

# 1 Staking Dashboard Issue

The staking dashboard requires the user to simultaneously account for multiple factors. This will likely make the user's brain hurt. We want the user to think a little bit to keep things interesting, but the current level of complexity is excessive.

## 1.1 Problem 1: MM fees may vary from epoch to epoch.

All else equal, delegators will prefer to stake an MM that generates consistent fee revenue every epoch. To evaluate this factor, delegators must open a separate tab and look at a graph of fees generated vs. time. Seems hard. Everything you need to consider should be available on one tab.

## 1.2 Problem 2: MM may choose different 'operator shares'

All else equal, delegators will prefer to stake an MM that chooses a lower 'operator share.' This becomes confusing because we are also presenting a 'staking percentage' and the two statistics are likely to point the user in different directions.

## 1.3 Problem 3: MM may choose different 'staking percentages'

All else equal, delegators will prefer to stake an MM that has a lower current 'staking percentage.' This becomes confusing because we are also presenting an 'operator share' and again the two statistics are likely to point the user in different directions.

# 2 Minor Proposed Improvements

## 2.1 Improvement 1: Report the median fees the MM generated over the past 6 epochs.

This statistic could replace the fees generated in current epoch stat that is currently on the dash (or we could report both).

## 2.2 Improvement 2: Hide small/inactive MM pools from the user.

When listing MM pools, I suggest we default to displaying only the top 5 MM pools ranked according to median fees generated over the past 6 epochs. These will probably be the only pools worth staking. We can include a show more button at the bottom, allowing an interested user to see more options.

### 3 Main Proposed Improvement

#### 3.1 Report an 'adjusted staking percentage' that accounts for effects of the 'operator share' and the 'staking percentage' simultaneously

A user could then just pick whichever of the top 5 MMs has the lowest 'adjusted staking percentage.' We could still report the 'operator share' and 'the staking percentage,' but a user would no longer need to look at these stats to decide whom to stake.

In Equation 1, I show a proposed formula that maps the statistics currently displayed on the dashboard: the 'operator share,'  $\theta_i$ , and the 'staking percentage,'  $s_i$  into an 'adjusted staking percentage,'  $\tilde{s}_i$ .

$$\tilde{s}_i = \left( \frac{1 - \alpha}{1 - \theta_i} \right)^{\frac{1}{\alpha}} s_i \quad (1)$$

To identify the pool with the highest rate of return per staked ZRX, all the user would need to do is pick the pool with the lowest value of  $\tilde{s}_i$ . Furthermore, when the system is in equilibrium (i.e. all delegators and MMs are making approximately optimal choices), we should observe that  $\tilde{s}_i \approx s_i \approx 1$ . Thus, the adjusted staking percentage can be interpreted in approximately the same way as the original 'staking percentage' statistic.

### 4 Staking Math To Obtain Equation 1

#### 4.1 Summary

Our first objective is to do the math for ZRX holders, so that they just see an 'adjusted staking percentage' and will know that the pool with the lowest 'adjusted staking percentage' offers the best return.

The ZRX holder would need to look at the MM  $i$ 's fees generated,  $t_i$ , the MMs operator share,  $\theta_i$ , and the MMs current stake,  $z_i$ , to figure out which pool offers the best return. We are going to map  $(t_i, \theta_i, z_i)$  into an 'adjusted staking percentage' that is inversely related to the pool's expected return.

A second objective is to have the 'adjusted staking percentages' hover around 100% when the system is in a healthy state. To achieve that we need to forecast what the operator shares,  $\theta_i$ , are likely to be and normalize our 'adjusted staking percentage' using this expected value of  $\theta_i$ .

The MMs will want to select an operator share,  $\theta_i$ , that maximize their expected income. If they get greedy and set  $\theta_i$  too high, then they will not attract enough ZRX and the pool will not generate enough revenue. If they are too generous and set  $\theta_i$  too low, then the pool will attract lots of ZRX and generate more revenue, but the MMs slice of the fee pie will be too small. There is thus an optimal value of  $\theta_i$  that we would expect MMs to converge upon.

## 4.2 Delegator's Choice of Pool

I will assume that the delegator observes the fees that will be generated by pool  $i$ ,  $t_i$ , the stake currently allocated to pool  $i$ ,  $z_i$ , and the operator share of pool  $i$ ,  $\theta_i$ . This info is sufficient to allow the delegator to identify the pool with the highest rate of return on staked ZRX.

The total delegator payout of pool  $i$ ,  $r_{i,d}$ , is shown in Equation 2, where in addition to the variables previously defined,  $\hat{\tau}$  is the total pot of rewards to be distributed,  $\hat{t}$  is the total fee earnings of all pools, and  $\hat{z}$  is the total ZRX stake of all pools.

$$r_{i,d} = \frac{\hat{\tau}}{\hat{t}^\alpha \hat{z}^{1-\alpha}} (1 - \theta_i) (t_i)^\alpha (z_i)^{1-\alpha} \quad (2)$$

We can rearrange Equation 2 to obtain an expression for pool  $i$ 's delegator payout per staked ZRX as shown in Equation 3.

$$\frac{r_{i,d}}{z_i} = \frac{\hat{\tau}}{\hat{t}^\alpha \hat{z}^{1-\alpha}} (1 - \theta_i) \left( \frac{t_i}{z_i} \right)^\alpha = \frac{\hat{\tau}}{\hat{z}} (1 - \theta_i) \left( \frac{\frac{z_i}{t_i}}{\frac{\hat{z}}{\hat{t}}} \right)^{-\alpha} \quad (3)$$

We can define  $s_i = \frac{\frac{z_i}{t_i}}{\frac{\hat{z}}{\hat{t}}}$ , and rewrite Equation 3 as shown in Equation 4.

$$\frac{r_{i,d}}{z_i} = \frac{\hat{\tau}}{\hat{z}} \left( \frac{1 - \theta_i}{s_i^\alpha} \right) \quad (4)$$

Note, that  $s_i$  is our current definition of a pool's 'staking percentage' of pool  $i$ . Furthermore, note that, holding  $\theta_i$  fixed,  $\frac{r_{i,d}}{z_i}$  is strictly decreasing in  $s_i$  for all  $\alpha > 0$ . This means that, if all pool's adopt the same operator share, then the delegator would want to pick whichever pool has the lowest staking percentage.

But what if both  $\theta_i$  and  $s_i$  vary across pools? To simplify this case, we could rewrite Equation 4 in terms of an adjusted staking percentage,  $\tilde{s}_i = \left( \frac{1-c}{1-\theta_i} \right)^{\frac{1}{\alpha}} s_i$  and a normalizing constant,  $c \in (-\infty, 1)$ , as shown in Equation 5.

$$\frac{r_{i,d}}{z_i} = \frac{\hat{\tau}}{\hat{z}} \left( \frac{1}{1-c} \right)^{\frac{1}{\alpha}} \left( \frac{1}{\tilde{s}_i^\alpha} \right) \quad (5)$$

Note that now, even if both  $\theta_i$  and  $s_i$  vary simultaneously, the delegator would always want to pick the pool with the lowest adjusted staking percentage,  $\tilde{s}_i$ . Note also that we can pick any value of  $c \in (-\infty, 1)$  we want and still preserve this property. However, ideally we would want to set  $c$  close to the operator's optimal choice of  $\theta_i$ , so that when the system is functioning as expected (i.e. in equilibrium),  $\frac{1-c}{1-\theta_i} \approx 1$  and  $\tilde{s}_i \approx s_i \approx 1$ .

## 4.3 Operator's Choice of $\theta_i$

In equilibrium, delegators are expected to equalize the returns to ZRX stake across pools, so that for all  $i$ ,  $\frac{r_{i,d}}{z_i} = \frac{\hat{\tau}}{\hat{t}^\alpha \hat{z}^{1-\alpha}} (1 - \theta_i) \left( \frac{t_i}{z_i} \right)^\alpha = \hat{k}$  where  $\hat{k}$  is a

constant rate of reward per ZRX staked. If we arrange this expression as shown in Equation 6, we can obtain a formula that maps the pool operator attributes  $\theta_i$  and  $t_i$  into an implied amount of staked ZRX,  $z_i$ , that the pool will receive.

$$z_i(\theta_i, t_i) = \left( \frac{\hat{\tau}}{\hat{t}^\alpha \hat{z}^{1-\alpha} \hat{k}} \right)^{\frac{1}{\alpha}} (1 - \theta_i)^{\frac{1}{\alpha}} t_i \quad (6)$$

An expression for the operator's reward,  $r_{i,mm}$ , is shown in Equation 7.

$$r_{i,d} = \frac{\hat{\tau}}{\hat{t}^\alpha \hat{z}^{1-\alpha}} \theta_i t_i^\alpha z_i^{1-\alpha} \quad (7)$$

If we substitute Equation 6 into Equation 7, we obtain the expression for the operator's reward as a function of his attributes  $\theta_i$  and  $t_i$  as shown in Equation 8.

$$r_{i,d}(\theta_i, t_i) = \left( \frac{\hat{\tau}}{\hat{t}^\alpha \hat{z}^{1-\alpha} \hat{k}} \right)^{\frac{1}{\alpha}} (1 - \theta_i)^{\frac{1-\alpha}{\alpha}} \theta_i t_i \quad (8)$$

The operator is then going to choose a value of  $\theta_i$  to maximize his expected rewards. In this calculation, we will assume that the operator is too small to affect attributes of the entire network, so that  $\left( \frac{\hat{\tau}}{\hat{t}^\alpha \hat{z}^{1-\alpha} \hat{k}} \right)^{\frac{1}{\alpha}}$  is constant. Furthermore, we will assume that the reward payments are too small to meaningfully alter the operator's trading fee generation, so that  $t_i$  can be treated as constant as well.<sup>1</sup> Under these assumptions, maximizing Equation 8 is equivalent to maximizing the concave function  $(1 - \theta_i)^{\frac{1-\alpha}{\alpha}} \theta_i$ . To do this we just take the first derivative and set it equal to 0 to obtain the expression for the optimal operator share shown in Equation 9.

$$\theta^* = \alpha \quad (9)$$

By setting the normalizing constant  $c$  equal to  $\theta^* = \alpha$ , we obtain the suggested formula for the adjusted staking percentage shown in Equation 1.

---

<sup>1</sup>Neither of these assumptions are 100% accurate. However, these types of effects are going to be negligible, so that ignoring them won't affect the answer.