

SoLever: A Unique Possibility in Leverage Restaking

SoLever Team

September 2024

Abstract

Restaking platforms of any real value necessarily create the possibility for slashing of operators or those delegating stakes to them. Liquid restaked tokens (LRTs) make these delegations fungible, but are consequently sensitive to the operators that they trust getting slashed. At the same time, significant TVL could be driven into LRTs through leverage restaking, but the commonly-used platform on Ethereum does not account for fundamental realities of restaking and consequently introduces both unnecessary risks and limitations for borrowers, disincentivizing leverage restaking. On Solana, no platform that allows for leverage restaking exists in the first place. We propose a new leverage restaking platform which is particularly suited for Solana and accounts for the fundamental realities of restaking to make doing so incredibly lucrative for both lenders and borrowers.

1 Background

1.1 The Need For Restaking (Briefly)

While blockchains like Ethereum and Solana allow for many different kinds of smart contracts to be secured on them, one limitation is that all these smart contracts must be deterministic[10][11]. That is, all operations performed on the EVM or SVM must have the same outcomes for everyone who performs them. This allows all validators to agree on the state of the virtual machine based on the accepted transactions, but precludes things like external price feeds or information about other blockchains from being incorporated directly into smart contracts. Applications that require such external data sources initially resorted to a single trusted source of truth (e.g. having only a trusted party with a particular key pair provide this information via transactions that only it can perform). However, this diminishes claims to decentralization: what would happen if a single, trusted oracle provided bad data? To address this Chainlink in 2017 pioneered oracle networks—networks of many providers of particular pieces of information, like price of an asset, where operators have stakes that are slashed in case their information conflicts with the others'[5]. However,

there is a need to extend this: new protocols may need new oracles around new information, or nodes for bridges, or similar things. For all this, they would ordinarily need to build a custom-purpose “trust network” themselves. To alleviate this problem, and to further secure such services (so-called “actively-validated services” or AVSs), Eigenlayer abstracted the task of running all these different AVSs so that anyone could create an AVS and operators could opt in to running any number of these AVSs that they want[4]. These operators could use the same stake to secure multiple AVSs at the same time. And, anyone could delegate their own funds to an operator that they trusted; the delegated stake would be lost in the case that the operator misbehaved, but otherwise a fraction of the yield paid by the AVS would go to the person delegating the stake. On Solana, several platforms have followed in the footsteps of Eigenlayer: namely, Solayer[7], Cambrian[2] and Jito[6].

1.2 *Liquid Restaking*

One issue with directly restaking with operators is liquidity of the restaked funds—once you restake, it may take time to unstake and you can’t do anything with that locked value in the meantime[8]. One way to address this could in principle be through tokenization of the restaked position, so instead of unstaking, someone who had previously restaked could just sell tokens representing ownership of the restaked assets and generated yield. However, two people who delegated to different operators would have incomparable positions: the operators may be supporting different numbers of different AVSs, and may have different degrees of reliability (e.g. one could have a more consistent internet connection or better hardware than another). For a tokenized position to be fungible, every token must represent the same position—delegation to the same operator(s) running the same AVSs.

The practical implementation of these fungible liquid restaked tokens (LRTs) involves all depositors putting their funds into the same pool that is managed in an opinionated way; different LRTs may trust different operators or AVSs. For this reason, restaking platforms like Eigenlayer (including Solayer, Cambrian, etc) cannot directly run their own LRTs, because that requires them to make opinionated decisions about which operators and AVSs they trust, destroying any claim they may lay to decentralization.

LRTs have been incredibly popular: at the time of writing, Eigenlayer has a TVL of \$11B USD[3] whereas the top 22 Ethereum LRTs on DefiLlama have a combined TVL of \$9.7B USD[9]—meaning that the vast majority of the restaked TVL comes from LRTs.

1.3 *Leverage platforms and LRTs*

Lending platforms like Aave allow users to borrow assets without the platform needing to trust them, using the requirement that the borrowers first put down more in collateral than they intend to borrow[1]. A common use case for this

was to gain temporary access to assets in one token while having assets in a different token, without needing to sell that other token.

Platforms like this have been used as well for ‘looping’[12], whereby a user puts down collateral in asset A, borrows asset B, trades the borrowed B for more A, and then uses that A as collateral to repeat the process. In this way, users can essentially take leveraged positions.

To demonstrate the utility of this in the context of leverage trading, consider first that a user trades asset B for asset A, and then the prices of both change respectively from $B(t)$ and $A(t)$ to $B(t + \Delta t)$ and $A(t + \Delta t)$. In this case, the trader changed what he started with by a factor of

$$\frac{B(t)A(t + \Delta t)}{A(t)}$$

On the other hand, through looping, the corresponding factor can be approximated by

$$\frac{B(t)A(t + \Delta t)}{A(t)} \sum_{n=1}^{\infty} LTV^n = \frac{B(t)A(t + \Delta t)}{A(t)(1 - LTV)}$$

Where LTV is the loan-to-value ratio, which is less than 1. Here, we make the assumption that the protocol and transaction fees are zero, and hence it is ideal to loop infinitely. In all cases where the value of A increases with respect to B, (since $0 < LTV < 1$), the trader stands to gain more through looping than directly trading.

In the case of liquid restaking, in the case that yield is paid through an increasing price of the LRT itself, a user could earn yield with leverage through looping where asset B is the blockchain native token (for example, ETH) and asset A is the LRT (for example, ezETH).

However, this doesn’t work in cases where users are rewarded specifically for holding the LRT, for example in the case of point campaigns or when holders are paid yield directly. In these cases, users turned to a platform called Gearbox, which allowed for borrowing in a different way: instead of simply giving assets to a borrower, and requiring overcollateralization, Gearbox allowed undercollateralized positions by simply taking the position on behalf of the user. For example, in the case of leverage restaking, the Gearbox contracts would restake lenders’ money on behalf of the borrower, and, in the case of a price drop of the LRT, liquidate it (sell the LRT to liquidators, and pay the lenders and liquidators out of the borrower’s collateral).

2 Problems With This

In the case that the operators to which LRTs delegate get slashed, the assets underlying the LRT itself are lost; and logically this should cause a drop in the price of the LRT. For this reason, LRTs are not risk-free: direct restakers, who restake into the LRT with their own funds, are at risk in the case of slashing.

Similarly, lending platforms that make 1:1 assumptions between the price of an LRT and the native underlying asset put lenders at risk. Instead, lending or leverage platforms need a metric for the health of an LRT—commonly, they use the price of the LRT, since this fits well with the lending and borrowing models for leverage trading that were described previously.

2.1 Volatility, Danger For Leverage Restakers

Since leverage platforms often use the prices of LRTs as metrics for the health of the token, just as in the case of other assets that are margin-traded (meaning that positions are liquidated once the value of the position drops below a certain point), leverage restakers take on two risks: one from the possibility of slashing; and another from the possibility of volatility. And this volatility is particularly problematic given that in many DEXes, the liquidity of restaked assets is relatively shallow (relatively, to the underlying native token).

Already, there have been several cases where assets that were fundamentally sound (meaning LRTs or LSTs whose operators or nodes were not slashed) temporarily dropped in price, causing borrowers or leverage restakers to lose huge amounts of money. Two notable examples, respectively on Ethereum and Solana, are the ezETH de-peg and its effects on Gearbox, and the mSol de-peg and its impacts on Marginfi.

3 SoLever: Slashing-proof LRTs, Volatility-Proof Leverage Restaking

We propose a leverage platform specifically for LRTs that mitigates the aforementioned problems.

The leverage platform described here is in the style of Gearbox, whereby a leverage restaker puts down collateral and a smart contract restakes an amount of lenders’ money on the users behalf. The smart contract then holds onto the LRTs it receives for this restaking action, which we call the ‘position’ here. Leverage restakers can close their position at a later time and claim the underlying SOL or LSTs and accrued yield, less the principal lent out by the lenders and the interest that the lenders charge.

3.1 Modified liquidation conditions

Traditional leverage platforms trigger liquidation in the case that the price-based value of a position drops so that the loan-to-value ratio is below a particular point. This point is set so that, in the time between the eligibility for liquidation and the liquidation itself, the price is unlikely to change drastically enough to cause a loss that cannot be covered by the borrower’s collateral.

Instead, we propose a leverage platform specifically for leverage restaking, that liquidates based on slashing rather than volatility. Specifically, when an operator trusted by a Jito Vault or LRT is slashed, a known amount of the

restaked assets underlying the LRT or VRT is lost—a known fraction of a known TVL in that LRT platform—leading to a known change in value of the LRT.

3.1.1 Definitions of important quantities and relations

Here, the term ‘value’ is used and considered to be distinct from ‘price’ in that price is volatile and may depend on which exchanges are considered together with random market conditions (for example, a whale dumping a huge amount of an LRT on a decentralized exchange), whereas the value of an LRT is based on the underlying unslashed stake. In our proposed leverage platform, liquidation only happens in the case of slashing: positions are fractionally liquidated as their value changes due to slashing, and borrowers’ collateral is used to cover these changes in value.

In the proposed leverage platform, the number of liquid restaked tokens held for a particular position at time t is $N(t)$, the amount of restaked value underlying that position is $R(t)$, and the total yield earned at time t since the position was opened is $Y(t)$. We say that the value of the position is *just* $R(t)$, denominated in whatever token was lent out to leverage-restake. This implies that the value of the LRT (per unit) is $V(t) = R(t)/N(t)$, in which case the prospective yield plays no role—this can be justified across the market by saying that the LRT in question is in a risk category of its own and its yield is the minimum acceptable rate of return for its own risk category (so its present value just remains the same for all times).

Note that R is time-dependent because it changes with slashing and partial liquidation, and Y is time-dependent because it is the total yield accrued by a certain time—it is monotonically increasing with time.

In the proposed leverage platform, both yield and collateral are sort of kept hostage—they can be used to repay lenders for losses due to slashing events. The term ‘hostage’ is used to distinguish from collateral, which is the borrower’s personal money put into the leverage platform in order to borrow. The amount that the platform has ‘hostage’ is, at a time t **before any slashing**,

$$H(t) = C(t) + Y(t) - I(t) - F_p(t)$$

Where $I(t)$ is the total accrued interest expected by the lenders as of time t , and $F_p(t)$ is the total accrued protocol fees expected by time t .

When slashing happens, a fraction s of the TVL underlying the LRT or VRT is lost, meaning that, if t_{before} is the time before the slashing and t_{after} is the time after slashing, R changes as:

$$R(t_{after}) = R(t_{before}) \cdot (1 - s)$$

Correspondingly, if the previous (prior to slashing) loan-to-value ratio of a particular leverage position is $LTV = R(t)/C(t)$, the fraction of the hostage lost to just repay the lenders is $\Delta H = H(t_{after}) - H(t_{before})$:

$$\Delta H_{slashing} = \Delta R = -s \cdot R(t_{before}) = -s \cdot C(t_{before}) \cdot LTV(t_{before})$$

With this, we can write the following expression for H that is always true:

$$H(t) = C(t) + Y(t) + \Delta H_{slashing, total} - I(t) - F_p(t)$$

In this way, we always have that

$$R(t) + Y(t) + C(t) - H(t) - F_p(t) = R(t) - \Delta R_{total} + I(t) = R(t_0) + I(t)$$

where $\Delta H_{slashing, total} = \Delta R_{total}$ is the total change in the H over many slashing events—note that $\Delta H_{slashing, total} < 0$. This says that the sum of the value of the position’s LRTs and their earned yield, less the hostage owed to the leverage restaker and the protocol fee, is always the initial amount borrowed from the lenders plus the expected interest accrued by that time.

Of course, this can only continue to a point: it is possible that not enough is held hostage to cover potential slashing events. Algebraically, this happens when:

$$\frac{H(t)}{R(t)} < s$$

The potential slashing exposure of LRTs can be seen from the operators that they rely on, and the policies of the AVSs that those operators support. For example, an assumption can be made that only one operator at a time will misbehave out of all the operators that an LRT relies on, and *market liquidation* has enough time to take place between two slashing events. The operator that could cause maximal s (meaning a maximal product of (fraction of stake underlying the LRT that is delegated to this operator) and (sum of slashing fractions of all AVSs that that operator supports), which is maximized with respect to the different operators supporting the LRT) gives an upper-bound estimate for s .

In the case that this condition (that $H(t)/R(t) > s$) is violated, it can be restored by selling off, or *liquidating*, a fraction of the LRT that the platform holds. Note that this condition is complicated by the fact that lenders also expect interest paid, the leverage protocol is paid a fee, and liquidation conditions should also ensure that the lenders and the protocol can always be repaid the principal, the interest, and the fees that they expect.

To express a strategy to avoid this, we define a owed-to-hostage ratio, or OTH , is defined as the ratio of value owed at a given time to the amount of hostage H remaining at that time. The amount owed includes the total accrued interest charged by the lenders $I(t)$ at a given time, the total accrued protocol fee $F_p(t)$, and the value of the position that is at risk due to slashing:

$$OTH(t) = \frac{R(t)}{H(t)} = \frac{R(t)}{C(t) + Y(t) - I(t) - F_p(t)}$$

Note that C is time-dependent because it decreases due to partial liquidations and increases when a leverage restaker adds collateral.

The OTH can be compared to LTV commonly used in other protocols. When this drops below an OTH_{max} specified for the asset, the position is partially liquidated to restore the OTH to what is acceptable.

3.2 Modified liquidation mechanism

‘Liquidation’ is divided into two categories: one which occurs due to slashing, and doesn’t require any actual sale on the open market—this one is called ‘slashing compensation’ here; and another which happens in the case that OTH gets too high—this one is called ‘market liquidation’ here.

3.2.1 Slashing compensation

At its core, when slashing happens, there is simply a decrease in the value of the LRTs owned by the platform, and lenders are compensated from the collateral of the borrower. This doesn’t actually require any immediate transactions to take place, as it simply means that the C used for other calculations is different (including the calculation for return of collateral when the position is closed).

Generally, this means reading from the contracts involved in AVS-specific slashing for all later calculations to check what slashing has been performed.

3.2.2 Market liquidation

Once the OTH of a given position exceeds its OTH_{max} , that position becomes eligible for market liquidation. From a smart contract perspective, this liquidation is performed by liquidators who initiate the liquidation transaction and pay for fees as per Gearbox—the OTH requirement mentioned above is a check performed by the smart contract at the time that the liquidation transaction is executed.

The mechanism of this liquidation is, similarly to Gearbox, liquidators purchasing the liquidated LRTs at a discounted rate.

The amount sold off to liquidators is (if t is the time before liquidation):

$$-\Delta N = \begin{cases} N(t) - \left(H(t) - l_p\right) \frac{OTH_{max}}{V(t)} & H(t) \geq 2l_p \\ N(t) & H(t) < 2l_p \end{cases}$$

This is sold off at a discounted rate—we let l_p be the premium that essentially goes to liquidators in the form of a discount, so the amount for which they can buy the given $-\Delta N$ is $-\Delta NV(t) - l_p$ (note that ΔN is negative). A liquidator could do whatever they want with this, and anyone can be the liquidator in question. In principle, the liquidator could just turn around and sell the LRT on a decentralized exchange to make a profit l_p . All the $-\Delta NV(t) - l_p$ goes back to the lender pool.

When this liquidation happens, H is decreased by l_p .

3.3 Closing the position

What is meant by closing the position in this case is the protocol either un-restaking the LRT associated with a particular position or market-selling them.

In this case, the borrower is paid back $H(t)$.

The protocol is paid $F_p(t)$. The way this may be calculated is, for example, using a constant rate $r_p \in (0, 1)$

$$F_p(t) = \int_{t_0}^t r_p R(\tilde{t}) d\tilde{t}$$

Finally, the lender is paid back:

$$\begin{aligned} & R(t) + Y(t) - H(t) - F_p(t) \\ &= R(t_0) + I(t) \end{aligned}$$

An example of the way $I(t)$ could be calculated is, setting a fixed simple interest $i \in (0, 1)$,

$$I(t) = \int_{t_0}^t i R(\tilde{t}) d\tilde{t}$$

note that every time market liquidation happens, the l_p decrease in H means that those funds are returned to the lenders in the end.

4 Key players in the proposed model and how they benefit

With the description of the mechaism given above, it may be clear what the benefits are of the proposed changes, but the reader may still be left with the question of, “why would someone use *this* platform?” To answer that question, this section goes over all the actors in our model and why they benefit.

Before going there, it should be noted that at the time of writing, **no platform on Solana meets the need for leverage restaking**. This is because the platforms on Solana, namely Solend and Marginfi, allow for looping as mentioned in Section 1.3. However, the proposed LRTs on Solana (the big one being Fragmetric) intend to reward users directly for holding the token—the looping would lead to all yield being paid to the smart contract. A platform such as what is proposed is needed anyway for leverage restaking on Solana.

That being said, here are the actors and their motivations.

4.1 Leverage restakers

Leverage restakers, also known as borrowers on other platforms, expect that the LRT in question will pay more yield over a long time scale than it will lose in slashing. These leverage restakers potentially earn a yield on their collateral that’s been hugely multiplied, less a constant interest that the lenders charge. For example, if a particular LRT has an expected maximum slashing of 2% at a time, the OTH_{max} may be as high as 50—meaning an initial loan-to-value ratio of 50x! If the LRT normally pays 5% yield per annum on the restaked amount, a leverage restaker using our platform would earn 250% interest on the leverage restaking investment per annum (less some interest that the lenders charge)!

4.2 Lenders

Lenders want a fixed interest on the assets they lend out, without any risk due to the slashing etc that decreases the value of the LRT. These lenders want to be sure that whatever they lend out, they will get back with interest (denominated in an asset that doesn't get slashed). Essentially, these lenders pay for security with the fact that their interest is fixed and independent of the yield paid by the LRT (and we expect it to be slightly smaller than the LRT yield in order to keep leverage restakers incentivized).

4.3 VRT/LRT creators

The people who launch LRTs and VRTs want to maximize the TVL in these platforms. Through a leverage restaking platform, the large value brought by people who would otherwise be too risk-averse to invest (these are the people who end up as lenders) can end up as TVL in these LRTs—this is a potential TVL multiplier for LRTs.

References

- [1] *AAVE Protocol Whitepaper V1.0*. 2020. URL: https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf.
- [2] *Cambrian Introduction*. URL: <https://docs.cambrian.one/> (visited on 09/10/2024).
- [3] *Eigenlayer - DefiLlama*. URL: <https://defillama.com/protocol/eigenlayer#information> (visited on 09/10/2024).
- [4] *EigenLayer: The Restaking Collective*. URL: https://docs.eigenlayer.xyz/assets/files/EigenLayer_WhitePaper-88c47923ca0319870c611decd6e562ad.pdf (visited on 09/10/2024).
- [5] Steve Ellis, Ari Juels, and Sergey Nazarov. *ChainLink: A Decentralized Oracle Network*. Tech. rep. ChainLink, 2017. URL: <https://research.chain.link/whitepaper-v1.pdf>.
- [6] Jito Foundation. “Announcing Jito (Re)staking, A Next-Gen Infrastructure Platform”. In: *Jito Network Blog* (2024). URL: <https://www.jito.network/blog/announcing-jito-restaking/>.
- [7] *Introduction*. URL: <https://docs.solayer.org/getting-started/introduction> (visited on 09/10/2024).
- [8] Stader Labs. “Liquid Restaked Token — A new DeFi primitive”. In: *Medium* (2023). URL: <https://blog.staderlabs.com/liquid-restaked-token-lrt-a-new-liquid-staking-paradigm-4ef0fc002c2d>.
- [9] *Liquid Restaking TVL Rankings - DefiLlama*. URL: <https://defillama.com/protocols/Liquid%20Restaking/Ethereum> (visited on 09/10/2024).

- [10] *Oracles*. URL: <https://ethereum.org/en/developers/docs/oracles/#why-do-smart-contracts-need-oracles> (visited on 09/10/2024).
- [11] *Oracles and Oracle Networks*. URL: <https://solana.com/developers/courses/connecting-to-offchain-data/oracles> (visited on 09/10/2024).
- [12] *What is looping?* URL: <https://medium.com/contango-xyz/what-is-looping-78421c8a1367> (visited on 09/10/2024).