Composant 5

RASOLOHERY Tianome

DJEUKAM YOUDJEU Richelle

HILALY Nawfal

NAJI Mohammed Hamza

VERSION 1

**Description**

Contexte

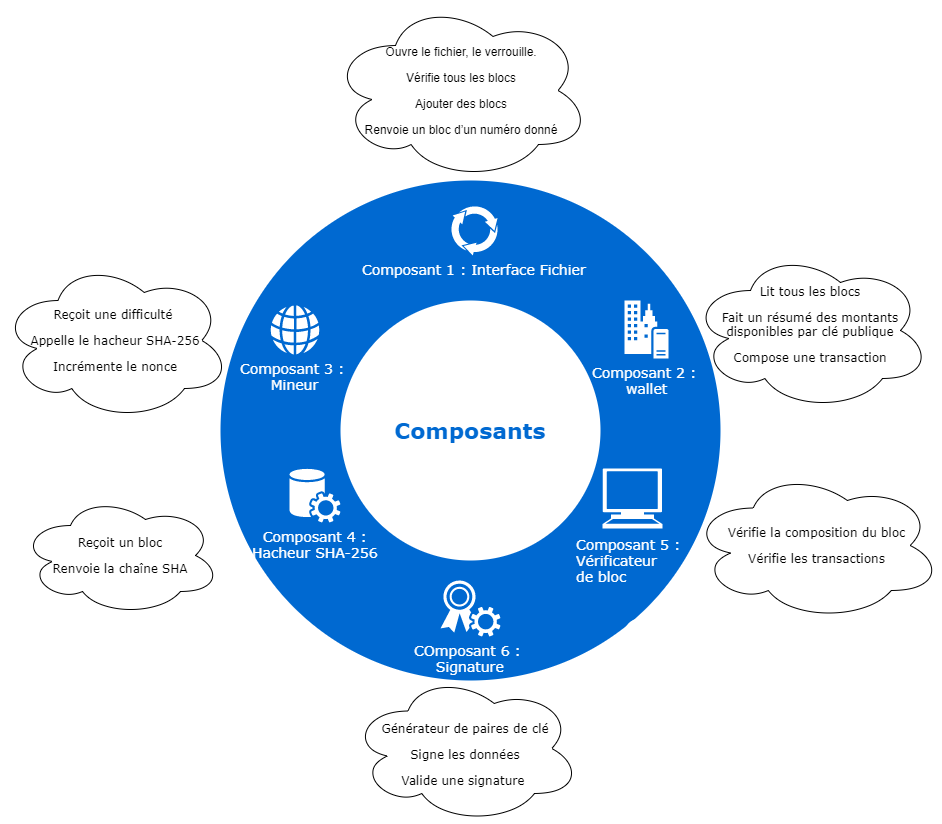
La blockchain est une technologie de stockage et de transmission d’informations, transparente, sécurisée, et fonctionnant sans aucun organe central de contrôle. Ce projet vise donc à mettre sur place une blockchain locale à travers la réalisation d’un ensemble de composants qui permettront l’acheminement sécurisée et complète d’une transaction.

Lorsqu’un hôte A a effectué une transaction vers un hôte B, celle-ci va être acheminée dans un bloc; Ce bloc va être validé par les noeuds du réseau; une fois le bloc validé, il est ainsi ajouté à la blockchain et la transaction ainsi générée sera visible au destinataire et à l’ensemble l’ensemble du réseau.

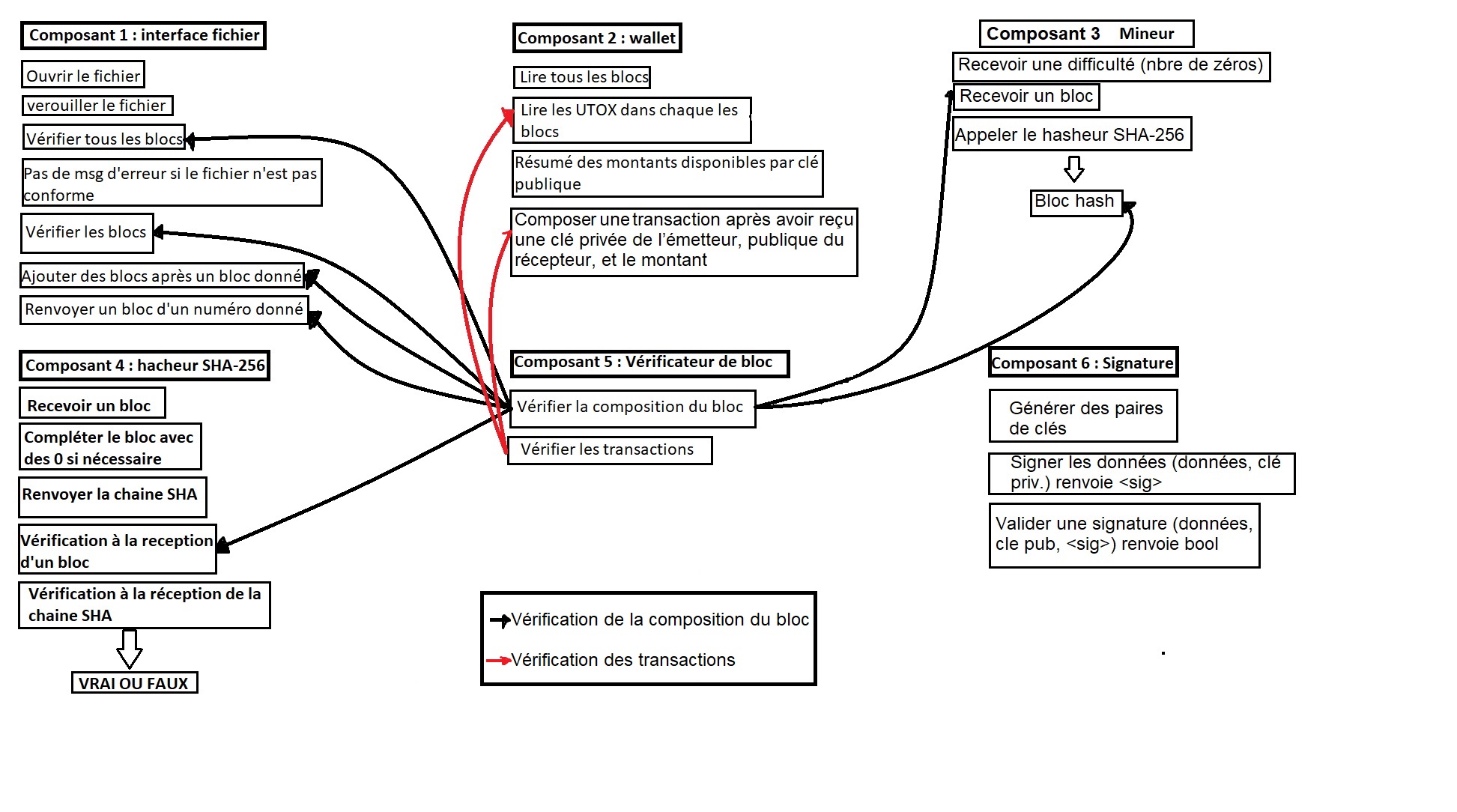
Afin d’assurer la sécurité au sein de la blockchain, chaque bloc et chaque transaction doivent vérifier certaines conditions de validation.

Ce document illustre ainsi les différentes spécifications ou encore les différentes étapes de vérification par lesquelles chaque transaction et chaque bloc devront passer afin d'être validés et par la suite être acheminés dans la blockchain.

Schéma bloc incluant les composants connexes



Interface et interaction avec chaque autre composant



Concernant l’interaction du composant 5 avec les autres composants, nous avons remarqué que ce dernier interagit avec les composants 1, 2, 3, 4.

En effet, nous avons déduit que dès qu’un bloc est créé il faudra que la vérification intervienne afin d’éviter de propager des informations erronées.

La partie vérification des transactions intervient quand à elle à la création ou la lecture d’une transaction.

Résumé (headers)

Dans cette partie, on va présenter les signatures des classes et des fonctions qui devront être implémentées pour la validation d’un bloc et la vérification d’une transaction.

Valider l'état d’un bloc

Un bloc est constitué de:

. Numéro du bloc

. Hash du bloc précédent

. Nonce

. Transactions

. Gain du mineur

Par conséquent, l’implémentation de la classe “Bloc” contiendra les attributs cité-ci dessus et la fonction vérification d’un bloc sera définit comme suit:

* Signature de la méthode: bool verify\_bloc(Bloc\* b);

Ainsi, cette fonction “verify\_bloc” vérifiera que la version du bloc est supérieure ou égale à la version du bloc dans l'état blockchain; sinon, elle s’arrête et renverra false. En d’autres termes elle doit vérifier que le bloc actuel commence par un nombre minimum de zéros.

Valider l'entrée de la transaction

En ce qui concerne les transactions, on a des entrées TXI (Transaction Input) caractérisées par:

. Nombre d’entrées

pour chaque entrée on a:

. Numéro de bloc

. Numéro de UTXO

. Signature de la UTXO

On a également des sorties UTXO (Unspent Transaction Output) caractérisées par:

. Nombre de sorties

pour chaque sortie on a:

. Montant

. Destinataire

Une transaction est dite *valide* par rapport à un état de blockchain particulier si elle est bien formée et si les sorties qu'elle tente de dépenser existent dans l'état, et elle satisfait les prédicats dans ces sorties.

* Signature de la méthode: bool validate\_entry(UTXI\* ui);

Cette fonction vérifie si les entrées (UTXI) que l’on reçoit sont valides.

Vérifier que la transaction est bien formée

* Signature de la méthode: bool retrieve\_utxo(UTXI ui);

Dans cette première étape, on cherche à savoir si la transaction peut être analysée en tant que structure de données de transaction c’est à dire si on peut retrouver la chaîne UTXO correspondante.

Deuxièmement, on vérifie si notre UTXI n’a pas été déjà référencée par une autre UTXI dans un bloc de numéro inférieur.

* Signature de la méthode: bool verify\_referenced(UTXI ui);

Et à la fin si les deux étapes de vérifications ont réussi, on décrypte la signature accompagnant UTXI à l’aide de la clé publique contenue dans la UTXO pour retrouver le hash de la UTXO. Ainsi, la fonction de décryptage peut réussir (retourner la chaine UTXO) ou échoué (retourner NULL)

Note: Hash de UTXO: montant, numéro de bloc, numéro de transac dans le bloc

* Signature de la méthode: UTXO decrypt\_signature(UTXI ui) ;