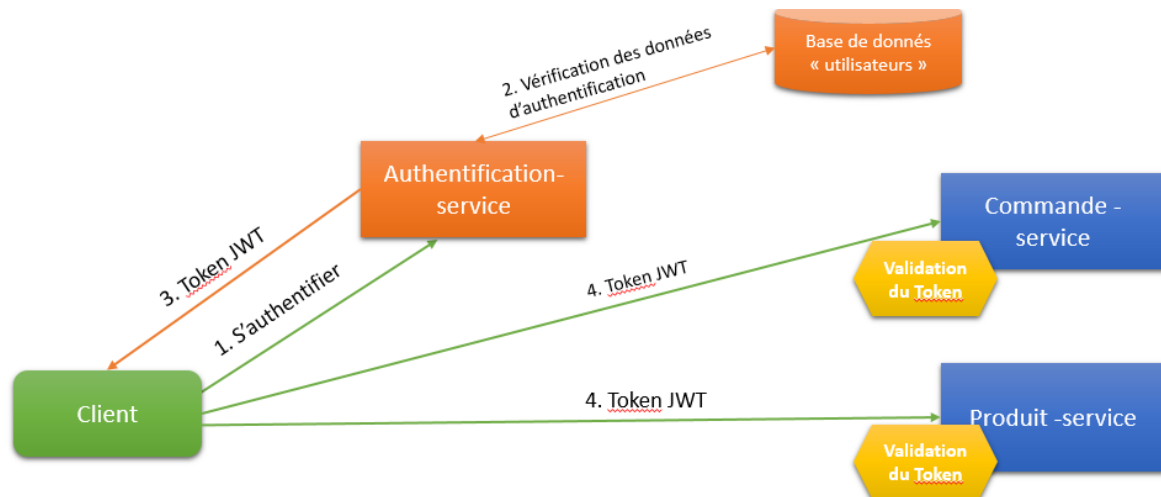


Authentication-service :

Avant de pouvoir exploiter les différentes méthodes des API créées, un utilisateur doit être connecté. Ainsi, on doit ajouter une étape intermédiaire d'authentification avant de consulter n'importe quel service ;

Le schéma suivant représente le modèle à suivre :



Ajoutons un nouveau service appelé « **auth-service** »:

| main ▾ | exp-microservice / auth-service |
|-------------------|---------------------------------|
| Name | Last con |
| .. | |
| node_modules | Ajout du |
| Utilisateur.js | Ajout du |
| index.js | Ajout du |
| package-lock.json | Ajout du |
| package.json | Ajout du |

8. Sous le dossier « exMicroservice », créer un sous dossier « auth-service »;
9. Initialiser npm et exécuter sous le terminal : **npm install express nodemon mongoose jsonwebtoken bcryptjs dotenv**
10. Modifier le script « test » sous package.json par : **"start": "nodemon index.js"**
11. Ajouter le schéma **utilisateur.js** pour stocker le nom, email, mot_passe et la date de création
12. Ajouter le fichier index.js pour définir deux routes suivantes :
 - /auth/register : permettra de créer et d'ajouter un nouvel utilisateur à la base de données
 - /auth/login : permettra de retourner un token après vérification de l'email et du mot de passe. La clé secrète du token va être stocké dans un fichier .env
13. Tester les deux routes sur Postman

Après avoir développé le service d'authentification et avoir généré le token suite à la connexion (email et mot de passe);

Les services Produit et commande recevront, désormais, des tokens lors des appels de leurs APIs exposées.

Ainsi, lesdits services doivent valider le token reçu avant de répondre à n'importe quelle requête.

14. Ajouter à chacun des services : Produit-service et Commande-service un fichier appelé : **isAuthenticated.js pour valider le token reçu qui va servir d'un middleware**
15. Modifier, par la suite, nos méthodes API, pour qu'elles prennent en considération l'intermédiaire (le middleware) **isAuthenticated**; Celui là vérifiera le token avant de répondre à la requête entrante.
Note : L'email de l'utilisateur sera récupéré par le biais du middleware **isAuthenticated** au lieu de le passer comme paramètre de la requête POST.
16. Pour tester, il faut, premièrement, démarrer les trois services, de s'authentifier et de récupérer le token et deuxièmement de d'appeler la méthode commander en lui passant ce token.