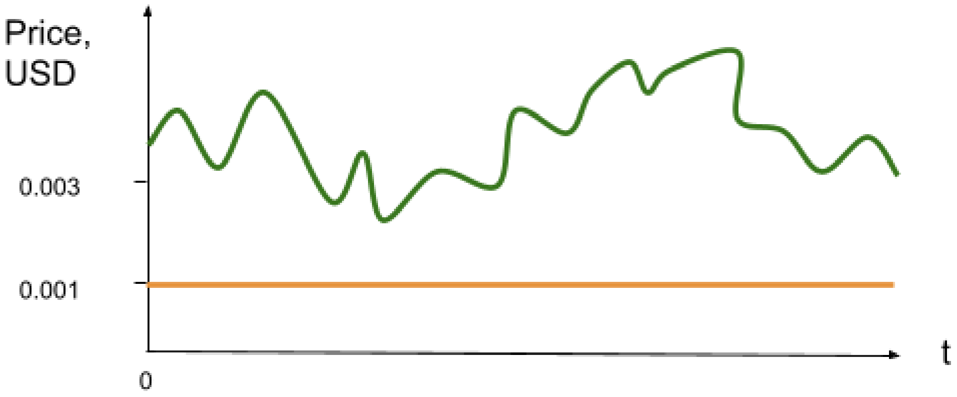
ReservedUSD

Introduction

ReservedUSD is a protocol for the decentralized issuance of the USD token on the Tera blockchain platform. The exchange rate is pegged to the real currency USD, and its stability is ensured by the reserve fund of the collateral asset. Having a reliable stablecoin protocol is very important for the development of the blockchain ecosystem. It is useful for traders and for other smart contracts that require the Oracle of the TERA rate.

Usually, the USD / TERA exchange chart looks like this:



On it, you can highlight the border below which the exchange rate has never fallen - this is the orange line with the rate of 0.001. This allows you to use the Tera as a collateral asset.

Basic properties:

1. To motivate the creation of stablecoins, we will issue a USDAO liquidity token, which will serve to profit from commissions and manage the parameters of the smart contract. Its name is a combination of two words USD + DAO.
2. For convenient use of the stablecoin, the smart contract will have a built-in exchange for the Tera/USD pair. The commission for the exchange operations will be sent to the owners of the USDAO.
3. DAO government. Voting will be possible for:
    a. Course backup Tera relative to USD (the Tera can change in foreign markets, for example, if the real exchange rate Panthers will significantly grow will be not profitable to create a new stablein at a low rate. Therefore, the reserve rate should follow the real rate, but always remain in a safe zone in case of a sharp fall, for example, have a safety margin from a fall of up to 3 times.
    b. The amount of the fee for USD/TERA exchange operations

Mechanics of operation

Pools

Pools are created (where tokens are stored on smart contract accounts):

1. TERA Collateral Asset Reservation Pool (*TeraReservePool*)
2. USD Exchange Pool (*USDSwapPool*)
3. TERA Exchange Pool (*TeraSwapPool*)

The calculation formula is a snapshot of courses

1. The reserve rate sets the number of USD tokens that can be issued per Tera:

   KReserve=*USDAllAmount* / *TeraReservePool*

2. The exchange rate sets the number of USD tokens that can be exchanged for one Tera:

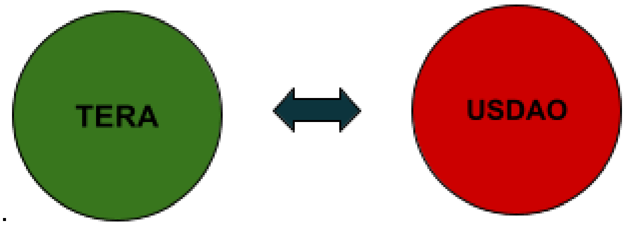
   KSwap=*USDSwapPool* / *TeraSwapPool*

3. The liquidity token rate sets the number of USDAO liquidity tokens that can be obtained in one Tera:

   KDAO=1/(1+*FeePool*/*DAOAll*)

Liquidity

Anyone who owns some of the Tera coins can issue a USD stablecoin and thus become a liquidity provider. It sends Tera coins to a smart contract, and in return receives a USDDAO liquidity token.



Adding Liquidity

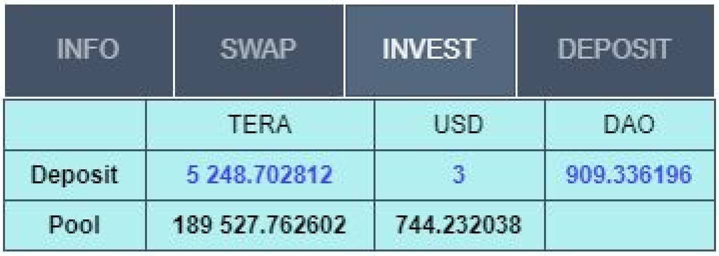
With increased liquidity, one part of the Tera coins fall into the reserve pool (`TeraToReserve`), and the other into the exchange pool (`TeraToPool`) according to the instant rates given above. The exact calculation, taking into account the commission, will be as follows:

```
TeraToReserve=TDeposit/(KReserve/KSwap+1);
TeraToPool=TDeposit-TeraToReserve;
USDToPool =TeraToReserve*KReserve;
DAO=TDeposit*KDAO;
```

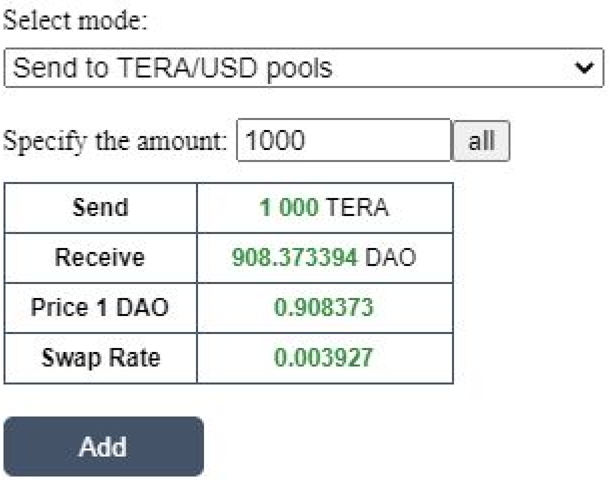
For market exchange rate management, there are three modes of adding liquidity:

1. Only to the TERA pool
2. Only to the USD pool
3. To TERA/USD pools (the exchange rate does not change)

GUI



	TERA	USD	DAO
Deposit	5 248.702812	3	909.336196
Pool	189 527.762602	744.232038	



Select mode: Send to TERA/USD pools

Specify the amount: 1000

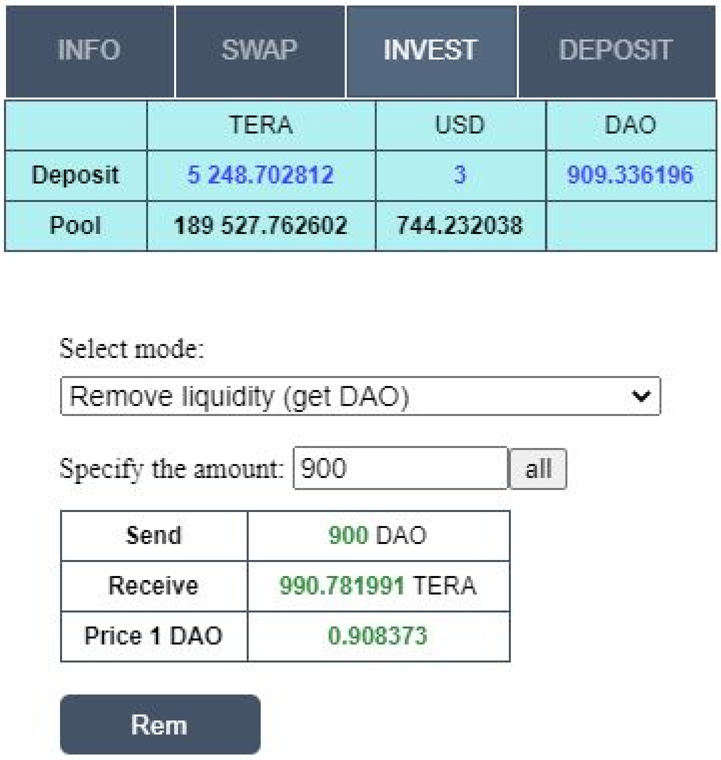
Send	1 000 TERA
Receive	908.373394 DAO
Price 1 DAO	0.908373
Swap Rate	0.003927

Removing liquidity

Presenting liquidity tokens (***DAOBurned***), you get Tera coins (*TeraWithdrawn*) in return. The rates are determined by the current state of the pools and their formulas are written above.

```
TeraWithdrawn = DAOBurned/KDAO;
TeraFromReserve = TeraWithdrawn /(KReserve/KSwap+1);
TeraFromPool   = TeraWithdrawn  - TeraFromReserve;
USDFromPool = TeraFromPool*KSwap;
```

GUI



	TERA	USD	DAO
Deposit	5 248.702812	3	909.336196
Pool	189 527.762602	744.232038	

Select mode:
Remove liquidity (get DAO)

Specify the amount: 900

Send	900 DAO
Receive	990.781991 TERA
Price 1 DAO	0.908373

Change in the reservation rate

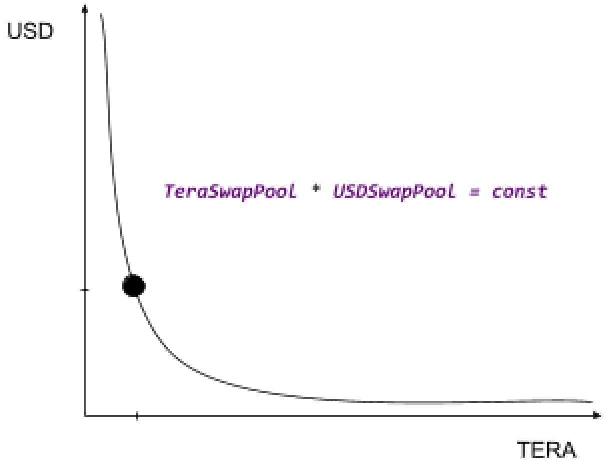
The reserve rate that determines the reliability of the USD token can be changed by voting with USDDAO tokens. To apply the new rate (***KReserveNew***), you need to rebalance the pools.

```
DeltaTera=(TeraReservePool-USDAllAmount/KReserveNew)/(1+KSwap/KReserveNew)
DeltaUSD = DeltaTera*KSwap
```

If these values are positive, then you need to issue additional USD by reducing the reserve pool and add them to the USD/TERA trading pair, and if the values are negative, then perform the opposite operations - by reducing the pools of the trading pair, increase the reserve pool.

Swap

The smart contract contains the TERA and USD liquidity reserves, and transactions are made directly against these reserves. Prices are set automatically using the market maker mechanism (x*y=k), which keeps the total reserves in relative equilibrium. The exchange is carried out under the assumption of a constant value: `TeraSwapPool * USDSwapPool = Invariant`. This invariant does not change and remains practically the same constant for all exchanges (the commission has little effect on the invariant), it changes significantly only when adding or removing liquidity.



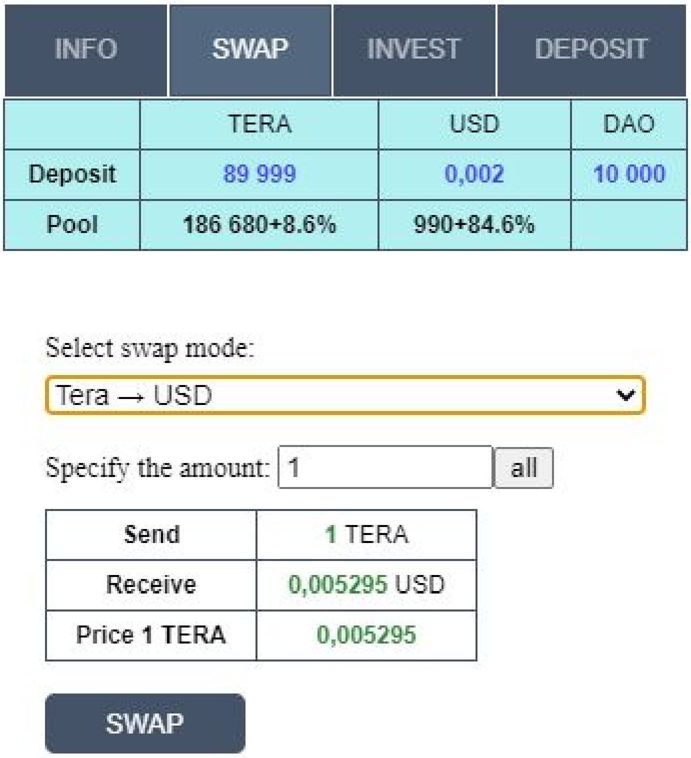
Tera to USD swap

With this exchange, `TeraSwapPool` increases and `USDSwapPool` decreases

```
var AmountFee=AmountTera*KFee;
var Invariant=TeraSwapPool * USDSwapPool;
var TeraSwapPoolNew = TeraSwapPool + AmountTera;
var USDSwapPoolNew = Invariant/(TeraSwapPoolNew - AmountFee);
var USDOut = USDSwapPool - USDSwapPoolNew;

TeraSwapPool = TeraSwapPoolNew;
USDSwapPool = USDSwapPoolNew;
```

GUI



	TERA	USD	DAO
Deposit	89 999	0,002	10 000
Pool	186 680+8.6%	990+84.6%	

Select swap mode:
Tera → USD

Specify the amount: 1

Send	1 TERA
Receive	0,005295 USD
Price 1 TERA	0,005295

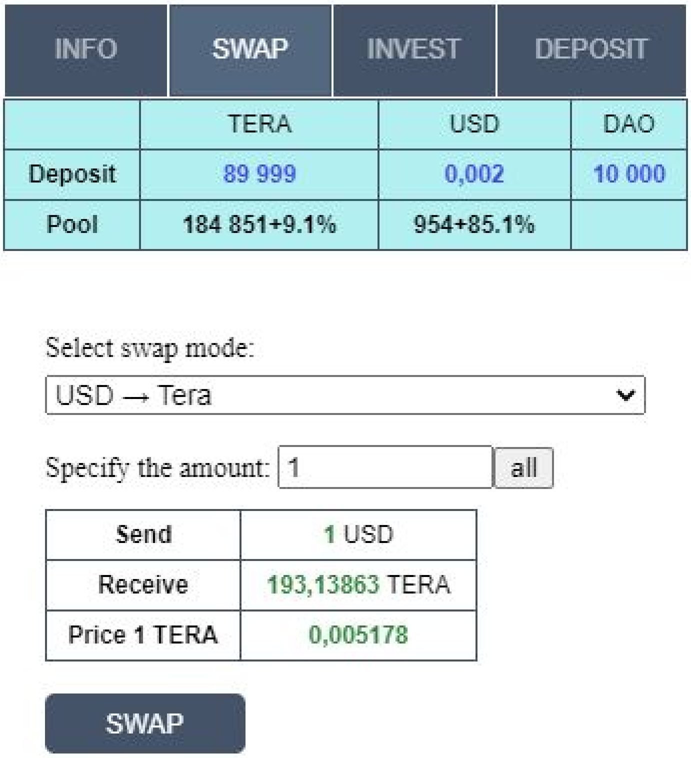
USD to Tera swap

With this exchange, the *USDSwapPool* increases and the *TeraSwapPool* decreases

```
var AmountFee=AmountUSD*KFee;
var Invariant=TeraSwapPool * USDSwapPool;
var USDSwapPoolNew = USDSwapPool + AmountUSD;
var TeraSwapPoolNew = Invariant / (USDSwapPoolNew - AmountFee);
var TeraOut = TeraSwapPool - TeraSwapPoolNew;

TeraSwapPool = TeraSwapPoolNew;
USDSwapPool = USDSwapPoolNew;
```

GUI



	TERA	USD	DAO
Deposit	89 999	0,002	10 000
Pool	184 851+9.1%	954+85.1%	

Select swap mode: USD → Tera

Specify the amount: 1

Send	1 USD
Receive	193,13863 TERA
Price 1 TERA	0,005178

Managing courses

Task stablein to have a constant exchange rate relative to the target of USD and as collateral TERA, you should exactly repeat the pair USD/TERA relative to other markets (exchanges).

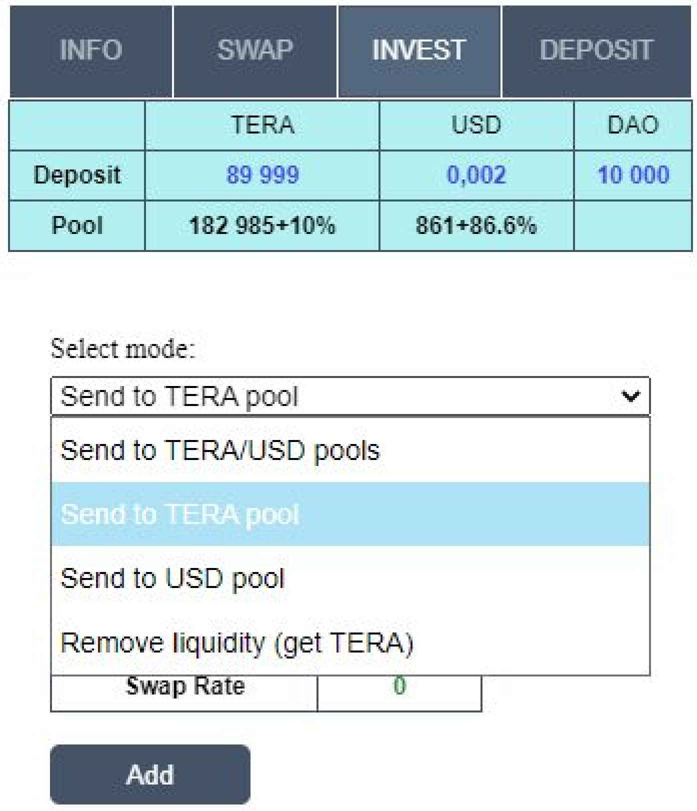
As you know the instant rate is determined by the formula:

$$KSwap=USDSwapPool\ /\ TeraSwapPool$$

Therefore, it is possible to influence the exchange rate by changing the pools (USD and TERA).

- If you increase only the TERA pool, the rate will drop
- If you only increase the USD pool, the exchange rate will increase

GUI



	TERA	USD	DAO
Deposit	89 999	0,002	10 000
Pool	182 985+10%	861+86.6%	

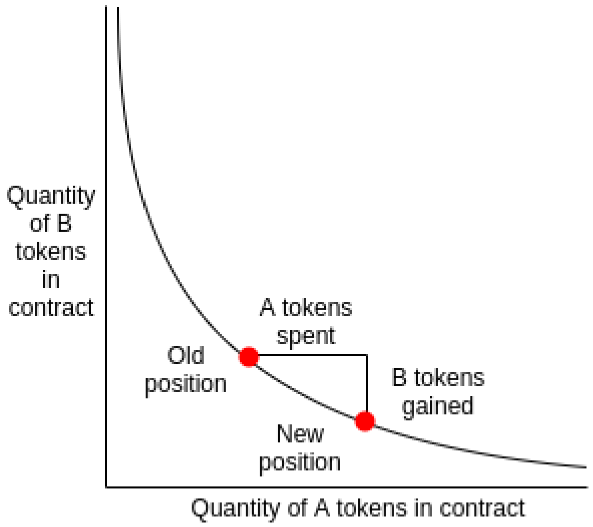
Select mode:

Send to TERA pool

Swap Rate	0

Protection against manipulation of the TERA/USD exchange rate

> Quote:
> Anyone can buy or sell coins, essentially moving the position of the market maker on the curve x∗y=k; if he will move the point to the right, then the amount by which it will move it to the right will be equal to the sum of A token that it needs to be, but the amount that he will move the point down, corresponds to how many token B he will get.


Quantity of B tokens in contract
A tokens spent
Old position
B tokens gained
New position
Quantity of A tokens in contract

Front running attack

In the process of sending an exchange order, the market situation may change, and there may be other orders ahead of our order that significantly worsen the exchange rate. Such a situation can be of a special nature, when a malicious miner can "wrap" our order with two of its own orders.

As a protection against unpredictable market conditions, you can enter the expected price and the maximum percentage of deviation from it in the parameters. At the time of execution of the transaction, a check is made for compliance with the expected values and if they do not match, the transaction is canceled.

One-way pool replenishment

This question is relevant when replenishing liquidity in the one-way pool selection mode - when only USD or only TERA is selected. With large amounts and immediate use of the pool, the balance can be significantly disturbed.

As an option for such protection, you can use a delayed application by time. Coins are frozen for a while, and then gradually thawed. If there is enough time, for example, a day, the market will have time to react to a change in the exchange rate either by issuing liquidity on another pool, or through SWAP operations.

Time interval pools

The application is postponed for 6 hours, and then gradually applied throughout the day.

Example, at time t1:
1) The liquidity of 1000 coins is sent to the TERA pool
2) These coins are placed divided into 4 parts, which are placed in time pools T2, T3,T4, T5, which are divided into 6 hour intervals and are fixed. If a similar replenishment of liquidity occurs within a short period of time, then the distribution of pools will be the same, but if for example after 6 hours, then as follows: T3,T4,T5,T6
3) After 6 hours, the gradual smooth use of temporary pools begins, coins from this pool are transferred to the main pool

Weighted average defrost

The same idea can be made through a mathematical formula-the weighted average of the amount of coins.
Let's say we have a certain amount of coins S1, which needs to be unfrozen after time T1, let's say a new amount of coins S2 is added, which needs to be unfrozen after time T2. You need to calculate the new value of S and T.

In this case, the new time will be as follows:

$$T \ = (S1 * T1 + S2 * T2)/(S1 + S2)$$

Accordingly, the new amount:

$$S \ = S1 + S2$$

The amount of defrosting per unit of time (block):

$$\Delta S \ = S/\, T$$

Instant change in liquidity

There remains an attack option in the form of:

- Exchange
- Replenishment of liquidity (for a common pair of pools)
- The reverse exchange

- Withdrawal of liquidity

Example:
Let's first have the liquidity: 5000 TERA and 15 USD
Steps:

1. Changing 10,000 TERA to 10 USD
2. We top up the liquidity by 20,000 TERA (the liquidator became 30,000 TERA and 10 USD)
3. We change 10 USD to 15 000 TERA
4. We take away the liquidity of 20 000

As a result, the prize is 5000 TERA

As a protection:

- Introduce a freeze on the withdrawal of liquidity within 24 hours.
- Or freeze pools in all modes.

Protection against freezing of pools

An attack is possible when all the pools are frozen.
Attack:

1. Add Liquidity
2. To remove liquidity
3. Repeat 1,2 in a loop

Even a small value is enough to zero out a useful part of all the liquidity.

Protection: freezing of DAO tokens