Leverage Protocol: A Composable, Undercollateralized Margin Framework for Starknet

Author(s): Jorge Zerpa

Version: 0.1.0

Date: October 14, 2025

Abstract

The **Leverage Protocol** is a decentralized, non-custodial liquidity protocol on the **Stark net** ecosystem designed to facilitate **undercollateralized lending** for margin trading. It provides a **unified margin framework** that allows traders to open leveraged long or short positions across a wide array of integrated third-party DeFi protocols.

By abstracting capital sourcing from trade execution, the protocol enhances **capital efficiency** and unlocks novel trading strategies. Its architecture is centered around two core components:

- An **ERC-4626 compliant Liquidity Pool** for capital provision.
- A **PositionManager** smart contract that serves as a transparent, on-chain broker for trade execution and position tracking.

This paper outlines the protocol's architecture, core mechanics, risk parameters, and its potential to augment the **composability** of the Starknet DeFi landscape.

---

1. Introduction

Starknet's DeFi ecosystem suffers from **fragmented capital**, limiting traders' ability to efficiently deploy leverage. Leverage Protocol addresses this by introducing an on-chain, undercollateralized lending framework inspired by traditional margin trading systems.

The primary objective is to create a **capital-efficient layer** that aggregates liquidity and allows traders to use it to gain multiplied exposure in positions on *other* DeFi protocols.

Core Mechanism:

1. Users deposit an **initial margin**.
2. The protocol provides the additional capital required to achieve the desired leverage, sourcing these funds from a common liquidity pool.
3. This amplifies the trader's exposure using a fraction of the required capital.

By acting as an integration layer, Leverage Protocol leverages the inherent composability of blockchain technology, allowing it to connect with any compliant third-party DeFi protocol for synergistic trading experiences.

---

2. System Architecture

The protocol utilizes a **modular architecture**. Primary smart contracts are instantiated for each unique trading pair and integration, which effectively **isolates risk** between different asset markets and external protocols.

2.1 Liquidity Pool (ERC-4626 Compliant)

The Pool contract is the **liquidity backbone** of the protocol. It is a fully compliant implementation of the **ERC-4626 Tokenized Vault Standard**, offering a standardized interface for deposits and withdrawals.

Role	Function
Liquidity Providers (LPs)	Deposit the base asset (the "underlying," typically a stablecoin like USDC/USDT) and receive tokenized shares representing their pro-rata claim.

Role	Function
Yield Generation	LPs earn a **variable yield** generated from the interest and fees paid by traders who borrow funds.
Asset Management	Assets are exclusively managed by the associated **PositionManager** contract to fund trades.

2.2 PositionManager (The Core Operational Engine)

The PositionManager contract orchestrates all trading activities and is responsible for maintaining system **solvency**.

- **Position Management:** Handles the opening, closing, and modification of all trading positions. It withdraws the required capital (trader's margin + borrowed funds) from the Pool and executes the trade on the designated third-party protocol via an Adapter.
- **State Tracking:** Maintains a comprehensive record of each position's state (margin, leverage ratio, entry price, current debt, etc).
- **On-Chain Broker:** Technically owns the position on the external protocol, but the **o riginal trader retains full control** over the position's lifecycle.

2.3 Integration Framework (Adapter-Based Extensibility)

The protocol is designed for extensibility using a modular integration framework. For each third-party protocol, a new, specific instance of the PositionManager, Pool and a dedicated **Adapter contract** are deployed.

- **Goal:** Ensures the core protocol logic remains secure and isolated.
- **Adapter's Role:** Translates standardized commands from the PositionManager (e.g., `executeTrade`, `closePosition`) into the specific function calls and data structures required by the target protocol.

---

3. Core Mechanics & Position Lifecycle

3.1 Opening a Position

1. **Margin Deposit:** Trader deposits the underlying asset as margin into the PositionManager.
2. **Trade Parameters:** Trader specifies the asset, direction (**long or short**), desired **l everage** (e.g., 5x, 10x), and other necessary parameters.
3. **Execution:** PositionManager calculates total capital, **borrows the necessary amount** from the Pool, and executes the trade via the Adapter. The PositionManager becomes the custodian of the resulting assets.

3.2 Position Health & Liquidation

Position solvency is determined by its **Margin Ratio**, which is the current value of the trader's margin relative to its initial value. An **Off-Chain Keeper Network:** Monitors the Margin Ratio of all open positions.

Threshold	Description	Action Triggered
Margin Call Threshold	A warning level (e.g., 50% ratio).	Trader is notified of approaching liquidation risk.
Liquidation Threshold	The point at which the position is undercollateralized (e.g., 20% ratio).	Keeper can trigger the `liquidate()` function.

Liquidation Process:

1. `liquidate()` function triggers when the ratio drops below the threshold.
2. **On-Chain Safeguards** re-calculate the ratio using the asset price at the moment of execution to prevent faulty liquidations.
3. Position is closed on the third-party protocol.
4. Borrowed funds are **repaid to the liquidity pool**, and a **PROTOCOL_FEE** is taked from such funds. The rest is considered LP's profit as a compensation for risk assumption.

The **Margin Ratio** (position health) is fetched by the **PositionManager** from its dedicated **Adapter**. Consequently, the logic for calculating the health factor resides within the Adapter, which allows a high level of customization for each integrated protocol. The Position Manager only fetches this value and acts in consequence.

3.3 Interest Rate & Fee Model

Fees are incurred **only when a position is closed or liquidated**. The structure is designed to compensate Liquidity Providers (LPs) for their capital risk and fund the protocol's operation via its treasury. The fee calculation logic is separated into two primary scenarios: closing in profit and closing in loss.

Key Fee Parameters

Parameter	Description	Destination
FEE	Fixed percentage applied to the **Net Profit** or **Net Loss** of the closed position.	Shared between Protocol Treasury and Liquidity Pool.
PROTOCOL_FEE	Fixed percentage cut taken from the **FEE** amount.	Protocol Treasury (`fee_recipient` address).

Case 1: Position Closed in Profit ($\mathrm{P}_{\text{profit}} > 0$)

The fee is extracted as a percentage of the **Net Profit**, which is calculated as:

$$\text{Net Profit} = \text{Current Position Value} - \text{Initial Position Value}$$

1. **Fee Calculation:** A fixed **FEE** percentage is applied to the Net Profit to determine the Fee Amount.
2. **Protocol Share:** A **PROTOCOL_FEE** percentage is extracted from the Fee Amount and transferred to the `fee_recipient` address (Protocol Treasury).
3. **LP Yield:** The remaining portion of the Fee Amount is sent back to the **Liquidity Pool**, increasing the yield for LPs.
4. **Trader Payout:** The remaining profit ($\text{Net Profit} - \text{Fee Amount}$) is added to the user's available margin register.

Case 2: Position Closed in Loss ($\mathrm{P}_{\mathrm{loss}} > 0$)

The protocol assesses a fee based on the **Net Loss**, calculated as:

$$\text{Net Loss} = \text{Initial Position Value} - \text{Current Position Value}$$

1. **Loss Fee Calculation:** A fixed **FEE** percentage is applied to the Net Loss to determine the Fee Amount.
2. **Protocol Share:** A **PROTOCOL_FEE** percentage is extracted from the Fee Amount and transferred to the fee_recipient address (Protocol Treasury).
3. **Margin Subtraction:** The total amount subtracted from the user's available margin to cover losses is:

$$\text{Total Margin Subtracted} = \text{Net Loss} + \text{Protocol Fee Amount}$$

4. **LP Compensation:** The remainder of the loss ($\text{Net Loss} - \text{Protocol Fee Amount}$) goes back to the **Liquidity Pool**. This subtraction from the user's margin acts as compensation for LPs, mitigating the risk they assumed by backing the loan.

Corner Case: Loss Exceeds Available Margin (Black Swan Event)

> **This logic should never be executed** under normal conditions, as keepers are mandated to liquidate positions before the Net Loss exceeds the user's available margin (including fees).
>
> **However**, in the highly unlikely event of a keeper failure or extreme traded asset volatility resulting in a loss greater than the collateral:
>
> 1. All available position collateral is seized to minimize protocol losses.
> 2. **NO protocol fees are taken** in this scenario. The entire remainder is used to repay the Liquidity Pool.

In a standard liquidation event (where loss does **not** exceed margin), the trader's remaining margin (after accounting for Net Loss and Fee Amount) is treated as a final margin return, ensuring LPs are fully repaid.

---

4. Use Cases & Composability

The true strength of the Leverage Protocol is its ability to interact with the broader Starknet DeFi ecosystem:

- **Leveraged Spot Trading:** Deposit 1,000 USDT to open a **5x leveraged long on wBTC** by borrowing an additional 4,000 USDT and executing the swap on a DEX like Ekubo.
- **Leveraged Staking:** Deposit ETH as margin to borrow more ETH, staking the **tot al amount** in a liquid staking protocol to **amplify rewards**.
- **Enhanced Derivatives Exposure:** Borrow assets to increase the size of positions on derivatives protocols (futures, options, perps…) for more complex and capital-efficient hedging or speculative strategies.

---

5. Risk Analysis

A comprehensive understanding of the risks is paramount.

Risk Cate gory	Description	Mitigation Strategy
Third -Part y Prot ocol Risk	Inherited risks from integrated external DeFi platforms (e.g., a vulnerability in an integrated DEX).	Risk is **isolated** by deploying separate PositionManager and Adapter contract instances for each integration.
Orac le Risk	Protocol solvency depends on **accurate and timely price data** (sources variates depending on implementation, could be a AMM TWAP, a current price of a pool, a price feed from Chainlink or another provider, etc.). A manipulated or lagging oracle could lead to improper liquidations.	Oracle manipulations are a common vector attack on DeFi ecosystem. It’s important to consider this during integration design and data source selection.

Risk Category	Description	Mitigation Strategy
Liquidity Risk	Sudden, large withdrawals by LPs could reduce available funds for borrowing, potentially affecting traders' ability to open new positions.	This risk is inherent to pooled lending.
Pool's funds freezing	A malicious actor can open a position and then never close it, making a part of the total funds on the pool impossible to withdraw by the LPs.	Each position has a `deadline` parameter, when such deadline is reached, any one can close the position.

---

6. Future Work

The initial release is a foundational layer. The roadmap includes:

- Integration with a **wider range** of Starknet DeFi protocols.
- Support for **multi-asset collateralization**.
- Development of a decentralized **governance structure (DAO)** to manage protocol parameters, risk settings, and new asset integrations.
- Research into **cross-chain margin trading** capabilities.
- Implement a **time based interest rate** to generate extra yield for LPs.
- Use unutilized LP's deposits and unutilized traders' margin deposits to generate passive extra yield. For example, could be deposited in an AMM pool the get LP rewards.

---

7. Conclusion

Leverage Protocol introduces a fundamental financial primitive to Starknet, enabling **undercollateralized lending for margin trading** in a decentralized and composable manner. By creating a unified framework for leverage, the protocol enhances capital efficiency and empowers traders with more sophisticated tools. With a strong focus on

security and modularity, Leverage Protocol is poised to become a **core building block** in the future of Starknet DeFi.