

Le RMM, comment ça marche...

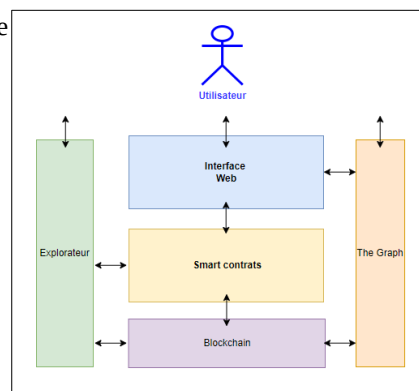
En complément du guide utilisateur, pour les curieux qui souhaiteraient comprendre comment ça marche (« DYOR »), le présent document va (tenter) d'expliquer progressivement les bases théoriques (mode de calcul) de la solution, ainsi que la façon dont cela se traduit dans l'application.

1 - Les composants d'une l'application WEB 3 (comme RMM)

Les Applications distribuées (dAPP) fonctionnent sur une blockchain. Elles sont constituées :

- d'une partie Interface Web (Front-end), que vous accédez via l'url du site et sur laquelle vous vous connectez avec votre wallet,
- de smart contrats, cœur de l'application, qui sont enregistrés et qui s'exécutent sur une blockchain (ici Gnosis),
- et dans le cas présent, d'un service d'accès et d'indexation des données sur la blockchain (The Graph).

L'utilisateur peut aussi accéder aux smart contrats de l'application, directement sans passer par l'interface, au moyen d'un explorateur de blockchain.



Pour la Gnosis Chain, deux explorateurs sont possibles : <https://gnosisscan.io/> et <https://gnosis.blockscout.com/> . (c'est bien utile d'en avoir deux, car ils sont parfois ponctuellement coincés)

Le service d'indexation de données, The Graph est accessible par l'utilisateur, c'est assez complexe et ça ne sera pas détaillé dans ce document (juste un exemple, pour les liquidations).

L'accès à l'application, à partir de l'interface, est détaillé dans le guide utilisateur. Dans le présent document, nous allons accéder à l'application sans passer par l'interface et ainsi analyser ce qui se passe aux niveaux inférieurs...

2 - Les sources de l'application RMM : AAVE

Si vous allez sur l'application AAVE (<https://app.aave.com/>), vous allez retrouver une présentation très semblable au RMM.

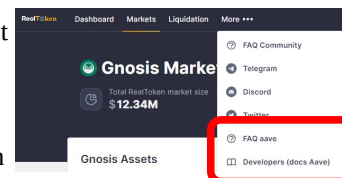
AAVE a déployé, en 2020, une dAPP open source de prêt / emprunt sur Ethereum (en s'inspirant de la solution, que Compound avait lancée en 2018).

La solution d'AAVE, auditée par différentes sociétés (<https://github.com/aave/aave-v3-core/tree/master/audits>) et sans gros incident depuis, a été la base de la solution RMM mise en place par RealT en 2022.

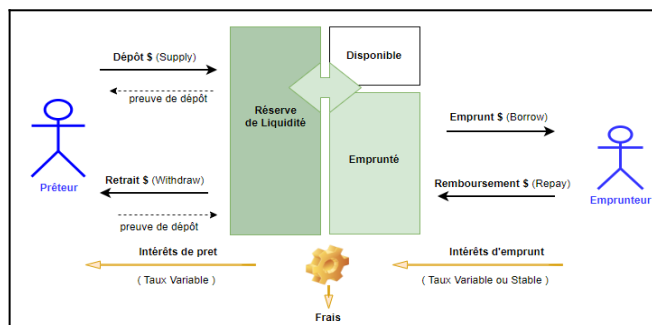
Voilà pourquoi, vous avez accès à des informations techniques d'AAVE, dans l'onglet « More » de l'application RMM.

Le présent document est constitué à partir de ces informations.

RealT a ajouté quelques fonctions complémentaires, notamment : la partie liquidation et le Wrapper (pour la version 3).



3 – Les tokens (et leur cinématique dans RMM)

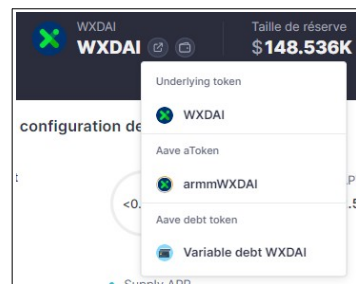


Les dépôts et emprunts d'Actifs (WXDAI, USDC, USDT..) sont tokenisés. Ce qui signifie, par exemple pour du WXDAI, que :

- Lorsque vous déposez des WXDAI : ils vont être transférés et bloqués dans RMM (dans un smart contrat). En échange, RMM va créer des armmWXDAI (preuve de dépôt), qu'il va vous envoyer sur votre wallet (pour la même quantité que le nombre de WXDAI).
- A l'inverse, lorsque vous souhaitez retirer les WXDAI que vous avez déposés : RMM va reprendre des armmWXDAI dans votre wallet, les détruire («bruler») et vous rendre des WXDAI.
- Lors d'un emprunt, RMM va :
 - bloquer une partie de vos armmToken pour assurer la garantie de votre prêt,
 - puis, envoyer sur votre wallet, les WXDAI souhaités ainsi que des tokens de dette.
- Lors du remboursement de votre emprunt : RMM va reprendre dans votre wallet les WXDAI et les tokens de dette, brûler ces derniers et débloquer les armmToken en garantie.

Pour un Actif donné (/token), il y a donc trois type de token : l'armmToken, le token de dette variable et le token de dette stable (**fonction actuellement désactivée**)

L'interface ne donne accès au tokens de dette stable, dans la suite du document, nous verrons comment obtenir cette adresse (via un des smart contrat du RMM).



Les tokens de dépôts et d'emprunts sont liés à leur actif sous-jacent dans un rapport de 1:1 (valeur d'échange), par contre leur quantité évolue en fonction des intérêts associés.

Ainsi :

- la quantité de vos armmToken, augmente automatiquement de la quantité des intérêts de dépôt qui vous sont dues,
- la quantité de vos tokens de dette, augmente automatiquement de la quantité des intérêts d'emprunt qui vous devez.

Les jetons de dépôt sont transférables, ce qui n'est pas le cas des jetons de dette :

- si vous transférez un armmToken, le destinataire pourra soit les garder, soit les convertir dans l'actif sous-jacent,
- par contre vous ne pouvez transférer de jetons de dette (pour que le destinataire, les garde ou les rembourse à votre place), car cette dette est liée à une garantie qui ne serait pas transférée en même temps, ce qui créerait une incohérence, qui rendrait impossible une liquidation, si nécessaire.

4 - Les Smart Contrats

Comme évoqué ci-avant, vous pouvez accéder à l'application sans passer par l'interface : directement auprès des smart contrats. Soit pour contourner une défaillance (ponctuelle) de l'interface, soit pour visualiser des informations que l'interface n'affiche pas, soit pour gagner en rapidité (cas des bots..).

Pour accéder a un smart contrat avec un explorateur, vous allez avoir besoin de son adresse. Chaque smart contrat a une adresse unique, sur la blockchain sur laquelle il est enregistré.

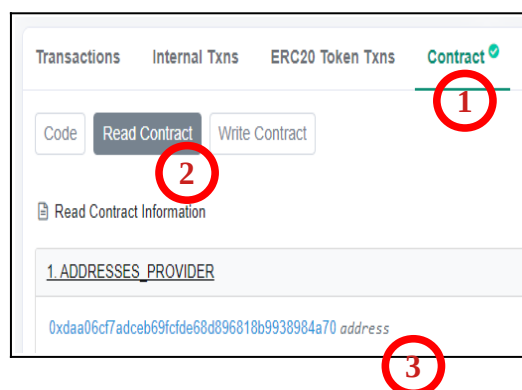
L'ensemble des adresses des smart contrats de l'application RMM, sont regroupées dans un smart contrat nommé «Adresses Provider».

Vous trouverez l'adresse de ce contrat à partir, par exemple, du lien vers le contrat de stratégie d'intérêt :



Ce lien ouvre l'explorateur sur le contrat :

1. Sélectionner la partie « Contract » dans l'explorateur,
2. puis la partie « Read Contract »,
3. et enfin le premier champ d'information : « Addresses Provider ».



4.1 - Smart Contrat : Adresses Provider

RMM v3 : <https://gnosisscan.io/address/0xdaa06cf7adceb69fcfde68d896818b9938984a70>

Principales informations, accessibles en « Read Contract » :

- «4. *getMarketID*» : pour le nom du pool (RMM v3),
- «5. *getPool*» : pour l'adresse du principal smart contrat de l'application. Celui dont vous voyez l'adresse s'afficher dans votre MetaMask, lorsque vous approuver des mouvements sur le RMM
- «7. *getPoolDataProvider*» : pour l'adresse d'un smart contrat qui consolide de nombreuses informations sur l'état du pool de réserves,
- «8. *getPriceOracle*» : pour les parités en USD des Actifs et Propriétés,
- d'autre champs d'information sont disponibles et dédiés à la gestion du pool.

Ce smart contrat, donne les adresses d'autres smart contrats, qui donnent eux même d'autres adresses. On se retrouve dans une arborescence dont l'AdressesProvider est l'origine.

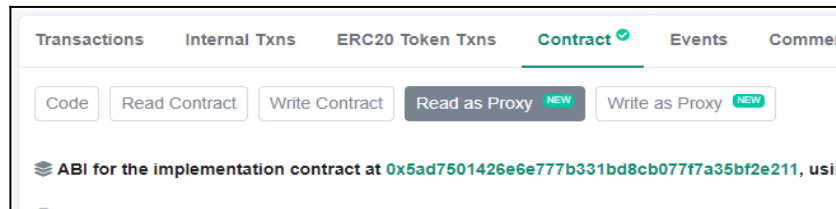
4.2 - Smart contrat : Lending Pool

RMM v3 : <https://gnosisscan.io/address/0xf9b9b496519fca8473fba1af0850b6b8f476bfdb3>

Ce smart contrat est celui qui supporte les principales actions sur le RMM (Dépôt, Emprunt, Remboursement, Retrait, Liquidation..).

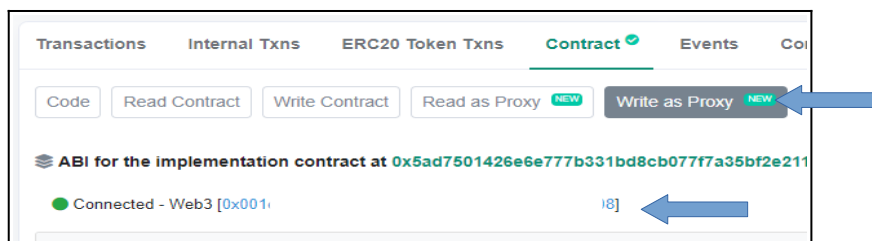
L'accès à ce smart contrat se fait au travers d'un Proxy :

Les smart contrats, une fois enregistrés à une adresse, sont non modifiables. C'est une des forces de la blockchain, mais ce peut être une contrainte lorsqu'on développe et qu'on a besoin de faire des mises à jours. Pour ce faire, on utilise la technique du Proxy, qui permet de modifier la logique du contrat, sans changer son adresse et les valeurs qu'il stocke. Lorsque vous accédez à un smart contrat avec Proxy (comme celui du Pool), la partie «contract» de l'explorateur se présente alors comme suit :



Pour accéder aux données du smart contrat, il faut sélectionner «Read as Proxy». Et, si vous souhaitez lire la logique du contrat qui s'exécute, il faudra aller dans l'adresse «d'implémentation» qui est mentionnée.

L'explorateur permet d'accéder à un smart contrat en lecture et en mise à jour. Dans ce dernier cas, il vous faudra vous connecter avec votre wallet. Cette connexion est l'équivalent de celle que vous faite lorsque vous vous identifiez sur l'application RMM (puisque en accédant directement aux smart contrats, vous ne passez pas par l'interface).



Un exemple d'action, directement sur le smart contrat, sera détaillé dans un prochain chapitre.

En mode lecture, les principaux champs d'information sont :

- «11. *getReserveData*» : pour l'état de chaque réserve du pool. L'information est fournie au format tuple (suite d'infos, séparée par des virgules) : une forme plus lisible est présentée dans le smart contrat suivant,
- «15. *getUserAccountData*» : pour obtenir la position globale d'un wallet sur le RMM (notamment son HF),
- d'autres champs sont disponibles, notamment pour des fonctions qui sont inactives dans le RMM (Flashloan, eMode,...).

```
[ getUserAccountData method Response ]
>> totalCollateralBase uint256 : 686237263852
>> totalDebtBase uint256 : 340039244112
>> availableBorrowsBase uint256 : 3079387814
>> currentLiquidationThreshold uint256 : 7000
>> ltv uint256 : 5000
>> healthFactor uint256 : 141267836878768563
```

4.3 - Smart contrat : *Data Provider*

RMM v3 : <https://gnosisscan.io/address/0x11b45acc19656c6c52f93d8034912083ac7dd756>

Ce smart contrat ne fait que donner des informations (aucune mise à jour de donnée n'est possible).

De nombreuses informations sont disponibles (19), sur le fonctionnement du pool :

- «4. *getAllReservesTokens*» : pour la liste des réserves : soit pour le RMM v3 : WXDAI, USDC et RTW (pour les RealTokens wrappés).

Les adresses, sont celles demandées dans d'autres champs, qui détaillent les caractéristiques de chaque réserve (« asset»),

- «3. *getAllATokens*» : pour les adresses des aToken (preuve de dépôt) de chaque réserve,

```
[ getAllReservesTokens method Response ]
>> tuple[] :
[[WXDAI,0xe91D153E0b41518A2Ce8Dd3D7944Fa863463a97d]
[USDC,0xDDAfb505ad214D7b80b1f830fcC89B60fb7A83]
[RTW-USD-01,0xd3DFf217818b4F33eB38a243158FBd2BBB029D3]]

[ getAllReservesTokens method Response ]
>> tuple[] :
[[ammv3WXDAI,0x0cA4f554Dd9Da6217d62D8d2816c82bba4157b]
[ammv3USDC,0xeD56F76E9cB6A64b821e9c016eAFbd3db5436D1]
[ammv3RTW-USD-01,0xf3220C8f66AEB86fC2A82502977EAb4Bfd2f647]]
```

- «2. *getATokenTotalSupply*» : pour la quantité déposée dans une réserve donnée :
L'adresse « asset » est celle figurant au point 4 ci-dessus (exemple avec le XDAI),
Les quantités de XDAI sont exprimées avec une précision de 18 décimales (cf info dans le point suivant). Pour obtenir la valeur déposée, il faut donc diviser la valeur affichée par 10 puissance 18 (10^18), Ce qui donne ici un montant de 148 252,68\$.

```
2. getATokenTotalSupply

asset (address)
0xe91D153E0b41518A2Ce8Dd3D7944Fa863463a97d

Query

uint256

[ getATokenTotalSupply method Response ]
>> uint256 : 148252687490809951297990
```

- «12. *getReserveConfigurationData*» : pour les caractéristiques de chaque réserve : Décimale, LTV, Seuil de liquidation, Pénalité de liquidation, Reserve factor, Collatéralisable, Empruntable, Empruntable à taux Stable, Réserve active, réserve figée.

```
[ getReserveConfigurationData method Response ]
>> decimals uint256 : 18
>> ltv uint256 : 7500
>> liquidationThreshold uint256 : 8000
>> liquidationBonus uint256 : 10500
>> reserveFactor uint256 : 1000
>> usageAsCollateralEnabled bool : true
>> borrowingEnabled bool : true
>> stableBorrowRateEnabled bool : false
>> isActive bool : true
>> isFrozen bool : false
```

- «13. *getReserveData*» : pour les valeurs instantanées de chaque réserve : valeur déposée, empruntée, taux..

```
[ getReserveData method Response ]
>> unbacked uint256 : 0
>> accruedToTreasuryScaled uint256 : 502203223723313924
>> totalAToken uint256 : 148252715291654080944275
>> totalStableDebt uint256 : 0
>> totalVariableDebt uint256 : 68847445706610172556844
>> liquidityRate uint256 : 18196139818996685085386368
>> variableBorrowRate uint256 : 43536550519597760499464403
>> stableBorrowRate uint256 : 87902436701306517366630960
>> averageStableBorrowRate uint256 : 0
>> liquidityIndex uint256 : 1000030488367151790368430553
>> variableBorrowIndex uint256 : 1000083748095749800680248570
>> lastUpdateTimestamp uint40 : 1707238135
```

- «15. *getReserveTokenAddresses*» : pour les adresses des tokens de dépôts et d'emprunt de chaque réserve,

```
[ getReserveTokenAddresses method Response ]
>> aTokenAddress address : 0x0cA4f554Dd9Da6217d62D8d2816c82bba4157b
>> stableDebtTokenAddress address : 0x8ACD88D494cF56F542234f8924F06024b5795B5
>> variableDebtTokenAddress address : 0x9908801dF7902675C3FEDD6F6a0294D18D5d5d34
```

- «5. *getDebtCeiling*» : pour la limite d'emprunt (pour les réserves en mode isolé),
- «8 *getInterstRateStrategyAddress*» : pour l'adresse du smart contrat qui fixe les caractéristique du modèle d'emprunt (ce smart contrat est détaillé dans le chapitre suivant),
- «19 *getUserReserveData*» : pour détailler la position d'un wallet sur une réserve.

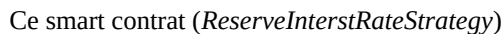
5 - Fonctionnement du RMM

Pour le flux du capital : tout commence par des dépôts, pour alimenter la réserve de liquidité. Sans argent dans la réserve, il ne peut y avoir de prêts !.

Mais à l'inverse, pour le flux des intérêts : tout commence par des emprunts. Sans intérêts payés par les emprunteurs, il ne peut y avoir d'intérêts payés aux prêteurs.

Commençons donc, par le mode de calcul des intérêts des emprunteurs, dont le taux varie en fonction de l'usage de la réserve.

Pour chacun des Actifs du RMM, il y a un smart contrat qui définit les paramètres du modèle de taux (cf lien d'accès)



fixe de nombreux paramètres (14), dont les principaux sont :

- La valeur indiquée dans le smart contrat :

[illegible]

- ```
8. getBaseVariableBorrowRate
0 uint256
```

[illegible]

```
14. getVariableRateSlope2
```

```
9_getMaxVariableBorrowRate
```

```
7. getBaseStableBorrowRate
```

- ```
11. getStableRateSlope1  
5000000000000000000000000000 uint256
```

```
12. getStableRateSlope2  
7500000000000000000000000000 ui
```

- en v2, l'ajout dans certains cas d'un taux complémentaire (moyenne des taux du marché),
- en v3,
 - l'ajout d'un **taux complémentaire** (8 % dans l'exemple),
 - lorsque la proportion d'emprunt à taux stable dépasse un **seuil** par rapport à l'ensemble des emprunts (20 % dans l'exemple)

```
10. getStableRateExcessOffset  
80000000000000000000000000000000 uint2
```

```
4. OPTIMAL_STABLE_TO_TOTAL_DEBT_RATIO  
2000000000000000000000000000 uint256
```

The interest rate R_t follows the model:

$$\begin{aligned}
 \text{if } U \leq U_{\text{optimal}} : \quad R_t &= R_0 + \frac{U_t}{U_{\text{optimal}}} R_{\text{slope1}} \\
 \text{if } U > U_{\text{optimal}} : \quad R_t &= R_0 + R_{\text{slope1}} + \frac{U_t - U_{\text{optimal}}}{1 - U_{\text{optimal}}} R_{\text{slope2}}
 \end{aligned}$$

Dans la quelle :

- U_t : est le taux d'utilisation de la réserve,
- U_{optimal} : le taux Optimal d'utilisation, mentionné ci-avant,
- R_0 : le taux de base,
- R_{slope1} : la première pente,
- R_{slope2} : la seconde pente,
- R_t est le taux exprimé en APR (cad sans composition des intérêts).

Sauf que pour le taux stable, il faut ajouter une correction supplémentaire fonction de la proportion d'emprunt stable utilisé. La formule pour cette correction, est la suivante :

$$\begin{aligned}
base_s &= slope_{v,1} + offset_{base}, & E_{util} &= 10^{27} - O_{util} & (1) \\
ratio &= \frac{debt_{stable}}{debt_{stable} + debt_{variable}} & (2) \\
rate_s &= \begin{cases} base_s + slope_{s,1} + slope_{s,2} * \frac{(util - O_{util})}{E_{util}}, & \text{if } util > O_{util} \\ base_s + slope_{s,1} * \frac{util}{O_{util}}, & \text{otherwise} \end{cases} & (3) \\
rate_s &= \begin{cases} rate_s + offset_{excess} * \frac{ratio - O_{ratio}}{10^{27} - O_{ratio}}, & \text{if } ratio > O_{ratio} \\ rate_s, & \text{otherwise} \end{cases} & (4)
\end{aligned}$$

https://github.com/aave/aave-v3-core/blob/master/techpaper/Aave_V3_Technical_Paper.pdf

- (1) bases : correspond au taux de base stable (point 7 du smart contrat de stratégie d'intérêt),
- (2) ratio : est la proportion d'emprunt stable par rapport à l'ensemble des emprunts (stable et variable),
- Oratio : est le seuil d'emprunt à taux stable, évoqué ci-avant (point 4 du smart contrat de stratégie d'intérêt),
- (4) rates : est le taux d'emprunt stable, qui est calculé différemment suivant le ratio :

- si les emprunts à taux stable sont inférieurs au seuil ($\text{ratio} < \text{Oratio}$), le calcul se fait comme expliqué précédemment mais avec des paramètres pour le taux stable (3),
- si les emprunts à taux stable dépassent le seuil ($\text{ratio} > \text{Oratio}$), un taux complémentaire (offsetexcess) est ajouté au taux précédemment calculé au prorata du ratio par rapport à 100 % ($\text{ratio} - \text{Oratio}$) / ($1 - \text{Oratio}$)

Un exemple sera détaillé, dans un prochain chapitre, pour être plus parlant ..

5.1.1 - Taux d'utilisation de la réserve

Le taux d'utilisation, d'une réserve, est le rapport entre l'ensemble des montants empruntés et l'ensemble des montants déposés.

A chaque fois qu'il y a un dépôt ou un emprunt, il y a création d'un token correspondant.

Comme vu précédemment, dans le point 15 du smart contrat *Data Provider* figurent les adresses des tokens de dépôt et d'emprunt.

```
[ getReserveTokensAddresses method Response ]
>> aTokenAddress address : 0x0cA4f5554Dd9Da6217d62D8df2816c82bba4157b
>> stableDebtTokenAddress address : 0x8ACD88D494cFA56F542234f8924F06024b5795B5
>> variableDebtTokenAddress address : 0x9908801df7902675C3FEDD6F6a0294D18D5d5d34
```

Il suffit donc d'entrer ces adresses dans un explorateur, cliquer sur le nom du «Tracker» et voir apparaître son «Supply»

Dans l'exemple, on trouve 148 252,61 WXDAI.

Faites la même chose, pour les deux autres tokens de dette (en variable et en stable),

Le prêt stable étant désactivé il n'y a pas de tracker

Le taux d'utilisation de la réserve (WXDAI dans l'exemple) est alors égale à la somme des tokens de dette (variable+Stable) divisé par le supply du token de dépôt. (dans l'exemple : $(68\,847,52 + 0) / 148\,252,61 = 46,44\%$)

Nous retrouvons ces chiffres dans l'interface :

Borrow info Total borrowed 68,845.25 <small>\$68.847K</small> APY, variable ⓘ 4.45 %	Supply info Total supplied ⓘ 148.25K of 2.00B <small>\$148.255K of \$2.000B</small> APY 1.84 %	Interest rate model Utilization Rate 46.44 %
--	--	--

On peut aussi retrouver la quantité des 3 tokens, avec la query 13 sur le smart contrat *Data Provider*.

Cette query donne aussi les taux d'emprunt associés.

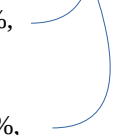

```
[ getReserveData method Response ]
>> unbacked uint256: 0
>> accruedToTreasuryScaled uint256: 502203223723313924
>> totalAToken uint256: 148252715291654080944275
>> totalStableDebt uint256: 0
>> totalVariableDebt uint256: 68847445706610172556844
>> liquidityRate uint256: 18196139818996685085386368
>> variableBorrowRate uint256: 43536550519597760499464403
>> stableBorrowRate uint256: 87902436701306517366630960
>> averageStableBorrowRate uint256: 0
>> liquidityIndex uint256: 100003048836751790368430553
>> variableBorrowIndex uint256: 1000083748095749800680248570
>> lastUpdateTimestamp uint40: 1707238135
```

5.1.2 - Exemple de calcul des taux d'emprunt

Validons notre compréhension des calculs, à partir de la formule théorique

$$U \leq U_{optimal} : R_t = R_0 + \frac{U_t}{U_{optimal}} R_{slope1}$$

et des valeurs précédemment citées :

- Le taux variable (en APR) est égale :
 - Taux de base Variable + Taux d'utilisation / Taux Optimal x Pente 1 Variable
 - soit : 0 % + 46,44 % / 80 % x 7,5 % = 4,35 %, 
- Le taux stable est égale :
 - Taux de base Stable + Taux d'utilisation / Taux Optimal x Pente 1 Stable
 - soit : 8,5 % + 46,44 % / 80 % x 0,5 % = 8,79 %, 
 - Le ratio d'emprunt Stable par rapport à l'ensemble des emprunts étant 0 (taux stable désactivé) Nous sommes donc au dessous du seuil de 20 % (point 4 du smart contrat *Data provider*),
Aucun taux complémentaire est à ajouter.

Ces chiffres correspondes à ceux de la query 13 évoqué ci-dessus, mais pas exactement a celui affiché pour le taux variable ... car ce dernier est affiché en APY.

Borrow info	Total borrowed	APY, variable 
	68,845.25	4.45 %
	\$ 68.847K	

5.1.3 - APR vs APY

L'APY est le taux annuel composé des intérêts.

Les intérêts sont calculés sur la base de la seconde et viennent s'ajouter au capital. La nouvelle valeur (capital + intérêt) devient la nouvelle base calcul d'intérêt. Il y a donc des intérêts sur des intérêts : c'est le mécanisme de composition des intérêts.

Pour passer d'un taux APR à un taux APY, il faut appliquer la formule suivante :

APR -> APY

To convert the APR to APY compounded per second the formula is:

$$APY = (1 + (APR / secondsPerYear))^{secondsPerYear} - 1$$

<https://docs.aave.com/developers/v/2.0/guides/apy-and-apr>

Le nombre de secondes par an étant égale à 60 x 60 x 24 x 365 = 31 536 000.

A partir de l'exemple précédent :

- l'APY correspondant à un APR de 4,3536 % est
- $(1 + 4,3536 \% / 31\,536\,000)^{31\,536\,000} - 1$, soit 4,45 %
- ce qui correspond à l'APY Stable qui est affiché dans l'interface.

5.2 - Calcul du taux de dépôt

Les intérêts collectés auprès des emprunteurs sont intégralement distribués en intérêt aux déposants (hors frais).

Donc : Somme empruntée * Taux d'emprunt = Somme déposée * Taux de dépôt,

Ou : Taux de dépôt = Taux d'emprunt * Somme empruntée / Somme déposée,

Taux de dépôt = Taux d'emprunt * Taux d'utilisation de la réserve.

Pour tenir compte des deux natures d'emprunt (à taux variable et taux stable) : un taux moyen pondéré est calculé.

Soit Taux moyen d'emprunt = Taux variable * Proportion d'emprunt variable + Taux Stable * Proportion d'emprunt Stable

d'où : Taux de dépôt = (Taux variable * Proportion d'emprunt variable + Taux Stable * Proportion d'emprunt Stable) * Taux d'utilisation de la réserve.

Pour financer son développement, AAVE et donc le RMM (/RealT) collecte des frais au travers du *Reserve Factor*.

Le *Reserve factor*, correspond au pourcentage des emprunts collectés qui sont affectés à ces frais.

Nous avons donc : Taux de dépôt = (Taux variable * Proportion d'emprunt variable + Taux Stable * Proportion d'emprunt Stable) * Taux d'utilisation de la réserve * (1 – Reserve Factor)

La formule de calcul du taux de dépôts (S_t), donnée dans la documentation d'AAVE correspond exactement à celle détaillée ci-avant :

<https://docs.aave.com/risk/liquidity-risk/borrow-interest-rate>

$$S_t = U_t(SB_tS_t + VB_tV_t)(1 - R_t)$$

- U_t , the utilisation ratio
- SB_t , the share of stable borrows
- S_t , the average stable rate
- VB_t , the share of variable borrows
- V_t , the variable rate
- R_t , the reserve factor

La valeur du *Reserve Factor*, fixée pour une réserve, est visible :

- soit sur l'interface :

- soit dans la query 12 du smart contrat *Data Provider* :

```
[ getReserveConfigurationData(address) method Response ]
>> decimals uint256: 18
>> ltv uint256: 7500
>> liquidationThreshold uint256: 8000
>> liquidationBonus uint256: 10500
>> reserveFactor uint256: 1000 ←
>> usageAsCollateralEnabled bool: true
>> borrowingEnabled bool: true
>> stableBorrowRateEnabled bool: true
>> isActive bool: true
>> isFrozen bool: false
```

Le montant accumulé par le *Reserve Factor*, est visible en utilisant le lien du contrat Collecteur qui est dans l'interface :

5.3 - Fonctionnement des indexes

Vos positions vis à vis du RMM, sont stockées dans les tokens de dépôts et de dette, de chaque réserve.

Comme évoqué précédemment, la quantité de ces tokens que vous possédez, évolue automatiquement et continûment (pour tenir compte des intérêts).

L'état d'une réserve est mise à jour à chaque interaction d'un utilisateur avec celle-ci (lors d'un dépôt, retrait, emprunt, remboursement ou liquidation).

Si à chaque interaction, les soldes des tokens de dépôt et de dette devaient être mis à jour, dans tous les wallets; cela coûterait des frais extrêmement conséquents. Et pourtant, votre solde évolue constamment, sans aucun frais !

La solution réside dans l'astucieux mécanisme d'indexes....

Prenons par exemple, le cas du token de dépôt armmWXDAI :

- Lorsque vous déposez des WXDAI, le smart contrat armmWXDAI n'enregistre pas le montant de WXDAI déposé par votre wallet, mais ce montant divisé par l'index de dépôt, propre à la réserve WXDAI (montant nommé « *ScaledBalance* »),
- Cet index correspond à l'ensemble des intérêts de dépôt cumulés, depuis la création de la réserve WXDAI,
- L'index (quantité d'intérêt) va évoluer à chaque interaction avec la réserve : il va augmenter en fonction du taux de dépôt multiplié par le temps passé depuis la dernière mise à jour,
- Lorsque vous interrogerez à nouveau votre montant déposé, le smart contrat armmWXDAI va faire le produit de votre *ScaledBalance* avec l'index du moment et vous afficher le résultat. Comme l'index aura augmenté, vous verrez votre solde augmenter, sans qu'aucune mise à jour n'ait été faite de votre compte, donc sans frais !

Deux Indexes sont mis en place, un pour les dépôts et un pour les dettes.

Si vous voulez en savoir plus :

- dans le support AAVE : <https://docs.aave.com/developers/guides/rates-guide#how-is-yield-accrued>
- ou, dans le smart contrat correspondant : <https://github.com/aave/aave-v3-core/blob/6070e82d962d9b12835c88e68210d0e63f08d035/contracts/protocol/tokenization/AToken.sol#L128>

Après cette plongée, dans les rouages d'AAVE/RMM, pour terminer remontons à la surface afin de voir quelques usages des compétences acquises.

6 - Informations complémentaires, à partir de l'explorateur

6.1 – A partir du token de dépôt

A partir du token de dépôt armmxDai, sur RMM v2 : 0x7349c9eaa538e118725a6130e0f8341509b9f8a0

Vous trouvez, la liste des prêteurs :

<https://blockscout.com/xdai/mainnet/token/0x7349C9eaA538e118725a6130e0f8341509b9f8A0/token-holders>

Vous constaterez que, le premier apporteur de liquidité a fourni 1,5 M\$ sur les 3,2 M\$ de la réserve !..
(au moment où ces lignes ont été écrites...)

Token Holders	
0x00	1,532,890.198 armmWXDAI 46.2957%
0x00	357,096.74 armmWXDAI 10.7849%
0x25b248911b9551f48cd3f0f01699ef7653d78433	186,016.219 armmWXDAI 5.6180%

Pour le RMM v3, comme il y a deux réserves de liquidité :

- pour l'USDC : <https://gnosisscan.io/token/0xeD56F76E9cBC6A64b821e9c016eAfbd3db5436D1#balances>
- pour le WXDAI : <https://gnosisscan.io/token/0x0cA4f5554Dd9Da6217d62D8df2816c82bba4157b#balances>

6.1.2 – Historique de vos dépôts et retraits

En indiquant l'adresse de votre wallet



6.2 – A partir des tokens d'emprunt

6.2.1 – Liste des plus gros emprunteurs

A partir du token de dette variable en wXDAI sur RMM v2 : 0x6a7ced66902d07066ad08c81179d17d0fbe36829
Vous trouvez la liste emprunteurs :

<https://blockscout.com/xdai/mainnet/token/0x6a7CeD66902D07066Ad08c81179d17d0fbE36829/token-holders>

Soit plus de 77 K\$ pour le premier !

Token Transfers	Token Holders
Token Holders	
0x7A3c35F91c472E83Dc0Ae5c81B87cd983345488B	77,477.005 variableDebtmmWXDAI 3.1433%
0x166ed9f7A56053c7c4E77CB0C91a9E46bbC5e8b0	52,571.373 variableDebtmmWXDAI 2.1328%

Pour le RMM v3, comme il y a deux réserves de liquidité :

- pour l'USDC : <https://gnosisscan.io/token/0x69c731aE5f5356a779f44C355aBB685d84e5E9e6#balances>
- pour le WXDAI : <https://gnosisscan.io/token/0x9908801dF7902675C3FEDD6Fea0294D18D5d5d34#balances>

6.2.1 – Historique de vos emprunts et remboursements

En indiquant l'adresse de votre wallet



6.3 - Position RMM d'un portefeuille :

Par exemple, quand ce portefeuille s'approche de la liquidation !

A partir du smart contrat de gestion du pool de liquidités (du RMM v2) :

<https://blockscout.com/xdai/mainnet/address/0x5B8D36De471880Ee21936f328AAB2383a280CB2A/read-proxy#address-tabs>

en mettant l'adresse du wallet, sur le point 11,

par ex avec l'adresse la plus proche actuellement de la liquidation

11. getUserAccountData →

↳ totalCollateralETH(uint256) totalDebtETH(uint256)

Vous retrouvez les valeurs suivantes :

- Total en collatéral = 7 917,80 \$ (valeur / 10^8)
- Total de l'emprunt = 5 467,32\$ (valeur / 10^8)
- Health Factor = 1,01374 (valeur / 10^{18})
valeur que l'on retrouve, par le calcul avec les deux premiers montants.

totalCollateralETH (uint256)	: 791779834739
totalDebtETH (uint256)	: 546732381529
availableBorrowETH (uint256)	: 0
currentLiquidationThreshold (uint256)	: 7000
Ltv (uint256)	: 5000
healthFactor (uint256)	: 1013742560422317817

Ces informations vous permettront, avec un petit calcul, de savoir quand ce portefeuille sera en liquidation (si il ne fait rien)...

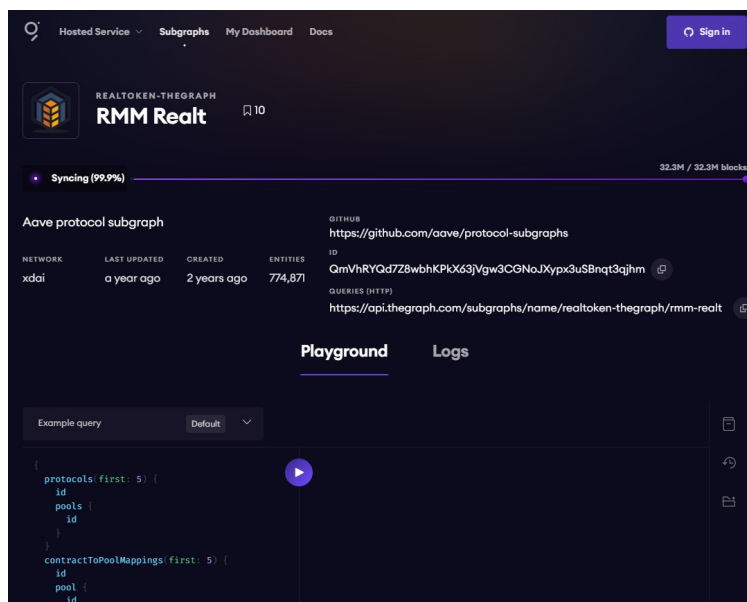
<https://community-realt.gitbook.io/tuto-community/aide-en-francais/defi-realt/rmm/evaluation-du-delaix-avant-liquidation>,

7 - Informations complémentaires, à partir de The Graph

The Graph est un protocole (/service) pour simplifier l'accès aux données des blockchains. RealT a mis en place un Subgraph, spécifique pour les données du RMM. Ces données sont accessibles soit via une API, soit via un explorateur (pour RMM v2) :

<https://thegraph.com/hosted-service/subgraph/realtoken-thegraph/rmm-realt>

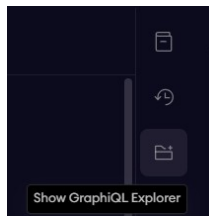
L'interrogation des données se fait au travers de requêtes en GraphQL.



Un exemple de requête apparaît en bas à gauche de l'explorer. Il suffit de cliquer sur la flèche au centre pour voir apparaître le résultat sur la droite :



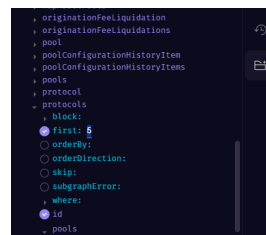
A droite, le dernier icône permet d'ouvrir l'explorateur des données



Vous voyez alors apparaître l'ensemble des données interrogeables et celles qui le sont (cochées) déjà dans la requête existante.

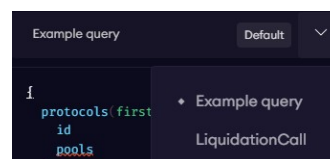
La requête se constitue donc au fur et à mesure que vous cochez ou décochez les données souhaitées.

Toute la difficulté étant de savoir à quoi correspondent chacune des données pour le RMM (il n'existe hélas pas de dictionnaire ...).



Dans les query par défaut, il y en a une qui permet de voir l'ensemble des liquidations :

Si vous la lancez, vous obtiendrez de nombreuses informations, mais hélas en format JSON (plutôt dédié à des programmes, qu'à des humains...)

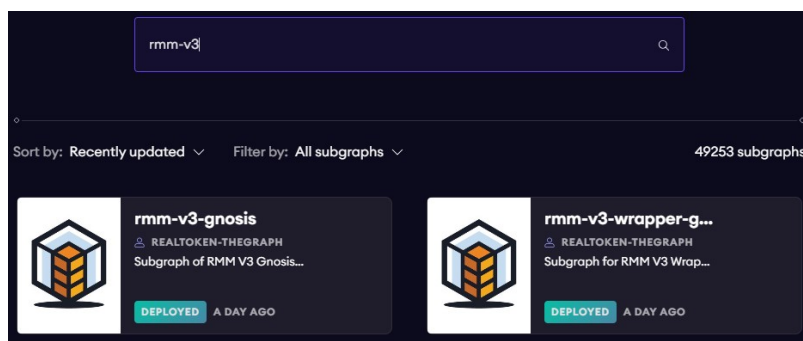


Pour une lecture simplifiée de ces données, il faut : soit faire un développement, soit se servir d'un programme qui sait lire ce format : par ex Google Sheet (avec l'extension API Connector) :

<https://docs.google.com/spreadsheets/d/1ksmZyjVRWi4ART3yIWczOtFE-jucIxcRCULDRYkq1Bs>

RealToken Liquidé	Wallet du Liquidateur	Wallet Liquidé	Date Liquidation	Montant WXDAI	Nb RealToken
RealToken S 19000 Fenton St Detroit MI	0xf0633940d64515379bbe743515fc1413834d393	0x49352b0e106f451d20eb981fb7e186a40245ef1c	27/02/2023	182.05	4.00
RealToken S 13835 La Salle Blvd Detroit MI	0xd58b712722371aa92c29272094c3a65482c0429	0x835c3e4296f60cdfa139ed6702e6cd5869bed999	28/03/2023	25.30	0.55
RealToken S 1389 Bird Ave Birmingham MI	0x46d0fb00d66d66e1a2fec6aeb670d53951c361c4	0x835c3e4296f60cdfa139ed6702e6cd5869bed999	10/05/2023	1.15	0.03
RealToken S 19750 Marx St Highland Park MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x40def0aa74bcb9107718ee767659004003603ebb	12/05/2023	0.00	0.00
RealToken S 13430 Tacoma St Detroit MI	0x46d0fb00d66d66e1a2fec6aeb670d53951c361c4	0x89902a85a218a69dd8dded2b5e9b980913c0b6	19/05/2023	44.49	0.97
RealToken S 19041 Lenore Ave Detroit MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x4da0ca5cbcf06d41c1935f3a7cfd3a5608017d96	19/06/2023	182.78	4.00
RealToken S 3747 Scovel Pl Detroit MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x835c3e4296f60cdfa139ed6702e6cd5869bed999	24/06/2023	12.34	0.27
RealToken S 10411-10421 Cadieux Rd Detroit MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x263c039fe227d3529a99dbbe6ae0b098c894ba6	09/07/2023	75.91	1.66
RealToken S 1610 E State Fair Ave Detroit MI	0x46d0fb00d66d66e1a2fec6aeb670d53951c361c4	0xaa59df1422c9010d5d9772567e56004270d711a1	31/07/2023	254.10	5.56
RealToken S 14918 Joy Rd Detroit MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x6f61dba437d5dc71963817c37e0bef392179fb2d	06/08/2023	226.22	5.06
RealToken S 618 E 79th St Chicago IL	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x5600a0b2db9a362aa3b1a0ea92db61550744d39	12/08/2023	50.23	1.09
RealToken S 19751 Marx St Highland Park MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x71d5919e7cf762a6b0913e008f4315965e9c01a1	08/09/2023	132.98	2.88
RealToken S 14918 Joy Rd Detroit MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0xd21d3f05cd1e7bb229473387f4099b89e33b09	15/09/2023	4923.57	109.55
RealToken S 10617 Hathaway Ave Cleveland OH	0xa22dc341c8d53ab1dff6e6228e779832a449bf	0x39dbefbae84c6fa5b4aa223e706315a2db636920	01/10/2023	4.01	0.08
RealToken S 10411-10421 Cadieux Rd Detroit MI	0xa22dc341c8d53ab1dff6e6228e779832a449bf	0x263c039fe227d3529a99dbbe6ae0b098c894ba6	04/11/2023	39.00	0.85
RealToken S 618 E 79th St Chicago IL	0xa22dc341c8d53ab1dff6e6228e779832a449bf	0x1ceb00630377aa16074bcb4de0760631fac53480	21/11/2023	38.40	0.83
RealToken S 18980 Fenton St Detroit MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x433dd11e3f2c6c6619c729f078c1ed7042ec5f1e	05/12/2023	38.07	0.83
RealToken S 3280 W Boston Blvd Detroit MI	0xa22dc341c8d53ab1dff6e6228e779832a449bf	0x5846fc589dcfd5e56838d031d57b61bfc63c562	15/12/2023	26.00	0.56
RealToken S 14918 Joy Rd Detroit MI	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0xd21d3f05cd1e7bb229473387f4099b89e33b09	19/01/2024	2 415.29	53.74

Dans les « Hosted Service » de The graph (<https://thegraph.com/hosted-service>) , deux Subgraph sont disponibles pour RMM v3 :



8 – Environnements de test

Pour réaliser des tests, sans risques voire sans frais, plusieurs environnements sont disponibles :

- celui mis en place par RealT, pour la migration au RMM v3,
- celui d'AAVE v3.

L'application RMM est disponible en PréProduction sur l'adresse suivante : <https://staging-rmm.realtoken.network/>

L'application sur le staging peut être utilisée :

- sur Gnosis Chain : en utilisant vos propres tokens, et en payant les frais.
- sur le Testnet Sepolia : en utilisant des tokens fournis par RealT et sans frais.

Procédure pour obtenir des tokens sur Sepolia et être whitelisted :

https://docs.google.com/document/d/1spk1WrqdKHILwFRi_yRDf_fbFz3ZYCuZh5g_ni0laQA/edit#heading=h.w11hkzqboxwj

Vous pouvez tester AAVE (application semblable, mais sans les RealTokens), en allant sur l'application <https://app.aave.com/> et en basculant le commutateur à droite sur réseau de test (l'icone Testnet apparaît à en haut gauche).



Vous pouvez obtenir des tokens de test via l'onglet « Faucet ». Des ETH seront nécessaires (cf procédure RealT, pour en obtenir).

9 – Fonctions RMM v3 non activés

Lors de la phase de test, des fonctions étaient disponibles qui n'ont pas été déployées en production.

9.1 – Mode isolé

Les Actifs considérés comme « à risque », peuvent être inclus dans la version 3, grâce à la fonction d'isolation de ce type d'Actif. Un Actif ainsi répertorié à une capacité d'emprunt limité. Il ne peut être mis en garantie que de façon isolée (cad seul, sans autre Actif) et ne donne alors droit, qu'à un emprunt d'Actifs sélectionnés dans un montant limité.

Dans l'exemple suivant, l'Actif USDT ne peut être collatéralisable qu'en mode isolé (comme indiqué dans la liste des Actifs) :

Actifs à déposer					Cacher —
Actifs	Solde du portefeuille	APY	Peut être collatéral		
WXDAIRealT	20.1	< 0.01 %	✓	Dépot	Détails
USDCRealT	11	< 0.01 %	✓	Dépot	Détails
USDTRealT	0.1	< 0.01 %	Isolé	Dépot	Détails

Pour que le mode isolé soit actif (bouton vert) et qu'un dépôt d'USDT puisse servir de collatéral à un emprunt :

Vos actifs déposés

Solde \$ 29.998 | APY < 0.01 % | Collatérale \$ 14.999

Actif	Solde	APY	Collatérale	
USDTRealT	15 \$ 14.999	< 0.01 %	Isolé 1	Retirer
WXDAIRealT	10 \$ 10	< 0.01 %	2	Retirer
USDCRealT	5 \$ 5	< 0.01 %		Retirer

Vos propriétés déposées

Solde \$ 0 **3**

Vos emprunts

Aucun emprunt pour l'instant

Actifs à emprunter **4**

Le pouvoir d'emprunt et les actifs sont limités en raison du mode d'isolement.

Actif	Disponible	APY, variable
USDCRealT	11.14	< 0.01 %
USDTRealT	11.14	< 0.01 %
WXDAIRealT	11.14	< 0.01 %

- il doit être le seul en collatéral ,
- d'autres Actifs peuvent être en dépôt (pour toucher des intérêts), mais ils ne doivent pas être en collatérale,
- aucune propriété ne doit être déposée (puisque'elles sont automatiquement en collatéral),
- les Actifs empruntables et leur quantité sont limités.

Dès qu'un autre Actif que l'USDT (ou une propriété) est en collatéral (bouton vert) :

Vos actifs déposés

Solde \$ 29.998 | APY < 0.01 % | Collatérale \$ 10

Actif	Solde	APY	Collatérale	
USDTRealT	15 \$ 14.999	< 0.01 %	Isolé 1	Retirer
WXDAIRealT	10 \$ 10	< 0.01 %		Retirer
USDCRealT	5 \$ 5	< 0.01 %		Retirer

Vos propriétés déposées

Solde \$ 0

Vos emprunts

Aucun emprunt pour l'instant

Actifs à emprunter **2**

Actif	Disponible	APY, variable
USDCRealT	7.43	< 0.01 %
USDTRealT	7.43	< 0.01 %
WXDAIRealT	7.43	< 0.01 %

- le mode isolé est inactivable.
L'USDT en dépôt rapporte des intérêts de dépôt, mais ne peut servir de collatéral à un emprunt,
- les Actifs empruntables ne sont plus limités par le mode isolé.

Le suivi de la dette de l'Actif isolé est visible dans sa page détail :

Usage de collatéral

L'actif ne peut être utilisé comme garantie qu'en mode isolé.
 In isolation mode you cannot supply other assets as collateral for borrowing. Assets used as collateral in Isolation mode can only be borrowed to a specific debt ceiling. [Learn more](#)

Max LTV ⓘ 75 % Seuil de liquidation ⓘ 80 % Pénalité de liquidation ⓘ 5 %

Plafond de la dette isolée ⓘ \$0 de \$5.00M

En résumé : Si vous avez déposé des RealTokens, vous ne pouvez activer le mode isolé et l'USDT que vous déposerez ne générera que des intérêts de dépôt et pas de capacité d'emprunt supplémentaire.

9.2 – Emprunt à taux stable

En Novembre 2023, une vulnérabilité concernant les emprunts en mode stable sur AAVE a été détecté : l'emprunt en mode stable a donc été désactivé.

Comment il pouvait être utilisé :

A partir de l'onglet « Tableau de bord », dans la partie « Actifs à emprunter » vous pouvez choisir le type de taux qui sera appliqué à votre emprunt :

- Un taux variable : plus faible, mais variable dans le temps suivant la liquidité de la réserve,
- ou un taux Stable : qui offre plus de prévisibilité, ce qui a un coût. Il est plus stable, mais pas fixe pour autant. Le différentiel avec le taux variable augmente progressivement en fonction de la proportion des emprunts à taux stable par rapport au total des emprunts de l'Actif.

Emprunter USDCRealT ✕

Taux APY d'emprunt ⓘ

Variable < 0.01 % Stable 7.82 %

10 \$10.001 Disponible 57.06 MAX

Aperçu des transactions

Facteur de santé ∞ → 6.14 Liquidation à < 1.0

Emprunter

Lorsque vous aurez emprunté, il est possible de changer le type de taux à tout moment :

Vos emprunts Cacher —

Solde \$10 APY < 0.01 % ⓘ Puissance d'emprunt utilisée 17.35 % ⓘ

Actif	Dette	APY	APY type ⓘ
USDCRealT	10 \$10	< 0.01 %	VARIABLE

Actifs à emprunter

Actif : Disponible ⓘ : APY, variable ⓘ :

☒ APY, variable < 0.01 %
☐ APY, stable 7.78 %

VOIR LES GRAPHIQUES

Notre exploration prend fin, en espérant que ce document vous aura aidé, voire donné envi d'en savoir encore plus ;-)