Tenor Protocol: Fixed Rate Market Infrastructure

Pierre-Yves Gendron
py@tenor.finance

November 10, 2024

WORKING DRAFT

Abstract

Tenor is a non-custodial, self-executing, fixed rate lending and borrowing protocol implemented for the Ethereum Virtual Machine. The protocol offers a simple template to create fixed rate markets using pre-existing Morpho money markets as their reference, simplifying the market creation process. The protocol also uses Morpho as its settlement layer at maturity. The protocol enables market curators to deploy fixed interest rate pools. These pools enable the efficient matching of lenders and borrowers at fixed interest rates for set periods of time using a bulletin board, order book-like system. The protocol also implements an aggregation layer between pools to unify liquidity. This layer improves the protocol's capital efficiency, while simplifying loan management for lenders and borrowers.

1 Introduction

The protocol's smart contracts enable the permissionless creation of fixed rate lending and borrowing markets on top of existing Morpho money markets, using a layered approach. The Tenor protocol architecture has several key design features:

- **Minimal Fixed Rate Markets:** Each Tenor market is created in reference to a pre-existing Morpho money market, leveraging the money market's existing parameters (e.g., oracles, max LTVs) thus simplifying the market creation process. Each fixed rate market enables the creation of Fixed Tokens of different maturities. These Fixed Tokens act as principal tokens, enabling lenders and borrowers to lend and borrow at fixed rates for set periods of time. At maturity, Fixed Tokens can be converted back to the money market's variable rate token. Additionally, the use of a reference money market enables borrowers to be settle in the money market after maturity if necessary. Finally, using Morpho money markets as the protocol's base layer makes all unmatched positions on the Tenor protocol earn the Morpho money market variable rate, thereby improving the protocol's capital efficiency for users.
- **Efficient Interest Rate Quoting Mechanism:** Fixed rate markets created using the protocol interact with fixed rate pools of different maturities. These pools, initialized by curators, allow for the matching between lenders and borrowers at fixed interest rates. Each pool implements a simple concentrated liquidity Interest Rate AMM (IR-AMM) where lenders and borrowers can specify their preferred borrowing and lending rates using limit orders. Limit orders offer a simple and efficient matching mechanism to enable peer-to-peer transactions between lenders and borrowers, allowing them to lock their desired rates until maturity. Unmatched lend limit orders in the protocol earn the Morpho variable rate while idle.
- **Governance Minimization:** Tenor fixed rate markets inherit the parameters (e.g. collateral asset, LLTV, liquidation discount, oracle) of their reference Morpho money market, simplifying market creation. Fixed rate pools created using the protocol also have unopinionated interest rate models, which depend solely on the immutable *tickSpacing* and *maxRate* parameters specified at market creation, minimizing overhead for curators. Finally, lending positions in any maturity

can programmatically be used to collateralize debts in longer dated maturities, provided that they are within the same market. This enables the efficient quoting of interest rates between the money market's variable rate and the various Fixed Tokens along the interest rate curve.

- **Re-aggregation of Liquidity:** Fixed-rate markets can face liquidity fragmentation issues due to the creation of individual pools for each maturity. To address this, the protocol implements a liquidity aggregation layer, allowing lenders to simultaneously place limit orders across pools of different maturities. These limit orders can be executed by third parties on behalf of the lenders, provided that the realized interest rate meets or exceeds the lender-specified minimum, and the maturity aligns with the lender's preferences. This mechanism helps re-aggregate liquidity across pools, reduces fragmentation, and provides lenders with a passive auto-rolling lending mechanism, mitigating the typically active management required with fixed-rate lending. This aggregation of liquidity also gives borrowers greater flexibility to source liquidity in their preferred maturity.

2 Loan Tokens

In the Tenor protocol, Loan Tokens refer to the tokenized version of the reference Morpho market's variable rate token. The Loan Token's reference Underlying Token corresponds to the market's loan asset. For example, for a WETH collateral, USDC loan market, the reference Underlying Token would be USDC and the Loan Token would be the Morpho variable rate yield bearing USDC token.

3 Fixed Tokens

To enable lending and borrowing at fixed interest rates, Tenor markets enable the minting of principal tokens called Fixed Tokens. At maturity, each Fixed Token converts to a claim on one Underlying Token in the form of Loan Tokens. For example, 1 Fixed USDC 2025-12-31 Token converts to a claim on 1 USDC at maturity (in the form of Morpho USDC yield bearing tokens). Fixed Tokens are defined by a currency type and a maturity date, for example USDC that matures Dec 31st 2025. Fixed tokens represent a claim on Underlying Tokens at a future date. As Fixed Tokens mature, they are converted back to Loan Tokens.

3.1 Minting Process

Fixed Tokens are always minted in pairs with Fixed Debts. Specifically, when users mint one Fixed Token, they simultaneously mint a corresponding unit of Fixed Debt. This ensures that all Fixed Tokens are offset by an equivalent amount of Fixed Debts. Consequently, when a user mints a set of Fixed Tokens and Fixed Debts, the overall effect on the value of the user's account is null, as they offset each other.

3.2 Fixed Debts

Holders of Fixed Debts must repay 1 unit of the Underlying Token (e.g., USDC) in the form of Loan Tokens for each Fixed Debt they hold at maturity.

4 Fixed Rate Pools

4.1 Pool Token Types

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

4.2 Interest Rate Ticks

Each fixed rate IR-AMM is made of a set of discrete interest rate ticks. The set of ticks for a given pool is 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%). This tick structure effectively creates a concentrated liquidity Interest Rate AMM. Borrowers and lenders can lend or borrow at one or a set of ticks in a similar fashion to Uniswap V3. Ticks are discrete, so that trades occur at precise interest rates. In the absence of fees, a user would experience no slippage if a trade was executed entirely within a single tick.

4.3 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 Tokens, the limit order is marked as filled and becomes inactive. Once filled, the Fixed 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 or partially filled. Users can also relist their Fixed Tokens or Loan Tokens in the IR-AMM at any time before maturity.

4.4 Interest Rate AMM Architecture

Within each pool, ticks can hold a combination of Loan Tokens or Fixed Tokens aligning with the pool's maturity. The pool's IR-AMM liquidity distribution could take the following forms:

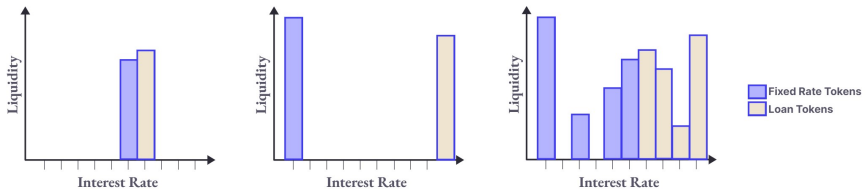

Figure 1: Example IR-AMM Liquidity Distributions.

4.4.1 Exchange Rate

The exchange rate between Loan Tokens and Fixed Tokens is computed using continuous compounding. The amount of Fixed Tokens that can be obtained in exchange for Underlying Tokens (e.g., USDC) is defined as follows:

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

Where timeToMaturity corresponds to:

$$\text{timeToMaturity} = \frac{\text{marketMaturityUnixTimestamp} - \text{currentUnixTimestamp}}{SecondsInYear} \tag{2}$$

4.4.2 Tick Bitmap

The protocol uses a tick bitmap inspired by the Uniswap v3 [1] design to track liquidity in the fixed rate AMMs. This tick bitmap helps identify the next tick with available liquidity during trades and enforces restrictions when adding limit orders.

4.4.3 Swapping

The protocol implements a similar trade function to Uniswap V3 [1] that finds the next tick with available liquidity and fills the swap sequentially by looping through ticks. At each tick the protocol fills swaps starting with the liquidity available in partially filled limit orders and then unfilled limit orders. This ensures that at any time a tick's state is held in a maximum of two different storage slots for a given interest rate tick.

4.4.4 Lending

When a user lends spot, they effectively swap Loan Tokens for Fixed Tokens in one of the protocol's fixed rate pool. The protocol first deposits the user's Underlying Tokens (e.g., USDC) in the reference Morpho money market for Loan Tokens (e.g., Morpho-USDC). It then finds the highest tick with Fixed Tokens and swaps the user's Loan Tokens for Fixed USDC Tokens using the exchangeRate formula 1. The Fixed Tokens are then sent to the lender's account.

4.4.5 Borrowing

When a user borrows spot, they swap Fixed Tokens for Loan Tokens in one of the protocol's fixed rate pool. The user must first deposit collateral assets in the fixed rate market, then the user mints an offsetting pair of Fixed Tokens and Fixed Debts. The protocol then finds the lowest tick with Loan Tokens and swaps the user's Fixed Tokens for Loan Tokens using the exchangeRate formula 1. The Loan Tokens are then used to redeem Underlying Tokens and sent to the borrower's wallet. This leaves the borrower with collateral assets and a Fixed Debt in their Tenor account and the borrowed amount in their wallet.

4.4.6 Fees

Any time a user borrows or lends spot in a fixed rate pool they pay a fee. The fee corresponds to the pool's tick_spacing and is subtracted from the tick interest rate on lend trades and is added on borrow trades. Only spot lenders and borrowers pay fees since they act as liquidity takers. Fees paid by spot lenders and borrowers are split between the pool's limit order users (liquidity providers) and the curator according to the Limit_Fee_Share parameter. This parameter is set by the curator and impacts all pools listed within the curator's pool manager.

5 Pool Managers

Tenor Fixed Rate pools are deployed within contracts known as Pool Managers, which use a singleton design pattern. This allows multiple fixed rate pools to be managed under a single contract instance. While grouped under the same contract, each pool functions independently—interactions with one do not affect the others. The owner of a Pool Manager can update the Limit_Fee_Share parameter for pools created within their contract.

6 Fixed Rate Markets

Anyone can permissionlessly create a fixed rate market using the Tenor protocol template. Curators can deploy fixed rate markets within their own Market Manager contract which will point to their dedicated pool manager contract. While fixed rate markets are all created within the same Market Manager smart contract, each fixed rate market operates in isolation.

6.1 Market Creation

Each fixed rate market is characterized by one or a set of collateral assets (e.g., WETH) and one loan asset (e.g., USDC). At creation, the following parameters define a Tenor fixed rate market:

- Morpho Money Market ID
- Ticks Spacing (e.g., 100 BPS)

- Maximum Rate (e.g., 25%)

Fixed rate markets inherit and track the Morpho money market parameters (e.g., LLTV, Liquidation discount, Oracle). This is necessary for the fixed rate market users to seamlessly settle to the reference money market after maturity. The market creator needs to set the interest rate granularity and maximum rate at which these pools are initialized.

6.2 Collateral Structure

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 *healthCheck* function that calculates the adjusted value of an account's collateral assets and compares it to the value of the account's debts. Users interacting with the Tenor protocol can utilize the following tokens as collateral:

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

6.2.1 Collateral Assets

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

$$\text{adjustedValue}_i = \text{collateralAssetBalance}_i \times \text{LLTV}_i \times \text{oracleExchangeRate}_i \tag{3}$$

6.2.2 Fixed Rate Debts

The protocol values Fixed Debts at their face value during the *healthCheck* calculation process.

6.2.3 Fixed Tokens

To adjust Fixed Tokens, the protocol uses the following method:

$$\text{adjustedValue} = \text{fixedRateTokenBalance} \cdot 0.99 \tag{4}$$

This adjustment ensures that Fixed Tokens are always discounted at a minimum exchange rate of 0.99 before maturity capping the amount of Fixed Debt that can be minted against Fixed Tokens. Outside of this adjustment, Fixed Tokens are recognized at their face value. This is possible given that the protocol only enables users to add Fixed Tokens as collateral if they are shorter dated than their shortest-dated Fixed Debt.

6.2.4 Limit Orders

A user can hold lend limit orders as collateral in their account on the Tenor protocol. Limit orders can have a claim on Loan Tokens and Fixed Tokens. The risk adjusted value of limit orders increases monotonically over time. Moreover adding a limit order as collateral will always decrease the user's collateral versus holding the initial balance of tokens directly at the moment the user's executes the transaction. These two properties are important to ensure that an account's health never decreases due to a discrete change in the limit order's balances.

The protocol performs the following adjustment for Lend limit orders:

$$\text{adjustedValue} = (\text{fixedRateTokenClaims} + \text{loanTokenClaim} \cdot \text{loanToAsset}) \cdot 0.99 \tag{5}$$

The protocol performs the following adjustment for Borrow limit orders:

$$\text{adjustedValue} = \text{loanTokenClaims} \cdot \text{loanToAsset} + \frac{\text{fixedRateTokenClaims} \cdot 0.99}{e^{\text{timeToMaturity} \cdot \text{tickInterestRate}}} \tag{6}$$

The adjustment above values the limit order's Fixed Tokens as if they were converted to Loan Tokens at the tick's exchange rate. It the adds the Loan Tokens to the actual Loan Token claims converted to their value in the market's underlying asset.

6.3 Interest Rate Quoting

The protocol's collateral framework allows users to borrow and lend across multiple pools simultaneously as long as their debts are longer dated than their lending positions. By connecting the variable rate market with the multiple fixed rate pools, the Tenor protocol enables interest rate changes to be propagated across the whole interest rate curve.

6.4 Liquidations

To ensure that borrowers can repay their debts, the protocol enforces that borrowers overcollateralize their debts with assets of greater value. If the *health check* function ever returns a negative value, the account is considered unhealthy and becomes eligible for liquidation.

The protocol implements a liquidation method allowing 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 its oracle price and repays the account's debt by depositing Loan Tokens (e.g., yield-bearing Morpho-USDC) into the unhealthy account's portfolio. Liquidations in Tenor may require the full face value of the debt to be repaid at the time of the liquidation. The early repayment cost of the loan before its maturity may be compensated by the Loan Token's yield over time.

6.5 Settlement

After maturity, any third party can call the settleFixed function. When called, the protocol records the exchange rate between the Loan Token (e.g., Morpho-USDC) and the Underlying Loan Token (e.g., USDC). For instance, if the exchange rate is 1.05 and a borrower has a 1000 Fixed-USDC position, the borrower will need to repay 952.4 Loan Tokens (1000 USDC / 1.05), equivalent to 1000 USDC.

6.5.1 Lender Settlement

Lenders automatically start accruing the money market variable rate at maturity as soon as the settledFixed function is called. Lenders don't have to take action at maturity.

6.5.2 Borrower Settlement

Borrowers that have not rolled forward or repaid their debts at maturity start paying the variable interest rate of the Morpho reference money market on their loan. Any third party can flash settle a borrower's account back to the Morpho money market if that action is required in order to facilitate lender redemptions. If the reference money market is illiquid such that a borrower's account can't be flash swapped, settlers will need to wait for the reference money market to become liquid again.

7 Lending Aggregation

The Tenor protocol implements a lending aggregation mechanism allowing lenders to set an authorization to lend across pools of different maturities. This enables lenders to opt-into an auto-rolling lending mechanism while allowing their liquidity to be borrowed across multiple pools. When granting this authorization, lenders set the following parameters on a per market basis:

- Minimum Lending Rate
- Maximum Duration

The Minimum Lending Rate ensures that any third-party lending on the user's behalf does so at or above a specified rate. The Maximum Duration restricts lending to pools with maturities within a certain timeframe from the current timestamp.

7.1 Auctions

When a lender's Fixed Token position matures within the protocol, a third party can re-lend to a longer-dated maturity on behalf of the lender. To mitigate the likelihood of lending below market rates (even if above the user's minimum rates), the protocol automatically initiates an auction at maturity. The auction starts at the market's maximum interest rate and gradually decreases towards the user's minimum lending rate. A third party can step in to lend at any point. This mechanism protects lenders from potential sandwich-style attacks when their minimum rate is significantly below the market rate, protecting the execution rate of passive lenders.

The protocol uses the previous position's maturity timestamp and the current timestamp to calculate the rate at which a third party can lend on behalf of the user if within the auction period. This enables the auction to run seamlessly.

References

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

Disclaimer

This technical document is for general information purposes only. The opinions reflected herein are subject to change without being updated.