

Tokenomics for DAO Governance and CES Exchange

v1.0

(under development)

Our DAO governance and community exchange system (CES) of the decentralized cooperative (DC) Ecofintech Coop works with 3 native cryptographic tokens called LABOR, LOVE and VOTE. These three tokens each have a specific function within the ecosystem.

LABOR Token

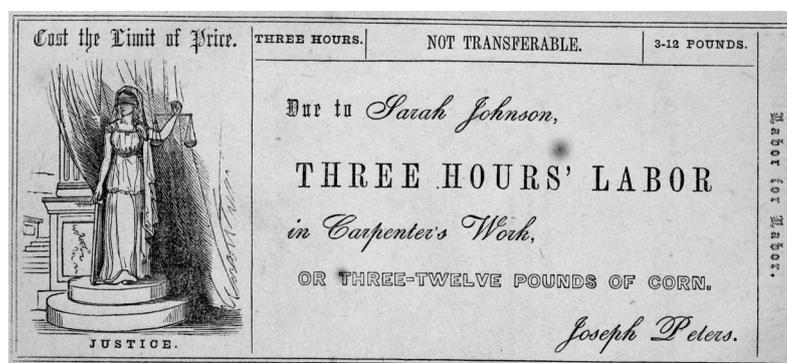
The Labor-value Token (LABOR) is a token based on the RRC-20 standard on the RSK federated network. This token is transferable and interchangeable between the members of the decentralised cooperative, so that it can be freely obtained at an exchange and traded with other members as well as received as a form of payment for work carried out. When performing a task of, for example, developing a front end or training a new participant, a quantity of LABOR will be obtained and at the same time a quantity of LOVE (care work) based on the amount of LABOR, in a relationship:

$$1 \text{ LABOR} = 0,5 \text{ LOVE}$$

The amount of LABOR received for the jobs in relation to the time spent is 3h = 1 LABOR.

As we see producing value through work it also generates a part of care work for the cooperative, which is accounted for using the non-transferable LOVE reputation token.

Website: <https://labor-token.github.io/>



LOVE Token

Unlike LABOR, it is not interchangeable between members and cannot be bought or sold. The only way for a member to obtain the LOVE token is by being

recognized in their contribution in care or by generating LABOR, while at the same time obtaining a part of LOVE. This token will not be received through the purchase and sale of LABOR, since it only corresponds to a reputation granted and withdrawn by the community to each specific participant. The mechanism of acquisition of LOVE and the subsequent calculation of VOTE avoid the risk of losing governance rights to external investors, so that the workers of the cooperative will always be able to balance in their favor most of the decision-making power.

The community of participants in the DC may assign (to mint) or remove (to burn) LOVE to other members, whether or not they have required it. In this way the care and reproductive work token is established as a reputable token with meritocracy function for participants.



VOTE Token

This token allows DC members to participate in the voting of proposals made. VOTE will be accumulated over time as a basket, therefore, the members who contribute more to the community, either in the form of work or making other contributions, and have acquired more LABOR & LOVE will have more influence on the votes.

For starters, members who are approved in the DC, will have 1 VOTE and from that initial amount they will be able to acquire more VOTE that will accumulate over time, thanks to the profit of LABOR & LOVE.

Holistic Weighting of Vote Protocol: *(in R&D process)*

From the platform cooperativism and the open value networks we understand by a consensus model just the one that visualizes a greater influence in the decision making to the members that contribute more in work and care of the project. By means of the vote with holistic weighting factor designed by Ecofintech Coop, each participant in the ecosystem has a weight of vote calculated through the history of the work and care credits contributed in an integral way to the ecosystem and accounted for by means of the cryptographic cards that we have described previously.

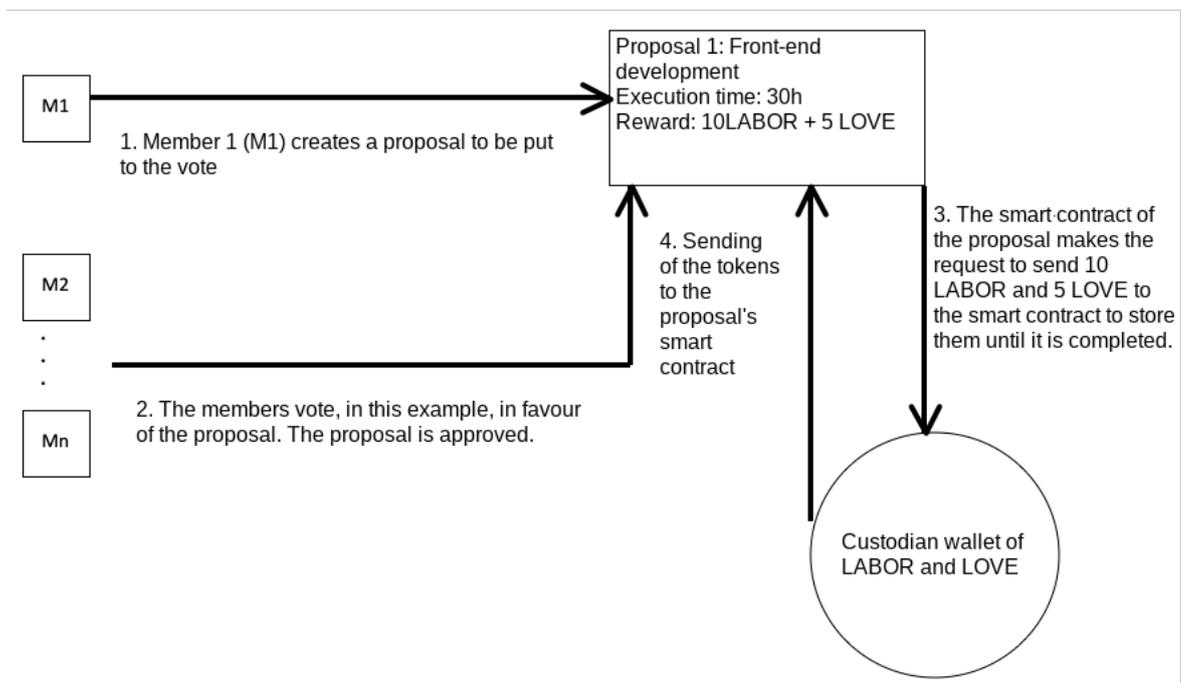
Now let's see how the three components interact with each other, and what role each plays within the functioning and governance of the DC. In order to do this, we are going to define the basic elements that must compose the system, these will be: members and proposals.

- **Members:** are the members of the DC, charged with voting on proposals to be rejected or carried out. They carry out the different development and administration tasks and in this way obtain LABOR and LOVE tokens as a

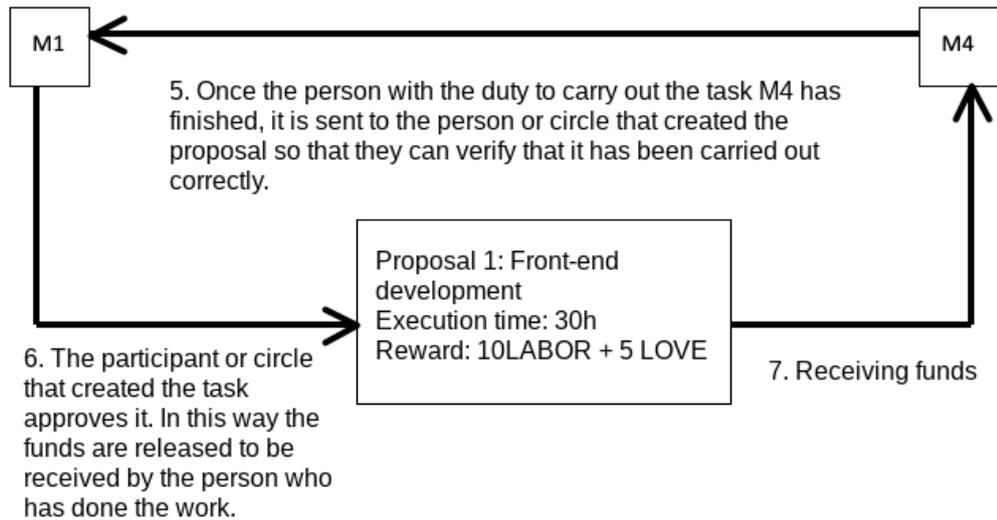
reward for their work. In this way they obtain additional VOTE to be members of more weight within the network according to their work or care provided. They can sell in the market or redeem with the own DC the obtained LABOR and thus to obtain a retribution in cash or cryptocurrency.

- Proposals: These are a fundamental part of the DC ecosystem. Thanks to the proposals, progress in the organization is achieved and the different actions are promoted. The idea is that the proposals are a smart contract that interacts with the members and the wallet that keeps the LABOR and the LOVE in the following way:

In the first instance, the proposal is created to be put to a vote(1), the member who creates it allocates the time that he considers convenient for the completion of the task, which will lead to a reward based on that time in LABOR and its correspondent LOVE, and which will be taken from the DC fund. It can also be proposed by a private individual with their own funds in LABOR, fostering the internal economy of the cooperative. The next step (2) will be to vote for or against it to allow its execution. Once approved (3), the Smart contract of the proposal, requests the funds to the custodian wallet to store them until the task is executed and verified. Finally, the wallet custodian sends the amount to the smart contract(4). (note: we will modify this point so that instead of a wallet custodian, the tokens are coined directly by the contract).



Now we move on to the part where the task is performed, and tokens are sent from the Smart contract of the proposal to the member who has done the work.



This type of proposal is a subsistence work. There are also proposals that are just consultations for decision making, movements of funds for some external project, etc., and requests for remuneration for commons oriented work carried out autonomously.

Assignment of VOTE tokens: They are those that allow to interact with the proposals. When voting on proposals, the VOTE basket must be updated so that, if someone has won LABOR or LOVE, this holistic increase in influence is reflected in the basket.

The way in which this process is carried out is through the application and evaluation of the following formula:

- 1.- *Formula to know if the wallet has received new tokens and must update the basket (this way it is avoided that a malicious user, acquiring LABOR in market and adding one to one each token, adds VOTE over the limit that corresponds to him):*
x: Amount of LABOR contained in the user's wallet
y: Amount of LOVE contained in the user's wallet
z: Amount of VOTE contained in the user's wallet

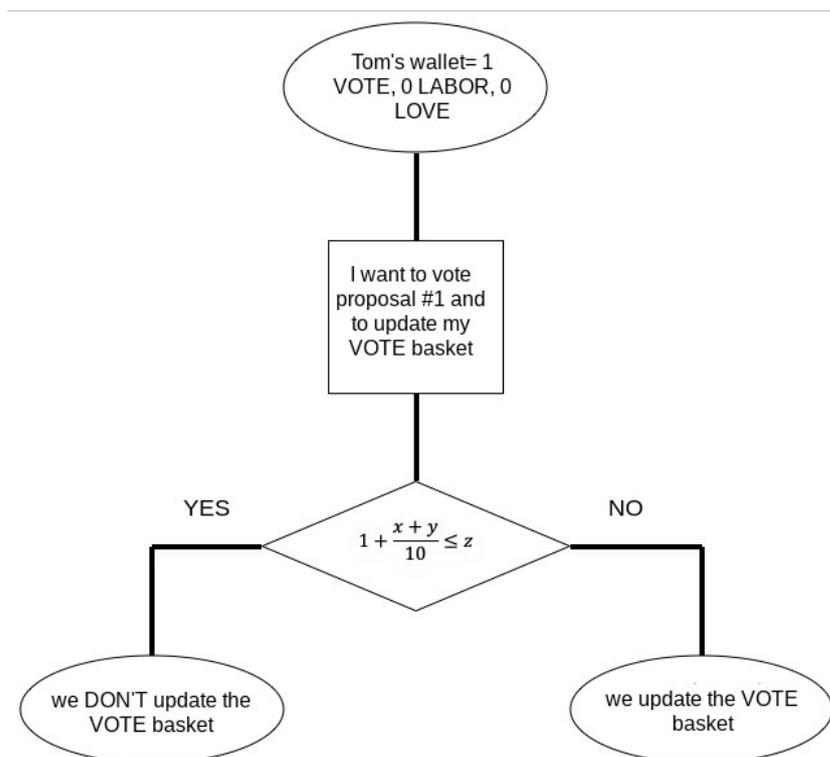
$$1 + \frac{x+y}{10} \leq z$$

- 2.- *Once it is calculated, in case of result not equal or less, the basket must be updated before a new voting proposal, the following formula is applied to add VOTE:*
C: Amount of VOTE to be added to the wallet

$$\frac{x+y}{10} - z + 1 = C$$

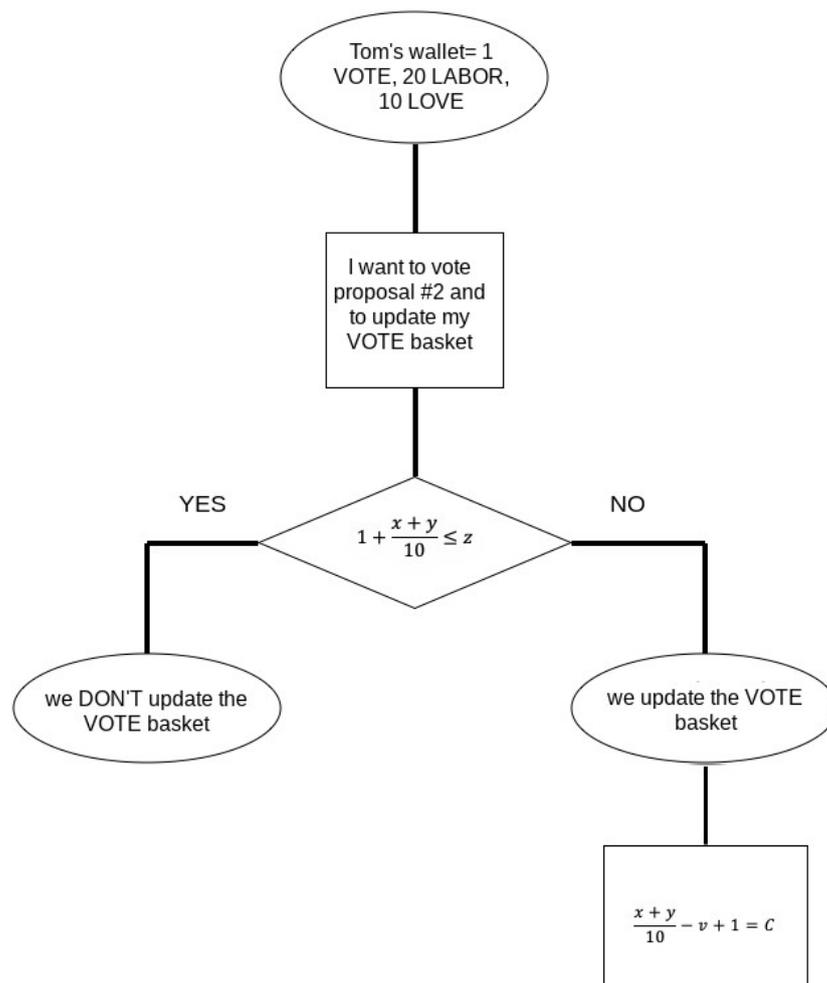
The application of this formula within the voting process will be shown through an example in which a member of the DC, Tom, enters the same, is approved by the community and therefore receives their respective VOTE token to interact in the organization and from there, interact with the different proposals.

What we are looking for is a method to prevent a malicious agent from getting indefinitely VOTE, making votes over and over again so that your basket is updated and receive the tokens without it having obtained more LABOR or LOVE. As we mentioned earlier, the example starts with member 1, which we will call Tom, joining the DC, being approved and receiving his 1st VOTE token so that he can immediately start taking sides in the organization's processes. This member wants to vote on proposal #1, and the next voting process is triggered, but first the VOTE basket has to be updated. So that the system does not allow VOTES to be won without new LABOR & LOVE having been achieved. The first formula is applied, where $x=0$, $y=0$ and $z=1$, which would result in a result of $1 + \frac{x+y}{10} \leq z$. When evaluating it, the check that we have not received tokens would be fulfilled, therefore, the VOTE basket would not have to be updated.



Following this example, time passes and Tom has done some work for the DC and obtained some tokens, 20 LABOR and 10 LOVE, since then he has not wanted to vote, as no proposal has caught his attention or he has simply not been able to participate (in this case the decentralised cooperative would continue to function despite situations of low participation, thanks to the Holographic Consensus protocol, see

below). Then a proposal appears in which he wishes to take part. This is where the previous process takes place again, but with a difference at the end of it:

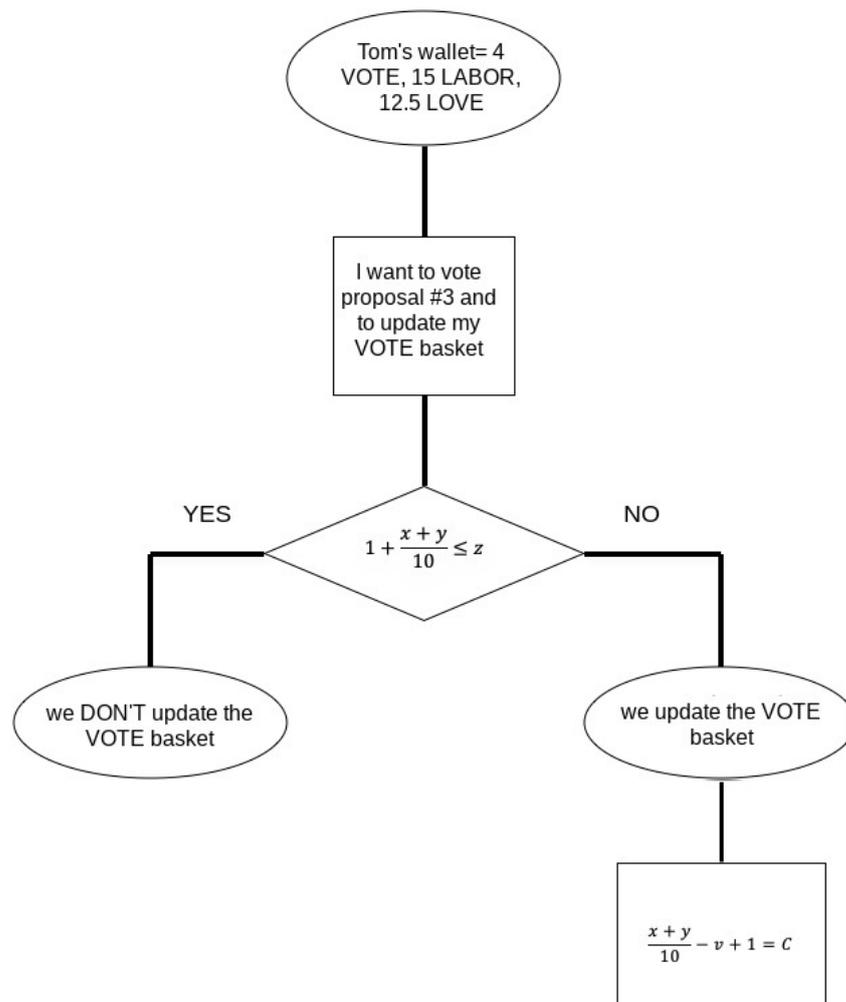


In this process, Tom will vote on proposal #2, so like the previous time with proposal 1, he must first update his VOTE basket. We apply the formula again and evaluate it, this time we have to $x=20$, $y=10$ and $z=1$. This time the equality is not fulfilled, since 4 is neither less nor equal to 1, which means that we have received LABOR and LOVE that has not been taken into account in our VOTE basket and therefore we have to update and receive the VOTE that correspond to us. The way to know how much VOTE has to be added to the wallet, is by applying the second formula.

This formula allows to take into consideration only the tokens added since the last update and exceeding the historical maximum. It also gives us the possibility to avoid that a malicious agent can acquire additional VOTE thanks to the purchase of LABOR in an exchange to pass it of 1 in 1 to his wallet and to make votes in different iterations to go accumulating VOTE, taking into account all the historical tokens instead of only the last added, of this form it would surpass the historical maximum. This mechanism also allows that when someone has won a number of tokens, then has received their VOTE, then has spent the LABOR tokens and lost LOVE; and then has won more LABOR and LOVE, these tokens that have been won will not be taken into account until the member accumulates as many as the last time his basket was

updated. In this example, we have calculated that $C=3$, this amount will be added to our wallet and we can now vote. Leaving Tom with 20 LABOR, 10 LOVE AND 4 VOTE. If Tom spends 10 LABOR and then wins 5 LABOR and 2.5 LOVE he will have in his wallet 15 LABOR, 12.5 LOVE and 4 VOTE, applying the formula, therefore, his basket will not be updated even if new tokens have been entered. As you have not stored enough, the VOTE increase will not occur until the equation exceeds the number of 4 VOTES. This system will make historical accumulation increasingly difficult for members and thus avoid monopoly on voting.

We continue with the example and now Tom spends 10 LABOR and then wins 5 LABOR and 2.5 LOVE, then wants to vote a third proposal. As before, we evaluate with the formula and in this case we have to $x=15$, $y=12.5$ and $z=4$, leaving the equality. As 3.75 is less than 4, equality is fulfilled and we do not update the VOTE basket. The same process can be applied if, after voting on this proposal, Tom would like to vote on a fourth proposal just after the third. Since there would be no changes to his wallet, the process would be identical.



Holographic Consensus:

This protocol designed by DAOstack allows us greater resilience and scalability in the decision-making process, establishing mechanisms that reward attention and allowing a democratic confederated decision-making model more efficient than the

flat assembly. Participants can structure voting by delegating in sections by topic or specialization, and proposals are not blocked in situations of low participation.

Read more:

<https://medium.com/daostack/decentralized-governance-first-principles-1fc6aa492ed>