

Verification Report

Elrond Standard Digital Tokens

September 14th, 2022



Prepared for Elrond Network by





Summary

This report presents our findings in regards to the ESDT tokens implementation in the Elrond Network's protocol. The objective was to review business logic and implementation in order to identify any issues that could cause unexpected behavior or be exploited.

Management of custom tokens on the Elrond blockchain is ensured by ESDTs - Elrond Standard Digital Tokens. Implementation of the ESDT mechanism at the protocol level removes the need for additional smart contracts or smart contract execution through the virtual machine.

Scope

The scope of this review is limited to the management and transfer of fungible ESDT tokens on the Elrond blockchain. The review has been conducted on the following repositories.

- elrond-vm-common - [64a0dfe69a18e0fdc3d69f41ddfa6ff40155fe5e](https://github.com/ElrondNetwork/elrond-vm-common/commit/64a0dfe69a18e0fdc3d69f41ddfa6ff40155fe5e)
- elrond-go - [8e402fa6d7e91e779980122d3798b2bf50892945](https://github.com/ElrondNetwork/elrond-go/commit/8e402fa6d7e91e779980122d3798b2bf50892945)
- wasm-vm - [1fcbc625c0eaae835e36e5ead9e57634cb18079d](https://github.com/ElrondNetwork/wasm-vm/commit/1fcbc625c0eaae835e36e5ead9e57634cb18079d)

The following artifacts have been reviewed.

- [esdt.go](#) - which handles token issuance by calls to the metachain
- [process.go](#) - the main file where execution of builtin functions is handled
- [metaProcess.go](#) - handling of transactions related to the metachain
- [shardProcess.go](#) - processing of shard transactions
- Functions managing ESDT transfers initiated by smart contracts
 - [TransferESDTNFTExecuteWithTypedArgs](#)
 - [TransferESDT](#)
 - [ExecuteESDTTransfer](#)
- ESDT builtin functions
 - [esdtDataStorage.go](#)
 - [esdtFreezeWipe.go](#)
 - [esdtGlobalSettings.go](#)
 - [esdtLocalBurn.go](#)
 - [esdtLocalMint.go](#)
 - [esdtMetaData.go](#)
 - [esdtRoles.go](#)
 - [esdtTransfer.go](#)



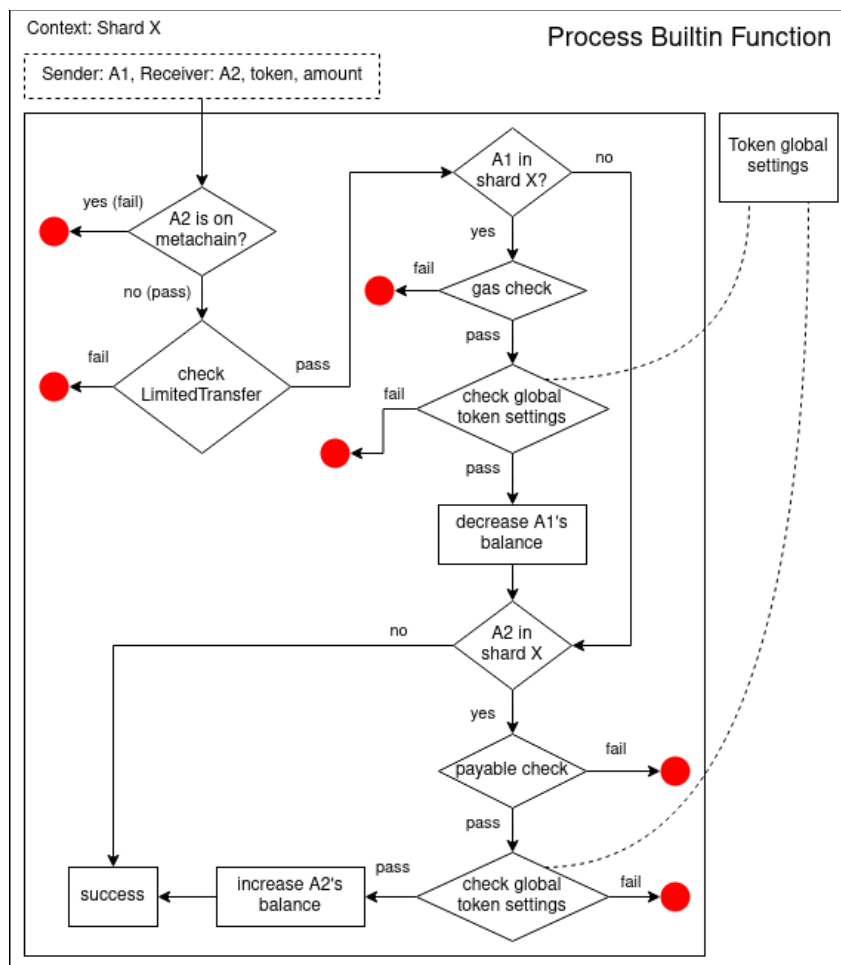
Disclaimer

This report does not constitute legal or investment advice. You understand and agree that this report relates to new and emerging technologies and that there are significant risks inherent in using such technologies that cannot be completely protected against. While this report has been prepared based on data and information that has been provided or is otherwise publicly available, there are likely additional unknown risks which otherwise exist. This report is also not comprehensive in scope, excluding a number of components critical to the correct operation of this system. This report is for informational purposes only and is provided on an "as-is" basis and you acknowledge and agree that you are making use of this report and the information contained herein at your own risk. The preparers of this report make no representations or warranties of any kind, either express or implied, regarding the information in or the use of this report and shall not be liable to you or any third parties for any acts or omissions undertaken by you or any third parties based on the information contained herein.

Protocol Description


ESDT Transfers

ProcessESDT common subflow



The graph above describes the core workflow of the ESDT transfer. Since Elrond is a multi-shard blockchain, the sender(receiver)'s balance is updated only when the sender(receiver) is in the current shard where the execution takes place. In each shard, there is a special storage which stores each token's global settings including:

- LimitedTransfer flag. Token owners can use this functionality to restrict token transfers. The "check LimitedTransfer" in the above graph first checks if the limitedTransfer flag is set to True. If the flag is true, it requires either the sender



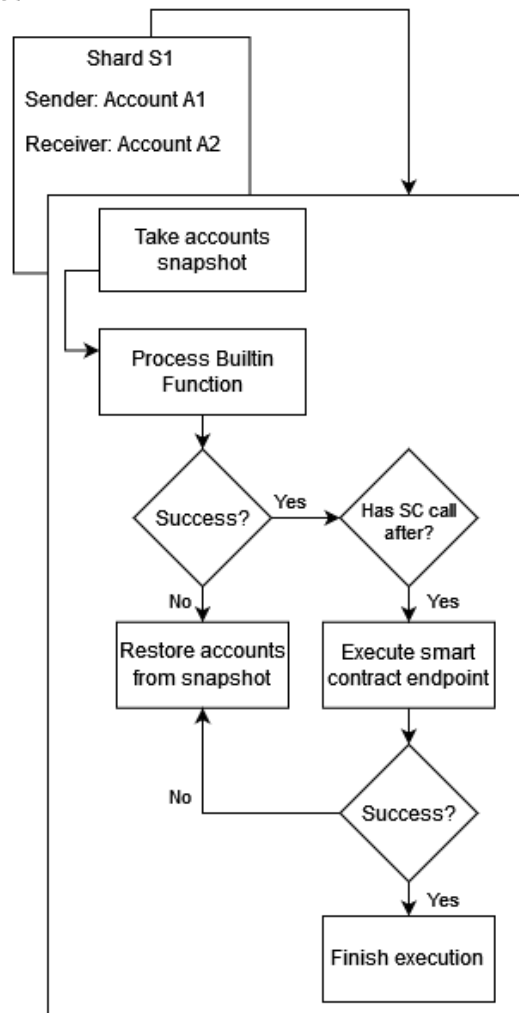
or the receiver to be whitelisted. The common usage is to restrict the addresses that locked tokens can be sent to.

- Pause flag. When the flag is set to true, all transactions of the token are paused.
- The accounts that are frozen for the token. The common usage is to blacklist hacker accounts.

The global settings will be checked when updating the balances of the sender and the receiver. If the receiver is a smart contract and the transaction is a simple transfer (no smart contract invocation), then the protocol will check if the smart contract is payable. If the transaction has a smart contract invocation, the payable check is done at the contract code level (not at the protocol level). The workflow will return “fail” if any of the checks fails (red dots in the graph) and return “success” if all the checks pass. Note that in the case of the same shard transfer, the context is where the sender and the receiver live. In the case of cross shard transfer, the context is the shard where the sender or the receiver lives.

Same shard Transfer and Execute

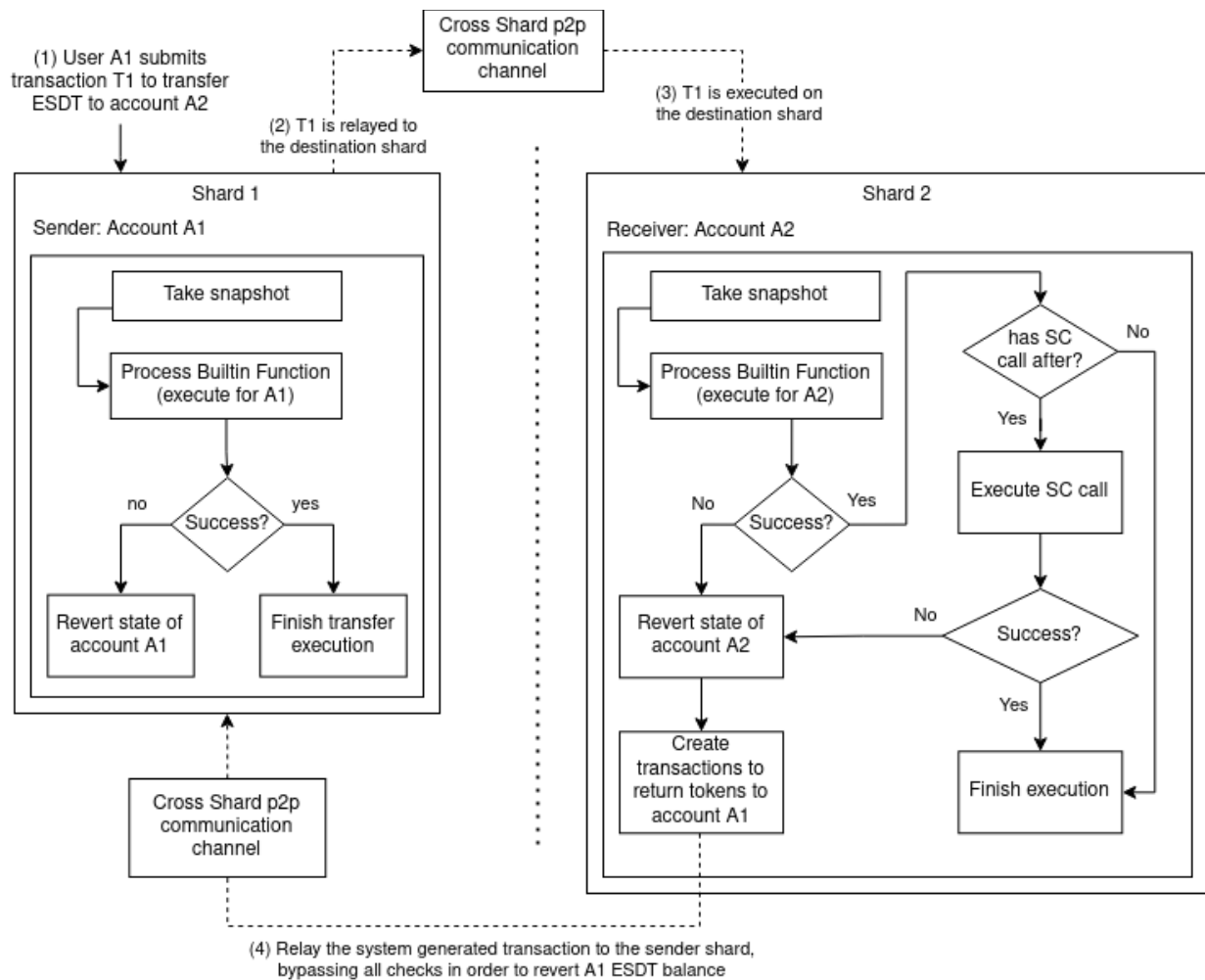
(1) Account A1 submits transaction T1 to transfer ESDT to account A2



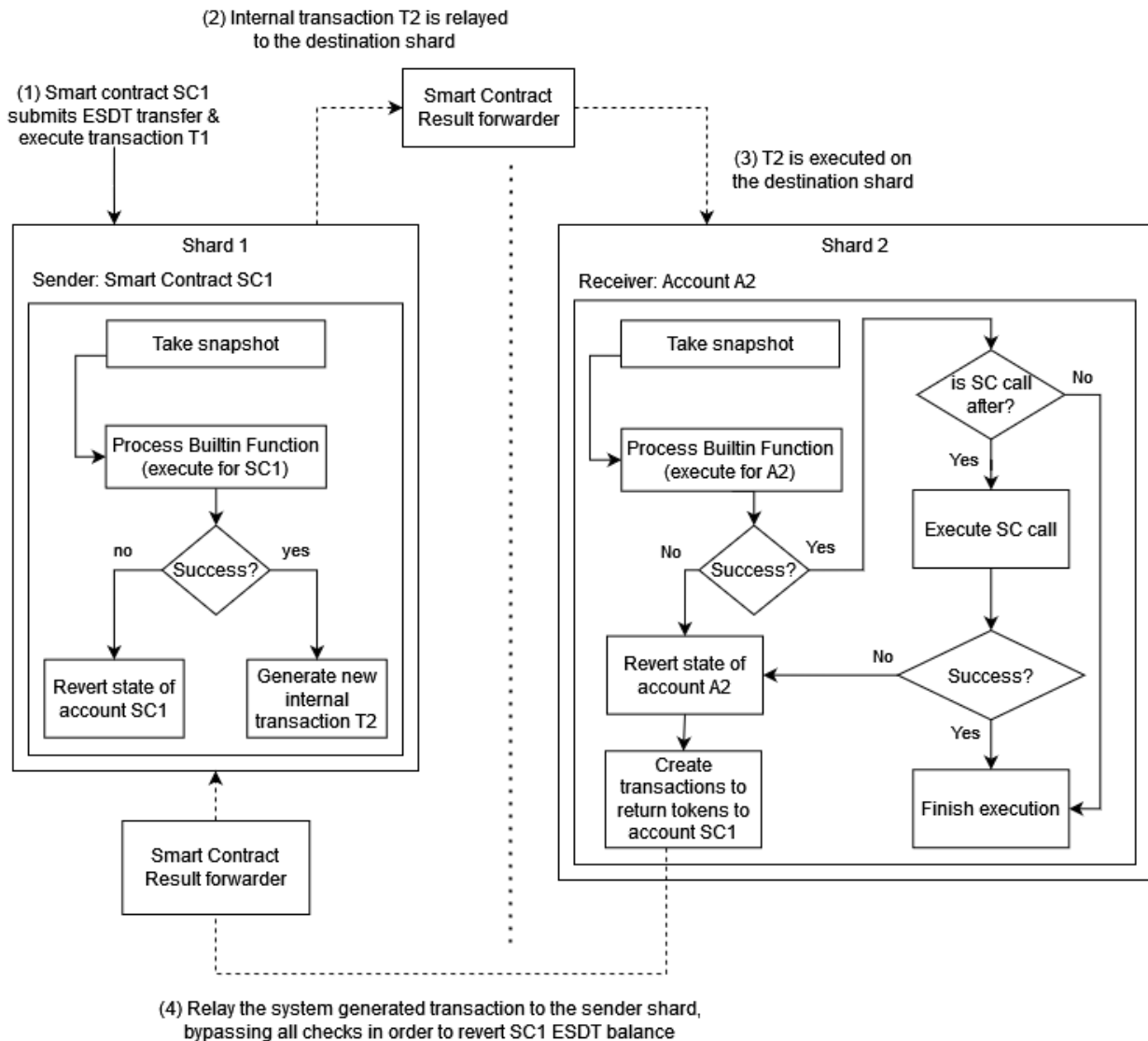
When the sender and the receiver are in the same shard, their balances are updated atomically. The protocol takes a snapshot before the transfer and in case of error, the account state is reverted to the snapshot.

Cross shard Transfer and Execute

- sender is an EOA account



- sender is a smart contract



When the sender and the receiver are in different shards, their balances are updated in two stages. In the first stage, the sender's balance is subtracted. If the execution on the sender succeeds, in the second stage, a transaction will be sent to the receiver to add the amount to the receiver's balance and, if needed, to execute the transaction's SC call. If the execution on the sender fails, the sender's state will get reverted and no transaction will be sent to the receiver. If the execution on the receiver fails, the receiver's state will get reverted and the receiver will send the same amount of token back to the sender.

Assumptions

- During the code review, we assume that no transactions will be altered or missed during the execution
- An account can only exist in one shard.

Important properties

- If an ESDT transfer succeeds eventually, the sender's and the receiver's balances are correctly updated. No token is lost during the transfer.
- If an ESDT transfer fails eventually, the sender's and the receiver's balances are updated to the values they would have if the transfer never happened.
- All account balances are greater or equal to 0.

ESDT builtin functions

In addition to the basic ESDT transfer functionality, we also checked the following ESDT builtin functions.

- Issue: The builtin function creates a new token on the metachain.
- Pause/Unpause(token T): Only the owner of token T can send the pause/unpause transaction to the system smart contract on the metachain and the smart contract will send a transaction to update the local pause flag of token T in each shard.
- Freeze/Unfreeze(account X, token T): Only the owner of token T can send the freeze/unfreeze transaction to the system smart contract on the metachain and the smart contract will send a transaction to update the local frozen accounts of token T in account X's shard.
- Wipe(account X, token T): Only the owner of token T can wipe out all data related to the T tokens held by a frozen account X. The wipe transaction is sent to the system smart contract on the metachain and the smart contract will send a transaction to wipe out the data. After wiping out the data, account X becomes unfrozen.
- LocalMint: Only an account that has ESDTRoleLocalMint role can mint tokens to itself by sending an ESDTLocalMint transaction to itself.
- LocalBurn: Only an account that has ESDTRoleLocalBurn role can burn its own tokens by sending an ESDTLocalBurn transaction to itself.



Important properties

- For an EOA account, no other account can decrease its token balance. For contracts, the balance can be decreased by other accounts only indirectly, i.e. by calling contract endpoints which take an explicit action to decrease it. The only exception is that if the token is Wipeable and Freezable, the token manager can wipe out all the tokens held by a frozen account.