User Stories :

US 001000-0010 « En tant que visiteur, je veux pouvoir me connecter à ‘monCoffre’ »
US 002000-0010 « En tant que visiteur, je veux pouvoir me créer un compte »
US 003100-0010 « En tant que visiteur, je veux pouvoir créer une base de données locale »
US 003200-0010 « En tant que visiteur, je veux pouvoir supprimer une base de données locale »
US 003300-0010 « En tant que visiteur je veux pouvoir consulter une base de données locale »
US 003300-0020 « En tant que visiteur, je veux pouvoir consulter une base de données locale par champ site»
US 003410-0010 « En tant que visiteur, je veux pouvoir ajouter un triplet »
US 003421-0010 « En tant que visiteur, je veux pouvoir modifier un site »
US 003422-0010 « En tant que visiteur, je veux pouvoir modifier un identifiant »
US 003423-0010 « En tant que visiteur, je veux pouvoir modifier un mot de passe »
US 003430-0010 « En tant que visiteur, je veux pouvoir supprimer un triplet »
US 004100-0010 « En tant qu’authentifié, je veux pouvoir créer une base de données distante »
US 004100-0020 « En tant qu’authentifié, je veux pouvoir exporter une base de données »
US 004200-0010 « En tant qu’authentifié, je veux pouvoir supprimer une base de données distante »
US 004300-0010 « En tant qu’authentifié, je veux pouvoir mettre à jour une base de données distante »
US 004400-0010 « En tant qu’authentifié, je veux pouvoir consulter les bases de données distantes »
US 005000-0010 « En tant qu’authentifié, je veux pouvoir importer une base de données »
US 006000-0020 « En tant qu’authentifié, je veux pouvoir modifier son mot de passe session »
US 007000-0010 « En tant qu’authentifié, je veux pouvoir modifier son mot de passe maître »
US 008000-0010 « En tant qu’authentifié, je veux pouvoir supprimer la session »

Technical stories :

Le serveur doit utiliser le système de fichier pour stocker.
L’utilisateur ne doit voir qu’une seule base de données locale à la fois.
L’utilisateur devra saisir son mot de passe maître et il sera stocké le temps de la durée de vie du client.
Un utilisateur peut se connecter via un token.
L’utilisateur peut avoir plusieurs bases de données distantes.
On utilise une base de données relationnelle côté serveur.
Les mots de passe et identifiants des triplets sont stockés en chiffrés dans le serveur.
Le client se chargera de chiffrer et déchiffrer les données.
Le client déchiffre les données à l’appel de l’utilisateur(accès au triplet).
Le serveur échange des données en JSON avec le client.
L’IHM du client doit correspondre à l’IHM de l’application mobile.
On ne fait pas de mise à jour en temps réel : On informe l’utilisateur qu’il y a des modifications sur la base de données locale qui n’ont pas été poussées sur le serveur.
L’utilisateur est informé si des changements n’ont pas été sauvegardé lors de la fermeture du client.
L’utilisateur peut entrer plusieurs sites identiques.
Un site d’une même base de données ne peut contenir plusieurs fois un même identifiant.
Un utilisateur ne peut lister que ses propres bases de données.

Development stories :

On utilise le serveur Wildfly dans lequel on code en Java 8 : Sur les conseils de notre professeur de D14.
On utilise Docker : Pour avoir une maîtrise de l’environnement.
On utilise GIT pour l’intégration continue.
On utilise Sqlite 3 car l’application mobile l’utilise.
On utilise REST : car le client n’a pas besoin de connaître l’architecture du serveur pour fonctionner.
On utilise MAVEN 3.5.4 : car on veut maîtriser notre environnement de compilation.
Client codé en JavaScript (Jquery 3) : .
On utilise HTML5 : Car il supporte le drag’n’drop et il permet d’utiliser un stockage local au navigateur.
On utilise CSS3 : car on utilise HTML5.
Navigateurs ciblés seront Firefox et Chrome et Safari par la suite : car ce sont les navigateurs les plus utilisés.
On utilise IndexedDB : IndexedDB est un moyen de stocker des données de manière persistante dans un navigateur, ce qui permet un mode hors ligne.
On utilise l’API WebCrypto : imposé par le client.
L’authentification doit intégrer des API externes : proposé par le client.