Oracle Module

The Maker Protocol's Oracles * Module Name: * Oracle Module * Type/Category: * Oracles —> OSM.sol & Median.sol * Associated MCD System Diagram* Contract Sources: * *Median * * OSM * * *

1. Introduction (Summary)

An oracle module is deployed for each collateral type, feeding it the price data for a corresponding collateral type to theVat . The Oracle Module introduces the whitelisting of addresses, which allows them to broadcast price updates off-chain, which are then fed into amedian before being pulled into theOSM . TheSpot 'ter will then proceed to read from theOSM and will act as the liaison between theoracles anddss .

1. Module Details

The Oracle Module has 2 core components consisting of theMedian andOSM contracts.

Oracle Module Components Documentation

- Median Documentation
- OSM Documentation
-

1. Key Mechanism and Concepts

?

Summary of the OracleModule Components

- TheMedian
- provides Maker's trusted reference price. In short, it works by maintaining a whitelist of price feed contracts which are authorized to post price updates. Every time a new list of prices is received, the median of these is computed and used to update the stored value. The median has permissioning logic which is what enables the addition and removal of whitelisted price feed addresses that are controlled via governance. The permissioning logic allows governance to set other parameters that control the Median's behavior—for example, thebar
- parameter is the minimum number of prices necessary to accept a new median value.
- TheOSM
- (named via acronym from "Oracle Security Module") ensures that new price values propagated from the Oracles are not taken up by the system until a specified delay has passed. Values are read from a designatedDSValue
- contract (or any contract that implements theread()
- andpeek()
- interface) via thepoke()
- method; theread()
- andpeek()
- methods will give the current value of the price feed, and other contracts must be whitelisted in order to call these. An OSM contract can only read from a single price feed, so in practice one OSM contract must be deployed per collateral type.
-

1. Gotchas (Potential sources of user error)

Relationship between the OSM and the Median:

- You can read straight from the median and in return, you would get a more real-time price. However, this depends on the cadence of updates (calls to poke).
- The OSM is similar but has a 1-hour price delay. It has the same process for reading (whitelist, auth, read and peek) as a median. The way the OSM works, is you cannot update it directly but you canpoke
- it to go and read from something that also has the same structure (thepeek
- method - in this case, its the median but you can set it to read from anything that conforms to the same interface).
- Whenever the OSM reads from a source, it queues the value that it reads for the following hour or followinghop
- property, which is set to 1 hour (but can be anything). When it ispoke
- 'd, it reads the value of the median and it will save the value. Then the previous value becomes that, so it is always off by an hour. After an hour passes, whenpoke
- d, the value that it saved becomes the current value and whatever value is in the median becomes the future value for the next hour.
- spot
-
    - if you poke it with an ilk (ex: ETH) it will read form the OSM and if the price is valid, it updates.
-

Relationship to theSpot 'ter:

- In relation to theSpot
- the oracle module handles how market prices are recorded on the blockchain. TheSpot
- ter operates as the interface contract, which external actors can use to retrieve the current market price from the Oracle module for the specified collateral type. TheVat
- in turn reads the market price from thespot
- ter.
-

1. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

2. Median

3.
    - there is currently no way to turn off the oracle (failure or returns false) if all the oracles come together and sign a price of zero. This would result in the price being invalid and would return false onpeek
4. , telling us to not trust the value.
5. OSM
6.
    - poke()
7.
    - is not called promptly, allowing malicious prices to be swiftly uptaken.
8.
    - Authorization Attacks and Misconfigurations.
9.
    - Read morehere.
10. *
11.

Previous Vow - Detailed Documentation Next Oracle Security Module (OSM) - Detailed Documentation Last updated4 years ago On this page * 1. Introduction (Summary) * 2. Module Details * Oracle Module Components Documentation * 3. Key Mechanism and Concepts * 4. Gotchas (Potential sources of user error) * 5. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

Export as PDF