Technical Documentation for the `MolNFT v1.0` Smart Contract

Storing Experimental Molecular Data On-Chain (GenesisL1)

by M

December 30, 2024

Contents

1 Introduction

This document describes a decentralized system for archiving and querying molecular data, relying on the **GenesisL1** blockchain for trustless storage and retrieval. The `MolNFT` smart contract is designed to handle expansive repositories such as the Protein Data Bank (PDB)—which grows annually—and other specialized or private molecular databases. By placing data on a permissionless ledger, research in biomedical and medical fields gains an auditable record of structures, including giant complexes with millions of atoms.

1.1 Motivation

Scientists routinely require secure, tamper-proof ways to share biomolecular findings. Centralized repositories can be inaccessible at times, or subject to unexpected alterations. This contract places the entire *bcif*-formatted PDB and other molecular archives into a globally shared space, fostering collaboration through:

- **Immutable Transactions:** Once uploaded, records cannot be covertly modified, preserving scientific integrity.
- **Trustless Accessibility:** Researchers worldwide interact without reliance on a singular gatekeeper.
- **Web3 Synergy:** EVM-compatible tooling connects on-chain data to a broader ecosystem of *decentralized applications*, enabling further innovation in bioinformatics and other fields.
- **Community Verification:** Contributions and updates remain transparent to all, lowering barriers to peer review.

2 Contract Overview

The `MolNFT` contract follows ERC721 standards, enhanced with features to handle large molecular data. Its *parent-child NFT structure* allows chunking of giant *bcif* files, giving each dataset the flexibility to span multiple tokens while maintaining logical unity. Key benefits include:

- **Parent-Child Linking:** Splits massive structures across multiple NFTs to avoid size limits.

- **On-Chain Metadata:** Keeps essential PDB fields, sequences, and descriptive text directly on the ledger.
- **Base64-Encoded Files:** Ensures entire data archives can be retrieved without centralized hosting.
- **On-Chain Search**: Metadata-based searching allows direct lookup on **GenesisL1**, fueling a new wave of EVM-based bioinformatics dApps.

3 Key Data Structures and Variables

3.1 `NFTData` Struct

```
struct NFTData {
string IDCODE;
string HEADER;
string ACCESSION_DATE;
string COMPOUND;
string SOURCE;
string AUTHOR_LIST;
string RESOLUTION;
string EXPERIMENT_TYPE;
string SEQUENCE;
string imageBase64;
string fileBase64;
}
```

Each token has fields for PDB metadata (`IDCODE`, `ACCESSION_DATE`, `RESOLUTION`, etc.) and Base64-encoded `bcif` data. These records ensure reproducibility of experimental findings and can expand to fit other databases (e.g., specialized private datasets in biomedical or medical research).

3.2 Token Identifiers

- **Parent Tokens (ID $<$ 100,000,000):** Represent the primary or "root" PDB entry. Indexed by `nextNFTId` in ascending order.
- **Child Tokens (ID $\geq$ 100,000,000):** Contain additional *bcif* fragments or extended data. Indexed by `nextChildId`.

3.3 Mappings and Arrays

- `mapping(uint256 => NFTData) private nftData;`
  Holds the metadata struct for each NFT.

- `mapping(uint256 => uint256[]) private children;`
  Maintains child tokens that belong to a parent token.

- `mapping(uint256 => uint256) private parent;`
  Indicates the parent token for each child ID.

- `uint256[] private allTokens;`
  Stores all token IDs for direct enumeration and searching.

4 Functionalities

4.1 `mintNFT`

```
function mintNFT(
...
uint256 parentId
) external
```

Creates a new NFT. If `parentId` is zero, it becomes a parent token. Otherwise, it appends a child token under an existing parent. This flexibility accommodates large PDB entries that cannot be stored in a single NFT.

4.2 `tokenExists`

```
function tokenExists(uint256 tokenId) public view returns (bool);
```

Validates whether a given token ID has been minted. Internally calls `ownerOf(tokenId)` with a try/catch.

4.3 `tokenURI`

```
function tokenURI(uint256 tokenId) public view returns (string memory);
```

Provides JSON-encoded data describing the NFT, including an embedded `imageBase64` field. Useful for NFT marketplaces or dApps that visualize molecular structures.

4.4 `updateMetadata`

```
function updateMetadata(... ) external;
```

Allows the owner to revise data if new experimental evidence arises, a key advantage for evolving molecular research and continuous PDB expansions.

4.5 `getChildren` & `getChildrenPaginated`

```
function getChildren(uint256 parentId) external view;
function getChildrenPaginated(...) external view;
```

Retrieves all children for a given parent, with optional pagination to handle long lists of fragments.

4.6 `getParent`

```
function getParent(uint256 tokenId) external view returns (uint256);
```

Returns the parent token ID of a child token, aiding relationship exploration.

4.7 `getCombinedData` and `getEntireNFT`

```
function getCombinedData(uint256 parentId) external view;
function getEntireNFT(uint256 parentId) external view;
```

`getCombinedData` concatenates parent and child `fileBase64` fields, assembling the entire structure. `getEntireNFT` likewise returns all core metadata plus a fully merged `bcif` string. This approach supports extremely large PDB structures and yearly expansions.

4.8 Search Functions

Built-in queries (`searchByIDCODE`, `searchByCOMPOUND`, etc.) allow direct metadata lookups on-chain. The contract also supports paginated searches to mitigate gas usage.

5 Scientific and Technical Benefits

5.1 Decentralized and Immutable

The **GenesisL1** blockchain ensures data remains tamper-proof and globally available. Once minted, each structure's authenticity is preserved, offering strong evidence for any data claims.

5.2 Web3-Ready

On-chain metadata can integrate with web3 platforms and EVM-based dApps. Developers can incorporate these NFTs into broader bioinformatics pipelines or combine them with analytic tools running on sidechains or off-chain computation layers.

5.3 Detailed Versioning and Transparency

As data evolves, researchers can maintain updates via transactions, leaving a permanent trail of changes. This record helps peer reviewers or collaborating labs understand how structures, sequences, or authorship info progressed over time.

5.4 Trustless Access to Private Repositories

Though the public PDB is a flagship use case, the same mechanism supports private data as well. Laboratories that wish to secure their specialized structures on a permissionless ledger can do so, ensuring controlled or open access based on ownership and NFT-based permissions.

6 Usage and Best Practices

6.1 Storing Entire Molecular Data Repositories

1. **Mint Parent NFT**: Record main fields, store initial `bcif` chunk.
2. **Mint Child NFTs**: Append more fragments for large or frequently updated structures.
3. **Search and Retrieval**: Use the `searchBy*` functions to locate entries, or `getEntireNFT` to reconstruct a full dataset.

Given annual expansions to the PDB and potential integration with other molecular databases, chunked uploading ensures near-limitless growth.

6.2 Future-Proofing

To deal with file size and gas considerations, efficient compression with `bcif` is advisable. Additional child tokens can be minted incrementally, mitigating one-time deployment costs.

7 Security and Gas Considerations

7.1 Gas Costs for On-Chain Storage

Uploading multi-megabyte scientific data to EVM-based systems carries high cost. This contract's parent-child design helps distribute these costs over time while preserving a cohesive reference.

7.2 Permission Settings

`onlyDeployerCanMint` can restrict minting if you wish to curate or manage the data. Alternatively, leaving it open fosters a broader, community-driven upload environment.

7.3 Upgrade Paths

If new compression methods emerge or data formats change, the ability to update metadata or add new child tokens allows adaptation without requiring a contract redeployment.

8 Conclusion

MolNFT introduces a robust on-chain framework for storing and managing large-scale molecular data on the **GenesisL1** blockchain, supporting both growing public databases and niche private repositories. Researchers benefit from a system that is verifiable, censorship-resistant, and fully integrated with the wider web3 ecosystem. This approach has direct implications for biomedical, medical, and broader scientific research, offering a novel path toward globally accessible, tamper-evident data archives. Through chunked storage, metadata search, and EVM-based interoperability, **MolNFT** aligns with a future where essential molecular knowledge remains unbounded, trustless, and fully transparent.