A new approach for OLAS top-ups (part of AIP-1)

Mariapia Moscatiello, Aleksandr Kuperman,
Andrey Lebedev, David Minarsch

22 September 2023

Contents

1 TL;DR

This document introduces a novel method for distributing OLAS top-ups to incentivise useful code based on their donors' veOLAS holdings. This approach encourages increased participation in locking OLAS in veOLAS, ultimately enhancing the ecosystem's robustness and security.

2 Introduction

The document introduces a novel method for distributing top-ups to incentivize users to lock more OLAS in veOLAS. This mechanism serves two primary objectives:

- Users who lock more OLAS can boost more useful code.
- Donors can boost up to an amount of top-up equal to their veOLAS holdings. This ensures that no useful code can receive more top-ups than the total veOLAS held by the donors of the relevant service.

The proposed mechanism, as outlined in Section 4, aims to enhance the distribution of top-ups for useful code by incorporating the veOLAS holdings of donors as weighting factors in the top-up formula. This is different from the current design, described in Section 3, where the formula relied on the amount of ETH donations. Here's how the new mechanism functions:

- Calculate the total amount of inflation reserved for code.
- Aggregate the veOLAS holdings of all donors from their donation transactions.
- Distribute the total top-ups to code in proportion to their veOLAS amounts of relevant donors.

Note. We do not mention anything about ETH rewards, as rewards are staying intact and follow the same logic as before for each donation.

Warning. To minimize alterations to the current design, preserve contract logic, and avoid significant impacts on gas costs, we present the simplest solution that is resistant to manipulation. The following set of actions is contemplated for this simple approach:

- Donors with veOLAS can use their veOLAS holdings only for the first donation transaction of each epoch in order to trigger top-ups. Specifically,
  - When a donor makes a donation to services s1, s2, ..., st in a unique transaction during the epoch, the veOLAS amount (at the beginning of the epoch) will be equally distributed among the code in these services.
  - If a donor makes an additional donation to any service, this transaction will solely trigger rewards in ETH and not top-ups.

  Accounting for multiple donations with boosts would be too complicated from an implementation perspective.
- Service owners who do not make donations will not activate OLAS boosts.

It's important to highlight that the ideal approach for boost, outlined below, necessitates substantial modifications to our current design hence is currently regarded as a backup option.

The ideal approach (backup-option) can be summarized as follows. Donors holding veOLA can choose the portion of their holdings allocated for boosting during each epoch. When their boosting reaches a cumulative total equal to their entire veOLAS holdings, any additional donations made during that epoch will exclusively trigger ETH rewards.

This arrangement woul be advantageous because, in this scenario, throughout the epoch, regardless of the number of donation transactions made by the veOLAS holder, they can consistently boost up to their total holdings. In contrast, in the current setup, only the initial donation transaction within the epoch permits the veOLAS holder to activate their boost for the service involved in that specific transaction.

3 Current design

Let's focus on components, the arguments are symmetric for agents. We recall that a component is considered useful if, during an epoch, any of the services in which the component is referenced received a donation in ETH. Useful components are rewarded with a share of ETH received by the services referencing them. Moreover, whether the donator or service owner owns a certain amount of veOLAS, the useful component receives a share of the OLAS inflation. Currently, there is a threshold over which anyone can trigger a large amount of OLAS top-ups. This should be improved for the following reasons. first of all, users are incentivized only to lock at 10k veOLAS to trigger OLAS top-ups, which is an arbitrary number and potentially too low. Secondly, holding only 10k veOLAS would allow triggering all the top-ups of the epoch with just a minimal ETH donation.

Currently, we weight both the ETH rewards and the top-ups for useful code by using the corresponding fraction amount of ETH received.

Specifically, if during epoch e services $s_1, \cdots, s_t$ received respectively $r_1, \ldots, r_t$ as donations, the current formula of the reward for the owner of the component c_i is the following:

$$Rew_{c_i}(e) = \text{componentFraction(e)} \cdot \sum_{1 \le j \le t} \frac{cit^{c_i}_{s_j}(e) r_j(e)}{\text{numComponents}(s_j)}. \quad (1)$$

Now, let assume that either the service owner or donator of the services s_j for $j = 1, \ldots w$ owns at least 10k veOLAS and let $\text{OLASTopUp}_C(e)$ be the total amount of OLAS inflation that during the e-th epoch is reserved to fund useful code top-ups. Then the top-ups for the component c_i during the e-th epoch can be calculated as follows.

$$topUp_{c_i}(e) = \frac{\text{OLASTopUp}_C(e)}{\sum_{1 \le j \le w} r_j} \sum_{1 \le j \le w} \frac{cit^{c_i}_{s_j}(e) r_j}{n_c(s_j)}. \quad (2)$$

For more details on the current design, see Autonolas Tokenomics v1 and Autonolas Tokenomics repository documentation.

4 New mechanism design

For top-ups, in order to incentivise users to lock more, instead of weighing by using the amount of ETH received by a service, as outlined in Section 3, we can

weight by using the amount of veOLAS holdings of the donators or the service owner as discussed below.
We focus on components, the proposal is symmetric for agents. Let

$$TopUpComp(e) = \text{compFrac(e)} \cdot \text{TopUps(e)}$$

be the amount of inflation reserved to fund useful components top-ups during the e-th epoch. Further, let assume that the service s_j received k_j donations made by **different** donators during the epoch, and let

$$v_j = \sum_{1 \leq n \leq k_j} \frac{v_{j,n}}{m_n},$$

be the sum of donators' holdings, $v_{k,n}$. The m_n is used to weight the donator holdings for the number of service they donated to into the same donation transaction. E.g. if donator with voting power $v_{1,j}$ donated to service s_j and s_l with a unique transaction, $m_1 = 2$. This implies a donator can at most boost $v_{1,j}$ per all the service they donated to the epoch. Then the new top-up formulas can be as follows:

$$top_{c_i}(e) = \begin{cases} \frac{TopUpComp(e)}{\sum_{1 \leq j \leq t} v_j} \cdot \sum_{1 \leq j \leq t} \frac{cit^{c_i} s_j(e) v_j}{\text{numComp}(s_j)}, & \text{if } \sum_{1 \leq j \leq t} v_j \geq TopUp(e) \\ \text{compFrac(e)} \cdot \sum_{1 \leq j \leq t} \frac{cit^{c_i} s_j(e) v_j(e)}{\text{numComp}(s_j)}, & \text{if } \sum_{1 \leq j \leq t} v_j < TopUp(e) \end{cases} \tag{3}$$

Remark. Note that a donator holding $v_{j,k}$ can at most boost reward with $v_{j,k}$ OLAS during one epoch. Specifically, $v_{j,k}$ will be accounted only once for an epoch. Notably, the first donation transaction of the aforementioned donator, trigger a 'distribution' of $v_{j,k}$ for all the services included in such a donation transaction. While any subsequent donation transaction from the same donator, will only trigger ETH rewards.

Example 1. We have the following useful[1] services s_j, agents a_t, components c_i, and donations r_i.

$$s1 = [a1, a1 = [c1, c2]], \text{ the donator (resp. service owner) owns } v_{1,1} = 100$$
$$s2 = [a2, a2 = [c2]], \text{ the donator (resp. service owner) owns } v_{2,1} = 1000$$
$$s3 = [a3, a3 = [c2, c3]], \text{ donator owns 0 veOLAS}$$
$$s4 = [a4, a4 = [c4, c5, c6]], \text{ donator owns 0 veOLAS}$$

[1]Recall that by "useful service" we mean that the during an epoch at least a donation was made for such service

Then, $v_1 = 50, v_2 = 1000$. Assuming that, $ToUpComp(e) < v_1 + v_2$, then

$$
\begin{aligned}
top_{c_1} &= \text{compFrac(e)} \cdot \frac{v_1}{2} = \text{compFrac(e)} \cdot \frac{100}{2} = \text{compFrac(e)} \cdot 50, \\
top_{c_2} &= \text{compFrac(e)} \cdot \left(\frac{v_1}{2} + v_2\right) = \text{compFrac(e)} \cdot 1050, \\
top_{c_3} &= 0, \\
top_{c_4} &= 0.
\end{aligned}
$$

Example 2. Let's consider services s_1 and s_2 and their respectively components as described in the previous example. Let's assume that $TopUpComp(e) = 50000$. Moreover, let's assume that, during the same epoch, the service s_1

- received first a donation $r_{1,1} = 100$ ETH by a donator with $v_{1,1} = 30000$ veOLAS,
- received first a donation $r_{1,2} = 1$ ETH by a donator with $v_{1,2} = 10000$ veOLAS

And, the service s_2

- received a donation $r_{2,1} = 10$ and $r_{2,2} = 1$ by the same donator owning $v_{2,1} = v_{2,1} = 20000$ veOLAS.

Therefore,

$$
v_1 = 30000 + 10000 = 40000, \qquad \text{and} \qquad v_2 = 20000.
$$

Since $TopUpComp = 50000 < v_1 + v_2$, we have the following

$$
\begin{aligned}
top_{c_1} &= \frac{TopUpComp}{v_1 + v_2} \cdot \frac{v_1}{2} = \frac{50000}{60000} \cdot \frac{40000}{2} = \frac{50000}{3} = 16666.7, \\
top_{c_2} &= \frac{TopUpComp}{v_1 + v_2} \cdot \left(\frac{v_1}{2} + v_2\right) = \frac{50000}{60000} \cdot \left(\frac{40000}{2} + 20000\right) = \frac{100000}{3} = 33333, 53.
\end{aligned}
$$

Example 3. We have the following services s_j, agents a_t, components c_i, and veOLAS holdings v_i.

$s1 = [a1, a1 = [c1, c2]]$, donator with $v_{1,1} = 80000$.

$s2 = [a2, a2 = [c2]]$, donator with $v_{2,1} = 20000$ veOLAS

$s3 = [a3, a3 = [c2, c3]]$, same donator of $s2$, donated to s_3 in the same transaction with $v_{3,1} = v_{2,1} = 20000$.

Therefore,

$$
v_1 = 80000, \qquad v_2 = \frac{20000}{2}, \qquad \text{and} \qquad v_3 = \frac{20000}{2}.
$$

Let's assume that $TopUpComp(e) = 60000$. Since $TopUpComp = 60000 < v_1 + v_2 + v_3$, we have the following

$$top_{c_1} = \frac{TopUpComp}{v_1 + v_2 + v_3} \cdot \frac{v_1}{2} = \frac{60000}{100000} \cdot \frac{80000}{2} = 24000$$
$$top_{c_2} = \frac{TopUpComp}{v_1 + v_2 + v_3} \cdot \left(\frac{v_1}{2} + v_2 + \frac{v_3}{2}\right) = \frac{60000}{10000} \cdot \left(\frac{80000}{2} + \frac{20000}{2} + \frac{20000}{4}\right) = 33000$$
$$top_{c_3} = \frac{TopUpComp}{v_1 + v_2 + v_3} \cdot \frac{v_3}{2} = \frac{60000}{100000} \cdot \frac{20000}{4} = 3000$$

Example 4. We have the following services s_j, agents a_t, components c_i, and veOLAS holdings v_i.

$s1 = [a1, a1 = [c1, c2]]$, donator with $v_{1,1} = 40000$ veOLAS makes two donations
$s2 = [a2, a2 = [c2]]$, donator with $v_{2,1} = 20000$ veOLAS.
$s3 = [a3, a3 = [c2, c3]]$, same donator of $s2$ that make a new donation in a new transaction with $v_{3,1} = 20000$.

Therefore,

$$v_1 = 40000, \qquad v_2 = 20000, \qquad \text{and} \qquad v_3 = 0.$$

Assuming that $TopUpComp \geq v_1 + v_2 + v_3$ yields

$$top_{c_1} = \text{compFrac(e)} \cdot \frac{v_1}{2} = \text{compFrac(e)} \cdot 20000,$$
$$top_{c_2} = \text{compFrac(e)} \cdot \left(\frac{v_1}{2} + v_2\right) = \text{compFrac(e)} \cdot 40000$$
$$top_{c_3} = 0$$

Example 5. We have the following services s_j, agents a_t, components c_i, and veOLAS holdings v_i.

$s1 = [a1, a1 = [c1, c2]]$, (different) donators with $v_{1,1} = 20000$, $v_{1,2} = 10000$ veOLAS
$s2 = [a2, a2 = [c2]]$, same of s_1 donator owning $v_{2,1} = v_{1,1} = 20000$ made donation in one tx

Therefore,

$$v_1 = \frac{20000}{2} + 10000 = 20000, \qquad \text{and} \qquad v_2 = \frac{20000}{2}.$$

Assuming that $TopUpComp \geq v_1 + v_2 + v_3$ yields

$$top_{c_1} = \text{compFrac(e)} \cdot \frac{v_1}{2} = \text{compFrac(e)} \cdot 10000,$$
$$top_{c_2} = \text{compFrac(e)} \cdot \left(\frac{v_1}{2} + v_2\right) = \text{compFrac(e)} \cdot 20000.$$

5 Implementation changes

To accommodate this proposed method, minimal implementation changes are required, ensuring minimal impact on gas costs for donations. Specifically, the following steps are necessary.

- **Donor Address Tracking:** Have a mapping to record donor addresses, preventing multiple boosts during the same epoch.
- **Boost Limits:** Enforce a limit on the total boost amounts that can be triggered by donors. This limit should ensure that the number of top-ups remains smaller than the veOLAS owned by all donors collectively. That is, a donor holding m veOLAS should have the capability to boost a maximum of m OLAS tokens within a single epoch
- **Algorithm Integration:** Incorporate the existing tokenomics algorithms, replacing reward weights with veOLAS weights for calculating pending top-ups.

6 Future ideas

One could replace the fixed “dev rewards” fraction per epoch with a dynamic one, that takes into account the current price of OLAS in ETH. This would allow targeting a certain quantity of boosts in ETH terms.

Furthermore, one could make boosts not only depend on veOLAS holdings but additionally on the amount of ETH donated. This has the benefit that boosts are proportional to not only value locked but also value donated. The downside is that it implies a rich-get-richer dynamic.