

Thank you to [@vbuterin](#) for his initial post kick-starting this topic: [Cross-rollup NFT wrapper and migration ideas](#)

Problem

A problem faced within the Ethereum blockchain as well as Layer 2s is the segmentation of assets, especially NFTs. When an NFT is created on a Layer 2, the NFT is stuck on that layer. This leads to the problem of NFT silos, where NFT assets on Layer 2 cannot be withdrawn to Layer 1. Having a way to create and transfer NFTs across multiple layers is important in order to introduce NFT applications to Layer 2s.

Solution

Create a way to easily transfer NFTs across layers and reduce the gas requirements by minting the NFTs on Layer 2. This lets the user decide when they would like to withdraw their NFTs to Layer 1.

Overview Diagram

[

NFT Metis-Initialization - Smart Contracts

798x796 33.1 KB

](<https://ethresear.ch/uploads/default/original/2X/7/7902ea6a682ac628322d0bf43184b23fcd9685f8.jpeg>)

Initializing the NFT Contracts

1. The Owner will provision the L1_NFT_Collection with the deployed L2_NFT_Collection, specifying:
 2. Address of the L2_NFT_Collection
 3. Chain ID of the L2_NFT_Collection
 4. Location of the NFT, mapping of the Chain ID by the NFT ID
 5. Range of Allocated NFTs, e.g. #1-100
- , 60-61, etc.

Range is a unit of 1000 NFTs... rangeid 0 means NFT #0

to NFT #999

.

Data Structure

mapping(chainid => address) addresses; mapping(rangeid => chainid) range_loc;

Claiming the NFT on Layer 2

1. The User will claim the NFT, specifying the NFT ID. The L2_NFT_Collection contract on Layer 2 will verify the ID assignment (i.e. this Layer 2 owns the NFT) and mints the NFT on this rollup accordingly. Users can freely transfer NFTs within the rollup as normal.

Withdrawing an NFT

1. The User sends a transaction to the L2_NFT_Collection smart contract, initiating the withdrawal to a target recipient on Layer 1.
2. The NFT gets deposited into the L2_Deposit contract, which locks the NFT for later retrieval if necessary.
3. The Locked NFT triggers a cross-message to the L1_NFT_Collection which allows the designated recipient to mint/withdraw the NFT on Layer 1.
4. A new record in L1_NFT_Collection will be added to update the chain ownership of this particular NFT, data structure:

```

mapping(id=>chainid) nft_loc;

nft_loc

always overrides range_loc

. To determine the actual chain ownership of a NFT, the logic should check nft_loc
first then range_loc

if nft_loc[id]

returns 0.

```

- If the NFT was already previously minted on Layer 1 before, i.e. it is not the first time this NFT has been withdrawn to Layer 1, the user would get that NFT with the updated metadata, with its nft_loc

updated.

1. The recipient will claim the NFT, specifying the NFT ID.

(Re)depositing the NFT on Layer 2

1. The User sends a transaction to the L1_NFT_Collection smart contract, initiating the transition to a target recipient on a target Layer 2 rollup.
2. The NFT gets deposited into the L1_Deposit contract, which locks the NFT for later retrieval if necessary.
3. The Locked NFT triggers a cross-message to the L2_NFT_Collection, which allows the user to mint/claim an existing NFT on Layer 2 by the NFT ID.
4. If the NFT was already created and was deposited to the target rollup, the user would receive the NFT with the updated metadata.
5. L1_NFT_Collection will have nft_loc

updated to reflect the updated chain ownership of the NFT.

Sequence Diagrams

Provisioning

[
 NFT Owner Sequence
 776×304 27.6 KB
](<https://ethresear.ch/uploads/default/original/2X/7/7433ba110be4749bfd03e7e0bdbd1c16dde22052.jpeg>)

Layer 1 => Layer 2

[
 NFT Sequence - L1 to L2
 1684×644 91.3 KB
](<https://ethresear.ch/uploads/default/original/2X/7/71d6af2af34c861cdd94dd4201b61c798d591c60.jpeg>)

Layer 2 => Layer 1

[
 NFT Sequence - L2 to L1
 1684×642 89.6 KB
](<https://ethresear.ch/uploads/default/original/2X/5/5c69dcd0310566bddd5ec9279489704d639a459f.jpeg>)

Pros & Cons

Pros

- Allows any user to claim an NFT on another Layer 2.
- The User can withdraw the NFT on Layer 2 and have the equivalent NFT created on Layer 1 with no chance of duplication.
- The User has the option to transfer their created NFT at any time from Layer 1 to Layer 2 and vice versa.
- Allows one NFT project to extend to multiple rollups.
- Easy and low cost initial setup.

Cons

- The Owner has to create an equivalent NFT contract on the supported layers.
- NFTs can only move from Layer 2 to the coordinator chain (Layer 1 in this case) and back, they cannot move from Layer 2 to Layer 2 directly.
- Higher cost when moving NFTs among rollups because involvement of two Layer 1 transactions (withdraw and deposit)
- Technical complexity to connect both Layers

Extensions

Using a rollup to track the chain ownership to reduce the transaction cost

A rollup can be used to manage the chain ownership. In this way, the transaction cost can be greatly reduced. Layer 1 in this case can be treated the same as other rollups to some extent.