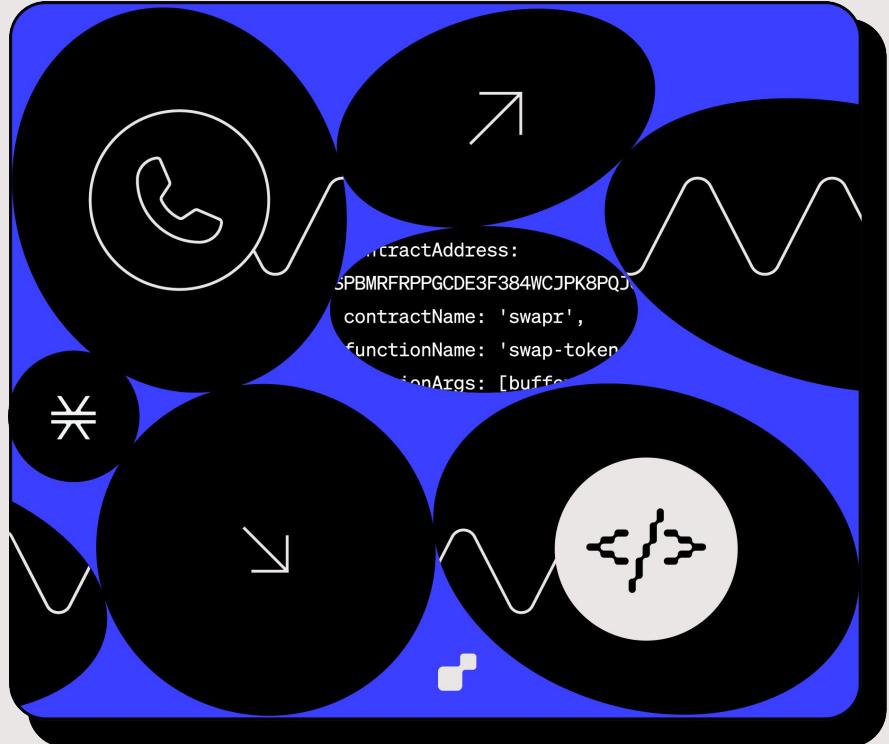


Hiro Monthly Developer Calls

#3 – June 30, 2022





Purpose

Hiro is
**developer
obsessed**

Strive to
engage
with the developer
community directly

Invite and
inspire
developers of
diverse backgrounds
to Stacks & Hiro

Listen
to your needs
carefully & improve
the DevX

Shoutouts & Thanks

June 2022 Contributors

Thank you!

Clarinet

@jhammond2012

@fariedt

@mcintyre94

@moodmosaic

Web Wallet

@talentdev0807

@mcintyre94

Stacks.js

@pradel

Explorer

@mcintyre94

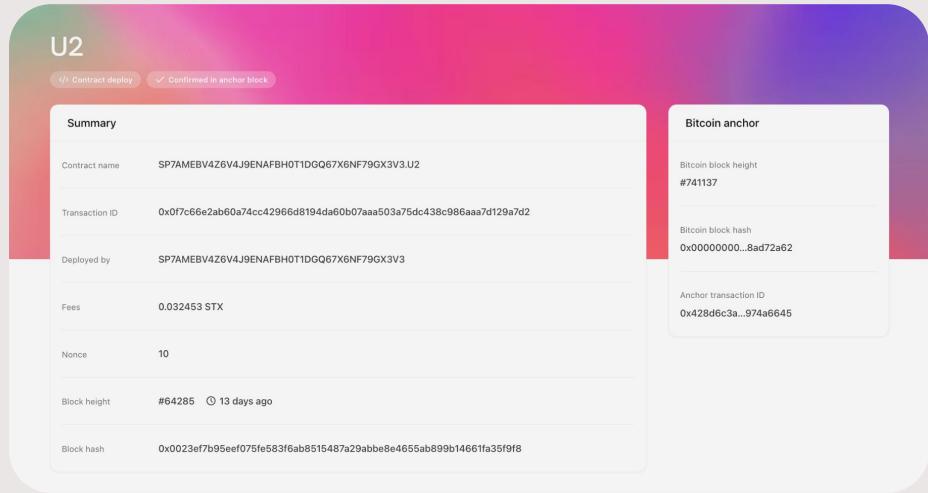
Smart Contract of the Month

```
contractAddress: "0xWCJP8PQJ...  
SPBMRFR...  
contractName: "swap-token"  
functionName: 'swap-token'  
functionArgs: [buffer]
```

Smart Contract of the Month: U2

- In 2014, U2 released their album on iTunes and it was automatically added to the “Purchased” playlist
- U2.clar is a non-fungible token built on Stacks that not only mimics, but pokes fun at, the original distribution method U2 used for the album in 2014.
- The U2 NFT was airdropped with the mechanism to mint a new one upon attempting a transfer

→ [Link to Contract in Stacks Explorer](#)
→ [Link to IPFS hosted on Cloudflare](#)



Chris Richards of The Washington Post called the release "**rock-and-roll as dystopian junk mail**".

Slate said that it was "extremely unsettling" that "**consent and interest are no longer a requisite for owning an album, only corporate prerogative**".

Smart Contract of the Month: U2

Functions Requiring Transaction

 burn	→
 buy-in-ustx	→
 claim	→
 claim-five-for	→
 claim-for	→
 claim-ten-for	→
 claim-twenty-five-for	→

 disable-premint	→
 enable-premint	→
 freeze-metadata	→
 list-in-ustx	→
 set-artist-address	→
 set-base-uri	→
 set-mint-limit	→

 U2 SP7AMEBV...F79GX3V3	
 37 functions	
 29 variables	
 4 maps	
 1 token	
 set-price	→
 toggle-pause	→
 toggle-sale-state	→
 transfer	→
 unlist-in-ustx	→

Smart Contract of the Month: U2

Read-Only Functions

API	get-paused	read_only
API	get-premint-enabled	read_only
API	get-price	read_only
API	get-sale-enabled	read_only
API	get-token-uri	read_only

API	get-balance	read_only
API	get-last-token-id	read_only
API	get-listing-in-ustx	read_only
API	get-mint-limit	read_only
API	get-mints	read_only
API	get-owner	read_only
API	get-passes	read_only



Smart Contract of the Month: U2

Implement deployed NFT trait

```
(impl-trait .nft-trait.nft-trait)
```

Define name of non-fungible token

```
(define-non-fungible-token Songs-of-Innocence uint)
```

Define some constants (e.g. deployer address)

```
; Constants
(define-constant DEPLOYER 'SP7AMEBV4Z6V4J9ENAFBH0T1DGQ67X6NF79GX3V3)
(define-constant COMM u99000)
(define-constant COMM-ADDR-ONE 'SP7AMEBV4Z6V4J9ENAFBH0T1DGQ67X6NF79GX3V3)
(define-constant COMM-ADDR-TWO 'SP7AMEBV4Z6V4J9ENAFBH0T1DGQ67X6NF79GX3V3)
```

Define standard error constants

```
(define-constant ERR-NO-MORE-NFTS u100)
(define-constant ERR-NOT-ENOUGH-PASSES u101)
(define-constant ERR-PUBLIC-SALE-DISABLED u102)
(define-constant ERR-CONTRACT-INITIALIZED u103)
(define-constant ERR-NOT-AUTHORIZED u104)
(define-constant ERR-INVALID-USER u105)
(define-constant ERR-LISTING u106)
(define-constant ERR-WRONG-COMMISSION u107)
(define-constant ERR-NOT-FOUND u108)
(define-constant ERR-PAUSED u109)
(define-constant ERR-MINT-LIMIT u110)
(define-constant ERR-METADATA-FROZEN u111)
(define-constant ERR-AIRDROP-CALLED u112)
(define-constant ERR-NO-MORE-MINTS u113)
```

Define necessary internal variables

```
; Internal variables
```

```
(define-data-var mint-limit uint u420420000000)
```

```
(define-data-var last-id uint u1)
```

```
(define-data-var total-price uint u0)
```

Define artist-address

```
(define-data-var artist-address principal 'SP7AMEBV4Z6V4J9ENAFBH0T1DGQ67X6NF79GX3V3)
```

Define string variable with IPFS link

```
(define-data-var ipfs-root (string-ascii 80) "https://cloudflare-ipfs.com/ipfs/QmUCSNZ674")
```

```
(define-data-var mint-paused bool false)
```

```
(define-data-var premint-enabled bool false)
```

```
(define-data-var sale-enabled bool true)
```

```
(define-data-var metadata-frozen bool false)
```

```
(define-data-var airdrop-called bool false)
```

```
(define-data-var mint-cap uint u420420000000)
```

Defining maps to store data

```
(define-map mints-per-user principal uint)
```

```
(define-map mint-passes principal uint)
```



Smart Contract of the Month: U2

Claim Functions

- For users to call the smart contract and claim an edition of the NFT
- Multiple types of claim functions
 - Claiming 1 for yourself
 - Claiming 1 for someone
 - Claiming multiple
 - 5, 10, 25



```
(define-public (claim (user (list 1 principal)))
  (mint user))

(define-public (claim-for (addr principal))
  (mint-many (list addr)))

(define-public (claim-five-for (addrs (list 5 principal)))
  (mint-many addrs))

(define-public (claim-ten-for (addrs (list 10 principal)))
  (mint-many addrs))

(define-public (claim-twenty-five-for (addrs (list 25 principal)))
  (mint-many addrs))
```

Smart Contract of the Month: U2

Defining Mint Functions

Defining mint function

Check to see if principal has mint pass

```
(define-private (mint (orders (list 25 principal)))
  (let
    (
      (passes (get-passes tx-sender))
    )
    (if (var-get premint-enabled)
        (begin
          (asserts! (>= passes (len orders)) (err ERR-NOT-ENOUGH-PASSES))
          (map-set mint-passes tx-sender (- passes (len orders)))
          (mint-many orders)
        )
        (begin
          (asserts! (var-get sale-enabled) (err ERR-PUBLIC-SALE-DISABLED))
          (mint-many orders)))
    )))

```

Defining the mint-many function

```
(define-private (mint-many (orders (list 25 principal)))
  (let
    (
      (last-nft-id (var-get last-id))
      (enabled true)
      (art-addr (var-get artist-address))
      (id-reached (fold mint-many-iter orders last-nft-id))
      (price (* (var-get total-price) (- id-reached last-nft-id)))
      (total-commission (/ (* price COMM u2) u10000))
      (total-artist (- price total-commission))
    )
    (var-set last-id id-reached)
    (ok last-nft-id)))

```

Defining the mint-many-iter function

```
(define-private (mint-many-iter (user principal) (next-id uint))
  (if (<= next-id (var-get mint-limit))
      (begin
        (print "'U2 sent a message to everyone that music is free, and that's disturbing."
        (unwrap! (nft-mint? Songs-of-Innocence next-id user) next-id)
        (+ next-id u1)
      )
      next-id)))

```





Smart Contract of the Month: U2 Defining Public Functions

Defining set-artist-address function

```
(define-public (set-artist-address (address principal))
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-INVALID-USER))
    (ok (var-set artist-address address))))
```

Defining set-price function

```
(define-public (set-price (price uint))
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-INVALID-USER))
    (ok (var-set total-price price))))
```

Defining toggle-pause function

```
(define-public (toggle-pause)
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-INVALID-USER))
    (ok (var-set mint-paused (not (var-get mint-paused))))))
```

Defining set-mint-limit function

```
(define-public (set-mint-limit (limit uint))
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-INVALID-USER))
    (asserts! (< limit (var-get mint-limit)) (err ERR-MINT-LIMIT))
    (ok (var-set mint-limit limit))))
```

Defining is-owner function

```
(define-private (is-owner (token-id uint) (user principal))
  (is-eq user (unwrap! (nft-get-owner? Songs-of-Innocence token-id) false)))
```

Defining set-base-uri function

```
(define-public (set-base-uri (new-base-uri (string-ascii 80)))
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-NOT-AUTHORIZED))
    (asserts! (not (var-get metadata-frozen)) (err ERR-METADATA-FROZEN))
    (var-set ipfs-root new-base-uri)
    (ok true))))
```

Defining freeze-metadata function

```
(define-public (freeze-metadata)
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-NOT-AUTHORIZED))
    (var-set metadata-frozen true)
    (ok true))))
```

All functions check if tx-sender = artist-address

Smart Contract of the Month: U2 Defining Functions Required for NFT Trait (SIP-009)

SIP-009

Defining the transfer function

```
;; Non-custodial SIP-009 transfer function
(define-public (transfer (id uint) (sender principal) (recipient principal))
  (begin
    (asserts! (is-eq tx-sender sender) (err ERR-NOT-AUTHORIZED))
    (asserts! (is-none (map-get? market id)) (err ERR-LISTING))
    (trnsfr id sender recipient)))
```

Defining get-owner function

```
;; read-only functions
(define-read-only (get-owner (token-id uint))
  (ok (nft-get-owner? Songs-of-Innocence token-id)))
```

Defining get-last-token-id function

```
(define-read-only (get-last-token-id)
  (ok (- (var-get last-id) u1)))
```

Defining get-token-uri function

```
(define-read-only (get-token-uri (token-id uint))
  (ok (some (concat (concat (var-get ipfs-root) "Songs-of-Innocence") ".json"))))
```

Defining get-paused function

```
(define-read-only (get-paused)
  (ok (var-get mint-paused)))
```

Defining get-price function

```
(define-read-only (get-price)
  (ok (var-get total-price)))
```

Defining get-mints function

```
(define-read-only (get-mints (caller principal))
  (default-to u0 (map-get? mints-per-user caller)))
```

Defining get-mint-limit function

```
(define-read-only (get-mint-limit)
  (ok (var-get mint-limit)))
```



Smart Contract of the Month: U2 More Non-Custodial Marketplace Functions

Define list-in-ustx function

Creates a listing in the market map, and checks if tx-sender owns the specific NFT

```
(define-public (list-in-ustx (id uint) (price uint) (comm-trait <commission-trait>))
  (let ((listing {price: price, commission: (contract-of comm-trait)}))
    (asserts! (is-owner? id) (err ERR-NOT-AUTHORIZED))
    (map-set market id listing)
    (print {merge listing {a: "list-in-ustx", id: id}}))
  (ok true))
```

Define unlist-in-ustx function

Unlists NFT and removes it from the market map

```
(define-public (unlist-in-ustx (id uint))
  (begin
    (asserts! (is-owner? id) (err ERR-NOT-AUTHORIZED))
    (map-delete market id)
    (print {a: "unlist-in-ustx", id: id})
    (ok true)))
```

Define buy-in-ustx function

Transfers the correct amount of STX from the price variable, and calls the trnsfr function

```
(define-public (buy-in-ustx (id uint) (comm-trait <commission-trait>))
  (let ((owner (unwrap! (nft-get-owner? Songs-of-Innocence id) (err ERR-NOT-FOUND)))
        (listing (unwrap! (map-get? market id) (err ERR-LISTING)))
        (price (get price listing)))
    (asserts! (is-eq (contract-of comm-trait) (get commission listing)) (err ERR-WRONG-COMMISSION))
    (try! (stx-transfer? price tx-sender owner))
    (try! (contract-call? comm-trait pay id price))
    (try! (trnsfr id owner tx-sender))
    (map-delete market id)
    (print {a: "buy-in-ustx", id: id}))
  (ok true)))
```



Smart Contract of the Month: U2 Extra Functions for Mintpass

Define public function toggle-sale-state

Gives deployer ability to enable or disable the premint and sale

```
; Extra functionality required for mintpass
(define-public (toggle-sale-state)
  (let
    (
      ;; (premint (not (var-get premint-enabled)))
      ;(sale (not (var-get sale-enabled)))
    )
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-NOT-AUTHORIZED))
    (var-set premint-enabled false)
    (var-set sale-enabled sale)
    (print { sale: sale })
    (ok true)))
  
```

Define enable-premint function

```
(define-public (enable-premint)
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-NOT-AUTHORIZED))
    (ok (var-set premint-enabled true))))
```

Define disable-premint function

```
(define-public (disable-premint)
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-NOT-AUTHORIZED))
    (ok (var-set premint-enabled false))))
```

Define read-only get-passes function

```
(define-read-only (get-passes (caller principal))
  (default-to u0 (map-get? mint-passes caller)))
```

Define read-only get-premint-enabled function

```
(define-read-only (get-premint-enabled)
  (ok (var-get premint-enabled)))
```

Define read-only get-sale-enabled function

```
(define-read-only (get-sale-enabled)
  (ok (var-get sale-enabled)))
```

Define burn function

A deceptive burn function that actually calls nft-mint? to create a new NFT

```
(define-public (burn (token-id uint))
  (let
    (
      (next-id (var-get last-id))
    )
    (var-set last-id (+ u1 next-id))
    (try! (nft-mint? Songs-of-Innocence next-id tx-sender))
    (print "You said burn, so I burned you another copy of the album.")
    (ok true)))
```

Upon trying to burn NFT, another is minted



Smart Contract of the Month: U2 Non-Custodial Marketplace Functions

Utilize commission trait on Mainnet

```
;; Non-custodial marketplace extras
(use-trait commission-trait .commission-trait.commission)
```

Define maps for NFT quantity per address, and prices/commissions

```
(define-map token-count principal uint)
(define-map market uint {price: uint, commission: principal})
```

Define get-balance function

```
(define-read-only (get-balance (account principal))
  (default-to u0
    (map-get? token-count account)))
```

Define trnsfr function

```
(define-private (trnsfr (id uint) (sender principal) (recipient principal))
  (match (nft-transfer? Songs-of-Innocence id sender recipient)
    success
      (let
        ((sender-balance (get-balance sender))
         (recipient-balance (get-balance recipient)))
        (map-set token-count
          recipient
          (+ recipient-balance u1))
        (try! (nft-mint? Songs-of-Innocence (var-get last-id) sender))
        (var-set last-id (+ (var-get last-id) u1))
        (try! (contract-call? .Friends-Are-Forever claim tx-sender))
        (print "'The delivery mechanism amounts to nothing more than spam with forced downloads.' -Wired ")
        (ok success))
      error (err error)))
```

If you try to transfer NFT, it mints another to your address



Check if user is also owner of contract

```
(define-private (is-sender-owner (id uint))
  (let ((owner (unwrap! (nft-get-owner? Songs-of-Innocence id) false)))
    (or (is-eq tx-sender owner) (is-eq contract-caller owner))))
```

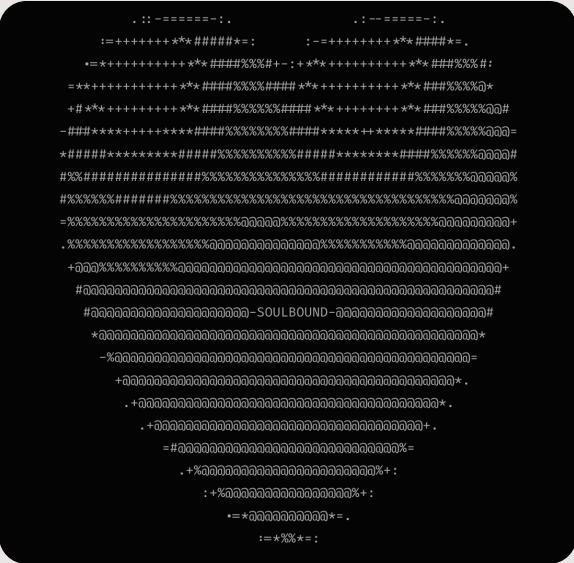
Read only function for getting listing in USTX

```
(define-read-only (get-listing-in-ustx (id uint))
  (map-get? market id))
```

Smart Contract of the Month: U2

- Contract begins with ASCII art heart with word "SOULBOUND"
- "[Decentralized Society: Finding Web3's Soul](#)"
(11 May 2022)
 - E. Glen Weyl, Puja Ohlhaver, and Vitalik Buterin
 - Non-transferable NFTs can act as passport
- More Hidden Messages
 - Transfer: "I thought you would have learned your lesson the first time..."
 - Burn: "I thought we were friends?"
 - Error on Burn

→ [Link to Contract on Stacks Explorer](#)



Songs-of-Innocence.json & Friends-are-Forever.json

```

"version": 1,
"name": "Songs of Innocence DELUXE EDITION",
"video": "ipfs://QmRBDKFs3e2naEKh8eMT27ir8fTKtwXcoqMdzLwk9nTqcp/songs-of-innocence.mov",
"description": "I had this beautiful idea, but (we) got carried away with ourselves. Art",
"edition": "1-of-Many",
"songs": [
  {
    "trait_type": "Track 1: The Miracle (Of Joey Ramone)",
    "value": "QmVzQZnM9ndN4hd359vDvp7boFMplfjSBnDx8CprxvtC.mp3"
  },
  {
    "trait_type": "Track 2: Every Breaking Wave",
    "value": "QmrAriBo2EmF56Gjs3LwMSTFa1FQNjxHt7jAhzPsyJ.mp3"
  },
  {
    "trait_type": "Track 3: California (There Is No End to Love)",
    "value": "Qmd2SjegHwdndmzgd7R6DpGSmNzcjWqfkzesCfskN0qi.mp3"
  },
  {
    "trait_type": "Track 4: Song for Someone",
    "value": "QmatmuWnzkcnUdBWLg2G86hRQPgqbq9qjyJUVSxth55ZT.mp3"
  },
  {
    "trait_type": "Track 5: Iris (Hold Me Close)",
    "value": "QmWUNLZ5f28n16yir8CGZ6B43tMhtpkMKHTGRufswGLBlu.mp3"
  },
  {
    "trait_type": "Track 6: Volcano",
    "value": "QmSYHPi94e24om6uBLYVReufpKVfydwsdHdbGkkbmXsV21.mp3"
  },
  {
    "trait_type": "Track 7: Raised by Wolves",
    "value": "QmPFzB4Pbj5162hbgGk5wfwr1cTvZxqF6Mq9VhSWerUAT.mp3"
  },
  {
    "trait_type": "Track 8: Cedarwood Road",
    "value": "QmRPLzlBub3b3RDbEGopvVmT2jijVXHVWmLE82wT5zjAK.mp3"
  },
  {
    "trait_type": "Track 9: Sleep Like a Baby Tonight",
    "value": "QmYUmzqzigEZhU2yoMHVALumQjmqrnW2BvJVk1hPgxmT.mp3"
  },
  {
    "trait_type": "Track 10: This Is Where You Can Reach Me Now",
    "value": "QmaKXGnKRWRwiSMjEeL9qBzj3GVXF2ALHHTDYSnNNVnK7.mp3"
  },
  {
    "trait_type": "Track 11: The Troubles",
    "value": "QmTihXxACbnvAYSzs6DuFZA2cimjEiKBQHZ78TbfUMkzFy.mp3"
  },
  {
    "trait_type": "Digital Booklet",
    "value": "QmbgX7k2mp6Onow7y9vgpBhBfQSLeMs4gqtFwFzzrYrZLZ.pdf"
  }
]
  
```

```

"properties": {
  "collection": "Apple Itunes",
  "collection_root": {
    "type": "string",
    "value": "ipfs://QmRBDKFs3e2naEKh8eMT27ir8fTKtwXcoqMdzLwk9nTqcp/friends.png"
  },
  "artist_name": {
    "type": "string",
    "value": "U2"
  },
  "album_name": {
    "type": "string",
    "value": "Songs of Innocence"
  },
  "recorded": {
    "type": "string",
    "value": "2009:2014"
  },
  "studio": {
    "type": "string",
    "value": "Electric Lady (New York City)Pull (New York C"
  },
  "release_date": {
    "type": "string",
    "value": "9 September 2014"
  },
  "length": {
    "type": "string",
    "value": "48 minutes:11 seconds "
  },
  "genre": {
    "type": "string",
    "value": "Rock"
  },
  "label": {
    "type": "string",
    "value": "Island, Interscope"
  },
  "Producers": {
    "type": "string",
    "value": "Danger MousePaul Epworth (add.)Ryan Tedder (add.)Declan Gaffney (add.)Flood (add.)"
  },
  "seed": "unknown"
}

{
  "version": 1,
  "name": "Songs of Innocence",
  "image": "ipfs://QmRBDKFs3e2naEKh8eMT27ir8fTKtwXcoqMdzLwk9nTqcp/friends.png",
  "description": "I had this beautiful idea, but (we) got carried away with ourselves. Artists are prone to that kind of thing. A drop of megalomania, a touch of generosity, a dash of self-promotion and deep fear that these songs, that we poured our life into over the last few years, mightn't be heard. There's a lot of noise out there. I guess we got a little noisy ourselves to get through it",
  "edition": "1-of-Many",
  "properties": {
    "collection": "Friends Are Forever",
    "collection_root": {
      "type": "string",
      "value": "ipfs://QmRBDKFs3e2naEKh8eMT27ir8fTKtwXcoqMdzLwk9nTqcp"
    },
    "tim-cook": {
      "type": "string",
      "value": "Apple CEO"
    },
    "bono": {
      "type": "string",
      "value": "Bono"
    },
    "seed": "LOVE"
  }
}
  
```



Hidden Messages in U2.clar

- **mint-many-iter**
 - "U2 sent a message to everyone that music is free, and that's disturbing. It's easy to do that when you're a multi-millionaire-billionaire and money isn't really something that you worry about, but when you're a working rock 'n' roll band and you count on every dollar, it's disappointing to see someone do that." -Keith Nelson
- **Commented in contract**
 - "I had this beautiful idea, but (we) got carried away with ourselves. Artists are prone to that kind of thing. A drop of megalomania, a touch of generosity, a dash of self-promotion and deep fear that these songs, that we poured our life into over the last few years, mightn't be heard. There's a lot of noise out there. Guess we got a little noisy ourselves to get through it" -Bono
- **trnsfr**
 - "The delivery mechanism amounts to nothing more than spam with forced downloads." -Wired
- **burn**
 - "You said burn, so I burned you another copy of the album." -Contract Developer



Smart Contract of the Month: U2



josh.stack
@Cryptochonn

Anyone who received the U2 Songs of Innocence airdrop on [@Stacks](#).

The contracts appear to have a potential exploit that can be triggered if the deployer calls the "toggle-sale-state" function in the U2 contract and the NFT is listed for sale.

6:34 PM · Jun 21, 2022 · Twitter Web App

Community Topics



Community Topics

Topic 1: Contract Testing/Debugging & Best Practices

Special guests:

The screenshot shows the homepage of SONGS RADIO. At the top left is the logo 'SONGS RADIO' with a stylized 'S'. At the top right is a link 'About STR'. Below the logo is a purple button with white text that reads 'Secured by Bitcoin'. The main title 'Discover, Invest and Own Music NFTs With Ease' is displayed in large, bold, white font. A subtext below it says: 'Music NFTs will continue to revolutionize the way that artists and fans create community together as we enter the upcoming year — undoubtedly changing the trajectory of countless budding music careers.' At the bottom are two buttons: 'Explore Market →' and 'Register an Artist →'.



Ludo Galabru
Staff Engineer @ Hiro



Community Topics

Topic 2: SIP-019 Update

[stacksgov / sips](#) Public

<> Code ⏺ Issues 18 ⏺ Pull requests 15 ⏺ Zenhub ⏺ Discussions ⏺ Actions ⏺ Projects ⏺ Insights

SIP-019: Notifications for Token Metadata Updates #72

[! Open](#) rafaelcr wants to merge 16 commits into [stacksgov:main](#) from [hirosystems:feat/token-metadata-updates](#)

Conversation 40 Commits 16 Checks 0 Files changed 2

rafaelcr commented on May 24 · edited

Preamble

SIP Number: 019

Title: Notifications for Token Metadata Updates

Author: Rafael Cárdenas (rafael@hiro.so), Matthew Little (matt@hiro.so)

Consideration: Technical

Type: Standard

Status: Accepted

Created: 17 May 2022

License: GPL-3.0

Sign-off: Jude Nelson jude@stacks.org

Layer: Traits

 Accepted

[hirosystems / stacks-blockchain-api](#) DevX Public

<> Code ⏺ Issues 71 ⏺ Pull requests 29 ⏺ Zenhub ⏺ Discussions ⏺ Actions ⏺ Projects ⏺ Security ⏺ Insights

refactor the Token Metadata Processor into an independent service #1222

[! Open](#) rafaelcr opened this issue 3 minutes ago · 0 comments

rafaelcr commented 3 minutes ago

Background

Our Stacks API currently exposes [a few endpoints](#) that attempt to serve FT metadata. This data is pulled by a background queue that listens to new handle token metadata update notifications, NFTs or SFTs. It currently is a complex part of the API that we're looking to compartmentalize as an ex

The goal of this issue is to track the process in building a new metadata micro-service, an independent system that manages its own internal logic separately.

Features

The new Token Metadata Service will support the following token metadata standards:

1. NFTs ([SIP-009](#))
2. FTs ([SIP-010](#))
3. SFTs ([SIP-013](#))
4. Token metadata update notifications ([! SIP-019: Notifications for Token Metadata Updates](#) stacksgov/sips#72)

NOTE: This service will only process and consume token metadata that follows the guidelines set by SIP-016: [stacksgov/sips#44](#)

After processing, all data will be exposed through REST endpoints for external services to consume (endpoint specification TBD).

 WIP

Community Topics

Topic 3: Hyperchains Q+A

Are you looking for
the roadmap,
timelines, open
feature requests,
or report problems?



github.com/hirosystems

The image shows two GitHub repository pages side-by-side, both titled "Q2-2022".

hirosystems / stacks-blockchain-api

- Issues: 67
- Pull requests: 22

Milestones

Q2-2022

Due by May 31, 2022 0% complete

Q2-2022

- 5 Open 0 Closed
- Optimize event replay process** Q2-2022
#1082 opened on Feb 25 by rafaelcr 4 of 8 tasks
- add arbitrary contract call support to ro**
#1095 opened on Mar 8 by rafaelcr
- Introduce an API Gateway to access our**
#1123 opened on Mar 24 by saralab
- Implement automated functional testing**
#1002 opened on Jan 25 by saralab
- Add blockchain stats endpoints** Q2-2022
#1130 opened on Mar 28 by rafaelcr 2 tasks

hirosystems / stacks.js

- Issues: 18
- Pull requests: 1

Milestones

Q2-2022

Due by May 31, 2022 0% complete

Q2 2022

- 5 Open 0 Closed
- Offload bn.js** ✓
#1150 opened on Dec 16, 2021 by al...
- Offload bip* dependency**
#1198 opened on Feb 23 by janniks...
- [wallet-sdk] Offload bip39**
#1202 opened on Feb 28 by janniks...
- [Bundle Size][Wallet-SDK]**
#1237 opened 7 days ago by ahsan...
- Create stacks.js starter t**
#1123 opened on Oct 20, 2021 by ky...

New to Hiro, and are looking for ways to contribute?



github.com/hirosystems

A screenshot of a GitHub search results page. The search bar at the top contains the query "good first issue". Below the search bar, there are three search results, each showing an issue title, labels, and a priority indicator (P4). To the right of the search results is a sidebar titled "Filter by label" with a search input field. Under the "Unlabeled" section, there is a checked checkbox for "good first issue" with the subtext "Good for newcomers". Another checkbox for "before code freeze" is also present with the subtext "Should be completed before code fr...".

Author ▾ Label ▾ P

Filter by label X

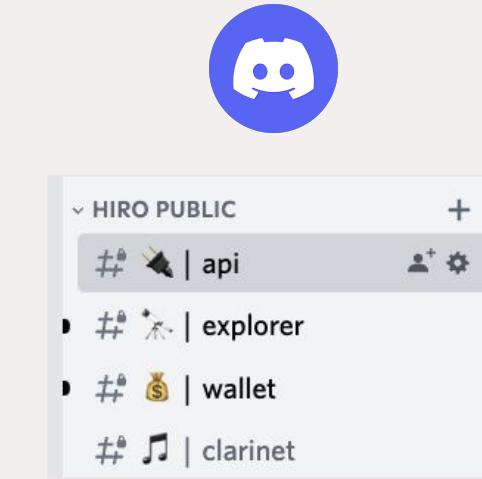
Filter labels

Unlabeled

✓ good first issue
Good for newcomers

before code freeze
Should be completed before code fr...

Are you wondering
how to engage with
Hiro or ask
questions?



Thank You

