Band Protocol Documentation /
BandChain Whitepaper / Terminology

Terminology

Data Sources

A data source is the most fundamental unit in BandChain's oracle system. It describes the procedure to retrieve a raw data point from a primary source and the fee associated with one data query. On BandChain, a data source can be registered into the system by anyone. This is done through the registrant sending a `MsgCreateDataSource` message to the

Search

ON THIS PAGE

various parameters the data source they wish to register, including

- the sender who wish to create the data source
- the owner of the data source, if specified
- the name of the data source
- (Phase 2+) the per-query fee that someone looking to use that data source needs to pay
- the content of the executable to be run by block validators upon receiving a data request for this data source

When registering the data source, the message sender can choose whether to specify an owner of the source. If an owner is specified, only the owner can the only party able to collect the accumulated request fees. On the other hand, if an owner is omitted, the data source can no longer be edited after it is registered. Note that the sender who creates the data source and the owner of the data source does not need to be the same.

In the case of unowned data sources, it is the data source's configuration on BandChain that cannot be changed. If the procedures associated with that source depend on centralized sources, the actual source of the data can still be controlled by centralized parties.

Examples

The following two examples illustrate what a data source executable might look like. Both examples are written in

Retrieve Cryptocurrency Price from CoinGecko

The data source requires that cURL ↗ and jq ↗ are installed on the executable runner's machine and expects one argument; the currency ticker symbol.

```
#!/bin/sh

# Cryptocurrency price endpoint: https://
URL="https://api.coingecko.com/api/v3/sim
KEY=".$1.usd"

# Performs data fetching and parses the r
curl -s -X GET $URL -H "accept: applicati
```

Resolve Hostname to IP Addresses

Again, this script assumes that getent ↗ and awk ↗ are available on the host and the host is connected to the DNS network

```
#!/bin/sh

getent hosts $1 | awk '{ print $1 }'
```

Oracle Scripts

When someone wants to request data from BandChain's oracle, they must do so by calling one of the available oracle scripts. An oracle script is an executable program that encodes:

- the set of raw data requests to the sources it needs
- the way to aggregate raw data reports into the final result

These sources can be the data sources published on the network, as well as other oracle scripts. Oracle scripts are

This composability and Turing-completeness makes oracle scripts very similar to smart contracts.

To create an oracle script, the creator must broadcast a `MsgCreateOracleScript` to BandChain. The contents of the message includes:

- the sender who wishes to create the oracle script
- the owner of the oracle script, if specified
- the name of the oracle script
- the OWasm compiled binary attached to this oracle script
- the schema detailing the inputs and outputs of this oracle script, as well as the corresponding types
- the URL for the source code of this

Similar to data sources, the sender who wishes to create the oracle script does not have to be the same as the owner of the oracle script specified in the message.

The execution flow of an oracle script can be broken down into two phases. In the first phase, the script outlines the data sources that are required for its execution. It then sends out a request to the chain's validators to retrieve the result from the required data sources. The contents of this consists of the data sources' execution steps and the associated parameters.

The second phase then aggregates all of the data reports returned by the validators, with each report containing the values the validator received from the required data sources. The script then proceeds to combine those values

Note that the specifics of the aggregation process is entirely up to the design of the oracle script. BandChain does not enforce any regulations when it comes to the aggregation method used, and entirely leaves that design decision to the creator of the script or any subsequent editors.

Example

The pseudocode below shows an example of an oracle script that returns the current price of a cryptocurrency. The script begins by emitting requests to validators to query the price from three data sources (i.e. the `request` function calls to CoinGecko, CryptoCompare, CoinMarketCap inside `prepare`). Once a sufficient number of validators have reported the prices, the

out the reported values results into a

single final result (the `aggregate` function).

In this particular oracle script, the aggregation process starts by summing all of the price values returned by the validators across all data sources, as well as the total number of reports returned. It then simply divides the summed price value with the number of data reports returned to arrive at the final average value.

```
# 1st Phase. Emits raw data requests that
def prepare(symbol):
    request(get_px_from_coin_gecko, symbc
    request(get_px_from_crypto_compare, s
    request(get_px_from_coin_market_cap,

# 2nd Phase. Aggregates raw data reports
def aggregate(symbol, number_of_reporters
    data_report_count = 0
    price_sum = 0.0
    for reporter_index in range(number_of
        for data_source in (
            get_px_from_coin_gecko,
            get_px_from_crypto_compare,
            get_px_from_coin_market_cap,
        ):
            price_sum = receive(reporter_
            data_report_count += 1
    return price_sum / data_report_count
```

Raw Data Reports

Raw data reports are the results that BandChain's validators return when they have successfully responded to a sources. In these reports, the validators list out the result they got from each data source, using the data source's external ID as the reference key. The external ID is the identifier used to reference a data source within an oracle script, and each data source's external ID is unique within the context of that script.

Oracle Data Proof

When the final data request result is successfully stored onto BandChain, an oracle data proof is produced. This proof is a Merkle proof that shows the existence of the final result of the data request on BandChain. In addition to the actual result value of the request, the proof contains information on the request parameters (oracle script hash,

etc) as well as as well as those of the

associated response (e.g. number of validators that responded to the request). This proof can then be used by smart contracts on other blockchain to verify the existence of the data as well as to decode and retrieve the result stored. Both of these can be done by interacting with our lite client ↗.

Found an Issue?

Help us improve this page by suggesting edits on GitHub.

← PREVIOUS **Introduction**

NEXT **System Overview** →

bandprotocol.com

This website is maintained by Band Protocol. The contents and opinions of this website are those of Band Protocol.