

DOCUMENTATION : SYSTEME DE RESERVATION DISTRIBUE

Mise en situation :

Contexte : Un groupe hôtelier togolais, présent sur tous les continents, souhaite développer une application web centralisée pour la gestion des réservations

Objectifs : Permettre à un client de réserver des chambres dans n'importe quel établissement du groupe depuis n'importe quelle agence locale

Fonctionnement : Un client se rend dans une agence au Togo pour réserver un séjour à Paris. L'agent utilise l'application web qui accède aux mêmes bases de données centralisées que toutes les autres agences mondiales, vérifie depuis son interface toutes les chambres disponibles de tous les établissements puis réserve une chambre disponible au nom du client

Avantage : La réservation est instantanément enregistrée et synchronisée dans le système global, garantissant une gestion cohérente et unifiée de toutes les agences

Résultat : le groupe offre ainsi une expérience cliente homogène et une visibilité en temps réel sur la disponibilité de l'ensemble de ses établissements partout dans le monde

NB : Nous nous intéresserons uniquement à la gestion des réservations dans ce projet

Les raisons du choix de l'architecture distribuée pour ce projet :

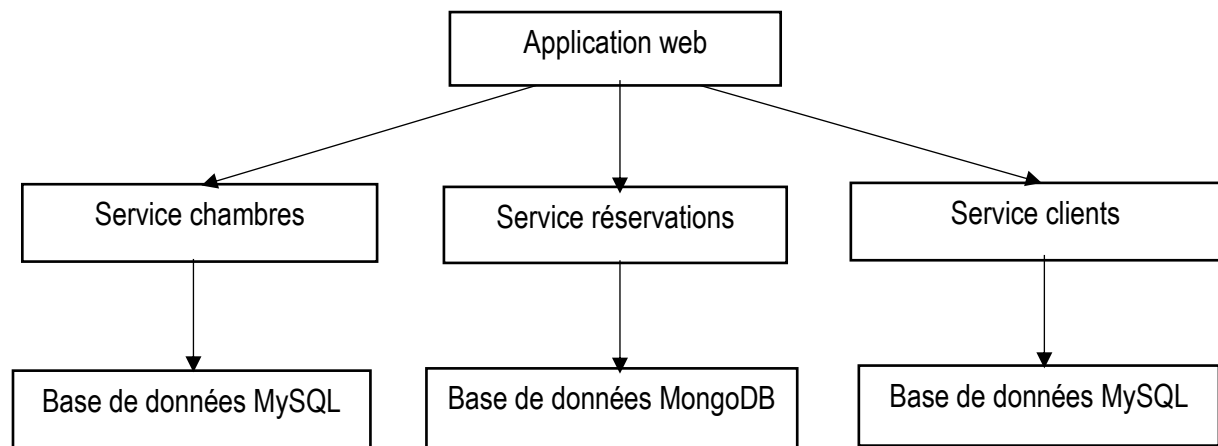
- Sécurité et conformité réglementaire
- Besoin de performance spécifique
- Maintenance et évolution
- Résilience et disponibilité
- Coûts et ressources

Synthèse des trois bases de données :

Bases	Types	Usages	Flexibilités
Clients	MySQL	Données sensibles	Faibles
Chambres	MySQL	Catalogue stable	Moyenne
Réservations	MongoDB	Données complexes évolutives	Elevée

Architecture système :

Schéma d'architecture



Technologies utilisées

Frontend : HTML, CSS, JavaScript

Backend : Python Flask

Base de données : MySQL, MongoDB

Communication : http/REST API

Architecture : micro services

MICROSERVICES

- Service chambre : Port 5001

Rôle : gestion du catalogue des chambres

Fichier : services_chambres.py

Fonctions :

- Liste des chambres disponibles
- Mise à jour de la disponibilité
- Gestion des prix

- Service client : Port 5002

Rôle : gestion des clients

Fichier : services_clients.py

Fonctions :

- Consultation des clients
- Vérification d'existence

- Service réservation : Port 5003

Rôle : gestion du cycle de vie des réservations

Fichier : services_reservation.py

Fonctions :

- Création des réservations
- Annulation des réservations
- Consultation des réservations

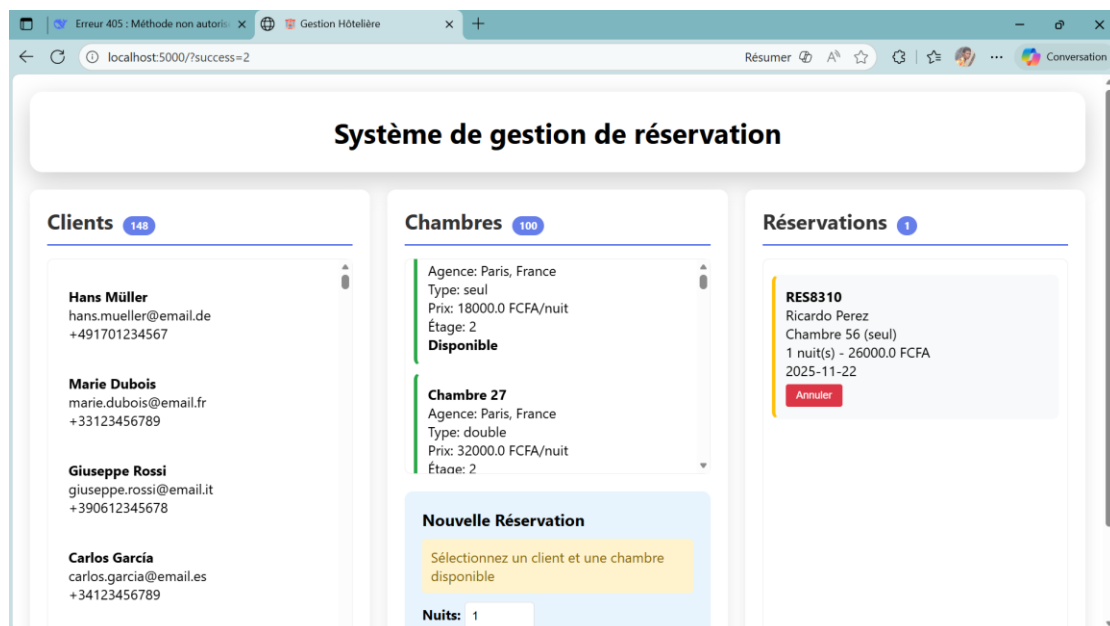
- Application web : Port 5000

Rôle : interface utilisateur et orchestration des services

Fichier : app_web.py

Fonctions :

- Affichage des données consolidées
- Gestion des formulaires de réservation
- Orchestration entre les micro services
- Gestion des erreurs



- Tableau de bord : Port 5005

Rôle : monitoring du système

Fichier : dashboard.py

Fonctions :

- Statut des services en temps réel
- Visualisation des métriques

