**Requirements Specification**

**for**

QNTU

**Version 1.0**

**IBL**

**03-15-2018**

**Table of Contents**

[**Introduction**](#_l5dio95tavbm) **4**

[**Purpose**](#_3znysh7) **4**

[**Product Scope**](#_tw4kw6a1n1tl) **4**

[**References**](#_hwlvj3k98pll) **4**

[**Overall Description**](#_8necbvgbvprz) **4**

[**QNTU Functions Description**](#_tb1wpql2x2kf) **10**

[**transferOwnership(address \_newOwner)**](#_rcaltsg0pyh8) **10**

[**changeTokenInformation(string \_name, string \_symbol)**](#_r7ddobm5h604) **10**

[**transfer(address \_to, uint \_value)**](#_hwhm30ltiboz) **10**

[**transferFrom(address \_from, address \_to, uint \_value)**](#_g2uqunzhc8gs) **11**

[**approve(address \_spender, uint \_value)**](#_cxinlldup4rl) **11**

[**increaseApproval(address \_spender, uint \_addedValue)**](#_e13t99nux891) **12**

[**decreaseApproval(address \_spender, uint \_subtractedValue)**](#_rlayr0y8tkhi) **12**

[**allowance(address \_owner, address \_spender)**](#_vwno6lhuzj7k) **13**

[**freezeTransfer()**](#_4ba8ojytmrp) **13**

[**unfreezeTransfer()**](#_sl6ur9mlsruq) **13**

[**freezeUpgrade()**](#_6xacb0jogbev) **13**

[**unfreezeUpgrade()**](#_keivs1ft3pnm) **14**

[**changeUpgradeMaster(address \_newMaster)**](#_gvozjx49w09a) **14**

[**changeUpgradeAgent(address \_newAgent)**](#_m3y6c7g5xce6) **14**

[**upgrade()**](#_sdhjbygck9dz) **15**

[**forceUpgrade(address[] \_holders)**](#_5ualzcpacv2r) **15**

[**transferToContract(address \_to, uint \_value)**](#_uwpq7a9u1frv) **16**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

*This document is for designing and developing QNTU. A ERC20 standard token smart contract on Ethereum.*

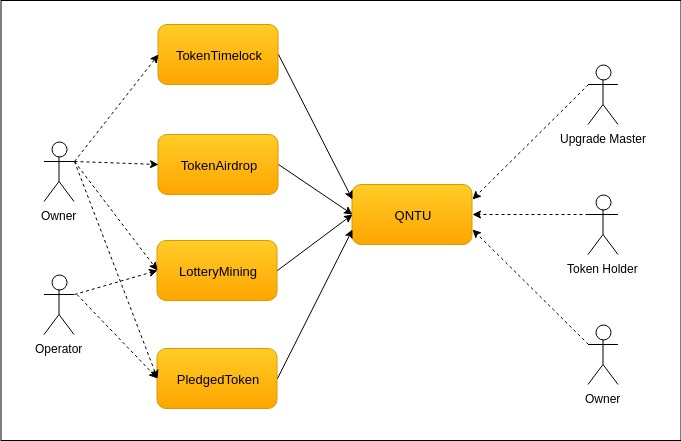
## **Product Scope**

*QNTU is the utility token of the Quanta ecosystem that the various participants will be able to put up as collateral. QNTU forms the basis of an internal econonomy that ensures the security of the network.*

## **References**

*Quanta White Paper: https://www.quantaplc.im/Quanta.pdf*

# **Overall Description**



# QNTU Functions Description

## transferOwnership(address \_newOwner)

* Description  
  - Allows the current Owner to transfer control of the smart contract to a new Owner.

- Caller is only Owner.

## changeTokenInformation(string \_name, string \_symbol)

* Description  
  - Change token name and symbol.

- Caller is only Owner.

## transfer(address \_to, uint \_value)

* Description  
  - Token Holder transfer their own tokens to another address.

## transferFrom(address \_from, address \_to, uint \_value)

* Description  
  - The person who are authorised can transfer tokens from the authorised address to another address.

## approve(address \_spender, uint \_value)

* Description  
  - Token Holder authorise another address to own a specific amount of their tokens.

## increaseApproval(address \_spender, uint \_addedValue)

* Description  
  - Token Holder increase the amount of tokens that the person who are authorised can transfer.

## decreaseApproval(address \_spender, uint \_subtractedValue)

* Description  
  - Token Holder decrease the amount of tokens that the person who are authorised can transfer.

## allowance(address \_owner, address \_spender)

* Description  
  - Get the remaining amount of tokens that the person who are authorised can transfer.

## freezeTransfer()

* Description  
  - Disallow to transfer token from an address to other address.

- Caller is only Owner.

## unfreezeTransfer()

* Description  
  - Allow to transfer token from an address to other address.

- Caller is only Owner.

## freezeUpgrade()

* Description  
  - Disallow to upgrade token to new smart contract.

- Caller is only Owner.

## unfreezeUpgrade()

* Description  
  - Allow to upgrade token to new smart contract.

- Caller is only Owner.

## changeUpgradeMaster(address \_newMaster)

* Description  
  - Change the Upgrade Master who perform to upgrade token to new smart contract.

- Caller is only Owner.

## changeUpgradeAgent(address \_newAgent)

* Description  
  - Change the new address of Upgrade Agent (new QNTU smart contract that tokens will be upgraded to it).

- Caller is only Owner.

## upgrade()

* Description  
  - Token Holder use this function for upgrade their tokens to a new smart contract.

## forceUpgrade(address[] \_holders)

* Description  
  - Upgrade Master force to upgrade token of holders to a new smart contract.

- Caller is only Upgrade Master.

## transferToContract(address \_to, uint \_value)

* Description  
  - Token Holder use this function to transfer their own tokens to specified ReceivingContract smart contract.