

# Projet Systèmes Distribués

Le projet de systèmes distribués avait pour but de simuler une blockchain. L'implémentation proposée est en Java et la bibliothèque de programmation distribué est RMI. Le sujet était passionnant, les lectures annexes étaient très intéressante et expliqué très bien la blockchain. Par rapport à la soutenance, j'ai modifié quelques problèmes et ajouter d'autres options.

Ma blockchain représente le Remote Object dans mon implémentation et permet d'être récupérer très facilement et de pouvoir être modifier. Un autre Remote Object est présent pour les serveurs afin qu'ils puissent se connecter entre eux et faciliter la modification des changements entre client/serveurs. Les clients représentent les nœuds participants et les serveurs sont les nœuds blocs récupérant la blockchain grâce à RMI. Le mérite est également présent avec un compte (d'entiers) qui augmente lorsque le client mine un bloc (il reçoit le même mérite, il n'augmente ou ne diminue pas en fonction de la durée de son minage) ou que des transactions sont effectués entre les nœuds participants. Un nœud participant peut changer de nœud bloc si il le souhaite. Les jeux de tests ne sont pas fournis, car un menu « interactif » est présent pour le serveur et le client afin que l'utilisateur puisse faire ce qu'il veut.

Malheureusement, l'implémentation n'est pas totalement respecté comme dans le sujet, car plusieurs points ne me paraissaient pas logique à faire tel que la transmission de certains blocs à d'autres blocs participants ou encore le mérite qui n'est pas le même en fonction du travail du client.

J'ai voulu implémenter une grosse sécurité avec DSA en utilisant la bibliothèque de BouncyCastle, mais j'ai rencontré énormément de soucis et j'ai repris le projet à zéro en choisissant de garder seulement certains aspects de la sécurité que je voulais mettre. En revanche, il y a tout de même une certaine sécurité grâce à l'utilisation de clés RSA permettant la génération d'une signature et sa vérification lors d'une transaction.

Pour lancer un serveur après avoir compiler ( ./compile ), il suffit de lancer : ./go\_serveur <numéro port>. On fera de même pour lancer un client avec : ./go\_client <numéro port>. J'ai pris la décision de rester avec une adresse locale afin de faciliter l'utilisation et les tests, mais les adresses globales fonctionnent également : en effet, j'ai testé avec mon adresse IP de mon box et avec Turing, tout fonctionnait très bien. Pour la suite de l'utilisation, l'affichage sur le terminal permet de faire toutes les demandes de l'utilisateur et d'afficher la blockchain sur le terminal.

En conclusion, le projet était très intéressant, sur un sujet d'actualité. J'aurais voulu que la sécurité fonctionne comme je le souhaitais, car j'ai perdu beaucoup de temps dessus et j'espère travailler dedans un jour, mais les erreurs ont été trop nombreuses. Le projet m'a surtout appris la notion de blockchain et comment une blockchain fonctionne.