Casper Smart Contracty

By Maciej Zieliński

Maciej Zieliński (30l.), 10 lat doświadczenia w IT.

- Politechnika Wrocławska.
- Quartic - JS, PHP, Java, Web.
- Divante - JS, PHP, Web.
- Commonwealth Bank of Australia - Scala, Big Data.
- Freelancer - Rust, Solidity, Smart Contracty.
- CasperLabs - Rust, Smart Contracty.

1. Wprowadzenie do blockchainów.

- Blockchain to baza danych.

1. Wprowadzenie do blockchainów.

- Blockchain to baza danych.
- Sieć serwerów, która osiąga konsensus.

1. Wprowadzenie do blockchainów.

- Blockchain to baza danych.
- Sieć serwerów, która osiąga konsensus.
- Blockchain składa się z transakcji.

1. Wprowadzenie do blockchainów.

- Blockchain to baza danych.
- Sieć serwerów, która osiąga konsensus.
- Blockchain składa się z transakcji.
- Transakcje są kawałkami kodu, które modyfikują stan blockchainu.

1. Wprowadzenie do blockchainów.

- Blockchain to baza danych.
- Sieć serwerów, która osiąga konsensus.
- Blockchain składa się z transakcji.
- Transakcje są kawałkami kodu, które modyfikują stan blockchainu.
- Transakcje wykonują się za pomocą maszyny wirtualnej.

1.1. Bitcoin Script

- Bitcoin VM jest maszyną stackową.

Input	Stack
6	[6]
2	[6, 2]
OP_SUB	[4]
4	[4, 4]
OP_EQUAL	[True]

- Pętle nie istnieją.

1.2. Ethereum

- Ethereum Virtual Machine (EVM) jest kompletna w sensie Turinga.

1.2. Ethereum

- Ethereum Virtual Machine (EVM) jest kompletna w sensie Turinga.
- Solidity kompiluje się do EVM Bytecodu.

1.2. Ethereum

- Ethereum Virtual Machine (EVM) jest kompletna w sensie Turinga.
- Solidity kompiluje się do EVM Bytecodu.
- Kontrakty mają swój address (hash) i pamięć.

1.3. Solidity

```
contract Token {
    mapping (address => uint256) private balances;

    constructor (uint256 tokenTotalSupply) public {
        balances[msg.sender] = tokenTotalSupply;
    }

    function balanceOf(address account) public view returns (uint256) {
        return balances[account];
    }

    function transfer(address recipient, uint256 amount) public {
        balances[msg.sender] = balances[msg.sender] - amount;
        balances[recipient] = balances[recipient] + amount;
    }
}
```

2. Architektura blockchainu Casper.

- Konsensus: PoS

2. Architektura blockchainu Casper.

- Konsensus: PoS

- Implementacja serwera: Rust

2. Architektura blockchainu Casper.

- Konsensus: PoS
- Implementacja serwera: Rust
- Maszyna wirtualna: WASM

2. Architektura blockchainu Casper.

- Konsensus: PoS
- Implementacja serwera: Rust
- Maszyna wirtualna: WASM
- Smart Contracty: Rust, AssemblyScript

3. Casper Smart Contracty - Rust.

- Struktura pliku WASM: importy, exporty, pamięć, funkcje lokalne.

3. Casper Smart Contracty - Rust.

- Struktura pliku WASM: importy, exporty, pamięć, funkcje lokalne.

- Biblioteki Rust do komunikacji z hostem: casper-contract, casper-types.

3. Casper Smart Contracty - Rust.

- Struktura pliku WASM: importy, exporty, pamięć, funkcje lokalne.
- Biblioteki Rust do komunikacji z hostem: casper-contract, casper-types.
- Kompilacja Rust do WASM.

3. Casper Smart Contracty - Rust.

- Struktura pliku WASM: importy, exporty, pamięć, funkcje lokalne.

- Biblioteki Rust do komunikacji z hostem: casper-contract, casper-types.

- Kompilacja Rust do WASM.

- Zalety: Rust i cały dojrzały ekosystem. Tysiące bibliotek. Szybkość testowania i kompilacji.

3. Casper Smart Contracty - Rust.

- Struktura pliku WASM: importy, exporty, pamięć, funkcje lokalne.

- Biblioteki Rust do komunikacji z hostem: casper-contract, casper-types.

- Kompilacja Rust do WASM.

- Zalety: Rust i cały dojrzały ekosystem. Tysiące bibliotek. Szybkość testowania i kompilacji.

- Wady: Tysiące potencjalnie niebezpiecznych bibliotek. Więcej kodu.

3.1 Casper w czystym Rust’cie.

```
#[no_mangle]

pub extern "C" fn add_one() {

    let number: i32 = runtime::get_named_arg("number");

    let result: CLValue = CLValue::from_t(number + 1).unwrap_or_revert();

    runtime::ret(result)

}
```

3.2 Casper DSL

```
#[casper_context]
struct Token {
    balances: Map<Address, Amount>;
}

#[casper_contract]
impl Token {
    fn new(mut self, total_supply: Amount) -> Token {
        self.balances[runtime::get_caller()] = total_supply;
    }
    fn balanceOf(&self, address: Address) -> Amount {
        return self.balances[address];
    }
    fn transfer(mut self, recipient: Address, amount: Amount) {
        let sender = runtime::get_caller();
        self.balances[sender] = self.balances[sender] - amount;
        self.balances[recipient] = self.balances[recipient] + amount;
    }
}
```

4.1 Przyszłość Smart Contractów.

- Systemy blockchainowe stają się globalną infrastrukturą finansową.

4.1 Przyszłość Smart Contractów.

- Systemy blockchainowe stają się globalną infrastrukturą finansową.
- Coraz więcej projektów.

4.1 Przyszłość Smart Contractów.

- Systemy blockchainowe stają się globalną infrastrukturą finansową.
- Coraz więcej projektów.
- Smart Contracty są coraz większymi projektami.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.

- Onchain: Smart Contracty.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.
- Onchain: Smart Contracty.
- Offchain: frontend, indeksery, exchange, wallety, IPFS.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.
- Onchain: Smart Contracty.
- Offchain: frontend, indeksery, exchange, wallety, IPFS.
- Audyty bezpieczeństwa.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.
- Onchain: Smart Contracty.
- Offchain: frontend, indeksery, exchange, wallety, IPFS.
- Audyty bezpieczeństwa.
- Solidity, Vyper.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.
- Onchain: Smart Contracty.
- Offchain: frontend, indeksery, exchange, wallety, IPFS.
- Audyty bezpieczeństwa.
- Solidity, Vyper.
- WASM.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.
- Onchain: Smart Contracty.
- Offchain: frontend, indeksery, exchange, wallety, IPFS.
- Audyty bezpieczeństwa.
- Solidity, Vyper.
- WASM.
- Analityka.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.
- Onchain: Smart Contracty.
- Offchain: frontend, indeksery, exchange, wallety, IPFS.
- Audyty bezpieczeństwa.
- Solidity, Vyper.
- WASM.
- Analityka.
- Kryptoekonomia.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.
- Onchain: Smart Contracty.
- Offchain: frontend, indeksery, exchange, wallety, IPFS.
- Audyty bezpieczeństwa.
- Solidity, Vyper.
- WASM.
- Analityka.
- Kryptoekonomia.
- Kryptografia: zero knowledge, podpisy.

4.2. Ścieżki kariery

- Moja droga do CasperLabs.
- Onchain: Smart Contracty.
- Offchain: frontend, indeksery, exchange, wallety, IPFS.
- Audyty bezpieczeństwa.
- Solidity, Vyper.
- WASM.
- Analityka.
- Kryptoekonomia.
- Kryptografia: zero knowledge, podpisy.
- Walidator w sieci PoS.

Pytania i odpowiedzi.