Pro Magento

PRO Magento
Meetup #7

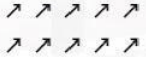
Применение блокчейн технологий в eCommerce.

Алексеенко Игорь
Senior Backend Magento Developer
IT Delight

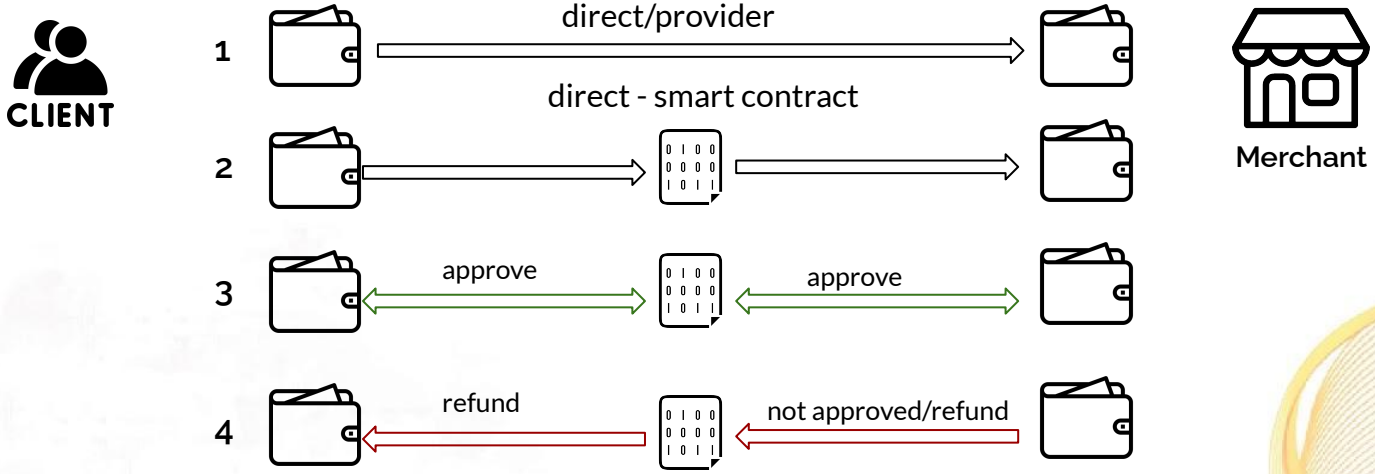
Magento Pro

О чем сегодня поговорим?

1. Способы оплаты
2. Оплата через Metamask
3. Создание, деплой и оплата через смарт-контракт
4. Практика

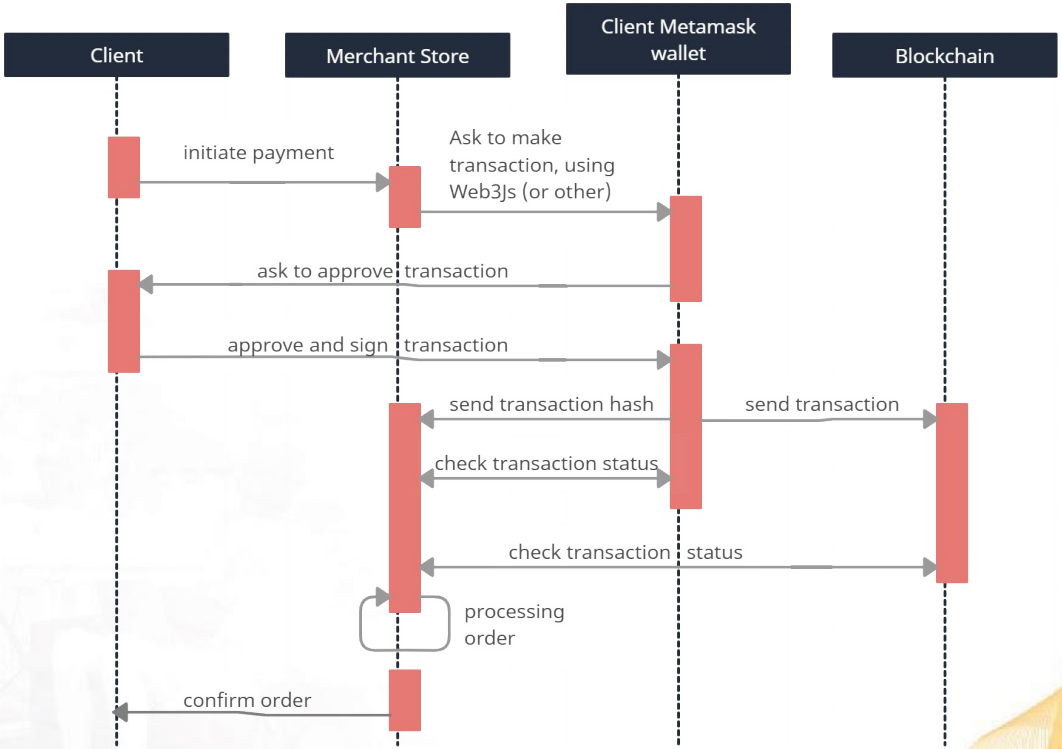


Способы реализации блокчейн оплаты



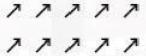
Оплата через metamask

Оплата через metamask

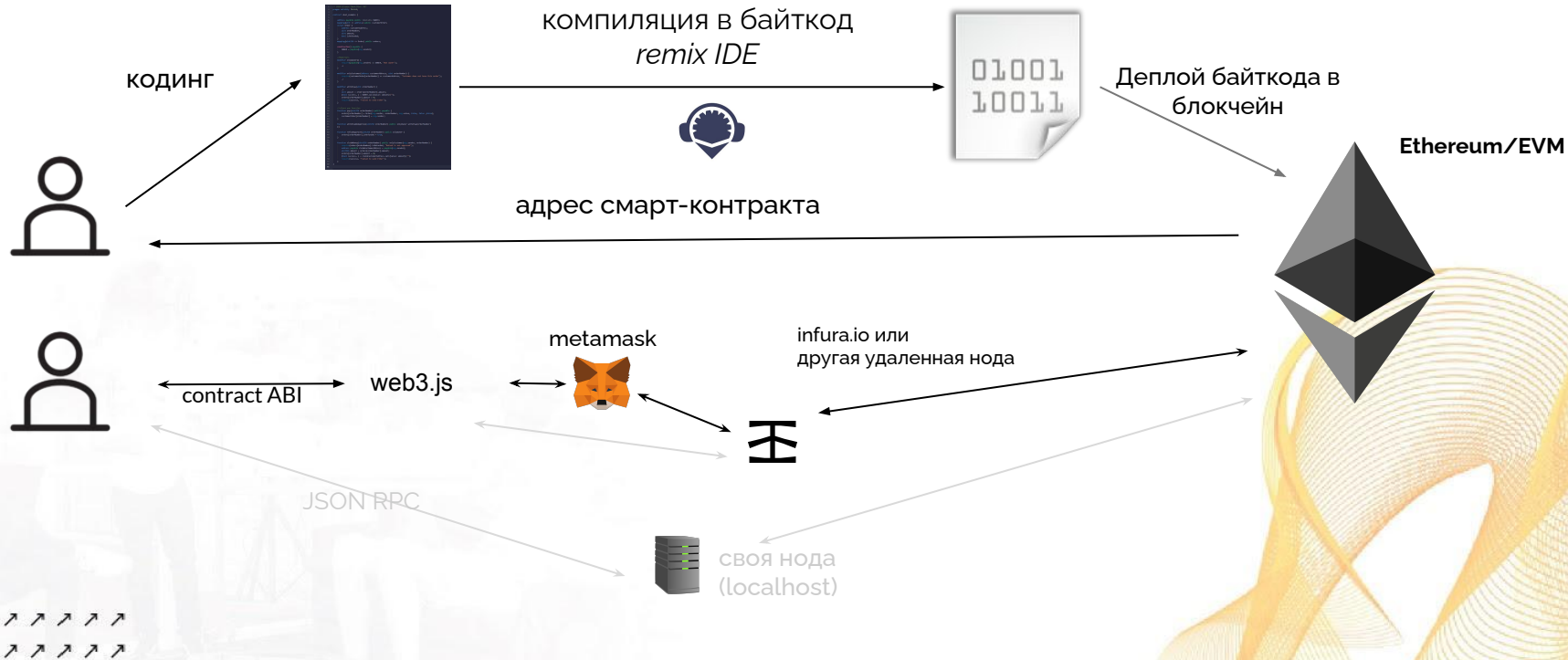


Практика. Примеры

Оплата через провайдера

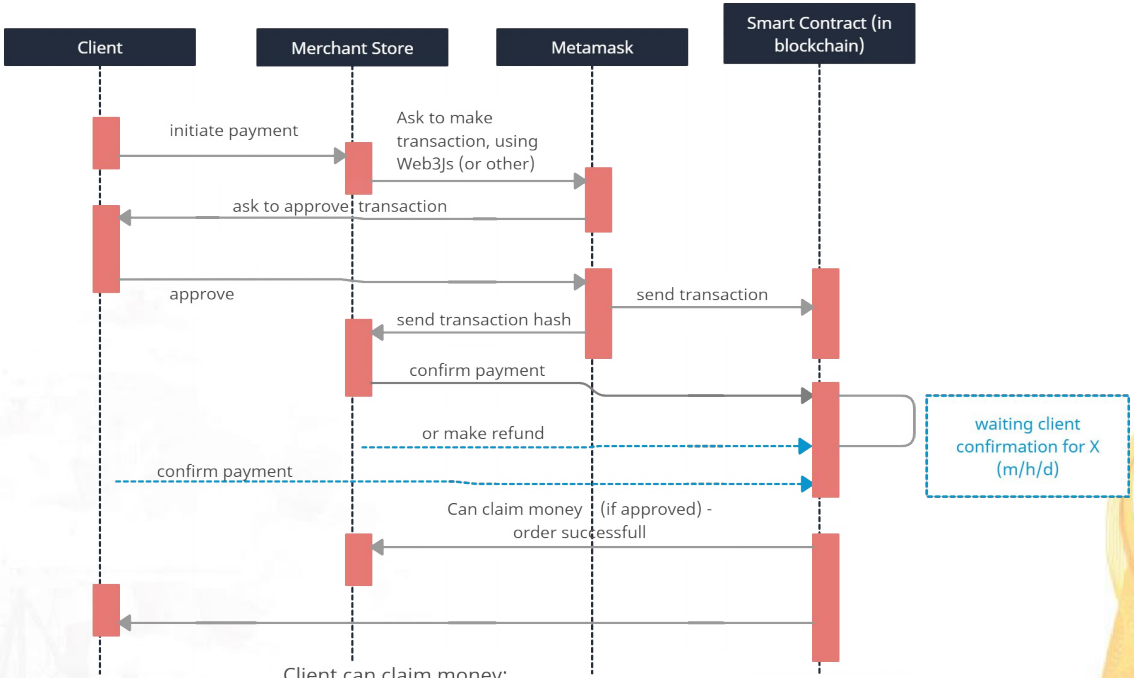


Оплата через смарт-контракт



Оплата через смарт-контракт

CLIENT



Client can claim money:
1) if not approved during some time
2) if merchant made refund

Magento Pro

Как выглядит смарт-контракт

Язык

Solidity (EVM)

IDE и деплой

Remix IDE

Solidity IntelliJ IDEA plugin

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract deal_example {

    address payable public immutable OWNER;
    mapping(uint => address) public customerOrder;
    struct Order {
        address customerAddress;
        uint orderNumber;
        uint amount;
        bool isRefunded;
    }
    mapping(uint256 => Order) public orders;

    constructor() payable {
        OWNER = payable(msg.sender);
    }

    //Modifiers
    modifier onlyOwner() {
        require(payable(msg.sender) == OWNER, "Not owner");
        _;
    }

    modifier onlyCustomer(address customerAddress, uint orderNumber) {
        require(customerOrder[orderNumber] == customerAddress, "Customer does not have this order");
        _;
    }

    modifier withdraw(uint orderNumber) {
        _;
        uint amount = orders[orderNumber].amount;
        (bool success, ) = OWNER.call{value: amount}("");
        orders[orderNumber].amount = 0;
        require(success, "Failed to send Ether");
    }

    //client pay function
    function pay(uint256 orderNumber) public payable {
        orders[orderNumber] = Order(msg.sender, orderNumber, msg.value, false, false ,false);
        customerOrder[orderNumber] = msg.sender;
    }

    function withdrawNoApprove(uint256 orderNumber) public onlyOwner withdraw(orderNumber)
    {}

    function refundApprove(uint256 orderNumber) public onlyOwner {
        orders[orderNumber].isRefunded = true;
    }

    function claimMoney(uint256 orderNumber) public onlyCustomer(msg.sender, orderNumber) {
        require(orders[orderNumber].isRefunded, "Refund is not approved");
        address payable claimCustomerAddress = payable(msg.sender);
        uint256 amount = orders[orderNumber].amount;
        orders[orderNumber].amount = 0;
        (bool success, ) = claimCustomerAddress.call{value: amount}("");
        require(success, "Failed to send Ether");
    }
}

```

Contract ABI


```
[{
    "inputs":[

    ],
    "stateMutability":"payable",
    "type":"constructor"
  },
  {
    "inputs":[

    ],
    "name":"OWNER",
    "outputs":[
      {
        "internalType":"address payable",
        "name":"",
        "type":"address"
      }
    ],
    "stateMutability":"view",
    "type":"function"
  },
  {
    "inputs":[
      {
        "internalType":"uint256",
        "name":"orderNumber",
        "type":"uint256"
      }
    ],
```

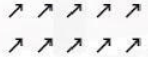

bytes4(keccak256("funcname(bytes,bool,uint256[])")

https://ropsten.etherscan.io/address/0x883af9f99e61cef58a12278f8fa6d39b779456de#code

Практика. Примеры

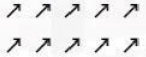
1) Смарт-контракт. Деплой смарт-контракта

2) Оплата через смарт-контракт



ProMagento

Спасибо за внимание



Ссылки

https://github.com/torys877/crypto-contract-test - модуль для теста смарт-контракта

https://github.com/torys877/crypto-metamask-eth-payment - модуль оплаты в Ether через Metamask