Title: **Oracles on XRP Ledger**
Revision: **3** (2023-10-18)

Author: Gregory Tsipenyuk
Affiliation: Ripple

Oracles on XRP Ledger

Abstract

This proposal adds on-chain `PriceOracle` object to XRPL ledger. A blockchain oracle is a system or service that acts as a bridge between a blockchain network and the external world, providing off-chain data or information to decentralized applications (dApps) on the blockchain. Oracles are used to bring real-world data, for instance market prices, exchange rates, interest rates, or weather conditions onto the blockchain, enabling dApps to access and utilize information that resides outside the blockchain. This document outlines the protocols involved in `PriceOracle` on XRPL ledger and provides guidelines for developers and system architects to implement and utilize this solution effectively. It introduces a new on-ledger `PriceOracle` object and the transactions to create, delete, and update the `PriceOracle` and adds `get_aggregate_price` API to retrieve an aggregate mean, trimmed mean, and median for the provided price oracles. This feature requires an amendment.

Terminology

- Oracle Provider is a service or technology that enables the integration of external data and real-world events into a blockchain network.
- dApp, short for decentralized application, refers to an application that is built on a blockchain network and operates using smart contracts or rely on other mechanisms or protocols for their functionality.

Creating `PriceOracle` instance on XRPL

On-Ledger Data Structure

The `PriceOracle` Object

The `PriceOracle` ledger entry represents the `PriceOracle` object on XRPL ledger and contains the following fields:

FieldName	Required?	JSON Type	Internal Type
`LedgerEntryType`	✓	`string`	`UINT16`
`Owner`	✓	`string`	`ACCOUNTID`
`Provider`	✓	`string`	`BLOB`
`PriceDataSeries`	✓	`array`	`ARRAY`
`LastUpdateTime`	✓	`number`	`UINT32`
`URI`		`string`	`BLOB`
`SymbolClass`	✓	`string`	`BLOB`
`PreviousTxnID`	✓	`string`	`UINT256`
`PreviousTxnLgrSeq`	✓	`string`	`UINT32`

- `LedgerEntryType` identifies the type of ledger object. The proposal recommends the value 0x0080 as the reserved entry type.
- `Owner` is the account that has the update and delete privileges. It is recommended that this account has an associated `signer list`.
- `Provider` identifies an Oracle Provider. It can be URI or any data, for instance `chainlink`. It is a string of up to 256 ASCII hex encoded characters (0x20-0x7E).
- `PriceDataSeries` is an array of up to ten `PriceData` objects, where `PriceData` represents the price information for a token pair. `PriceOracle` with more than five `PriceData` objects requires two owner reserves. `PriceData` includes the following fields:

FieldName	Required?	JSON Type	Internal Type
`Symbol`	✓	`string`	`CURRENCY`
`PriceUnit`	✓	`string`	`CURRENCY`
`SymbolPrice`		`number`	`UINT64`
`Scale`		`number`	`UINT8`

  - `Symbol` is the symbol to be priced. Any arbitrary value should be allowed and interpreted exactly like other asset code fields in the ledger. A new enum value STI_CURRENCY and class STCurrency are introduced to support the `CURRENCY` field.
  - `PriceUnit` is the denomination in which the prices are expressed. Any arbitrary value should be allowed and interpreted exactly like other asset code fields in the ledger.
  - `SymbolPrice` is the scaled asset price, which is the price value after applying the scaling factor. This is an optional field. It is not included if the last update transaction didn't include the `Symbol`/`PriceUnit` pair.
  - `Scale` is the price's scaling factor. It represents the price's precision level. For instance, if `Scale` is `6` and the original price is `0.155` then the scaled price is

155000. Formally, $scaledPrice = originalPrice * 10^{scale}$. Valid Scale range is {0-10}. This is an optional field. It is not included if the last update transaction didn't include the Symbol/PriceUnit pair.

- URI is an optional URI field to reference the price data off-chain. It is a string of up to 256 ASCII hex encoded characters (0x20-0x7E).
- SymbolClass describes a type of the assets, for instance "currency", "commodity", "index". It is a string of up to ten ASCII hex encoded characters (0x20-0x7E).
- LastUpdateTime is the specific point in time when the data was last updated. The LastUpdateTime is the ripple epoch time.
- PreviousTxnID is the hash of the previous transaction to modify this entry. (Same as on other objects with this field.).
- PreviousTxnLgrSeq is the ledger index of the ledger when this object was most recently updated/created. (Same as other objects with this field.)

The **PriceOracle** Object ID Format

We compute the PriceOracle object ID, a.k.a., OracleID, as the SHA-512Half of the following values, concatenated in order:

- The Oracle space key (0x52)
- The Owner Account ID
- The Oracle Sequence. This field must be passed to the transactions and it describes a unique Price Oracle sequence for the given account.

Example of **PriceOracle** JSON

```
{
    "LedgerEntryType": "PriceOracle",
    "OracleID":
"00070C4495F14B0E44F78A264E41713C64B5F89242540EE255534400000000000000
",
    "Owner": "rsA2LpzuawewSBQXkiju3YQTMzW13pAAdW",
    "Provider": "70726F7669646572",
    "SymbolClass": "63757272656E6379",
    "PriceDataSeries": [
      {
        "PriceData": {
          "Symbol": "XRP",
          "PriceUnit": "USD",
          "SymbolPrice": 74,
          "Scale": 2,
        }
      },
    ],
    "LastUpdateTime": 743609414,
    "PreviousTxnID":
"C53ECF838647FA5A4C780377025FEC7999AB4182590510CA461444B207AB74A9",
    "PreviousTxnLgrSeq": 56865244
}
```

Transactions

This proposal introduces several new transactions to allow for the creation, update, and deletion of the `PriceOracle` object.

Transaction for creating or updating `PriceOracle` instance

We define a new transaction **SetOracle** for creating or updating a `PriceOracle` instance. Before the transaction can be submitted to create a new `PriceOracle` instance, the Oracle Provider has to do the following:

- Create or own the `Account` XRPL account with sufficient XRP balance to meet the XRP reserve and the transaction fee requirements.
- The Oracle Provider has to publish the `Account` account public key so that it can be used for verification by dApp's.
- The Oracle Provider has to publish a registry of available Price Oracles with their unique Oracle Sequence. The hash of the `Account` and the `OracleSequence` uniquely identifies the Price Oracle on-ledger object.

Transaction fields for SetOracle transaction

FieldName	Required?	JSON Type	Internal Type
`TransactionType`	✓	`string`	UINT16

Indicates a new transaction type `SetOracle`. The integer value is 49.

FieldName	Required?	JSON Type	Internal Type
`Account`	✓	`string`	`ACCOUNTID`

`Account` is the account that has the Oracle update and delete privileges.

FieldName	Required?	JSON Type	Internal Type
`OracleSequence`	✓	`string`	`UINT32`

`OracleSequence` is a unique identifier of the Price Oracle for the given Account.

FieldName	Required?	JSON Type	Internal Type
`Provider`		`string`	`BLOB`

`Provider` identifies an Oracle Provider. `Provider` must be included when creating a new instance of `PriceOracle`.

FieldName	Required?	JSON Type	Internal Type
`URI`		`string`	`BLOB`

`URI` is an optional field to reference the price data off-chain.

FieldName	Required?	JSON Type	Internal Type
`SymbolClass`		`string`	`BLOB`

`SymbolClass` describes the assets type.

FieldName	Required?	JSON Type	Internal Type
`LastUpdateTime`	✓	`number`	`UINT32`

`LastUpdateTime` is the specific point in time when the data was last updated.

FieldName	Required?	JSON Type	Internal Type
`PriceDataSeries`	✓	`array`	`ARRAY`

`PriceDataSeries` is an array of up to ten `PriceData` objects, where `PriceData` represents the price information for a token pair. `PriceData` includes the following fields:

FieldName	Required?	JSON Type	Internal Type
`Symbol`	✓	`string`	`CURRENCY`

`Symbol` is the symbol to be priced.

FieldName	Required?	JSON Type	Internal Type
`PriceUnit`	✓	`string`	`CURRENCY`

`PriceUnit` is the denomination in which the prices are expressed.

FieldName	Required?	JSON Type	Internal Type
`SymbolPrice`	✓	`number`	`UINT64`

`SymbolPrice` is the scaled asset price, which is the price value after applying the scaling factor.

FieldName	Required?	JSON Type	Internal Type
`Scale`	✓	`number`	`UINT8`

`Scale` is the price's scaling factor.

The transaction fails if:

- A required field is missing.
- XRP reserve is insufficient. If the Oracle instance has less or equal than five token pairs then the XRP reserve requirements is one, otherwise the XRP reserve requirements is two.
- Transaction's `PriceDataSeries` array size is empty or exceeds ten when creating a new Oracle instance or Oracle's instance `PriceDataSeries` array size exceeds ten after updating the Oracle instance.
- `PriceDataSeries` has duplicate token pairs.
- `PriceDataSeries` has array elements with missing `SymbolPrice`.
- The `Account` account doesn't exist or the `Account` is not equal to the `Owner` field when updating the Oracle instance.
- The transaction is not signed by the `Account` account or the account's multi signers.
- The `URI` field length exceeds 64 bytes.
- The `Provider` field length exceeds 64 bytes.
- The `SymbolClass` field length exceeds 12 bytes.

If an object with the `OracleID` Object ID already exists then the new token pairs are added to the Oracle instance `PriceDataSeries` array. Note that the order of the token pairs in the `PriceDataSeries` array is not important since the token pair uniquely identifies location in the `PriceDataSeries` array of the `PriceOracle` object. Also note that not every token pair price has to be updated. I.e., even though the `PriceOracle` may define ten token pairs, `SetOracle` transaction may contain only one token pair price update. In this case the missing token pair will not include `SymbolPrice` and `Scale` fields. `PreviousTxnID` can be used to find the last updated Price Data for this token pair.

On success the transaction creates a new or updates existing `PriceOracle` object. If a new object is created then the owner reserve requirement is incremented by one or two depending on the `PriceDataSeries` array size.

Example of SetOracle transaction JSON

```
{
    "TransactionType": "SetOracle",
    "Account": "rsA2LpzuawewSBQXkiju3YQTMzW13pAAdW",
    "OracleSequence": 34,
    "Provider": "70726F7669646572",
    "LastUpdateTime": 743609014,
    "SymbolClass": "63757272656E6379",
    "PriceDataSeries": [
      {
        "PriceData": {
          "Symbol": "XRP",
          "PriceUnit": "USD",
          "SymbolPrice": 740,
          "Scale": 3
        }
      }
    ]
}
```

Transaction for deleting Oracle instance

We define a new transaction **DeleteOracle** for deleting an Oracle instance.

Transaction fields for DeleteOracle transaction

FieldName	Required?	JSON Type
`TransactionType`	✔	`string`

Indicates a new transaction type `DeleteOracle`. The integer value is 42.

FieldName	Required?	JSON Type	Internal Type
`Account`	✔	`string`	`ACCOUNTID`

`Account` is the account that has the Oracle update and delete privileges.

FieldName	Required?	JSON Type	Internal Type
`OracleSequence`	✔	`string`	`UINT32`

`OracleSequence` is a unique identifier of the Price Oracle for the given Account.

DeleteOracle transaction deletes the `Oracle` object from the ledger.

The transaction fails if:

- Object with the `OracleID` Object ID doesn't exist.
- The transaction is not signed by the `Account` account or the account's multi signers.

On success the transaction deletes the `Oracle` object and the owner’s reserve requirement is reduced by one or two depending on the `PriceDataSeries` array size.

Example of DeleteOracle transaction JSON

```
{
    "TransactionType": "DeleteOracle",
    "Account": "rsA2LpzuawewSBQXkiju3YQTMzW13pAAdW",
    "OracleSequence": 34
}
```

API's

Retrieving The Oracle

An Oracle object can be retrieved with the `ledger_entry` API call by specifying the `account` and `oracle_sequence`.

Example of `ledger_entry` API JSON

Request JSON

```
{
     "method ":  "ledger_entry ",
     "params" : [
         "account": "rsA2LpzuawewSBQXkiju3YQTMzW13pAAdW",
         "oracle_sequence":  34,
         "ledger_index ":  "validated "
     ]
}
```

Response JSON

```
{
   "index" :
"CF2C20122022DE908C4F521A96DC2C1E5EFFD1EFD47AA244E9EE9A442451162E",
   "ledger_current_index" : 23,
   "node" : {
      "Flags" : 0,
      "LastUpdateTime" : 743609014,
      "LedgerEntryType" : "Oracle",
      "Owner" : "rp847ow9WcPmnNpVHMQV5A4BF6vaL9Abm6",
      "SymbolClass" : "63757272656E6379",
      "Provider": "70726F7669646572",
      "PreviousTxnID" :
"6F120537D0D212FEA6E11A0DCC5410AFCA95BD98D451D046832E6C4C4398164D",
      "PreviousTxnLgrSeq" : 22,
      "PriceDataSeries": [
        {
          "PriceData: {
            "PriceUnit" : {
               "currency" : "USD"
            },
            "Symbol" : {
               "currency" : "XRP"
            },
            "Scale" : 1,
            "SymbolPrice" : "740",
          }
        }
        ],
        "index" :
"CF2C20122022DE908C4F521A96DC2C1E5EFFD1EFD47AA244E9EE9A442451162E"
    },
    "status" : "success",
    "validated" : true
}
```

Oracle Aggregation

`get_aggregate_price` API calculates the aggregate price of the specified PriceOracle's. `get_aggregate_price` returns three types of price statistics - average, median, and trimmed median if `trim` parameter is included in the API.

API fields are:

FieldName	Required?	JSON Type
`ledger_index`		`string` or `number` (positive integer)

The ledger index of the max ledger to use, or a shortcut string to choose a ledger automatically.

FieldName	Required?	JSON Type
`ledger_hash`		`string`

A 20-byte hex string for the max ledger version to use.

FieldName	Required?	JSON Type
`symbol`	✓	`string`

`symbol` is the symbol to be priced.

FieldName	Required?	JSON Type
`price_unit`	✓	`string`

`price_unit` is the denomination in which the prices are expressed.

FieldName	Required?	JSON Type
`oracles`	✓	`array`

`oracles` is an array of `oracle` objects to aggregate over. `oracle` object has two fields:

FieldName	Required?	JSON Type
`account`	✓	`string`

  `account` is the Oracle's account.

FieldName	Required?	JSON Type
oracle_sequence	✓	number

oracle_sequence is a unique identifier of the Price Oracle for the given Account.

FieldName	Required?	JSON Type
trim		number

trim is the percentage of outliers to trim. Valid trim range is 1-25. If this parameter is included then the API returns trimmed median in addition to the average and median.

FieldName	Required?	JSON Type
time_threshold		number

The time_threshold is used to define a time range in seconds for filtering out older price data. time_threshold is 4 seconds by default.

The price data to aggregate is selected based on specific criteria. The most recent Price Oracle object is obtained for the specified oracles. The most recent LastUpdateTime among all objects is chosen as the upper time threshold. A Price Oracle object is included in the aggregation dataset if it satisfies the conditions of containing the specified symbol/price_unit pair, including the SymbolPrice field, and its LastUpdateTime is within the time range of (upper threshold - time threshold) to the upper threshold. If a Price Oracle object doesn't contain the SymbolPrice for the specified token pair, then up to three previous Price Oracle objects are examined and include the first one that fulfills the criteria.

The get_aggregate_price fails if:

- The oracles array size is either 0 or greater than 200.
- The oracles array's object doesn't include account or oracle_sequence or those fields have invalid value.
- symbol or price_unit are missing.
- trim or time_threshold contain invalid uint value.
- If the resulting data set is empty.

On success, the response data contains the following fields:

- entire_set is an object of the following fields:
    - size is the size of the data set used to calculate the statistics.
    - average is the simple average.
    - standard_deviation is the standard deviation.
- trimmed_set is an object, which is included in the response if trim fields is set. The object has the following fields:
    - size is the size of the data set used to calculate the statistics.
    - average is the simple average.
    - standard_deviation is the standard deviation.
- median is the median.

- `time` is the most recent time stamp out of all `LastUpdateTime` values.

Example of get_aggregate_price API JSON

Request JSON

```
{
"method": "get_aggregate_price",
"params": [
    {
        "ledger_index": "current",
        "symbol": "XRP",
        "price_unit": "USD",
        "flags": 7,
        "trim": 20,
        "oracles": [
          {
            "account": "rp047ow9WcPmnNpVHMQV5A4BF6vaL9Abm6,
            "oracle_sequence": 34
          },
          {
            "account": "rp147ow9WcPmnNpVHMQV5A4BF6vaL9Abm7,
            "oracle_sequence": 56
          },
          {
            "account": "rp247ow9WcPmnNpVHMQV5A4BF6vaL9Abm8,
            "oracle_sequence": 2
          },
          {
            "account": "rp347ow9WcPmnNpVHMQV5A4BF6vaL9Abm9,
            "oracle_sequence": 7
          },
          {
            "account": "rp447ow9WcPmnNpVHMQV5A4BF6vaL9Abm0,
            "oracle_sequence": 109
          }
        ]
    }
]
}
```

Response JSON

```
{
  "entire_set" : {
     "average" : "74.75",
     "size" : 10,
     "standard_deviation" : "0.1290994448735806"
  },
  "ledger_current_index" : 25,
  "median" : "74.75",
  "status" : "success",
  "trimmed_set" : {
    "average" : "74.75",
    "size" : 6,
    "standard_deviation" : "0.1290994448735806"
  },
  "validated" : false
  "time" : 78937648
}
```