



DESCRIPTION DES ELEMENTS DE BASE REUTILISABLE

Imamou Chaima
Architecte logiciel MedHead+

Table des matières

I.	Introduction.....	2
II.	Architecture et communications des micro-services.....	2
III.	Les outils et dépendances utilisés.....	3
IV.	Description des micro-service	4
A.	Micro-service gestion des hôpitaux	4
B.	Micro-service réservation de lit	5
C.	Service configuration.....	7
D.	Service d'enregistrement	7
E.	Service API Gateway (proxy)	8
V.	Quelques exemples de différents d'Endpoint des services.....	9
VI.	Axes d'améliorations	10

I. Introduction

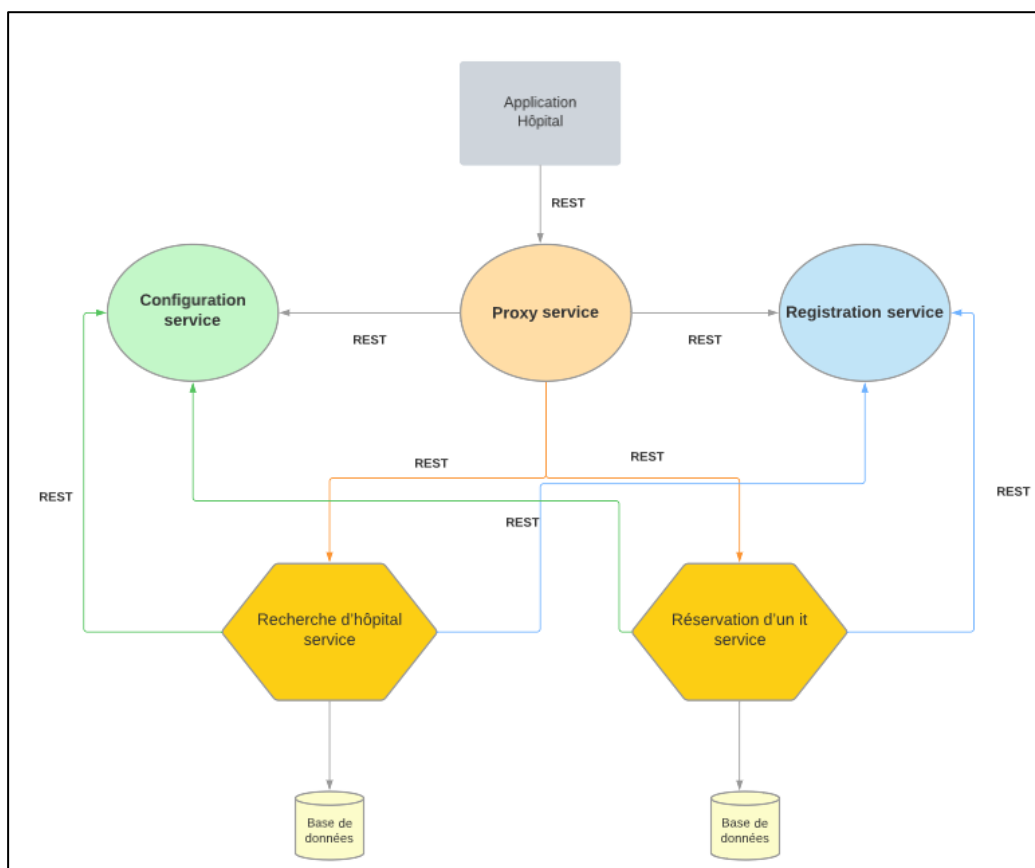
L'objectif du projet s'est concentré sur le système d'intervention d'urgence qui permet l'attribution en temps réel de lits d'hôpital en fonction de la pathologie.

En effet, le système d'intervention d'urgence en temps réel est destiné à recevoir une ou plusieurs spécialités médicales et une banque de données d'informations récentes sur les hôpitaux afin de suggérer l'hôpital le plus proche offrant un lit disponible, associé à une ou plusieurs spécialisations correspondantes. Le lieu de l'incident d'urgence doit également être fourni.

Par conséquent, nous avons mis en place deux micro-services :

- Gestion de recherche d'hôpitaux
- Gestion de réservation de lit

II. Architecture et communications des micro-services



- **Configuration service** : permet de centraliser toutes les configurations nécessaires pour l'ensemble des micro-services

- **Service d'enregistrement** : permet d'enregistrer la localisation de chaque micro-service.
C'est un annuaire qui publie l'ensemble des informations des micro-services (nom, adresse IP, numéro de port)
- **Service proxy** : permet d'orchestrer l'ensemble des fonctionnalités des micro-services.

Chaque micro-service avant de démarrer, il envoie une requête http REST vers le *service de configuration* pour récupérer ces paramètres de configuration et puis il démarre.

Le micro-service va ensuite se connecter au *service d'enregistrement* pour publier son nom, son adresse IP et son numéro de port.

En effet, toutes les requêtes http des applications (web ou mobiles), vient vers le *service proxy* en fournissant le nom du micro-service souhaité. Avec ce dernier le service proxy contact, l'annuaire pour obtenir les informations du micro-service (port, adresse IP). Ensuite, le service va envoyer la requête vers le micro-service concerné. Le service proxy va ensuite récupérer le résultat pour envoyer au client.

III. Les outils et dépendances utilisés

Sur le projet nous avons utiliser plusieurs outils et technologies :

	Outils/technologies
Développement backend et frontend du PoC	IDE Eclipse
	Maven
	Spring boot
	Jpa
	Proxy
	Spring apiGateway
	Thymleaf
Base de données	SQL Server
Test du Poc	JUnit
	Jenkins
Documentations	Swagger (documentation API)
	Jacoco (rapport des tests)

IV. Description des micro-service

A. Micro-service gestion des hôpitaux

Ce service permet de rechercher un hôpital en fonction de la spécialisation, la localisation et du nombre de lits disponibles.

En effet, le personnel médical pourra saisir sur la plateforme sa localisation et doit choisir la spécialisation sur une liste déroulante. Suite aux informations saisies , un hôpital doit lui être proposé.

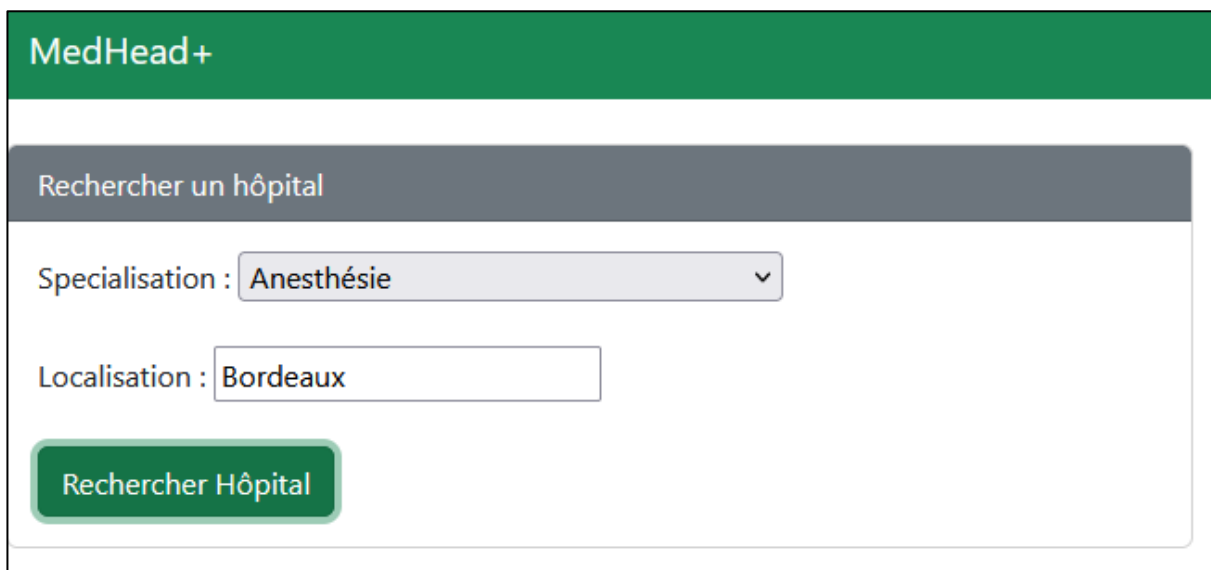
A l'exemple :

ET un patient nécessitant des soins en cardiologie.

QUAND le personnel sélectionne la spécialisation « cardiologie » ET que l'urgence est localisée à Bordeaux ALORS l'hôpital de Bordeaux va être proposé.

Extrait de l'interface utilisateur gestion des hôpitaux:

- Rechercher un l'hôpital



The screenshot shows the MedHead+ application interface. At the top is a green header with the text "MedHead+". Below this is a grey bar with the text "Rechercher un hôpital". The main area contains two input fields: "Specialisation :" with a dropdown menu showing "Anesthésie" and a downward arrow, and "Localisation :" with a text box containing "Bordeaux". At the bottom left of this area is a green button with the text "Rechercher Hôpital".

- Proposition de l'hôpital de la ville avec l'option réservation de lit :

MedHead+

Information sur l'hôpital

Nom hôpital : CHU Bordeaux

Adresse hôpital : 23 rue andré reinson,33300 Bordeaux

Specialité : Anesthésie

Nombre de lit disponible : 9

Reservation d'un lit

Nom du patient :

Prenom du patient :

Reserver un lit

B. Micro-service réservation de lit

Ce service permet d'attribuer en temps réel de lits d'hôpital en fonction de la pathologie et de l'hôpital. En cas de proposition d'un hôpital, la réservation du lit va être réalisée de manière automatique.

En effet, quand le personnel aura reçu la proposition d'un hôpital, si un lit est disponible et que le personnel choisi de lui réserver un lit, le micro-service va être en mesure de réserver un lit. Un évènement est déclenché automatiquement pour réserver un lit. Le personnel doit saisir le nom et prénom du patient pour réserver le lit en fonction du patient.

Si l'hôpital est réservé, le micro-service va ensuite réduire le nombre de lits disponibles dans l'hôpital ou la réservation a été effectuée (en fonction de la spécialisation).

Le service réservation de lit appelle et utilise le service gestion des hôpitaux afin de pouvoir réduire le nombre de lits disponibles au moment de la réservation de lit.

A l'exemple :

Etant donné qu'il y a un hôpital qui est proposé

Quand on saisit le nom et prénom du patient

Et qu'on clique sur le bouton réserver

Alors le système déclenche un événement pour réserver un lit en fonction du nom du patient.

Extrait de l'interface utilisateur réservation de lit :

- Réservation du lit

The screenshot shows a web application interface for 'MedHead+'. It features a green header with the logo. Below the header, there are two main sections. The first section, titled 'Information sur l'hôpital', contains four lines of text: 'Nom hôpital : CHU Bordeaux', 'Adresse hôpital : 23 rue andré reinson,33300 Bordeaux', 'Specialité : Anesthésie', and 'Nombre de lit disponible : 9'. The second section, titled 'Reservation d'un lit', contains two input fields: 'Nom du patient : Hamon' and 'Prenom du patient : Clara'. At the bottom of this section is a green button labeled 'Reserver un lit'.

MedHead+

Information sur l'hôpital

Nom hôpital : CHU Bordeaux

Adresse hôpital : 23 rue andré reinson,33300 Bordeaux

Specialité : Anesthésie

Nombre de lit disponible : 9

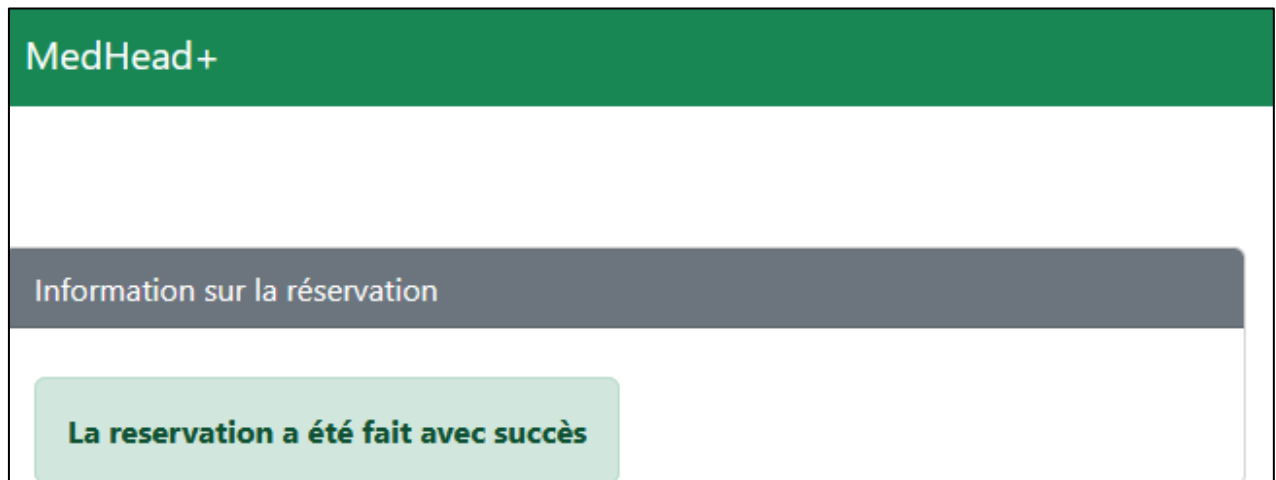
Reservation d'un lit

Nom du patient :

Prenom du patient :

Reserver un lit

- Réserve automatique avec un message de succès :



C. Service configuration

Ce service permet de centraliser toutes les configurations nécessaires pour l'ensemble des micro-services. Un fichier a été créer pour chaque micro-service et contient les informations de sa configuration.

L'application avant de démarrer complètement, il va d'abord consulter se fichier et récupère les informations nécessaires pour son démarrage et son fonctionnement (exemple pour récupérer son numéro de port).


Exemple : Le fichier du service réservation de lit va contenir :

- Son numéro de port
- L'url de connexion de sa base de données
- Le nom type de base de données utilisé
- ...

D. Service d'enregistrement

Ce service permet d'enregistrer la localisation de chaque micro-service. C'est un annuaire qui publie l'ensemble des informations des micro-services (nom, adresse IP, numéro de port).

Extrait du registre avec le nom du micro-service et son numéro de port :

Toggle navigation

System Status

Environment	test	Current time	2022-08-30T12:19:19 +0200
Data center	default	Uptime	00:02
		Lease expiration enabled	false
		Renews threshold	6
		Renews (last min)	6

DS Replicas

[localhost](#)

Instances currently registered with Eureka

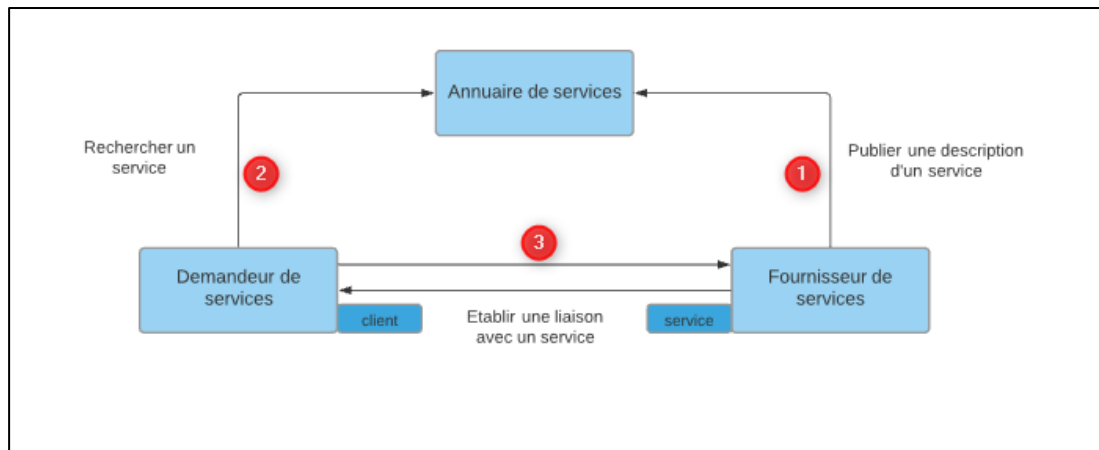
Application	AMIs	Availability Zones	Status
GESTION-HOPITAL 1	n/a (1)	(1)	UP (1) - LPT-CHAI.home:gestion-hopital:8084
PROXY 2	n/a (1)	(1)	UP (1) - LPT-CHAI.home:proxy:8083
RESERVATION-LIT 3	n/a (1)	(1)	UP (1) - LPT-CHAI.home:reservation-lit:8081

E. Service API Gateway (proxy)

Ce service permet d'orchestrer l'ensemble des fonctionnalités des micro-services. En effet, si l'application envoie des requêtes http, au lieu d'aller directement demander les réponses aux différents micro-services (gestion hôpitaux et réservation de lit), il va d'abord interroger l'API Gateway (proxy) puis ce dernier va interroger par la suite le service d'enregistrement qui va lui fournir les informations de configuration du service souhaité par la demande (exemple le numéro de port).

En effet, c'est le point d'entrée unique pour les API et micro-services. En effet, elle permet d'agréger différents micro-services. L'agrégation des requêtes sera faite directement au travers de l'API Gateway, cela permet de réduire la charge réseau et la multiplication des appels.

Exemple :



- *Le demandeur de service* : API Gateway
- *L'annuaire de service* : le service d'enregistrement
- *Fournisseur de service* : est les micro-services (gestion hôpitaux et réservation de li

V. Quelques exemples de différents d'Endpoint des services

Composants	Endpoint	Description
Rechercher Hopital	GET/specialisations	Obtenir une liste des spécialisations
	GET/hopitals/{codeSpecialisation}/{localisation}	Obtenir un hôpital en fonction de la spécialisation, la localisation et du nombre de lit disponible
	PUT/nombreLit/{codeHopital}/{codeSpecialisation}	Réduire le nombre de lit de l'hôpital recherché
Réservation lit	PUT/reservationLit/{codeHopital}/{codeSpecialisation}/{nomPatient}/{prenomPatient}	Réserver un lit d'hôpital

Extrait d'un exemple de la documentation Swagger :

The screenshot shows a Swagger UI interface. At the top, the Swagger logo is on the left, and the URL `/v3/api-docs` is in a search bar on the right. Below the header, the text 'OpenAPI definition' is displayed in large blue font, with 'v0' and 'OAS3' in small green circles to its right. Underneath, the URL `/v3/api-docs` is shown in a smaller blue font. A 'Servers' section contains a dropdown menu with the value `http://localhost:8084 - Generated server url`. The main title of the API is 'gestion-hopital-controller'. Below this, three API endpoints are listed in colored boxes: a PUT endpoint `/urgences/nombreLit/{hopital}/{specialisation}` in an orange box, a GET endpoint `/urgences/specialisations` in a light blue box, and another GET endpoint `/urgences/hopitals` in a light blue box.

VI. Axes d'améliorations

- Mise en place d'une API externe « L'API Distance Matrix » :
 - Elle proposera l'hôpital le plus proche de la localisation : il va pouvoir comparer les distances des hôpitaux qui sont autour de la localisation et proposer l'hôpital le plus proche
 - Elle fournira la distance et la durée des déplacements en temps réel