Trust-Machines / stacks-pyth-bridge Public

forked from hirosystems/stacks-pyth-bridge

Code   Pull requests   Actions   Projects   Security   Insights

2 Branches   0 Tags   Go to file   Go to file   Code

This branch is 55 commits ahead of, 1 commit behind `hirosystems/stacks-pyth-bridge:main`.

blaizew Merge pull request #17 from Trust-Machines/update_docs

6568456 · 2 weeks ago

.github/workflows	chore: update cbtc example (h...	11 months ago
.vscode	feat: init tests with clarinet sdk...	2 years ago
audits	Add files via upload	last month
contracts	fix QA-10	2 months ago
deployments	deploy pyth to mainnet	last month
dockerfiles	chore: update rust version in d...	2 years ago
docs	chore: update documentation	2 years ago
example	fix R-QA-01	2 months ago
settings	chore: update deployment plans	2 years ago
unit-tests	update helper functions	2 months ago
.gitattributes	chore: update gitattributes	2 years ago
.gitignore	chore: update gitignore	2 years ago
.prettierrc	chore: apply prettier	2 years ago
Clarinet.toml	fix R-QA-01	2 months ago
LICENSE	chore: add Apache2 License.	2 years ago
README.md	update readme with latest depl...	2 weeks ago
package-lock.json	update deployment	2 months ago
package.json	fix test	4 months ago
tsconfig.json	tests: use vitest and clarinet-sdk	2 years ago
vitest.config.js	chore: update cbtc example (h...	11 months ago

About

Retrieve trading pairs (BTC-USD, STX-USD, etc) from Clarity smart contracts.

- Readme
- Apache-2.0 license
- Activity
- Custom properties
- 4 stars
- 1 watching
- 0 forks

Report repository

Releases

No releases published

Packages

No packages published

Languages

● Clarity 68.5%   ● Rust 30.7%
● Other 0.8%

README   Apache-2.0 license

Stacks <> Pyth - edited by Granite team

```
   /     /   ▶ Stacks Pyth Bridge
  / --- /      Bridging Pyth price feeds to the Stacks blockchain.
 /     /       Retrieve trading pairs (BTC-USD, STX-USD, etc.) from Clarity smart contracts.
```

Introduction   # Features   # Quick Start   # Documentation   # Contribute

Introduction

Status: Beta

The Pyth protocol integration is available as a Beta on both testnet and mainnet networks, to help developers test, give feedback, and ensure the reliability and stability of the integration.

Stacks is a blockchain linked to Bitcoin by its consensus mechanism that spans the two chains, called Proof of Transfer. This enables Stacks to leverage Bitcoin's security and enables Stacks apps to use Bitcoin's state. Stacks is a Bitcoin layer that enables decentralized apps and smart contracts.

Pyth Network is an oracle that publishes financial market data to multiple blockchains. The market data is contributed by over 80 first-party publishers, including some of the biggest exchanges and market-making firms in the world. Pyth offers price feeds for several asset classes, including US equities, commodities, and cryptocurrencies. Each price feed publishes a robust aggregate of publisher prices that updates multiple times per second. Price feeds are available on multiple blockchains and can be used in off-chain applications.

Wormhole is a decentralized attestation engine that leverages its network of guardians to trustlessly bridge information between the chains it supports. Wormhole has a simple, elegant, and pragmatic design that has enabled it to be the first real solution to ship to market and has received wide recognition and support from its member chains.

Setup and and run the tests

The contracts are developed in Clarity and use clarinet-sdk for its test harnessing.

Git clone and compile **stacks-pyth-relayer**

```
$ git clone https://github.com/hirosystems/stacks-pyth-bridge.git
$ cd stacks-pyth-bridge
$ npm install
$ npm test
```

Consuming price feeds

Latest Deployments

network	address
testnet	ST20M5GABDT6WYJHXBT5CDH4501V1Q65242SPRMXH.pyth-oracle-v3
mainnet	SP3R4F6C1J3JQWWCVZ3S7FRRYPMYG6ZW6RZK31FXY.pyth-oracle-v3

Onchain

The `pyth-oracle-v3` contract is exposing the following method:

```
(define-public (read-price-feed
    (price-feed-id (buff 32))
    (pyth-storage-address <pyth-storage-trait>)))
```

That can be consumed with the following invocation:

```
(contract-call?
    'SP3R4F6C1J3JQWWCVZ3S7FRRYPMYG6ZW6RZK31FXY.pyth-oracle-v3                ;; Address of the helper contract
    read-price-feed
    0xe62df6c8b4a85fe1a67db44dc12de5db330f7ac66b72dc658afedf0f4a415b43      ;; BTC-USD price identifier
    'SP3R4F6C1J3JQWWCVZ3S7FRRYPMYG6ZW6RZK31FXY.pyth-storage-v3)
```

The authenticity of the price feeds is verified during their ingestion, making the cost of queries as light as possible.

Each Pyth Network price feed is referred to via a unique ID. Price feeds also have different IDs in mainnets than testnets or devnets. The full list of price feeds is listed on the pyth.network website. The price feed IDs page lists the ID of each available price feed on every chain where they are available. To use a price feed on-chain, look up its ID using these pages, then store the feed ID in your program for price feed queries.

Price Feed usage and best practices are described on the pyth.network developer documentation website.

Prices currently supported on Testnet and Mainnet

The full list of prices is available here.

Offchain

For every new price recorded and stored on chain, the `pyth-storage-v1` is emitting an event with the following shape:

```
{
  type: "price-feed",
  action: "updated",
  data: {
    price-identifier: 0xec7a775f46379b5e943c3526b1c8d54cd49749176b0b98e02dde68d1bd335c17,
    price: 46098556,
    conf: u37359,
    ema-price: 46167004,
    ema-conf: u36191,
    expo: -8,
    publish-time: u1695751649,
    prev-publish-time: u1695751648
  }
}
```

These events can be observed using Chainhook, using the `print` predicates.

Updating price feeds

Pyth Network uses a pull price update model that is slightly different from other oracles you may be more familiar with. Most oracles today use a push model, where the oracle runs an off-chain process that continuously sends transactions to update an on-chain price. In contrast, Pyth Network does not operate an off-chain process that pushes prices on-chain. Instead, it delegates this work to Pyth Network users.

Hermes is a web service that listens to the Pythnet and the Wormhole Network for Pyth price updates, and serves them via a convenient web API. It provides Pyth's latest price update data format that are more cost-effective to verify and use on-chain. Hermes allows users to easily query for recent price updates via a REST API, or subscribe to a websocket for streaming updates. The Pyth Network's Javascript SDKs connect to an instance of Hermes to fetch price updates.

```
$ curl https://hermes.pyth.network/api/latest_price_feeds?ids[]=ec7a775f46379b5e943c3526b1c8d54cd49749176b0b98e02dde
| jq -r '.[0]'.vaa \
| base64 --decode \
| hexdump -ve '1/1 "%.2x"'

504e41550100000003b8...a7b10321ad7c2404a910
```

This sequence of bytes is a Verified Action Approvals (VAA) including the price information including its cryptographic elements helping the Pyth contract ensuring the authenticity of the data.

This VAA can be encoded as a Clarity buffer, and submitted to the Pyth contract using the following:

```
(contract-call?
    'SP3R4F6C1J3JQWWCVZ3S7FRRYPMYG6ZW6RZK31FXY.pyth-oracle-v3   ;; Address of the helper contract
    verify-and-update-price
    0x504e41550100000003b8...a7b10321ad7c2404a910               ;; BTC-USD price update
    {
      pyth-storage-contract: 'SP3R4F6C1J3JQWWCVZ3S7FRRYPMYG6ZW6RZK31FXY.pyth-storage-v3,
      pyth-decoder-contract: 'SP3R4F6C1J3JQWWCVZ3S7FRRYPMYG6ZW6RZK31FXY.pyth-pnau-decoder-v2,
      wormhole-core-contract: 'SP3R4F6C1J3JQWWCVZ3S7FRRYPMYG6ZW6RZK31FXY.wormhole-core-v3
    })
```

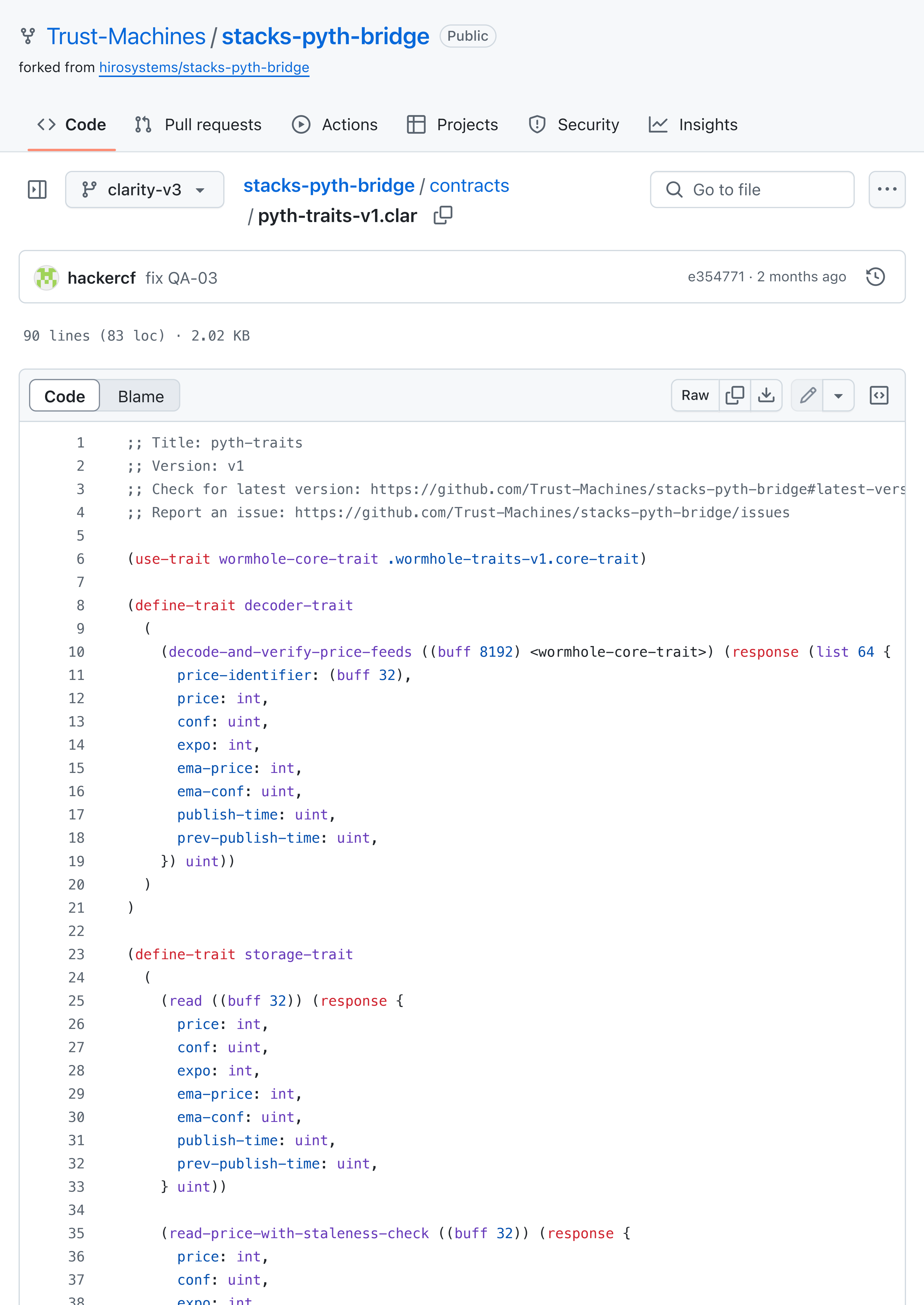
If the VAA is valid, the contract call will return a payload with the following signature:

```
(response
  (list 64 {
    price-identifier: (buff 32),
    price: int,
    conf: uint,
    expo: int,
    ema-price: int,
    ema-conf: uint,
    publish-time: uint,
    prev-publish-time: uint,
  })
  uint)
```

Including all the prices successfully updating the oracle. All of the implementation details can be found in Pyth documentation.

Upgrades

- Ported the codebase to Clarity V3
- bump up nodejs dependencies
- introduced a utility function `set-price-testnet` in the `pyth-storage-v3` contract to set the price data for a specific feed without the



Trust-Machines / stacks-pyth-bridge

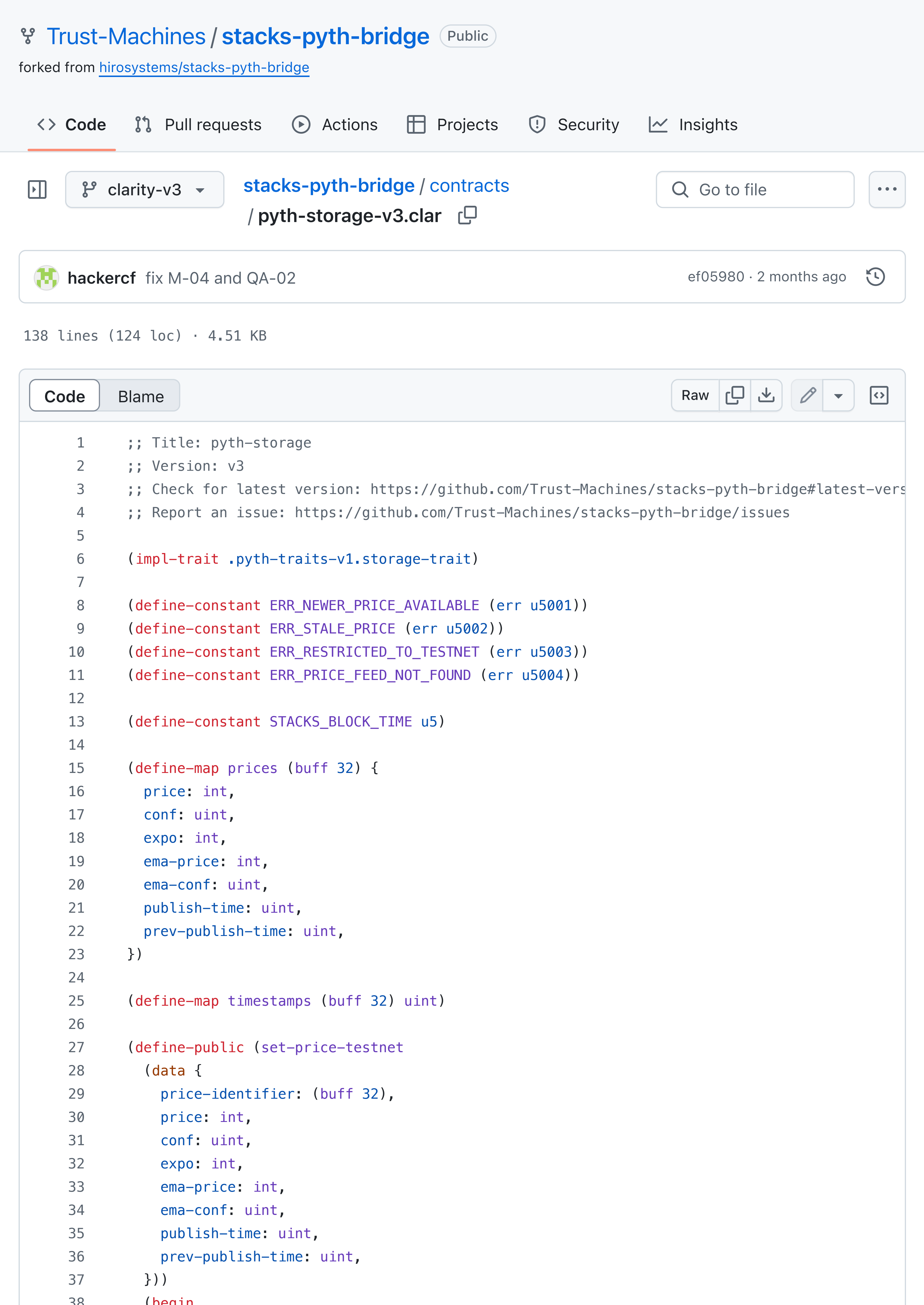
forked from hirosystems/stacks-pyth-bridge

stacks-pyth-bridge / contracts / pyth-traits-v1.clar

hackercf fix QA-03 e354771 · 2 months ago

90 lines (83 loc) · 2.02 KB

```
;; Title: pyth-traits
;; Version: v1
;; Check for latest version: https://github.com/Trust-Machines/stacks-pyth-bridge#latest-vers
;; Report an issue: https://github.com/Trust-Machines/stacks-pyth-bridge/issues

(use-trait wormhole-core-trait .wormhole-traits-v1.core-trait)

(define-trait decoder-trait
  (
    (decode-and-verify-price-feeds ((buff 8192) <wormhole-core-trait>) (response (list 64 {
      price-identifier: (buff 32),
      price: int,
      conf: uint,
      expo: int,
      ema-price: int,
      ema-conf: uint,
      publish-time: uint,
      prev-publish-time: uint,
    }) uint))
  )
)

(define-trait storage-trait
  (
    (read ((buff 32)) (response {
      price: int,
      conf: uint,
      expo: int,
      ema-price: int,
      ema-conf: uint,
      publish-time: uint,
      prev-publish-time: uint,
    } uint))

    (read-price-with-staleness-check ((buff 32)) (response {
      price: int,
      conf: uint,
      expo: int
      expo: int,
      ema-price: int,
      ema-conf: uint,
      publish-time: uint,
      prev-publish-time: uint,
    } uint))

    (write ((list 64 {
      price-identifier: (buff 32),
      price: int,
      conf: uint,
      expo: int,
      ema-price: int,
      ema-conf: uint,
      publish-time: uint,
      prev-publish-time: uint,
    })) (response (list 64 {
      price-identifier: (buff 32),
      price: int,
      conf: uint,
      expo: int,
      ema-price: int,
      ema-conf: uint,
      publish-time: uint,
      prev-publish-time: uint,
    }) uint))
  )
)

(define-trait proxy-trait
  (
    (read-price-feed ((buff 32)) (response {
      price: int,
      conf: uint,
      expo: int,
      ema-price: int,
      ema-conf: uint,
      publish-time: uint,
      prev-publish-time: uint,
    } uint))

    (verify-and-update-price-feeds ((buff 8192) <wormhole-core-trait>) (response (list 64 {
      price-identifier: (buff 32),
      price: int,
      conf: uint,
      expo: int,
      ema-price: int,
      ema-conf: uint,
      publish-time: uint,
      prev-publish-time: uint,
    }) uint))
  )
)
```



Trust-Machines / stacks-pyth-bridge Public
forked from hirosystems/stacks-pyth-bridge
Code
Pull requests
Actions
Projects
Security
Insights
clarity-v3
stacks-pyth-bridge / contracts / pyth-storage-v3.clar
Go to file
hackercf fix M-04 and QA-02
ef05980 · 2 months ago
138 lines (124 loc) · 4.51 KB
Code
Blame
Raw

```
;; Title: pyth-storage
;; Version: v3
;; Check for latest version: https://github.com/Trust-Machines/stacks-pyth-bridge#latest-vers
;; Report an issue: https://github.com/Trust-Machines/stacks-pyth-bridge/issues

(impl-trait .pyth-traits-v1.storage-trait)

(define-constant ERR_NEWER_PRICE_AVAILABLE (err u5001))
(define-constant ERR_STALE_PRICE (err u5002))
(define-constant ERR_RESTRICTED_TO_TESTNET (err u5003))
(define-constant ERR_PRICE_FEED_NOT_FOUND (err u5004))

(define-constant STACKS_BLOCK_TIME u5)

(define-map prices (buff 32) {
  price: int,
  conf: uint,
  expo: int,
  ema-price: int,
  ema-conf: uint,
  publish-time: uint,
  prev-publish-time: uint,
})

(define-map timestamps (buff 32) uint)

(define-public (set-price-testnet
  (data {
    price-identifier: (buff 32),
    price: int,
    conf: uint,
    expo: int,
    ema-price: int,
    ema-conf: uint,
    publish-time: uint,
    prev-publish-time: uint,
  }))
  (begin
  (begin
    (asserts! (not is-in-mainnet) ERR_RESTRICTED_TO_TESTNET)
    (ok (write-batch-entry data))
  )
)

(define-public (read (price-identifier (buff 32)))
  (let ((entry (unwrap! (map-get? prices price-identifier) ERR_PRICE_FEED_NOT_FOUND)))
    (ok entry)))

(define-read-only (get-price (price-identifier (buff 32)))
  (let ((entry (unwrap! (map-get? prices price-identifier) ERR_PRICE_FEED_NOT_FOUND)))
    (ok entry)))

(define-read-only (read-price-with-staleness-check (price-identifier (buff 32)))
  (let (
      (entry (unwrap! (map-get? prices price-identifier) ERR_PRICE_FEED_NOT_FOUND))
      (stale-price-threshold (contract-call? .pyth-governance-v2 get-stale-price-threshold))
      (latest-stacks-timestamp (unwrap! (get-stacks-block-info? time (- stacks-block-height u
    )
    (asserts! (>= (get publish-time entry) (+ (- latest-stacks-timestamp stale-price-threshol
    (ok entry)))

(define-public (write (batch-updates (list 64 {
    price-identifier: (buff 32),
    price: int,
    conf: uint,
    expo: int,
    ema-price: int,
    ema-conf: uint,
    publish-time: uint,
    prev-publish-time: uint,
  })))
  (let ((successful-updates (map unwrapped-entry (filter only-ok-entry (map write-batch-entry
    ;; Ensure that updates are always coming from the right contract
    (try! (contract-call? .pyth-governance-v2 check-execution-flow contract-caller none))
    (ok successful-updates)))

(define-private (write-batch-entry (entry {
      price-identifier: (buff 32),
      price: int,
      conf: uint,
      expo: int,
      ema-price: int,
      ema-conf: uint,
      publish-time: uint,
      prev-publish-time: uint,
    }))
    (let ((stale-price-threshold (contract-call? .pyth-governance-v2 get-stale-price-threshol
          (latest-stacks-timestamp (unwrap! (get-stacks-block-info? time (- stacks-block-heig
          (publish-time (get publish-time entry)))
      ;; Ensure that we have not processed a newer price
      (asserts! (is-price-update-more-recent (get price-identifier entry) publish-time) ERR_N
      ;; Ensure that price is not stale
      (asserts! (>= publish-time (+ (- latest-stacks-timestamp stale-price-threshold) STACKS_
      ;; Update storage
      (map-set prices
        (get price-identifier entry)
        {
          price: (get price entry),
          conf: (get conf entry),
          expo: (get expo entry),
          ema-price: (get ema-price entry),
          ema-conf: (get ema-conf entry),
          publish-time: publish-time,
          prev-publish-time: (get prev-publish-time entry)
        })
      ;; Emit event
      (print {
        type: "price-feed",
        action: "updated",
        data: entry
      })
      ;; Update timestamps tracking
      (map-set timestamps (get price-identifier entry) (get publish-time entry))
      (ok entry)))

(define-private (only-ok-entry (entry (response {
    price-identifier: (buff 32),
    price: int,
    conf: uint,
    expo: int,
    ema-price: int,
    ema-conf: uint,
    publish-time: uint,
    prev-publish-time: uint,
  } uint))) (is-ok entry))

(define-private (unwrapped-entry (entry (response {
    price-identifier: (buff 32),
    price: int,
    conf: uint,
    expo: int,
    ema-price: int,
    ema-conf: uint,
    publish-time: uint,
    prev-publish-time: uint,
  } uint))) (unwrap-panic entry))

(define-private (is-price-update-more-recent (price-identifier (buff 32)) (publish-time uint)
  (> publish-time (default-to u0 (map-get? timestamps price-identifier))))
```

Trust-Machines / **stacks-pyth-bridge** Public

forked from hirosystems/stacks-pyth-bridge

Code · Pull requests · Actions · Projects · Security · Insights

clarity-v3

stacks-pyth-bridge / contracts / **pyth-pnau-decoder-v2.clar**

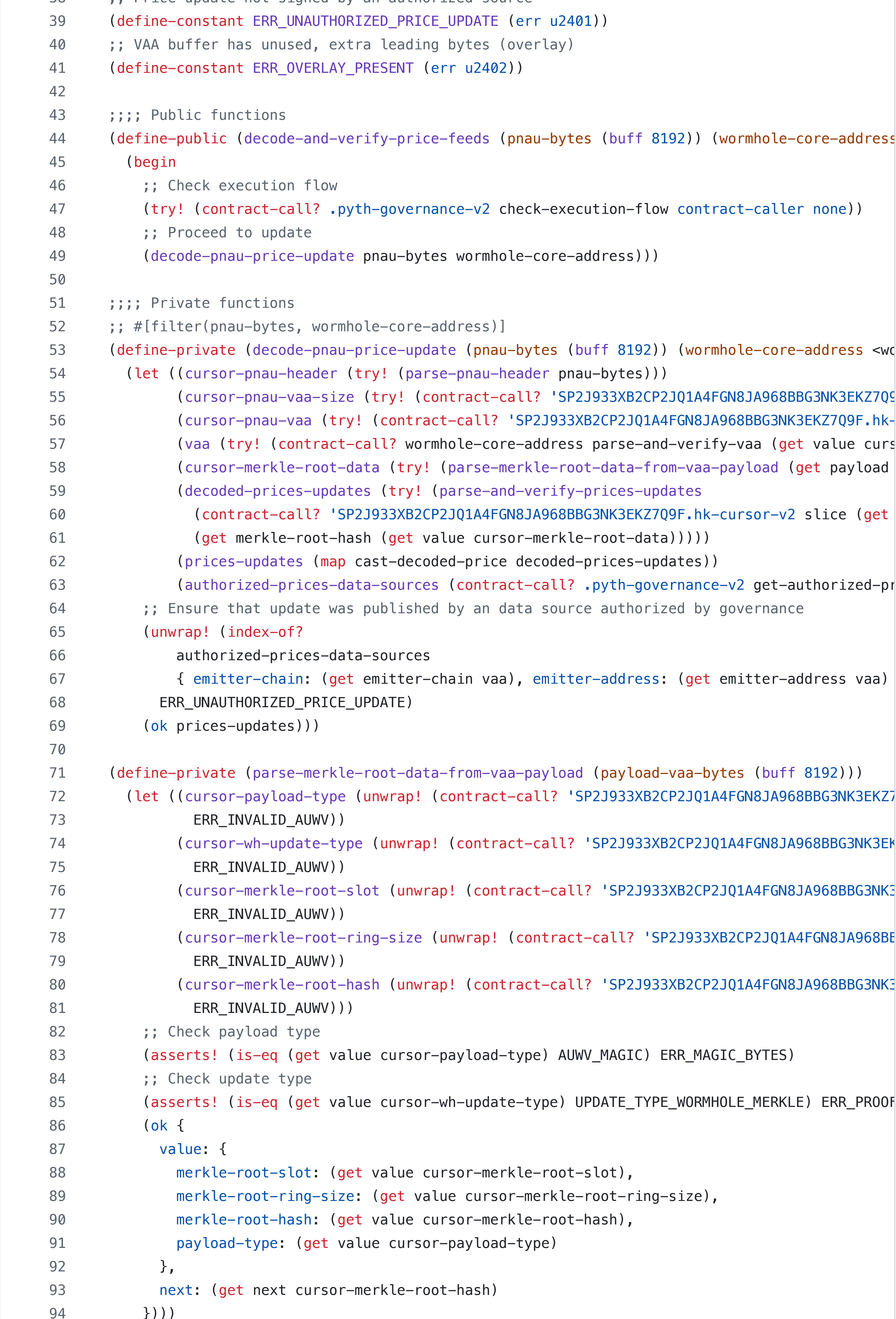
Go to file

hackercf fix R-QA-01 32320e0 · 2 months ago

327 lines (314 loc) · 15.9 KB

Code Blame Raw

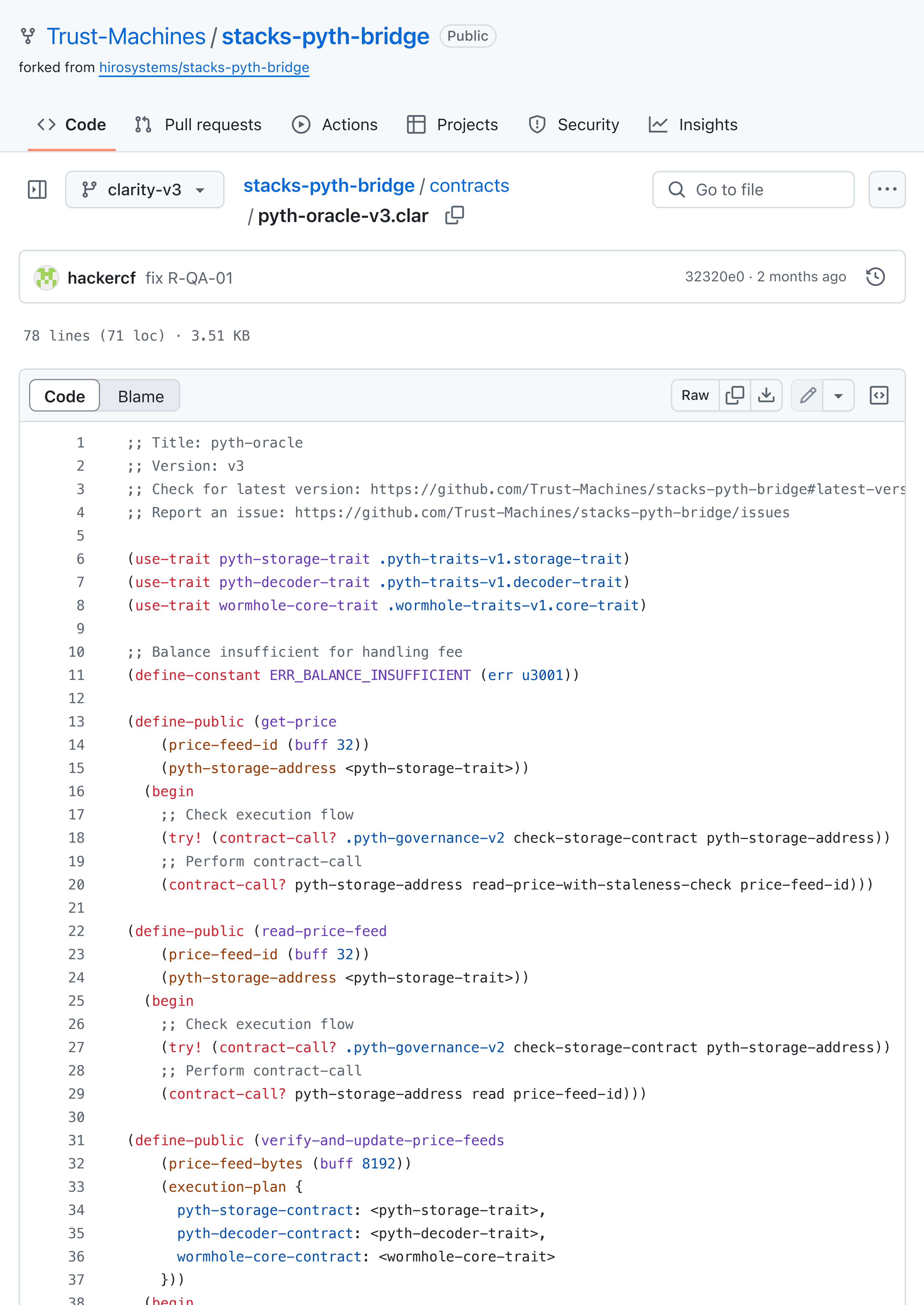
```
;; Title: pyth-pnau-decoder
;; Version: v2
;; Check for latest version: https://github.com/Trust-Machines/stacks-pyth-bridge#latest-vers
;; Report an issue: https://github.com/Trust-Machines/stacks-pyth-bridge/issues

;;;; Traits
(impl-trait .pyth-traits-v1.decoder-trait)
(use-trait wormhole-core-trait .wormhole-traits-v1.core-trait)

;;;; Constants

(define-constant PNAU_MAGIC 0x504e4155) ;; 'PNAU': Pyth Network Accumulator Update
(define-constant AUWV_MAGIC 0x41555756) ;; 'AUWV': Accumulator Update Wormhole Verification
(define-constant PYTHNET_MAJOR_VERSION u1)
(define-constant PYTHNET_MINOR_VERSION u0)
(define-constant UPDATE_TYPE_WORMHOLE_MERKLE u0)
(define-constant MESSAGE_TYPE_PRICE_FEED u0)
(define-constant MERKLE_PROOF_HASH_SIZE u20)

;; Unable to price feed magic bytes
(define-constant ERR_MAGIC_BYTES (err u2001))
;; Unable to parse major version
(define-constant ERR_VERSION_MAJ (err u2002))
;; Unable to parse minor version
(define-constant ERR_VERSION_MIN (err u2003))
;; Unable to parse trailing header size
(define-constant ERR_HEADER_TRAILING_SIZE (err u2004))
;; Unable to parse proof type
(define-constant ERR_PROOF_TYPE (err u2005))
;; Unable to parse update type
(define-constant ERR_UPDATE_TYPE (err u2006))
;; Merkle root mismatch
(define-constant ERR_INVALID_AUWV (err u2007))
;; Merkle root mismatch
(define-constant ERR_MERKLE_ROOT_MISMATCH (err u2008))
;; Incorrect AUWV payload
(define-constant ERR_INCORRECT_AUWV_PAYLOAD (err u2009))
;; Price update not signed by an authorized source
```



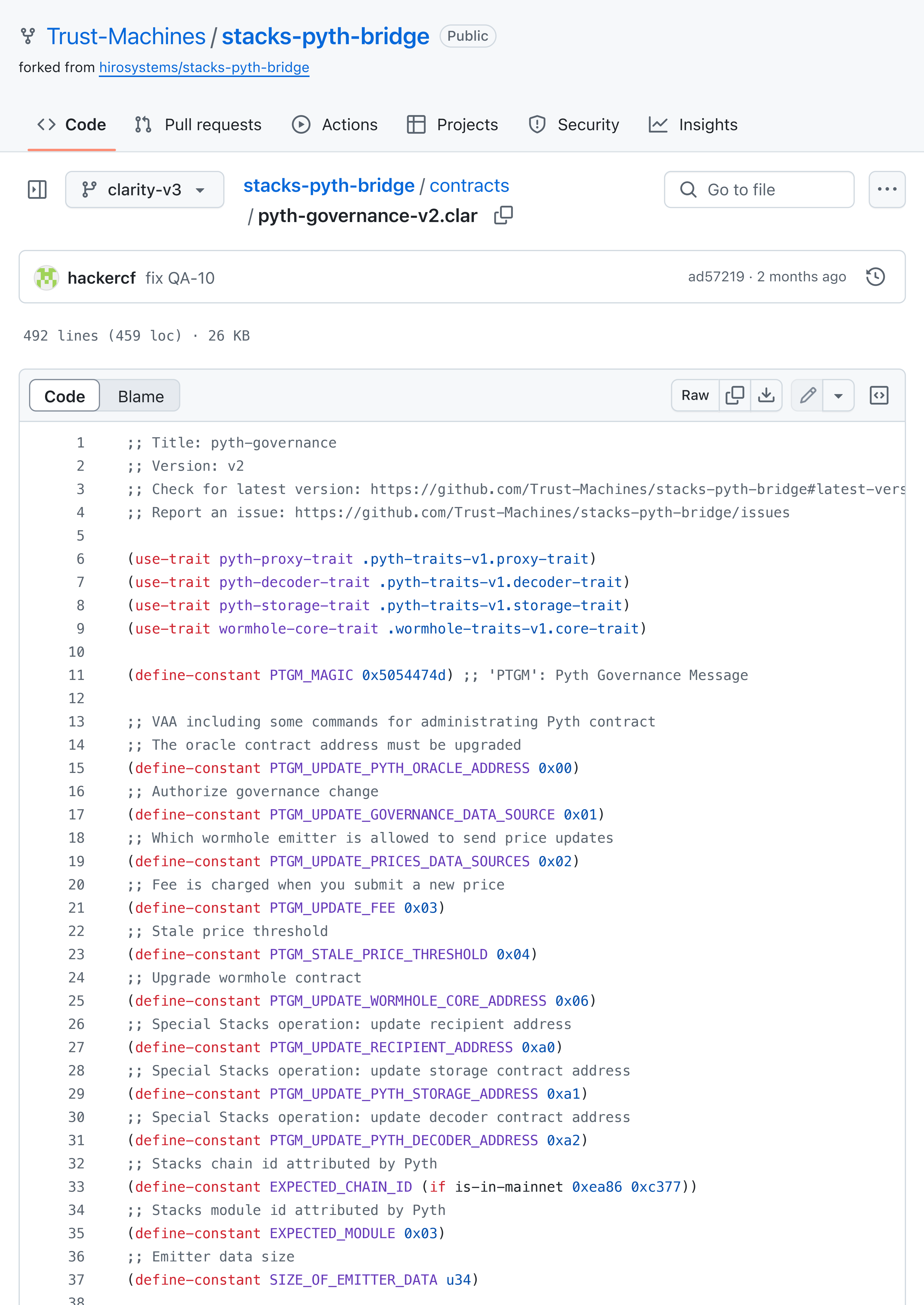
```
;; Price update not signed by an authorized source
(define-constant ERR_UNAUTHORIZED_PRICE_UPDATE (err u2401))
;; VAA buffer has unused, extra leading bytes (overlay)
(define-constant ERR_OVERLAY_PRESENT (err u2402))

;;;; Public functions
(define-public (decode-and-verify-price-feeds (pnau-bytes (buff 8192)) (wormhole-core-address
  (begin
    ;; Check execution flow
    (try! (contract-call? .pyth-governance-v2 check-execution-flow contract-caller none))
    ;; Proceed to update
    (decode-pnau-price-update pnau-bytes wormhole-core-address)))

;;;; Private functions
;; #[filter(pnau-bytes, wormhole-core-address)]
(define-private (decode-pnau-price-update (pnau-bytes (buff 8192)) (wormhole-core-address <wo
  (let ((cursor-pnau-header (try! (parse-pnau-header pnau-bytes)))
        (cursor-pnau-vaa-size (try! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9
        (cursor-pnau-vaa (try! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F.hk-
        (vaa (try! (contract-call? wormhole-core-address parse-and-verify-vaa (get value curs
        (cursor-merkle-root-data (try! (parse-merkle-root-data-from-vaa-payload (get payload
        (decoded-prices-updates (try! (parse-and-verify-prices-updates
          (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F.hk-cursor-v2 slice (get
          (get merkle-root-hash (get value cursor-merkle-root-data)))))
        (prices-updates (map cast-decoded-price decoded-prices-updates))
        (authorized-prices-data-sources (contract-call? .pyth-governance-v2 get-authorized-pr
    ;; Ensure that update was published by an data source authorized by governance
    (unwrap! (index-of?
        authorized-prices-data-sources
        { emitter-chain: (get emitter-chain vaa), emitter-address: (get emitter-address vaa)
      ERR_UNAUTHORIZED_PRICE_UPDATE)
    (ok prices-updates)))

(define-private (parse-merkle-root-data-from-vaa-payload (payload-vaa-bytes (buff 8192)))
  (let ((cursor-payload-type (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7
          ERR_INVALID_AUWV))
        (cursor-wh-update-type (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EK
          ERR_INVALID_AUWV))
        (cursor-merkle-root-slot (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3
          ERR_INVALID_AUWV))
        (cursor-merkle-root-ring-size (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BB
          ERR_INVALID_AUWV))
        (cursor-merkle-root-hash (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3
          ERR_INVALID_AUWV)))
    ;; Check payload type
    (asserts! (is-eq (get value cursor-payload-type) AUWV_MAGIC) ERR_MAGIC_BYTES)
    ;; Check update type
    (asserts! (is-eq (get value cursor-wh-update-type) UPDATE_TYPE_WORMHOLE_MERKLE) ERR_PROOF
    (ok {
      value: {
        merkle-root-slot: (get value cursor-merkle-root-slot),
        merkle-root-ring-size: (get value cursor-merkle-root-ring-size),
        merkle-root-hash: (get value cursor-merkle-root-hash),
        payload-type: (get value cursor-payload-type)
      },
      next: (get next cursor-merkle-root-hash)
    })))
```

```

(define-private (parse-pnau-header (pf-bytes (buff 8192)))
  (let ((cursor-magic (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F.hk-
          ERR_MAGIC_BYTES))
        (cursor-version-maj (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q
          ERR_VERSION_MAJ))
        (cursor-version-min (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q
          ERR_VERSION_MIN))
        (cursor-header-trailing-size (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG
          ERR_HEADER_TRAILING_SIZE))
        (cursor-proof-type (unwrap! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9
            bytes: pf-bytes,
            pos: (+ (get pos (get next cursor-header-trailing-size)) (get value cursor-header
          ERR_PROOF_TYPE)))
```

```
            }) u64)),
            limit: (get limit acc),
        })
        ;; Increment position
        {
            cursor: {
              index: (+ (get index (get cursor acc)) u1),
              next-update-index: (get next-update-index (get cursor acc)),
            },
            bytes: (get bytes acc),
            result: (get result acc),
            limit: (get limit acc),
        })))

  (define-private (parse-proof
        (entry (buff 1))
        (acc {
          cursor: {
            index: uint,
            next-update-index: uint
          },
          bytes: (buff 8192),
          result: (list 128 (buff 20)),
          limit: uint
        }))
    (if (is-eq (len (get result acc)) (get limit acc))
      acc
      (if (is-eq (get index (get cursor acc)) (get next-update-index (get cursor acc)))
        ;; Parse update
        (let ((cursor-hash (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F.hk-cursor
              (hash (get value (unwrap-panic (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK
          {
            cursor: {
              index: (+ (get index (get cursor acc)) u1),
              next-update-index: (+ (get index (get cursor acc)) MERKLE_PROOF_HASH_SIZE),
            },
            bytes: (get bytes acc),
            result: (unwrap-panic (as-max-len? (append (get result acc) hash) u128)),
            limit: (get limit acc),
          })
        ;; Increment position
        {
            cursor: {
              index: (+ (get index (get cursor acc)) u1),
              next-update-index: (get next-update-index (get cursor acc)),
            },
            bytes: (get bytes acc),
            result: (get result acc),
            limit: (get limit acc)
        })))

  (define-private (cast-decoded-price (entry
          {
            price-identifier: (buff 32),
            price: int,
            conf: uint,
            expo: int,
            publish-time: uint,
            prev-publish-time: uint,
            ema-price: int,
            ema-conf: uint,
            proof: (list 128 (buff 20)),
            leaf-bytes: (buff 255)
          }))
    {
      price-identifier: (get price-identifier entry),
      price: (get price entry),
      conf: (get conf entry),
      expo: (get expo entry),
      publish-time: (get publish-time entry),
      prev-publish-time: (get prev-publish-time entry),
      ema-price: (get ema-price entry),
      ema-conf: (get ema-conf entry)
    })
```


Trust-Machines / stacks-pyth-bridge Public
forked from hirosystems/stacks-pyth-bridge
Code
Pull requests
Actions
Projects
Security
Insights
clarity-v3
stacks-pyth-bridge / contracts / pyth-oracle-v3.clar
Go to file
hackercf fix R-QA-01
32320e0 · 2 months ago
78 lines (71 loc) · 3.51 KB
Code
Blame
Raw

```
;; Title: pyth-oracle
;; Version: v3
;; Check for latest version: https://github.com/Trust-Machines/stacks-pyth-bridge#latest-vers
;; Report an issue: https://github.com/Trust-Machines/stacks-pyth-bridge/issues

(use-trait pyth-storage-trait .pyth-traits-v1.storage-trait)
(use-trait pyth-decoder-trait .pyth-traits-v1.decoder-trait)
(use-trait wormhole-core-trait .wormhole-traits-v1.core-trait)

;; Balance insufficient for handling fee
(define-constant ERR_BALANCE_INSUFFICIENT (err u3001))

(define-public (get-price
    (price-feed-id (buff 32))
    (pyth-storage-address <pyth-storage-trait>))
  (begin
    ;; Check execution flow
    (try! (contract-call? .pyth-governance-v2 check-storage-contract pyth-storage-address))
    ;; Perform contract-call
    (contract-call? pyth-storage-address read-price-with-staleness-check price-feed-id)))

(define-public (read-price-feed
    (price-feed-id (buff 32))
    (pyth-storage-address <pyth-storage-trait>))
  (begin
    ;; Check execution flow
    (try! (contract-call? .pyth-governance-v2 check-storage-contract pyth-storage-address))
    ;; Perform contract-call
    (contract-call? pyth-storage-address read price-feed-id)))

(define-public (verify-and-update-price-feeds
    (price-feed-bytes (buff 8192))
    (execution-plan {
      pyth-storage-contract: <pyth-storage-trait>,
      pyth-decoder-contract: <pyth-decoder-trait>,
      wormhole-core-contract: <wormhole-core-trait>
    }))
  (begin
  (begin
    ;; Check execution flow
    (try! (contract-call? .pyth-governance-v2 check-execution-flow contract-caller (some exec
    ;; Perform contract-call
    (let ((pyth-decoder-contract (get pyth-decoder-contract execution-plan))
          (wormhole-core-contract (get wormhole-core-contract execution-plan))
          (pyth-storage-contract (get pyth-storage-contract execution-plan))
          (decoded-prices (try! (contract-call? pyth-decoder-contract decode-and-verify-price
          (updated-prices (try! (contract-call? pyth-storage-contract write decoded-prices)))
          (fee-info (contract-call? .pyth-governance-v2 get-fee-info))
          (fee-amount (* (len updated-prices) (* (get mantissa fee-info) (pow u10 (get expone
      ;; Charge fee
      (if (> fee-amount u0)
        (unwrap! (stx-transfer? fee-amount tx-sender (get address fee-info)) ERR_BALANCE_INSU
        true
      )

      (ok updated-prices))))

(define-public (decode-price-feeds
    (price-feed-bytes (buff 8192))
    (execution-plan {
      pyth-storage-contract: <pyth-storage-trait>,
      pyth-decoder-contract: <pyth-decoder-trait>,
      wormhole-core-contract: <wormhole-core-trait>
    }))
  (begin
    ;; Check execution flow
    (try! (contract-call? .pyth-governance-v2 check-execution-flow contract-caller (some exec
    ;; Perform contract-call
    (let ((pyth-decoder-contract (get pyth-decoder-contract execution-plan))
          (wormhole-core-contract (get wormhole-core-contract execution-plan))
          (decoded-prices (try! (contract-call? pyth-decoder-contract decode-and-verify-price
          (fee-info (contract-call? .pyth-governance-v2 get-fee-info))
          (fee-amount (* (len decoded-prices) (* (get mantissa fee-info) (pow u10 (get expone
      ;; Charge fee
      (if (> fee-amount u0)
        (unwrap! (stx-transfer? fee-amount tx-sender (get address fee-info)) ERR_BALANCE_INSU
        true
      )
      (ok decoded-prices))))
```



Trust-Machines / stacks-pyth-bridge Public

forked from hirosystems/stacks-pyth-bridge

stacks-pyth-bridge / contracts / pyth-governance-v2.clar

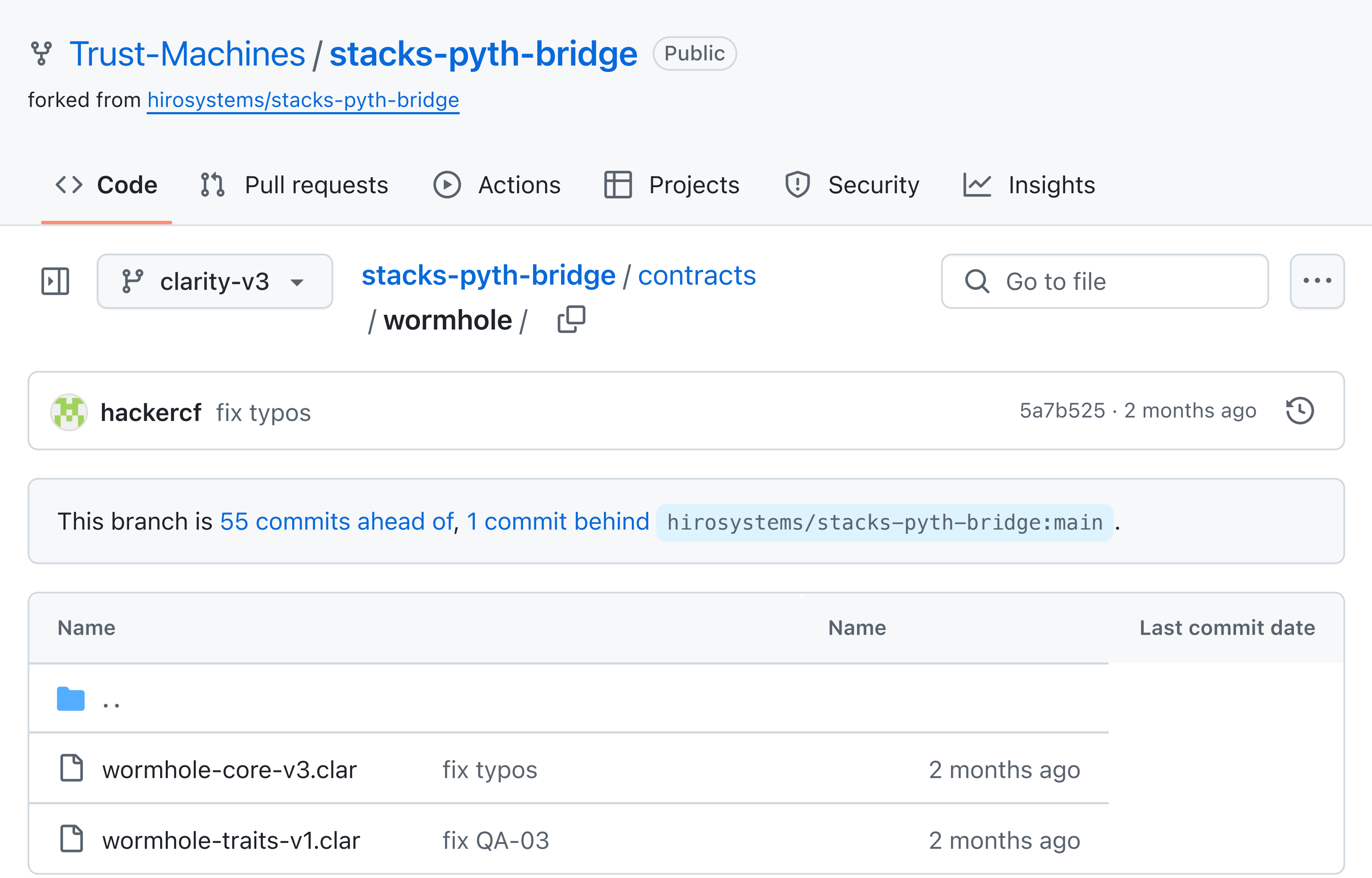
hackercf fix QA-10 ad57219 · 2 months ago

492 lines (459 loc) · 26 KB

```
;; Title: pyth-governance
;; Version: v2
;; Check for latest version: https://github.com/Trust-Machines/stacks-pyth-bridge#latest-vers
;; Report an issue: https://github.com/Trust-Machines/stacks-pyth-bridge/issues

(use-trait pyth-proxy-trait .pyth-traits-v1.proxy-trait)
(use-trait pyth-decoder-trait .pyth-traits-v1.decoder-trait)
(use-trait pyth-storage-trait .pyth-traits-v1.storage-trait)
(use-trait wormhole-core-trait .wormhole-traits-v1.core-trait)

(define-constant PTGM_MAGIC 0x5054474d) ;; 'PTGM': Pyth Governance Message

;; VAA including some commands for administrating Pyth contract
;; The oracle contract address must be upgraded
(define-constant PTGM_UPDATE_PYTH_ORACLE_ADDRESS 0x00)
;; Authorize governance change
(define-constant PTGM_UPDATE_GOVERNANCE_DATA_SOURCE 0x01)
;; Which wormhole emitter is allowed to send price updates
(define-constant PTGM_UPDATE_PRICES_DATA_SOURCES 0x02)
;; Fee is charged when you submit a new price
(define-constant PTGM_UPDATE_FEE 0x03)
;; Stale price threshold
(define-constant PTGM_STALE_PRICE_THRESHOLD 0x04)
;; Upgrade wormhole contract
(define-constant PTGM_UPDATE_WORMHOLE_CORE_ADDRESS 0x06)
;; Special Stacks operation: update recipient address
(define-constant PTGM_UPDATE_RECIPIENT_ADDRESS 0xa0)
;; Special Stacks operation: update storage contract address
(define-constant PTGM_UPDATE_PYTH_STORAGE_ADDRESS 0xa1)
;; Special Stacks operation: update decoder contract address
(define-constant PTGM_UPDATE_PYTH_DECODER_ADDRESS 0xa2)
;; Stacks chain id attributed by Pyth
(define-constant EXPECTED_CHAIN_ID (if is-in-mainnet 0xea86 0xc377))
;; Stacks module id attributed by Pyth
(define-constant EXPECTED_MODULE 0x03)
;; Emitter data size
(define-constant SIZE_OF_EMITTER_DATA u34)

```

```
;; Error unexpected action
(define-constant ERR_UNEXPECTED_ACTION (err u4001))
;; Error unexpected action
(define-constant ERR_INVALID_ACTION_PAYLOAD (err u4002))
;; Error unauthorized control flow
(define-constant ERR_UNAUTHORIZED_ACCESS (err u4003))
;; Error outdated action
(define-constant ERR_OUTDATED (err u4004))
;; Error unauthorized update
(define-constant ERR_UNAUTHORIZED_UPDATE (err u4005))
;; Error parsing PTGM
(define-constant ERR_INVALID_PTGM (err u4006))
;; Error not standard principal
(define-constant ERR_NOT_STANDARD_PRINCIPAL (err u4007))
;; Error Ptgm overlay bytes
(define-constant ERR_PTGM_CHECK_OVERLAY (err u4008))
;; Error invalid price data source
(define-constant ERR_INVALID_PRICE_DATA_SOURCES (err u4009))

(define-data-var governance-data-source
  { emitter-chain: uint, emitter-address: (buff 32) }
  { emitter-chain: u1, emitter-address: 0x5635979a221c34931e32620b9293a463065555ea71fe97cd623
(define-data-var prices-data-sources
  (list 255 { emitter-chain: uint, emitter-address: (buff 32) })
  (list
    { emitter-chain: u1, emitter-address: 0x6bb14509a612f01fbbc4cffeebd4bbfb492a86df717ebe92e
    { emitter-chain: u26, emitter-address: 0xf8cd23c2ab91237730770bbea08d61005cdda0984348f3f6
    { emitter-chain: u26, emitter-address: 0xe101faedac5851e32b9b23b5f9411a8c2bac4aae3ed4dd7b
(define-data-var fee-value
  { mantissa: uint, exponent: uint }
  { mantissa: u1, exponent: u0 })
(define-data-var stale-price-threshold uint (if is-in-mainnet (* u2 u60 u60) (* u5 u365 u24 u
(define-data-var fee-recipient-address principal (if is-in-mainnet 'SP3CRXBDXQ2N5P7E25Q39MEX1
(define-data-var last-sequence-processed uint u0)

;; Execution plan management
(define-data-var current-execution-plan {
  pyth-oracle-contract: principal,
  pyth-decoder-contract: principal,
  pyth-storage-contract: principal,
  wormhole-core-contract: principal
} {
    pyth-oracle-contract: .pyth-oracle-v3,
    pyth-decoder-contract: .pyth-pnau-decoder-v2,
    pyth-storage-contract: .pyth-storage-v3,
    wormhole-core-contract: .wormhole-core-v3
})

(define-read-only (check-execution-flow
  (former-contract-caller principal)
  (execution-plan-opt (optional {
    pyth-storage-contract: <pyth-storage-trait>,
    pyth-decoder-contract: <pyth-decoder-trait>,
    wormhole-core-contract: <wormhole-core-trait>
  })))
  (let ((expected-execution-plan (var-get current-execution-plan))
        (success (if (is-eq contract-caller (get pyth-storage-contract expected-execution-pla
          ;; The storage contract is checking its execution flow
          ;; Must always be invoked by the proxy
          (try! (expect-contract-call-performed-by-expected-oracle-contract former-contract-c
          ;; Other contract
          (if (is-eq contract-caller (get pyth-decoder-contract expected-execution-plan))
            ;; The decoding contract is checking its execution flow
            (try! (expect-contract-call-performed-by-expected-oracle-contract former-contract
            (if (is-eq contract-caller (get pyth-oracle-contract expected-execution-plan))
              ;; The proxy contract is checking its execution flow
              (let ((execution-plan (unwrap! execution-plan-opt ERR_UNAUTHORIZED_ACCESS)))
                ;; Ensure that storage contract is the one expected
                (try! (expect-active-storage-contract (get pyth-storage-contract execution-pl
                ;; Ensure that decoder contract is the one expected
```

```
  (asserts! (is-eq (get pos (get next cursor-emitter-address)) (len ptgm-body)) ERR_PTGM_CH
  (ok {
   emitter-chain: (get value cursor-emitter-chain),
   emitter-sequence: (get value cursor-emitter-sequence),
   emitter-address: (get value cursor-emitter-address)
  })))

(define-private (parse-principal (ptgm-body (buff 8192)))
 (let ((cursor-ptgm-body (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F.hk-curso
    (cursor-principal-len (try! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9
    (principal-bytes (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F.hk-cursor
    (new-principal (unwrap! (from-consensus-buff? principal principal-bytes) ERR_INVALID_
  (asserts! (is-eq (+ (get pos (get next cursor-principal-len)) (get value cursor-principal
  (asserts! (is-eq (+ (get pos (get next-cursor-principal-len)) (get value cursor-principa
  (asserts! (is-standard new-principal) ERR_NOT_STANDARD_PRINCIPAL)
  (ok new-principal)))

(define-private (parse-and-verify-prices-data-sources (ptgm-body (buff 8192)))
 (let ((cursor-ptgm-body (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F.hk-curso
    (cursor-num-data-sources (try! (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ
    (cursor-data-sources-bytes (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F
    (data-sources-bundle (fold parse-data-source cursor-data-sources-bytes {
     result: (list),
     cursor: {
      index: u0,
      next-update-index: u0
     },
     bytes: cursor-data-sources-bytes,
     limit: (get value cursor-num-data-sources)
    }))
    (data-sources (get result data-sources-bundle)))
  (asserts! (is-eq (get next-update-index (get cursor data-sources-bundle)) (len cursor-dat
  (asserts! (is-eq (get value cursor-num-data-sources) (len data-sources)) ERR_INVALID_PRIC
  (ok data-sources)))

(define-private (parse-data-source
   (entry (buff 1))
   (acc {
    cursor: {
     index: uint,
     next-update-index: uint
    },
    bytes: (buff 8192),
    result: (list 255 { emitter-chain: uint, emitter-address: (buff 32) }),
    limit: uint
   }))
 (if (is-eq (len (get result acc)) (get limit acc))
  acc
  (if (is-eq (get index (get cursor acc)) (get next-update-index (get cursor acc)))
   ;; Parse update
   (let ((buffer (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968BBG3NK3EKZ7Q9F.hk-cursor-v2 r
      (cursor-emitter-chain (unwrap-panic (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA968B
      (cursor-emitter-address (unwrap-panic (contract-call? 'SP2J933XB2CP2JQ1A4FGN8JA96
    {
     cursor: {
      index: (+ (get index (get cursor acc)) u1),
      next-update-index: (+ (get index (get cursor acc)) SIZE_OF_EMITTER_DATA),
     },
     bytes: (get bytes acc),
     result: (unwrap-panic (as-max-len? (append (get result acc) {
      emitter-chain: (get value cursor-emitter-chain),
      emitter-address: (get value cursor-emitter-address)
     }) u255)),
     limit: (get limit acc),
    })
   ;; Increment position
   {
     cursor: {
      index: (+ (get index (get cursor acc)) u1),
      next-update-index: (get next-update-index (get cursor acc)),
     },
     bytes: (get bytes acc),
     result: (get result acc),
     limit: (get limit acc)
   })))
```



Trust-Machines / stacks-pyth-bridge Public

forked from hirosystems/stacks-pyth-bridge

stacks-pyth-bridge / contracts / wormhole /

hackercf fix typos 5a7b525 · 2 months ago

This branch is 55 commits ahead of, 1 commit behind `hirosystems/stacks-pyth-bridge:main` .

Name	Name	Last commit date
..		
wormhole-core-v3.clar	fix typos	2 months ago
wormhole-traits-v1.clar	fix QA-03	2 months ago

Trust-Machines / **stacks-pyth-bridge** Public

forked from hirosystems/stacks-pyth-bridge

Code Pull requests Actions Projects Security Insights

clarity-v3

stacks-pyth-bridge / contracts / wormhole / **wormhole-traits-v1.clar**

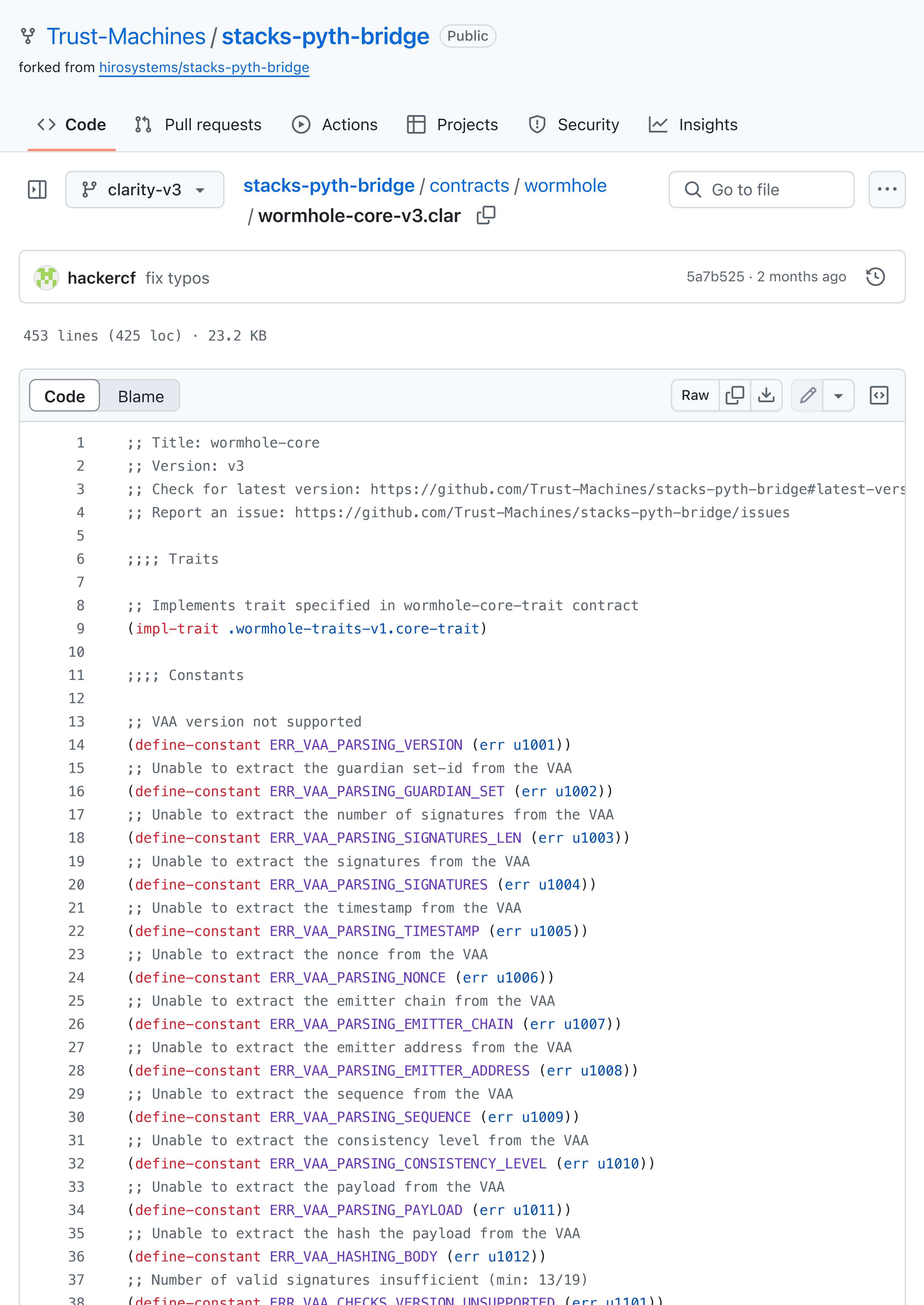
Go to file

hackercf fix QA-03 e354771 · 2 months ago

23 lines (22 loc) · 707 Bytes

Code Blame Raw

```
;; Title: core-traits
;; Version: v1
;; Check for latest version: https://github.com/Trust-Machines/stacks-pyth-bridge#latest-vers
;; Report an issue: https://github.com/Trust-Machines/stacks-pyth-bridge/issues

(define-trait core-trait
  (
    ;; Parse and Verify cryptographic validity of a VAA
    (parse-and-verify-vaa ((buff 8192)) (response {
      version: uint,
      guardian-set-id: uint,
      signatures-len: uint ,
      signatures: (list 19 { guardian-id: uint, signature: (buff 65) }),
      timestamp: uint,
      nonce: uint,
      emitter-chain: uint,
      emitter-address: (buff 32),
      sequence: uint,
      consistency-level: uint,
      payload: (buff 8192),
    } uint))
  )
)
```



Trust-Machines / stacks-pyth-bridge Public

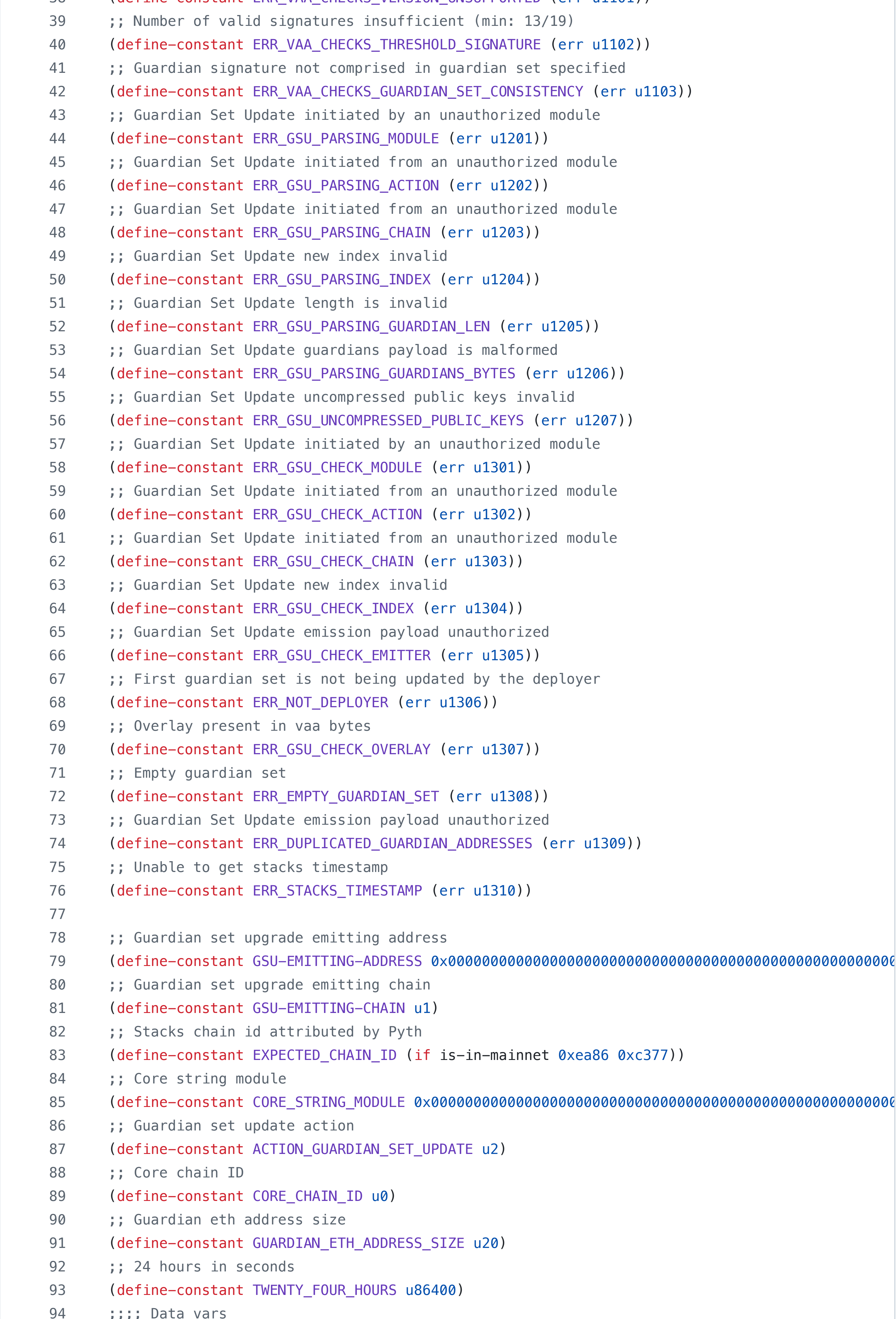
forked from hirosystems/stacks-pyth-bridge

stacks-pyth-bridge / contracts / wormhole / wormhole-core-v3.clar

hackercf fix typos 5a7b525 · 2 months ago

453 lines (425 loc) · 23.2 KB

```
;; Title: wormhole-core
;; Version: v3
;; Check for latest version: https://github.com/Trust-Machines/stacks-pyth-bridge#latest-vers
;; Report an issue: https://github.com/Trust-Machines/stacks-pyth-bridge/issues

;;;; Traits

;; Implements trait specified in wormhole-core-trait contract
(impl-trait .wormhole-traits-v1.core-trait)

;;;; Constants

;; VAA version not supported
(define-constant ERR_VAA_PARSING_VERSION (err u1001))
;; Unable to extract the guardian set-id from the VAA
(define-constant ERR_VAA_PARSING_GUARDIAN_SET (err u1002))
;; Unable to extract the number of signatures from the VAA
(define-constant ERR_VAA_PARSING_SIGNATURES_LEN (err u1003))
;; Unable to extract the signatures from the VAA
(define-constant ERR_VAA_PARSING_SIGNATURES (err u1004))
;; Unable to extract the timestamp from the VAA
(define-constant ERR_VAA_PARSING_TIMESTAMP (err u1005))
;; Unable to extract the nonce from the VAA
(define-constant ERR_VAA_PARSING_NONCE (err u1006))
;; Unable to extract the emitter chain from the VAA
(define-constant ERR_VAA_PARSING_EMITTER_CHAIN (err u1007))
;; Unable to extract the emitter address from the VAA
(define-constant ERR_VAA_PARSING_EMITTER_ADDRESS (err u1008))
;; Unable to extract the sequence from the VAA
(define-constant ERR_VAA_PARSING_SEQUENCE (err u1009))
;; Unable to extract the consistency level from the VAA
(define-constant ERR_VAA_PARSING_CONSISTENCY_LEVEL (err u1010))
;; Unable to extract the payload from the VAA
(define-constant ERR_VAA_PARSING_PAYLOAD (err u1011))
;; Unable to extract the hash the payload from the VAA
(define-constant ERR_VAA_HASHING_BODY (err u1012))
;; Number of valid signatures insufficient (min: 13/19)
(define-constant ERR_VAA_CHECKS_VERSION_UNSUPPORTED (err u1101))
```



```
(define-constant ERR_VAA_CHECKS_VERSION_UNSUPPORTED (err u1101))
;; Number of valid signatures insufficient (min: 13/19)
(define-constant ERR_VAA_CHECKS_THRESHOLD_SIGNATURE (err u1102))
;; Guardian signature not comprised in guardian set specified
(define-constant ERR_VAA_CHECKS_GUARDIAN_SET_CONSISTENCY (err u1103))
;; Guardian Set Update initiated by an unauthorized module
(define-constant ERR_GSU_PARSING_MODULE (err u1201))
;; Guardian Set Update initiated from an unauthorized module
(define-constant ERR_GSU_PARSING_ACTION (err u1202))
;; Guardian Set Update initiated from an unauthorized module
(define-constant ERR_GSU_PARSING_CHAIN (err u1203))
;; Guardian Set Update new index invalid
(define-constant ERR_GSU_PARSING_INDEX (err u1204))
;; Guardian Set Update length is invalid
(define-constant ERR_GSU_PARSING_GUARDIAN_LEN (err u1205))
;; Guardian Set Update guardians payload is malformed
(define-constant ERR_GSU_PARSING_GUARDIANS_BYTES (err u1206))
;; Guardian Set Update uncompressed public keys invalid
(define-constant ERR_GSU_UNCOMPRESSED_PUBLIC_KEYS (err u1207))
;; Guardian Set Update initiated by an unauthorized module
(define-constant ERR_GSU_CHECK_MODULE (err u1301))
;; Guardian Set Update initiated from an unauthorized module
(define-constant ERR_GSU_CHECK_ACTION (err u1302))
;; Guardian Set Update initiated from an unauthorized module
(define-constant ERR_GSU_CHECK_CHAIN (err u1303))
;; Guardian Set Update new index invalid
(define-constant ERR_GSU_CHECK_INDEX (err u1304))
;; Guardian Set Update emission payload unauthorized
(define-constant ERR_GSU_CHECK_EMITTER (err u1305))
;; First guardian set is not being updated by the deployer
(define-constant ERR_NOT_DEPLOYER (err u1306))
;; Overlay present in vaa bytes
(define-constant ERR_GSU_CHECK_OVERLAY (err u1307))
;; Empty guardian set
(define-constant ERR_EMPTY_GUARDIAN_SET (err u1308))
;; Guardian Set Update emission payload unauthorized
(define-constant ERR_DUPLICATED_GUARDIAN_ADDRESSES (err u1309))
;; Unable to get stacks timestamp
(define-constant ERR_STACKS_TIMESTAMP (err u1310))

;; Guardian set upgrade emitting address
(define-constant GSU-EMITTING-ADDRESS 0x00000000000000000000000000000000000000000000000000000
;; Guardian set upgrade emitting chain
(define-constant GSU-EMITTING-CHAIN u1)
;; Stacks chain id attributed by Pyth
(define-constant EXPECTED_CHAIN_ID (if is-in-mainnet 0xea86 0xc377))
;; Core string module
(define-constant CORE_STRING_MODULE 0x0000000000000000000000000000000000000000000000000000000
;; Guardian set update action
(define-constant ACTION_GUARDIAN_SET_UPDATE u2)
;; Core chain ID
(define-constant CORE_CHAIN_ID u0)
;; Guardian eth address size
(define-constant GUARDIAN_ETH_ADDRESS_SIZE u20)
;; 24 hours in seconds
(define-constant TWENTY_FOUR_HOURS u86400)
;;;; Data vars
```

```
;;;;

;; Guardian Set Update uncompressed public keys invalid
(define-data-var guardian-set-initialized bool false)
;; Contract deployer
(define-constant deployer contract-caller)
;; Keep track of the active guardian set-id
(define-data-var active-guardian-set-id uint u0)
;; Keep track of exiting guardian set
(define-data-var previous-guardian-set {set-id: uint, expires-at: uint} {set-id: u0, expires-

;;;; Data maps

;; Map tracking guardians set
(define-map guardian-sets uint (list 19 { compressed-public-key: (buff 33), uncompressed-publ
```

```
    ;; Ensure that this message is matching the expected chain
    (asserts! (or (is-eq (get value cursor-chain) (buff-to-uint-be EXPECTED_CHAIN_ID)) (is-ec
    (if (var-get guardian-set-initialized)
      ;; Ensure that next index = current index + 1
      (asserts! (is-eq (get value cursor-new-index) (+ u1 (var-get active-guardian-set-id)))
      ;; Ensure that next index > current index
      (asserts! (> (get value cursor-new-index) (var-get active-guardian-set-id)) ERR_GSU_CHE
    )

    ;; Good to go!
    (ok {
        guardians-eth-addresses: eth-addresses,
        module: (get value cursor-module),
        action: (get value cursor-action),
        chain: (get value cursor-chain),
        new-index: (get value cursor-new-index)
      })))

(define-private (get-quorum (guardian-set-size uint))
  (+ (/ (* guardian-set-size u2) u3) u1))

(define-private (is-guardian-cue (byte (buff 1)) (acc { cursor: uint, result: (list 19 uint)
  (if (is-eq u0 (mod (get cursor acc) GUARDIAN_ETH_ADDRESS_SIZE))
    {
      cursor: (+ u1 (get cursor acc)),
      result: (unwrap-panic (as-max-len? (append (get result acc) (get cursor acc)) u19)),
    }
    {
      cursor: (+ u1 (get cursor acc)),
      result: (get result acc),
    }))

(define-private (is-valid-guardian-entry (entry { compressed-public-key: (buff 33), uncompres
  (begin
    (try! prev-res)
    (let (
      (compressed (get compressed-public-key entry))
      (uncompressed (get uncompressed-public-key entry)))
      (if (or (is-eq 0x compressed) (is-eq 0x uncompressed))
        ERR_GSU_PARSING_GUARDIAN_LEN
        (ok true)
      )
    )
  )
)

(define-private (set-new-guardian-set-id (new-set-id uint))
  (if (var-get guardian-set-initialized)
    (let (
        (latest-stacks-timestamp (unwrap! (get-stacks-block-info? time (- stacks-block-height
        (previous-set-expires-at (+ TWENTY_FOUR_HOURS latest-stacks-timestamp))
      )
      (var-set previous-guardian-set {
          set-id: (var-get active-guardian-set-id),
          expires-at: previous-set-expires-at
        })
      (var-set active-guardian-set-id new-set-id)
      (ok true)
    )
    (begin (var-set active-guardian-set-id new-set-id) (ok true))
  )
)

(define-private (is-valid-guardian-set (set-id uint))
  (if (is-eq (var-get active-guardian-set-id) set-id)
    (ok true)
    (let (
      (prev-guardian-set (var-get previous-guardian-set))
      (prev-guardian-set-id (get set-id prev-guardian-set))
      (prev-guardian-set-expires-at (get expires-at prev-guardian-set))
      (latest-stacks-timestamp (unwrap! (get-stacks-block-info? time (- stacks-block-height u
    ) (ok (and (is-eq prev-guardian-set-id set-id) (>= prev-guardian-set-expires-at latest-st
  )
)
```

Stake

← back to price feeds

STX/USD

STACKS / US DOLLAR

0xec7a775f46379b5e943c3526b1c8d54cd49749176b0b98e02dde68d1bd335c17

Price
$0.980596

i Confidence
±$0.0012

Last Updated
<2s ago

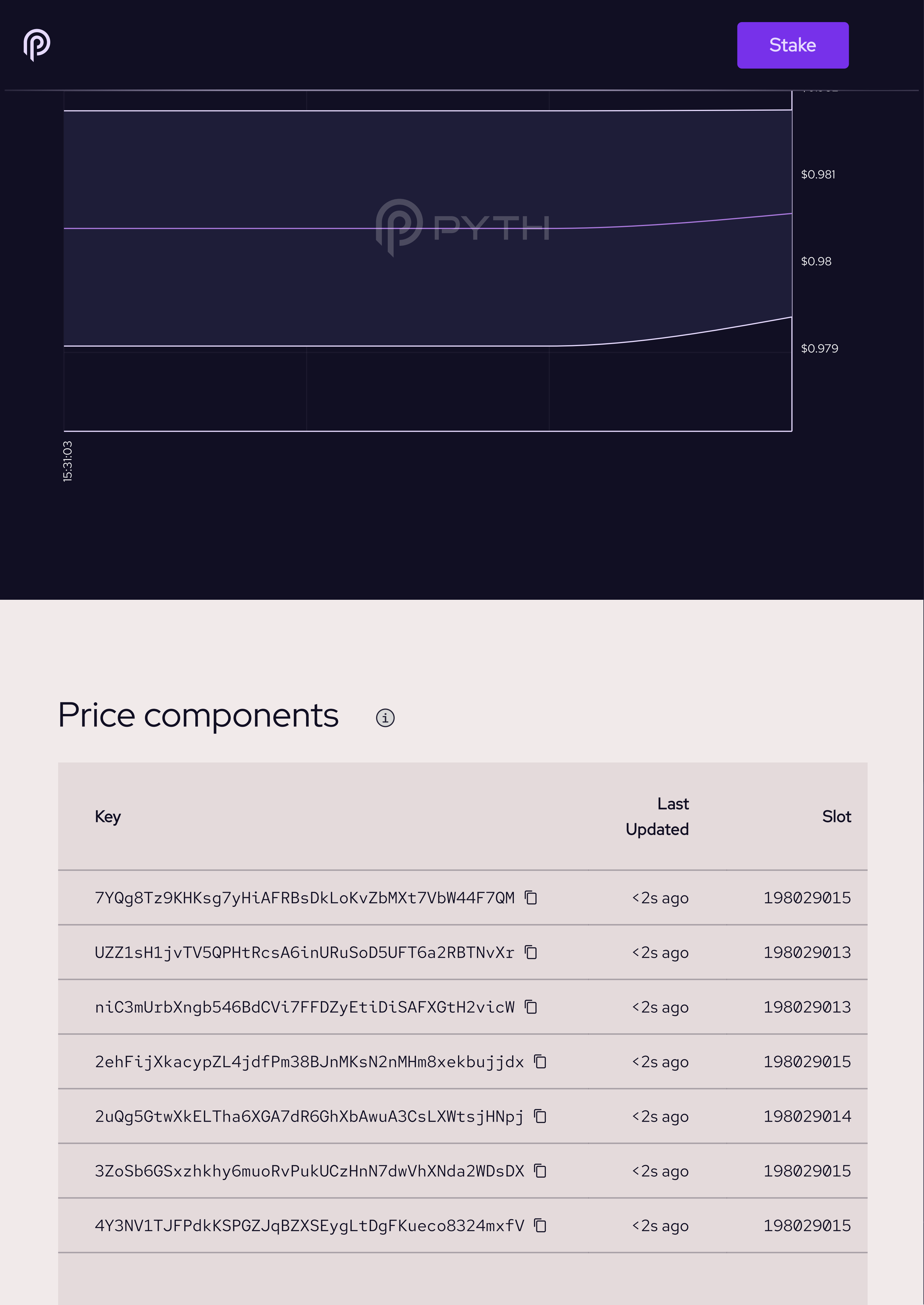
Asset Type
Crypto

i 1Hr EMAP
$0.984728

i 1Hr EMAC
$0.0013

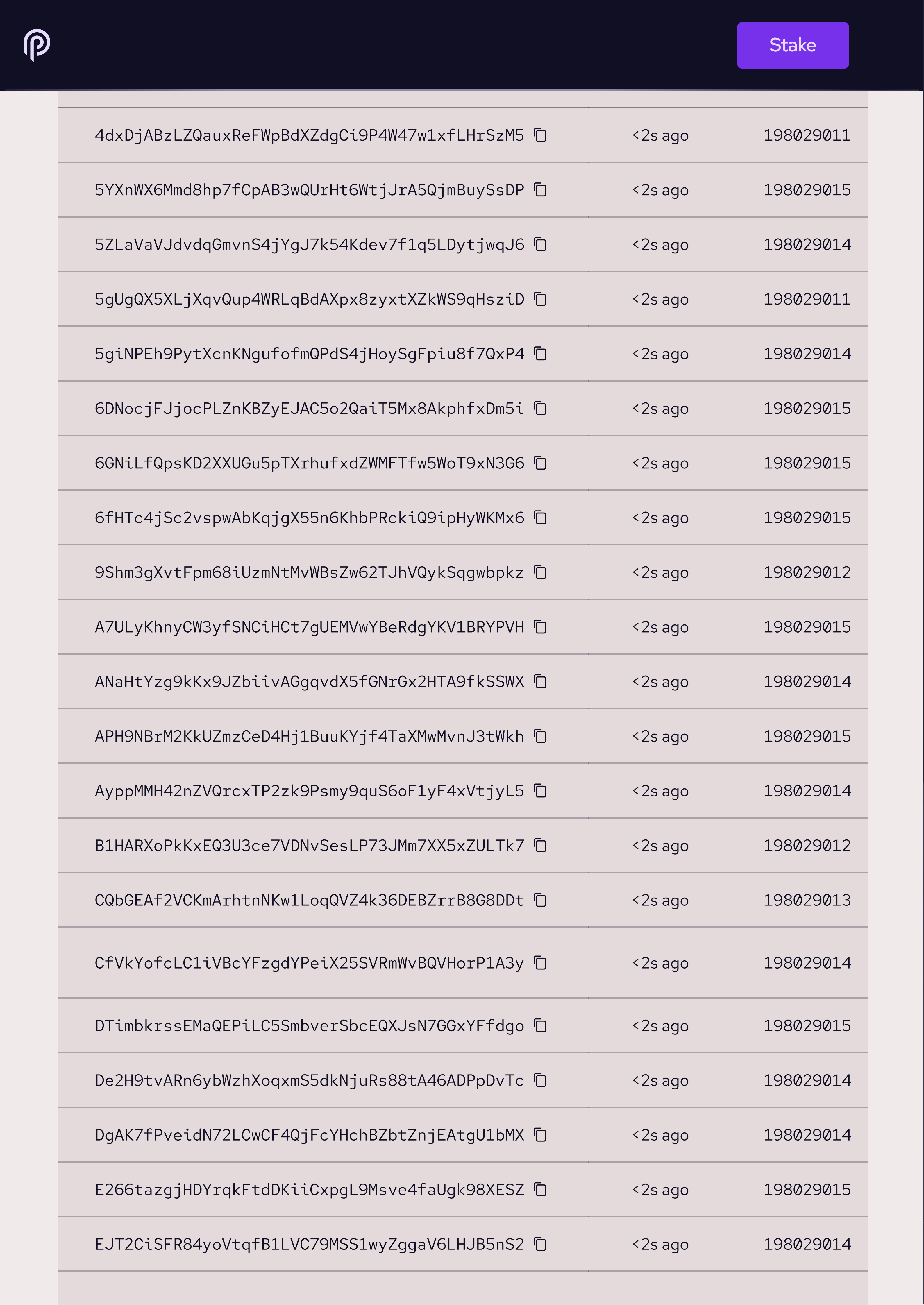
Live | 1 hour | 1 day | 1 week | 1 month

pythnet



Price components

Key	Last Updated	Slot
7YQg8Tz9KHKsg7yHiAFRBsDkLoKvZbMXt7VbW44F7QM	<2s ago	198029015
UZZ1sH1jvTV5QPHtRcsA6inURuSoD5UFT6a2RBTNvXr	<2s ago	198029013
niC3mUrbXngb546BdCVi7FFDZyEtiDiSAFXGtH2vicW	<2s ago	198029013
2ehFijXkacypZL4jdfPm38BJnMKsN2nMHm8xekbujjdx	<2s ago	198029015
2uQg5GtwXkELTha6XGA7dR6GhXbAwuA3CsLXWtsjHNpj	<2s ago	198029014
3ZoSb6GSxzhkhy6muoRvPukUCzHnN7dwVhXNda2WDsDX	<2s ago	198029015
4Y3NV1TJFPdkKSPGZJqBZXSEygLtDgFKueco8324mxfV	<2s ago	198029015



4dxDjABzLZQauxReFWpBdXZdgCi9P4W47w1xfLHrSzM5	<2s ago	198029011
5YXnWX6Mmd8hp7fCpAB3wQUrHt6WtjJrA5QjmBuySsDP	<2s ago	198029015
5ZLaVaVJdvdqGmvnS4jYgJ7k54Kdev7f1q5LDytjwqJ6	<2s ago	198029014
5gUgQX5XLjXqvQup4WRLqBdAXpx8zyxtXZkWS9qHsziD	<2s ago	198029011
5giNPEh9PytXcnKNgufofmQPdS4jHoySgFpiu8f7QxP4	<2s ago	198029014
6DNocjFJjocPLZnKBZyEJAC5o2QaiT5Mx8AkphfxDm5i	<2s ago	198029015
6GNiLfQpsKD2XXUGu5pTXrhufxdZWMFTfw5WoT9xN3G6	<2s ago	198029015
6fHTc4jSc2vspwAbKqjgX55n6KhbPRckiQ9ipHyWKMx6	<2s ago	198029015
9Shm3gXvtFpm68iUzmNtMvWBsZw62TJhVQykSqgwbpkz	<2s ago	198029012
A7ULyKhnyCW3yfSNCiHCt7gUEMVwYBeRdgYKV1BRYPVH	<2s ago	198029015
ANaHtYzg9kKx9JZbiivAGgqvdX5fGNrGx2HTA9fkSSWX	<2s ago	198029014
APH9NBrM2KkUZmzCeD4Hj1BuuKYjf4TaXMwMvnJ3tWkh	<2s ago	198029015
AyppMMH42nZVQrcxTP2zk9Psmy9quS6oF1yF4xVtjyL5	<2s ago	198029014
B1HARXoPkKxEQ3U3ce7VDNvSesLP73JMm7XX5xZULTk7	<2s ago	198029012
CQbGEAf2VCKmArhtnNKw1LoqQVZ4k36DEBZrrB8G8DDt	<2s ago	198029013
CfVkYofcLC1iVBcYFzgdYPeiX25SVRmWvBQVHorP1A3y	<2s ago	198029014
DTimbkrssEMaQEPiLC5SmbverSbcEQXJsN7GGxYFfdgo	<2s ago	198029015
De2H9tvARn6ybWzhXoqxmS5dkNjuRs88tA46ADPpDvTc	<2s ago	198029014
DgAK7fPveidN72LCwCF4QjFcYHchBZbtZnjEAtgU1bMX	<2s ago	198029014
E266tazgjHDYrqkFtdDKiiCxpgL9Msve4faUgk98XESZ	<2s ago	198029015
EJT2CiSFR84yoVtqfB1LVC79MSS1wyZggaV6LHJB5nS2	<2s ago	198029014

FBpNUB6YkYSoDhaSdyPnU8966Kp2FPTp2rGKaqXK1zwR	<2s ago	198029014
FMyhZGgD6QXoMpKdxTLJVpunWUpcvELEmMQihqHmwTjp	2s ago	198029010
GdHd8HjqX3we9WcKth1HkHUD9r8kCSSYuFfR1sb1v3nH	<2s ago	198029013
H2ViysxKfEYSWaK3fbdYeJvsr325z6u7kwm5sVwjfSoN	2s ago	198029009

Product

GZGPKYLFyCBDiVpvWG2TDikST6bRnXSc qdhRQDqiFBRM

Key	Value
Asset Type	Crypto
Base	STX
Description	STACKS / US DOLLAR
Display Symbol	STX/USD
Generic Symbol	STXUSD
Quote Currency	USD
Schedule	America/New_York;O,O,O,O,O,O,O;
Symbol	Crypto.STX/USD

Price

Gv7XY6jphWwjdpqfoip6gCMhUtH7 48DFJM1drLtbgoFU

Key	Value
Price Type	Price
Exponent	-8
Number of Price Components	32
Number of Quoters	32
Minimum Number of Publishers	5
Max Slot Latency	10
Last Slot	198029015
Valid Slot	198029014
Price	98059590
Confidence	118881
Status	Online
EMA Price	98472757

EMA Confidence	126197

How To Fetch Price Updates

The following guide explains how to fetch price updates. Price updates can be submitted to the Pyth Price Feeds contract to update the on-chain price. Please see What is a Pull Oracle? to learn more.

Price updates are served from Hermes, which provides three different ways to fetch price updates:

1. REST API
2. Streaming
3. SDK

ℹ Fetching a price from Hermes requires a price feed ID. This ID serves as a unique identifier for each price feed (e.g., BTC/USD). The complete list of Pyth price feed IDs is available at https://pyth.network/developers/price-feed-ids

REST API

Hermes exposes several endpoints to fetch the price updates. Use the `/v2/updates/price/latest` endpoint to fetch the latest price update for one or more feeds. This endpoint allows you to fetch the latest price updates for multiple feeds in a single request. For example, the following command retrieves the latest price updates for BTC/USD and ETH/USD:

```
curl -X 'GET' \
  'https://hermes.pyth.network/v2/updates/price/latest?ids%5B%5D=0xe62df6c8b4a85fe1a67db
```

The output will be similar to the following containing the requested price update:

```
{
  "binary": {
    "encoding": "hex",
    "data": [
      "504e41550100000003b801000000040d00561f4ceb8ce5eb58adda318009817714a017b0db9a7f1ef
    ]
  },
  "parsed": [
    {
      "id": "e62df6c8b4a85fe1a67db44dc12de5db330f7ac66b72
      "price": {
        "price": "6140993501000",
```

Ask me anything about Pyth

```
        "conf": "3287868567",
        "expo": -8,
        "publish_time": 1714746101
      },
      "ema_price": {
        "price": "6094004700000",
        "conf": "3792887800",
        "expo": -8,
        "publish_time": 1714746101
      },
      "metadata": {
        "slot": 138881186,
        "proof_available_time": 1714746103,
        "prev_publish_time": 1714746101
      }
    },
    {
      "id": "c96458d393fe9deb7a7d63a0ac41e2898a67a7750dbd166673279e06c868df0a",
      "price": {
        "price": "4959503",
        "conf": "5465",
        "expo": -8,
        "publish_time": 1714746101
      },
      "ema_price": {
        "price": "4982594",
        "conf": "5536",
        "expo": -8,
        "publish_time": 1714746101
      },
      "metadata": {
        "slot": 138881186,
        "proof_available_time": 1714746103,
        "prev_publish_time": 1714746101
      }
    }
  ]
}
```

Hermes offers several other endpoints for retrieving price updates. For more information, see the Hermes API Reference.

Streaming

Hermes also provides a Server-Sent Events (SSE) endpoint to stream price updates. The `/v2/updates/price/stream` endpoint continuously streams price updates for the requested feeds to the caller.

For example, to stream price updates for BTC/USD, run:

```
curl -N 'https://hermes.pyth.network/v2/updates/price/stream?ids[]=0xe62df6c8b4a85fe1a67
```

The output is a stream of events containing the requested price updates, similar to the following:

```
data:{"binary":{"encoding":"hex","data":["504e41550100000003b801000000040d00eabd2d495ed4

data:{"binary":{"encoding":"hex","data":["504e41550100000003b801000000040d00c225b810b047
```

SDK

Pyth provides a typescript SDK for Hermes to fetch price updates. The `HermesClient` class in this SDK connects to Hermes to fetch and stream price updates.

```
const connection = new HermesClient("https://hermes.pyth.network", {});

const priceIds = [
  // You can find the ids of prices at https://pyth.network/developers/price-feed-ids
  "0xe62df6c8b4a85fe1a67db44dc12de5db330f7ac66b72dc658afedf0f4a415b43", // BTC/USD price
  "0xff61491a931112ddf1bd8147cd1b641375f79f5825126d665480874634fd0ace", // ETH/USD price
];

// Get price feeds
// You can also fetch price feeds for other assets by specifying the asset name and asse
const priceFeeds = await connection.getPriceFeeds("btc", "crypto");
console.log(priceFeeds);

// Latest price updates
const priceUpdates = await connection.getLatestPriceUpdates(priceIds);
console.log(priceUpdates);
```

`HermesClient` also allows subscribing to real-time price updates over a Server-Sent Events (SSE) connection:

```
// Streaming price updates
const eventSource = await connection.getStreamingPriceUpdates(priceIds);

eventSource.onmessage = (event) => {
  console.log("Received price update:", event.data);
};

eventSource.onerror = (error) => {
  console.error("Error receiving updates:", error);
  eventSource.close();
};
```

```
await sleep(5000);

// To stop listening to the updates, you can call eventSource.close();
console.log("Closing event source.");
eventSource.close();
```

Last updated on January 28, 2025

Price Feeds > Use Real-Time Price Data

How to Use Real-Time Price Data

The following guides demonstrate how to consume Pyth real-time prices on various blockchains. These guides are intended for developers building on-chain applications that need price data, i.e., the price data must be on the blockchain.

Pyth price feeds are available on 40+ blockchain ecosystems. Check out the complete list of chains and implementation contract addresses at Contract Addresses.

If your blockchain is not supported, please ask in Discord. Then, consult the relevant ecosystem guide to get started using Pyth real-time price data:

- EVM
- Solana
- Aptos
- CosmWasm
- Sui
- Near

Pyth price feeds can also be used in off-chain applications. For example, an application may need to show real-time asset prices on a website. Developers building such applications can consult the following guide:

- Off-chain Apps

Off-chain application developers should also consider using Benchmarks. In addition to real-time data, Benchmarks provides access to historical Pyth prices. These historical prices are useful for building price charts or graphs.

Last updated on January 28, 2025

How To Schedule Price Updates

The following guides explain how to schedule Pyth price updates to occur at regular intervals. As a pull oracle, Pyth's users are typically responsible for updating the state of on-chain feeds. Please see What is a Pull Oracle? to learn more about pull updates.

The Pyth Data Association sponsors regular on-chain updates for some price feeds. See Sponsored Feeds for the current list of feeds and their update parameters. If you would like to see additional feeds on this list, please contact the association via this form.

There are also two different tools to schedule price updates:

- Gelato provides a turnkey automation solution for scheduled updates.
- Scheduler is a service that developers can run to trigger price updates when certain time or price change conditions are met.

For developers comparing these two options, Gelato is simpler, in that it does not require you to operate a service. However, Scheduler supports more blockchains than Gelato.

Last updated on January 28, 2025

Searc

Testnet

Explore

Sandbox

Testnet

$96,899.11

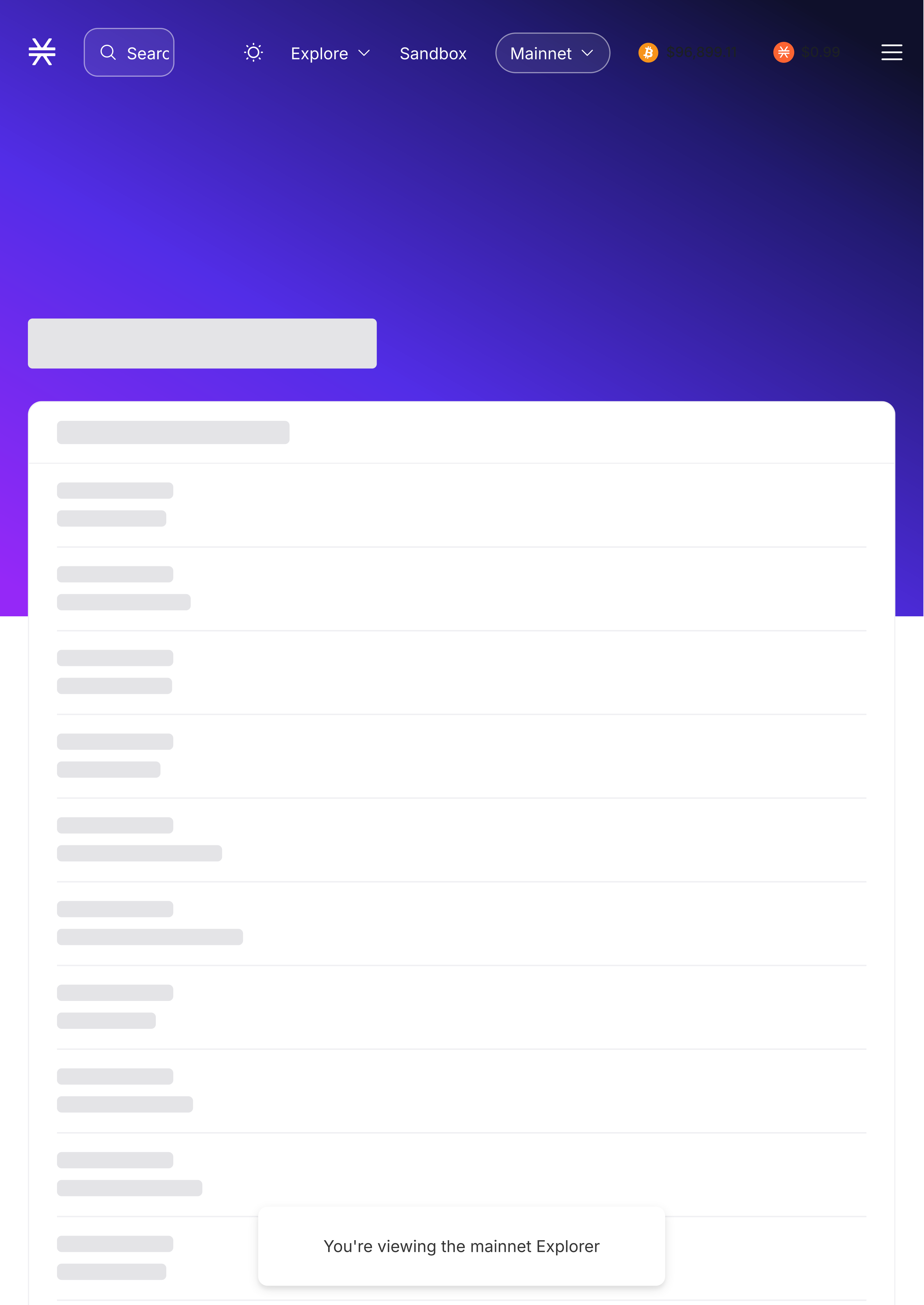
$0.99

You're viewing the testnet Explorer

You're viewing the testnet Explorer

You're viewing the testnet Explorer

You're viewing the testnet Explorer



You're viewing the mainnet Explorer

You're viewing the mainnet Explorer

You're viewing the mainnet Explorer

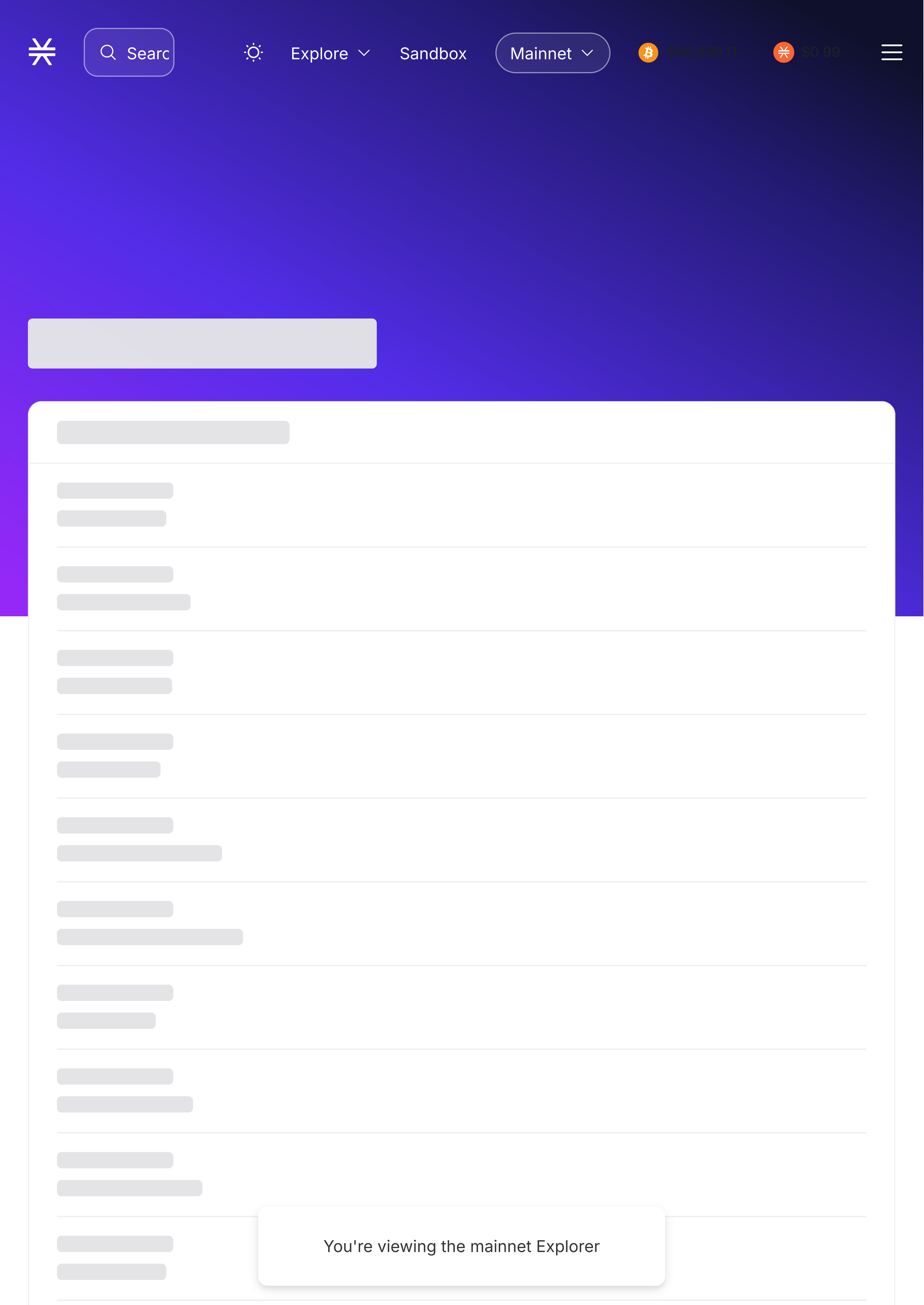
Recent transactions Sandbox Found a bug in the Stacks Blockchain?

Market data provided by LunarCrush

Support Submit bug or feature request Terms & Privacy

Version 1.249.0

You're viewing the mainnet Explorer



You're viewing the mainnet Explorer

You're viewing the mainnet Explorer

You're viewing the mainnet Explorer

You're viewing the mainnet Explorer