dYdX V4 Application Layer Incentive Design

Max Holloway [*]
max@xenophonlabs.com

Thomas Cintra
thomas@xenophonlabs.com

March 2023

Abstract

In the new version of the dYdX protocol – dYdX V4 – there will be a number of changes to the underlying orderbook and matching engine. These changes will usher in new requirements from dYdX's incentive mechanisms. Historically, dYdX has provided token incentives for three key components of the protocol: user fund security, orderbook liquidity, and trading volume. In this report, we provide recommendations for how the protocol should incentivize each of these components after the migration to dYdX V4.

[*]Disclosure. Reference to the DYDX token and its price is necessary for this research. The authors do not own DYDX token, nor are they affiliated with dYdX Trading Inc or any of its affiliates. This research was funded by the dYdX Grants Trust. Any opinions and results stated here are those of the authors, not of dYdX, its affiliates, nor the dYdX Grants Trust. This is not financial advice.

1 Introduction

In June 2022, dYdX Trading Inc announced that they would build dYdX V4: the dYdX Chain [Trab]. This new dYdX version will be a layer-1 blockchain with a perpetual futures orderbook built on the Cosmos SDK. This new version promises to maintain the smooth CEX-like user experience of dYdX V3, while completely changing the underlying orderbook and trade matching engine. Since these changes are so meaningful to the way that advanced traders and market makers will interact with the exchange, we believe it is appropriate to give a thorough analysis of the existing dYdX incentive mechanisms and provide guidance on the mechanisms that would help to grow the platform in dYdX V4.

In this report, we analyze ways that the protocol can optimize these objectives via protocol security mechanisms, trader incentive mechanisms, and liquidity incentive mechanisms. We extend our prior analyses on each of these mechanisms, and we give concrete guidance on the token budget that should be allocated to each initiative.

We analyze mechanisms for securing dYdX Chain in Section 2; we provide mechanisms for the dYdX chain to attract and maintain liquidity in Section 3; and we put forward a new design for attracting dYdX trading volume in Section 4.

We restrict this report's content of application-layer incentives. We intentionally omit discussion of the dYdX Chain's staking token; bridging DYDX to the dYdX Chain; dYdX Chain transaction fee mechanism design; dYdX Chain fee distribution; dYdX Chain governance; and other topics that require dYdX Chain implementation details. In the future, we intend to release a follow-up report that provides recommendations on these implementation-specific topics.

2 Security Incentives

In early November 2022, following the collapse of FTX, the price of SOL experienced a significant increase in volatility. Due to Solana's association with then FTX CEO Sam Bankman-Fried, interest in shorting Solana's token spiked across various crypto exchanges, particularly futures exchanges. The asymmetric increase in short selling lead spot and perpetual prices to diverge, widening funding rates and therefore increasing the risk of forced liquidations. Forced (and missed) liquidations can incur large losses for the protocol, especially in times of increased asset price volatility and asymmetric market liquidity. To address this risk, dYdX placed SOL-USD perps in close-only mode on November 9th, 2022 [Kes]. This kind of on-the-fly risk management will be difficult, if not impossible, when dYdX transitions to a decentralized app chain with V4.

A Cosmos app chain makes an important tradeoff: it increases decentralization but hampers dYdX Trading Inc's ability to dynamically manage risk on the protocol. It also increases the surface area for smart contract exploits, as more elements of dYdX's infrastructure are implemented on-chain. Robust security incentives are more important than ever as we transition to dYdX V4. We argue these security incentives should be comprised of both a prevention mechanism, such as a bug bounty program, and a mitigation mechanism, by the way of a protocol insurance fund.

2.1 The Safety Staking Module

The safety staking module (SSM) was, until November 2022, a mechanism for the protocol to insure against shortfall events. The SSM incentivized DYDX holders to deposit DYDX into an insurance pool. Depositors were paid interest for locking their DYDX for a period of at least 14 days, during which dYdX governance could vote to slash the depositors' funds in order to pay back protocol debts. We have already spoken extensively in previous works about the original SSM, and our recommendation was to migrate the SSM's deposit asset to USDC, rather than DYDX [CHa]. This avoids the wrong way risk of using DYDX to insure dYdX. Following our review of the safety module and a number of concerns raised by other community members, the module was wound down and no alternative form of insurance has been established[1].

This is in stark contrast with most other B2C financial services corporations, which hold some form of insurance for their depositors (e.g. charted banks and FDIC [FDI] or Robinhood and SIPC [Rob]). Unfortunately, crypto-native insurance providers such as Nexus Mutual or Unslashed Finance do not have sufficient size to insure a protocol as large as dYdX, whose TVL hovered around $1B between October 2021 and May 2022.

Here we detail the design of a USDC-denominated safety staking module to provide adequate risk capital for dYdX going into V4. In light of the $2.7B in losses due to smart contract exploits in 2022 alone [Banb], dYdX can pioneer a dynamic, decentralized, and efficient mechanism for protocol insurance. In this section we cover:

1. A formulation for how much DYDX must be allocated to the module to achieve a target insurance value over time.
2. Why this target insurance value is some percentage of protocol TVL.
3. How to properly lockup deposits to prevent withdraws following a shortfall event.
4. An incentive mechanism to dynamically allocate capital to the staking module via a multisig of high-context community members.

2.1.1 Sizing the Safety Staking Module

Since our proposed insurance pool is denominated in USDC[2], we can safely ignore any pro-cyclicality in the pool's TVL. That is, the pool's ability to pay protocol liabilities is not

[1] The liquidation module contains an insurance fund which grows and shrinks according to market liquidity and volatility. Profits and losses from liquidations are fed to this insurance fund as a capital buffer for the liquidation engine, but it is not equipped to protect the protocol against a "black swan" type shortfall event.

[2] For reasons why, please refer to our previous safety module review [CHa]

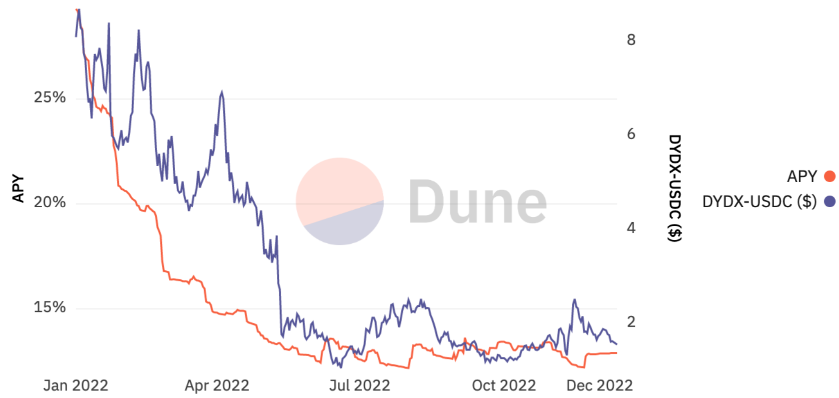

Figure 1: APY vs DYDX-USDC spot price for the legacy safety staking module.

affected by a shortfall event: it is denominated in the same asset as the liabilities. Assuming depositors are yield-maximizing, we can derive an equation for how much DYDX should be used to fund the module. Let TVL_{SSM} and R_{SSM} define the total value locked and the rewards constant for the SSM in USDC respectively, and p_{DYDX} be the spot price for DYDX-USDC, then for a 28 day epoch:

$$APY = \frac{365 \times R_{SSM} \times p_{DYDX}}{28 \times TVL_{SSM}} \tag{1}$$

Since depositors are yield-maximizing, they will be incentivized to deposit into the module as long as yields are more attractive than some risk-free rate r_{free} plus some risk-premium r_{prem}. It follows that TVL_{SSM} will grow until $APY = r_{free} + r_{prem}$, at which point there is no more incentive for deposits. Let's denote this equilibrium yield as APY'. For some detail on what risk free rates are, and why the module must pay a risk premium, refer to appendix B.

We can then derive the expected TVL of the SSM at equilibrium, using the equilibrium yield and equation 1:

$$TVL'_{SSM} = \frac{365 \times R_{SSM} \times p_{DYDX}}{28 \times (r_{free} + r_{prem})} \tag{2}$$

We can use equations 1 and 2 to fund the module and understand its risk profile.

2.1.2 Risk Management and Stress Tests

To discuss protocol risk management we can begin by overviewing risk management frameworks in traditional finance. Traditional Bank Holding Companies (BHCs) have complex frameworks to determine minimum capital requirements to comply with financial regulation. Most notably, The Basel committee is an international effort from global banks to ensure minimum risk compliance amongst important financial institutions. It was created

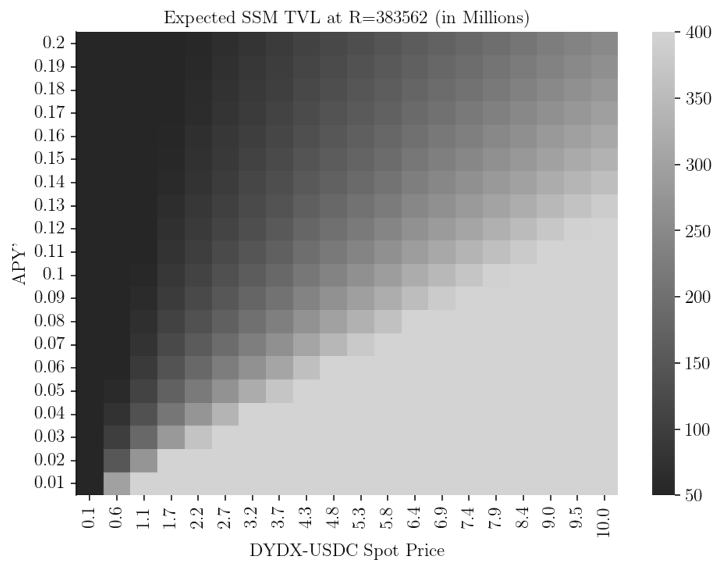

Figure 2: A heat map of TVL_{SSM} for intervals of r and p_{DYDX} using equation 2 and the original SSM allocation, $R_{SSM} = 383,562$. Notice that the white regions denotes any TVL above $400M$, the TVL as of Jan 30th, 2023.

after the 2008 global financial crisis and Basel III is the latest iteration of its guidelines, which major BHCs around the world comply with. While Basel III is particular to how BHCs operate, its framework for understanding risk and ensuring minimum risk capital requirements can be helpful in understanding how dYdX should manage risk. Part of this framework is laid out in detail in the Basel committee's *Minimum Capital Requirements for Market Risk* [Bana]. This framework states the minimum risk capital a BHC must hold is a percentage of its risk weighted assets and outlines the minimum regulatory models it must maintain to project market risk in its banking book[3].

Risk Weighted Assets are a great conceptual framework for how much risk capital a financial institution must hold. They are calculated by assigning some "riskiness" factor to each liability (such as a corporate loans or mortgage backed securities). Taking the dot product of the liability notional values, and their corresponding riskiness factor, the bank arrives at a scalar value representing their total risk weighted assets. dYdX's liabilities (ie. USDC collateral) are encapsulated by the protocol's TVL, and as we have discussed, their riskiness is a function of liquidations and smart contract exploits.

It follows that the risk weighted assets for dYdX are some percentage of the protocol's TVL given some expectation of potential risks. This is what the safety staking module should be sized against. Financial institutions, like the BHCs regulated by Basel III, are required to

[3]Risk weighted assets are used to determine the minimum amount of capital a bank must hold relative to the risk profile of that capital.

Table 1: Stress Test Results, Target Coverage = 15% of TVL

Scenario	TVL	DYDX Price	APY	Rewards Constant	Annualized Cost (USDC)	Coverage (USDC)
Baseline	709.8M	3.06	15%	403.3K	16.1M	106.5M
Adverse	972.9M	1.03	19%	2122.4K	28.4M	145.9M
Severely Adverse	1333.7M	0.34	25%	11168.4K	50.2M	200.1M

hold a minimum of $6\%-15\%$ of their risk weighted assets in reserves [Res] [4]. In decentralized exchanges, the "riskiness" of liquidations and (more importantly) smart contract exploits is not well understood, and the capital requirements are completely unregulated. We can collapse the two variables into a "capital ratio". The capital ratio dictates how much the community treasury (and therefore tokenholders) will pay for insurance, and consequently how many cents users will get on the dollar in the event of a shortfall.

In the worst case scenario where the entire protocol TVL is stolen, a capital ratio of 15% returns 15 cents on the dollar to dYdX users. In figure 1, we quantify the annualized cost to the community treasury to achieve this capital ratio in three different market scenarios.

Stress testing is a useful risk management guide, but is highly sensitive to how variable shocks are constructed. Based on 2022 average prices, yields, and TVL, covering $107M$ USDC in user funds would have coost treasury roughly $16.1M$ USDC, at a unit cost of insurance of 0.15 cents. Notice that the cost per unit of insurance is entirely dependent on the module's yield, which as we have discussed is a function of risk free rates in DeFi and the risk premium assigned to the safety staking module. This is approximately the capital ratio observed by the legacy staking module, which expensed roughly $15.3M$ USDC worth of DYDX token throughout 2022[5].

Determining the capital ratio for dYdX will ultimately be up to the community; this report provides a framework for how much this capital ratio is likely to cost treasury and why it is important to insure some percentage of protocol TVL going into V4. Suggesting a precise number would require assigning some probability to a smart contract exploit which, for an emerging Cosmos App chain, is unlikely to be accurate. For the remainder of this section, we will assume a capital ratio of 15%.

2.1.3 Staking Lockups

The legacy staking module enforced a blackout window, which prohibited withdraws 14 days before the end of the epoch. In our SSM review [CHa], we argued this was likely insufficient given historical voting patterns and the possibility for large stakers to stall the

[4] This is subject to many additional buffers. For example, firms considered to be Global Systemically Important Institutions (G-SIB) are required to hold an additional percentage of risk capital, and many firms like Goldman Sachs or JP Morgan hold an additional percentage on top of that.

[5] We approximate this by taking the DYDX price at the end of each epoch.

vote. A simple solution is to increase the blackout window, but this would still be limitted to the 28 day length of the epoch.

We propose a solution to this issue by introducing a lockup period N on top of the blackout window b. The current implementation of the module requires the `requestWithdraw()` function to be called $b = 3$ days before the end of the epoch[6], such that deposits are available for withdrawing in the following epoch ($N = 1$). Instead, the `requestWithdraw()` function can make funds available $N \geq 2$ epochs in the future, at which point the deposit stops being eligible for rewards

We propose a modification to the SSM's smart contract that:

1. Sets the blackout window back to a 14 day period.
2. Modifies the "requestWithdraw()" function such that funds become available (and rewards stop accruing) $N = 2$ epochs in the future, as opposed to the following epoch.

With $N = 2$, we simply ensure the lower bound on the protocol's risk capital for an additional epoch, providing a greater time buffer within which governance can vote to slash the module. We recommend a lockup period of $N = 2$ but invite discussion from governance to tune this parameter[7]. We can't make N too large without affecting the expected TVL_{SSM}, since increasing N increases the risk a depositor's funds get slashed, and therefore increases the risk premium they will charge to deposit their USDC.

2.1.4 Parameter Tuning

It follows from our stress testing in section 2.1.2 that the cost of safety staking insurance exhibits three key market risks:

1. Interest rate risk: the partial derivative of TVL_{SSM} with respect to r_{free} and r_{prem}. For our purposes, these terms can be bundled together to produce our interest rate risk term $r \equiv r_{free} + r_{prem}$ [8];
2. Delta risk: the partial derivative of TVL_{SSM} with respect to p_{DYDX};
3. TVL risk: movements in the protocol's total value locked.

Movements in either direction from any of these parameters, r, p_{DYDX}, TVL can lead the protocol to either under or overpay for its target insurance. An efficient allocation of risk capital then requires R_{SSM} be dynamically tuned according to material market movements, and crucially not exceed the maximum cost the community is willing to pay. For

[6] Previously, $b = 14$ days

[7] Choosing $N = 2$ is based on our current perception of the DIP life cycle, which can last several weeks and which would be complicated by the sensitive nature of a slashing vote.

[8] Notice r is the same as the $Yield$ term, but we replace with a letter for simplicity.

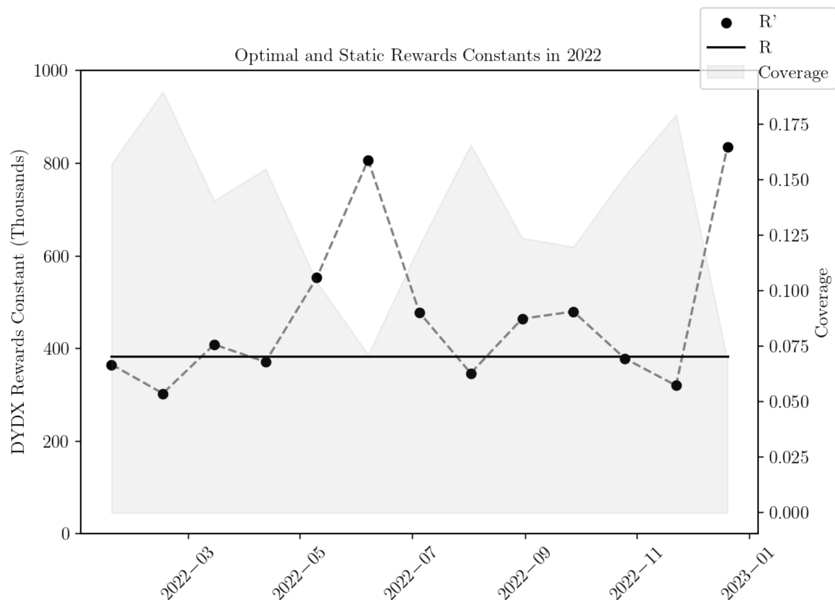

Figure 3: Optimal rewards allocation for a 15% capital ratio using end-of-epoch TVl, DYDX price, and APY throughout 2022. The solid black line is the static 383,562 DYDX initially allocated to the module, wheras the gray area chart shows the coverage that was actually achieved by the module using the static allocation. Notice the volatility in the cost of insurance over time.

example, if DYDX price suddenly rises by several standard deviations, the community might start unnecessarily overpaying for insurance over the course of several epochs before the rewards constant is tuned by the community. The current process to tune such parameters requires lengthy community discussions (DRC), followed by a dYdX Improvement Proposal (DIP) that can take weeks to be approved [Lab]. Furthermore, this process would involve community members at large to consistently fine-tune a somewhat esoteric parameter in the protocol, *without any clear incentive to do so.* Figure 2 provides an illustration of how movements in DYDX price or yields can materially affect the size of the safety staking module.

This is puzzling. Relatively small movements in DeFi yields or DYDX price can lead to significant inefficiency in capital allocation for the safety module. Unfortunately, the significant volatility in DYDX price, DeFi interest rates, and dYdX's total value locked [Coi] [Lab] [l2b] make it unlikely we can project an efficient optimal R'_{SSM}. This is amplified by the lack of diversification in risk factors, which is in stark contrast to the (mostly) hedged balance sheets of regulated BHCs. This is all to say: the efficient allocation of R_{SSM} requires regular maintenance.

We propose the creation of a multisig contract that incentivizes high-context community members to acelerate the DIP process and update the rewards constant epoch-over-epoch. Similar approaches have been taken be Aave governance to dynamically update Aave's risk parameters [Mon] [Can]. In both examples, Aave's community voted in favor of creating a privileged entity that receives funding from Aave's treasury to routinely submits AIPs (Aave

Improvement Proposals) and tune Aave's various risk paramters.

This allows dYdX's risk capital to be highly responsive to market environments with a relatively short turnover of 1 epoch[9]. However, routinely analyzing protocol data and tuning protocol parameters will not be done for free, but could save the protocol significant capital overtime. Therefore, we suggest funding an engagement contract with a community-driver multisig as follows:

- **Structure**
  1. The multisig will consist of 1-3 active community members/entities that are elected through a DIP. It is not required that they be risk or trading experts, but they must be capable of operating a multisig wallet.
  2. The dYdX DAO votes to fund a 6 epoch (∼semi-annual) engagement contract, at which point the members are up for re-election or the contract is wound down based on performance.
  3. The committee sets the SSM reward-per-epoch R_i for each epoch i. They should set the target SSM monthly reward as

$$R_i = \min\left[R_{i-1}\left(\frac{\overbrace{0.15 \cdot (\text{avg. protocol USD TVL in epoch } i-1)}^{\text{target SSM size for epoch } i-1}}{\underbrace{\text{avg. SSM USD TVL in epoch } i-1}_{\text{actual SSM size for epoch } i-1}}\right),\ 400,000\right]$$

     where $400,000$ is the maximum reward per epoch. In words, this formula says that the committee should set the new SSM monthly reward equal to the previous SSM monthly reward, multiplied by the target SSM TVL divided by the observed SSM TVL. The intuition here is that when the target SSM TVL is higher than the observed SSM TVL, then we under-incentivized, so we should increase the SSM rewards; if the target SSM TVL in epoch $i-1$ is smaller than the actual SSM TVL in epoch $i-1$, then we decrease the SSM rewards.

     By giving a formula for the SSM committee, we reduce the role of the committee to that of updating a spreadsheet to calculate a pre-formulated quantity for the SSM reward [10].

- **Responsibilities**

[9] Notice this will affect the risk premium depositors charge, since it increases the governance risk of staking into the module as stated in the beginning of 2.1.1

[10] Alternatively, we could allow the committee to set these parameters by their own accord. This would require the committee members to possess some risk-management expertise, and it would also require more active measures by the community to hold the committee members accountable. For instance, the committee's performance could be measured by the community based on the allocated R_{SSM} and the optimal R'_{SSM} using the observed input parameters (e.g. DYDX price). Let i represent an epoch and $R = R_{SSM}$ and

1. On the first day of every epoch, signers either initiate a DIP to rebalance the R_{SSM}, or a forum post discussing why no rebalancing is required.
2. Either way, the post must detail how the signers arrived at this conclusion. That is, signers are responsible for aggregating and sharing the relevant data for capital allocation (e.g. average DYDX price, staking yields, and protocol TVL throughout the epoch).
3. Signers are responsible for responding to community inquiries in a timely member and pushing the DIP through the DIP life cycle by the end of the epoch.
4. Any proposed parameter changes (unless voted against) must be implemented by the beginning of the following epoch.

- **Incentives and Accountability**

  1. We suggest an initial incentive allocation of \$200 per hour for a total of 5hrs an epoch for each signer.
  2. Signers must be KYC'ed.
  3. To prevent conflicts of interest, signers cannot be actively staking in the safety module, which can be verified given they are KYC'ed.

This, or course, creates an additional axis for centralization. We could imagine automating this process such that the module's contract itself performed this rebalancing act at the end of the epoch. This could be the endgame for a fully decentralized and efficient "insurance as a service" module. However, this would require a more significant engineering effort to prevent the module from being gamed, or from acting unexpectedly during sudden adverse market movements. Solutions such as Aera's treasury management protocol [Aer] or Element's Voting Vaults [Fin] could be effective alternatives to a multisig in the future.

2.2 Bug Bounty Program

Although the safety staking module is a practical way of insuring protocol assets, it does nothing to proactively avoid protocol exploits. Since avoiding exploits is much cheaper than paying for them, we believe bug bounties are a great approach to increase the security of the dYdX Chain. There are multiple platforms that enable protocols to offer bug bounty programs, including Immunefi [Imma] and Code4rena [Cod]. There have been a number of bounties paid in excess of \$1M through these platforms [Immb] [Haw], although none greater than \$10M in size.

$R' = R'_{SSM}$, then we evaluate the signers' performance using a root mean squared error on their allocations:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{6}(R_i - R'_i)^2}{6}} \tag{3}$$

Determining the exact size of the bug bounty program is more of an art than a science. Some combination of protocol value at risk, probability of exploit, and difficulty of exploit execution are all helpful factors to determine the exploit bounty size. Many bounty programs offer different size bounties depending on the size of the exploit. Although we do not yet have a framework for computing protocol value at risk, nor probability of protocol exploit, we can determine these once there are more details on dYdX V4. For the testnet, we recommend sponsoring a relatively small Immunefi competition with payouts ranging from $5,000 to $250,000 for bugs discovered on the dYdX Chain testnet. This payout size range is from the range of payouts offered by other projects on Immunefi today. This entire program could be funded with $1M worth of DYDX token.

2.3 Recommendations

There are many ways to insure the protocol against a shortfall event, and two stand out as the most practical: a USDC-based safety staking module and a bug bounty program.

We recommend the following.

1. Allocate $\$1M$ worth of DYDX token from the Treasury to the funding of a bug bounty program on Immunefi;
2. Fork the legacy implementation of the SSM and modify it to:
   (a) Deposit USDC instead of DYDX.
   (b) Modify `requestWithdraw()` to unlock funds (and stop rewards) $N = 2$ instead of $N = 1$ epochs from the current epoch.
   (c) Reinstate a blackout window of $b = 14$ days.
3. Create and fund a multisig composed of 1-3 active community members. These signers are responsible for rebalancing the allocation of DYDX to the safety module, targeting a 15% coverage of protocol TVL. Signers are compensated at a rate of $200 per hour, 5 hours an epoch, ($\$13k$ annualized per signer), and are re-elected (or the multisig is terminated) on a 6 epoch schedule. Signers set the safety staking module rewards rate once per epoch according to an easy-to-compute formula, thus making their role exceedingly simple. This should not be an expensive effort for the community to fund and maintain.

3 Liquidity Incentives

To provide traders with competitive execution prices, dYdX needs liquidity. In dYdX V3, liquidity was incentivized in two ways: liquidity provider rewards ("LP rewards") and the liquidity staking module ("LSM"). Although the LSM has since been wound down, the LP rewards formula remains vital to keeping the exchange's markets liquid [Sixa] [Sixb].

Here, we provide a brief review of each mechanism, and we share appropriate modifications that are necessary for dYdX V4.

3.1 Current Liquidity Provider Rewards Mechanism

The liquidity provider (LP) rewards mechanism has existed since the release of the DYDX token. There are 1,150,685 DYDX that are distributed to market makers following the conclusion of each 28-day epoch. Each market maker l receives a quality score, Q_l, that reflects the quality of their liquidity provision over the course of the epoch, and the amount of DYDX that they receive is

$$1,150,685 \cdot \frac{Q_l}{\sum_{l'} Q_{l'}}.$$

The quality score itself is computed as a weighted geometric mean of the market maker's depth, spread, uptime, and volume across dYdX markets over the course of the epoch. The specific formulas are not pertinent to this report, however they can be found in the dYdX documentation [Foue].

In dYdX V3, dYdX Trading Inc administers the orderbook, and they have complete access to historical orderbook state, indexed by each account that has placed an order. In dYdX V4, the current plan is to not track the orderbook in consensus, and thus there will not be as clear of a notion of "the" orderbook [Traa]. This will make it more challenging, if not impossible, to replicate the current LP rewards formula, which includes components that necessitate tracking an orderbook, such as depth, spread, and uptime.

Some possible solutions include (a) trusting a single instantiation of the orderbook as the one that computes rewards, (b) creating a mechanism by which each node operator reports their version of the orderbook at the end of each epoch and aggregating these to compute rewards, or (c) create a new LP quality formula that does not rely on orderbook-derived terms. Solution (a) has the obvious downside of trusting a central operator, and solution (b) has the downside of introducing complexity and potential gamability. Although solution (c) has the downside of eliminating the nuance of the current LP rewards formula, we believe it is the most practical of the aforementioned solutions, and we demonstrate here how it could be implemented.

3.2 Execution Surplus Liquidity Provider Rewards Mechanism

As a rough mechanism design heuristic, it is typically best for a mechanism to give payments to a user in proportion to the positive impact that the user has on the protocol, since this aligns the incentives of users with that of the protocol [11]. In the case of liquidity provider rewards, this would imply that we should reward market makers in proportion to their positive impact on the protocol.

[11] For a related idea, see the "Clarke Pivot Rule" [Wikb] and "Pigovian Tax" [Wika].

So how could we quantify the positive impact of a maker on the protocol? One approach is to look at the case where the market maker provides liquidity, then the case where they do not provide liquidity, and determine the difference in execution price for a taker order in each. For instance, suppose that Alicia places a taker order to go long 10 ETH-USD perp, and Bobby's maker orders fill 100% of the volume at an average price of $1,500. Now suppose that Bobby were not present, and that Alicia's order was filled at an average price of $1,600 USD. Then Bobby's order contributed to $(\$1,600 - \$1,500) \cdot 10 = \$1,000$ of cost savings for Alicia (ignoring fees). This quantity, which we will henceforth refer to as **execution surplus**, is the marginal benefit that Bobby's liquidity rendered for Alicia in her trade.

3.2.1 Execution Surplus Rewards: Formal Specification

More formally, each epoch would be defined as a set of blocks with heights in the range $[n, n+k]$, for $n, k \in \mathbb{Z}^+$. Let the set of taker orders in an epoch be called O_t. Furthermore, let the set of approved market makers be L, and the set of markets be M. Let $\{z_m\}$ be a family of functions, where $z_m : O_t \to \{0, 1\}$ is an indicator function that is 1 if an order is in market m and 0 otherwise. Let $d_m(l, o)$ be the execution surplus for the order given by l on order o in market m, as described in the Alicia/Bobby example above. Then for each market maker $l \in L$, and each market $m \in M$, we have that l's score is

$$Q_{l,m} = \sum_{o \in O \text{ s.t. } z_m(o)=1} d_m(l, o). \tag{4}$$

The amount of rewards given to each market, which we denote R_m, would be determined by governance. The amount of rewards that market maker l would get from market m, namely $R_{l,m}$ would be given by that market maker's share of the surplus, i.e.

$$R_{l,m} = R_m \cdot \frac{Q_{l,m}}{\sum_{l' \in L} Q_{l',m}}. \tag{5}$$

The amount of rewards that a market maker would earn over the course of an epoch, R_l, would be

$$R_l = \sum_m R_{l,m} \ . \tag{6}$$

This mechanism aligns the incentives of market makers with the incentives of traders, since the market maker's proportion of the total rewards pool is directly tied to the marginal positive impact they have on pricing.

Execution Surplus Rewards: Considerations.

This mechanism does not come without flaws.

For starters, tying volume to LP rewards has the possible effect of increasing incentives for wash trading. This has been a concern since the initial implementation of trader and LP rewards [Fouf], and increasing the portion of LP rewards that comes from volume only increases this concern. However, it should be noted that there are straightforward methods through which we can detect wash trading, such as running analytics of the bipartite graph of takers, makers, and the portion of volume that a taker pushes to each maker, as illustrated by Figure 4. Since wash trading detection algorithms cannot be tuned without account-level data, we leave it as an area of future work to provide an ironclad wash trading detection algorithm.

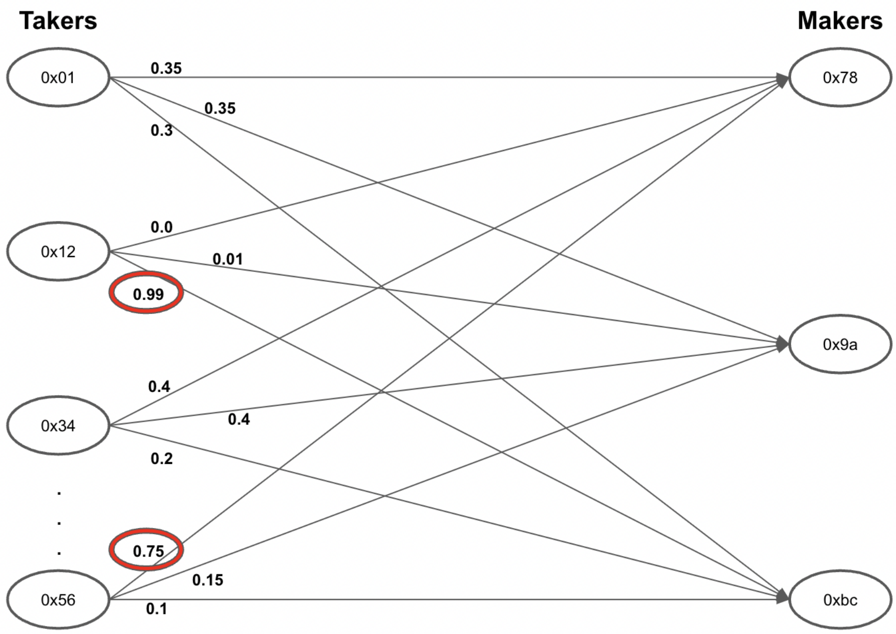

Figure 4: A maker/taker bipartite graph used for wash trade detection. The edge weight is given by the proportion of a taker's volume that goes to a particular maker over the course of an epoch. A wash trading detection algorithm might compute metrics on these edge weights, such as comparing a maker/taker weight to the median of other edge weights for the same maker, then identifying abnormally large values; to illustrate, we circle two abnormally high edge weights.

Another *potential* flaw is that this mechanism may not properly incentivize depth at very small spreads. For instance, if \$100M of volume over an epoch was all via small orders whose marginal execution surplus averaged .01 cents per dollar of volume, then the total execution surplus would be \$10,000. However, if there were a single trade of size \$1M that moved the market by 1%, then its execution surplus would be 1 cent per dollar, and its total execution surplus would be \$10,000. That is, a single sloppy order could contribute to a large portion of the execution surplus.

Although this scenario is not preferable, it does not seem very likely. Since market making rewards are so competitive, we would expect market makers to organize their liquidity to capture the execution surplus from these sloppy trades. For instance, if a large sloppy trade occurred approximately once per day, then market makers who are maximizing their rewards would surely place liquidity on the book in anticipation.

Still, this mechanism may not reward competitive, low spreads. Suppose maker A has an offer of one ETH-USD at \$1,500, maker B has an offer of one ETH-USD at \$1,500, and the next lowest offer of ETH-USD is \$1,600. If a trader places a market order for 1 ETH, then the execution surplus for maker A is zero, since B would fill the order at the same price if A's offer was not present; symmetrically, the execution surplus for maker B is also zero. However, if A and B agree that A will post the offer and B will abstain, then the execution surplus would be \$100, and they would share the rewards attributable to that execution surplus. We believe this example is unlikely to occur in practice, since these agreements require sophisticated communication between competing market makers. Still, it is undesirable for these collusion behaviors to be incentivized.

Finally, and perhaps most importantly, this mechanism's main drawback is that **it requires orderbook data**. If we operate under the assumption that high fidelity orderbook data is not safe to be used to determine rewards in dYdX V4, then we provide a similar, alternative mechanism: volume-based LP rewards.

3.3 Volume-Based Liquidity Provider Rewards Mechanism

In a volume-based LP rewards mechanism would be quite similar to the current trader rewards mechanism: each LP's proportion of the rewards for a given market is equal to their portion of the maker volume for that market.

Volume-Based Mechanism: Formal Specification Let the set of market makers be L, the set of markets be M. Then for each maker $l \in L$ and market $m \in M$, the quality score for the market maker would be

$$Q_{l,m} = \frac{v_{l,m}}{\sum_{l' \in L} v_{l',m}}, \tag{7}$$

where $v_{l,m}$ is the amount of maker volume that market maker l filled on market m. The rewards for market maker l would then be calculated as

$$R_{l,m} = R_m \cdot \frac{Q_{l,m}}{\sum_{l' \in L} Q_{l',m}}, \tag{8}$$

where R_m is the amount of rewards to be given to market m, as determined by governance.

Volume-Based Mechanism: Considerations

Assuming that block producers running the matching engine do not censor any maker orders, it follows that the only way for a maker to increase their volume share is to provide more competitive bids and offers than the other makers. This competition yields a positive outcome on traders, who are able to place trades against markets with better bids and offers.

Furthermore, the amount of volume that a market maker fills can be computed historically via on-chain transaction data. It is even easier to compute a market maker's volume in each market than it is to compute their execution surplus.

Although the volume-based mechanism is simple, it clearly comes with some drawbacks. For one, makers will get just as much rewards share for giving 1 cent better pricing as 1 dollar better pricing. This can lead to pennying, whereby a maker places orders that have only one tick better pricing than the competition, but which receive all of the rewards for filling an opposing trade's volume. This makes it essential for market makers to engage in competitive behavior that does not necessarily translate to better user outcomes. However, as a counterargument to this drawback, it should be noted that professional market makers have immense experience with adversarial continuous-time first price auctions, since first-price auction competitions are the foundation of orderbooks! Even if rewards are removed, there is competition among market makers to get more orderflow at more favorable prices, in which even marginally better bids/offers will receive more orderflow. Thus, it is possible that makers would not need to commit a meaningful amount of work to optimize for volume-based rewards.

Additionally, this mechanism is not exempt from collusion. Theoretically, many market makers could agree to widen their spreads to process less volume. However, collusion seems much less likely under a volume-based rewards mechanism than an execution surplus-based mechanism. This is due to the fact that volume-based mechanisms will always reward a maker for seeing volume, whereas there is a great deal of volume that would not be rewarded under the surplus-based mechanism. In the volume-based mechanism, it is *easier* to accumulate quality score, whereas there are situations in the execution surplus mechansim in which quality score can only be accrued by colluding.

Overall, the volume-based mechanism is quite similar to the execution surplus mechanism, but it has much lower threat of leading to collusion, and it does not require orderbook-level data to compute. This makes it a realistic contender to replace the current dYdX LP rewards mechanism.

3.4 Liquidity Staking Module

The liquidity staking module was a mechanism in dYdX V3 that aimed to increase exchange liquidity by giving interest-free USDC loans to market makers. We wrote extensively about this mechanism in our liquidity staking module review [JCH], and it was decommisioned due to inactivity. There have since been proposals for initiating the LSM again via a third-party service provider [Rod]. Although the idea behind the LSM is quite compelling, the reasons for its lack of use are not well-understood. Some claim that this lack of use was due to poor technical support of the module, while others claim that the lack of use was due to orderflow toxicity and the limited profitability of organic market making on dYdX V3. Overall, we think that the liquidity staking module was an excellent idea, and we believe it should come into consideration if liquidity provider rewards are insufficient to attract sufficient liquidity in dYdX V4. However, we firmly believe that the decision for how much to incentivize a liquidity staking module should be compared to other, high-value initiatives that could be funded by the dYdX treasury, such as marketing outreach or additional grants programs. We suggest revisiting this module if liquidity is an issue following the launch of dYdX V4.

3.5 Recommendations

1. Transfer the LP rewards program for dYdX V3 into dYdX V4, beginning with the same quantity of rewards. Due to unreliability of orderbook data as a source of truth for computing rewards, we suggest that the LP rewards forumla be simplified to only account for volume.

2. Pause for now on the liquidity staking module. If liquidity on dYdX V4 is a problem, then consider partnering with a third-party loan administrator to create a new market maker borrowing pool.

4 Trader Incentives

dYdX launched the Trader Rewards Module in August 2021 to provide additional incentives for trading in dYdX v3, allocating to it 25% of the initial token supply ($250M$ DYDX). The first iteration of the module rewarded traders for holding open interest and paying fees throughout an epoch. Xenophon Labs published an in-depth review of Trader Rewards in early 2022 [CHb]. Since then, there have been two significant changes to the module:

1. DIP 13: The community voted to simplify the rewards formula. The original formula scored each trader using a Cobb-Douglas utility function: $w = f^a d^b$, where w is the trader weight, f is their fees paid, and d is their average open interest (a and b are constants). The simplified formula weighs each trader as their fees paid $w = f$ [Foua].

2. DIP 16: The community voted to reduce the token emissions per epoch from $3,835,616$ DYDX by 25% to $2,876,712$ DYDX [Foub].

4.1 Current Trader Rewards Mechanism

The objective of the trader rewards module was to accelerate adoption and market liquidity for v3 [Fouc]. However, it is not clear that trader rewards has increased trading activity outside of the rewards module itself. In a previous report [CHb] we derived a pure-strategy Nash equilibrium for how traders would game the rewards module. Using this model, we showed that there is an approximately linear relationship between protocol revenue (fees paid) and DYDX price. As the module matured, we saw trading fees approaching the equilibrium predicted by our model, particularly for epochs 4-7.

A corollary of this observation is that the increase in trading activity was predicated on the profitability of trader rewards. From the DYDX API data shown in figure 5 we see that DYDX/USDC spot price correlates strongly with protocol fees paid. We argue

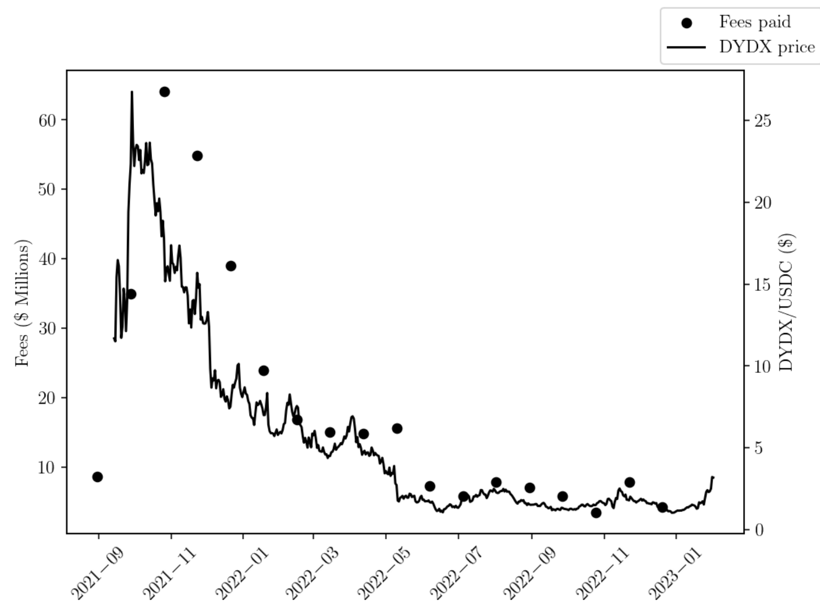

Figure 5: Fees paid taken from the DYDX API plotted against DYDX/USDC spot price taken from Kraken exchange. Fees paid are plotted on epoch end dates. Notice that despite DIPs 13 and 16, the relationship between fees paid and DYDX price is still apparent, suggesting the importance of trader rewards in protocol revenue is still significant.

a significant driver of this (which we discussed at length in our previous review) is that the optimal strategy for those gaming trading rewards is primarily a function of DYDX price. Unlike credit card cash back programs, traders could actually earn more in rewards than they spend in fees for using the protocol. This quickly transformed the module into a game for savvy traders to farm DYDX by wash trading[12] and end-of-epoch fee farming. Unfortunately, neither of these behaviors are effective vehicles for accelerating user adoption or market liquidity outside of the rewards module itself.

The tentative plan for trader rewards is to divert DYDX emissions from v3 to V4. Diverting emissions would accelerate adoption as those gaming the rewards module must transition over to V4. Presumably, this means these traders would remain trading on V4, even outside of the rewards module. If the community decides the Trading Rewards module has failed at increasing user adoption, and has become mostly a game to rack up fees and earn DYDX (which is presumably sold shortly thereafter), the module could then be deprecated some number of epochs into V4 launch. This could look something like:

1. 25% of the current Trading Rewards allocation is diverted to V4, the remainder is sent back to treasury.
2. Trader rewards are set to be emitted for the first 6 epochs after V4 launch to accelerate

[12] Technically illegal/prohibited, see this wash trading forum post by the dYdX Foundation [Foud]

early adoption.

3. After 6 epochs all the DYDX rewards would be depleted, and the module would be functionally deprecated.

However, we argue the Trader Rewards module can be a much more meaningful vector for product adoption going into V4. Instead of designing the module to optimize for accelerated adoption and market liquidity, we propose optimizing for "user stickiness".User stickiness is a measure of retention: higher stickiness means you retain greater user engagement over broader periods of time. This metric can be understood as a ratio. For example, Google tracks user stickiness by measuring the pairwise ratios of daily active users, weekly active users, and monthly active users [Goo]. For dYdX, a comparison of these user profiles throughout 2021 would show poor user retention. As we argued in our review, this is particularly due to the drop in DYDX price, leading to lower activity in the trading rewards module.

4.2 Non-Fungible Trader Rewards

In 2Q22, Starbucks CEO Howard Schults told equity analysts that Starbucks' rewards program drove a record 53% of US company operated revenue [PYM]. CFO Patrick Grismer had previously stated that when "customers join our rewards program, their total spend with Starbucks increases meaningfully." [H]. Their rewards program revolves around the Starbucks app, which offers two key services: improved access to products (mobile payment, pre-ordering, etc.), and rewards for purchases.

Unlike dYdX's trader rewards, Starbucks rewards are denominated in coffee. More specifically, loyal customers are rewarded with a non-fungible claim to a free drink. They cannot sell their free drink in the open market, nor does it represent shares in Starbucks' business. Similarly, airline miles, a product with significant market research to support it, rewards customers with non-fungible claims that can only be redeemed by purchasing another flight.

Generally speaking, customer loyalty programs (CLPs) increase user stickiness primarily through financial incentives [Mäg03]. The effectiveness of a CLP is tied to how well it targets prevailing customer motives with adequate benefit structures [KM14]. For dYdX, the prevailing customer motive is getting a better quoted price. Market liquidity aside, this is achieved by lowering transaction fees. Since DYDX is fungible, rewarding traders with DYDX is somewhat similar to lowering their fees by a commensurate amount in future epochs. But there is a key distinction between this design, and Starbucks rewards and airline miles: there is no incentive for traders to continue to use the protocol; they can just market sell the DYDX. Conversely, a points based program can only be redeemed with further purchases (trades), creating a sunk cost that drives customer retention. Meaningfully, a points-based program is not predicated on the price of DYDX token, and it does not require the current significant emission of DYDX to support it.

We propose a points-based redesign of the trader rewards program:

- Similar to current module, we allocate X points to the rewards module.
- At the end of an epoch, X points are distributed pro-rata to users based on their fees paid (not including any points spent)
- Points can then be used to pay for fees when placing transactions (essentially 1:1 swapping points for USDC). Additionally, points could be used elsewhere within the dYdX ecosystem, such as in buying merch.
- Points balances reset every epoch, meaning points that are accrued on epoch t, but not used on epoch $t+1$, are gone by epoch $t+2$.
- Trader weights are multiplied by a parameter τ, the number of consecutive epochs a trader has paid over Γ in fees, capped at 3 epochs. For example, a user that has paid over $\Gamma = 1k$ in fees for 3 epochs has their points weight multiplied by 3. These would be called "Multipliers" and appear side-by-side with the trader points.

Tentative rewards weighting:

$$r_i = \min\left(f_i, R \cdot \frac{f_i}{\sum_n f_n}\right) \tag{9}$$

This ensures that no user can earn more in rewards than they pay in fees, eliminating the possibility of the module being gamed for profit. However, it also means some rewards might be left unclaimed at the end of an epoch, so the module must be implemented such that any surplus is reinvested to the module, or diverted to the community treasury.

4.3 Recommendations

The dYdX trader rewards module, as it stands today, requires a significant overhaul. We propose that the community experiment with a new trader rewards module that mirrors customer loyalty programs from a number of other industries.

We recommend that the dYdX community institute the non-fungible trader rewards program from Section 4.2. Set the points amount, X, equal to the dYdX trader rewards per-epoch emission rate at the time of instituting the program. Set Γ equal to \$100.

We emphasize that these recommendations are made given the information currently present regarding the dYdX Chain, and we are open to community feedback and further modifications before the release of dYdX Chain Mainnet.

5 Conclusion

As dYdX transitions to the dYdX Chain, its incentive mechanism needs will shift. In this report, we outline three areas – security, liquidity, and volume – where incentives will help drive better outcomes for dYdX's traders, market makers, and token holders. In broad strokes, we recommend three incentive changes: a more optimized safety staking module with an oversight committee, a volume-based LP rewards mechanism, and a customer-loyalty-program-inspired trader rewards revamp. These changes are all implementable by governance, and they are each straightforward on the implementation side.

A Alternative Security Mechanisms

A.1 Third Party Insurance

An alternative to the safety staking module is to have a collection of third parties with large balance sheets insure the protocol. Thus, instead of the protocol paying individuals to stake DYDX, the protocol would pay a group of large, trusted insurers to insure against a shortfall event.

This process would resemble a procurement auction, whereby insurers place bids for the amount of insurance they are willing to provide, as well as the rate they will charge for the amount of insurance. A possible mechanism to implement this is a reverse Dutch Auction: there is a slowly ascending insurance premium over the auction time interval, insurers submit their bid for a certain amount of coverage at the rate at the time they submit, and this continues until either protocol-set max-premium is hit or all of the protocol's insurance coverage demand is met. Any other multi-unit reverse auction mechanism could also fit this use case.

Although this could potentially be a more capital-efficient method of insuring the protocol's debts, this process would realistically be a logistical nightmare. Conveying a "fair" insurance price for a shortfall event would require careful analysis of the dYdX V4 protocol by each of the insurers, which would either be priced into their premium or paid explicitly by the protocol. Furthermore, the institutions would need to agree to interface with dYdX governance, which eliminates most potential insurers. This mechanism is likely unrealistic.

An alternative approach, whereby we utilize existing on-chain mutuals / insurance funds, are also impractical, since the largest one is Nexus Mutual, and it could not feasibly insure a significant portion of dYdX's deposited assets.

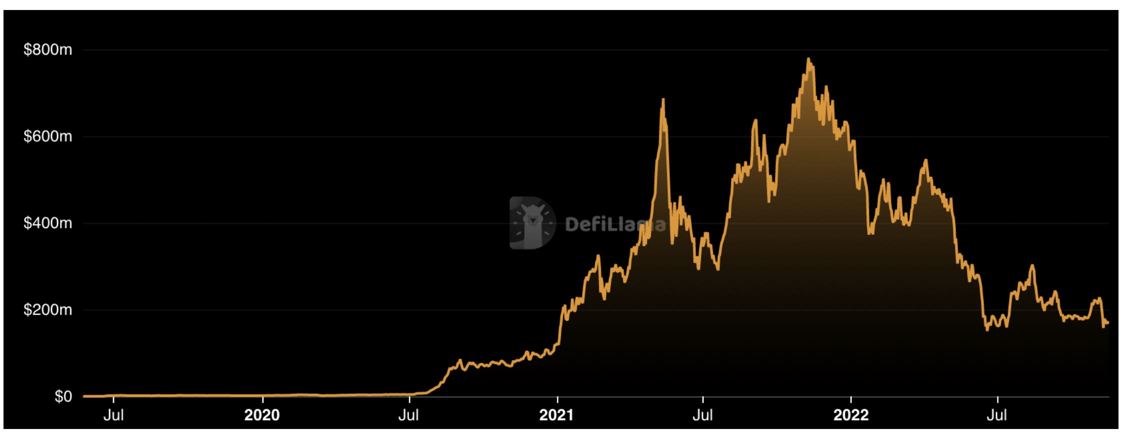

Figure 6: Nexus Mutual Total-Value-Locked [DeF]. Although we do not know precisely how many deposits will be in dYdX V4, it is unlikely that Nexus Mutual could cover a meaningful portion of the protocol's debt.

A.2 Debt Seniority Tranches

Yet another alternative to insure the protocol would be an opt-in self insurance. The idea here is that users who have severe risk aversion to a protocol shortfall event would pay a premium for withdrawal priority over other users who have less risk aversion for a shortfall event. We could implement this by making three [13] seniority tranches: high, medium, and low seniority. Users' USDC would be in the medium tranche by default, and there would be a market where users could buy their way into the high priority tranche, or sell their way into the low priority tranche. Each tranche unit would represent $1 of USDC to be put into the high seniority tranche. If a shortfall event were to occur, then the USDC in the high seniority tranche would be allowed first withdrawals, then the mid priority tranche, then finally the low priority tranche.

Although this mechanism does allow users some control over their withdrawal seniority, tranche units are not equivalent to insurance coverage, and they are much more difficult to reason about. Due to the exoticness of tranche units, as well as the general difficulty of estimating the distribution of shortfall event sizes, we anticipate that the market for tranche units would be illiquid. Furthermore, implementing different tranches would require a different withdrawal UX for the different tranches, as well as an oracle solution for reporting shortfall events to the protocol.

Seniority tranches would enable users to more granularly specify their tolerance for principal risk. However, the market for tranche units would likely be illiquid, and enforcing seniority requirements would severely hinder withdrawal UX. Thus, we do not believe seniority tranches are a practical solution.

[13] We could, in fact, generalize this mechanism to any positive odd number of tranches. Say there were 5 tranches, where tranche with index 1 is the most senior. Tranche 3 would be the default; users could pay to get into tranche 2 while their counterparty goes to tranche 4; users could pay to get into tranche 1 while their counterparty goes to tranche 5.

B Risk Free Rates and Risk Premiums

B.1 DeFi's Risk-Free Rate

The risk-free rate is the rate of return on an investment with no risk of financial loss. It is traditionally defined as the yield on a 10 year treasury bond, because treasuries are backed by the "good faith" of the US treasury, and the probability of the US government defaulting on their debt obligations is thought to be negligible. Since the US dollar is the world's reserve currency, and the US treasury reserves the right to print more dollars, investors generally don't worry about the small (but non-zero) probability that they won't get the face value of their government bond at maturity. This makes US treasuries a useful tool to compare other investments to: if a risky investment expects a lower return than the risk-free rate, it is (generally) a bad investment. Basically all investments in traditional finance are benchmarked against the yield on long-dated US treasuries (think Sharpe ratios).

In DeFi, there is no perfect analogy. Some argue that lending rates for stablecoins on established lending protocols such as Aave offer a comparable benchmark [Nue]. Aave might offer a comparable benchmark to US treasuries because the probability of a depositor losing all their money is low, but much like US treasuries it is not zero. For our purposes, this might be sufficient, but we should be careful in ascribing "risk-free" status to investments that carry with them numerous nuanced risks: smart contract risks, poor risk-management risks, etc. Unlike US treasury holders, Aave depositors can't rely on their counterparty simply printing more money.

Notice that analysts will usually try to align the tenor of the risk-free asset with the tenor of the investment it's compared to. For example, a 1 year investment would be compared to a 1 year US treasury note, instead of a 10 year US treasury bond (which would presumably, but not always, offer a higher interest rate).

B.2 The Risk Premium

The risk premium is the additional yield the protocol must pay depositors given the following risks:

- Slashing risk: the risk depositors lose part (or all) of their principal.
- Smart contract risk: another risk in which depositors might lose part (or all) of their principal.
- Yield asset volatility risk: higher volatility in expected cashflows means a lower Sharpe investment, so it demands a greater yield than an equivalent deposit with stablecoin-denominated yield.
- Governance risk: the risk governance votes to modify the module against the investor, say by lowering R_{SSM} before depositors can request withdraws.

All these risks require the protocol to offer a higher expected rate of return than the risk-free. If dYdX's SSM offered the same return as Aave, then depositors would not deposit their USDC into the SSM. That is, their expected risk-adjusted return would be higher on Aave. It follows that the risk-premium is the additional yield the SSM must offer to be an attractive alternative to depositing on Aave. As long as the pool's APY exceeds the risk-free rate plus the risk premium, depositors will deposit their USDC into the pool, driving down the APY, until we reach some equilibrium state.

References

[Aer] Aera. *Risk-Aware DAO Treasury Management*. URL: https://uploads-ssl.webflow.com/62cd150e5e9efc960319c44d/6346c4525380f8fa6435c2b5_Aera_Whitepaper.pdf. accessed: 12/26/2022.

[Bana] Basel Committee on Banking Supervision. *Minimum capital requirements for market risk*. URL: https://www.bis.org/bcbs/publ/d457.pdf. accessed: 12/14/2022.

[Banb] BanklessTimes. *Funds Lost Trhough Smart Contract Hacks in 2022 Stand at $2.7B, Representing a 1250% Jump Since 2022*. URL: https://www.banklesstimes.com/news/2022/12/08/funds-lost-through-smart-contract-hacks-in-2022-stand-at-dollar27b-representing-a-1250percent-jump-since-2020/. accessed: 12/26/2022.

[Can] Nicholas Cannon. *Proposal: Dynamic Risk Parameters*. URL: https://governance.aave.com/t/proposal-dynamic-risk-parameters/4854. accessed: 12/26/2022.

[CHa] Thomas Cintra and Max Holloway. "dYdX Safety Staking Module Review". In: (). URL: https://xenophonlabs.com/papers/dydx_safety_staking.pdf.

[CHb] Thomas Cintra and Max Holloway. *Trader Rewards Module Review*. URL: https://xenophonlabs.com/papers/dydx_trade_rewards.pdf. accessed: 12/22/2022.

[Cod] Code4rena. *Code4rena Homepage*. URL: https://code4rena.com. accessed: 11/17/2022.

[Coi] CoinMarketCap. *DYDX Price*. URL: https://coinmarketcap.com/currencies/dydx/. accessed: 12/26/2022.

[DeF] DeFiLlama. *Nexus Mutual TVL*. URL: https://defillama.com/protocol/nexus-mutual. accessed: 11/17/2022.

[FDI] FDIC. *Goldman Sachs Bank USA*. URL: https://banks.data.fdic.gov/bankfind-suite/bankfind/details/33124. accessed: 12/26/2022.

[Fin] Element Finance. *Voting Vault*. URL: https://docs.element.fi/governance-council/council-protocol-smart-contracts/voting-vaults. accessed: 12/26/2022.

[Foua] dYdX Foundation. *DIP 13*. URL: https://github.com/dydxfoundation/dip/blob/master/content/dips/DIP-13.md. accessed: 12/26/2022.

[Foub] dYdX Foundation. *DIP 16*. URL: https://github.com/dydxfoundation/dip/blob/master/content/dips/DIP-16.md. accessed: 12/26/2022.

[Fouc] dYdX Foundation. *Trading Rewards*. URL: https://docs.dydx.community/dydx-governance/rewards/trading-rewards. accessed: 12/26/2022.

[Foud] dYdX Foundation. *Wash Trading – Epoch 0*. URL: https://forums.dydx.community/discussion/1677-wash-trading-epoch-0. accessed: 12/26/2022.

[Foue] dYdX Foundation. *Liquidity Provider Rewards*. URL: https://docs.dydx.community/dydx-governance/rewards/liquidity-provider-rewards. accessed: 11/17/2022.

[Fouf] dYdX Foundation. *Review of Epoch 0*. URL: https://dydx.foundation/blog/epoch-0. accessed: 11/17/2022.

[Goo] Google. *User Stickiness*. URL: https://support.google.com/analytics/answer/12993725?hl=en#:~:text=User%20stickiness%20compares%20engagement%20by,good%20engagement%20and%20user%20retention.. accessed: 12/26/2022.

[H] Leah H. *Starbucks: Winning on rewards, loyalty, and data*. URL: https://d3.harvard.edu/platform-digit/submission/starbucks-winning-on-rewards-loyalty-and-data/. accessed: 12/22/2022.

[Haw] Jessica Haworth. *Polygon pays out record $2 million bug bounty reward for critical vulnerability*. URL: https://portswigger.net/daily-swig/polygon-pays-out-record-2-million-bug-bounty-reward-for-critical-vulnerability. accessed: 11/27/2022.

[Imma] ImmuneFi. *ImmuneFi Homepage*. URL: https://immunefi.com. accessed: 11/17/2022.

[Immb] Immunefi. *Wormhole Uninitialized Proxy Bugfix Review*. URL: https://medium.com/immunefi/wormhole-uninitialized-proxy-bugfix-review-90250c41a43a. accessed: 11/27/2022.

[JCH] Justin Jiang, Thomas Cintra, and Max Holloway. "dYdX Liquidity Staking Module Review". In: (). URL: https://xenophonlabs.com/dydx_liquidity_staking.pdf.

[Kes] Sam Kessler. *DeFi Exchange Platform dYdX Places Solana in 'Close Only' Mode*. URL: https://www.coindesk.com/tech/2022/11/09/defi-exchange-platform-dydx-places-solana-in-close-only-mode/. accessed: 01/21/2022.

[KM14] Henning Kreis and Alexander Mafael. "The influence of customer loyalty program design on the relationship between customer motives and value perception". In: *Journal of Retailing and Consumer Services* 21 (July 2014), pp. 590–600. DOI: 10.1016/j.jretconser.2014.04.006.

[l2b] l2beat. *dYdX TVL*. URL: https://l2beat.com/scaling/projects/dydx. accessed: 12/14/2022.

[Lab] Xenophon Labs. *Safety Staking Review Dune Dashboard*. URL: https://dune.com/tncintra/dydx-safety-staking. accessed: 12/18/2022.

[Mäg03] Anne W. Mägi. "Share of wallet in retailing: the effects of customer satisfaction, loyalty cards and shopper characteristics". In: *Journal of Retailing* 79.2 (2003), pp. 97–106. ISSN: 0022-4359. DOI: https://doi.org/10.1016/S0022-4359(03)00008-3. URL: https://www.sciencedirect.com/science/article/pii/S0022435903000083.

[Mon] Monet-Supply. *Proposal: Aave Risk DAO*. URL: https://governance.aave.com/t/proposal-aave-risk-dao/4729/1. accessed: 12/26/2022.

[Nue] William Nuelle. *DeFi's Risk Free Rate*. URL: https://www.galaxy.com/research/insights/defis-risk-free-rate. accessed: 01/29/2022.

[PYM] PYMNTS. *Starbucks Invests in New Rewards as Loyalty Members Help Inflation-Proof the Brand*. URL: https://www.pymnts.com/earnings/2022/starbucks-invests-new-rewards-loyalty-members-help-inflation-proof-brand. accessed: 12/22/2022.

[Res] Federal Reserve. *Dodd-Frank Act Stress Test Publications*. URL: https://www.federalreserve.gov/publications/large-bank-capital-requirements-20210805.htm. accessed: 12/14/2022.

[Rob] Robinhood. *Insurance*. URL: https://robinhood.com/us/en/support/articles/insurance/. accessed: 12/26/2022.

[Rod] Ryan Rodenbaugh. *[Proposal] Move the dYdX Borrower Pool to TrueFi*. URL: https://forums.dydx.community/discussion/6434-proposal-move-the-dydx-borrower-pool-to-truefi. accessed: 11/17/2022.

[Sixa] Sixtant. *DRC - Winding down the Borrowing Pool*. URL: https://forums.dydx.community/discussion/5661-drc-winding-down-the-borrowing-pool. accessed: 11/17/2022.

[Sixb] Sixtant. *Wind down the Borrowing Pool?* URL: https://snapshot.org/#/dydxgov.eth/proposal/0x6a0c11c6749661fa59f887e521deada80d30dcbc0961af555227a5c4 accessed: 11/17/2022.

[Traa] dYdX Trading. *dYdX V4 - Full Decentralization*. URL: https://dydx.exchange/blog/v4-full-decentralization. accessed: 11/17/2022.

[Trab] dYdX Trading Inc. *Announcing dYdX Chain*. URL: https://dydx.exchange/blog/dydx-chain. accessed: 01/27/2022.

[Wika] Wikipedia contributors. *Pigouvian tax — Wikipedia, The Free Encyclopedia*. [Online; accessed 25-November-2022]. URL: https://en.wikipedia.org/w/index.php?title=Pigouvian_tax&oldid=1123394474.

[Wikb] Wikipedia contributors. *Vickrey–Clarke–Groves mechanism — Wikipedia, The Free Encyclopedia*. URL: https://en.wikipedia.org/w/index.php?title=Vickrey%E2%80%93Clarke%E2%80%93Groves_mechanism&oldid=1074505319. [Online; accessed 17-November-2022].