Gimmer Token Sale Smart Contracts Specifications

Lucas Assis Ribeiro

December 27th, 2017

1 Gimmer Token

Gimmer Token is the contract for the GMR Tokens in the Gimmer ecosystem.
It's a Mintable Token, meaning the Crowdsale actually creates the tokens before each transaction.

1.1 Parameters

Maximum Token Cap: **100.000.000 GMR**
Maximum Token Cap (With Comission): **110.000.000 GMR**
Name: **GimmerToken**
Decimals: **18**
Symbol: **GMR**

2 Gimmer Token Sale

The Token Sale contract manages contributions to the Pre Sale and Public Sale.

2.1 Constructor

Fund Wallet:
KYC Manager Wallet:
Wei Limit Without KYC: **30 ETH**
Max Transaction Gas: **50 GWEI**

2.2 Parameters

2.2.1 Token

Maximum possible minted tokens: **110.000.000 GMR**
Total tokens for sale: **100.000.000 GMR**
Pre Sale Cap/Crowd Sale Soft Cap: **4.800 ETH/15.000.000 GMR**

Crowd Sale Hard Cap: **35.000 ETH**
Minimum Crowdsale transaction: **0.1 ETH**

2.2.2 Pricing

Base conversion: **1 ETH = 2500 GMR**
Pre Sale 30 Eth Rate: **1 ETH = 3125 GMR**
Pre Sale 300 Eth Rate: **1 ETH = 3250 GMR**
Pre Sale 3000 Eth Rate: **1 ETH = 3500 GMR**
Crowd Sale Week 1 Rate (20% Bonus): **1 ETH = 3000 GMR**
Crowd Sale Week 2 Rate (15% Bonus): **1 ETH = 2875 GMR**
Crowd Sale Week 3 Rate (10% Bonus): **1 ETH = 2750 GMR**
Crowd Sale Week 4 Rate (5% Bonus): **1 ETH = 2625 GMR**

2.2.3 Dates

Pre Sale Start Week Date: **12:00 17th February 2018**
Pre Sale End Date: **12:00 31th January 2018**
Crowd Sale Week 1 Date: **12:00 1st February 2018**
Crowd Sale Week 2 Date: **12:00 8th February 2018**
Crowd Sale Week 3 Date: **12:00 15th February 2018**
Crowd Sale Week 4 Date: **12:00 22th February 2018**
Crowd Sale End Time (Week 4's end): **12:00 1st March 2018**

2.3 Specifications

2.3.1 Fair-Chance

To make sure that no one gets to buy tokens ahead of anyone, we included a transaction gas price limit.
This makes sure that no address can send exorbitant transaction prices to get mined faster and get priority in the acquisition of tokens.
Start Value: **50 GWei**
As the Ethereum network is potentially unstable, this value can be changed by the KYC Manager to better reflect current network gas prices.

2.3.2 KYC

Know-Your-Customer rules how much a wallet can spend on the contract, depending on which stage of the contract were in.
In Pre-Sale, the wallet needs KYC approval to do any transactions.
In Crowd Sale, any wallet can spend up to 30 Eth before needing approval.

The KYC Manager Wallet can only approve/disapprove users and change the max transaction gas price. The permissions are limited as to not facilitate hacking.
The owner of the contract can change the KYC Manager wallet to a different one if any issue comes (compromised KYC account/changing the account that the website uses).

2.3.3 (Crowd) Sale Wei Limit Without KYC

The limit of Wei an user can spend in this contract during the crowd sale before being blocked and requiring KYC approval.
This can be updated by the owner of the contract by using setSaleWeiLimitWithoutKYC.
Start Value: **30 ETH**

2.4 Bug Bounty

2.4.1 Project

Github page, where the latest version of all contracts are stored - `https://github.com/GimmerBot/gmr-token`
Generated coverage tests output data can be found together with the contracts.

2.4.2 Files

GimmerToken.sol - Contract file for the GMR token. Basically a MintableToken with the addition that it can only be traded after minting is complete. `https://github.com/GimmerBot/gmr-token/blob/master/contracts/GimmerToken.sol`
GimmerTokenSale.sol - Manages PreSale and Crowd Sale transactions - `https://github.com/GimmerBot/gmr-token/blob/master/contracts/GimmerTokenSale.sol`
GimmerTokenSale.js - Automated tests for both the Token Sale and GMR Token contracts. - `https://github.com/GimmerBot/gmr-token/blob/master/test/GimmerTokenSale.js`

2.4.3 Bugs

Found bugs will be rewarded proportionately to their severity.

2.4.4 Tips

Recommendations and tips about the structure of the contract can also be accepted, though rewards will vary based on their usefulness to us, with the rules we have to implement.

2.5 Testing

2.6 List

- Token Sale Contract:
  - Should not deploy the contract with any incorrect constructor arguments
  - Should not accept Eth after sale's final date
  - Wallet that deployed the contract should be the owner of it
  - Received funds should arrive at the fundWallet
  - Should reject calling finishContract() before the sale's end date
  - Should be able to call finishContract() after final date
  - Fund wallet should end with 10% of token cap
- KYC:
  - Should only flag KYC when executing from the KYC Manager wallet
  - Should execute setKYCManager from the owner account to change the KYC manager
- Fair Chance:
  - Should accept transaction at exactly max gas price (50 GWei)
  - Should reject transaction above max gas price (50.000000001 GWei)
  - Should change the max tx gas price to 100 GWei
  - Should accept transaction at exactly max gas price (100 GWei)
  - Should reject transaction above max gas price (100.000000001 GWei)
- PreSale:
  - Should not accept Eth before PreSale start date
  - Should not accept Eth if wallet is not KYCed
  - Should reject buying less than the PreSale minimum (29.999999999999999999 ETH)
  - Should accept buying exactly the PreSale minimum (30 ETH)
  - Should buy 30 ETH to receive 25% bonus on PreSale
  - Should buy 300 ETH to receive 30% bonus on PreSale
  - Should buy 3000 ETH to receive 40% bonus on PreSale
- Sale:
  - Should not accept Eth before Sale start date
  - Should not accept more than 30 Eth if user is not KYCed

  - Should not accept less than minimum 0.1 Eth
  - 1st Week should have 20% bonus
  - 2nd Week should have 15% bonus
  - 3rd Week should have 10% bonus
  - 4th Week should have 05% bonus

- Gimmer Token:
  - Should be owned by GimmerTokenSale contract before calling finishContract
  - Should have mintingFinished set to false before calling finishContract
  - Should not be able to call mint in the token contract
  - Should be tradeable ONLY after contract's ending

2.6.1 Automated Tests

Results
Should reject deploying the Token Sale contract with an empty fund wallet (113ms)
Should reject deploying the Token Sale contract with an empty KYC wallet (161ms)
Should reject deploying the Token Sale contract with no Wei Limit (170ms)
Should reject deploying the Token Sale contract with no Max Tx Gas Price (196ms)
Should deploy the Token Sale contract (353ms)
Should reject calling finishContract() during period before sale (74ms)
Contract should not be on finished state
Contract should not have finished presale
Should be the owner of the contract
Should await untill the start of the PreSale period (405ms)
Should reject calling finishContract() during presale period (72ms)
Gimmer Token should be owned by Gimmer Token Sale (61ms)
Gimmer Token should have mintingFinished be false (62ms)
Should reject minting 1 GMR from the fund wallet (or any other wallet) (60ms)
Should reject invoking finish minting from the fund wallet (or any other wallet) (58ms)
Should reject changing the KYC Manager to an empty wallet (64ms)
Should change KYC Manager to correct wallet (163ms)
Should not approve KYC when executing from old KYC Manager (test wallet that was input on constructor) (73ms)
Should revert when invoking approveUserKYC with an empty wallet (107ms)
Should revert when invoking disapproveUserKYC with an empty wallet (115ms)
Should approve KYC of the investor wallet when executing from KYC Manager wallet (187ms)
Should approve KYC of the 2nd investor wallet and read the KYC logs (188ms)
Should disapprove KYC of the 2nd investor wallet and read the KYC logs (173ms)
Should reject buying 1 token more than PreSale cap: (15000000.000000000000000001 GMR/4285.714285714285717785 ETH) (484ms)

Should reject buying less than minimum during presale (29.999999999999999999/30 ETH) (214ms)

Should increase tokensSold (682ms)

Should log purchase (629ms)

Should assign tokens to sender (709ms)

Should forward funds to wallet (1134ms)

Should buy exactly the minimum (30 ETH) from kyced wallet (737ms)

Should reject buying 500 ETH from un-kyced wallet (236ms)

Should increase userWeiSpent() during presale (697ms)

Should buy 30 ETH worth of tokens at (PreSale 25% Bonus) (713ms)

Should buy 300 ETH worth of tokens at (PreSale 30% Bonus) (696ms)

Should buy 3000 ETH worth of tokens at PreSale (PreSale 40% Bonus) (656ms)

Should await until the end of the PreSale (366ms)

Contract should not be on finished state

Contract should have finished presale

Should reject calling finishContract() during nothing-day period (59ms)

Should reject buying 50 ETH during nothing day (221ms)

Should reject transfering 100 GMR from Investor 1 to 2 (77ms)

Should reject approving Investor 3 to manage 100 GMR from Investor 1 (96ms)

Should reject increaseApproval Investor 3 to manage 100 GMR from Investor 1 (79ms)

Should reject decreaseApproval Investor 3 to manage 100 GMR from Investor 1 (86ms)

Should await until the 1st week of public sale (328ms)

Should reject buying 1 token more than Token Sale cap: (100000000.000000000000000001 GMR) (425ms)

Should reject calling finishContract() during Week 1 of crowd sale period (60ms)

Should reject buying less than the minimum during the crowd sale (205ms)

Should accept buying exactly the minimum (568ms)

Should accept transaction with max gas price (50 GWei) (556ms)

Should reject transaction with gas price above the maximum (50.000000001 GWei) (61ms)

Should reject changing the max tx gas price to 0 GWei (96ms)

Should change the max tx gas price to 100 GWei (159ms)

Should accept transaction with max gas price (100 GWei) (563ms)

Should reject transaction with gas price above the maximum (100.000000001 GWei) (65ms)

Should reject buying more than allowed without KYC (275ms)

Should buy 1 ETH worth of tokens at Week 1 (20% Bonus) (711ms)

Should call buyTokens (632ms)

Should increase userWeiSpent() during crowd sale (684ms)

Should await until the 2nd week of public sale (340ms)

Should reject buying more than allowed without KYC (265ms)

Should buy 1 ETH worth of tokens at Week 2 (15% Bonus) (688ms)

Should await until the 3rd week of public sale (338ms)

Should reject buying more than allowed without KYC (269ms)
Should buy 1 ETH worth of tokens at Week 3 (10% Bonus) (676ms)
Should reject setting the saleWeiLimitWithoutKYC to 0 ETH (92ms)
Should change saleWeiLimitWithoutKYC to 30 ETH (157ms)
Should reject buying 30.000000000000000001 ETH worth of tokens without KYC (250ms)
Should buy 30 ETH worth of tokens without KYC (735ms)
Should await until the 4th week of public sale (323ms)
Should reject calling finishContract() during Week 4 of crowd sale (54ms)
Should reject buying more than allowed without KYC (259ms)
Should buy 1 ETH worth of tokens at Week 4 (5% Bonus) (629ms)
Contract should not be on finished state
GimmerTokenSale presale should have ended
Should await until the end of the public sale (338ms)
Contract should have finished presale
Contract should be on finished state
Should reject buying 1 Wei after sale ends (235ms)
Should call finishContract (362ms)
Should reject calling finishContract() after already calling it (89ms)
Should reject calling transfer with empty wallet (111ms)
Should reject calling transfer to GMR contract (166ms)
Should transfer 100 GMR from Investor 1 to 2 (204ms)
Should allow Investor 3 to transfer 100 GMR from Investor 1 (123ms)
Should increase the allowed amount to be managed by Investor 3 by 70 GMR (119ms)
Should decreaseApproval the transfer to be managed by Investor 3 by 20 GMR (133ms)
Should transferFrom the total approved value of 150 GMR (250ms)
Should reject minting 1 GMR from the fund wallet (even though it owns the GimmerToken contract now) (45ms)

89/89 passing (33s)

2.7 Variables

2.7.1 Tokens Sold

The total amount of tokens sold through the Token Sale contract.

2.7.2 Wei Raised

The total amount of Wei that has passed through the Token Sale contract.

2.7.3 KYC Manager Wallet

The wallet that is able to approve the KYC of supporters.
Can be changed by the owner of the contract by calling setKYCManager.

Start Value: **0xbb3628807c424b58b8db1ae5e304255b84581af1**

2.8 Constants

2.8.1 Fund Wallet

The wallet that the Wei funds will be forwarded to.
Value: **0x8c789463412b2697185e103ded5f75b9b3931a84**

2.8.2 One Million

Reference value to calculate the Pre Sale Token cap
Value: **1.000.000**

2.8.3 Pre Sale GMR Token Cap

The total amount of tokens that can be sold during the PreSale period.
Value: **15.000.000 tokens**

2.9 Public State-Changing Functions

2.9.1 ()

Fallback function, calls buyTokens().

2.9.2 buyTokens()

Buys tokens based on the amount of Wei sent to the contract.
KYC approval is needed for the transaction to fullfill.

2.9.3 approveUserKYC(address user)

Approves the KYC for the specified wallet address.
Only the KYC Manager wallet can execute this.

2.9.4 disapproveUserKYC(address user)

Disapproves the KYC for the specified wallet address.
Only the KYC Manager wallet can execute this.

2.9.5 setKYCManager(address newKYCManager)

Changes the KYC Manager wallet address to a new one.
Only the Owner of the contract can execute this.

2.9.6 transferOwnership(address newOwner)

Changes the owner of the contract to a new one.
Only the Owner of the contract can execute this.

2.9.7 updateMaxTxGas(uint256 newMaxTxGas)

Updates the maximum gas price allowed in transactions.

2.9.8 finishContract()

Ends operation of the contract after all the sales are done.

2.10 Events

2.10.1 TokenPurchase

Occurs everytime a user sends Wei to the contract.
Sends the address of the person buying, the amount of Wei sent and the total amount of tokens bought.

2.10.2 KYC

Occurs everytime the KYC Manager changes the KYC enabled state of an user.
Sends the address of the wallet that the manager changed and the state that it was changed to (approved/disapproved).