Whitepaper: DeepBook Token

September 5, 2024

1 Introduction

The DeepBook token (DEEP) is designed to make the DeepBook liquidity platform on Sui **the preeminent venue for wholesale liquidity**, between and amongst professional traders and DeFi protocols. DeepBook is a special example of a decentralized, permissionless, and *on-chain* central limit orderbook, made possible by Sui's fast execution paths and low gas fees. Central limit orderbooks are highly efficient for active liquidity providers, and so DeepBook as a protocol is intrinsically well-suited to allocating liquidity amongst professional participants. The goal of the DeepBook token, thus, is to enable and further that goal. This positions the DeepBook system as a public good: key financial infrastructure that is inexpensive for users, benefits ecosystem-wide liquidity, and supports the rich consumer-facing DeFi platform.

Equilibrium Dynamics

The DeepBook token model relies on four core flows:

1. **Volume-Based Taker Fees**: Fees to take liquidity in a DeepBook pool start at moderate levels and decline to low levels with respect to a trader's volume taken in that pool over an epoch. This means that participants who actively participate in swapping liquidity pay low fees on the margin (in contrast to those who only occasionally participate).

2. **Maker Incentives**: Liquidity providers on DeepBook ("makers") receive incentives, which also start at elevated levels in a pool and then decrease with respect to the amount of total liquidity provided by other makers over the epoch. In other words, makers are financially encouraged to provide incremental liquidity when liquidity provision is low. Those incentives then phase out as the total liquidity provision turns healthy. This mechanism allows a DeepBook pool to maintain good liquidity in all periods, by programmatically turning incentives on and off as needed.

3. **Stake-Based Participation**: In order to participate in these two programs (volume-based fees and incentives), traders have to stake a minimum number of DEEP tokens on the pool in advance and for the duration of the epoch. These stakes earn no explicit rewards. Any trader that does not stake this minimum number is not eligible for either program: they will pay the full fee to take liquidity (regardless of amount of traded) and they will not receive maker

incentives from the pool. This staking is critical for ensuring alignment between the pool's core participants and DeepBook, both in terms of direct governance and indirect stewardship.

4. **Stake-Based Governance**: Separately, stakers in a given pool will govern that pool. Specifically, parameters around fees and staking requirements for a given pool will be set by stakers in that pool, with governance rights that increase with stake weights. There are no minimum stake requirements to participate in governance.

Vulnerabilities and Defenses

At first glance, this token model has two vulnerabilities.

1. **Wash Trading**: Since incentives are paid to active liquidity makers, traders may place inorganic volume in an attempt to earn more of these incentives.

2. **Governance Capture**: Through governance channels, large stakers in the pool may try to extract surplus from small stakers and non-stakers in setting parameters, to the detriment of the pool's long-term success.

The original token model is augmented with four additional components to mitigate these vulnerabilities:

1. Against wash trading, there are two defenses:

   - To protect DeepBook against wash trading holistically, tokens collected in an epoch by a given pool can never be less than tokens distributed, with the residual tokens being burned as a form of credible commitment.
   - To disincentivize wash trading for individual participants, once some minimal liquidity threshold has been reached, the incentives given to a maker for a given trade will be less than the fees collected on that same trade.

2. Against governance capture, there are also two defenses:

   - Pool governance can only adjust two fee-related parameters, and those adjustments are subject to restrictions. This prevents a cabal of traders from exercising monopolistic pricing on other participants in the pool.
   - Governance rights – while increasing in stake – are increasing at sub-linear rates (square root) past some threshold. This will allow for greater representation to small entities, to counterbalance large institutional traders. This mechanism is uniquely resistant to the traditional vulnerability of Sybil wallet creation, because of minimum staking requirements needed to earn discounted fees and incentives.

Token Allocation

To achieve the equilibrium model, the DEEP token needs to be distributed across the ecosystem, and to participants who are aligned with the success of the protocol and Sui more generally. As such, the token will be distributed across four buckets: exactly 10% for the community in an initial airdrop, approximately 28% for core contributors and early backers, and approximately 62% for future grants, community programs, and community initiatives. This allocation thus gives over two-thirds of the token supply to the broader community.

Paper Roadmap

The whitepaper proceeds as follows. Section 2 discusses the flow of tokens during trading, including a deeper dive into how fees and incentives are set. Section 3 outlines the governance of pools. Section 4 discusses a set of other operational considerations, e.g. price conversions and timelines. Section 5 lays out the allocation of DEEP tokens. Finally, Section 6 concludes.

2 Token Flow

Tokens have the following flow during trading:

- Prior to an epoch and for a given pool, wallets choose how many DEEP tokens to stake (if any) with that pool.
- During the epoch, takers and makers pay trade fees for every trade that occurs in that pool.
  - Taker fees are constant for wallets that do not stake a predetermined number of DEEP tokens in a given pool.
  - For wallets that do stake this minimum number, taker fees start at the same level. Once the account hits some minimum volume, the fees fall by half.
  - Maker fees are constant for wallets, regardless of the number of tokens staked or the amount of volume made.
- Once the epoch concludes, these trade fees are potentially rebated back to makers as incentives.
  - Makers are only eligible for these incentives if they have staked the same predetermined number of DeepBook tokens on a pool.
  - The residual fees are burned for economic security reasons.
- Stakes can be rolled over to the next epoch, or they can be released back to the original wallets.

The next sections provide more detail into the three critical components in this system: fees, maker incentives, and residual collected fees.

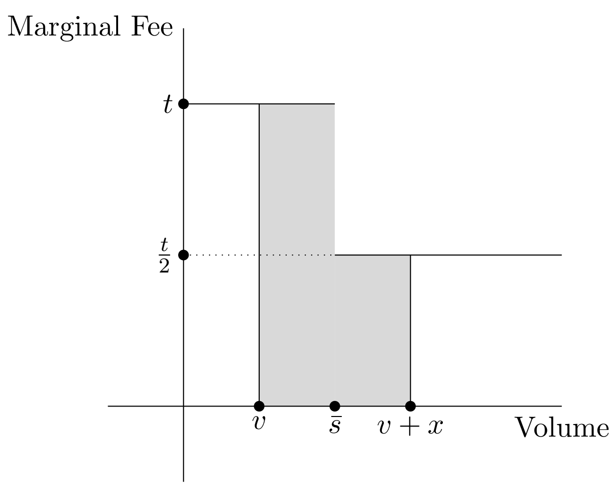
2.1 Fees

The fee schedule for takers and makers can be described most easily through specifying the *marginal* fees paid by each, i.e. the fee levied on a per-unit basis for a given trade.

First, consider a liquidity taker. There are two parameters that govern this wallet's fees: whether they have staked a sufficient number of DeepBook tokens prior to the epoch for this particular pool, i.e. whether their direct stake s exceeds the pool-level threshold $\bar{s}$, and the baseline taker fee t. These parameters jointly describe a linear schedule for the marginal fee against total volume in an epoch, where the marginal fee for a trade is t for non-stakers (or, more accurately, those who stake insufficiently) and either t or $t/2$ for stakers, depending on whether the stakers have traded at least $\bar{s}$ in the pool. It is important to note that $\bar{s}$ thus plays two roles: it is both the minimum value of DEEP tokens that a staker must stake to qualify for volume discounts, and it is the trading volume that a staker must cross in an epoch to earn discounted fees.

This visually manifests as Figure 1. In Figure 1, the area under the marginal fee curve is the total fee paid by a liquidity taker for an example trade. For notation, assume the wallet has previously traded in this epoch volume v for this particular pool as a taker, is placing a trade of size x, and has sufficient stake to qualify for discounted fees.

Figure 1: Marginal Taker Fee



The figure depicts the taker fee schedule for a hypothetical scenario. The marginal fee on trading volume starts at t, until trading volume reaches $\bar{s}$, at which point it falls by 50%. For a trade of size x, which moves a taker from volume v for that epoch to volume $v + x$, the taker pays the summation of the marginal fee over the trade size, i.e. the area under the curve.

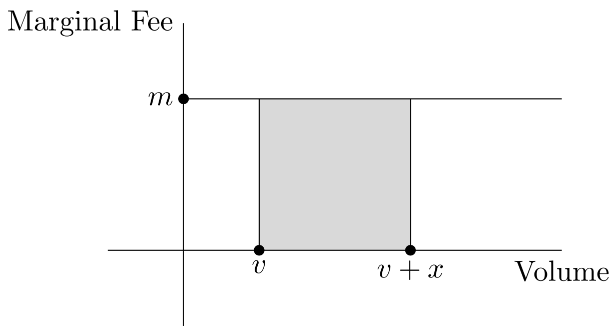
The total trade fee paid by a taker, given the parameters determining the marginal fee and the decision variables, can be formalized in Equation (1).

$$\text{Taker Fee} = \begin{cases} tx & \text{when } s < \bar{s} \\ tx & \text{when } s \geq \bar{s} \text{ and } v + x \leq \bar{s} \\ \frac{tx}{2} & \text{when } s \geq \bar{s} \text{ and } v \geq \bar{s} \\ \frac{t}{2}(\bar{s} + x - v) & \text{otherwise} \end{cases} \tag{1}$$

Second, consider a liquidity maker. These fees are simple, as they do not depend on stake or volume (unlike maker *incentives*). Indeed, the only parameter determining the marginal fee is m, which is the marginal per-unit fee. For completeness, Figure 2 visualizes an example and Equation (2) formalizes the total trade fee paid by a maker on a given trade.

$$\text{Maker Fee} = mx \tag{2}$$

Figure 2: Marginal Maker Fee



The figure depicts the maker fee schedule. The marginal fee on trading volume is constantly m. For an illustrative given trade size of x, which moves a maker from volume v for that epoch to volume $v + x$, the maker pays the summation of the marginal fee over the trade size, i.e. the area under the curve.

2.2 Maker Incentives

The calculation of maker incentives is more complex, as it balances four principles. First, the incentives should be structured in a way that protects the solvency of the protocol. Second, the incentives should be designed to be countercyclical, i.e. largest when total liquidity is lowest. Third, the incentives should not encourage wash trading or other forms of inorganic activity by DeepBook makers. Fourth and finally, the schedule of incentives should never accidentally disincentivize liquidity provision by makers.[1]

The solution for the DeepBook protocol respects all four principles. At a high level, it issues incentives to a given maker that depend on *other* makers' aggregate liquidity provided. This allows for two core features. First, the mechanism provides countercyclical incentives: when other makers

[1]This framework rules out many simple designs. As one example, incentives that phase out with the aggregate liquidity provided violate the fourth principle, as a maker that provides liquidity is impairing its own incentives.

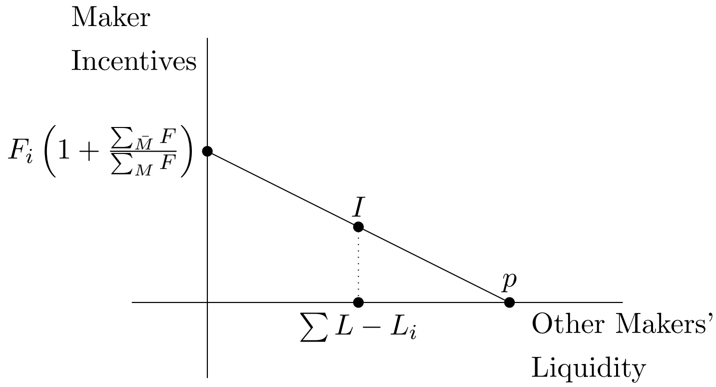
provide little liquidity, a given maker earns high incentives on a per-unit basis; and when other makers provide high liquidity, that maker earns low incentives. Second, the mechanism does not dissuade that particular maker from providing liquidity, because its decisions do not impair its own incentives (and only impair *other* makers' incentives). In addition, the mechanism is parameterized to keep pools solvent and dissuade wash trading.

The computation of incentives – which happens after an epoch ends and is only given to makers who have staked the required number of DEEP tokens in advance – is calculated in Equation (3) for a given maker i. Equation (3) introduces several new variables. First, M refers to the set of makers who stake a sufficient number of DEEP tokens, and $\bar{M}$ refers to the set of makers who do not fulfill this condition. Second, F refers to total fees (collected both from takers and the maker) that a maker's volume has generated in a given epoch. Third, L refers to the total liquidity provided by a maker – and specifically the liquidity traded, not just the liquidity quoted. Finally, the critical point p is the "phaseout" point, at which – if total liquidity provided by other makers' crosses this point – incentives are zero for the maker in that epoch. This point p is constant for all makers in a pool and epoch.

$$\text{Incentives for Maker } i = \max\left[F_i\left(1+\frac{\sum_{j\in\bar{M}}F_j}{\sum_{j\in M}F_j}\right)\left(1-\frac{\sum_{j\in M\cup\bar{M}}L_j - L_i}{p}\right), 0\right] \tag{3}$$

Equation (3) can be visualized in Figure 3 for maker i. In this figure, the x-axis measures liquidity provided by other makers (both stakers and non-stakers), and the y-axis measures total incentives available to the maker. Equation (3) is a line on this graph anchored by two points. The first point is $(p, 0)$, i.e. at the point at which other makers have provided sufficient liquidity so as to phase out all incentives, the incentives are indeed zero for the maker in question. The second point is $(0, f)$, where f refers to the total two-sided trading fees generated by the maker, "grossed up" to include a pro-rata share of fees generated by non-staking makers. This point's interpretation is the following: when other makers have provided no liquidity, all of the fees generated by a maker (plus bonus fees from non-staking makers) are paid to that maker as incentives.

Figure 3: Maker Incentives



The figure depicts the incentives for a maker in an epoch, as a function of liquidity provided by other makers and fees generated by all makers. The schedule determining this incentive is a simply a line, anchored by two point: the phase-out point p on the x-axis, at which paid incentives are zero, and the fees generated by the maker plus a pro-rata share of fees generated by non-staking makers on the y-axis.

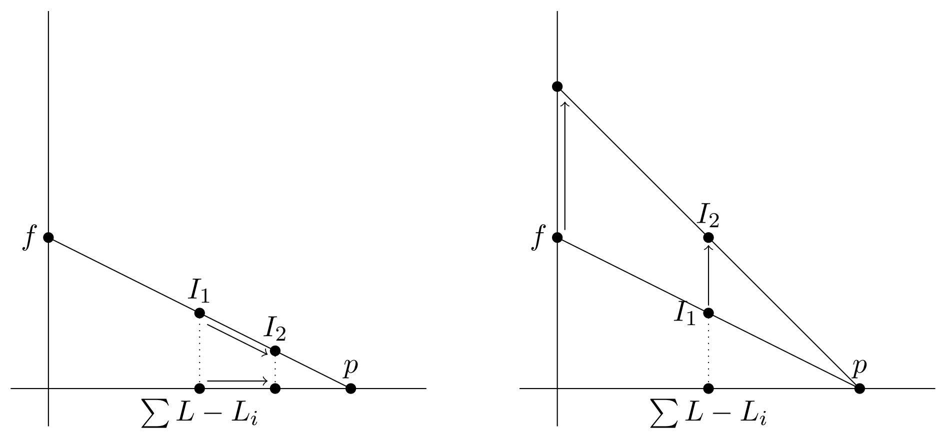
The framework in Figure 3 allows us to visualize two types of changes, demonstrated in Figure 4: improvements in the liquidity of other makers, and improvements in the liquidity of the maker. In Figure 4a, other makers provide more liquidity, revising the incentives available to the maker down from I_1 to I_2.[2] This demonstrates the principle that incentives are more generous when aggregate liquidity is low and less generous when aggregate liquidity is high. In Figure 4b, the maker provides more liquidity, raising incentives from I_1 to I_2.[3] Because there is no change in other makers' liquidity, incentives only go up, demonstrating the principle that the mechanism does not dissuade marginal liquidity provision. Operationally, this increase is driven by a rise in the maker's total trading fees, which raises the y-intercept of the incentives schedule. Finally, with respect to Figure 4b, the marginal change in incentives shrinks towards zero as other makers provide more liquidity, i.e. the gap between I_2 and I_1 declines as other liquidity approaches the critical point p. This reinforces both principles: marginal incentives get less generous as aggregate liquidity increases, but there is never an explicit disincentive for the maker to provide more liquidity.

[2] This makes the assumption that the ratio of fees collected from non-staking makers' liquidity to fees collected from staking makers' liquidity remains constant, i.e. that other makers who provide liquidity do so in the same proportion across stakers and non-stakers, which keeps point $(0, f)$ constant. If this does not hold, then a new line is drawn and I_2 could be higher or lower than I_1. Figure 8 in the Appendix demonstrates this accordingly.

[3] The explanation in Figure 4b abstracts away from the change in the ratio of fees earned from non-stakers' liquidity to stakers' liquidity, which will slightly dampen any rise in F_i, but not neutralize it except in degenerate cases.

Figure 4: Changes in Maker Incentives

(a) Scenario: Others provide more liquidity (b) Scenario: Maker provides more liquidity



The figures depict how incentives for a maker change under two scenarios. In the first scenario, other makers provide more liquidity, moving the maker's incentives along the existing schedule to the right (from I_1 to I_2) and thus lowering them. In the second scenario, the maker provides more liquidity and earns more fees, generating a new schedule of incentives. This moves the maker's incentives up from I_1 to I_2, which corresponds to an increase in incentives.

Finally, consider the other two principles: ensuring solvency in the pool, and defending against wash trading. Solvency can be taken by summing up the payout to each maker from Equation 3, where $b_i \in [0, 1]$ reflects the second parenthetical term, which adjusts payouts by the liquidity of other makers relative to the phaseout point p. Equation (4) compares this summation, on the lefthand side, to the total fees collected, on the righthand side. The first term on the left is weakly smaller than the first term on the right, as $b_i \leq 1$; and for the same reason, the second term on the left is weakly smaller than its counterpart. This ensures solvency, as the aggregate incentives paid out are always weakly less than the aggregate fees collected by the pool over an epoch.

$$\begin{gathered}\sum_{i \in M} \left[F_i \left(1 + \frac{\sum_{j \in \bar{M}} F_j}{\sum_{j \in M} F_j} \right) b_i \right] \leq \\ \left(1 + \frac{\sum_{j \in \bar{M}} F_j}{\sum_{j \in M} F_j} \right) \sum_{i \in M} (F_i b_i) \leq \\ \sum_{i \in M} F_i b_i + \sum_{i \in \bar{M}} F_i \left(\frac{\sum_{j \in M} F_j b_j}{\sum_{j \in M} F_j} \right) \leq \sum_{i \in M} F_i + \sum_{i \in \bar{M}} F_i\end{gathered} \tag{4}$$

The defense against wash trading is imperfect, and can best be understood in terms of a maker's marginal propensity to wash trade a unit of the underlying pool. On this trade, the maker pays both maker and taker fees per Equations (1) and (2), while collecting higher incentives per Equation (3). This yields the marginal payoff in Equation (5), under the simplifying assumption that changes to the ratio of aggregate non-staking maker fees to aggregate staking maker fees are second-order.

$$\pi = (m + x) \left[\left(1 + \frac{\sum_{j \in \bar{M}} F_j}{\sum_{j \in M} F_j} \right) \left(1 - \frac{\sum_{j \in M \cup \bar{M}} L_j - L_i}{p} \right) - 1 \right] \quad \text{where } x \in \{t/2, t\} \tag{5}$$

Equation (5) is positive, i.e. there exists an incentive to wash trade, if the product of the "grossed-up" fees and the scaling factor due to other makers' liquidity exceeds one. This product will often be less than one, particularly if other makers are moderately active; but it is sometimes possible for this term to exceed one, especially in low liquidity periods. This is the unfortunate and accepted side effect of trying to make incentives as generous and effective as possible during low liquidity periods.[4] Liquidity is indeed the greater priority in DeepBook, and so this risk is tolerated in such scenarios.

2.3 Residual Collected Fees

In general, not all fees collected in a pool for an epoch will be recycled as maker incentives, particularly as aggregate liquidity approaches and crosses the phase-out point p. The final step of the token flow is to burn the residual collected fees, as a means of fortifying the protocol's integrity against wash trading.

The underlying assumption in Equation (5), i.e. the marginal payoff to wash trading, is that fees paid and not directly recouped as incentives *can not be recouped in any other way.* If this statement is violated and those residual fees can be collected, even imperfectly or with some lag, the conditions in Equation (5) weaken, and wash trading becomes economical across a wider range of scenarios and parameterizations. This will undermine DeepBook's integrity.

To illustrate, consider the stylized model in Equation (6), in which a maker is considering whether to put on a wash trade ($t = 1$) or not ($t = 0$). The maker recoups $\alpha < 1$ of fees f paid as incentives. The residual fees $(1 - \alpha)f$ can be collected back with some probability π. Thus, the maker faces the following tradeoff.

$$\max_{t \in \{0,1\}} t\left(\alpha f + \pi(1 - \alpha)f - f\right) \tag{6}$$

The most robust solution that ensures wash trading does not occur along the widest range of parameterizations is to set $\pi = 0$. Any other solution, such as redistributing residual fees in later epochs, collecting residual fees in a community treasury program, cross-subsidizing other pools, and so on, allows the maker to potentially recoup those fees and thus pushes π up. In turn, the only way to credibly set $\pi = 0$ is to burn the residual fees, as an irreversible action that ensures credible commitment for the mechanism.

This aspect of DeepBook is strongly inspired by Ethereum's Improvement Proposal 1559, which

[4] Specifically, this vulnerability stems from the conversion of fees from non-staking makers into incentives to staking makers. The only guaranteed solution against wash trading would be to block fees from non-staking makers from circulating to staking makers, although this would be less effective in boosting liquidity during low-liquidity epochs.

burns base fees as a security measure to prevent off-chain collusion. As Roughgarden (2020) explains:

> The base fee revenues of a block must be burned or otherwise withheld from the block's miner, as otherwise the miner could collude with users off-chain to costlessly simulate a first-price auction.[5]

Finally, note that wash trading often offers extra-protocol benefits to the trader. In particular, wash trading participants on DeepBook may have broader motives beyond direct maker incentives, e.g. triggering liquidations, manufacturing volatility, or spoofing to generate price dislocations on other exchanges. This reinforces the importance of setting $\pi = 0$, rather than setting it simply low but positive, as the strongest defense against wash trading.

3 Governance

Governance within the context of a DeepBook pool has the following flow:

- During an epoch, stakers can issue a proposal to change the three customizable parameters for the subsequent epoch, regarding taker fees t, maker fees m, and minimum staking requirements $\bar{s}$.
- Once a proposal or set of proposals is live, stakers can vote for a single proposal (or for the status quo) in that same epoch, with their votes computed as a function of their stake.
- At the end of the epoch, a proposal that earns over 50% of the votes cast will be implemented for the subsequent epoch, subject to total voting reaching quorum. If no proposal earns 50% of votes cast, the status quo earns 50%, or quorum is not reached, no proposal is implemented and the parameters remain at their current values.
- There are no minimum staking requirements to vote or issue proposals.

Note that broader governance, e.g. changes to the constraints, changes to other parameters, or changes to other aspects of the system altogether, is outside the scope of this mechanism. This would fall under the broader governing framework behind DeepBook, which is maintained through the Sui Improvement Proposals (SIPs) process.

The next sections provide more detail into the two critical components in this system: governance proposals and the function mapping stake to votes.

[5] Roughgarden, Tim (2020). "Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559." arXiv:2012.00854.

3.1 Governance Proposals

Governance proposals can change parameters, subject to certain constraints. These constraints are important for ensuring that taker and maker fees are set at competitive levels with respect to competitor centralized and decentralized exchanges, and are not set at unduly low or high levels that would reflect monopolistic pricing.

In particular, there are three parameters that governance participants can change: the taker baseline fee t, the maker fee m, and the minimum staking requirement $\bar{s}$. Any proposal must respect the following two rules for these parameter values.

1. The taker baseline fee t can not be higher than ten basis points nor lower than five basis points for non-stableswap pools. For stableswap pools, e.g. between two stablecoins or between a token and wrapped token, the taker baseline fee t cannot be higher than one basis point nor lower than 0.5 basis points.

2. The maker fee m can not be higher than five basis points nor lower than two basis points for non-stableswap pools. For stableswap pools, the maker fee m can not be higher than 0.5 basis points nor lower than 0.2 basis points.

Section 4.2 discusses the classification of pools into non-stableswap and stableswap pools. Note that there are no direct constraints on minimum stake $\bar{s}$, as the optimal level will vary by pool.

3.2 Stake into Votes

Since there are no direct constraints on $\bar{s}$, i.e. the minimum stake required to participate in reduced taker fee and maker incentive programs, it is important that this pool governance sets this responsibly. However, this is difficult in standard governance settings. Large stakers would generally be incentivized to raise $\bar{s}$ to exclude all other participants, and give themselves exclusive access to such programs. To mitigate this vulnerability, DeepBook utilizes a mixture of linear and concave functions in turning stake into votes.

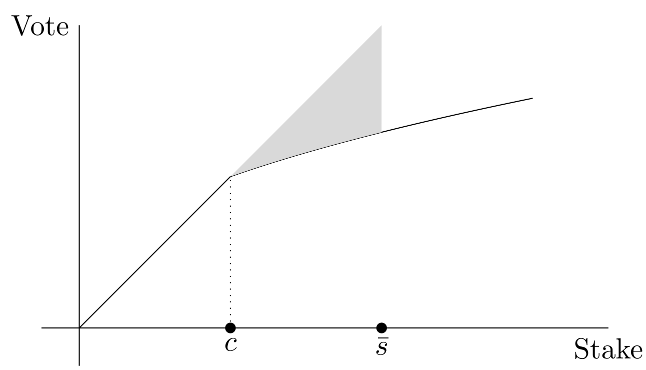
Any voting function must balance two extremes. A purely linear function (e.g. every four staked tokens translates into four votes) would give large stakers with 50% of the vote the ability to carry out a very specific action: raise the minimum stake requirement, $\bar{s}$, to exactly the minimum stake amongst the set, and thus exclude all other parties. A purely concave function (e.g. every four staked tokens translates into two votes) would open up a Sybil vulnerability, wherein participants could split their stake into many tiny micro-stakes to wrest voting control. This is an especially acute vulnerability given Sui's low gas fees.

The class of voting functions in DeepBook are thus linear up to a point, and concave thereafter. Specifically, Equation (7) demonstrates a function for mapping stake s into votes v, along with some "cutoff" parameter c at which point the linear function becomes concave.

$$v(s, c) = \min(c, s) + \max\left(\sqrt{s} - \sqrt{c}, 0\right) \tag{7}$$

To understand this function, consider how vote-optimizing traders would react to this function (and assuming $c \leq \bar{s}$). Traders that had less than $\bar{s}$ stake would split their stake into smaller tranches controlled by separate wallets, all bounded by the cutoff point c, such that their effective voting power would be the same as their stake. But traders that had more than $\bar{s}$ would face a tradeoff. They too could exhibit such Sybil behavior, but this would make them ineligible for any taker fee discounts or maker incentives. More likely, therefore, they would have one "primary" tranche of $\bar{s}$ (from their main wallet) and a set of residual tranches (run by separate wallets), each bounded by c.[6] In short, they would convert $s - \bar{s}$ into linear voting rights, but they would lose some voting power from c to $\bar{s}$ in one wallet due to the function's concavity. This is illustrated in Figure 5.

Figure 5: Voting Function



The figure depicts the voting function mapping stake to vote. In addition, it shows the relative loss of voting power (in gray) that a large staker faces to enjoy their staking benefits, versus the strategy of splitting up their stake across multiple accounts to maximize voting power.

Figure 5 thus illustrates the tradeoffs around c. A high cutoff c, e.g. $c = \bar{s}$, would turn this into a traditional linear scheme. A low c, e.g. $c = \bar{s}/100$, would severely dilute the power of the stakers, perhaps to the point that they could be overwhelmed by capital-light adversaries. The cutoff is thus set as $c = \bar{s}/2$. In other words, the voting function is linear up to $\bar{s}/2$ and concave thereafter.[7] Specifically, this yields the voting function in Equation (8).

$$v(s) = \max\left(\sqrt{s} - \sqrt{\frac{1}{2}\bar{s}}, 0\right) + \min\left(\frac{1}{2}\bar{s}, s\right) \tag{8}$$

Although this section is framed largely in the context of governance attacks, concavity is useful more generally for democratization in the DeepBook protocol. In almost all protocols, a few critical token holders ("whales") control all governance decisions. DeepBook is thus rare in operationalizing

[6] While such traders could, in principle, take one epoch to maximize voting power and have no tranche cross c, this would not be sustainable. They would not enjoy any benefits of staking; and in order to enjoy such benefits, they would immediately face a loss in governance votes.

[7] This is almost equivalent to setting a minimum stake requirement of $\bar{s}/2$ to vote. However, the primary difference is that Equation (8) allows governance participation for wallets with less capital than this threshold.

concave voting and giving smaller token holders an outsized voice; and that has many broader benefits, as discussed by Lalley and Weyl (2018).[8] This mechanism is only made possible by gating certain benefits on a minimum stake, which weakens the otherwise-inevitable Sybil attack.

4 Operational Considerations

This section covers four general operational considerations around DeepBook and the DEEP token. These are the automated adjustment of the phaseout point for the maker incentives p, price conversions between pool tokens and DEEP, the permissionless creation of new pools, and the timeline of DeepBook operations.

4.1 Adjusting the Phaseout

The governance mechanism in Section 3 binds on three parameters, but an automated mechanism oversees p, i.e. the phaseout point for maker incentives in a pool. This can not be left to governance for two reasons. First, pool stakers will always want this as high as possible to maximize their direct earnings, making them a conflicted steward of the parameter. Second, calculating a reasonable value for the parameter is straightforward, making it a suitable candidate for automated mechanisms instead.

Specifically, p should be set to be equal to the median of total liquidity distributed in the last twenty-eight epochs (i.e. four weeks), across both staking and non-staking makers alike. This means that, if the subsequent epoch follows the same distribution, there is a 50% chance that no incentives are given and a 50% chance that some incentives are given.[9]

This design is difficult to game. In particular, in order to raise p and sustain it at elevated levels, makers have to provide generous liquidity over repeated epochs. DeepBook would benefit from such behavior.

Finally, DeepBook deploys medians over twenty-eight epochs to separate out transient from systematic trends. To illustrate, consider two scenarios. In the first scenario, liquidity systematically declines, perhaps because of a decline in usage of the underlying assets. The phaseout point is adjusted down over time, making incentives less generous for the pool's previous volume but correctly-sized for its new volume. This is reasonable behavior for permanent shifts in usage. However, in the second scenario, liquidity temporarily declines, perhaps because of an unrelated shock (e.g. a holiday). Again, the phaseout point might be adjusted down, but this adjustment – and the corresponding downwards revision in incentives – makes it harder for the pool to recover once the shock passes. As such, this is undesirable behavior. This second scenario motivates the selection of a lengthy rather than short window, and further justifies the choice of a median over

[8] Lalley, Steven and Glen Weyl (2018). "Quadratic Voting: How Mechanism Design Can Radicalize Democracy." *AEA Papers & Proceedings*, 108: 33-37.

[9] In practice, it is slightly more than a 50% chance that at least some incentives are deployed, because this point phases out incentives for a maker when the sum of liquidity provided by only *other* makers crosses the threshold, whereas p is computed off the liquidity provided by *all* makers.

a mean. Both features make the mechanism robust to temporary shocks, and more responsive to persistent trends than temporary ones.

4.2 Price Conversions

In any pool, there are three currencies: the two traded currencies and the DeepBook token. The ideal flow is that fees are expressed and computed in the currencies in the pool on a per-unit basis, but are paid in DEEP itself. However, this mechanism only works if there is a way to price at least one and ideally both of the pool tokens against DEEP.

The solution is to rely on a protocol-approved oracle to price DEEP against those reference currencies. For simplicity, this conversion rate will be updated at low frequencies (e.g. a few times per epoch, either as a snapshot or a time-weighted average). This is to keep DeepBook trades, which should be rapid and efficient, as unencumbered as possible. By contrast, updating the price at high frequencies would both add computational complexity to the transaction and further add possible contention around shared real-time price or pool objects.

An oracle can be used for a second purpose, which is identifying stableswap pairs from non-stableswap pairs. Section 3.1 in particular notes that maker and taker fees have different bounds depending on whether the traded pair qualifies as a stableswap or not, and an oracle that prices two tokens against each other (either directly or via the DEEP token) can be used to determine this fairly and programmatically.

This design generates three follow-up questions and concerns. The first is that the DeepBook token price might drift in the time between updates. The second and related one is that the price might be explicitly manipulated in the reference markets used by an oracle, by takers and makers trying to lower their trading fees or raise fees on others. These are fair concerns, but they are not fatal. Under these scenarios, trading fees are slightly lower or higher than their true value, but this does not pose a threat to the underlying solvency of the protocol or any individual pool. Each pool's trading fees are segmented from other pools, and even from other epochs of the same pool. Moreover, any inaccuracies in the price can be corrected rapidly, by pushing another price update to the pool.

Third and most importantly, there may be pools where neither underlying coin is priced against DEEP. This is a more critical issue. DeepBook would not want to necessitate an oracle as a requirement for opening new pools, as that would undermine its flexibility and its ability to react to market conditions. Thus, the solution is twofold.

1. When oracle pricing between pool tokens and DEEP does not exist, fees are collected in pool tokens at the highest taker t and maker m tiers, regardless of the volume traded. These fees are not returned to makers as incentives, nor are they burned. DeepBook governance will ultimately be responsible for deciding what to do with the accumulated non-native tokens, if they are collected in meaningful quantities.[10]

[10] One such idea could be an auction mechanism, where the collected fees are auctioned off to private participants for DEEP.

2. When oracle pricing against DEEP does exist (triggered by the pool ingesting valid reference prices from supported oracles), fees must be collected in DEEP.[11]

DeepBook can solve gaps in oracle coverage creatively. For instance, DeepBook can establish a DEEP price against a coin vis-a-vis an intermediary coin, like SUI, if both pathways are supported by the protocol's chosen oracles.

4.3 Pool Permissionlessness

DeepBook is a permissionless protocol, both for trades and entrepreneurs. Anyone can create a new liquidity pool, and that principle is respected by the token design in this whitepaper.

This principle motivates the lack of cross-subsidization between pools. Pool incentives are restricted only to fees earned in that pool, such that low-liquidity pools can not draw extra funding from profitable pools. While, in theory, such a mechanism would deliver benefits from an efficiency standpoint, it would weaken the principle of permissionlessness. Specifically, to prevent spam pools from draining funds from mature pools, the creation of any pool would have to be approved by protocol-level governance or some other check, and that barrier would chip away at the permissionless ethos behind DeepBook.

The only potential friction is an oracle-supported exchange rate between pool tokens and DEEP, which may not exist for the most cutting-edge of pools. However, this does not stop the pool itself from being created, and only slows its ability to take advantage of the full range of DeepBook's tokenomics.

4.4 Timeline of Operations

DeepBook has many moving parts: trading fees, maker incentives, governance proposals, price updates, and so on. This section outlines the timeline of how these various operations take place on chain.

Figure 6 details the design, which spans two epochs and requires one manual transaction call in each epoch. This transaction can be issued by any party, whether the Sui Foundation or an interested private participant.

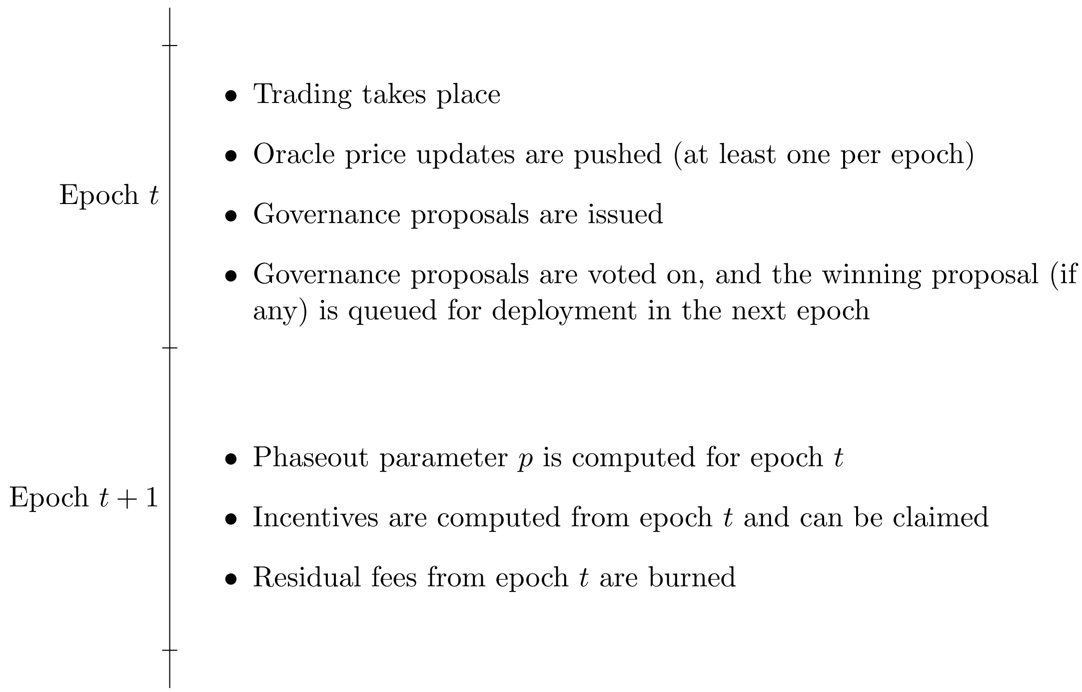
As Figure 6 illustrates, trading, oracle updates, and voting take place during epoch t. Trading is tied to no other operations, to keep it as fast as possible. Oracle updates can be done as standalone updates by any party, but the one manual transaction call each epoch should also provide a price update as a fallback mechanism. Finally, governance is more complex: issuing proposals is simple, but when users vote on proposals through transactions, votes are ideally tallied in tandem. This would allow a governance proposal, once it passes the conditions for ratification, to be queued up for immediate deployment in the subsequent epoch. (Similarly, if a governance proposal first attains and subsequently falls out of the conditions for ratification, it can be de-queued from deployment.)

[11] It is possible that DeepBook could continue supporting collection of non-DEEP fees, at marked up rates over the standard rates and without counting towards the calculation of volume discounts or maker incentives, for rare cases where participants refuse to use or can not use DEEP tokens.

As Figure 6 illustrates next, the reconciliation of the pools operations in epoch t happens in epoch $t+1$, upon receipt of the one manual transaction. Specifically, the transaction will compute the relevant phaseout parameter p for the previous epoch t.[12] It will then compute maker incentives and distribute DEEP tokens accordingly for makers to claim. Finally, it will burn residual fees. In tandem, it will provide an oracle update the live price between DEEP tokens and pool tokens. In general, this step is not time sensitive, and can be done at any point in the epoch, as the state it operates on is settled in the previous epoch and cannot be altered.

For pools that lack such oracle support and so do not collect DEEP fees, this manual transaction and more generally all operations in epoch $t+1$ can be skipped. There are no phaseout points to compute, no prices to update, no maker incentives to distribute, and no fees to burn. Trading and governance in epoch t are all that are needed.

Figure 6: DeepBook Pool Operations



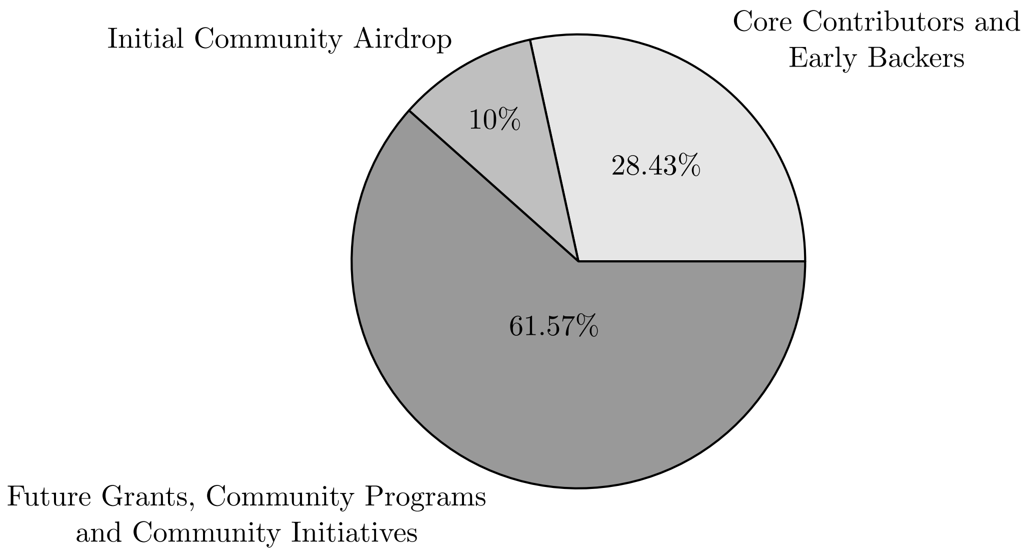
The figure depicts the timeline of DeepBook operations in a given pool, for operations associated with epoch t.

5 Token Allocation

This section briefly outlines the allocation of DeepBook tokens. There are exactly ten billion DeepBook tokens at launch, with no further DeepBook tokens minted. This will be split into three buckets, as visualized in Figure 7.

[12]This parameter is known in epoch t, but it does not need to be computed until epoch $t+1$.

Figure 7: Token Allocation



The figure depicts the allocation of the total supply of ten billion DeepBook tokens.

6 Conclusion

DeepBook has long paired the efficiency of orderbooks and the censorship-free world of blockchain. This upgrade – in tandem with its best-in-class execution – positions DeepBook as the *premier* global liquidity venue in web3. Specifically, the design of the DEEP token incentivizes its many diverse participants into working together to offer ample and around-the-clock liquidity, and thus strengthens its particular speciality in wholesale liquidity.

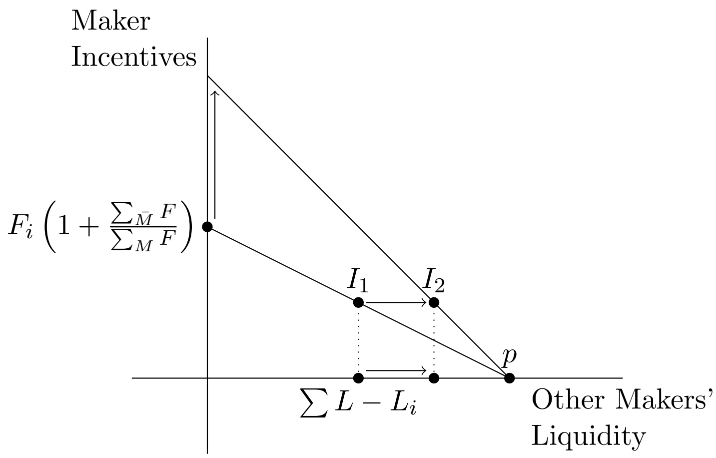
The decentralized finance ecosystem on Sui has many key actors, from consumer-facing protocols to yield aggregators to professional liquidity providers. DeepBook's place in this rich ecosystem is to serve as financial infrastructure, enabling liquidity reallocation between those actors. Future developers can rely on DeepBook when building their protocols, and can leverage composability that only exists with on-chain applications. Future market makers can rely on DeepBook to support other on-chain operations, rather than trying to stitch together on-chain and off-chain surfaces. Future front-end liquidity pools can use DeepBook to smooth imbalances programmatically, and build creative and loss-resistant protocols.

Above all, by virtue of DeepBook's robust and easy-to-use liquidity layer, consumer-facing applications can truly focus on the *consumer*. DeepBook resides in the bedrock of Sui's ecosystem, and this robust foundation enables applications to innovate, build, compete, and tailor for the end user. Together with Sui's robust technical architecture and its many other novel applications, DeepBook positions the Sui blockchain as the hub for the next generation of web3 finance.

A Appendix: Maker Incentives

This short appendix shows a more accurate version of Figure 4a, in which other makers provide more liquidity. In Figure 4a, this only causes a shift along the x-axis; but in practice, it will cause second-order alterations to point $(0, f)$ and thus also change the slope of the line. To illustrate, Figure 8 shows an exaggerated scenario in which *only* other non-staking makers, i.e. makers $\in \bar{M}$, provide more liquidity. This leads to a change in both the shape of the function and the input to the function, causing I_2 to be comparable to I_1.

Figure 8: Change in Maker Incentives: Other non-staking makers provide more liquidity



The figure depicts a version of Figure 4a in which all increased liquidity from other makers only comes from non-staking makers. This both changes the function shape, by changing the ratio term, while changing the function input. The maker moves from I_1 to I_2, but it is unclear in this case whether this is a net increase or decrease in the incentives.

In practice, this is implausible. Non-staking makers are not likely to provide ample liquidity without staking, as they would deprive themselves of a major source of revenue. Thus, in practice, the change in the function shape should be second-order relative to the change in the function input, when other makers provide more liquidity.