Compte rendu Mercredi 8 février 2023

BROUILLET Arthur, POIRIER Louis, FROEHLY Jean-Baptiste

1. Présentation du projet

L'idée étant de regrouper différents éléments au programme, nous avons choisi de trouver un projet incluant de la POO en python, de l'algo python classique et faire le lien avec une interface web-based.

La première idée étant d'implémenter une blockchain, nous nous étions donc orienté sur le projet de faire une pseudo crypto monnaie. Cependant, le projet était un peu trop logique vis à vis du thème choisi. L'idée a donc été de trouver un autre projet qui, basé sur la blockchain, soit moins commun. Nous nous sommes donc rappelé de l'histoire des votes en ligne et du manque de sécurité de ces derniers et donc de les baser sur le principe de blockchain afin de rendre le tout inviolable.

L'idée est donc de faire une plateforme de vote en ligne qui, suivant la technologie de la blockchain, permet à l'utilisateur de voter et selon le vote, créer un bloc qui sera ajouté à la chaine et la base du vote suivant le tout en montrant à chaque fois les différentes preuves d'inviolblité de la chaine et rendre le tout user-friendly et fonctionnel.

1.1 Choix des technologies

Pour ce qui est de la blockchain en elle même, nous aurions pu utiliser lune lib dédiée mais cela aurait rendu le projet moins intéressant. Nous avons donc décidé d'implémenter nous même la blockchain en python en orienté objet.

Pour ce qui est de la liaison python-web, nous avons utilisé le framework Flask qui permet de faire une interface web en python. De plus étant entièrement structuré en orienté objet, manipuler ce framework permet de consolider nos connaissances en POO.

Finalement ce projet apport également des bases cryptographiques et notamment de chiffrement et de hashage.

Le travail

A l'état actuel, n'étant pas tous encore trop disponibles, nous avons pu commencer une version alpha de blockchain qui est fonctionnelle mais améliorable et une strucure flask découpée en différentes parties permettant de gérer les user-side et les admin-side avec leurs chemins et autorisations respectives. Il est aussi question de rajouter une petite db pour stocker les utilisateurs et les votes de ceux-ci pour en retirer des statistiques mais cela restera à déterminer en fonction de l'avancement global du projet. Il reste donc encore à faire un joli front-end adapté avec toutes les options, et mécaniques possibles, relier cet ensemble au programme et à ses données tout en améliorant notre algo blockhain de base.

```
Idée générale de la structure du projet sous flask

├──project
│   ├──admin
│   │   ├──__init__.py
│   │   └──routes.py
```

```
│   ├──main
│   │   ├──__init__.py
│   │   └──routes.py
│   ├──static
│   │   ├──css
│   │   ├──images
│   │   └──js
│   └──templates
│       └──(différentes pages html)
├──data.db
└──start.py
```

Nous avons fait un hashage en SHA256 (par protocole et précision ce qui nous donne des empreintes numériques de 64 caractères héxadécimaux contre 32 que l'on trouverait en md5) et le système de vérifcation d'intégrité de la chaîne se base simplement sur la vérification de correspondance entre le hash précédent d'un bloc, permettant d'obtenir le hash actuel, et le hash dit "actuel" du bloc précédent.

Le travail pourra réellement commencer lorsque nous serons tous disponibles et bien au courant des ce qu'il y a exactement à faire, par qui et comment.