Tenor Protocol: Ethereum's Fixed Rate Order Book

Pierre-Yves Gendron
py@shippooor.xyz

August 15, 2024

WORKING DRAFT

Abstract

The Tenor Protocol is an onchain primitive designed to facilitate the lending and borrowing of ERC20 tokens at fixed interest rates using a set of fully onchain interest rate order books. Tenor fixed rate markets can be deployed on top of any pre-existing money markets allowing users to opt in to a fixed rate experience. Fixed rate markets can be deployed by curators with a limited set of immutable parameters. Tenor markets inherit the risk parameters of the underlying money markets making them simple to deploy. At any time before maturity users can roll their lending or borrowing positions forward to a new fixed rate maturity. If not extended, fixed rate positions convert to the underlying money market variable rate after maturity.

1 Introduction

In this paper, we introduce the design of the Tenor protocol, a set of smart contracts that facilitates the deployment of fixed rate lending and borrowing markets on top of pre-existing money markets in a layered approach. The Tenor architecture enables several key features:

- **Capital efficiency:** Each fixed rate market supports a set of unopinionated onchain interest rate order books. Tenor order books are designed to allow users to trade interest rates similarly to how Uniswap enables users to trade prices. The protocol's order books natively support limit orders allowing users to express their interest rate preferences. Limit orders enable borrowers and lenders to set orders at specific interest rates, allowing them to secure their desired rates until maturity. Limit orders can enable peer-to-peer trades with no interest rate slippage, provided a sufficiently narrow interest rate spread. Limit orders also facilitate the bootstrapping of fixed-rate markets by allowing users to place limit orders even in the absence of pool liquidity. This reduces the market's dependence on liquidity providers, a historically challenging aspect limiting the growth of pool-based fixed-rate protocols. Additionally, pending lend limit orders continue to earn the money market variable interest rate until they are filled thereby minimizing the opportunity cost for lenders. Finally, positions and limit orders in any maturity can be used to collateralize debts in another maturity, provided they are within the same market. This offers market makers programmatic leverage allowing them to efficiently quote rates along the curve, maximizing liquidity for users and mitigating liquidity fragmentation.
- **Governance minimized:** Tenor fixed rate markets inherit the risk parameters of the underlying money market (e.g., collateral assets, max LTVs, liquidation discounts, oracles) simplifying market listings for market curators and minimizing additional overhead. Fixed rate markets order books operate with unopinionated interest rate models, which depend solely on the *tickSpacing* and *maxRate* parameters specified during the market deployment. These unopinionated interest rate models enable lenders and borrowers to set limit orders at their preferred interest rates providing users with greater flexibility.
- **Composability:** By listing fixed rate markets on top of pre-existing money markets, Tenor allows a pre-existing user base to opt-in to a fixed rate experience. The Tenor fixed rate AMM is implemented using a custom Uniswap V4 NoOp hook. This enables users to lend (e.g., swap USDC for a USDC fixed rate zero coupon bond instrument) directly through Uniswap.

- **Simple UX:** At maturity, fixed-rate positions can be rolled forward to a new maturity passively or will default to start accruing the money market variable rate. This process transitions fixed-rate positions back to the variable rate at maturity without the need for user intervention, simplifying the user experience. In addition to enabling fixed rate lending and borrowing, Tenor also natively supports order book based yield trading in a similar fashion to Pendle.
- **Embedded Maker and Taker fees:** The protocol's structure imposes fees on liquidity "takers." For instance, a lender or borrower who draws from the pool pays a fee determined by the annualized interest rate of his trade. These fees result in spot lenders lending at slightly lower rates and spot borrowers borrowing at slightly higher rates. A portion of these fees is redistributed to "makers"- liquidity providers (LPs) who deposit liquidity in the pool- thereby incentivizing them to engage in market making strategies.

2 Fixed Rate Tokens

To facilitate the exchange of fixed interest rates, Tenor markets enable the minting of tokenized zero coupon bond like instruments called Fixed Rate Tokens (FRTs). The Fixed Rate Token design is inspired by the Yield protocol fyTokens [1] and Notional Finance fCash [3] design. At maturity each Fixed Rate Token converts to a claim on one underlying token. For example, 1 Fixed USDC 2025-12-31 Token converts to a claim on 1 USDC at maturity. Fixed Rate Tokens are defined by a currency type and a maturity date, for example USDC that matures Dec 31st 2025. Fixed Rate Tokens are ERC-4626 compliant and can be used by other DeFi apps.

2.1 Minting Process

Fixed Rate Tokens are always minted in pairs with Fixed Rate Debt Tokens. Specifically, when users mint one Fixed Rate Token, they simultaneously mint a corresponding unit of Fixed Rate Debt. This ensures that all zero coupon bond assets can be offset by an equivalent amount of debts. Consequently, when a user mints a set of Fixed Rate Tokens and Fixed Rate Debts, the overall effect on the value of the user's account is null, as they offset each other. The supply of Fixed Rate Tokens and Fixed Rate Debt Tokens in the protocol should always equal zero.

Fixed Rate Tokens represent the asset side of the zero coupon bond like instruments minted by the protocol. Conversely, Fixed Rate Debts represent the debt side of the zero coupon bond like instruments minted by the protocol. Holders of Fixed Rate Debts must repay 1 unit of loan token (ex: USDC) for each Fixed Rate Debt token they hold at maturity.

2.2 Pricing

The value of Fixed Rate Tokens is computed using continuous compounding. The amount of Fixed Rate Tokens that can be obtained in exchange for Underlying Tokens (ex: USDC) is defined as follow:

$$\text{exchangeRate} = e^{\text{tickInterestRate} \cdot \text{timeToMaturity}} \tag{1}$$

Alternatively the Present Value (PV) of Fixed Rate Tokens in Underlying Tokens (ex: USDC) is defined as follow:

$$\text{fixedRateTokenPV} = \frac{1}{e^{\text{tickInterestRate} \cdot \text{timeToMaturity}}} \tag{2}$$

Where timeToMaturity corresponds to:

$$\text{timeToMaturity} = \frac{\text{marketMaturityUnixTimestamp} - \text{currentUnixTimestamp}}{31536000} \tag{3}$$

Where 31536000 corresponds to the number of seconds in 365 days.

3 Fixed Rate Pools

3.1 Pool Token Types

Each fixed rate pool can hold a combination of two tokens: Fixed Rate Tokens and Loan Tokens. A market's Loan Token is the yield bearing version of the market's Underlying Token. For example, a WETH collateral USDC debt market's Underlying Token would be USDC and the Loan Token would be the yield bearing USDC asset from the underlying money market. The use of a yield bearing asset as the Loan Token enables outstanding liquidity and pending limit orders to be yield bearing which improves the protocol's capital efficiency.

3.2 Interest Rate Ticks

Each fixed rate order book is made up of a set of discrete interest rate ticks. The set of ticks for a given pool are dictated by the market's immutable *tickSpacing* and *maxRate* parameters. For example, if a pool's *tickSpacing* is 1% and the *maxRate* is 10% users can trade at the following rates: (0%, 1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%).

3.3 Order Book Architecture

A fixed rate pool ticks can hold a combination of Loan Tokens or Fixed Rate Tokens aligning with the pool's maturity. The pool's order book liquidity distribution could take the following forms:

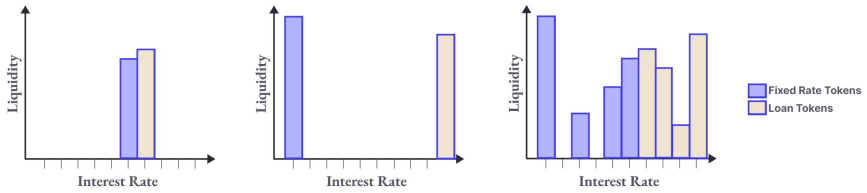

Figure 1: Example Order Book Liquidity Distributions.

Conceptually, borrowers and lenders can lend or borrow at one or a set of ticks in a similar fashion to Uniswap V3. Ticks are discrete, such that trades occur at precise interest rates. If a trade is entirely executed in a single tick, the user experiences no slippage.

3.3.1 Liquidity

Users can add liquidity to a pool using either Loan Tokens or Fixed Rate Tokens. When adding Loan Tokens, users must deposit at a tick higher than the highest tick with Fixed Rate Tokens (the pool's bid). Conversely, when adding Fixed Rate Tokens, users must deposit at a tick lower than the lowest tick with Loan Tokens (the pool's ask). This restriction prevents liquidity providers (LPs) from adding liquidity at sub-optimal market rates.

Consequently, only one tick can simultaneously hold both Fixed Rate Tokens and Loan Tokens. This tick is referred to as the "spot tick." When adding liquidity at the spot tick, users must do so proportionally. As users borrow and lend within a given pool, liquidity providers' positions are effectively converted between Loan Tokens and Fixed Rate Tokens. Therefore, liquidity providers (LPs) accrue a combination of the money market rate and the fixed rate, in addition to collecting trading fees. These trading fees automatically compound in the pool.

3.3.2 Limit Orders

Users can set lending or borrowing limit orders within a pool. For example, a lender might set a limit order at the 10% interest rate by adding Loan Tokens at that specific tick. If a borrower takes all the Loan Tokens in the limit order in exchange for Fixed Rate Tokens, the limit order is marked as filled

and becomes inactive. Once filled, the Fixed Rate Tokens from the limit order cannot be swapped back to Loan Tokens by any other lender. Users can redeem their limit orders at any time, regardless of whether they are unfilled, partially filled or filled.

3.3.3 Lending

When a user lends spot, he effectively trades Loan Tokens for Fixed Rate Tokens in the fixed rate pool (ex: exchange USDC for a USDC zero coupon bond like instrument). The protocol first deposits the user's Underlying Tokens (ex: USDC) in the underlying money market for Loan Tokens (ex: cUSDC). It then finds the highest tick with Fixed Rate Tokens and swap the user's Loan Tokens for Fixed USDC Tokens using the exchangeRate formula 1. The Fixed Rate Tokens are then sent to the lender's account.

3.3.4 Borrowing

When a user borrows spot, he effectively trades Fixed Rate Tokens for Loan Tokens in the fixed rate pool. The user must first deposit collateral assets, then the user mints an offsetting pair of Fixed Rate Tokens and Fixed Rate Debt Tokens. It then finds the lowest tick with Loan Tokens and swap the user's Fixed Rate Tokens for Loan Tokens using the exchangeRate formula 1. Loan Tokens are redeemed for Underlying Tokens and sent to the borrower's wallet. This leaves the borrower with collateral assets and a Fixed Rate Debt tokens in his Tenor account and the borrowed amount in his wallet.

3.3.5 Fees

Any time a user borrows or lends spot in a fixed rate pool he pays a fee. The fee is based on the interest rate at which the user is trading and the FEE_PERCENT parameter. FEE_PERCENT is set by the market curator when deploying the fixed rate market. If the FEE_PERCENT is 10% and a user lends at the 5% tick he effectively lends a 4.5% (5% / 1.1) after accounting for the fee. If a user borrows at the 5% tick he effectively pays a 5.5% (5% * 1.1) rate on his borrow position. Limit order users do not pay fees on their borrow or lend transactions. Only spot lenders and borrowers pay fees since they act as "liquidity takers."

Fees paid by spot lenders and borrowers are split between the market's LPs and the market curator. The fee split between LPs and the market curator is determined by the CURATOR_FEE_SHARE parameter. Market curator fees incentivize risk curators to deploy fixed rate markets on top of their money market. Part of the market curator fees are subject to a protocol fee share.

3.4 Implementation

The protocol uses a tick bitmap inspired by the Uniswap v3 [2] design to track liquidity in the fixed rate order books. This tick bitmap helps identifying the next tick with available liquidity during trades and enforces restrictions when adding liquidity and setting limit order. Tenor fixed rate order books are implemented directly in Uniswap V4 utilizing a custom NoOp hook. Implementing Tenor's order book using Uniswap V4 has the benefit of enabling users to lend at fixed rates only by interacting with Uniswap. Furthermore, it allows Tenor to leverage Uniswap's infrastructure, including Uniswap X to simplify interest rates quoting at scale. To support onchain limit orders, each pool implements a set of order books:

- LP Order Book (one per pool)
- Unfilled Limit Order Book (up to one per tick)
- Partially Filled Limit Order Book (up to one per tick)

3.4.1 LP Order Book

Each pool has a main order book for passive LPs. Liquidity deposited in the main order book ticks can be flipped from Loan Tokens to Fixed Rate Tokens and vice versa as traders interact with the pool.

3.4.2 Limit Order Books

The pool creates limit order books for individual ticks. These limit order books only support swapping in one direction. Given the different behavior of limit orders, limit order books are necessary to track limit order positions distinctly from the main order book. When a limit order book tick becomes partially filled, a new limit order book has to be created for new limit order users interacting in that given tick.

3.4.3 Trading

The protocol implements a similar trade function to Uniswap V3 [2] that finds the next tick with available liquidity and fills the trade in a sequential order by looping through ticks. At each tick the Tenor protocol fills trades sequentially starting with the liquidity available in partially filled limit orders, then unfilled limit orders and finally the liquidity in the main LP order book.

3.4.4 Position Accounting

Because liquidity is not fungible across ticks, each tick LP position or limit order is accounted for distinctly.

4 Fixed Rate Markets

4.1 Factory

The Tenor protocol is constituted of instances that market curators can use to deploy fixed rate markets on top of money markets from protocols such as Euler and Morpho.

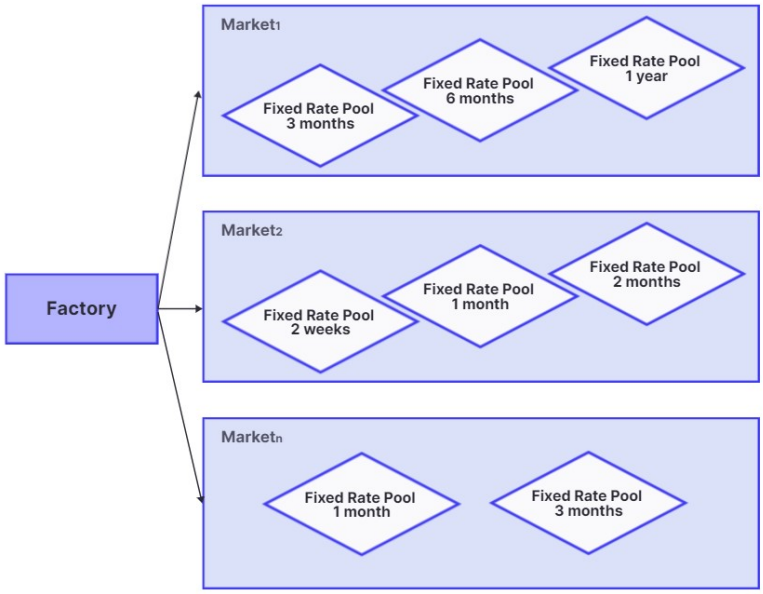

Figure 2: Market and pool deployment structure.

Curators who list fixed rate markets on top of their own money markets enhance the utility of their markets by allowing users to opt in to a fixed-rate lending or borrowing experience. Additionally, market curators can structure their fixed rate markets so that the Total Value Locked (TVL) in the fixed-rate market entirely flows down to their money markets. Fixed rate market curators also benefit from receiving part of the fees generated on all lending and borrowing transactions. Furthermore, they generate fees from excess Oracle Extractable Value (OEV) captured during liquidations. These fees incentivize market curators to list fixed rate markets.

4.2 Markets

By default, fixed rate markets are isolated from one another. After the deployment of a fixed rate market by a market curator, they can simply list fixed rate trading pools of varying maturity dates. Multiple pools of different maturities can be listed under the umbrella of a single fixed rate market, with all pools adhering to the same parameters within a specific market, making them simple to deploy.

Each fixed rate market is characterized by a set of collateral assets (e.g., WBTC, wstETH, WETH) and one loan currency (e.g., USDC). The parameters defining fixed rate markets include:

- Collateral Assets (e.g. ETH)
- Loan Token (e.g. USDC)
- Maximum LTVs
- Liquidation discounts
- Oracle addresses
- Maximum market length (e.g., 1 year)
- Ticks Spacing (e.g., 100 BPS)
- Maximum Rate (e.g., 25%)
- Fee Rate
- Market Curator Fee Share

Fixed rate markets inherit and track the money market risk parameters (e.g., Maximum LTVs, Liquidation discounts, Oracles). This is necessary for the fixed rate market users to seamlessly settle to the underlying money market after maturity. It also simplifies the creation of Tenor Fixed rate markets for market curators by leveraging the risk management decisions made in the underlying market.

4.3 Collateral Structure

Each fixed rate market supports a set of trading pools of different maturities. Each fixed rate market implements a collateral framework to ensure that borrowers overcollateralize their debts with assets of greater value. To achieve this, each market implements a *getNetCollateral* function that calculates the risk adjusted value of an account in Underlying Tokens (e.g., USDC). For maximal capital efficiency, accounts can utilize the following as collateral:

- Collateral Assets
- Fixed Rate Tokens
- Loan Tokens
- LP Tokens
- Limit Orders

4.3.1 Collateral Assets

Collateral assets such as WETH, wstETH, WBTC are risk adjusted using the following formula:

$$\text{riskAdjustedValue}_i = \text{collateralAssetBalance}_i \times \text{collateralHaircut}_i \times \text{oracleExchangeRate}_i \quad (4)$$

4.3.2 Fixed Rate Tokens

To risk-adjust Fixed Rate Tokens, the protocol must first determine the user's net Fixed Rate Token balance. A user can hold Fixed Rate Tokens directly in their account and indirectly through liquidity providing (LP) positions and limit orders.

The protocol will aggregate the user's LP and limit order claims on Fixed Rate Tokens and add them to the user's balance. Subsequently, the net Fixed Rate Token balance is risk-adjusted using the following method:

$$\text{riskAdjustedValue} = \frac{\text{fixedRateTokenBalance}}{\max(1.01, e^{\text{timeToMaturity} \times (\text{maxRate} + \text{buffer})})} \tag{5}$$

This risk adjustment method effectively values Fixed Rate Tokens at the present value (PV) of the pool's maximum rate, augmented by an interest rate buffer. This ensures that Fixed Rate Tokens are always valued more conservatively compared to the maximum rate at which a user could acquire Fixed Rate Tokens. Additionally, the protocol ensures that Fixed Rate Tokens are always discounted at a minimum rate of 1.01 (resulting in a PV of 0.99). This constraint serves to limit the amount of leverage a user can assume on their Fixed Rate Tokens as they approach maturity.

4.3.3 Loan Tokens

A user can hold Loan Tokens directly in their account and indirectly through liquidity provider (LP) positions and limit orders. For LP token and limit order claims on Loan Tokens, the following risk adjustment is performed:

$$\begin{aligned} \text{riskAdjustedValue} = \min \big(& 0.99 \times \text{loanTokenClaims}, \\ & \text{loanTokenClaims} \times e^{\text{timeToMaturity} \times (\text{tickInterestRate} - \text{maxRate} - \text{buffer})} \big) \end{aligned} \tag{6}$$

This adjustment for Loan Tokens is executed as follows. Since a user can hold a claim on Loan Tokens (e.g., USDC) within a specific tick (e.g., 10% tick), it is understood that a borrower could convert the user's position from Loan Tokens to Fixed Rate Tokens (e.g., convert the tick's balances from USDC to fUSDC). In such a scenario, the protocol would discount the fUSDC at the maximum rate plus a buffer (as detailed in the fixed rate token section above). To preempt a discrete change in the valuation of the user's LP position, it is necessary to discount the position as if it were entirely held in Fixed Rate Tokens. Thus, the protocol calculates the present value (PV) of the position under the assumption that it converts to Fixed Rate Tokens, and subsequently discount this position at the maximum rate plus buffer. Finally, the risk-adjusted Loan Token claims are added to the user's Loan Token balance.

4.3.4 Fixed Rate Debts

Fixed rate debts are valued at a present value (PV) of 1, which assumes that debts are always worth their full face value. This conservative approach is used to effectively risk-adjust debts.

$$\text{riskAdjustedValue} = \text{fixedRateDebtBalance} \tag{7}$$

4.3.5 Programmatic Leverage

This collateral framework enables sophisticated market participants to execute yield trading strategies across pools with leverage. For instance, a participant could deposit 1,000 USDC into their account, mint 50,000 fUSDC, and utilize those Fixed Rate Tokens to engage in market-making activities across different maturities. As Fixed Rate Tokens approach maturity, the amount of leverage one can take on Fixed Rate Tokens increases. If a market is listed with a lower *maxRate* parameter it will effectively allow users to use more leverage that if the *maxRate* was set at a higher rate.

4.4 Settlement

After maturity, any external account can call the settlePool function. When called, the protocol records the exchange rate between the Loan Token (e.g., cUSDC) and the Underlying Loan Token (e.g., USDC). For instance, if the exchange rate is 1.05 and a borrower has a 1000 fUSDC position, the borrower must repay 952.4 Loan Tokens (1000 USDC / 1.05), equivalent to 1000 USDC.

4.4.1 Lender Settlement

Lenders automatically start accruing the money market variable rate at maturity. They can set a flag in their account allowing to get automatically rolled forward to lend in a new maturity as long as the new rate clears a certain rate threshold (ex: Money market rate + X%, or a set rate). This simplifies position management and enables users to opt in to a passive lending experience.

4.4.2 Borrower Settlement

Post-maturity, borrowers with outstanding Fixed Rate Debt positions start paying the underlying money market interest rate plus a penalty according to a set schedule. Users can always flash settle their account back to the money market if they want to default back to the money market. If the underlying money market is illiquid such that a borrower's account can't be flash swapped, the borrower himself can always repay his position or can wait for the underlying money market to become liquid again.

They can also set a flag in their account allowing to get automatically rolled forward to borrow in a new maturity as long as the new rate clears a certain rate threshold (ex: Money market rate + X%, or a set rate). This simplifies position management and enables users to opt in to a passive borrowing experience.

4.5 Liquidations

To ensure that borrowers can repay their debts, the protocol enforces that borrowers overcollateralize their debts with assets of greater value. The *getNetCollateral* function continuously risk adjusts the value of user balances. If the *getNetCollateral* function ever returns a negative value, the account is considered unhealthy and becomes eligible for liquidation. Fixed rate markets enforce three liquidation methods:

- Collateral liquidations
- Fixed Rate Token liquidations (Post maturity)
- LP Token liquidations (Post maturity)

4.5.1 Collateral liquidations

This method enables a liquidator to liquidate the collateral assets (e.g., WETH) of an unhealthy account in exchange for repaying the user's debt (e.g., USDC). The liquidator purchases the collateral assets at a discount to the oracle price and repays the account's debt by depositing Loan Tokens (e.g., yield-bearing USDC) into the unhealthy account's portfolio. The protocol implements a liquidation method that mitigates OEV and supports delayed liquidations.

4.5.2 Fixed Rate Token liquidations

After maturity, a user might hold a long-dated Fixed Rate Token position against a matured Fixed Rate Token position. As the interest rate on the matured debt increases, the account's risk adjusted collateral value might become negative. A third party can't settle the account in the underlying money market since it does not accept Fixed Rate Tokens as collateral.

This liquidation method allows for the liquidation of Fixed Rate Tokens in exchange for Loan Tokens at an exchange rate determined by the Fixed Rate Token's present value at the maximum

rate plus a buffer. In practice a liquidator could flash liquidate an account's Fixed Rate Tokens by purchasing them at a low PV (lend at a high interest rate), deposit the Fixed Rate Tokens on a Tenor account and borrow Loan Tokens in a short dated maturity against the Fixed Rate Token position. If there is no Loan Token liquidity in any fixed rate pool it would be the case that this liquidation method requires a liquidator to purchase the Fixed Rate Tokens and hold them over some period of time.

4.5.3 LP Token liquidations

After maturity, a user might hold a long-dated LP position or limit order against a matured Fixed Rate Token position. As the interest rate on the matured debt increases, the account's net collateral might fall below zero.

This liquidation method allows the conversion of the unhealthy account's LP position or limit order to its underlying claim. This liquidation method effectively removes liquidity from the order book on behalf of the liquidated account. The liquidator receives part of the positive net collateral benefit associated with the removal of the Loan Tokens from the tick. If the LP position claim is in Fixed Rate Tokens, the liquidator will need to call the Fixed Rate Token liquidation method described above in order to profit from the transaction.

5 Acknowledgments

I would like to thank Teddy Woodward and Jeff Wu for their feedback.

References

[1] Allan Niemerg Dan Robinson. The yield protocol: On-chain lending with interest rate discovery, 2020.

[2] Moody Salem River Keefer Dan Robinson Hayden Adams, Noah Zinsmeister. Uniswap v3 core, 2021.

[3] Teddy Woodward Jeff Wu. Notional v1 whitepaper, 2020.

Disclaimer