

# RIF On Chain

## Stablecoin Protocol Collateralized with RIF

[Technical whitepaper — release 2 — revision 1]

December 2023 - [www.rifonchain.com](http://www.rifonchain.com)

<sup>1</sup>**Abstract** — A RIF-collateralized stablecoin, USDRIF, that minimizes counterparty risk through a set of smart contracts. USDRIF, referred to as the pegged tokens(TP), is backed by the Rootstock network's native token, RIF, which acts as collateral (AC) – though other RRC20 tokens may be used as collateral to maintain pegs. The pegged tokens (TP), which are RRC20 tokens, are pegged to distinct fiat currencies (or other assets) by means on an oracle, e.g US\$, €, £, ¥, Au etc. The Collateral Tokens (TC), also RRC20 tokens, maintain Collateral Asset's price volatility e.g. RIF. The RIF on Chain Protocol (RoC), a set of smart contracts, returns the amount of TPs equivalent to the amount received in AC at the time of confirmation of the transaction. TC are issued when ACs are sent to the RoC Protocol. The system returns the equivalent amount at the time the transaction is recorded on the blockchain. Participants mint TC with AC, and redeem them for AC at which time they are burned.

---

<sup>1</sup> The RoC protocol is based on the MoC Protocol which is a set of smart contracts developed to represent the model behind the DAO MoneyOnChain. You can find the details of the MoC Protocol in the [MOC white paper](#)  
This is an adaptation of the [MOC white paper](#) to implement a fork of said protocol for RIF tokens and their derivatives

<b>LEGAL DISCLAIMER</b>	<b>5</b>
<b>Executive Summary</b>	<b>7</b>
<b>Description of the model behind RoC Protocol</b>	<b>9</b>
<b>Introduction</b>	<b>10</b>
<b>Nomenclature</b>	<b>11</b>
Collateral Bag	11
Peg Container	11
Variables	11
<b>The (TC) Collateral Token</b>	<b>13</b>
TC issuance	13
TC redemption	13
<b>The (TP) Pegged Token</b>	<b>13</b>
TP Issuance	13
TP Redemption	14
<b>Explaining the “bucket” concept</b>	<b>14</b>
Collateral Bag	15
Peg Container	15
<b>Definitions</b>	<b>15</b>
Global model coverage	15
Moving averages	15
Ctargema	16
Ctargematpi	17
Ctargemaca	17
Collateral Token Price	17
Maximum amount of TC that can be redeemed	17
Maximum amount of TPi that can be issued	18
Maximum amount of TPi that can be redeemed	18
<b>Success Fee (Disabled)</b>	<b>18</b>
Definitions	18
Functions	19
Pseudo code	19
qTPi emission atomic process	19
qTC emission atomic process	20
qTC emission atomic process	20
Atomic settlement process	21
Multi peg (Disabled)	21
Diagram	22

Definitions	22
<b>Liquidation</b>	<b>23</b>
Forced system liquidation	23
Mechanisms to prevent the liquidation of the system	23
<b>Joint issuance and redemption of TPs and TCs</b>	<b>23</b>
About the proportionality of the Tokens	24
Joint issuing	24
Case of use	24
Pseudocode	25
Joint redemption	26
Case of use	26
Pseudocode	26
Swaps	28
<b>Allow Different Recipient parameter</b>	<b>28</b>
<b>Vendors</b>	<b>29</b>
Vendors	29
Vendors Guardian	29
Vendors API	29
Diagram	29
Data flow	30
<b>Flux Capacitor</b>	<b>30</b>
Introduction	30
Implementation	30
Components	30
Description of operation	31
About the Decay Factor	32
Pseudo Linear Model	32
Exponential Model (Not used)	32
Graphic	33
Conditions of Acceptability & Verification	33
Limits	34
Maximum Mint/Redeem Admitted MMRA	34
<b>Queue</b>	<b>38</b>
Introduction	38
Transaction queue	38
Additional Considerations	39
<b>Apendix-Some formula explanation</b>	<b>39</b>
Maximum amount of TC that can be redeemed	39

Maximum amount of TPi that can be issued	40
Joint issuance and redemption of TPs and TCs	41

## LEGAL DISCLAIMER

This whitepaper is for information purposes only and may be subject to change. We cannot guarantee the accuracy of the statements made or conclusions reached in this whitepaper and we expressly disclaim all representations and warranties (whether express or implied by statute or otherwise) whatsoever, including but not limited to:

- any representations or warranties relating to merchantability, fitness for a particular purpose, suitability, title or non-infringement;
- that the contents of this document are accurate and free from any errors; and
- that such contents do not infringe any third-party rights. We shall have no liability for losses or damages (whether direct, indirect, consequential or any other kind of loss or damage) arising out of the use, reference to or reliance on the contents of this whitepaper, even if advised of the possibility of damages arising.

You should not definitively rely upon it or use it to form the definitive basis for any decision, contract, commitment or action whatsoever, with respect to any investment decision.

Any views or terms contained herein are preliminary only, and are based on financial, economic, market and other conditions prevailing as of the date of this paper and are therefore subject to change. We undertake no obligation or responsibility to update any of the information contained in this paper.

This paper and the information contained herein do not constitute an offer to sell or the solicitation of an offer to buy any security, commodity or instrument or related derivative, nor do they constitute an offer or commitment to lend, syndicate or arrange a financing, underwrite or purchase or act as an agent or advisor or in any other capacity with respect to any transaction, or commit capital, or to participate in any trading strategies, and do not constitute legal, regulatory, accounting or tax advice to the recipient. We recommend that the recipient seek independent third party legal, regulatory, accounting and tax advice regarding the contents of this paper. This paper does not constitute and should not be considered as any form of financial opinion or recommendation by us or any of our affiliates.

No regulatory authority has examined or approved any of the information set out in this paper. No such action has or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution, or dissemination of this paper does not imply that applicable laws or regulatory requirements have been complied with.

Participation in the transactions described in this paper carries substantial risk and may involve special risks that could lead to an economic substantial loss. Please ensure that you have read, understood and are prepared to accept the risks of participating in the transaction described herein before participating.

This paper is for the sole purpose of providing preliminary and not binding information related to the functionality of the token. Thus, this paper does not constitute an offer or recommendation to participate in any transaction and /or acquire any token.

There is no guarantee that any of the premises, estimations, projections, results (in total or partially) or conclusions used or presented in this paper will be effectively reached or will be verified in total or partially. The final results may differ from the projections and these differences may be significant for any reasons, such as, but not limited to market conditions.

Before deciding to participate in any transaction, you shall take all measures that you deem necessary to ensure the due understanding of the transaction in all its aspects and make an independent assessment of its convenience and its objectives, particularly in relation to risks and benefits of entering in such a transaction. You should also seek advice from specialized advisors (financial, tax, legal, among others) to make such an assessment.

When starting the analysis of this paper, you declare that you have agreed to all terms stated above.

## **Executive Summary**

A RIF-collateralized stablecoin is a type of stablecoin that is backed by RIF as collateral. The idea behind this type of stablecoin is that the value of the stablecoin is pegged to a stable asset, usually the US dollar but may be other, while its value is backed by the volatile asset of RIF.

To create a RIF-collateralized stablecoin, a user would need to deposit RIF as collateral into a smart contract. The smart contract would then issue stablecoins in return, which the user could use for transactions or hold as a store of value. The value of the stablecoins would remain stable as long as the value of RIF does not drill a minimum price floor referred to the asset to which you must be attached. If the value of RIF drops significantly, however, the smart contract shall take preventive and corrective measures to ensure the value of the stablecoin. This mechanism helps ensure that the stablecoin remains stable in value and can be redeemed for its underlying collateral, RIF.

RIF-collateralized stablecoins aims to provide a more stable alternative to holding RIF while still allowing users to benefit from its potential price appreciation.

This whitepaper serves a dual purpose. Firstly, it introduces the reader to the concept of RIF-collateralized stablecoins and explains how they work, including their benefits and potential drawbacks. This helps the reader gain a deeper understanding of the technology and the underlying math behind the model. Secondly, the white paper also serves as a technical document, detailing the mathematical models and algorithms used to create and maintain the stablecoin.

This section of the whitepaper may include topics such as the calculation of the collateralization ratio and the issuance and redemption of stablecoins. By providing this technical information, the whitepaper allows interested parties to fully assess the viability of the RIF-collateralized stablecoin model and potentially contribute to its development.

### **Key Features:**

1. Collateralized with RIF: RIF-collateralized stablecoins are backed by RIF, which is a volatile asset, but also an established utility token in the Rootstock ecosystem.
2. Stable Value: These stablecoins aim to maintain a stable value, usually pegged to a stable asset, such as the US dollar.
3. Decentralized: These stablecoins are created and maintained on a decentralized blockchain network, which provides transparency, security, and resistance to censorship. A decentralized

stablecoin contract is not controlled by a single entity, but instead operates on a distributed blockchain network, making it resistant to manipulation or shutdown. This ensures that the stablecoin remains transparent and accessible to all users, regardless of their location or political affiliation. Additionally, a decentralized stablecoin contract allows for peer-to-peer transactions, without the need for intermediaries, reducing the cost and time associated with traditional financial transactions. Overall, decentralization is a critical aspect of a stablecoin contract, providing trust, security, and accessibility to users.

4. Programmable: The smart contracts used to create these stablecoins can be programmed to automate various functions such as issuance, redemption, and liquidation.
5. Efficient: Transactions involving RIF-collateralized stablecoins can be conducted quickly and efficiently, without the need for intermediaries such as banks.

#### **Benefits:**

- Reduced Volatility: RIF-collateralized stablecoins offer a more stable alternative to holding RIF, reducing the risk of significant price fluctuations.
- Greater Access: These stablecoins can provide greater access to cryptocurrencies for individuals and institutions that may be hesitant to invest in volatile assets.
- Potential Yield: Holders of RIF-collateralized stablecoins may earn interest on their holdings if the smart contract allows for it.
- Borderless Transactions: Transactions with RIF-collateralized stablecoins can be conducted globally, without the need for conversion to local currencies.
- Transparency: The blockchain technology used to create and maintain these stablecoins provides transparency and accountability to users, ensuring trust in the system.

RIF-collateralized stablecoins have the potential to significantly impact the cryptocurrency industry. By providing a more stable alternative to holding RIF and other volatile cryptocurrencies, these stablecoins may attract new users and increase the adoption of cryptocurrencies. This could lead to greater mainstream acceptance and use of cryptocurrencies as a means of payment and store of value. Additionally, RIF-collateralized stablecoins could offer a way for individuals and institutions to access the benefits of cryptocurrency without the risks associated with volatility. This could lead to increased investment in the cryptocurrency industry and greater overall growth. However, there are also potential drawbacks, such as the risk of a significant drop in the value of RIF, which could lead to liquidation of collateral and potentially cause instability in the market. However, in our model, preventive and corrective measures have been implemented that prevent liquidation, which have been put to the test over the past years, enduring without being affected several abrupt falls and long periods of bear markets. Overall, RIF-collateralized stablecoins have the potential to reshape the cryptocurrency industry by offering a stable, efficient, and programmable alternative to existing cryptocurrencies.



## **Description of the model behind RoC Protocol**

RoC Protocol is a set of smart contracts developed in order to represent the model behind the DAO MoneyOnChain.

The version 2.0 of the model is described here. However, to achieve a complete understanding of the model, it is necessary to have internalized the operation of the previous version described in the former [whitepaper](#).

The following features have been added:

- Multipeg. There can be multiple TPs associated with a collateral
- The success fee instead of the flat fee, thus being more fair for long-term holders.
- Joint issuance and redemption of TPs and TCs in order to allow TP mint and TC redeem when the coverage is under de target.
- Swap operations of TPs for TCs and TCs for TPs, in both cases checking that the global coverage remains above the target coverage
- The transaction Queue is a mechanism that lifts the price Oracle demands to a high-frequency price to ensure enhanced protection of the protocol. In its essence, the transaction Queue is a smart contract that allows the collateral bag to queue orders for their execution after a certain period of time. When the user enters a transaction, it is sent to the queue. The user will be required to pay gas fee for the new queued transaction. Once the transaction enters the queue, cancellation is not permitted, and the funds become locked. The queued transaction could be executed by any address ensuring decentralization. Once the execution starts, the queue sends blocks of queued transactions to the collateral bag for execution. The collateral bag will report 1 by 1 if it was successful or failed, and If it fails, the locked funds will returned to the owner

The RIFX token, which are a 2X+ leveraged positions, have been removed from this version of the protocol. Concomitantly, the payment of interest paid by leveraged positions with all the associated complexity has been removed.

The settlement functions have also been greatly simplified.

For all the examples and descriptions, it is enlightening to assume that the stablecoin is pegged to the US dollar and that the collateral used is RIF, however, with the appropriate parameters, it is possible to

use another collateral or maintain another peg or pegs since the protocol is multipeg, but in that case, the less inflationary asset should be used as collateral of the other.

## **Introduction**

The model was designed to align the economic incentives of the different actors with the objective of maintaining pegs e.g.  $1\text{USDRIF} = 1\text{USD}$ . It was designed based on the ver 0.1 protocol but not taking other stablecoin implementations, however it solve its liquidity known problems<sup>2</sup>

It Identifies and aligns the interests of three types of cryptocurrency users.

1. The TP (USDRIF), is a RRC20 token pegged to a fiat currency (US\$) for risk-averse individuals.
2. The TC (RIFPRO), is a RRC20 token for RIF holders seeking a passive income in RIF. RIF holders, people who save their RIFs for the long term and seek an income on their RIFs, use TCs, which is a token that receives profits in AC, in addition to being slightly leveraged, which increases the holding of AC when the price goes up. The profit is variable, and depends on market conditions. 65% of all RIFs are in possession of these type of holders.
3. Other RRC20 tokens, pegged to other fiat currencies or assets. Those who need to pay in local currency, or need to transfer local currency across borders and receive value in a different currency. The MultiPeg feature is disabled in this version of the protocol.

Example:

Let's suppose Alice is a long-term holder of RIFs. Alice will find it attractive to buy TCs with her RIFs to get a passive income. Alice's purchase provides collateralization to the model.

Bob instead is not risk prone and prefers to buy TPs with his RIFs. Bob's TPs will use Alice's RIFs as collateral. Bob may use his TPs to make payments at a predictable value.

---

<sup>2</sup> "(In) Stability for the Blockchain: Deleveraging Spirals and ...." 5 jun.. 2019, <https://arxiv.org/abs/1906.02152>.

Finally, Carol, who has interests in different countries, pays his expenses with a token pegged to his local currency, or can transfer value to other countries receiving the recipient in local currency or any other..

## **Nomenclature**

Here are the definitions of the variables that are used in mathematical formulas throughout the entire document. If necessary, you should return to this paragraph to confirm the meaning of the variables when they appear in a formula.

### **Collateral Bag**

$nACcb$	Amount of Collateral Asset in Collateral Bag
$nTCcb$	Amount of Collateral Tokens in Collateral Bag
$prot\_thrld$	Protected state threshold

### **Peg Container**

$nTP(i)$	Amount of TP of the Peg Container "i"
$pACtp(i)lstop$	AC price expressed in TPs of the last operation

### **Variables**

<u><math>A</math></u>	_____
$AC$	Collateral Asset - RIF It can be an ERC-20 or the Basecoin of the chain
$ACC$	Collateral Token Coverage

<b><u>C</u></b>	
<i>Cglb</i>	Global model coverage
<i>Ctarg</i>	Overall target coverage of the model
<i>Ctargemaca</i>	Global objective coverage of the model adjusted by the moving average of the AC
<i>Ctargematpi</i>	Global objective coverage of the model adjusted by the moving average of the TPi
<b><u>D</u></b>	
<i>d(c)</i>	Discount rate e.g. %1
<i>DPC</i>	Protected mode threshold coverage
<i>DTC</i>	Discount threshold coverage
<b><u>E</u></b>	
<i>EMA(i)</i>	Exponential moving average of pACtp(i)
<b><u>L</u></b>	
<i>ITC</i>	Leverage spot del TC
<b><u>M</u></b>	
<i>maxTCrdm</i>	Maximum amount of TC that can be redeemed
<i>maxTPdsc</i>	Maximum amount of TPi that can be redeemed at a discount
<i>mocGain</i>	Refers gains due to Success Fees (see chapter)
<b><u>N</u></b>	
<i>nACcb</i>	Amount of Collateral Asset in the collateral bag
<i>nTCcb</i>	Amount of Collateral Token in the collateral bag
<i>nTPi</i>	Amount of Pegged Token in Peg Container “i”
<b><u>P</u></b>	
<i>pACtp(i)</i>	AC price in TPi
<i>pACtp(i)lstop</i>	AC price in TPi of the last operation
<i>pTCac</i>	TC price in AC
<i>TP</i>	Pegged Token - USDRIF
<i>pTPac(i)</i>	TPi price in AC 💡 $pTPac(i) = 1 / pACtp(i)$
<b><u>T</u></b>	

<i>TC</i>	Collateral Token - RIFPRO
<i>TPGain</i>	Refers gains due to Success Fees (see chapter)
<i>TPi</i>	Pegged Token i

### **The (TC) Collateral Token**

TCs receive the volatility or leverage that the TPs give them. Additionally receive other profits from other origins analyzed later.

#### **TC issuance**

In order to issue TC, ACs are sent to the RoC Protocol, in its turn, the system gives back the amount of TC equivalent to the quotation at the time the transaction is recorded in the blockchain.

#### **TC redemption**

△ Should the global coverage of the model fall below the *Ctargema* the redemption of TC was inhibited. See at **Joint issuance and redemption of TPs and TCs**

The system allows redemptions of TCs provided that the global coverage is greater than *Ctargema*. Otherwise the tokens may be traded in the secondary market.

### **The (TP) Pegged Token**

The TP was designed to represent a value as close as possible to a fiat currency. Strictly speaking, TP is not that currency nor pretends to be.

#### **TP Issuance**

△ Should the global coverage of the model fall below the *Ctargema* the issuance of TP was inhibited.

See at **Joint issuance and redemption of TPs and TCs**

TPs are issued when an address sends a sum of AC to the system requesting the issuance of the same. RoC Protocol, returns the amount of TPs equivalent to the amount of TP received in AC at the time of confirmation of the transaction.

## TP Redemption

The peg RoC Protocol ensures that it will return the equivalent of 1TP (less fee) in AC to any address that sends 1 TP token to be destroyed (Redemption)

### Explaining the “bucket” concept

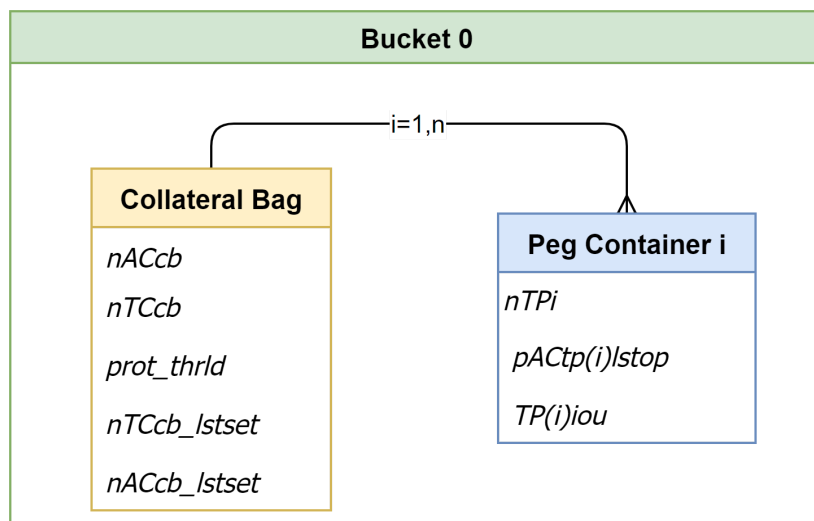
The "bucket" serves as a mental concept to comprehend the operation of the model.

A bucket is basically a container where the amount of different tokens are stored, and within it the tokens keep relationships that define the behavior of the model in that environment.

Additionally, there are global parameters, and each bucket has its own unique set..

There are established rules dictating how tokens move inside buckets, as well as procedures for getting in and getting out them (minting and redeeming).

In this instance of the model 1 bucket is defined, "Bucket 0", and is composed by a “Collateral Bag” (CB) and n “Peg Container”



## Collateral Bag

The Collateral Bag holds two tokens CT and AC. Here is where the relationship between those is defined.

The quantities of these tokens are the state variables of the bucket.

$nACcb$	Amount of Collateral Asset in the collateral bag
$nTCcb$	Amount of Collateral Token in the collateral bag
$prot\_thrld$	Protected state threshold
$nTCcb\_lstset$	Amount of Collateral Token in the collateral bag @ Last Settlement
$nACcb\_lstset$	Amount of Collateral Asset in the collateral bag @ Last Settlement

## Peg Container

There will be one Peg Container for each Pegged Token TPi

$nTPi$	Amount of Pegged Token in Peg Container “i”
$pACtp(i)lstop$	AC price in TPi of the last operation
$TPiou[i]$	Auxiliary temporary accumulator for Success fee and vaults

## Definitions

### Global model coverage

$$cglb = \frac{nACcb - mocGain}{\sum_1^n \frac{nTPi + tpGain(i)}{pACtpi}}$$

## Moving averages

A moving average will be taken by TP. The moving average for each TP will be that of its pACtp, which indicates the number of TPs equivalent to an AC.

Therefore there will be a moving average for each Peg container

*Examples:*

*If AC is RIF and TP is USD. 1 RIF equals 0.1 USD*

*$pACtp = 0.1$*

*If AC is the USD and TP is the ar\$. 1 USD is equivalent to ar\$ 300*

*$pACtp = 300$*

The moving average will have a duration expressed in days for each TP.

The moving average is calculated as follows:

$$EMApt_0 = pACtp + (1 - SF) * EMApt_{-1}$$

Where:

$EMApt_0$  is the spot moving average

$SF$  is the smoothing factor

$EMApt_{-1}$  is the current moving average so far

$pTPac0$  is the spot price of the TP

And the smoothing factor is defined as:

$$SF = 1 / (1 + D)$$

Where

$D$  is the number of days in memory of the  $EMA$ .

*Example:*

*If  $D = 20$  days*

*$SF = 1 / (1 + 20) = 0.047619048$*

## Ctargema

It is the value of the objective coverage adjusted by the moving average of the value of each  $TP$  and of the  $AC$ . Therefore there is a *Ctargema* for each  $TP$  and one for the  $AC$ .

Be them: AC

- $Ctargema_{tpi}$  the Ctargema of the  $TP_i$
- $Ctargema_{ac}$  the Ctargema of the AC



### Ctargema<sub>tpi</sub>

$Ctargema_{tpi}$  is calculated by multiplying the target coverage of the model ( $Ctar$ ) by a correction factor  $FCtpi$

$$FC_{pti} = \frac{pACtpi}{EMAtpi}$$
$$FCtp_i \leq 1 \Rightarrow FCtp_i = 1$$

Note that if  $FCtpi$  were less than 1 it takes the value 1

### Ctargema<sub>ca</sub>

This case is more complex than the previous one.

Just as we said, there is a FC for TP, but there is only one FCca. This is calculated taking into account how much collateral there is of each TPi and performing the weighted average of the FCtpi without the condition that it be greater than 1, which will then be applied to the weighted average.

$$FCca = \frac{\sum_1^n \frac{nTPi+TPgain(i)}{EMAi}}{\sum_1^n \frac{nTPi+TPgain(i)}{pACtpi (spot)}}$$

### Collateral Token Price

$$pTCac = \frac{nACcb - mocGain - \sum_1^n \frac{nTPi - tpGain}{pACtpi}}{nTCcb}$$

### Maximum amount of TC that can be redeemed

Let us take into account that when redeeming  $\Delta TC$ ,  $\Delta AC$  corresponding to its value will leave the system

$$\Delta AC = \Delta TC * pTCac$$

$TCs$  can be redeemed until coverage reaches  $Ctargema_{ca}$

$$\Delta TC = \frac{nACcb - mocGain - ctargema_{ca} * \sum_1^n \frac{nTPi + tpGain}{pACtpi}}{pTCac}$$

Maximum amount of TPi that can be issued

$$\Delta TPi = \frac{nACcb + mocGain - \sum_1^n \frac{(nTPi + TPgain)}{pACtpi} * ctargema_{tpi}}{(ctargema_{tpi} - 1)} * pACtpi$$

Maximum amount of TPi that can be redeemed

$$nTP_i + TPgain$$

### **Success Fee (Disabled)**

#### Definitions

Appreciation factor

FASF is the percentage of profits split between the Success Fee and the governance token holders

FASF = FA + SF

FA is the percentage of profit that is returned to the Success Fee.

SF is the percentage of profit that is passed on to MOC token holders.

For example, if FASF = 60%

FA = 50%, and SF = 10%%

This means that 10% will go to the MOC token holders and 50% will go to the Success Fee.

## Functions

On the Fly P&L TPi

$$otfPnLtpi = \frac{nTPi}{pACtpilstop} - \frac{nTPi}{pACtpi}$$

Adjusted Fly P&L TPi

$$adjPnLtpi = otfPnLtpi + TPiou[i] \quad *$$

TPigain

$$TPigain = si(adjPnLtpi < 0; 0; else adjPnLtpi * FA)$$

Gain for TP1 on TP1

$$TPitpgain = TPigain * pACtp(i)$$

Gain for moc in TPi

$$mocdueTPigain = si(adjPnLtpi < 0; 0; else adjPnLtpi * SF)$$

mocGain

$$mocGain = \sum_1^n mocdueTPgain$$

pTCacreal

$$pTCreal = \frac{nACcb - mocGain - \sum_1^n \frac{nTPi + TPitpgain}{pACtpi}}{nTCcb}$$

## Pseudo code

qTPi emission atomic process

TPiou[i] = TPiou[i] + otfPnLtpi; //to TPiou is added on the fly P&L

nTPi = nTPi + qTPI; //nTPi is added to the amount that is issued

nACcb = nACcb + (qTPi / pACtpi); //to the collateral nACcb the value in AC of the qTPi that are issued is added

```
pACtplstop[i] = pACtpi; // the price of the last operation is updated
```

This code processes the emission of a certain amount of Tokenized Positions (TPi). When this process occurs, the following happens:

- The on-the-fly profit and loss (P&L) is added to the TPi-iou variable for TPi.
- The amount of TPi issued (qTPi) is added to the total amount of Tokenized Positions issued (nTPi).
- The value of the qTPi issued is added to the total amount of collateral (nCAcb) by dividing it by the current real-time value of pACtpi.
- The price of the last operation for TPi is updated.

### qTC emission atomic process

```
nTCcb = nTCcb + qTC; // add the newly issued Token Collateral (TC) to the total amount of TC collateral (nTCcb)
```

```
nCAcb = nCAcb + (qTC / pTCacreal); // add the value of the newly issued TC to the total amount of collateral (nCAcb) by dividing the TC amount by the real-time value of pTCacreal
```

This code processes the emission of a certain amount of Token Collateral (TC). When this process occurs, the following happens:

- The newly issued amount of Token Collateral (qTC) is added to the total amount of Token Collateral (nTCcb).
- The value of the newly issued TC is added to the total amount of collateral (nCAcb) by dividing it by the real-time value of pTCacreal.

### qTC emission atomic process

```
nTCcb = nTCcb + qTC; // add the newly issued tokens to the Token Collateral amount
```

```
nCAcb = nCAcb + (qTC / pTCacreal); // add the value of the issued tokens to the collateral nCAcb
```

The first line adds the amount of newly issued tokens to the existing Token Collateral amount, and the second line adds the value of the newly issued tokens to the existing collateral (nCacb). The value of the tokens is calculated by dividing the quantity of tokens issued (qTC) by the actual price of the Token Collateral (pTCacreal).

### Atomic settlement process

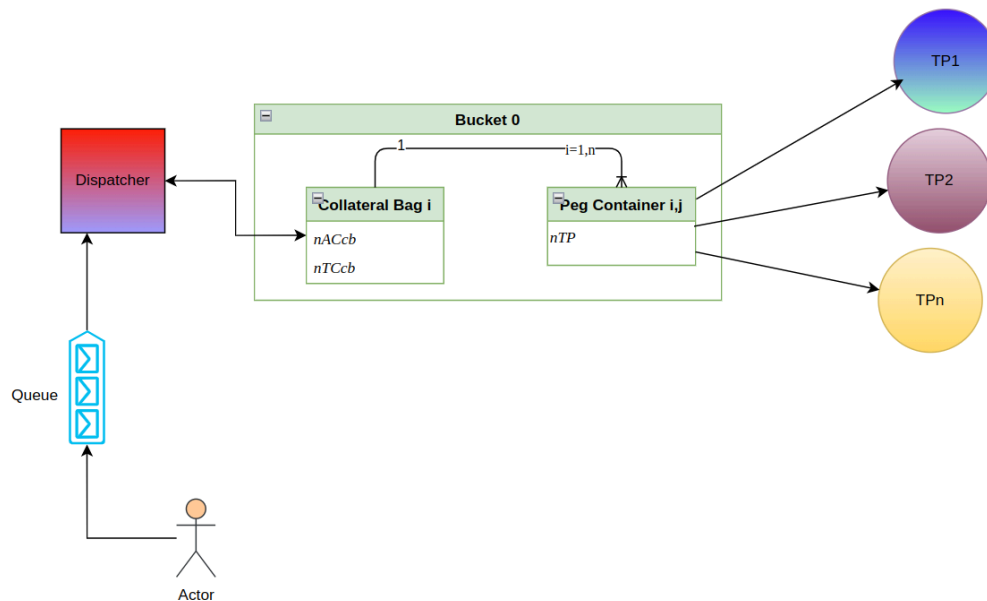
```
for (uint i = 1; i <= n; i++) { // iterate over all the TPi
    TP(i).iou = TP(i).iou + otfPnLtpi; // add the on-the-fly P&L to TPiiou
    if (TP(i).iou > 0) { // if TPiiou is greater than 0
        nTPi = nTPi + TPtpcgain; // add TPtpcgain to nTPI
        nCacb = nCacb - mocTPigain; // subtract mocTPigain from nCacb
        mocflow.transfer(mocTPigain); // send mocTPigain to mocflow
        TP(i).iou = 0; // set TPiiou to 0
    } else { // otherwise
        // do nothing
    }
    pACtp(i).lstop = pACtpi; // update the price of the last operation
}
```

These lines of code describe the different operations that are performed inside the for loop. The loop iterates over all the TPi from 1 to n. The first line adds the on-the-fly P&L to the TPiiou variable. The if statement checks if the TPiiou is greater than 0. If it is, the code adds TPtpcgain to nTPI, subtracts mocTPigain from nCacb, and sends mocTPigain to mocflow. It also sets TPiiou to 0. If TPiiou is not greater than 0, the code does nothing. Finally, the code updates the price of the last operation for the ith TPi.

### Multi peg (Disabled)

For the collateral bag there will be several linked tokens associated (just a few). The model maintains the amounts of each linked token for the collateral bag.

## Diagram



## Definitions

- TP = Token Peg
- TC = Token collateral
- CA = Collateral Asset
- TG = Token Governance
- CB = Collateral bAG
- There will be one Collateral Bag
- There will be a Peg Container for each TP a in this case the  $nTP_{i,j}$  represent amounts, but those that correspond to the same peg are fungible among themselves, they are different amounts of the same token.  $TP(1,j)=TP(2,j)=\dots=TP(n,j)$
- The Collateral Bag has its own
  - Target coverage
  - Spot price
  - EMA
  - Target coverage adjusted by exponential moving average (Ctargema)
- The Target global coverage adjusted by the exponential moving average of each collateral bag (Ctargema) is the weighted average of their own Peg Containers.

- Given a token  $TP(x,j)$  the collateral "i" may be redeemed according to the spot coverage of the collateral bag.
- In the event of liquidation, it will be done at a coverage greater than 1, the excess of 1 being the penalty for liquidation.

## **Liquidation**

### **Forced system liquidation**

If a "black swan event" occurs and the global coverage index reaches the liquidation threshold UC, the system is liquidated. In such an event the RIFPros are cleared and the USDRIFs are available to be redeemed at the PEG value.

There is a mechanism to deposit the USDRIFs in the system and receive the equivalent in RIF at the time of the liquidation.

The protocol's forced liquidation mechanism could be enabled or disabled via a governance vote by the community.

### **Mechanisms to prevent the liquidation of the system**

From the formulas expressed above it is clear that the system is designed, depending on the parameters used, with a strong overcollateralization<sup>3</sup>. The controls in the issuance and redemption of USDRIFs and RIFPro make the system tend to equilibrium.

## **Joint issuance and redemption of TPs and TCs**

In order to issue TPs and/or redeem TCs, it is necessary that the global coverage of the system is greater than the target coverage. This coverage is a parameter, 5.5 for the RoC contract, corrected by the relationship between the value of the AC spot and its moving average.

When a user wants to issue TCs and the contract is in the non-issuance state, he cannot do so, just as if he wants to redeem TCs, he cannot.

To remedy this effect, joint issuance and redemption are incorporated, where a carefully calculated amount is issued or redeemed so that it does not affect the global coverage of the protocol in any way.

---

<sup>3</sup> "Overcollateralization (OC) - Investopedia." 23 mar..2023 <https://www.investopedia.com/terms/o/overcollateralization.asp>.

Below you can see how the token proportions are calculated in both operations

## About the proportionality of the Tokens


Being *cobop* the coverage to which the joint operation must be carried out

$$\begin{aligned}nTP &= qAC * pACtp * \frac{1}{cobop} \\ nTC &= qAC * pACtc * \frac{cobop-1}{cobop}\end{aligned}$$

The ratio  $nTC/nTP$  is:

$$\frac{nTC}{nTP} = \frac{qAC * pACtc * \frac{cobop-1}{cobop}}{qAC * pACtp * \frac{1}{cobop}} = \frac{pACtc * (cobop-1)}{pACtp}$$

$$\begin{aligned}nTC &= \frac{pACtc * (cobop-1)}{pACtp} * nTP \\ nTP &= \frac{pACtp}{pACtc * (cobop-1)} * nTC\end{aligned}$$

 Suppose 2 tokens TPo (source) and TPd (destination) where I want to exchange equal value of TPo for equal value of TPd.

in the case where:

Ctargema\_destination <= Ctargema\_origin  $\Rightarrow$  without checking coverage

But

Ctargema\_destination > Ctargema\_origin  $\Rightarrow$  check availability

## Joint issuing

### Case of use

- The user wants to issue a quantity  $qTP$  and for this he has to jointly issue a proportional quantity of TC.
- If the  $qAC$  that it sends does not reach the process, it is canceled.
- If  $qAC$  are left over in the process, they are returned to the user.
- In the UI, a slippage that increases the  $qAC$  must be taken into account.



- If the user declares 0 in the number of TPs, it emits with all the qAC

## Pseudocode

```
function jointEmission(uint qTP, uint qAC, uint pACtp, uint pACtc, uint ctargema, uint fee) public {
    // qTP: amount of TP the user wants to emit
    // qAC: amount of AC the user sends for the joint issuance
    // pACtp: spot price of AC expressed in TP
    // pACtc: spot price of AC expressed in TC
    // ctargema: target coverage adjusted by EMA of TP
    // fee: operation fee
    uint prop = pACtc * (ctargema - 1) / pACtp;
    if(qTP > 0) {
        uint nTCtoMint = qTP * prop;
        uint netAC = (nTCtoMint / pACtc) + (qTP / pACtp);
        uint feeCharge = netAC * fee;
        require(qAC >= netAC + feeCharge, "Insufficient AC sent for issuance");
        // Mint new TP and TC tokens
        mint(nTP);
        mint(nTCtoMint);
        uint change = qAC - netAC - feeCharge;
        // Send fee to MoCkflow contract
        sendto(MoCkflow, feeCharge);
        // Send new TP, new TC, and change tokens to the user
        sendtouser(nTP, nTCtoMint, change);
    } else {
        uint feeCharge = qAC * fee;
        uint netAC = qAC - feeCharge;
        uint nTCtoMint = netAC * pACtc * (ctargema - 1) / ctargema;
        uint nTPtoMint = nTCtoMint / prop;
        // Mint new TP and TC tokens
        mint(nTPtoMint);
        mint(nTCtoMint);
        // Send fee to MoCkflow contract
        sendto(MoCkflow, feeCharge);
        // Send new TP and new TC tokens to the user
        sendtouser(nTPtoMint, nTCtoMint, 0);
    }
}
```

The code defines a function called jointEmission that takes six input parameters: qTP, qAC, pACtp, pACtc, ctargema(i), and fee.

The purpose of the function is to issue a joint emission of two types of tokens, called "TP" and "TC", where the number of TP tokens to be issued is determined by the user's input  $q_{TP}$ , and the number of AC tokens sent by the user for the joint issuance is determined by the input  $q_{AC}$ .

The function then calculates the exchange rate between TP and TC tokens based on the spot prices of AC tokens denominated in TP and TC, as well as a target coverage ratio adjusted by an exponential moving average of TP tokens denoted by  $ctargema(i)$ .

If the user has specified a positive value for  $q_{TP}$ , the function proceeds to calculate the number of TC tokens that need to be minted in order to achieve the desired issuance of TP tokens, based on the exchange rate previously calculated. It then checks if the user has sent enough AC tokens to cover the cost of issuing the TP and TC tokens as well as a fee for the transaction. If the user has not sent enough tokens, the function will return the AC tokens to the user and terminate the transaction. Otherwise, the function will mint the required TP and TC tokens, subtract the transaction fee from the total amount of AC tokens, and send the transaction fee to a "MoCflow" address, while sending the TP and TC tokens to the user.

If the user has specified a zero or negative value for  $q_{TP}$ , the function proceeds to calculate the number of TP and TC tokens that need to be minted based on the total number of AC tokens sent by the user, subtracting a transaction fee from the total amount. It then sends the transaction fee to the "MoCflow" address and sends the TP and TC tokens to the user.

Overall, the purpose of the function is to facilitate a joint issuance of two types of tokens, based on the user's input and the current market conditions, while ensuring that the transaction is properly executed and a transaction fee is charged.

## Joint redemption

### Case of use

- The user wants to redeem an amount  $q_{TC}$  and for this he has to jointly redeem a proportional amount of TP.
- If the  $q_{TP}$ s it sends do not reach the process, it aborts.
- In the UI, a slippage that increases the  $q_{TP}$  must be taken into account.
- If  $q_{TP}$  are left over in the process, they are returned to the user.

## Pseudocode

```
function jointRedemption(uint qTP, uint qTC, uint pACtp, uint pACtc, uint cglob, uint fee)
// qTC: amount of TC that the user sends for redemption
// qTP: amount of TP that the user sends for joint redemption
// pACtp: spot price of the AC expressed in TP
// pACtc: spot price of the AC expressed in TC
```

```

// cglob: global coverage of the model
// fee: transaction fee
uint prop = (cglob-1)*(ctargemaTP-1)/(ctargemaCA-1);
prop = pACtc * prop / pACtp
if ((qTC/qTP) > prop) {
    sendback(qTP, qTC);
    return;
}
uint TPtoRedeem = qTC/prop;
uint change = qTP - TPtoRedeem;
uint qAC = (qTC/pACtc) + (TPtoRedeem/pACtp);
uint feeCharge = qAC * fee;
uint int = getint(TPtoRedeem);
uint netAC = qAC - feeCharge - int;
redeem(TPtoRedeem, qTC);
sendtouser(netAC, change);
sendtomocflow(feeCharge);
send(int, address);
}

```

This function handles the joint redemption of Token Collateral (TC) and Tokenized Positions (TP) for an Asset Coverage (AC) expressed in both TC and TP. Here's what it does step by step:

- Calculates the proportion (prop) of the AC that is covered by TC and the remaining amount that is covered by TP. This is done by dividing the TC price (pACtc) by the TP price (pACtp) multiplied by the global coverage of the model (cglob) minus 1.
- Checks if the proportion of TC sent by the user for redemption is greater than the calculated prop. If so, the function returns the TP and TC amounts back to the user.
- Calculates the amount of TP to be redeemed based on the amount of TC sent by the user and the calculated prop.
- Calculates the amount of AC (qAC) that the user will receive after redemption. This is done by adding the amount of TC and the amount of TP redeemed, divided by their respective spot prices.
- Calculates the transaction fee based on the amount of AC.
- Calculates the amount of TP to be charged as an additional fee.
- Calculates the net amount of AC that the user will receive after deducting the transaction fee and the fee.
- Calls the redeem() function to redeem the TP and TC amounts.
- Sends the net amount of AC and the remaining TP back to the user.
- Sends the transaction fee to the MocFlow contract.
- Sends the fee to the specified address.

## **Swaps**

Swaps are joint operations encapsulated in atomic transactions. Several types of these, namely:

TP<sub>a</sub> by TP<sub>b</sub>

In this case, the swap atomically resolves the redemption of the TP<sub>a</sub>, and with the collateral obtained it mints TP<sub>b</sub> and delivers them to the user. In order to use this feature there must be more than one TP defined and the number of possible swaps will be the combinatorics of the number of swaps

TP by TC

In this case, the swap atomically resolves the redemption of the TP, and with the collateral obtained it mints TC and delivers them to the user. There are one swap per TP defined

TC by TP

In this case, the swap atomically resolves the redemption of the TC, and with the collateral obtained it mints TP and delivers them to the user. There are one swap per TP defined

Mint TC and TP

It is thoroughly explained in the following link

[Joint issuing](#)

Redeem TC and TP

It is thoroughly explained en el following link

[Joint redemption](#)

## **Allow Different Recipient parameter**

All nine operation types now offer the option to either send tokens to the same account that executed the operation or to a different account.

A new parameter, "allowDifferentRecipient," has been introduced to control this functionality. When set to "true," it enables the ability to send tokens to a different account. However, for security reasons, in the case of RIF on Chain Protocol, this parameter will remain permanently deactivated, ensuring that operations can only be performed within the user's own account.

## Vendors

### Vendors

Vendors are a third party that wants to integrate with the RoC protocol to allow USDRIF mint and redeem transactions on their platform, and the system allows them to add a markup fee on the basis applied.

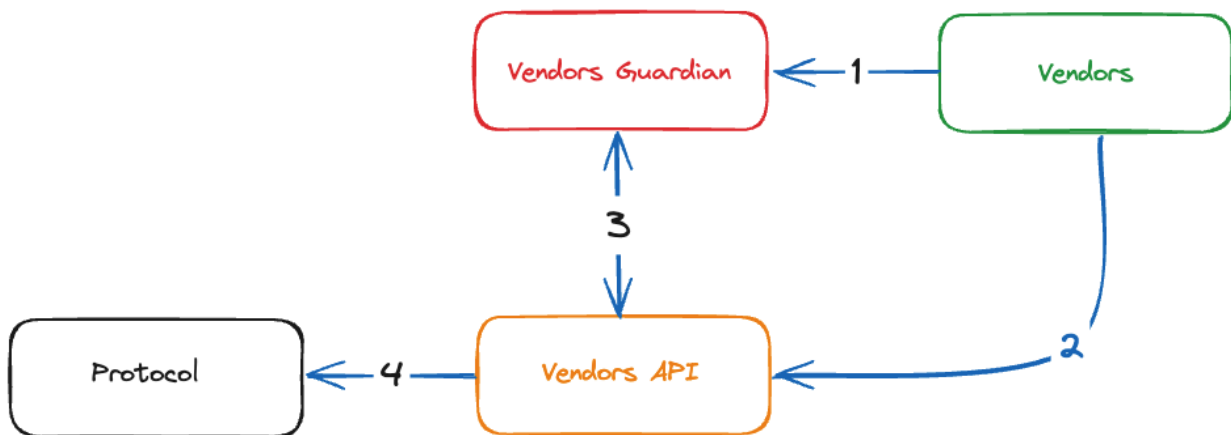
### Vendors Guardian

Is an address allowed to modify vendors markups that they will apply to the transactions they carry out.

### Vendors API

It is a smart contract where vendors send their transactions. This contract verifies that the vendor is registered and applies the markup established in the registry to it. If it is not registered, the transaction is unsuccessful. Finally, this contract communicates with the protocol API to send the transaction.

### Diagram



## Data flow

1. Vendors register in the Vendors Guardian and establish their markup
2. Vendors submit transactions to the Vendors API
3. Vendors API interacts with the Vendors Guardian to verify that the vendor is registered and, if so, receives its markup
4. Vendors API sends the correctly formed transaction to the protocol

## **Flux Capacitor**

### Introduction

The Flux Capacitor is a transaction doser based on a set of methods to be added in the flow to the protocol, with or without the transaction queue, that prevent the execution of a price variation or price manipulation attack.

The Flux Capacitor operation is based on two concepts.

1. A maximum transaction value for a given number of blocks.
2. Detects and prevents a train of opposite issuance-redemption transactions.

### Implementation

#### Components

##### **Absolute Accumulator - AA**

Accumulates the absolute value of minting and redemption transactions.

Both one and the other add their value

##### **Differential Accumulator - DA**

Accumulates the signed value of minting and redemption transactions.

The emission ones add up, while the redemption ones subtract

**Block Number of the Last Accepted Transaction - BNLAT**

It is a storage where the block number of the entered transaction is stored and which is updated with each transaction

**Parameters****Absolute Maximum Transaction Allowed - AMTA**

It is the absolute maximum transaction allowed for a certain number of Blocks.

**Maximum Operational Difference Allowed - MODA**

See the variable OD, MODA is the maximum value of OD allowed.

**Decay Factor - DF**

It is the value at which the ability to transact must be released block by block.

**Variables****Operational Difference - OD**

$$OD = AA - |DA|$$

$$OD' = AA' - |DA'|$$

$$OD'' = AA' + |MR| - |DA' + MR|$$

**Maximum Transaction Admitted - MTA**

$$MTA = AMTA - AA$$

**Description of operation**

At the beginning the accumulators are at 0, therefore  $OD = 0$  and  $MTA = AMTA$

Suppose an issuance transaction is entered with value  $V < MTA$

In this case, the accumulators will both have the value  $V$ ,  $AA = DA = V$

Therefore  $MTA = AMTA - V$  and  $OD = |V - V| = 0$

$$OD = AA + |V| - |DA + V| = AA + |V| - |DA| - |V|$$

$$OD = AA - |DA| = V - V = 0$$

Now suppose that a redemption transaction of value  $U < MTA$  is entered

The new values of the accumulators and variables will be

$$AA = V + U$$

$$DA = V - U$$

$$OD = |AA - DA| = |V + U - V + U| = 2 * |U|$$

$$OD = AA + |U| - |DA - U|$$

$$MTA = AMTA - AA = AMTA - V - U$$

## About the Decay Factor

### Pseudo Linear Model

In this case one more parameter must have been taken into account. This is Block Span BS. It is the number of blocks that have to elapse for the linear decay factor to be 0.

Validating the transactions implies verifying points 1 and 2 of Conditions of acceptability.

$$n = CB - BNLAT$$

$LDF = \text{Linear Decay Factor}$

$$LDF = \frac{-n}{BS} + 1$$

$$\text{if } LDF < 0 \rightarrow LDF = 0$$

$$\textcircled{1} AA' = AA * LDF$$

$$\textcircled{2} DA' = DA * LDF$$

If the transaction is validated:

$$CB = BNLAT$$

Otherwise revert  $\textcircled{1}$  &  $\textcircled{2}$

### Exponential Model (Not used)

The Exponential Decay Factor ( $EDF$ ) is a number between 0 and 1 by which both  $AA$  and  $DA$  must be multiplied and updated block by block.

Since the number of blocks elapsed since the last transaction can be 0, 1 or “n”, the current block “ $CB$ ” and the Block Number of the Last Accepted Transaction “ $BNLAT$ ” must be taken into account.

Validating the transactions implies verifying points 1 and 2 of Conditions of acceptability.



Therefore, before validating the transaction, it must be done

$$n = CB - BNLAT$$

$$\textcircled{1} AA = AA * EDF^n$$

$$\textcircled{2} DA = DA * EDF^n$$

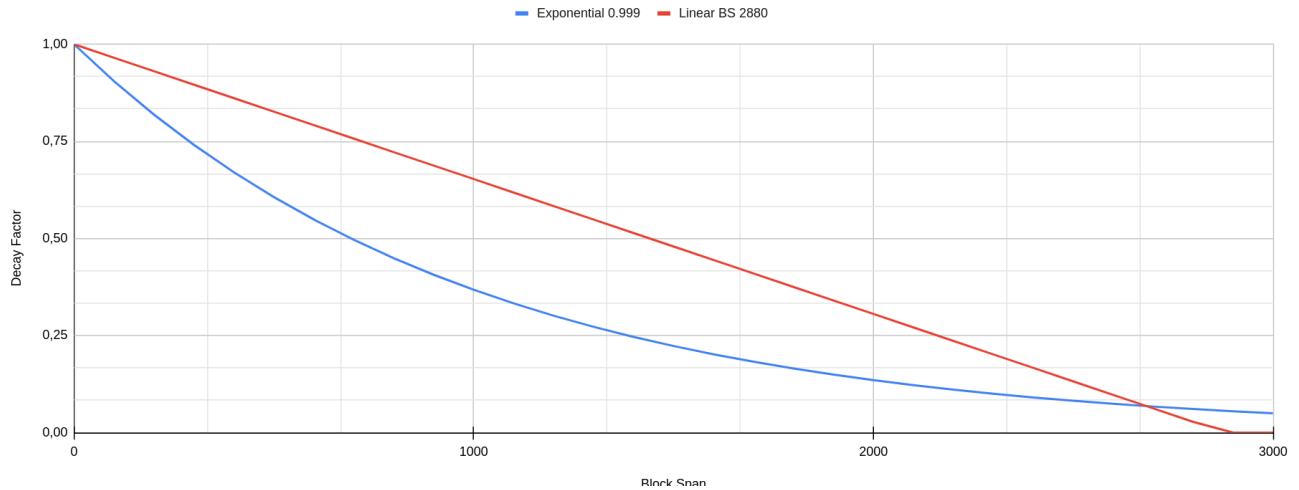
If the transaction is validated:

$$CB = BNLAT$$

Otherwise revert  $\textcircled{1}$  &  $\textcircled{2}$

## Graphic

The curve of the decay function will be bounded by the exponential curve and the straight line



## Conditions of Acceptability & Verification

When a candidate transaction is proposed, two conditions must be verified.

1. The transaction value must not exceed the MTA  
 $MTA = AMTA - AA$
2. The absolute value resulting from the OD after the transaction has been carried out must not exceed MODA.

In the event that a condition is not verified, the transaction must be rejected and fail.

These two conditions are expressed in the next conditional expression:

IF(abs(MR)<=(AMTA-AA");TRUE;FALSE) from  $\textcircled{1}$

AND

IF((abs(OD'')<=MODA;TRUE;FALSE) from ②

## Limits

### Maximum Mint/Redeem Admitted MMRA

Some definitions:

MR	Mint / Redeem value
AA	<i>Absolute Accumulator as is in the Storage</i>
AA'	<i>AA adjusted by the Decay Factor</i>
AA''	$AA' +  MR $
DA	<i>Diferencial Accumulator as is in the Storage</i>
DA'	<i>DA adjusted by the Decay Factor</i>
DA''	$DA' + MR$
OD	$AA -  DA $
OD'	$AA' -  DA' $
OD''	$AA' +  MR  -  DA' + MR $

### Conditions of acceptance

1) IF(abs(MR)<=(AMTA-AA'');TRUE;FALSE) from ①

$AA' + ABS(MR) \leq AMTA$

$|MR| \leq AMTA - AA'$

2) IF((abs(OD'')<=MODA;TRUE;FALSE) from ②

$|OD''| \leq MODA$

$OD'' = AA' + |MR| - |DA' + MR|$

$AA' + |MR| - |DA' + MR| \leq MODA$

## Conditions of acceptance

①  $AA' + \text{ABS}(MR) \leq AMTA$

$$AA' + |MR| \leq AMTA$$

$$|MR| \leq AMTA - AA' \quad (\alpha) \quad \checkmark$$

②  $\text{IF}((\text{abs}(\text{OD})) \leq \text{MODA}; \text{TRUE}; \text{FALSE})$

$$AA' + |MR| - |DA' + MR| \leq MODA$$

	MR (+)	MR (-)
DA' (+)	Ⓐ	Ⓑ
DA' (-)	Ⓒ	Ⓓ

Case Ⓐ  $DA' (+) \wedge MR (+)$

$$AA' + |MR| - |DA' + MR| \leq MODA$$

$$AA' + MR - (|DA'| + |MR|) \leq MODA$$

$$AA' + |MR| - |DA'| - |MR| \leq MODA$$

$$AA' - |DA'| \leq MODA \quad (\beta) \quad \checkmark$$

Case Ⓓ  $DA' (-) \wedge MR (-)$

$$AA' + |MR| - |DA' + MR| \leq MODA$$

$$AA' + |MR| - (|DA'| + |MR|) \leq MODA$$

$$AA' + |MR| - |DA'| - |MR| \leq MODA$$

$$AA' - |DA'| \leq MODA \quad (\beta) \quad \checkmark$$

Case Ⓑ  $DA' (+) \wedge MR (-)$

$$|AA' + |MR| - |DA' + MR|| \leq MODA$$

Case Ⓑ<sub>1</sub>  $|DA'| \geq |MR|$

$$AA' + |MR| - |DA' + MR| \leq MODA$$

$$AA' + |MR| - (|DA'| - |MR|) \leq MODA$$

$$AA' + |MR| - |DA'| + |MR| \leq MODA$$

$$AA' - |DA'| + 2|MR| \leq MODA (\gamma)$$

$$|MR| \leq \frac{MODA - AA' + |DA'|}{2} (\gamma) \checkmark \checkmark$$

Case  $\textcircled{b}_2$   $|DA'| < |MR|$

$$AA' + |MR| - |DA' + MR| \leq MODA$$

$$AA' + |MR| - ||MR| - |DA'|| \leq MODA$$

$$AA' + |MR| - |MR| + |DA'| \leq MODA$$

$$AA' + |DA'| \leq MODA (\beta) \checkmark \checkmark \checkmark$$

Case  $\textcircled{c}$   $DA' (-) \wedge MR (+)$

$$AA' + |MR| - |DA' + MR| \leq MODA$$

Case  $\textcircled{c}_1$   $|DA'| \geq |MR|$

$$AA' + |MR| - |DA' + MR| \leq MODA$$

$$AA' + |MR| - |MR + DA'| \leq MODA$$

$$AA' + |MR| - |MR - DA'| \leq MODA$$

$$AA' + |MR| - ||DA'| - |MR|| \leq MODA$$

$$AA' + |MR| - |DA'| + |MR| \leq MODA$$

$$|MR| \leq \frac{MODA - AA' + |DA'|}{2} (\gamma) \checkmark \checkmark$$

Case  $\textcircled{c}_2$   $|DA'| < |MR|$

$$AA' + |MR| - |DA' + MR| \leq MODA$$

$$AA' + |MR| - |-DA' + MR| \leq MODA$$

$$AA' + |MR| - ||MR| - |DA'|| \leq MODA$$

$$AA' + |MR| - |MR| + |DA'| \leq MODA$$

$$AA' + |DA'| \leq MODA (\delta) \checkmark \checkmark$$

In cases  $\textcircled{b}_1$  and  $\textcircled{c}_1$  the limitation is given by:

$$|MR| \leq \frac{MODA - AA' + |DA'|}{2}$$

There are borderline cases in which the sign of DA' is inverted when applying the total amount available for redemption or minting. In these cases the maximum that should be applied is:

$$|MR| \leq AMTA - AA'$$

In other cases there is no limitation for the mint and redeem amount other than that imposed by the condition ① which is:

$$|MR| \leq AMTA - AA'$$

If for any case the maximum to be redeemed or minted is less than 0, it must be forced to 0

The general expression is:

```

if (DA' = 0):
    maxMint = maxRedeem = AMTA-AA'

if (DA' > 0):
    maxMint = AMTA-AA'
    max Redeem = min(AMTA-AA';(MODA-AA'+|DA'|)/2)
    If sign(DA')<>sign(DA'- maxRedeem):
        maxRedeem =AMTA-AA'

if (DA' < 0):
    maxRedeem = AMTA-AA'
    max Mint = min(AMTA-AA';(MODA-AA'+|DA'|)/2)
    If sign(DA')<>sign(DA'+ Mint):
        maxMint =AMTA-AA'

if (maxMint < 0):
    maxMint = 0

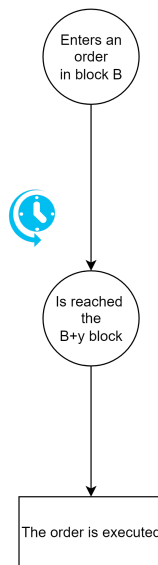
if (maxRedeem < 0):
    maxRedeem = 0

```

## Queue

### Introduction


The transaction queue is a mechanism that allows the protocol to work with slow and less precise oracles, and therefore less expensive ones. This consists of the implementation of a queue of transactions, which receive a delay for their execution. State machine model



### Transaction queue

The transaction queue is a smart contract that communicates the collateral bag contracts with the user. It has the following functions:

- Add an order to an execution queue.
- The queue is executed every certain period of time
- It starts in the pending state, it is in this pending state for at least 1 block, that is, the transaction cannot be executed in the block that the request is mined.
- There is a waiting time of "y" blocks for the transaction to be executed

-  Interesting to note that if the parameter “y” is set to a number greater than the value of the protocol's “no price” status then transactions are only made after a price update.
- The transaction queue does not verify that the requested operation is achievable
  - Neither for sufficient coverage
  - Nor for sufficient quantities
- Verification is done as usual by the main contract
- In case the order is not feasible it fails and is discarded giving the funds back to the owner.
- If any [flux capacitor](#) accumulators are reached, that's means that if the current operation fails by that condition it not be discarded and any other cannot be executed

## Additional Considerations

The transaction is entered in the queue and will be executed at its time.

Could be executed or not. In the event that the order would not be viable, it is discarded and the funds are returned to the user.

When the order is effectively executed, everything happens as usual.

1. The execution of the transaction queue is public and the executor receives the execution fee accumulated for each operation enqueued
2. The Oracle's nodes will run the transaction queue “ticks” and receive an economic incentive for doing so
3. There will be some transaction batching and paging when there is high volume. This could be avoided depending on the size of the chunk executed at once. Eventually, it could be an order at once
4. Execution of orders is strictly FIFO
5. “y” is a fixed parameter
6. The user will pay a fee in advance when he/she enters the tx
7. The tx cannot be canceled after it enters the queue, the funds will be locked

## Appendix-Some formula explanation

### Maximum amount of TC that can be redeemed

Let us take into account that when redeeming  $\Delta TC$ ,  $\Delta AC$  corresponding to its value will leave the system

$$\Delta AC = \Delta TC * pTCac$$

TCs can be redeemed until coverage reaches  $Ctargema_{ca}$

$$cglob = \frac{nACcb - mocGain}{\sum_1^n \frac{nTPi + tpGain}{pACtpi}}$$

$$Ctargema_{ca} = \frac{nACcb - mocGain - \Delta AC}{\sum_1^n \frac{nTPi + tpGain}{pACtpi}}$$

$$Ctargema_{ca} = \frac{nACcb - mocGain - \Delta TC * pTCac}{\sum_1^n \frac{nTPi + tpGain}{pACtpi}}$$

$$Ctargema_{ca} * \sum_1^n \frac{nTPi + tpGain}{pACtpi} = nACcb - mocGain - (\Delta TC * pTCac)$$

$$\Delta TC * pTCac = nACcb - mocGain - (Ctargema_{ac} * \sum_1^n \frac{nTPi + tpGain}{pACtpi})$$


---

Maximum amount of TPi that can be issued

The global model coverage is:

$$cglb = \frac{nACcb - mocGain}{\sum_1^n \frac{nTPi + tpGain(i)}{pACtpi}}$$

Now, when we issue TP, the amount of TPi increases by  $\Delta TPi$ , and the collateral increases by  $\Delta TPi * pTPac(i)$



$$cglb = \frac{nACcb - mocGain + (\Delta TPi * pTPac(i))}{\Delta TPi * pTPac(i) + \sum_1^n \frac{nTPi + tpGain(i)}{pACtpi}}$$

Coverage should not be less than *ctargema*

$$ctargema = \frac{nACcb - mocGain + (\Delta TPi * pTPac(i))}{\Delta TPi * pTPac(i) + \sum_1^n \frac{nTPi + tpGain(i)}{pACtpi}}$$

$$ctargema * \Delta TPi * pTPac(i) = nACcb - mocGain + (\Delta TPi * pTPac(i))$$

$$\Delta TPi = \frac{nACcb - mocGain - ctargema * \sum_1^n \frac{(nTPi + TPitomint)}{pACtpi}}{(ctargema - 1)} * pACtpi$$

$$\Delta TPi = \frac{nACcb - mocGain - \sum_1^n \frac{(nTPi + TPitomint)}{pACtpi} * ctargema_{tpi}}{(ctargema_{tpi} - 1)} * pACtpi$$

$$\Delta TPi = \frac{nACcb - mocGain - \sum_1^n \frac{(nTPi + TPitomint)}{pACtpi} * ctargema_{tpi}}{(ctargema_{tpi} - 1)} * pACtpi$$

## Joint issuance and redemption of TPs and TCs

About the proportionality of the Tokens

The quantity of AC that is worth a quantity of TP is:

$$nAC = nTP / pACtp$$

The quantity of AC that is worth a quantity of TC is:

$$nAC = nTC / pACtc$$

If we send an amount *qAC* to make a joint issue, the following must be fulfilled:

$$\textcircled{1} \quad qAC = \frac{nTP}{pACtp} + \frac{nTC}{pACtc}$$

$$\textcircled{2} \text{ cobop} = qAC * \frac{pACtp}{nTP}$$

Being *cobop* the coverage to which the joint operation must be carried out

From  $\textcircled{2}$ :

$$\textcircled{3} nTP = qAC * pACtp * \frac{1}{cobop}$$

Applying  $\textcircled{3}$  in  $\textcircled{1}$  :

$$qAC = \frac{qAC * pACtp * \frac{1}{cobop}}{pACtp} + \frac{nTC}{pACtc}$$

$$qAC = \frac{qAC}{cobop} + \frac{nTC}{pACtc}$$

$$qAC - \frac{qAC}{cobop} = \frac{nTC}{pACtc}$$

$$\frac{cobop * qAC - qAC}{cobop} = \frac{nTC}{pACtc}$$

$$\frac{qAC * (cobop - 1)}{cobop} = \frac{nTC}{pACtc}$$

$$nTC = \frac{pACtc * qAC * (cobop - 1)}{cobop}$$

$$nTP = qAC * pACtp * \frac{1}{cobop}$$

$$nTC = qAC * pACtc * \frac{cobop - 1}{cobop}$$

The relationship  $nTC/nTP$  is:

$$\frac{nTC}{nTP} = \frac{qAC * pACtc * \frac{cobop - 1}{cobop}}{qAC * pACtp * \frac{1}{cobop}} = \frac{pACtc * (cobop - 1)}{pACtp}$$

$$nTC = \frac{pACtc * (cobop - 1)}{pACtp} * nTP$$

$$nTP = \frac{pACtp}{pACtc * (cobop - 1)} * nTC$$