**SMT Smart Contract**

SMT is an Ethereum token. It complies with and extends EIP-20 – the only token standard in Ethereum and a widely used token API. SMT Smart Contract guarantees:

**1. Transparency**

**1.1. Balance.** The information on the number of tokens held by any user is public.

**1.2. Transfers.** All information on transfers is public and can be traced back in time.

**2. Ownership**

**2.1. Scope.** Only Ethereum users and contracts can be token holders.

**2.2. Uniqueness.** Each token belongs to one user-owner. There are no shared tokens.

**2.3. Right to transfer.** A token can be transferred to another user only by the direct command of its owner or by the command of the receiver directly authorized by the owner. No token transfer may be initiated by another user.

**3. Token Supply**

**3.1. Multiple issuance.** Tokens can be issued multiple times by the contract owner.

**3.2. Supply.** The token supply can increase or decrease since the time of deployment.

**3.3. Destruction.** Every user can destroy (burn) some or all of his tokens, which are then deducted from the total supply.

**4. Contract Management**

**4.1 Replacement.** The contract owner can relinquish the ownership in favor of any other Ethereum user or contract.

**4.2 Blockade.** The contract owner can stop or resume token transfers between token holders  at any time.

**5. Miscellaneous**

**5.1 Recovery.** Any call to the contract which results in an error does not change the users' tokens or Ether balance, except for the gas spent on the transaction.

**5.2 Safe Approval.** The token contract supports two approve() functions: a 2-parameter approve() (the EIP-20 standard) and a 3-parameter approve(), which guarantees that spender gets new allowance only if current allowance equals presumed allowance. It is recommended to use the 3-parameter approve for all approve() calls taking the presumed allowance as the second input.

**Uncertainty provisions**

The SMT Smart Contract does *not* guarantee the following:

**6.1 User validity.** Details:an Ethereum address with positive token balancemay not correspond to any actual user or a private key, as it can be a result of a mistake. Tokens transferred to such users will likely be lost.

**6.2 Zero Ether balance.** Details: the contract prohibits most, but not all means by which Ether could be sent to it by users who are not contract owners.

**6.3 Complete registry.** Details: The contract does not provide a list of all token holders. However, it is guaranteed that every token holder is either the contract creator or a token recipient in the Transfer() event. The list of token holders can be obtained by checking all these events or by exploring the contract storage using blockchain explorers.