# Introduction

The Gross Protocol signifies a transformation from a decentralized peer-to-peer lending approach to a strategy centered around collective funding. Loans are no longer required to be paired individually, rather they draw upon the collective resources, as well as the borrowed sums and their corresponding collateral. This facilitates immediate loans with attributes determined by the condition of the fund pool.

The interest rate for both borrowers and lenders is decided algorithmically:

- For borrowers, it depends on the cost of money - the amount of funds available in the pool at a specific time. As funds are borrowed from the pool, the amount of funds available decreases which raises the interest rate.
- For lenders, this interest rate corresponds to the earn rate, with the algorithm safeguarding a liquidity reserve to guarantee withdrawals at any time.

# Basic Concepts

At the core of a lending pool is the principle of reserve: each pool maintains reserves in various currencies, with the total amount in Ethereum defined as total liquidity. A reserve accepts deposits from lenders. Users can borrow these funds, provided they secure a greater value as collateral, which supports the borrowing position. Specific currencies in the pooled reserves can be configured as collateral or not for borrowing positions, with only low-risk tokens being considered. The amount one can borrow is dependent on the currencies deposited that are still available in the reserves.

Each reserve has a specific Loan-To-Value (LTV), calculated as the weighted average of the different LTVs of the currencies forming the collateral. The weight for each LTV is the equivalent amount of the collateral in BTC. Figure 3, for instance, illustrates an example of these parameters.

Every borrowing position can be initiated with a stable or variable rate. Borrowings have an indefinite duration, and there's no repayment schedule: partial or full repayments can be made at any time. This flexible structure allows for a dynamic lending environment that can adapt to the needs of the users and the state of the market.

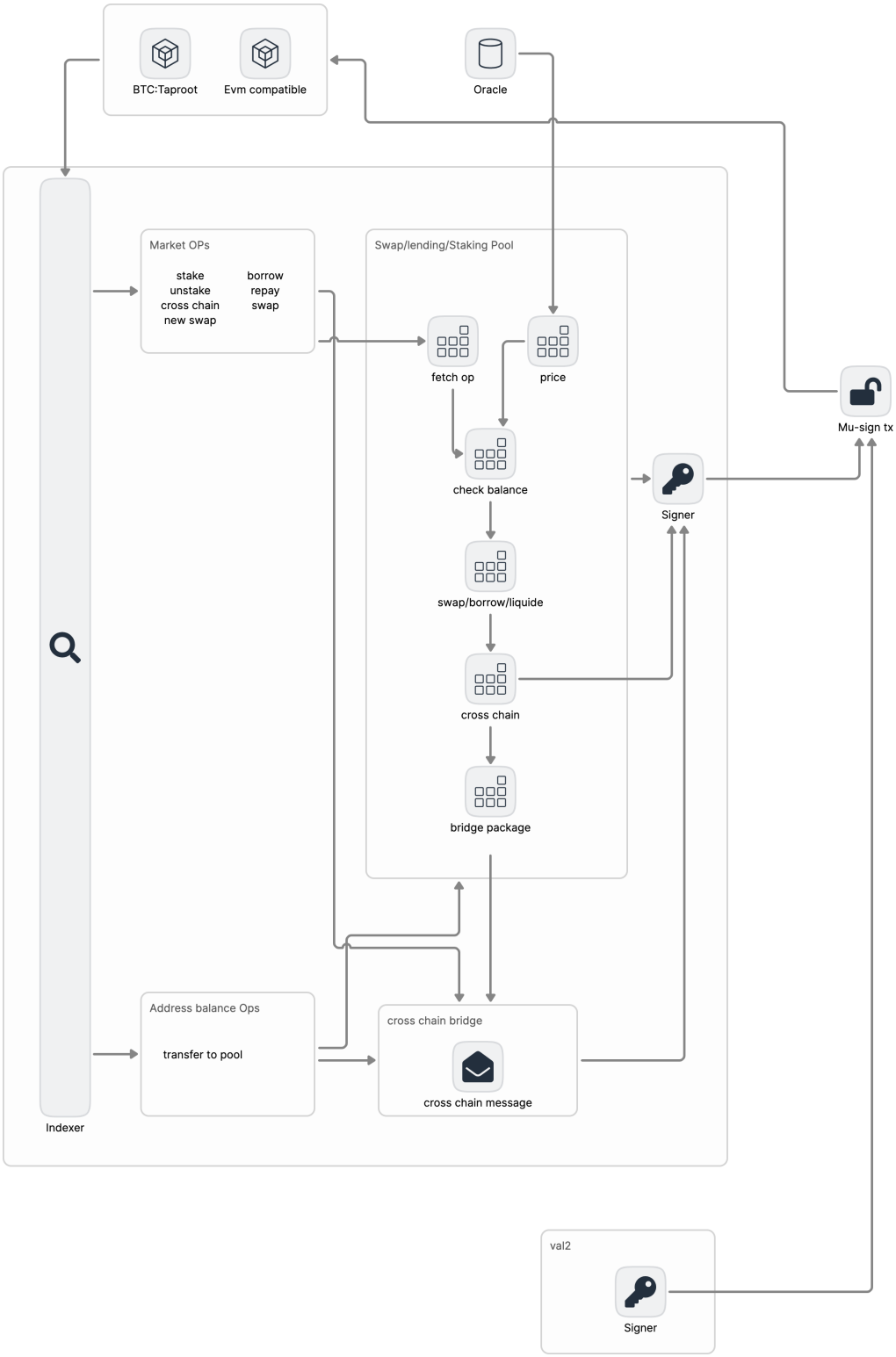
In the event of price fluctuations, a borrow position might be liquidated. A liquidation event occurs when the price of the collateral falls below a certain threshold, known as the liquidation threshold (LQ). When this ratio is reached, a liquidation bonus is triggered, incentivizing liquidators to purchase the collateral at a discounted price. Each reserve has a specific liquidation threshold, calculated in the same manner as the Loan-To-Value (LTV).

The average liquidation threshold (LaQ) is calculated dynamically, using the weighted average of the liquidation thresholds of the collateral's underlying assets.

At any given moment, a borrow position is characterized by its health factor (Hf), a function of the total collateral and the total borrows. The health factor is a measure that determines if a loan is undercollateralized. If the health factor falls below a certain level, it indicates that the loan is at risk of being liquidated

because the value of the collateral is not sufficient to cover the loan amount. This system provides a safeguard to ensure the stability of the lending pool and protect the interests of the lenders.

# Protocol Architecture


BTC:Taproot
Evm compatible
Oracle
Market OPs
stake
unstake
cross chain
new swap
borrow
repay
swap
Swap/lending/Staking Pool
fetch op
price
check balance
swap/borrow/liquide
cross chain
bridge package
Mu-sign tx
Signer
Indexer
Address balance Ops
transfer to pool
cross chain bridge
cross chain message
val2
Signer

# Lending Pool Core

The LendingPoolCore serves as the heart of the Gross Protocol. It plays a central role in managing the state of the system and executing key functions. Here's what the LendingPoolCore does:

- Maintains the State of Every Reserve and All Deposited Assets: The LendingPoolCore keeps track of the current state of every reserve in the lending pool. This includes the total amount of each currency in the reserve, the current interest rates, and other key parameters. It also tracks all the assets that have been deposited into the lending pool by lenders.

- Handles Basic Logic: The LendingPoolCore is responsible for executing the basic logic of the Gross Protocol. This includes the cumulation of the indexes, which is a measure of the total amount of interest that has been earned or paid over time. It also involves the calculation of the interest rates, which are determined algorithmically based on the current state of the lending pool.

# LendingPool

The LendingPool, facilitates interaction with the reserves through a variety of actions. These actions are:

Deposit: Users can deposit funds into a reserve, which increases the pool's liquidity.

Borrow: Users can borrow funds from the pool, provided they have sufficient collateral.

Rate Swap: Users can switch between stable and variable interest rates for their loans.

Flash Loan: This feature allows users to borrow and repay a loan within a single transaction, provided the loan is repaid by the end of the transaction.

Redeem: Users can withdraw their deposited funds, along with any accrued interest.

Repay: Borrowers can repay their loans, either partially or in full, at any time.

Liquidation: If a borrow position becomes undercollateralized (the value of the collateral falls below a certain threshold), it can be liquidated. The collateral is sold to repay the loan.

When a user opens a borrow position, the tokens used as collateral are locked and cannot be transferred. This ensures that the collateral remains in place to secure the loan.

## Governance

# Stable Rate

In the context of the Gross Protocol, the Stable Rate Theory refers to the concept of offering borrowers a rate that remains relatively consistent over the life of the loan. This rate is typically higher than the variable rate at the time of loan initiation, but it provides the borrower with the certainty of a fixed repayment amount, shielding them from potential interest rate volatility. This approach can be particularly beneficial in a volatile market environment.

The implementation of a fixed rate model on top of a pool is indeed complex. Fixed rates are difficult to manage algorithmically since the cost of borrowing can fluctuate with market conditions and available liquidity. Consequently, there may be scenarios (such as sudden market changes or bank runs) where managing stable rate borrow positions would necessitate the use of specific heuristics based on time or economic constraints. Following this line of thought, we identified two potential ways of managing fixed rates:

The subsequent section elucidates the application of steady rates within the system and their respective constraints.

Establishing a constant rate model on a liquidity pool is intricate. Indeed, constant rates are challenging to manage algorithmically, as the borrowing cost fluctuates with market conditions and available liquidity. Consequently, there could be scenarios (unforeseen market shifts, liquidity crunches...) where managing steady rate borrow positions would necessitate the use of distinct heuristics grounded in temporal or economic restrictions. Following this logic, we've pinpointed two potential methods for managing constant rates:

- Applying temporal restrictions: constant rates could function flawlessly within a time-limited context. If a loan has a fixed term, it should withstand extreme market conditions, as the borrower is obligated to repay at the end of the loan term. Regrettably, time-limited constant rate loans do not align with our specific open-ended loan use case. This would necessitate a degree of user experience friction, where users would need to generate and manage multiple loans with varying time restrictions.

- Applying rate restrictions: An interest rate determined at the outset of a loan could be influenced by market conditions, preventing it from remaining constant. If the rate deviates excessively from the market, it can be recalibrated. This wouldn't constitute a pure constant rate, open-term loan - as the rate could fluctuate over the loan's lifespan – nonetheless, users will encounter actual constant rates during specific time frames, or when sufficient liquidity is present. This particular implementation has been selected for integration into Gross's Protocol, identified as the stable rate.

The LendingPool unveils a function, rebalanceStableBorrowRate, which enables the rebalancing of a particular user's stable rate interest. This function can be invoked by anyone; however, there is no direct incentive for the caller to adjust a specific user's rate. For this reason, Gross will deploy an agent that will periodically scrutinize all stable rate positions and rebalance those deemed necessary. The rebalance strategy will be determined offchain by the agent, implying that users meeting the rebalance conditions may not be rebalanced immediately. Since these conditions hinge on available liquidity and market state, there may be temporary scenarios where an immediate rebalance is not required.

This does not introduce any centralization aspects to the protocol. Even if the agent ceases to function, anyone can invoke the rebalance function of the LendingPool. While there isn't a direct incentive to do so ("why should I bother?"), there exists an indirect incentive for the ecosystem. Indeed, even if the agent were to discontinue, depositors might still desire to trigger a rebalance of the lowest borrow rate positions to enhance the liquidity rate and/or compel borrowers to close their positions, thereby increasing available liquidity. Conversely, in a downscale scenario, borrowers have a direct incentive to perform a rebalance of their positions to reduce the interest rate.