

Sujet de projet

- Réaliser l'équivalent d'Uber qui respecte la vie privée, en utilisant une blockchain (i.e., sans intermédiaire de confiance).

Blocs de base

- Blockchain
 - Identifiants
 - Demande de trajet
 - Acceptation d'un trajet
 - Validation d'un trajet effectué
 - Mining
 - Choix de la branche
- Réseau:
 - Découverte de noeuds
 - Routage des messages entre les noeuds
 - Broadcast de messages
- Stockage
 - Stockage des données
 - Récupération des données en cas de litige

Methodologie

- Groupes de 4, un scrum master
- Sprints de 3h
 - 10 premières de réunion de groupe
 - A la fin d'un sprint un deliverable fonctionnel est commité sur le git
 - A chaque nouveau sprint on change les groupes.
- Présentation des résultats et décision des tâches à faire au CM suivant.

Envoi du résultat du sprint

- Mettez vos documents dans une branche à vous (que vous nommerez `<spring_name_group>`)
- Générer un bundle:
 - `$ git bundle create <sprint_name_group>.git --all`
- Envoyer le fichier `.git` à `damien.saucez@inria.fr` et le mettre à disposition des autres étudiants sur la plateforme du CERI.

Note

- Chaque étudiant•e participe donc à 4 sprints.
- Chaque sprint est noté sur 5 sur base de son livrable.
- Les points d'un•e étudiant•e correspondent à la somme des points des sprints auxquels il•elle a participé.

Informations complémentaires

Git?

- <http://rogerdudler.github.io/git-guide/index.fr.html>
- Merger <project-a> dans <project-b>?
 - \$ cd </path/project-b/>
 - \$ git remote add project-remote </path/project-a>
 - \$ git fetch project-remote
 - \$ git merge --allow-unrelated-histories project-remote/<branch>
 - \$ git remote remove project-remote

Flask?

- <https://pythonspot.com/flask-web-app-with-python/>
- Exemple minimaliste
 - <https://github.com/dsaucez/avignon18/blob/master/src/server.py>

Requests?

- <http://docs.python-requests.org/en/master/user/quickstart/>
- Exemple minimaliste
 - <https://github.com/dsaucez/avignon18/blob/master/src/requests.py>