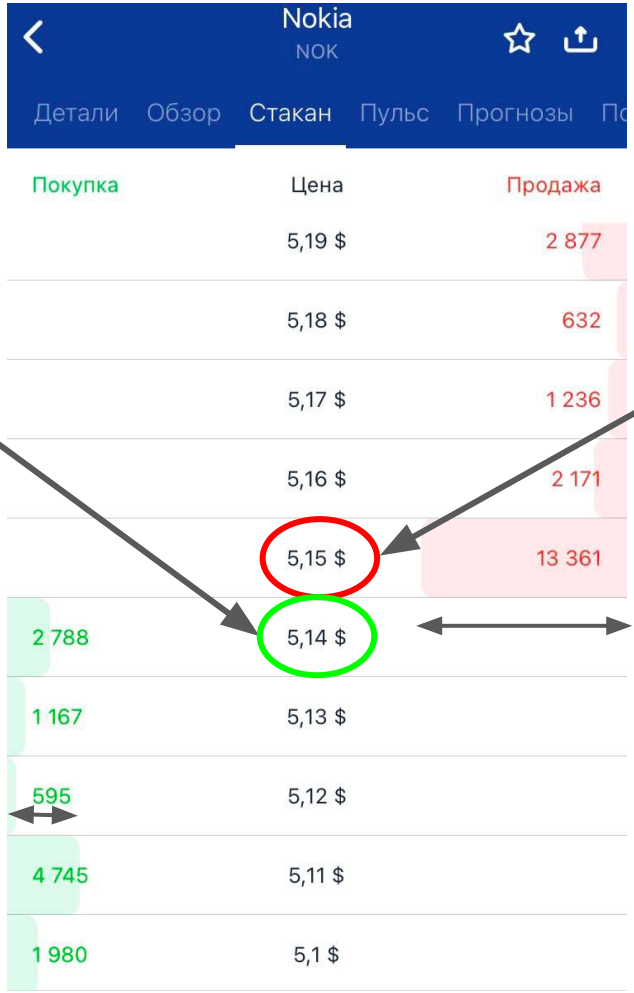
InnoDEX

[S21] Smart Contracts Development in Distributed Ledger Systems

Igor Krasheninnikov 😎 Roman Solovev 🤗 Mariia Charikova 😜
(BS17-SB)

Stock Exchange



Best Buyer Price (Highest)

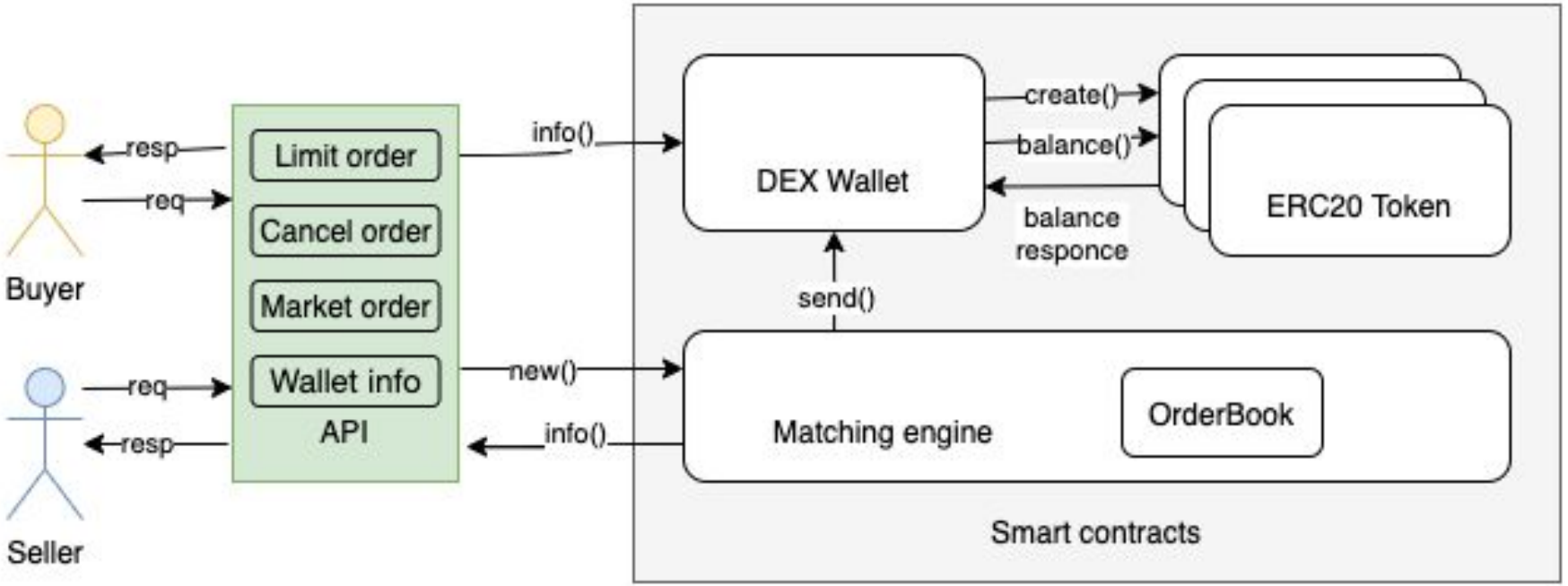
Best Seller Price (Lowest)

Spread = 0.01$

number of stocks that people want to buy at a given price

number of stocks that people want to sell at a given price

Architecture diagram



Token Contracts


```
contract ERC20 {
    string public name;
    string public symbol;
    uint8 public decimals;

    address public owner_;
    address public baseAddress;

    event Approval(address indexed tokenOwner, address indexed spender, uint tokens);
    event Transfer(address indexed from, address indexed to, uint tokens);


    mapping(address => uint256) public balances;

    mapping(address => mapping (address => uint256)) allowed;

    uint256 totalSupply_;

    using SafeMath for uint256;
```

Wallet

```
function create_wallet (address user) public {
    wallets[user] = userWallet(0, true);
}
```

```
function createToken (address user, uint256 total, string memory name, string memory symbol, uint8 decimals) public {
    require (wallets[user].valid, "You should create wallet before usage");
    ERC20 c = new ERC20(total, name, symbol, decimals, user);
    wallets[user].balances[address(c)] = total;
}
```

Wallet

```
function deposit_eth (address user) public payable {
    require (wallets[user].valid, "You should create wallet before usage");
    wallets[user].eth_balance.add(msg.value);
    ethDeposit(user, msg.value);
}
```

```
function withdraw (address payable user, uint256 amount) public {
    require (wallets[user].valid, "You should create wallet before usage");
    require (amount <= wallets[user].eth_balance, "You don't enough balance to withdraw");
    wallets[user].eth_balance.sub(amount);
    user.transfer(amount);
    ethWithdraw(user, amount);
}
```

Wallet

```
function send_token (address user, address receiver, address tokenAddr, uint numTokens) public returns (bool){
    require (wallets[user].valid, "You should create wallet before usage");
    ERC20 tmp = ERC20(tokenAddr);
    tmp.transfer(user,receiver,numTokens);
    wallets[user].balances[tokenAddr] = tmp.balanceOf(user);
    wallets[receiver].balances[tokenAddr] = tmp.balanceOf(receiver);
    return true;
}

function send_eth (address user, address receiver, uint256 amount) public view {
    require (wallets[user].valid, "You should create wallet before usage");
    wallets[user].eth_balance.sub(amount);
    wallets[receiver].eth_balance.add(amount);
}
```

Wallet

```
function eth_balanceOf (address tokenOwner) public view returns (uint) {
    require (wallets[tokenOwner].valid, "You should create wallet before usage");
    return wallets[tokenOwner].eth_balance;
}

function token_balanceOf (address tokenOwner, address token) public view returns (uint) {
    require (wallets[tokenOwner].valid, "You should create wallet before usage");
    return wallets[tokenOwner].balances[token];
}
```

```
contract MatchingEngine {

    using SafeMath for uint;

    struct Offer {
        uint amount;
        address user;
    }

    struct OrderList {
        uint nextPrice;
        uint prevPrice;

        mapping (uint => Offer) offers;
        uint firstOffer;
        uint numOfOffers;
    }

    struct OrderBook {
        mapping (uint => OrderList) buyOffers;
        uint maxBuyPrice;
        uint minBuyPrice;
        uint buyCount;

        mapping (uint => OrderList) sellOffers;
        uint minSellPrice;
        uint maxSellPrice;
        uint sellCount;
    }

    mapping (address => OrderBook) tokenBooks;
```

Matching Engine

Matching Engine

Functions:

- `buyOffer(address user, address token, uint price, uint amount)`
- `storeBuyOrder(address user, address token, uint price, uint amount)`
- `sellOffer(address user, address token, uint price, uint amount)`
- `storeSellOrder(address user, address token, uint price, uint amount)`
- `removeOrder(address user, address token, bool sellorder, uint price)`

Interface: python + flask + javascript

Name	Address	
Igor	0xc25F9650C3547fd66B3f0665f68C2a9E8b22DF23	Login
Roma	0x76AF5D41470eDE5Eb417c6ea846AaE6814d0b156	Login
Mariia	0x6a2416f4D96cE501CebB093E2779b8B6D4a5EAF9	Login
Leonid	0xc7D768AecCAc834FfbCFcba326A6db2eFC291F39	Login
Hamza	0x935D87b2f8FefbB28df4a4Cab045450e67510a62	Login

Interface

Hello, (0xc25F9650C3547fd66B3f0665f68C2a9E8b22DF23)! List of available coins:

Ticker	Address	Your Balance		
ABB	0xa747924F8b0E713e127B2830DEF988a0Fe0A922F	0	Sell	Buy
PKC	0x2a25468715927A71C1797AE2bA8693dA7c21B584	100	Sell	Buy

Wallet Dashboard

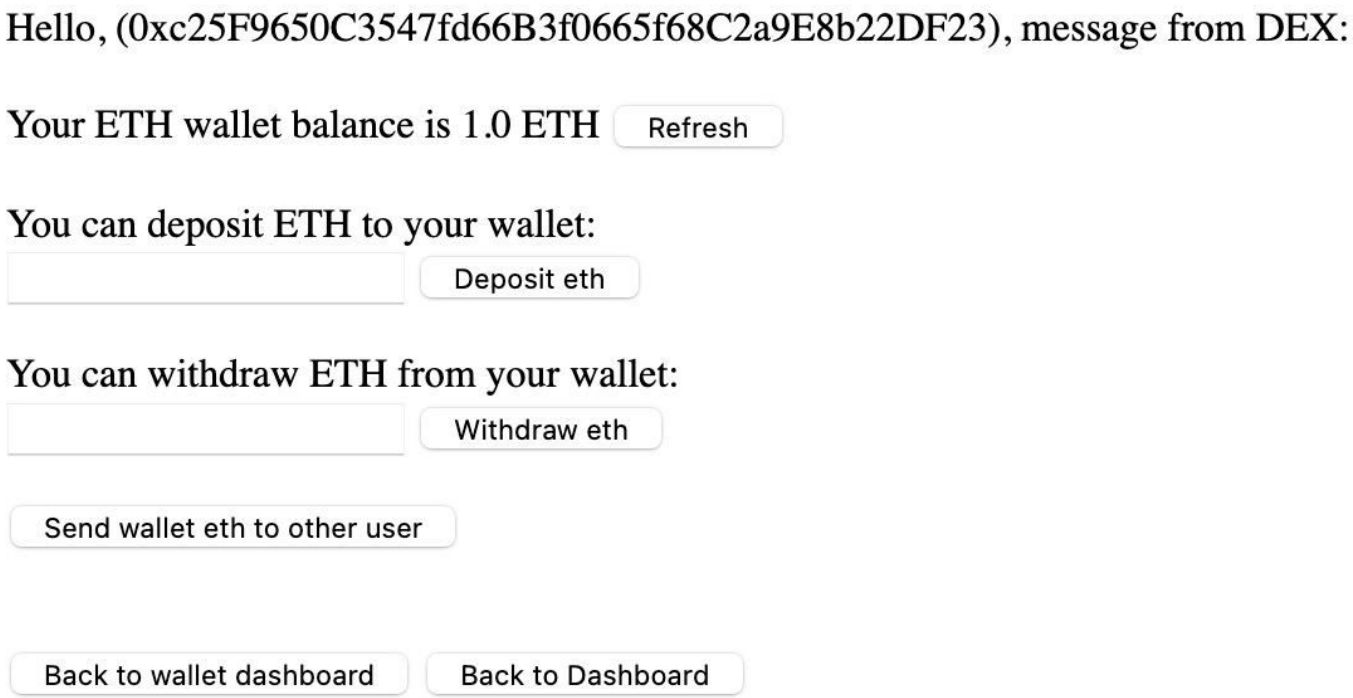
Interface

Hello, Igor (0xc25F9650C3547fd66B3f0665f68C2a9E8b22DF23)! Your ETH account balance is:

Ticker	Balance
ETH	78.4523196199

Back to accounts Wallet Dashboard

Interface


Hello, (0xc25F9650C3547fd66B3f0665f68C2a9E8b22DF23), message from DEX:
Your ETH wallet balance is 1.0 ETH
Refresh
You can deposit ETH to your wallet:
Deposit eth
You can withdraw ETH from your wallet:
Withdraw eth
Send wallet eth to other user
Back to wallet dashboard
Back to Dashboard

Interface

Hello, (0xc25F9650C3547fd66B3f0665f68C2a9E8b22DF23)

Create token form:

Total Supply

Name

Symbol

Decimals

Create token

Back to wallet dashboard

Back to Dashboard

Interface

Hello, (0xc25F9650C3547fd66B3f0665f68C2a9E8b22DF23)

Create token form:

Total Supply

Name

Symbol

Decimals

Create token

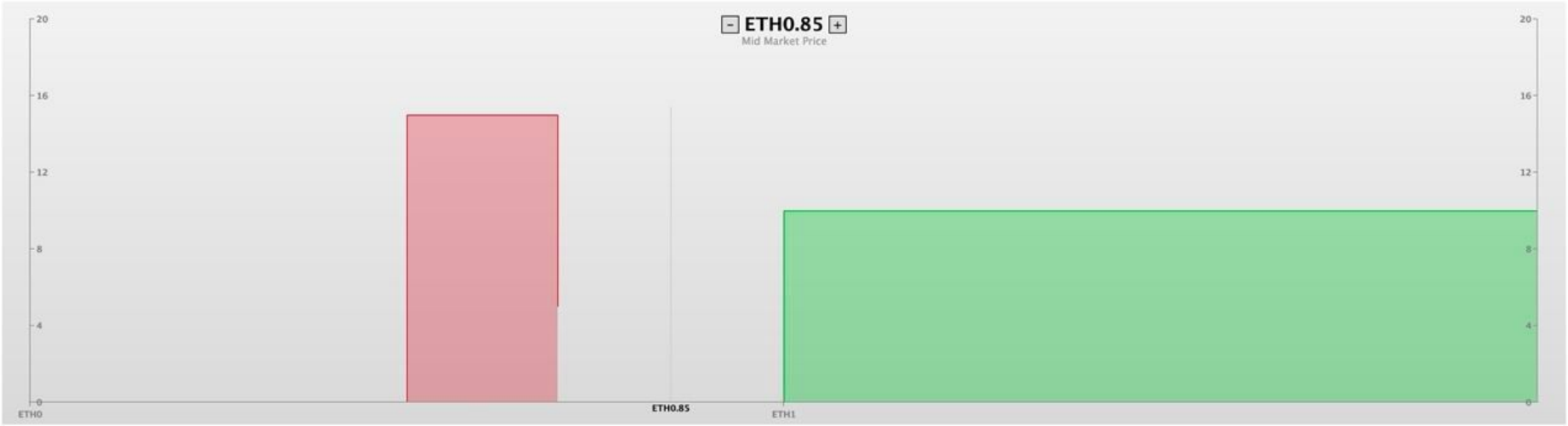
Back to wallet dashboard

Back to Dashboard

Interface

Hello, (0xc25F9650C3547fd66B3f0665f68C2a9E8b22DF23). Here is DOM for lhc:
Buys [ETH, amount]: [[0.85, 0], [0.5, 15], [0.7, 5]]
Sells [ETH, amount]: [[0.85, 0], [1.0, 10], [2.0, 20], [3.0, 30]]

Load Default Style | Load Custom Style



Back Available tokens | Back to wallet dashboard | Back to Dashboard

Demo

*buy money