Le RMM, comment ca marche…

PhilP – v5 Nov. 2025

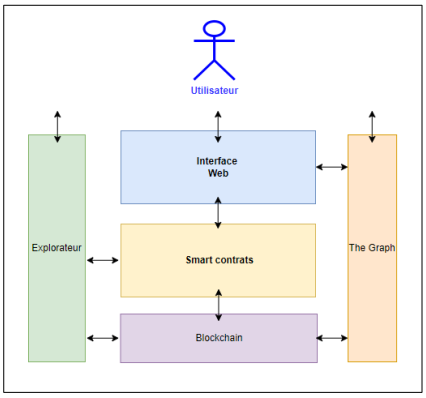
En complément du guide utilisateur, pour les curieux qui souhaiteraient comprendre comment ça marche (« DYOR »), le présent document va détailler :

- les composants de l'appllication : architecture, tokens, smart contrats (chapitres 1 à 4),
- le fonctionnement et mode de calcul de la solution (chapitre 5),
- comment agir directement sur les smart contrats (chapitre 6),
- une introduction à l'indexeur TheGraph (chapitre 7),
- l'environnement de test et quelques fonctions inactivées (chapitre 8 et 9),
- l'accès au code de l'application (chapitre 10).

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 s'exécutent sur une blockchain (ici Gnosis),
- et dans le cas présent, d'un service d'accès et d'indexation aux 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/ . (ce qui est bien utile, lorsque l'un des deux est indisponible ..)

Le service d'indexation de données, The Graph, est accessible par l'utilisateur : c'est d'un usage assez complexe et une seimple présentation sera faite en fin de document (chapitre 7).

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…

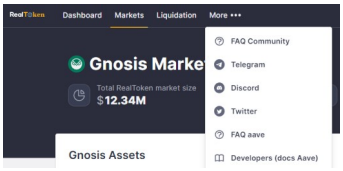
2 - Les sources de l'application RMM : AAVE

Si vous allez sur l'application AAVE (https://app.aave.com/), vous allez retrouver un Interface très semblable au RMM.

AAVE a deployé, en 2020, une dAPP open source de prêt / emprunt sur Ethreum.

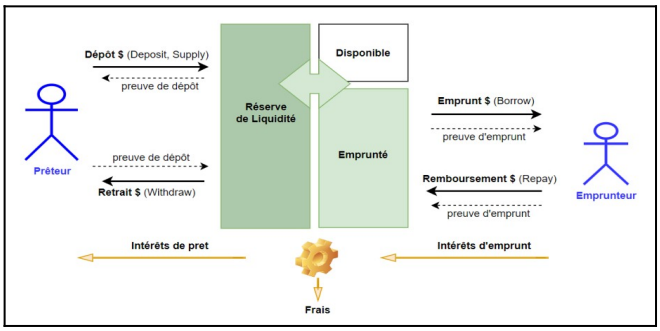
La solution d' AAVE , auditée par différentes sociétés (https://github.com/aave/aave-v3-core/tree/master/audits) et sans gros incident depuis, a été la base de la solution RMM mise en place par RealT en 2022.

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.



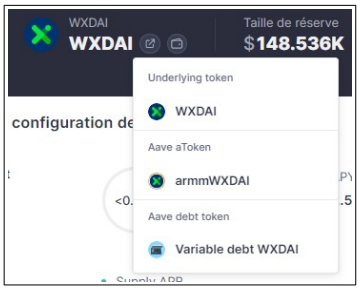
RealT a ajouté quelques fonctions complémentaires, notamment : la partie liquidation et le Wrapper (pour la version 3).

3 – Les tokens (et leur cinématique dans RMM / AAVE)



Les dépôts et emprunts d'**Actifs (WXDAI, USDC)** 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 emprunts de WXDAI, RMM va :
  - bloquer une partie de vos armmToken pour assurer la garantie de votre prêt,
  - puis, envoyer sur votre wallet, les WDXAI souhaités ainsi que des tokens (preuve) 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 types de token associés :

- la preuve de dépôt : armmToken,
- les preuves de dette : variable et stable (fonction actuellement désactivée).

Les adresses des différents token sont disponible dans l'interface de l'application, cf image :

Les token 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 (on dit qu'ils sont « rebasing »)
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 :

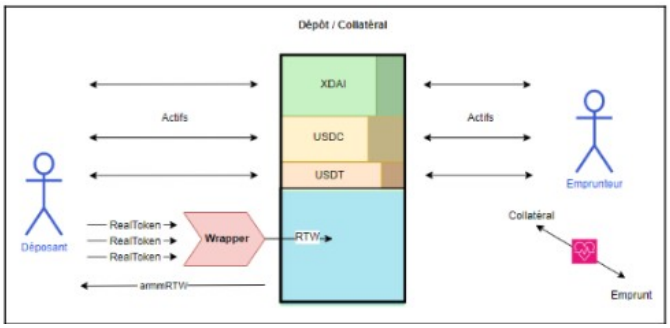
- si vous transférez un armmToken, le destinataire pourra soit les garder, soit les convertir dans l'actif sous-jacent,
- par contre vous ne pouvez transférer de jetons de dette (pour que le destinataire, les garde ou les rembourse à votre place), car cette dette est liée à une garantie qui ne serait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

Concernant les **Realtoken** : Ils ont, comme les Actifs, leur armmToken (preuve de dépôt) mais puisqu'ils ne sont pas empruntables, ils n'ont pas de token de dette.

En version 2, le nombre de Realtoken était limité et chaque Realtoken avait sont armmToken propre.

En version 3, le nombre de Realtoken déposable étant bien supérieur à la limite d'AAVE (128), RealT a du développer un système de Wrapper.

Le Wrapper est une smart contrat qui reçoit des Realtoken et transmet en échange des tokens RTW-USD dont la quantité correspond à la valeur en USD des Realtoken échangés.



RMM gère une pool de RTW-USD, qui à la différence des WXDAI et USDC : n'est pas visible dans l'interface de l'application, car ces tokens restent interne au RMM. Par contre, en échange du dépôt de Realtoken, wrappé en RTW-USD déposés sur RMM, l'utilisateur reçoit des armmv3RTW-USD.

Lorsque les Realtoken, déposés dans le wrapper, sont réévalué, la valeur des RTW-USD et armmv3RTW-USD est modifié en conséquence.

Les smart contract du wrapper ont été audités :
https://github.com/abdk-consulting/audits/blob/main/realt/ABDK_RealT_RealTokenWrapper_v_1_0.pdf

Des fonctions complémentaires ont été ajoutées au wrapper, afin de traiter certains cas particullier de retrait de Realtoken : https://wiki.realtoken.community/fr/dapp-ecosystem/rmm

En résumé, liste des Tokens avec leur adresse :

- WXDAI : https://gnosisscan.io/token/0xe91d153e0b41518a2ce8dd3d7944fa863463a97d
  - armmv3WXDAI (preuve de dépôt de WXDAI) : https://gnosisscan.io/token/0x0cA4f5554Dd9Da6217d62D8df2816c82bba4157b
  - variableDebtrmmv3WXDAI (dette variable en WXDAI) : https://gnosisscan.io/token/0x9908801dF7902675C3FEDD6Fea0294D18D5d5d34
- USDC : https://gnosisscan.io/token/0xddafbb505ad214d7b80b1f830fccc89b60fb7a83
  - armmv3USDC (preuve de dépôt d'USDC) : https://gnosisscan.io/token/0xeD56F76E9cBC6A64b821e9c016eAFbd3db5436D1
  - variableDebtrmmv3USDC (dette variable en USDC) : https://gnosisscan.io/token/0x69c731aE5f5356a779f44C355aBB685d84e5E9e6
- RTW-USD -01: https://gnosisscan.io/token/0xd3DFf217818b4F33eB38a243158FBeD2BBB029D3
  - armmv3RTW-UDS_01 (preuve de dépôt de RTW-USD) : https://gnosisscan.io/token/0xF3220Cd8F66AEB86fC2A82502977EAb4BFd2f647

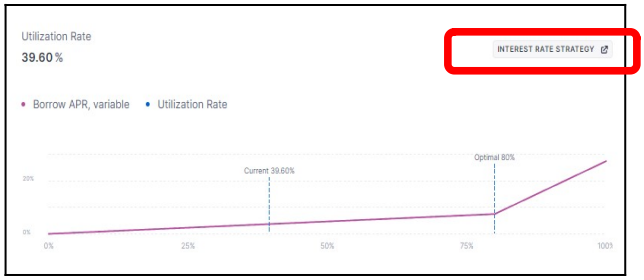
4 - Les Smart Contrats

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 dans l'interface :



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 lecture (« 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.

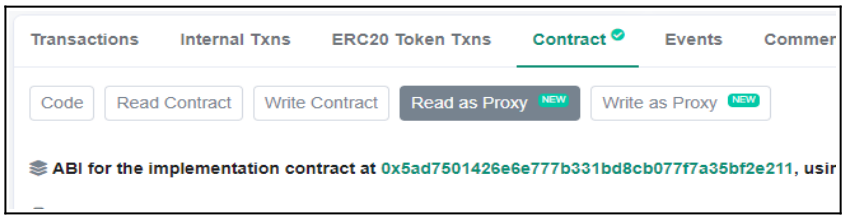
4.2 - Smart contrat : *Lending Pool*

RMM v3 : https://gnosisscan.io/address/0xfb9b496519fca8473fba1af0850b6b8f476bfdb3

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

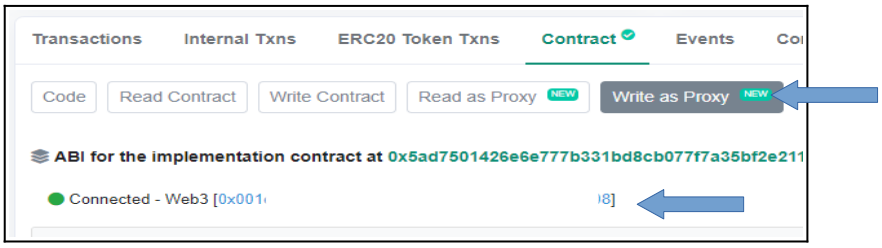
L'accès à ce smart contrat se fait au travers d'un Proxy :
Les smart contrats, une fois enregistrés à une adresse, sont non modifiables. C'est une des forces de la blockchain, mais ce peut être une contrainte lorsqu'on développe et qu'on a besoin de faire des mises à jours. Pour ce faire, on utilise 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.

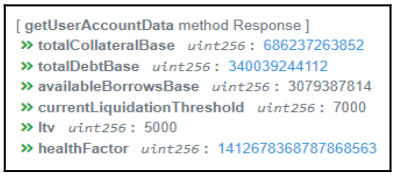
L'explorateur permet d'accéder à un smart contrat en lecture et en mise à jour. Dans ce dernier cas, il vous faudra vous connecter avec votre wallet. Cette connexion est l'équivalent de celle que vous faite lorsque vous vous identifiez sur l'interface de l'application RMM (puisqu'en accédant directement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



Des exemples d'actions, directement sur les smart contrat, seront détaillés 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. getUserAcountData*» : pour obtenir la position globale d'un wallet sur le RMM (notamment son HF),



- d'autre champs sont disponibles, notamment pour des fonctions qui sont inactives dans le RMM (Flashloan, eMode,..).

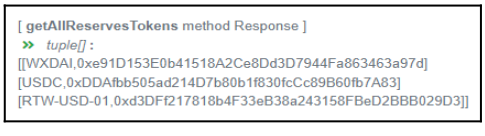
4.3 - Smart contrat : *Data Provider*

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).

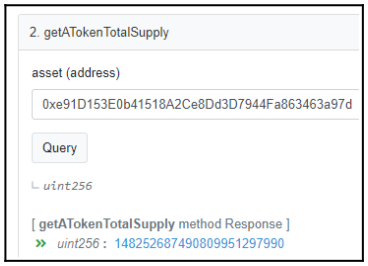
De nombreuses informations sont disponibles (19), sur le fonctionnement du pool :

- «*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»),

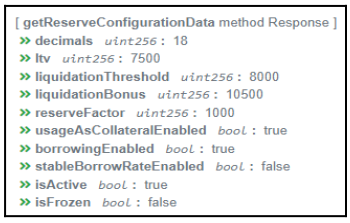


- «*3. getAllATokens*» : pour les adresses des aToken (preuve de dépôt) de chaque réserve,

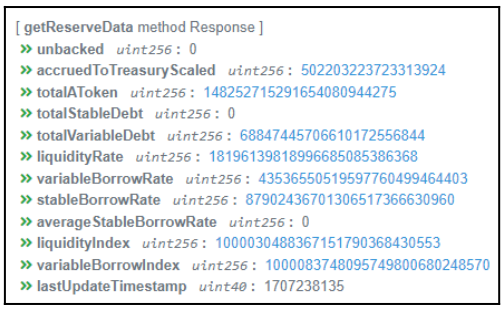
- «*2. getATokenTotalSupply*» : pour la quantité déposée dans une réserve donnée :
  L'adresse « asset » est celle figurant au point 4 ci-dessus (exemple avec le XDAI),
  Les quantités de XDAI sont exprimées avec une précision de 18 décimales (cf info dans le point suivant). Pour obtenir la valeur déposée,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



- «*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.

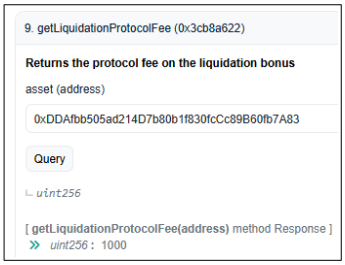


- «*13. getReserveData*» : pour les valeurs instantanées de chaque réserve : valeur déposée, empruntée, taux..



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

- «*5. getDebtCelling*» : pour la limite d'emprunt (pour les réserves en mode isolé),
- «*8 getInterstRateStrategyAddress*» : pour l'adresse du smart contrat qui fixe les caractéristiques 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,
- « 9 getLiquidationProtocolFee » : pour les frais de protocole prélevés lors des liquidations (sur le Bonus). Exprimé en BPS (soit 0,01%). Sur l'image, cela correspond à 10 % du Bonus, pour la réserve USDC.



Adresse des principaux smart contracts du RMM v3 :

Smart Contract	**Addres**
ACLManager-Realt	0x6a1163DAF9F5909990B547C15Dbd672169464055
AToken-Realt	0x565CFf7a77BA690FC9D860530413761d77c2DDD3
AaveOracle-Realt	0xb4AE809Ad7CEB7e5B579dEdD0De7c213aD5AB516
BorrowLogic	0x5c6D6267cDAe7E863EE6B05E59081E914D9Db40a
BridgeLogic	0x2CA4FC4a3CDB423dc3B5B387fd3402f74864cFfB

ConfiguratorLogic		0xB5C65aF5f255183Aaf37D9DBF4758CBbb6Fe6648
DelegationAwareAToken-Realt		0xAeF1077f2d79A8a7e9827CD2247d9fcb5B8b8f83
EModeLogic		0xe8295699b0c6C6953D972692B274Eb421c6F70a8
EmissionManager		0xA3990aDdC83F603e4b368cF6630Ec975a5c65D51
FlashLoanLogic		0x7eD919ecF07d0b002d835d8bf3a8371e9875529A
IncentivesProxy		0x11898E9F1C7DE79D944dAD26DC75485e8E360F36
IncentivesV2	implementation	0xeA8f6500F1B0EEce7b08Ea4e709F462F6Dbf4179
LiquidationLogic		0x40caEEAa5187a5c5f0bFc6813f05f74656F7d765
Pool	implementation	0x5ad7501426e6e777B331Bd8cb077F7a35Bf2E211
Pool-Proxy-Realt		0xFb9b496519fCa8473fba1af0850B6B8F476BFdB3
PoolAddressesProvider-Realt		0xdAA06Cf7adCEb69fCFDe68d896818b9938984A70
PoolAddressesProviderRegistry		0xC6c4b123e731819AC5f7F9E0fe3A118e9b1227Cd
PoolConfigurator	implementation	0x2c1134079676Babe28F8239Db4B88dCd8999dC92
PoolConfigurator-Proxy-Realt		0xc2af0FFE79Eb3e0110C108F7b2d849818c338e8D
PoolDataProvider-Realt		0x11B45acC19656c6C52f93d8034912083AC7Dd756
PoolLogic		0x43682a8f83d67e1289b9b619b8084baC3b9390DA
ReserveStrategy-rateStrategyStableOne		0xec016116537dC3f8a42B81728D8DE04eCC45853a
ReserveStrategy-rateStrategyStableTwo		0x600763b246d20c198a2697604185E91b6bf1aa96
ReserveStrategy-rateStrategyVolatileOne		0xd134fb028e83c40c6136224F2270f373150a2434
ReservesSetupHelper		0xB8E3505F26282ffff800DFDa2dF8c6fde1d7E8E6
StableDebtToken-Realt		0x671Ab2985bC1DFCF9CAFD37235FD41BF756Ff2b3
SupplyLogic		0xAE93423A8C03B281A6869Ac6B592d7161585183e
Treasury-Controller		0x062EEE59Cdf2bB4a7B6f1bc98acBA92DE99e6065
Treasury	implementation	0x3E5c150A97DEF25B4E4fDD1F968Be20f762b2Ad2
TreasuryProxy		0x586B572EDF0916D2aFEa1f909B1ff8D8eC8a4210
USDC-AToken-Realt		0xeD56F76E9cBC6A64b821e9c016eAFbd3db5436D1
USDC-StableDebtToken-Realt		0x3D1Dae285860153169E17A5365492C6bbA16979e
USDC-VariableDebtToken-Realt		0x69c731aE5f5356a779f44C355aBB685d84e5E9e6
UiIncentiveDataProviderV3		0x64FADA70290F3375125182FB7bfd6443DB8Ec6C5
UiPoolDataProviderV3		0x0ACBD24A3804d57eDdDC6A0c0d67Ed33f690bcDF
VariableDebtToken-Realt		0x0ac8b3F53E610Ed89C56B21f835Bb1332F405bfc
WXDAI-AToken-Realt		0x0cA4f5554Dd9Da6217d62D8df2816c82bba4157b
WXDAI-StableDebtToken-Realt		0x8ACD88D494cFA56F542234f8924F06024b5795B5
WXDAI-VariableDebtToken-Realt		0x9908801dF7902675C3FEDD6Fea0294D18D5d5d34
WalletBalanceProvider		0x6fCed45102121855Fd0264ACc71AA448eDb9cbC7
WrappedTokenGatewayV3		0x2A36C23f113053e08BBDF4a6DB30Fc37fE0b1068
RealTokenWrapper	proxy	0x10497611Ee6524D75FC45E3739F472F83e282AD5
RealTokenWrapper	implementation	0x12a000a8A2Cd339D85119C346142Adb444bc5ce5
RTW	proxy	0xd3DFf217818b4F33eB38a243158FBeD2BBB029D3
RTW	implementation	0x690Aa27EE8ab8ee0840bA7F07260F1dfe333319E
ATokenRTW	implementation	0xf3DE3A9719D3Cc1444F9dF05a168C42296648bA7

5 - Fonctionnement de RMM / AAVE

Une fois les paramètre fixés, dans les smart contrats mentionnés dans le chapitre précédent, l'application RMM fonctionne sans intervention humaine, de façon complètement automatque (d'où le nom aussi utilisé de « Protocol »). Le présent chapitre va décrire l'algorythme utilisé afin que la solution soit continuellement en équilibre financier. Nous allons suivre les flux financiers, pour comprendre le mode de fonctionnement.

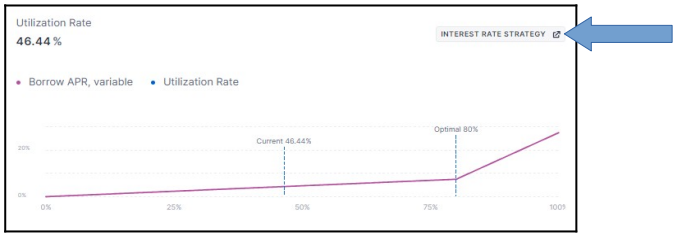
Pour le flux du capital : tout commence par des dépôts, pour alimenter la réserve de liquidité. Sans argent dans la réserve, il ne peut y avoir de prêts !.

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 préteurs.

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

5.1 - Calcul des taux d'emprunt

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



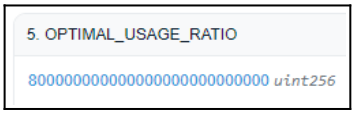
Ce smart contrat (*ReserveInterstRateStrategy*)

RMM v3 : https://gnosisscan.io/address/0xec016116537dC3f8a42B81728D8DE04eCC45853a#readContract

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

- Le **taux optimal d'utilisation** : c'est la valeur à partir de laquelle la pente de taux est bien plus «forte» ..
  Ce mécanisme est important, car il concoure à l'équilibre financier du système :
  - en utilisation «normale» (de 0 % à l'optimal) : le taux d'emprunt est faible et monte doucement au fur et à mesure de l'utilisation de la réserve,
  - quand on commence à manquer de liquidité (de l'optimal à 100%) : le taux d'emprunt augmente plus fortement afin :
    - de pousser les emprunteurs à rembourser (le taux de leur emprunt devenant élevé), on retrouve ainsi de la liquidité,
    - les taux de dépôts étant lié aux taux d'emprunt, ils augmentent eux aussi. Ce qui attire plus de préteurs et améliore la liquidité.

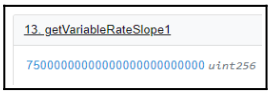
  La valeur indiquée dans le smart contrat :
  Les taux sont exprimé en « ray » : 10^27. Pour obtenir le % en décimal, de l'optimal, il faut donc diviser la valeur affichée par 10^27.
  (80 % dans l'exemple)



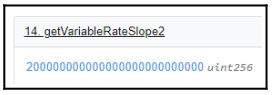
- Pour un emprunt à **taux variable** :

- **Le taux de base**, point de départ de la courbe ci-dessus :
  Il s'agit de la valeur du taux d'emprunt à 0 % d'utilisation de la réserve.
  En général, c'est 0.

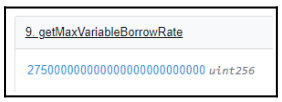
- **La première pente :** soit l'augmentation du taux quand on passe de 0 à l'optimal (7,5 % dans l'exemple).



- **La seconde pente :** soit l'augmentation du taux quand on passe l'optimal à 100 % (20 % dans l'exemple).



- A saturation de l'utilisation de la réserve (100%), le taux d'emprunt sera donc la somme des trois valeurs qui précèdent (27,5 % dans l'exemple). Chiffre qu'on retrouve au point 9 du smart contrat (cf image).

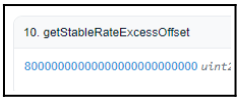


- Pour un emprunt à **taux stable** (fonction actuellement désactivée) :
  - La courbe est du même type, mais avec des paramètres différents,
  - Le taux de base, est généralement pas 0 (8,5 % dans l'exemple),

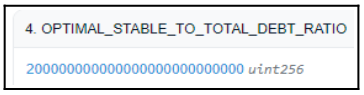
  - La première pente (0,5 % dans l'exemple).
    En taux stable, lors l'utilisation « normal » de la réserve, le taux est donc assez stable (passant dans l'exemple de 8,5 % à 9 %).

  - La seconde pente (75 % dans l'exemple),

  Pour le bon fonctionnement en taux stable, des mécanismes complémentaires sont ajoutés :
  - 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)



Pour passer des paramètres à la courbe (affichée dans l'interface), il suffit d'appliquer les formules suivantes :

The interest rate R_t follows the model:

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

$$if\ U > U_{optimal}: \qquad R_t = R_0 + R_{slope1} + \frac{U_t - U_{optimal}}{1 - U_{optimal}} R_{slope2}$$

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

Dans la quelle :

- Ut : 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).

La formule ci-dessus s'applique pour les deux type d'emprunt à taux variable et à taux stable.

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 :

$$base_s = slope_{v,1} + \text{offset}_{base}, \qquad E_{util} = 10^{27} - O_{util} \tag{1}$$

$$ratio = \frac{debt_{stable}}{debt_{stable} + debt_{variable}} \tag{2}$$

$$rate_s = \begin{cases} base_s + slope_{s,1} + slope_{s,2} * \frac{(util - O_{util})}{E_{util}}, & \text{if } util > O_{util} \\ base_s + slope_{s,1} * \frac{util}{O_{util}}, & \text{otherwise} \end{cases} \tag{3}$$

$$rate_s = \begin{cases} rate_s + \text{offset}_{excess} * \frac{ratio - O_{ratio}}{10^{27} - O_{ratio}}, & \text{if } ratio > O_{ratio} \\ rate_s, & \text{otherwise} \end{cases} \tag{4}$$

page 12 :
https://github.com/aave/aave-v3-core/blob/master/techpaper/Aave_V3_Technical_Paper.pdf

La formule peut paraître complexe, mais c'est en fait assez simple ;-) :

- (1) $base_s$: 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),
- O_{ratio} : est le seuil d'emprunt à taux stable, évoqué ci-avant (point 4 du smart contrat de stratégie d'intérêt),
- (4) $rate_s$: est le taux d'emprunt stable, qui est calculé différemment suivant le ratio :
  - si les emprunts à taux stable sont inférieur au seuil (ratio < O_{ratio}), le calcul se fait comme expliqué précédemment mais avec des paramètres pour le taux stable (3),
  - si les emprunts à taux stable dépasse le seuil (ratio> O_{ratio}), un taux complémentaire ($offset_{excess}$) est ajouté au taux précédemment calculé au prorata du ratio par rapport à 100 % (ratio – O_{ratio}) / (1 – O_{ratio})

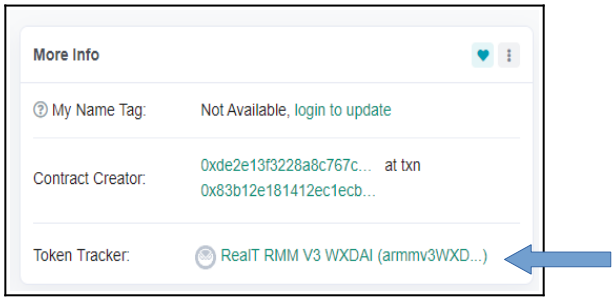
Un exemple sera détaillé, dans un prochain chapitre, pour être plus parlant ..

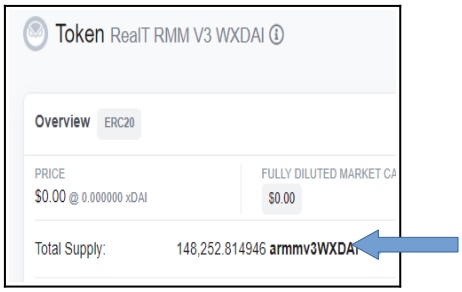
5.1.1 - Taux d'utilisation de la réserve

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 dépôts 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.

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.

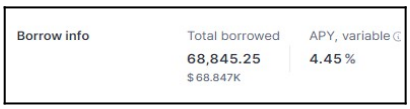
Faites la même chose, pour les deux autres tokens de dette (en variable et en stable),

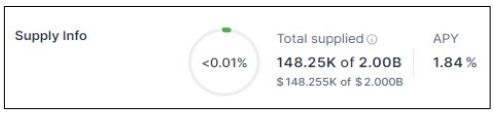
0

Le prêt stable étant desactivé 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. (dan l'exemple : (68 847,52 + 0) / 148 252,61 = 46,44%)

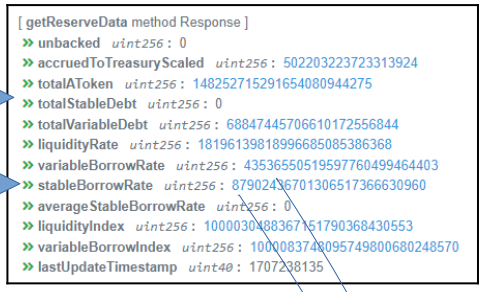
Nous retrouvons ces chiffres dans l'interface :





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.



5.1.2 - Exemple de calcul des taux d'emprunt

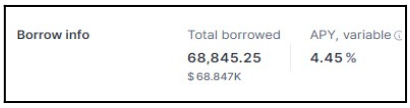
Validons notre compréhension des calculs, à partir de la formule théorique

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

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.

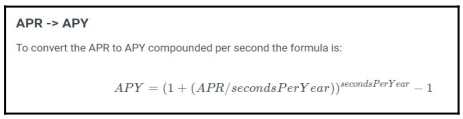
Ces chiffres correspondes à ceux de la query 13 évoqué ci-dessus, mais pas exactement a celui affiché pour le taux variable … car ce dernier est affiché en APY.



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 :



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

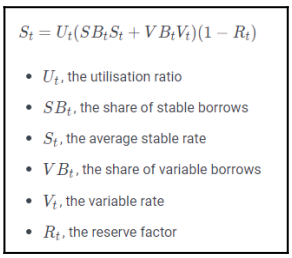
Les intérêts collectés auprès des emprunteurs sont intégralement distribués en intérêt aux déposants (hors frais).
Donc : Somme empruntée * Taux d'emprunt = Somme déposée * Taux de dépôt,
Ou : Taux de dépôt = Taux d'emprunt * Somme empruntée / Somme déposée,
Taux de dépôt = Taux d'emprunt * Taux d'utilisation 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'ou : 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)

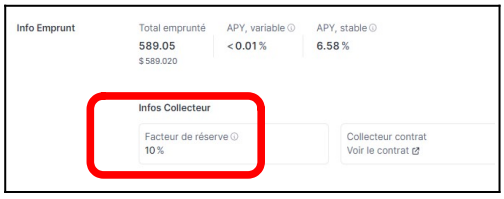
La formule de calcul du taux de dépôts (S_t), donnée dans la documentation d'AAVE correspond exactement à celle détaillée ci-avant :

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

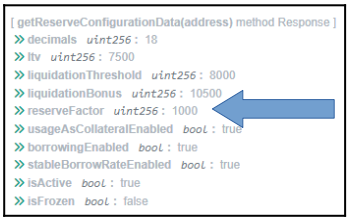


La valeur du *Reserve Factor,* fixée pour une réserve, est visible :

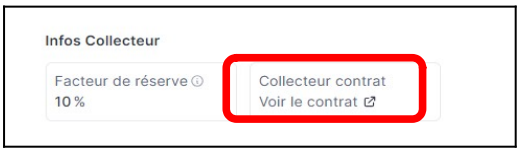
- soit sur sur l'interface :

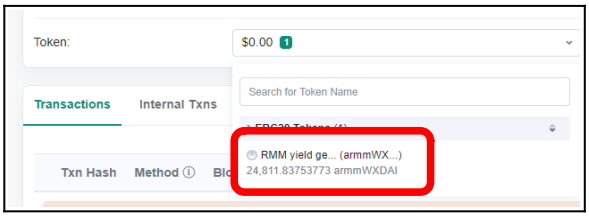


- soit dans la query 12 du smart contrat *Data Provider :*



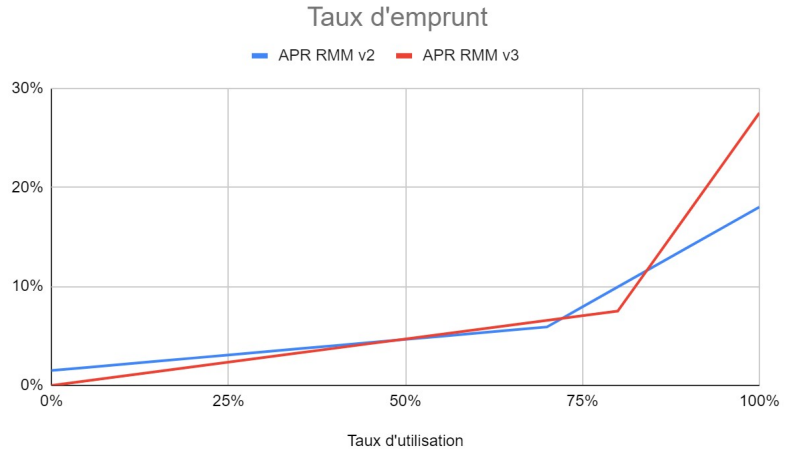
Le montant accumulé par le *Reserve Factor,* est visible en utilisant le lien du contrat Collecteur qui est dans l'interface :





5.3 – Synthèse des paramètres

	RMM v3			RMM v2	
	XDAI	**USDC**	**RealToken (RTW)**	**XDAI**	**RealToken**
Max LTV (« Pouvoir d'emprunt »)	75 %	80 %	50 %	75 %	50 %
Seuil de liquidation	80 %	85 %	70 %	80 %	70 %
Pénalité de liquidation	5 %	5 %	10 %	5 %	10 %
Collatéralisable	x	x	x	x	x
Empruntable	x	x		x	
Facteur de réserve (frais sur intérêts)	10 %	10 %		10 %	
Modèle de taux d'intérêt (variable) :					
Taux d'utilisation Optimal	80 %	80 %		70 %	
Taux d'emprunt Min. (Utilisation 0%)	0,0 %	0,0 %		1,5 %	
Pente initiale (avant Taux Opt.)	7,5 %	7,5 %		4,4 %	
Pente secondaire (après Taux Opt.)	20,0 %	20,0 %		12,1 %	
Taux d'emprunt Max. (Utilisation 100%)	27,5 %	27,5 %		18,0 %	



5.4 - 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 nomé « *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'indexe aura augmenté, vous verrez votre solde augmenter, sans qu'aucune mise à jour n'ai 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 :

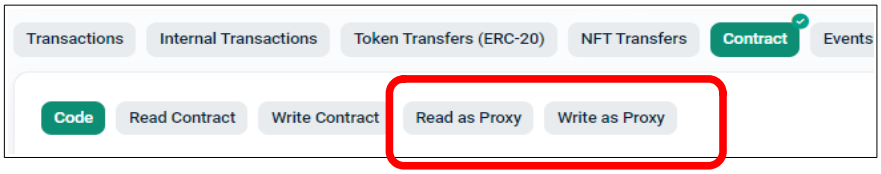
- dans le support AAVE : https://docs.aave.com/developers/guides/rates-guide#how-is-yield-accrued
- ou, dans le smart contrat correspondant : https://github.com/aave/aave-v3-core/blob/6070e82d962d9b12835c88e68210d0e63f08d035/contracts/protocol/tokenization/AToken.sol#L128

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

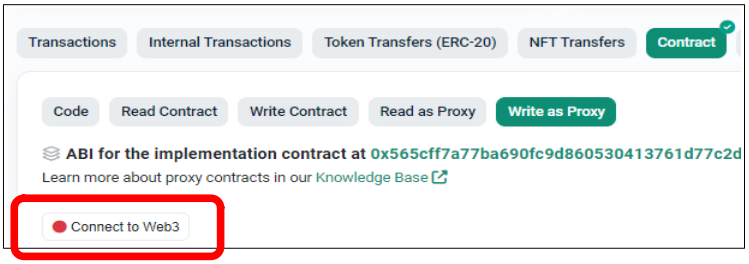
6 – Comment agir directement sur les smart contrats

(A partir de l'explorateur, sans passer par l'interface de l'application)

Les actions peuvent être en lecture seule ou en écriture.



En écriture, vous devrez vous connecter avec le wallet qui va effectuer et approuver les transactions :



Commençons par le plus simple et le moins risqué : des exemples d'accès en lecture ...

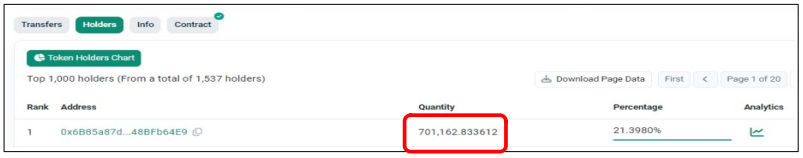
6.1 – A partir d'un des tokens de dépôt :

- **Liste des principaux apporteurs de liquidité**

  Exemple (en lecture) à partir du token de dépôt USDC du RMM v3 :

  https://gnosisscan.io/token/0xeD56F76E9cBC6A64b821e9c016eAFbd3db5436D1#balances

  Au moment où sont écrites ces lignes, un wallet a déposé 701 K$!



  Pour le WXDAI : https://gnosisscan.io/token/0x0cA4f5554Dd9Da6217d62D8df2816c82bba4157b#balances

- **Historique de vos dépôts et retraits**

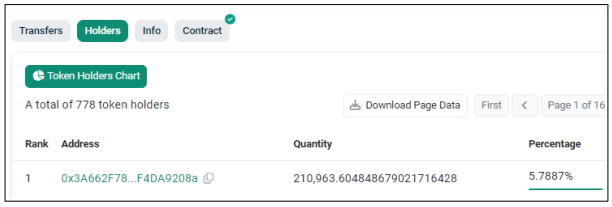
  En indiquant l'adresse de votre wallet

6.2 – A partir d'un des tokens d'emprunt :

- **Liste des plus gros emprunteurs**

A partir du token de dette variable en wXDAI sur RMM v3 :
https://gnosisscan.io/token/0x9908801dF7902675C3FEDD6Fea0294D18D5d5d34#balances

Soit plus de 210 K$ pour le premier !



Pour l'USDC : https://gnosisscan.io/token/0x69c731aE5f5356a779f44C355aBB685d84e5E9e6#balances

- **Historique de vos emprunts et remboursements**

En indiquant l'adresse de votre wallet

6.3 - Position RMM d'un portefeuille :

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

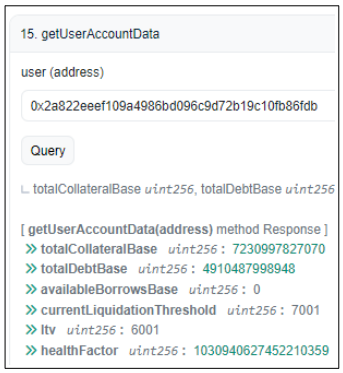
A partir du smart contrat de gestion du pool de liquidités (du RMM v3) :

https://gnosisscan.io/address/0xFb9b496519fCa8473fba1af0850B6B8F476BFdB3#readProxyContract

en mettant l'adresse du wallet, sur la fonction *15. getUserAccountData*
par ex avec l'adresse la plus proche actuellement de la liquidation

Vous retrouvez les valeurs suivantes :

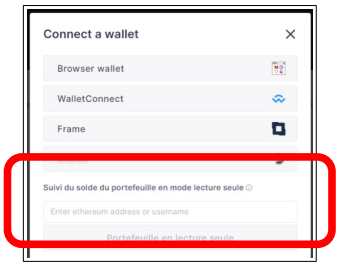
- Total en collatéral = 72 310 $ (valeur / 10^8)
- Total de l'emprunt = 49 105 $ (valeur / 10^8)
- Seuil de liquidation = 70,01 %
- Health Factor = 1,0309 (valeur / 10^18)
  Valeur que l'on retrouve, par le calcul avec les trois premières valeurs (72 310 * 70% / 49 105)



Ces informations vous permettrons, 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-delais-avant-liquidation,

Pour mémoire, sur l'inteface vous pouvez voir la position d'un wallet en vous connectant à l'interface RMM en mode lecture



6.4 – Actions, en mise à jour, sur les smart contrats

A la différence des actions précédentes en lecture, qui n'induisent aucun risque, les actions en écritures engagent vos fonds, nécessitent une (ou des) approbation (s) et donc toute votre attention. Par précaution, avant de faire une transaction sur une grosse somme, faites un test avec un petit montant (comme dans les exemples qui suivent).

Les principales actions en mise à jour sur RMM, sont synthétisés dans le tableau suivant :

			Fonction du smart contrat, pour chacune des actions sur RMM			
Token	**Réserve RMM**	**Smart contrat**	**Dépôt**	**Retrait**	**Emprunt**	**Remboursement**
XDAI	WXDAI	Gateway	2. depositETH	8. withdrawETH	1. borrowETH	6. repayETH
WXDAI		Pool RMM v3	24. supply	29. withdraw	2. borrow	15. repay
USDC	USDC					16. repayWithATokens
aWXDAI	WXDAI					
aUSDC	USDC					
RealToken	RTW-USD-01	RealToken Wrapper	13. supply	22. withdraw		
Approval préalable			WXDAI, USDC, Realtoken	aWXDAI pour XDAI		WXDAI, USDC

	Décimal	**Adresse smart contrat**
XDAI	18	Coin Gnosis
WXDAI	18	0xe91D153E0b41518A2Ce8Dd3D7944Fa863463a97d
USDC	6	0xDDAfbb505ad214D7b80b1f830fcCc89B60fb7A83
aWXDAI	18	0x0cA4f5554Dd9Da6217d62D8df2816c82bba4157b
aUSDC	6	0xeD56F76E9cBC6A64b821e9c016eAFbd3db5436D1
Gateway		0x2A36C23f113053e08BBDF4a6DB30Fc37fE0b1068
Pool RMM v3		0xFb9b496519fCa8473fba1af0850B6B8F476BFdB3
RealToken Wrapper		0x10497611Ee6524D75FC45E3739F472F83e282AD5

En fonction du Token objet de la transaction, sont indiqués (dans le tableau supérieur) :

- la réserve RMM v3 correspondante,
- le smart contrat à partir duquel une fonction sera à executer, suivant l'action souhaitée sur RMM,
- Le token sur lequel sera éventuellement nécessaire un approval, au préalable de l'execution de la fonction.

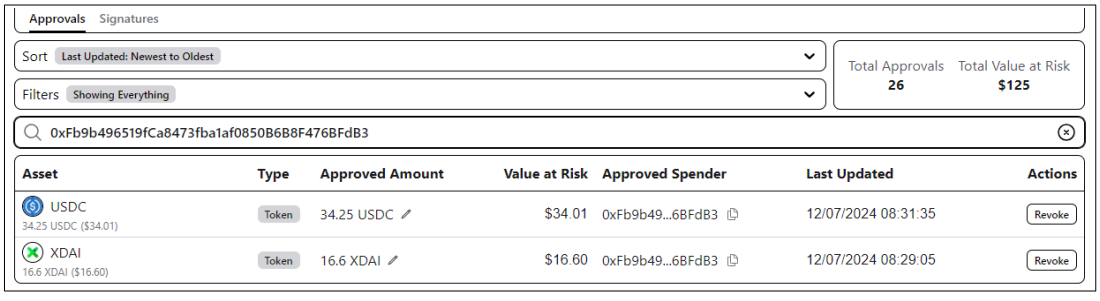
Les fonctions nécessitent différentes données pour leur execution (tableau inférieur) :

- des montants, à formaliser en fonction du nombre de décimal du token,
- des adresses de token,
- ainsi que d'autre informations, qui seront précisé dans les exemples qui suivent.

Approval préalable :

Si lors pour l'execution de la fonction, le smart contrat a besoin de transférer des tokens sur votre wallet, il est nécessaire qu'avant l'execution de la fonction vous donniez l'autorisation correspondante.

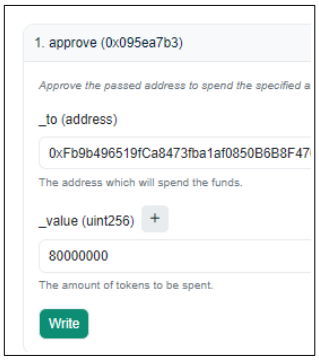
Par exemple lors d'un dépôt ou d'un remboursement, le smart contrat RMM va devoir transférer des WXDAI ou USDC à partir de votre wallet. Il doit donc avoir été autorisé au préalable (allowance).
Pour cela, une autorisation doit avoir été donnée sur le token (allowance) avec l'adresse du contrat autorisé et le montant maximum de l'approval.
Vous pouvez déjà vérifier les approbations existantes, via l'application https://revoke.cash/.
En connectant votre wallet (et en étant sur la blockchain Gnosis), vous pouvez lister vos approvals en les triant par smart contrat (ici celui du pool RMM v3)



On voit ici, combient le smart contrat du pool RMM peut transférer de chaque token à partir du wallet (connecté à revoke.cash) : 34,35 USDC et 16,6 WXAI.

Si ces montants ne sont pas suffisants, vous pouvez :

- soit les modifier en utilisant le crayon à coté de « l'Approved Amount »,
- soit (si ca ne fonctione pas ou que la ligne n'existe pas) aller directement sur le smart contrat du token et executer la fonction approve avec le montant qui convient.
  Exemple, à partir du smart contrat de l'USDC et de sa fonction approve, pour passer l'autorisation :
  - du smart contrat du pool RMM (_to (address))
  - à transférer jusq'à 80 USDC (_value (uint256))
    (attention à bien respecter les décimales..)



Décimales des Tokens :

Chaque token à sa propre précision en format décimal (18 ou 6 dans les exemples). Par contre, le langage des smart contrat (solidity) ne supporte pas le format des nombres décimaux, les nombres doivent être indiqué sans virgule et avec tous les chiffres de la précision décimale.
Par exemple, ci-dessus :

- le token USDC sur Gnosis à une précision de 6 décimales,
- pour indiquer 80, il faut donc ajouter à 80 : 6 zéro (ou le multiplier par 10 puissance 6).

Pour le token WXDAI, qui a une précision de 18 décimales, pour indiquer 80 WXDAI il faudrait y ajouter à 80 : 18 zéro !..

Attention à ne pas vous tromper, et ne pas ajouter un chiffre en trop (avec autant de chiffre, cela pourrait arriver ..).
Le wallet MetaMask, redonne le montant en valeur décimale avant l'approbation : vous pourrez ainsi vérifier.

Exemples de transaction :

- **Transactions en Coin XDAI**

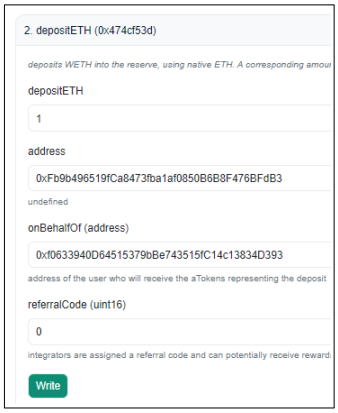
Il est possible de faire des transactions à partir des XDAI, alors qu'il n'existe pas de réserve en tant que telle. Les transaction en XDAI se feront au travers d'un smart contrat particulier qui opérera la convertion XDAI / WXDAI (WrappedTokenGatewayv3) pour échanger avec la réserve en WXDAI :

https://gnosisscan.io/address/0x2A36C23f113053e08BBDF4a6DB30Fc37fE0b1068#writeContract

Le XDAI étant le coin (monnaie pour payer les frais des transactions) de la blockchain Gnosis, les transactions en XDAI sont très simple car elle ne necessite qu'une seule approbation (pour le transfert de XDAI et le paiement des frais).

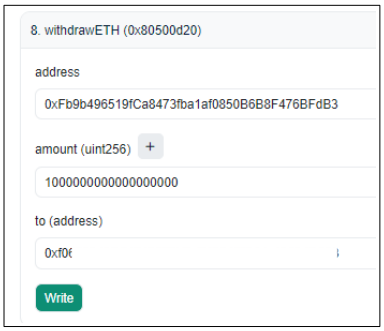
Le **dépôt de XDAI,** dans la réserve WXDAI, se fait avec la fonction « *2. depositETH* » (du smart contrat Gateway) en indiquant :

- le montant de WXDAI à déposér (ici 1, sur l'ensemble des exemples),
- l'adresse du Pool RMMv3,



- l'adresse du wallet qui recevra les preuves de dépôt (armmV3WXDAI),
  Cette adresse peut être différente de celle connectée au smart contrat Gateway, donnant son approbation et cédant ses XDAI.
- et un referralCode à 0.

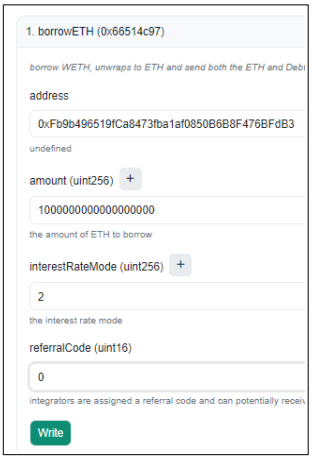
Le **retrait en XDAI,** à partir de la réserve WXDAI, se fait avec la fonction « *8. WithdrawETH* », en indiquant :



- l'adresse du Pool RMMv3,
- le montant de WXDAI retiré, donc exprimé en décimal x 10 puissance 18,
- et l'adresse du wallet qui recevra les XDAI (après conversion par la gateway).

Pour mémoire, cette fonction nécessite un approval au préalable : sur le smart contrat aWXDAI, afin que le smart contrat de la Gateway puisse transférer vos aWXDAI dans la quantité souhaitée.

L'**emprunt de XDAI**, à partir de la réserve WXDAI, se fait avec la fonction « *1. BorrowETH* », en indiquant :



- l'adresse du Pool RMMv3,
- le montant de WXDAI, exprimé en décimal x 10^18 (ici 1 $),
- le type de taux d'intérêt de l'emprunt : 2 pour variable,
- et le referralCode à 0

Nota : Lors du premier emprunt XDAI, une approbation préalable est faite.
Sur le contrat de dette variable WXDAI, la fonction approveDelegation donne le droit d'emprunt au smart contrat Gateway pour le token WXDAI pour un montant infini.
Si vous avez déjà emprunté en XDAI, cette approbation n'est plus à faire.

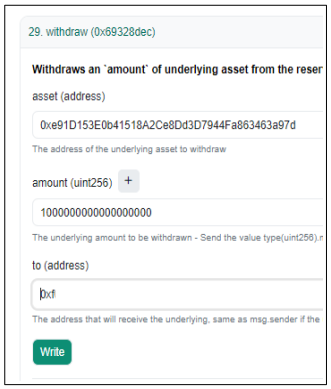
Le **remboursement de XDAI**, se fait de façon similaire avec la fonction « *6. RepayETH* »

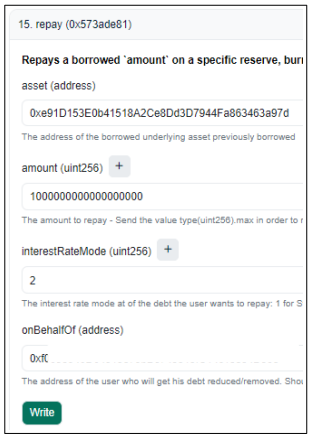
Commme pour le dépôt, le wallet connecté au smart contrat (ici qui rembourse) peut être différent de celui qui est addressé par la fonction « onBehalfOf » (ici qui a fait l'emprunt qui sera remboursé).
Ainsi vous pouvez rembourser l'emprunt d'un tier, ce qui n'est pas possible par l'application front-end.

- **Transactions en token USDC ou WXDAI**

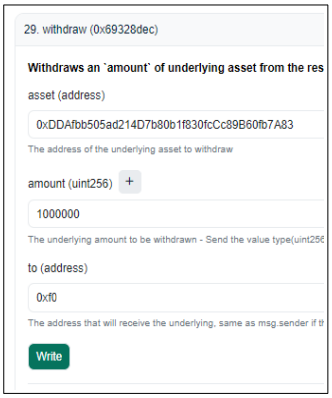
Dans ce cas, les transcations se font à partir du contrat du Pool RMM v3 directement :
https://gnosisscan.io/address/0xFb9b496519fCa8473fba1af0850B6B8F476BFdB3#writeProxyContract

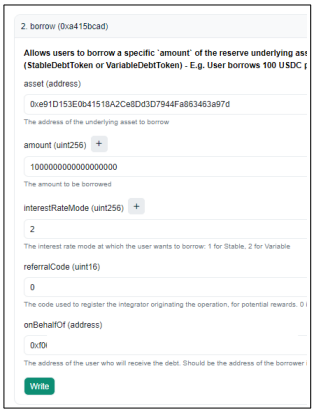
WXDAI : Dépôt (*) Retrait Emprunt Remboursement (*)

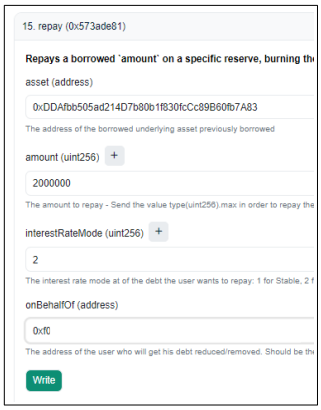




USDC : Dépôt (*) Retrait Emprunt Remboursement (*)







(*) Un Approval préalable est nécessaire sur le smart contrat du token WXDAI ou USDC afin que le smart contrat du pool RMM puisse transférer le token WXDAI ou USDC dans la quantité souhaitée.

- **Transactions en aToken**

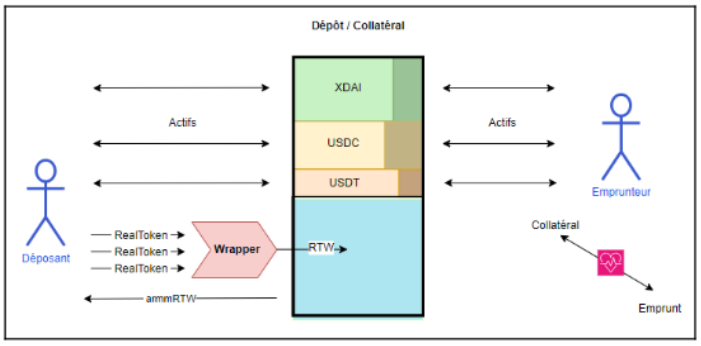
Comme expliqué au chapitre 2.4 du Guide utilisateur (https://community-realt.gitbook.io/tuto-community/defi-realt/rmm#guide-utilisateur), en v3 il est possible de rembourser un emprunt directement avec des token de preuve de dépôt (aToken).

La fonction à utiliser, sur le smart contrat du pool RMM, est : *16. repayWithATokens* de façon similaire au paragraphe précédent.

- **Transactions en RealToken**

Comme indiqué au chapitre 2.2.4 du Guide utilisateur

En version 3, le nombre de token dépassant la limite d'AAVE (128) , RealT a du créer un contrat intermédiaire pour consolider les RealTokens en RTW (Real Token Wrapped), et ainsi n'avoir qu'un nombre restreint d'actifs RTW (plutôt qu'autant d'actifs que de RealToken, comme en v2).
Le nombre de RTW créé, lors du dépôt correspond à la valeur en $ du RealToken déposé (ici 52,41).
En contre-partie du dépôt, on ne reçoit plus des armmToken comme en v2, mais des armmRTW et avec une quantité égale au montant du RealToken déposé.
Ces armmRTW sont bloqués sur votre wallet, vous ne pouvez les transférer.
Nota : Si le Realtoken déposé, venait a être réévalué : votre nombre d'armmRTW sera actualisé en conséquence.



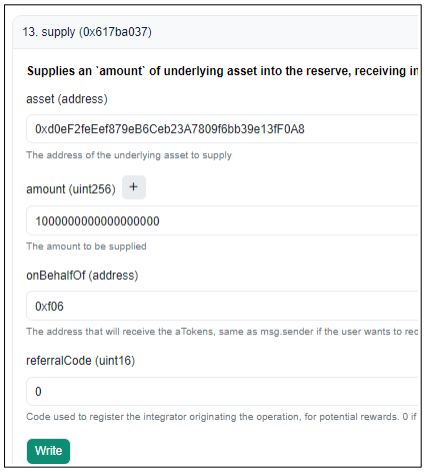
Les transactions de Realtoken, s'execute avec le smart contrat Realtoken Wrapper :

https://gnosisscan.io/address/0x10497611Ee6524D75FC45E3739F472F83e282AD5#writeProxyContract

Seuls les fonctions de dépôt et retrait existent, puisque les Realtoken ne sont pas empruntables.

Dépôt de Realtoken, avec la fonction *13. supply* en indiquant :

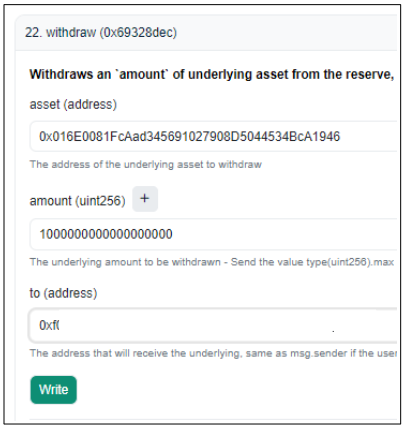
- l'adresse du Realtoken déposé,
- sa quantité en décimale x 10 puissance 18,
- le wallet cédant les Realtoken et recevant les preuves de dépôt aRTW
- et le referralcode 0



Un aproval préalable est nécessaire : sur le contrat du Realtoken, afin que le Realtoken Wrapper puisse le transférer dans la quantité souhaité.

Retrait de Realtoken, avec la fonction *22. withdraw,* en indiquant :

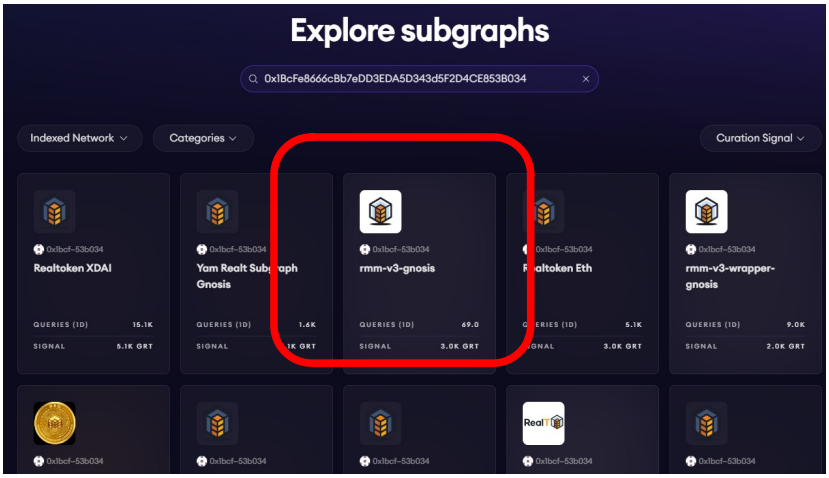
- l'adresse du Realtoken retiré,
- sa quantité en décimale x 10 puissance 18,
- le wallet recevant les Realtoken (ce Realtoken devant être whitelisté pour le wallet),
  Le wallet cédant les aRTW est celui connecté au smart contrat pour executer la fonction
- et le referralcode 0



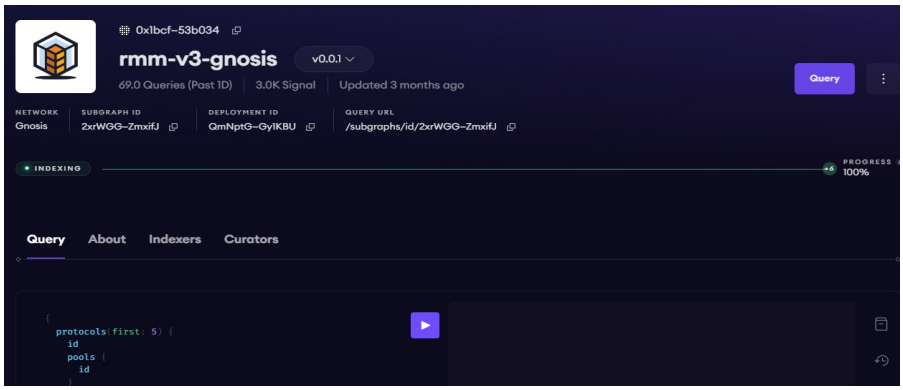
7 – Services The Graph

The Graph est un protocol (/service) pour « simplifier » l'accès aux données des blockchains. La simplification est surtout au niveau des applications, car : en une seule requette, on peut accéder à un ensemble d'informations, qui autrement aurait réclammé de multiples accès au smart contract (option plus lente et couteuse).
RealT utilise TheGraph et plusieurs Subgraph sont disponibles en relation avec les applications qu'ils ont développées.
Les Subgraph précédement disponibles sur le service « TheGraph Hosted » ont été migrés sur la solution décentralisée de TheGraph, depuis le 12 juin 2024.

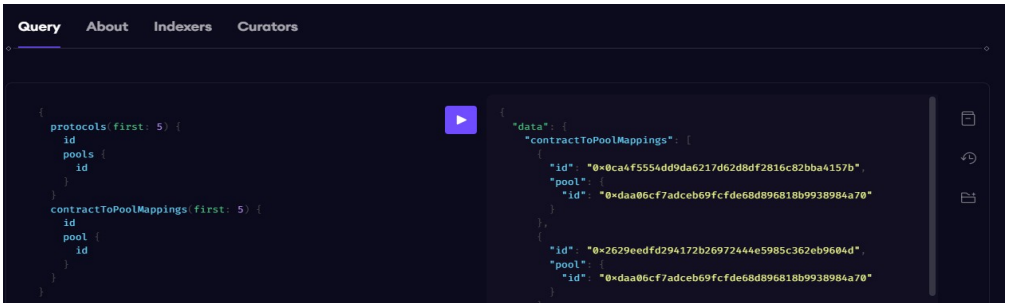
La liste des subgraph RealT : https://thegraph.com/explorer/profile/0x1bcfe8666cbb7edd3eda5d343d5f2d4ce853b034?view=Subgraphs&chain=arbitrum-one



Celui du RMM v3 :

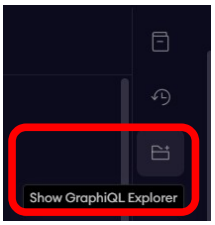


L'interrogation des données se fait au travers de requêtes en GraphQL :
Dans l'explorer (onglet Query) : elles sont écrites sur la gauche, en cliquant sur la flèche central (violette) le résultat apparaît sur la droite.



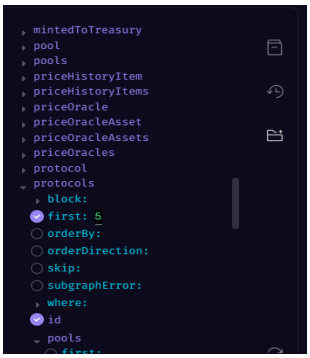
Les résultats sont présentés au format JSON : plutôt adaptés à des programmes, qu'a des humains, quand les résultats sont nombreux..

Pour explorer les données et fabriquer sa requette, vous disposez d'un outils sur la droite :



Vous voyez alors apparaître l'ensemble des données interrogeables (celles cochées correspondent à la requête existante).
La requête se constitue donc au fur et à mesure de ce que vous cochez ou décochez des données souhaitées.

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



Les services The Graph peuvent être accédés : « manuellement » comme décrit ci-avant, mais le plus souvent ce sera via une application.

Cette application peut être un simple tableur, au moyen de script ou de connecteurs :

- L'accès se fait via une url, indiquée dans la page du Subgraph.
- L'accès est maintenant sécurisé par une clé API qui s'obtient avec l'application Studio (https://thegraph.com/studio/apikeys/) et qu'il faudra inclure dans l'url.
- le script convertissant les résultats, du format Json en format tableur.

Vous trouverez un exemple de ce genre de tableur, qui liste les liquidations RMM v2 et v3 :

https://docs.google.com/spreadsheets/d/1K7UY-IYJ-Cs8-YF9oGgCUyBq84Cj7XEYc7mljOPXWK4

Date de la liquidat	Liquidateur	Utilisateur liquidé	Token liquidé	Montant remboursé	Quantité liquidée
2023-06-19	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x4da0ca5cbcf06d41c1935f3a7cfd3a5b608017d96	RealToken S 19041 Lenore Ave Detroit MI	182.78	4.00
2023-06-24	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x835c3e4296fd0cdfa139ed6702e6cd5869bed999	RealToken S 3747 Scovel Pl Detroit MI	12.34	0.27
2023-07-09	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x263c03f9fe227d3529a99ddbe6ae0b098c894ba6	RealToken S 10411-10421 Cadieux Rd Detroit MI	75.91	1.66
2023-07-31	0x46d0fb00d66d66e1a2fec6aeb670d53951c361c4	0xaa59df1422c9010d5d9772567e56004270d711a1	RealToken S 1610 E State Fair Ave Detroit MI	254.10	5.56
2023-08-06	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x6f61dba437d5dc71963817c37e0bef392179fb2d	RealToken S 14918 Joy Rd Detroit MI	226.22	5.06
2023-08-12	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x5c600a0b2db9a362aa3b1a0ea92db61550744d39	RealToken S 618 E 79th St Chicago IL	50.23	1.09
2023-09-08	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x71d5919e7cf762a6b0913ed08f4315965e9c01a1	RealToken S 19751 Marx St Highland Park MI	132.98	2.88
2023-09-15	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0xd21d3fd5cd1e7bb22947338f7f4099b89e33bf09	RealToken S 14918 Joy Rd Detroit MI	4'923.57	109.55
2023-10-01	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x39dbefbae84c6fa5b4aa223e706315a2dbe36920	RealToken S 10617 Hathaway Ave Cleveland OH	4.01	0.08
2023-11-04	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x263c03f9fe227d3529a99ddbe6ae0b098c894ba6	RealToken S 10411-10421 Cadieux Rd Detroit MI	39.00	0.85
2023-11-21	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x1ceb00630377aa16074bcb4de0760631fac53480	RealToken S 618 E 79th St Chicago IL	38.40	0.83
2023-12-05	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x433dd11e3f2c6c6619c729fd78c1ed7042ec5f1e	RealToken S 18980 Fenton St Detroit MI	38.07	0.83
2023-12-15	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x5846fc589dcf5de5683f8d031d57b61fbc63c562	RealToken S 3280 W Boston Blvd Detroit MI	26.00	0.56
2024-01-19	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0xd21d3fd5cd1e7bb22947338f7f4099b89e33bf09	RealToken S 14918 Joy Rd Detroit MI	2'415.29	53.74
2024-02-08	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x9e48c1cdd88b1ef6fb295efbdd1654c3bae68759	Wrapped XDAI	12.27	12.89
2024-03-23	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0xd21d3fd5cd1e7bb22947338f7f4099b89e33bf09	RealToken S 17809 Charest St Detroit MI	1'363.86	26.80
2024-04-01	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x5ecac835ae39dee09689dc86b11c8c79693f58fc	Wrapped XDAI	24.87	26.12
2024-04-02	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x08192acad66610e1656f6728378cb743d1027c4b	Wrapped XDAI	387.96	407.36
2024-04-07	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x1b57fe61c28cec7cfc9a55acff876bd619b9f61d	RealToken S 1610 E State Fair Ave Detroit MI	28.72	0.63
2024-04-21	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x6adce7c7378a70ba4876e462079fc75460af0012	Wrapped XDAI	3.96	4.16
2024-04-28	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x1da85a0625e2576b434eca1fde155558e5078d2a	RealToken S 1630 E State Fair Ave Detroit MI	2'275.88	50.00
2024-05-20	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0xd21d3fd5cd1e7bb22947338f7f4099b89e33bf09	RealToken S 19750 Marx St Highland Park MI	457.30	10.00
2024-05-20	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x16d56e22268eae7d4ec11506a152b5a7fc7ad69b	RealToken S 1630 E State Fair Ave Detroit MI	340.65	7.49
2024-05-26	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x1da85a0625e2576b434eca1fde155558e5078d2a	RealToken S 3747 Scovel Pl Detroit MI	1'003.77	22.00
2024-05-26	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x1b57fe61c28cec7cfc9a55acff876bd619b9f61d	RealToken S 19751 Marx St Highland Park MI	0.01	0.00
2024-05-26	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x4c968a3a650f76f655e7e7010075aa63cec87de2	RealToken S 3280 W Boston Blvd Detroit MI	0.00	0.00
2024-05-26	0xb2cc0d719b71fb91082e050967b644a5e55b33a0	0x4245fbf228968ce8813331ccc63c2b96f5805ff6	RealToken S 618 E 79th St Chicago IL	5.59	0.12
2024-05-26	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0x08192acad66610e1656f6728378cb743d1027c4b	RealToken S 15930 Monica St Detroit MI	0.00	0.00
2024-05-26	0xa22dc341c8dd53ab1dff6e66228e779832a449bf	0xd1b6c36b9f281fcfac5f6939fad7a13b26788af3	RealToken S 18980 Fenton St Detroit MI	0.02	0.00

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 whitelisté : https://docs.google.com/document/d/1spk1WrqdKHILwFRi_yRDf_fbFz3ZYCuZh5g_ni0laQA/edit#heading=h.wl1hkzqboxwj

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).

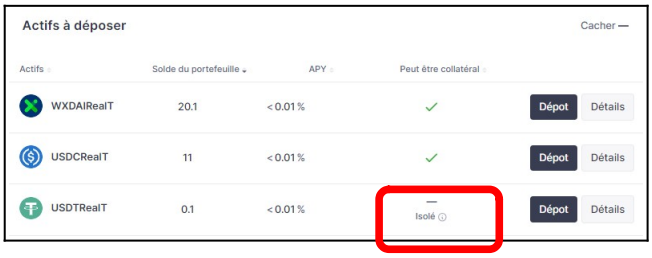
9 – Fonctions RMM v3 non activés

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

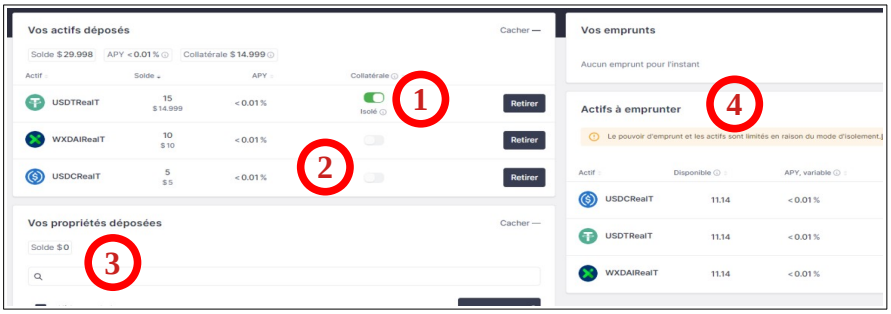
9.1 – Mode isolé

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) :

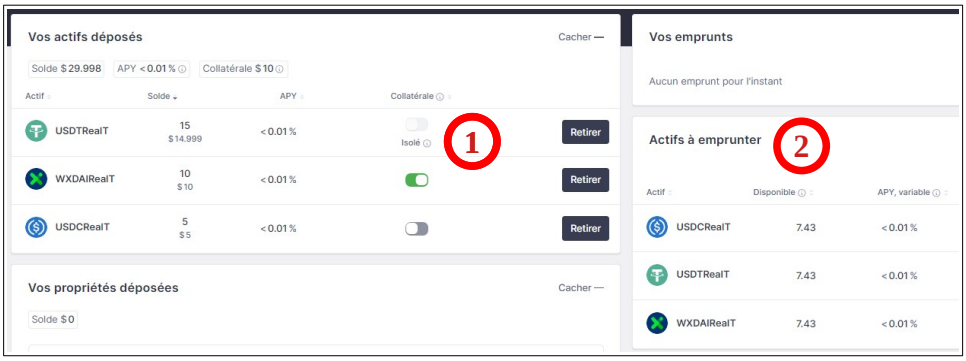


Pour que le mode isolé soit actif (bouton vert) et qu'un dépôt d'USDT puisse servir de collatéral à un emprunt :



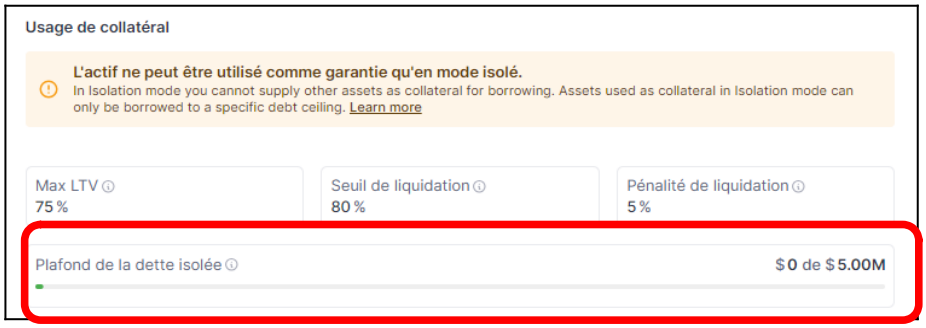
1. il doit être le seul en collatéral ,
2. d'autres Actifs peuvent être en dépôt (pour toucher des intérêts), mais ils ne doivent pas être en collatérale,
3. aucune propriété ne doit être déposée (puisqu'elles sont automatiquement en collatéral),
4. les Actifs empruntables et leur quantité sont limités.

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



1. le mode isolé est inactivable.
   L'UDST en dépôt rapporte des intérêts de dépôt, mais ne peut servir de collatéral à un emprunt,
2. 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 :



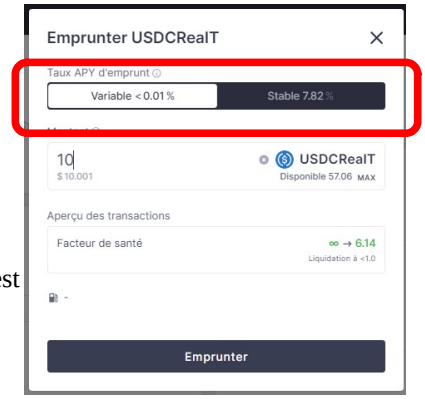
En résumé : Si vous avez déposé des RealTokens, vous ne pouvez activer le mode isolé et l'USDT que vous déposerez ne générera que des intérêts de dépôt et pas de capacité d'emprunt supplémentaire.

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 desactivé.

Comment il pouvait être utilisé :

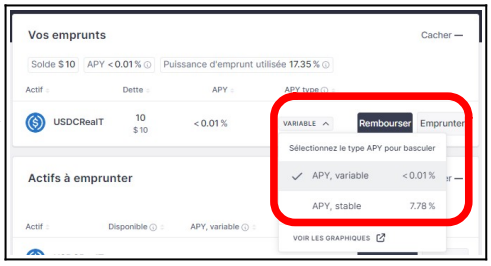
A partir de l'onglet « Tableau de bord », dans la partie « Actifs à emprunter » vous pouvez choisir le type de taux qui sera appliqué à votre emprunt :



- 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.

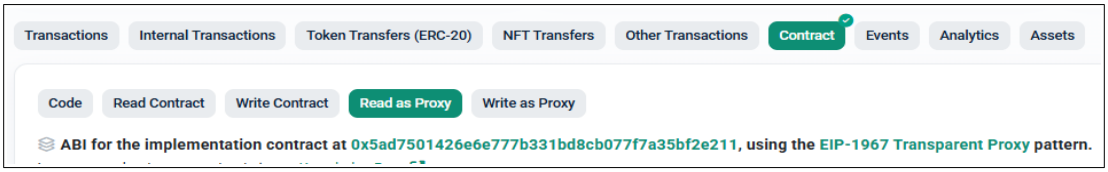
Lorsque vous aurez emprunté, il est possible de changer le type de taux à tout moment :



10 – Code de l'application

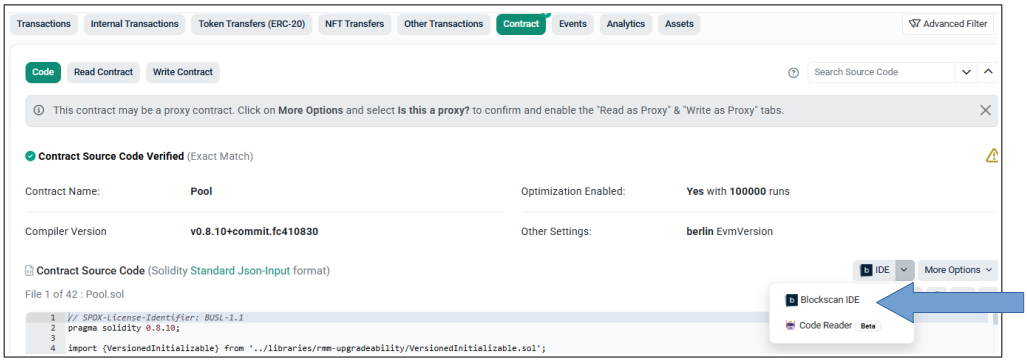
La liste des smart contract est indiqué en bas de la page 6.
Le contrat principale (Pool-Proxy-Realt) du RMM v3 est accessible à l'adresse suivante https://gnosisscan.io/address/0xfb9b496519fca8473fba1af0850b6b8f476bfdb3.

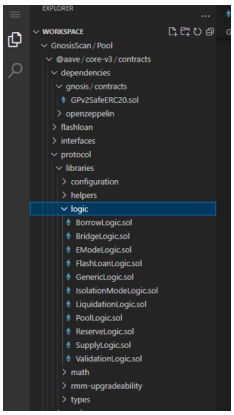
Cette adresse est celle du Proxy, qui pointe vers son implémentation :



https://gnosisscan.io/address/0x5ad7501426e6e777b331bd8cb077f7a35bf2e211#code

Vous pouvez voir le source via un Editeur (IDE) intégré à l'explorateur GnosisScan :





La logique des principales fonctions, se trouve dans le dossier protocol/librairies/logic

Vous y trouverez les différents code Solidity, pour pousser votre analyse...

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