)

Coinecta Vesting Contracts

Vesting contracts are a way to guarantee tokens are only released into circulation at a certain rate. This can be beneficial in multiple scenarios, for example if tokens are sold during a presale event with a price under the targeted IDO price any participant in the presale has an unfair advantage and could negatively affect the market.

1 High Level Requirements

For Coinecta we have the following High Level Requirements (HLR) that we would like the contracts to fulfill.

1. Unvested tokens need to start vested at a certain point in time

2. Tokens need to vest at a certain linear rate

3. Vested tokens are to be claimed by the user with a vesting NFT

4. The vesting NFT should be claimed by the user

5. The admin should be able to specify how much each user can claim

6. The user can decide how much to lock behind a vesting NFT, so their allocation can be split into multiple vesting NFT's

7. The admin can remove unclaimed tokens after a certain time has passed.

8. If a user claims tokens after initial release time has passed vested assets are released along with vesting NFT

1.a HLR 1 - Unvested tokens need to start vested at a certain point in time

It should be possible to control when tokens locked in the contract start being vested, to support preventing tokens coming into circulation before for example an IDO.

1.b HLR 2 - Tokens need to vest at a certain linear rate

Once the initial time of vesting (HLR1) has been reached the unvested tokens locked in the contract should be vested gradually at a specific linear rate, such that every X amount of time Y% of the initially unvested tokens are to be vested.

1.c HLR 3 - Vested tokens are to be claimed by the user with a vesting NFT

Those tokens that are vested according to (HLR1) and (HLR2), are to be claimed by a unique NFT. This allows the unvested tokens to be transferrable even though they are still locked.

1.d HLR 4 - The vesting NFT should be claimed by the user

The vesting NFT described in (HLR3) is not airdropped to eligible users but must be claimed/minted by the user. Preventing high ADA costs incurring on the project running the vesting setup.

1.e HLR 5 - The admin should be able to specify how much each user can claim

When the user claims their vesting NFT the amount available to them to lock should be defined by the admin of the protocol. The admin also decides the parameters such as vesting rate.

1.f HLR 6 - The user can decide how much to lock behind a vesting NFT, so their allocation can be split into multiple vesting NFT's

When the user claims their vesting NFT they can decide to lock a smaller amount than their full allocation, giving them the option to divide their allocation into smaller amounts.

1.g HLR 7 - The admin can remove unclaimed tokens after a certain time has passed.

If any user does not claim their vesting NFT in time the admin can recover the unclaimed tokens, so they can be used for other purposes.

1.h HLR 8 - If a user claims tokens after initial release time has passed vested assets are released along with vesting NFT

If a user claims their vesting NFT after the initial release time has passed (HLR1) they will receive vested tokens following (HLR2) along with the vesting NFT. If all tokens are vested no vesting NFT is minted and the user receives all tokens.

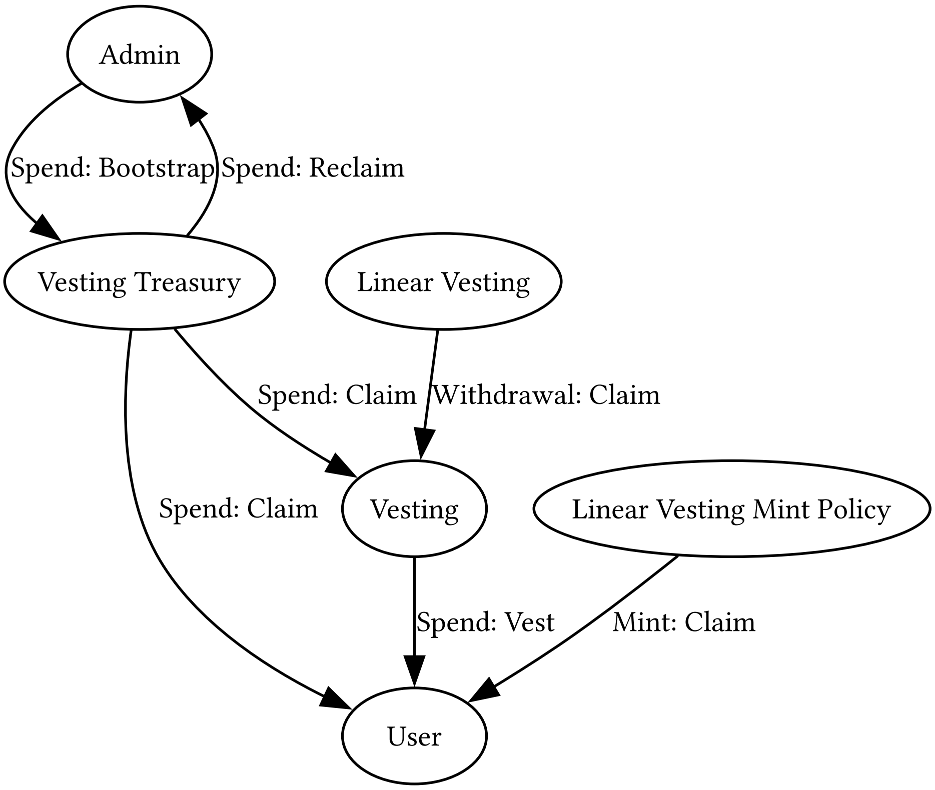
2 Protocol overview

Test

2.a Actors

Actor name	**Actor description**
Admin	Admin (wallet) bootstrapping the protocol
Vesting Treasury	Allocation smart contract containing the allocation for all users in a merkle tree
Vesting	Vested contract holding unvested tokens for the user
User	User (wallet) interacting with the protocol
Linear Vesting	Contract controlling the correct settings for a linear vesting lock
Linear Vesting Mint Policy	Contract controlling the correct minting and burning of vesting nft's

2.b Visual Protocol Overview


Admin
Spend: Bootstrap
Spend: Reclaim
Vesting Treasury
Linear Vesting
Spend: Claim
Withdrawal: Claim
Spend: Claim
Vesting
Linear Vesting Mint Policy
Spend: Vest
Mint: Claim
User