KILLER WHALE FINANCE

Whitepaper

By Mitja Goroshevsky and Andrey Lyashin

"***Unus pro omnibus, omnes pro uno***"

Abstract

This document describes practical decentralized project tokenization and funding models by introducing the following mechanisms: **Soft Cooperative Voting (SCV)** for managing Decentralized Funds and **Fair Decentralized Autonomous Organization (FDAO)** for aligning interests between all DAO participants[1].

[1]Killer Whale Finance (kw.finance) is a project produced by Killer Whale Pod Cast (kwpc.io). Killer Whale Token (KWT) is a **Launchpool** on top of the FLEX decentralized order book exchange on Everscale. Killer Whale Finance (KWF) is a **Launchpad** for the FLEX Initial DeX Offering (IDO). While KWT provides liquidity for young blockchain projects, KWF provides them with funding.

Soft cooperative voting

The idea of Soft Cooperative Voting (further - SCV) comes from two goals of decentralization: give the active voters a direct ability to influence the processes while defending the passive majority from unwanted *binary* decisions.

Most voting procedures answer only binary questions: to-elect or not-to-elect, to-invest or not-to-invest, to be or not to be… But the financial world is quantitative — it deals with questions like *how much to invest*? *how many people to elect*? *how often to distribute*? and so on. The proposed SCV procedure addresses these questions in a decentralized manner.

Every actor has a quantitative measure of participation in some campaign, which can be interpreted as a share of campaign stakes. Actors with such stake can pro-actively participate in voting as campaign representatives. And as representatives they can vote and have their vote accounted for immediately, without waiting for others. That is why we call it *soft*. There is no need for any kind of majority or any other binary (or discrete) decomposition notion.

A campaign is something common, integrated over participants' ventures or interests and therefore voting from a representative should reflect on others. Thus an actor literally votes literally affects the interests of others too, proportionally to their stake share, of course. That is why we call it *cooperative*.

For example, let's imagine a Token (let's call it a Killer Whale Finance Token or KWF for short) which will be issued to every investor in a pool (we will refer to it as "the Pool" further), proportionally to their investment. Let investors vote for the Pool capital allocation to several projects. If investor I_1 which has 10% of the KWF Tokens will vote with all its Tokens for project P_1 while no other investors of the Pool vote for it, the total of 10% of the Tokens will be allocated to project P_1, but from all of the Pool funds. In other words investor I_1 will end up allocating 1% of their stake in the Pool while the sum of all other investors that did not vote will allocate another 9% proportionally, simply because they hold the other 90% of the Pool.

In order to protect investors who did not participate in the voting process for whatever reason or did not want to participate in this round at all, we are introducing two additional important mechanisms: **Minimum Investment Threshold** (MIT)[2] and **Monthly Investment Subscription** (MIS).

KW Finance uses SCV procedure symmetrically. First, to gradually vote for project allocation until the MIT is reached to ensure the broader consensus among Pool investors, and second,

[2] Is defined by **Minimum value of monthly burning rate (Pmin)**, see below

to later decide what fund share percentage is to be directed to support the project in quantitative expression via MIS mechanism to mitigate the so-called Rug pull attack[3].

The quantitative (continuous) control is much more adaptive than a binary (discrete) one and two decades ago it gave green light to the automated car industry pioneered by Sebastian Thrun et. al. with their "Probabilistic robotics"[4]: "Traditionally… uncertainty has mostly been ignored in robotics. However, as robots are moving away from factory floors into increasingly unstructured environments, the ability to cope with uncertainty is critical for building successful robots". This holds true for decentralized finance as well.

Yet in the end, if an investor completely disagrees with one of the Fund's allocation decisions, they can always exit the Fund through the FDAO mechanism. For example, if at a certain point one of the participants in the Pool disagrees with the one of the Pool's allocations, they can exit the Pool by burning their unlocked KWF Tokens via the Fund smart contract which will result in a proportional share of all investments of this participant being returned to them in exchange for the burned KWF Tokens. We will further show in the FDAO section below that this mechanism works for any project revenues allocation as well..

Project Allocation

Let's see now how the Fund will allocate its resources to a new Project. A Project that wishes to be considered for the Fund support will have to submit the following parameters:

a. **Minimum value of monthly burning rate (Pmin).**
   If the Minimum Investment Threshold (MIT) is not reached, in other words, if the project doesn't get the minimum coverage during the initial voting, it is assumed to be rejected.
b. **Maximum value (Pmax).** Is a value provided by the Fund current settings.
c. **The maximum total Project tokens (TPT*)** emission — the maximum total number of Tokens to be emitted. The real emitted number TPT of project Tokens can be less than TPT* because emission (minting and farming) depends on the project funding level. If all of the lifetime of the project is 100% funded (at Pmax level) then the real emitted number should equal TPT*.
d. **The number of months (MN)** of the project run. It defines the **Monthly Investment Subscription** (MIS), where the Minimum and Maximum levels are determined by the Fund settings.

[3] A rug pull involves the abandonment of a project by the project foundation after collecting investor's funds

[4] http://www.probabilistic-robotics.org/ by Sebastian THRUN, Wolfram BURGARD, Dieter FOX (2002)

e. **The minting Token number for 100% funding (PTM).** The number corresponds to the number of Tokens minted if the project has full funding support. This parameter defines the initial price suggestion for the project Token.
f. **The Initial Funding Level (IFL) and Current Funding Level (CFL)** constitute the total amount committed by the Fund at the IDO event and dynamically every month. **The farming program's exponential base (0<E<100)** defines project farming as FaPT = (CFL/MN) * E^mn , where *mn* is the current month (zero based), so that E represents how much more will be farmed in a month at the end of the program than during the first month' coverage. This parameter is not to be calculated by the Project's initiators but rather provided automatically based on other, clearer parameters (see below).
g. **Investor Participation — IP%, Reserve Participation — RP%** correspond to investors' IP% and reserve shares formed on any minting act, respectively. The **Founders Participation — FP% =** 100%-IP%-RP% corresponds to the founders' share of Tokens.
h. **Investor Project Tokens (IPT)**+ **Reserve Project Tokens (RPT)** + **Founders Project Tokens (FPT)** = **Total Project Tokens** <= TPT* are tightly connected with the Tokens quantity for all parties and are to be defined at the very end of project funding.

Voting with KWF

For each unallocated KWF Token, 1 KWF Voting Token (KWFV) is issued to each investor. These Tokens are used to vote for the Project allocations. They are always equivalent to the Investor's current unallocated pool share.

When all the requirements are fulfilled, the Project is going to the SCV procedure where each investor can vote with their KWFV Tokens for the project. The initial level of funding is determined by the formula

IFL = V/FKWFV*FVL,

where V is a number of KWFV Tokens voted for the project, FKWFV is a number of free KWFV Tokens (voting bulletins), FVL - is the available value of EVERs locked in the Fund. So each FKWFVT corresponds to a certain amount of EVERs locked and available in the Fund.

Next, IFL should be compared with two other numbers:

- IFL >= Pmin*MN
- IFL <= Pmax*MN

If the first condition holds, the project is considered accepted. The second condition is to verify that the project is not going to take too much from the fund and if IFL > Pmax*MN, it is then defined that IFL = Pmax*MN.

According to the CSV model it is not necessary for all investors to vote to have a project accepted, only a share of Pmin*MN/FVL is enough. But after acceptance every investor has a proportional share in all of the projects unless and until all investors decide to exit a particular project at a certain moment.

Now we can estimate TFT* expressed in other parameters' values:

TFT* = PTM + (PTM/MN) * [(E^MN-1) / (E - 1)] / MN

After the project allocation the sum IFL is locked. Part of it can be gradually unlocked later anticipating new projects or if a project is losing support of the Fund participants.

Project Farming is exponential for the "greed" reason. in order for investors not to rebalance their portfolio out of the earlier projects too easily, the longer the project has been supported, the harder the decision to exit it should be.

~Sample simulations of different voting dynamics can be found at the end of this Paper.

Project farming formulas

To achieve the farming goals the project initiators define the APY parameter when deploying the Project contract. It is defined as the final fraction of total farmed tokens to total minted tokens. Then the monthly farming amount is set at

$FPT_k=(MPT_k/K)*E^k,$

where
k - is the zero-based current month number,
K - arbitrary constant reflecting the level of skewness of the plot (we can choose it equal to MN),
E - parameter to be found from the relation
$\sum MPT_k = APY * \sum FPT_k$
where k is the current month' number.

Then the right hand side is equal to

$(E^{MN}-1)/(E-1)*APY/K*\sum MPT_k$ [5]

and after cutting the $\sum MPT_k$ factor from the both sides the equation reads:

$(E^{MN} - 1)/(E-1) = K/APY$

which can be numerically solved with respect to **E** (here is no general analytical solution, but Newton method will give a good approximation, with starting value for iteration E0 = 2).

Fair Decentralized Autonomous Organization (FDAO)

Above we have described how a Project is evaluated for Funding in KWF and how the KWF Token holders are voting to provide quantitative support for it. In exchange for this support the Project Tokens (GOV) are issued to the Fund through minting and farming mechanisms.

Let's now discuss a Project Tokenization model which such a Project may employ. In most cases this framework should be used by Projects that would like to receive support from KW Finance. As we will show below this model provides a "Fair" mechanism for stakeholders, investors and entrepreneurs alike.

Let's assume we have an on-chain business (Project) which generates some revenue (REV). Let's make all revenues go to a special Revenue Accumulator smart contract (RAC) and be locked there. Now let's issue some Governance Tokens (GOV) of the Project. Remember that the final number of such GOV Tokens will be determined at the end of the funding stage.

An owner of the GOV Token can trade it against Accumulator by burning it at any time. The price of such trade will be determined as follows:

$Total_{REV}/Total_{GOV}$.

That will provide a way to exit the Project through its accumulated revenues (something like an accumulated dividend payment). Since all project revenues go to the RAC, the amount of Evers divided by the amount of GOV Tokens will constitute the "intrinsic" or a "floor" value of the GOV Token.

[5] the calculation is using the MPT monthly constant

Thus if the Project needs to spend additional funds after spending all the funds that were received through the KWF Token sale, Founders will have to burn their GOV Tokens against the Project's accumulated revenues.

A Token holder of course would rarely (or never) use such a mechanism because most of the time the open market price of GOV will be higher than revenues divided by tokens outstanding because of future revenues expectations and decreasing supply mechanism built into the market price. Even if the market is not liquid enough it still gives Projects an ability to fund future development as long as they believe in the ultimate Project's success.

There is a deep internal meaning of having GOV Tokens instantly decreasing RAC. Every Project by its very nature has a limited lifecycle, and therefore there is no need to permanently pay for the development. At the end the payer contract will be emptied, frozen or burned. All unspent Tokens will be proportionally distributed between shareholders (if any) or burned, which will have the identical economic effect, as will be proved below.

Should a Project only come to the support-stage without the need to develop at all, the shareholders can issue other Tokens for that stage. It will be essentially another Project with different economic characteristics[6].

In some cases shareholders can come to an agreement that they are to help the Project by selling their Tokens. Each such time they must make their own decision since there is no automated procedure (yet).

The RAC should not exchange the GOV tokens for the accumulated Evers by trading with the Accumulator. What is important here, is that this trade can be made at any time, thus providing arbitrage protection to the Token price.

The market price can be the same as the Accumulator price only when there are no expectations of any future income, indicating the Project is over and all stakeholders can just take the accumulated revenues and leave.

Now let's prove that:

- The GOV token price will always increase over time if the project accumulates revenues;
- The price of the GOV token on the open market will always be higher than a price in the Accumulator contract due to future revenue expectations;

[6] As a side effect this mechanism prevents monopolization by preventing projects from constantly pivoting or acquiring new business within current token structures.

- The developer selling such token against revenues to fund activities will not dilute their holding in monetary terms as we explain below, but will be more beneficial to stakeholders;
- This model is preserving shareholder value at least as good as other token “buy back” mechanisms using revenues;
- It provides a graceful way to dissolve stalled Projects since all of their accumulated revenues are stored.

Let us have a G number of GOV tokens in total and let the overall accumulated revenue be R, and market price p, so we suggest

G*p >= R:

Let GS be tokens in token holders hands and GP be Tokens at the Team contract that pays the bills, let’s call it a Payer contract, so

G=GS+GP

We trade some g number of GOV tokens from GP and pay bills later. Now we have (G-g) total tokens, and the same amount of GS shareholders’ tokens. Let us compare two options:

1. The trade is made against the Accumulator. Accumulator price before trade is a1=R/G, and after is a2=(R-g*R/G)/(G-g) = R/G = a1 . Note this is the same value as before. That means that the floor price value is held. Market value was: p1 = a1+F1, where a1 is the Accumulator price and F1 is the estimated future discounted revenue over the total GOV supply. And after the trade it became p2 = a2+F2. However F1*G = F2*(G-g) and F2 > F1 because it is about the future and only supply has changed, and a1 = a2 as we just showed. So in this case shareholder’s savings will increase GS*p2 > GS*p1.
2. When trading is made on the open market:
    a. a1 = R/G, a2 = R/G because no revenues were taken, and no GOV tokens were burned. F1 = F2 then, and therefore there is no economic reason to have the market price intrinsically change.
    b. Another difference is that the Payer contract spends less GOV tokens to cover the bills in the second case, which has no effect on shareholders either, as the revenues are saved too.
    c. That means that shareholders can force management to take revenues from the Accumulator rather than to make open market operations, however they do not lose anything in either case.

So let us go through the points:

1. The token price will always increase over time if the project accumulates revenues

a. Let us compare two moments p(t1) = a1 + F1 and p(t2) = a2 + F2.
b. a1=R1/G, a2=R2/G (let's temporarily ignore trades in between) .
c. F1*G = ((R2-R1) + F2*G)/r, when r>1 is a discount of period (t2-t1).
d. R1/G+F1 and R2/G+(F1*G*r-(R2-R1))/G
e. R1/G+F1 and R1/G+F1*r , r>1 and
f. R1/G+F1 < R1/G+F1*r QED.

2. The price of the GOV token on the open market will always be higher than the price in the Accumulator contract due to future revenues expectations.
3. A developer selling such a token against revenue to fund activities will not dilute their holding in monetary terms but be more beneficial to stakeholders. As was shown before, that is even more so when developer trades on the open market as compared to trades with Accumulator.
3. This model is preserving shareholder value more or equal to token "buy back" using revenue mechanisms:
    a. What will happen if shareholders sell g GOV tokens:
        i. To the Accumulator
            1. p1 = R/G+F1
            2. p2 = (R - g*R/G)/(G-g)+F2 = p1-F1+F2
            3. F1*G = F2*(G-g) as before
            4. Compare holdings value H1 = GS*p1 and H2=(GS-g)*p2 + g*R/G
            5. GS*p1 and (GS-g)*(p1-F1+F2) + g*R/G
            6. GS*p1 and (GS-g)*(p1-F1+F1*G/(G-g)) + g*R/G
            7. GS*p1 and (GS-g)*(p1+F1*g/(G-g)) + g*R/G
            8. 0 and -g*(p1+F1*g/(G-g))+GS*F1*g/(G-g) + g*R/G
            9. 0 and -g*p1+g/(G-g)*(GS*F1-F1*g) + g*R/G
            10. 0 and -g*p1+g/(G-g)*(GS-g)*F1 + g*R/G
            11. p1 and (GS-g)*F1/(G-g)+R/G
            12. p1 = R/G + F1 > F1*(GS-g)/(G-g) +R/G because GS<G
            13. So the market value will be less in this case.
            14. Selling GOV tokens to the Accumulator is the simulation of a "buy back" mechanism as in the latter the revenues belong to the Company (read - management) which we have shown to be less profitable than holding the GOV tokens, however is still possible in our model.
        ii. Against the open market
            1. p1 = p2
            2. GS*p1 and (GS-g)*p2+g*p1 Equal
4. Model provides a graceful way to dissolve stalled Projects since all of their accumulated revenues are stored.

a. The Process of project closing is conducted by selling all of the tokens to the Accumulator since there is now a difference with market value).
b. We should ignore the Payer contract tokens in this case as they have no economic meaning since there are neither “bills” to pay nor tokens in this contract anymore.
c. That also means that we should think of G in the previous formulas as real total G minus the rest of Tokens on the Payer contract at the moment of Project closing. It doesn’t change the math.

Avoiding the Tragedy Of The Commons

By mere coincidence the mechanism we described is also under certain conditions solving for the Tragedy of The Commons. V.Buterin is talking about the Tragedy Of The Commons in cryptoeconomics (for example here). Let us consider how SCV can solve this without a requirement for any public funding.

Let’s consider the SCV model in the management of common resources such as a pasture.

1. Let’s assume that this pasture yield is decreasing exponentially with the growth of grazing cattle number

$P(x) = P0*\exp(-x/a)$,

where P0 is yield with no cattle and a is the decrease rate, so when x = a, yield decreases by e = 2.7… times;

2. Let the farmer to have an unlimited number of cattle without and no additional expenses when cattle size increases;
3. A rational farmer will calculate their benefit as follows:
   $Q(x) = P(x+c)*x$
   where x — is their own livestock, and c — is livestock of everyone else;
4. Then $Q’(x) = P’(x+c) * x + P(x+c)$ and with expected behavior of P(x)
   a. $Q’(x) = -P0*\exp(-(x+c)/a)/a*x+P0*\exp(-(x+c)/a) = P0*\exp(-(x+c)/a) (1- x/a)$
   b. Meaning when all farmers act independently they will increase their livestock towards “a” because with a they receive maximum benefits.
   c. The total sum of pasture yield be

$QT = P(n*a)*a*n = P0*\exp(-n)*a*n$

thus quickly tending to zero with the number of farmers increasing

5. In the SCV model when farmers are not independent
   Q(x) = P(A*x)*x,

   where A — is a reverse share of the current farmer in the system so that A*x — is total livestock, and increasing their own livestock leads to proportional growth of everybody's:
   a. Q'(x) = A*P'(A*x)*x+P(A*x)
   b. Q'(x) = -A*P0*exp(-A*x/a)/a*x+P0*exp(-A*x/a) = P0*exp(-A*x/a)*(1-A/a*x)
   c. x -> a/A
   d. QT = P(a) * N = P(a)*a , where N total livestock, N = SUM (a/Ai) = a

It means the use of this pasture stabilizes for common livestock = a and does not allow for pasture depletion.

Conclusion

Described mechanism of Revenue distribution ensures the incentive to hold the token and supports the floor price of the EVER/GOV pair, therefore constantly pushing the price higher.

Other outcomes of such a mechanism are transparency and alignment of interest between all project stakeholders, including developers. Since all revenues are going into the Accumulator contract, the developers will need to sell GOV tokens to the Accumulator (or in the market) in order to continue funding development and marketing activities should those be needed. Since revenues in Accumulator are constantly increasing unless the project is not generating revenues anymore, in which case it will be dissolved by the forces of the Game Theory. If the developers decide they need to sell part of the revenues by selling some of the tokens, their holdings in Evers terms won't change.

This mechanism also provides additional protection against regulatory risks because developers have the exact same interest as stakeholders and can not be therefore considered an "interested party". The fact that theoretically any stakeholder can sell their stake and fund whatever activities around the Project's open source documentation they deem necessary, the main assumptions of the Howey test do not hold, making GOV therefore fully ready even for institutional investors.

Appendix I

Examples

Project minting and farming example

In the example below:
MN = 12, IFL =Pmax*MN, total maximum number to be minted = 1 bln, E = 1.406,
APY = 100%.

Month	Level of funding	Minted	Total minted	Farmed	Total farmed	Total PT generated
		1,000,000,000.00		100%		
1	100%	83,333,333.33	83,333,333.33	6,944,444.44	6,944,444.44	90,277,777.78
2	100%	83,333,333.33	166,666,666.67	9,763,888.89	16,708,333.33	183,375,000.00
3	100%	83,333,333.33	250,000,000.00	13,728,027.78	30,436,361.11	280,436,361.11
4	100%	83,333,333.33	333,333,333.33	19,301,607.06	49,737,968.17	383,071,301.50
5	100%	83,333,333.33	416,666,666.67	27,138,059.52	76,876,027.69	493,542,694.35
6	100%	83,333,333.33	500,000,000.00	38,156,111.69	115,032,139.37	615,032,139.37
7	100%	83,333,333.33	583,333,333.33	53,647,493.03	168,679,632.40	752,012,965.73
8	100%	83,333,333.33	666,666,666.67	75,428,375.20	244,108,007.60	910,774,674.27
9	100%	83,333,333.33	750,000,000.00	106,052,295.53	350,160,303.13	1,100,160,303.13
10	100%	83,333,333.33	833,333,333.33	149,109,527.52	499,269,830.65	1,332,603,163.98
11	100%	83,333,333.33	916,666,666.67	209,647,995.69	708,917,826.33	1,625,584,493.00
12	100%	83,333,333.33	1,000,000,000.00	294,765,081.94	1,003,682,908.27	2,003,682,908.27

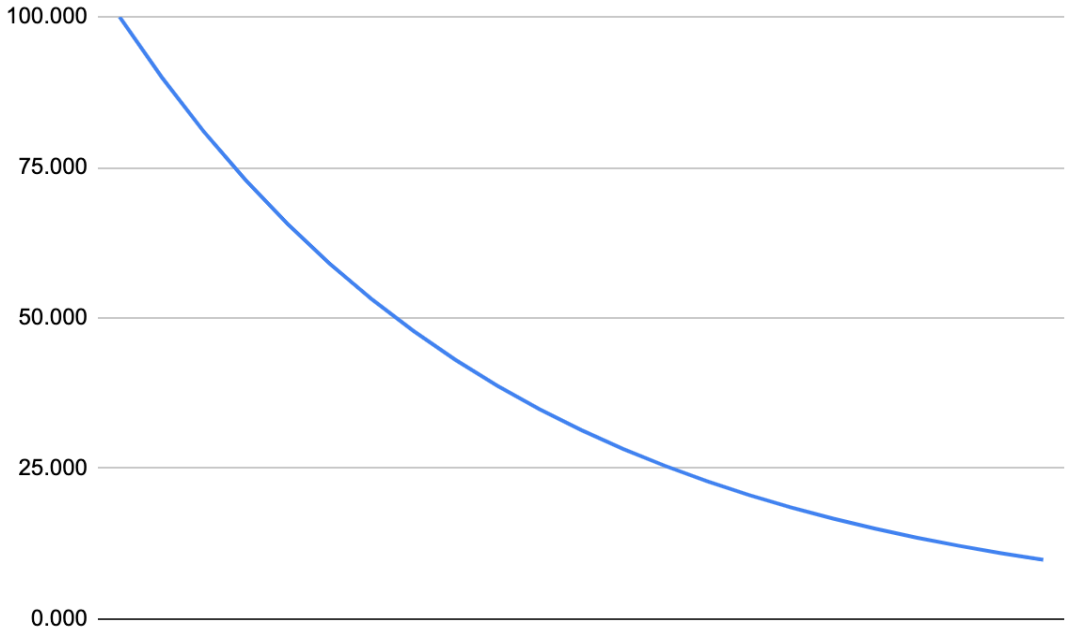
SCV for declining project funding support example

In some cases an investor might want to stop to support the project. In this case they might remove their share from it. But the same way as while voting for, the removal procedure is cooperative. So that at one phase investor can take only I^2 of their support back, where I is their current fund share, while at the same time they frees I*(1-I) from other investors. The new share we show for two cases:

- All other investors are passive. In this case the funding support gradually changes accordingly to I. The proportion of investors participation in the project holds.

	Participant1	Others	Support Level	
Initial share	10.000	90		
	10.000	90	100.000	Const = P1/Others
1	9.000	81	90.000	0.1111111111
2	8.100	72.90	81.000	0.1111111111
3	7.290	65.61	72.900	0.1111111111
4	6.561	59.05	65.610	0.1111111111
5	5.905	53.14	59.049	0.1111111111
6	5.314	47.83	53.144	0.1111111111
7	4.783	43.05	47.830	0.1111111111
8	4.305	38.74	43.047	0.1111111111
9	3.874	34.87	38.742	0.1111111111
10	3.487	31.38	34.868	0.1111111111
11	3.138	28.24	31.381	0.1111111111
12	2.824	25.42	28.243	0.1111111111

We see here that having 10% of Fund share, an investor can gradually decline the project support from 100% to ~30% in a year.



- If other investors decide to save the share of the project they can make a counteraction until the monthly deadline. That is, suggest one of the investors having 10% released the support from the project by 10% at some moment. And other investors counteract
    - Before: 90 + 10 in the project
    - After the release: 81 + 9 in the project, 9 + 1 released
    - After the counteraction: (81+0.9*9 = 89.1) + (9+0.9*1 = 9.9) in the project, 0.9 + 0.1 released
    - The parties can continue the game with restriction to make a fixed number of orders inside a certain time frame - otherwise the sides can significantly change the support level in a row.

  The final restricting values (if any) will be introduced at the development phase.