CIP 25 - NFT Metadata Standard

Contents

Authors

Prof Philip Schlump, Assistant Director, University of Wyoming Center for Blockchain and Digital Innovation, email:pschlump@uwyo.edu

Andrew Thornhill, TODO-your-title, email:andrew@itconsult.ge

Support

Work on this standard is supported by NFT-DAO, LLC and funds from the Cardano Catalyst Fund.

Abstract

This proposal defines a NFT Metadata Standard for Native Tokens on the Cardano chain and the file format for storage of off-chain digital assets associated with the NFT.

Motivation

Tokens on Cardano are a part of the ledger. Unlike on Ethereum, where metadata can be attached to a token through a smart contract. This isn't possible on Cardano because tokens are native and Cardano uses a UTxO ledger, which makes it hard to directly attach metadata to a token. So the link to the metadata needs to be established differently. By using Cardano's ability to send metadata as a part of a transaction, the necessary linkage between a transaction and the NFT's metadata can be established.

Given a token in an EUTXOma ledger, we can ask "Where did this token come from?" Since tokens are always created in specific forging operations, we can always trace them back through their transaction graph to their origin.

(Section 4.1 in the paper: https://hydra.iohk.io/build/5400786/download/1/eutxoma.pdf)

Considerations

That being said, we have a unique metadata link to a token and can always prove that with 100% certainty. No one else can manipulate the link except if the policy allows it to (update mechanism).

Since the cost of the transaction is determined by the size of the transaction, the amount of data stored on the chain is minimized.

Also digital assettes are not assumed to be single files or a small set of data or a single image. NFTs have a clear application in tying real world data to on chain tokens. For example, the use of NFTs in tracking weld certifications for bridges would combine engendering specifications with a video of the weld and x-ray images. Another example of a NFT would be piano concerto with a `.mp3` file containing the music and the sheet music as a `.pdf` .

Specification

721 NFT or 1155 NFT Metadata included in this transaction.

On Chain Structure

The structure allows for multiple token mints, also with different policies, in a single transaction. This data must comply with the CBOR data representation used in Cardano.

```
        "metadata": {
                "1155": {
                        "<Policy_ID>": {

                                "<nft0>": {
                                        "name": "<required example:TOKEN0>",
                                        "hash": "<requried desc:sha256-hash-of-file>"
                                        "description": "<optional>",
                                        "sample_image": "<optional>",
                                        "location": {
                                                "url": "<required>",
                                                "ipfs": "<optional>",
                                                "arweave": "<optional>"
                                        }
                                },
                                ...
                        },
                        ...
                        "version": "1.0.0"
                }
        }
```

The “1155” is a CBOR data key and referees to the ERC-1155 standard NFT token specification. It is a constant string.

The “<Policy_ID>” is the rules for minting and burning of this kind of token.

“<nft0>” is a unique name within this set of tokens describing the minted item.

“name” is a unique name that more fully describe the NFT.

“location” is the URI or URL of a location to get the off-chain storage for the token. The format of the off chain storage is specified in the next section.

“hash” is the sha256 hash of the off chain storage.

“sample_image” is an appropriate sample image ULR for a representative image for this NFT.

“version” is a version number for this set of tokens and is a reserved constant. Tokens can not be named “version.” Policy_ID can not be “version.”

Token retrieval

Retrieve valid metadata for a specific token

As mentioned above, this metadata structure allows to have either one token, or multiple tokens, with different policies in a single mint transaction. A third party tool can then fetch the token metadata seamlessly. It doesn’t matter if the metadata includes just one token or multiple tokens. The procedure for the third party is always the same:

Find the latest mint transaction with the label 1155 in the metadata of the specific token

Lookup the Policy Id of the token.

Lookup the name of the token and the payload.

This allows you to determine the contents and associated data for the token.

File Structure

The set of files comprising the digital asset for the NFT is combined into a single .zip file with an index.json added that describes the attributes of this NFT.

The file contains a JSON file, `index.json` .

```
{
    "Name": "NFT 0",
    "FilesHash": "sha256-hash-of-concatenated-file-Hash",
    "Title": "The Title of this NFT",
    "Description": "What is this NFT",
    "Creator": "John Q Person",
    "CreationTimestamp": "2010-12-31T15:59:00-07:00",
    "License": "CC-BY-SA-4.0",
    "Location": {
        { "latitude": 42.43,, "longitude": -109.31 }
    },
    "RepFile": 0,
    "AdditionalData": "{\"some\":\"data\"}",
    "Files": [
        {
            "MimeType": "mime type from rfc6838",
            "OrigialFileName": "an-original-file-name.mp3",
            "FileDescription": "Description of this file in set",
            "FileName": "file-in-set.mp3" ,
            "Hash": "sha256-hash-of-file"
        }
    ]
}
```

“Name” is the required name of the NFT and must mach with the “name” field in the NFT metadata.

“FileHash”, a required value, is the sha256 hash of each of the “Hash” fields in the “Files” array concatenated in order. It functions in a similar fusion to a Merkel Hash. The characters in the hash must be in [0-9a-fA-F].

“Title” is a required text title for this NFT.

“Description” is a required text description of what this file is as a part of the digital assets of the NFT.

“Creator” is a person or entity or set of entities that created this NFT.

“CreationTimestamp” is the RFC3339 formatted date when this NFT was created.

“Location” is the geographical location of the NFT if applicable.

“RepFile” is an index to one of the items in the “Files” array that is representative of the NFT. This may be an icon or a reduced size image that is used to display the NFT.

“License”: is an optional text field that describes the license of this NFT.

“AdditionalData” is an optional string with user defined data.

“Files” is an array of objects. The array must contain at least 1 object. The array is order dependent and must match with the order that the “Hash” field in each object is conatenated to produce the “FilesHash.” Each object contains:

“MimeType” the mime type for the file.

“OrigialFileName” the name of the file before it was added to this .zip archive.

“FileDescription” the description of this file in the set. For example “sheet music.”

“FileName” the name of this file in the .zip archive.

“Hash” the sha256 bit hash as a hexidecimal string with characters from [0-9a-f] of the contents of the file.

Backward Compatibility

To keep NFT metadata compatible with changes coming up in the future, we use the version property. Version 1.0.0 is used in the current metadata structure of this CIP.

References

- Mime type: https://tools.ietf.org/html/rfc6838.

- JSON file Standard: https://www.reddit.com/r/CardanoDevelopers/comments/mkhlv8/nft_metadata_standard/
- CIP about reserved labels: https://github.com/cardano-foundation/CIPs/blob/master/CIP-0010/CIP-0010.md
- EIP-721: https://eips.ethereum.org/EIPS/eip-721
- URI: https://tools.ietf.org/html/rfc3986, https://tools.ietf.org/html/rfc2397
- ERC-1155 Standard: https://eips.ethereum.org/EIPS/eip-1155
- CBOR data Standard: https://datatracker.ietf.org/doc/html/rfc8610
- JSON Schema: https://json-schema.org/draft/2020-12/schema
- Timestamp Format: https://datatracker.ietf.org/doc/html/rfc3339

Appendix

Example Transaction creation of NFT

An example transaction

```
#!/bin/bash

curl --request POST \
  --url http://localhost:1337/v2/wallets/5076b34c6949dbd150eb9c39039037543946bdce/trans
  --header 'Content-Type: application/json' \

  --data '{
    "passphrase": "password123",
    "payments": [
        {
            "address": "addr_test1qpg2eglv9gf2rksvdj53t6ajfgzkycaadlt2fatjyn4etpze0592a
            "amount": {
                "quantity": 1000000,
                "unit": "lovelace"
            }
        }
    ],
    "metadata": {
                "1155": {
                        "cbc34df5cb851e6fe5035a438d534ffffc87af012f3ff2d4db94288b": {
                                "nft0": {
                                        "name": "NFT 0",
                                        "location":{
                                                "url": "https://ipfs.io/ipfs/QmUrNv5yof
                                        },
                                        "hash": "d3b71a414945ff13ee4e2b21697ab6ff9a4ff1
                                },
                                "nft1": {
```

```
                                                "name": "NFT 1",
                                                "location":{
                                                        "url": "http://nft-metadata-storeage.s3
                                                },
                                                "hash": "2db31c4b604cac43817e19f174a6d3fa1c12a4
                                        }
                                }
                        }
        }
}'
```

JSON Schema for Validation of index.json

The following is the JSON Schema for validation of the `index.json` file in the .zip archives.

```
{
  "$id": "https://example.com/cip25nftstandardfiledata.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "CIP-25 NFT Standard File Data",
  "required": [ "Name", "FileHash", "Title", "Description", "RepFile", "Files" ],
  "type": "object",
  "properties": {
    "Name": {
      "type": "string",
      "description": "The uqique name of this NFT."
    },
    "FileHash": {
      "type": "string",

      "description": "The sha256 hash of the concatenated hashes of files in the Files
    },
    "Title": {
      "type": "string",
      "description": "The title of this NFT."
    },
    "Description": {
      "type": "string",
      "description": "A description of what this NFT is."
    },
    "Creator": {
      "type": "string",
      "description": "A name or set of names of who created this digital asset."
    },
    "CreationTimestamp": {
      "type": "string",
      "description": "A timestamp in RFC3339 format for when this asset was created."
    },
    "License": {
      "type": "string",
      "description": "A description of how this item is licensed."
    },
    "AdditionalData": {
      "type": "string",
      "description": "A set of user specified data for this token."
    },
        "Location": {
                "required": [ "latitude", "longitude" ],
                "type": "object",
                "properties": {
                "latitude": {
                "type": "number",
                "minimum": -90,
                "maximum": 90
        },
        "longitude": {
                "type": "number",
                "minimum": -180,
                "maximum": 180
        }
          }
        },
    "RepFile": {
      "description": "The position in the Files array of a representative image for thi
      "type": "integer",
      "minimum": 0
    },
        "Files": {
                "type": "array",
        "description": "A non-empty list of files for the NFT token.",
                "items": { "$ref": "#/$defs/files" }

        }
  },

  "$defs": {
    "files": {
      "type": "object",
      "required": [ "MimeType", "OriginalFileName", "FileDescription", "FileName", "Has
      "properties": {
                      "MimeType": {
                        "type": "string",
                        "description": "The mime type of the file in the set."
                      },
                      "OriginalFileName": {
                        "type": "string",
                        "description": "The name that the file was originally stored
                      },
                      "FileDescription": {
                        "type": "string",
                            "description": "A description for this file."
                          },
                          "FileName": {
                            "type": "string",
                            "description": "The name of this file in the set."
                          },
                          "Hash": {
                            "type": "string",
                            "description": "The sha256 hash of the file's data in the set
                          }
          }
        }
  }
}
```

Usage Examples

An example of an `index.json` file:

```
    {
            "Name": "NFT 0",
            "FileHash": "sha256-hash-of-concatenated-file-Hash",
            "Title": "The Title of this NFT",
            "Description": "What is this NFT",
            "Creator": "John Q Person",
            "CreationTimestamp": "TODO",
            "License": "CC-BY-SA-4.0",
            "Location": { "latitude": 42.43, "longitude": -109.31 } ,
            "RepFile": 0,
            "AdditionalData": "{\"some\":\"data\"}",
            "Files": [
                    {
                            "MimeType": "TODO",
                            "OriginalFileName": "TODO",
                            "FileDescription": "TODO",
                            "FileName": "TODO",
                            "Hash": "sha256-hash-of-file"
                    }
            ]
    }
```

Copyright