

Vesting smart contract

What is going on

Gaimin is going to have a private GMRX token sales round. It will happen near the date of the public sale (most probably a little bit earlier). During that private round, investors will get some amount of GMRX tokens. To prevent them from selling all the tokens once public sales begin Gaimin needs to have some vesting period for investors. That means that investors will own GMRX tokens but will be able to sell them only after a specific date, and there will more than one unlock date, so investors will get access to their tokens in batches.

An example:

Gaimin will have a public sale on the 1st of November.

On the 20th of October Gaimin makes private sales for investors, with a 1-month cliff and 4 vesting periods of 3 months each.

Alise gets 100000 GMRX tokens on her locked wallet during private sales.

So she will be able to withdraw 25000 GMRX on the 1st of December.

25000 GMRX on the 1st of March 2022

25000 GMRX on the 1st of June 2022

25000 GMRX on the 1st of September 2022

Gaimin is going to have multiple investors groups with different cliff and vesting periods.

Basic contracts overview

We already have GMRX tokens smart contract deployed.

On the mainnet GMRXs are bridged from Ethereum: <https://polygonscan.com/address/0x73f56124a34e0214067b7e5f42a132b3ea072014>

On testnet it is a standard ERC20 smart contract: <https://mumbai.polygonscan.com/token/0x9037dd49bed73b3b2a99fce722d2f9207027bc3e>

There will be 2 smart contracts: Time Locked Wallet (TLW) that will hold GMRXes for one user, and Factory that will create new TLWs for every user. The Factory will create TLWs as [minimal clones](#), so the original TLW should already be deployed before the Factory would be used.

Original mainnet TLW: --missing--

Original testnet TLW: <https://mumbai.polygonscan.com/address/0x06E5D057280B9Cba511809c393c8c66C64ad675b>

TLW API:

Read functions:

`_lockBoxes(int)` - accept index of locking period starting from 0, return information about this locking period.

Response example:

`amount uint256 : 1000`

`unlockTime uint256 : 1650000000`

`paid bool : false`

`_owner()` - return address of beneficiary who will get the locked tokens.

`_tokenAddress()` - address of locked tokens contract (should be the address of GMRXes).

`lockedAmount()` - return total locked tokens left on this wallet.

`readyToWithdraw()` - return amount of tokens that are ready to be withdrawn.

Write functions:

`initialize(...)` - could be called only once, and will be called by the Factory automatically.

input parameters:

`owner` - beneficiary for whom tokens will be locked

`tokenAddress`

`amount`

`numberOfPeriods` - number of the locked periods (number of withdrawals)

`firstUnlockTime` - in second

`periodDuration` - locked period duration

validations:

`owner` should be address and not smart contract

`numberOfPeriods` > 0

`firstUnlockTime` should be in the future

check that this TLW already have tokens on the balance

`withdraw()` - transfer to the `owner` tokens from all unlocked periods. It could be more than one unlocked period.

Factory mainnet: --missing--

Factory testnet: - <https://mumbai.polygonscan.com/address/0x07b1e78Ea789FD6755b6C62D23301FC7074be6b0>

Factory API:

constructor(tokenAddress, tlwAddress)

Read functions:

_tlwAddress - return address of original TLW to make clones from.

_tokenAddress() - address of locked tokens contract (should be the address of GMRXes).

getWallets(address) - accept user address and return list of TLWs created to and created by this user.

owner - address of the Factory owner

Write functions:

newTimeLockedWallet(...)

input parameters:

owner - beneficiary for whom tokens will be locked

amount

numberOfPeriods - number of the locked periods (number of withdrawals)

firstUnlockTime - in second

periodDuration - locked period duration

validations:

owner should be address and not smart contract

amount > 0

numberOfPeriods > 0

firstUnlockTime should be in the future

periodDuration >= 1 day for mainnet (or >= 1 minute for testnet)

check that token allowance is correct

clone original TLW to the new address, transfer tokens to new TLW and return address.

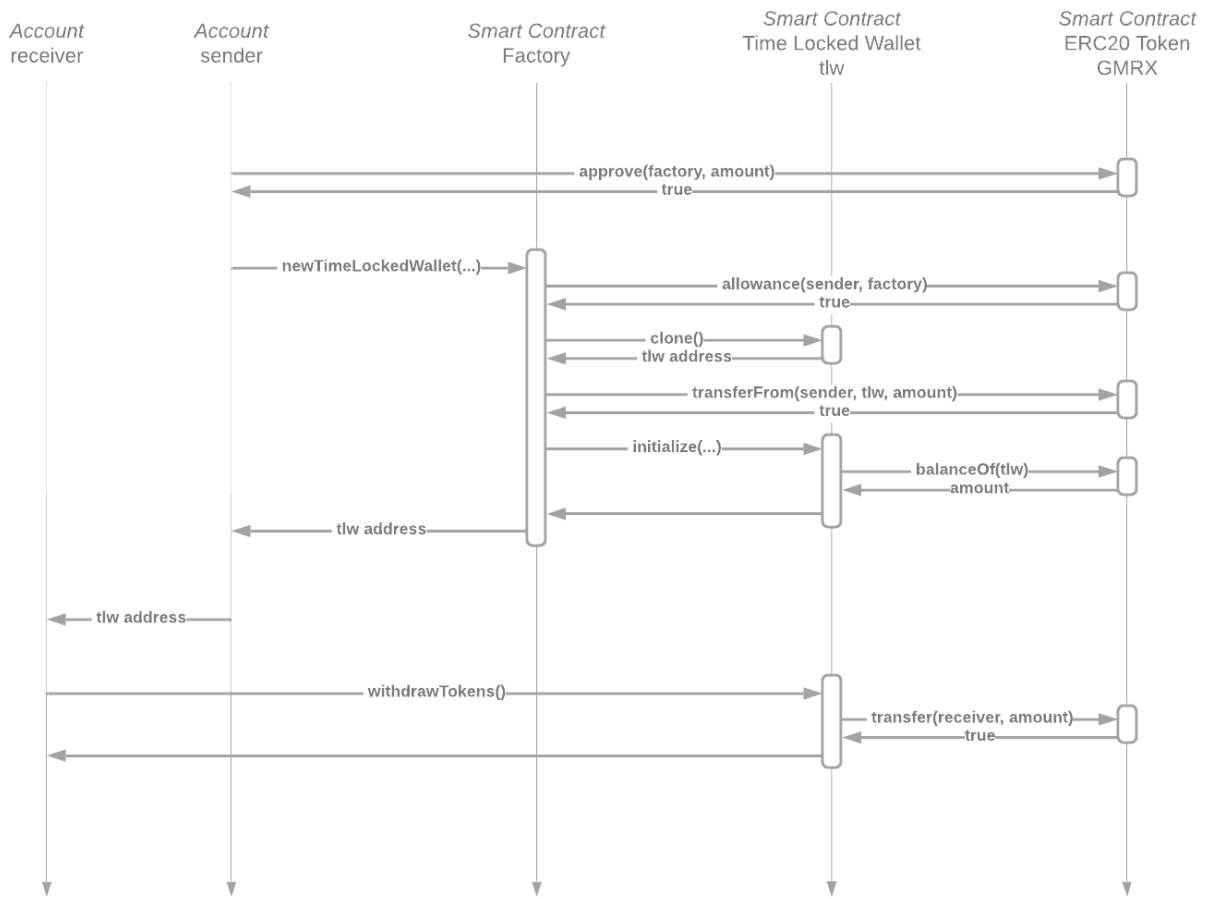
renounceOwnership() - remove owner from contract, so contract will stay immutable after that

setTokenAddress(address)

setTLWAddress(address)

transferOwnership(address) - transfer ownership to new address

Factory flow



Example

Let's assume that Alice wants to lock 1000 GMRXes for Bob. Tokens should be locked in 4 equal periods, Bob should be able to get his money starting from 09.10.2021 and then each day gets another 25% of the locked amount.

Alice address: 0xBD8911B2967efE7C98A731f5332A76526902AEe4

Bob address: 0xF1746359eFeaA3468e4C521187c57D3ee189F561

GMRX: 0x9037dD49BeD73b3b2a99fCE722d2F9207027Bc3e

Original TLW: 0x06E5D057280B9Cba511809c393c8c66C64ad675b

Factory: 0x07b1e78Ea789FD6755b6C62D23301FC7074be6b0

1. Alice approve that Factory could spend 1000 GMRXes

Alice makes

`approve(0x07b1e78Ea789FD6755b6C62D23301FC7074be6b0, 10000000000000000000)`

call to 0x9037dD49BeD73b3b2a99fCE722d2F9207027Bc3e.

10000000000000000000 - is 1000 GMRX including 10^{18} decimals

<https://mumbai.polygonscan.com/tx/0x183cd9380f54a46f858a0dead8e33589ed2662ef184d23208d63e9947e06048a>

?

From:

0xbd8911b2967efe7c98a731f5332a76526902aee4

?

To:

Contract 0x9037dd49bed73b3b2a99fce722d2f9207027bc3e

?

Value:

0 MATIC (\$0.00)

?

Transaction Fee:

0.000140679 MATIC (\$0.00)

?

Gas Limit:

70,339

?

Gas Used by Transaction:

46,893 (66.67%)

?

Gas Price:

0.000000003 MATIC (3 Gwei)

?

Nonce

234

Position

2

?

Input Data:

| # | Name | Type | Data |
|---|---------|---------|--|
| 0 | spender | address | 07b1e78ea789fd6755b6c62d23301fc7074be6b0 |
| 1 | amount | uint256 | 10000000000000000000 |
| | | | |

2. Alice creates a new TLW through Factory

Alice makes

`newTimeLockedWallet(0xF1746359eFeaA3468e4C521187c57D3ee189F561, 10000000000000000000, 4, 1633726800, 86400)`

call to 0x07b1e78Ea789FD6755b6C62D23301FC7074be6b0

10000000000000000000 - is 1000 GMRX including 10¹⁸ decimals

1633726800 - is timestamp of 09.10.2021

86400 - one day in seconds

<https://mumbai.polygonscan.com/tx/0xc2b73ebd23b61fae7bd203747ea7c505e339b031edc6f7330c03ab4f22b5b831>

?

From:

0xbd8911b2967efe7c98a731f5332a76526902aee4

?

Interacted With (To):

Contract 0x07b1e78ea789fd6755b6c62d23301fc7074be6b0

✓

?

Tokens Transferred:

►

From 0xbd8911b2967ef... To 0x4ebf558387857... For 1,000

Matic Gaimin... (M_GMRX)

?

Value:

0 MATIC (\$0.00)

?

Transaction Fee:

0.001276659 MATIC (\$0.00)

?

Gas Limit:

440,553

?

Gas Used by Transaction:

425,553 (96.6%)

?

Gas Price:

0.000000003 MATIC (3 Gwei)

?

Nonce

Position

235

0

?

Input Data:

| # | Name | Type | Data |
|---|-----------------|---------|--|
| 0 | owner | address | f1746359efaaa3468e4c521187c57d3ee189f561 |
| 1 | amount | uint256 | 10000000000000000000 |
| 2 | numberOfPeriods | uint256 | 4 |
| 3 | firstUnlockTime | uint256 | 1633726800 |
| 4 | periodDuration | uint256 | 86400 |

With this transaction new TLW was created 0x4ebf5583878576b987386b82832ae7343c548acc

Now it holds all 1000 GMRXes

Contract 0x4ebF5583878576b987386b82832AE7343c548acC

Contract Overview

Balance: 0 MATIC

Token: \$0.00

More Info

My Name Tag: Not Available

Contract Creator: 0x07b1e78ea789fd6755... at txn 0xc2b73ebd23b61fae7b...

Transactions Internal Txns **ERC-20 Token Txns** Contract Events

Latest 1 ERC-20 Token Transfer Event

| Txn Hash | Age | From | To | Value | Token |
|-------------------------|-------------|-------------------------|----------------------------|-------|--------------------------|
| 0xc2b73ebd23b61fae7b... | 11 mins ago | 0xbd8911b2967efe7c98... | IN 0x4ebf5583878576b987... | 1,000 | Matic Gaimin... (M_GMRX) |

Timestamp:

4 mins ago (Oct-11-2021 08:06:09 AM +UTC)

From:

0xf1746359efaaa3468e4c521187c57d3ee189f561

Interacted With (To):

Contract 0x4ebf5583878576b987386b82832ae7343c548acc

Tokens Transferred: 3

From 0x4ebf558387857... To 0xf1746359efaaa3... For 250 Matic Gaimin... (M_GMRX)
From 0x4ebf558387857... To 0xf1746359efaaa3... For 250 Matic Gaimin... (M_GMRX)
From 0x4ebf558387857... To 0xf1746359efaaa3... For 250 Matic Gaimin... (M_GMRX)

Value:

0 MATIC (\$0.00)

Transaction Fee:

0.000448755 MATIC (\$0.00)

Gas Limit:

151,549

Gas Used by Transaction:

149,585 (98.7%)

Gas Price:

0.000000003 MATIC (3 Gwei)

Nonce

Position

340 4

Input Data:

Function: withdraw() ***

MethodID: 0x3ccfd60b

View Input As

Links:

1. Github repository <https://github.com/Gaimin-io-Limited/gmr-x-vesting>
2. ERC20 token <https://docs.openzeppelin.com/contracts/4.x/erc20>
3. Minimal Clone https://docs.openzeppelin.com/contracts/4.x/api/proxy#minimal_clones

4. Ownable extension <https://docs.openzeppelin.com/contracts/4.x/api/access#Ownable>
5. Initializable extension <https://docs.openzeppelin.com/contracts/4.x/api/proxy#Initializable>