**Création d’une application permettant de gérer des Produits appartenant à des Catégories avec Spring Security, JWT, Angular 7.**

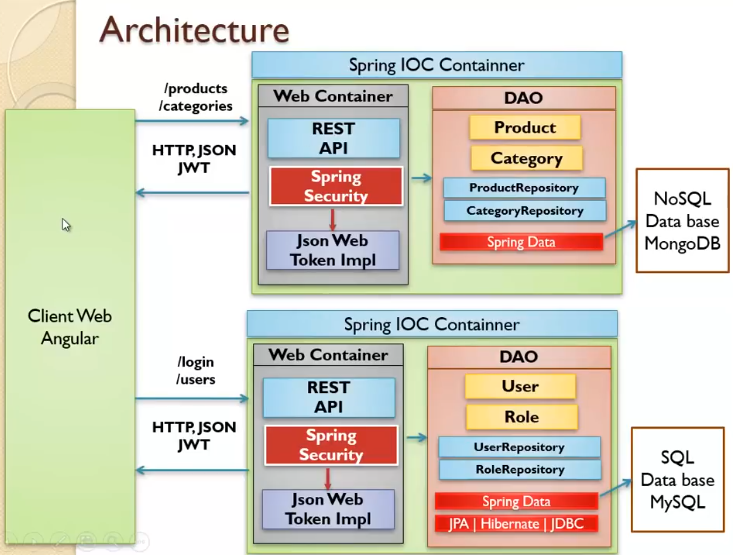
**Exigences fonctionnelles :**

* L’ application doit permettre de
* Saisir ou Ajouter un produit (ou catégorie), seulement si tu as le rôle ADMIN
* Chercher des produits (ou catégories), par son Id ou un mot clé
* Editer et modifier un produit (ou catégorie), seulement si tu as le rôle ADMIN
* Supprimer un produit (ou catégorie), seulement si tu as le rôle ADMIN
* Chaque utilisateur doit s’enregistrer puis s’authentifier pour avoir accès aux produits.

**Exigences Techniques :**

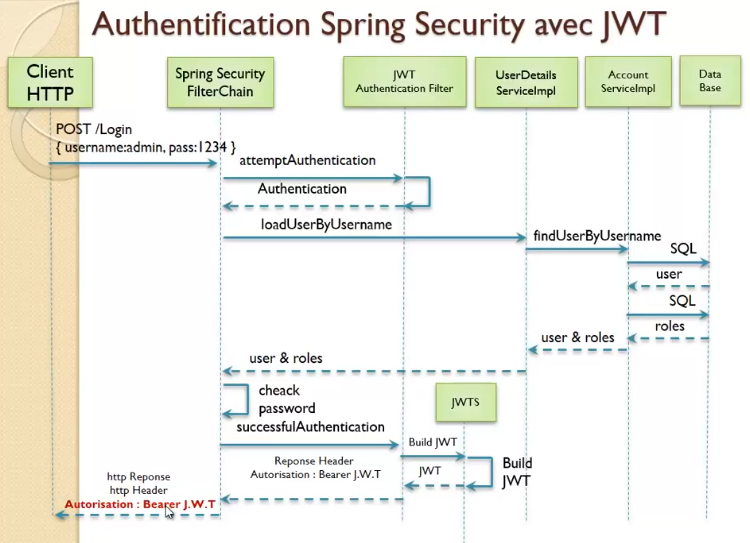
* On a deux micro services :
* Un micro service qui gère les produits et les catégories.
* Ils sont stockés dans une base de données NoSQL, en utilisant MongoDB.
* L’application se compose de plusieurs éléments :
* La couche DAO qui est basée sur Spring Data, MongoRepository,
* Pour L’API REST on utilise Spring Data Rest.
* Spring Security en utilisant l’API Json Web Token
* Pour la partie front end, on utilise Angular 7
* Un micro service qui gère les utilisateurs et les rôles, pour l’authentification, et l’accès aux ressources ou url.
  + Un utilisateur peut avoir plusieurs rôles, et un rôle concerne plusieurs utilisateurs.
  + Les rôles et les utilisateurs sont stockés dans une base de données MySQL.
  + L’application se compose de plusieurs éléments :
* La couche DAO qui est basée sur Spring Data, JPA,
* La couche Web qui est basée sur l’API REST, et utilisation de Spring Data Rest
* Spring Security en utilisant l’API Json Web Token
* Pour la partie front end, on utilise Angular 7

**Architecture Technique**



**Mise en place de deux Filtres**

* Un filtre nommé **Authentication Filter** pour l’authentification :



* Un filtre nommé Autorisation Filter qui intervient à chaque demande de ressource nécessitant l’authentification.
* Un filtre nommé **Autorisation Filter** qui intervient à chaque demande de ressource nécessitant l’authentification.

