

## Price oracle

The price oracle smart contract receives input from the redistribution game. Notably the input constitutes information on storage supply relative to the current demand. Since demand is maxed out with storage depth, this information is simply captured by the number of honest revealers per neighbourhood. Rounds when there are no revealers should be treated as rounds where the number of revealers is zero therefore the number of skipped rounds is passed to the price update function too.

### Definition 1 – price function.

The *Price* function determines the unit price of storage (PLUR/chunk/block). It is defined in such a way that the ratio between the price of rent in consecutive rounds is an exponential function of supply. Supply is the deviation from the optimal number of peers in a neighbourhood as measured by the number of honest revealers in the previous round.

$$Price : \Gamma \mapsto uint256 \quad (1)$$

$$Price(\gamma) \stackrel{\text{def}}{=} Price(Prev(\gamma)) \cdot 2^{\sigma \cdot d} \quad (2)$$

where

$$\text{SUPPLY} \quad d = \text{NHOOD\_PEER\_COUNT} - |Reveals(\gamma)| \quad (3)$$

$$\text{REACTIVITY} \quad \sigma = \frac{1}{\text{PRICE\_2X\_ROUNDS}} \quad (4)$$

First, note that for a consistent deviation signal  $d$ , the price change after  $n$  rounds is given by multiplying the starting price with  $(2^{\sigma \cdot d})^n$ . The price has doubled if this expression is 2, i.e.,  $n \cdot d \cdot \sigma = 1$ . With the lowest signal of undersupply ( $r = 3$ ),  $d = 1$ , and therefore the reactivity parameter is expressed as  $\sigma = \frac{1}{n}$ . In other words  $\frac{1}{\sigma}$  measures the number of rounds it takes for the price to double.