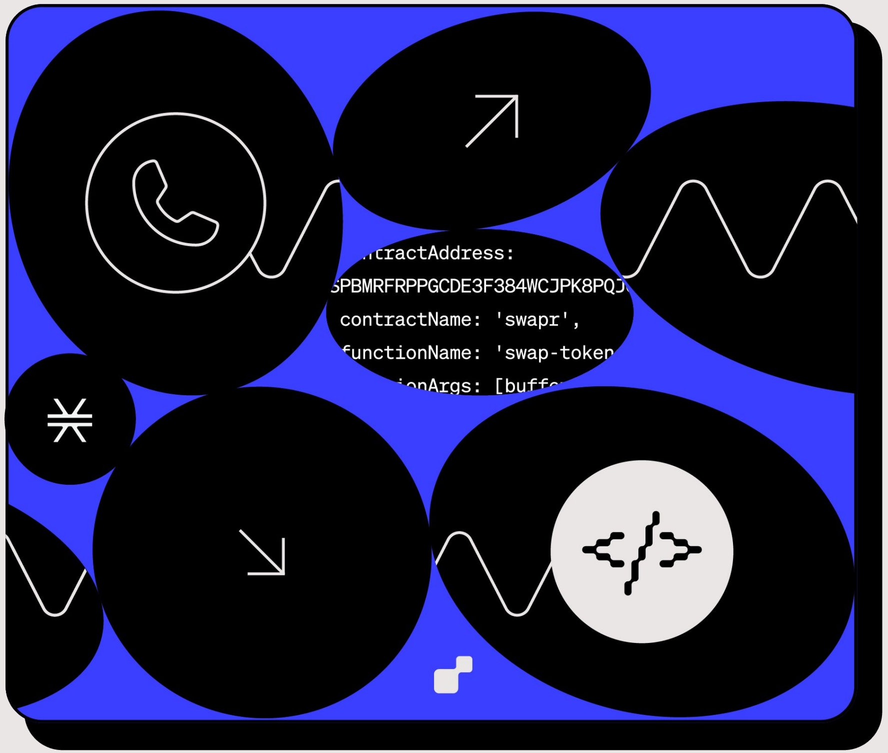
Hiro Monthly Developer Calls

#2 – May 26th, 2022



Hiro Developers

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

May 2022 Contributors

Thank you!

Clarinet

- @friedger
- @jhammond2012
- @fariedt
- @LNow
- @mcintyre94
- @moodmosaic

Web Wallet

- @pradel
- @mcintyre94

Stacks.js

- @mcintyre94
- @pradel

API

- @josecanhelp

Explorer

- @mcintyre94

Smart Contract of the Month

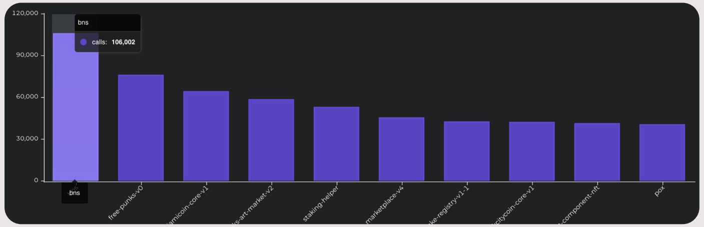
```
ntractAddress:
WCJPK8PQJ
',
en
```

```
ntı
SPBMRFR
contractNa
functionName: 'swap-token
ionArgs: [buff
```

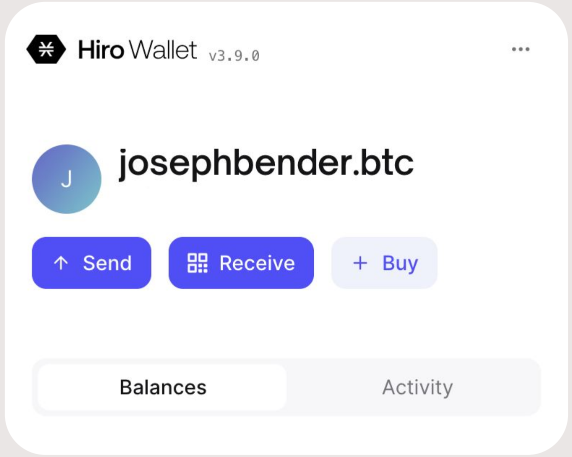
BNS = Blockchain Naming System

- Blockchain Naming System (BNS) is a network system that binds Stacks usernames to off-chain state without relying on any central points of control.
- Introduced in Stacks 1.0.
- Stacks accounts can be referenced with unique, user-owned, and human-readable usernames.
- BNS is implemented through the Clarity smart contracting language itself. The BNS contract provides a set of public and read-only methods to interact with the naming system.
- Developers can leverage the BNS package and the API to obtain data related to a username.

→ Link to Smart Contract in Stacks Explorer



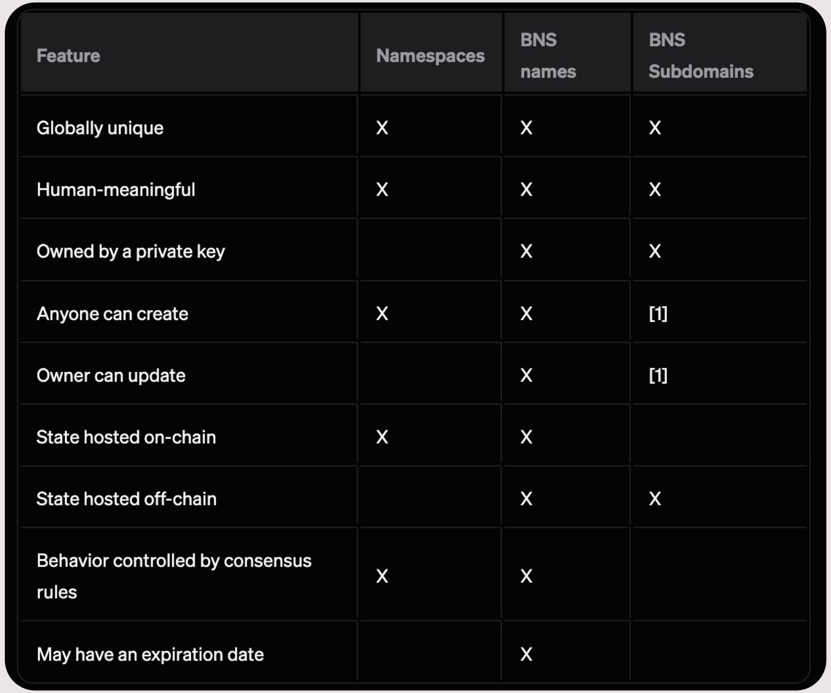
Most called contract in last 61,457 Stacks blocks



BNS name auto-populating in Hiro Web Wallet

BNS = Blockchain Naming System

- BNS names are organized into a global name hierarchy, with three layers:
  - **Namespaces**
    - Top Level domains
    - **.id**
  - **BNS Names**
    - Names recorded on blockchain
    - **Muneeb.id**
  - **Subdomains**
    - Off-chain, but anchored to blockchain
    - **joe.personal.id**



Feature	Namespaces	BNS names	BNS Subdomains
Globally unique	X	X	X
Human-meaningful	X	X	X
Owned by a private key		X	X
Anyone can create	X	X	[1]
Owner can update		X	[1]
State hosted on-chain	X	X	
State hosted off-chain		X	X
Behavior controlled by consensus rules	X	X	
May have an expiration date		X	

bns

SP000000...02Q6VF78

- 41 functions
- 46 variables
- 5 maps
- 1 token

Functions Requiring Transaction

- name-import
- name-preorder
- name-register
- name-renewal
- name-revoke
- name-transfer
- name-update
- namespace-preorder
- namespace-ready
- namespace-reveal
- namespace-revoke-function-price-edition
- namespace-update-function-price

Read-Only Functions

- can-name-be-registered — read_only
- can-namespace-be-registered — read_only
- can-receive-name — read_only
- check-name-ops-preconditions — read_only
- get-name-price — read_only
- get-namespace-price — read_only
- get-namespace-properties — read_only
- is-name-in-grace-period — read_only
- is-name-lease-expired — read_only
- name-resolve — read_only
- resolve-principal — read_only

Stacks API Names Endpoints:

- GET Get Namespace Price
- GET Get Name Price
- GET Get All Namespaces
- GET Get Namespace Names
- GET Get All Names
- GET Get Name Details
- GET Get Name History
- GET Get Zone File
- GET Get Historical Zone File
- GET Get Names Owned by Address

Defining the names NFT:

```
(define-non-fungible-token names { name: (buff 48), namespace: (buff 20) })
```

Price table for namespaces:

```
;; Price tables
(define-constant NAMESPACE_PRICE_TIERS (list
  u640000000000
  u64000000000 u64000000000
  u6400000000 u6400000000 u6400000000 u6400000000
  u640000000 u640000000 u640000000 u640000000 u640000000 u640000000 u640000000 u640000000 u640000000 u640000000 u640000000 u640000000 u640000000))
```

Map time

namespaces:

```
(define-map namespaces
  (buff 20)
  { namespace-import: principal,
    revealed-at: uint,
    launched-at: (optional uint),
    lifetime: uint,
    can-update-price-function: bool,
    price-function: {
      buckets: (list 16 uint),
      base: uint,
      coeff: uint,
      nonalpha-discount: uint,
      no-vowel-discount: uint
    }
  })
```

name properties:

```
(define-map name-properties
  { name: (buff 48), namespace: (buff 20) }
  { registered-at: (optional uint),
    imported-at: (optional uint),
    revoked-at: (optional uint),
    zonefile-hash: (buff 20) })
```

namespace preorders:

```
(define-map namespace-preorders
  { hashed-salted-namespace: (buff 20), buyer: principal }
  { created-at: uint, claimed: bool, stx-burned: uint })
```

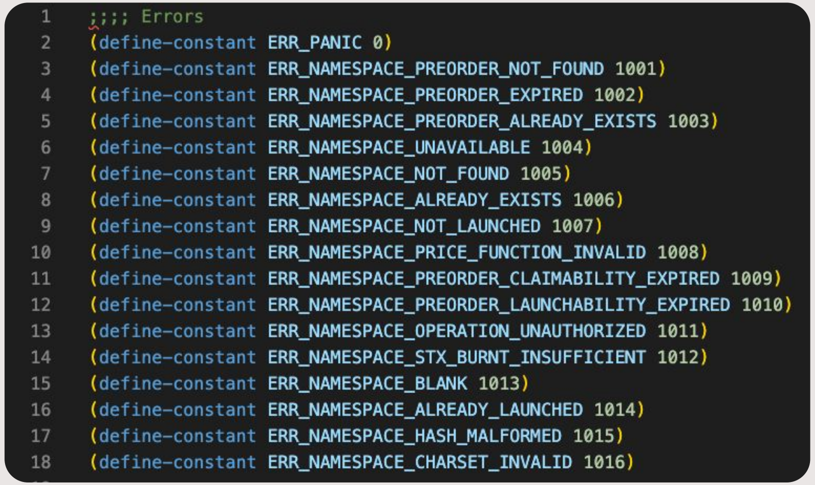
owner names:

```
;; Rule 1-1 -> 1 principal, 1 name
(define-map owner-name principal { name: (buff 48), namespace: (buff 20) })
```

name preorders:

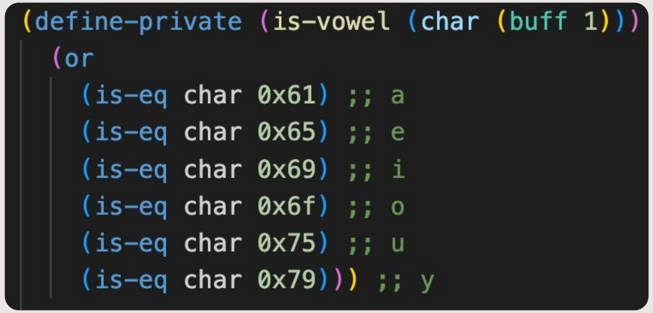
```
(define-map name-preorders
  { hashed-salted-fqn: (buff 20), buyer: principal }
  { created-at: uint, claimed: bool, stx-burned: uint })
```

Errors:

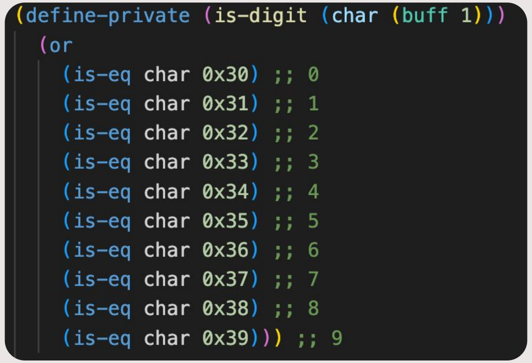


```
;;;; Errors
(define-constant ERR_PANIC 0)
(define-constant ERR_NAMESPACE_PREORDER_NOT_FOUND 1001)
(define-constant ERR_NAMESPACE_PREORDER_EXPIRED 1002)
(define-constant ERR_NAMESPACE_PREORDER_ALREADY_EXISTS 1003)
(define-constant ERR_NAMESPACE_UNAVAILABLE 1004)
(define-constant ERR_NAMESPACE_NOT_FOUND 1005)
(define-constant ERR_NAMESPACE_ALREADY_EXISTS 1006)
(define-constant ERR_NAMESPACE_NOT_LAUNCHED 1007)
(define-constant ERR_NAMESPACE_PRICE_FUNCTION_INVALID 1008)
(define-constant ERR_NAMESPACE_PREORDER_CLAIMABILITY_EXPIRED 1009)
(define-constant ERR_NAMESPACE_PREORDER_LAUNCHABILITY_EXPIRED 1010)
(define-constant ERR_NAMESPACE_OPERATION_UNAUTHORIZED 1011)
(define-constant ERR_NAMESPACE_STX_BURNT_INSUFFICIENT 1012)
(define-constant ERR_NAMESPACE_BLANK 1013)
(define-constant ERR_NAMESPACE_ALREADY_LAUNCHED 1014)
(define-constant ERR_NAMESPACE_HASH_MALFORMED 1015)
(define-constant ERR_NAMESPACE_CHARSET_INVALID 1016)
```

Character checks:

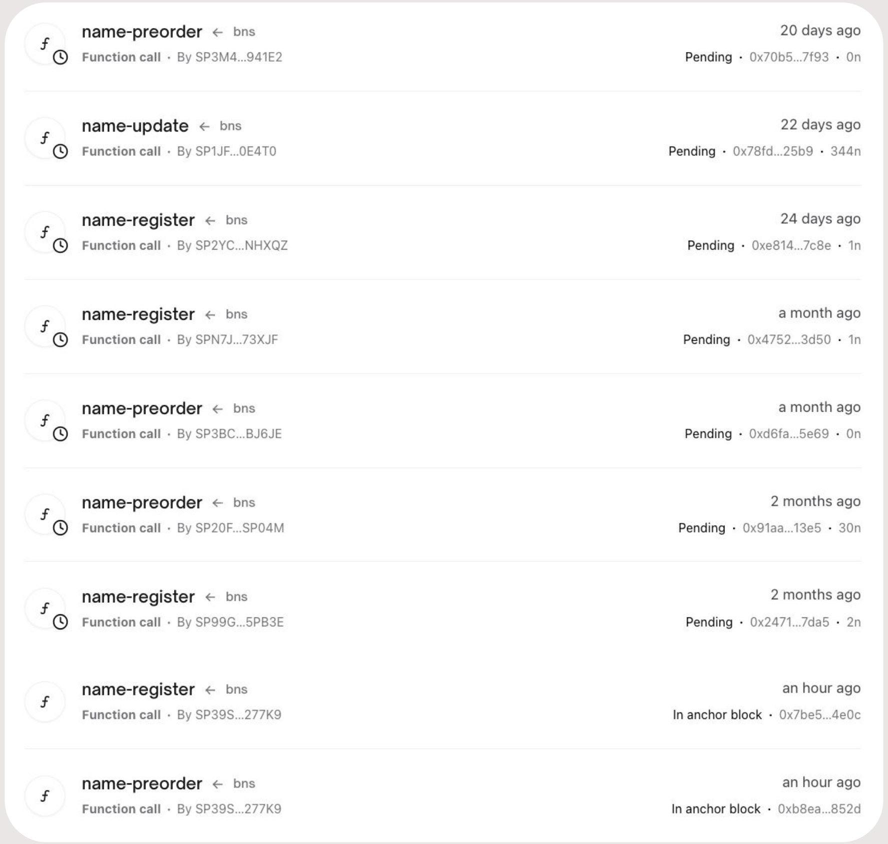


```
(define-private (is-vowel (char (buff 1)))
  (or
    (is-eq char 0x61) ;; a
    (is-eq char 0x65) ;; e
    (is-eq char 0x69) ;; i
    (is-eq char 0x6f) ;; o
    (is-eq char 0x75) ;; u
    (is-eq char 0x79))) ;; y
```



```
(define-private (is-digit (char (buff 1)))
  (or
    (is-eq char 0x30) ;; 0
    (is-eq char 0x31) ;; 1
    (is-eq char 0x32) ;; 2
    (is-eq char 0x33) ;; 3
    (is-eq char 0x34) ;; 4
    (is-eq char 0x35) ;; 5
    (is-eq char 0x36) ;; 6
    (is-eq char 0x37) ;; 7
    (is-eq char 0x38) ;; 8
    (is-eq char 0x39))) ;; 9
```

name-preorder and name-register function calls in Stacks Explorer



name-preorder ← bns
Function call · By SP3M4...941E2
20 days ago
Pending · 0x70b5...7f93 · 0n

name-update ← bns
Function call · By SP1JF...0E4T0
22 days ago
Pending · 0x78fd...25b9 · 344n

name-register ← bns
Function call · By SP2YC...NHXQZ
24 days ago
Pending · 0xe814...7c8e · 1n

name-register ← bns
Function call · By SPN7J...73XJF
a month ago
Pending · 0x4752...3d50 · 1n

name-preorder ← bns
Function call · By SP3BC...BJ6JE
a month ago
Pending · 0xd6fa...5e69 · 0n

name-preorder ← bns
Function call · By SP20F...SP04M
2 months ago
Pending · 0x91aa...13e5 · 30n

name-register ← bns
Function call · By SP99G...5PB3E
2 months ago
Pending · 0x2471...7da5 · 2n

name-register ← bns
Function call · By SP39S...277K9
an hour ago
In anchor block · 0x7be5...4e0c

name-preorder ← bns
Function call · By SP39S...277K9
an hour ago
In anchor block · 0xb8ea...852d

name-preorder function

Preorders a name by telling all BNS nodes the salted hash of the BNS name. It pays the registration fee to the namespace owner's designated address.

- Salted hash: adding random data to force uniqueness and increase complexity

Defining the function

Beginning the let expression

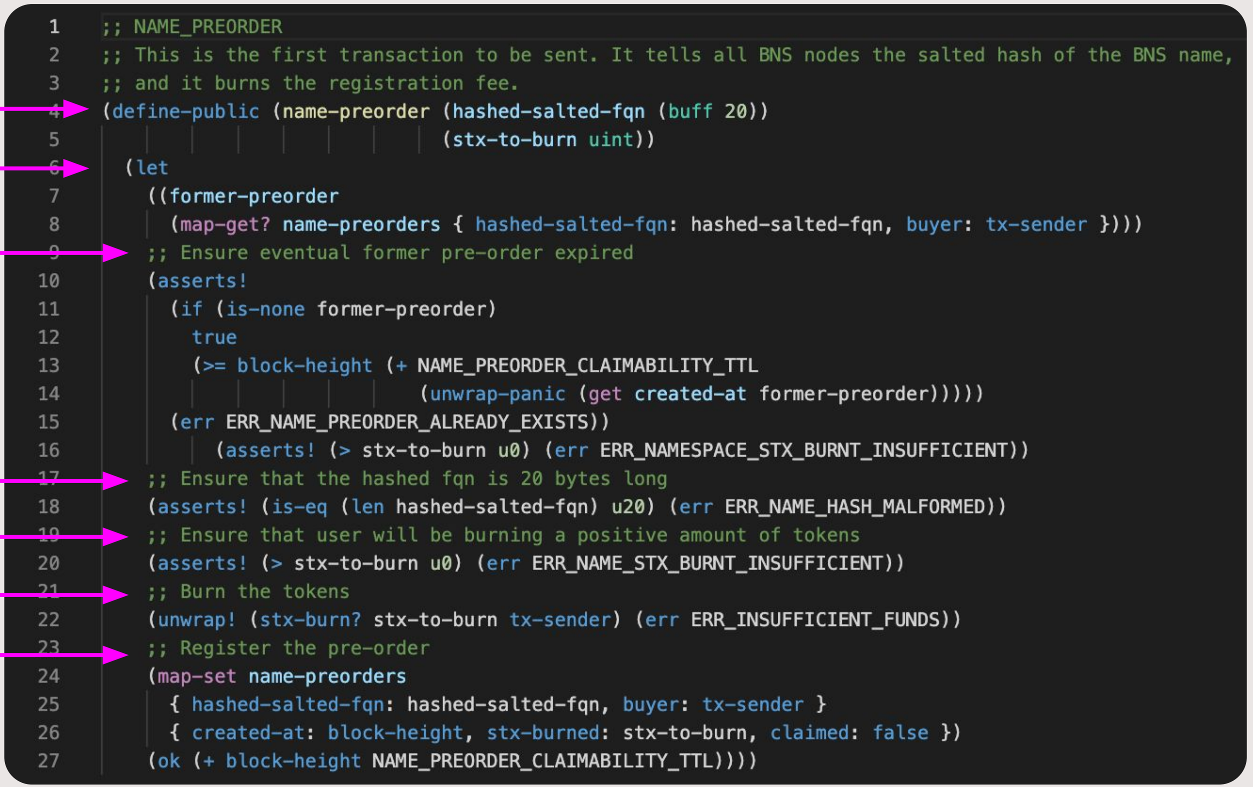
Ensure any former preorders have expired

Check that the provided salted hash function is the correct length

Checking that txn is burning non-zero amount of STX

Execute the burn of the tokens!

Finalize the preorder



```
;; NAME_PREORDER
;; This is the first transaction to be sent. It tells all BNS nodes the salted hash of the BNS name,
;; and it burns the registration fee.
(define-public (name-preorder (hashed-salted-fqn (buff 20))
                              (stx-to-burn uint))
 (let
   ((former-preorder
     (map-get? name-preorders { hashed-salted-fqn: hashed-salted-fqn, buyer: tx-sender })))
   ;; Ensure eventual former pre-order expired
   (asserts!
     (if (is-none former-preorder)
       true
       (>= block-height (+ NAME_PREORDER_CLAIMABILITY_TTL
                          (unwrap-panic (get created-at former-preorder)))))
     (err ERR_NAME_PREORDER_ALREADY_EXISTS))
       (asserts! (> stx-to-burn u0) (err ERR_NAMESPACE_STX_BURNT_INSUFFICIENT))
   ;; Ensure that the hashed fqn is 20 bytes long
   (asserts! (is-eq (len hashed-salted-fqn) u20) (err ERR_NAME_HASH_MALFORMED))
   ;; Ensure that user will be burning a positive amount of tokens
   (asserts! (> stx-to-burn u0) (err ERR_NAME_STX_BURNT_INSUFFICIENT))
   ;; Burn the tokens
   (unwrap! (stx-burn? stx-to-burn tx-sender) (err ERR_INSUFFICIENT_FUNDS))
   ;; Register the pre-order
   (map-set name-preorders
     { hashed-salted-fqn: hashed-salted-fqn, buyer: tx-sender }
     { created-at: block-height, stx-burned: stx-to-burn, claimed: false })
   (ok (+ block-height NAME_PREORDER_CLAIMABILITY_TTL))))
```

let = function that binds a list of variables to expressions

asserts! = checking boolean

name-register function

Reveals the salt and the name to all BNS nodes, and assigns the name an initial public key hash and zone file hash.

Defining the function

Begin let expression

Check if the name can be registered

Ensure the namespace has been previously launched

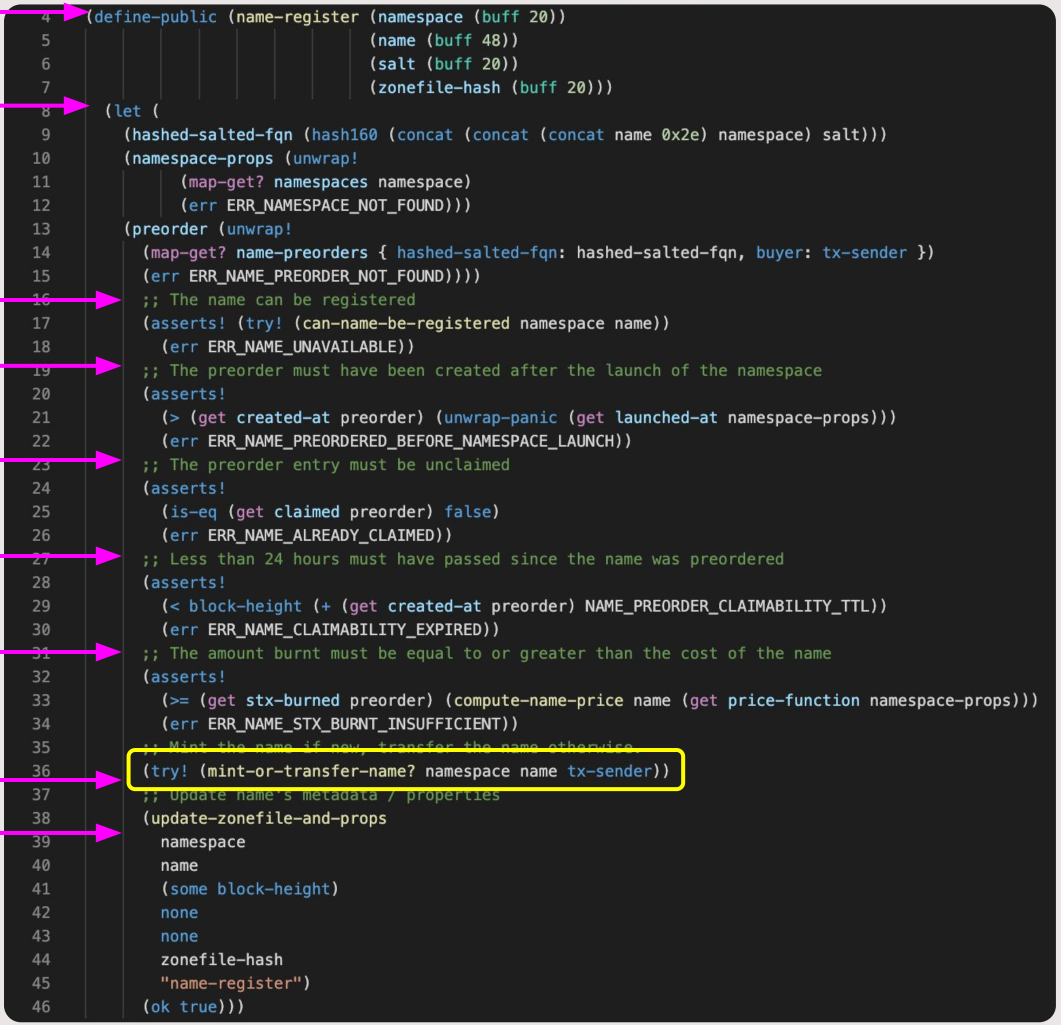
Check that the preorder is unclaimed

Ensure it has been less than 24 hours since preorder

The amount of burnt STX must be >= cost of name

Once all checks are complete, mint or transfer the name!

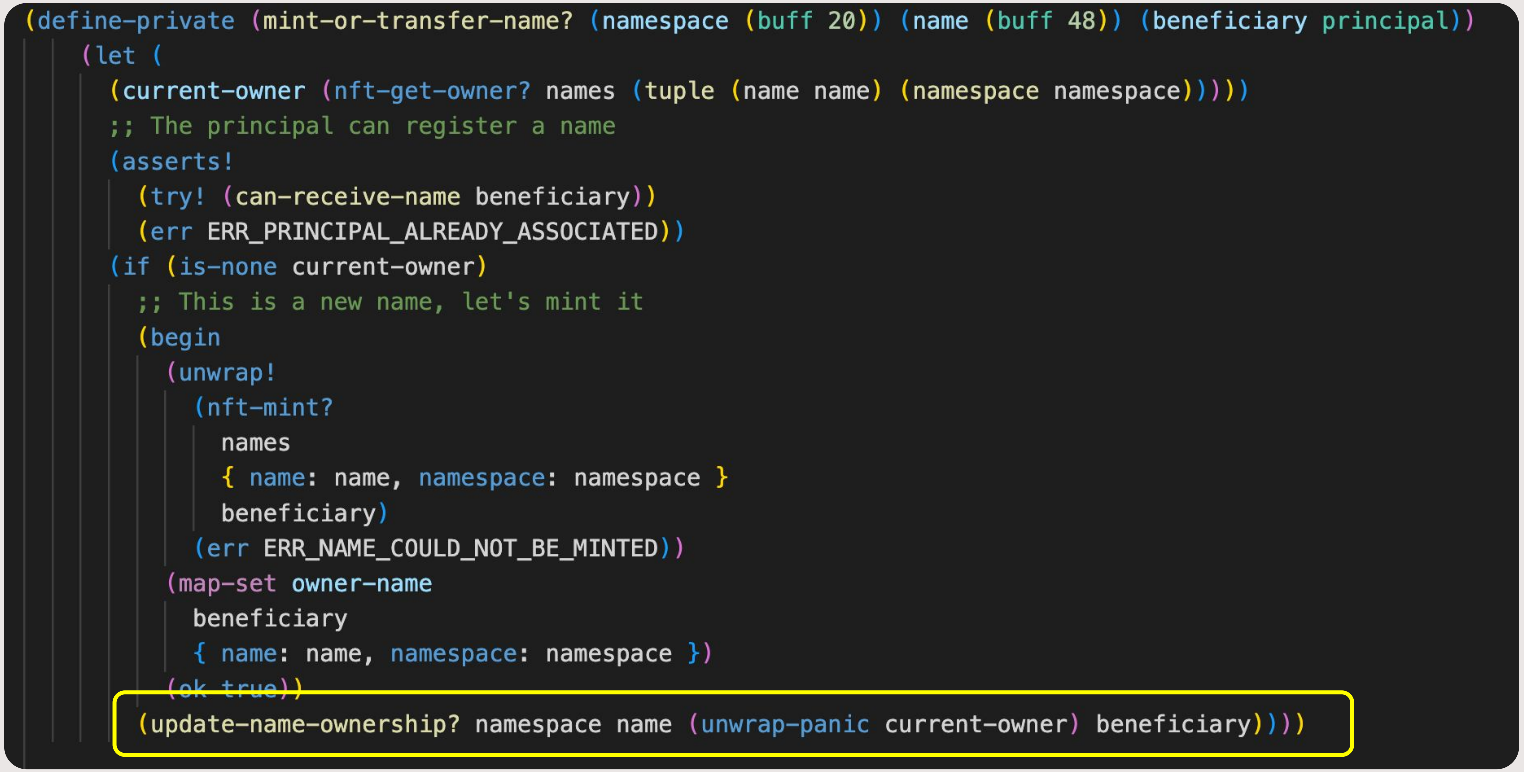
Finally, update metadata of name



```
(define-public (name-register (namespace (buff 20))
                              (name (buff 48))
                              (salt (buff 20))
                              (zonefile-hash (buff 20)))
  (let (
    (hashed-salted-fqn (hash160 (concat (concat (concat name 0x2e) namespace) salt)))
    (namespace-props (unwrap!
          (map-get? namespaces namespace)
          (err ERR_NAMESPACE_NOT_FOUND)))
    (preorder (unwrap!
      (map-get? name-preorders { hashed-salted-fqn: hashed-salted-fqn, buyer: tx-sender })
      (err ERR_NAME_PREORDER_NOT_FOUND))))
      ;; The name can be registered
      (asserts! (try! (can-name-be-registered namespace name))
        (err ERR_NAME_UNAVAILABLE))
      ;; The preorder must have been created after the launch of the namespace
      (asserts!
        (> (get created-at preorder) (unwrap-panic (get launched-at namespace-props)))
        (err ERR_NAME_PREORDERED_BEFORE_NAMESPACE_LAUNCH))
      ;; The preorder entry must be unclaimed
      (asserts!
        (is-eq (get claimed preorder) false)
        (err ERR_NAME_ALREADY_CLAIMED))
      ;; Less than 24 hours must have passed since the name was preordered
      (asserts!
        (< block-height (+ (get created-at preorder) NAME_PREORDER_CLAIMABILITY_TTL))
        (err ERR_NAME_CLAIMABILITY_EXPIRED))
      ;; The amount burnt must be equal to or greater than the cost of the name
      (asserts!
        (>= (get stx-burned preorder) (compute-name-price name (get price-function namespace-props)))
        (err ERR_NAME_STX_BURNT_INSUFFICIENT))
      ;; Mint the name if new, transfer the name otherwise.
      (try! (mint-or-transfer-name? namespace name tx-sender))
      ;; Update name's metadata / properties
      (update-zonefile-and-props
        namespace
        name
        (some block-height)
        none
        none
        zonefile-hash
        "name-register")
      (ok true)))
```

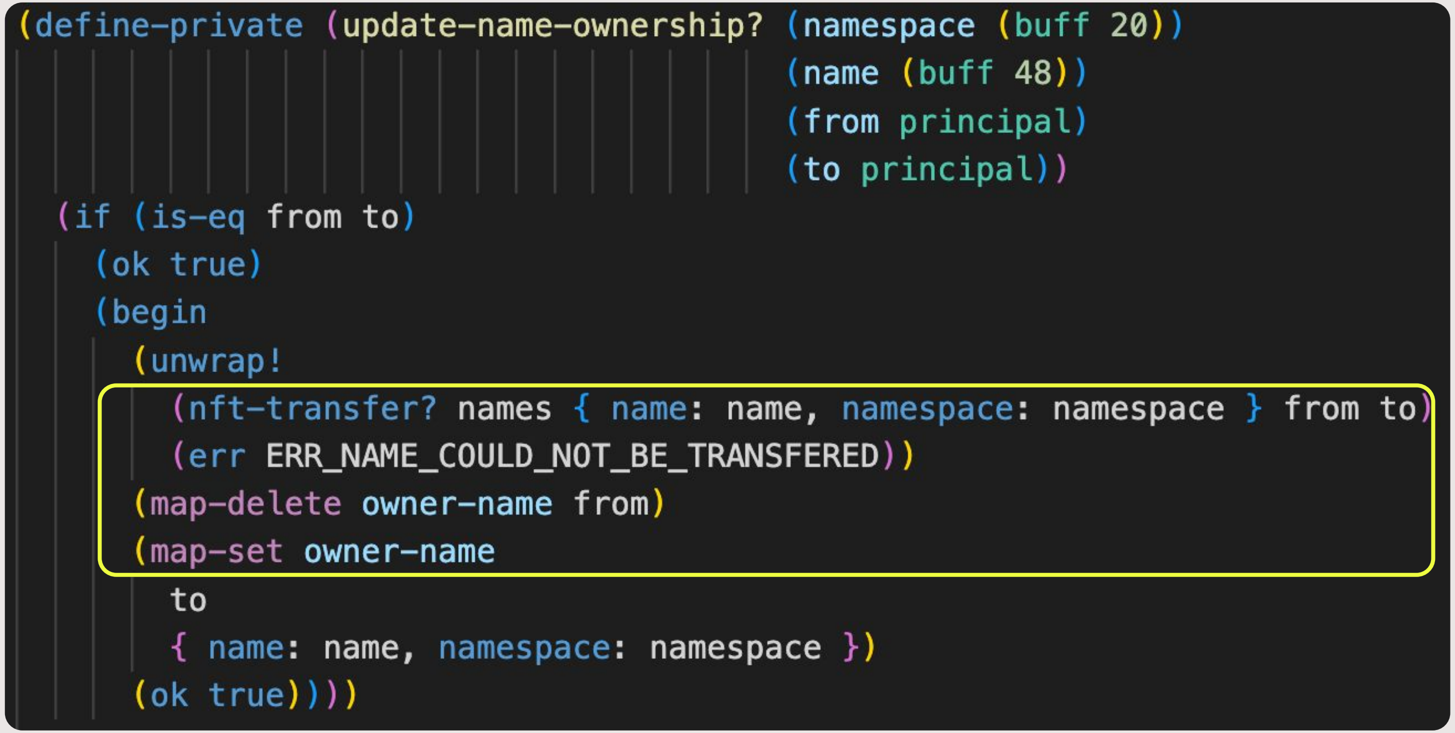
Hiro Developers

mint-or-transfer-name? function



```
(define-private (mint-or-transfer-name? (namespace (buff 20)) (name (buff 48)) (beneficiary principal))
    (let (
      (current-owner (nft-get-owner? names (tuple (name name) (namespace namespace)))))
      ;; The principal can register a name
      (asserts!
        (try! (can-receive-name beneficiary))
        (err ERR_PRINCIPAL_ALREADY_ASSOCIATED))
      (if (is-none current-owner)
        ;; This is a new name, let's mint it
        (begin
          (unwrap!
            (nft-mint?
              names
              { name: name, namespace: namespace }
              beneficiary)
            (err ERR_NAME_COULD_NOT_BE_MINTED))
          (map-set owner-name
            beneficiary
            { name: name, namespace: namespace })
          (ok true))
        (update-name-ownership? namespace name (unwrap-panic current-owner) beneficiary))))
```

update-name-ownership? function



```
(define-private (update-name-ownership? (namespace (buff 20))
                                        (name (buff 48))
                                        (from principal)
                                        (to principal))
  (if (is-eq from to)
    (ok true)
    (begin
      (unwrap!
        (nft-transfer? names { name: name, namespace: namespace } from to)
        (err ERR_NAME_COULD_NOT_BE_TRANSFERED))
      (map-delete owner-name from)
      (map-set owner-name
        to
        { name: name, namespace: namespace })
      (ok true))))
```

Learn more about BNS on Stacks:

→ hiro.so/blog → 🔍 bns **How to get started**

→ docs.stacks.co/build-apps/references/bns **Stacks Docs**

→docs.stacks.co/noteworthy-contracts/ bns-contract **BNS Contract**

→stacks-js-git-master-blockstack.vercel.app /modules/bns.html **Stacks.js reference**

Community Topics

Topic 1: Arbitrary Message Signing

Use cases:

DAOs with off-chain voting, governance, compliance, and auditing processes (e.g. when multiple signatures required)

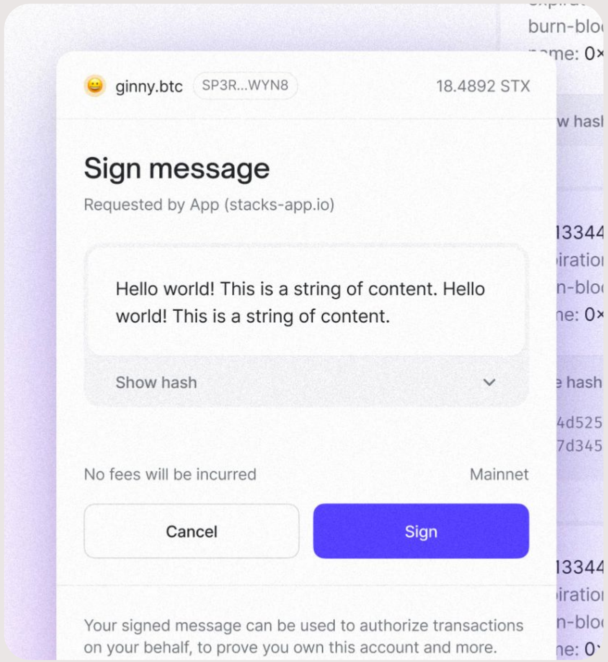
Settling orders for **decentralized exchange** in off-chain order book or L2 / hyperchain for later settlement on-chain / with Stacks mainnet

Server-side auth such as when adding Stacks wallet as option for sign in to Web 2.0 product or bridging auth with desktop apps (e.g. games)

Community Topics

Topic 1: Arbitrary Message Signing

Solution:


ginny.btc SP3R...WYN8 18.4892 STX

Sign message

Requested by App (stacks-app.io)

Hello world! This is a string of content. Hello world! This is a string of content.

Your signed message can be used to authorize transactions on your behalf, to prove you own this account and more.

→ Arbitrary Message Signing in Hiro Wallet

hiro.so/blog

HYPERCHAINS COMMS PLAN

01
MAY 26th [LIVE]
Community Call
Timeline updates, Q&A
Call for Beta Testers

02
MAY 26th
Primer Video ✓
Timeline updates, links back to the Primer Video
Call for Beta Testers

03
MAY 26th
Forum Post ✓
Timeline updates, links back to the Primer Video
Call for Beta Testers

04
JUNE 7th
Blog Post
Deep Dive to address most often asked questions on Architecture, Use cases, performance, timelines
Call for Beta Testers

05
JUNE 7th [LIVE]
Twitter Spaces
Conversation with community, active Stacks developers, Q&A
Call for Beta Testers

06
JUNE 14th [LIVE]
Discord Chat
Conversation with community, active Stacks developers
Call for Beta Testers

HYPERCHAINS ROAD TO MAINNET

TESTNET LAUNCH

- TestNet Deployment
- NFT Use Case
- BFT Changes
- NFT Use Case Testing

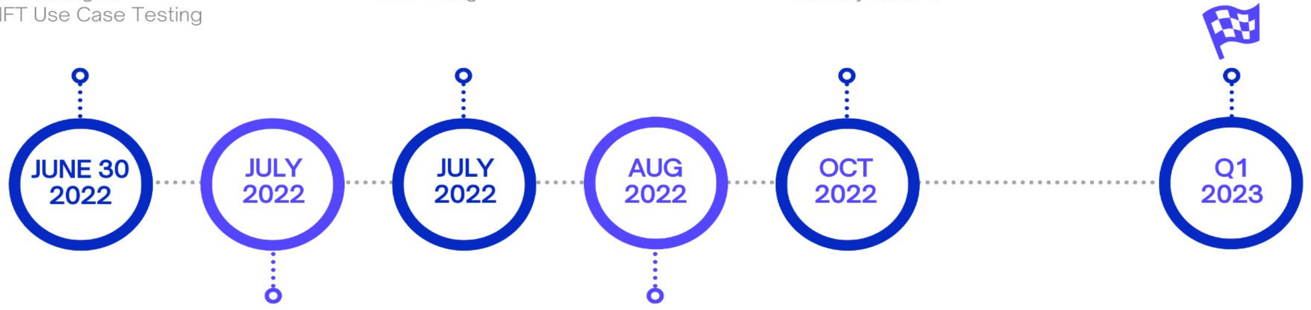
INTEGRATIONS

- API Integration and TestNet Setup
- Wallet Integration
- UX Testing

REVIEWS/AUDITS

- Legal Review
- Code Audits
- Security Review

MAINNET LAUNCH



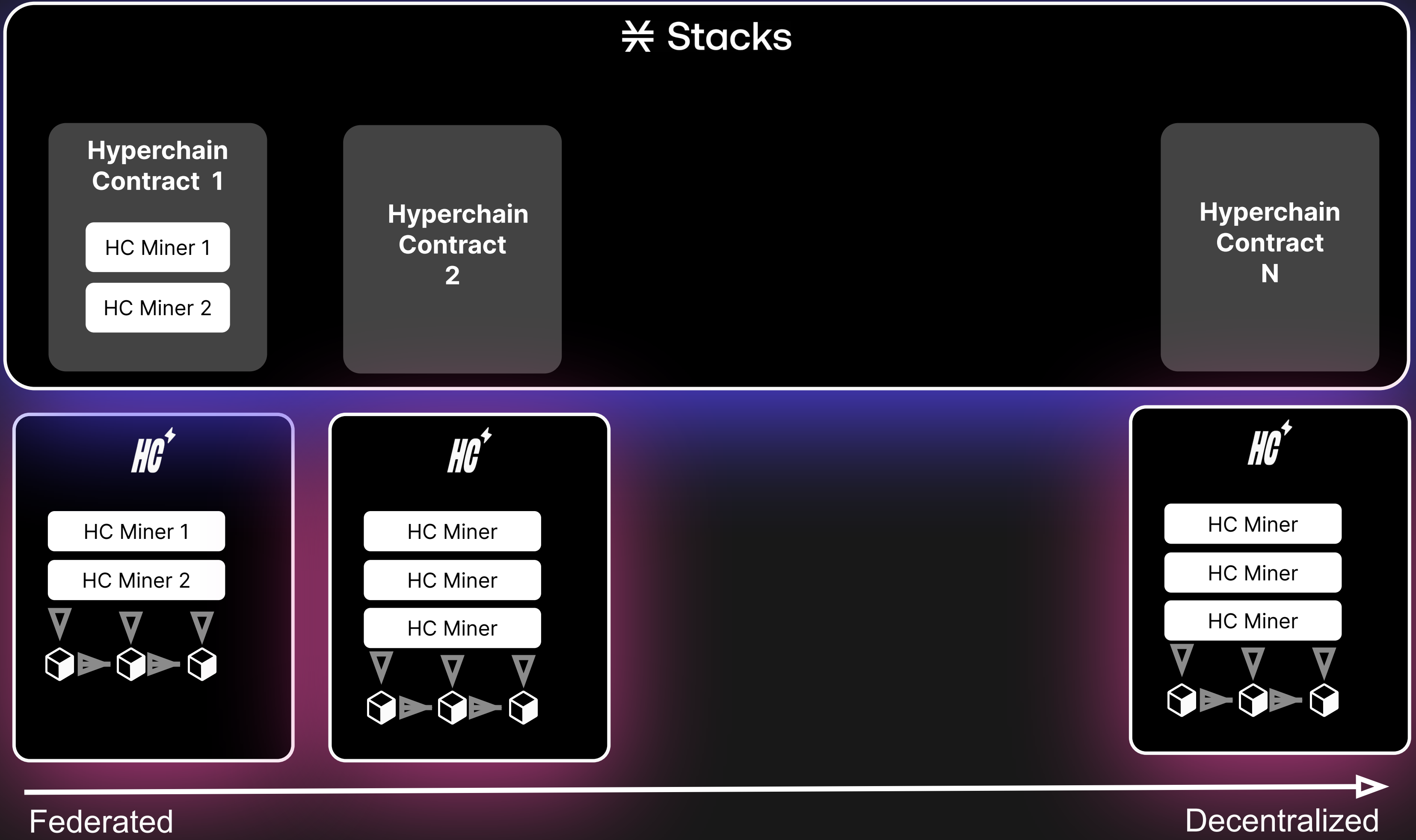
2.1 READINESS

- Upgrade Hyperchains Contract
- TestNet SetUp
- Commence Beta Testing

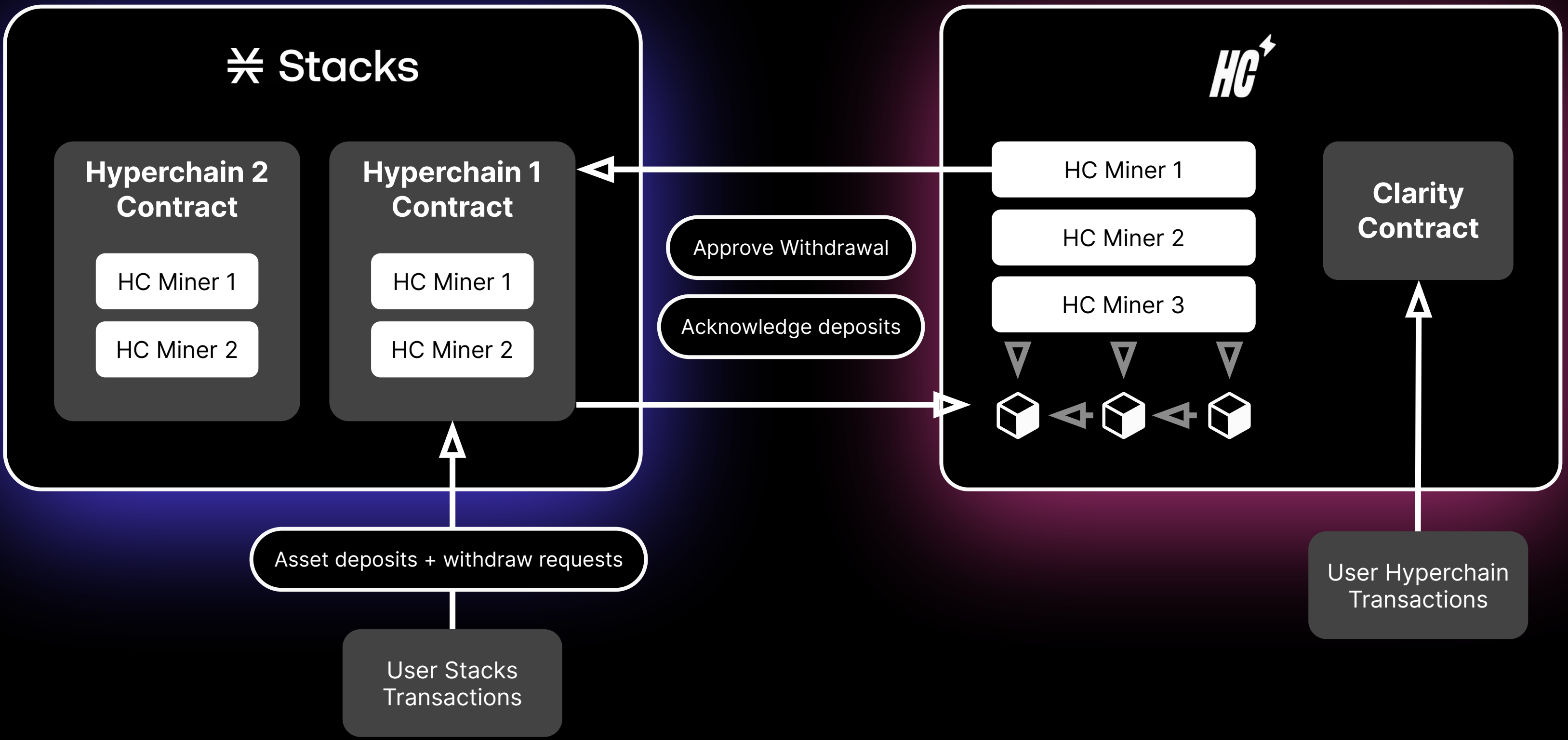
MINERS SETUP

- Instructions, Docs, Setup
- Layout Incentives details

Multiverse of Hyperchains



Hyperchains Architecture



Topic 3: Notifications for Token Metadata Updates

Use cases:

Create a fully audited new token on Stacks, and change its metadata such as token symbol or name later after the launch.

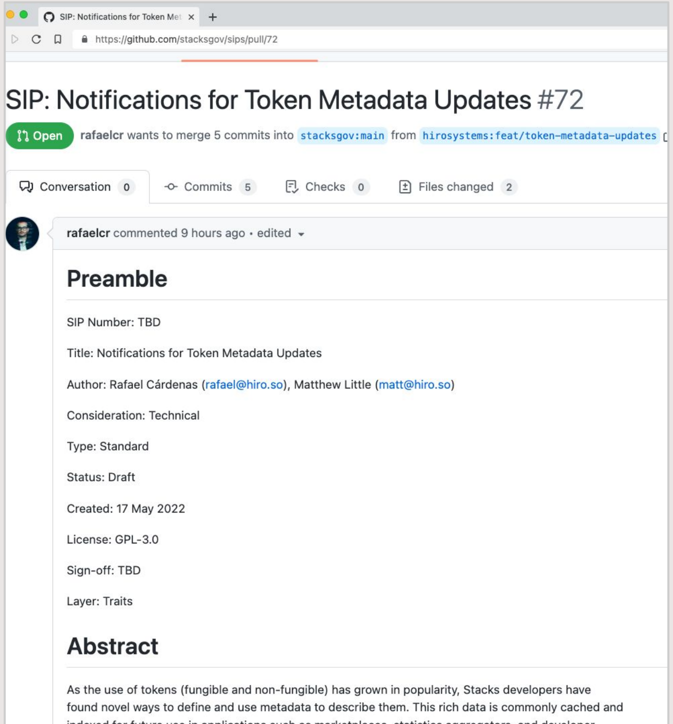
Launch a new NFT project with 'placeholders', but want to delay the reveal of artwork & properties of the NFTs.

Problem:

There is no standard method to notify the metadata state changes, so developer tools like Stacks Blockchain API cannot react to and reconcile the metadata changes.

Topic 3: Notifications for Token Metadata Updates

Solution:


SIP: Notifications for Token Met
https://github.com/stacksgov/sips/pull/72

SIP: Notifications for Token Metadata Updates #72

Open rafaelcr wants to merge 5 commits into stacksgov:main from hirosystems:feat/token-metadata-updates

Conversation 0
Commits 5
Checks 0
Files changed 2

rafaelcr commented 9 hours ago • edited

Preamble

SIP Number: TBD

Title: Notifications for Token Metadata Updates

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

Consideration: Technical

Type: Standard

Status: Draft

Created: 17 May 2022

License: GPL-3.0

Sign-off: TBD

Layer: Traits

Abstract

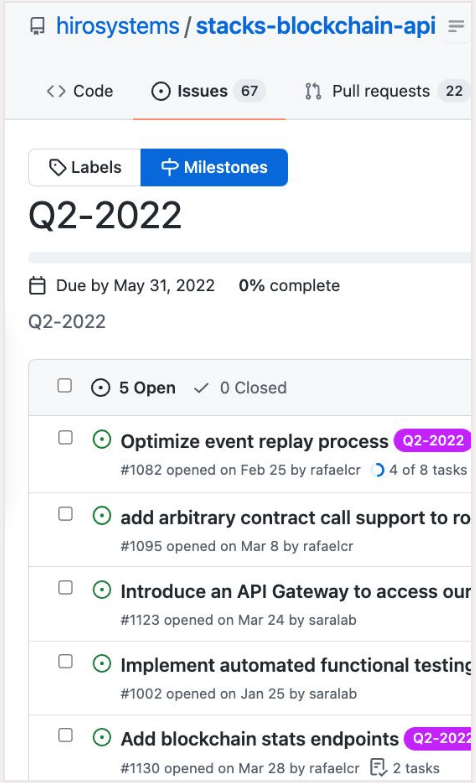
As the use of tokens (fungible and non-fungible) has grown in popularity, Stacks developers have found novel ways to define and use metadata to describe them. This rich data is commonly cached and

→ Stacks Improvement Proposal #72

github.com/stacksgov/sips/pull/72

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

github.com/hirosystems


hirosystems / stacks-blockchain-api

Code | Issues 67 | Pull requests 22

Labels | Milestones

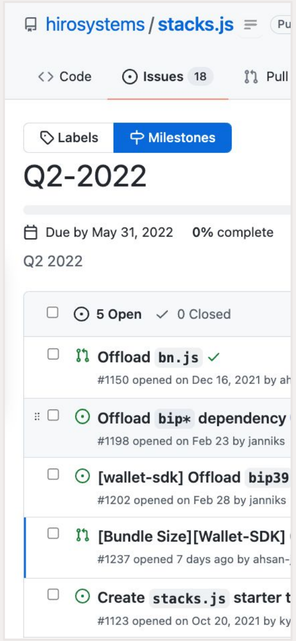
Q2-2022

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

Code | Issues 18 | Pull

Labels | Milestones

Q2-2022

Due by May 31, 2022 0% complete

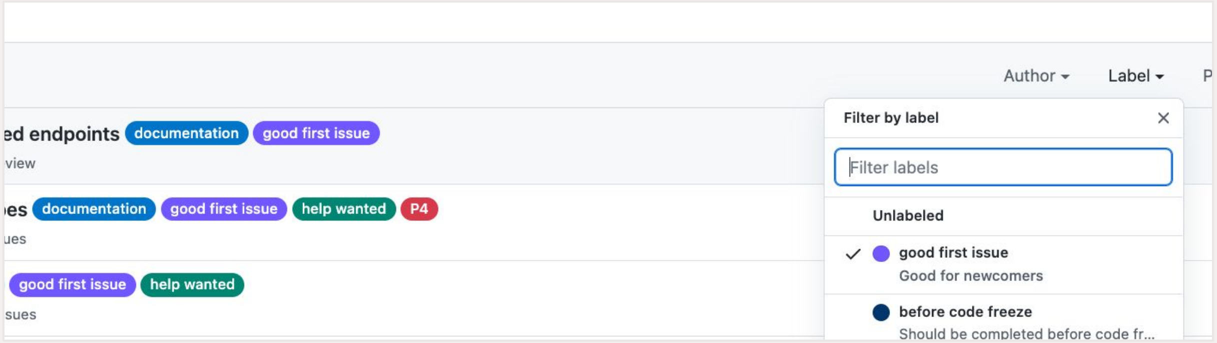
Q2 2022

5 Open ✓ 0 Closed

- Offload `bn.js` ✓
  #1150 opened on Dec 16, 2021 by ah
- 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



Author ▾ Label ▾ P

ed endpoints documentation good first issue
view

es documentation good first issue help wanted P4
ues

good first issue help wanted
sues

Filter by label

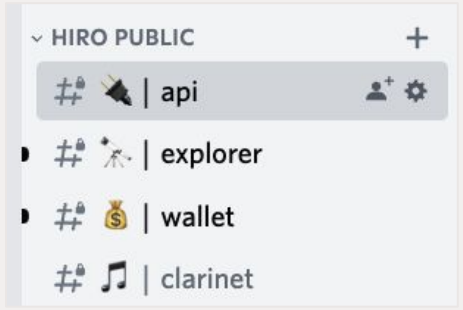
Filter labels

Unlabeled

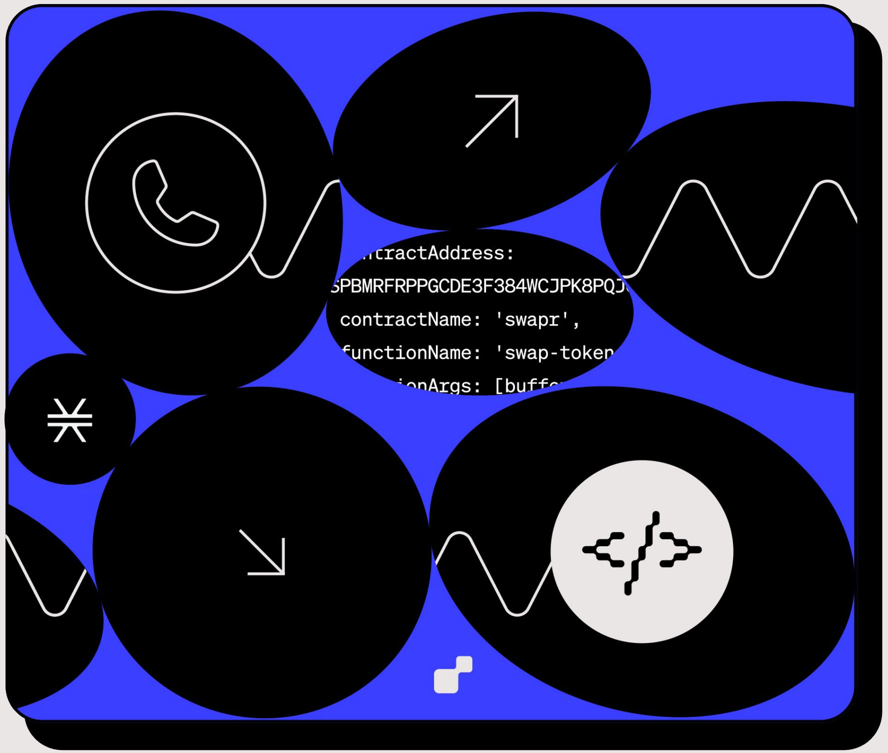
✓ ● good first issue
Good for newcomers

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

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



Thank You



Hiro Developers