Xenon: Universal Margin Protocol on Solana

Muneeb Mohammed
muneeb@xenon.so

June 22, 2022

Abstract

Xenon is a Universal Margin Protocol built on Solana, which enables its users to trade with leverage across DeFi protocols all from a single Margin Account. For long, lending protocols in DeFi have suffered from low capital efficiency due to over-collateralization and Xenon aims to solve this issue through leveraging the power of composability. Xenon provides under-collaterized loans which can be used on any whitelisted protocol Xenon integrates with. It also provides a "one-stop" liquidity avenue where lenders would be able to provide liquidity to the entire Solana DeFi ecosystem, and traders can use the same liquidity across a variety of protocols. Be it a DEX, a derivatives market or even another lending platform for that matter, all can be integrated with Xenon and therefore be leveraged upon.

Contents

Motivation

1. **Liquidity Fragmentation**: Liquidity is often an issue in developing ecosystems. Due to the prevalence of a large number of protocols often isolated/disconnected from each other, liquidity keeps on getting more and more diluted and fragmented. This calls for a unification in the ecosystem, where the focus needs to shift from isolation to composability. Instead of building in silos, protocols need to compose and build on top of each other in order to give way for a well-connected and healthy ecosystem.
2. **Capital Efficiency**: Current lending protocols are mainly directed towards providing over-collaterized loans, which by design underutilize capital. A platform which can provide under-collaterized loans would greatly increase capital efficiency, consequently providing sustainable yields to the lenders.
3. **Restricted Leverage**: Although most derivative markets and leveraged farming platforms do offer in-house leverage, its usage is quite limited - with the capital being restricted to the protocol itself. Thus, there is a strict need to democratize leverage so that even protocols without native margin support are able to access a liquidity boost.
4. **Portfolio Fragmentation**: Many users deal with having isolated positions in their portfolios. More often than not, portfolios span over multiple platforms and consequently are disconnected from each other. A Unified Margin Account would not only connect these isolated positions (by virtue of Aggregated Valuation) but also provide avenues for **portfolio margining** - where a profitable position can be used to hedge against a failing position, thereby reducing overall portfolio risk. What's more, the whole of the portfolio can be used as a collateral since funds can be borrowed against the cumulative value and not restricted to any isolated position.

Introduction

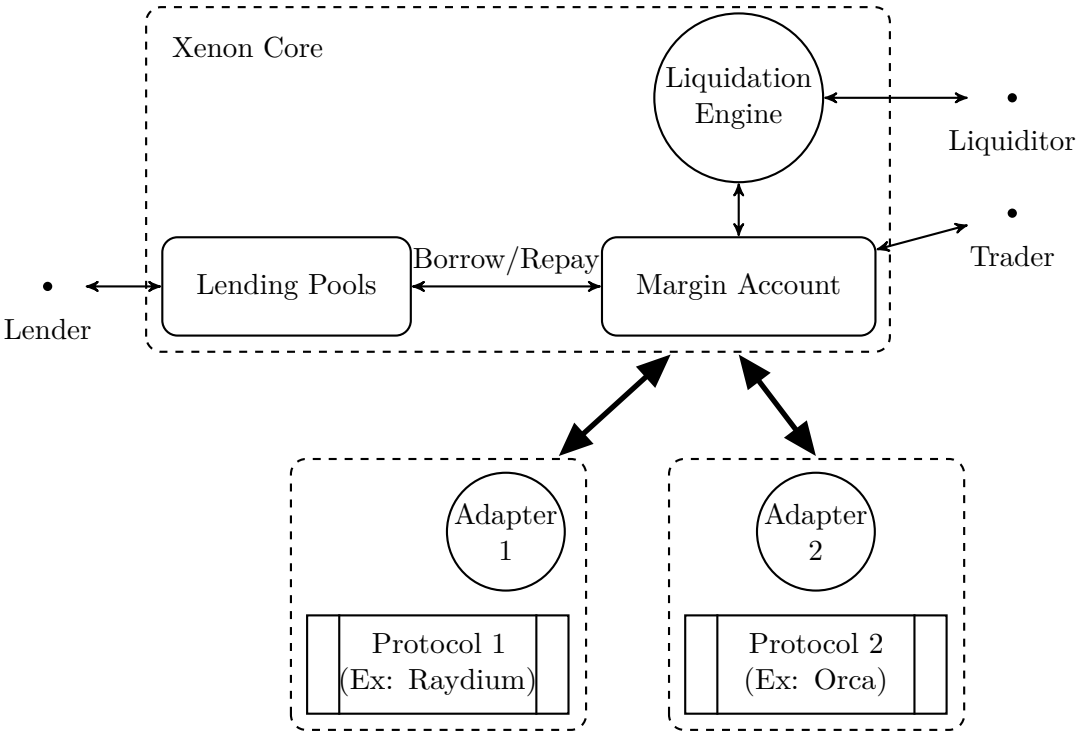
Xenon works by connecting lending pools to margin accounts with embedded leverage, which can execute whitelisted actions on any DeFi protocol Xenon has composed with. These margin accounts are hosted by the protocol, and hold the borrowers' positions, initial collateral, and other intermediary assets, and acts as security over borrowed capital to mitigate risk of loss for the lending pool. To borrow, users are required to first deposit capital into the margin account, before they are able to access credit from the lending pools, and interact with external DeFi protocols. Before a user is able to withdraw funds from their margin account, they must first repay the entirety of the loan to minimise risk of insolvency. A liquidation engine ensures that margin accounts are sufficiently capitalised, and that accounts at risk of insolvency are closed prior to any potential impact to the lending pool

What enables Xenon is **Composability**. By composing over protocols, Xenon creates a meta-ecosystem of sorts where the borrowed assets, even though transferred out of the lending pool and traded across protocols - are always under the purview of Xenon and as a result *safu.*

System Architecture

Xenon employs a modular architecture where additional protocols can be seamlessly integrated. The architecture is extensible, allowing new integrations and protocols to be easily added. The architecture is split into two components:

- **Xenon Core** - consisting of the core lending pools, margin accounts and the liquidation engine
- **Adapters** - which contain the protocol-specific logic to interact with the underlying protocols



Lending Pools

Interest Rate Model

Lending pools in Xenon are similar to Aave's, with an interest-rate model determined by the utilization ratio of assets in the pool (U)

$$U = \frac{Borrows}{Borrows + Available\ Deposits}$$

The interest rate of the pool (I_t) is determined by

$$|I_t| = \begin{cases} \frac{U(t)}{U_{opt}} I_{opt}, & U(t) \leq U_{opt} \\ I_{opt} + \frac{U(t) - U_{opt}}{1 - U_{opt}} (I_{max} - I_{opt}), & U(t) > U_{opt} \end{cases}$$

where I_{opt} is the optimal interest rate when $U = U_{opt}$ and I_{max} is the maximum interest rate.

Liquidity Provision

In order to supply liquidity to the pool, lenders deposit assets (initially only USDC/SOL) into the pool and get back **xeTokens** in return. These xeTokens represent the lenders' share in the pool and are transferable as well as redeemable. xeTokens are **interest-bearing tokens** so their value always accrues over time and their price reflects the cumulative interest rate accrual in the asset pool. The amount of xeTokens transferred to the lender is calculated as

$$XE_i = \frac{D_i}{\epsilon_t}$$

where D_i is the amount of asset supplied and ϵ_t is the exchange rate of the xeToken at the time of supply. The exchange rate is the cumulative interest accrued over time and is calculated as

$$\epsilon_t = \epsilon_{t-1} + I_t$$

In order to withdraw liquidity from the pool, lenders would need to exchange their xeTokens. Upon withdraw, these xeTokens would be burned and equivalent amount of assets would be then transferred back to the lender from the pool. The amount of assets withdrawn (W_i) would be determined by the exchange rate of the xeTokens at the time of withdraw (ϵ_t)

$$W_i = XE_i * \epsilon_t$$

Margin Account

Margin accounts in Xenon are implemented as Programmatically Derived Addresses (PDAs) which hold the deposited funds (by the account owner) as well as the borrowed funds (liabilities from the lending pool)

The privileges to the PDA are programmatically controlled in a way so as to allow the account owner to only borrow funds in a ratio of the collateral deposited, and only withdraw when the whole of the loan is repaid back to the lending pool.

Margin Accounts also act as a Point-of-Interaction of the trader/account owner with the Target Protocols since it is the one holding the custody of assets.

Account Variables

The **Assets** (A) represent the aggregated valuation (in base asset) of all positions owned by a margin account. Assets can be held by the margin account natively or be deposited in a target protocol. They're calculated as

$$A = \sum p_i * (F_i + D_i)$$

where p_i is the Oracle Price of the ith Asset (in base asset), F_i is the Free balance of the ith asset held by the margin account and D_i is the cumulative balance of the asset deposited/locked in any of the target protocols.

The **Liabilities** (L) represent the net value borrowed (in base asset) from the lending pool (B) along with the interest accrued (I) on the borrow. This is the amount which is owed and needs to be repayed back to the lending pool

$$L(t) = B + I(t)$$

The **Account Value** (AV) is the net value of a margin account and is calculated as a difference between Assets and Liabilities. Whereas the **Collateral Ratio** (C) is defined as the ratio of Assets to Liabilities

$$AV(t) = A - L(t)$$
$$C(t) = A/L(t)$$

Theoretically the account debt is secured as long as $AV > 0$ and $C > 1$ such that there are enough assets to repay back the liabilities to the lending pool but in practice, this is seldom the case. Often there are execution delays and shortfall events which warrant a need for a buffer between assets and liabilities. Also each asset has different properties (such as volatility & liquidity) and therefore needs to be weighed in for risk accordingly.

The **Health Factor** (HF) is a much better indicator of the account quality since it takes into account the weighted asset value (WAV) instead of the bare asset valuation (A). These are calculated as

$$WAV = \sum p_i * (F_i + D_i) * w_i$$

$$HF = \frac{WAV}{L(t)}$$

where w_i is the asset weight and calculated depending on the volatility and liquidity of the asset (which ultimately translates to varying liquidation risk)

$$w_i = 1 - l_i - l_f$$

where l_i is the liquidation premium for the asset, and l_f is the liquidation fee

In the event where $HF < 1$, the margin account is liable to be liquidated

Key Interactions

1. **Depositing Collateral:** In order to borrow funds, the trader first has to create a margin account and deposit collateral. This collateral would be held in the custody of the margin account and can be used to trade on any of the integrated protocols.
2. **Borrowing**: Once the collateral has been deposited, the trader is eligible to borrow funds from the lending pool. The precise amount which is available for borrow would initially be set to 3x of the collateral (or the cumulative account value) but can later be amended through governance. Similar to the deposited collateral, the borrowed funds would also be held in custody of the margin account and can be used for trading.
3. **Repaying Debt**: All borrows into the margin account accrue interest. The interest is cumulative and is added to the account liabilites. At the time of repayment, the trader would need to repay the initial borrowed amount along with the interest accrued (in the base asset), and the proceeds would then be transferred back to the lending pool
4. **Withdraw**: Once the whole of the debt (along with interest) is paid back to the lending pool, i.e. the account liabilites are null, the trader can close the margin account and withdraw the remaining account assets back into his wallet.
5. **Liquidations**: In the event where the Health Factor (HF) of the account goes below 1, the account is liable to be liquidated. The liquidation process is designed to be as atomic as possible in order to reduce risk of shortfall events. A detailed discussion of it can be found in the "Liquidation Engine" section of the paper.

Account Auctions

Another distinguishing feature of margin accounts is that they are transferable, which allows them to be "auctioned" or "sold" to prospective buyers. Instead of market closing all the account

positions and incurring slippage costs, the account owner if he wishes to exit, may simply put the account up for sale at a discounted price. Upon matching, the buyer transfers the agreed valuation in the base asset to the account owner and in turn receives ownership of the margin account.

This further strengthens the concept of "account as a primitive" and goes on to show that these margin accounts can indeed develop into into a key infrastructure lego in the DeFi ecosystem.

Protocol Interactions

Composability is the key feature of Xenon, and it is what enables it to provide leverage to its users without compromising the safety of its funds. From the previous section, it is clear that it is the margin account which holds the custody of assets and all protocol interactions are routed through it. Upon each interaction, the margin account checks for account health to make sure the account debt is always secured.

But designing a composable architecture on Solana comes with its own challenges. Currently, there is no standard for protocols to adhere to on Solana. Due to the flexibility provided in defining custom interfaces to programs on Solana, each protocol has its unique accounting architecture. This poses a problem to Xenon as it would then need to have custom wrapper code for each additional protocol it needs to interact with.

Xenon solves this by the use of **Adapters**. The adapters (or connectors) allow the core of Xenon's logic to be separated from the protocol-specific bits. This not only allows for modularity but also makes future protocol integrations seamless. Adapters are protocol-specific and would essentially implement an interface standard to be used by the Xenon core. Adapters also play a key role in implementing security whitelists which are discussed later.

Security

Security is of the utmost importance in any lending protocol, and more so in case of margin protocols which provide under-collaterized loans. Both preventive as well as corrective measures need to be in place to ensure that enough there are enough safeguards and the protocol is also prepared for "black swan" events.

Liquidation Engine

Liquidations are the most safety-critical feature of any lending protocol. In order to guarantee that the protocol is always over-collateralized and no loss is incurred to the lenders, the liquidation engine has to be responsive and airtight. Below is the sequence of events which happen at liquidation:

- The Health Factor (HF) of a Margin account goes below 1 and a liquidation event is triggered
- Liquidators who get notified of the trigger, call the **liquidate** instruction which does the following:
    1. Check that the Health Factor (HF) has indeed gone below threshold, abort if false
    2. Repay the liabilites of the margin account in the base asset (borrows + accrued interest) back to the lending pool
    3. Once the debt is transferred to the pool, the liabilites in the margin account are zeroed out and the account ownership is transferred to the liquidator

Upon ownership transfer, the liquidator is free to either hold on to the account assets and use the account for future trading or exit account positions into the base asset and close the account.

The entire liquidation process is designed to be as atomic as possible so that execution delays are minimized and there is minimal risk of a shortfall event

Protocol and Asset Whitelist

Building over protocols and redirecting leveraged assets outside of the protocol's safe world sounds great, but it comes with risk. The target protocols need to be thoroughly vetted so as to ensure the safety of the funds. A vulnerability in any of the target protocols would put whole of Xenon at risk. Even interactions with the protocols would have to be doctored to an extent. Also, since all tradable assets contribute to the account valuation, there is a need to restrict only assets which are healthy (liquid) to be traded through Xenon. Thus Xenon maintains the following whitelists for safety:

- **Protocol Whitelist**: Only whitelisted protocols (programs) would be allowed to integrate with Xenon. This whitelist makes sure that funds are sent only to protocols which have been vetted/audited and deemed to be safe
- **Instruction Whitelist**: Taking the protocol whitelist a bit further, the instruction whitelist would allow only specific interactions between Xenon and the target protocol.

- **Asset Whitelist**: This is a global whitelist and ensures that the only those assets which are safe and have enough liquidity are allowed to be traded through Xenon

The protocol and the asset whitelists are maintained globally within the Xenon core while the instruction whitelist is handled by the protocol-specific adapters.

The whitelists are fixed at initial launch but can be later amended through governance. All in all, whitelists add an extra layer of security and gatekeeping to the protocol

Insurance Fund

Although rare, protocols should always be prepared for shortfall events. Shortfall Events happen when the liabilites of an account exceed the asset value, thereby resulting in a deficit to the lenders. These can be caused by various factors such as liquidation delays, smart contract exploits or oracle attacks. To insure protocol risk, the Xenon Insurance fund is split into two reserves

- **Reserve 1:** This would act as a first line of defense in case of a shortfall, and denominated in USDC. The reserve would be initially funded through a XEN-incentivized staking pool, with a fixed unstaking period. As the value-flows through interest payments and liquidation fees build up the Insurance Fund reserves over time, the staking pool can be gradually disincentivized with reduced emissions
- **Reserve 2**: This reserve would act as a second line of defense, in case when the shortfall loss exceeds the amount in Reserve 1. The reserve would be denominated in XEN tokens and would be funded at token genesis and also through through XEN staking. In case of shortfall, the requisite amount of tokens would be auctioned off and used to offset the remaining loss

Example Use-Cases

1. **Leveraged Arbitrages**
   - Spot → Spot: AMM - CLOB, AMM - AMM
   - Derivative → Spot: Cash and carry, basis trading
   - Interest Rate Arbitrages between Lending markets
2. Unlocking liquidity of LP tokens/yield assets
3. Spot Leverage on DEXes
4. Protocol-Agnostic Liquidity Provision
5. Leveraged Yield Farming
6. Portfolio Margining