SecurityToken(ERC1400) Incarnation

Newly Introduced functions

1. `isIssuable()` - This function is used to know whether the token is mintable or not If a token returns FALSE for `isIssuable()` then it MUST never allow additional tokens to be issued.
2. `canTransfer()` - This function will used to understand the cause of failure by getting the byte value. Which will be the ESC that follows the EIP 1066. ESC can be mapped with a reason string to understand the failure cause, table of Ethereum status code will always reside off-chain.
   Return values -
   a. `bool` It signifies whether the transaction will be executed or not.
   b. `byte` Ethereum status code (ESC)
   c. `bytes32` Application specific reason code.
3. Similarly `canTransferFrom()` will be used.
4. `setDocument()` - Used to attach a new document to the contract, or update the URI or hash of an existing attached document (Only called by the owner).
5. `removeDocument()` - Used to remove an existing document from the contract by giving the name of the document.
6. `isControllable()` - In order to provide transparency over whether `controllerTransfer()` / `controllerRedeem()` are useable or not `isControllable()` function will be used.
7. `getDocument()` - Used to return the details of a document with a known name (`bytes32`).
8. `getAllDocuments()` - Used to retrieve a full list of documents attached to the smart contract.
9. `balanceOfByPartition()` - It is used to retrieve the balance of token holder according to the given partition. Initially we only introduced two partitions `LOCKED` & `UNLOCKED`. It will work by looping every TMs to get the balance by using `getTokensByPartition()` which is present in the `ITransferManager`.

Function name changes

1. `freezeMinting()` → `freezeIssuance()` .

2. `mintWithData(address _investor, uint256 _value, bytes memory _data)` → `issue(address _tokenHolder, uint256 _value, bytes memory _data)`. issue function works similar as the earlier one but it doesn't return a boolean anymore (No return statement).
3. `mintMulti()` → `issueMulti()`.
4. `burnWithData()` → `redeem()`.
5. `burnFromWithData()` → `redeemFrom()`.
6. `forceTransfer()` → `controllerTransfer()` where parameter `_log` is renamed with the `_operatorData` parameter.
7. `forceBurn()` → `controllerRedeem()` where parameter `_log` is renamed with the `_operatorData` parameter.

Removed functions

1. `mint()` It is removed and now everywhere `issue()` function will be used.
2. public version of `verifyTransfer()` is removed. canTransfer() & canTransferFrom() will be used instead.

Changed Events

1. `Minted(address indexed _to, uint256 _value)` → `Issued(address indexed _operator, address indexed _to, uint256 _value, bytes _data)`.
2. `Burnt(address indexed _from, uint256 _value)` → `Redeemed(address indexed _operator, address indexed _from, uint256 _value, bytes _data)`.
3. `ForceTransfer` → `ControllerTransfer`
4. `ForceBurn` → `ControllerRedemption`

Note - add `STGetter` contract address in the securityToken constructor as delegate. This change is not related to ERC's while it is in the favour of contract size issue.