STBT - Short-term Treasury Bill Token

March 18th, 2023

Abstract

STBT is a token designed for accredited investors seeking T-bill yields with their stablecoin holdings. It offers exposure to US T-bill yields while maintaining the stability of stablecoin investments.

Each STBT token is pegged to 1 USD and backed by US Treasury securities with maturities within 6 months and reverse repurchase agreements. STBT enables diversification of stablecoin portfolios with exposure to US Treasury Bills and introduces risk-free interest into the DeFi ecosystem.

1. Introduction

The cryptocurrency market and other major tradable markets are undergoing significant changes due to a macro shift. This shift involves the move from easy money to higher interest rates across most central banks, resulting in pressure on asset prices.

The US Federal Reserve aims to combat inflation by raising interest rates, fulfilling a mandate of ensuring full employment and price stability. By doing so, the Fed intends to slow down investment and spending by offering an attractive yield for holding cash rather than investing or spending it. The interest rate paid by the Fed on deposits is considered the risk-free rate. As investors anticipate further rate hikes, they have responded by selling equities, cryptocurrencies, and other risk assets and depositing their cash into treasury-backed savings products such as money market funds, repurchase agreements, or US Treasury securities like T-bills.

On the crypto side, the market has responded differently. The "risk-free" rate on USD stablecoins has decreased in response to the Fed's actions. Additionally, to earn that yield, significant counterparty or protocol risk must be taken. Many anticipate that the Fed will maintain a high interest rate environment for an extended period. Market participants who prefer to hold stablecoins rather than fiat require the ability to earn the risk-free rate while navigating this challenging market.

We introduce STBT, the Short-term Treasury Bill token, a token backed by the underlying assets of reverse repurchase agreements collateralized by U.S. Treasury securities and U.S. Treasury securities maturing within six (6) months(overnight loan fully backed by T-bill treasuries).

STBT is developed according to the ERC 1400 standard and is equipped with the latest security features, as certified by BlockSec and Zellic.io. It's created exclusively for accredited investors looking to earn T-bill yields and is transferable solely to account-holders that have been pre-approved via a KYC/AML white-listing mechanism.

Investors can either mint and redeem STBT through Matrixdock, or trade STBT directly via STBT/3CRV pool on Curve Finance.

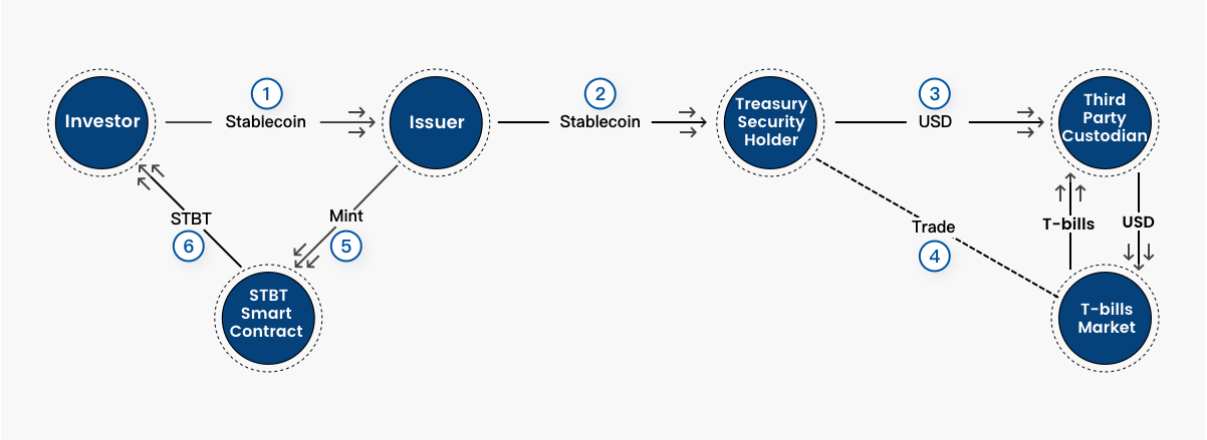
2. How it works

2.1 Issuance (Mint)

USDC/USDT/DAI can be deposited through APP and STBT tokens will be minted once the underlying T-bill subscription and/or reverse repo is confirmed, via interaction with the STBT smart contract.

Alternatively, investors can request STBT minting by transferring USDC/USDT/DAI to the STBT Issuer's dedicated address, upon execution of the over-the-counter agreement. The STBT minting takes up to 3 New York Banking Days given the operational process.

STBT Mint Flow Chart:



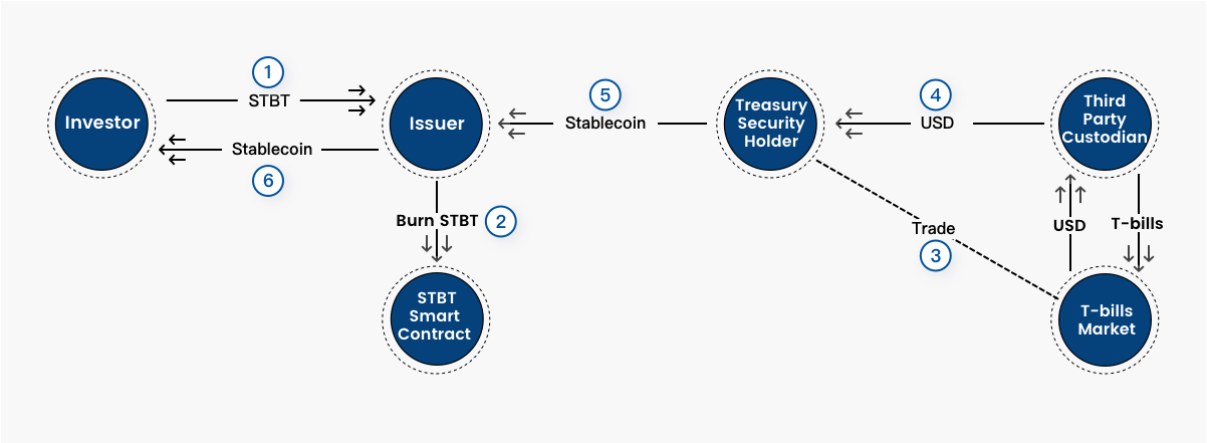
2.2 Redemption (Burn)

STBT redemption can be requested through APP or by transferring STBT to the Issuer's dedicated address. The timeline for redemption is T+4 (New York Banking Day only).

In the event that the redemption results in a T-bill disposal prior to its maturity, the Execution Price is calculated as the T-bill settlement price divided by the fair market value (“FMV”) on the preceding day. The T-bill settlement price can be lower than the fair market price for such T-bill on the preceding day, causing the STBT/USD ratio (the "Execution Price") to be less than 1.

The final settlement amount is then calculated as the STBT redemption amount * Execution Price * (1 - 0.1% redemption fee).

STBT Burn Flow Chart



2.3 Interest Rebase

For all STBT holders, the interest rebase mechanism will be as following:

The total STBT supply is equivalent to the Net Asset Value (NAV) of the underlying asset portfolio, which equates to the sum of US T-bills, reverse repurchase agreements (overnight loans fully-backed by T-bills) and cash held at any point in time. The expiry date for each T-bill corresponds to the day that its par value will be realized. To calculate the STBT rebased interest, reference is made to Bloomberg's 3pm NY time closing price (using the "BGN" source on Bloomberg Terminal's "HP" function) (the “Fair Market Value” or “FMV”) and compare it to the previous day's FMV.

- Daily Interest Distributed = NAV of the current day - NAV of the last rebase day - Expenses*

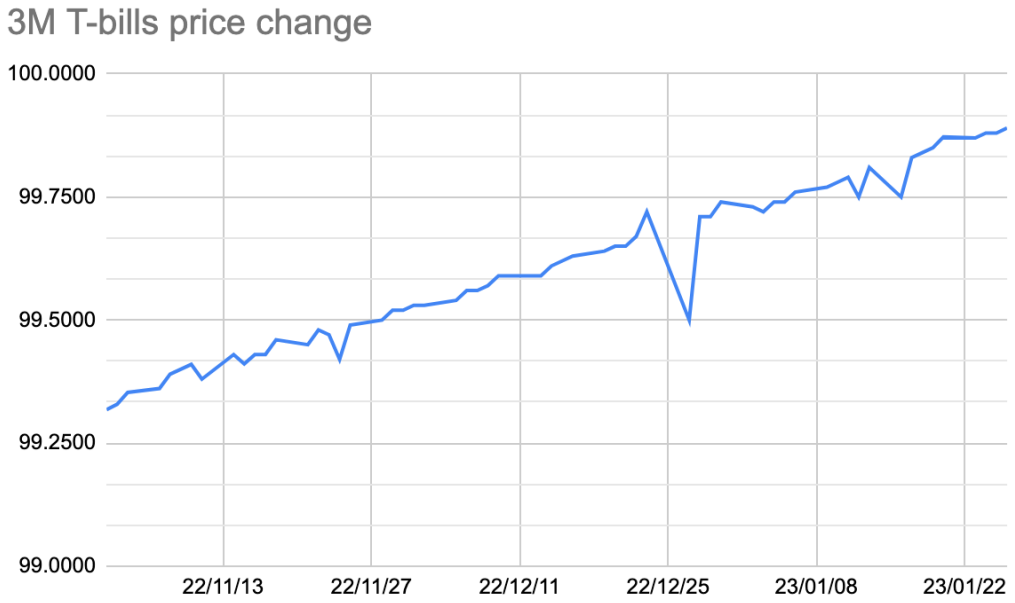
- APR on the Rebase Day = Daily Interest Distributed / STBT Market Cap / Number of Interest Accrual Day * 365 . Matrixdock Website will update the 7 day-average APR for reference.

- If there is a new T-bills transaction, the intraday price movement(the price movement between marked price and settled price) will not be included into interest rebasing on the transaction settlement day as the corresponding STBT were not minted yet. The intraday price movement will be rolled over and distributed together when we rebase the interest on the next day after the settlement day.

**Expenses including T-bill Custodian Fee, Reverse Repo Brokerage Fee and Matrixdock Service Fee (0.1% APY). The all-inclusive expenses are estimated to be approximately 0.3% APY, subject to periodical adjustment.*

Due to the volatility of the T-bill market, it is possible that in rare cases the FMV on a certain day traded lower than the FMV of the preceding day, resulting in a negative interest temporarily. STBT Issuer refrains from rebasing negative interest to STBT holdings due to restrictions within DeFi protocols and the potential risks associated (most DeFi protocols don't accept negative interest rebase token). As such, when such a rare occasion of negative interest day occurs, no rebase will be made on that day itself until (positive) interest will be rebased when the FMV on a following day becomes higher than the FMV on the previous interest rebase day.

Here's a chart of price changes for a January 31, 2023 T-bill:



2.4 Secondary Market Liquidity of STBT

The STBT/3CRV pool has been established with the aim of providing secondary market liquidity and increasing returns for STBT holders. The 3CRV pool comprises stablecoins, specifically USDC, USDT, and DAI. Verified accredited investors are given two methods of participating in the STBT/3CRV pool, namely, token swapping and liquidity provisioning.

Token swapping enables verified accredited investors to exchange STBT for stablecoins and vice versa. This process can be executed through a whitelisted address only. The swap rate is not dependent on the STBT Issuance or Redemption mechanism, and the Curve pool real-time rates apply. This feature facilitates a 24/7 no waiting time swap experience.

Through liquidity provisioning, verified accredited investors can provide a single or multiple stablecoins to earn rewards from the pool, which consist of STBT native yields and transaction fees.

3. STBT Smart Contract

3.1 Key Functions

STBT balances are dynamic and represent the holder's share in the total value of underlying assets controlled by the protocol, the concept of ‘user share’ is introduced into the contract, so the contract also stores the sum of all shares to calculate each account's token balance which equals to: **sharesOf(user) * totalPooledSTBT / totalShares**

3.1.1 Token Issurance

```javascript
function mintShares(address _to, uint256 _userDepositAmount) external {
uint256 shareToMint = _userDepositAmount * totalShares / totalPooledUSD;
totalShares += shareToMint;
totalPooledSTBT += userDepositAmount;
}
```

3.1.2 Distribute Interests

```javascript
function distributeInterests(uint256 _distributedInterestSTBT) external {
totalPooledSTBT += _distributedInterestSTBT;

}
```

3.1.3 Token Redemption

```javascript
function burnShares(address _account, uint256
_userWithdrawalAmount) external {
uint256 shareToBurn = _userWithdrawalAmount * totalShare
/ totalPooledUSD;
totoalPooledSTBT -= userWithdrawalAmount
}
```

3.2 ERC1400 Security Token Standards And Authority Management

ERC1400 standard is split into several modular sub-standards :

- ERC1410 which defines partially fungible tokens where balances of tokens can have an associated metadata
- ERC1594 which defines transfer restrictions and core security token functionality
- ERC1643 which defines document management functionality.
- ERC1644 which defines controller operation functionality

In particular, the STBT token contract implements the ERC1594 Transfer Restrictions specs, and the ERC1644 Controller Operations specs

3.2.1 ERC1594 Restricted Transfer

```javascript
interface IERC1594 {
  // // Transfer Validity
```

```
  function canTransfer(address _to, uint256 _value, bytes calldata _data)
external view returns (bool, byte, bytes32);

  function canTransferFrom(address _from, address _to, uint256 _value, bytes
calldata _data) external view returns (bool, byte, bytes32);
}
```

STBT utilizes the whitelist mechanism to determine whether transmission and reception of addresses are authorized and to specify the time-lock parameter. The data structure is presented below:

```
JavaScript
mapping (address => Permission) public permissions; // Address-specific
transfer permissions

struct Permission {
  bool sendAllowed; // default: true
  bool receiveAllowed;

  // Address holder's KYC will be validated till this time, after that the holder
needs to re-KYC.
  bool expiryTime; // default:0 validated forever
}
```

3.2.2 ERC1644 Controller Operations

```
JavaScript
interface IERC1644 {
  // Controller Operation
  function controllerTransfer(address _from, address _to, uint256 _value, bytes
calldata _data, bytes calldata _operatorData) external;

  function controllerRedeem(address _tokenHolder, uint256 _value, bytes
calldata _data, bytes calldata _operatorData) external;
  function isControllable() external view returns (bool);
```

}

STBT retains the ability to perform forced transfers between addresses to reverse fraudulent transactions or the loss of a private key.

3.2.3 Authority Management

The permission framework of the STBT contract is broadly classified into three distinct roles:

- **ControllerRole**, which defines the role of the contract controller and have the authority to perform mandatory transfer and redemption defined by ERC1644
- **IssuerRole**, which defines the role of the token issuer, who has the authority to perform normal token issuance and redemption
- **ModeratorRole**, which defines the role of the contract moderator, who has the authority to perform the configuration of permission of an address defined by ERC1594

All three roles mentioned above are denoted as Administrators. The initial proposal is to designate a single external control smart contract with a timelock function as the controller contract for all three roles. All STBT administrator activities are carried out via this controller contract, with each operation subject to two stages of proposing and executing, and the execution must conform to the delay specified by the timelock.

For the implementation of controller, we have the following assumption:

- Remove the `minDelay` design to prevent discrepancies in timelock duration for different operations. If `minDelay` is set to 60s, the security of the `mint` operation cannot be ensured, as it requires a 4-hour delay, whereas the `burn` operation can be delayed for only 60s.
- Avoid passing in delay parameters from the caller to prevent security issues arising from a compromised business system.
- Add custom logic in the controller smart contract implementation to set different `delay` values for various operations (parsed according to the first four bytes of calldata).
- To modify `delay` values in the future, redeploy a new controller smart contract and set the admin role of STBT to the new controller via the old controller.

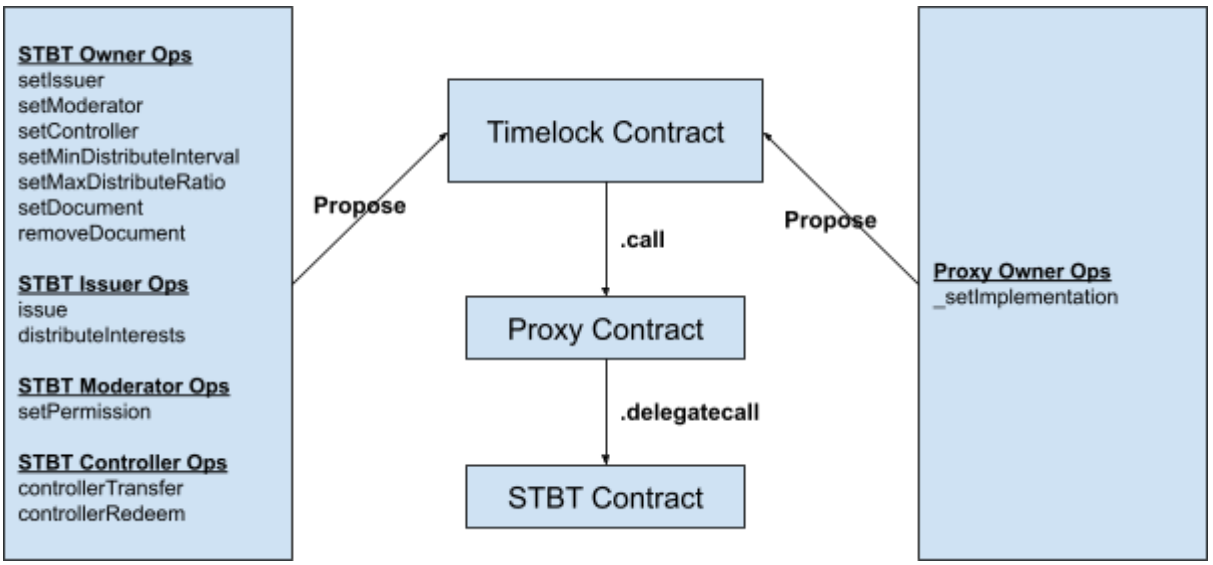
3.3 Overall Structure

- The timelock controller contract is the entry contract calling into the STBT contract, and the roles involved in the contract :
    - Owner: contract itself, to realize self-management
    - Proposer/Executor/Canceller: They are the most important EOA addresses, which control any calls involving the admin role in the Proxy and STBT contracts. These addresses are specified when the contracts are deployed.
- The proxy contract is the proxy of the STBT contract and is responsible for the upgrade of STBT
    - Set the owner of the proxy contract as a timelock contract when deploying, and the subsequent upgrade of the STBT contract is controlled by the timelock contract.
- STBT contract
    - The owner is the timelock controller contract.
    - Controller/issuer/moderator are all set as the timelock controller after STBT deployed.

3.3.1 Security

By the above settings, the timelock contract becomes the only admin role, and the security of the STBT contract is normalized to the security of each role of proposer/executor/canceller of the timelock controller.

Proposer/executor/canceller are three different EOA addresses, among which proposer and executor are Cactus custody addresses. Any admin operation of the contract [see the figure below] needs to be initiated by the proposer through the timelock contract, the executor will call and execute the operation after timelock expired, both proposer and executor should sign the operation, and both are indispensable. Canceler is one (or two) self-hosted hardware wallets, which can call the timelock controller to cancel operation that has been scheduled during the timelock period. Currently, in order to balance safety and operational efficiency, we set the timelock period of Mint, Burn, and Rebase to 4 hours.



3.3.2 Redemption

There are three methods in the contract to complete burning tokens, namely: `redeem`, `redeemFrom`, `controllerRedeem`, and only admins can operate these three methods.

- **RedeemFrom**
  To do the redemption, the user first needs to approve the corresponding amount of tokens to the admin of the contract, and then the admin initiates the `redeemFrom` method call to the smart contract. If the user address belongs to the Matrixdock account, the system will automatically help users to call the `approve` method. Otherwise,the user needs to call the `approve` method on his own initiative to authorize a certain amount of tokens to the admin of smart contract which is the timelock controller.

- **Redeem**
  Admin of the contract initiates this method call to the smart contract to redeem tokens belonging to the admin.

- **controllerRedeem**

- User tokens can be burned without the user's permission, which conflicts with the decentralized nature of the blockchain
- This method is part of the ERC1400 specification, and the rationale when discussing this EIP specification is explained as : *A token representing ownership in a security may require authorized operators to have additional controls over the tokens.In some jurisdictions the issuer (or an entity delegated to by the issuer) may*

need to retain the ability to force transfer tokens. This could be to address a legal dispute or court order, or to remedy an investor losing access to their private keys.

- Matrixdock will strictly manage this permission internally, and will find several reputable partner institutions to manage the private keys of this function with Matrixdock in the future to further reduce this risk.

3.4 Smart contract upgrade

3.4.1 Upgrade of STBT

This pertains primarily to the scenario where the STBT contract logic requires modification.

The specific actions include redeploying the STBT smart contract and invoking the `resetImplementation` method of the proxy contract through the timelock controller to specify the new implementation of the STBT contract.

3.4.2 Upgrade of timelock controller

This primarily addresses the scenario where the length of the timelock duration requires an update.

The specific actions include redeploying the timelock controller smart contract and updating the proxy contract owner and STBT controller/issuer/moderator to the new timelock by invoking the "old timelock contract".

This will result in subsequent STBT contract calls being directed to the new timelock controller. It should be noted that all existing operations of the "old timelock contract" must have already been executed.

3.4.3 Redeployment of all contracts

This pertains primarily to severe attacks of unknown origin.

The process involves taking a snapshot of the contract state at a designated block height and reconstructing the user's balance based on the post-deployment events such as issue, redeem, transfer, etc., that occurred until that block. Tokens can be reissued by invoking the issue method of the newly deployed contract.

Conclusion

STBT is a blockchain-based store of value that is fully backed by short-dated U.S. treasuries and reverse repo secured by U.S treasury securities, ensuring that each token is redeemable 1:1 for U.S. dollars (minus T-bill sale slippage, if any). On-chain statements are available for STBT holders to verify that the amount of underlying assets is at least the amount of STBT in circulation each New York Banking day. Matrixdock holds these assets in the custody and management of credible financial institutions.