

Lunar Foundation Q4 2022 Contract Updates

Requirements

Prepared for:
Lunar Foundation

Authors:
Shepard, MasterJedi

Last Updated:
December 9th, 2022



Table of Contents

Lunar Foundation Q4 2022 Contract Updates	1
Requirements	1
What is a Requirements Document?	3
Objective Outline	3
2022 Q4 Contract Updates Overview	4
Introduction	4
Project Priority	4
Efforts Estimate	4
Contract Update Requirements	5
Business Objectives	5
Objective #1: ERC-20 - Expand Fee Distribution Granularity	5
Objective #2: ERC-20 - Support Larger Wallets for Centralized Exchanges	6
Objective #3: ERC-721 - Enforce NFT Royalties	7
Implementation Objectives	8
Objective # 1: ERC-20 - Establish Development Wallet and Modify Fee Structures	8
Objective # 2: ERC-20 - Implement New Safeguards for Centralized Exchanges	9
Objective # 3: ERC 721 - Research Royalty Standards for OpenSea	10
Objective # 4: ERC-721 - Implement EIP-2981 Royalty Standards	11
Contract Update Implementation Details	12
ERC-20 Update Details	12
Structures	12
ERC-721 Update Details	13
OpenZeppelin Royalty Standards	13
Conclusion	14
Sign-Off	14

What is a Requirements Document?

The Requirements Document is the most important document in a software development project. This document outlines objectives a specific product must meet to provide value to the customer. It is imperative that all parties agree on the contents of this document before a single line of code is written, as misinterpretation of Requirements Documents is the #2 cause of software project failure, right behind inaccurate estimates.

Objective Outline

The objective of this engagement shall be to update two smart contracts currently managed by the Lunar Dev Team: the ERC-20 and ERC-721 contracts. Both token contracts are currently available via the BSC Mainnet.

Upgrade tasks shall be performed by the assigned contractors as part of this engagement. These tasks shall include but are not limited to:

ERC-20 Contract

- Add new SafeGuard structures to ensure centralized exchange rates and calculated values are available.
- Add a new Development wallet address and Update FeeRates Structures.
- Update the calculation of fees to properly reflect the desired fee structures.
- Update FeesPaidHistory to include new Development wallet.

ERC-721 Contract

- Determine the royalty percentage being set by the Lunar Foundation.
- Research OpenSea marketplace royalty standards.
- Establish royalty percentage and royalty payment percentages using ERC-2981.
- Conduct Internal Testing to ensure royalty calculations are processed as desired.

Now that we have provided an outline of the tasks at hand, we can jump into the priorities and an initial timeline of this engagement, with clear descriptions of the business objectives.

2022 Q4 Contract Updates Overview

This section provides some additional details to support the objective outline.

Introduction

After months of managing and maintaining our Lunar contracts on the BSC Mainnet, the Lunar Foundation and Lunar Advisors have uncovered several new features that are needed to enhance our operational capabilities.

For the ERC-20 token, we would like more transparency around which operational fees are geared towards development, and which are for other day-to-day expenses. Additionally, since the LNR v2 migration, centralized exchanges require access to LNR in amounts that surpass the current anti-whale guards. We've had to mark those wallets as "internal" to bypass those restrictions, which currently allows those wallets to operate unchecked.

For the ERC-721 token, OpenSea's sudden and unannounced support for the BNB Chain has forced us to dig into their business practices and understand how they pay out royalties. They have varying standards on different chains that will force us to implement certain features if we want to play on Ethereum.

Project Priority

After identifying the operational needs of the existing smart contracts, the Lunar Dev Team has made the Lunar Foundation aware of the need to make updates to the contracts. Without these updates, the team will continue to add non-administrative wallets to our bypass lists, continue to improperly divide fee collections and completely miss out on the royalty opportunities from existing NFT collections.

The priorities are driving operational overhead, and as a result, the team would like this project completed as quickly as possible, even if this means adjusting some of the expectations.

Efforts Estimate

To help understand the development efforts necessary to conduct this engagement, we have outlined an estimate of the efforts below.

- **Research, Analysis & Design**
 - Description: ERC-20 Struct Manipulation after deployments, ERC-2981 standards research, OpenSea royalty requirement research,
 - Effort: 14 hrs.
- **ERC-20 Updates**
 - Description: New SafeGuards, New Development wallet, Updates to fee calculations, updates to fee history structs, updates to transaction tracking processes
 - Effort: 24 hrs.
- **ERC-721 Updates**
 - Description: ERC-2981 EIP research, OpenSea royalties standard research, royalty percentage implementation
 - Effort: 24 hrs.

- **Tests**
 - Description: Unit test and internal integration testing
 - Effort: 12 hrs.
- **Deployment & Validation**
 - Description: Deployment to beta and production environments, user acceptance testing, acquire final sign-off
 - Effort: 20 hrs.

Contract Update Requirements

Business Objectives

This section outlines what the Lunar Foundation is attempting to accomplish by making the suggested changes.

Objective #1: ERC-20 - Expand Fee Distribution Granularity

Objective
Description <p>The Lunar Foundation has expressed a desire to create a separate fee on the ERC-20 contract to set aside a specific budget for developing the Lunar platform. Such a capability will allow the Lunar DAO to better understand what funds are available to the Dev Team. To make that happen, we must expand the fee collection structures on the contract to support another wallet and adjust the calculations accordingly.</p>
Current Shortcomings <p>The existing ERC-20 Token contract does not support separating fees to fund Lunar Dev Team expenses. This shortcoming is preventing the Dev Team from having access to a specific dedicated budget that cannot be used for other purposes.</p>
Outstanding Questions <p>N/A</p>

Objective #2: ERC-20 - Support Larger Wallets for Centralized Exchanges

Objective
Description
<p>We currently support AllowLists for a limited number of wallets to bypass transaction fees and anti-whale guards.</p> <p>Certain exchanges, specifically MEXC, have required us to give them the ability to hold LNR quantities that exceed our current anti-whale limits. To solve that problem today, we've had to add those wallets to the ERC-20 "internal" wallet list, which is functionally correct but logically incorrect and could lead to future confusion about wallet management.</p> <p>To resolve this operational issue, we will need to establish a new set of safeguards and bypass lists to specifically support centralized exchanges.</p>
Current Shortcomings
<p>The current contract does not support the separation of administrative wallets and those considered to be maintained / managed by CEXs. This shortcoming is resulting in the pollution of our 'internal' bypass list and may result in erroneous bypassing in the future.</p>
Outstanding Questions
<p>N/A</p>

Objective #3: ERC-721 - Enforce NFT Royalties

Objective
Description
<p>During the requirements gathering process for the ERC-721 tokens, Lunar DeFi and the Dev Team decided to not build creator royalties into the contract, because those fees are typically tacked on at the Marketplace level.</p> <p>OpenSea's surprise support for the BNB Chain has forced us to expand our understanding of the royalty ecosystem and listing with them requires us to support certain capabilities that are not currently present in the contract.</p> <p>To future-proof the contract as we work to bring the Lunar Mint platform to market, and to potentially capture additional chain-native revenue, we are adding these capabilities now while we're updating other Solidity code and tests.</p>
Current Shortcomings
<p>Royalty features do not exist on any version of the upgradeable smart contract and therefore royalties must be managed in each individual marketplace. This shortcoming makes the Lunar Foundation reliant on 3rd party marketplaces and means that if an NFT holder lists their NFT on a marketplace we don't support, we won't collect royalties on the transaction.</p>
Outstanding Questions
<p>N/A</p>

Implementation Objectives

This section describes the process by which we will implement the Business Objectives, the potential issues we might encounter along the way, and how we intend to mitigate those issues.

Objective # 1: ERC-20 - Establish Development Wallet and Modify Fee Structures

Objective
Description
We must establish a new Development Wallet to help gather operational fees for the Development Team. We must then also modify the existing fee structures to allow the development wallet to reside within the fee transaction calculations. And we must do so without breaking the existing Solidity memory space.
Current Shortcomings
The current ERC20.sol Lunar contract does not have a Development Wallet and therefore operations costs are being shared only between marketing, operations, and liquidity wallets. This fee structure prevents the Engineering team from receiving the necessary funds to operate.
Issues & Mitigations
Issue: Setting up a new wallet address in the Smart Contract will require changes to the current memory space as the FeeRates and FeeHistory tracking is stored via a Structs on the contract. We must ensure that adding a new property to the Struct will not corrupt the memory space and potentially comprise the entire contract.
Mitigation: Conduct proper research to ensure that changes to an existing struct will not have adverse consequences against the Smart Contract memory space. This may require us to re-work the entire data structure to avoid these issues in the future.
Implementation Details
We intend to incorporate a new wallet for “Development” and add the appropriate fee structure as necessary. <ul style="list-style-type: none">• Add Development Address• Update Fee Structures
Outstanding Questions
N/A

Objective # 2: ERC-20 - Implement New Safeguards for Centralized Exchanges

Objective
Description
We will add a new type of safeguard to the contract to differentiate exchange wallets from other wallets and set maximum sizes for wallets and transactions. We will also expand the unit tests to support complete code coverage against these additional safeguards.
Current Shortcomings

The current anti-whale guards are too restrictive to support the expanded requirements of some of the exchanges we work with. Therefore, we are having to utilize bypass lists in ways that were not intended, and those exchange wallets may end up with additional capabilities they are not supposed to have.

Over time, this could become a larger security risk, and will also pollute our bypass lists and make them unmanageable.

Issues & Mitigations

Issue: We must not forget to remove the existing exchange wallets from the Internal list as a part of this process.

Mitigation: Make sure a task is created as a part of the process to also remove the MEXC wallet from the internal bypass list.

Implementation Details

We intend to incorporate a new set of SafeGuard structures to allow for exchanges and organizations to be set on different rate and calculated fee structures.

- ERC-20 Token Address:
 - <https://bscscan.com/address/0xc1a59a17f87ba6651eb8e8f707db7672647c45bd#readProxyContract>

Outstanding Questions

N/A

Objective # 3: ERC 721 - Research Royalty Standards for OpenSea

Objective
Description
In our initial review of OpenSea's support for the BNB Chain, we found they have specific requirements regarding royalty implementations. We must research OpenSea's business practices vs industry standards to understand how these variables affect our platform operations moving forward.
Current Shortcomings
We have not utilized OpenSea's Marketplace on the Ethereum chain and therefore the standards set forth by their Agreements are not clear and may have negative impacts on NFT collection holders.
Issues & Mitigations
Issue: Improper implementation of royalties could result in negative publicity associated with an entire NFT collection, dramatically impacting the value of any NFT asset under a given NFT collection.
Mitigation: Conduct proper research to ensure that implementation of Royalties will not contradict the existing standards set by OpenSea and general royalty standards.
Implementation Details
We will be researching the standards associated to OpenSea and general royalty standards to ensure a proper implementation of royalty capabilities. <ul style="list-style-type: none">• Research OpenSea requirements for proper royalty implementation<ul style="list-style-type: none">◦ https://twitter.com/sofiagarcia_io/status/1600151036273926144?s=46&t=SgaoWRf3UDaLO_t8dlwnEg
Outstanding Questions
N/A

Objective # 4: ERC-721 - Implement EIP-2981 Royalty Standards

Objective
Description
We must implement royalty standards already set by the Ethereum Improvement Proposal (EIP) process to properly enable royalty features in our ERC-721 contract and tokens.
Current Shortcomings
We do not support any native royalty capabilities in the existing ERC-721 contract. This shortcoming prevents the Lunar Foundation from enforcing royalty payments on every NFT marketplace.
Issues & Mitigations
Issue: In this process, we may unintentionally implement in ways that actually reduce the royalties collected or may conflict with 3 rd party marketplaces.
Mitigation: Properly research the necessary steps to implement EIP-2981 aligned royalty standards while also conforming to the OpenSea royalty standards and other royalty programs.
Implementation Details
We will be researching the standards associated with common royalty transaction execution and how those exist alongside OpenSea standards. <ul style="list-style-type: none">• EIP-2981 standards<ul style="list-style-type: none">◦ https://eips.ethereum.org/EIPS/eip-2981#optional-royalty-payments
Outstanding Questions
N/A

Contract Update Implementation Details

ERC-20 Update Details

Here's a breakdown of the specific modifications that shall occur as part of the latest upgrade to the ERC-20 Lunar Smart Contract.

Structures

Capability
Description
Structures allow us, as Solidity developers, to craft class-like definitions within a given Smart Contract. We would be adding a new Structure for our "Exchange" based Rates and Calculated Fee values.
Use Case
Structures like the existing ones in our Lunar ERC-20 Smart Contract: <ul style="list-style-type: none">• https://bscscan.com/address/0xc1a59a17f87ba6651eb8e8f707db7672647c45bd#readProxyContract
Research
We intend to implement the new "Structures" by adding a new instance of "Exchange" structures: <ul style="list-style-type: none">• Structs -> https://solidity-by-example.org/structs/• Existing Structs -> https://bscscan.com/address/0xb8448630a74ad7e871265ae661b8c3e470f7b5a4#code#F1#L250

ERC-721 Update Details

Here's a breakdown of the specific modifications that shall occur as part of the latest upgrade to the ERC-20 Lunar Smart Contract.

OpenZeppelin Royalty Standards

Capability
Description
OpenZeppelin already provides a clear path to implementing royalty capabilities in upgradeable smart contracts. These royalty standards derive from the original Ethereum Improvement Proposal (EIP), EIP-2981.
Use Case
Implementing Royalties via OpenZeppelin's standard contracts: <ul style="list-style-type: none">• https://bscscan.com/address/0xc1a59a17f87ba6651eb8e8f707db7672647c45bd#readProxyContract
Research
Implement EIP-2981 using OpenZeppelin's Standard Smart Contract for ERC-721, already available here: <ul style="list-style-type: none">• https://github.com/OpenZeppelin/openzeppelin-contracts-upgradeable/blob/master/contracts/token/ERC721/extensions/ERC721RoyaltyUpgradeable.sol We cannot simply set a percentage on every transfer as defined by the Finalized EIP here. <ul style="list-style-type: none">• https://eips.ethereum.org/EIPS/eip-2981#optional-royalty-payments• Original Quote regarding Royalties:<ul style="list-style-type: none">o It is impossible to know which NFT transfers are the result of sales, and which are merely wallets moving or consolidating their NFTs. Therefore, we cannot force every transfer function, such as <code>transferFrom()</code> in ERC-721, to involve a royalty payment as not every transfer is a sale that would require such payment. ERC2981Upgradeable.sol is not recommended if you already established an ERC-721, as explained here: <ul style="list-style-type: none">• https://github.com/OpenZeppelin/openzeppelin-contracts-upgradeable/tree/master/contracts/token/common Original, Finalized EIP-2981 https://eips.ethereum.org/EIPS/eip-2981#optional-royalty-payments

Conclusion

By the end of this engagement the customer shall have the ability to collect separate fees for platform development, manage exchange wallets as a separate process, and collect NFT royalties more consistently.

Sign-Off

By signing below, the Lunar Foundation indicates it understands the information presented and has given the Dev Team the go-ahead to proceed as outlined above.

Foundation Sign-off: Marcus Yarbro
[Marcus Yarbro \(Dec 13, 2022 09:39 CST\)](#)
