

Le RealToken Wallet, comment ça marche ?

Le *RealToken wallet* est un portefeuille d'actifs numériques, dont les deux principales caractéristiques sont :

- La simplicité d'accès.
Comme l'était le *walletless*, il nécessite pas de compétences en crypto. Et notamment pas de sécurisation de clé privée.
- L'ouverture.
A la différence du *walletless* qui n'était accessible qu'à partir du site RealT.co, ce portefeuille va permettre l'accès à de nombreuses applications sur la blockchain (notamment celles de l'écosystème des RealTokens : YAM et RMM).

Pour développer ce Wallet plus facilement et rapidement, RealT a utilisé des composants d'autres sociétés. Les deux majeurs sont :

- Web3Auth, pour la connexion au Wallet avec les identifiants de comptes sociaux existants,
- Etherspot, pour la création et l'usage du Wallet au standard d'un Abstract Account.

L'Account Abstraction

(sur Ethereum et les blockchains compatibles comme Gnosis)

L'abstraction de compte vise à séparer la gestion du compte (son contenu) et l'exécution de transaction (avec sa clé privée).

Avec un comptes "classiques" (EOA, Externally Owned Accounts), - comme celui qu'on crée avec MetaMask par exemple -, le compte qui contient vos cryptos est associé à sa clé privée, qui est utilisée pour signer vos transactions. Si vous perdez la clé privée du compte, vous perdez les fonds qui y résident.

Les comptes abstraits eux séparent :

- le smart contrat, qui détient les crypto et la programmabilité du wallet,
- du signataire/propriétaire du wallet, qui peut être un compte owner associé, une authentification social, un multi-signataire,..., Permettant la récupération des fonds, si un des moyens de signature est perdu.

Ainsi le compte abstrait réduit le point de vulnérabilité de la clé privée et simplifie les moyens d'authentification avec du social login. La programmabilité du smart contrat apportant bien d'autres avantages (comme des paiements groupés, ou automatisés, ou limités, ..).



Dans le cas du *Realtoken Wallet*, il y a donc deux comptes :

- un smart contrat, qui va détenir vos cryptos, il est créé et géré avec la solution Etherspot,
- un compte "Owner" (EOA) qui permettra l'approbation de vos transactions, il est créé et accédé avec la solution Web3Auth.

Web3Auth (W3A)

(Web3Auth a été racheté par Consensys - Editeur de MetaMask - en juin 2025...)

W3A fourni :

- du code que les développeurs intègrent dans leurs applications (partie W3A Plug n Play du schéma ci-après),

- et un réseau de gestion de clés (partie Auth Network, en bas à droite).

RealT a intégré le code dans l'application [Realt.co](https://realt.co) ainsi que dans la nouvelle application “[Realtoken Wallet](#)”.

Ce code pourra être intégré par la suite aux applications YAM, RMM, ... afin que vous puissiez vous y connecter directement avec votre *RealToken Wallet*.

Pour autoriser une transaction à partir de votre *RealToken Wallet*, vous devez vous connecter au compte Owner associé.

La première étape de la connexion est l'authentification au moyen de “Login Providers”. Ce sont des applications comme Google, Facebook, Twitter, GitHub..., sur lesquelles vous possédez déjà un compte et qui peuvent être utilisées pour vous authentifier.

La seconde étape consiste à reconstruire la clé privée du contrat Owner.

Lors de la création du *RealToken Wallet*, W3A va générer une clé privée (et une adresse publique) qui correspondra au compte Owner.

W3A ne stocke jamais la clé privée complète à un seul endroit. Il la décompose en plusieurs parties (shares) au moyen de techniques de partage de secret cryptographique (SSS, Secret Shamir Sharing), qu'il stocke à différents endroits et qu'il réassemblera ponctuellement lors d'une signature.

Si vous optez pour une sécurisation avec double facteur d'authentification (2FA), il y aura trois shares qui seront distribués :

- un share sur le réseau de W3A et accessible à partir du social login,
- un share sur l'appareil de l'utilisateur,
- et un dernier share pour la récupération à un endroit différent des deux précédents.

Si vous n'avez pas opté pour le 2FA, il n'y aura qu'un seul share (SFA) qui sera géré sur le réseau W3A.

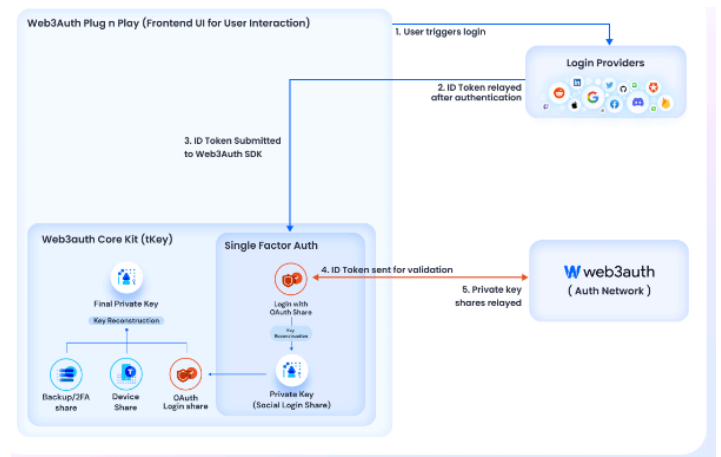
Pour reconstituer la clé privée, le SSS fonctionne avec un seuil : il n'a besoin que de 2 shares sur les 3. Ainsi il est possible de perdre un des shares sans perdre l'accès au compte Owner (et au *RealToken Wallet*). Et inversement celui qui possède qu'un seul share (par ex W3A) ne pourra exécuter une transaction.

Il est aussi possible d'augmenter le nombre de shares et donc le nombre de moyens d'authentification (par ex lors de la connexion avec un nouvel appareil ou navigateur).

Le share qui est sur le réseau d'authentification de W3A est lui-même divisé en neuf parties différentes et distribué sur un réseau de nœuds de stockage d'entreprises et d'institutions géographiquement diverses. Le seuil est ici de 5 sur les 9 pour reconstruire le share chez W3A. Il sera possible à terme que ce réseau soit hébergé par RealT lui-même.

Parmi les différents codes (SDK) proposés par W3A, RealT a retenu pour commencer le SDK Plug n Play (cf schéma ci-dessus), qui comprend notamment la partie interface (écrans de connexion). Progressivement la partie interface de W3A sera remplacée par une interface propre au *RealToken wallet*. Le SDK qui sera alors conservé sera : W3A Core Lit (tKey).

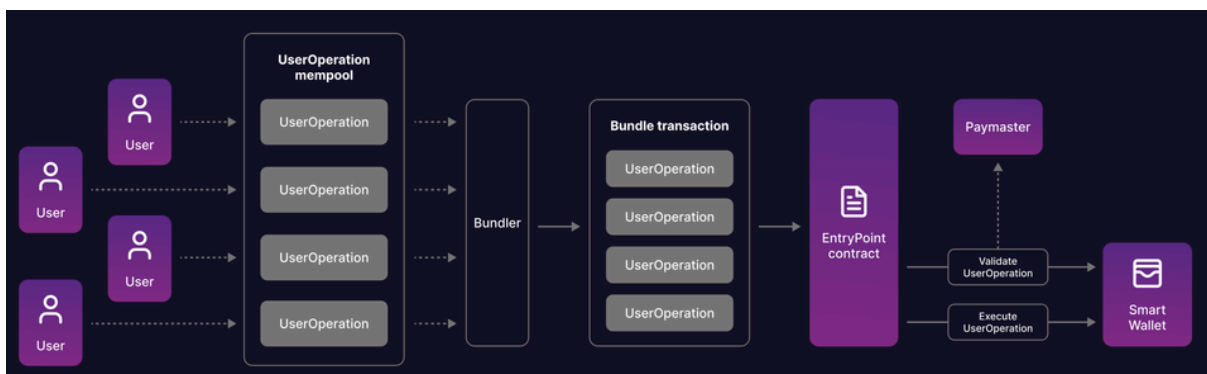
Pour en savoir plus : <https://web3auth.io/docs>



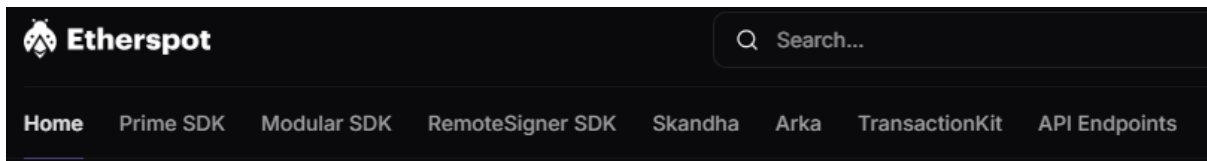
Etherspot

La solution Etherspot fournit les différents composants d'une infrastructure d'Account Abstraction (ERC-4337) :

- Un service Off-chain de "Bundler", qui agit comme un intermédiaire entre l'utilisateur et la blockchain (un peu comme les RPC des wallets EOA). Il regroupe plusieurs opérations utilisateur à partir d'un pool de mémoire pour créer un bundle qui est signé et transmis à la chain sous forme d'une seule transaction.
- On-chain, les composants (smart contrat) pour :
 - l'Entry Point qui traite les groupes de UserOperation,
 - Le Paymaster qui peut éventuellement payer les frais de gaz à la place du wallet (pas utilisé dans notre cas),
 - La Factory qui déploie les nouveaux wallets,
 - et enfin le wallet (modulaire).



Pour en savoir plus : <https://etherspot.fyi/introduction>

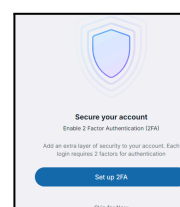
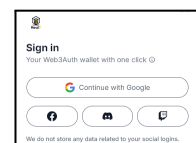


- Le wallet utilisé par RealT est le "Modular SDK",
- Skanda est le Bundler,
- Arka le Paymaster.

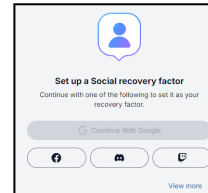
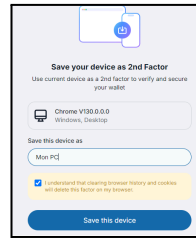
Création du *Realtoken wallet*

La création se fait sur le site Realt.co, au travers des étapes suivantes :

- Un premier écran W3A pour choisir le social login qui sera associé au wallet lors de sa création,
- A l'issue de la connexion, W3A génère la clé privée et l'adresse public du compte Owner, de façon déterministe à partir de l'identifiant du social login,



- W3A propose l'option 2FA, si elle est retenue il sépare la clé en 3 shares (comme évoqué précédemment) :
 - le premier share est stocké sur le réseau W3A et associé au social login utilisé,
 - le second share est stocké sur le matériel utilisateur (navigateur),
 - pour le troisième share de récupération, W3A est paramétré pour proposer un des autres social login que celui déjà utilisé pour l'authentification,



- Si vous n'adoptez pas pour le 2FA ("Skip for Now") en bas de l'écran, vous passez directement à la phase d'initialisation.



L'initialisation du *RealToken Wallet*, comprend les étapes suivantes :

- L'adresse du *RealToken Wallet*, sur Gnosis, est généré par Etherspot à partir des trois éléments suivants
 - l'adresse de la Factory, qui sert pour déployer les comptes abstraits.
 - le Bytecode du smart contrat du compte abstrait qui sera déployé,
 - un Salt : valeur unique dérivée de la clé publique du compte Owner.

Nota : Pour initialiser le *Realtoken wallet* sur une autre blockchain avec la même adresse, les trois informations précédentes devront être identiques (le SDK Etherspot étant initialisé sur la blockchain cible).

A ce stade, l'adresse du futur *Realtoken Wallet* est déterminée, mais le smart contrat du wallet n'est pas encore déployé.

- Realt transfère vers cette adresse 0,1 XDAI pour les frais des transactions qui suivent,
 - Une première transaction (User Operation) est envoyée sur l'Entry Point à destination de l'adresse du *Realtoken wallet*,
 - L'Entry Point détecte que le smart contrat du wallet n'est pas déployé, il demande à la Factory de le faire,
 - La Factory va déployer le smart contrat du wallet à partir d'un modèle (que nous détaillerons plus loin)
 - Une fois le smart contrat du wallet déployé, plusieurs paramétrages vont avoir lieu :
 - Ajout de trois adresses de "Guardian". Ce sont des contrats Safe (Multi-signature) contrôlés par RealT qui permettent certaines récupérations (qui seront détaillés plus loin),
- Nota : Lors du passage en V3, cette partie était manquante et est rajoutée à postériori.
- Ajout du contrat Owner qui contrôlera les transactions effectuées avec le *Realtoken wallet*,
- La création est terminée et l'adresse du *RealToken wallet* est indiquée sur le site Realt.co.

Vous pouvez voir les tâches d'initialisation de votre propre *RealToken Wallet* en allant sur les "internal transactions" :

Transactions Internal Transactions Token Transfers (ERC-20) Contract Events Analytics Multichain Portfolio						
⌵ Latest 3 Internal transactions Download Page Data						
Parent Transaction Hash	Block	Age	From	To	Amount	
0xe4ac77a6c4...	34421962	131 days ago	0xb2989354...7b3f8e5aD	ERC-4337: Entry Point ...	0.0008066 xDAI	
0xe4ac77a6c4...	34421962	131 days ago	0x7fd8f10...A6a05d40E	Contract Creation	0 xDAI	
0x4e12d757fc9...	34421836	131 days ago	0xA44947FF...41a8fdF0	0xb2989354...7b3f8e5aD	0.1 xDAI	

Smart contrat du *RealToken Wallet*

Les adresses des différents composants de la solution Etherspot, qui sont déployés sur Gnosis figure ici :

https://github.com/etherspot/etherspot-prime-contracts/blob/master/DEPLOYMENTS_MODULAR.md

On y trouve l'adresse (d'implémentation) du smart contrat du *RealToken wallet* (ModularEtherspotWallet) :

<https://vscode.blockscan.com/gnosis/0x202a5598bdba2ce62bffa13ecccb04969719fad9>

Ce modèle de wallet, très récent chez Etherspot (sept. 2024), est déployé par RealT en tant que Version 3 du *RealToken Wallet*. L'avantage majeur de cette version est sa modularité (en conformité avec le standard ERC-7579). Il sera ainsi possible d'ajouter de nouvelles fonctionnalités sans avoir à créer un nouveau contrat/adresse (comme cela a été nécessaire pour passer de la v1/2 à la v3). Etherspot développe par ailleurs un "Modular Store" qui sera disponible sur la plateforme [PillarX](#). Ce magasin permettra :

- Aux utilisateurs d'installer des modules comme des add-ons pour leurs comptes abstraits,
- Aux dApps d'utiliser ces modules pour améliorer l'expérience utilisateur.

Les principales fonctions du smart contrat sont :

- La gestion des modules (validateurs, exécuteurs, gestionnaires de fallback, et hooks).
- La gestion des Owner et Guardians
- La validation de UserOp. et Signataire, et l'exécution de l'UserOp

Le Owner à un contrôle total sur le wallet :

- Son Initialisation,
- L'Installation/désinstallation de modules,
- L'ajout/retrait d'Owner et Guardians,
- L'exécution de transactions (UserOp) et les validations associées

Les Guardians ont des droits moindres :

- Il ne peuvent que proposer un nouveau Owner, proposition qui ne sera exécutée que si un quorum de Guardians est atteint,
- Ils ne peuvent valider et exécuter aucune transaction.

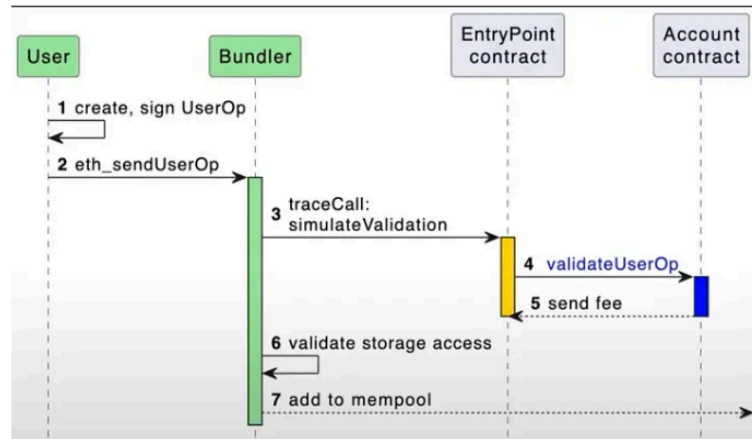
Exécution des transactions avec un Abstract Account

L'initialisation et l'exécution d'une transaction avec un Abstract Account est assez différente qu'avec un wallet classique (Metamask par ex).

Comme vous le voyez dans le schéma :

Les deux premières étapes sont off-chain:
1 - L'application (Realt.co ou celle du wallet), créer une "UserOperation" avec un format spécifique.
2,3 - Le bundler groupe les UserOp et les transmet à l'Entry Point sur la Blockchain.

Les étapes suivantes sont on-chain :
4,5 - L'entry Point soumet la UserOp pour validation et exécution au smart contrat du wallet.



Les applications comme MetaMask ou Rabby ne peuvent être utilisées pour un Abstract Account, d'où la création par RealT d'une application spécifique ([RealToken Wallet](#)) qui joue le rôle de wallet. Les fonctions actuellement disponibles sur ce wallet sont rudimentaires, et se développeront progressivement.

Récupération

Comme nous l'avons vu précédemment :

- W3A associe votre social login et la clé privé du compte Owner (de votre *RealToken Wallet*),
- W3A propose un 2FA pour sécuriser le social login,
- Le compte Owner autorise les transactions du *Realtoken Wallet*,
- Les Guardians peuvent modifier le compte Owner.

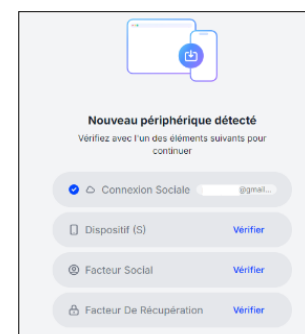
Voyons maintenant les différents incidents qui peuvent se produire et le traitement associé :

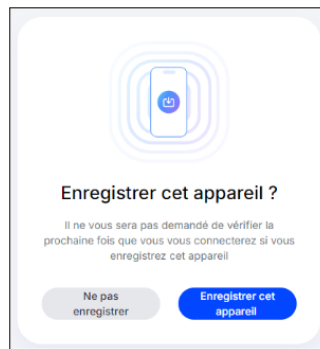
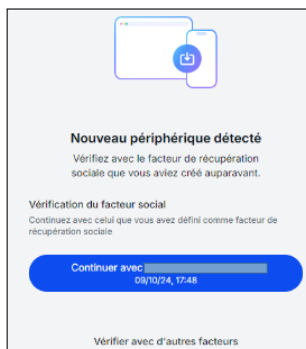
1. Le cas de base est la perte du social login :
 - Actuellement la seule solution est : de recréer un nouveau compte Owner avec le nouveau social login, puis de changer le compte Owner de votre *RealToken Wallet* à l'aide des Guardians. Actions possibles uniquement par RealT.
 - A l'avenir, un interface de récupération de compte pourra être inclus dans les applications RealT afin de se servir du 2FA pour modifier le login associé au compte owner existant.
2. Si vous avez activé le 2FA et que vous changez de matériel ou de navigateur (ou que vous effacez les cookies, faite une mise à jour, ..)

W3A ne va pas trouver le second share du device et va donc demander le troisième share de récupération.

Allez dans "Vérifier" le "Facteur Social" et authentifiez vous.

W3A vous proposera pour terminer d'enregistrer ce nouvel "appareil".



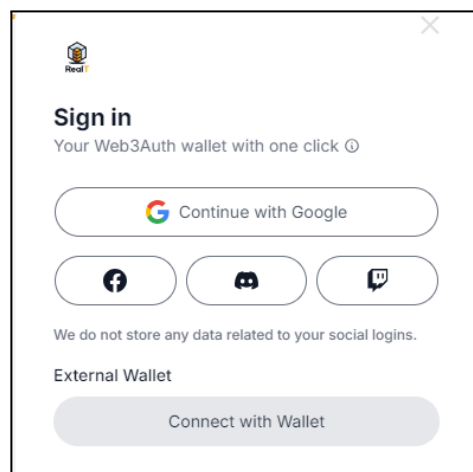
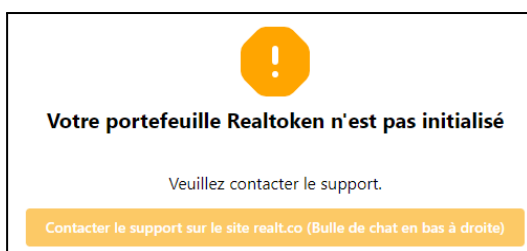


3. Si vous avez perdu le facteur social de récupération, Seule la solution décrite en 1 est actuellement disponible,
4. Si vous souhaitez récupérer l'entière propriété de votre *RealToken wallet* (sans dépendance vis à vis de RealT). Il faudra que l'application *RealToken Wallet* permette de changer les compte guardians gérés par RealT par vos propres comptes.
L'objectif du *RealToken wallet* étant d'éviter la complexité d'un EOA, là vous en aurez trois à gérer !..

L'application RealToken Wallet

Lors de la connexion avec [l'application](#), un champ vous a peut être intrigué .."Connect with wallet".

Lorsque vous cliquez dessus, il vous propose de vous connecter avec un compte classique (via MetaMask, Rabby, ..) et généralement vous aboutissez à l'erreur suivante :



Cette fonction d'external wallet permet d'avoir un *RealToken Wallet* auquel on se connecte avec un wallet classique, sans passer par un social login. Option plutôt réservée à des personnes expérimentées.

Dans ce cas, pour créer le *RealToken wallet* associé à ce wallet externe; les tâches d'initialisation (évoquées dans le chapitre de création du *RealToken Wallet*), doivent être réalisées par RealT (d'où le message d'erreur lorsqu'elle n'ont pas été faites et le renvoi vers le support RealT).

Nota : avec une telle solution, les crypto qui sont sur le wallet externe ne sont pas pour autant sur le *RealToken wallet* (puisque comme expliqué au début, les actifs d'un abstract account se situent sur son smart contrat).

Ce wallet "externe" correspond en fait au compte Owner évoqué précédemment. Si vous connaissez son adresse publique, il est donc possible de vous connecter à votre *RealToken Wallet* par ce moyen..

L'adresse du compte Owner d'un *RealToken wallet* n'est pas diffusée, il est néanmoins possible de la trouver en allant dans la console de votre navigateur. Elle apparaît lors de la connexion.

```
script: account is connecting  
got ownerwallet address 0xf0f  
account not deployed yet  
account: 0xaB9  
web3auth listeners off
```

PhilP - 21 Oct. 2024