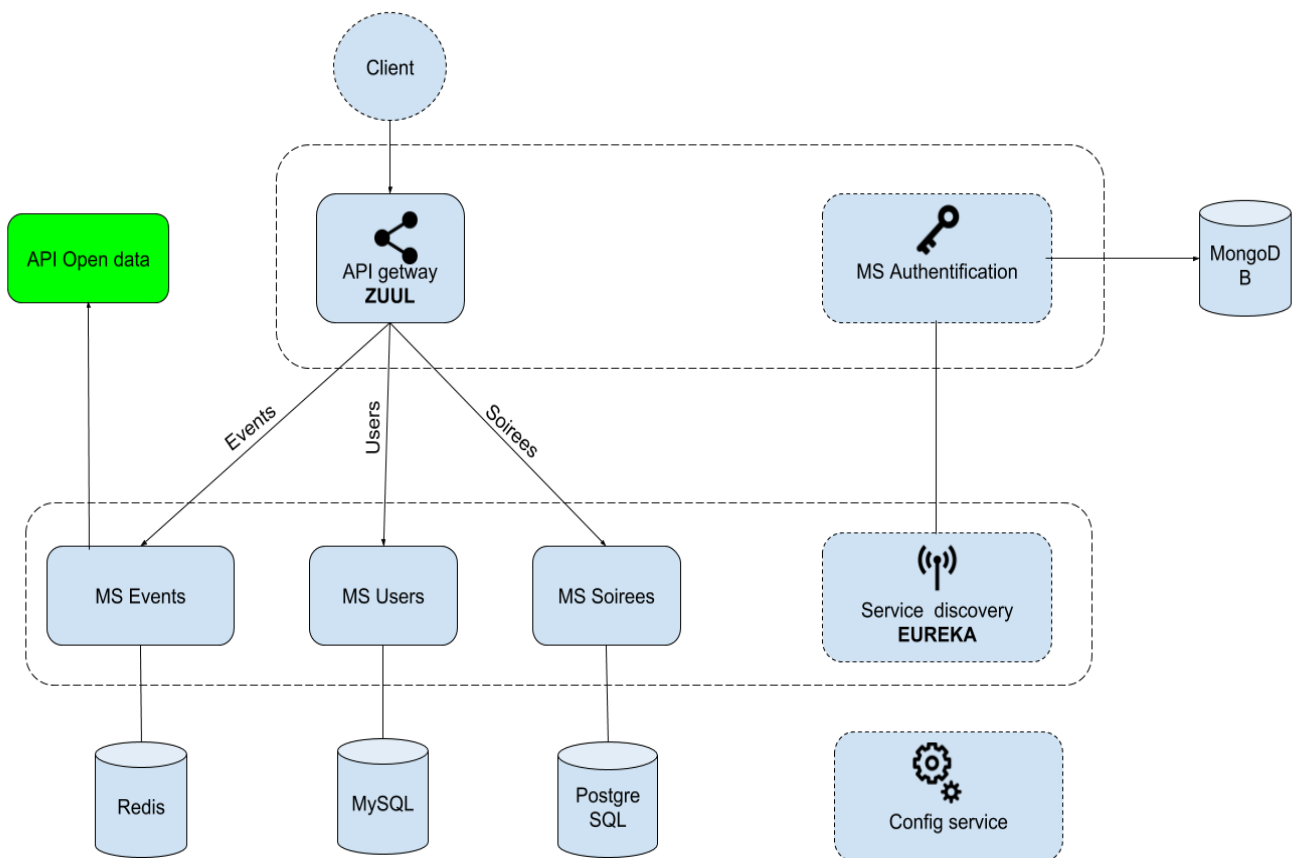


I. Architecture



A. Description de l'architecture & déploiement :

Chacun de nos micro-service est relié à une base de données. Ces services sont déployés indépendamment les uns des autres.

Le fonctionnement globale de l'application s'articule de la manière suivante :

- Premièrement il faut lancer la configuration du serveur dont laquelle on a centralisé les paramétrage de nos micro-services.
- Sachant que nous faisons une redirection dynamique avec Eureka, il faut lancer celui-ci pour que nos micro-services s'enregistrent afin de permettre une communication entre la Gateway (passerelle) et le micro-services.
- En suite, on peut lancer nos différents micro-services.

B. Technologie utilisées :

Pour le front , nous avons choisi d'utiliser :

- VueJS
- Ajax
- JQuery
- Thymleaf
- Bootstrap 4.0

Pour le bac, on est partis sur le classique des technologies à savoir :

- Spring Boot
- Spring MVC
- Spring Data
- Framework Jersey

II. APIs

URI \ VERB	GET	POST	PUT	DELETE
API « Users » : Gestion des utilisateurs et gestion des amis				
/users/signout		TODO		
/users/signin		TODO		
/users	In : none Out : JSON {liste de tous les utilisateurs} Success code : 200 Ok Error code : 404 not found	Inscription d'un utilisateur In : JSON {User Object Info} Out : URI location		
/users/{pseudo}	In : pseudo de l'utilisateur Out : JSON {User Object Info} Success code : 200 Ok Error code : 404 not found		Modification et mise à jour des infos d'un utilisateur. In : JSON {User Object Info}, pseudo de l'utilisateur Out : URI location Success code : 201 created Error code : 400 Bad request	
/users/{pseudo}/friends	In : pseudo de l'utilisateur Out : JSON {liste de tous les amis de l'utilisateur pseudo} Success code : 200 Ok Error code : 404 not found	Ajout d'un ami dans la liste d'amis de l'utilisateur pseudo. In : JSON {User Object Info} Out : URI location Success code : 201 created Error code : 400 Bad request		

/users/{pseudoU}/friends /{pseudoF}				<p>Suppression de pseudoF dans la liste d'amis de pseudoU.</p> <p>In : pseudo de l'utilisateur pseudoU pour lequel l'ami pseudoF va être supprimé.</p> <p>Out : _____</p> <p>Success code : 204 No content</p> <p>Error code : 400 bad request</p>
--	--	--	--	--

URI \ VERB	GET	POST	PUT	DELETE
API « Events » : Gestion des évènements				
/private-events	In : pseudo Out : JSON {liste de tous les evenements privees}	In : JSON {evenements} Out : URI location		
/private-events/{idEvent}	In : idEvent Out : JSON evenement		In : Event Out : 201 created	In : idEvent Out : 204 No content
/public-events	In : start, rows out : JSON {list des evenements publics}			
/public-events/{idEvent}	In : idEvent out : JSON {un evenement public}			

URI \ VERB	GET	POST	PUT	DELETE
API « Soirée » : Gestion des soirées				
/soirees	In : pseudo Out : JSON {list des soirees }	In : Soiree, pseudo Out : URI location		
/soirees/{idSoiree}	In : idSoiree out :JSON soiree			In : idSoiree out : 204 No content
/soirees/inv-soirees	In : pseudp Out : JSON {liste soirees auxquelles je suis invite} Success code : 200 Ok			
/soirees/inv-soirees/{idSoiree}			In : idSoiree, pseudo, etatSoiree Success code : 201 created	
/soirees/{idSoiree}/commentaires	In : idSoiree Out : JSON {liste de tous les commentaire d'une soiree} Success code : 200 Ok Error code : 404 not found	In : idSoiree, pseudo, commentaire (JSON) Out: URI location Success code : 201 created Error code : 400 Bad request		
/soirees/{idSoiree}/commentaires/{idCom}				In : idSoiree, idCom out : Success code : 204 No content Error code : 400 bad request
/soirees/{idSoiree}/evenement-prive/{idEvent}				in : idSoiree, idEvent Success code : 204 No content

				Error code : 400 bad request
/soirees/{idSoiree}/evenement-public/{idEvent}				in : idSoiree, idEvent Success code : 204 No content Error code : 400 bad request

URI \ VERB	GET	POST	PUT	DELETE
API « Authentification				
/uaa/users/current	in : out : current connected client			
/uaa/users/.	in : username, password out : JSON{User}			
/uaa/ouah/token	in : pseudo, motdepasse out: JSON User			

III. Base de données

MySQL: Microservice Users « ms_users »

MySQL est un SGBD de modèle relationnel

Utilisée pour la gestion des utilisateurs et leurs amis, elle a été choisie car nos entités présentent des relations entre elles mais pas que.

- Principaux intérêts :
 - o Le serveur MySQL est très rapide
 - o Beaucoup plus simple à utiliser
 - o Tourne sur divers systèmes tels qu'Unix, Windows ou Linux

PostgreSQL: Microservice Soirées « ms_soirees »

PostgreSQL est un SGBD relationnel et objet, robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données, mêmes dans des situations critiques.

Nous supposons que pour les soirées, nous aurons dans des données massives c'est en partie dans cette optique que nous avons choisi cette BD.

Redis: Microservice Evénements « ms_evenements »

L'usage de base de Redis, c'est de stocker des valeurs associées à des clés. Le serveur Redis est comme un gigantesque Hash Map, dictionnaire Python, object Javascript, Array associatif en PHP... En premier lieu, donc, on utilise Redis pour stocker des choses ainsi :

```
"cle1" => valeur1
```

```
"cle2" => valeur2
```

```
Etc.
```

Le module événement ne dispose que d'une table dans la BD sur laquelle nous effectuons des opérations du CRUD. C'est la cause radicale de son utilisation afin d'avoir pour chaque clé, un événement associé.

MongoDB : Microservice authentification « ms_authentification »

Dans MongoDB, les données sont modélisées sous forme de document sous un style JSON.

On ne parle plus de tables, ni d'enregistrements mais de collections et de documents. Ce système de gestion de données nous évite ainsi de faire des jointures de tables car toutes les informations propres à une certaine donnée sont stockées dans un même document.

Nous l'avons particulièrement appréciée pour :

- Assurer face à une montée en charge rapide.
- Pouvoir restructurer son schéma à la volée.
- Un besoin de redondance et de continuité de service.
- Une répartition géographique des données.
- Un besoin de système de recherche, de stockage de fichier ou d'un système information géographique.

