Blockchain Project 2022

Primiano Arminio Cristino: <u>primiano.cristino@studio.unibo.it</u> Francesco Palmisano: <u>francesco.palmisano2@studio.unibo.it</u>

Outlines

- DAPP infrastructure
 - Metamask
 - Buying & Selling
- Contract infrastructure
 - o DEX
 - o ERC-20 Token
 - o ERC-721 Token
- Security treats
 - NFT term conditions
 - o NFT forging and stealing
- Conclusion

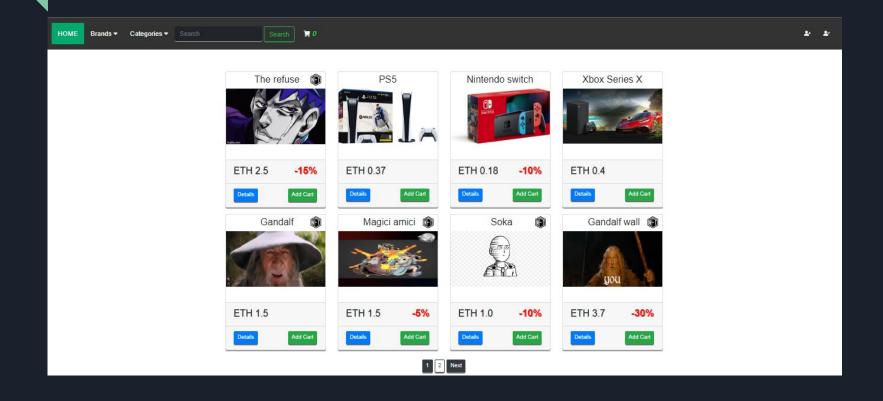
DAPP infrastructure

MetaMask

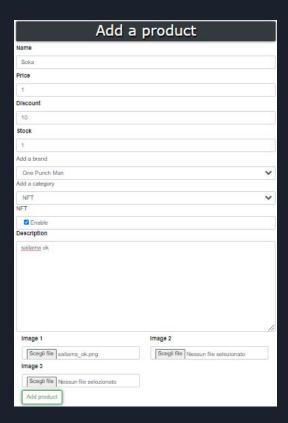


- To test the interaction with contracts we tried to use the Web3 python library.
- Web3py directly manages users private key.
- For this reason we decided to use Web3js that interacts with a proper remote wallet (e.g. MetaMask).
- Thus, we created a MetaMask wrapper to execute solidity transactions.

Home Page



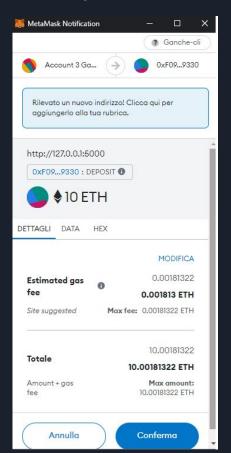
Add and edit a product

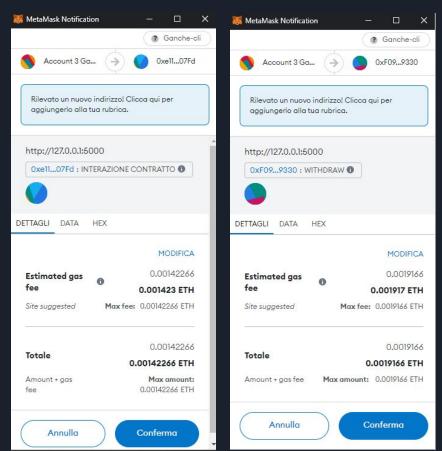


 When a product is added, the customer can decide whether the product is an NFT or not.

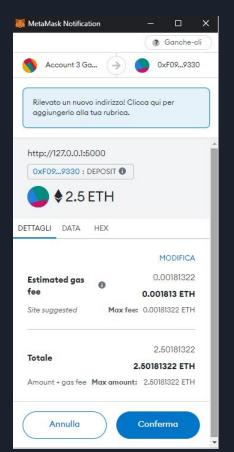
 In particular the stock and image for an NFT must be one.

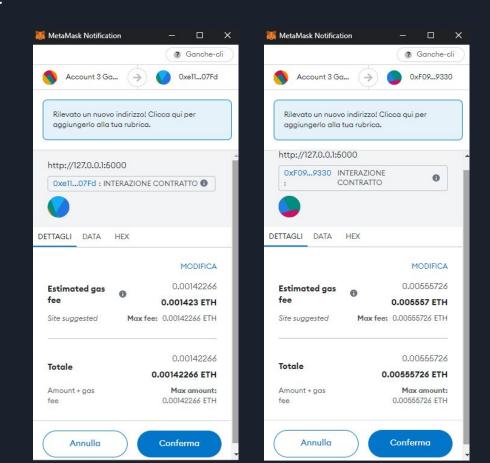
Deposit and Withdraw





Buy a Product





Contracts infrastructure

DEX (Decentralized Exchange)



DEX (Decentralized Exchange)



- Deposit/Withdraw tokens
- Add/Update/Delete products
- Store products information

ERC-20 Token



- Approve and transfer tokens from and address to another with DEX as intermediate.
- Associate a proper allowance to DEX contract.

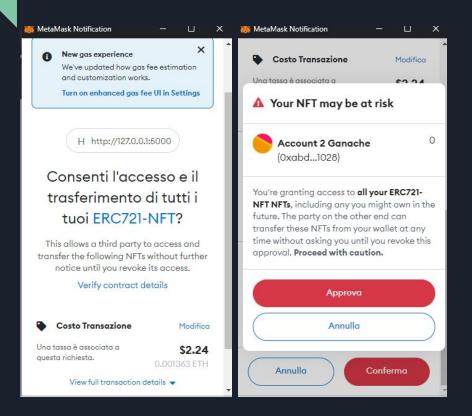
ERC-721 Token



- Generate/Burn NFTs.
- Ensure NFT uniqueness.
- Avoid NFT stealing and forging (contract-side).
- Provide NFT URI.

Security threats

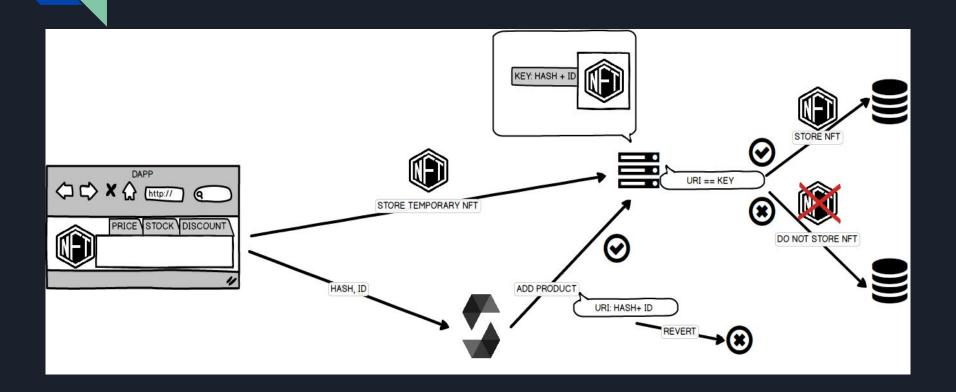
NFT Term conditions



As soon as a new customer registers at the DAPP, he/she can decide whether let the application managing its NTFs.

Otherwise, the customer can approve them singularly.

Dealing with NFT



Conclusion

Let's see the demo.