# Market Making on Microtick

There are three main reasons to make a market and place quotes on Microtick. Number one: to participate in price discovery and add token backing, accuracy and stability to the on-chain consensus price. Number two: to earn a return on quote backing (pegged DAI on the main Microtick chain). Number three: to earn more TICK (Microtick's utility token).

This white paper covers the profit seeking motives for market making - reasons 2 and 3 above. We will discuss the general topics every market maker should understand and consider, but will leave it to individual market makers to create, discover and innovate with their own competitive algorithms.

Note that Microtick's TICK tokens can also be pegged to other chains as quote backing for cross-chain price discovery, simultaneously increasing their utility and providing a mechanism for other chains to join the real-time consensus without specifying a staking mechanism. Cross chain price discovery is a compelling use case for Microtick but is out of scope for this paper.

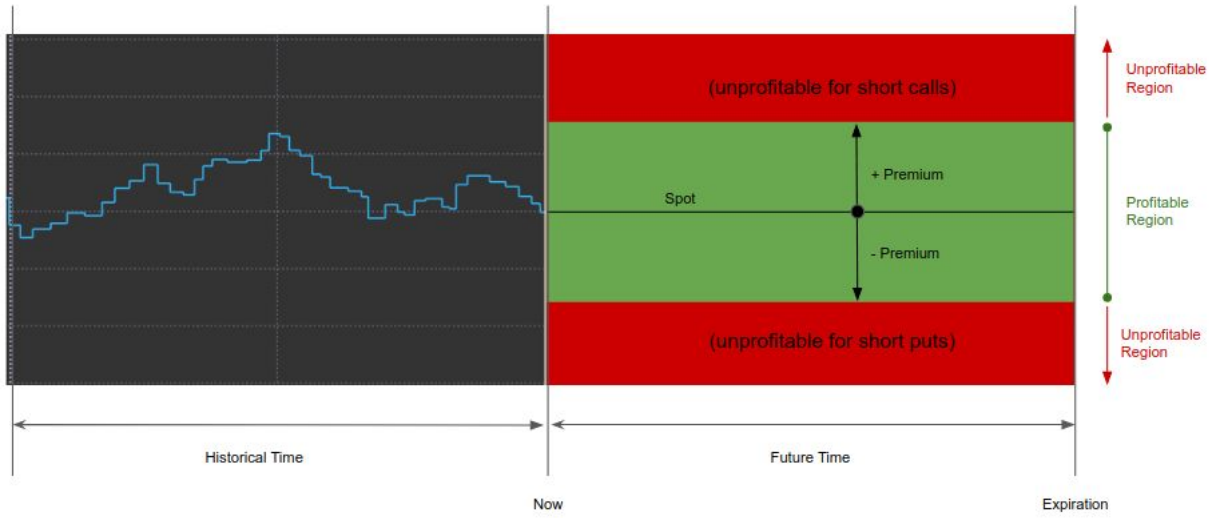
## Market Making to Earn DAI

In order to discuss the details, let's first take a step back and revisit how the Microtick system operates.

### Quote Basics

At its core a Microtick quote is simply a *spot price* and a *premium* that relate to a *fixed time duration*. The spot price is just a number representing any real-world condition such as a price for an asset, index or any other value. The premium is a second number that sets an accuracy range for the spot price, taking into consideration the time duration of the quote. These values work together to create a game theoretic profitable region and two unprofitable regions as shown in the following diagram. The blue line on the left side of the diagram is the consensus price history for the selected time duration and the right hand side of the diagram represents an equal amount of future time.

The quote is also funded by an on-chain token backing that underwrites the quote. This backing is not explicitly shown in the diagram, but is implicit to this discussion. It is the token that backs the quote that both market makers and traders seek to earn more of through trading.




**Microtick quote: basic game theory**

## Trade Basics

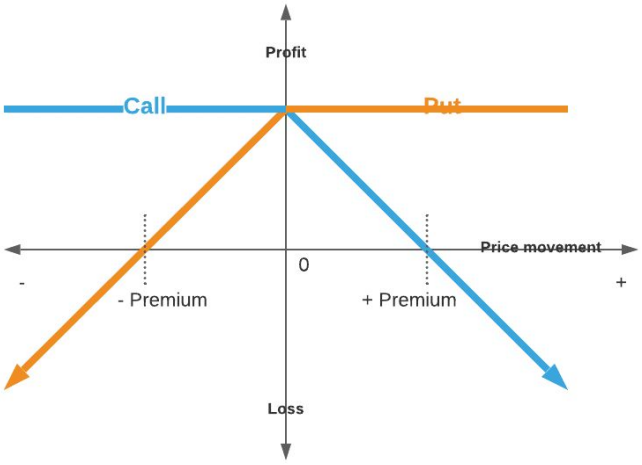
Time does not start ticking until a trade is placed against the quote. To start a trade, a trader pays the premium amount directly to the market maker and chooses a direction: long (increasing spot price, known as a ***call***), or short (decreasing spot price, known as a ***put***). The premium is transferred immediately from the trader's account to the market maker's account, a strike price equal to the current consensus price is assigned to the trade, and an expiration time is set as the current time plus the quote's time duration. Token backing from the quote is then moved into the new trade contract and if the entire amount of backing is moved, the quote is subsequently removed from the order book (and the consensus price is adjusted - more on this mechanism later).

In general, if the blue consensus line ends within the green profitable region at expiration time the market maker will be net profitable. Otherwise, the trader earns a profit on the trade - depending on the trade direction.

By choosing a trade direction at the start, the trader eliminates one of the two unprofitable regions from the diagram based on whether the trade is a call or a put. If the trader goes long (call), the trader will only make a profit for positive price movement. The same in reverse for a

short (put) position. Specifically, the actual unprofitable regions for a market maker are only one of the two regions, depending on the direction the trader chooses. The unprofitable region in the top right in the diagram does not exist for a put trade, and the unprofitable region on the bottom right of the diagram does not exist for a call trade.

In addition, the exact amount of profit for a market maker varies based on the amount of price movement. The precise profit / loss graph for calls (blue) and puts (orange) is shown in the figure below. The premium paid by the trader to the market maker is the break even point for calls as the price moves up and for puts as the price moves down.




]

**Market maker profit based on consensus price movement**

The market maker obviously wants to collect as much premium as possible in order to make a profit. On the other hand, they need to keep premiums reasonable otherwise there will be no traders willing to take the opposing side of the bet. The market maker approaches the premium pricing decision very much like the “house” in a casino - the premium’s fair value represents a probability of price movement; the market maker seeks to earn a profit over time even though specific outcomes may result in the trader earning potentially large payouts along the way.

The market maker can only earn a return if a trader is willing to create a trade. Until that point in time, a quote is just an ongoing offer that is continuously updated with new spot and premium values as market conditions change both on-chain and in the real world. The job of a market maker is to evaluate these real-world and on-chain conditions and make quote changes that

anticipate what other market makers are likely to do, and at the same time to maximize their profitability based on what direction traders are likely to trade.

Quote parameters are summarized in the table below.

| Quote Parameter | Description |
|---|---|
| Duration | Time duration for the quote. Current choices are 5-minute, 15-minute, 1-hour, 4-hour and 12-hour. |
| Spot | Best estimate of the current real-world consensus price. |
| Premium | Fair value for expected price movement over the time duration of the quote, in either the positive or negative direction. |
| Backing | On-chain token amount of value that backs this quote (not explicitly shown). Amount "at-risk" that traders stand to win if the price moves out of the green area into the red. |

## On-chain Consensus Price

In the above description we have discussed a simple market where the quote's spot price is always equal to the market consensus. This can only happen when there is a single quote in the market. In reality, there will almost always be multiple quotes of various time durations from different market makers operating together to create a market.

In Microtick, all quotes across all time durations are combined together into a single weighted average. This is done by assigning a weight to each quote:

$$Weight = \frac{Backing}{Premium}$$

There are mathematical reasons for using this formula having to do with calculating market sensitivity to manipulation that are out of scope for this paper, but intuitively this makes sense:

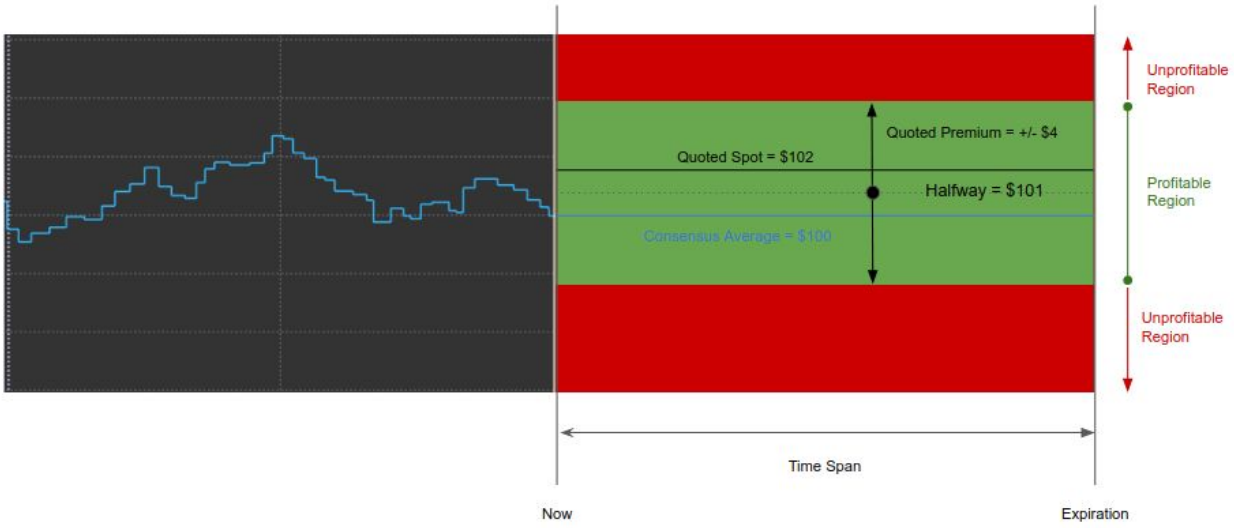
- The more backing a market maker places behind a quote, the more weight it gets in the consensus average.
- The tighter a market maker is willing to make the profitable region, the more confidence a trader should have in the accuracy of the quote.
- Shorter time durations will have smaller premiums in general, which tends to give more weight to quotes with shorter time durations. This happens naturally - there is nothing in the smart contract logic that changes quote weight based on time duration.

Using the calculated weightings, the on-chain consensus price at any given block is the weighted average across all the quotes from all time durations:

$$Consensus = \frac{\sum_{quotes} Weight_q * Spot_q}{\sum_{quotes} Weight_q}$$

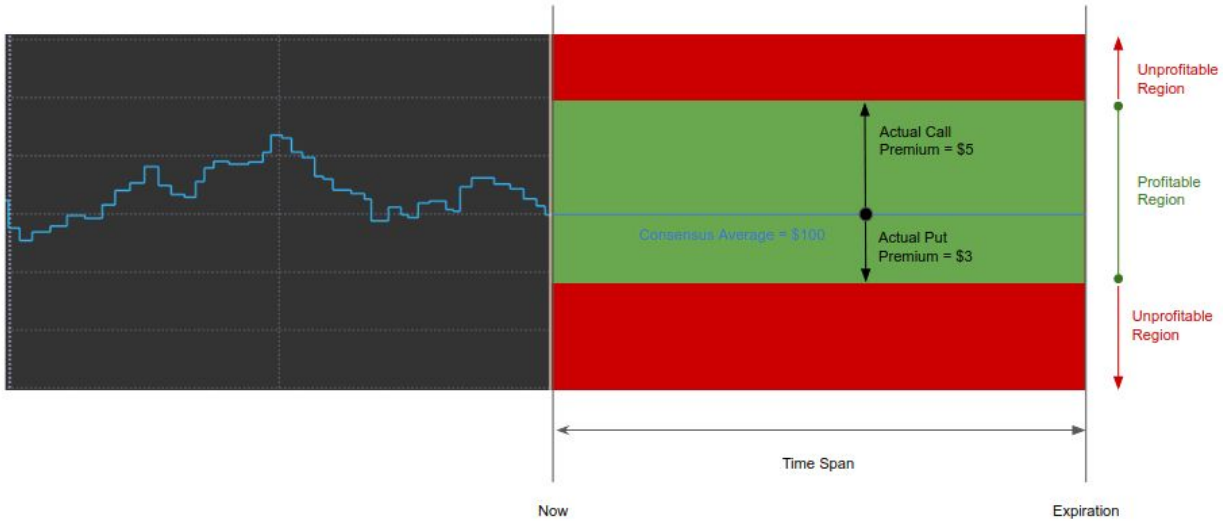
## Dynamic Quote Premiums

The final consideration a market maker must take into account when placing and maintaining quotes is how the profitable region for a quote is adjusted when the market consensus price is different from a quote's spot price. The profitable region is adjusted up or down to the halfway point in-between the quoted spot and the current consensus weighted average. As an example, assume the quoted spot is $102 with a premium of +/- $4, and the consensus is at $100. Note that the +/- premiums are based on the halfway price point, and are only indirectly related to the quoted spot price as the consensus changes. Compare this to the first figure.



**Microtick quote when spot is different from consensus**

Because trades are always initiated with a strike price at the consensus price (not the quoted spot), the on-chain premiums are adjusted dynamically, block by block, to reflect the updated profitable region. This happens automatically through smart contract logic, with no action required from the market maker who placed the quote.



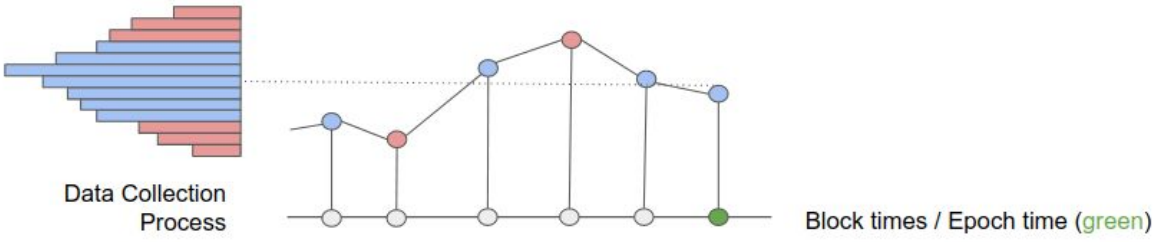

There is mathematical justification for handling the quotes in this manner that has to do with option pricing theory and the linear approximation Microtick makes to fair valuation of at-the-money options, but that topic is out of scope for this paper. For more information see the Microtick website https://www.microtick.com.

# Factors Affecting Premium Pricing

Now that we understand how the marketplace itself operates, let's look into the factors that affect how a market maker might want to adjust their on-chain premium.

## Real-world Time Volatility

There are significant challenges in sampling real-world prices on a blockchain. Most oracles work using epochs (a period of N blocks) where samples are collected and then outliers are slashed. This method is certainly not foolproof - even if a real world sample is collected correctly, if it's sampled during a window of time when the real-world price is at a relative minimum or maximum (due to short-term imbalances in supply and demand) it is still in danger of being slashed, as shown in the figure below.




Microtick solves this by not relying on epoch-based data collection, but by allowing market makers to specify their own accuracy individually. The more accurate the quote as assessed by the market maker, the more weight in the average. If the market maker is too aggressive with their accuracy assessment, the quote becomes the least expensive (in terms of premium) and will be the first traded off the market.

Accurate assessment of real-world volatility is likely to be the most significant factor affecting on-chain premiums, so market makers should strive to develop algorithms that accurately estimate this value - and can respond to changes quickly, for example if a market starts trending or experiences a sudden increase in time volatility.

## Price Variability Due to Sampling and Distribution Error

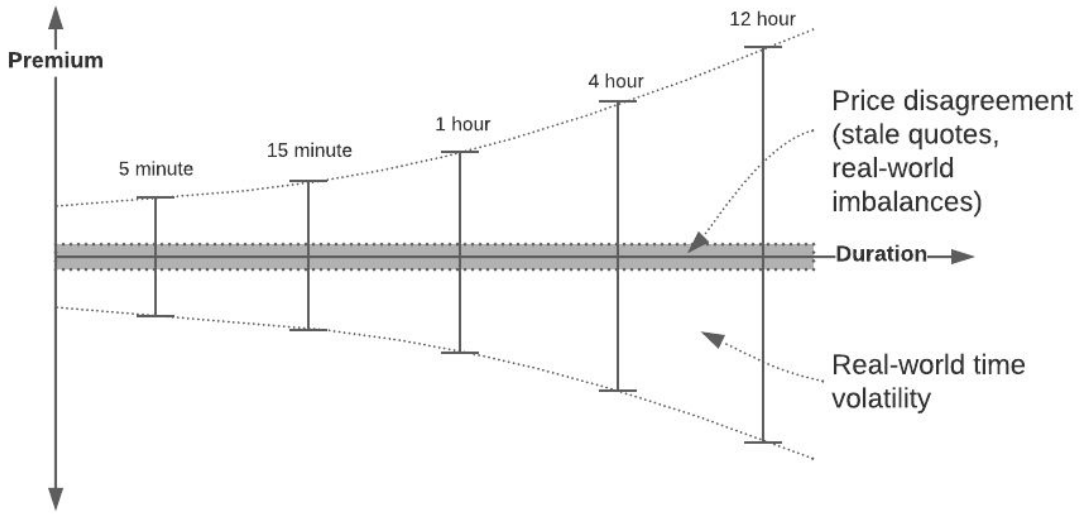
Real world price information is traditionally observed from trading activity. Regional or temporal supply and demand imbalances can affect short-term price information so it's best to observe multiple sources and aggregate the observations when collecting data for placing a quote on Microtick.

Because Microtick is a Schelling point consensus the game theoretic expected behavior of other market participants needs to also be considered. Not all of them may be aggregating multiple price sources and some may use less reliable or stale price information, which will inevitably lead to a distribution of spot prices on the market at any given moment.

In addition, quotes themselves on a Microtick market may become stale over time to varying degrees - and as these are updated the price may change as a result.

The best way for a market maker to account for this error due to price disagreement is to account for it as an additional factor in the premium for the quotes. The time variability

(volatility) can be added in because the uncertainty is additive. This is illustrated in the figure below.




It is expected that the disagreement in price resulting from stale quotes or variability of prices among real-world sources will usually be small with respect to price volatility over time, especially for longer time durations. For smaller time durations it may become a significant factor that market makers need to account for in their pricing algorithms.

## On-chain Weighting

In order to be competitive in premium pricing, market makers need to take into account the on-chain weight distribution across time durations for a market.

Microtick is a weighted average, and the longer the time duration for a quote, the less frequently it will need to be updated. This can lessen on-chain volatility as compared to the real-world (think of a moving average on a stock price chart versus the bars themselves) As a specific example, below is a screenshot of a weight distribution for a market running the various time durations:

| ■ 5 minute | ■ 15 minute | ■ 1 hour | ■ 4 hour | ■ 12 hour |
|---|---|---|---|---|
| ⚖ 0.4185 | ⚖ 0.5972 | ⚖ 0.5807 | ⚖ 0.181 | ⚖ 0.0696 |

Of note for this market is most of the weight is in the 5-minute, 15-minute, and 1-hour time durations. This will typically result in a more responsive price than a market running mostly with longer 4-hour or 12-hour durations.

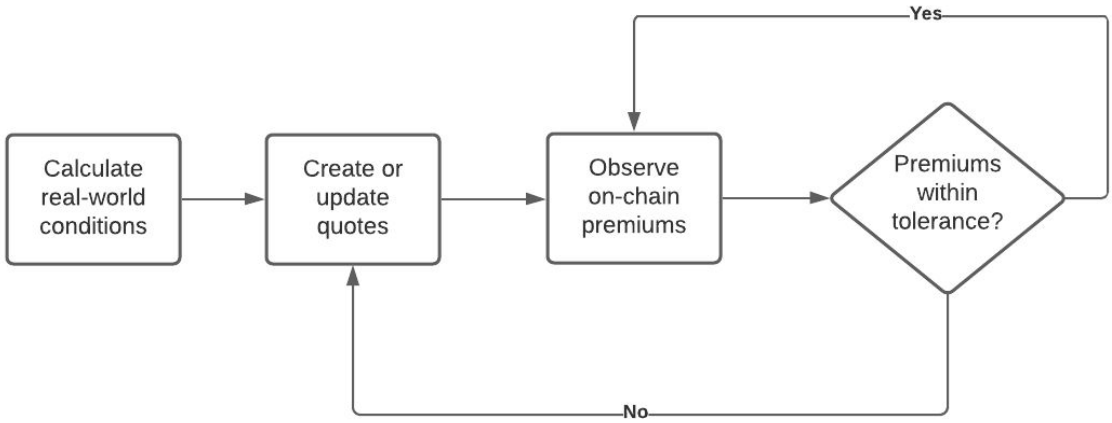
An interesting side note here: due to this averaging effect and the fact that real-world prices use order book price discovery which is inherently less stable over short time frames, Microtick on-chain volatility should always be less than real-world volatility. The Microtick price will tend to lag its real-world counterpart (alternatively leading it when prices mean revert). Because the Microtick price is tradable, this will likely provide a stable index price available for arbitrage if a real-world price runup or selloff happens.

# On-chain Premium Management

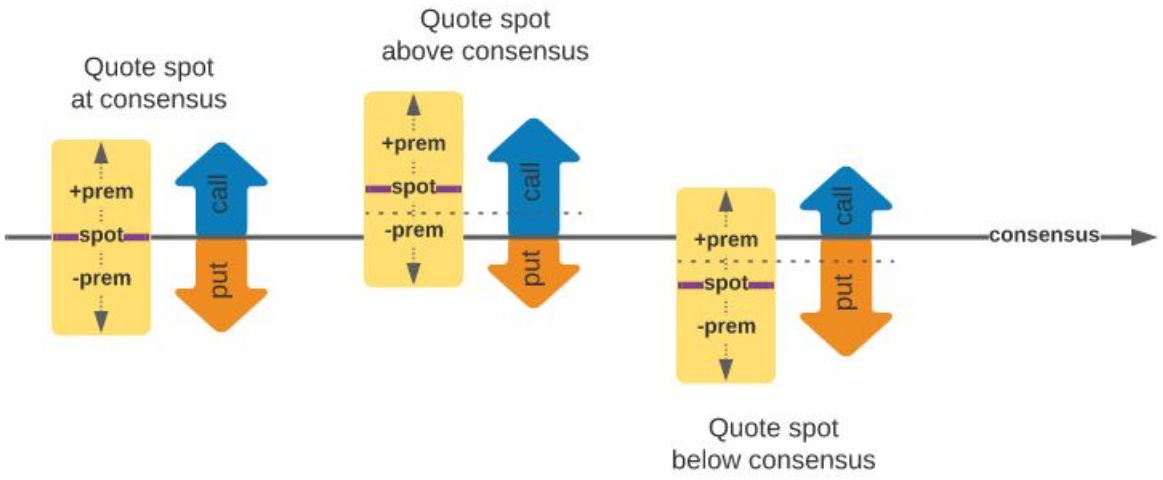
Once the market maker has a picture of real world price distributions, expected real-time volatility and the on-chain weights that indicate the sensitivity of the Microtick consensus to quote updates, we can start to create a model for managing quote premiums on-chain as the consensus price changes.

## Monitoring Loop

The following diagram is a flowchart of how a market maker might operate on a Microtick system. Note that the market maker does not necessarily need to monitor the spot price or the resulting consensus price. As long as the premiums for the market maker's quotes are within tolerances (minimum, maximum) no changes to spot price needs to be made on-chain.



As the consensus price drifts over time with respect to the market maker's quote, the on-chain premiums will automatically adjust. This is the same mechanism discussed above under "Dynamic Quote Premiums", but illustrated differently to show the actual call and put premiums next to the original quote.




The goal is to manage the quote's premium such that the minimum (the price of a put when the quote's spot is above consensus, or the price of a call when the quote's spot is below consensus) doesn't go below the minimum target premium the market maker has predetermined. Once the premium falls below tolerance, the market maker simply re-adjusts the quote by recentering it at the current real-world aggregate spot price.

*There are certain situations where a market maker might recenter a quote with a spot above or below the consensus on purpose*. For example, if the real-world market is trending higher and the market maker's open interest is short mostly calls and no puts, the market maker might opt to place the quote's spot above the consensus to increase the call premium and discount put premium to incentivize traders to buy put premium. *Because price can only move in one direction, a balanced open interest is generally desirable (depending on expiration times of course) because the premium in one of the directions will represent pure profit.*

## Update Frequency

When market conditions are relatively static and the consensus is not changing rapidly the market maker does not need to update quotes as frequently. *Of note: there is a slashing*

*condition for quotes that have not been updated in 2x their time duration*. This means that a 1-hour quote that has not been updated for 2 hours can be removed from the market by anyone, with the slash amount being paid from the quote's backing to the party that issued the cancellation transaction. The current slash rate can be queried from the microtick module's params:

```
$ mtm query microtick params
cancel_slash_rate: "0.010000000000000000"
```

This means for faster order books like 5-minute time durations, market makers should manage risk of lost connectivity or system downtime much more aggressively than for 4-hour or 12-hour markets. From a market perspective, the shorter time durations have more weight in the market so it becomes much more critical for system operation to ensure quotes do not become stale.

### Freeze Time

There is an on-chain "freeze time" enforced by the smart contract logic that prevents quotes from being updated too frequently. This parameter is adjustable through governance. Market makers should be aware of this value and adjust their premiums accordingly (the larger the freeze time, the larger the premium should be in general). This parameter exists to allow traders to take advantage of misplaced or manipulative quotes that are placed by untrustworthy market makers.

```
$ mtm query microtick params
freeze_time: 30
```

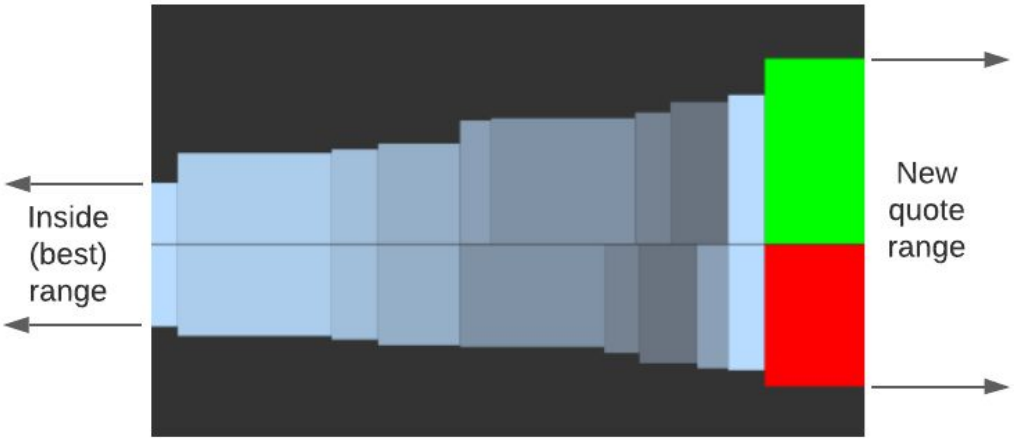
## Market Making to Earn TICK

Microtick's on-chain token is "TICK". Market makers earn TICK as a reward for paying commissions for their quote creation and subsequent updates. The ratio of TICK earned is set in the on-chain parameters relating to quoting:

```
$ mtm query microtick params
commission_quote_percent: "0.000400000000000000"
commission_cancel_percent: "0.000100000000000000"
commission_update_percent: "0.000050000000000000"
freeze_time: 30
mint_ratio: "0.500000000000000000"
```

The commission paid for placing a quote is significantly greater than the commission for updating, so if the goal of the market maker is to earn TICK it makes sense to pay the creation commission, wait for at least “freeze_time” seccnds, then cancel the quote, paying the commissions for quote creation “commission_quote_percent” and cancellation “commission_cancel_percent”. All parameters are on-chain and adjustable through governance.

In order to encourage competition with quotes the full mint reward applies only to the inside premium as illustrated below on the left hand side of the diagram. All other quotes that have premiums greater than the inside range are discounted proportionally to the ratio of the quote’s range to the inside range at the time the quote is placed or updated.




As an example - the new quote’s range on the right is approximately twice the inside range for the order book. As a result the TICK earned by placing this quote will only be about half the maximum rate indicated by the mint_ratio parameter (in this case 0.25 instead of 0.5). This is an incentive for market makers interested in earning TICK to be aggressive in their premiums and even place trades against the inside quotes taking those off the order book.