

Development of an ERC404 NFT Marketplace with Cross-Chain Functionality

Krishna Kalaspurkar¹, Prof. Sunil Mane², Prof. Rohini Sarode³

^{1,2,3}*Department of Computer Engineering Pimpri Chinchwad College of Engineering Pune, India*

Abstract—This paper presents the design and implementation of an ERC404-based NFT marketplace with cross-chain functionality between Ethereum and Polygon networks. The developed system allows users to mint, trade, and transfer NFTs while supporting fractional ownership through the ERC404 token standard. To enable secure asset transfers across different blockchains, the marketplace uses the Burn-to-Claim mechanism, which ensures that NFTs are transferred atomically without relying on centralized intermediaries. The system consists of Solidity smart contracts, a React-based frontend integrated with MetaMask, and a custom JavaScript bridge script for handling cross-chain operations. The implemented marketplace supports core functionalities such as NFT minting, listing, buying, cancellation, fractional redemption, and cross-chain transfers. Experimental testing shows that the system performs these operations successfully with reasonable transaction time, throughput, and gas consumption. The results demonstrate that ERC404, combined with cross-chain interoperability, can be effectively used to build secure, decentralized, and scalable NFT marketplace applications.

Index Terms—ERC404, NFT marketplace, cross-chain, blockchain interoperability, Burn-to-Claim, Ethereum, Polygon, fractional ownership

I. INTRODUCTION

The proliferation of blockchain technology has revolutionized how digital assets are managed, providing immutable, decentralized, and transparent platforms for transactions and ownership. A persistent challenge, however, is that most blockchain networks operate as isolated silos incapable of moving assets and data across heterogeneous networks. This lack of interoperability hinders composability, restricts scalability, and often leads to high transaction costs especially in the rapidly expanding NFT (Non-

Fungible Token) industry. Established NFT marketplaces typically rely on ERC721, which, although robust, does not natively facilitate fractional ownership and is predominantly single-network, contributing to high fees and limited accessibility on Ethereum. The ERC404 standard represents a critical innovation in this space. By merging the properties of ERC20 (fungible) and ERC721 (non-fungible) tokens, ERC404 introduces native fractionalization in the ownership and trading of NFTs, opening new opportunities in Web3. Its theoretical potential for platform-level support and secure cross-chain functionality remains largely unexplored.

In response to these challenges, this paper presents the design, development, and empirical evaluation of an ERC404-based NFT Marketplace with cross-chain asset transfer between Ethereum and Polygon. The implementation is inspired by the Burn-to-Claim protocol a system that employs self-verifiable proofs and cryptographic tools to ensure security, correctness, and atomicity when transferring assets by burning them on a source chain and re-issuing them on a destination chain. This approach does not require external middleware or trusted third parties, achieving trustless and decentralized interoperability.

The marketplace enables effective minting, listing, trading, bridging, and redeeming of fractional NFTs via smart contracts, a user-friendly React interface, and a custom bridge script. Performance metrics including transaction time, throughput, latency, and gas consumption confirm that the system supports practical, safe interoperability. This work is intended as a blueprint for future blockchain applications that not only build on theoretical foundations but deploy them in operational systems.

The remainder of this paper is organized as follows. Section II presents a review of existing research on blockchain interoperability, cross-chain protocols, and

ERC404-based NFT marketplaces to provide the background for this work. Section III describes the overall system architecture, including the major components of the marketplace, smart contracts, frontend application, and cross-chain bridge mechanism. Section IV explains the implementation details, covering the deployment process, smart contract functionalities, and the logic used for cross-chain asset transfers. Section V presents the workflow of the proposed system through a high-level algorithm describing the complete NFT lifecycle. Section VI discusses the experimental results and performance analysis, including transaction time, throughput, latency, and gas consumption for different marketplace operations. Finally, Section VII concludes the paper by summarizing the key contributions of the implemented marketplace and outlines possible directions for future improvements.

II. LITERATURE REVIEW

A. Blockchain Interoperability and Cross-Chain Protocols

Blockchain interoperability has been a primary research concern, aimed at enabling the exchange of assets, data, and services between incompatible networks. By design, blockchains are isolated ledgers, aggravating cross-network composability and creating “value islands.” Early approaches include:

Sidechains: Auxiliary chains (e.g., Bitcoin’s Liquid Network) enable asset transfers between a primary chain and sidechains via a two-way peg, adding scalability and privacy. Reliance on third-party custodians, however, raises concerns about decentralization and safety.

Hashed Time-Lock Contracts (HTLCs): HTLCs enable atomic swaps via cryptographic locks, guaranteeing binary security on chain interchange. They require compatible primitives, which can be challenging in heterogeneous ecosystems.

Blockchain Bridges: Projects such as the Polygon Bridge and Wormhole link networks to facilitate asset and information transfer. However, bridges remain vulnerable to attacks; the \$320 million Wormhole exploit demonstrated that existing bridge designs can present significant security risks.

Relay Chains: Relay chains such as Polkadot and Cosmos enable secure communication and asset transfer among connected blockchains using protocols

like the Inter-Blockchain Communication (IBC) standard.

Despite these developments, persistent barriers to smooth blockchain interoperability remain: lack of universal standards, inconsistent smart contract languages, security threats, and governance complexities [10].

B. Cross-Chain Burn-to-Claim Protocol

The Burn-to-Claim protocol provides a self-verifiable, trust-less mechanism for cross-blockchain asset transfer without middleware, third-party validators, or intermediate tokens. It employs a two-phase system:

- 1) Exit Transaction: Assets are irreversibly burned on the source chain, and cryptographic, time-locked proofs are generated to verify the state transition.
- 2) Entry Transaction: The proofs are submitted to the destination chain, which validates them and reconstructs the asset or its representation.

Burn-to-Claim offers strong security, correctness, and atomicity guarantees, addressing concerns such as double-spending. Unlike earlier proof-of-burn constructs, it includes a reclaiming feature activated by time-locks, allowing senders to retrieve assets if the transfer does not complete within a set period [10].

C. NFT Marketplaces and ERC404

The NFT market has grown exponentially, enabling tokenization of digital assets such as art, collectibles, and in-game items on platforms like Ethereum. Traditional NFT standards (ERC721), while robust, are limited by the absence of native fractional ownership and liquidity, and by single-network operation resulting in high fees and reduced accessibility. The ERC404 standard addresses these limitations by combining ERC20 and ERC721 properties to allow native NFT fractionalization, dividing NFTs into fungible shares that improve market liquidity [5]. Early experiments such as Pandora [9] demonstrate how ERC404 can pave the way for new DeFi opportunities and broader user access.

III. SYSTEM ARCHITECTURE

A. Overview

Fig. 1 illustrates the overall ERC404 NFT Marketplace architecture, showing the major system components and their interconnections across the

Ethereum local (ethLocal) and Polygon local (polygonLocal) networks.

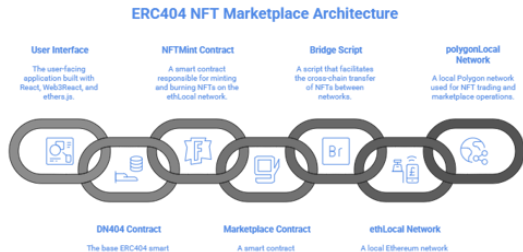


Fig. 1. ERC404 NFT Marketplace Architecture showing the User Interface, smart contracts (DN404, NFTMintDN404, NFTMarketplace), Bridge Script, and local network deployments.

Fig. 2 presents the system flow diagram, illustrating user interactions from wallet connection through minting, listing, buying, canceling, and burning NFTs, and the corresponding smart contract invocations.

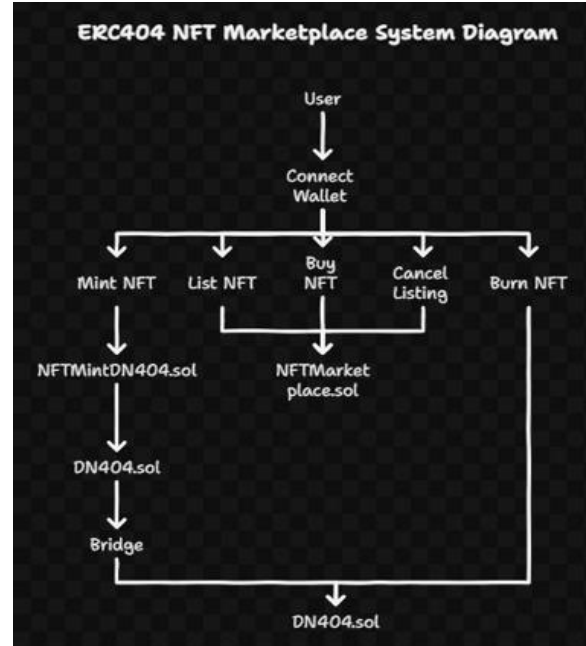


Fig. 2. ERC404 NFT Marketplace System Diagram depicting the interaction flow between the user, frontend, and smart contracts (NFTMintDN404.sol, NFTMarketplace.sol, DN404.sol) and the cross-chain bridge.

B. Smart Contracts

The core programmable logic is written in Solidity and deployed to local Ethereum and Polygon blockchains:

- DN404.sol: Manages the hybrid ERC20/ERC721 (ERC404) token logic, supporting standard NFT operations and fractionalization.
- NFTMintDN404.sol: Extends minting and burning capabilities and implements the Burn-to-Claim mechanism, allowing users to redeem ERC721 NFTs by burning ERC20 tokens.
- NFTMarketplace.sol: Manages NFT listing, purchasing, and approval flows, as well as the trade lifecycle.

C. Frontend Application

A React-based web application uses Web3React and ethers.js to connect to the user’s MetaMask wallet, inter-act with contracts, and facilitate seamless network switching between ethLocal and polygonLocal.

D. Bridge Script

A custom JavaScript script (bridge.js) supports cross-chain NFT transfers by: monitoring the source chain for burn events; triggering minting and claiming on the destination chain via proof relaying; and coordinating lock/burn and mint/redeem operations across contracts on both networks.

IV. IMPLEMENTATION DETAILS

A. Local Networks and Deployment

Two local blockchain environments were used:

- ethLocal: Ethereum local network (Chain ID 1337) for primary ERC404 NFT minting and issuance.
- polygonLocal: Polygon local network (Chain ID 80002) for bridging and trading with lower gas costs.

Deployment was managed using Hardhat:

1. deploy DN404, NFTMintDN404, and NFTMarketplace on ethLocal;
2. mirror contracts on polygonLocal;
3. configure bridge.js with network endpoints.

B. Smart Contract Features

- Minting: Creates unique (ERC721) and fractional (ERC20) tokens simultaneously.
- Listing and Approvals: Users grant marketplace approval via standard or permit functions to reduce gas.
- Buying and Trading: Supports purchase of whole or fractional NFTs with automated ownership transfers.

- Cancellation: Sellers may withdraw listings and restore tokens to their wallets.
- Burn-to-Claim: Fractional ERC20 holders burn tokens to redeem the underlying ERC721 NFT.

C. Cross-Chain Bridging Logic

NFTs are locked or burned on the source chain and a transaction indicator is relayed to the destination chain via the bridge script. Upon successful verification, a corresponding NFT is minted or transferred on the destination chain. Atomicity is ensured via hash-time-locks and validation, guaranteeing all-or-nothing transfers. The bridge script listens for relevant on-chain events (e.g., Burn, Mint) and triggers corresponding actions on the destination chain.

V. WORKFLOW AND PSEUDOCODE

Algorithm 1 presents the high-level workflow of the ERC404 NFT Marketplace, covering all major user operations.

VI. RESULTS AND OBSERVATIONS

A. Performance Metrics

Table I summarizes the measured performance metrics across all major marketplace operations on ethLocal and polygonLocal. Gas fee estimates based on Polygon Amoy pricing (40 gwei/gas).

Table I: Performance Metrics for ERC404 NFT Marketplace Operations

Operation	Time	TPS	Lat. (s)	Gas	USD
Minting	~0.5 s	2-10	0.1-0.5	~200k	\$0.0032
Listing	2-3 s	0.5-2	2-3	~500k	\$0.0084
Buying	~0.5 s	2-10	0.1-0.5	~150k	\$0.0024
Cancel	~0.5 s	2-10	0.1-0.5	~100k	\$0.0016
Bridging	1-2 s	1-5	1-2	~400k	\$0.0064

B. Analysis

Minting and buying operations completed in approximately 0.5 seconds, achieving 2-10 TPS. Listing throughput was lower (0.5-2 TPS) due to multi-step approval sequences. Cross-chain bridging achieved 1-5 TPS over 1-2 seconds. The hybrid ERC404 logic consumed approximately 30% more

gas than standard ERC721 operations. The Burn-to-Claim mechanism functioned correctly, allowing fractional token holders to redeem underlying ERC721 NFTs securely. End-to-end lifecycle tests were successful across both local networks.

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper demonstrated the feasibility of the ERC404 NFT Marketplace with cross-chain functionality by deploying smart contracts for minting, listing, fractionalization, and trading, a React frontend, and a custom bridge script across local Ethereum and Polygon networks. Performance testing confirmed efficient operation, with minting and purchasing taking ~0.5 s, listing 2-3 s, and bridging 1-2 s, at throughputs up to 10 TPS. Gas costs were reasonable, though ERC404's hybrid logic added ~30% overhead relative to ERC721. The Burn-to-Claim mechanism enabled atomic cross-chain interactions.

Algorithm 1 ERC404 NFT Marketplace Workflow

Require: User wallet, NFT contract address, amount, listing ID, price

Ensure: NFT minted, bridged, listed, bought, canceled, or claimed

- 1: CONNECT wallet via Web3 provider (MetaMask)
- 2: if connection fails then
- 3: DISPLAY error; RETURN
- 4: end if
- 5: SELECT network
- 6: if network = ethLocal (1337) then
- 7: CALL mintNFT(contractAddress, amount)
- 8: if mint fails then
- 9: DISPLAY error; RETURN
- 10: end if
- 11: OUTPUT "NFT minted with ERC20 tokens"
- 12 if bridge requested then
- 13: CALL bridgeNFT(contractAddress)
- 14: if bridge fails then
- 15: DISPLAY error; RETURN
- 16: end if
- 17: OUTPUT "NFT bridged to polygonLocal"
- 18: end if
- 19: else if network = polygonLocal (80002) then
- 20: INPUT action ∈ {list, buy, cancel, burn}
- 21: if list then
- 22: CALL setApprovalForAll, nftApprove,

```

setNFTContract
23: CALL listWithPermit(contractAddress, price,
listingID)
24: OUTPUT "NFT listed"
25: else if buy then
26: CALL buyNFT(listingID, price);
OUT-PUT "NFT purchased"
27: else if cancel then
28: CALL cancelListing(listingID);
OUT-PUT "Listing canceled"
29: else if burn then
30: CALL burnToClaim(contractAddress, amount);
OUTPUT "ERC721 NFT claimed"
31: end if
32: else
33: DISPLAY error "Invalid network"; RETURN
34: end if
    
```

The project advances the state of the art in applying new NFT token standards and cross-chain protocols to practical Web3 applications.

B. Future Work

Key future directions include: (1) deployment to public testnets and mainnet; (2) gas optimization via batching and signature aggregation; (3) extension of the bridge to multiple chains with decentralized relayer incentives and on-chain event proofs; (4) rigorous security audits under adversarial conditions; and (5) improved user authentication and support for additional NFT standards.

REFERENCES

- [1] Nakamoto, S., "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [2] Buterin, V., "Ethereum: A next-generation smart contract and decentralized application platform," Ethereum White Paper, 2013.
- [3] Wood, G., "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Yellow Paper, 2014.
- [4] Entriken, W., Shirley, D., Evans, J., and Sachs, N., "ERC-721: Non-Fungible Token Standard," Ethereum Improvement Proposal (EIP-721), 2018.
- [5] "ERC-404: Hybrid NFT Standard," Official Documentation, 2024.
- [6] OpenZeppelin, "OpenZeppelin Contracts: Smart contract library and security," 2024. [Online]. Available: <https://docs.openzeppelin.com>
- [7] Hardhat, "Hardhat development framework documentation," 2024. [Online]. Available: <https://hardhat.org>
- [8] Polygon, "Bridging resources between Ethereum and Polygon," Polygon Documentation, 2024. [Online]. Available: <https://docs.polygon.technology>
- [9] Pandora Labs, "Pandora ERC-404 NFT Project," 2024.
- [10] Pillai, B., Biswas, K., Hao, Z., and Muthukkumarasamy, V., "Burn-to-Claim: A blockchain interoperability protocol," Computer Networks, vol. 200, Art. no. 108495, 2021.
- [11] Vogelsteller, F., and Buterin, V., "ERC-20 Token Standard," Ethereum Improvement Proposal (EIP-20), 2015.
- [12] Radomski, W., et al., "ERC-1155 Multi Token Standard," Ethereum Improvement Proposal (EIP-1155), 2019.
- [13] MetaMask, "MetaMask developer documentation," 2024.
- [14] ethers.js, "Ethereum JavaScript library," 2024.
- [15] React, "React official documentation," Meta Platforms, 2024.
- [16] Solidity, "Solidity programming language," Ethereum Foundation, 2024.
- [17] Web3.js, "Web3.js documentation," 2024.
- [18] Nomic Foundation, "Hardhat network documentation," 2024.
- [19] OpenSea, "The world's largest NFT marketplace," 2024.
- [20] Polygon Labs, "Polygon Proof-of-Stake architecture," 2024.