



UNIVERSITE D'ORLEANS

# Interopérabilité et innovation dans les SI

Année universitaire 2018-2019

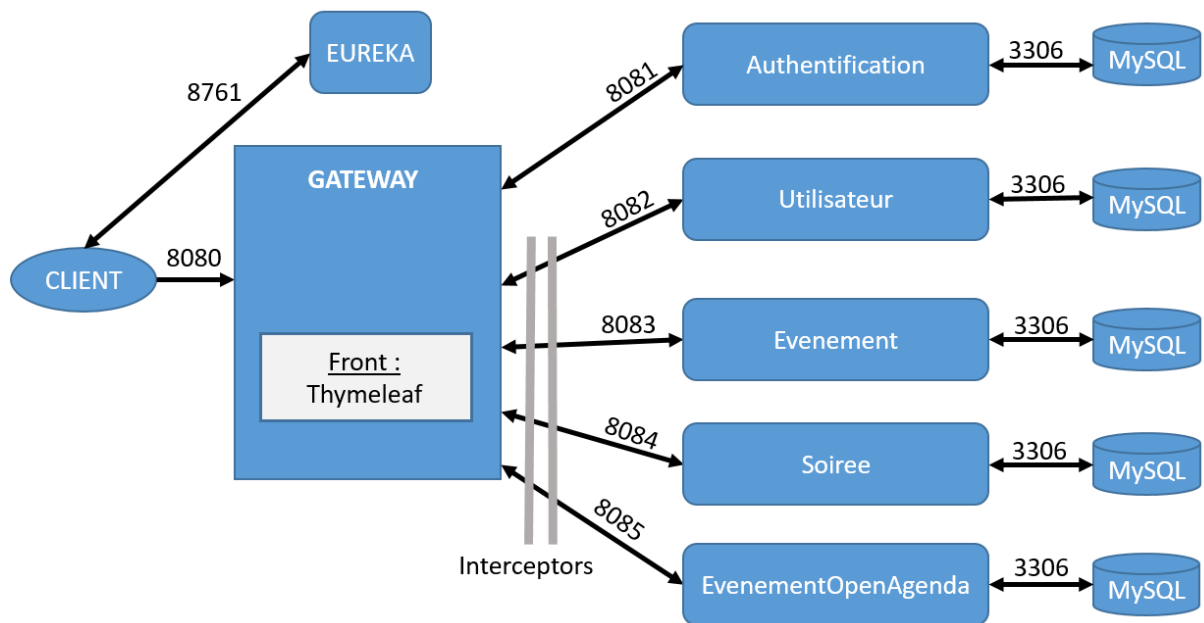
GROUPE C  
BAYRAKLI Levent-Gabriel  
DESSUS Arthur  
LABATUT Maxime  
TROUILLET Quentin

## Table des matières

<b>I. Architecture technique .....</b>	<b>1</b>
<b>1. Choix d'implémentations.....</b>	<b>1</b>
<b>2. Découpage et interactions entre les services .....</b>	<b>4</b>
a. Authentification .....	4
b. Utilisateur.....	4
c. Événement .....	4
d. EvenementOpenAgenda .....	4
e. Soirée .....	4
<b>II. Déploiement .....</b>	<b>5</b>
<b>1. Build .....</b>	<b>5</b>
<b>2. Lancer les conteneurs Docker .....</b>	<b>5</b>

## I. Architecture technique

### 1. Choix d'implémentations

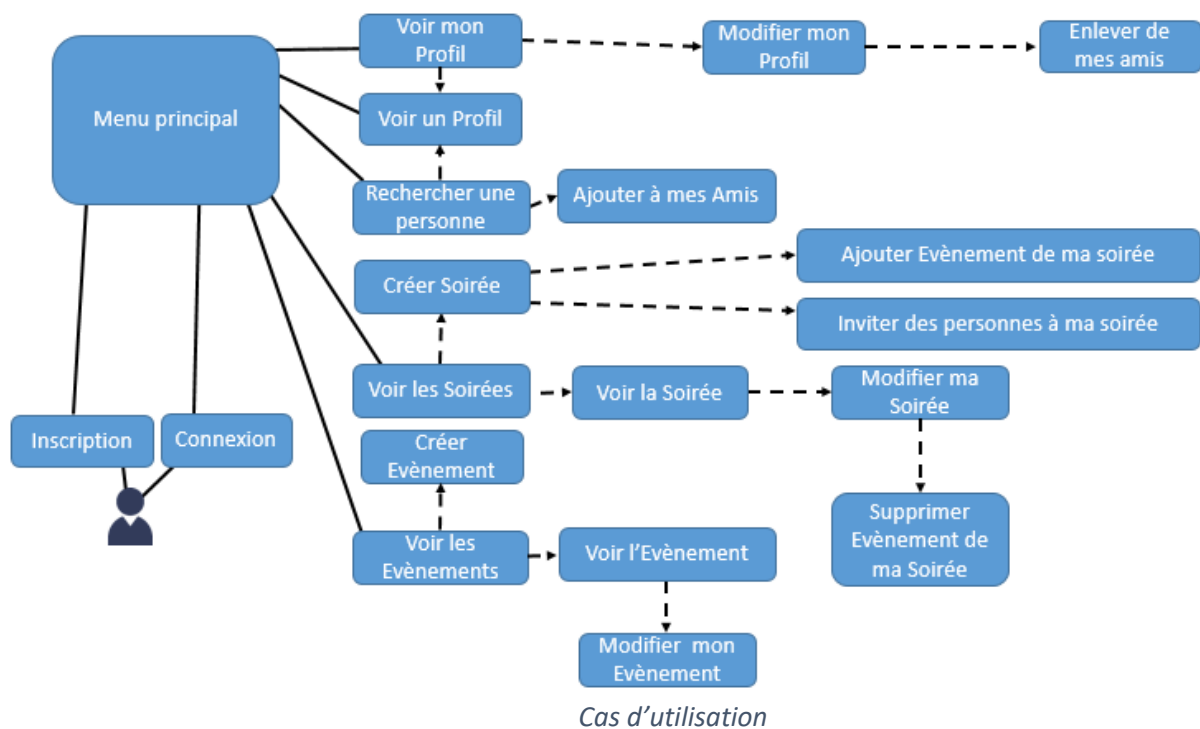


*Schéma de l'Architecture technique*

Notre Implémentation se découpe en 7 micros services et une Gateway :

- **Authentification** : Micro-service s'occupant exclusivement de l'authentification soit : la connexion, inscription, déconnexion...
- **Utilisateur** : Micro-service s'occupant de la gestion des utilisateurs mais aussi, on utilise ce micro-service pour gérer les invitations que peuvent se faire les utilisateurs pour ajouter/supprimer un ami et accepter/refuser une invitation à un évènement...
- **Évènement** : Micro-service s'occupant exclusivement de la gestion des évènements soit la création, suppression, modification...
- **Soirée** : Il correspond à la gestion des Soirées.
- **EvenementOpenAgenda** : Ce micro-service correspond à la gestion des événements de l'API Orléans Data métropole que l'on récupère depuis le site : <https://data.orleans-metropole.fr/pages/home/>
- **Eureka** : micro-service reposant sur la dynamisation des ports. Chaque micro-service s'enregistre auprès de celui-ci et ainsi.

L'interface utilisateur se trouve en dans la gateway.



Lors du lancement de l'application nous avons le choix entre nous connecter et nous inscrire. Si nous choisissons de nous inscrire, on sera basculé directement sur le menu principal sans passer par le menu connexion.

#### **Sur le menu principal, nous avons plusieurs informations différentes :**

Un panneau de navigation propre à la page

Le résumé des notifications d'invitations aux soirées et de demandes d'amis reçues où on peut accepter ou refuser ces demandes.

Un module informant les soirées à venir pour les soirées auxquelles la personne connectée serait inscrite.

#### Dans un premier menu :

Sur la page de consultation du profil de la personne qui est connectée, on peut voir le « fil d'Ariane » qui nous indique les vues par lesquelles nous sommes passé. Nous pouvons voir notre liste d'amis avec un lien pour consulter leur profil.

Nous avons le détail de notre profil avec un bouton « modifier » qui permet de changer les caractéristiques du profil.

Dans la modification de profil, on peut changer les infos sur l'e-mail, le nom, le prénom, l'adresse la ville, le code postal et la date de naissance. Nous avons toujours notre liste d'amis avec un bouton pour les supprimer.

#### Dans un deuxième menu :

On peut consulter le profil des autres personnes et amis. On peut voir la liste d'amis que la personne possède ainsi que les détails de son profil sans pouvoir modifier ses informations.

Dans un troisième menu :

On peut accéder aussi au menu de recherche de personne qui liste toutes les personnes de connectées. Pour chaque personne, si la personne connectée n'est pas ami avec certaines personnes alors on lui propose à côté du nom le fait d'envoyer une invitation d'ami.

On peut bien évidemment aussi cliquer sur le nom de la personne pour consulter son profil.

Dans un quatrième menu :

Nous pouvons consulter les soirées ayant été créées mais aussi créer une nouvelle soirée.

Lors de la création d'une nouvelle soirée, nous devons entrer les caractéristiques principales (voir paragraphes en dessous). De plus, nous pouvons inviter nos certains de/tous nos amis à la soirée.

On peut cliquer sur leur nom pour accéder au détail de la soirée. Dans le détail de la soirée, on peut voir son nom, le nombre de places disponible, la date de la soirée, l'heure de début de la soirée.

Pour une soirée, on peut voir aussi les événements privés et les événements publics rattachés à la soirée ainsi que les participants à la soirée.

Si nous sommes le propriétaire de la soirée, nous pouvons la modifier. Durant la modification, on peut bien évidemment retoucher toutes les caractéristiques principales de la soirée, mais aussi enlever un événement rattaché à la soirée.

Dans un cinquième menu :

Nous pouvons consulter les événements ayant été créés mais aussi créer un nouvel événement.

Il y a deux types d'événements, les événements privés créés par les utilisateurs pour inviter entre amis et les événements publics récupérés sur l'API Orléans Data métropole. Les événements publics sont visualisés en détail dans le menu où on affiche tous les événements. Pour tout événement public, on peut voir plus de détail et choisir de créer une soirée affiliée à celui-ci.

Lors de la création d'un nouvel événement, nous devons entrer les caractéristiques principales présentées ci-dessous.

On peut cliquer sur leur nom pour accéder aux détails de l'événement. Dans le détail de l'événement, on peut voir son nom, le lieu et la date de l'événement.

Si nous sommes le propriétaire de l'événement, nous pouvons le modifier. Durant la modification, on peut retoucher toutes les caractéristiques de l'événement, mais aussi enlever un événement rattaché à la soirée.

## 2. Découpage et interactions entre les services

### a. Authentification

URL					Méthode											
Protocole	Nom du serveur	Port	Chemin de la ressource		GET			POST			PUT			DELETE		
			Service	Ressource	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles
http://	localhost	8080	/authentication	/connexion		-		Connexion d'un utilisateur		200 409 404	-		Déconnexion d'un utilisateur	-	200 401 404	
http://	localhost	8080	/authentication	/inscription		-		Inscription d'un utilisateur		201 401 409	-		-	-		

### b. Utilisateur

URL			Méthode														
Protocole	Nom du serveur	Port	Chemin de la ressource		GET			POST			PUT			DELETE			
			Service	Ressource	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	
http://	localhost	8080	/utilisateur	/	Recherche tous les utilisateurs	Collection d'Utilisateurs	200 401	Ajout d'un utilisateur	Utilisateur	201 401 404	-	-	-	-	-	-	
http://	localhost	8080	/utilisateur	/idUtilisateur	Recherche d'un utilisateur	Utilisateur	200 401 404	-	-	-	Modification d'un utilisateur	200 401 404	-	-	-	-	
http://	localhost	8080	/utilisateur	/idUtilisateur/amis	Recupère les amis	Collection d'Utilisateurs	200 401 404	Ajoute d'un ami	Invitation d'ami	201 401	-	-	-	Supprime un ami	-	200 401	
http://	localhost	8080	/utilisateur	/idUtilisateur/invitationsAmis	Recherche tous les invitations d'amis d'un utilisateur	Collection d'invitations d'amis	200 401 404	Ajout d'une invitation d'ami	Utilisateur	201 401	-	-	-	-	-	-	
http://	localhost	8080	/utilisateur	/idUtilisateur/invitationsAmis/idinvitation	-	-	-	-	-	-	Accepter invitation d'ami	200 401	Refuser invitation d'ami	-	-	200 401	
http://	localhost	8080	/utilisateur	/idUtilisateur/invitationsSoirees	Recherche tous les invitations de Soirées d'un utilisateur	Collection d'invitations de soirées	200 401 404	Ajout d'une invitation de Soirée	Invitation de soiree	201 401	-	-	-	-	-	-	
http://	localhost	8080	/utilisateur	/idUtilisateur/invitationsSoirees/idinvitation	-	-	-	-	-	-	Accepter invitation de soirée	200 401	Refuser invitation d'une soirée	-	-	200 401	

### c. Événement

URL				Méthode												
Protocole	Nom du serveur	Port	Chemin de la ressource		GET			POST			PUT			DELETE		
			Service	Ressource	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles
http://	localhost	8080	/evenement	/	Recherche tous les événements	Collection d'Event	200 401	Ajout d'un événement	Evenement	201 401		-			-	
http://	localhost	8080	/evenement	/idEvenement	Recherche d'un événement	Evenement	200 401 404		-		Modification d'un événement	Evenement 200 401		Suppression d'un événement	-	200 401
http://	localhost	8080	/evenement	/idEvenement)/utilisateur	Rechercher tous les Utilisateurs dans l'évènement	Collection Utilisateur	200 401 404	Ajout d'un utilisateur à un événement	Evenement	200 401 404		-			-	

### d. EvenementOpenAgenda

URL					Méthode											
Protocole	Nom du serveur	Port	Chemin de la ressource		GET			POST			PUT			DELETE		
			Service	Ressource	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles
http://	localhost	8080	/evenementOpenAgenda	/	Rechercher tous les événements open agenda	Collection d'Event	200 401	-	-	-	-	-	-	-	-	-
http://	localhost	8080	/evenementOpenAgenda	/idEvenement	Recherche d'un Event	Evenement	200 401 404	-	-	-	-	-	-	-	-	-

### e. Soirée

URL				Méthode												
Protocole	Nom du serveur	Port	Chemin de la ressource		GET			POST			PUT			DELETE		
			Service	Ressource	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles	Objectif	Données renvoyées	Codes HTTP possibles
http://	localhost	8080	/soirée	/	Rechercher toutes les soirées	Collection de soirée	200 401	Ajout d'une soirée	Soirée	201 401	-	-	-	-	-	-
http://	localhost	8080	/soirée	/idSoirée	Rechercher une soirée	Soirée	200 401 404	-	-	-						
http://	localhost	8080	/soirée	/idSoirée/evenements	Rechercher tous les événements d'une soirée	Collection d'événement	200 401 404	Ajout d'un événement à une soirée	-	200 401	-	-	-	-	-	-
http://	localhost	8080	/soirée	/idSoirée/evenementsOpenAgenda	Rechercher tous les événements open agenda de la soirée	Collection d'événement	200 401 404	Ajout d'un événement open agenda à une soirée	-	200 401	-	-	-	-	-	-

## II. Déploiement

Le processus de déploiement est expliqué dans le README.md du projet disponible à l'adresse : <https://pdicost.univ-orleans.fr/git/projects/INTEROP/repos/groupepec/browse> cependant en voici le contenu.

### 1. Build

Rendez-vous dans le répertoire projet-microservices-docker et exécutez la commande `mvn clean package -Dmaven.test.skip=true` afin de générer les fichiers jar des différents micro-services.

```
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] projetMicroServices 1.0-SNAPSHOT ..... SUCCESS [ 0.378 s]
[INFO] gateway 1.0-SNAPSHOT ..... SUCCESS [ 5.625 s]
[INFO] soiree 1.0-SNAPSHOT ..... SUCCESS [ 1.974 s]
[INFO] utilisateur 1.0-SNAPSHOT ..... SUCCESS [ 1.848 s]
[INFO] evenement 1.0-SNAPSHOT ..... SUCCESS [ 1.012 s]
[INFO] authentication 1.0-SNAPSHOT ..... SUCCESS [ 1.028 s]
[INFO] evenementOpenAgenda 1.0-SNAPSHOT ..... SUCCESS [ 0.981 s]
[INFO] eureka-server 1.0.0-SNAPSHOT ..... SUCCESS [ 0.508 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 14.180 s
[INFO] Finished at: 2019-03-30T18:10:56+01:00
[INFO] -----
```

Résultat de la commande "mvn clean"

### 2. Lancer les conteneurs Docker

Tout d'abord vous devez build les images. Pour cela, exécutez la commande `docker-compose build`. Ainsi les images seront et installées.

Un exemple d'image correctement build :

```
Building eurekaServer
Step 1/5 : FROM java:8-jre
----> e44d62cf8862
Step 2/5 : ENV PORT=8761
----> Using cache
----> 928840405912
Step 3/5 : ADD target/eurekaServer.jar eurekaServer.jar
----> 17857f1aa86b
Step 4/5 : ENTRYPOINT ["java", "-jar", "/eurekaServer.jar"]
----> Running in a48723a15baf
Removing intermediate container a48723a15baf
----> ae94258a0d3a
Step 5/5 : EXPOSE 8761
----> Running in 1c4559de40be
Removing intermediate container 1c4559de40be
----> 967a7c130f61
Successfully built 967a7c130f61
Successfully tagged projet-microservices eurekaServer:latest
```

Résultat de la commande "docker-compose build"

Une fois l'installation terminée, vous pouvez vérifier que les images sont correctement installées avec la commande `docker images`.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
projet-microservices_evenement	latest	53971128f6fe	4 seconds ago	375MB
projet-microservices_evenement_openagenda	latest	53971128f6fe	4 seconds ago	375MB
projet-microservices_authentification	latest	4c254734a7b6	7 seconds ago	364MB
projet-microservices_utilisateur	latest	1d8cd1062421	10 seconds ago	375MB
projet-microservices_gateway	latest	78232ca1737a	14 seconds ago	357MB
projet-microservices_eurekaserver	latest	3af78c940b1d	17 seconds ago	356MB
mysql	5.7	98455b9624a9	3 days ago	372MB
java	8-jre	e44d62cf8862	2 years ago	311MB

Résultat de la commande "docker images"

Vous pouvez maintenant démarrer les conteneurs avec la commande `docker-compose up -d`. Le paramètre `-d` signifie que ceux-ci s'exécutent en tâche de fond.

```
Creating network "projet-microservices_default" with the default driver
Creating utilisateur_db      ... done
Creating evenement_db       ... done
Creating soiree_db          ... done
Creating evenement_openagenda_db ... done
Creating authentication_db   ... done
Creating eurekaserver       ... done
Creating gateway            ... done
Creating utilisateur        ... done
Creating evenement          ... done
Creating soiree             ... done
Creating evenement_openagenda ... done
Creating authentication      ... done
```

Résultat de la commande "docker-compose up -d"

La commande `docker ps -a` permet d'afficher tous les conteneurs (en cours ou terminés) ce qui peut être intéressant.

Si vous souhaitez accéder aux logs d'un conteneur en particulier, vous pouvez utiliser la commande `docker logs [nom-du-conteneur]`. Les noms des conteneurs peuvent être retrouvés grâce à la commandé précédemment citée : `docker ps -a`

Vous pouvez accéder à :

- L'application : <http://localhost:8080/ui/>
- Le panneau de contrôle Eureka : <http://localhost:8761/>



Vous pouvez arrêter tous les conteneurs avec la commande `docker-compose down` :

```
Stopping authentication    ... done
Stopping evenement        ... done
Stopping utilisateur      ... done
Stopping soiree           ... done
Stopping gateway          ... done
Stopping evenement_db     ... done
Stopping eurekaserver     ... done
Stopping soiree_db        ... done
Stopping authentication_db ... done
Stopping utilisateur_db   ... done
Stopping evenement_openagenda_db ... done
Removing evenement_openagenda ... done
Removing authentication    ... done
Removing evenement        ... done
Removing utilisateur      ... done
Removing soiree           ... done
Removing gateway          ... done
Removing evenement_db     ... done
Removing eurekaserver     ... done
Removing soiree_db        ... done
Removing authentication_db ... done
Removing utilisateur_db   ... done
Removing evenement_openagenda_db ... done
Removing network projet-microservices default
```

Résultat de la commande “`docker-compose down`”