

Olympus System for Range-Bound Stability Specification

This document codifies the requirements for a system to implement the Range-Bound Stability model defined in the [recent paper](#) by Zeus et al.

Initial spec by Zeus (has changed some since)

System is comprised of three components:

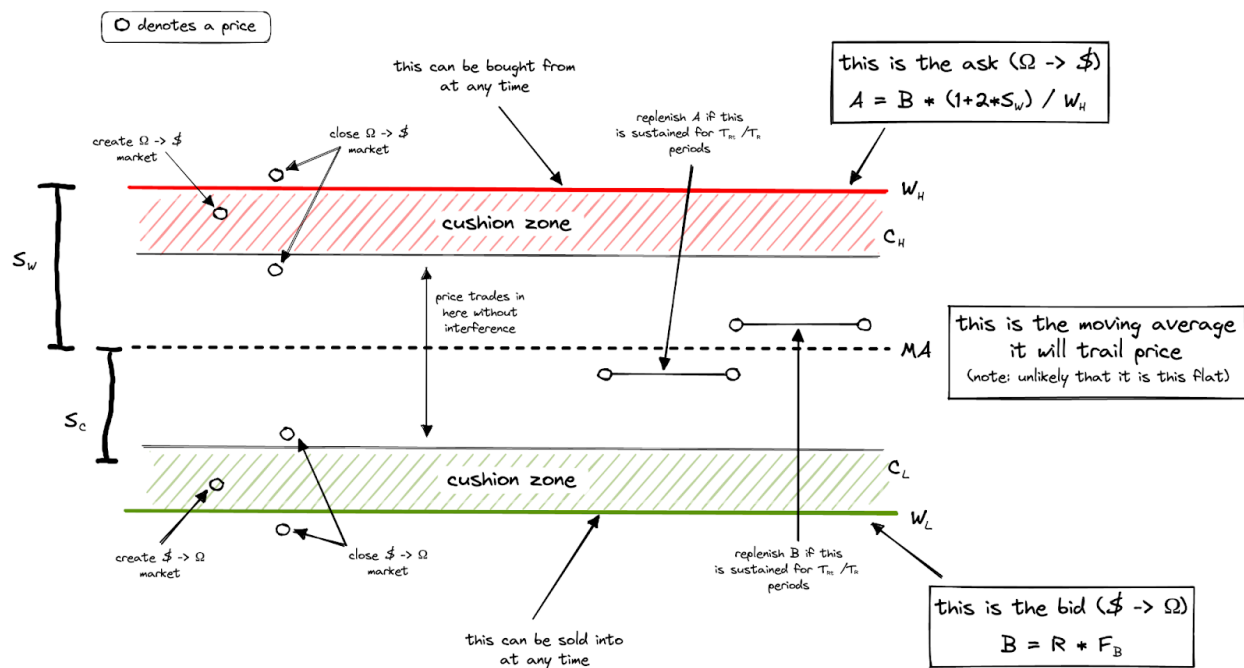
1. the range setter
 - price data (from oracle)
 - moving average period length (from governance)
 - spread (from governance)
 - treasury data (from oracle)
 - depth (from governance, % reserves)
 - regeneration rate (from governance)
2. the treasury balancer
 - asset data (from oracle)
 - liquidity/reserve balance (from governance)
 - offering length/reoffering period length (governance)
 - vesting speed (closest liquid expiration?)
 - bond depo integration
3. the liquidity maintainer
 - pool list (need to verify)

the range setter keeps track of a moving average of price spanning T_{ma} days. it applies a spread Sw to that moving average to determine a range ($MA - Sw$, $MA + Sw$). it calculates the amount to buy B at the range low using treasury T and the depth factor F_b , and the amount to sell at the range high by scaling the bid up by $2x$ radius and dividing by price at top of the range ($B * (1 + (2 * Sw)) / (MA * 1 + Sw)$). it accepts trades at range (low, high) up to depth (low, high). depth is regenerated if price sits on the opposite side of the MA for T_t epochs in a T_r period (ie 18 out of 21 epochs).

the treasury balancer maintains a balance of liquidity and reserve assets. when assets in liquidity change (as a result

of trading activity) or assets in reserves change (as a result of range activity), this ratio is skewed. the balancer creates bond markets to correct this skew (ie underweight reserves, create bond market selling ohm for dai. overweight reserves, create bond market selling dai for ohm).

the liquidity maintainer maintains liquidity by minting into xyk liquidity pools. it keeps a list of verified pools to which it mints tokens at the staking reward rate and calls the sync() function to update the pools virtual balances.



Requirements:

Protocol-enforced Range

1. Calculate and maintain a moving average price for OHM against a specified reserve asset for a configurable duration. The MA price should be updated each system epoch.
2. Calculate lower and upper bounds for the price OHM against a specified reserve asset, both “wall” and “cushion” components, from the moving average price and configurable spread variables.
 - a. $(C_H, C_L) = MA * (1 \pm S_C)$
 - b. $(W_H, W_L) = MA * (1 \pm S_W)$
 - c. *where $S_C \leq S_W$*
3. Allow users to swap OHM for reserves at the lower wall price (W_L) and reserves for OHM at the upper wall price (W_H) up to the specified capacity of the current walls.
 - a. Capacity of the lower wall (bid) in reserves should be the amount of reserves in the Olympus Treasury multiplied by a configured bid factor (percent of reserves to use for each wall). $B_{reserve} = R * F_B$
 - b. Capacity of the upper wall (ask) in OHM should be the capacity of the lower wall (bid) divided by the upper wall price and scaled up by the difference in spread from the lower wall to the upper wall. $A_{OHM} = B \div W_H * (1 + (S_W * 2))$
 - c. When the capacity of either wall is depleted, the system should not allow additional swaps on that side until it is regenerated.
 - d. When the capacity of the upper wall (ask) is depleted, the system should not reinstate a new wall with additional capacity until the current price is observed to be below the MA price for a X out of the last Y system epochs, where X and Y are configured parameters representing the regeneration threshold (X) and total number of observations (Y). Additionally, the wall should not regenerate until a minimum configured amount of time has past.
4. Deploy a bond market to sell OHM for reserves when the current price of OHM against the reserve asset is greater than or equal to the upper cushion price at the system epoch with a capacity equal to the configured percentage of the upper wall capacity to use for the cushion.
 - a. If a bond market is active on a system epoch, the system should close the market if the current price of OHM against the reserve is back below the upper cushion price or has exceeded the upper wall price.
 - b. The bond market should be closed if the upper wall capacity is depleted.
 - c. The bond market should be instant-swap with no vesting.
 - d. The bond market should start at the upper wall price and have a minimum price of the upper cushion price.

5. Deploy a bond market to buy OHM with reserves when the current price of OHM against the reserve asset is less than or equal to the lower cushion price at the system epoch with a capacity equal to the configured percentage of the lower wall capacity to use for the cushion.
 - a. If a bond market is active on a system epoch, the system should close the market if the current price of OHM against the reserve is back above the lower cushion price or has gone below the lower wall price.
 - b. The bond market should be closed if the lower wall capacity is depleted.
 - c. The bond market should be instant-swap with no vesting.
 - d. The bond market should start at the lower wall price and have a minimum price of the lower cushion price.

Treasury Rebalancing

1. Maintain a configured target percent of the treasury's reserve assets in the specified OHM-reserve liquidity pool for the reserve.
 - a. The system should maintain this balance through the use of time-weighted automated market maker (TWAMM) orders that are executed over a configured amount of time and interval.
 - b. The system should check the current percentage of treasury-owned reserves in the liquidity pool each system epoch and input TWAMM orders to correct imbalances that exceed a configured threshold. If an order is active on a system epoch, it should determine whether to close or continue the order before issuing a new one.

User-provided Range (Wall of Ohmies)

1. Allows users to stake OHM, the reserve asset, or both to provide funds for a second wall at the cushion prices.
 - a. The stake should be locked on deposit.
 - b. Users should be able to unlock their stake which starts a configurable cooldown period. Once the cooldown period expires, they can withdraw their stake.
2. Maintain a target ratio of value between the two assets in the pool to provide sufficient capital for both walls to support the range.
 - a. The system should incentivize users to balance the ratio to the target by incentivizing them to provide the mix of tokens that moves the system closest to the target ratio.
 - b. The target ratio should adjust based on the movement of price inside the range to avoid staker dilution from new staker incentives on price movements within the range.
3. Allow users to withdraw their stake and receive a system defined mix of OHM and the reserve asset according to what will best help it correct to the target ratio. The system

defined mix of assets prevents manipulation of incentives and circular incentive calculations on withdrawals.

4. Pay a protocol defined token as a reward for staking.
 - a. Allow the protocol to configure the rewards issued per day.
 - b. Only users with locked stakes should receive

