Composant 3 : blockchain

Auteurs

Louis Ledoux

Théo Chennebault

Nikola Spaci

Tom Moore

Do Thanh Long LE

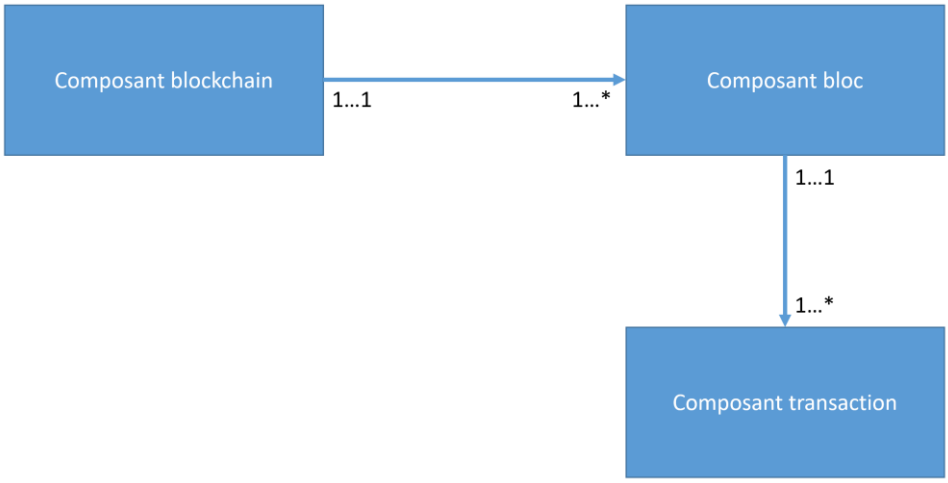
Version :

1.0.0

Description :

Ce composant interagit avec le composant bloc. En effet, la blockchain est une liste chainée ordonnée de blocs. Ainsi, le composant blockchain encapsule le composant bloc. Il permet également de sauvegarder la blockchain au format Json. Enfin, il doit pouvoir charger la blockchain à partir d'un fichier Json. Il est à noter que le composant bloc encapsule le composant transaction. Ainsi, lors du mapping au format Json, une dépendance à ce composant est ajoutée.

Schéma de dépendances aux autres composants :



Signature méthode Python :

saveAsJson() : Sauvegarde l'ensemble de la structure blockchain (ensemble des objets blocs de liste chainée)

importFromJson(json) : Récupération de l'ensemble de la blockchain et stockage dans une liste de bloc (blockchain)

getNbBloc() : Retourne un entier correspondant au nombre de bloc dans la liste chainéé de la blockchain

getBloc(index) : Récupère le bloc à la position i de la blockchain

addBloc(bloc) : Ajoute un bloc dans la liste blockchain

Tests :

Les premiers tests à réaliser consistent à s'assurer que les données issues d'un objet blockchain sérialiser au format JSON peut être désérialisé et stocké au sein d'un objet blockchain (ensemble des blocs). Il faut s'assurer que chaque attribut du bloc soit bien chargé, même quand l'attribut est associé à une autre classe. Par exemple comme c'est le cas pour les champs tx0 qui a un TXM ainsi que pour la liste de TX (txs). Pour s'assurer que le chargement est opérationnel, on réalise un mock en JSON d'une blockchain. C'est ce mock qui sera utilisé dans la phase de test. On s'assure ensuite que la sérialisation est correcte en sauvegardant les différents blocs de la blockchain dans un JSON, on s'assurera que ce nouveau JSON est identique au Mock JSON précédemment utilisé.

Ensuite on poursuivra sur des tests autour de la gestion de la blockchain, c'est à dire l'ajout, la lecture d'un bloc au sein de la liste blockChain. En premier lieu il faut tester la lecture, pour se faire on va encore se baser sur notre mock JSON que l'on va désérialiser et on va ensuite parcourir la liste bockchain pour vérifier que la méthode getBloc en charge de la lecture d'un bloc nous retourne bien chaque bloc. Lorsque la lecture d'un bloc est fonctionnelle, on va s'assurer du fonctionnement de la méthode d'ajout en l'appelant puis en vérifiant que le bloc ajouté à la fin de la liste blockchain est bien identique au bloc précédemment ajouté.