Tokenomics of VirtuSwap

Evgeny Lyandres

evgeny@virtuswap.io

August 24, 2023

Abstract

This document describes VirtuSwap tokenomics, via VirtuSwap governance token, *VRSW*, and its extension, *veVRSW*. The document consists of the following parts:

1. Description of *VRSW* allocation to various parties and release schedule of *VRSW* tokens.
2. A brief description of fees and incentives.
3. A general description of *VRSW* utility. In designing *VRSW* utility, our approach has been to follow currently available best practices and to improve upon them. In particular, we introduce an elegant mechanism for mixed incentivization of staking and locking of *VRSW* tokens, under which locking tokens for a pre-specified period of time is not required to receive boosts to rewards for liquidity provision.
4. A description of per-block reward boosts, involving staking and locking *VRSW* tokens.
5. A detailed algorithm for optimized computation of rewards accrual and redemption over time.

1 Token allocation and distribution

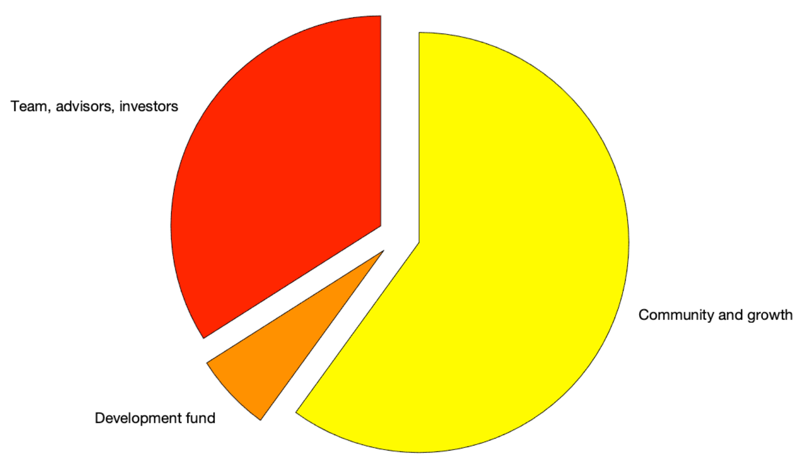
1.1 Initial token allocation

The total number of *VRSW* tokens that will ever be minted is $\overline{R} = 1,000,000,000$. The token allocation is as follows:

- 34% – team, advisors, investors.
- 6% – development fund;
- 60% – community and growth.

The token allocation is illustrated in the following chart:

VRSW token distribution



The founding team, employees, and advisors are all subject to 36-month vesting and 12-month lockup. Investors are subject to 12-month emission.

1.2 Emission of *VRSW* tokens as LP incentivization mechanism

The *VRSW* tokens reserved for community and growth ($0.6\overline{R}$ or 600,000,000 tokens) will be released as follows:

- 10% (100,000,000 tokens) will be released immediately to the VirtuSwap DAO treasury to support pre-launch airdrops and lockdrop, partnerships, and strategic initiatives.
- 50% (500,000,000 tokens) will be released using a pre-determined gradual algorithmic schedule over 10 years. Out of these:
  - 30% (300,000,000 tokens) will be released directly to pools chosen by a combination of VirtuSwap proprietary machine-learning-based liquidity optimization module and voting by VirtuSwap ecosystem participants;
  - 20% (200,000,000 tokens) will be released to the VirtuSwap DAO treasury and will be also used for liquidity provider incentivization but at discretionary times decided by VirtuSwap DAO.

The overall number of *VRSW* tokens released algorithmically to chosen liquidity pools and to VirtuSwap DAO during block t since launch, R_t^A, is described by the following schedule:

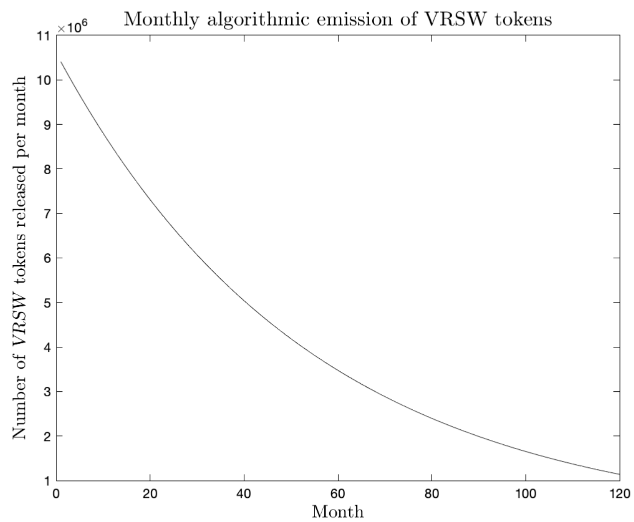
$$R_t^A = \int_{s=\tau_{t-1}}^{\tau_t} V e^{vs} ds = \frac{V}{v}(e^{v\tau_t} - e^{v\tau_{t-1}}) \text{ for } \tau_t \leq \overline{\tau}, \tag{1}$$

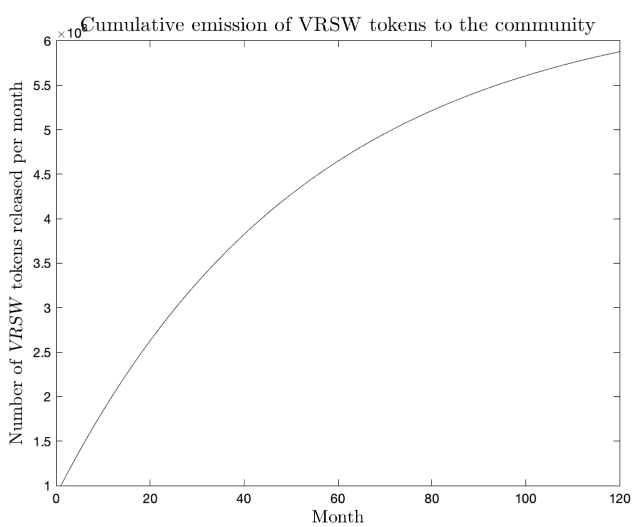
where:

- τ_t is the timestamp (in milliseconds) of block t;
- $\overline{\tau} = (365 * 8 + 366 * 2) * 24 * 60 * 60 * 1,000 = 315,532,800,000$ milliseconds, i.e. the number of miliseconds in a 10-year period beginning in 2023;
- $V = 0.00347030585160354$ and $v = -0.000000000005842017$ are constants chosen so as to ensure that:
  - 20% of all *VRSW* tokens released via the algorithmic release (i.e. $0.1\overline{R}$ or 100,000,000 tokens) will be released in the first year, and 16.83% fewer tokens will be released in each subsequent year until the end of year 10;

– The total number of *VRSW* tokens to be released according to this schedule within 10 years will equal $0.5\overline{R}$, i.e. 500,000,000 tokens.

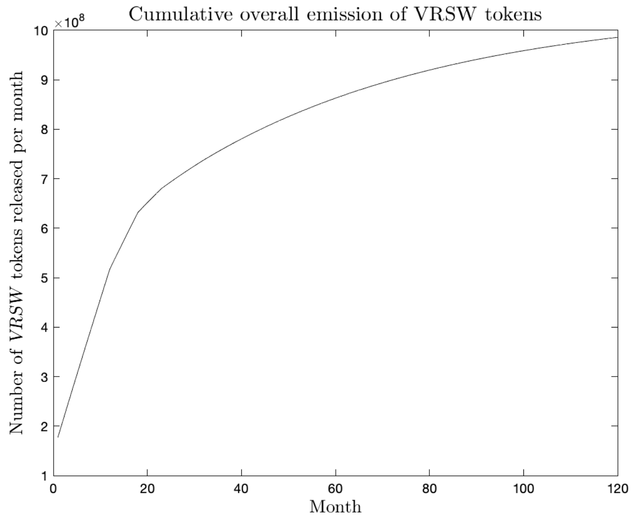
The marginal and cumulative algorithmic release of *VRSW* tokens to community and growth is illustrated in the following two figures, where x-axis represents months:





1.3 Overall VRSW emission over time

The previous section described the emission of *VRSW* tokens to the community. Combining this emission with tokens allocated to the team, advisors, and investors in VirtuSwap results in the following overall cumulative emission schedule:

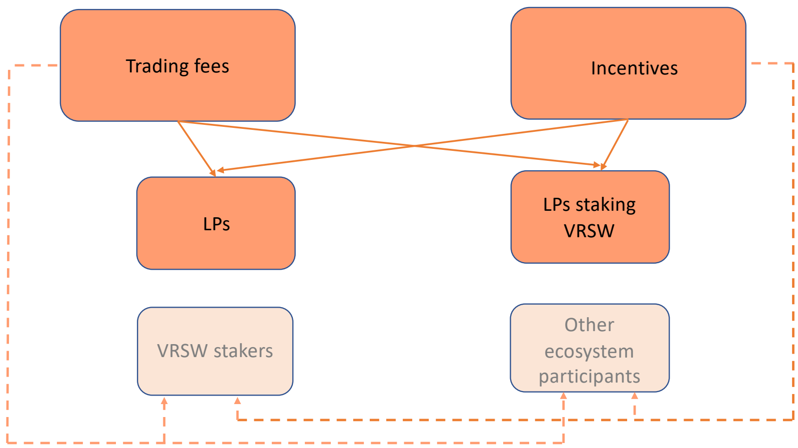


2 Fees and incentives

The fee for each trade within each pool participating in the trade, as well as reward boost for this pool's liquidity providers who have staked their LP tokens and/or staked/locked *VRSW* tokens will be split into the following components, as illustrated in the next figure. The initial distribution of these components is illustrated in dark orange:

- Fraction $F_{pool,fees}$ and $F_{pool,rewards}$ of trading fees in a pool and rewards (in *VRSW* and, potentially in other tokens such as project or blockchain native tokens), respectively, in a given block will stay in the pool. These returns are proportional to the number of LP tokens owned by each wallet.

- Fraction $F_{boost,fees}$ and $F_{boost,rewards}$ of trading fees in a pool and rewards, respectively, in a given block will accrue to liquidity providers in the pool, who have staked their LP tokens (LP token stakers hereafter) and (potentially) staked or locked *VRSW* tokens.



The proportion of rewards accruing to each liquidity provider in a given pool will depend on her relative number of LP tokens staked as well as on the relative number and duration of staked and/or locked $VRSW$ tokens, as described below.

Initially, $F_{pool,fees} = 100\%$, $F_{pool,rewards} = 0$, $F_{boost,fees} = 0$, and $F_{boost,rewards} = 100\%$, i.e. trading fees will stay in the pool, whereas $VRSW$ (and other potential) rewards will accrue to LP token stakers. Later on, changes to fee distribution will be possible, following community decisions made according to VirtuSwap governance mechanism. In addition to possible changes to F_{pool} and F_{boost}, examples of possible future distributions of pool fees and incentives are illustrated in pale orange:

- Protocol-level rewards for staking $VRSW$ tokens (without a requirement for staking LP tokens), F_{VRSW}. Initially, F_{VRSW}=0. Protocol-level rewards can only be turned on and off by the decision of VirtuSwap governance. F_{VRSW} cannot exceed 10% of the per-block algorithmic token emission.

- Future rewards to additional activities in the future versions of the protocol. Examples include decentralized insurance, i.e. taking on the risk of reserve asset price fluctuations, and single-sided liquidity provision (this list of fee distribution channels is not exhaustive). Future rewards will be decided by VirtuSwap governance.

3 *VRSW* utility design

State-of-the art DEX utility design involves the following functions of a protocol token:

- Staking;
- Locking;
- Governance via voting.

VirtuSwap tokenomics via *VRSW* and its extension, *veVRSW*, is based on the three aforementioned functions, with the following important improvement relative to existing DEX token utility structures. It is not necessary to lock *VRSW* tokens in order to receive a reward boost; incentives for both staking and locking are achieved within a single elegant mechanism. Relaxation of the requirement for pre-determined lockup periods is meant to increase the liquidity of *VRSW* token, without compromising its utility, as discussed below.

VRSW and *veVRSW* tokens are tightly connected:

- A new *veVRSW* token is minted every time *VRSW* token is staked (not necessarily locked!).
  - *veVRSW* tokens can be used for submitting governance proposals and voting on them, where the voting power is proportional to the number of *veVRSW* tokens held.
- Staking *VRSW* tokens boosts LP rewards in pools to which the staker provides liquidity.
  - Every *VRSW* token can also be locked ex-ante for a pre-specified period of time. Locking *VRSW* tokens provides an added boost to LP rewards, as described below.
  - If/when protocol rewards are turned on, staking *VRSW* will result in rewards unrelated to liquidity provision.
- *VRSW* tokens that are not locked can be unstaked at any time, resulting in burning of associated *veVRSW* tokens.

4 Accrual of reward boosts to staked LP tokens

4.1 Main objectives

The main goal of the formulation described in this section is threefold:

- Provide incentives for staking *VRSW* tokens for prolonged periods of time, without a requirement of locking them ex-ante for pre-specified time periods.
- Provide added incentives for pre-determined ex-ante locking of *VRSW* tokens.
- Allow flexibility for rewarding prolonged durations of both staking and locking of *VRSW* tokens.

The share of reward boosts to an LP of a particular pool who has staked/locked *VRSW* tokens depends on the following factors:

- Number of staked LP tokens of a given pool;
- Number of staked *VRSW* tokens in each staking position;
- Current duration of staking of *VRSW* tokens in each staking position;
- Locking period of *VRSW* tokens in each staking position;

4.2 Conceptual computation of reward boost accrual during a block

Let us use the following notation:

- Pool consisting of native assets X and Y: $[XY]$;
- Proportional trading fee in pool $[XY]$: f_{XY};
- The number of LP tokens in pool $[XY]$ that wallet i has staked but not locked: LP^i_{XY};
- *VRSW* staking position number p by wallet i: $VRSW^i_p$;
- Current duration of *VRSW* tokens staked by wallet i in staking position p: D^i_p;

- The locking period of *VRSW* tokens in position p by wallet i: $k_p^i \geq 0$. If $k_p^i = 0$ then *VRSW* in staking position p is not locked and is available for immediate withdrawal;
- The compound rate of duration of reward boost calculation: r;
- The reward boost coefficients for locking *VRSW* per unit of locking time: b;
- Parameter describing the curvature of reward boost for locking *VRSW* as a function of locking duration, γ;
- Parameter describing the importance of LP token staking in reward boost computation: α;
- Parameter describing the importance of *VRSW* staking/locking in reward boost computation: β;
- "Allocation points" of pool $[XY]$ during block t, i.e. the fraction of *VRSW* emission in block t accruing to pool $[XY]$: $\eta_{XY,t}$;
- The total amount of asset X (Y) sent to pool $[XY]$ in exchange for asset Y (X) in block t: $U_{XY,t}^X$ ($U_{XY,t}^Y$).

Then, the reward boost accruing to LPs of pool $[XY]$ who staked their LP tokens consists of the following elements:

- Discrete reward boosts (occurring in blocks in which the pool facilitated trades):
  - $W_{XY,t}^X = f_{XY} F_{boost,fees} U_{XY,t}^X$ of asset X;
  - $W_{XY,t}^Y = f_{XY} F_{boost,fees} U_{XY,t}^Y$ of asset Y.
- Continuous reward boosts in *VRSW* tokens:
  - $W_{XY,t}^{VRSW} = R_t^A \eta_{XY,t} F_{boost,rewards}$ of *VRSW* tokens.

Denote by $W_{XY,t}^d$ total rewards (in various assets (denominations) above, denoted by d) during block t. The reward boost accruing to wallet i, $\pi_{XY,t}^{d_i}$, equals:

$$\pi_{XY,t}^{d_i} = W_{XY,t}^d \frac{\mu_{XY,t}^i}{\sum_j \left(\mu_{XY,t}^j\right)}, \tag{2}$$

where

$$\mu^i_{XY,t} = (LP^i_{XY,t})^\alpha (1 + \sum_{p,i} (VRSW^i_{p,i} e^{D^i_{p,i} r} (1 + b(k^i_{p,i})^\gamma)))^\beta, \tag{3}$$

j denotes all wallets that staked LP tokens of pool $[XY]$, and p, i denotes all staking positions (p) of wallet i.

If/when general staking rewards would be turned on, they will be computed similarly to (2) and (3) above, while replacing pool $[XY]$ by $[Protocol]$ and assuming $LP^i_{Protocol,t} = 1$ for all i.

4.3 Example of reward boost accrual

Let us consider an example to illustrate the calculation of reward boost shares of wallets in pool *[XY]* during block t. This example assumes $F_{boost,fees} = 0$ and $F_{boost,rewards} = 100\%$. Assume the following values of parameters that are true as of August 23, 2023:

- Algorithmic emission of *VRSW* tokens in epoch (week) t: $R^A_t = 1,235,764.56$;
- Share of rewards accruing to pool $[USDT - USDC]$ in block t: $\eta_{XY,t} = 14.59\%$;
- The importance of LP token staking parameter: $\alpha = 1$;
- The importance of VRSW staking parameter: $\beta = 1$;
- Curvature parameter of reward for VRSW locking period: $\gamma = 1$;
- The reward for VRSW locking per period of time: $b = 0.000000126769705$ per second or 0.01095 per day;
- Compound rate of reward boost calculation: 0.000000081289468 per second or 0.007023 per day;

The following table describes the number of staked LP tokens, as well as *VRSW* tokens staked, current staking duration (in days), and the length of locking period of various *VRSW* staking positions by two wallets, A and B (in days).

Wallet	# LP	VRSW staking position 1			VRSW staking position 2			VRSW staking position 3		
Wallet	tokens	# VRSW tokens	Staking duration	Locking period	# VRSW tokens	Staking duration	Locking period	VRSW tokens	Staking duration	Locking period
A	50	100	50	60	200	20	0	50	100	200
B	50	50	10	100	100	0	0			

The resulting rewards of wallets A and B are computed using equations (2)-(3). In particular:

$$\mu_{XY}^{A} = 50^1(1 + 100e^{0.01095*50}(1 + 0.007023 * 60^1) + 200e^{0.01095*20}(1 + 0.007023 * 0^1) + 50e^{0.01095*100}(1 + 0.007023 * 200^1))^1 = 42,755.08$$

$$\mu_{XY}^{A} = 50^1(1 + 50e^{0.01095*10}(1 + 0.007023 * 100^1) + 100e^{0.01095*0}(1 + 0.007023 * 0^1))^1 = 9,798.23$$

implying that epoch (weekly) rewards in USDT-USDC pool for each of the two liquidity providers are: $\pi_{XY,t}^{A} = \frac{42{,}755.08}{42{,}755.08+9{,}798.23} \times 1,235,764.56 \times 14.59\% = 146,683$ *VRSW*, whereas $\pi_{XY,t}^{B} = \frac{9{,}798.23}{42{,}755.08+9{,}798.23} \times 1,235,764.56 \times 14.59\% = 33,615$ *VRSW*.

5 Implementation of reward boost calculation

In the absence of gas costs, the computation of aggregate reward boosts accruing to wallet i during a particular time period would amount to a straightforward aggregation of block-level reward boost in Eqs. (2) and (3). In the presence of gas costs, computing the evolution of reward boost for every wallet every block is prohibitively expensive. This section outlines an algorithm that achieves rewards accrual as in Eqs. (2) and (3), while requiring very few calculations. In particular, updates of reward boost accruals are performed:

- only during blocks in which certain events (trading in the pool; changes in pool allocation points and discretionary reward parameter; staking and unstaking of LP tokens; staking and unstaking of *VRSW* tokens, locking and unlocking *VRSW* tokens);

- only for the wallet directly involved in a particular event as well as for aggregation of all wallets – but, importantly, not for any other wallet.

In the remainder of this section, we describe the calculations that need to be performed and the updates that need to be recorded for each type of event.

Let us define the following quantities:

- Series of events (i.e. stake/lock VRSW, stake LP token, claim rewards etc.): e_τ, τ being an integer index of events;
- Timestamp of event e_τ: t_{e_τ};
- Total rewards allocated at time t across all wallets, $M_{XY,t} = \sum_j \mu^j_{XY,t}$.

Note first that $\mu^i_{XY,t}$ stays unchanged if wallet i does not trigger any event at time t. Similarly, $M_{XY,t}$ stays unchanged if no wallet triggers any event at time t.

Let us consider equation (2), in which the denomination of reward asset d takes the form of VRSW:

$$\pi^{VRSW_i}_{XY,t} = W^{VRSW}_{XY,t} \frac{\mu^i_{XY,t}}{\sum_j (\mu^j_{XY,t})} = R^A_t \eta_{XY} \frac{\mu^i_{XY,t}}{M_{XY,t}}. \tag{4}$$

For simplicity, we will omit $VRSW$ and XY henceforth.

Now let us consider the series of events and find the way of rewards calculation only at times during which some event is triggered. For this, suppose first that we would like to calculate rewards accrued to wallet i in time interval $[t_{e_\tau}; t_{e_{\tau'}}]$ (where $\tau' > \tau$), assuming that both events were triggered by wallet i, and no other events were triggered in between t_{e_τ} and $t_{e_{\tau'}}$.

$$\pi^i_{t_{e_{\tau'}}} = \pi^i_{t_{e_\tau}} + \int_{t=t_{e_\tau}}^{t_{e_{\tau'}}} R^A_t \eta \frac{\mu^i_t}{M_t} dt = \pi^i_{t_{e_\tau}} + \frac{\mu^i_{t_{e_\tau}}}{M_{t_{e_\tau}}} \int_{t=t_{e_\tau}}^{t_{e_{\tau'}}} R^A_t \eta dt. \tag{5}$$

Now suppose that only events e_τ and $e_{\tau'}$ (for $\tau < \tau'$) were triggered by wallet i. Every other event $\tau*$ in between τ and τ' was triggered by one of the other wallets and, thus, has modified only value of M_t, but not the value of μ^i_t.

$$\pi^i_{t_{e_{\tau'}}} = \pi^i_{t_{e_\tau}} + \sum_{j=\tau+1}^{\tau'} (\pi^i_{t_{e_j}} - \pi^i_{t_{e_{j-1}}}) = \pi^i_{t_{e_\tau}} + \mu^i_{t_{e_\tau}} \sum_{j=\tau+1}^{\tau'} (\frac{1}{M_{t_{e_{j-1}}}} \int_{t=t_{e_{j-1}}}^{t_{e_j}} R^A_t \eta dt). \tag{6}$$

As a result, we can effectively calculate rewards for wallet i only at the time of event triggered by that wallet.

In light of the discussion above, the algorithm will consist of the following general steps:

1. When some event $e_{\tau'}$ is triggered by wallet i, update state (both global and local for wallet i) variables:
   - VRSW cumulative rewards in the pool as: $\mathbb{R}(t_{e_{\tau'}}) = \mathbb{R}(t_{e_\tau}) + \int_{t_{e_\tau}}^{t_{e_{\tau'}}} R^A_t \eta dt$,
   - Rewards for wallet i as: $\pi^i_{t_{e_{\tau'}}} = \pi^i_{t_{e_\tau}} + \mu^i_{t_{e_\tau}} E^{t_{e_{\tau'}}}_{t_{e_\tau}}$, where $E^{t_{e_{\tau'}}}_{t_{e_\tau}} = \sum_{j=\tau+1}^{\tau'} (\frac{1}{M_{t_{e_{j-1}}}} \int_{t=t_{e_{j-1}}}^{t_{e_j}} R^A_t \eta dt)$.
2. Process the event (i.e. stake/lock/unstake VRSW, stake/unstake LP tokens etc.)
3. Update both $\mu^i_{t_{e_{\tau'}}}$ and $M_{t_{e_{\tau'}}}$ according to equation (3).