

5 Février 2026



Présentation Technique

Plateforme Freelance: Architecture Microservices

Notre équipe



Ridha Ferchichi



Ela Tebourbi



Yassine Zemzem



Baha Edine
Ferjeni



Oussama Abdallah



Ahmed Iheb
Ayed

Sommaire

1-Titre du projet

2-Objectif du projet

3-Architecture globale

4-Description des microservices

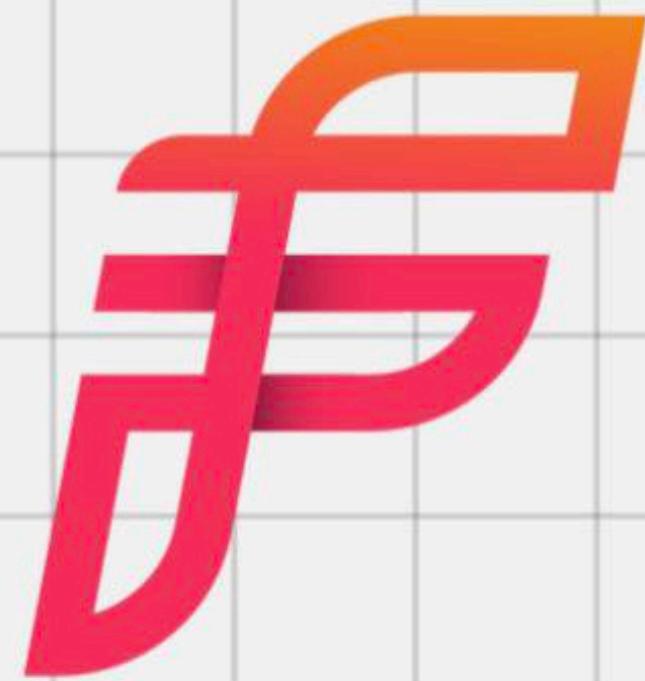
5-Communication & flux

6-Infrastructure & déploiement

7-Conclusion

Titre du projet

Freelancia - Smart Freelance & Project Matching
Platform



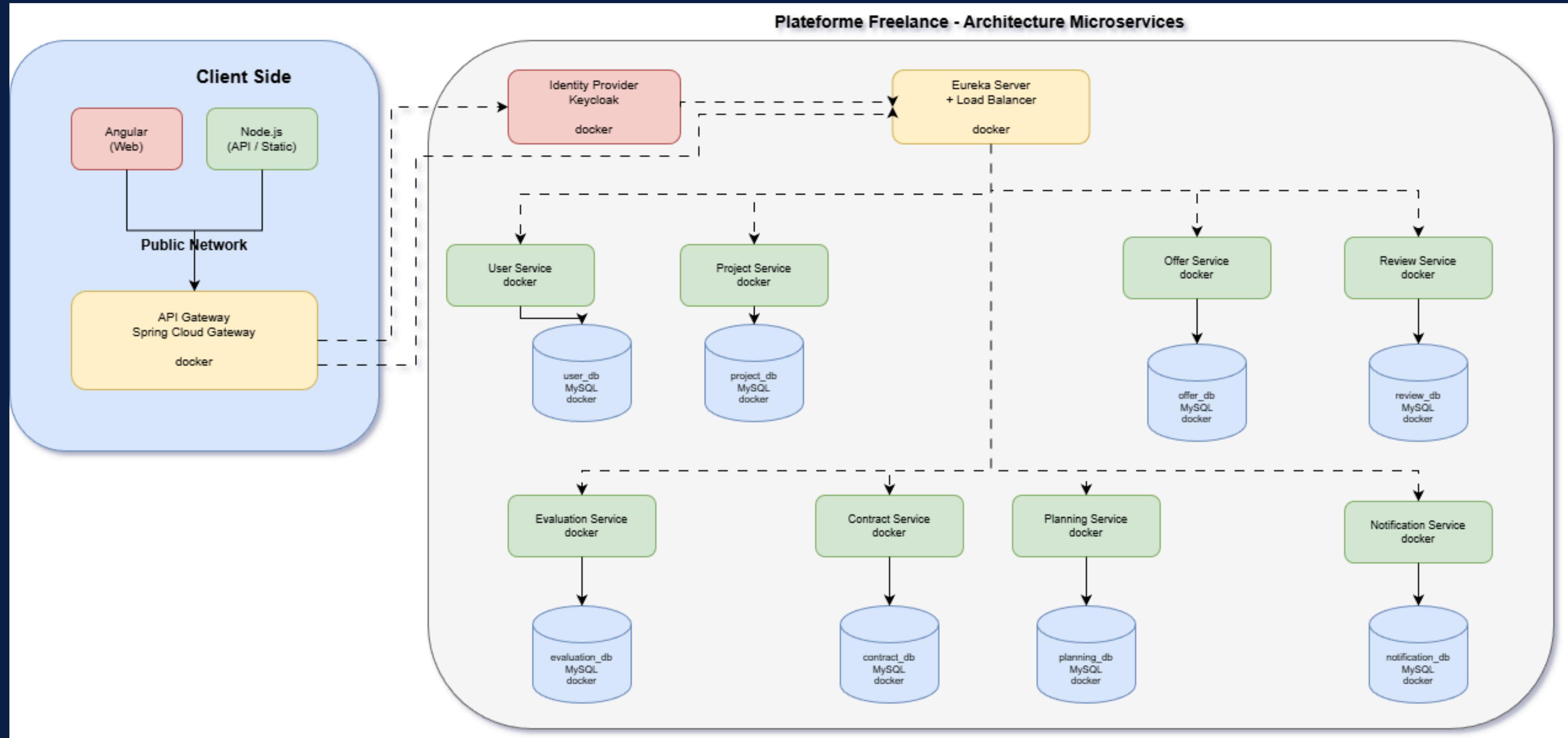
FREELANCIA

Objectif du projet

Freelancia est une plateforme digitale qui met en relation des freelancers et des entreprises grâce à un système de matching intelligent basé sur les compétences, le budget, la disponibilité et les besoins du projet.

L'objectif est de faciliter l'accès à des opportunités professionnelles, d'optimiser le recrutement freelance et de soutenir l'économie numérique.

Architecture globale



Description des microservices

User Management Microservice :

Ce microservice est responsable de la gestion complète des utilisateurs de la plateforme Freelancia.

Responsabilités :

- Inscription et authentification des utilisateurs
- Gestion des rôles : Client, Freelancer, Administrateur
- Création et mise à jour des profils utilisateurs
- Gestion des autorisations et du contrôle d'accès

Project Management Microservice :

Ce microservice permet aux clients de gérer leurs projets et aux freelancers d'y postuler.

Responsabilités :

- Création et publication des projets par les clients
- Gestion des candidatures via l'entité ProjectApplication
- Suivi de l'état des projets : Open, In Progress, Completed, Cancelled
- Consultation et mise à jour des informations liées aux projets

Description des microservices

Offer Management Microservice :

Ce microservice gère les offres de services proposées par les freelancers.

Responsabilités :

- Création et gestion des offres de services par les freelancers
- Postulation des clients via OfferApplication
- Suivi du statut des offres : Available, Accepted, Expired
- Mise à jour et suppression des offres

Reviews Management Microservice :

Ce microservice est dédié à la gestion des avis et de la réputation des utilisateurs.

Responsabilités :

- Création d'avis après une collaboration entre utilisateurs
- Réponses aux avis reçus
- Calcul et suivi des notes et de la réputation globale
- Consultation de l'historique des avis

Description des microservices

Evaluation Management Microservice :

Ce microservice permet l'évaluation des compétences des freelancers.

Responsabilités :

- Définition des compétences et des critères d'évaluation
- Gestion des tests d'évaluation
- Évaluation des freelancers selon leurs résultats
- Attribution de certifications basées sur les performances

Contract Management Microservice :

Ce microservice assure la gestion contractuelle entre clients et freelancers.

Responsabilités :

- Génération automatique de contrats numériques après acceptation
- Gestion du workflow de signature des contrats
- Suivi de l'état des contrats
- Gestion des litiges via des réclamations

Description des microservices

Planning and Tracking Microservice :

Ce microservice permet le suivi de l'avancement des projets.

Responsabilités :

- Publication des mises à jour d'avancement par les freelancers
- Commentaires et validations par les clients
- Suivi des étapes et du progrès global des projets
- Historique des actions liées au projet

Notification Management Microservice :

Ce microservice gère la communication événementielle avec les utilisateurs.

Responsabilités :

- Envoi de notifications basées sur des événements (in-app, email, push)
- Gestion des préférences de notification des utilisateurs
- Génération de messages à partir de modèles
- Notification des actions importantes (acceptation, signature, avis, etc.)

Communication & flux

Types de communication

- REST API : communication synchrone entre microservices
- Message Broker (RabbitMQ / Kafka) : communication asynchrone
- JSON comme format d'échange

Infrastructure & déploiement

Conteneurisation

- Chaque microservice est conteneurisé avec Docker
- Images indépendantes pour chaque service

Bases de données

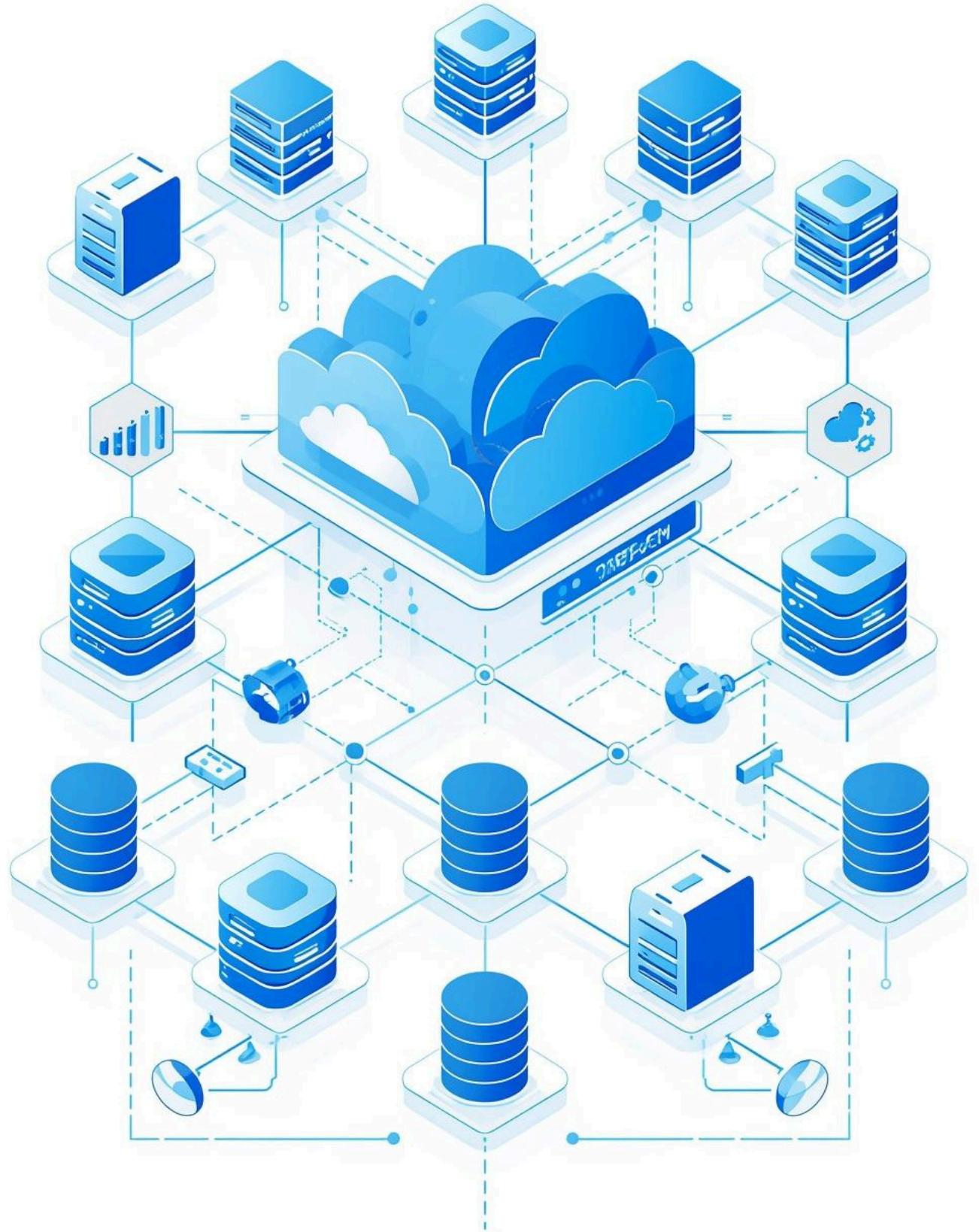
- Une base de données par microservice
- MySQL / PostgreSQL selon le besoin

Déploiement

- Docker Compose (en environnement académique)
- Possibilité d'évolution vers Kubernetes

Communication avec le Front-end

- Front-end Web (Angular)
- Communication via API REST sécurisée (JWT)



Conclusion

Le choix de l'architecture microservices pour Freelancia permet :

- Une meilleure évolutivité du système
- Une séparation claire des responsabilités
- Une maintenance simplifiée
- Une base solide pour l'intégration future de l'IA et du Big Data

Cette architecture soutient pleinement la vision de Freelancia comme une plateforme moderne, scalable et intelligente.

