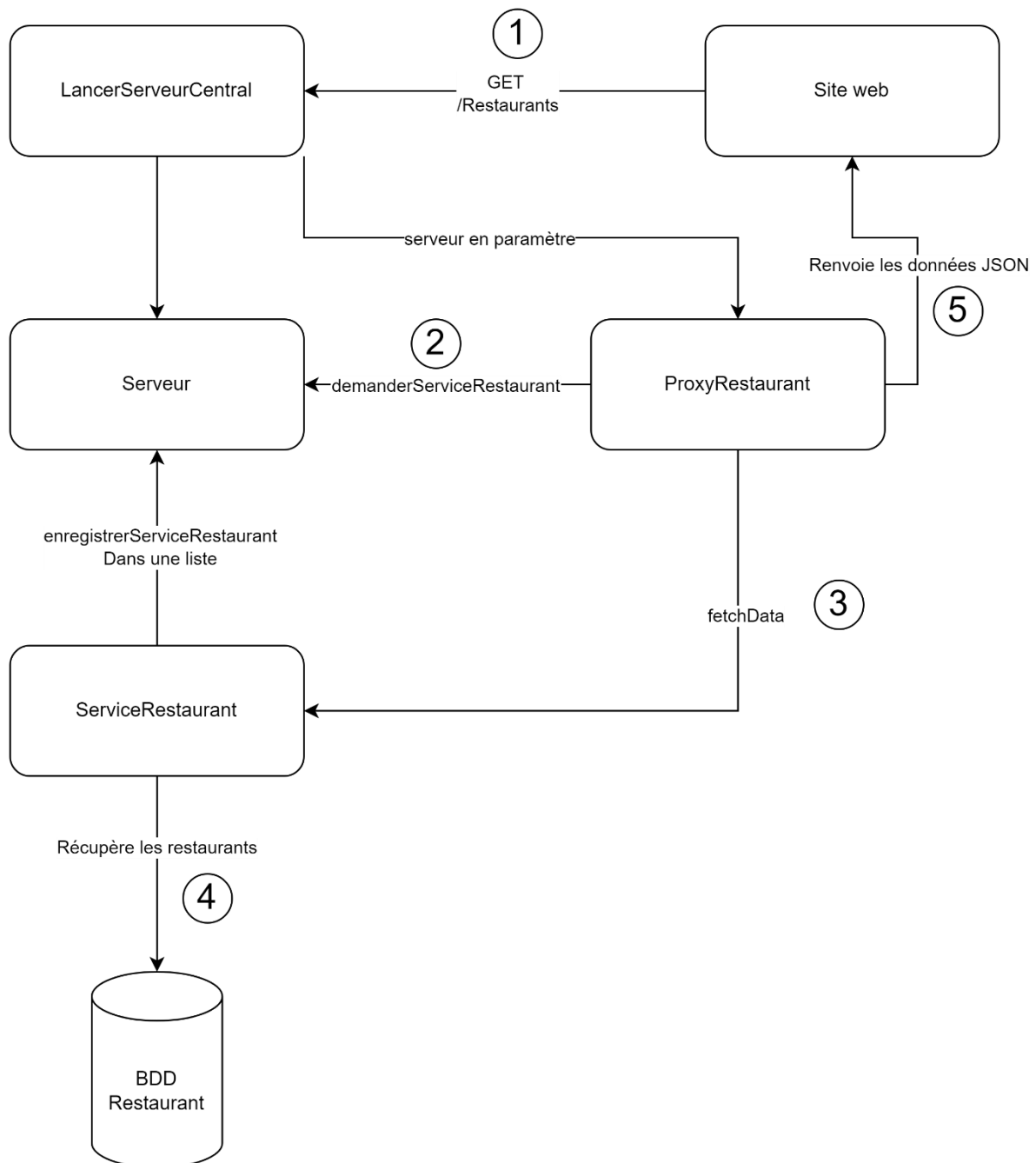


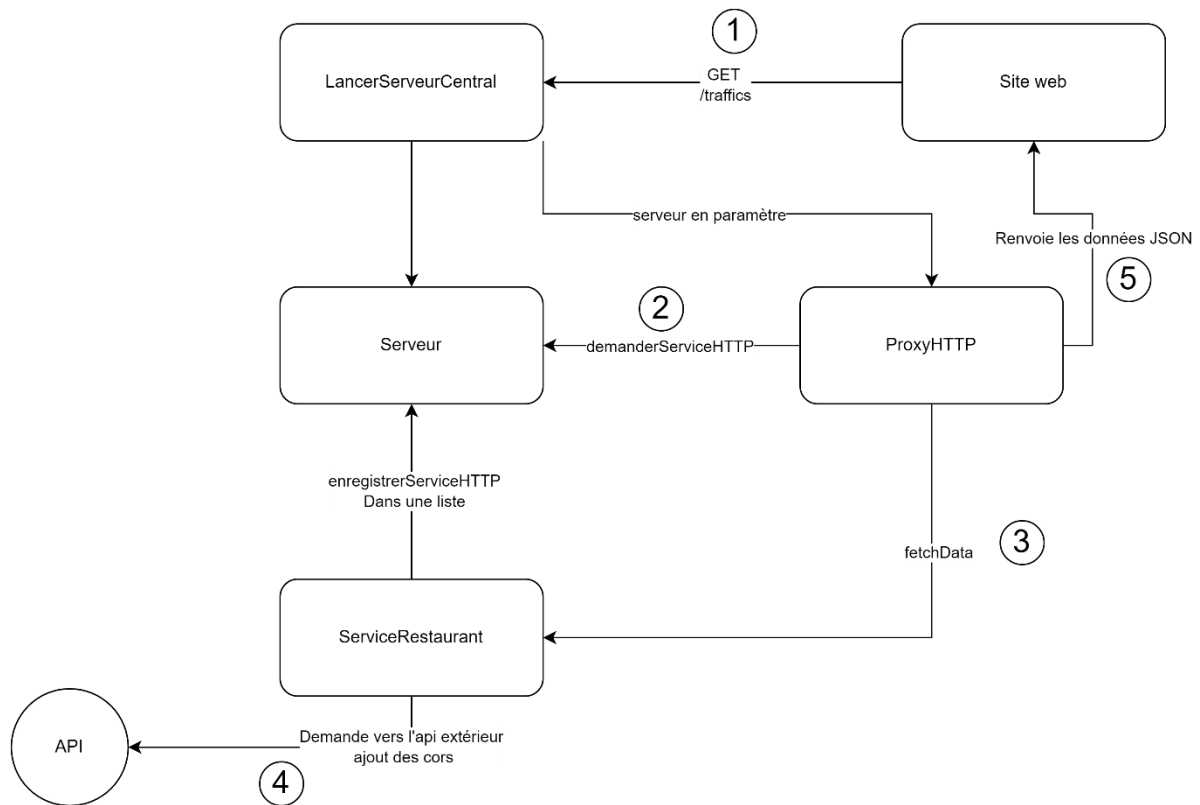
SAE : PROJET REPARTIE

HAROUNA Laetitia – RAGOT Yogan

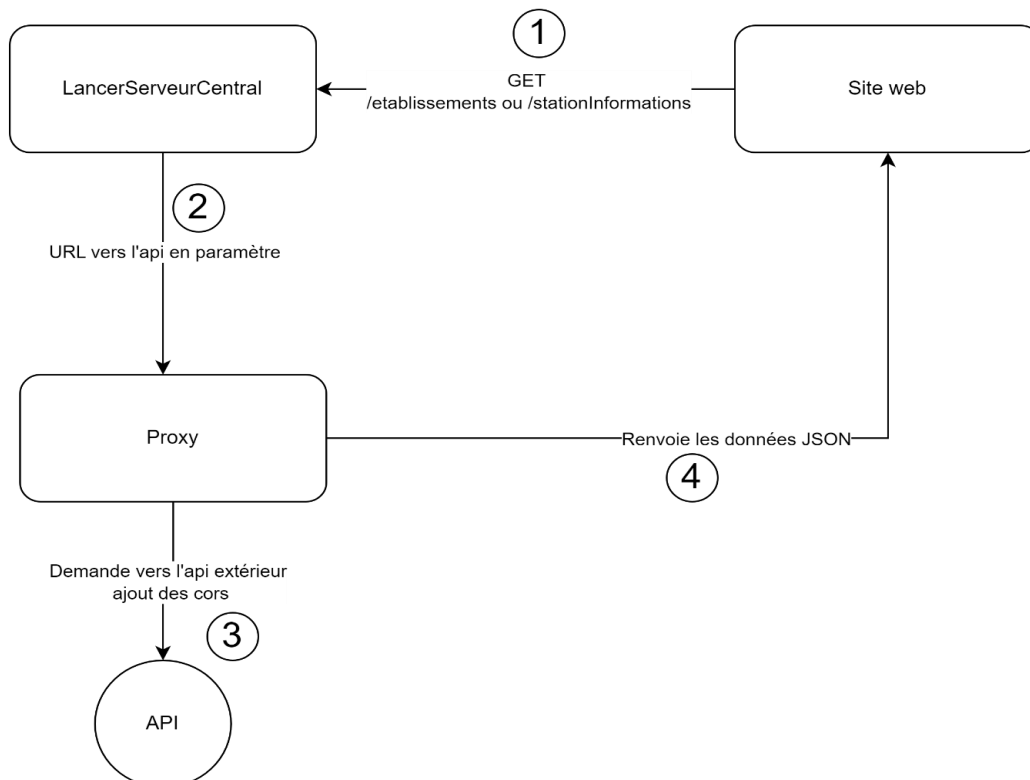
ARCHITECTURE (Centré sur les restaurants avec un scénario)



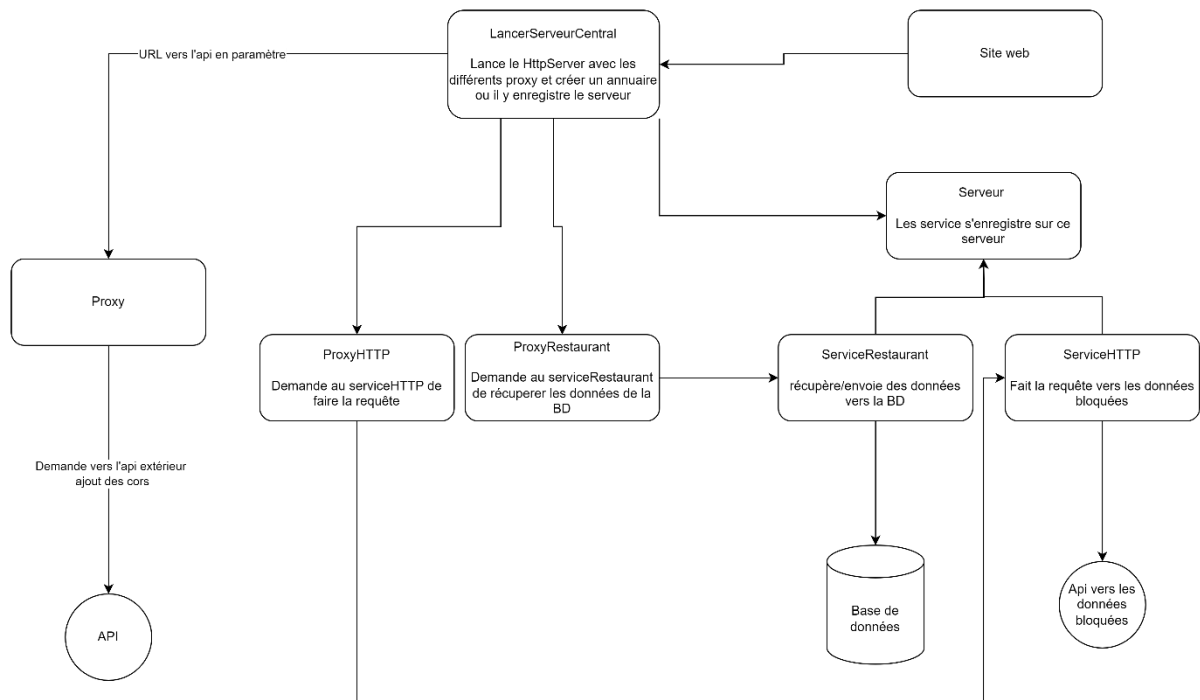
ARCHITECTURE (Centré sur les incidents avec un scénario)



ARCHITECTURE (Centré sur les autres requêtes API (Vélo, établissements))



ARCHITECTURE Générale



TECHNOLOGIES

- **Frontend:** HTML, CSS, JavaScript, Leaflet.js
- **Backend:** Java, RMI
- **Base de données:** ORACLE JDBC
- **Déploiement:** Webetu

REALISATION

- **Création de la base de données des restaurants:**
 - Création d'une base de données avec les tables réservations, tabl et restaurants
 - Il y a un script dans le répertoire bd pour que vous prenez connaissance de ce que contient chaque table.
- **Développement des services RMI:**
 - Un service permettant de récupérer toutes les coordonnées des restaurants et permettant de réserver une table dans un restaurant en renseignant nom, prénom, nombre de convives, et coordonnées téléphoniques.
 - Un service permettant d'appeler les données fermées sur les incidents à Nancy
- **Visualisation des données sur une carte Leaflet:**

- Affichage des stations Vélib avec l'adresse de la station, le nombre de vélos disponibles et le nombre de places de parking libres.
- Ajout des établissements d'enseignement supérieur.
- Ajout des incidents de circulation liés aux travaux dans la Métropole du Grand Nancy avec leur adresse exacte, leur cause et les dates d'incidence.
- Ajout des restaurants
- **Création d'un Proxy:**
 - Un proxy pour interroger des données ouvertes comme Les stations ou les établissements.
- **Réservation de table directement depuis la carte:**
 - Intégration d'une fonctionnalité permettant de réserver une table en cliquant sur la marque d'un restaurant sur la carte.

Informations Nécessaires à l'Utilisation du Projet

1. Accès au site:

- Hébergé sur webetu, accessible via L'URL (cf Git Repository et Webetu)

2. Lancer les différents programmes:

Un script bash se trouve dans le dépôt github mais nous n'avons pas pu le tester voici les 3 commandes à lancer dans des terminaux différents si la version de java compiler est la bonne :

- Les commandes sont à réaliser depuis /src/

```
java serveur/LancerServeurCentral.java
serviceRestaurant/LancerServiceRestaurant.java localhost
java serviceHTTP/LancerServiceHTTP.java localhost
```

Erreurs :

- Si vous avez une erreur de version vous trouverez à la fin de ce pdf toutes les commandes pour javac tous les fichiers et déplacer les fichiers class aux bons endroits
- Si vous avez une erreur JDBC voici la commande pour réaliser la compilation du serviceRestaurant :

```
java -cp ".;ojdbc11.jar" serviceRestaurant/LancerServiceRestaurant localhost
```

3. Utilisation de la carte:

- La carte Leaflet est centrée sur Nancy et permet de visualiser les stations Vélib,
- Si vous cliquez sur le bouton Etablissement les établissements d'enseignement supérieur s'affichent
- Sur le bouton Trafic on aura les incidents de circulation.
- En cliquant sur Restaurant on a les marques des restaurants, et en cliquant sur l'une d'elles un bouton réserver est contenu dedans si vous cliquez une fenêtre de réservation s'ouvre en bas de page.

4. **Service RMI Restaurant:**

- Inscription du service sur le serveur.
- Utilisation des services pour récupérer les coordonnées des restaurants et effectuer des réservations.

5. **Service RMI HTTP:**

- Inscription du service sur le serveur.
- Interroge des données fermées.

INSTALLATION ET EXECUTION

Pour exécuter le projet, il faut exécuter les commandes suivantes si problème compilation :

Compilation des interfaces Java

```
javac serveur/ServeurInterface.java
```

```
javac serviceHTTP/ServiceHTTPInterface.java
```

```
javac serviceRestaurant/ServiceRestaurantInterface.java
```

```
javac serviceRestaurant/Restaurant.java
```

```
javac serviceRestaurant/Reservation.java
```

Copie des fichiers .class

```
cp serviceRestaurant/ServiceRestaurantInterface.class serviceHTTP/
```

```
cp serviceRestaurant/ServiceRestaurantInterface.class serveur/
```

```
cp serviceRestaurant/Restaurant.class serveur/
```

```
cp serviceRestaurant/Reservation.class serveur/
```

```
cp serviceHTTP/ServiceHTTPInterface.class serviceRestaurant/
```

```
cp serviceHTTP/ServiceHTTPInterface.class serveur/
```

```
cp serveur/ServeurInterface.class serviceRestaurant/
```

```
cp serveur/ServeurInterface.class serviceHTTP/
```

Lancer le serveur central

- Allez dans le répertoire suivant depuis votre terminal

src/serveur/

- Faites la commande suivante :

javac LancerServeurCentral.java

java LancerServeurCentral

Le serveur est maintenant lancé.

Il va lancer 3 Proxys : Un proxy pour les stations de vélo, établissement (Proxy : il va recevoir une URL et il fait la demande lui-même), un proxy pour les trafic (ProxyHTTP) et un proxy pour Restaurant (ProxyRestaurant).

Lancer les services

- Allez dans le répertoire :

src/serviceRestaurant

- Faites la commande suivante :

javac LancerServiceRestaurant.java

java -cp ".;ojdbc11.jar" LancerServiceRestaurant localhost (ceci est la commande en Windows, nous ne savons pas si elle marche sous Linux)

Vous pouvez maintenant accéder aux restaurants enregistrés dans notre base de données si vous cliquez sur le bouton Restaurants.

Vous avez également la possibilité de réserver si vous cliquez sur le bouton réserver, un formulaire s'affichera en bas.

- Allez dans le répertoire :

src/serviceHTTP

Faites la commande suivante :

javac LancerServiceHTTP.java

javaLancerServiceHTTP localhost

Vous pouvez maintenant voir afficher les incidents en appuyant sur le bouton Trafic.

Git Repository et WebEtu

- Git : https://github.com/ItsNezio/Projet_Repartie_HAROUNA_RAGOT
- Webetu : <https://webetu.iutnc.univ-lorraine.fr/~harouna4u/RMI/src/js/>

Explication Technique

Service RMI : ServiceRestaurant

Le service RMI Restaurant permet d'accéder à la base de données des restaurants et offre les fonctionnalités suivantes :

- Récupération de toutes les coordonnées des restaurants.
- Réservation d'une table avec les informations suivantes : nom, prénom, nombre de convives et coordonnées téléphoniques.

Service RMI : ServiceHTTP

Le service RMI HTTP est utilisé pour interroger des données fermées et contourner les problèmes de blocage de données sur le client

Proxy HTTP

Le proxy HTTP est utilisé pour le service RMI.

Proxy Restaurant

Le proxy Restaurant est utilisé pour utiliser le service Restaurant et envoyé les données vers la Base de données et les récupérés.

Proxy

Proxy permet de faire toutes les requêtes depuis HttpServer au lieu de faire avec le JS cela permet de séparer le front-end et le back-end.

Visualisation des Données sur Leaflet

La carte Leaflet affiche les informations suivantes :

- Stations Vélib : adresse, nombre de vélos disponibles, nombre de places de parking libres.
- Établissements d'enseignement supérieur.
- Incidents de circulation : adresse, cause et dates d'incidence.

Conclusion

Ce projet combine plusieurs technologies pour offrir une application répartie affichant des informations sur Nancy. Il permet de visualiser des données en temps réel et d'interagir avec elles via un navigateur web.