

# TRS Use-Case & Actor Definitions

Yao Sun & Ali Sharif

2017-09-06

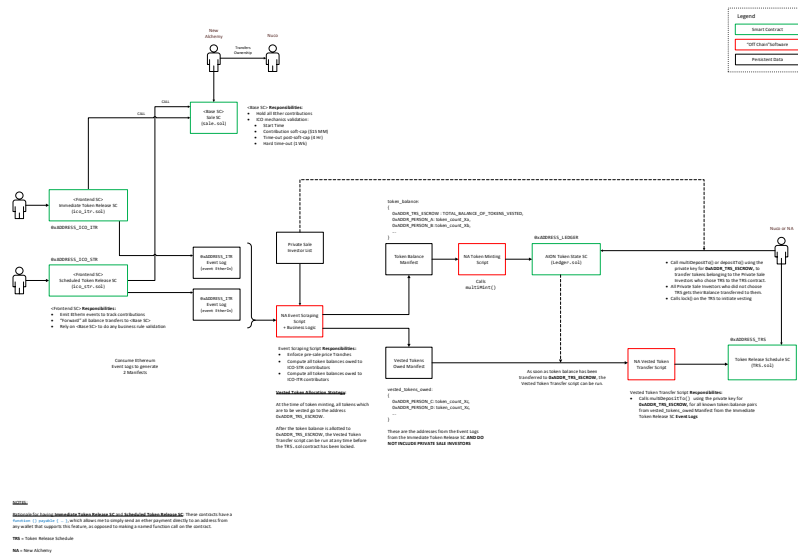


Figure 1: System Interactions Between ICO Contract Suite, TRS Contract, NA Software and Ledger.sol

The following are use case definitions initially concluded by the team with respect to the TRS contract. Proceeding, we will be designing test cases around the following actors. These are derived from a series of meeting with Matt Spoke, New Alchemy team and Nuco Engineering team, in which the following system architecture defined: Figure 1.

## **Actors**

### **Private and Pre-Sale Liquid Token Holders**

They are address owners who have been allocated a balance in the ERC20 Token contract and are looking to get into the TRS before `lock()` is called

They require:

- Ability to contribute to the TRS contract

### **Private Pre-Sale Contributors**

These are people who privately sent contributions (in any form) to Aion Foundation to be included in the time-tranched pre-sale. They have the option at the time of the commitment to choose between TRS or Liquid and they are minted tokens accordingly.

They require:

- Same ability to interact with TRS as a TRS contributor

### **Private Sale Contributors**

These are people who contributed privately (in any form) to Aion Foundation in exchange for Tokens at a fixed price. At token minting, the Foundation holds onto their token for a period of 1 week, at which time, they will inform the foundation if they want liquid allocation or TRS allocation, at which point the Foundation will appropriately distribute the Tokens to the source addresses or TRS contract

They require:

- Same ability to interact with TRS as a TRS contributor

### **Regular Pre-Sale Contributors**

These are the people who contributed to the pre-sale through the STR (scheduled token release contract). At the time of minting, they are immediately allocated tokens in the TRS.

They require:

- Same ability to interact with TRS as a TRS contributor

## TRS Contributors

These are parties with token deposits in the TRS contract, they will be interested in reviewing that their stored amounts are correct, and that the payout is correct. They need to be able to:

- Query the TRS contract to figure out current period
- Query the TRS account on the contract to see [total tokens owned by the contract after locking]
- Query the TRS account on the contract to see [total face value deposited; sum of balanceOf]
- Query the TRS account on the contract to see [number of withdrawals they've done]
- Assert that there are no discrepancies between the TRS contract & its owned token value
- Query their balance once tokens have been deposited to TRS (this may be a multi-block process)
- Withdraw their own tokens!!!

## Foundation / NA

- lock() contract
- start() token release schedule
- transfer ownership