

dApp CardMaster

<https://github.com/gloryan0829/cardMaster>

목차

- 개요
- 사용자 인증
- 마이페이지
- 이더와 토큰 구매
- 아이템 마켓
- 용 잡기
- 회고

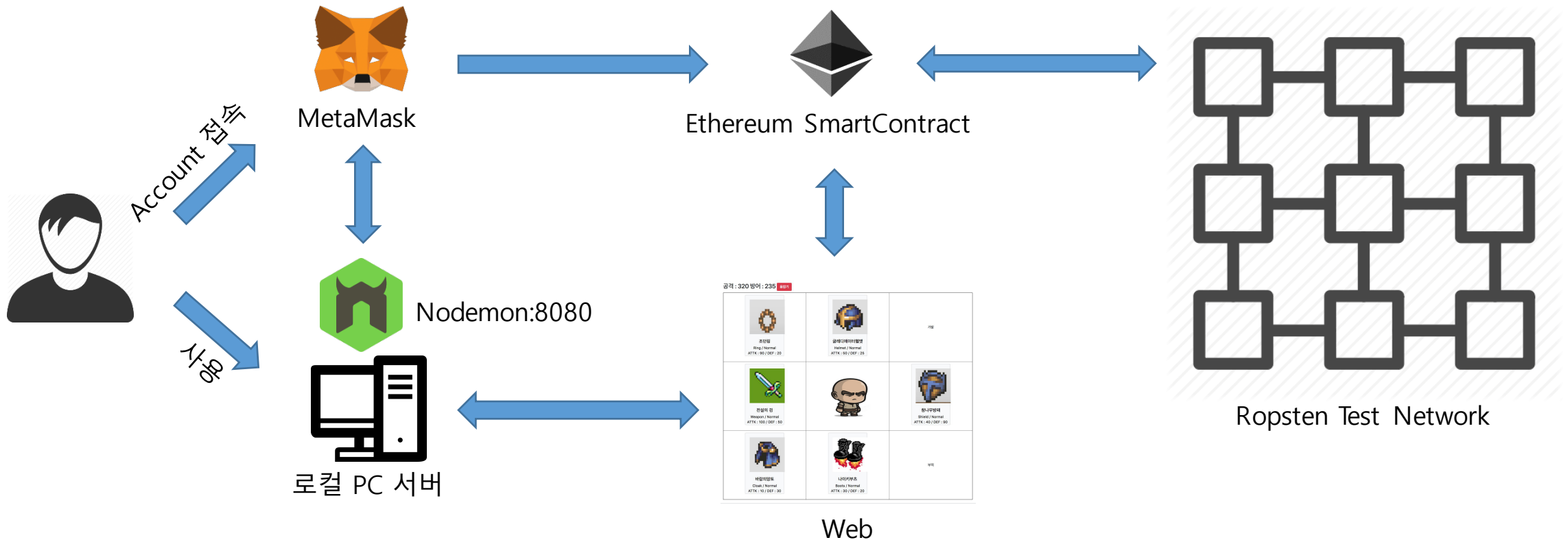
개요

이더리움 네트워크와 dApp 을 간단한 심플 카드게임 개발을 통해 이해하고자 함

사용 기술

- Solidity, Truffle Framework, NodeJS, Express, jQuery, Bootstrap, HTML


시스템 구조



사용자 인증 - 크립토키티 사례

Confirm account details

Your wallet address is Ox25777f...



Nickname Other users will see your nickname instead of your wallet address throughout CryptoKitties.

Nickname (optional)

Email options We require your email to send you product and account-related updates.

Email address

User privacy & rights We improved a few things, so please review your settings [Terms of Use](#) and [Privacy Policy](#).


I agree to CryptoKitties' Terms of Use. ☐

I agree to CryptoKitties' Privacy Policy. ☐

I want to get marketing updates (optional) ☐

Continue

Ropsten Test Net











 **Account 5** ...

Ox25777f...

3,979 ETH
1903.16 USD

BUY SEND

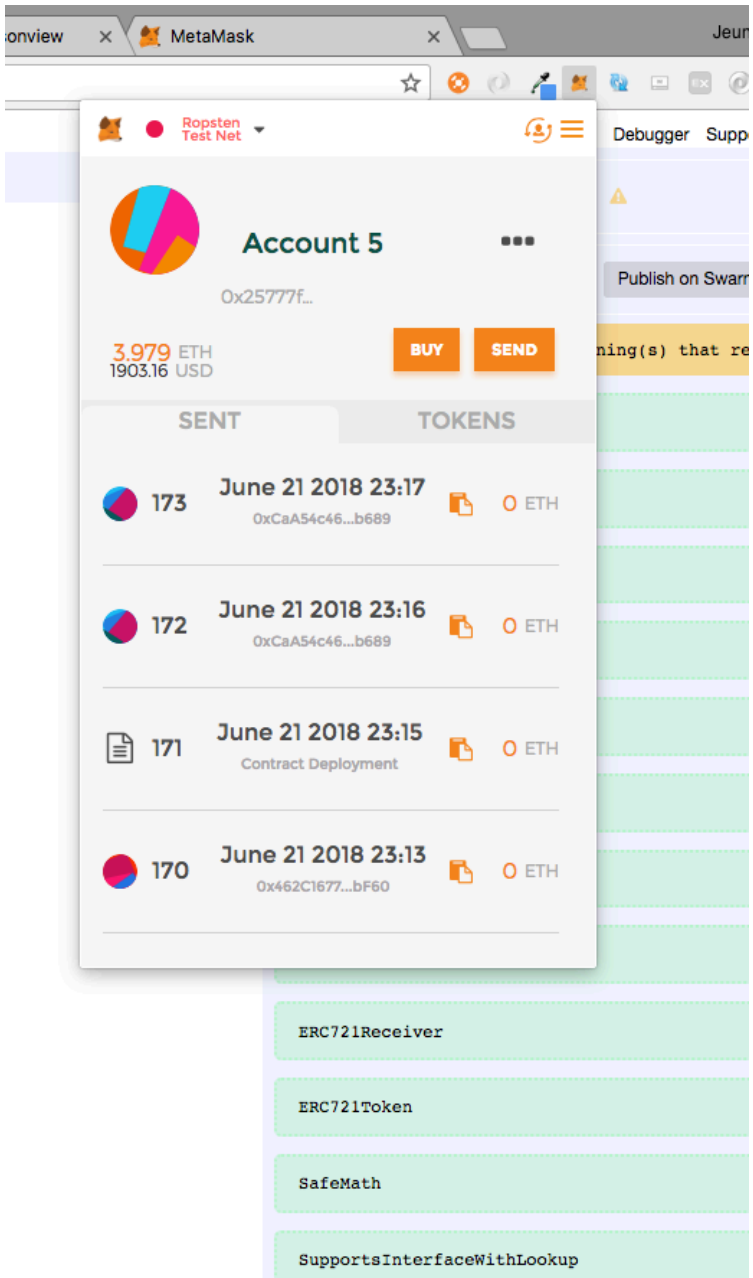
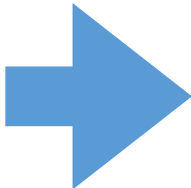
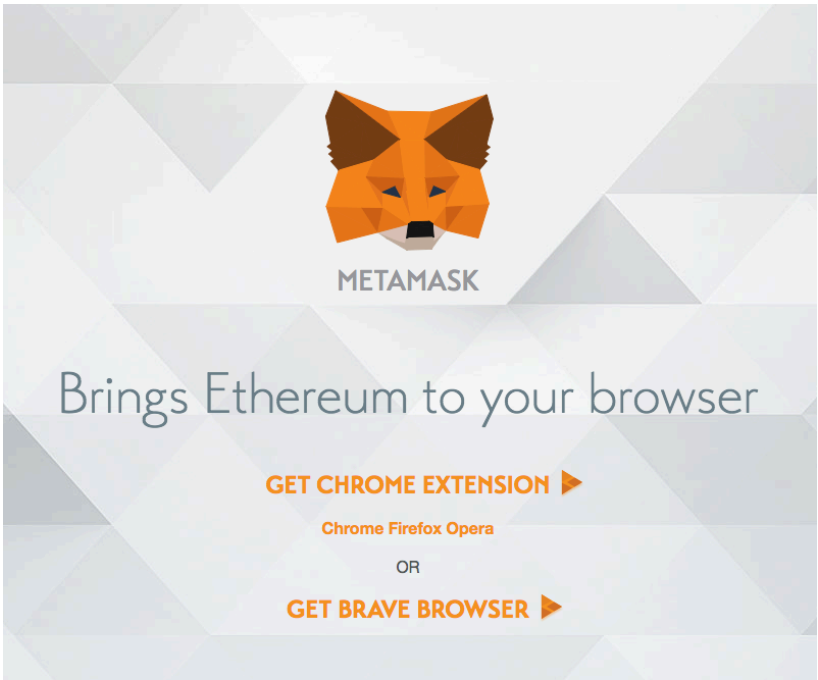
SENT TOKENS

 173	June 21 2018 23:17	 OxCaA54c46...b689	 ETH
 172	June 21 2018 23:16	 OxCaA54c46...b689	 ETH
 171	June 21 2018 23:15	Contract Deployment	 ETH
 170	June 21 2018 23:13	Ox462C1677...bF60	 ETH

회원가입시 MetaMask 본인 인증 > 네트워크 선택 > 서비스 이용

MetaMask를 통한 로그인

<https://metamask.io/>



브라우저 Web3js 라이브러리를 통해 MetaMask 와 통신하여 사용자 인증을 할 수 있다.

사용자 인증

MetaMask를 통해 본인이 계정을 관리하고 그 외 프로파일 정보를 위해 카카오톡 로그인 Open API로 접속하여 로그인 함

web3.js 를 통한 접속 - 자바스크립트 코드

```
if (typeof window.web3 !== "undefined") {  
    //MetaMask 접속 시 window 객체 web3.currentProvider 멤버 변수, 함수가 생긴다.  
    web3js = new Web3(window.web3.currentProvider);  
    console.log("Connect to Mist/MetaMask");  
} else {  
    // 직접 JSON-RPC로 해당 Peer에 접속할 수 있는 방법이다.  
    // web3js = new Web3(new Web3.providers.HttpProvider("http://localhost:9545"));  
    // console.log("Connect to Localhost");  
}
```

카카오 로그인 API



```
Kakao.Auth.createLoginButton({  
    container: "#kakao-login-btn",  
    size : "small",  
    success: function (authObj) {  
        console.log(authObj);  
        window.location = "./mypage";  
    },  
    fail : function (err) {  
        console.log(err);  
    }  
});
```

마이페이지 (1/3) Account 정보



관련 URL

```
./mypage ( mypage.html )
```

SmartContract 소스

ItemFactory.sol

Token721.sol

Token.sol

- 1 function loginStatus();
 - Kakao.Auth.getStatus 로그인 상태 체크 후 프로파일 정보 가져오기
- 2 function balanceOf();
 - 보유 이더와 ERC20으로 발급된 토큰을 가져옴
- 3 function getGasPrice();
 - <http://ethgasstation.info/> 사이트에 공시된 Gas 시장가 가져오기

마이페이지 (2/3) Character와 Item 장착 관련 정보

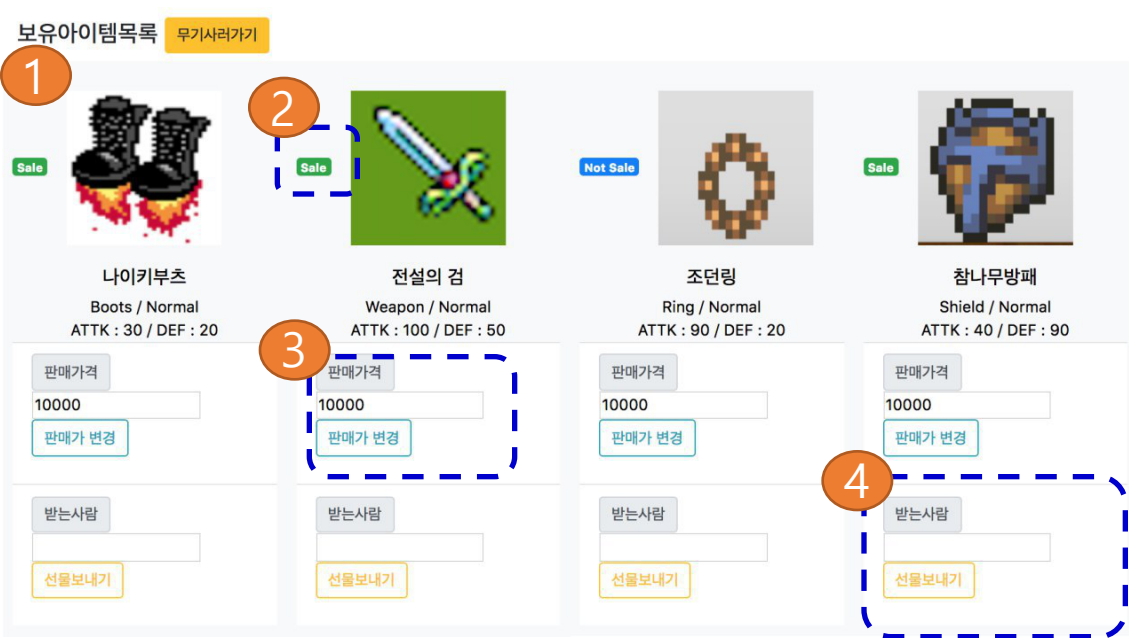
공격 : 320 방어 : 235 용접기



- 1 function equip(equipment);
 - 보유 아이템 리스트에서 아이템을 클릭 시 아이템 분류에 따라 장착되어짐 (localStorage 저장)
 - 2 function unEquip(data);
 - 해당 아이템을 화면에서 해제하고 localStorage에서 제거함
 - 3 function itemPositionSetup();
 - 페이지 로딩 시 localStorage 저장된 정보를 불러와 장착함
- ↓
- 4 무기 장착/해제 시 공격력과 방어력은 가감함

* 저장소에 대한 부분은 DB를 사용하지 않고 window.localStorage를 사용함

마이페이지 (3/3) Character와 Item 장착 관련 정보



- 1 `function ownerOfItems();`
 - 보유 아이템 리스트를 불러옴
- 2 `function saleStatusChange(tokenSeq, status);`
 - 현재 판매 상태를 바꾼다. 판매 상태를 [Sale] 로 바꿀 시 ./marketPlace 에 등록됨
- 3 `function salePriceChange(tokenSeq, price);`
 - 마켓에 등록된 판매 가격을 변경할 수 있음
- 4 `function sendGiftToFriends(_recipient, _tokenSeq);`
 - 수신 Account에게 해당 ERC721로 만들어진 아이템을 선물함

* 아이템에 대한 이미지, 공격력, 방어력 등 Attributes 에 대한 정보는 Node Server API를 통해 가져오고 해당 TokenID 는 블록체인 상에서 만들고 소유권이 누구인지 관리함 (아이템과 해당 리소스에 대한 정보는 어드민쪽에서 관리한다는 가정으로 함)

이더와 토큰 구매 (1/2) - 이더 구매

MetaMask Ether Faucet

faucet

address: 0x81b7e08f65bdf5648606c89998a9cc8164397647
balance: 9591739.77 ether

request 1 ether from faucet

user

address: 0x5cb484e7b57a05f53b65e0760c22f88fe9bf92df
balance: 11.67 ether
donate to faucet:

1 ether

10 ether

100 ether

transactions

Ropsten Network 에서는 기본적으로 Ether를 무료로 나누어주는 페이지가 존재한다. 해당 [request 1ether from faucet] 버튼을 눌러 이더를 확보하고 테스트네트워크가 아닌 별도의 Geth 를 Private Network로 활용한다면 계정에 Transfer 해주면 됨

이더와 토큰 구매 (2/2) - 토큰 구매

GameToken Wallet

내 계정 : 0x5cb484e7b57a05f53b65e0760c22f88fe9bf92df
내 보유 이더 : 11670607862299888777 wei
보유 게임 토큰 : 99999999999999999900100000 GameToken

시장의 현재 Gas Price : 1.2

토큰 선물 하기

수신자: 전송 토큰: GameToken GT

토큰 구매 하기 (판매 가능 토큰 : 0 GameToken)

이더 지출 금 : 0 wei * 1000 비율 (구입 토큰 수량 GameToken) 배율

로그:

2

1

```
function callTransfer();
```

- 해당 토큰을 누군가에게 선물할 수 있는 기능 일반적인 ERC20 규약의 transfer 함수를 이용하였음

2

```
function buyTokens();
```

- 보유이더가 있어야 함. CrowdSale 스마트컨트랙트에 이더를 전송하여 설정된 배율만큼 토큰으로 돌려받을 수 있음

관련 URL

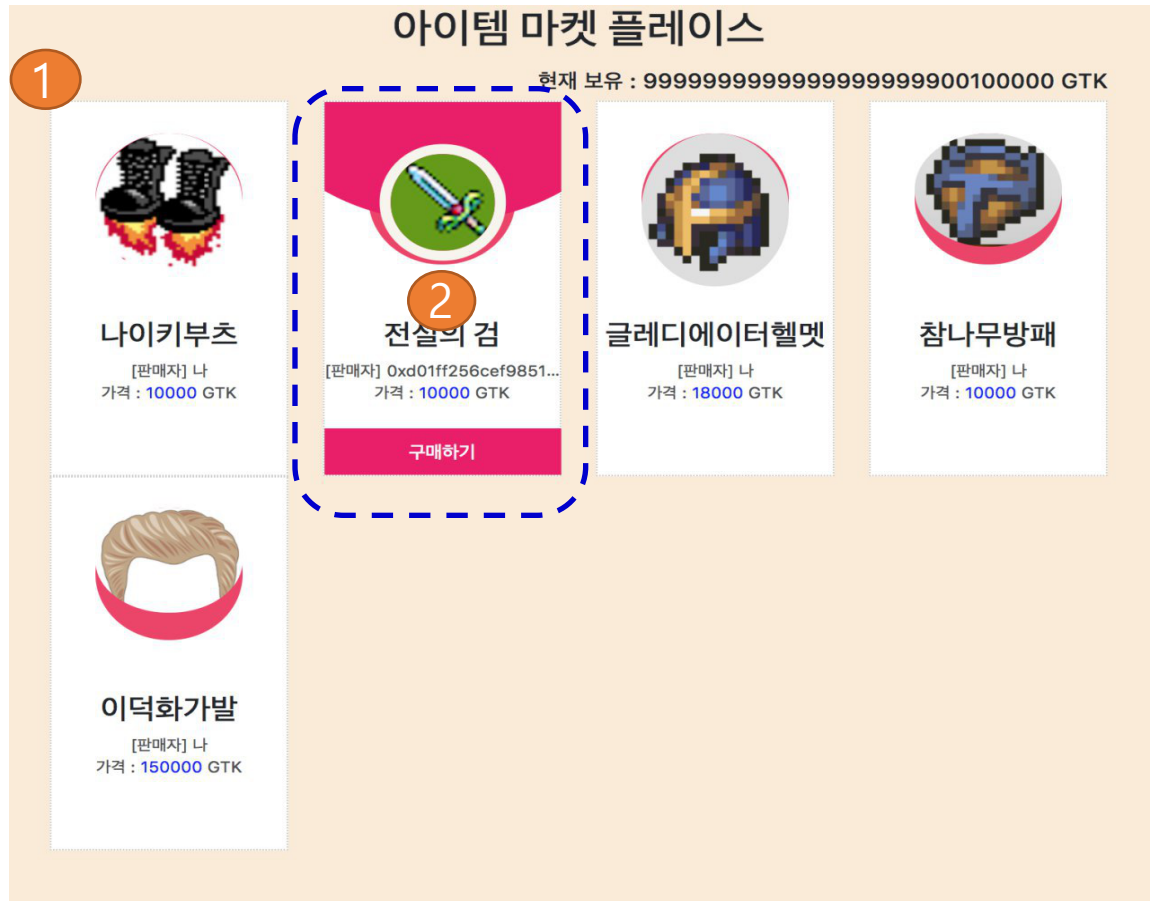
```
./tradeToken ( tradeToken.html )
```

SmartContract 소스

CrowdSale.sol

Token.sol

아이템 마켓



1

```
function cardList();
```

- 블록체인상에서 생성된 아이템을 가져오고 상태가 Sale 인 아이템에 한하여 해당 마켓플레이스에 보여줌

2

```
function purchaseItem(_price, _tokenSeq);
```

- 카드를 이더가 아닌 GTK 토큰으로 구매할 수 있는 함수임

관련 URL

```
./marketPlace( marketPlace.html )
```

SmartContract 소스

ItemTrading.sol ItemFactory.sol

Token.sol

용잡기



1 function characterApperance(ability);

- localStorage에 저장된 아이템 공격력, 방어력 총합에 따라 해당 캐릭터가 바뀔 수 있도록 함

2 function getDragon(_dragonId);

- 컨트랙트에서 생성된 랜덤 드래곤을 가져와 화면에 그리고 공격, 생명력과 공격으로 인한 남은 생명력을 보여줌

3 function damage();

- 내 공격력만큼 드래곤의 생명력에 영향을 미치게 함

4 function resetCoolTime();

- 쿨타임이 30분 지나기 전 리셋 하여 공격할 수 있게 함

5 function dragonAttackerInfo(_dragonId);

- 해당 계정에서 드래곤에게 입힌 총 데미지와 남은 쿨타임을 보여줌

관련 URL

./battle(battle.html)

SmartContract 소스

Battle.sol

DragonFactory.sol

* 드래곤은 한마리만 관리자가 생성할 수 있도록 스마트 컨트랙트를 작성함

* 게임의 많은 변수를 제외하고 최대한 심플하게 작성함

* 한번 공격 후 쿨타임은 30분이다. 쿨타임 리셋시 일정 토큰을 보낸다는 가정을 함

회고

트랜잭션 처리 속도 문제

사용자가 수수료를 지불해야 하는 문제

트랜잭션을 처리 후 사용자가 콜백을 받았을 때 UI 화면에서 Pending Transaction의 처리 방법

MetaMask를 통한 트랜잭션 처리 과정들 복잡성 (사용자 변화 관리 측면)