# ERLC IEXEC

This report contains the results of our engagements with iExec in reviewing the eRLC codebase for KYC-enabled ERC-20 tokens (ERC-677), as well as the integration these tokens with the iExec PoCo Smart Contracts.

## 1 | EXECUTIVE SUMMARY

This report contains the results of our engagements with iExec in reviewing the eRLC codebase for KYC-enabled ERC-20 tokens (ERC-677), as well as the integration these tokens with the iExec PoCo Smart Contracts.

Shayan Eskandari, Nicholas Ward and Nicholas Ward reviewed the documents over a period of 10 days (January 4th through January 8th, 2021).

## 2 | SCOPE

Our review focused on the commit hash b16266d4940f9cc695859a47c483485c48fbda66 for eRLC and the KYC additions to the PoCo delegate modules at commit hash 96a39c9d53668896321556d23351a4e79d4d46a8. The scope did not include the PoCo delegate mechanism and any other functions within the PoCo system. You can find the complete list of files within each repository's scope in the Appendix.

### 2.1 | Objectives

We identified these priorities together with the iExec group for our review:

1. Ensure that the system works in accordance with its intended functionality and without any unintended side effects.
2. Identify potential vulnerabilities in smart contract systems as described in Smart Contract Best Practices and Smart Contract Weakness Classification Registry.
3. Evaluate the design and implementation for the eRLC token contract. This includes KYC access controls, deposit and withdrawal functionality, and KYC access controls.
4. Review the addition KYC-aware transfer hooks to the PoCo system and authorization checks.

## 3 | SECURITY SPECIFICATION

This section describes the security implications of the system under scrutiny. This section is not intended to replace documentation. This section identifies security properties that have been validated by the audit team.

### 3.1 | Actors

Below are the relevant actors and their abilities:

eRLC Actors:

Admin (DEFAULT_ADMIN_ROLE):

**Manage KYC Admins**

| Start Snapshots of user balances snapshots
| Any tokens that were sent to the eRLC Contract will be claimed (amount to admin's balance).
| Unintentional deposits of the token underlying will be refunded (will be added as an admin's balance).

**KYC Admin** (KYC_ADMIN_ROLE): kyc admin, manages kyc members

| GrantKYC and revoke KYC roles to a group of addresses grantKYC, revokeKYC

**KYC Member** (KYC_MEMBER_ROLE):

| You can deposit, withdraw, receive, or transfer tokens.

**Anyone else:**

| Due to the verification of KYC in the ERC-677 _beforeTokenTransfer() callback, no account without the KYC_MEMBER_ROLE can perform any actions in the contract.
| It is a desirable property, but it is important to remember that the KYC Admin can revoke KYC at anytime, effectively freezing funds for a particular address.

### 3.2 | Security Properties

This is not an exhaustive list of security properties that were checked during this audit:

- eRLC, as explained in the Executive summary, is primarily a KYC token. Therefore, trusting admins to follow the regulations is a key feature of the system. This includes, but is not limited to, locking accounts (non-KYC members) out of the system, minting tokens and claiming extra tokens in contract.
- Only the KYC Admin has the authority to grant or revoke KYC approval
- KYC_MEMBER_ROLE is required to deposit, transfer or withdraw eRLC tokens.
- Tokens cannot be minted without depositing an equivalent amount of the underlying assets or by the Admin for the recovery of the excess balance of eRLC contracts via recover()
- Without burning an equal amount of eRLC token, no amount of the underlying asset may be withdrawn from the eRLC agreement

## 4 | RECOMMENDATION

### 4.1 | Share status codes between `ERC20KYC` and `IexecERC20CoreKYC`

Both `ERC20KYC` and the PoCo system's `IexecERC20CoreKYC` utilize the same status codes for `ERC20KYC.detectTransferRestriction`, with a status code of 0 indicating no restriction. It would be advantageous to have both the status codes defined in one library or contract to prevent any future changes from causing conflicts between them.

**code/eRLC/contracts/ERC20KYC.sol:L28-L30**

```
uint8 internal constant _RESTRICTION_OK              = uint8(0);
uint8 internal constant _RESTRICTION_MISSING_KYC_FROM = uint8(0x01);
uint8 internal constant _RESTRICTION_MISSING_KYC_TO   = uint8(0x02);
```

**code/PoCo/contracts/modules/delegates/IexecERC20CoreKYC.sol:L36-L40**

```
uint8 restrictionCode = m_baseToken.detectTransferRestriction(from, to, amount);
if (restrictionCode != uint8(0))
{
        revert(m_baseToken.messageForTransferRestriction(restrictionCode));
}
```

### 4.2 | eRLC: Include `Transfer(address,address,uint256,bytes)` event

**Acknowledge**

#### Description

The ERC-677 standard includes an event, Transfer(address,address,uint256,bytes), to be emitted from the transferAndCall() function. This event is required to prevent any deviations from the token standard. It also makes it easier for external calls to be traced back from the token contract.

We believe ERC-677 is under-specified in its current form. This suggests that a deviation from the standard might not be justified if the event has been excluded. This deviation may be made explicit in the user-facing documentation.

### 4.3 | eRLC: Require a delay period before granting KYC_ADMIN_ROLE

**Acknowledge**

**Already, the development team has plans to use a TimelockController for the KYC_DEFAULT_ADMIN in the eRLC contract. This is a satisfactory solution that also avoids unnecessary code complexity in eRLC contracts.**

Any user can be frozen by the KYC Admin at any time. This is done by revoking their `KYC_MEMBER_ROLE`. You can reduce the trust requirements of users by delaying the grant of this ability to new addresses.

Although the review does not cover the management of admin access and private keys, it is possible to add a delay to protect both the development team as well as the system in the event that a private key compromise occurs.

### Examples

Batch granting and revoking the `KYC_MEMBER_ROLE`. These functions cannot be called except by the `KYC_ADMIN_ROLE`

**code/eRLC/contracts/KYC.sol:L56-L72**

```
function grantKYC(address[] calldata accounts)
external virtual override
{
    for (uint256 i = 0; i < accounts.length; ++i)
    {
        grantRole(KYC_MEMBER_ROLE, accounts[i]);
    }
}

function revokeKYC(address[] calldata accounts)
external virtual override
{
    for (uint256 i = 0; i < accounts.length; ++i)
    {
        revokeRole(KYC_MEMBER_ROLE, accounts[i]);
    }
}
```

## APPENDIX 1 - FILES IN SCOPE

The following files were covered in this review:

eRLC Repository:

| File Name | SHA-1 hash |
|---|---|
| ERC677.sol | 9a187e76516e352fec834f2f77612b717e6d7bd1 |
| KYC.sol | e922130045a37dcd7ce791ac3566a08f6e3d1fbe |
| ERLC.sol | da987bd06bda51a2b8a2b76d6e505a9ba61c25a7 |
| ERLCTokenSwap.sol | 20ad99f44374a4e9ed43e8cd245a83201455b5f2 |
| ERC20KYC.sol | 48b443f2127724e50b6c93e67f2b5c108a113ce2 |
| Claimable.sol | cf0ab5b6f255aa38a669032f791c6ed3089ca971 |
| ERC20Softcap.sol | d81ba880a2879360356f31ac2f117c6bdca6ea42 |
| interfaces/IKYC.sol | f8b8601a1bef3f8ee15b8697284e97c93397138e |
| interfaces/IERC20KYC.sol | 9c2f0d2348444de354f5490ab47b4eec9e89904f |
| interfaces/IERC677Receiver.sol | 55b4232894ca3cabb81433bf192da40fbeed54cd |
| interfaces/IERC677.sol | 410c708e946bdaa845c8d263c2d5dbbeaf75bc86 |
| interfaces/IERC1404.sol | dad28efcd76127a3bc6a9cbcd74038e44a680ebc |

PoCo Repository:

| File Name | SHA-1 hash |
|---|---|
| modules/delegates/IexecERC20CoreKYC.sol | 6d516c02f71c1ff38970b8e246d133486a21804f |
| modules/delegates/IexecERC20DelegateKYC.sol | 7b946cecff33f5bee80715103bb9f1def3c6f5c4 |
| modules/delegates/IexecPoco1DelegateKYC.sol | fa9e7ba731f11a5a58dcdf48a9a581f529ef1b72 |
| modules/delegates/IexecEscrowTokenDelegateKYC.sol | e221d94d3b248cc36c333da7a6c8c4b8a2015d41 |
| modules/delegates/IexecPoco2KYCDelegate.sol | 904a694194f47e43c3695949c45ae19c6bb82384 |

## APPENDIX 2 DISCLOSURE

ConsenSys Dialigence ("CD") receives compensation from clients (the Clients) for performing the analysis in these reports (the Reports). Reports can be distributed via ConsenSys publications or other distributions.

Reports are not intended to endorse or indict any project or team. They also do not guarantee security for any project. This Report doesn't consider or have any bearing on the economics of token sales, tokens, or other products, services, or assets. Cryptographic tokens, which are emerging technologies, carry high technical risks and uncertainties. Any Report does not provide any representation or warranty to Third-Parties in any way. This includes regarding the bug-free nature of code, any business model or proprietors, or the legal compliance of such businesses. The Reports should not be relied upon by any third party, even if it is used to make decisions about buying or selling tokens, products, services, or assets. This Report is not intended as investment advice and should not be relied on as such. It is also not an endorsement of the project or its team. Furthermore, it does not guarantee absolute security. CD is not obligated to any Third-Party for publishing these Reports.

PURPOSE OF THE REPORTS Reports and analysis contained therein are only for Clients. They can be published with their permission. Our review will only cover Solidity code. We are limited to reviewing the Solidity codes we have identified as being included in this report. Solidity language is still under development. It may have flaws and risks. The review does NOT cover the compiler layer or any other areas that could pose security risks beyond Solidity. Cryptographic tokens, which are emerging technologies, carry high technical risk and uncertainty.

CD makes the Reports accessible to clients and other parties (i.e. "third parties") via its website. CD hopes that the public availability of these analyses will help the blockchain ecosystem to develop best practices in this rapidly changing area of innovation.

LINKS TO OTHER WEBSITES FROM THIS WEB site You can, via hypertext or other computer hyperlinks, gain access web sites owned by people other than ConsenSys. These hyperlinks are provided only for your convenience and are not intended to replace the owners of these web sites. ConsenSysys or CD are not responsible or liable for any content or operation of these Web sites. You also agree that ConsenSysys or CD will not be liable for any third-party Web site. Except as stated below, linking from this Web Site to another site does not mean or imply that ConsenSysys or CD endorses that site's content or its operator. It is up to you to decide whether or not you can use content from any other websites to which the Reports link. ConsenSys or CD will not be responsible for third-party software used on the Web Site. They also assume no responsibility and shall have no liability to any person or entity as regards the accuracy and completeness of any result generated by such software.

TIMELINESS CONTENT. The Reports are current as of the Report's date. However, they can be modified at any time. ConsenSys or CD are the only sources of information, unless otherwise indicated.