Export an NFT from Cardano to Ethereum

This process was applied on Test networks for Cardano and Ethereum

Workflow

- Import Cardano NFT metadata
- Create Ethereum accounts
- Deploy an ERC721 smart contract
- Minting a Cardano NFT on Ethereum
- Reading on-chain smart contract data

Import Cardano NFT metadata

- An NFT owner address is needed to start this process

```
In[∘]:= owners =
          {"addr_test1qznxhpnmh2lrr436kyapq290s8qstndkctqrvx3xdulsqcta5nwc39ejfjvduc0gjag⋱
               fkvcj5ln8ldd96hrp3p53r5nsnv0ueu",
           "addr_test1vp7ncuaksvhz9dgrrutpm800c67wu46red407mykk7kyaqsqzt2jr",
           "addr_test1vpnr0japmkm3f488rtckx7mngchmcd99hk0ryhd5ppdlstcqtfuhx"};
```

- Using BlockchainAddressData to have details on AssetBalances

```
In[∘]:= assetData = BlockchainAddressData[First@owners, "AssetBalances",
            BlockchainBase → {"Cardano", "Testnet"}] // First // Dataset
```

Out[∘]=

As	544d504c3031
As	TMPL01
Pc	5c514a19ca505405f12e4e3005a54f945222d112fd3919c30ca94941
Fi	asset1dyzlhg7m7c8g6pp23gw9rk83swawcuehnpwhs8
Qu	1

- Looking the token data by its fingerprint

In[]:=
```
BlockchainTokenData[assetData["Fingerprint"],
  BlockchainBase → {"Cardano", "Testnet"}] // Dataset
```

Out[]=

Fingerprint
asset1dyzlhg7m7c8g6pp23gw9rk83swawcuehnpwhs8

columns 1–10 of **12**

In[]:=
```
tokenDataCardano = BlockchainTokenData[assetData["Fingerprint"],
      BlockchainBase → {"Cardano", "Testnet"}] // First // DeleteMissing // Dataset
```

Out[]=

Fi	asset1dyzlhg7m7c8g6pp23gw9rk83swawcuehnpwhs8
As	5c514a19ca505405f12e4e3005a54f945222d112fd3919c30ca94941544d504c3031
Pc	5c514a19ca505405f12e4e3005a54f945222d112fd3919c30ca94941
As	544d504c3031
As	TMPL01
To	{ …1 }

- "TokenMints" has all the transaction ID of the minting token,

In[]:=
```
txIDCardanoMinting = tokenDataCardano["TokenMints"] // First // #["TransactionID"] &
```

Out[]=

```
a448740f17acb352e031e21ea34c5459dec2ad8da268392e64377900a97e23de
```

- BlockchainTransactionData can provide the token metadata using the previous transaction ID and the Policy ID. In this example, pick only the first one

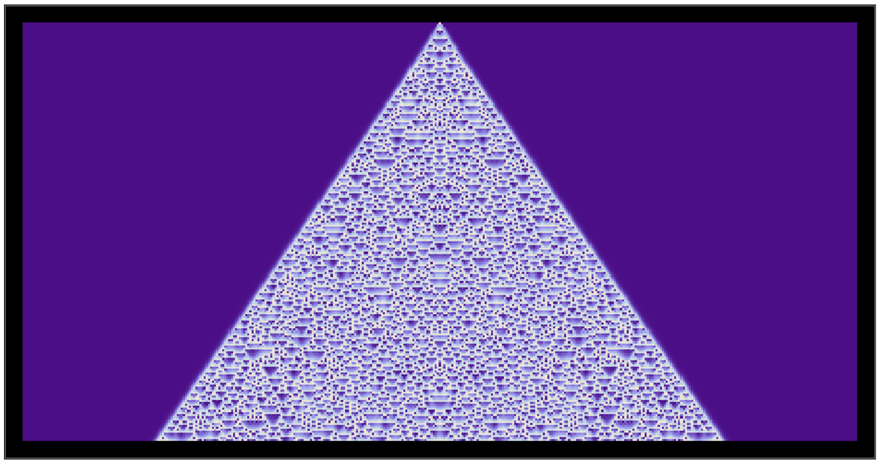
In[ ]:=
```
metadata =
 BlockchainTransactionData[txIDCardanoMinting, "Metadata", BlockchainBase →
      {"Cardano", "Testnet"}]["721"][assetData["PolicyID"]] // Dataset // First
```
Out[ ]=

na	Ice Temple
fil	{ …1 }
im	ipfs://QmazNAHXSgaKmm2NmbhGuFqwQ8DTnsNSWxsKceaM7FbkqP
m	image/png
at	<| type → Lake, rarity → uncommon |>
de	C.A. Temples Collection v0.1

- Get the NFT image

In[ ]:=
```
cid = metadata["files"] // First // #["src"] &
```
Out[ ]=
```
ipfs://QmNUN89yrTkp9YbodcpApkNYjmAQYsgPniSWd58tG4uDR6
```

In[ ]:=
```
Import@ExternalStorageDownload[cid, ExternalStorageBase → "IPFS"]
```
Out[ ]=



- Export the metadata expression to a local file in JSON

In[ ]:=
```
Export["/Users/dsuarez/Documents/CardanoNFTMetadata.json", metadata, "JSON"]
```
Out[ ]=
```
/Users/dsuarez/Documents/CardanoNFTMetadata.json
```

- Upload NFT’s metadata to IPFS

```
In[ ]:= ipfsMetadata =
         ExternalStorageUpload[File["/Users/dsuarez/Documents/CardanoNFTMetadata.json"],
          ExternalStorageBase → "IPFS"]

        ExternalStorageUpload : None is not a recognized MIME Type.

Out[ ]=
                              CID: QmQRsvZ72ED7i9LkHtrHiU6F2dFLjzNKy7zDQ7iz3VikEd
        ExternalStorageObject[ ResourceType : SingleFile                          ]
                              Service: IPFS
                              FileHash: 5d9abb3a4eb1ba4bd5315df74f2e82a2

In[ ]:= Import[ExternalStorageDownload[ipfsMetadata]]
Out[ ]=
        {
            "name": "Ice Temple",
            "files": [{"src": "ipfs://QmNUN89yrTkp9YbodcpApkNYjmAQYsgPniSWd58tG4uDR6",
          "name": "Ice Temple", "mediaType": "image/png"}],
            "image": "ipfs://QmazNAHXSgaKmm2NmbhGuFqwQ8DTnsNSWxsKceaM7FbkqP",
            "mediaType": "image/png",
            "attributes": {"type": "Lake", "rarity": "uncommon"},
            "description": "C.A. Temples Collection v0.1"
        }
```

- The following CID is needed for the minting process

```
In[ ]:= ipfsMetadata["CID"]
Out[ ]=
        QmQRsvZ72ED7i9LkHtrHiU6F2dFLjzNKy7zDQ7iz3VikEd
```

Create Ethereum accounts

Account 1 - ERC721 Contract’s owner

```
In[ ]:= testKeys = GenerateAsymmetricKeyPair[Method → <|
            "Type" → "EllipticCurve", "CurveName" → "Ethereum", "Compressed" → False|>]

In[ ]:= testAddress = BlockchainKeyEncode[testKeys["PrivateKey"],
          "Address", BlockchainBase → {"Ethereum", "Testnet"}]
Out[ ]=
        7f7e831c1914371A483042590ef115Da89a1d5f1
```

Account 2 - NFT’s recipient

```
        testKeys2 = GenerateAsymmetricKeyPair[Method → <|
            "Type" → "EllipticCurve", "CurveName" → "Ethereum", "Compressed" → False|>]
```

```
In[ ]:= testKeys2 = ⟨|"PrivateKey" → PrivateKey[ Type: Elliptic curve (secp256k1 )
                                                  Private key size: 256 b
                                                  Public key size: 512 b ],
           "PublicKey" → PublicKey[ Type: Elliptic curve (secp256k1 )
                                    Public key size: 512 b ]|⟩;

In[ ]:= testAccount2 = BlockchainKeyEncode[testKeys2["PublicKey"],
          "Address", BlockchainBase → {"Ethereum", "Testnet"}]
Out[ ]=
        c729Dd19989C15770E099Cc7056C9fC62408D18B
```

Faucet transaction 1

Faucet transaction 2

Deploy an ERC721 smart contract

Deploy an NFT Smart contract

- Local API connected to Goerli, an API upgrade is required to compile Solidity version 0.8.x

```
In[ ]:= Blockchain`$TemplateBase = "http://localhost:8000"
Out[ ]=
        http://localhost:8000
```

- The following code is an ERC721 smart contract for an NFT called CardanoNFT

```
solidityCode = "
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import \"@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol\";
import \"@openzeppelin/contracts/utils/Counters.sol\";
import \"@openzeppelin/contracts/access/Ownable.sol\";

contract CardanoNFT is ERC721Enumerable, Ownable {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIdCounter;

    // Mapping from token ID to metadata URI
    mapping(uint256 => string) private _tokenURIs;

    constructor() ERC721(\"CardanoNFT\", \"CNFT\") {}

    function _setTokenURI(uint256 tokenId, string memory _tokenURI) internal virtual {
        _tokenURIs[tokenId] = _tokenURI;
    }

    function tokenURI(uint256 tokenId) public view virtual override returns (string memory
        require(_exists(tokenId), \"ERC721Metadata: URI query for nonexistent token\");
        return _tokenURIs[tokenId];
    }

    function mint(address recipient, string memory metadataURI) public onlyOwner {
        _tokenIdCounter.increment();
        uint256 newTokenId = _tokenIdCounter.current();
        _safeMint(recipient, newTokenId);
        _setTokenURI(newTokenId, metadataURI);
    }
}
";
```

■ Build the transaction

In[ ]:=
```
trxNFT = BlockchainTransaction[
  <|
   "TransactionCount" → 1,
   "GasPrice" → Quantity[1, "GWei"],
   "Contract" → solidityCode,
   "BlockchainBase" → {"Ethereum", "Testnet"}
   |>]
```

Out[ ]=
```
BlockchainTransaction[ blockchain base: {Ethereum, Testnet }
                       signed: False
                       fee: 3 469 549 000 000 000 wei ]
```

- Sign the transaction

In[ ]:=
```
trxNFT = BlockchainTransactionSign[trxNFT, testKeys["PrivateKey"]]
```

Out[ ]=
```
BlockchainTransaction[ blockchain base: {Ethereum, Testnet }
                       signed: True
                       fee: 3 469 549 000 000 000 wei ]
```

- Submit the transaction on the Ethereum Testnet

In[ ]:=
```
trxNFT = BlockchainTransactionSubmit[trxNFT]
```

Out[ ]=
```
BlockchainTransaction[ blockchain base: {Ethereum, Testnet }
                       signed: True
                       fee: 3 469 549 000 000 000 wei ]
```

In[ ]:=
```
trxNFT["TransactionID"]
```

Out[ ]=
```
303fd20523943d673cc5529433b8d62f57bf5b9f837fc442a2e7ca2eb292680a
```

In[ ]:=
```
BlockchainTransactionData[trxNFT["TransactionID"],
 BlockchainBase → {"Ethereum", "Testnet"}]
```

```
BlockchainTransactionData : TxID:
    0x303fd20523943d673cc5529433b8d62f57bf5b9f837fc442a2e7ca2eb292680a has not been
    mined yet.
```

Out[ ]=
```
Missing[PendingToMine]
```

- After some time

```
In[∘]:= BlockchainTransactionData[trxNFT["TransactionID"],
          BlockchainBase → {"Ethereum", "Testnet"}] // Dataset
Out[∘]=
```

TransactionID	303fd20523943d673cc5529433b8d62f57bf5b9f837fc442a2e7ca2
BlockHash	825c98c37837839eb11a883d919b06613cf33fbf37f1f25a1b3a5c8
BlockNumber	9 541 823
Confirmations	5
Timestamp	Fri 18 Aug 2023 19:02:00
Status	True
TransactionIndex	16
Sender	7f7e831c1914371A483042590ef115Da89a1d5f1
ContractAddress	2a3af98eaFc15Bf61Ac3C1DdDd001104209FBc8A
Amount	0 wei
GasUsed	2 911 191
GasPrice	1 000 000 000 wei
Fee	2 911 191 000 000 000 wei
TransactionCount	1
Size	14 220 B
InputData	ByteArray[ ]
TransactionDigest	ByteArray[ ]
DigitalSignature	DigitalSignature[Type : Elliptic curve (secp256k1) Hashing method: None Signature size: 512 b]
SenderPublicKey	PublicKey[Type: Elliptic curve (secp256k1) Public key size: 512 b]
EventList	{<|Address → 2a3af98eaFc15Bf61Ac3C1DdDd001104209FBc8A,

■ Here is the contract address

```
In[•]:= trxNFTContract = BlockchainTransactionData[
           "303fd20523943d673cc5529433b8d62f57bf5b9f837fc442a2e7ca2eb292680a",
           BlockchainBase → {"Ethereum", "Testnet"}]["ContractAddress"]
Out[•]=
        2a3af98eaFc15Bf61Ac3C1DdDd001104209FBc8A
```

- Getting the contract’s owner, this is an on-chain call

```
In[•]:= contractOwner = BlockchainContractValue[
          trxNFTContract, <|"Function" → Typed["owner", {} → "address"], "Inputs" → {}|>,
          BlockchainBase → {"Ethereum", "Testnet"}]
Out[•]=
        7f7e831c1914371A483042590ef115Da89a1d5f1
```

Minting a Cardano NFT on Ethereum

- Review ERC721 smart contract address

```
In[•]:= trxNFTContract
Out[•]=
        2a3af98eaFc15Bf61Ac3C1DdDd001104209FBc8A
```

- Recipient address

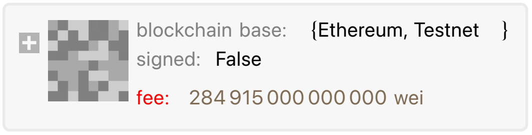
```
In[•]:= recipient = BlockchainKeyEncode[testKeys2["PrivateKey"],
          "Address", BlockchainBase → {"Ethereum", "Testnet"}]
Out[•]=
        c729Dd19989C15770E099Cc7056C9fC62408D18B
```

- Build a transaction for minting an NFT

```
In[•]:= BlockchainTransaction[
         <|
          "BlockchainBase" → {"Ethereum", "Testnet"},
          "TransactionCount" → 2,
          "Address" → trxNFTContract,
          "GasPrice" → Quantity[1, "GWei"],
          "FunctionCall" → <|"Function" → Typed["mint", {"address", "string"} → {}],
            "Inputs" → {recipient, ipfsMetadata["CID"]}|>
          |>]
        ••• BlockchainTransaction : Calling contract returned following error: Ownable: caller is not the owner
Out[•]=
        $Failed
```

- In this case the contracts owner must be the Sender

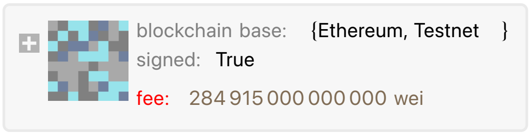
```
In[ ]:= trxMint = BlockchainTransaction[
          <|
           "BlockchainBase" → {"Ethereum", "Testnet"},
           "TransactionCount" → 2,
           "Address" → trxNFTContract,
           "GasPrice" → Quantity[1, "GWei"],
           "FunctionCall" → <|"Function" → Typed["mint", {"address", "string"} → {}],
             "Sender" → contractOwner,
             "Inputs" → {recipient, ipfsMetadata["CID"]}|>
           |>]
Out[ ]=
```



```
BlockchainTransaction[ blockchain base: {Ethereum, Testnet }
                       signed: False
                       fee: 284 915 000 000 000 wei ]
```

■ Sign

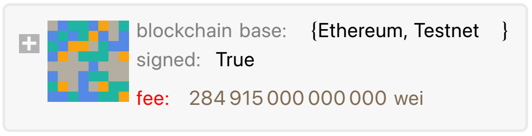
```
In[ ]:= trxMint = BlockchainTransactionSign[trxMint, testKeys["PrivateKey"]]
Out[ ]=
```



```
BlockchainTransaction[ blockchain base: {Ethereum, Testnet }
                       signed: True
                       fee: 284 915 000 000 000 wei ]
```

■ Submit

```
In[ ]:= trxMint = BlockchainTransactionSubmit[trxMint]
Out[ ]=
```



```
BlockchainTransaction[ blockchain base: {Ethereum, Testnet }
                       signed: True
                       fee: 284 915 000 000 000 wei ]
```

■ Where the transaction ID of the minting transaction is

```
In[ ]:= trxMint["TransactionID"]
Out[ ]=
ce05deda32fb55b47d5493fc20537ac1b74fc7f624c8ae0c90da3599ed80de84
```

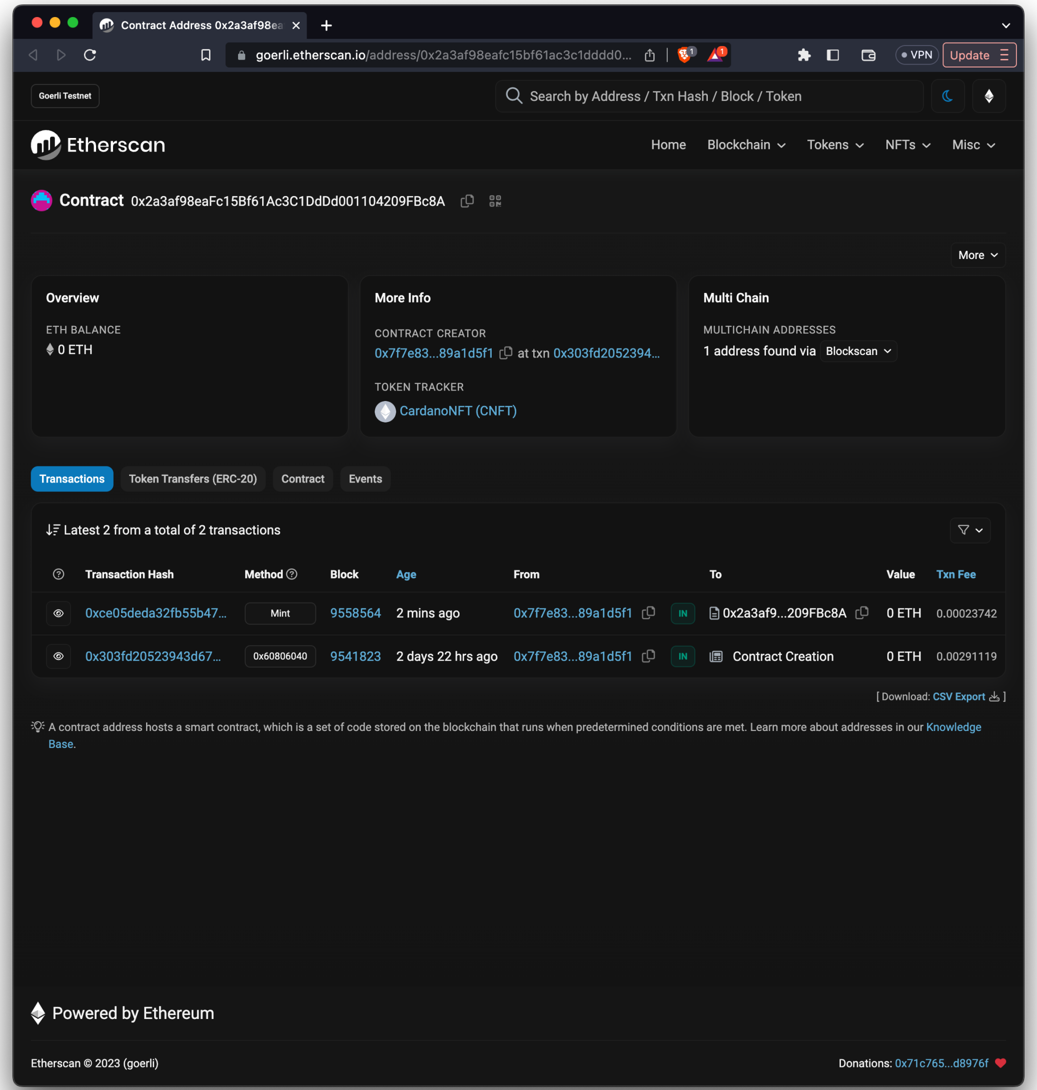
■ Reading transaction data

In[∘]:=
```
BlockchainTransactionData[trxMint["TransactionID"],
  BlockchainBase → {"Ethereum", "Testnet"}] // Dataset
```

Out[∘]=

TransactionID	ce05deda32fb55b47d5493fc20537ac1b74fc7f624c8ae0c90da359
BlockHash	9de0039a1b44409740bc72e0a4d59a5813384e5f9c255b73af6850
BlockNumber	9 558 564
Confirmations	2
Timestamp	Mon 21 Aug 2023 17:57:12
Status	True
TransactionIndex	22
Sender	7f7e831c1914371A483042590ef115Da89a1d5f1
Receiver	2a3af98eaFc15Bf61Ac3C1DdDd001104209FBc8A
Amount	0 wei
GasUsed	237 429
GasPrice	1 000 000 000 wei
Fee	237 429 000 000 000 wei
TransactionCount	2
Size	268 B
InputData	ByteArray[ ]
TransactionDigest	ByteArray[ ]
DigitalSignature	DigitalSignature[Type : Elliptic curve (secp256k1) Hashing method: None Signature size: 512 b]
SenderPublicKey	PublicKey[Type: Elliptic curve (secp256k1) Public key size: 512 b]
EventList	{<|Address → 2a3af98eaFc15Bf61Ac3C1DdDd001104209FBc8A,

■ Verification of the ERC721 contract using Goerli Ethescan



Contract Address 0x2a3af98ea

goerli.etherscan.io/address/0x2a3af98eafc15bf61ac3c1dddd0...

Goerli Testnet

Search by Address / Txn Hash / Block / Token

Etherscan

Home Blockchain Tokens NFTs Misc

Contract 0x2a3af98eaFc15Bf61Ac3C1DdDd001104209FBc8A

Overview

ETH BALANCE

0 ETH

More Info

CONTRACT CREATOR

0x7f7e83...89a1d5f1 at txn 0x303fd2052394...

TOKEN TRACKER

CardanoNFT (CNFT)

Multi Chain

MULTICHAIN ADDRESSES

1 address found via Blockscan

Transactions | Token Transfers (ERC-20) | Contract | Events

Latest 2 from a total of 2 transactions

Transaction Hash	Method	Block	Age	From		To	Value	Txn Fee
0xce05deda32fb55b47...	Mint	9558564	2 mins ago	0x7f7e83...89a1d5f1	IN	0x2a3af9...209FBc8A	0 ETH	0.00023742
0x303fd20523943d67...	0x60806040	9541823	2 days 22 hrs ago	0x7f7e83...89a1d5f1	IN	Contract Creation	0 ETH	0.00291119

[Download: CSV Export]

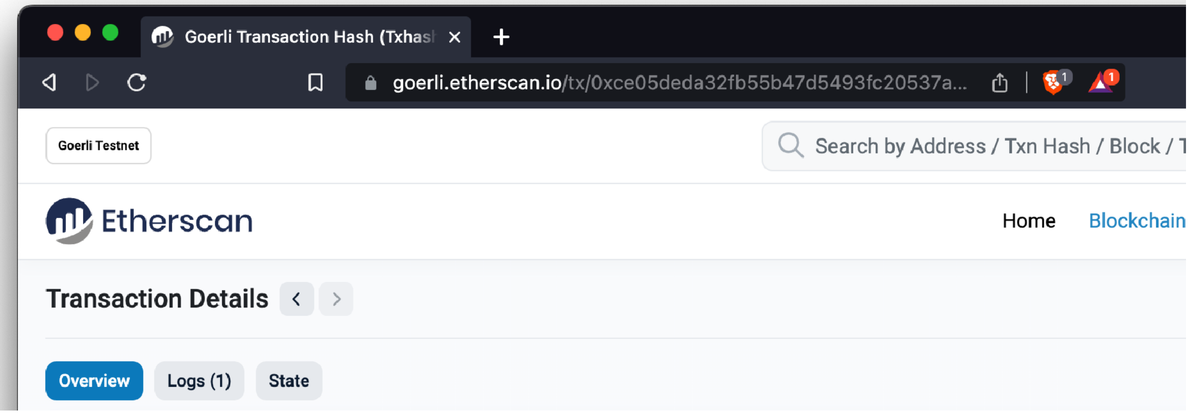
A contract address hosts a smart contract, which is a set of code stored on the blockchain that runs when predetermined conditions are met. Learn more about addresses in our Knowledge Base.

Powered by Ethereum

Etherscan © 2023 (goerli)

Donations: 0x71c765...d8976f

■ Verification of the minting transaction on Goerli Etherscan



Goerli Transaction Hash (Txhas

goerli.etherscan.io/tx/0xce05deda32fb55b47d5493fc20537a...

Goerli Testnet

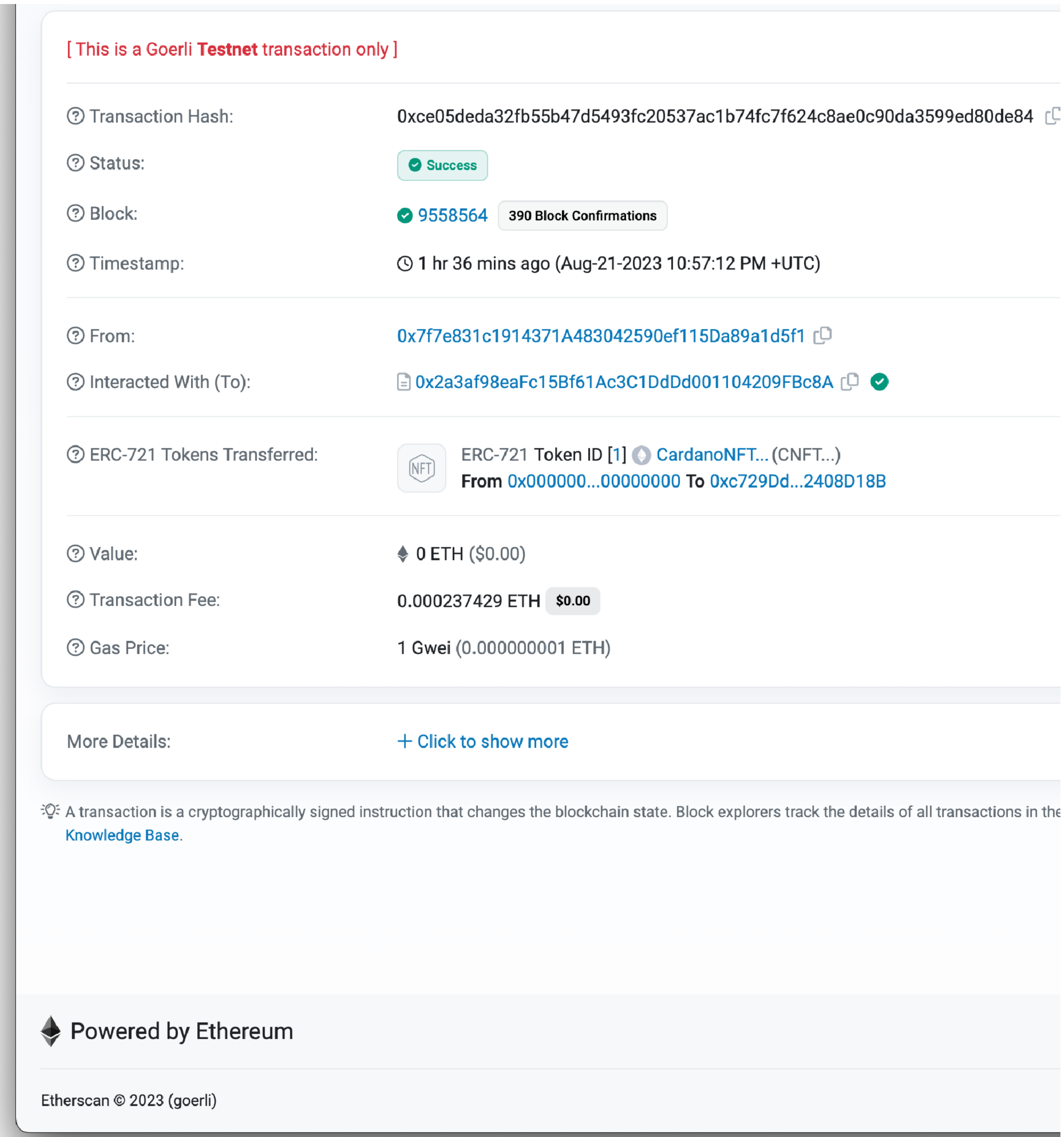
Search by Address / Txn Hash / Block / T

Etherscan

Home Blockchain

Transaction Details

Overview | Logs (1) | State



[This is a Goerli **Testnet** transaction only]

Transaction Hash:	0xce05deda32fb55b47d5493fc20537ac1b74fc7f624c8ae0c90da3599ed80de84
Status:	Success
Block:	9558564 390 Block Confirmations
Timestamp:	1 hr 36 mins ago (Aug-21-2023 10:57:12 PM +UTC)
From:	0x7f7e831c1914371A483042590ef115Da89a1d5f1
Interacted With (To):	0x2a3af98eaFc15Bf61Ac3C1DdDd001104209FBc8A
ERC-721 Tokens Transferred:	ERC-721 Token ID [1] CardanoNFT... (CNFT...) **From** 0x000000...00000000 **To** 0xc729Dd...2408D18B
Value:	0 ETH ($0.00)
Transaction Fee:	0.000237429 ETH $0.00
Gas Price:	1 Gwei (0.000000001 ETH)
More Details:	+ Click to show more

A transaction is a cryptographically signed instruction that changes the blockchain state. Block explorers track the details of all transactions in the
Knowledge Base.

Powered by Ethereum

Etherscan © 2023 (goerli)

Reading on-chain smart contract data

- Fix BlockchainAddressData harvester for Testnet
- Get the balance of tokens from the recipient address

```
In[∘]:= BlockchainContractValue[
          trxNFTContract, <|"Function" → Typed["balanceOf", {"address"} → "uint256"],
           "Inputs" → {recipient}|>, BlockchainBase → {"Ethereum", "Testnet"}]
Out[∘]=
        1
```

■ Get the token metadata URI of the token Id

```
In[∘]:= tokenId = 1
Out[∘]=
        1
```

```
In[∘]:= tokenURI = BlockchainContractValue[
           trxNFTContract, <|"Function" → Typed["tokenURI", {"uint256"} → "string"],
            "Inputs" → {tokenId}|>, BlockchainBase → {"Ethereum", "Testnet"}]
Out[∘]=
        QmQRsvZ72ED7i9LkHtrHiU6F2dFLjzNKy7zDQ7iz3VikEd
```

■ Download the metadata from IPFS

```
In[∘]:= tokenMetadata =
         Import[ExternalStorageDownload[tokenURI, ExternalStorageBase → "IPFS"],
           "RawJSON"] // Dataset
Out[∘]=
```

na	Ice Temple
fil	{…₁}
im	ipfs://QmazNAHXSgaKmm2NmbhGuFqwQ8DTnsNSWxsKceaM7FbkqP
m	image/png
at	<| type → Lake, rarity → uncommon |>
de	C.A. Temples Collection v0.1

■ Download the NFT image from IPFS

```
In[ ]:= Import@ExternalStorageDownload[
          tokenMetadata["files"] // First // #["src"] &, ExternalStorageBase → "IPFS"]
```

Out[ ]=