Huge thanks to @Mol_Eliza for the co-authoring and review

This is a supplementary note for the original CSM proposal

The main goal of this note is to outline one of the possible approaches to capital efficiency estimation. Thanks to @micho for pointing out the importance of these calculations during LidoConnect in Istanbul.

To have a metric to compare bonding designs, sizes, and fee structures for different protocols, let's consider two metrics: "rewarded capital" (RC)

and "rewarded capital multiplier" (RCM)

.

RC

RC

(rewarded capital) signifies the equivalent capital required for vanilla solo-staking to yield commensurate rewards compared to employing a staking provider solution.

To illustrate the RC concept, consider a full Ethereum validator demanding a stake of 32 ETH. Envision a scenario where a staking provider uses only users' ETH and charges a 10% fee on staking rewards. This translates to a provider claiming 10% of rewards generated by a 32 ETH capital or 100% of rewards generated by 3.2 ETH. This quantity is denoted as RC for a staking provider. Now, envision a staking provider contributing 8 ETH out of the required 32 ETH, retaining all rewards from their 8 ETH, and charging a 5% fee on the remaining 24 ETH. In this instance, the RC for a staking provider is calculated as 8 * 1 + 24 * 0.05 = 9.2 ETH

.

The basic formula for RC (full bond rewards + fee from external capital) is:

RC_generic = Bond + (32 - Bond) * MaxFee

If we assume a possible CSM approach with bond converted to stETH

, the RC formula should be changed to account for the different fee mechanics (regular stETH

rewards on bond + fee from entire 32 ETH):

RC_csm = Bond * (1 - LidoStakingFee) + 32 * MaxFee

To account for additional forms of bond capital not directly connected to ETH, let's name it BondALT

, some form of valuation of normalization of yield in ETH may be introduced in the form of the K

coefficient. With this coefficient, the initial formula will take the form of:

RC_alt = BondETH + (32 - BondETH) * MaxFee + BondALT * K

where BondALT * K

- equivalent ETH amount with vanilla staking APR providing same rewards. K > 1

means that additional bond capital form is more rewarding than vanilla ETH staking, and with K < 1

- additional bond capital is less rewarding.

Note that the actual formula of the K

coefficient depends on the actual alt token mechanics and should be defined individually for each case.

RCM

RCM

(rewarded capital multiplier) establishes a relationship between RC

and the actual provided capital. Using the previous example where RC is 9.2 ETH, and the provided capital is 8 ETH, the RCM

in this scenario is calculated as 9.2 / 8 = 1.15

.

Alternative bond example. RPL

As an illustration of the K

coefficient and for future RCM comparison, let’s consider RPL (the most developed and well-documented token on the market).

The RPL bond portion of the minipools bond is measured in ETH. To mitigate the impact of unpredictable ETH/RPL exchange rates on our calculations, we employ Relative Purchasing Power Parity (RPPP). Adhering to this principle, our focus is on comparing the inflation rates between the two. Following “The Merge”, ETH inflation has exhibited a relatively stable trend, hovering around 0%. In contrast, RPL inflation is set at a fixed rate of 5% annually. RPL staking APR can be recalculated into ETH staking APR using the following formula:

RPL_StakingAPR_in_ETH = (1 + RPL_StakingAPR) * C_parity - 1

where

C_parity = (1 + ETH_inflation) / (1 + RPL_inflation)

While the actual value for the C_parity = 1 / 105 ~= 0.952

, RPL_StakingAPR

depends on the amount of RPL staked in the following way:

RPL_StakingAPR = RPL_inflation * RPL_inflation_allocation_to_stakers / RPL_staked_percent

The RPL_inflation = 0.05

and RPL_inflation_allocation_to_stakers = 0.7

(see rewards section). The only variable in the formula above is RPL_staked_percent

.

Given all the above and taking ETH_StakingAPR = 0.04 (or 4%)

, one can get the following:

It is clear that for a low percentage of RPL staked, the resulting RPL_StakingAPR

is higher than ETH_StakingAPR

. At the same time, if more than 38% of RPL is staked, the resulting RPL_StakingAPR

is lower than ETH_StakingAPR

.

Coefficient K

, in this case, can be defined as:

K = RPL_StakingAPR_in_ETH / ETH_StakingAPR

At the moment of writing, approx. 48% of RPL’s total supply isstaked. Hence,

RPL_StakingAPR_in_ETH = 0.022 (or 2.2%)

Substituting actual values, one can get:

K = 0.022 / 0.04 = 0.55

RCM Comparison

Given all the above at the moment of writing, numbers are the following:

Project

RP LEB8

Stader *

Swell **

CSM (4) ***

Bond ETH

8

4

16

4

Additional bond

2.4

0.4

0

0

APR ratio (K)

0.55

5.82

0

0

MaxFee

14%

6%

10%

7.5%

RC

12.68

8.01

17.6

6

RCM

1.22

1.82

1.1

1.5

-
    - SD tokenomics and staking APR are not transparent

** - permissionless access is not available yet

*** - actual bond size and staking fee are not yet approved by the Lido DAO

As one can see from the table, at the moment of writing, Rocket Pool offers LEB8 minipools with an RCM of 1.22

, Stader offers permissionless staking with a 1.82

multiplier (number is super high mostly due to enormous SD APR that does not seem sustainable), and Swell does not have permissionless staking live yet but has plans on launching with a 1.1

multiplier.

Project

RP LEB4

RP LEB4 (no RPL)

Stader (normal SD)

CSM (2) *

Bond ETH

4

4

4

2

Additional bond

2.8

0

0.4

0

APR ratio (K)

0.55

0

1

0

Max fee

14%

14%

6%

7.5%

RC

9.46

7.92

6.08

4.2

RCM

1.39

1.98

1.38

2.1

-
    - actual bond size and staking fee are not yet approved by the Lido DAO

Also, looking into the future, one can assume that Rocket Pool will launch LEB4 minipools (1.39

) or even LEB4 with no RPL requirement (1.98

). At the same time, SD rewards will probably end up on par with ETH_StakingAPR

or even lower, resulting in Stader RCM reduction to 1.32

(still very compatible!). To ensure CSM positions in the hypothetical future staking market, a 2 ETH bond can be introduced, resulting in RCM of 2.1

.

Complete calculations can be found here