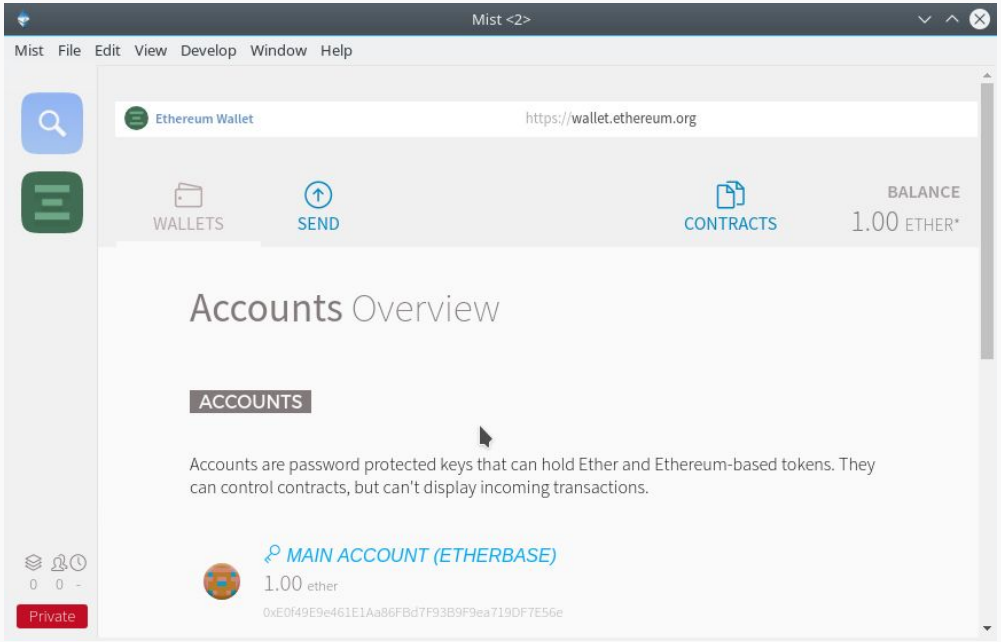
```
{
  "config": {
    "chainId": 2456,
    "homesteadBlock": 0,
    "eip155Block": 0,
    "eip158Block": 0
  },
  "difficulty": "1",
  "gasLimit": "2100000",
  "alloc": {
    "0a824020dca880540874976c451bf7c372bd9a22": { "balance": "1000000000000000000" }
  }
}
```

Działający geth

```
rhoqx@retvi1:~/workspace/blockchain$ ./geth --datadir=./ethereum --networkid 2456 --nodekey node.key
INFO [10-08|10:44:28] Starting peer-to-peer node               instance=Geth/v1.7.1-stable-05101641/linux-amd64/go1.9
INFO [10-08|10:44:28] Allocated cache and file handles         database=/home/rhoqx/workspace/blockchain/ethereum/geth/chaindata cach
e=128 handles=1024
WARN [10-08|10:44:28] Upgrading database to use lookup entries
INFO [10-08|10:44:28] Database deduplication successful        deduped=0
INFO [10-08|10:44:28] Initialised chain configuration          config="{ChainID: 2456 Homestead: 0 DAO: <nil> DAOSupport: false EIP15
0: <nil> EIP155: 0 EIP158: 0 Byzantium: <nil> Engine: unknown}"
INFO [10-08|10:44:28] Disk storage enabled for ethash caches   dir=/home/rhoqx/workspace/blockchain/ethereum/geth/ethash count=3
INFO [10-08|10:44:28] Disk storage enabled for ethash DAGs     dir=/home/rhoqx/.ethash                                   count=2
INFO [10-08|10:44:28] Initialising Ethereum protocol           versions="[63 62]" network=2456
INFO [10-08|10:44:28] Loaded most recent local header          number=0 hash=bb9eb5…b1a5b1 td=200000000
INFO [10-08|10:44:28] Loaded most recent local full block      number=0 hash=bb9eb5…b1a5b1 td=200000000
INFO [10-08|10:44:28] Loaded most recent local fast block      number=0 hash=bb9eb5…b1a5b1 td=200000000
INFO [10-08|10:44:28] Loaded local transaction journal         transactions=0 dropped=0
INFO [10-08|10:44:28] Regenerated local transaction journal    transactions=0 accounts=0
INFO [10-08|10:44:28] Starting P2P networking
INFO [10-08|10:44:30] UDP listener up                          self=enode://979d6e5410410581a0930dbd2bfdd31e1e4fb7425f724721fe46ca482
b7462a3179b79dfc816bb6ea3613925841d283645f3e13813a5c1b1e24304e672f7b46e@[::]:30303
INFO [10-08|10:44:30] RLPx listener up                         self=enode://979d6e5410410581a0930dbd2bfdd31e1e4fb7425f724721fe46ca482
b7462a3179b79dfc816bb6ea3613925841d283645f3e13813a5c1b1e24304e672f7b46e@[::]:30303
INFO [10-08|10:44:30] IPC endpoint opened: /home/rhoqx/workspace/blockchain/ethereum/geth.ipc
```

Konto Ethereum (Portfel)

- Klient Mist - przeglądarka DApps
- $ mist --rpc ./ethereum/geth.ipc



Kontrakt!

```
pragma solidity ^0.4.17;

contract Hello {
    string _text;

    function Hello(string text) public {
        _text = text;
    }

    function showText() public constant returns(string) {
        return _text;
    }
}
```

Kompilacja, BIN oraz ABI

- $ npm install -g solc
- solcjs Hello.sol --bin
- solcjs Hello.sol --abi

```
rhoqx@retvi1:~/workspace/blockchain$ cat Hello_sol_Hello.bin
6060604052341561000f57600080fd5b6040516102af3803806102af8339810160405280805182019190505080600090805
5190602001906100419291906100485565b50506100ed565b828054600181600116156101000203166002900490600052602
0600020906001f0160209004810192826001f1061008957805160ff1916838001178555610b7565b828001600101855582
156100b7579182015b828111156100b6578251825591602001919060010190610009b565b5b5090506100c491906100c856
5b5090565b6100ea91905b808211156100e6576000816000905550600101610ce565b5090565b90565b6101b3806100fc
6000396000f300606060405260003576c0100000000000000000000000000000000000000000000000000000000900463ff
ffffff16806307bc88291461003d57600080fd5b34156100485760008fd5b6100506100cb565b60405180806020018281
038252838181518152602001915080519060200190808383600005b838110156100905780820151818401526020810190
610075565b50505050905090810190601f1680156100bd5780820380516001836020036101000a031916815260200191
5b509250505060405180910390f35b6100d3610173565b60008054600181600116156101000203166002900480601f0160
208091040260200160405190810160405280929190818152602001828054600181600116156101000203166002900480
6101695780601f1061013e576101008083540402835291602001916101695b820191906000526020600020905b815481
529060010190602001808311610148c57829003601f168201915b5050505050905090565b602060405190810160405280600
00815250905600a165627a7a72305820e0caa57d53fe3b186d09af6df8a691ef77e013c2fc2fd6d38bd7c943c03a516700
29rhoqx@retvi1:~/workspace/blockchain$
```

Deployment - deploy.js

```
var text = "Hello World";
var helloContract = web3.eth.contract([{
    "constant": true,
    "inputs": [],
    "name": "showText",
    "outputs": [{"name": "", "type": "string"}],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
}, {
    "inputs": [{"name": "text", "type": "string"}],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "constructor"
}]);
```

Deployment - deploy.js

```
var hello = helloContract.new(
    text, {
        from: web3.eth.accounts[0],
        data: '<BIN DATA>',
        gas: '2100000'
    }, function (e, contract) {
        console.log(e, contract);
        if (typeof contract.address !== 'undefined') {
            console.log('Contract mined! address: ' + contract.address + ' transactionHash: '
+ contract.transactionHash);
        }
    });
```

Czas na geth!

```
$ ./geth attach ipc:/ethereum/geth.ipc
> personal.unlockAccount("0x0a824020dca880540874976c451bf7c372bd9a22",
"12345678", 0)
```

Nasz miner musi działać - coś / ktoś musi potwierdzać nasze bloki.

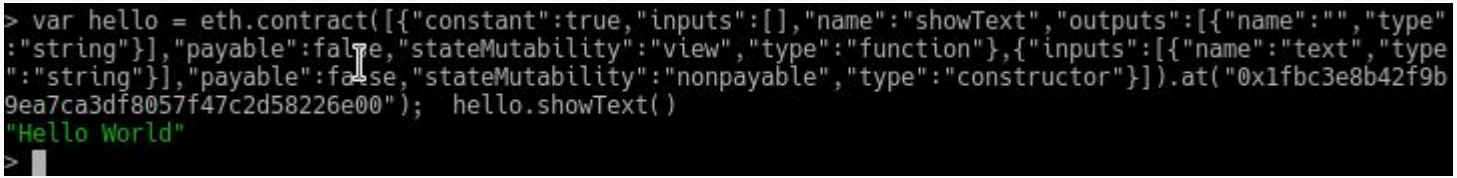
Czas na transakcję!

```
> loadScript("deploy.js")
null [object Object]
true
Submitted contract creation
fullhash=0x347f83c32bc41f862009470e5f35dfea3e1a535d1cb20c5622c8b60ff0377e07
contract=0x1Fbc3e8b42f9b9EA7CA3df8057f47C2d58226E00
> null [object Object]
Contract mined! address: 0x1fbc3e8b42f9b9ea7ca3df8057f47c2d58226e00
transactionHash:
0x347f83c32bc41f862009470e5f35dfea3e1a535d1cb20c5622c8b60ff0377e07
```

Hello World

> var hello = eth.contract([{"constant":true,"inputs":[],"name":"showText","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"},{"inputs":[{"name":"text","type":"string"}],"payable":false,"stateMutability":"nonpayable","type":"constructor"}]).at("0x1fbc3e8b42f9b9ea7ca3df8057f47c2d58226e00")

> hello.showText()
"Hello World”



Pytania?

Dzięki za uwagę!