Blockchain and Distributed Ledger Technologies - Final Project

Politecnico di Milano

Student: Andrea Prati, Simone Buranti
Advisor: Prof. Francesco Bruschi

8 October 2023

Table of contents

01 Purpose of the project

02 Introduction to the use case

03 Technology and System design

04 Considerations

05 Annex

01

Purpose of the project

Purpose of the project

Explore new **use cases** for **Blockchain technology** and test **how ready** the technology is for a general B2C use, from the point of view of a business person and of the engineers hired to implement the idea.

Note: we want to provide **a proof that the technology could be already used in the real world**

02

Introduction to the Use Case

NFT - a new ownership model

Blockchain and **Distributed Ledgers** represent revolutionary technologies which are **radically changing** several **traditional domains** and are allowing completely **new scenarios**

Technology Push

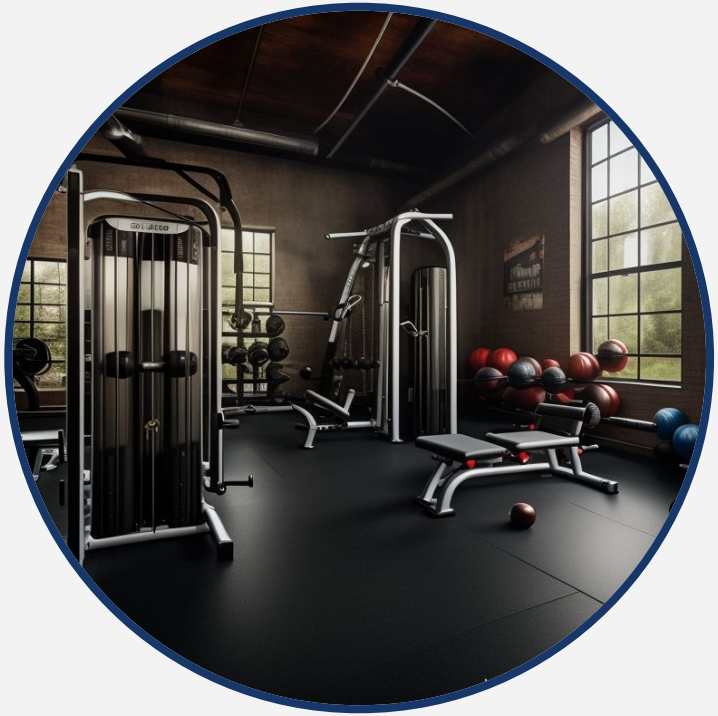
NFT - key features

- **Uniqueness**
- **Non-fungibility**
- **Programmability**
- **Ownership**
- **Transparency**
- **Transferability**

NFT as a **subscription**

The Scenario

A business person, Mr/Mrs Alex decides to build his own gym franchise. They are experienced on their field, they have already built a few businesses and now they heard about blockchain...



Use case - gym subscription

Key idea:
Currently most gym structures present a **rigid** subscription system and either **do not** allow the **reselling** of client subscriptions or **strictly constrain** it.

Pain for gym customers

Solution: implement gym subscription as **NFT** which can be **easily** resold to other users

Short-term investment

Long-term return

The players

Business person

- Why should I use blockchain?
- Advantages on the competitors?
- Drawbacks?

Tech team

- Blockchain? What?
- You sure?
- Hope to find something useful out there

Use case - proper dynamics

Business person

- **Collect** margins from business activity

Customers

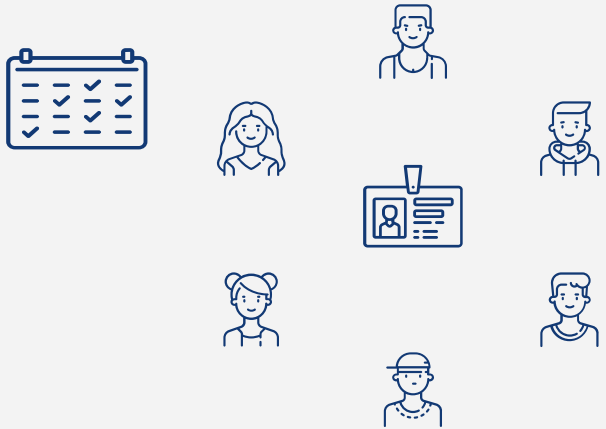
- **Subscribe** to the gym
- **Access** the gym services
- **Sell** valid subscriptions to other users at the **current value**

… freely during the validity period
… indefinitely and **without constraints**

Use case - undesired scenarios I

Since NFTs can be freely and limitlessly exchanged between clients . . .

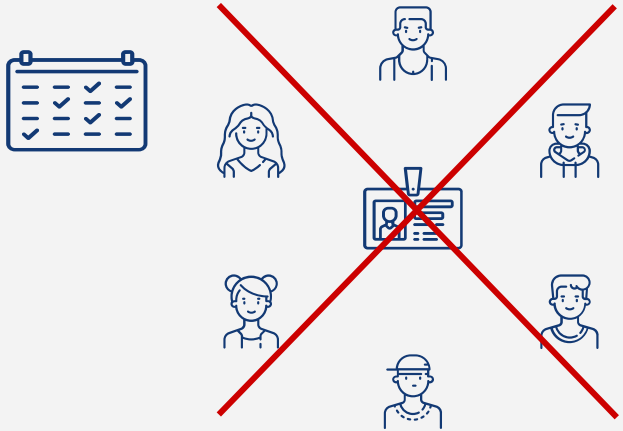
What if clients intentionally hold the subscription just for the training session and then resell it?



f.i. a group of customers might schedule the training sessions of each member so that the whole group can access the gym service with the exchange of only **one token**

Use case - solution I

Solution: each transfer of ownership **involves fees to the gym** corresponding to the **equivalent value** of **one (seven) day(s) of subscription**

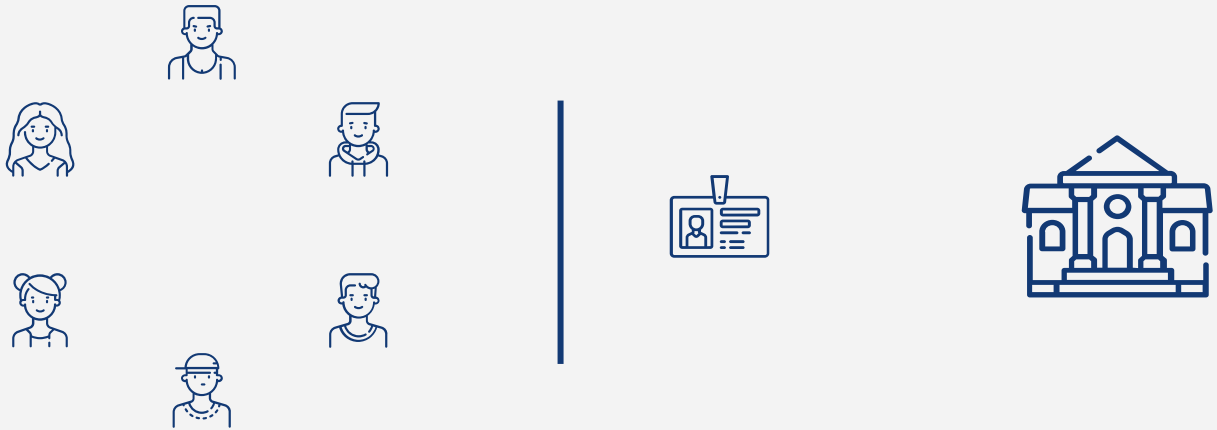


Assuming that the subscription is exchanged **every day** such a group would be of **2 - 7** people.
A fee of **(n - 1) day value** would make the “hack” equivalent to the purchase of a brand new subscription.
In the implementation we adopted a **one day value fee**

Use case - undesired scenarios II

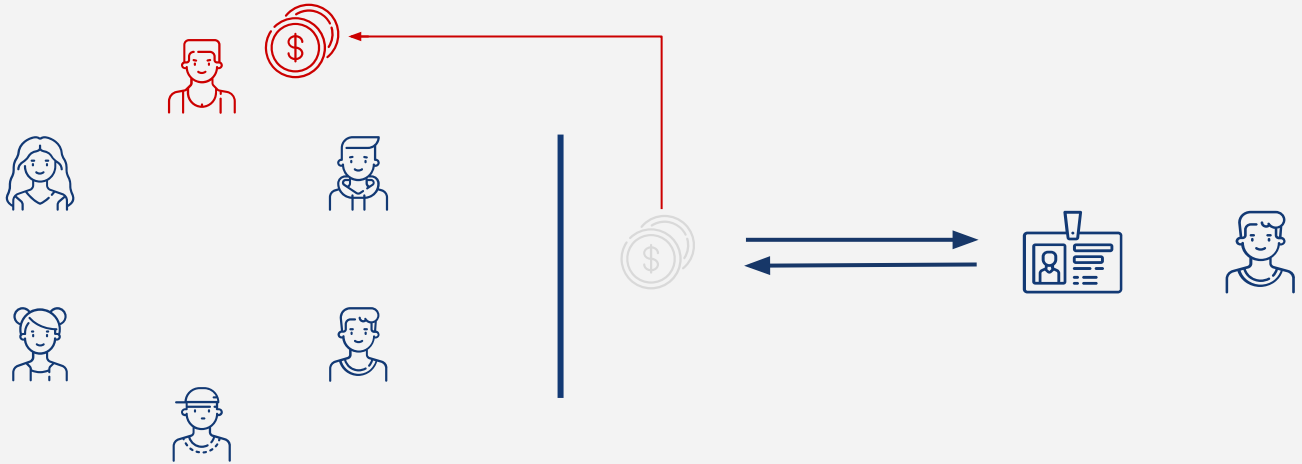
Since it is quite handy to create a new wallet ...

What if a group of "malicious" customers share the credentials of an ad-hoc created wallet to bypass the fee system?



Use case - solution II

Mitigation: since each subscription holds a **value** according to the remaining validity period, it is **worthwhile** for a group member to **sell the subscription** and **transfer the gained ethers to another wallet owned by themselves only.**



Use case - solution II

However the advanced scenario is still viable for **small groups** where "malicious" customers trust one another to keep the wallet shared.

Solution: store **off-chain** profile data on the gym back-end

Open Question: store or not to store profile information ?

Store local off-chain customer-related data

- ✓ Allows further applications in real world use cases
- ✗ Compromise the anonymity of blockchain technology

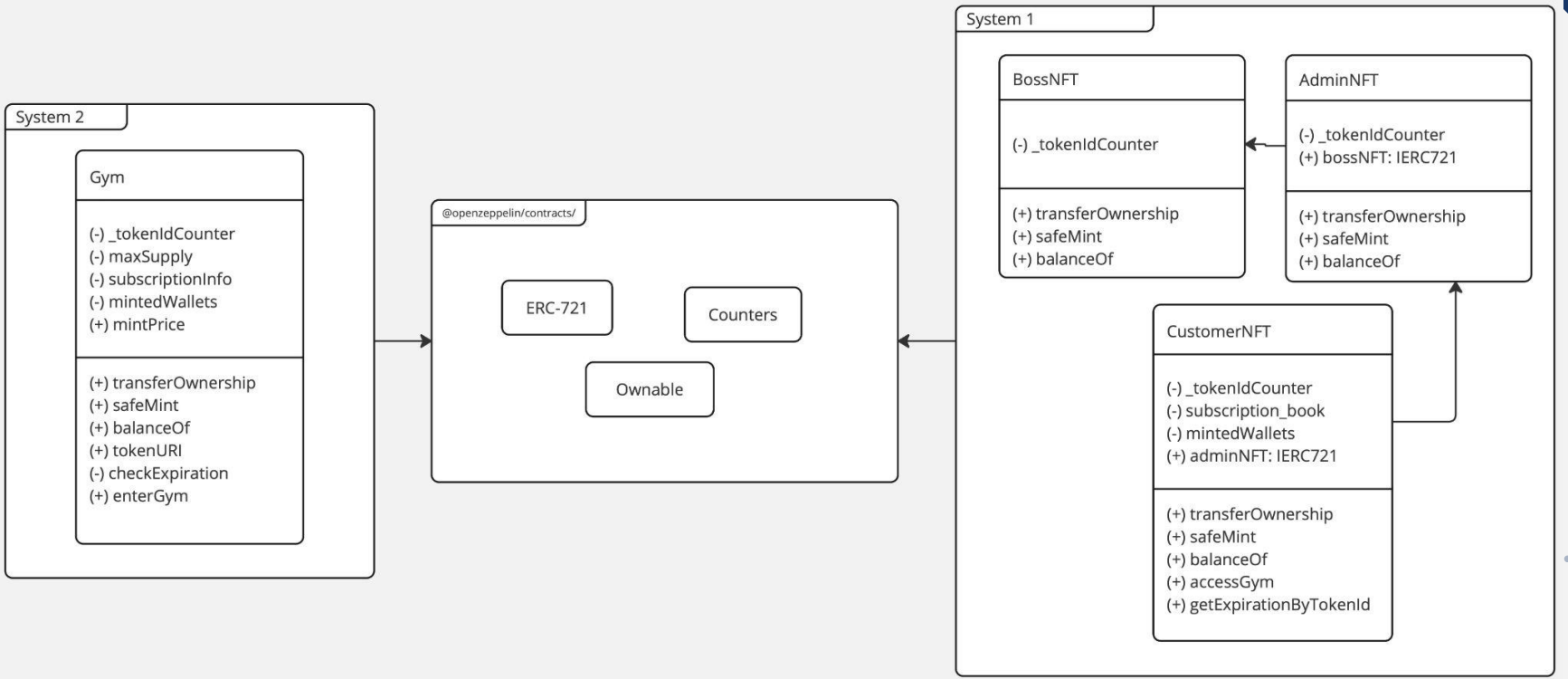
Not to store any customer-related data

- ✗ Unable to fully handle real world scenarios
- ✓ Keeps customer pseudo anonymity in real-world scenarios

03

Technology and System design

UML


System 2
Gym
(-) _tokenIdCounter
(-) maxSupply
(-) subscriptionInfo
(-) mintedWallets
(+) mintPrice
(+) transferOwnership
(+) safeMint
(+) balanceOf
(+) tokenURI
(-) checkExpiration
(+) enterGym
@openzeppelin/contracts/
ERC-721
Counters
Ownable
System 1
BossNFT
(-) _tokenIdCounter
(+) transferOwnership
(+) safeMint
(+) balanceOf
AdminNFT
(-) _tokenIdCounter
(+) bossNFT: IERC721
(+) transferOwnership
(+) safeMint
(+) balanceOf
CustomerNFT
(-) _tokenIdCounter
(-) subscription_book
(-) mintedWallets
(+) adminNFT: IERC721
(+) transferOwnership
(+) safeMint
(+) balanceOf
(+) accessGym
(+) getExpirationByTokenId

System 1 - features

Assumption: customers **are not** skilled enough to correctly handle a **crypto wallet**

- Possibility to buy the subscription as usual (FIAT money)
- Help the client to create the wallet.
- **Mobile application** to handle the subscription and the check on the subscription

System 1 - More technology used

Flutter

System 1 - Role of the players

What must they be able to do?

Owner

- To pass the ownership of the gym
- To hire Admins
- To fire Admins

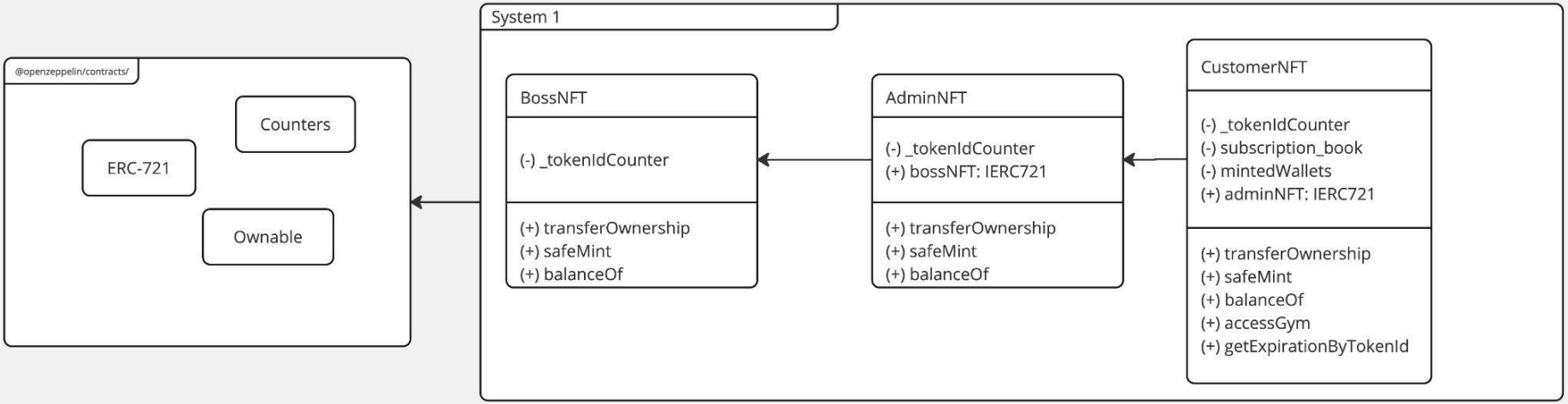
Admin

- To subscribe new customer (and extend the sub)
- To ban a customer

Subscriber

- To request subscription extension
- To subscribe to a service
- To send the subscriptions

UML - system 1


@openzeppelin/contracts/
Counters
ERC-721
Ownable
System 1
BossNFT
(-) _tokenIdCounter
(+) transferOwnership
(+) safeMint
(+) balanceOf
AdminNFT
(-) _tokenIdCounter
(+) bossNFT: IERC721
(+) transferOwnership
(+) safeMint
(+) balanceOf
CustomerNFT
(-) _tokenIdCounter
(-) subscription_book
(-) mintedWallets
(+) adminNFT: IERC721
(+) transferOwnership
(+) safeMint
(+) balanceOf
(+) accessGym
(+) getExpirationByTokenId

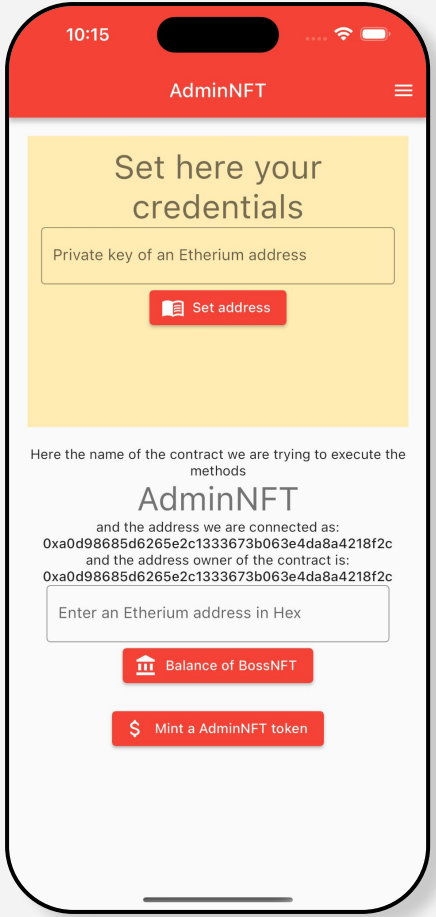
System 1 - Modifiers

```
modifier onlyAdminNFTOwner () {
    // require that the sender has at least 1 BossNFT
    require(
        adminNFT.balanceOf(msg.sender) > 0,
        "Only AdminNFT owners can call this function"
    );
    _; // placeholder for the statement
}
```

System 1 - Mobile app

Note

The current mobile application has been developed with the **purpose of proving** a full **connection with the blockchain** can be done **with a well-known technology**



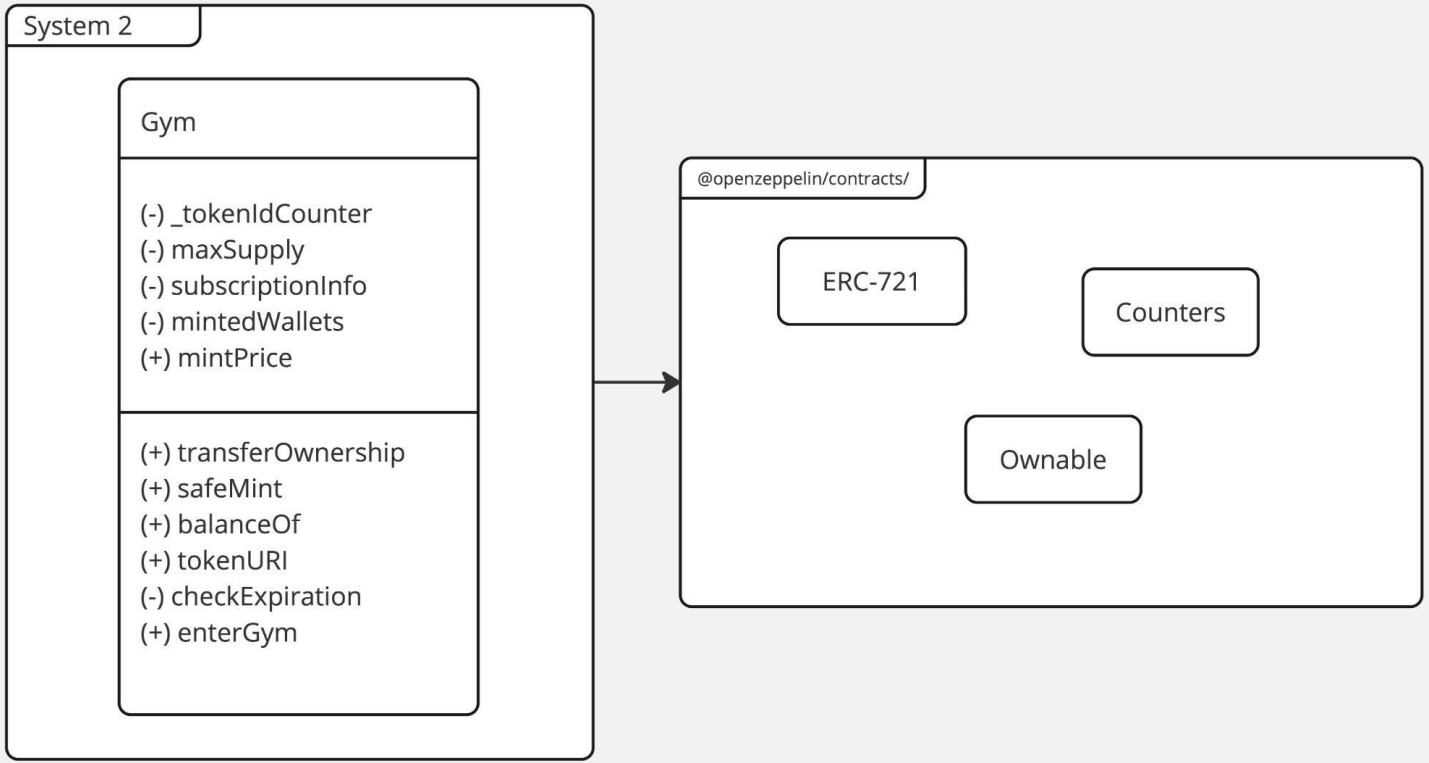
System 2- features

Assumption: customers **are** skilled enough to correctly handle a **crypto wallet**

Long-term solution

- Direct interactions with the **smart contract** handling the business of the gym
- Interactions with the gym occur completely **online**
- Benefit from an **agile acquisition** and **exchange** of subscriptions

UML - system 2


System 2
Gym
(-) _tokenIdCounter
(-) maxSupply
(-) subscriptionInfo
(-) mintedWallets
(+) mintPrice
(+) transferOwnership
(+) safeMint
(+) balanceOf
(+) tokenURI
(-) checkExpiration
(+) enterGym
@openzeppelin/contracts/
ERC-721
Counters
Ownable

System 2 - design choices

On-chain metadata
Light static metadata → completely **on-chain**

```
struct Info {
    uint256 price;
    uint256 expirationDate;
}
```

Subscription reselling
Customers can **publish** their **"for sale"** subscription on the **Gym.sol Smart Contract**, where interested users can purchase them for a **reduced price (rescaled on remaining days)**.

Subscription expiration
Subscription expiration is **hardcoded** in the token metadata and checked on customers' interactions

Withdraw
The SC **owner (only)** can withdraw the current balance whenever they want

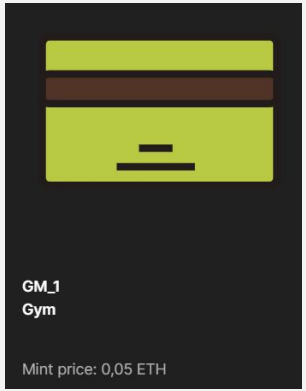
System 2 - deployed

Etherscan

[This is a Sepolia **Testnet** transaction only]

Transaction Hash:	0xea6f54315c32d82f65c44416aa4c7aeb81a8f87bafca926c55a8b22c1ef3b7e5
Status:	Success
Block:	4587989 716 Block Confirmations
Timestamp:	2 hrs 35 mins ago (Oct-29-2023 02:46:12 PM +UTC)
Method:	0x60806040
From:	0x37f83864E608fe7983Cb89FaE42F8324d55B8E55
To:	[0xc552699d9b4e6773e5c6e2cce86a490e07c6e5e5 Created]
Value:	0 ETH ($0.00)
Transaction Fee:	0.011868480056968704 ETH ($0.00)
Gas Price:	2.500000012 Gwei (0.000000002500000012 ETH)

OpenSea



04 Considerations

Consideration

- Development experience
  - Dynamic technology, hard to follow best practices and to find updated solutions (our source of truth has been OpenZeppelin) ✗
  - Confused community ✗
  - Tools are often bugged ✗
  - Control system fast to develop (nature of blockchain) ✓
- Business
  - Need less administration ✓
- General feedback
  - Disruptive innovation → various use-cases like luxochain, gaming, and so on. ✓

Thank you for the attention

Andrea Prati
Simone Buranti

Questions?

andrea.prati@mail.polimi.it
simone.buranti@mail.polimi.it