TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Vladimir Andrianov 203860IABM

MetaMex NFT marketplace platform development on the Elrond blockchain

Master's thesis

Supervisor: Avar Pentel
MA

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Vladimir Andrianov 203860IABM

MetaMex NFT turuplatvormi arendus Elrondi plokiahelas

Magistritöö

Juhendaja: Avar Pentel

MA

s

Tallinn 2022

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Vladimir Andrianov

28.04.2022

Abstract

We are entering the fourth industrial revolution, and we have heard a lot about the global-scale changes and narrative of our lives from the World Economic Forum. The fourth industrial revolution was accelerated by constant lock-downs enforcing the massive digitalization and automation of all societal processes. The World Economic Forum executive chairman Professor Klaus Schwab also mentioned that "The pandemic represents a rare but narrow window of opportunity to reflect, reimagine, and reset our world" [1]. The world's central banks are discussing conversion to digital currencies as an alternative to the model of money we previously used to see. Digital currencies such as the digital renminbi, ruble, euro, and dollar will take the first place soon as the world moves more to digital assets. Facebook turned into Meta preparing for the metaverse already, and the global industries adopt decentralized technologies quickly, the game and art industries adopt blockchain crypto space for NFT trading, while slowly everything is moving towards the integration with the metaverse, where the borders will not exist anymore, and we will live the new way.

Alongside the above, the technologies evolve faster than the society thinks of it, when the fourth industrial revolution was announced in the international newspapers, many of us found that some places of the world have not passed the third industrial revolution even, the same processes are happening to the Web industry too, the Web 3.0 marches twenty paces, leaving the web 2.0 far behind. The Decentralized networks with decentralized autonomous organizations have become the new reality already, and these are not just theories in the books we read anymore. The global industry has started adopting Web 3.0 concepts for business processes already, its main idea is striving for openness and transparency. Blockchain is the connecting link between transparency and decentralization, focusing on security and integrity. Third-generation blockchains quickly adopt the concepts of smart contracts and non-fungible tokens, distributed networking and distributed storage, all of this facilitates the Web 3.0 ideas. It is an uncharted land with the potential to become one of the world's largest industries.

The main problem of the Web 3.0 industry is too much demand in the market, which means that the current number of experienced developers is far from meeting the need for a workforce. The blockchain technology stack requires much research before it can be used effectively. The marketplaces and digital wallets that are new for mass media are already out of date and collecting the technical debt.

The transactions on Blockchain and Ethereum networks are too expensive and slow, it hinders the development of the market and makes many products financially and economically nonviable.

The purpose of the research is to obtain an overview of third-generation blockchains compared to Bitcoin (first generation) and Ethereum (second generation), get actual transaction costs and speed on Elrond (third-generation) blockchain and evaluate platform financial viability with the actual infrastructure costs. The research is composed of a blockchain theory and hybrid application following the Web 3.0 principles, which is an NFT marketplace platform on Elrond, the underlying work covers the entire team and project planning, development of platform front-end, back-end, smart contracts and DevOps aspects and financial model development for service to achieve a financially viable platform. The front-end will serve the UI needs for the dApp, which will be integrated with the back-end service that uses the database as a persistence layer to store the user data and smart contracts to facilitate token operations on the blockchain, such as to create an NFT collection, manage the collection, mint NFTs, set NFT for sale and purchase processes. The discovery phase also includes the Elrond WASM VM and IPFS solutions research. The NFT marketplace platform will serve as a testing ground to get data such as transaction processing time and gas fees in Elrond, this data will be collected and elaborated in the table of results.

This thesis is written in English and is 109 pages long, including 6 chapters, 41 figures and 10 tables.

Annotatsioon
MetaMex NFT turuplatvormi arendus Elrondi plokiahelas

Oleme jõudmas neljandasse tööstusrevolutsiooni ja oleme maailma majandusfoorumilt palju kuulnud globaalsetest muutustest ja oma elu narratiivist. Neljandat tööstusrevolutsiooni kiirendasid pidevad sulgemised, mis sunnivad kõigi ühiskondlike protsesside tohutut digitaliseerimist ja automatiseerimist. Maailma Majandusfoorumi tegevjuht professor Klaus Schwab mainis ka, et "The pandemic represents a rare but narrow window of opportunity to reflect, reimagine, and reset our world" [1]. Maailma keskpangad arutavad digitaalvaluutadele konverteerimist alternatiivina rahamudelile, mida me varem nägime. Digitaalsed valuutad, nagu digitaalsed renminbi, rubla, euro ja dollar, võtavad esikoha peagi, kui maailm liigub rohkem digitaalsete varade poole. Facebook muutus metaversumiks valmistuvaks Meta-ks ja globaalsed tööstused võtavad kiiresti kasutusele detsentraliseeritud tehnoloogiad, mängu- ja kunstitööstused võtavad NFT-ga kauplemiseks kasutusele plokiahela krüptoruumi, samas kui aeglaselt liigub kõik metaversumiga integreerumise poole, kus piire ei eksisteeri. enam ja me elame uutmoodi.

Lisaks eelnevale arenevad tehnoloogiad kiiremini, kui ühiskond arvata oskab, kui rahvusvahelistes lehtedes neljandast tööstusrevolutsioonist välja kuulutati, leidsid paljud meist, et mõnel pool maailmas pole isegi kolmandat tööstusrevolutsiooni läbitud, samad protsessid on Juhtub ka veebitööstusega, Web 3.0 marsib paarkümmend sammu, jättes web 2.0 kaugele maha. Detsentraliseeritud võrgustikud koos detsentraliseeritud autonoomsete organisatsioonidega on juba saanud uueks reaalsuseks ja need pole enam ainult teooriad raamatutes, mida me loeme. Ülemaailmne tööstus on juba alustanud Web 3.0 kontseptsioonide kasutuselevõttu äriprotsessides, selle põhiideeks on püüdlus avatuse ja läbipaistvuse poole. Plokiahel on ühenduslüli läbipaistvuse ja detsentraliseerimise vahel, keskendudes turvalisusele ja terviklikkusele. Kolmanda põlvkonna plokiahelad võtavad kiiresti omaks nutikate lepingute ja mittevahetatavate tokenidele, hajutatud võrgunduse ja hajutatud salvestusruumi kontseptsioonid – kõik see hõlbustab Web 3.0 ideede loomist. See on kaardistamata maa, millel on potentsiaal saada üheks maailma suurimaks tööstuseks.

Web 3.0 tööstuse põhiprobleemiks on liiga suur nõudlus turul, mis tähendab, et praegune kogenud arendajate arv ei vasta kaugeltki tööjõuvajadusele. Plokiahela tehnoloogia virn nõuab palju uurimistööd, enne kui seda saab tõhusalt kasutada. Massimeedia jaoks uued turuplatsid ja digitaalsed rahakotid on juba aegunud ja koguvad tehnilist võlga sisse.

Tehingud Blockchaini ja Ethereumi võrkudes on liiga kallid ja aeglased, takistab turu arengut ning muudab paljud tooted rahaliselt ja majanduslikult elujõuetuks.

Uurimistöö eesmärk on saada ülevaade kolmanda põlvkonna plokiahelatest võrreldes Bitcoiniga (esimene põlvkond) ja Ethereumiga (teine põlvkond), saada tegelikud tehingukulud ja kiirus Elrondi (kolmas põlvkond) plokiahelas ning hinnata platvormi rahalist elujõulisust, infrastruktuuri tegelikud kulud. Uuring koosneb plokiahela teooriast ja Web 3.0 põhimõtteid järgivast hübriidrakendusest, mis on NFT turuplatvorm Elrondil, mille aluseks olev töö hõlmab kogu meeskonna ja projekti planeerimist, platvormi esiotsa, tagaotsa, nutikate lepingute arendamist. ja DevOpsi aspektid ning teenuse finantsmudeli arendamine, et saavutada rahaliselt elujõuline platvorm. Esiots teenindab kasutajaliidese vajadusi dAppi jaoks, mis integreeritakse taustateenusega, mis kasutab andmebaasi püsikihina kasutajaandmete ja nutikate lepingute salvestamiseks, et hõlbustada plokiahelas tokenoperatsioone, näiteks looge NFT kollektsioon, looge NFT-sid, määrake NFT müügi- ja ostuprotsesside jaoks. Avastamisfaas hõlmab ka Elrond WASM VM ja IPFS lahenduste uurimist. NFT turuplatsi platvorm on katsepolügooniks selliste andmete saamiseks nagu tehingute töötlemise aeg ja gaasitasud Elrondis, need andmed kogutakse ja täpsustatakse tulemuste tabelis.

See lõputöö on kirjutatud inglise keeles ja on 109 lehekülge pikk, sisaldab 6 peatükki, 41 joonist ja 10 tabelit.

List of abbreviations and terms

NFT	A unique token stored on the blockchain. NFT is non-fungible, meaning it cannot be interchanged.
ESDT	Elrond Standard Digital Token.
TPS	Transactions per second, a maximum number of transactions can be processed per second on a blockchain platform.
dApp	A decentralized application that runs on a decentralized blockchain system and is facilitated by smart contracts.
VM	A virtual machine that emulates a computer system is often used to isolate computer resources and use this for specific purposes inside the system.
DAO	Decentralized autonomous organization.
IPFS	InterPlanetary File System, a protocol for the peer-to-peer decentralized network storage system.
eDNS	Elrond Distributed Name Service.
ASIC	The application-specific integrated circuit is designed for highly-efficient computations for specific goals, such as Bitcoin mining.
WASM	Web Assembly is a virtual machine to run binary code in the browser, such as smart contracts written in Rust.
SC	Smart contract – a self-executing program that automatically executes and controls the process of the deal or any agreement.
Headless CMS	A content management system that is used for content management only, it is an independent service that has no tight coupling with a website as it provides all data through HTTP API and has no code dependencies with other components.
VPC	Virtual private cloud.
VPS	Virtual private server.

Table of contents

List of figures

List of tables

1 Introduction

Blockchains are evolving quickly, becoming more popular and generating more business value. A wide range of blockchains and cryptocurrencies is available with different applications, such as widely known Bitcoin and Ethereum blockchains and solutions like IBM Blockchain to help transform businesses or STORJ cryptocurrency used in Storj project to pay users for the disk space they provide. Blockchain technologies are becoming popular for governance to serve as a public ledger, this brings blockchains to the masses and makes it an efficient platform for managing the ownership rights such as for digital assets, e.g., non-fungible tokens. The function of NFT is tightly tied with smart contracts, a digital token that exists within the blockchain and can store the information associated with it. The NFT digital assets such as artworks and music appeal to vast masses into blockchains, allowing artists from all around the globe to obtain freedom to sell their artworks and bound an intellectual property to NFT to create manageable digital assets. The OpenSea NFT marketplace illustrates an increasing demand that "has quadrupled its value in less than three months to cross the $10 billion mark" [2] and in March. 11, 2021 the Beeple's "EVERYDAYS: THE FIRST 5000 DAYS" NFT was sold for USD 69,346,250 [3].

1.1 The problem

The first-generation blockchain architecture limitations made people decide which 2 of 3 aspects they aim for in the blockchain: decentralization, security, and scalability. This concept is also well-known as the blockchain trilemma, coined by Vitalik Buterin, the co-founder of Ethereum [4]. The security, decentralization, scalability (Figure 1) aspects cover different aspects of blockchain benefits, the first generation blockchains strive towards security and decentralization, resulting in high transaction fees and small network throughput, e.g., Bitcoin, or security and scalability resulting in low decentralization, or high decentralization and scalability but low security. Bitcoin and Ethereum blockchains both suffer from low throughput and high transaction fees, slowing down the blockchain adoption to the masses.

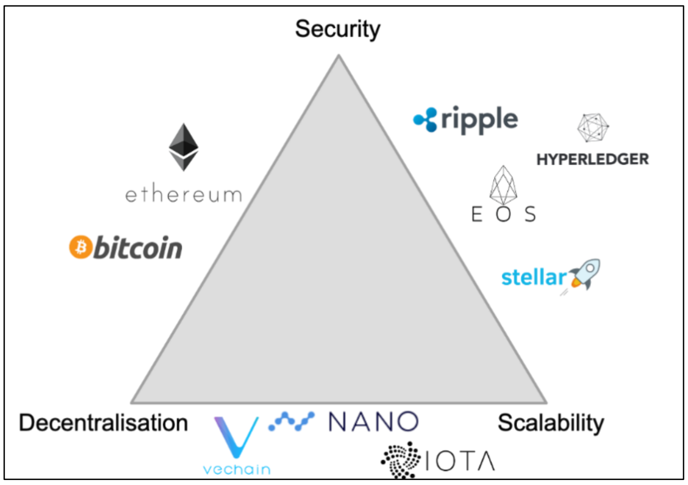
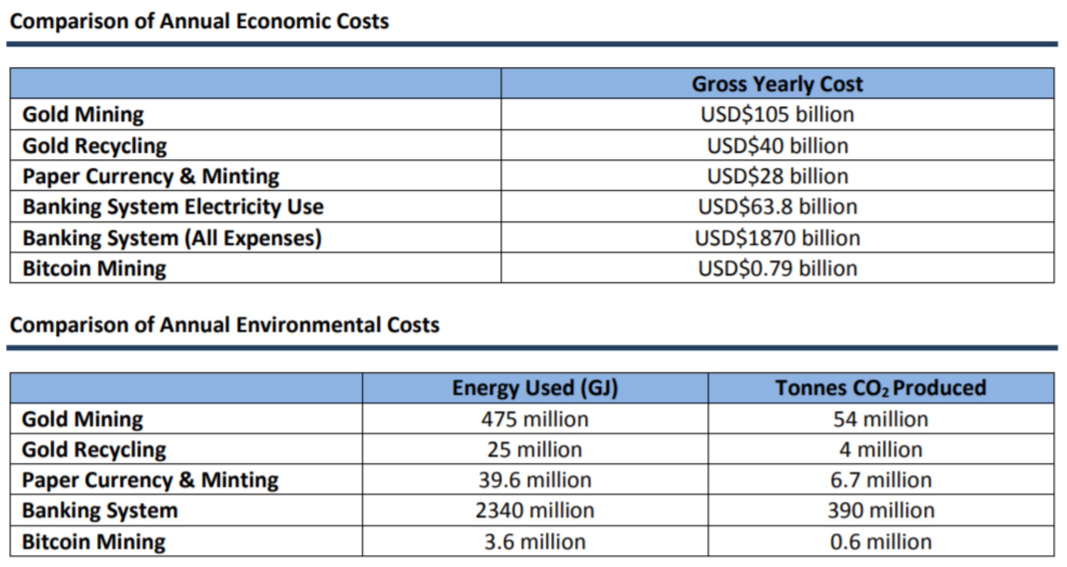


Figure 1. Scalability trilemma [4]

1.2 Blockchain generations

1.2.1 First generation

The first-generation proof of work blockchains was designed to provide people with a decentralized monetary platform with openness and transparency, bringing control back into the hands of people, these were simple ledgers that record transactions, e.g., Bitcoin. The proof-of-work concept problem is that mining requires too much computing power, meaning people are not capable of confirming new blocks via CPU power on their computers as it requires specified ASICs making machines to run it efficiently, however, it is still more efficient compared to conventional banking systems (Table 1).



Comparison of Annual Economic Costs

	Gross Yearly Cost
Gold Mining	USD$105 billion
Gold Recycling	USD$40 billion
Paper Currency & Minting	USD$28 billion
Banking System Electricity Use	USD$63.8 billion
Banking System (All Expenses)	USD$1870 billion
Bitcoin Mining	USD$0.79 billion

Comparison of Annual Environmental Costs

	Energy Used (GJ)	Tonnes CO_2 Produced
Gold Mining	475 million	54 million
Gold Recycling	25 million	4 million
Paper Currency & Minting	39.6 million	6.7 million
Banking System	2340 million	390 million
Bitcoin Mining	3.6 million	0.6 million

Table 1. Comparison of annual costs [5]

1.2.2 Second generation

The second-generation blockchains, e.g., Ethereum, add a smart contract layer that serves as "conditions" to transactions and extends first-generation blockchains' limited functionality, e.g., Bitcoin. Ethereum massively extended the use-cases of blockchains and made it a platform for different fields of application, e.g., new-generation projects for land registration and governance. However, Ethereum did not completely solve the actual problem of scalability coming from the first blockchain generation architecture.

1.2.3 Third generation

The third-generation blockchains improve performance, energy-efficiency and solve the scalability problem from previous generations. These blockchains are much more scalable, secure and decentralized, these also have smart contracts support and cross-chain transaction ability.

1.3 Proof of work vs proof of stake

1.3.1 Consensus mechanism

On decentralized networks, the transaction validation problem cannot be solved through a centralized authority, instead, there is a “consensus mechanism” developed that allows all the nodes in a network to agree about which transactions are legitimate, the two primary mechanisms nowadays are “proof of stake” and “proof of work”.

1.3.2 Proof of work

The proof of stake concept requires all nodes on the network to compete to solve the problem, and these are the new block hashes calculation problems, e.g., who finds the valid hash has the right to update the blockchain with a new block of verified transactions and gets rewarded by the network. That is a robust and straightforward method to incentivize more miners to the network to increase its power and security, while on the other hand, it is hard to scale as the mining complexity always grows. In competitive situations, when there is an alternative network branch with the same block, the networks pick the longest chain.

1.3.3 Proof of stake

In the proof of stake networks, the validator node first requires a certain amount of cryptocurrency to be “staked” for the node. After achieving a certain amount invested, these nodes are allowed to enter into the node pool with a chance to be chosen to process a transaction. The nodes with the most significant investment and the biggest uptime have a bigger chance of processing the transaction. The currency can always be “unstaked” from the node, making it unable to stay active when the amount is less than the minimum required. The benefit of the proof of stake mechanism is that it has much less computing power than proof of work, as only owning the tokens is required to enter the queue, making it much more accessible and affordable to participate in the network.

The proof of work network attack is possible with the hardware representing the majority of the network, however, it is unrealistic due to the immense scale of the network resources united. The proof of stake network can be attacked when the buyer owns over half of all the tokens available and sets up enough validators to make up over half the network.

1.3.4 Elrond’s secure proof of stake

Elrond has developed its unique consensus called “secure proof of stake”, which combines proof of stake consensus with random validator selection, random sampling of consensus group and random reshuffling of nodes into other shards, and there are different types of nodes available, validators, observers that only act as read & relay, and fisherman nodes to catch the malicious actors [6]. That means the nodes are picked randomly from the pool and randomly distributed into different shards (4 in Elrond), meaning the probability of obtaining the majority of nodes in a single shard and being picked for the same round simultaneously is very unrealistic and malicious actors can be spotted by fisherman nodes also [7]. In each round, a new consensus group is picked, and only one node is allowed to be a block proposer, the node is known only one for the current round, and the rest of the validators in the consensus group validate and sign it [7].

1.4 Elrond

Elrond is a third-generation proof-of-stake blockchain that solves the blockchain trilemma and has outstanding performance metrics compared to first and second-generation blockchains. Elrond aims to become an internet-scale blockchain, offering developers the integrated WASM VM engine, tokens (FT, SFT, NFT), support of different programming languages for smart contracts, and splits the network into test and development branches for development purposes [8]. Elrond offers a very competitive platform for building decentralized applications with high TPS, low transaction fees and short-latency time with the support of WASM-compatible smart contracts.

1.4.1 Blockchain trilemma and Elrond

1.4.1.1 Scalability

Elrond has a relatively high performance of 15,000 current TPS with a scale up to 100,000 TPS with \$0.001 transaction cost and 6 seconds latency and network setup of 3200 nodes with 4 shards. Elrond has an extensive financial model that supports rewards, the developers receive ~30% of generated smart contract transactions gas fees, and validator nodes are also rewarded with the scheme "5,000 EGLD (up from 4,000 EGLD currently) will be distributed to the Delegator Waiting List + Validator Queue each Monday, proportional to the amount staked and time spent in the Queue in the previous week. Validators must stake 2,500 EGLD to join the Queue." and rewards for processed transactions [9]. Elrond works on average computers (i3, 8GB RAM, SSD storage), making it easy to scale and be rewarded for doing this [10].

1.4.1.2 Decentralization

Elrond is based on a certain number of nodes taken from the node pool, the nodes are chosen randomly from the pool to process the transaction, nodes are independent and can be run by every person with a computer, but only the highest runtime nodes will be taken from the queue into the pool of nodes, and being taken out if going down. The algorithm automatically scales network shards when it's required to use more nodes from the node pool.

1.4.1.3 Security

The network is divided into shards, and nodes are being selected randomly from the pool, it is barely possible to obtain a majority in one shard by pure luck as "One possible way to attack a shard would be to have more than ⅓ malicious nodes into the shard, but this is highly unlikely due to the random assignment of nodes to shards, and the assumptions that no more than ¼ of nodes in the network are malicious" [11].

1.5 Elrond NFT Marketplace

Unlike Bitcoin, Ethereum and first and second-generation blockchains, Elrond users benefit from security, performance, scalability, smart contracts, and custom tokens support. Marketplace based on Elrond blockchains will offer lower transaction fees, faster transaction execution and a better user experience. An integrated rewards mechanism makes the project more financially viable with a growing number of transactions generated. An Elrond blockchain-based marketplace should offer better a price/cost balance than Bitcoin or Ethereum based solutions and more flexibility in financial viability and functionality with an embedded WASM machine for the smart contract's execution.

1.5.1 The First Carbon Negative European Blockchain

Elrond has also solved the CO2 problem by becoming the first carbon-negative European blockchain, meaning the blockchain is highly energy-efficient as "Elrond is a high-speed Proof of Stake network that is more than 6 million times more energy-efficient than Bitcoin", and Elrond team is actively investing into sustainability projects, biodiversity protection and social programs [12].

1.6 Elrond herotags

Elrond has native support of herotags, these are custom usernames which can be set once per wallet by the owner used as a wallet address for the transfers. A herotag gives people an option to have a human-readable name for the wallet and use it on the network.

1.6.1 eDNS and herotags

eDNS is an Elrond Distributed Name Service, a decentralized naming system such as DNS solutions running on Ethereum. This allows addresses to map to specific usernames on top of Elrond domain called herotags, enabling more accessible interactions using short herotags instead of address hashes. Digital wallets like Maiar can use eDNS to transfer money using the herotag or username.

2 Background

2.1 Borderless NFT art

Since NFT tokens are manageable on blockchains, these can be transferred inside the blockchain, meaning there are no geographical borders for the art in NFT space unless the network is functioning correctly. Blockchain provides every artist with equal possibilities and opportunities to trade, exchange and transfer tokens. NFTs allow artists to preserve their ownership rights and use NFTs as proof of ownership with a trackable history record.

2.2 IPFS storage

IPFS storage is an InterPlanetary file system protocol that allows storing files in a peer-to-peer network, which is more reliable than traditional centralized storage. IPFS storage solutions enable artists to store digital assets in a reliable decentralized space. NFT stores the digital assets URI that was set on the NFT creation stage, meaning once the URI leads to dead storage, the token loses a connection to the allocated digital assets, preserving only the URI

2.3 NFT marketplaces

NFT marketplaces are growing blazingly fast, and CNBC states, “Non-fungible tokens (NFTs) may seem like a passing craze, but with over $10 billion traded in the third quarter of 2021 alone” [13]. NFT becomes a good digital assets investment for the companies and serves as proof of purchase with ownership rights, each purchase on the blockchain cannot be repeated, preserving its distinctiveness of it, and the token is non-fungible.

2.3.1 Bitcoin NFT marketplaces

Since Bitcoin is a first-generation blockchain, there is no support for NFT and smart contracts, thus, there are no native NFT marketplaces on first-generation blockchains, and if there were any, the transaction fees would make it hardly financially feasible.

2.3.2 Ethereum NFT marketplaces

Ethereum, unlike, for example, Blockchain, is a second-generation blockchain with support of smart contracts and custom tokens, meaning it is a good fit for the marketplace, and the most prominent marketplaces on the market, such as OpenSea, are operating on Ethereum, Polygon and Klaytn second-generation blockchains [14].

2.3.3 Solutions on Elrond

The number of marketplaces available on Elrond is noticeably smaller than on Ethereum, but there are NFT marketplaces running exclusively on Elrond, e.g., TrustMarket, Isengard or Deadrare, Trust.market. One popular marketplace called eMoon's wallet has been compromised, its trending NFTs were stolen, causing the project to shut down immediately. "On Friday 02/04, someone successfully broke into our system and stole about 500 of the rarest NFTs from our wallet in 30min" [15]. The solution made for this research must avoid any NFT ownership through the marketplace's wallet, meaning only the smart contract that owns NFTs for sale is a secure way to facilitate sales. The transaction cost might vary on the contents of the smart contract endpoints, the price depends on the number of computing resources it needs for execution.

2.4 Elrond World community

Elrond World, which is a global blockchain community that is specifically orientated around the Elrond Network blockchain ecosystem and supports NFT projects, did not have any NFT marketplace platform, the goal of the project is also to provide a platform for the Elrond World community with joint efforts of Elrond World team.

2.5 Tokens on Elrond

The native token on Elrond network is the EGLD. ESDT is Elrond Standard Digital Token and facilitates any number of tokens to be created by developers to use the Network with as a fungible token. Mex is the native ESDT token for the Maiar Exchange, and Maiar is the joint Company of Elrond Network, which has developed the Maiar Wallet and Maiar Exchange service that focuses exclusively on Elrond.

3 Methodology and NFT marketplace development

3.1 Work process

3.1.1 Team

The team working on MVP has two developers and a project manager. The Gokai Labs team put much effort into the web-app design and branding (logos), the author took the full-stack developer role, taking the front-end, back-end, CMS, infrastructure and smart contracts development for the application, the Elrond World community leadership worked through the financial model and the project roadmap.

3.1.2 Git

Git is the most commonly used version control system today. Git offers much flexibility in how teams track changes and commit to the main source code. The Git was used alongside the GitHub repositories. First, the requirements were gathered, and the project goals were specified and well-defined, the separate Git repositories were created for different parts of the application, such as back-end, front-end, Strapi CMS and smart contracts. The "monorepo" concept for an entire platform was not chosen to enforce the code security and the principle of components responsibility segregation to expand the teams in the future more manageable and to be able to manage the access to infrastructure, front-end and back-end for different teams separately. The work has been held in different branches according to tasks. The code has been reviewed and merged through pull requests into the main repository. Pull requests were used to review the changes, discuss and refactor the code before it gets merged into main branch and taken by pipeline into deployment. The Git flow is illustrated in Figure 2 below.

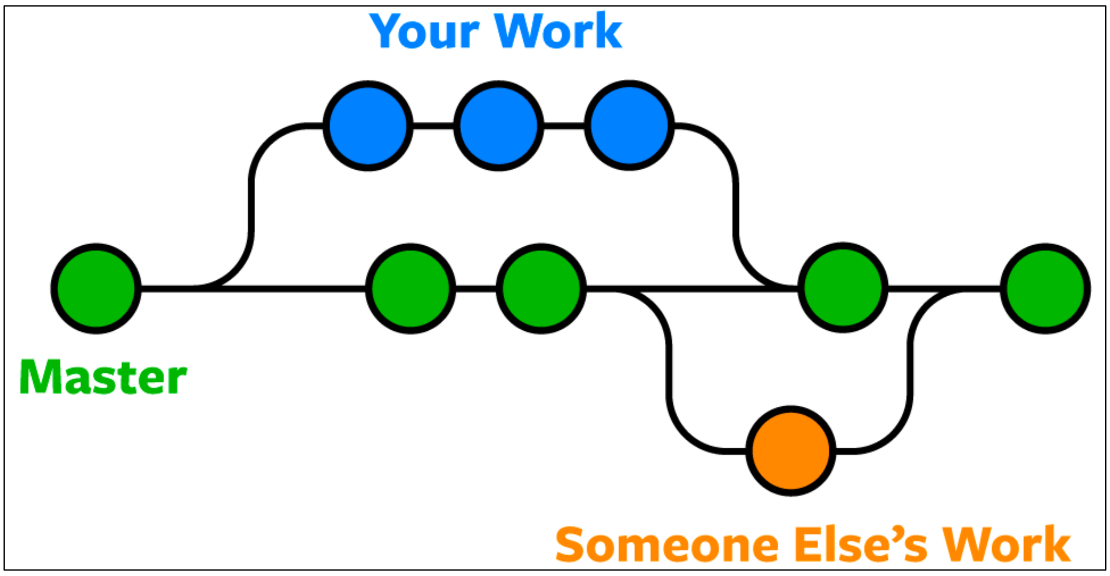


Figure 2. Git flow visualized [16].

3.1.3 GitHub

GitHub was chosen to host the Git repository, allowing team members to use GitHub authentication to access licenses and tools integrated with GitHub, e.g., GitKraken.

3.1.4 Team communication

The Discord and Telegram channels were chosen to provide dedicated and secure channels and workgroups for communication with the core product team and external workforce, e.g., hired workforce from Upwork and Elrond World Discord server space.

3.2 Front-end development

The front-end development includes a set of frameworks, such as Tailwind CSS framework and Next.js React framework.

3.2.1 Tailwind CSS

Tailwind CSS provides a developer-friendly approach for inline component style customization right from the component code and also allows global styling. Tailwind also optimizes the CSS output and also generates a static file for all style contents, as mentioned, "Tailwind CSS works by scanning all of your HTML files, JavaScript components, and any other templates for class names, generating the corresponding styles and then writing them to a static CSS file." [17]

3.2.2 Next.js

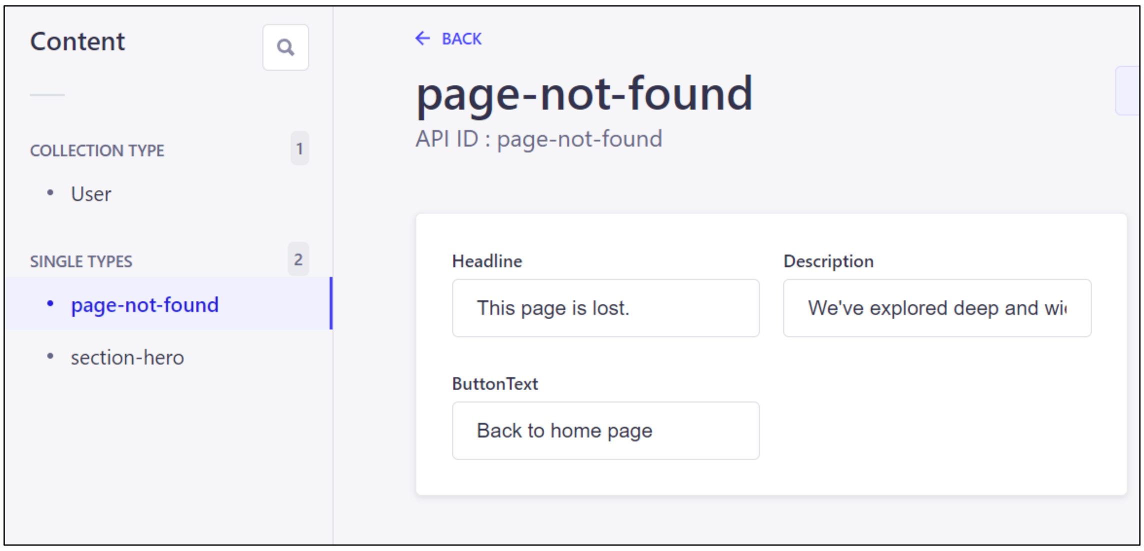
Next.js is a framework based on React built on top of Node.js. Next.js provides built-in support for server-side generation and server-side rendering, which is a superior approach for SEO and search engines. Also, it has built-in image optimization, static site build capability, e.g., static assets are fetched on the build time and stored as static assets, extended router support, API methods generation on-fly, and the serverless functions support, which can be used to proxy the cookies and traffic. Next.js also supports TypeScript, which has a native type check and interfaces support built-in [18].

3.3 CMS

Next.js has great support for static site assets allowing these to be fetched on build time and served infinitely from local static assets. The headless CMS was used to extend the flexibility of components with text/image static content fetched from the headless CMS.

3.3.1 Strapi

Strapi is a headless CMS which supports custom content types, and API endpoints are built on the fly based on content types provided with the specified structure. Strapi works as a Node.js application and requires a database to run, such as PostgreSQL. It is serving the text for the homepage, forms and images for the missing and broken images for the NFT overview component below in Figure 3 is an example of a 404-page data type, and the actual JSON in Figure 4 is a response from the CMS with content set up in the CMS.



page-not-found

API ID : page-not-found

Headline

This page is lost.

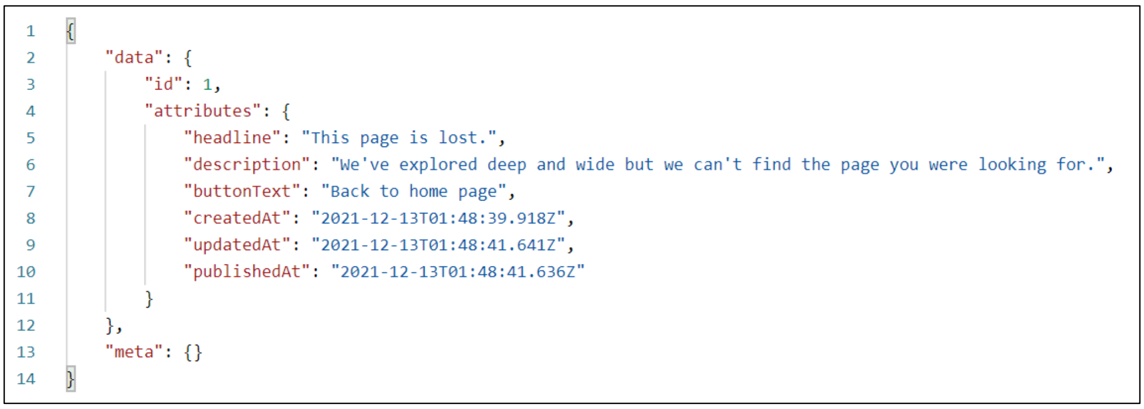
Description

We've explored deep and wi

ButtonText

Back to home page

Figure 3 Strapi 404-page content type



```
{
    "data": {
        "id": 1,
        "attributes": {
            "headline": "This page is lost.",
            "description": "We've explored deep and wide but we can't find the page you were looking for.",
            "buttonText": "Back to home page",
            "createdAt": "2021-12-13T01:48:39.918Z",
            "updatedAt": "2021-12-13T01:48:41.641Z",
            "publishedAt": "2021-12-13T01:48:41.636Z"
        }
    },
    "meta": {}
}
```

Figure 4 Strapi 404-page JSON response

3.4 Back-end development

The back-end is a supply of custom data for user profiles which cannot be retrieved from the public blockchain API endpoints.

3.4.1 Golang/C#/Python

.NET core was considered to be used for the back-end, the benefit of C# is built-in support for Entity Framework and rich ecosystem, but the long build time language complexity was considered excessive for the simple microservices to launch the project. Python is also supported by the Elrond team, and the native libraries in Python are provided, but the library version in Go is usually supported better than Python. Due to the simplicity, most prosperous community support, project structure simplicity and single-file deployment, the decision was made to proceed with Golang of the latest version 1.18.

3.5 Infrastructure

The infrastructure for the dApp requires back-end and front-end and CMS services to be running simultaneously with the database for the back-end and database for the CMS, the smart contracts are able to run within the customer’s web browser within WASM virtual machine, so there was no server used to execute the smart contracts, sign and confirm the transactions even.

3.5.1 Cloud

The custom VPC or set of VPS-s was an excessive solution requiring manual maintenance, so the cloud hosting was chosen to run the platform. Cloud-managed database services are a better alternative to a self-hosted database in terms of database scalability. The scalability for the database can be achieved within a custom cluster with auto-scale rules set, while still, the cloud database managed solution provides private subnetworks segregation out of the box to allow only the explicitly allowed application's connections, and that is a huge advantage of cloud offered services. The network traffic is free within the VPC in the cloud, which is another huge advantage of cloud setup compared to a set of VPSs connected through a public network.

3.5.2 DigitalOcean vs AWS

DigitalOcean is a better fit for the small business and starter projects, this is a cloud with the typical features support, such as managed database, DNS management, file storage and pipeline templates built-in. AWS is an enterprise-grade solution which will cost more than DigitalOcean service fulfilling the same needs.

DigitalOcean supports a new DigitalOcean App platform which can handle Docker containers, it offers horizontal scaling and can deployed instances can be white-listed for the database, the drawback of the DigitalOcean App platform is that compared to the DigitalOcean droplet, the Apps platform cannot provide free traffic between services, while this doable with Droplets within the same VPC, meaning the network traffic between apps is not free and causes traffic expenses. Another drawback is that DigitalOcean does not support docker-compose services, while AWS supports it, so running Apps within the same VPC in Apps is not feasible, the services are deployed separately with no dependencies in between, so it did not affect the initial platform architecture. The DigitalOcean was a cheaper solution with the support of all required services and offered free trial credits worth 100$ to set up the platform and test it.

3.5.3 Build pipelines

The DigitalOcean has access to the GitHub repositories and is watching for events on the main branch, so when the new commit is pushed to the main branch of the front-end repository, the DigitalOcean is triggering a pipeline to rebuild the front-end application, if there is some error in the build stage that won’t affect the live version as will deploy only the version that builds without errors. DigitalOcean Apps platform also offers zero-downtime deployments. The same pipeline logic has been applied for the back-end service and Strapi CMS. Strapi CMS stores manually added content types as a local structure of files and builds API based on these, preserving all the data in the database. The Strapi CMS is rebuilding and deploying when new changes to the data model structures are committed, which are made on the local machine, committed to the repository and appear on live deployment.

3.6 Domains and DNS

DigitalOcean provides a domain management system with 99.99% uptime SLA DNS servers with cloud firewall and load balancers features. The #domain_name has been reserved for the marketplace and configured on DigitalOcean for usage.

3.7 Storage

DigitalOcean offers a cloud storage solution called Spaces, S3-compatible object storage with a built-in CDN and a public API so the files can be uploaded from the application through the back-end service, keeping keys secret. This is not viable decentralized storage, and this cannot be used as IPFS storage to store digital assets for NFT, so this is used to store the user data such as profile pictures or banners to arrange customer profile pages.

3.8 IPFS Storage

The marketplace does not upload digital assets on behalf of the customer to the IPFS storage, the customer does this manually to avoid responsibility for the digital assets uploaded to IPFS storage, the integration can be done in the further development once there is a dedicated team to verify and administer IPFS upload requests. NFT can be taken out from marketplace listings, but the content uploaded to IPFS will persist in the public space.

3.9 Financial model

The marketplace financial strategy is based on a simple model of transaction rewards and sale commission. Every 3% of sale value is taken as commission to the marketplace wallet, and around 30% of generated transaction gas fees will be obtained as a reward for generating the transactions on the network. That means the marketplace not only gets sale commission, but it also does it when customers create a collection, set permissions for collection, create NFT, list NFT for sale, update listed NFT price, cancel NFT listing, buy NFT, transactions are committed on every step and marketplace gets rewarded for each.

3.9.1 Business model

The customers are the key partners for the marketplace platform, customers generate content on the platform and bring digital assets for listing, expanding the marketplace assortment of goods. The platform provides artists with a simple NFT management system, it will serve as a control panel for the digital assets, platform also provides customers with a personal profile that can be adjusted and published for marketing purposes. The platform scope also includes a launchpad feature that can be used to enable customers to announce collections available to mint. In general, the marketplace is a platform that facilitates customers' campaigns, sales and digital assets management, and the launchpad page can be used for basic marketing campaign purposes.

3.9.2 Operational costs

The operational costs include only cloud infrastructure. Strapi CMS, Golang and Next.js are open-source solutions available for the public domain and are free to use. The CMS and back-end require a database as a persistence layer, the databases are running on DigitalOcean Managed Database service PostgreSQL node for a fixed price 15$ per month with free daily backups. For a start, the CMS, back-end and front-end can be run on the lowest-tier App instances 5$ each resulting in 15$ per month for all computing resources. The Space storage service to store the profile images and banners comes with a basic plan including 250 GB of storage space, 1 TB bandwidth and whole month uptime resulting in 5$ per month, image size limit and compression were applied to the profile image and banner not to overuse the storage space. In total, the entire infrastructure with front-end, back-end, CMS, storage and a managed database cluster ends up in 35$ a month bill before tax and 42$ bill a month with VAT Estonia (20.00%) included.

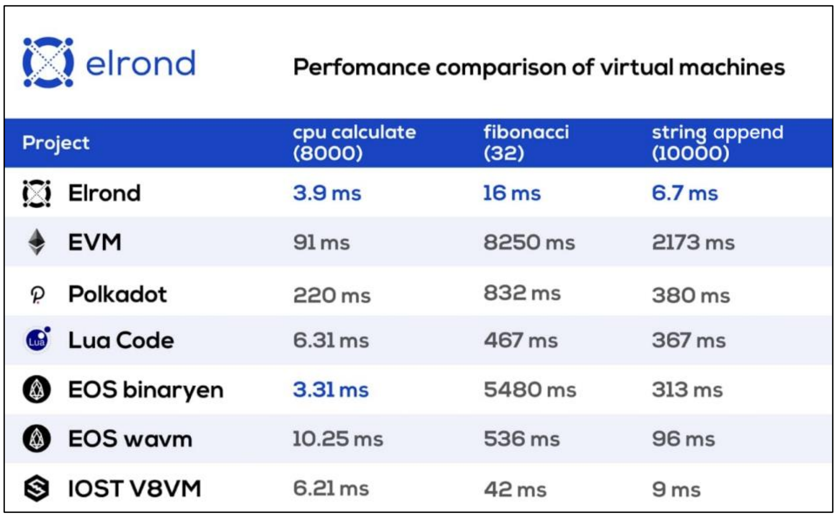
The low computing costs make the marketplace very affordable and profitable, taking the peak EGLD price in 2021 into consideration, the 542.48$ per 1 EGLD, just three sales a month each worth 1 EGLD will result in 48.82$ marketplace revenue, which is more than 42$ computing costs and leaves 6.82$ as profit already. By 18.03.2022, the OpenSea total trade volume reached 2.11 billion USD, meaning there is a potential to reach a much higher income and the base cost of 42$ per month is an excellent result.

3.10 Smart contracts

For a reliable and credible sales platform, smart contracts are a must, these bring transparency to the platform, and anyone on the network can validate this. The smart contracts manage the sales and operations with the customer's NFT.

3.10.1 WASM

Elrond supports the custom implementation of a virtual machine with the support of WebAssembly, which enables customers to run smart contracts on their side, and sign and send a transaction to the network. Elrond's WASM supports any language that can be compiled into a byte executive binary file. As shown in Figure 5 below, Elrond's VM is very efficient when it comes to calculations, VM was built with composability kept in mind, meaning the smart contracts support asynchronous calls to one another, even remaining composable across different shards.



elrond

Perfomance comparison of virtual machines

Project	cpu calculate (8000)	fibonacci (32)	string append (10000)
Elrond	3.9 ms	16 ms	6.7 ms
EVM	91 ms	8250 ms	2173 ms
Polkadot	220 ms	832 ms	380 ms
Lua Code	6.31 ms	467 ms	367 ms
EOS binaryen	3.31 ms	5480 ms	313 ms
EOS wavm	10.25 ms	536 ms	96 ms
IOST V8VM	6.21 ms	42 ms	9 ms

Figure 5 Blockchain VM performance comparison [19]

3.10.2 Rust

The smart contracts are primarily written in Rust, which is a memory-safe and type-safe compiled language with native sandbox support. Rust does not include garbage collector or advanced optimizations, enables the writing of low-level code without the overhead and minimal runtime, making smart contracts more efficient and less gas consuming.

4 Implementation

4.1 NFT ownership model of Elrond

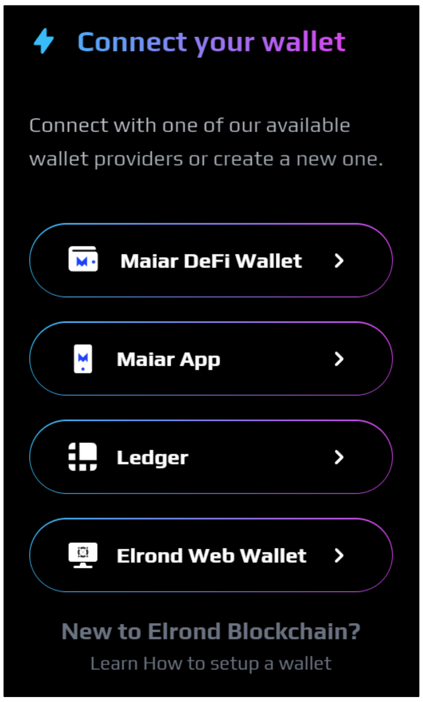
Unlike in Ethereum, where the marketplace owns the smart contract and NFTs are derived from it, and the smart contract always owns it and sells the ownership rights to it, in Elrond, the customer has 100% ownership of the NFT and NFTs can be created in the customer's wallet and sold with complete transfer of all the ownership to the new owner. The 100% token ownership in the model makes ownership management significantly easier, the token will always have only one current owner.

4.2 MVP scope

The marketplace scope is limited for the first stage of the product and follows the research goals. The planned marketplace scope was an essential feature set with the aim of security, which contains: support of multiple authentication methods based on blockchain, hybrid app architecture, cloud infrastructure architecture, cloud storage, ease of deployment, core functionality integration with Elrond API, smart contracts to facilitate the entire sale flow of the marketplace and low operational costs for the first stage of the marketplace platform roadmap. EGLD is used for all payments on stage 1 of the marketplace MVP, full MEX token support for trading was also considered as the next phase of platform implementation, so the MVP has wallet integration with MEX for further integrations.

4.3 Multiple authentication methods

The marketplace supports Maiar DeFi Wallet, Maiar App, Ledger and Elrond Web Wallet authentication methods as listed in Figure 6. All these options connect the marketplace web platform with customer wallets on the blockchain, allowing the initialization of wallet components on the customer side and also enabling operations with the wallet through Elrond SDK. Maiar DeFi Wallet isn't currently supported within Elrond Web Wallet integration, the custom component makes this option available.



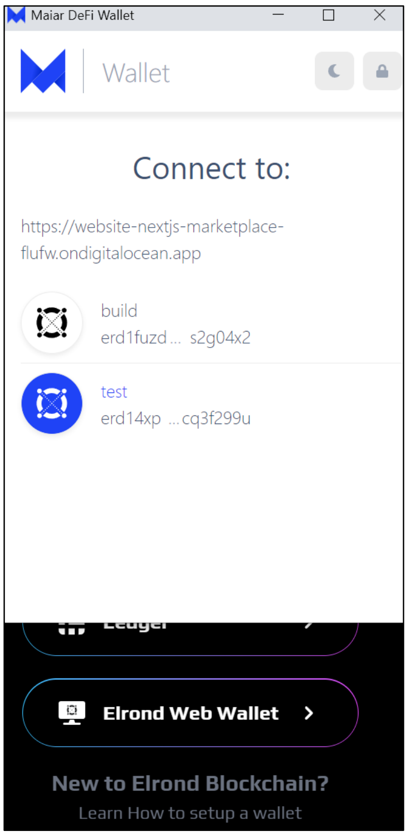
Connect your wallet

Connect with one of our available wallet providers or create a new one.

New to Elrond Blockchain?
Learn How to setup a wallet

Figure 6 Authentication methods

Maiar DeFi Wallet from Figure 7 is a web browser extension developed by Maiar, it connects the web app with the wallet on the blockchain directly from the browser.



Connect to:

https://website-nextjs-marketplace-flufw.ondigitalocean.app

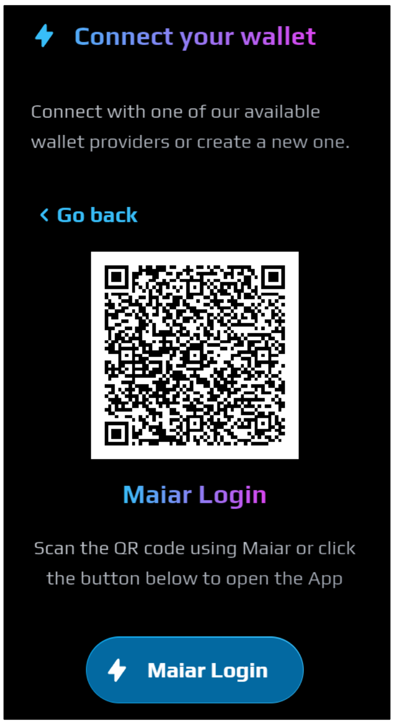
build
erd1fuzd ... s2g04x2

test
erd14xp ...cq3f299u

New to Elrond Blockchain?
Learn How to setup a wallet

Figure 7 Maiar DeFi Wallet browser extension

An alternative solution to web extension is an integration with the Maiar App from Figure 8 below that can scan QR codes and authorize access to the wallet.


Connect your wallet

Connect with one of our available wallet providers or create a new one.

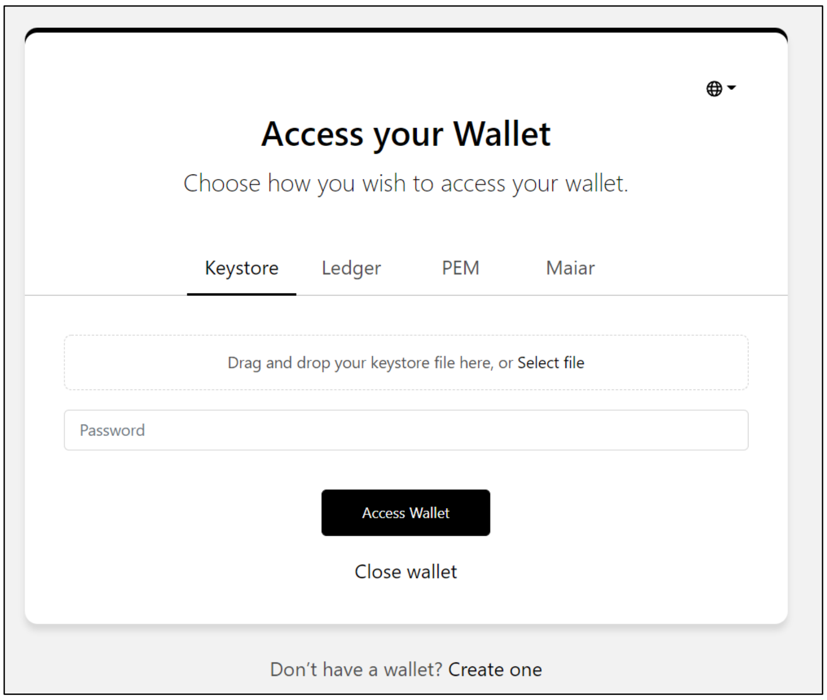
< Go back

Maiar Login

Scan the QR code using Maiar or click the button below to open the App

Figure 8 QR code authentication

Elrond also provides developers with Elrond Web Wallet integration that can process Keystore, Ledger, PEM file and integration with the Maiar app as listed in Figure 9 below.


Access your Wallet

Choose how you wish to access your wallet.

Drag and drop your keystore file here, or Select file

Don't have a wallet? Create one

Figure 9 Elrond Web Wallet

4.3.1 Wallet balance

When authorization has been passed successfully, the web platform prepares a wallet component for the customer wallet, which is persisted in the customer's web-browser local storage, which persists wallet data such as balance and address. In order to display a readable balance in USD, the EGLD balance is fetched from the wallet through the SDK gateway and MEX as the custom token is fetched from the Elrond API endpoint. Once data has been fetched from the integration with CoinGecko API, the latest EGLD and MEX exchange rates are fetched, calculated with the latest retrieved exchange rates, converted into USD value and displayed at the customer's wallet component in Figure 10.

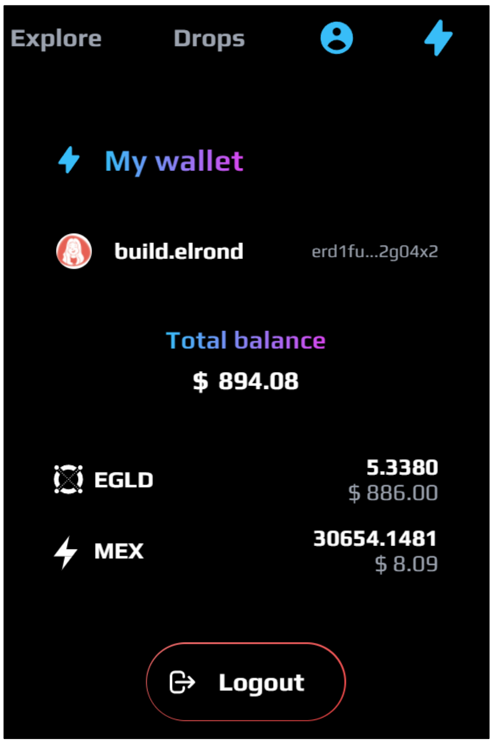


Figure 10 Wallet component balance

4.4 Hybrid app architecture

The platform has a hybrid architecture, meaning it is integrated with public Elrond API, Elrond blockchain network, cloud storage and custom services as back-end and Strapi CMS, which are connected to databases as persistence layer as in Figure 11 below.

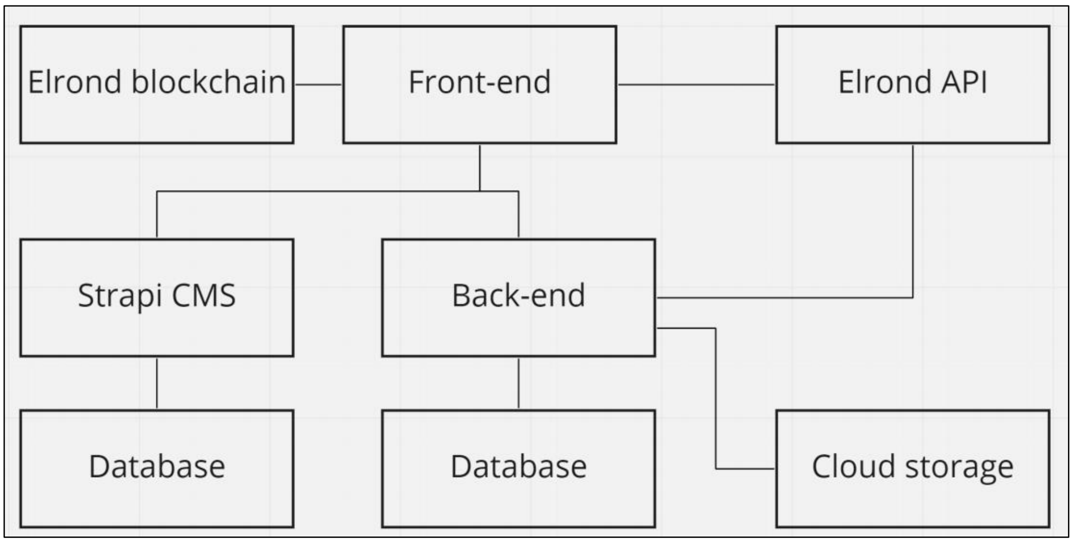


Figure 11 Hybrid app architecture

4.4.1 Cloud infrastructure architecture

The diagram in Figure 12 below shows the DigitalOcean components and databases used for the back-end and Strapi services and three services deployed in the App platform listed in Figure 13, these are back-end, CMS and web services.

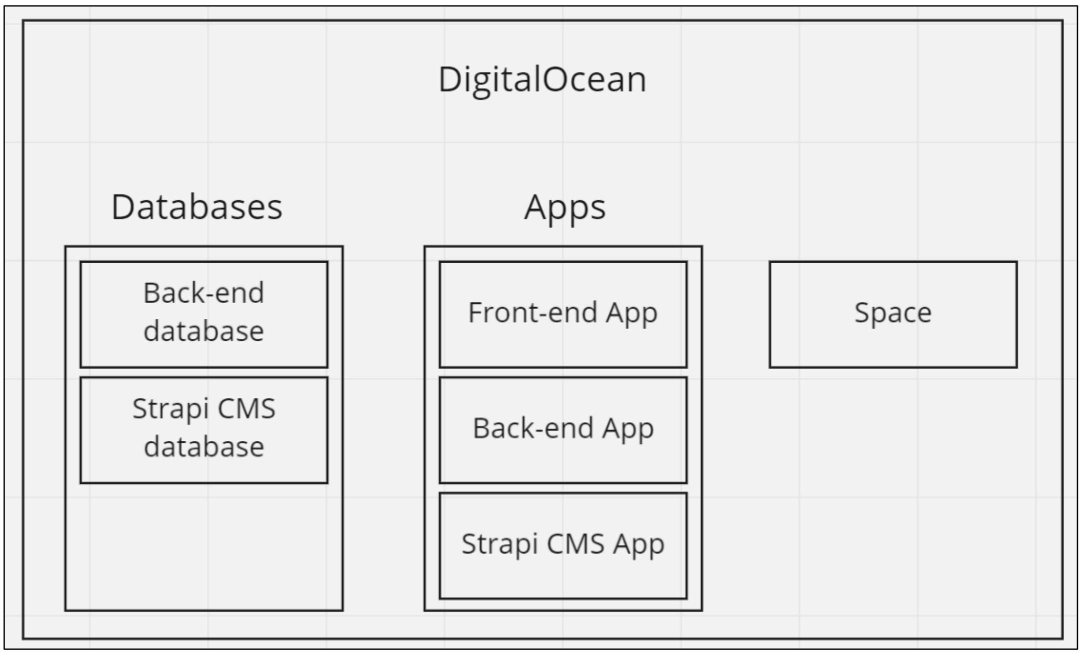
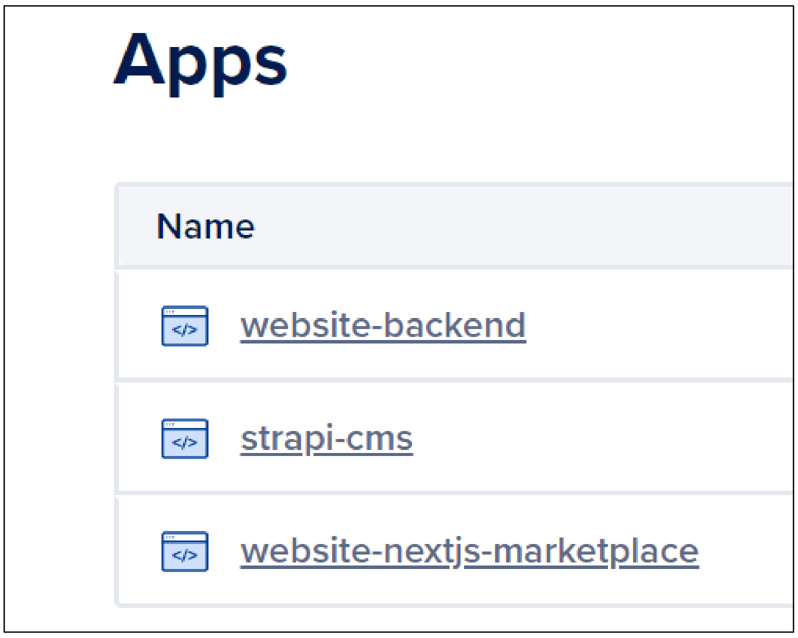


Figure 12 Cloud infrastructure architecture



Apps

Name
website-backend
strapi-cms
website-nextjs-marketplace

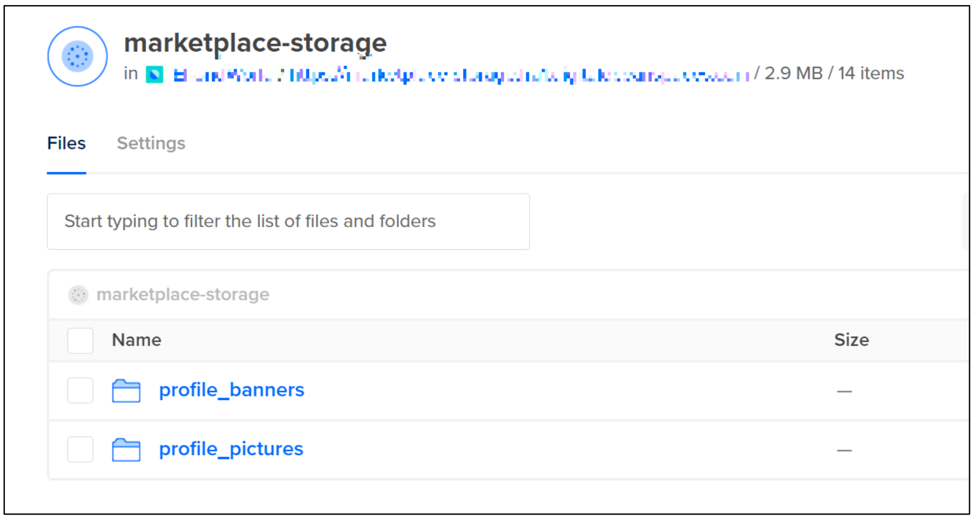
Figure 13 Deployed DigitalOcean Apps

4.4.2 Docker for deployment

The Golang back-end and Next.js front-end are packed into Dockerfile images for deployment to the App platform and docker-compose file to run the database with database viewer in a container for easier development, the example container for Strapi CMS PostgreSQL database container with PgAdmin is shown on Appendix 2.

4.4.3 Cloud storage

DigitalOcean Space provides simple programmatic access to files, for the MVP scope, the profile pictures and banners are stored in the storage as shown in Figure 14 below.



marketplace-storage

in [redacted] / 2.9 MB / 14 items

Name	Size
profile_banners	—
profile_pictures	—

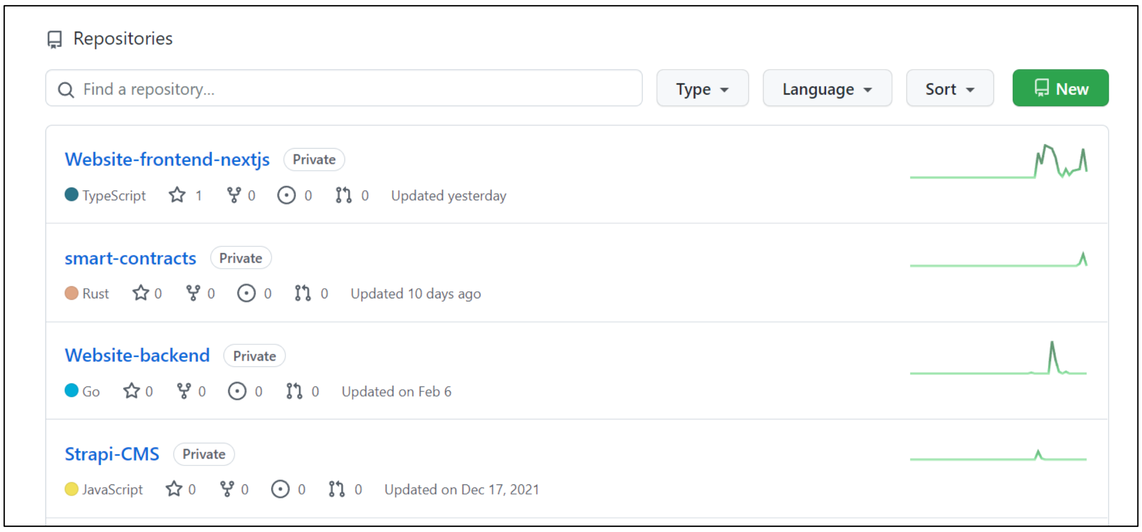
Figure 14 DigitalOcean Space

4.4.4 Elrond API integration

The web application is integrated with blockchain through Elrond HTTP API, Elrond API is public and does not require any access tokens. The wallet information, such as wallet balance, tokens, digital assets, and herotag, is fetched directly from the Elrond API. The list of NFTs for the profile page, EGLD and MEX token balance for the wallet to show the USD value in the wallet component are fetched from API, and Elrond's SDK gateway is used for the authentication methods on Elrond, the NFTs listed for sale are also fetched from smart contract's NFT balance through the Elrond API.

4.5 GitHub project structure

The project Git space was split into 4 private repositories for the Next.js web application, smart contracts for Elrond written in Rust, back-end written in Golang, and the Strapi CMS repository to persist the data models and settings. This micro-repository space from Figure 15 below enables maintainers to have granular control over repositories and send private member invitations individually to different repositories, e.g., not revealing the back-end and front-end code and CMS configuration to the contractor asked to provide services only for the smart contract code.



Website-frontend-nextjs Private
TypeScript 1 0 0 0 Updated yesterday

smart-contracts Private
Rust 0 0 0 0 Updated 10 days ago

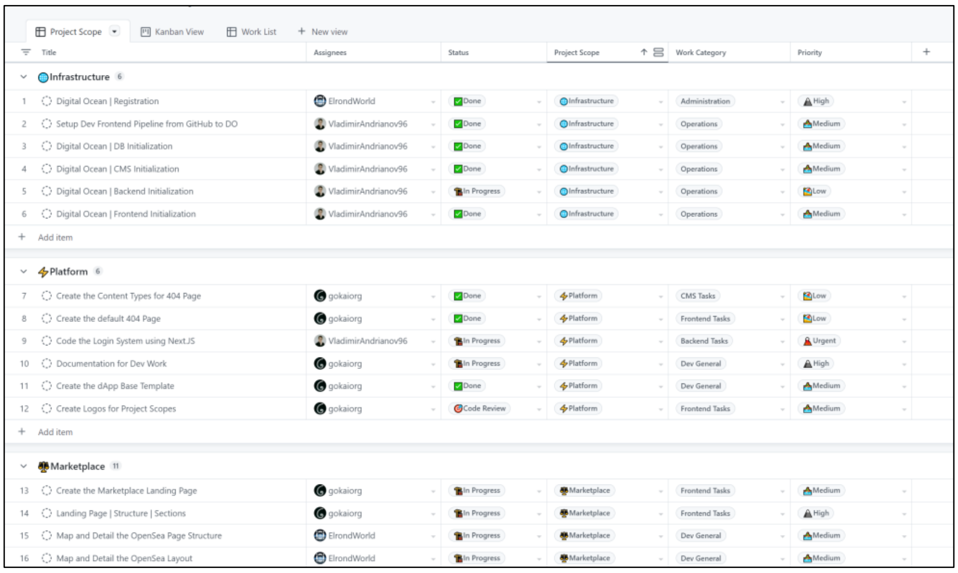
Website-backend Private
Go 0 0 0 0 Updated on Feb 6

Strapi-CMS Private
JavaScript 0 0 0 0 Updated on Dec 17, 2021

Figure 15 Git repository structure

4.5.1 GitHub project management

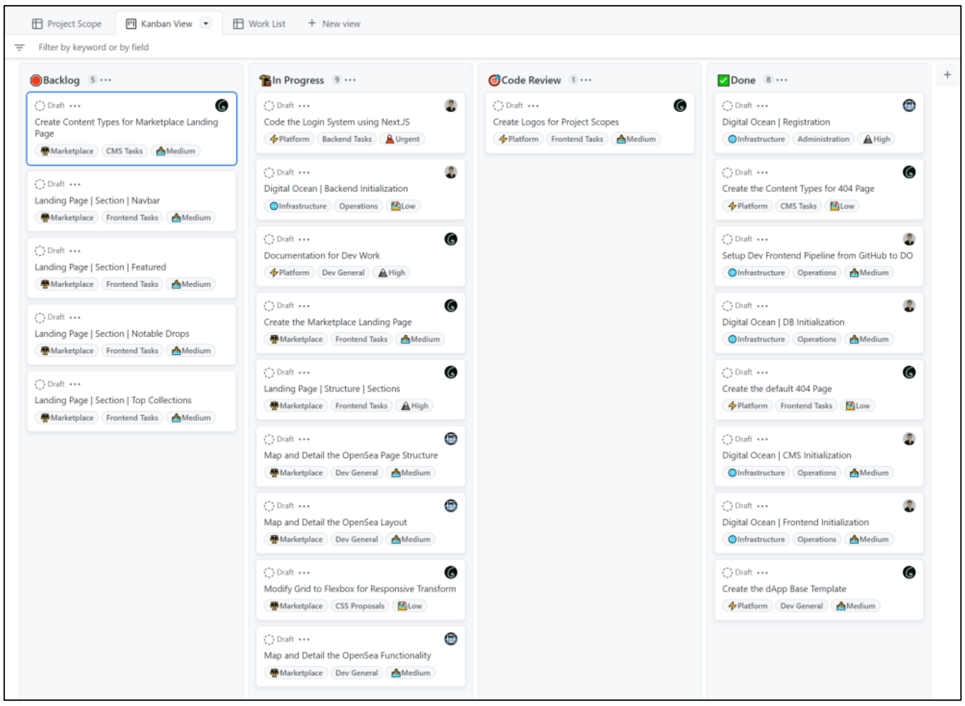
GitHub offers a project space with a Kanban board to organize projects and tasks in one place. The MVP scope tasks were defined for the first stage of development as in the project scope section in Figure 16.



	Title	Assignees	Status	Project Scope	Work Category	Priority
	Infrastructure 6					
1	Digital Ocean | Registration	ElrondWorld	Done	Infrastructure	Administration	High
2	Setup Dev Frontend Pipeline from GitHub to DO	VladimirAndrianov96	Done	Infrastructure	Operations	Medium
3	Digital Ocean | DB Initialization	VladimirAndrianov96	Done	Infrastructure	Operations	Medium
4	Digital Ocean | CMS Initialization	VladimirAndrianov96	Done	Infrastructure	Operations	Medium
5	Digital Ocean | Backend Initialization	VladimirAndrianov96	In Progress	Infrastructure	Operations	Low
6	Digital Ocean | Frontend Initialization	VladimirAndrianov96	Done	Infrastructure	Operations	Medium
+	Add item					
	Platform 6					
7	Create the Content Types for 404 Page	gokaiorg	Done	Platform	CMS Tasks	Low
8	Create the default 404 Page	gokaiorg	Done	Platform	Frontend Tasks	Low
9	Code the Login System using NextJS	VladimirAndrianov96	In Progress	Platform	Backend Tasks	Urgent
10	Documentation for Dev Work	gokaiorg	In Progress	Platform	Dev General	High
11	Create the dApp Base Template	gokaiorg	Done	Platform	Dev General	Medium
12	Create Logos for Project Scopes	gokaiorg	Code Review	Platform	Frontend Tasks	Medium
+	Add item					
	Marketplace 11					
13	Create the Marketplace Landing Page	gokaiorg	In Progress	Marketplace	Frontend Tasks	Medium
14	Landing Page | Structure | Sections	gokaiorg	In Progress	Marketplace	Frontend Tasks	High
15	Map and Detail the OpenSea Page Structure	ElrondWorld	In Progress	Marketplace	Dev General	Medium
16	Map and Detail the OpenSea Layout	ElrondWorld	In Progress	Marketplace	Dev General	Medium

Figure 16 GitHub Project Scope

The project space has the Kanban board, the tasks can be viewed by the project team and moved between different columns representing the status, as shown below in Figure 17.



Backlog 5

- Draft — Create Content Types for Marketplace Landing Page — Marketplace, CMS Tasks, Medium
- Draft — Landing Page | Section | Navbar — Marketplace, Frontend Tasks, Medium
- Draft — Landing Page | Section | Featured — Marketplace, Frontend Tasks, Medium
- Draft — Landing Page | Section | Notable Drops — Marketplace, Frontend Tasks, Medium
- Draft — Landing Page | Section | Top Collections — Marketplace, Frontend Tasks, Medium

In Progress 9

- Draft — Code the Login System using NextJS — Platform, Backend Tasks, Urgent
- Draft — Digital Ocean | Backend Initialization — Infrastructure, Operations, Low
- Draft — Documentation for Dev Work — Platform, Dev General, High
- Draft — Create the Marketplace Landing Page — Marketplace, Frontend Tasks, Medium
- Draft — Landing Page | Structure | Sections — Marketplace, Frontend Tasks, High
- Draft — Map and Detail the OpenSea Page Structure — Marketplace, Dev General, Medium
- Draft — Map and Detail the OpenSea Layout — Marketplace, Dev General, Medium
- Draft — Modify Grid to Flexbox for Responsive Transform — Marketplace, CSS Proposals, Low
- Draft — Map and Detail the OpenSea Functionality — Marketplace, Dev General, Medium

Code Review 1

- Draft — Create Logos for Project Scopes — Platform, Frontend Tasks, Medium

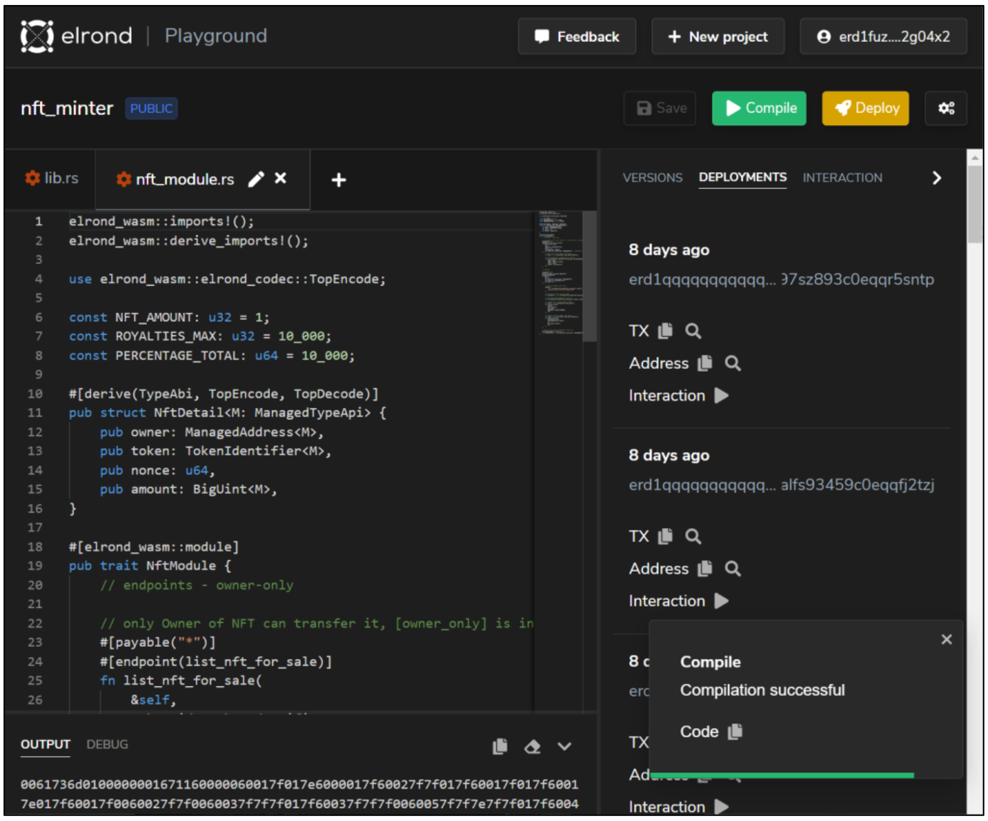
Done 8

- Draft — Digital Ocean | Registration — Infrastructure, Administration, High
- Draft — Create the Content Types for 404 Page — Platform, CMS Tasks, Low
- Draft — Setup Dev Frontend Pipeline from GitHub to DO — Infrastructure, Operations, Medium
- Draft — Digital Ocean | DB Initialization — Infrastructure, Operations, Medium
- Draft — Create the default 404 Page — Platform, Frontend Tasks, Low
- Draft — Digital Ocean | CMS Initialization — Infrastructure, Operations, Medium
- Draft — Digital Ocean | Frontend Initialization — Infrastructure, Operations, Medium
- Draft — Create the dApp Base Template — Platform, Dev General, Medium

Figure 17 GitHub project Kanban View

4.6 Elrond playground smart contract web IDE

Elrond’s rich developer tool pack also includes a web IDE for smart contracts development in Rust. The web IDE is free of charge and can be used to write code, compile it into WASM code, deploy a smart contract to the network and interact with smart contract endpoints from playground UI. The web IDE also supports multi-project files and dependencies and can be run without installing any dependencies on the local machine, the build and debug outputs are also available in UI, as shown below in Figure 18.



```rust
elrond_wasm::imports!();
elrond_wasm::derive_imports!();

use elrond_wasm::elrond_codec::TopEncode;

const NFT_AMOUNT: u32 = 1;
const ROYALTIES_MAX: u32 = 10_000;
const PERCENTAGE_TOTAL: u64 = 10_000;

#[derive(TypeAbi, TopEncode, TopDecode)]
pub struct NftDetail<M: ManagedTypeApi> {
    pub owner: ManagedAddress<M>,
    pub token: TokenIdentifier<M>,
    pub nonce: u64,
    pub amount: BigUint<M>,
}

#[elrond_wasm::module]
pub trait NftModule {
    // endpoints - owner-only

    // only Owner of NFT can transfer it, [owner_only] is in
    #[payable("*")]
    #[endpoint(list_nft_for_sale)]
    fn list_nft_for_sale(
        &self,
```

```
0061736d0100000001671160000060017f017e6000017f60027f7f017f60017f017f6001
7e017f60017f0060027f7f0060037f7f7f017f60037f7f7f0060057f7f7e7f7f017f6004
```

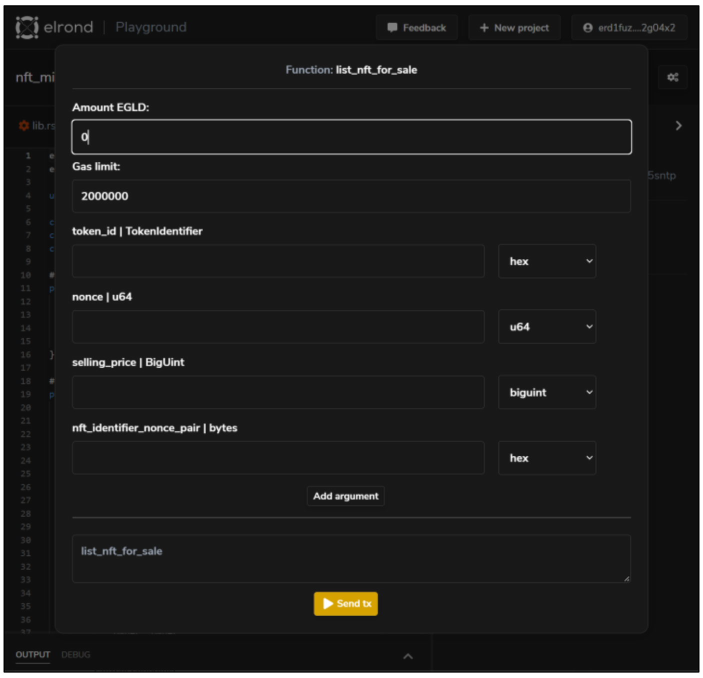
8 days ago
erd1qqqqqqqqqqq... 97sz893c0eqqr5sntp
TX
Address
Interaction

8 days ago
erd1qqqqqqqqqqq... alfs93459c0eqqfj2tzj
TX
Address
Interaction

Compile
Compilation successful
Code

Figure 18 Elrond Playground web-IDE UI

The smart contracts can also be tested within the test network from the interaction tab, where every smart contract endpoint is listed and can be picked for interaction, the form fields for endpoint input arguments are auto-generated, the transactions can also be signed and sent to the network within the interaction UI as on Figure 19.



Function: list_nft_for_sale

Amount EGLD:	0
Gas limit:	2000000
token_id | TokenIdentifier	hex
nonce | u64	u64
selling_price | BigUint	biguint
nft_identifier_nonce_pair | bytes	hex

Add argument

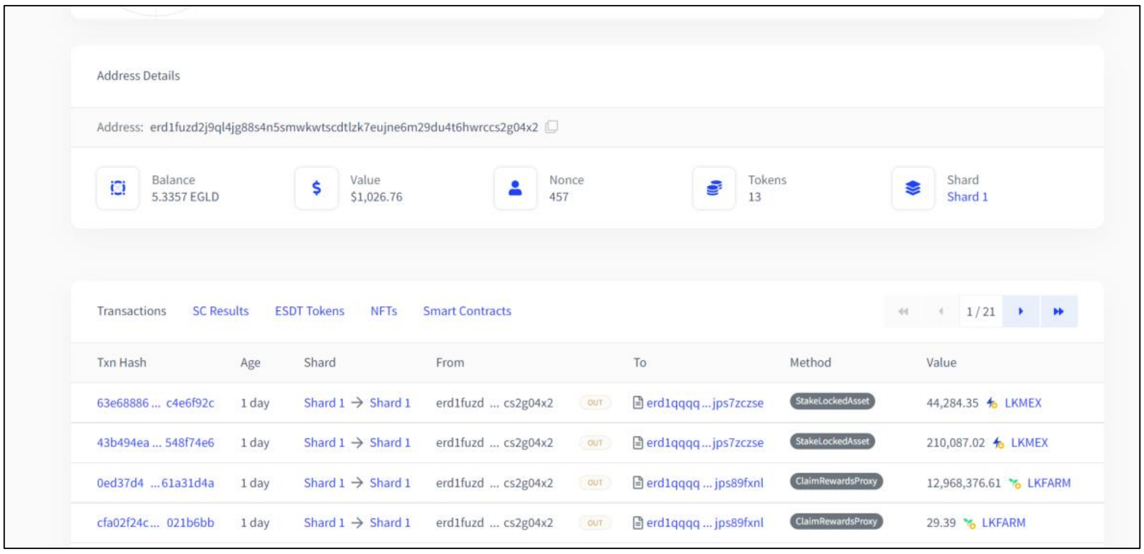
list_nft_for_sale

Send tx

Figure 19 Smart contract interaction within Elrond Playground

4.7 Elrond Explorer

Elrond Explorer is provided by the Elrond team, this is a web wallet explorer that shows current blockchain economics and enables search on the wallet and retrieves information in a human-readable format. Elrond Explorer lists transactions, smart contract results and endpoint used, ESDT tokens, NFTs and Smart Contracts per selected wallet. Elrond Explorer has been used to track the state of the wallet and explore transactions, data input and smart contract output, the general overview is shown in Figure 20.


Address Details

Address: erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2

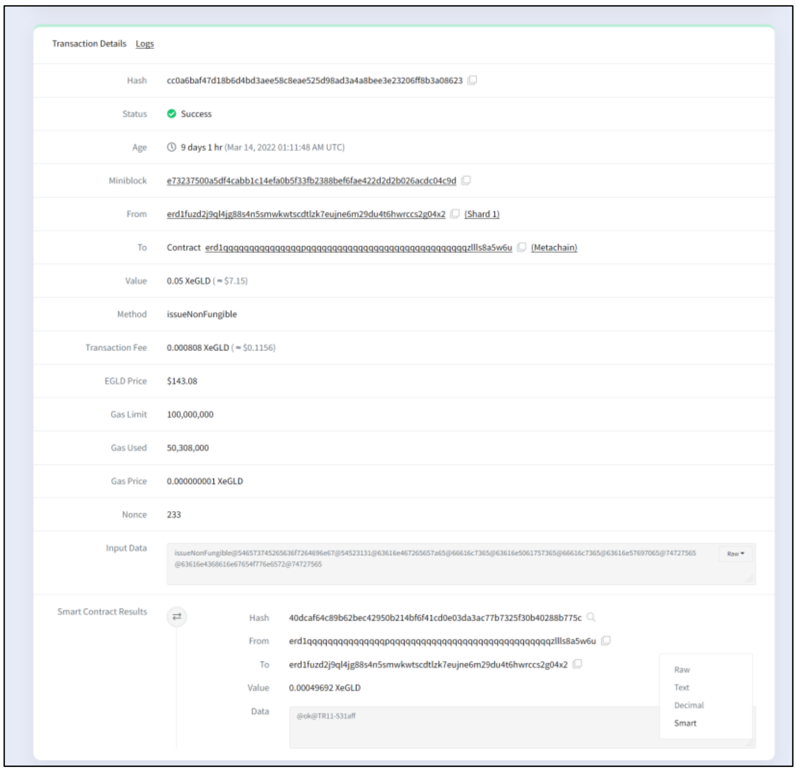
Balance	Value	Nonce	Tokens	Shard
5.3357 EGLD	$1,026.76	457	13	Shard 1

Transactions   SC Results   ESDT Tokens   NFTs   Smart Contracts   1 / 21

Txn Hash	Age	Shard	From		To	Method	Value
63e68886 ... c4e6f92c	1 day	Shard 1 → Shard 1	erd1fuzd ... cs2g04x2	OUT	erd1qqqq ...jps7zczse	StakeLockedAsset	44,284.35 LKMEX
43b494ea ... 548f74e6	1 day	Shard 1 → Shard 1	erd1fuzd ... cs2g04x2	OUT	erd1qqqq ...jps7zczse	StakeLockedAsset	210,087.02 LKMEX
0ed37d4 ...61a31d4a	1 day	Shard 1 → Shard 1	erd1fuzd ... cs2g04x2	OUT	erd1qqqq ... jps89fxnl	ClaimRewardsProxy	12,968,376.61 LKFARM
cfa02f24c... 021b6bb	1 day	Shard 1 → Shard 1	erd1fuzd ... cs2g04x2	OUT	erd1qqqq ... jps89fxnl	ClaimRewardsProxy	29.39 LKFARM

Figure 20 Customer wallet overview

Every transaction can be viewed in Elrond Explorer, the current example in Figure 21 shows the transaction to the smart contract's endpoint "issueNonFungible" to issue the non-fungible token used as a collection for the NFTs. By default, the data is shown in Raw format, but this can be formatted on choice as Raw, Text, Decimal or Smart view that formats raw data as both text and decimal based on input.



Hash	cc0a6baf47d18b6d4bd3aee58c8eae525d98ad3a4a8bee3e23206ff8b3a08623
Status	Success
Age	9 days 1 hr (Mar 14, 2022 01:11:48 AM UTC)
Miniblock	e73237500a5df4cabb1c14efa0b5f33fb2388bef6fae422d2d2b026acdc04c9d
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	Contract erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u (Metachain)
Value	0.05 XeGLD (≈ $7.15)
Method	issueNonFungible
Transaction Fee	0.000808 XeGLD (≈ $0.1156)
EGLD Price	$143.08
Gas Limit	100,000,000
Gas Used	50,308,000
Gas Price	0.000000001 XeGLD
Nonce	233
Input Data	issueNonFungible@546573745265636f7264696e67@54523131@63616e467265657a65@66616c7365@63616e5061757365@66616c7365@63616e57697065@74727565 @63616e4368616e67654f776e6572@74727565

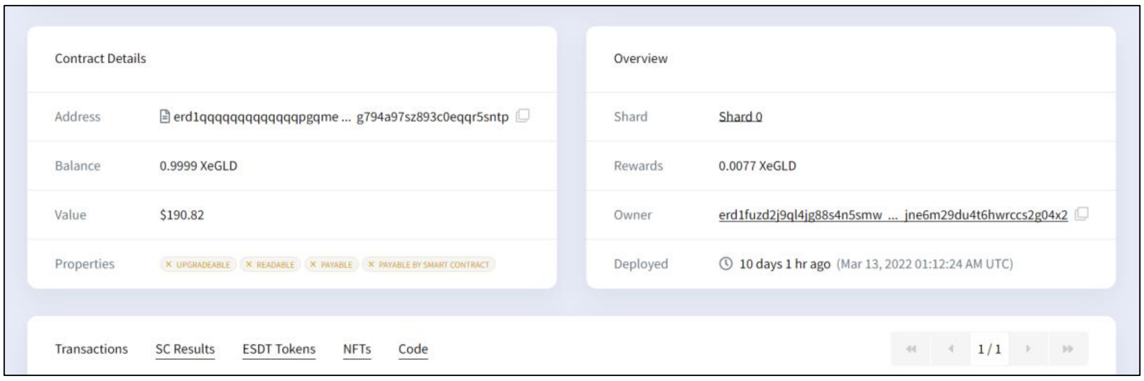
Smart Contract Results

Hash	40dcaf64c89b62bec42950b214bf6f41cd0e03da3ac77b7325f30b40288b775c
From	erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.00049692 XeGLD
Data	@ok@TR11-531aff

Figure 21 The "issueNonFungible" transaction overview in Elrond Explorer

4.8 Smart contracts

The Smart contract is a self-executing computer program on blockchain that controls the execution, these are typically used to automate processes and run independently. Smart contracts are deployed to the blockchain meaning these are immutable programs and can be tracked and validated by anyone eliminating the intermediary from operations. Smart contracts on Elrond are similar to wallets, these have balance, persist state, can own ESDT tokens, NFTs, smart contract results, process transactions, be viewed from Elrond Explorer, the main difference is that smart contract has a "code" section which is a WASM compiled code. The major difference from the ordinary wallet is that smart contracts can use storage to persist state, another significant difference is that the smart contract has an owner and rewards, rewards are: 30% of transaction gas fees generated on the network by the smart contract, these can be claimed as seen on Figure 22.



Contract Details

Address	erd1qqqqqqqqqqqqqpgqme ... g794a97sz893c0eqqr5sntp
Balance	0.9999 XeGLD
Value	$190.82
Properties	✕ UPGRADEABLE ✕ READABLE ✕ PAYABLE ✕ PAYABLE BY SMART CONTRACT

Overview

Shard	Shard 0
Rewards	0.0077 XeGLD
Owner	erd1fuzd2j9ql4jg88s4n5smw ... jne6m29du4t6hwrccs2g04x2
Deployed	10 days 1 hr ago (Mar 13, 2022 01:12:24 AM UTC)

Figure 22 Smart contract overview

Smart contracts facilitate the entire market flow bringing trust and transparency to the platform. The platform does not own any digital assets listed for sale; everything is done through the deployed smart contract only, meaning digital assets cannot be stolen from the marketplace, so it cannot be compromised as an eMoon marketplace. Owners manage the digital assets, and only the owners can sign transactions for any operations with digital assets on the smart contract, securing customers' assets too. The only issue is that the marketplace smart contract completely manages assets, meaning a mechanism must be developed to let owners of assets retrieve their assets from the smart contract balance.

4.8.1 Data serialization

The data must be serialized before it can be placed in a transaction input data, in Rust, the Elrond-codec crate handles the data serialization. That means the strings and numbers must be in hexadecimal encoding. The numbers like royalties that can be any number from 0 to 10000 must be in hexadecimal encoding and also must have even length, e.g., decimal 1 must have leading zero as only 01 is a correctly encoded decimal number that can be deserialized.

4.8.2 Integrated smart contract endpoints

The collection token and NFT creation are handled by the default smart contract provided by the Elrond team that facilitates these operations, and the Rust code functions in the smart contract SDK also hit the same smart contract by default for operations on the collection and NFT, meaning that the custom smart contract will hit the same smart contract endpoint, meaning custom endpoints for these operations do not make sense as it only introduces extra gas costs thus this must be avoided.

The optimal flow is to integrate collection creation, collection role management and NFT creation with Elrond's default smart contract to avoid extra costs and complexity. The business logic for marketplace trade functionality isn't provided by Elrond smart contract, so the custom smart contract was developed to provide endpoints to facilitate the business needs, these are the "list_nft_for_sale", "update_price", "unlist_nft" and "buy_nft" endpoints.

The trade flow combines Elrond and marketplace smart contracts with the customer's wallet, the Figure 23 shows that the wallet uses Elrond smart contract for basic interactions to create the NFT, and marketplace smart contract to list it for sale, update price, buy it or unlist it from the sale.

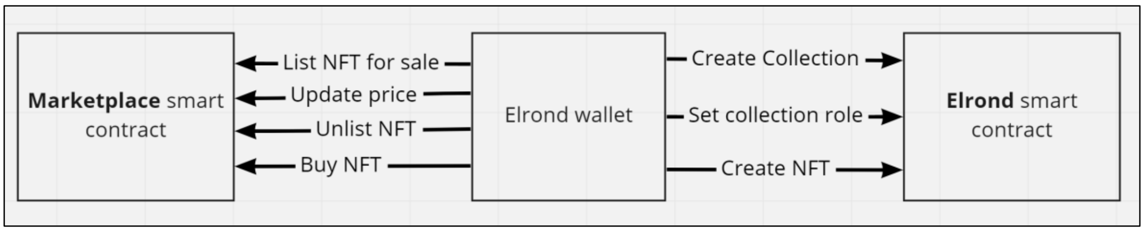


Figure 23 Trade flow smart contract endpoints

4.8.3 Author royalties and payments

On Elrond, every NFT has a "royalties" field to allow creators to receive rewards for any transaction involving their NFT, royalties are set as a numeric value between 0 and 10000, representing royalties from 0% to 100%, the royalties are not paid automatically on every transaction, this must be explicitly handled within the smart contract. The marketplace business model contains a 3% fee from any sale committed within the platform, which must be explicitly handled within the smart contract as NFT owner's profit as well. Every buy operation of NFT will send payments to the author, NFT owner and marketplace wallet. A smart contract gets rewarded for every transaction it generates, meaning there must be a mechanism to withdraw rewards too.

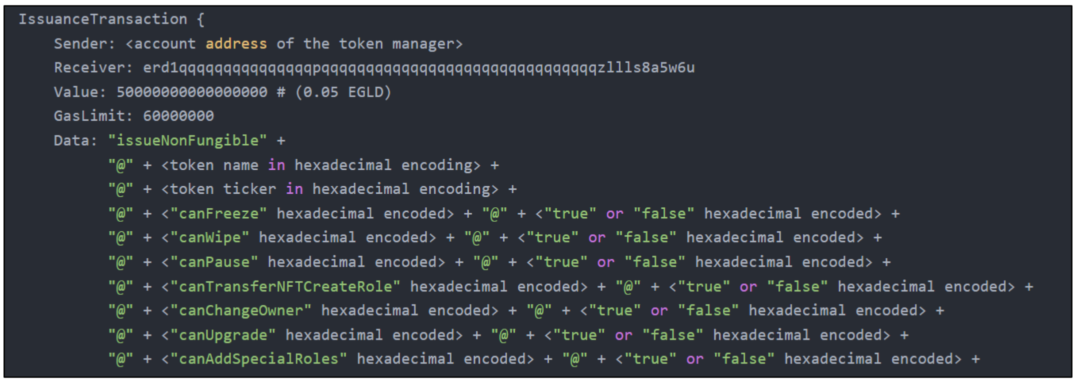
4.8.4 Endpoints and transactions data

Every transaction to the smart contract must include input data, input data must contain the smart contract method and all required attributes for the smart contract endpoint. The input data must be converted to hexadecimal values and have an even length. Input data might include another smart contract address for a nested call, this allows NFT owners to transfer their NFT to the smart contract and also execute the endpoint to trigger the smart contract to do some action with NFT. Only the owner of NFT can transfer it to another wallet since smart contracts cannot take it from the NFT owner’s wallet, customer must do this transfer by calling the Elrond’s smart contract combined with the nested call to

4.8.5 Elrond default methods on the marketplace

4.8.5.1 Issue non-fungible token for collection

Every NFT created on Elrond requires a collection token to be created, so in order to create NFTs, the collection must always be created first, this transaction from Figure 24 executes the “issueNonFungible” endpoint of Elrond’s smart contract, which is an asynchronous action and result of this call is a token added created on the caller’s wallet, and the token identifier will be printed in smart contract’s output. The token identifier is a unique identifier on the blockchain, it is a combination of a user-defined token ticker which is a non-human readable identifier for the token, and the random string added to it, e.g., collection with the human-readable name “HauntedHouseCrazyGhost” with ticker “HHCG” could have a token identifier looking like HHCG-6258d2.



```
IssuanceTransaction {
    Sender: <account address of the token manager>
    Receiver: erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u
    Value: 50000000000000000 # (0.05 EGLD)
    GasLimit: 60000000
    Data: "issueNonFungible" +
          "@" + <token name in hexadecimal encoding> +
          "@" + <token ticker in hexadecimal encoding> +
          "@" + <"canFreeze" hexadecimal encoded> + "@" + <"true" or "false" hexadecimal encoded> +
          "@" + <"canWipe" hexadecimal encoded> + "@" + <"true" or "false" hexadecimal encoded> +
          "@" + <"canPause" hexadecimal encoded> + "@" + <"true" or "false" hexadecimal encoded> +
          "@" + <"canTransferNFTCreateRole" hexadecimal encoded> + "@" + <"true" or "false" hexadecimal encoded> +
          "@" + <"canChangeOwner" hexadecimal encoded> + "@" + <"true" or "false" hexadecimal encoded> +
          "@" + <"canUpgrade" hexadecimal encoded> + "@" + <"true" or "false" hexadecimal encoded> +
          "@" + <"canAddSpecialRoles" hexadecimal encoded> + "@" + <"true" or "false" hexadecimal encoded> +
```

Figure 24 “issueNonFungible” transaction [20]

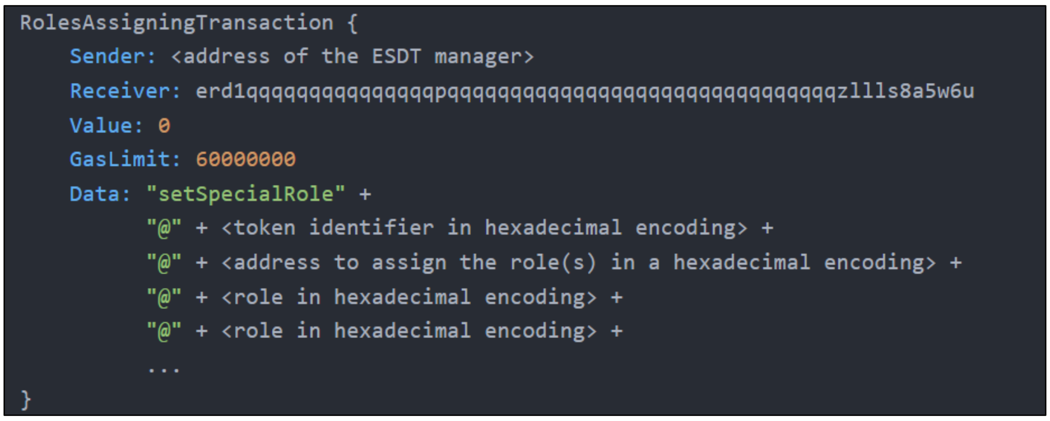
The results of this transaction can be found under Appendix 3 – “issueNonFungible” transaction view and execution results in Table 2 below.

Endpoint	issueNonFungible
Sender	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Receiver	erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u
input	
Decoded data	issueNonFungible@TestRecording@TR11@canFreeze@false @canPause@false@canWipe@true@canChangeOwner@true
output	
Decoded data	@ok@TR11-531aff

Table 2 "issueNonFungible" transaction

4.8.5.2 Set special role for collection

To create NFTs for collection, the collection must have the “ESDTNFTCreate” role set that allows the “ESDTNFTCreate” calls. This transaction requires an identifier obtained from the previous “issueNonFungible” transaction to be passed in as input, and the address to assign the role is a caller’s address who owns the collection and sets rules for it, as shown in Figure 25, as only the owner of collection is allowed to manage token at this point.



```
RolesAssigningTransaction {
    Sender: <address of the ESDT manager>
    Receiver: erd1qqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u
    Value: 0
    GasLimit: 60000000
    Data: "setSpecialRole" +
          "@" + <token identifier in hexadecimal encoding> +
          "@" + <address to assign the role(s) in a hexadecimal encoding> +
          "@" + <role in hexadecimal encoding> +
          "@" + <role in hexadecimal encoding> +
          ...
}
```

Figure 25 "setSpecialRole" transaction [20]

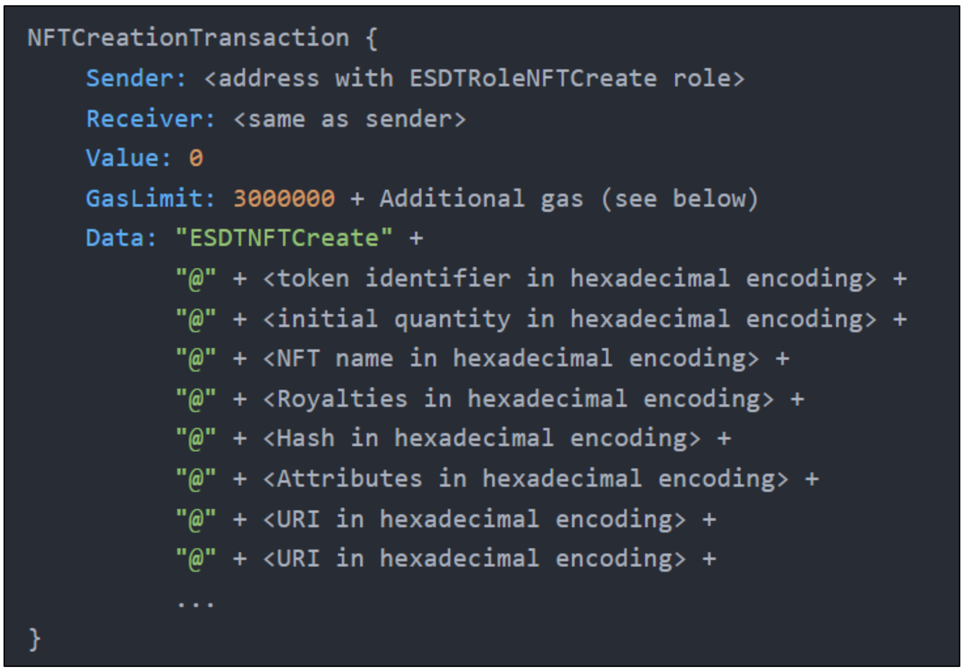
The results of this transaction can be found under Appendix 4 – “setSpecialRole” transaction smart view and execution results in Table 3 below.

Endpoint	setSpecialRole
Sender	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Receiver	erd1qqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u
input	
Decoded data	setSpecialRole@TR11-531aff @erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 @ESDTRoleNFTCreate
output	
Decoded data	ESDTSetRole@TR11-531aff@ESDTRoleNFTCreate
Decoded data	@ok

Table 3 "setSpecialRole" transaction

4.8.5.3 Create NFT

After the collection has been created and the “ESDTCreate” rule has been set for it, the NFT can be created, and it will be attached to the created collection, that endpoint requires the same token identifier returned from the first step, and the NFT related data, the difference is that the sender and receiver addresses are the same for NFT creation endpoint as shown on Figure 26. The result of the transaction is NFT created and attached to the caller’s collection.



```
NFTCreationTransaction {
    Sender: <address with ESDTRoleNFTCreate role>
    Receiver: <same as sender>
    Value: 0
    GasLimit: 3000000 + Additional gas (see below)
    Data: "ESDTNFTCreate" +
          "@" + <token identifier in hexadecimal encoding> +
          "@" + <initial quantity in hexadecimal encoding> +
          "@" + <NFT name in hexadecimal encoding> +
          "@" + <Royalties in hexadecimal encoding> +
          "@" + <Hash in hexadecimal encoding> +
          "@" + <Attributes in hexadecimal encoding> +
          "@" + <URI in hexadecimal encoding> +
          "@" + <URI in hexadecimal encoding> +
          ...
}
```

Figure 26 "ESDTNFTCreate" transaction [20]

The results of this transaction can be found under Appendix 5 – "ESDTNFTCreate" transaction view and execution results in Table 4 below.

Endpoint	ESDTNFTCreate
Sender	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Receiver	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
input	
Decoded data	ESDTNFTCreate@TR11-531aff@01@Build the world! @1500@@key1:value1;key2:value2 @https://ipfs.io/images/command-line-hex.png
output	
Decoded data	@ok@01

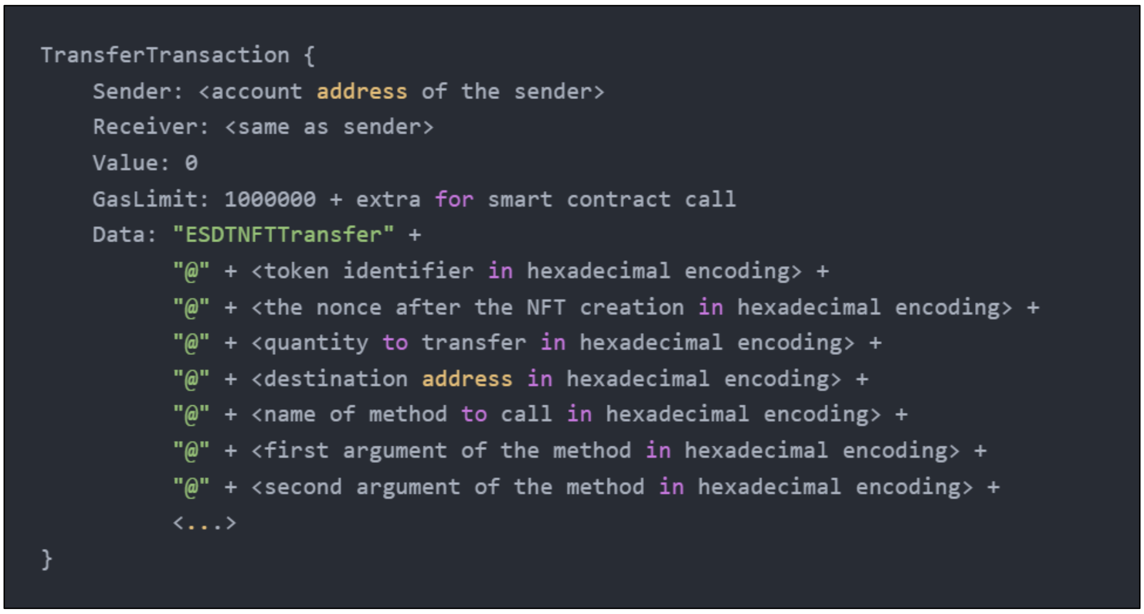
Table 4 "ESDTNFTCreate" transaction

4.8.6 Custom smart contract methods on the marketplace

4.8.6.1 List NFT for sale

Any existing NFT on Elrond can be listed for sale on the marketplace if it is available on the customer's wallet, meaning any customer with NFTs on Elrond can come to the marketplace and list NFTs for sale or create NFTs on the marketplace platform and list these for sale also.

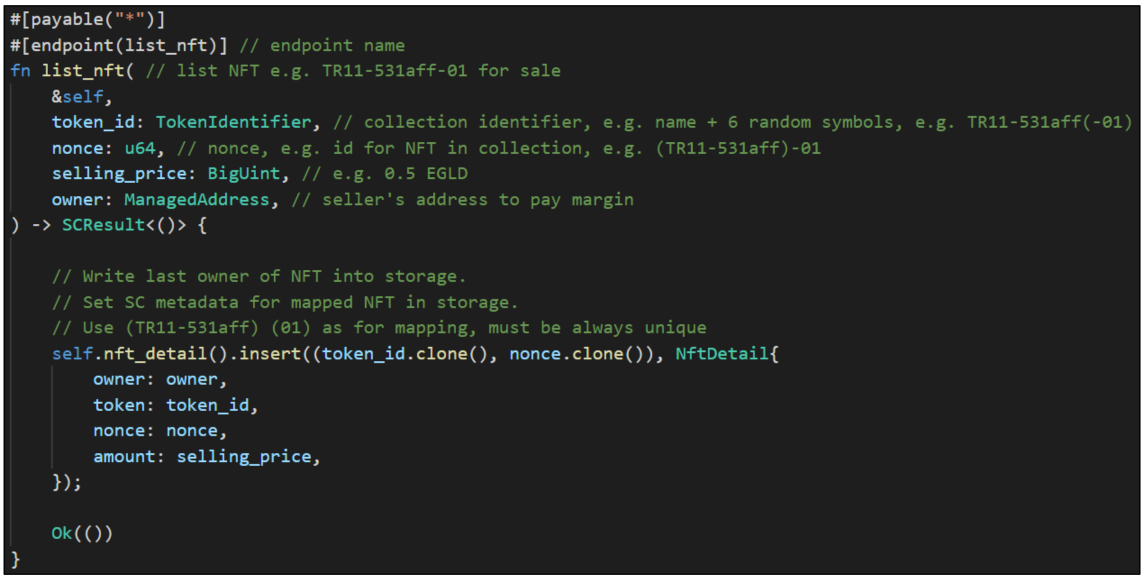
The listing cannot be done directly from the customer wallet to the smart marketplace contract due to the architecture constraints of Elrond. Elrond smart contracts have only owner annotation that allows the only owner of token to manage it, that constraint is applied to Elrond's default functions, such as role management for collection, NFT creation and actions such as NFT transfer to marketplace wallet, that can be done only through Elrond built via ESDTNFTTransfer function call. The ESDTNFTTransfer call supports simple NFT transfer to another wallet and data transfers to the smart contract method. The ESDT NFT transfer to smart contract transaction data must contain the destination address and arguments if any, as specified in the documentation in Figure 27.



```
TransferTransaction {
    Sender: <account address of the sender>
    Receiver: <same as sender>
    Value: 0
    GasLimit: 1000000 + extra for smart contract call
    Data: "ESDTNFTTransfer" +
          "@" + <token identifier in hexadecimal encoding> +
          "@" + <the nonce after the NFT creation in hexadecimal encoding> +
          "@" + <quantity to transfer in hexadecimal encoding> +
          "@" + <destination address in hexadecimal encoding> +
          "@" + <name of method to call in hexadecimal encoding> +
          "@" + <first argument of the method in hexadecimal encoding> +
          "@" + <second argument of the method in hexadecimal encoding> +
          <...>
}
```

Figure 27 "ESDTNFTTransfer" transaction [20]

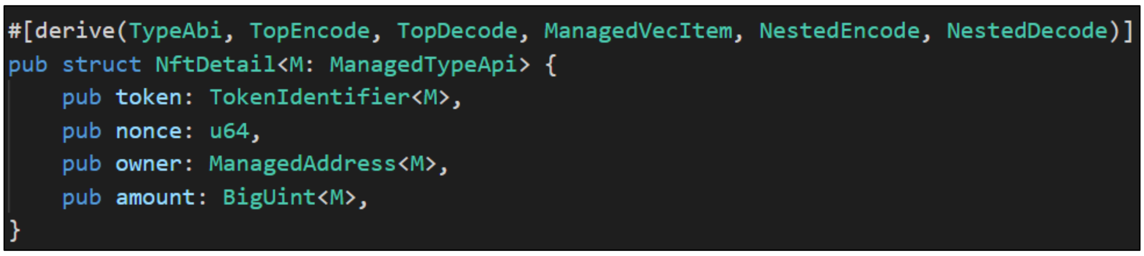
This call allows NFT to be sent, the receiver address is the same as the sender's account, so the smart contract endpoint is actually a user's wallet, that is an owner of the NFT thus, it is allowed by the owner only annotation and has the rights to commit operations with the token. After the transaction is signed, it executes the destination smart contract's method with the input data specified as arguments, meaning there is a nested call to the external smart contract method that can fall. The error handling is handled by default in Elrond, the ESDTNFTTransfer method reserves NFT for the entire transaction, waiting until all the smart contract results are available, and if there is an error returned from nested smart contract methods, that will roll back the NFT transfer, and send it back to the original owner of it. This rollback functionality works by default on Elrond, this secures NFTs from being transferred to the marketplace for a listing and being stuck forever on marketplace balance if the listing method has failed. If the listing method fails when NFT is transferred to the marketplace, that will mean the NFT will be on the balance of the smart marketplace contract, but it will not be mapped to the smart contract's storage or have any price set or have any information about the seller in smart contract storage so that NFT will not get back to the owner or be able to be ever sold if sale logic is based on marketplace's storage.



```
#[payable("*")]
#[endpoint(list_nft)] // endpoint name
fn list_nft( // list NFT e.g. TR11-531aff-01 for sale
    &self,
    token_id: TokenIdentifier, // collection identifier, e.g. name + 6 random symbols, e.g. TR11-531aff(-01)
    nonce: u64, // nonce, e.g. id for NFT in collection, e.g. (TR11-531aff)-01
    selling_price: BigUint, // e.g. 0.5 EGLD
    owner: ManagedAddress, // seller's address to pay margin
) -> SCResult<()> {

    // Write last owner of NFT into storage.
    // Set SC metadata for mapped NFT in storage.
    // Use (TR11-531aff) (01) as for mapping, must be always unique
    self.nft_detail().insert((token_id.clone(), nonce.clone()), NftDetail{
        owner: owner,
        token: token_id,
        nonce: nonce,
        amount: selling_price,
    });

    Ok(())
}
```

Figure 28 “list_nft” method

The screenshot consists of a part of the actual smart contract code written in Rust language, as listed above in Figure 28, the custom list_nft_for_sale method has arguments that will be set in the ESDTNFTTransfer transaction, the arguments for listing are specified on the front-end platform sets the listing related data into a transaction before it is signed. Endpoint requires the token identifier for a collection, e.g., “TR11-531aff”, and a nonce for NFT, e.g., “01” these are used to map the collection and NFT to the smart contract storage that will keep required data for listing. The smart contract stores the data structured in a custom struct called “NftDetail” from Figure 29, that struct links the owner, token, nonce and selling price together so the NFT can be sold with some price in future. Since this method is called from Elrond's smart contract that transfers the NFT, the caller of the transaction is not an NFT owner but the Elrond smart code, so the original owner's address is also passed as an argument and stored internally.



```
#[derive(TypeAbi, TopEncode, TopDecode, ManagedVecItem, NestedEncode, NestedDecode)]
pub struct NftDetail<M: ManagedTypeApi> {
    pub token: TokenIdentifier<M>,
    pub nonce: u64,
    pub owner: ManagedAddress<M>,
    pub amount: BigUint<M>,
}
```

Figure 29 “NftDetail” struct

The transaction has nested transaction arguments, so transaction data is noticeably larger than in other transactions, example is for listing with 1 EGLD price set, the transaction arguments are listed in Table 5 below and transaction view in Appendix 6.

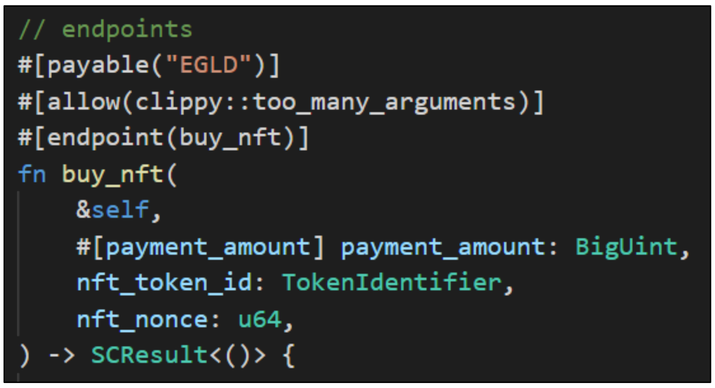
Endpoint	ESDTNFTTransfer
Sender	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Receiver	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Nested receiver	erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g
input	
Decoded data	ESDTNFTTransfer@TR11-531aff@01@01 @erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g @list_nft@TR11-531aff@01 @1000000000000000000 @erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
output	
Status	success

Table 5 “list_nft” transaction

The successfully transferred NFT with successfully executed nested listing method results in an NFT added to listing on the marketplace that can be fetched and listed for sale on the front-end. The endpoint processed the NFT, and NFT data with additional price data is stored in the smart contract storage, so it can be retrieved through the view functions and can be used to process the purchase and validate the data.

4.8.6.2 Buy NFT

The endpoint for buying NFTs can be called directly from the buyer's wallet to the marketplace's smart contract as this does not require any assets being sent from the buyer's wallet that require special permissions, the EGLD value from the wallet balance can be sent directly to the transaction value, and the marketplace's smart contract as the owner of NFT is entitled to send it to the buyer's wallet. The endpoint requires only token identifier and NFT nonce to be sent as listed in Figure 30, and the value of the transaction is auto-added to the transaction and fetched from there, so this does not need to be sent as input data.



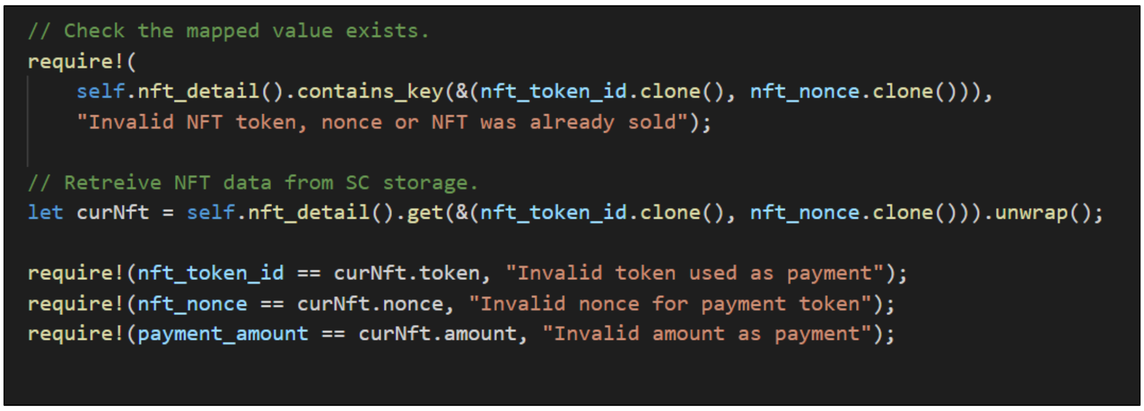
```
// endpoints
#[payable("EGLD")]
#[allow(clippy::too_many_arguments)]
#[endpoint(buy_nft)]
fn buy_nft(
    &self,
    #[payment_amount] payment_amount: BigUint,
    nft_token_id: TokenIdentifier,
    nft_nonce: u64,
) -> SCResult<()> {
```

Figure 30 “buy_nft” method arguments

As the purchase is handled manually within the smart contract, the total transaction value must match the value of the NFT listed for sale to prevent NFTs from being sold for the wrong price and also to prevent NFT operations on the smart contract if the NFT is not available on the listing even it is on the balance, e.g., NFT has been sent manually to the smart contract and did not go through the listing endpoint as a result of the attempt of exploiting it. The listing and input data are validated against the smart contract storage as shown in Figure 31, and transaction data from appendix 7 is as in Table 6 below.

Endpoint	buy_nft
Sender	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	1 EGLD
Receiver	erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g
input	
Decoded data	buyNFT@TR11-531aff@01
output	
Status	success

Table 6 “buy_nft" transaction

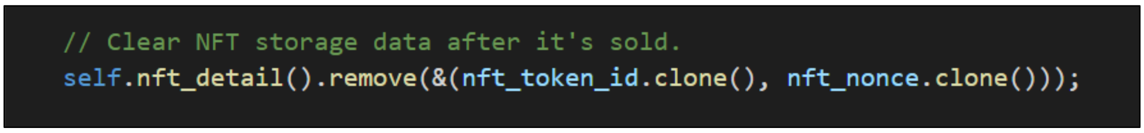


```
// Check the mapped value exists.
require!(
    self.nft_detail().contains_key(&(nft_token_id.clone(), nft_nonce.clone())),
    "Invalid NFT token, nonce or NFT was already sold");

// Retreive NFT data from SC storage.
let curNft = self.nft_detail().get(&(nft_token_id.clone(), nft_nonce.clone())).unwrap();

require!(nft_token_id == curNft.token, "Invalid token used as payment");
require!(nft_nonce == curNft.nonce, "Invalid nonce for payment token");
require!(payment_amount == curNft.amount, "Invalid amount as payment");
```

Figure 31 “buy_nft” method data validation

The input token identifier and the NFT nonce are used to retrieve the listing object from the storage, this prevents any operations with NFTs that aren't listed for sale through the listing endpoint, e.g., NFTs were sent to the balance, so only the correct listings will be processed, and token identifier with NFT nonce and the value of the sale from the listing will be compared against the passed values from arguments and transaction.

When all data has been validated, the creator royalties, marketplace service fee and amount to be paid to the NFT seller are calculated, and the author royalties are fetched from the blockchain via the Elrond WASM API “get_esdt_token_data” function, which also requires valid owner's address and validates it. After all amounts were calculated, the NFT is sent to the buyer's wallet, the marketplace service fee is paid to the owner of the marketplace smart contract owner, royalties are paid to the creator, and leftover value is transferred to the seller's wallet as revenue from the sale as shown on Appendix 8. To prevent double spending, after NFT is sold, it is removed from the smart contract storage as it's done in Figure 32, meaning the same listing cannot be used twice, but the new owner can list it again on the marketplace, and the selling process will work again.



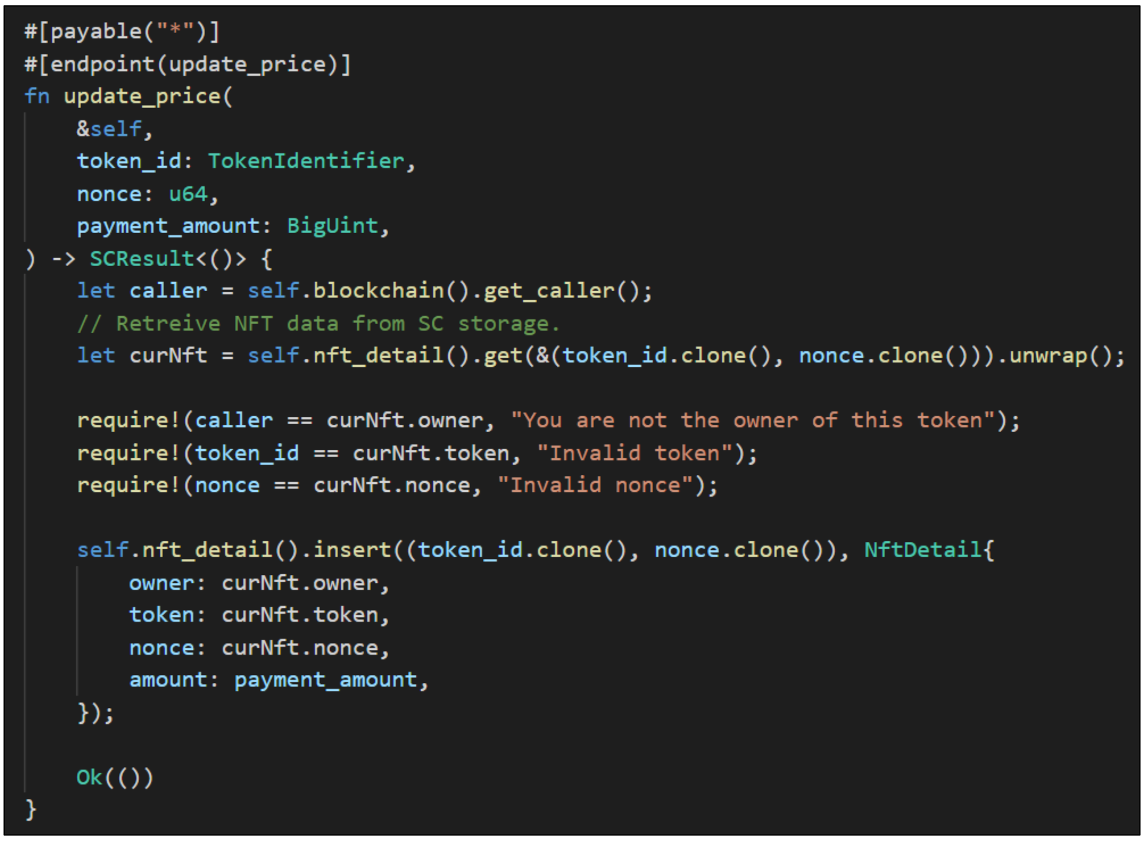
```
// Clear NFT storage data after it's sold.
self.nft_detail().remove(&(nft_token_id.clone(), nft_nonce.clone()));
```

Figure 32 Clearing the smart contract storage mapped value

To prevent double spending meanwhile the smart contract executes the purchase endpoint flow, the ESDTNFTTransfer function reserves NFT, taking it from the smart contract balance, so the NFT transfer cannot be executed twice, meaning the transaction will fail.

4.8.6.3 Update listing price

The smart contract allows to update the listing price update with no need to unlist and pay for unnecessary transactions, so the NFT is not transferred back and forth to the marketplace's smart contract as the price as a value in object kept in smart contract storage, so it is just being updated with the new value in a single transaction as shown on Figure 33.



```
#[payable("*")]
#[endpoint(update_price)]
fn update_price(
    &self,
    token_id: TokenIdentifier,
    nonce: u64,
    payment_amount: BigUint,
) -> SCResult<()> {
    let caller = self.blockchain().get_caller();
    // Retreive NFT data from SC storage.
    let curNft = self.nft_detail().get(&(token_id.clone(), nonce.clone())).unwrap();

    require!(caller == curNft.owner, "You are not the owner of this token");
    require!(token_id == curNft.token, "Invalid token");
    require!(nonce == curNft.nonce, "Invalid nonce");

    self.nft_detail().insert((token_id.clone(), nonce.clone()), NftDetail{
        owner: curNft.owner,
        token: curNft.token,
        nonce: curNft.nonce,
        amount: payment_amount,
    });

    Ok(())
}
```

Figure 33 “update_price” method

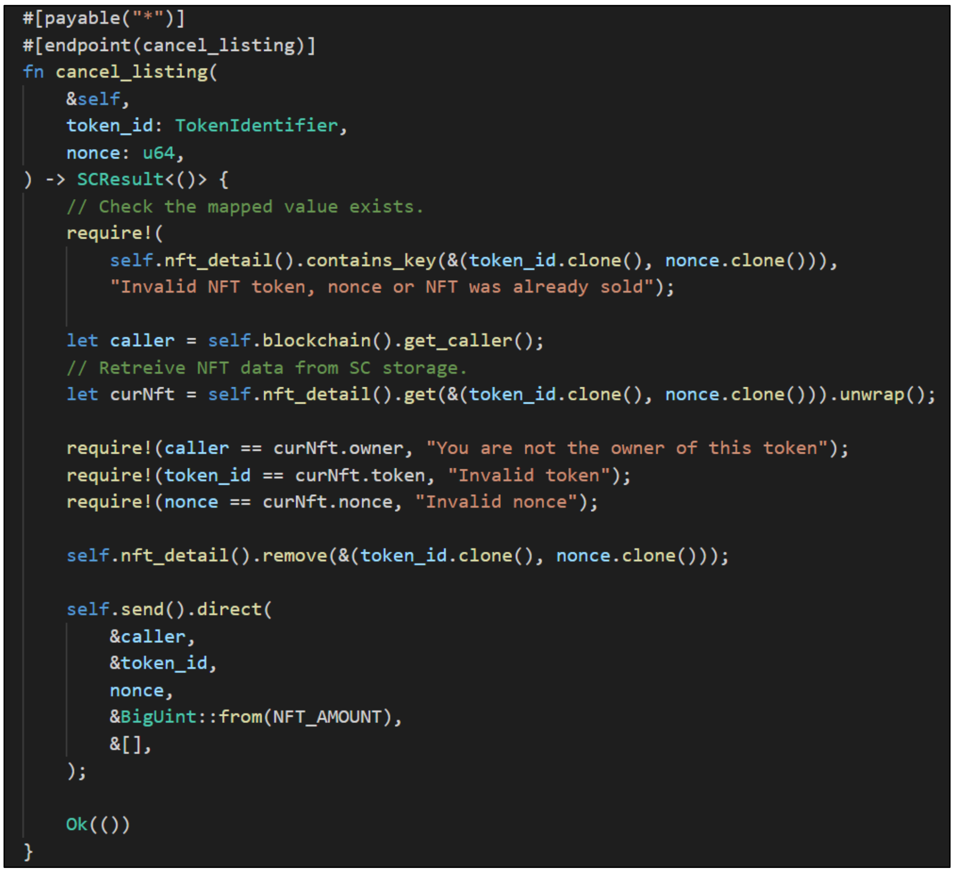
The transaction body and results are listed in Appendix 9, only one smart contract execution is required for the entire operation and required transaction data is listed below in Table 7.

Endpoint	update_price
Sender	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Receiver	erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g
input	
Decoded data	update_price@COLNAME-5d5729@02@500000000000000000
output	
Status	success

Table 7 "update_price" transaction

4.8.6.4 Cancel listing

Cancel listing endpoint allows removing listing from the marketplace, removing it from the marketplace storage and sending NFT back to the seller's wallet with no need to pay any price. Only the owner of the NFT is allowed to cancel the listing, receiving NFT not paying the actual sale price, the caller's owner is validated against the seller's address stored in the smart contract storage, as shown in Figure 34.



```rust
#[payable("*")]
#[endpoint(cancel_listing)]
fn cancel_listing(
    &self,
    token_id: TokenIdentifier,
    nonce: u64,
) -> SCResult<()> {
    // Check the mapped value exists.
    require!(
        self.nft_detail().contains_key(&(token_id.clone(), nonce.clone())),
        "Invalid NFT token, nonce or NFT was already sold");

    let caller = self.blockchain().get_caller();
    // Retreive NFT data from SC storage.
    let curNft = self.nft_detail().get(&(token_id.clone(), nonce.clone())).unwrap();

    require!(caller == curNft.owner, "You are not the owner of this token");
    require!(token_id == curNft.token, "Invalid token");
    require!(nonce == curNft.nonce, "Invalid nonce");

    self.nft_detail().remove(&(token_id.clone(), nonce.clone()));

    self.send().direct(
        &caller,
        &token_id,
        nonce,
        &BigUint::from(NFT_AMOUNT),
        &[],
    );

    Ok(())
}
```

Figure 34 “cancel_listing” method

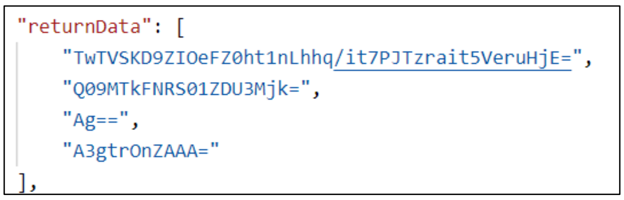
The transaction body and results are listed in Appendix 10, only one smart contract execution is required for the entire operation and required transaction data is listed below in Table 8.

Endpoint	cancel_listing
Sender	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Receiver	erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g
input	
Decoded data	cancel_listing@COLNAME-5d5729@02
output	
Status	success

Table 8 "cancel_listing " transaction

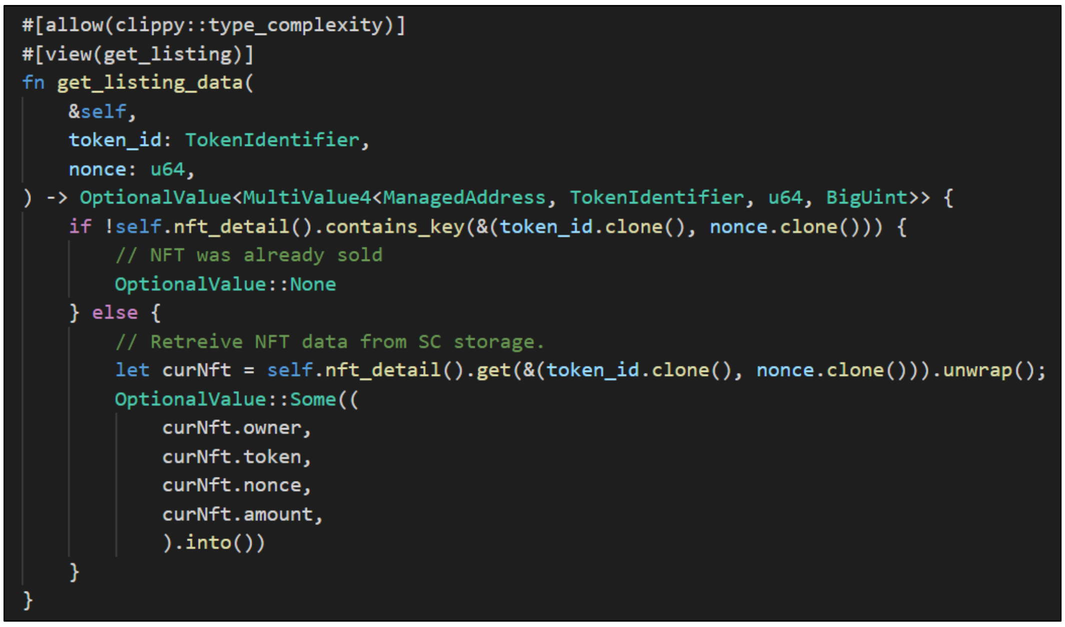
4.8.6.5 Get listing data view

The view endpoints don’t require transactions to be committed to the blockchain or require any money to be run, the view method of the smart contract can be used directly by the client as shown in Appendix 11 in Postman request, that is the common way to read the smart contract items. The WASM machine is running on the client’s web browser and fetches the data from the SC, so it is always free to use, the “get_listing” view method returns the encoded NFT owner address, token identifier, nonce and the listing price as shown on Figure 35 below.



```
"returnData": [
    "TwTVSKD9ZIOeFZ0ht1nLhhq/it7PJTzrait5VeruHjE=",
    "Q09MTkFNRS01ZDU3Mjk=",
    "Ag==",
    "A3gtrOnZAAA="
],
```

Figure 35 Listing data



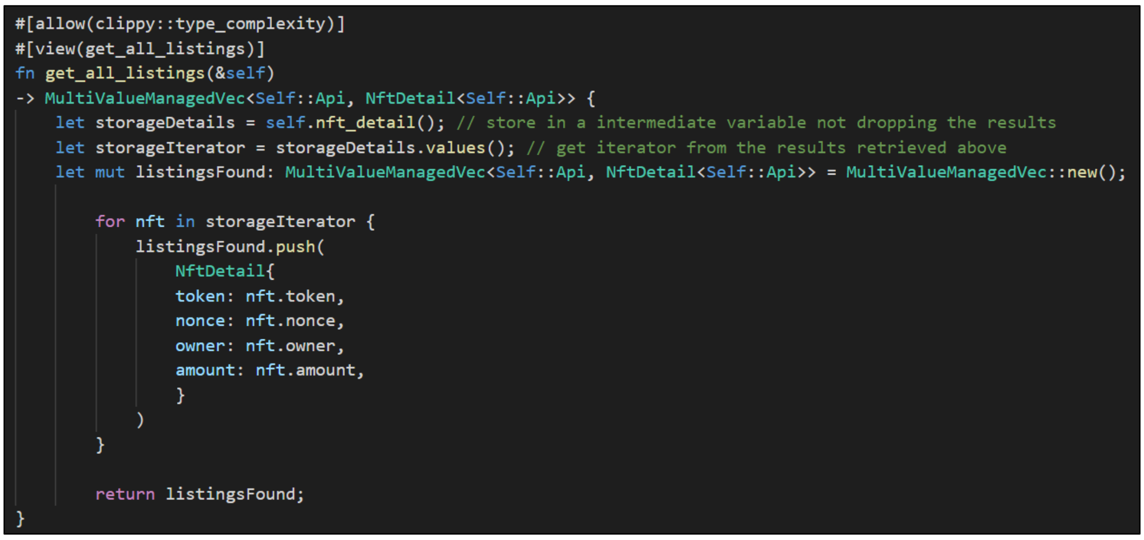
```rust
#[allow(clippy::type_complexity)]
#[view(get_listing)]
fn get_listing_data(
    &self,
    token_id: TokenIdentifier,
    nonce: u64,
) -> OptionalValue<MultiValue4<ManagedAddress, TokenIdentifier, u64, BigUint>> {
    if !self.nft_detail().contains_key(&(token_id.clone(), nonce.clone())) {
        // NFT was already sold
        OptionalValue::None
    } else {
        // Retreive NFT data from SC storage.
        let curNft = self.nft_detail().get(&(token_id.clone(), nonce.clone())).unwrap();
        OptionalValue::Some((
            curNft.owner,
            curNft.token,
            curNft.nonce,
            curNft.amount,
            ).into())
    }
}
```

Figure 36 “get_listing” view method

The request and response from the “get_listing” endpoint from Figure 36 above are shown in Appendix 11, all the data returned in the “returnData” section is base64 encoded, meaning every value comes in Elrond’s serialization format with base64 encoding on top of it, so after the decoding, the data must be encoded from HEX to, e.g., SHA hash of address or hexadecimal value of the nonce. The “OptionalValue” type has been used since there might be no data response in case the NFT is not listed anymore, meaning there is no listing information stored at the SC anymore.

4.8.6.6 Get all listings view

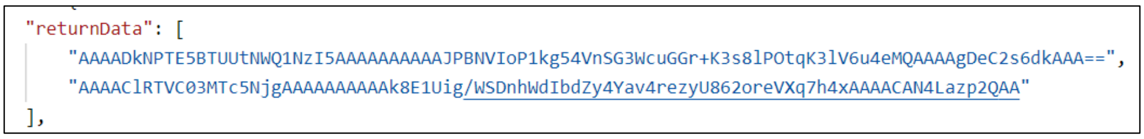
Since not all NFTs on the balance of the marketplace can be listed for sale, e.g., when some NFT was sent directly not triggering the listing endpoint, there must be a mechanism to provide only valid listings in the response, and only the internal storage of marketplace has knowledge about the values of NFT set, so the data must be exposed to the frontend. In order to achieve that, the "SingleValueMapper" cheapest mapper does not fit here as it does not provide any method to iterate through all the mapped values [21]. The "MapMapper" was selected to utilize the mapper needs with the ability to iterate over all keys, it works using the mapping key similarly to "SingleValueMapper", with the difference that it is extended structure and any operations with it will cost more gas compared to simple solutions with no iteration ability. The view endpoint is shown above in figure 37 below, showing the listings data is extracted from the smart contract memory before it is returned. The available data structure to return the array of objects is the "MultiValueManagedVec", which is a vector of values that can accept any number of values of a specified type pushed to it.



```rust
#[allow(clippy::type_complexity)]
#[view(get_all_listings)]
fn get_all_listings(&self)
-> MultiValueManagedVec<Self::Api, NftDetail<Self::Api>> {
    let storageDetails = self.nft_detail(); // store in a intermediate variable not dropping the results
    let storageIterator = storageDetails.values(); // get iterator from the results retrieved above
    let mut listingsFound: MultiValueManagedVec<Self::Api, NftDetail<Self::Api>> = MultiValueManagedVec::new();

        for nft in storageIterator {
            listingsFound.push(
                NftDetail{
                token: nft.token,
                nonce: nft.nonce,
                owner: nft.owner,
                amount: nft.amount,
                }
            )
        }

        return listingsFound;
}
```

Figure 37 get_all_listings" view method

“MapMapper” supports the iteration, so the iterator has been extracted and used to iterate over the map of values, where each value has been read and added to the vector of values, and when all the values from the storage were read, the vector of values is returned to the caller. Since “NftDetail” is a custom struct, there is no support for the generics required for the vector’s methods, so the “<Self::Api>” generic type has been added to the custom struct, and interface requirements were satisfied. The format of listings data in the listings array returned differs from the single listing data format, as the vector concatenates all the values it has in a single string with separators and applies to encode on the entire string of values making it look different as shown on Figure 38 from the request on Appendix 12.



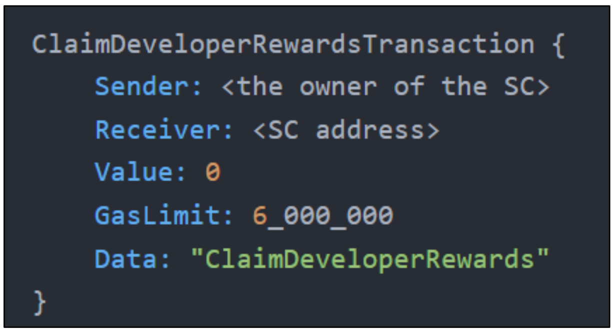
```
"returnData": [
    "AAAADkNPTE5BTUUtNWQ1NzI5AAAAAAAAAAJPBNVIoP1kg54VnSG3WcuGGr+K3s8lPOtqK3lV6u4eMQAAAAgDeC2s6dkAAA==",
    "AAAAClRTVC03MTc5NjgAAAAAAAAAAk8E1Uig/WSDnhWdIbdZy4Yav4rezyU862oreVXq7h4xAAAACAN4Lazp2QAA"
],
```

Figure 38 All listings data

4.8.1 Marketplace governance smart contract endpoints

4.8.1.1 Claim developer rewards

As rewards are built into Elrond’s protocol, and every transaction’s 30% gas fees are returned to the smart contract that generated the transaction, all the rewards are accumulated on the marketplace balance until the smart contract owner claims these. Developer rewards can be claimed by the transaction sent from SC owner’s wallet to SC with the raw payload “ClaimDeveloperRewards”, no encoding to HEX or any other encoding needed as shown on endpoint structure below on Figure 39, the function makes the payment with all available rewards to owner address.



```
ClaimDeveloperRewardsTransaction {
    Sender: <the owner of the SC>
    Receiver: <SC address>
    Value: 0
    GasLimit: 6_000_000
    Data: "ClaimDeveloperRewards"
}
```

Figure 39 “ClaimDeveloperRewards” transaction [22]

The transaction is captured in Appendix 13, transaction data is listed below in Table 9.

Endpoint	
Sender	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Receiver	erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g
input	
Raw data	ClaimDeveloperRewards
output	
Status	success

Table 9 "ClaimDeveloperRewards" transaction

4.9 Issues on Elrond

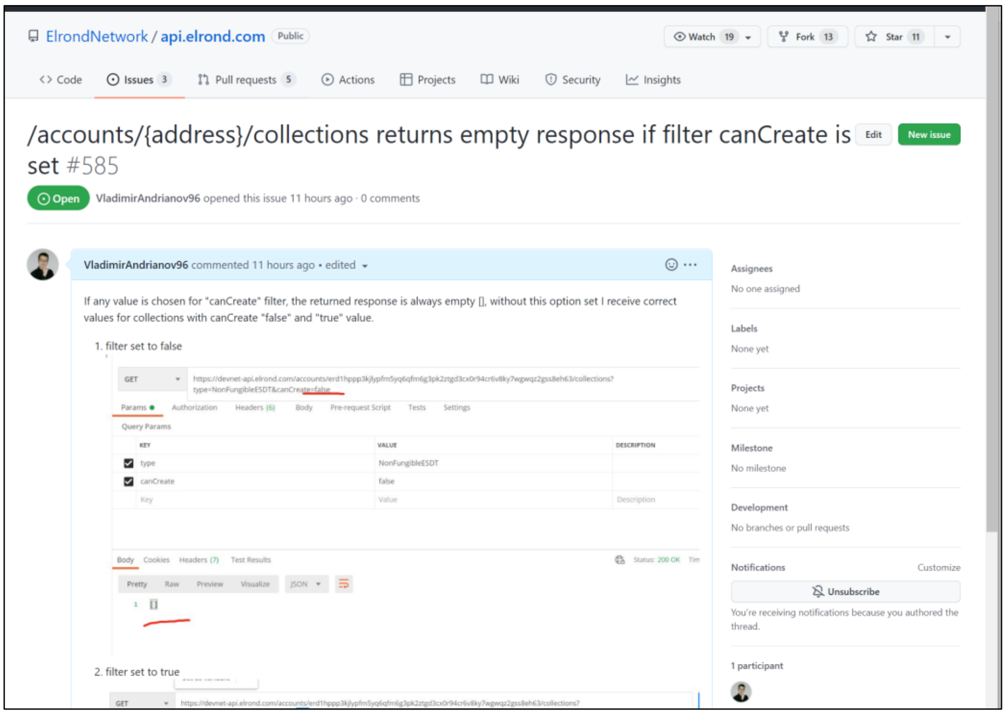
During the development, some problems occurred that blocked the development that affected the API endpoints on mainnet and eDNS data on the blockchain in devnet.

4.9.1 Herotags missing on devnet

After the devnet, the development network got wiped, the data of the smart contracts deployed there was missing, and the herotags functionality also was broken, meaning the marketplace staging environment running on devnet did not show the herotags, the issue existed for a month on devnet so the part of marketplace functionality, e.g., herotags, were switched to the mainnet, meaning the application is using data from mainnet and devnet, but there is no other way around to get the actual data, and the mainnet is always the source of truth for herotags.

4.9.2 Broken collections method on mainnet API

The method used to fetch the creatable and non-creatable collections used the filtering argument in the request, the live mainnet API started providing always empty response if any filtering option was set, so the issue was introduced to the Elrond GitHub api.elrond.com repository as shown below on Figure 40.



/accounts/{address}/collections returns empty response if filter canCreate is set #585

Open VladimirAndrianov96 opened this issue 11 hours ago · 0 comments

VladimirAndrianov96 commented 11 hours ago • edited

If any value is chosen for "canCreate" filter, the returned response is always empty [], without this option set I receive correct values for collections with canCreate "false" and "true" value.

1. filter set to false

```
GET https://devnet-api.elrond.com/accounts/erd1hppp3kjlypfm5yq6qfm6g3pk2ztgd3cx0r94cr6v8ky7wgwqz2gss8eh63/collections?type=NonFungibleESDT&canCreate=false
```

Query Params

KEY	VALUE	DESCRIPTION
✓ type	NonFungibleESDT	
✓ canCreate	false	
Key	Value	Description

Status: 200 OK

```
1 []
```

2. filter set to true

```
GET https://devnet-api.elrond.com/accounts/erd1hppp3kjlypfm5yq6qfm6g3pk2ztgd3cx0r94cr6v8ky7wgwqz2gss8eh63/collections?
```

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

Notifications
Customize
Unsubscribe
You're receiving notifications because you authored the thread.

1 participant

Figure 40 Collections API method issue in GitHub

5 Results

5.1 Smart contract processing costs

All of the transactions with blockchain are listed in Table 10 below, showing how expensive the transaction fees are on the marketplace, taking into account the value as Elrond takes 0.5 EGLD to register a collection, all transactions cost between 0.1-0.2 US dollars, which is distinctly cheaper compared to two years lowest 1.04 USD fee per average transaction on Bitcoin by April 17, 2022, and peak 62.79 USD at April 21, 2021 [23]. The view functions have no transaction fee or any gas needed, as these represent the storage details from the smart contract and are executed in the WASM machine on the client-side thus, no fees are applied to them.

Action	Transaction fees	EGLD price
Create collection	Value: 0.05 EGLD ($7.15) Fee: 0.000808 EGLD ($0.1156)	$143.08
Set special role	Value: 0 Fee: 0.0007555 EGLD ($0.1081)	$143.08
Create NFT	Value: 0 Fee: 0.0003791 EGLD ($0.0542)	$143.08
List NFT	Value: 0 Fee: 0.00046084126 EGLD ($0.0680)	$147.54
Buy NFT	Value: #NFT_PRICE_AMOUNT# Fee: 0.00014738404 EGLD ($0.0217)	$147.56
Update listing price	Value: 0 Fee: 0.00016709223 EGLD ($0.0247)	$147.73
Cancel listing	Value: 0 Fee: 0.00015358636 EGLD ($0.0227)	$147.53
Get listing details	none	none
Get all listings	none	none

Table 10 Transactions fees and values

5.2 Marketplace maintenance

The list of registered users is stored within the database cluster on the cloud, any registered user is kept there along with the user details. The data from the database is used to return the profile settings, while it is not only limited to it, it can also store the session details, legal information (KYC check), and status of the user to mark banned users as banned and build the logic around them or any whitelisted users with special privileges.

The user database will be used for the governance model, which will be developed outside of the thesis MVP scope, this will include governance endpoints for the smart contract to allow marketplace administrators to remove any listing, remove any NFT from the balance, withdraw EGLD balance or any ESDT token such as MEX, whitelist and blacklist can be implemented in the future versions of the smart contract as well to avoid any vulnerable logic on the back-end services.

The regular maintenance must include the cloud storage cleanup routines to reduce the costs for the cloud storage, this can be applied for any banned or inactive users, and the database can be cleaned up too to archive inactive users' details and save a bit on the monthly database bills.

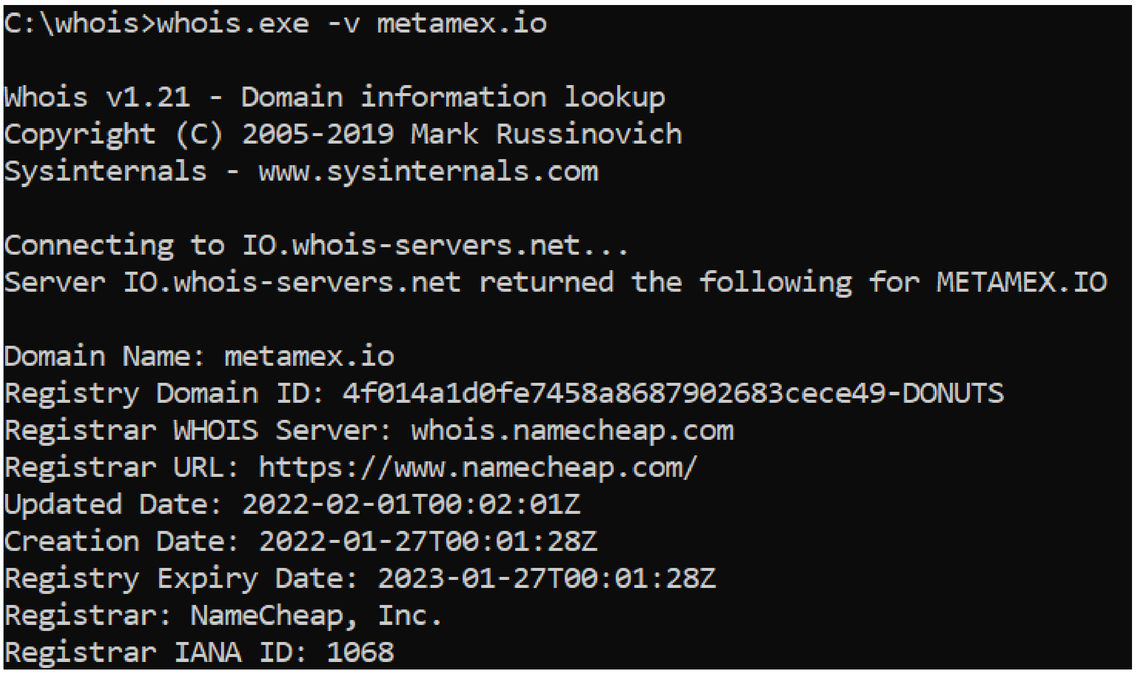
5.3 Infrastructure scaling

The cloud-based infrastructure allows easy vertical and horizontal scaling, e.g., the database is run within the managed database service, which is a cluster running on the cloud which can adjust and scale based on the performance needs, the data from instances can be replicated, and more computing resources can be added by simply increasing the number of stateless back-end services instances running in the cloud. The Apps service also allows the team to select instance type and a number of containers to be running in the cloud, and the cloud Space storage solution can cache files in the CDN network to save traffic and reduce infrastructure expenses.

The blockchain will handle the smart contract storage and process the transactions, so this is out of the marketplace's responsibility, the network's performance will grow with the increasing number of nodes and audiences coming to the Elrond network.

5.4 Development start and related expenses

The idea of the marketplace development was born in the summer of 2021, the active development of the marketplace project began later at the end of the year, the earlier versions were based on Elrond libraries that used the dApp template provided by the Elrond team, but this carried much redundant dependencies and functionality was based on the react-router that is conflicting with the Next.js router, so the application was rebuilt with Next.js framework and Elrond's functionality refactored for Next.js usage in @superciety/pwa-core-library library. The first commit to the Next.js version of the marketplace repository is dated by 12 December 2021, and the metamex.io domain was registered on 26 January 2022, as specified in the registration records in Figure 41 below.



```
C:\whois>whois.exe -v metamex.io

Whois v1.21 - Domain information lookup
Copyright (C) 2005-2019 Mark Russinovich
Sysinternals - www.sysinternals.com

Connecting to IO.whois-servers.net...
Server IO.whois-servers.net returned the following for METAMEX.IO

Domain Name: metamex.io
Registry Domain ID: 4f014a1d0fe7458a8687902683cece49-DONUTS
Registrar WHOIS Server: whois.namecheap.com
Registrar URL: https://www.namecheap.com/
Updated Date: 2022-02-01T00:02:01Z
Creation Date: 2022-01-27T00:01:28Z
Registry Expiry Date: 2023-01-27T00:01:28Z
Registrar: NameCheap, Inc.
Registrar IANA ID: 1068
```

Figure 41 Domain registration records

Team core developers did the main work, so this was free of charge for the community, but there were expenses for outsourcing and consulting services from developers hired primarily on Upwork. Also, the licenses were bought for software such as GitKraken for easier git management, WebStorm IDE for front-end development, and the cloud infrastructure cost real money. The Upwork developers were hired to consult the team on the structure of the smart contracts for Elrond as at the beginning of development, there was no blockchain development expertise within the project team, so every knowledge was acquired within the active development time from Elrond documentation and Elrond's live support in Telegram answering the key questions about Elrond architecture.

5.5 Platform UI

The platform UI built on the Next.js framework is the key component which unites customers with the back-end and all infrastructure components united, so the visual part was a massive part of the dApp, meaning there was much time spent on the development of the designs with the colour scheme, fonts and logo. Many features included in the designs are outside of the thesis scope, such as collections and custom collection drops functionality, the available at the time of writing parts of the marketplace are captured and brought in the form of screenshots to the appendixes part.

The deployed application is not fully available on the internet, it’s locked behind the password form, as captured in Appendix 14. When the valid password is provided, the request is sent to the serverless function of the Next.js application, which validates the password and provides a session cookie to access the marketplace, if the password is changed or the cookie was lost, the password form will be opened again on any of the routes and sub-routes accessed in the browser as web application validates cookie using the secret key to compare hashes.

The main page from Appendix 15 opens a view on the trending collections and also the navigation pane and sliding side navigation bar, the unauthorized customers will see the side pane showing four authentication methods from Appendix 16; Appendix 17 is a Maiar App authentication method that shows up a QR code to be scanned through a mobile app, and Maiar DeFi wallet opens the Maiar Wallet web browser extension as captured on Appendix 18. When the customer has logged in, the side panel show his session details as on Appendix 19, such as herotag, profile picture, available EGLD and MEX balance, the total balance in USD and the option to log out from the system.

Navigation between pages is available in the navigation pane on top of the screen, it has “Create” options, as shown in Appendix 20, which allow to manage collections and create NFTs on the network, the “Trade” dropdown from Appendix 21 opens the option to get into the collections overview page and the listed NFTs for sale overview, Discover will lead to collection drops, and profile dropdown from Appendix 22 leads to the profile page and profile settings to set up the profile space which will show up on the personalized profile page. The “Create” dropdown provides access to the “Create collections” screen from Appendix 23 that allows a customer to register a collection on the network and set properties to in on the creation step, the next step is the “Activate collections” screen from Appendix 24 that lets customer set up the “ESDTNFTCreate” role for the collection so that the collection will be ready to create NFTs and will be available at “Update the Metaverse” dropdown from Appendix 25, to be selected to proceed to the “Create NFT” page from Appendix 26, the NFTs will be created in the customer’s wallet and will be available for listing on the marketplace, the NFTs will appear under the profile page NFT section.

The not fully implemented “explore collections” page from Appendix 27 will be used to list the trending collections on the marketplace once it is finished, and the “Listings” screen from Appendix 28 is a grid component that lists all available NFT listings on the marketplace, any NFT can be bought there when corresponding “Buy” button is triggered, more discovery options are available at preliminary mock of Discovery page from Appendix 29, this will list all collections available for minting or drops, and the corresponding collection details will be listed on the “mint collection” screen from Appendix 30 or page with NFTs belonging to the particular collection as shown on Appendix 31. The NFTs that have been chosen under the “Listings” or “Collection details” page will open an NFT detail screen from Appendix 32, which will show the integrated media content as image or video or music, along with NFT history, attributes, rarity and price, with the auction and “make offer” features in the future.

The user profiles from the Profile page, such as Gokai Labs profile from Appendix 33 and Elrond World from Appendix 34, represent the personalized userspace, where the profile page, banner, and collections tab from Appendix 35 and profile bio from Appendix 36 are fully customized by the customer, these options have default fallback values. The user is not forced to set up any of these, it is voluntary for the marketplace customers, so they can always set it up later when they feel the need for it. The owner of the profile, the wallet owner, sees his profile as different from other users on the marketplace, profile owner has special options for NFTs in his space so that only the owner can list NFTs and see these buttons as “List for sale” button from Appendix 37, this button will trigger a modal form to appear, where customer can set the price for NFT and list it, in the future development there will be a tab to see listed NFTs and manage these separately, to update the price or cancel the listing. The profile page loads details set in the “Settings” page from Appendix 39, it stores fields data set in fields in back-end service, which processes this data and serves it back to the profile visitors.

6 Summary

The platform development was long exhausting, there were many bugs and unexpected changes to functionality or documentation on Elrond, such as a bug with API the issue was created for and other issues, e.g., missing herotags on devnet, there were a lot of blind walkthroughs before the goal was achieved. The web app was built on the Next.js framework that allows server-side rendering and serverless functions to ease the development and save time from implementing everything on the back-end, including the password protection for the deployed marketplace instance available on the internet. Cloud infrastructure simplified the database management for back-end services serving the profile details and data for CMS and also enabled the easy automatic deployment of every component to the cloud. The smart contract built for the marketplace turned out to be a very robust and transparent solution, and deployment was robust and prevented any failed builds from passing to live deployment, resulting in the absence of downtime during the development phase. TypeScript has prevented type issues and preserved the uptime for the marketplace web platform too. A lot of development and architectural knowledge was acquired from the development, and there was conducted deep research into smart contract structure with different mappers and Rust language basics to optimize the code.

The end solution has achieved the full MVP scope for the thesis, this is a platform which is available to any Elrond wallet owner on the internet, where customers can log in through every available wallet integration, such as Elrond Wallet integration, Maiar App with QR code scanner, Maiar DeFi Wallet browser extension or the ledger solution and customers are free and not obligated to personalize their profile pages and share it, and there is always an option to remove every detail from the profile. The marketplace is integrated with the Elrond network, so the actual data is fetched directly from the network into the marketplace and is extended with data from the custom smart contract, which was specially designed for the marketplace needs. The platform facilitates the sale processes and can be used to manage collections to create NFTs, show available listings, update listings, cancel listings, and process the sales with payments.

The main problem of blockchains was the inefficiency and high costs of operations, which eventually can make transaction fees too high and sometimes more than the value of the purchase itself. The MetaMex marketplace proves it can be used as a solution to this problem, as it allows list NFT just in less than 7 cents, or buy, unlist and update price for less than 3 US dollar cents, which makes it affordable to most the world population, of course, the EGLD price on the market will vary and the actual fees might go higher or lower, the more the network will grow the more computing resources will be available to process transactions. The actual scope set for the MetaMex marketplace is larger than the thesis MVP scope, there are designs prepared for the collections management, such as marketing campaigns for the drop and mint campaigns, the visuals of the future planning were added to the work and will be worked on, these listed features can be redesigned completely, and visuals will change depending on the future development.

References

1. Now is the time for a 'great reset'. Schwab, K. (03.06.2020) [WWW] https://www.weforum.org/agenda/2020/06/now-is-the-time-for-a-great-reset/ (12.12.2021).

2. OpenSea’s NFT trade quadruples to 10 billion in under three months but Axie Infinity continues to be the crowd favourite. Gill, P. (09.11.2021) [WWW] https://www.businessinsider.in/investment/news/openseas-nft-trade-quadruples-to-10-billion-in-under-three-months-but-axie-infinity-continues-to-be-the-crowd-favourite/articleshow/87609137.cms (14.12.2021).

3. Beeple | The First 5000 Days. CHRSTIE’S. (11.03.2021) [WWW] https://onlineonly.christies.com/s/beeple-first-5000-days/lots/2020 (14.12.2021) .

4. Blockchain Trilemma. Grodzicka, H. (22.07.2020) [WWW] https://blog.fingo.pl/blockchain-trilemma/ (14.12.2021).

5. An Order-of-Magnitude Estimate of the Relative Sustainability of the Bitcoin Network, 3nd ed., Working Paper. McCook, H. (11.02.2015) [WWW] https://www.academia.edu/7666373/_OBSOLETE_An_Order_of_Magnitude_Estimate_of_the_Relative_Sustainability_of_the_Bitcoin_Network (20.12.2021).

6. Elrond Staking: Beginners’ Guide. Arsov, A. (22.12.2021) [WWW] https://egg.fi/crypto-academy/staking/elrond-staking-beginners-guide/ (26.12.2021).

7. Secure Proof of Stake. Elrond. [WWW] https://docs.elrond.com/technology/secure-proof-of-stake/ (30.12.2021).

8. Maiar Exchange - Testnet functionalities practicing tutorial. MGStaking. (25.08.2021) [WWW] https://medium.com/mgstaking/maiar-exchange-devnet-functionalities-practicing-tutorial-ae05efee5bed (05.01.2022).

9. Elrond Staking Phase 2 - The Validators Queue Launches On December 1st. Mincu, L. (16.10.2020) [WWW] https://elrond.com/blog/staking-phase-2/ (05.01.2022).

10. System Requirements. Elrond. [WWW] https://docs.elrond.com/validators/system-requirements/ (08.01.2022).

11. Frequently Asked Questions. Elrond. [WWW] https://elrond.com/faq/ (08.01.2022).

12. Elrond Becomes The First Carbon Negative European Blockchain, Opening A New Wave Of Sustainable Innovation In Line With The European Climate Policy. Mincu, B. (05.08.2021) [WWW] https://elrond.com/blog/elrond-carbon-negative-offsetra/ (08.01.2022).

13. NFT trading volume hit $10.7 billion last quarter—here are 2 reasons why people are spending thousands on digital assets. Locke, T. (06.10.2021) [WWW] https://www.cnbc.com/2021/10/06/nft-trading-volume-hit-10-billion-2-reasons-why-people-are-buying.html (08.01.2022).

14. Which blockchains does OpenSea support? OpenSea. [WWW] https://support.opensea.io/hc/en-us/articles/4404027708051-Which-blockchains-does-OpenSea-support (19.01.2022).

15. eMoon is closing its doors. eMoon. [WWW] https://emoon.space/ (27.02.2022).

16. How to Git a Branch: Work Safety. Rolie. (05.08.2021) [WWW] https://medium.com/codex/how-to-git-a-branch-work-safety-421a8320b93c (30.02.2022).

17. Get started with Tailwind CSS. Tailwind CSS. [WWW] https://tailwindcss.com/docs/installation (30.02.2022).

18. The Next.js Handbook. Copes, F. (19.11.2019) [WWW] https://www.freecodecamp.org/news/the-next-js-handbook/ (30.02.2022).

19. The Elrond Virtual Machine - Smart Contracts At Internet Scale. Mincu, B. (04.12.2020) [WWW] https://elrond.com/blog/elrond-virtual-machine-arwen-wasm-vm/ (05.03.2022).

20. NFT tokens. Elrond. [WWW] https://docs.elrond.com/developers/nft-tokens/ (05.04.2022).

21. Storage Mappers. Elrond. [WWW] https://docs.elrond.com/developers/best-practices/storage-mappers/ (10.04.2022).

22. Built-In Functions. Elrond [WWW] https://docs.elrond.com/developers/built-in-functions/ (12.04.2022)

23. Bitcoin average transaction fees lowest in two years at $1.04. Sarkar, A. (18.04.2022) [WWW] https://cointelegraph.com/news/bitcoin-average-transaction-fees-lowest-in-two-years-at-1-04 (28.04.2022)

Acknowledgements

Throughout the writing of this thesis, I have received a great deal of support and assistance from the core team members, I would first like to thank the Elrond World (Twitter - @ElrondWorld), which is a global blockchain community that is specifically orientated around the Elrond Network blockchain ecosystem and Gokai Labs (Twitter - @GokaiLabs) which is a community-driven collectible NFT DAO on Elrond Network highly supported and facilitated the research and prototyping of my work on this thesis.

I would also like to thank Yi Yang for any support and evaluation of the thesis and thank the company I work for, the Dialog exe AS, also known as DX, for providing me with every support and freedom to work on the thesis. DX is an innovative company that invests a lot in the development of new technologies and highly supports employee development.

In addition, I would like to thank Admiral Markets and XRP Ledger for paying interest in this work and supporting students in coordination with Taltech and for their willingness to support me.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Vladimir Andrianov

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “NFT marketplace platform development on Elrond blockchain”, supervised by Avar Pentel
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

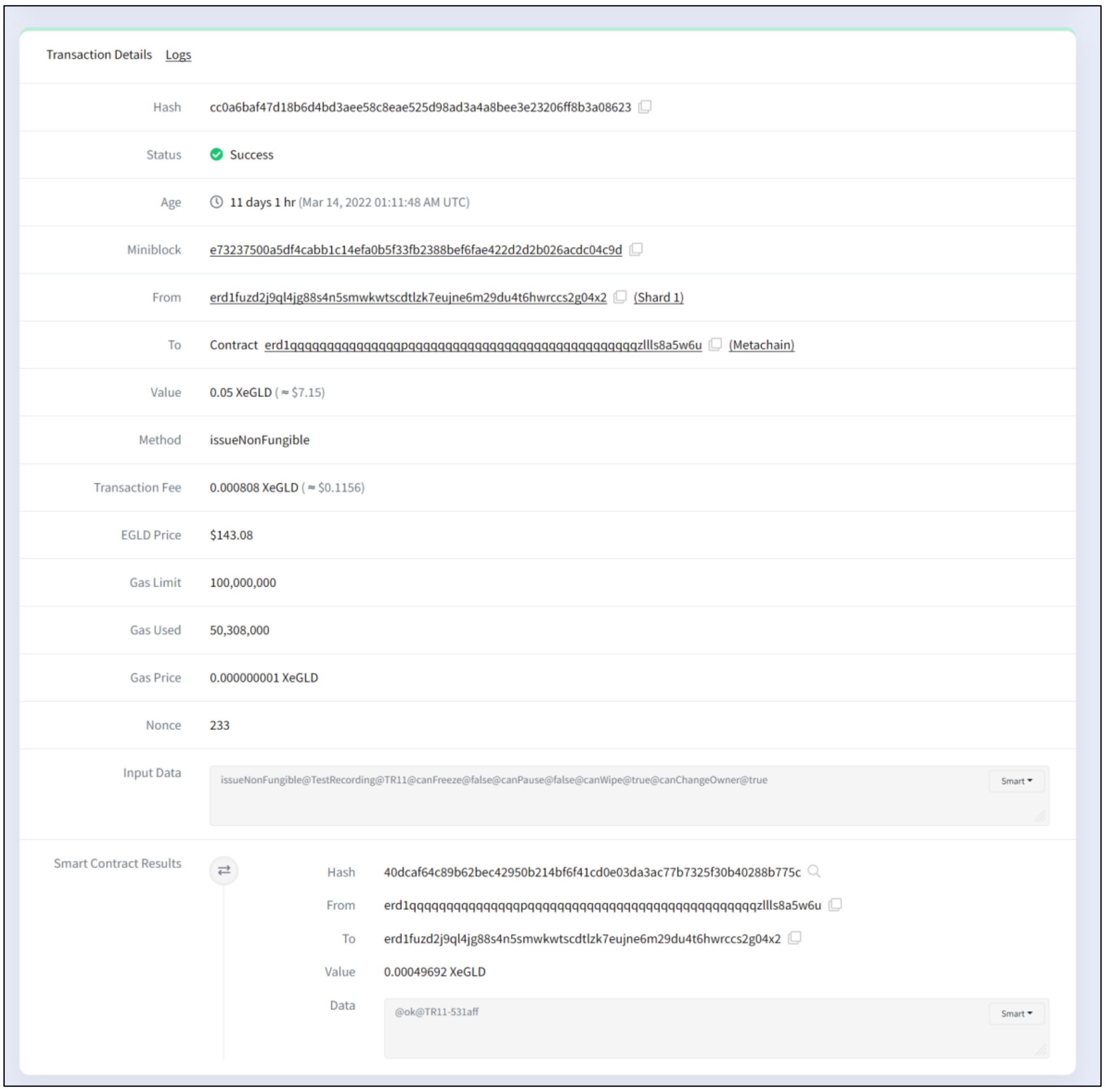
28.04.2022

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – Docker-compose file for container with PostgreSQL and PgAdmin

```
version:
'3'
services:
 # Use healthcheck to let back-end know when the db is up.
  postgres:
    image: postgres:alpine
    container_name: #container_name#
    environment:
      - POSTGRES_USER=#username#
      - POSTGRES_PASSWORD=#password#
      - POSTGRES_DB=#db_name#
    ports:
      - "6500:5432" #remap ports to avoid clashing.
    volumes:
      - postgres:/var/lib/postgres/data
    restart: unless-stopped
    networks:
      - db_network
    healthcheck:
      test: "exit 0"
  pgadmin:
    image: dpage/pgadmin4
    container_name: #container_name#
    environment:
      PGADMIN_DEFAULT_EMAIL: #email#
      PGADMIN_DEFAULT_PASSWORD: #password#
    depends_on:
      - postgres
    ports:
      - "5050:80" #remap ports to avoid clashing.
    restart: unless-stopped
    networks:
      - monorepo_network
volumes:
  postgres:
# Networks to be created to facilitate communication between containers
networks:
  db_network:
    driver: bridge
```

Appendix 3 – “issueNonFungible” transaction



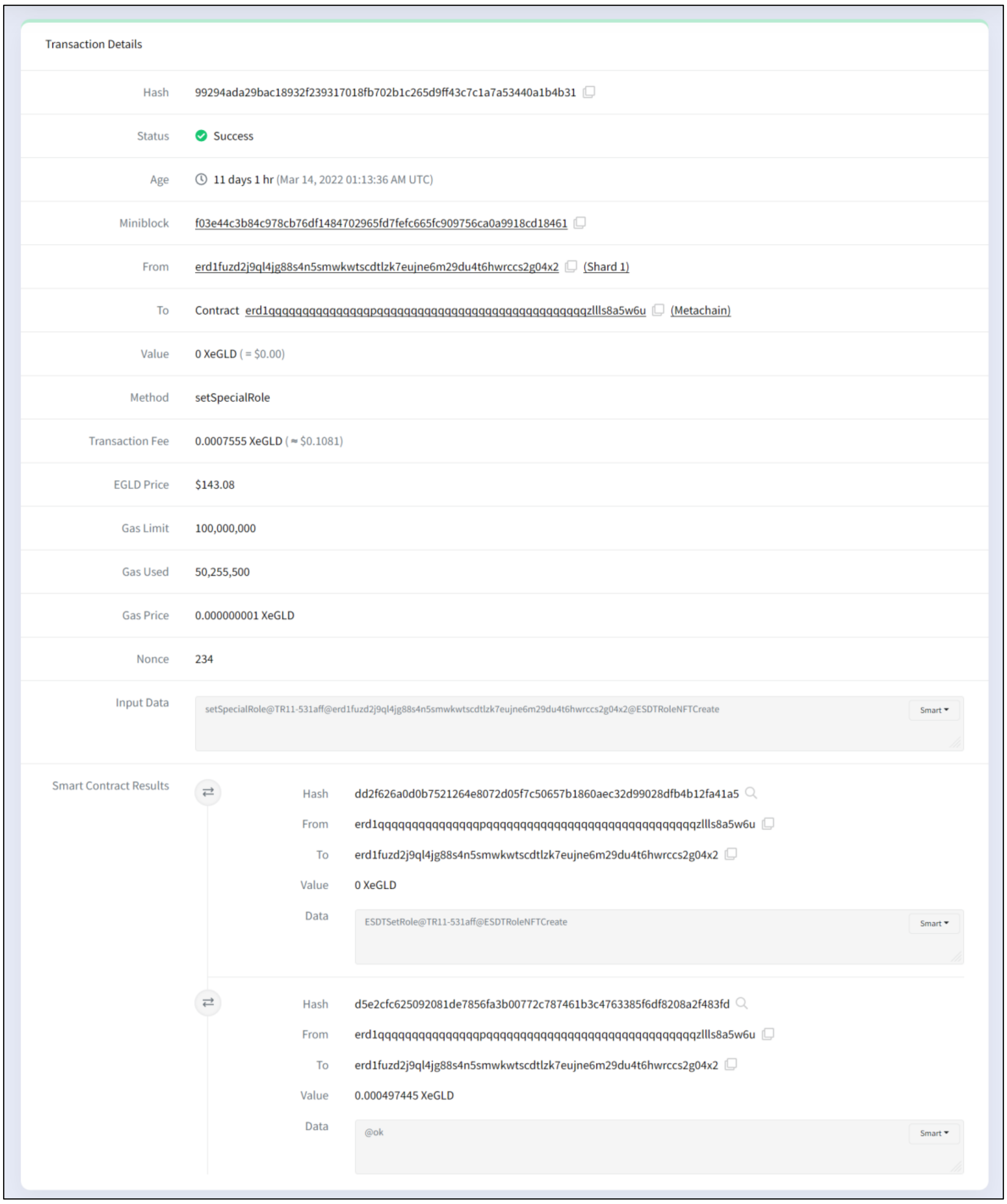
Transaction Details Logs

Hash	cc0a6baf47d18b6d4bd3aee58c8eae525d98ad3a4a8bee3e23206ff8b3a08623
Status	Success
Age	11 days 1 hr (Mar 14, 2022 01:11:48 AM UTC)
Miniblock	e73237500a5df4cabb1c14efa0b5f33fb2388bef6fae422d2d2b026acdc04c9d
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	Contract erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u (Metachain)
Value	0.05 XeGLD (≈ $7.15)
Method	issueNonFungible
Transaction Fee	0.000808 XeGLD (≈ $0.1156)
EGLD Price	$143.08
Gas Limit	100,000,000
Gas Used	50,308,000
Gas Price	0.000000001 XeGLD
Nonce	233
Input Data	issueNonFungible@TestRecording@TR11@canFreeze@false@canPause@false@canWipe@true@canChangeOwner@true

Smart Contract Results

Hash	40dcaf64c89b62bec42950b214bf6f41cd0e03da3ac77b7325f30b40288b775c
From	erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.00049692 XeGLD
Data	@ok@TR11-531aff

Appendix 4 – “setSpecialRole” transaction



Transaction Details

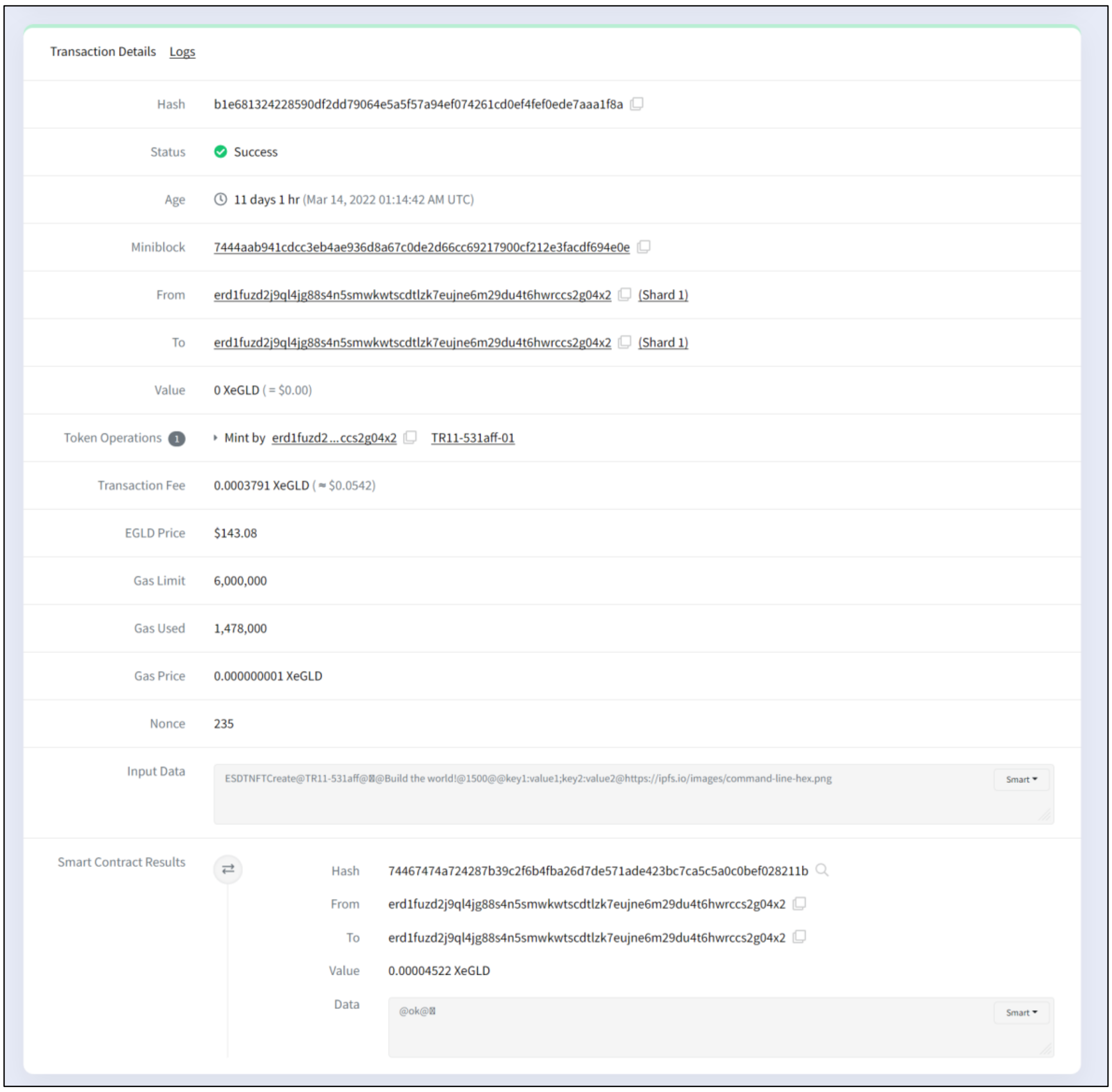
Hash	99294ada29bac18932f239317018fb702b1c265d9ff43c7c1a7a53440a1b4b31
Status	✓ Success
Age	11 days 1 hr (Mar 14, 2022 01:13:36 AM UTC)
Miniblock	f03e44c3b84c978cb76df1484702965fd7fefc665fc909756ca0a9918cd18461
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	Contract erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u (Metachain)
Value	0 XeGLD (= $0.00)
Method	setSpecialRole
Transaction Fee	0.0007555 XeGLD (≈ $0.1081)
EGLD Price	$143.08
Gas Limit	100,000,000
Gas Used	50,255,500
Gas Price	0.000000001 XeGLD
Nonce	234
Input Data	setSpecialRole@TR11-531aff@erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2@ESDTRoleNFTCreate

Smart Contract Results

Hash	dd2f626a0d0b7521264e8072d05f7c50657b1860aec32d99028dfb4b12fa41a5
From	erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0 XeGLD
Data	ESDTSetRole@TR11-531aff@ESDTRoleNFTCreate

Hash	d5e2cfc625092081de7856fa3b00772c787461b3c4763385f6df8208a2f483fd
From	erd1qqqqqqqqqqqqqqqpqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqzllls8a5w6u
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.000497445 XeGLD
Data	@ok

Appendix 5 – “ESDTNFTCreate” transaction



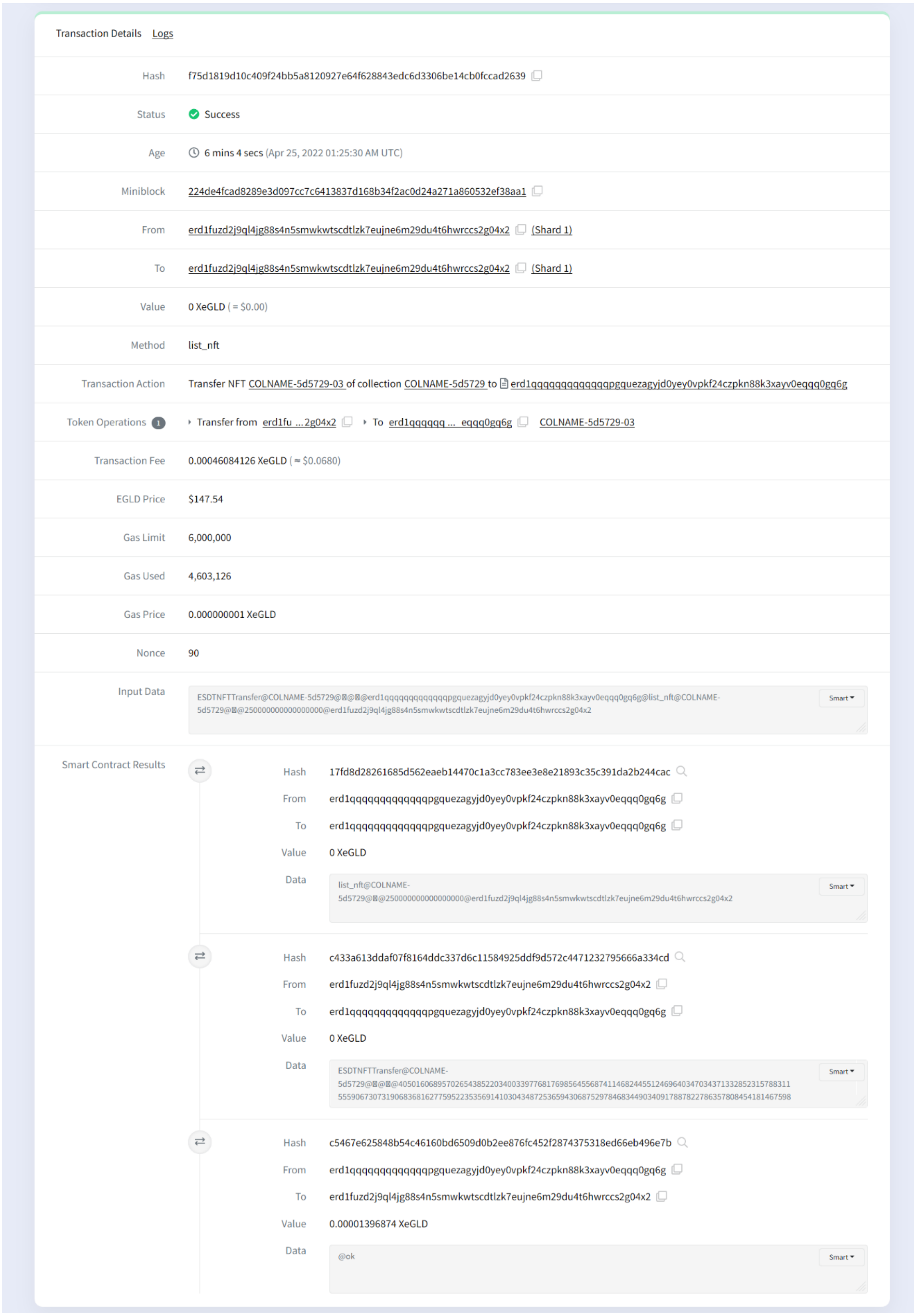
Transaction Details Logs

Hash	b1e681324228590df2dd79064e5a5f57a94ef074261cd0ef4fef0ede7aaa1f8a
Status	Success
Age	11 days 1 hr (Mar 14, 2022 01:14:42 AM UTC)
Miniblock	7444aab941cdcc3eb4ae936d8a67c0de2d66cc69217900cf212e3facdf694e0e
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
Value	0 XeGLD (= $0.00)
Token Operations 1	Mint by erd1fuzd2...ccs2g04x2 TR11-531aff-01
Transaction Fee	0.0003791 XeGLD (≈ $0.0542)
EGLD Price	$143.08
Gas Limit	6,000,000
Gas Used	1,478,000
Gas Price	0.000000001 XeGLD
Nonce	235
Input Data	ESDTNFTCreate@TR11-531aff@⊠@Build the world!@1500@@key1:value1;key2:value2@https://ipfs.io/images/command-line-hex.png

Smart Contract Results

Hash	74467474a724287b39c2f6b4fba26d7de571ade423bc7ca5c5a0c0bef028211b
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.00004522 XeGLD
Data	@ok@⊠

Appendix 6 – “ESDTNFTTransfer” with nested “list_nft_for_sale” transaction



Transaction Details   Logs

Hash	f75d1819d10c409f24bb5a8120927e64f628843edc6d3306be14cb0fccad2639
Status	Success
Age	6 mins 4 secs (Apr 25, 2022 01:25:30 AM UTC)
Miniblock	224de4fcad8289e3d097cc7c6413837d168b34f2ac0d24a271a860532ef38aa1
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
Value	0 XeGLD (= $0.00)
Method	list_nft
Transaction Action	Transfer NFT COLNAME-5d5729-03 of collection COLNAME-5d5729 to erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
Token Operations 1	Transfer from erd1fu ...2g04x2 To erd1qqqqqq ... eqqq0gq6g COLNAME-5d5729-03
Transaction Fee	0.00046084126 XeGLD (≈ $0.0680)
EGLD Price	$147.54
Gas Limit	6,000,000
Gas Used	4,603,126
Gas Price	0.000000001 XeGLD
Nonce	90
Input Data	ESDTNFTTransfer@COLNAME-5d5729@�@�@erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g@list_nft@COLNAME-5d5729@�@250000000000000000@erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2

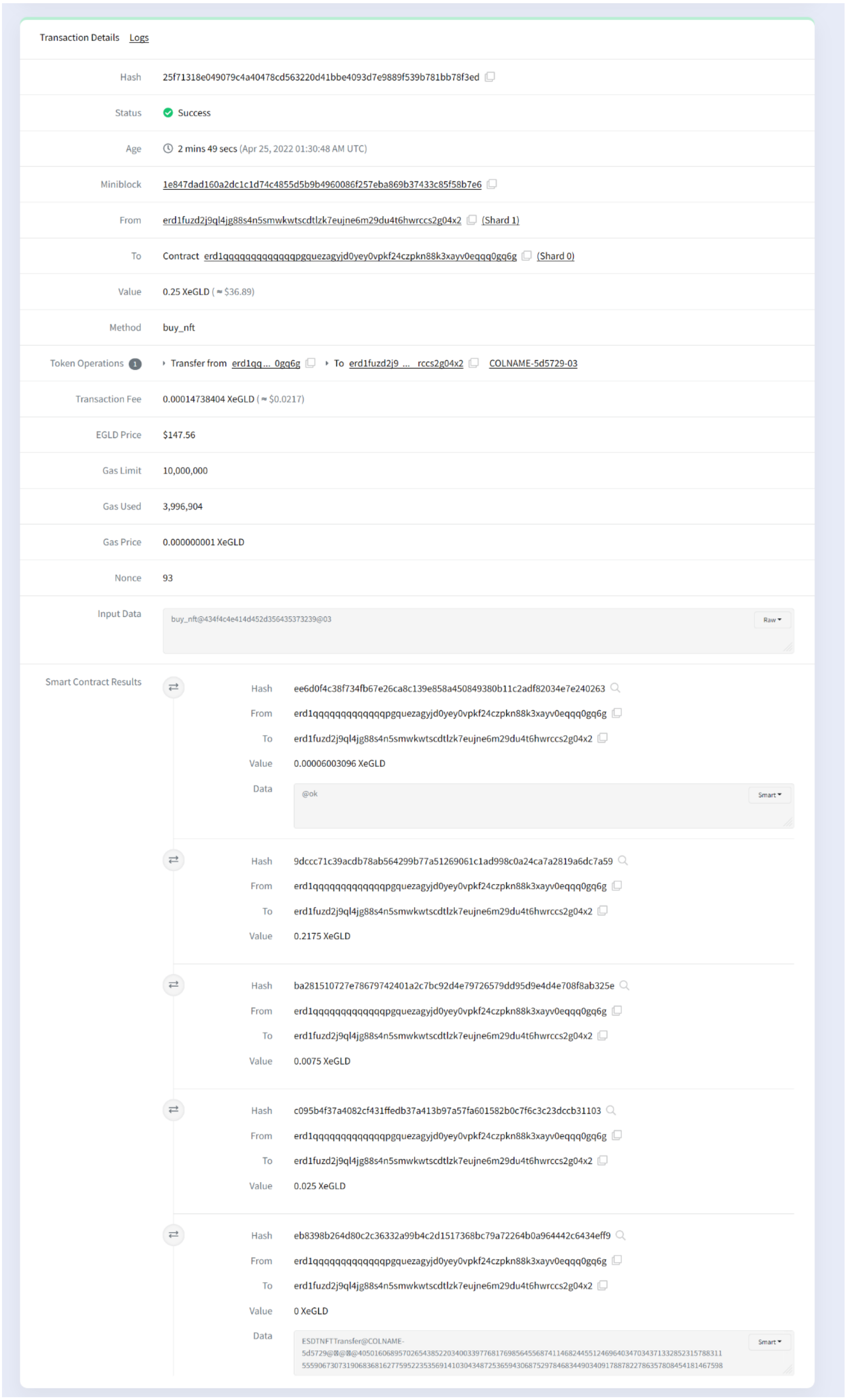
Smart Contract Results

Hash	17fd8d28261685d562eaeb14470c1a3cc783ee3e8e21893c35c391da2b244cac
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
Value	0 XeGLD
Data	list_nft@COLNAME-5d5729@�@250000000000000000@erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2

Hash	c433a613ddaf07f8164ddc337d6c11584925ddf9d572c4471232795666a334cd
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
To	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
Value	0 XeGLD
Data	ESDTNFTTransfer@COLNAME-5d5729@�@�@40501606895702654385220340033977681769856455687411468244551246964034703437133285231578831155590673073190683681627759522353569141030434872536594306875297846834490340917887822786357808454181467598

Hash	c5467e625848b54c46160bd6509d0b2ee876fc452f2874375318ed66eb496e7b
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.00001396874 XeGLD
Data	@ok

Appendix 7 – "buyNFT" transaction



Transaction Details Logs

Hash	25f71318e049079c4a40478cd563220d41bbe4093d7e9889f539b781bb78f3ed
Status	✓ Success
Age	2 mins 49 secs (Apr 25, 2022 01:30:48 AM UTC)
Miniblock	1e847dad160a2dc1c1d74c4855d5b9b4960086f257eba869b37433c85f58b7e6
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	Contract erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g (Shard 0)
Value	0.25 XeGLD (≈ $36.89)
Method	buy_nft
Token Operations 1	Transfer from erd1qq… 0gq6g To erd1fuzd2j9 … rccs2g04x2 COLNAME-5d5729-03
Transaction Fee	0.00014738404 XeGLD (≈ $0.0217)
EGLD Price	$147.56
Gas Limit	10,000,000
Gas Used	3,996,904
Gas Price	0.000000001 XeGLD
Nonce	93
Input Data	buy_nft@434f4c4e414d452d356435373239@03

Smart Contract Results

Hash	ee6d0f4c38f734fb67e26ca8c139e858a450849380b11c2adf82034e7e240263
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.00006003096 XeGLD
Data	@ok

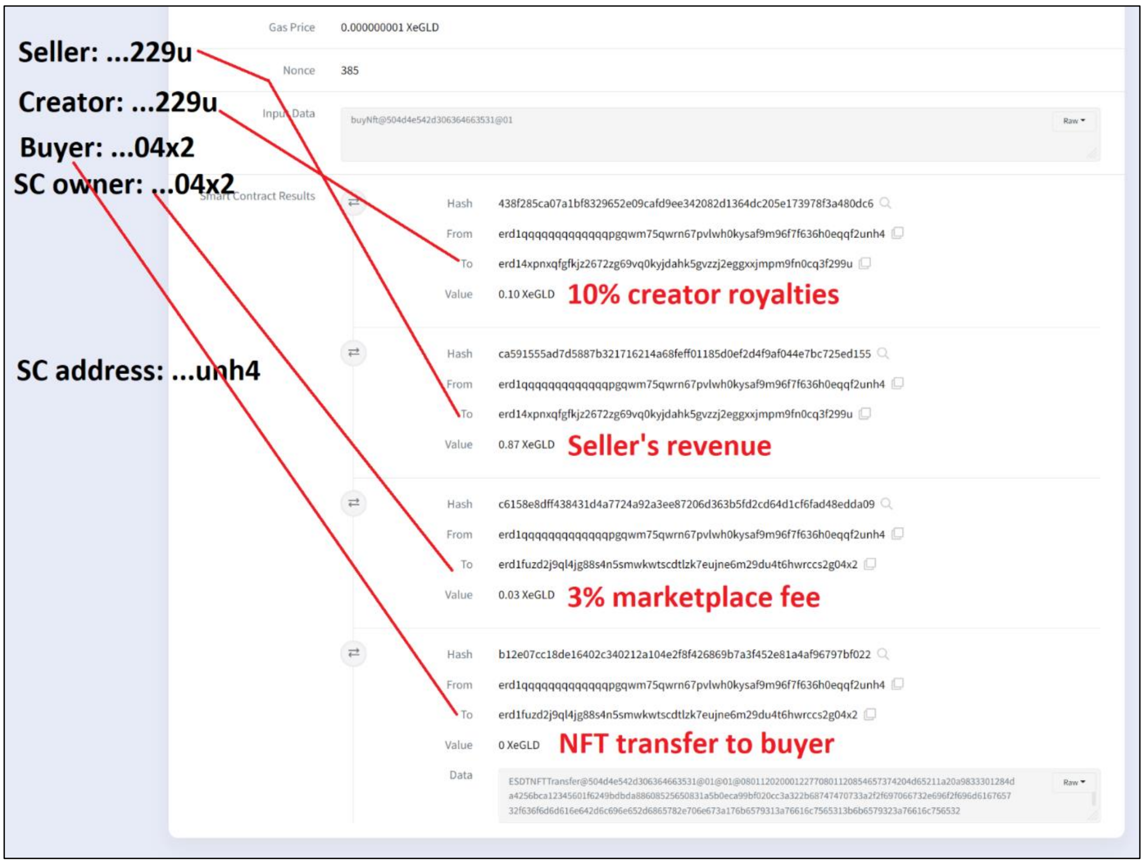
Hash	9dccc71c39acdb78ab564299b77a51269061c1ad998c0a24ca7a2819a6dc7a59
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.2175 XeGLD

Hash	ba281510727e78679742401a2c7bc92d4e79726579dd95d9e4d4e708f8ab325e
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.0075 XeGLD

Hash	c095b4f37a4082cf431ffedb37a413b97a57fa601582b0c7f6c3c23dccb31103
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.025 XeGLD

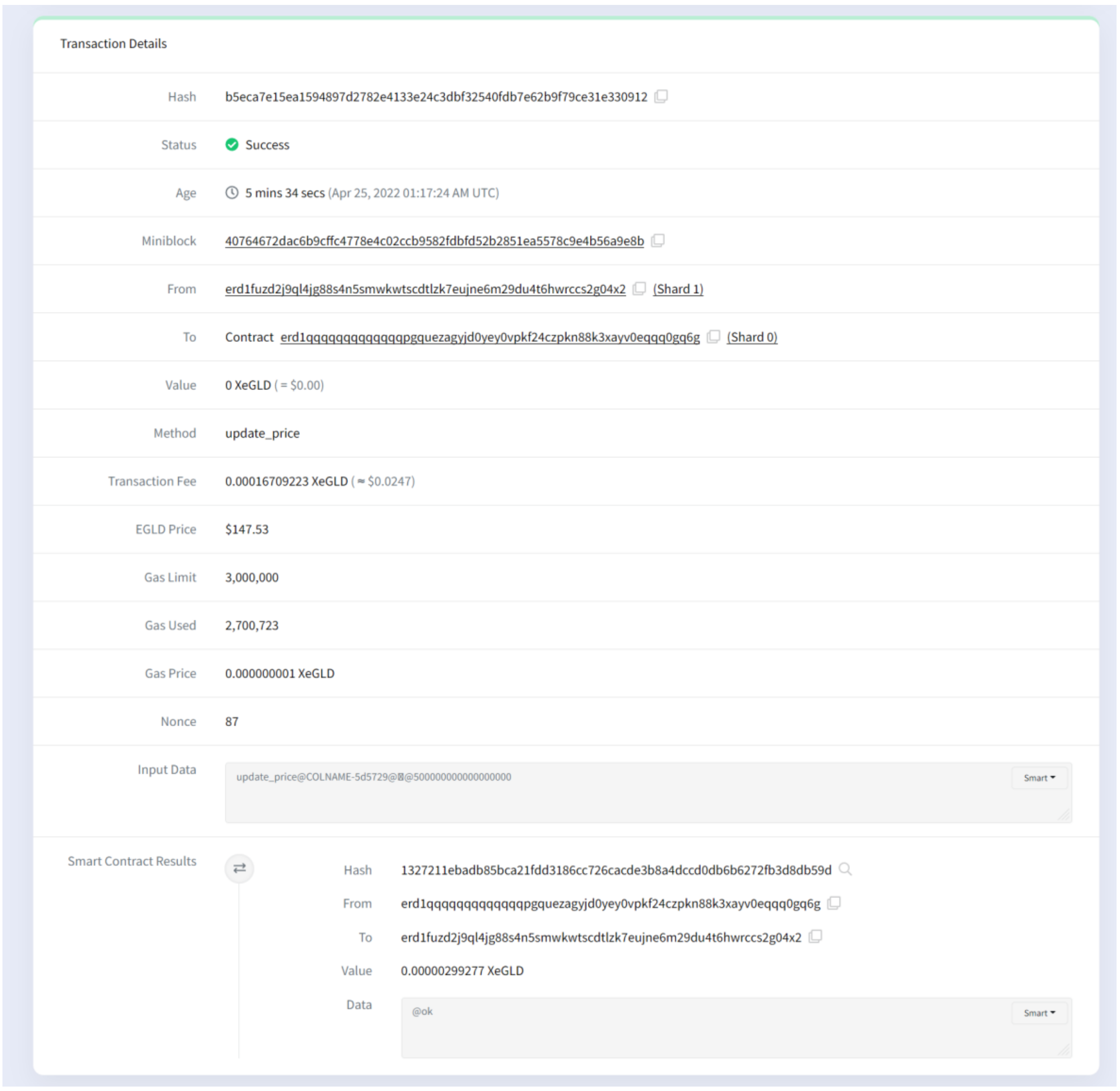
Hash	eb8398b264d80c2c36332a99b4c2d1517368bc79a72264b0a964442c6434eff9
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0 XeGLD
Data	ESDTNFTTransfer@COLNAME-5d5729@▯@▯@40501606895702654385220340033977681769856455687411468244551246964034703437133285231578831155590673073190683681627759522353569141030434872536594306875297846834490340917887822786357808454181467598

Appendix 8 – "buyNFT" transfers



```
Seller: ...229u
Creator: ...229u
Buyer: ...04x2
SC owner: ...04x2
SC address: ...unh4

Gas Price   0.000000001 XeGLD
Nonce       385
Input Data  buyNft@504d4e542d306364663531@01

Smart Contract Results
    Hash   438f285ca07a1bf8329652e09cafd9ee342082d1364dc205e173978f3a480dc6
    From   erd1qqqqqqqqqqqqqpgqwm75qwrn67pvlwh0kysaf9m96f7f636h0eqqf2unh4
    To     erd14xpnxqfgfkjz2672zg69vq0kyjdahk5gvzzj2eggxxjmpm9fn0cq3f299u
    Value  0.10 XeGLD   10% creator royalties

    Hash   ca591555ad7d5887b321716214a68feff01185d0ef2d4f9af044e7bc725ed155
    From   erd1qqqqqqqqqqqqqpgqwm75qwrn67pvlwh0kysaf9m96f7f636h0eqqf2unh4
    To     erd14xpnxqfgfkjz2672zg69vq0kyjdahk5gvzzj2eggxxjmpm9fn0cq3f299u
    Value  0.87 XeGLD   Seller's revenue

    Hash   c6158e8dff438431d4a7724a92a3ee87206d363b5fd2cd64d1cf6fad48edda09
    From   erd1qqqqqqqqqqqqqpgqwm75qwrn67pvlwh0kysaf9m96f7f636h0eqqf2unh4
    To     erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
    Value  0.03 XeGLD   3% marketplace fee

    Hash   b12e07cc18de16402c340212a104e2f8f426869b7a3f452e81a4af96797bf022
    From   erd1qqqqqqqqqqqqqpgqwm75qwrn67pvlwh0kysaf9m96f7f636h0eqqf2unh4
    To     erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
    Value  0 XeGLD   NFT transfer to buyer
    Data   ESDTNFTTransfer@504d4e542d306364663531@01@01@08011202000122770801120854657374204d65211a20a9833301284d
           a4256bca12345601f6249bdbda88608525650831a5b0eca99bf020cc3a322b68747470733a2f2f697066732e696f2f696d6167657
           32f636f6d6d616e642d6c696e652d6865782e706e673a176b6579313a76616c7565313b6b6579323a76616c756532
```

Appendix 9 – UpdateNFT price



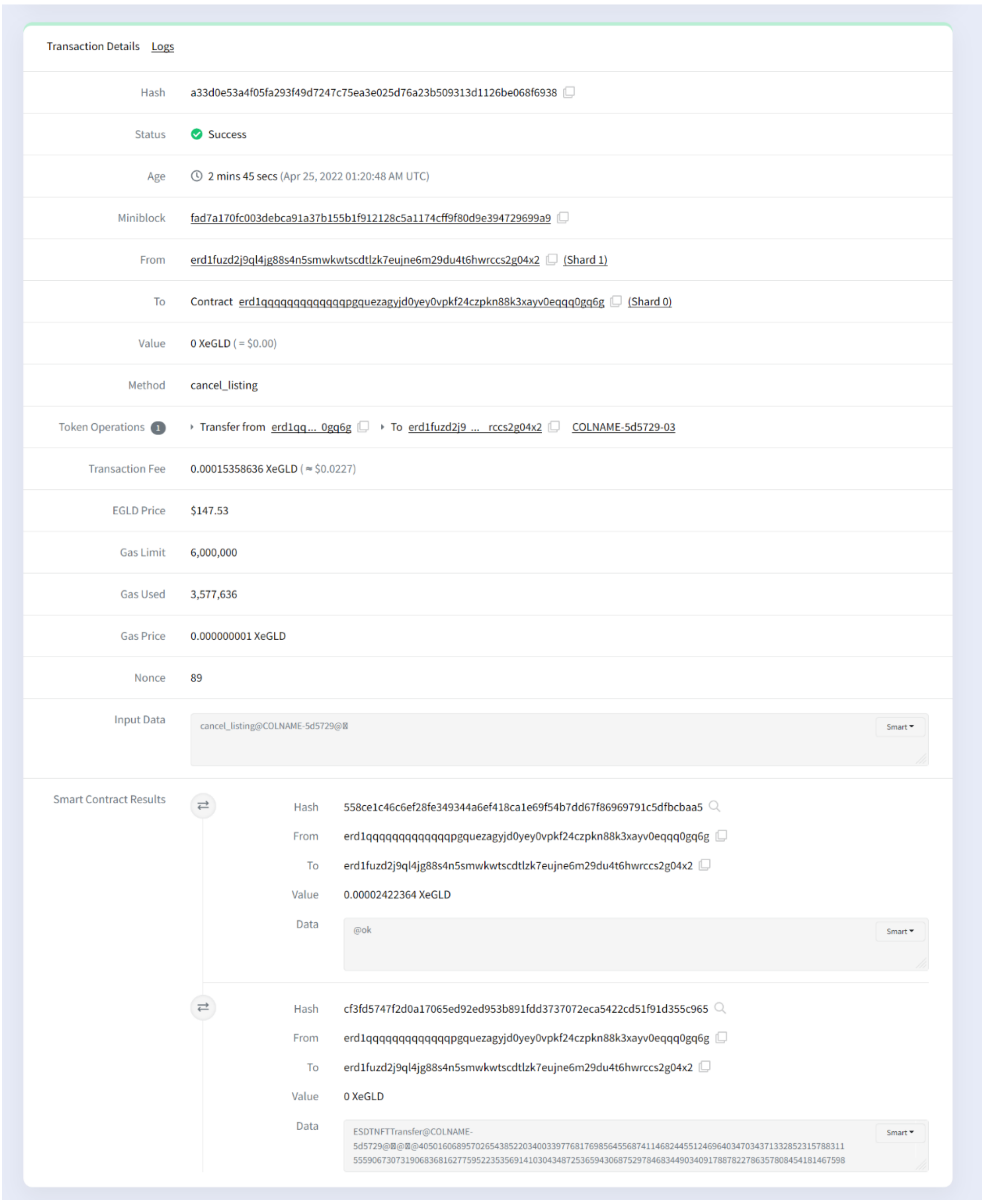
Transaction Details

Hash	b5eca7e15ea1594897d2782e4133e24c3dbf32540fdb7e62b9f79ce31e330912
Status	Success
Age	5 mins 34 secs (Apr 25, 2022 01:17:24 AM UTC)
Miniblock	40764672dac6b9cffc4778e4c02ccb9582fdbfd52b2851ea5578c9e4b56a9e8b
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	Contract erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g (Shard 0)
Value	0 XeGLD (= $0.00)
Method	update_price
Transaction Fee	0.00016709223 XeGLD (≈ $0.0247)
EGLD Price	$147.53
Gas Limit	3,000,000
Gas Used	2,700,723
Gas Price	0.000000001 XeGLD
Nonce	87
Input Data	update_price@COLNAME-5d5729@@500000000000000000

Smart Contract Results

Hash	1327211ebadb85bca21fdd3186cc726cacde3b8a4dccd0db6b6272fb3d8db59d
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.00000299277 XeGLD
Data	@ok

Appendix 10 – Cancel listing



Transaction Details Logs

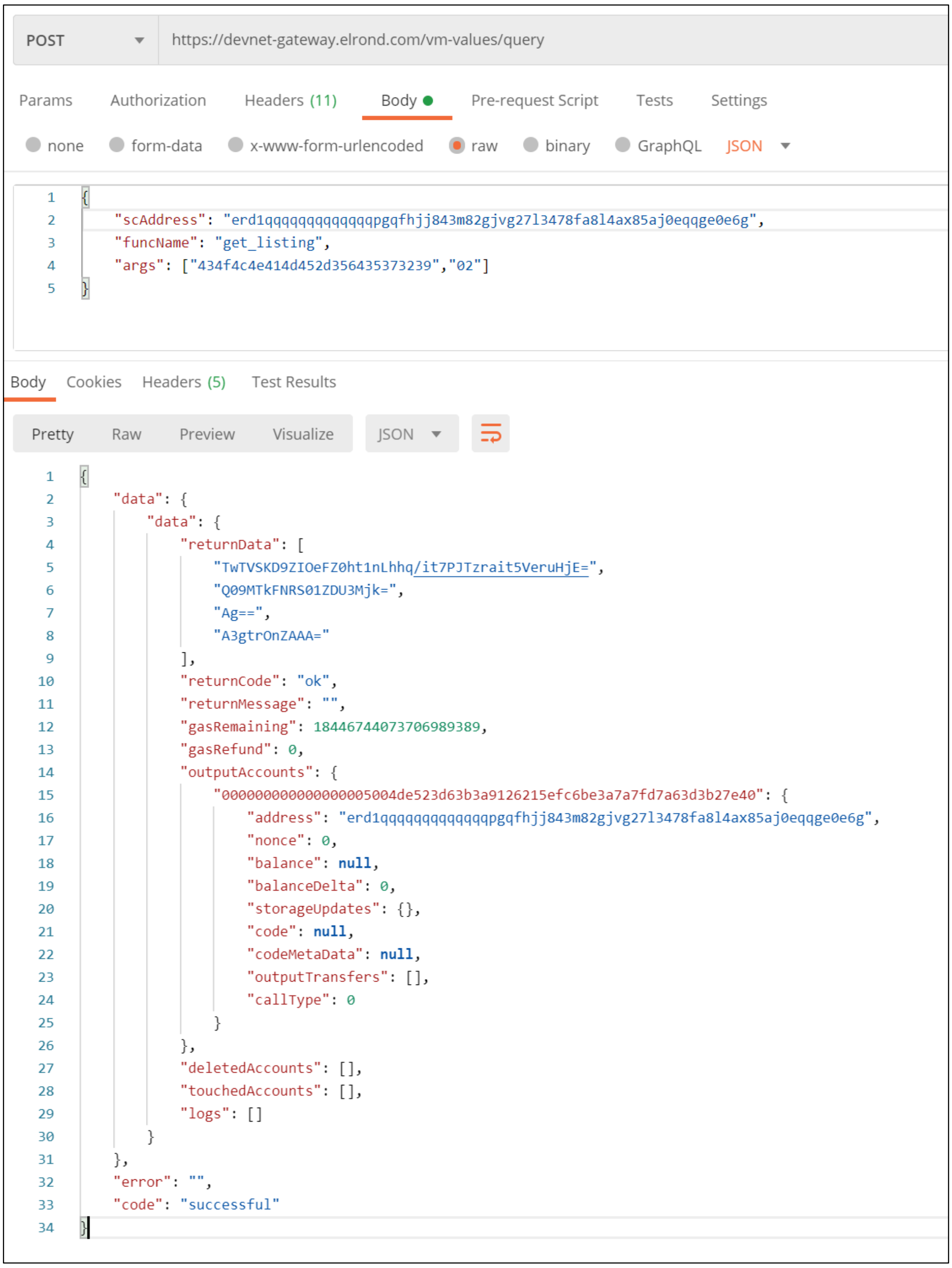
Hash	a33d0e53a4f05fa293f49d7247c75ea3e025d76a23b509313d1126be068f6938
Status	Success
Age	2 mins 45 secs (Apr 25, 2022 01:20:48 AM UTC)
Miniblock	fad7a170fc003debca91a37b155b1f912128c5a1174cff9f80d9e394729699a9
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	Contract erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g (Shard 0)
Value	0 XeGLD (= $0.00)
Method	cancel_listing
Token Operations 1	Transfer from erd1qq... 0gq6g To erd1fuzd2j9 ... rccs2g04x2 COLNAME-5d5729-03
Transaction Fee	0.00015358636 XeGLD (≈ $0.0227)
EGLD Price	$147.53
Gas Limit	6,000,000
Gas Used	3,577,636
Gas Price	0.000000001 XeGLD
Nonce	89
Input Data	cancel_listing@COLNAME-5d5729@

Smart Contract Results

Hash	558ce1c46c6ef28fe349344a6ef418ca1e69f54b7dd67f86969791c5dfbcbaa5
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0.00002422364 XeGLD
Data	@ok

Hash	cf3fd5747f2d0a17065ed92ed953b891fdd3737072eca5422cd51f91d355c965
From	erd1qqqqqqqqqqqqqpgquezagyjd0yey0vpkf24czpkn88k3xayv0eqqq0gq6g
To	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2
Value	0 XeGLD
Data	ESDTNFTTransfer@COLNAME-5d5729@@@405016068957026543852203400339776817698564556874114682445512469640347034371332852315788311555906730731906836816277595223535691410304348725365943068752978468344903409178878227863578084541814675598

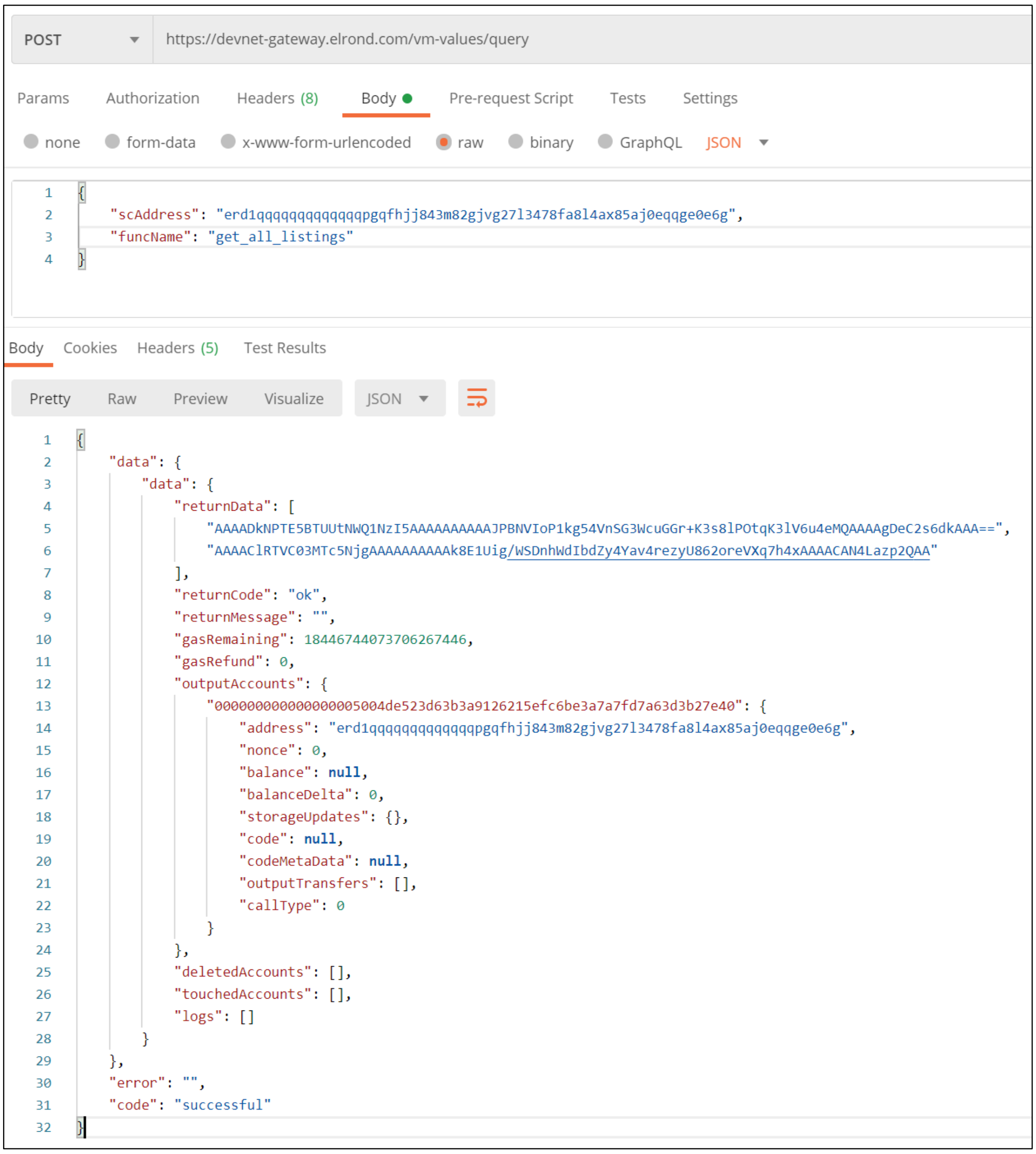
Appendix 11 – Get listing data WASM request



```
{
    "scAddress": "erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g",
    "funcName": "get_listing",
    "args": ["434f4c4e414d452d356435373239","02"]
}
```

```
{
    "data": {
        "data": {
            "returnData": [
                "TwTVSKD9ZIOeFZ0ht1nLhhq/it7PJTzrait5VeruHjE=",
                "Q09MTkFNRS01ZDU3Mjk=",
                "Ag==",
                "A3gtrOnZAAA="
            ],
            "returnCode": "ok",
            "returnMessage": "",
            "gasRemaining": 18446744073706989389,
            "gasRefund": 0,
            "outputAccounts": {
                "000000000000000005004de523d63b3a9126215efc6be3a7a7fd7a63d3b27e40": {
                    "address": "erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g",
                    "nonce": 0,
                    "balance": null,
                    "balanceDelta": 0,
                    "storageUpdates": {},
                    "code": null,
                    "codeMetaData": null,
                    "outputTransfers": [],
                    "callType": 0
                }
            },
            "deletedAccounts": [],
            "touchedAccounts": [],
            "logs": []
        }
    },
    "error": "",
    "code": "successful"
}
```

Appendix 12 – Get all listings WASM request



POST https://devnet-gateway.elrond.com/vm-values/query

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

none | form-data | x-www-form-urlencoded | [✓ raw] | binary | GraphQL | JSON

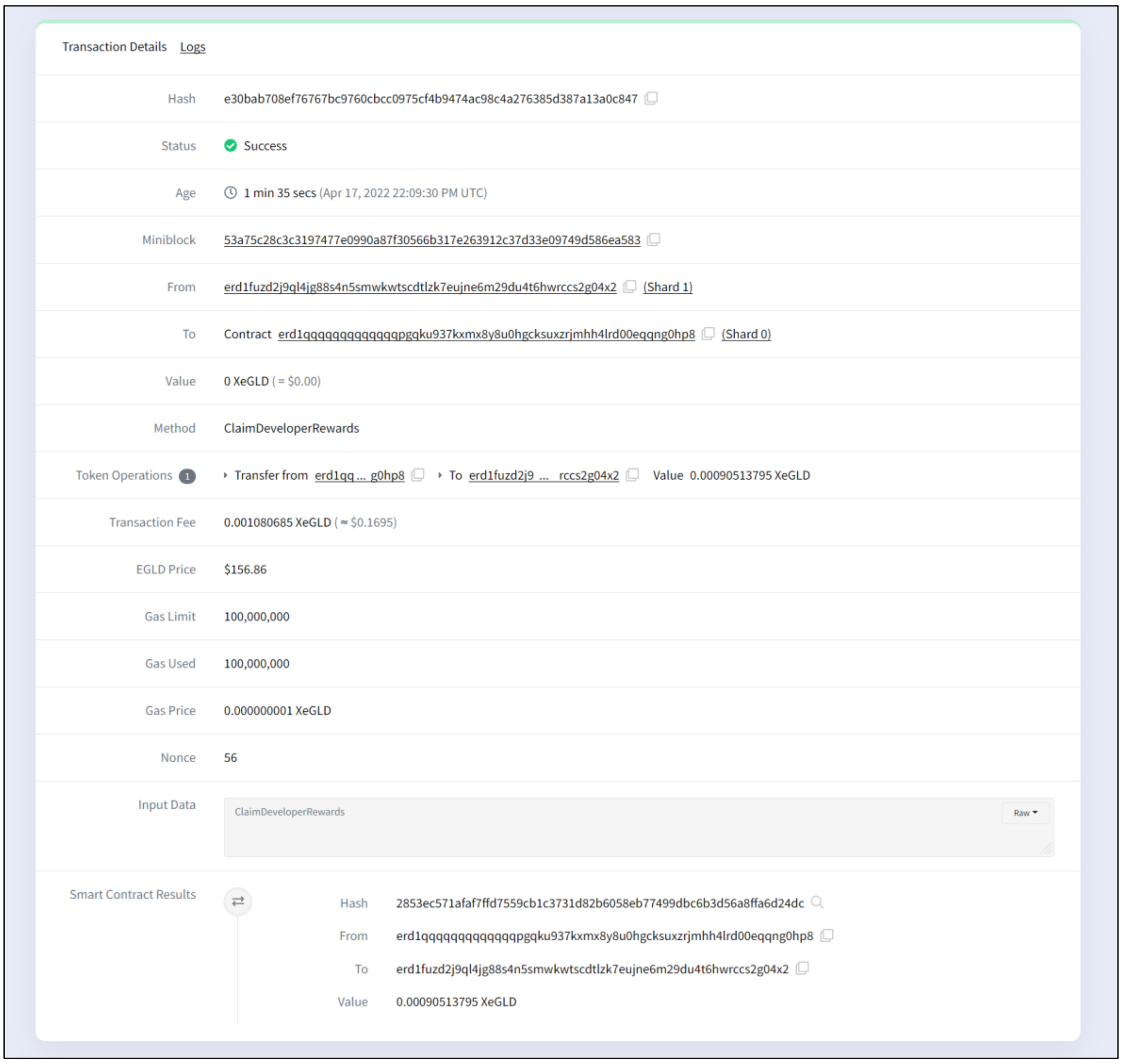
```
{
    "scAddress": "erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g",
    "funcName": "get_all_listings"
}
```

Body | Cookies | Headers (5) | Test Results

Pretty | Raw | Preview | Visualize | JSON

```
{
    "data": {
        "data": {
            "returnData": [
                "AAAADkNPTE5BTUUtNWQ1NzI5AAAAAAAAAAJPBNVIoP1kg54VnSG3WcuGGr+K3s8lPOtqK3lV6u4eMQAAAAgDeC2s6dkAAA==",
                "AAAAClRTVC03MTc5NjgAAAAAAAAAAk8E1Uig/WSDnhWdIbdZy4Yav4rezyU862oreVXq7h4xAAAACAN4Lazp2QAA"
            ],
            "returnCode": "ok",
            "returnMessage": "",
            "gasRemaining": 18446744073706267446,
            "gasRefund": 0,
            "outputAccounts": {
                "000000000000000005004de523d63b3a9126215efc6be3a7a7fd7a63d3b27e40": {
                    "address": "erd1qqqqqqqqqqqqqpgqfhjj843m82gjvg27l3478fa8l4ax85aj0eqqge0e6g",
                    "nonce": 0,
                    "balance": null,
                    "balanceDelta": 0,
                    "storageUpdates": {},
                    "code": null,
                    "codeMetaData": null,
                    "outputTransfers": [],
                    "callType": 0
                }
            },
            "deletedAccounts": [],
            "touchedAccounts": [],
            "logs": []
        }
    },
    "error": "",
    "code": "successful"
}
```

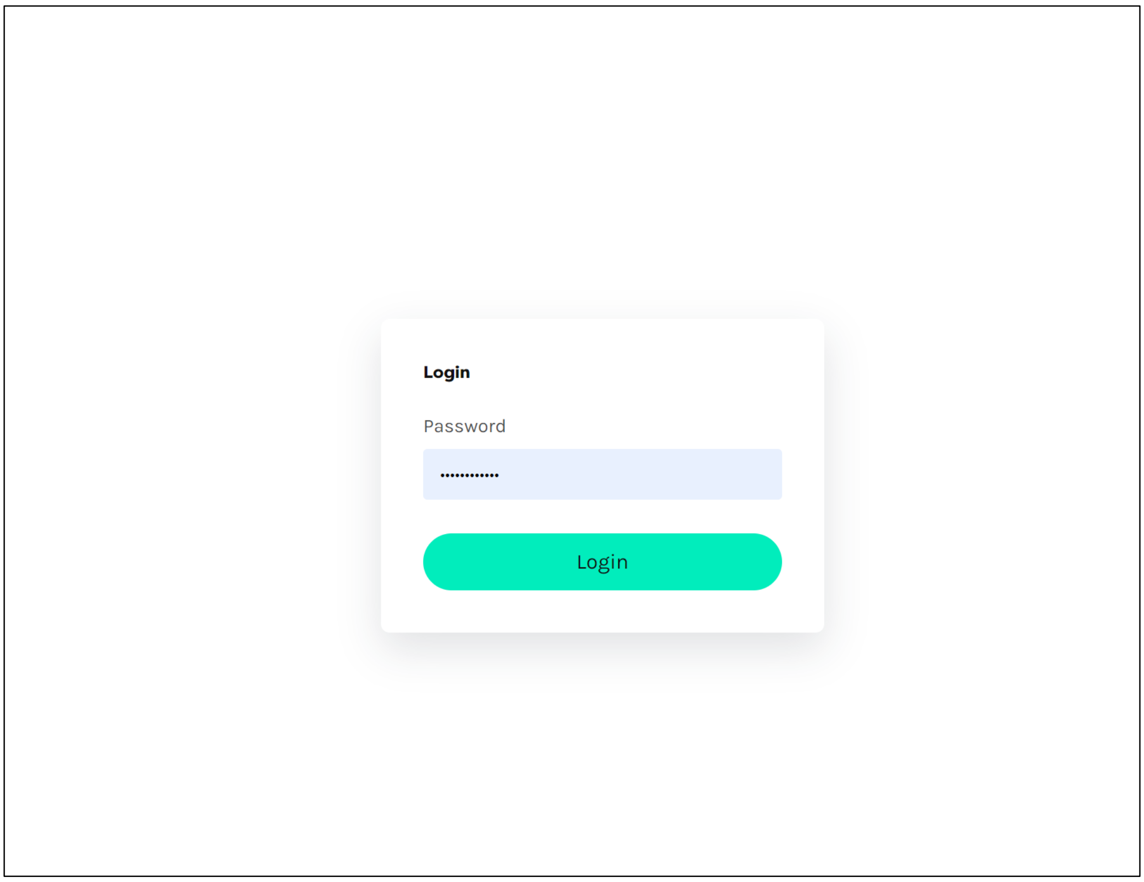
Appendix 13 – Claim developer rewards



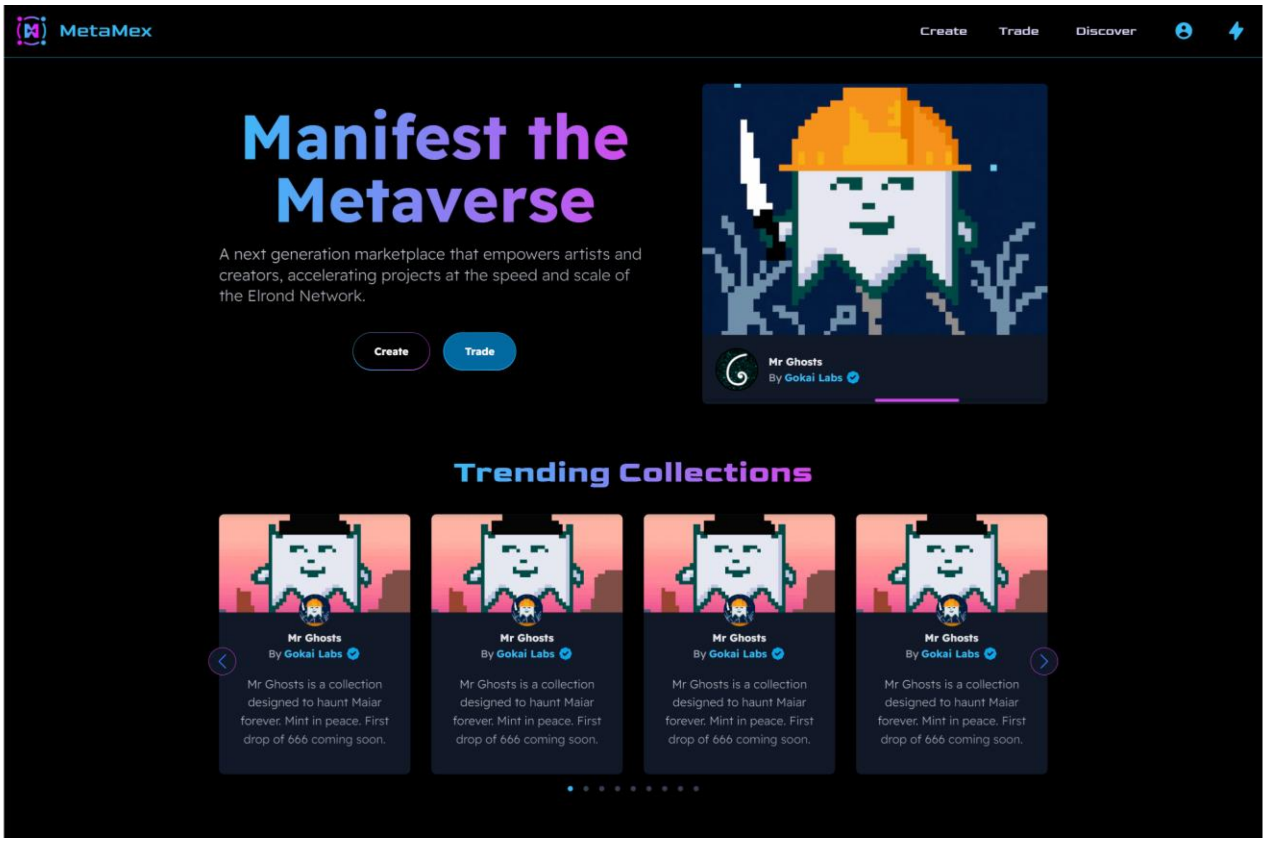
Transaction Details Logs

Hash	e30bab708ef76767bc9760cbcc0975cf4b9474ac98c4a276385d387a13a0c847
Status	Success
Age	1 min 35 secs (Apr 17, 2022 22:09:30 PM UTC)
Miniblock	53a75c28c3c3197477e0990a87f30566b317e263912c37d33e09749d586ea583
From	erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 (Shard 1)
To	Contract erd1qqqqqqqqqqqqqpgqku937kxmx8y8u0hgcksuxzrjmhh4lrd00eqqng0hp8 (Shard 0)
Value	0 XeGLD (= $0.00)
Method	ClaimDeveloperRewards
Token Operations 1	Transfer from erd1qq ... g0hp8 To erd1fuzd2j9 ... rccs2g04x2 Value 0.00090513795 XeGLD
Transaction Fee	0.001080685 XeGLD (≈ $0.1695)
EGLD Price	$156.86
Gas Limit	100,000,000
Gas Used	100,000,000
Gas Price	0.000000001 XeGLD
Nonce	56
Input Data	ClaimDeveloperRewards
Smart Contract Results	Hash 2853ec571afaf7ffd7559cb1c3731d82b6058eb77499dbc6b3d56a8ffa6d24dc From erd1qqqqqqqqqqqqqpgqku937kxmx8y8u0hgcksuxzrjmhh4lrd00eqqng0hp8 To erd1fuzd2j9ql4jg88s4n5smwkwtscdtlzk7eujne6m29du4t6hwrccs2g04x2 Value 0.00090513795 XeGLD

Appendix 14 – Main page password protected



Appendix 15 – Main page



Manifest the Metaverse

A next generation marketplace that empowers artists and creators, accelerating projects at the speed and scale of the Elrond Network.

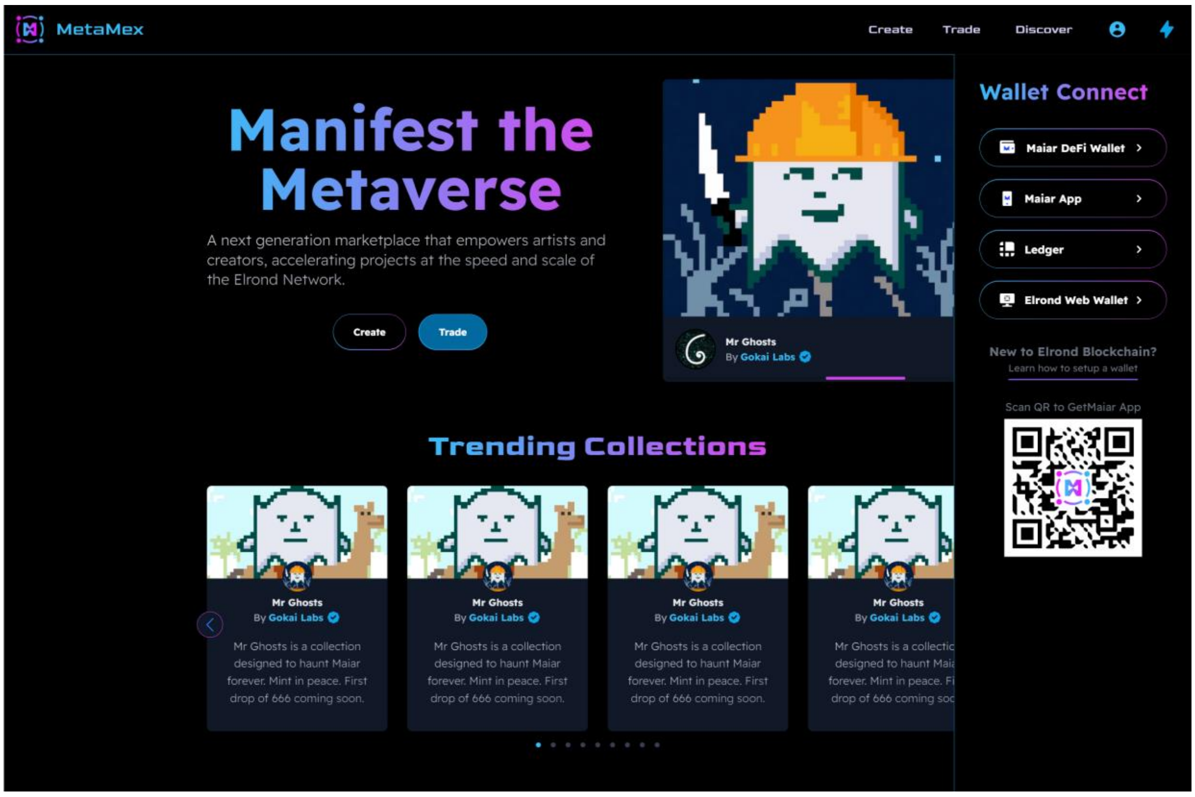
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Appendix 16 – Unauthorized sidebar



Manifest the Metaverse

A next generation marketplace that empowers artists and creators, accelerating projects at the speed and scale of the Elrond Network.

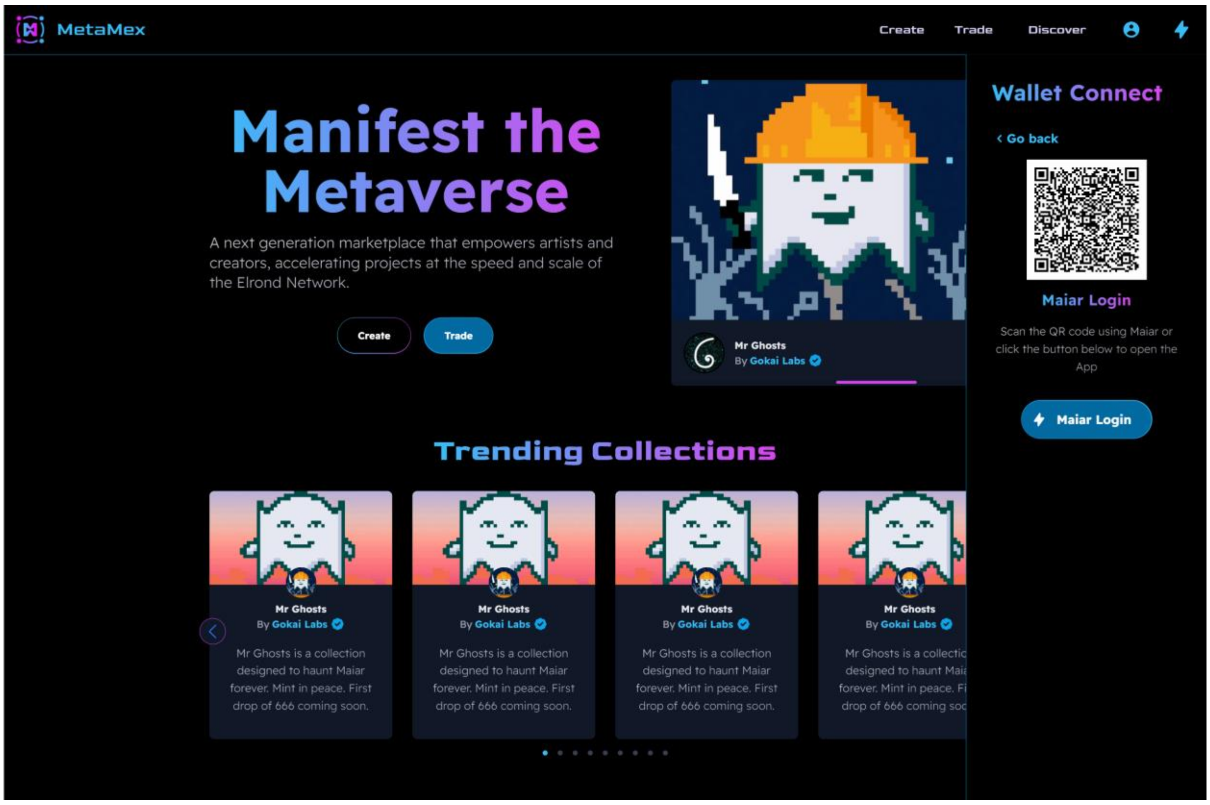
Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collectic designed to haunt Maia forever. Mint in peace. Fi drop of 666 coming soo

Appendix 17 – Sidebar Maiar App login

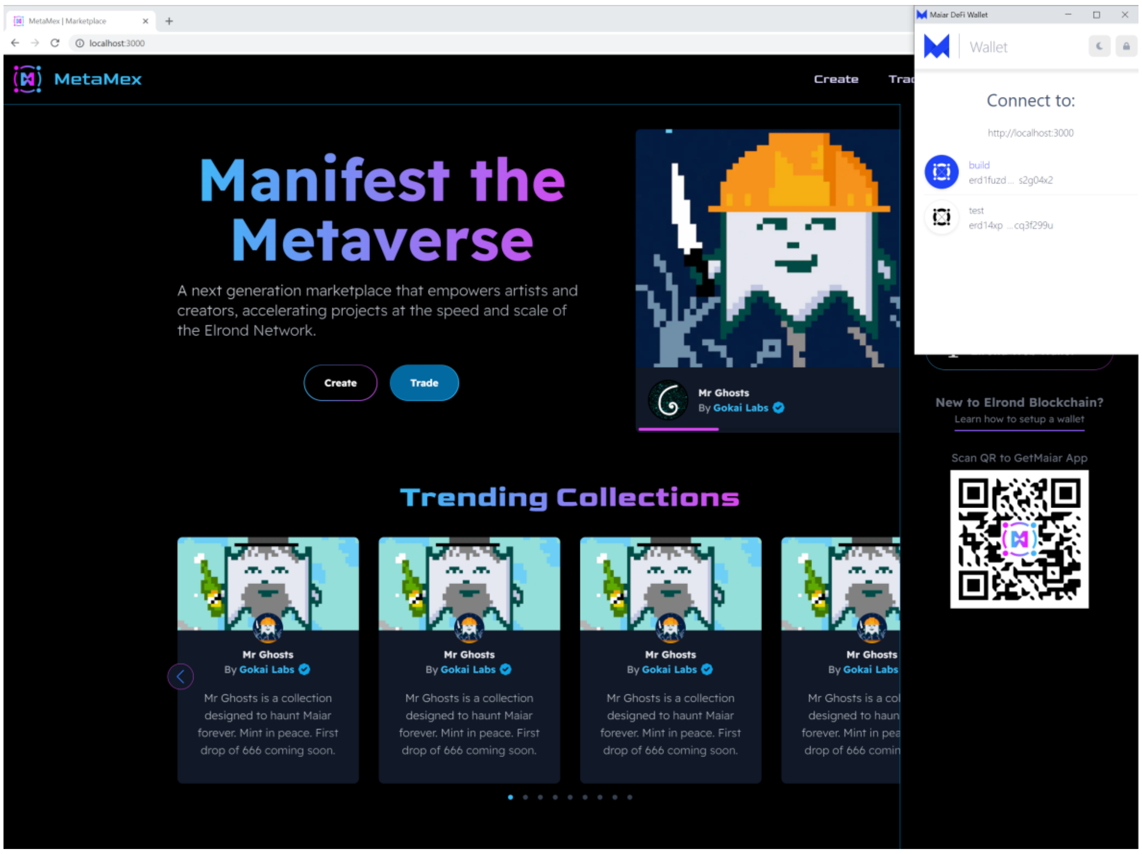


A next generation marketplace that empowers artists and creators, accelerating projects at the speed and scale of the Elrond Network.

Scan the QR code using Maiar or click the button below to open the App

Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

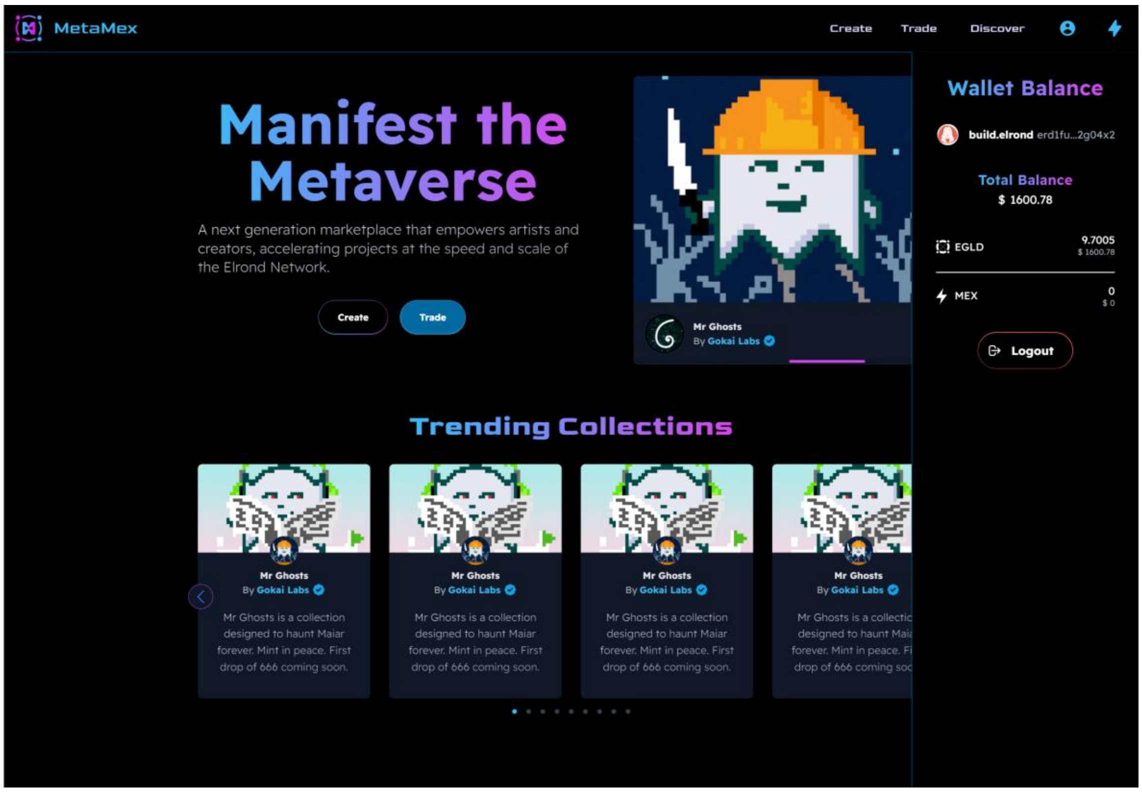
Appendix 18 – Sidebar Maiar DeFi Wallet login



A next generation marketplace that empowers artists and creators, accelerating projects at the speed and scale of the Elrond Network.

Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Appendix 19 – Sidebar when customer logged in



Manifest the Metaverse

A next generation marketplace that empowers artists and creators, accelerating projects at the speed and scale of the Elrond Network.

Mr Ghosts
By Gokai Labs

Wallet Balance

build.elrond erd1fu...2g04x2

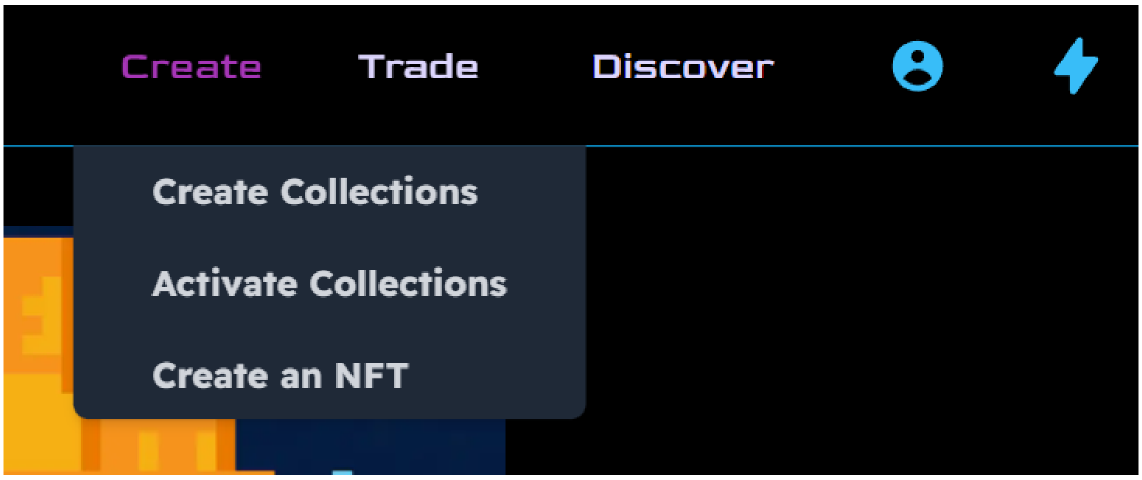
Total Balance
$ 1600.78

EGLD 9.7005
$ 1600.78

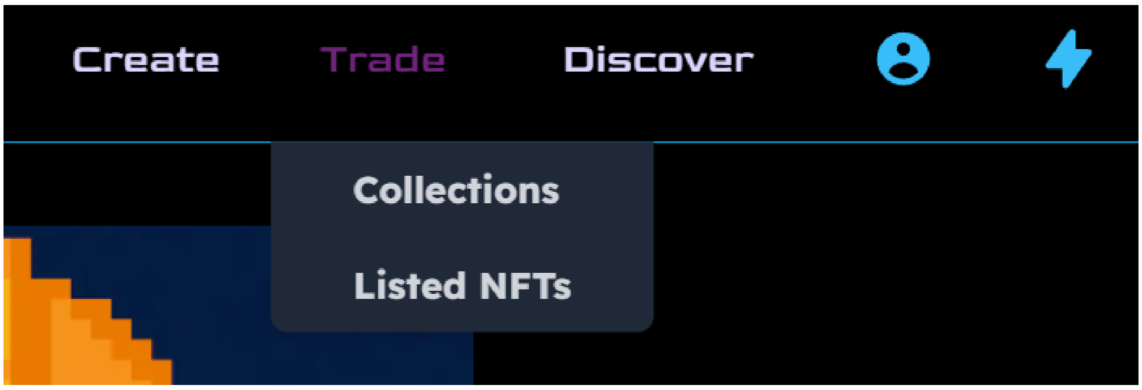
MEX 0
$ 0

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

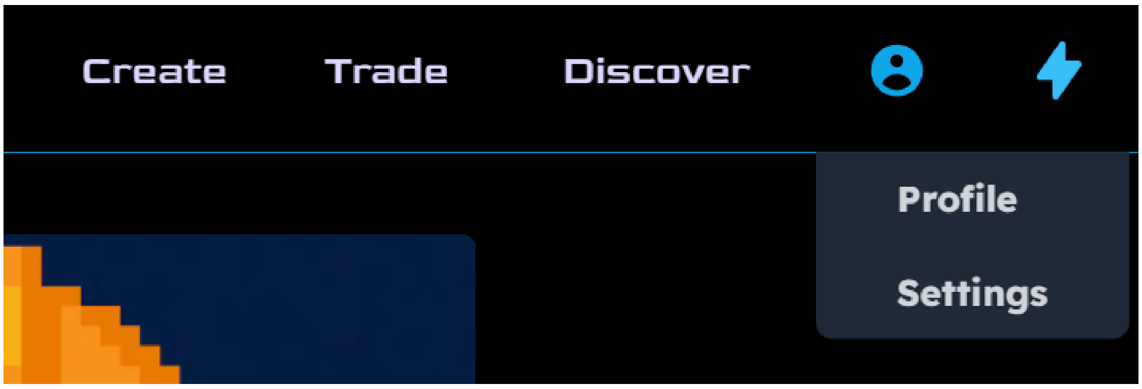
Appendix 20 – Create menu dropdown



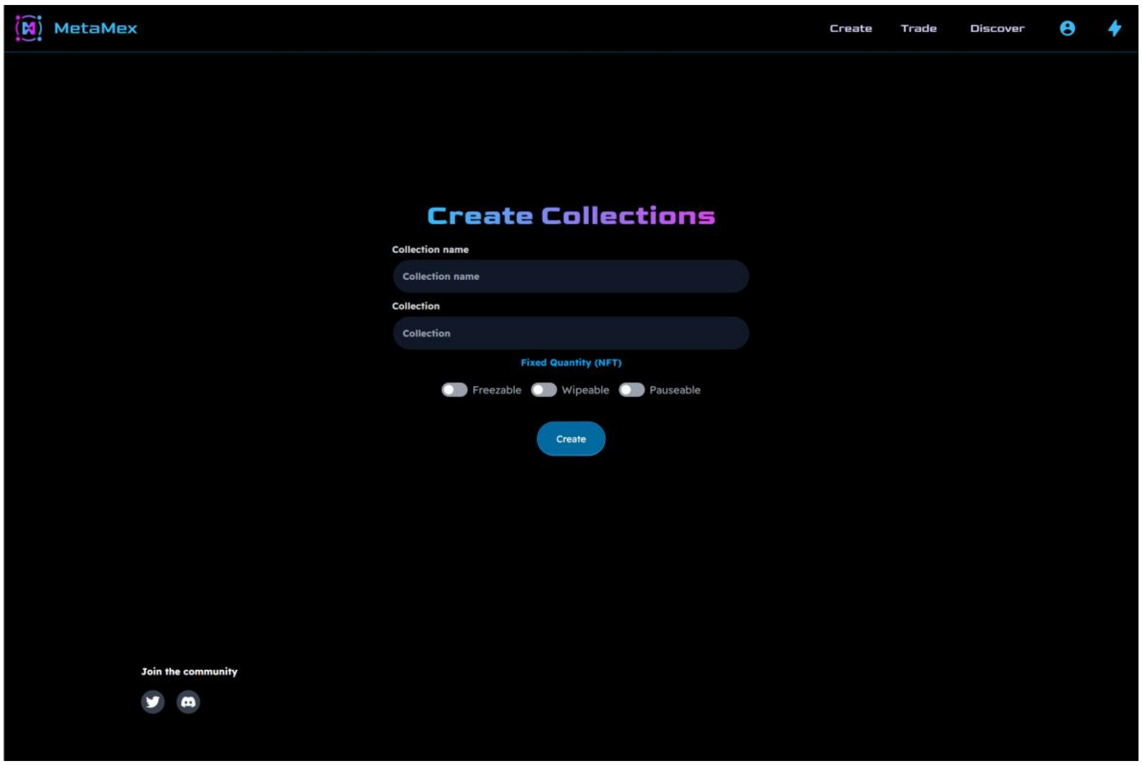
Appendix 21 – Trade menu dropdown



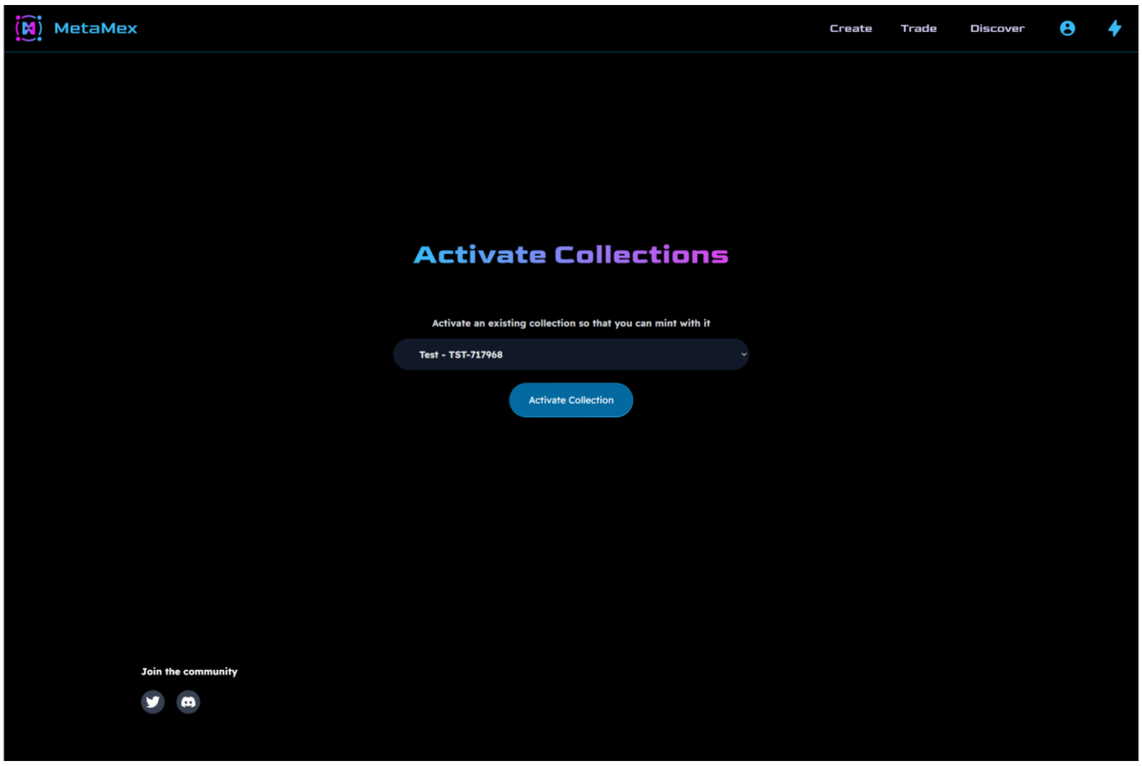
Appendix 22 – Profile menu dropdown



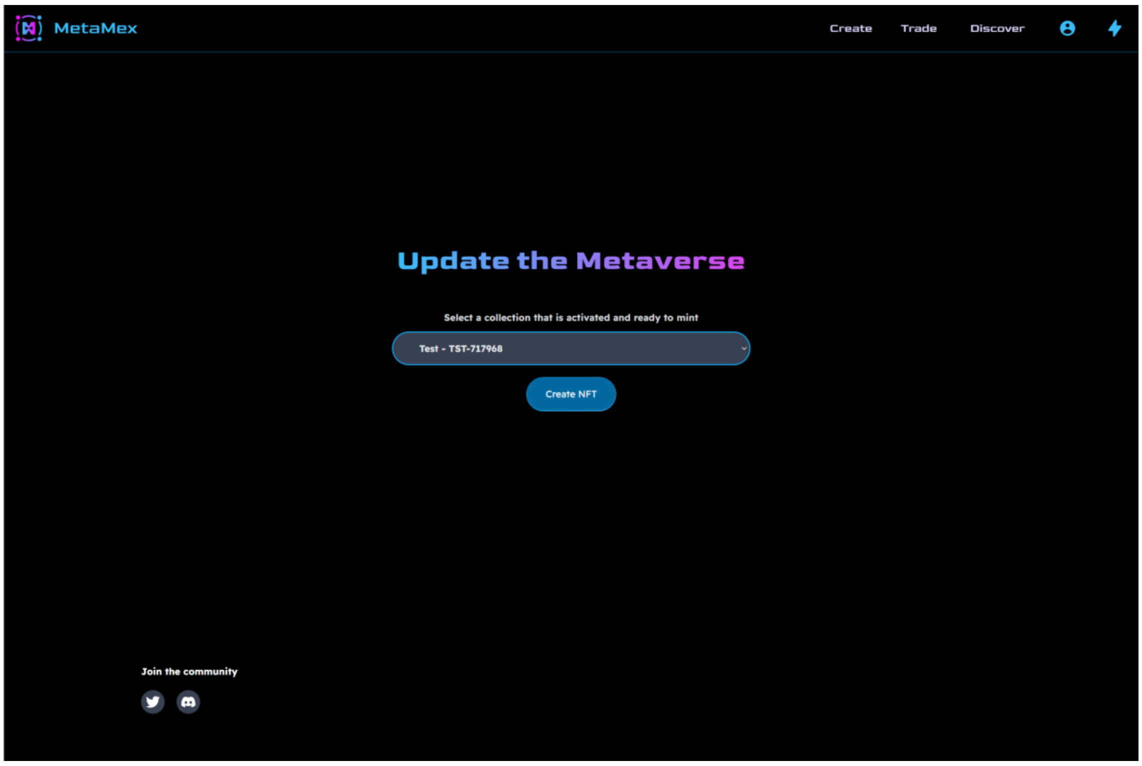
Appendix 23 – Create collections screen



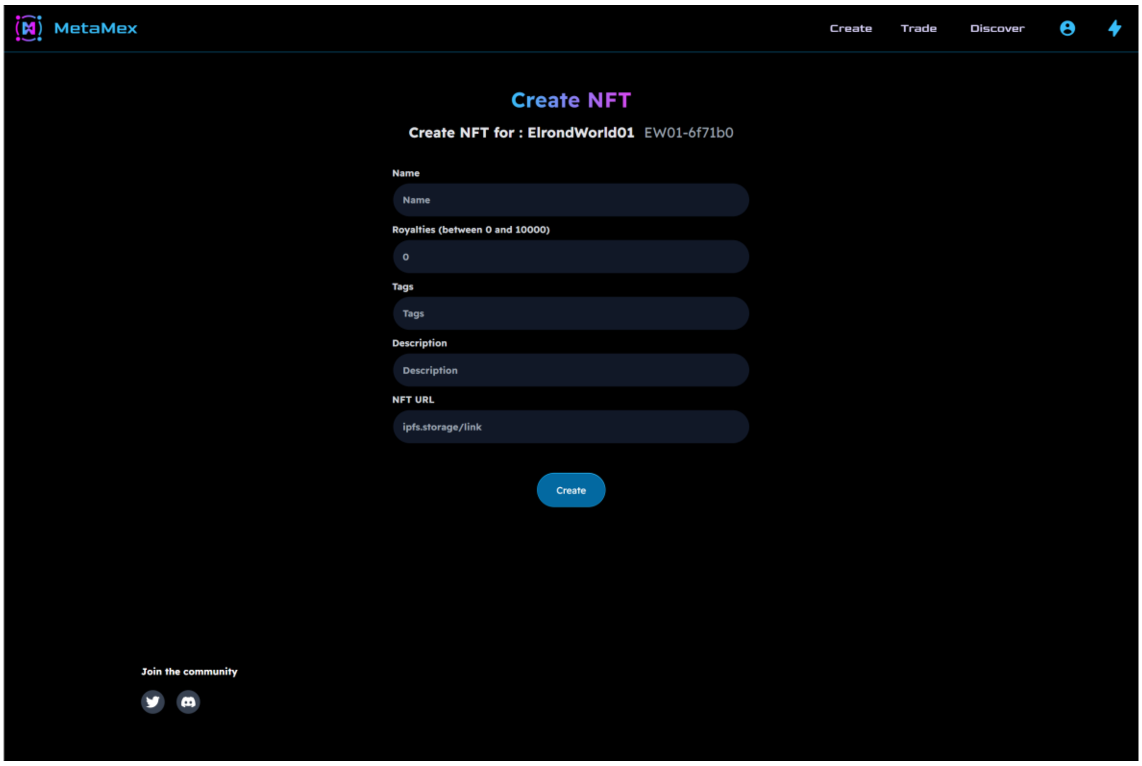
Appendix 24 – Activate collections screen



Appendix 25 – Pick collection for NFT creation screen



Appendix 26 – Create NFT screen



Create NFT

Create NFT for : ElrondWorld01 EW01-6f71b0

Name

Name

Royalties (between 0 and 10000)

0

Tags

Tags

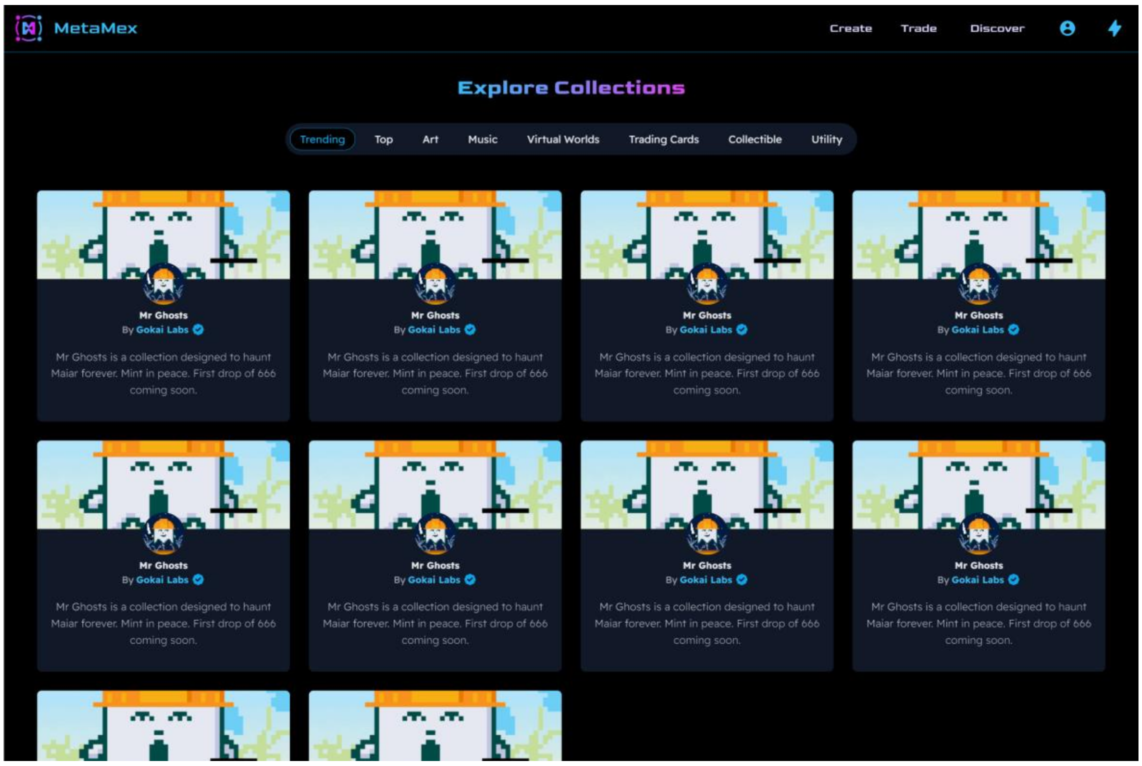
Description

Description

NFT URL

ipfs.storage/link

Appendix 27 – Explore collections screen



Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

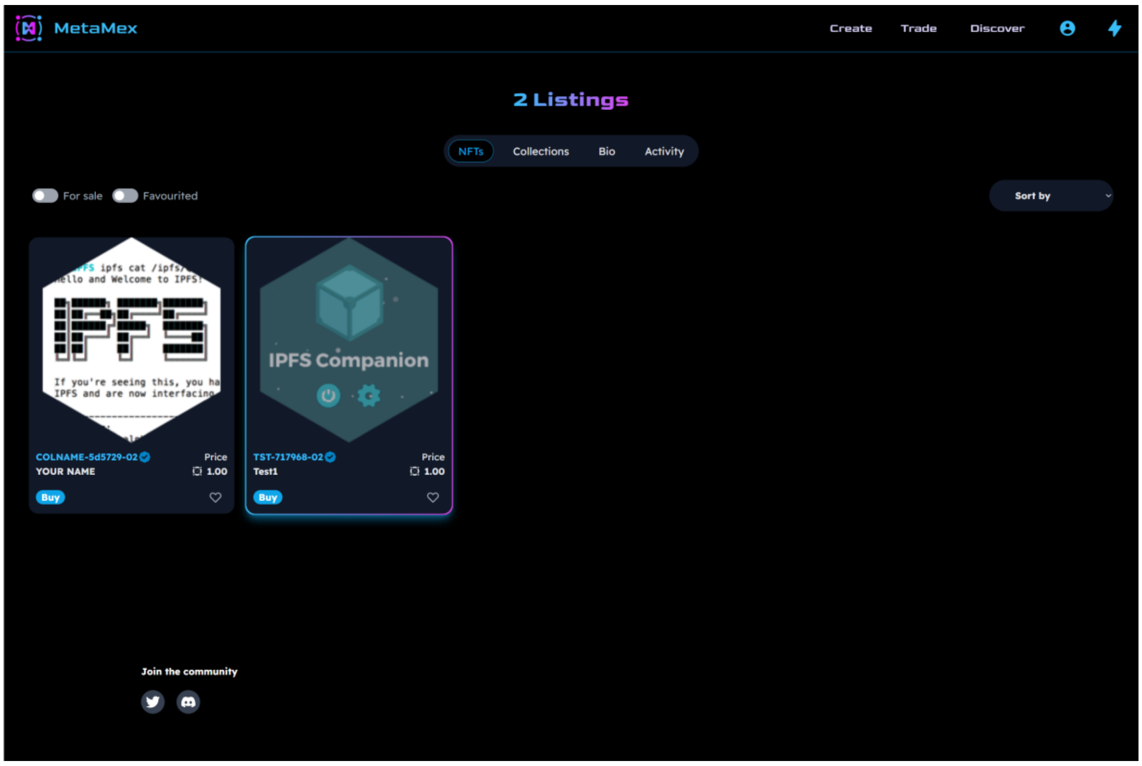
Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

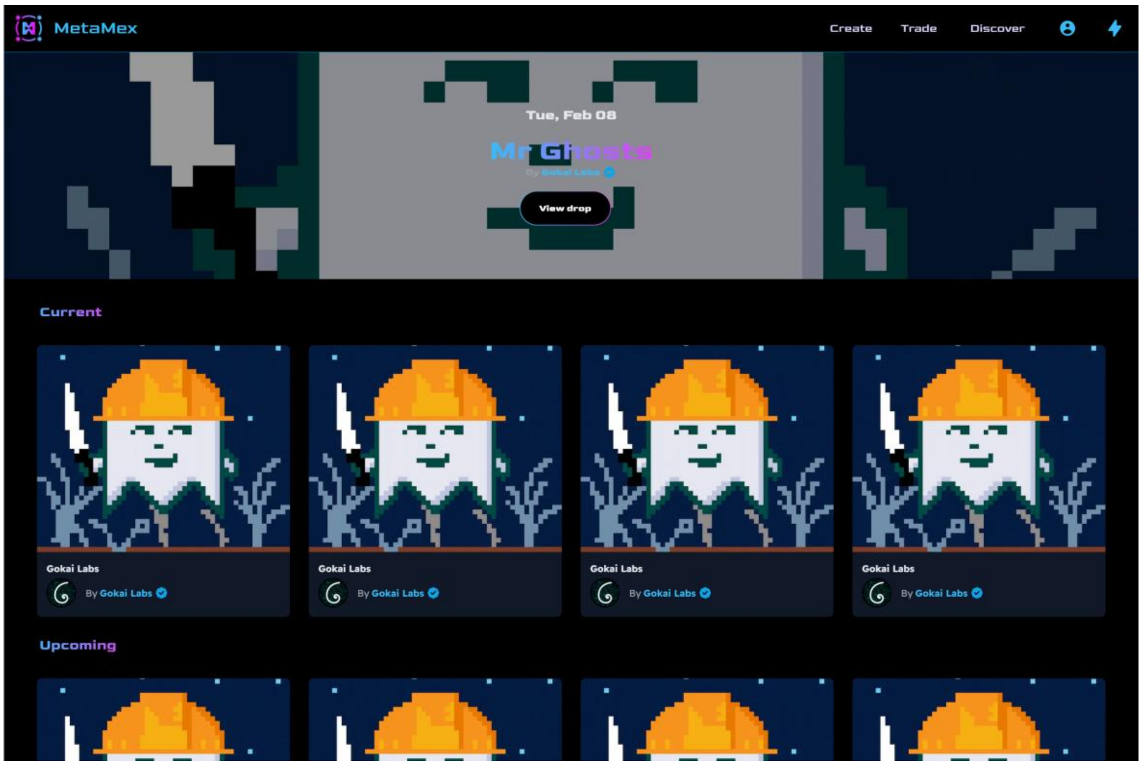
Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Mr Ghosts
By Gokai Labs
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

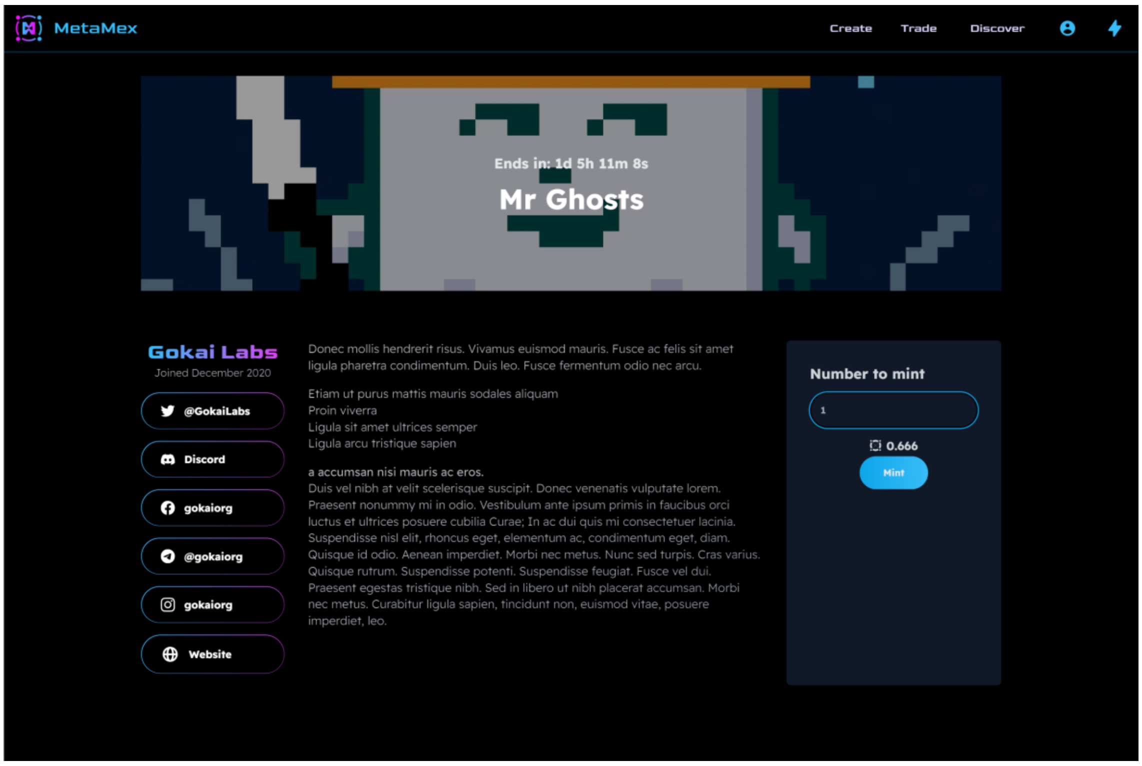
Appendix 28 – Explore listings screen



Appendix 29 – Discover screen



Appendix 30 – Mint collection screen

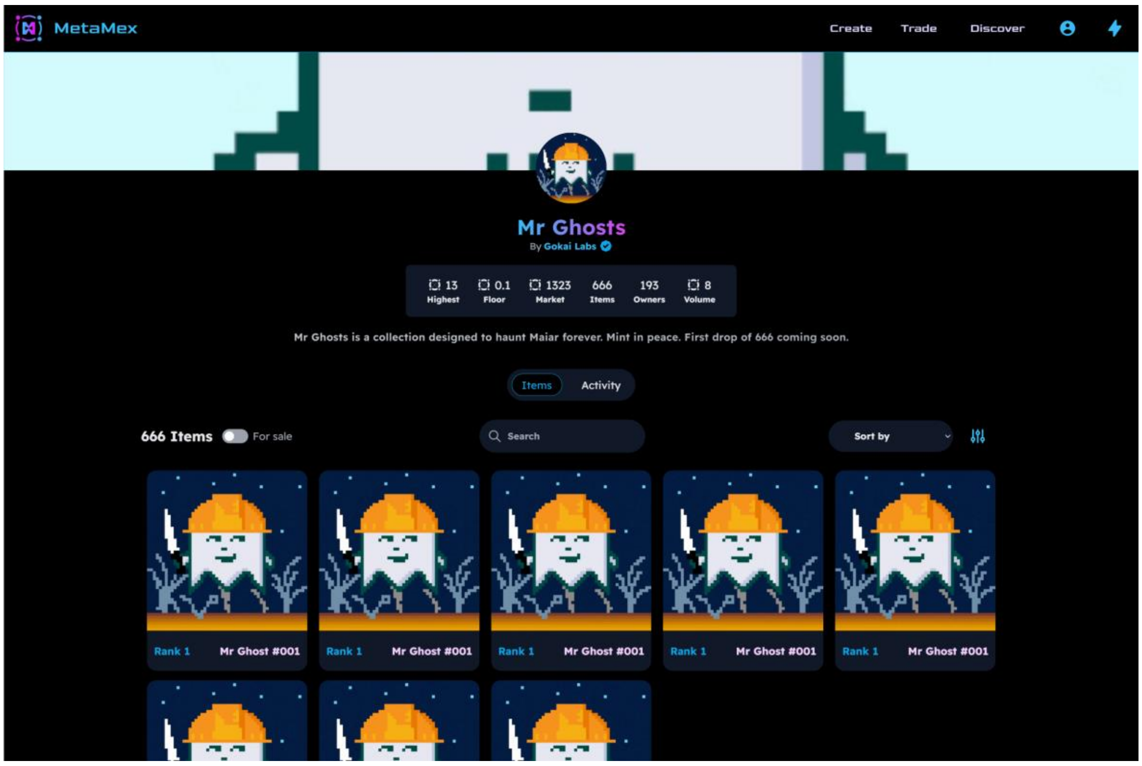


Donec mollis hendrerit risus. Vivamus euismod mauris. Fusce ac felis sit amet ligula pharetra condimentum. Duis leo. Fusce fermentum odio nec arcu.

Etiam ut purus mattis mauris sodales aliquam
Proin viverra
Ligula sit amet ultrices semper
Ligula arcu tristique sapien

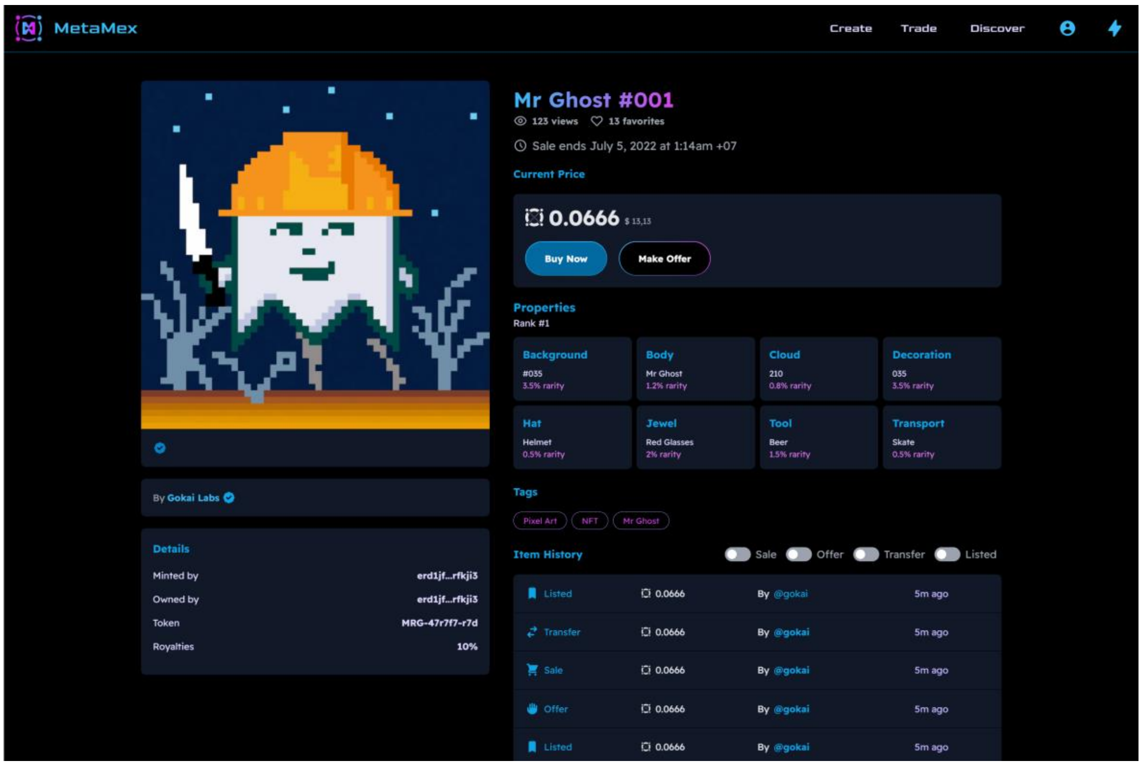
a accumsan nisi mauris ac eros.
Duis vel nibh at velit scelerisque suscipit. Donec venenatis vulputate lorem. Praesent nonummy mi in odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; In ac dui quis mi consectetuer lacinia. Suspendisse nisl elit, rhoncus eget, elementum ac, condimentum eget, diam. Quisque id odio. Aenean imperdiet. Morbi nec metus. Nunc sed turpis. Cras varius. Quisque rutrum. Suspendisse potenti. Suspendisse feugiat. Fusce vel dui. Praesent egestas tristique nibh. Sed in libero ut nibh placerat accumsan. Morbi nec metus. Curabitur ligula sapien, tincidunt non, euismod vitae, posuere imperdiet, leo.

Appendix 31 – Collection detail screen



Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Appendix 32 – NFT detail screen



Mr Ghost #001

123 views 13 favorites

Sale ends July 5, 2022 at 1:14am +07

Current Price

0.0666 $ 13,13

Properties

Rank #1

Background	Body	Cloud	Decoration
#035 3.5% rarity	Mr Ghost 1.2% rarity	210 0.8% rarity	035 3.5% rarity
Hat	**Jewel**	**Tool**	**Transport**
Helmet 0.5% rarity	Red Glasses 2% rarity	Beer 1.5% rarity	Skate 0.5% rarity

Tags

By Gokai Labs

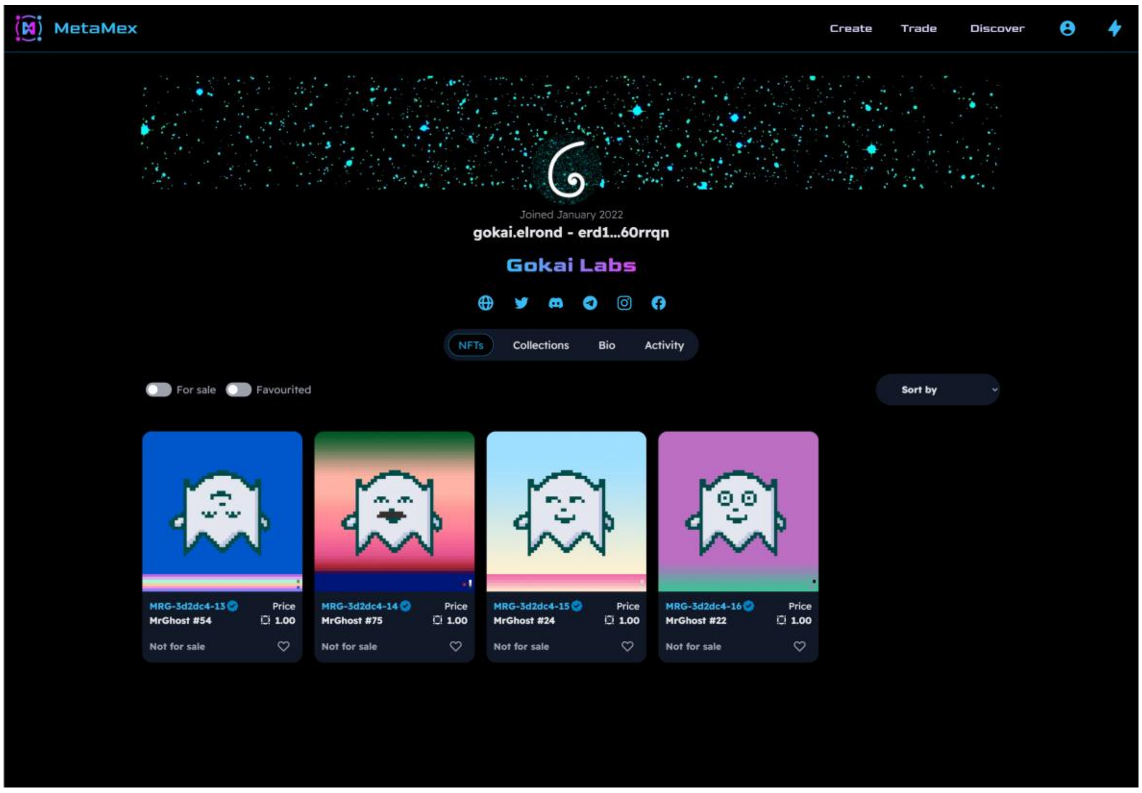
Details

Minted by	erd1jf...rfkji3
Owned by	erd1jf...rfkji3
Token	MRG-47r7f7-r7d
Royalties	10%

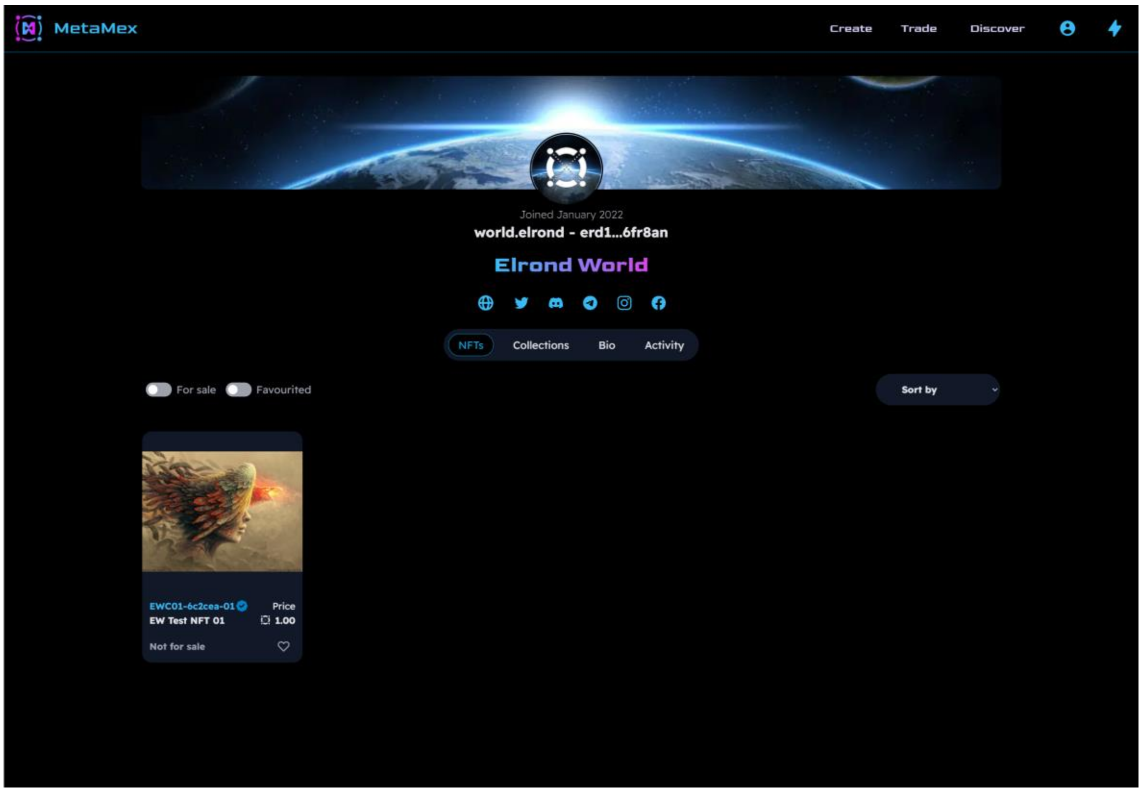
Item History

Listed	0.0666	By @gokai	5m ago
Transfer	0.0666	By @gokai	5m ago
Sale	0.0666	By @gokai	5m ago
Offer	0.0666	By @gokai	5m ago
Listed	0.0666	By @gokai	5m ago

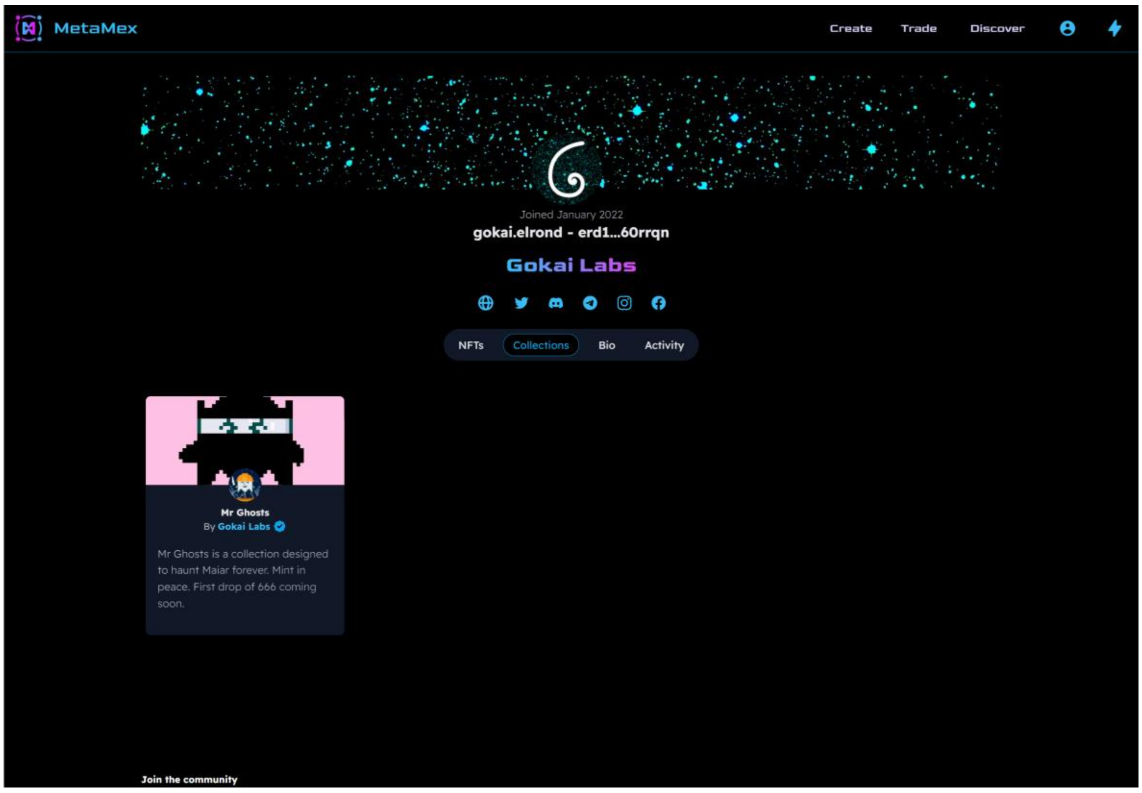
Appendix 33 – Gokai Labs profile screen



Appendix 34 – Elrond World profile screen

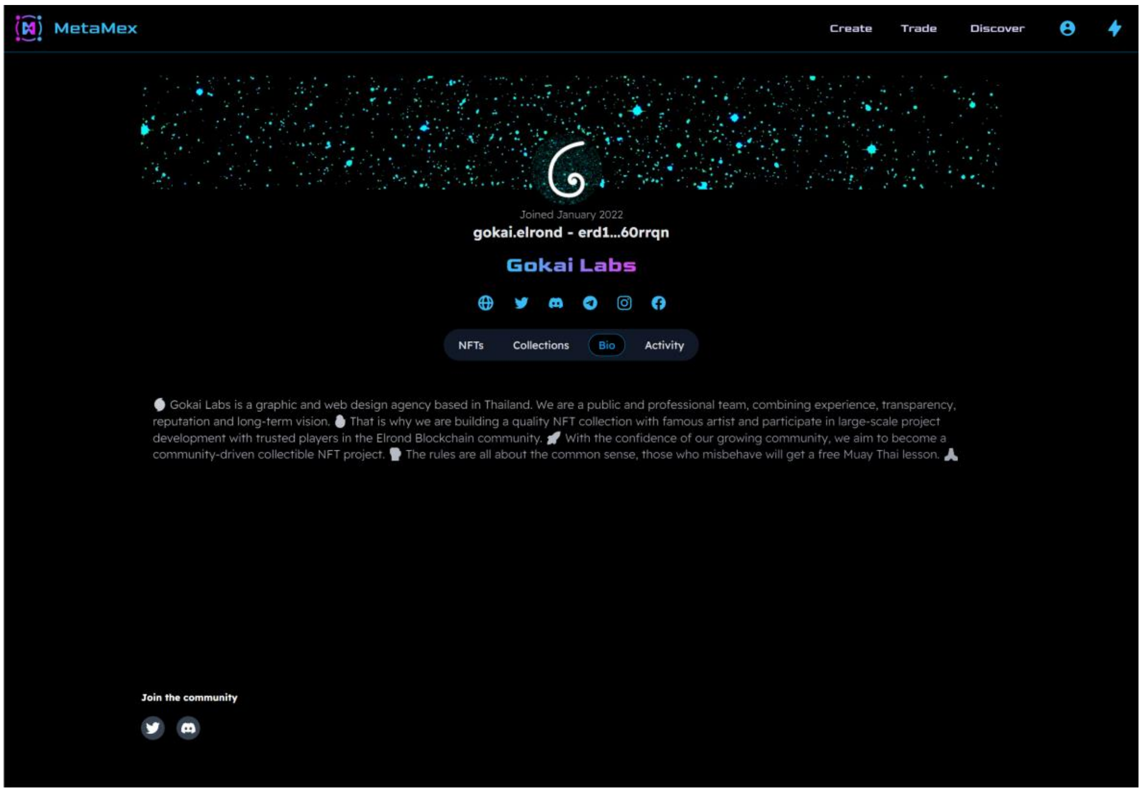


Appendix 35 – Profile collections screen



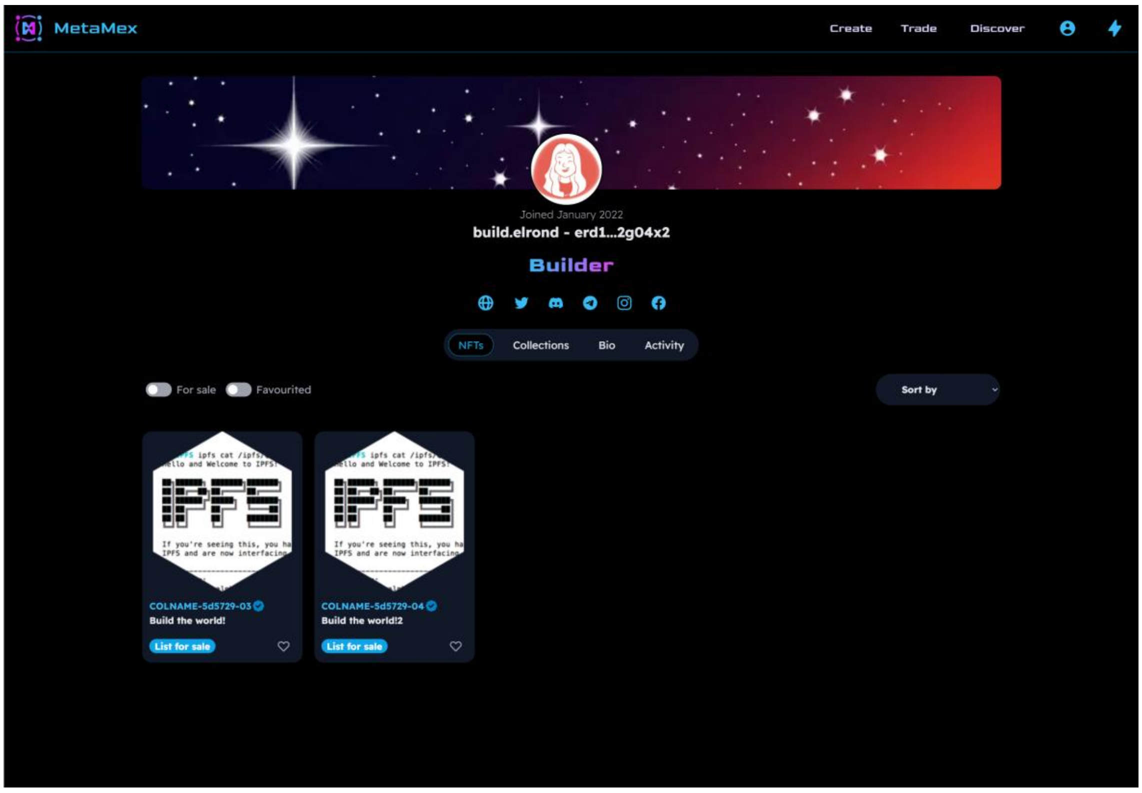
Mr Ghosts is a collection designed to haunt Maiar forever. Mint in peace. First drop of 666 coming soon.

Appendix 36 – Profile bio screen

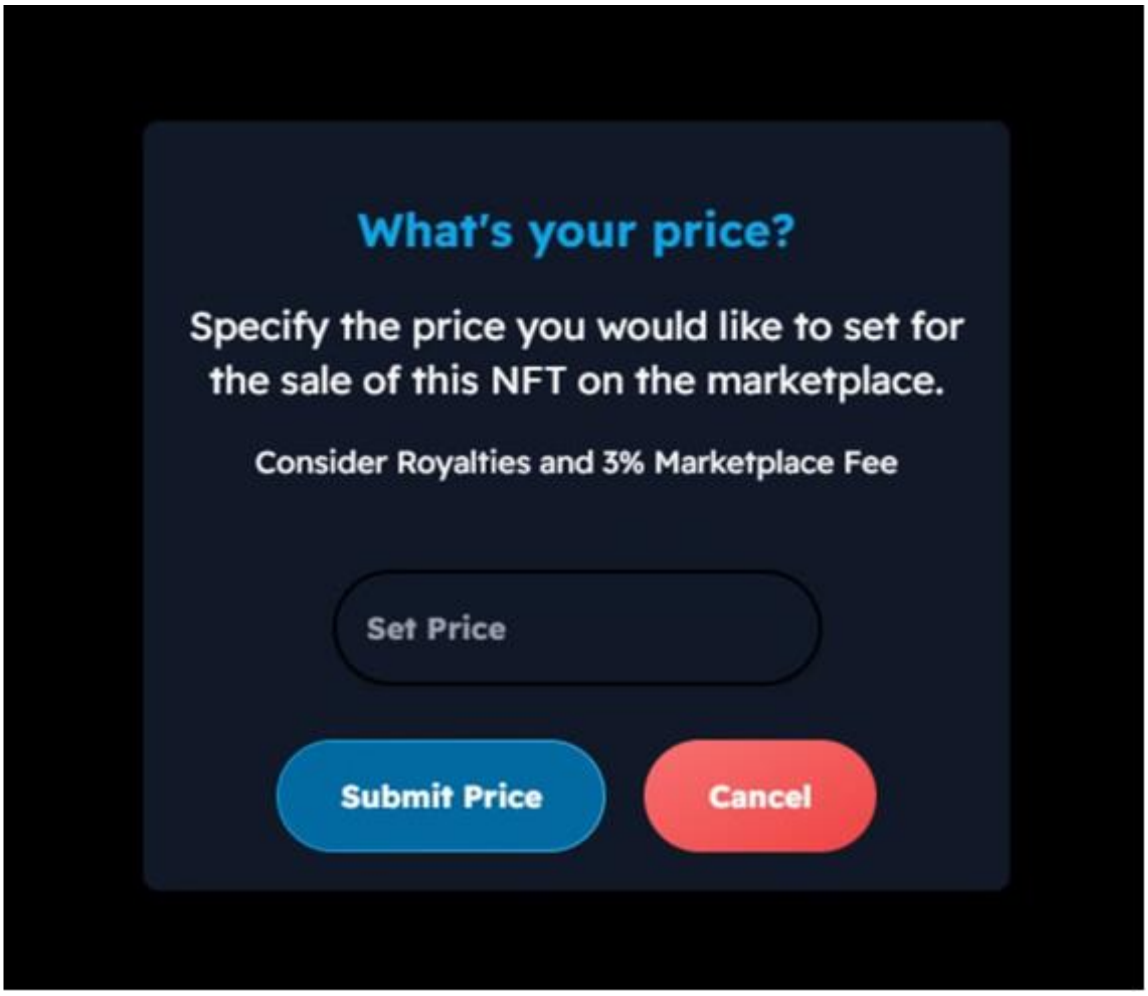


🌑 Gokai Labs is a graphic and web design agency based in Thailand. We are a public and professional team, combining experience, transparency, reputation and long-term vision. 🥚 That is why we are building a quality NFT collection with famous artist and participate in large-scale project development with trusted players in the Elrond Blockchain community. 🚀 With the confidence of our growing community, we aim to become a community-driven collectible NFT project. 🗣 The rules are all about the common sense, those who misbehave will get a free Muay Thai lesson. 🙏

Appendix 37 – Current customer's NFT management



Appendix 38 – Submit listing form

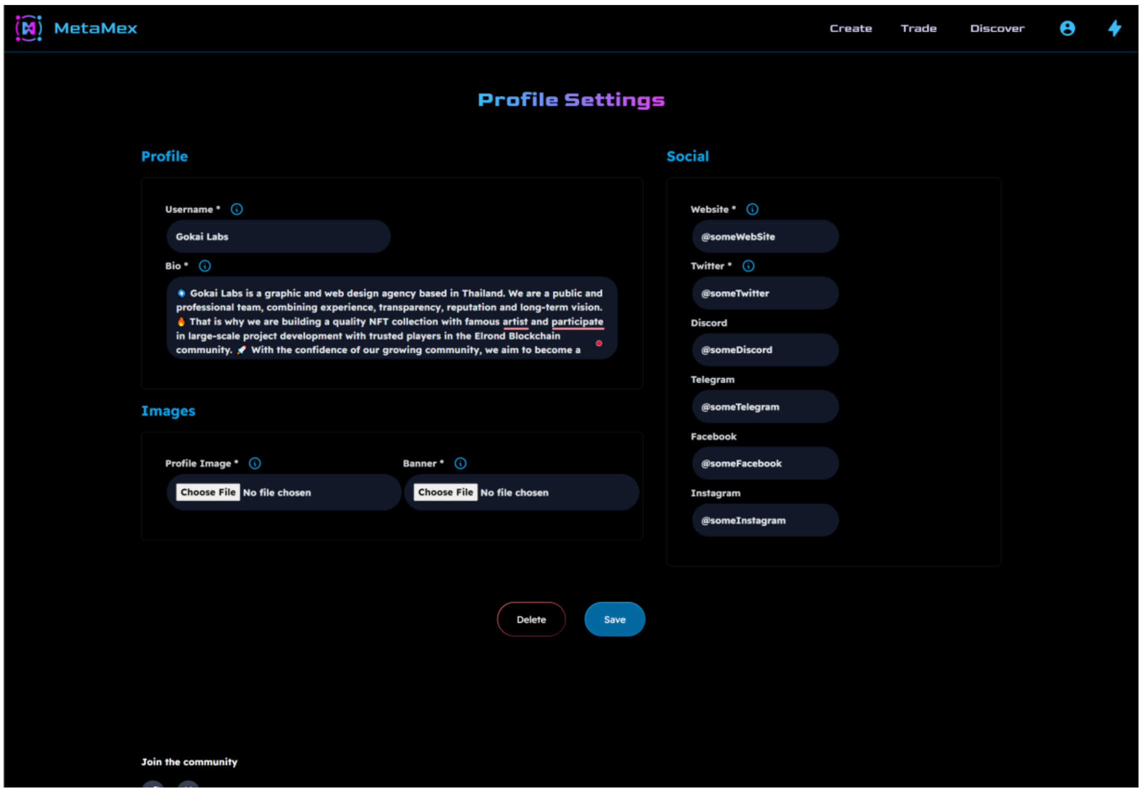


What's your price?

Specify the price you would like to set for the sale of this NFT on the marketplace.

Consider Royalties and 3% Marketplace Fee

Appendix 39 – Profile settings screen



Profile Settings

Profile

Username *

Gokai Labs

Bio *

🔹 Gokai Labs is a graphic and web design agency based in Thailand. We are a public and professional team, combining experience, transparency, reputation and long-term vision. 🔥 That is why we are building a quality NFT collection with famous artist and participate in large-scale project development with trusted players in the Elrond Blockchain community. 🚀 With the confidence of our growing community, we aim to become a

Images

Profile Image * — Choose File No file chosen

Banner * — Choose File No file chosen

Social

Website *	@someWebSite
Twitter *	@someTwitter
Discord	@someDiscord
Telegram	@someTelegram
Facebook	@someFacebook
Instagram	@someInstagram