Document de spécifications du composant 3

Auteurs: LEPERS Antoine, XU Victor, MEGHARA Lyes, LAKHDARI Adlane Badr

HISTORIQUE DES VERSIONS

Dernière modification le :	Par :	Objet de la modification :	Version
18.02.2019	Antoine LEPERS	Création du document	1.0
20.03.2019	Antoine LEPERS	Première version	1.1

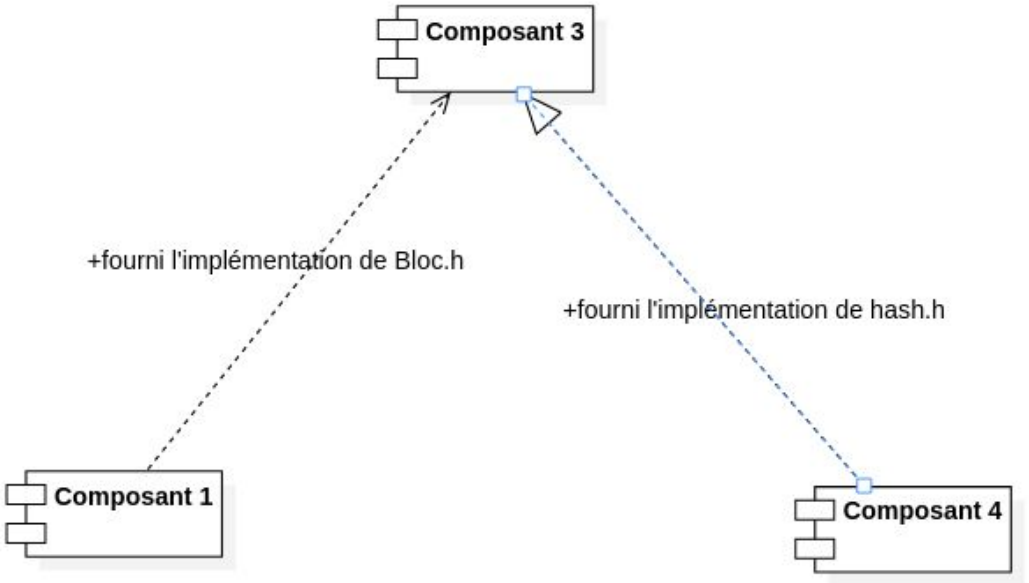
Contexte

La sécurité de la plupart des cryptomonnaies repose en grande partie sur des hashs. Le hash de chaque bloc de la blockchain doit contenir un certain nombre de zéros. Et comme les blocs sont liés (d'où le nom de blockchain), le hash de chaque bloc dépend du bloc précédent. Ainsi altérer un bloc est quasiment impossible car cela implique de recalculer tous les blocs suivant le bloc altéré pour que leurs hashs gardent le bon nombre de zéros.

Nous allons ici décrire le fonctionnement du composant 3. C'est le mineur. Il prend en entré un bloc et une nombre de zéros et renvoie le bloc en ayant modifié le nonce pour que le hash comporte le bon nombre de zéros.

Ce composant fait appel à deux autres composant. Le composant 1 qui définit notamment la classe Bloc. Et le composant 4 qui définit la fonction hash.

Schéma des interactions avec les composants annexes



Résumé header

Bloc.h: défini les la classe bloc de M. Luu

hash.h: défini la fonction string hash(Bloc b) qui pour un bloc donné renvoie un string de 63 caractères

Mineur.h: défini deux fonctions. Une fonction int nbZerosCounter(std::string str) qui renvoie le nombre de zéros à la fin d'un string. Et une fonction Bloc miner(unsigned int difficulty, Bloc b) qui fait une boucle jusqu'à trouver un hash terminant par un nombre de zéros supérieur au paramètre difficulty.

Test

Pour les tests nous avons implémentés les autres headers dans dans fichiers appelés Dummy. Nous avons donc un DummyBloc.cpp et un DummyHash.cpp.

DummyBloc.cpp implémente le builder d'un bloc en initialisant le nonce à 0. DummyHash.cpp implémente la fonction de hash. Elle renvoie un string avec 63 zéros (le 64e caractère étant le 0 de fin de tableau).

Nous avons ensuite écrit un programme de test appelé tester.cpp. Il crée un bloc et appelle la fonction miner avec en paramètre le bloc et 4 comme nombre de zéros exigés.

Le nombre de zéros renvoyé par la fonction de hash étant 63 une seule boucle suffit au programme pour trouver un hash correct.