SPECIFICATIONS COMPOSANT TRANSACTION.H

Antoine PERRIN, Ahmed AHOUILI, Abdelouheb MOUHOUBI, Hang LI, Aymeric ROINEL

[NOM DE LA SOCIETE] [Adresse de la société]

Contexte.

La transaction est à la base de la blockchain et donc des cryptomonnaies.

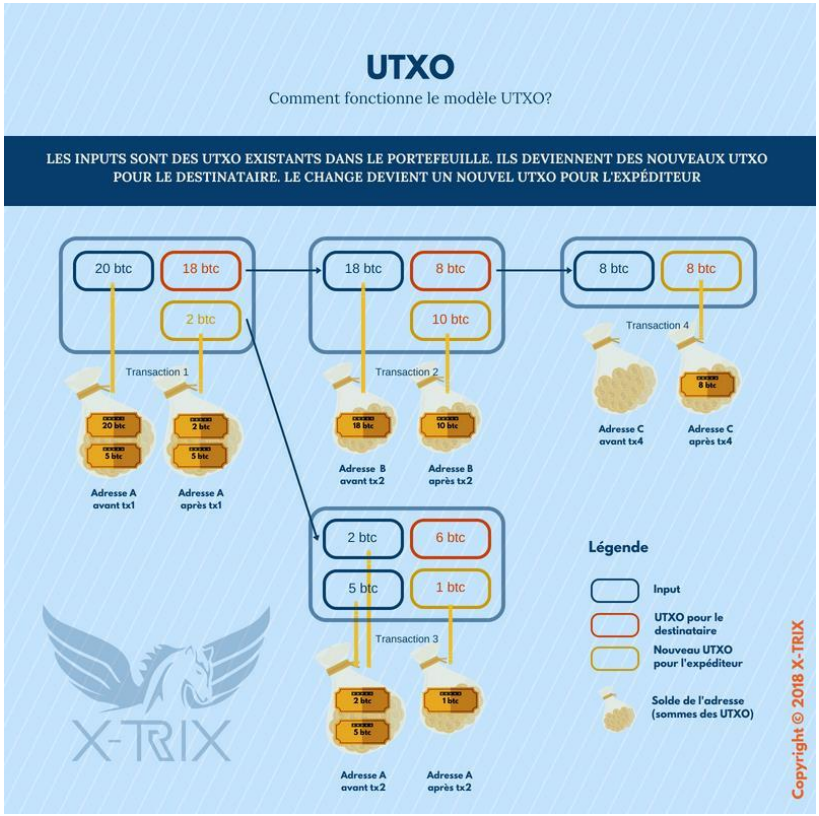
Il y a deux objets au cœur de chaque transaction : UTXO (Unspent Transaction Output) et TXID (Transaction ID).

UTXO : L'objet le plus important, car il porte les informations clés dans une transaction (montant et destinataire de ce montant).

Dans ce modèle (contrairement au modèle *Account/Balance model*), chaque transaction a pour origine un output d'une transaction antérieure et génère à son tour de nouveaux outputs. Les unspent transactions (non dépensées) sont conservées tout au long des blocs. Le portefeuille électronique garde la trace de toutes les UTXO associées aux adresses qu'il possède. Lorsque l'on consulte son solde de bitcoins, le logiciel réalise en fait un calcul : il somme tous les UTXO présents afin d'afficher le solde du portefeuille.

Une analogie facile à comprendre pour visualiser le fonctionnement de ce système serait avec des billets. Dans cet exemple un UTXO représenterait un billet, billet qui ne peut être utilisé qu'une fois. Imaginons que notre portefeuille soit de 20 BTC, et que ces 20 BTC soient « stockés » dans un seul UTXO/« billet ». Nous souhaitons maintenant donner 10 BTC à un ami. Pour cela, le système va créer 2 UTXO de 10 BTC chacun à partir de l'UTXO de 20 BTC, puis marquer l'UTXO de 20 BTC comme *spent*. Ensuite les deux UTXO de 10 BTC seront chacun distribués (c'est-à-dire que l'attribut destinataire sera attribué), un à notre ami, et un pour nous.

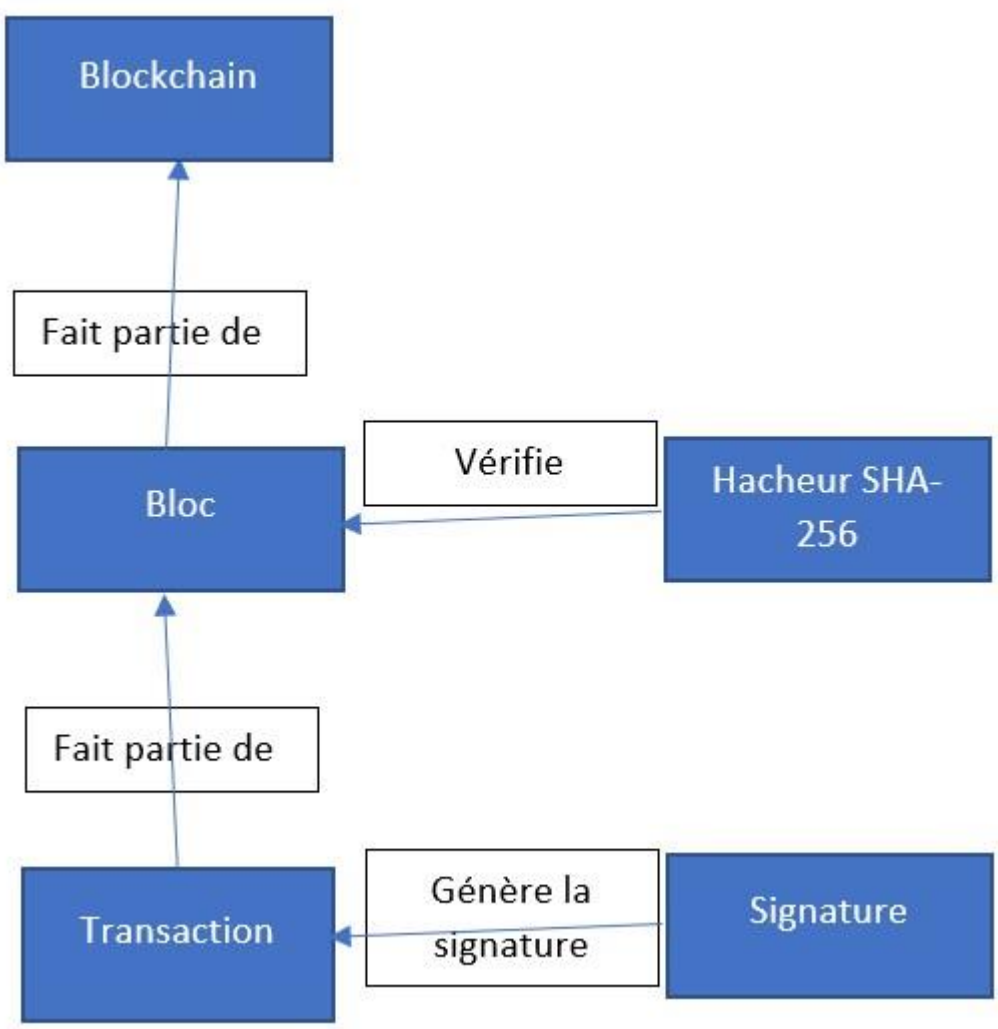
Un exemple avec un schéma :



TXID : C'est l'empreinte cryptographique d'une transaction. C'est en quelque sorte une « étiquette » associée à chaque UTXO, avec des informations cette fois-ci de type administrative.

Notre travail ici sera donc de mettre en place ce processus de génération d'UTXO à partir des inputs fournis.

Interface et interaction avec chaque autre composant.



Fonctions :

Classe TX :

- Constructeur à partir de json (le json contient les données complètes de la transaction, on ne fait que recopier)
- to_json() : renvoie l'objet TX sous format json (seulement de la recopie)
- faireTransaction(json) : fais une transaction (en calculant notamment la signature de la transaction) à partir d'un fichier json contenant les TXI et UTXO d'imputs, et les couples montant-destinataire
- verifierTransaction() : vérifie qu'une transaction est correcte à partir de sa signature

Classe TXM :

- Constructeur à partir de json (le json contient les données complètes de la transaction, on ne fait que recopier)
- to_json() : renvoie l'objet TXM sous format json (seulement de la recopie)

Test.

- Vérifier que les UTXO en outputs sont bien inférieurs à la somme en inputs