Technical Smart Contract Architecture Document

Overview

Contract Name: NFTLock
Version: Solidity 0.8.24

Purpose

The `NFTLock` contract is designed to enable the locking of NFTs (Non-Fungible Tokens) for a specified period of time. While it can handle any ERC721 NFT, it is specifically optimized for UniV3 liquidity provider NFTs. The contract allows users to lock their liquidity in the form of NFTs, providing a mechanism to commit liquidity over a predetermined duration. This functionality is particularly useful for liquidity providers who wish to signal their long-term commitment to a liquidity pool.

Features

- Locking mechanism for ERC721 and ERC721 extension NFTs.
- Ability to extend the lock period.
- Transferability option for locked NFTs.
- NFT unlocking after the lock period expires.

Contract Inheritance

`NFTLock` inherits from:

- `ERC721Enumerable`: For enumeration functionality over all NFTs.
- `IERC721Receiver`: To ensure the contract can safely receive NFTs.
- `ReentrancyGuard`: To prevent reentrancy attacks.

Key Structures

Lock

- `address nft`: The address of the NFT being locked.
- `uint256 tokenId`: The token ID of the NFT being locked.
- `uint256 unlockDate`: The timestamp when the NFT can be unlocked.
- `bool transferable`: Indicates if the locked NFT can be transferred.

Main Functions

Constructor

Initializes the contract with a name and symbol for the lock tokens.

lockNFT

Allows a user to lock an NFT by transferring it to the contract. The function requires the NFT's owner to approve the transfer in advance.

onERC721Received

Handles the receipt of an NFT, enabling the locking process through the `safeTransferFrom` method. It decodes the provided data to extract locking parameters.

extendLock

Permits the extension of an NFT's lock period, provided the new unlock date is in the future.

unlockNFT

Enables the unlocking and return of an NFT after the lock period has expired.

Events

- `LockCreated`: Emitted when a new lock is created.
- `LockExtended`: Emitted when a lock is extended.
- `NFTUnlocked`: Emitted when an NFT is unlocked.

Security Features

- `ReentrancyGuard`: Prevents reentrant calls to sensitive functions.
- Ownership checks: Ensures that only the lock owner can extend or unlock the NFT.
- Transferability checks: Enforces the transferability rules set during the locking process.

Potential Use Cases

This contract can be utilized by DeFi projects and liquidity providers looking to incentivize long-term liquidity provision by locking liquidity provider tokens (NFTs). It's particularly suited for protocols using UniV3 NFTs to represent liquidity positions.

Conclusion

The `NFTLock` contract is a robust solution for locking NFTs to secure and incentivize long-term liquidity provision in DeFi protocols. Its integration of ERC721 standards, along with customization options such as transferability and extension of lock periods, offers flexibility and security for both users and DeFi projects.