

DOSSIER ARCHITECTURE TECHNIQUE SHOT YOUR PET



Sommaire

- Présentation de l'application
- Détail du projet
- Schéma Architecture Technique
- Description technique

Shot Your Pet

Présentation de l'application

Shot Your Pet est une application inspirée du concept de BeReal.

Le principe est simple : chaque jour, l'utilisateur prend une photo avec son animal de compagnie.

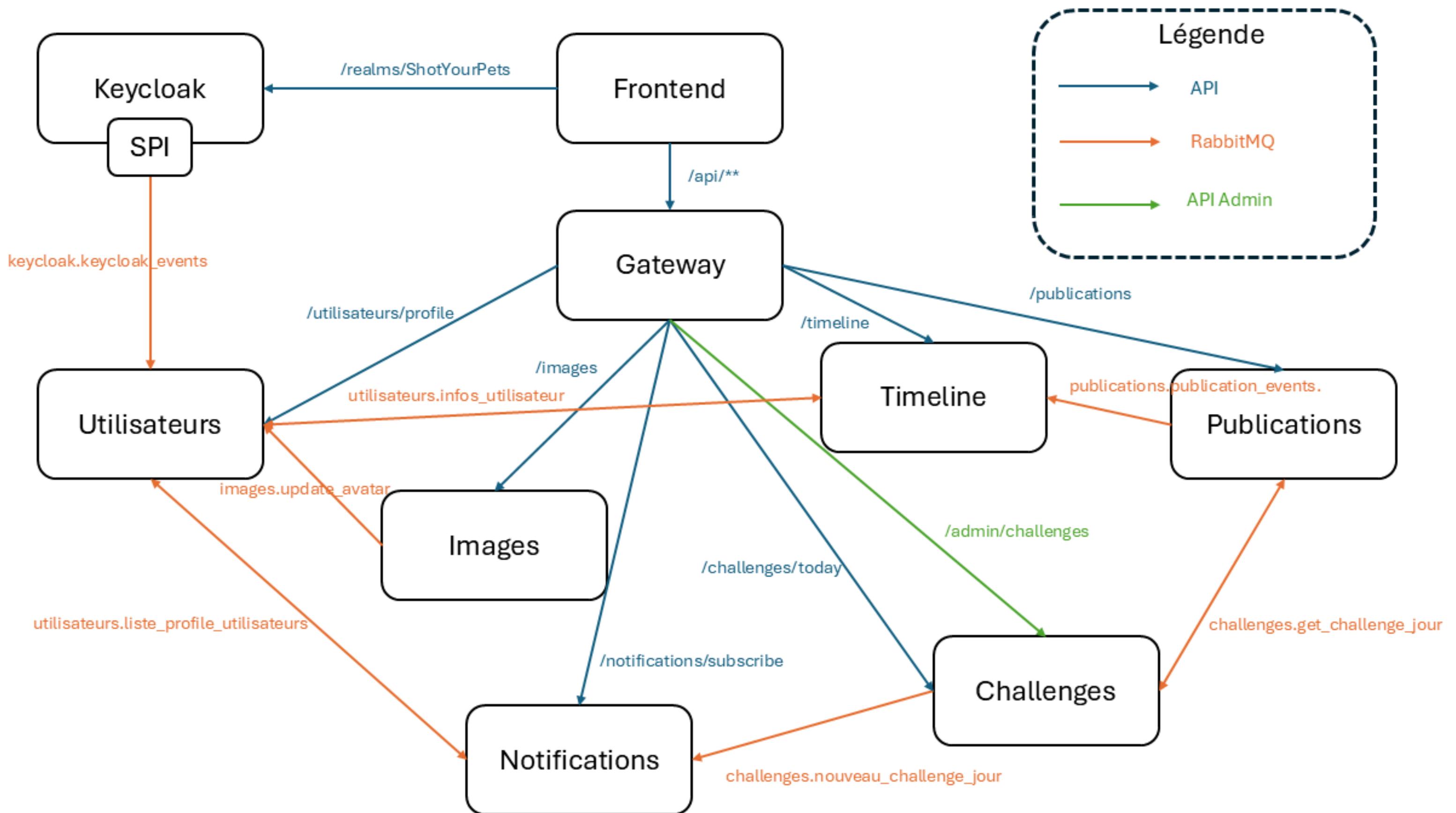
À 18h, une notification lui propose un petit défi à relever avec son compagnon.

Une fois la photo prise, il peut découvrir celles des autres utilisateurs. Chaque cliché est ensuite archivé dans un calendrier, permettant de revoir tous les souvenirs partagés au fil du temps.



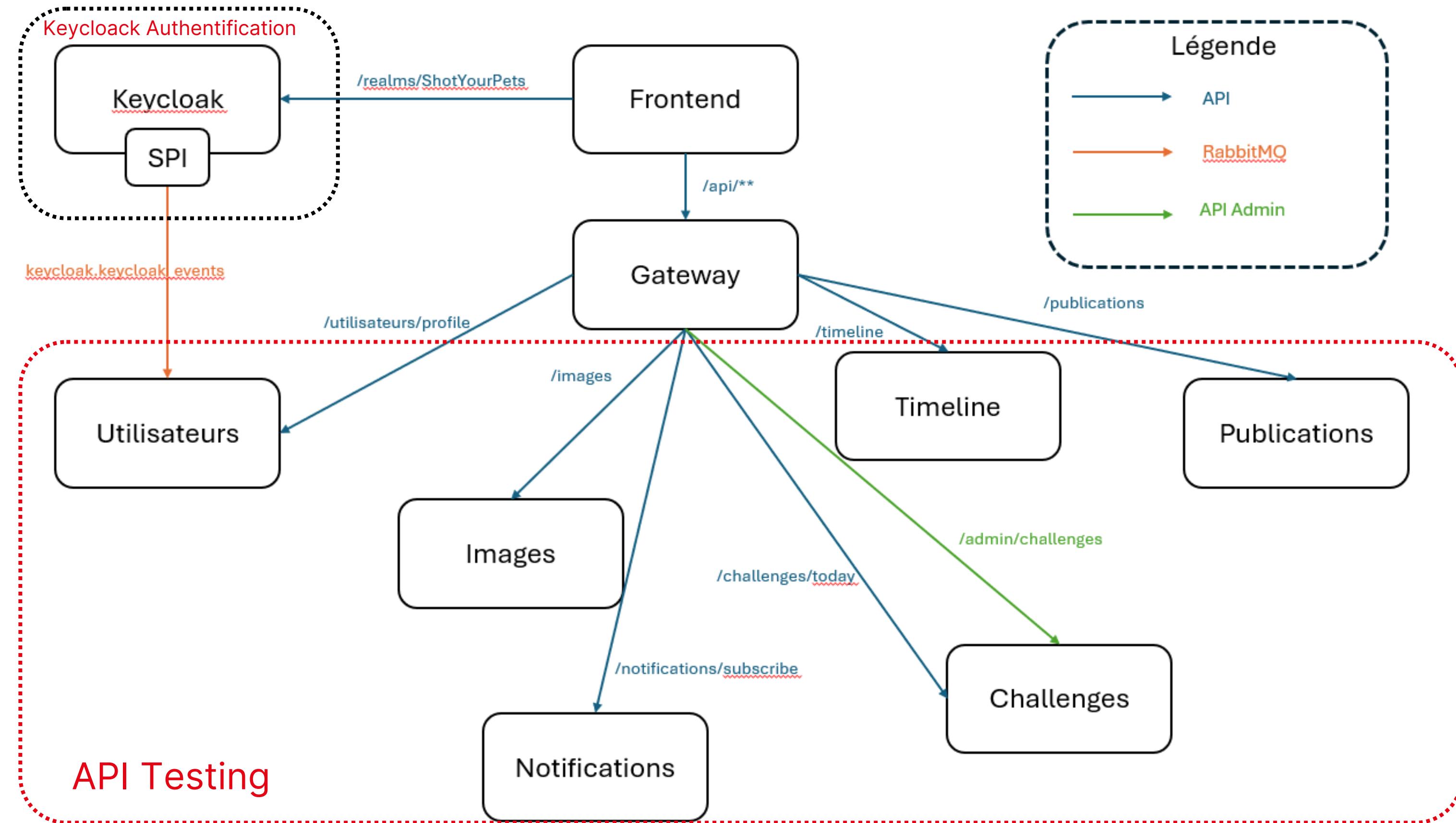
Détails du projet

Architecture globale



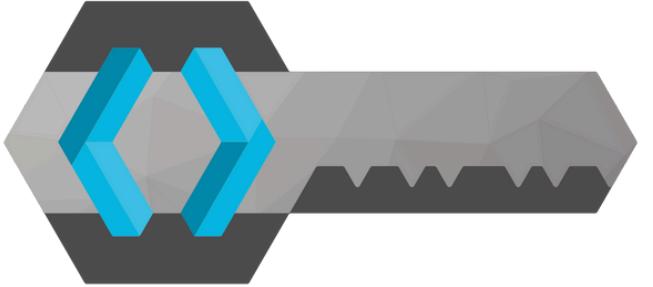
Détails du projet

Architecture de tests d'intégration



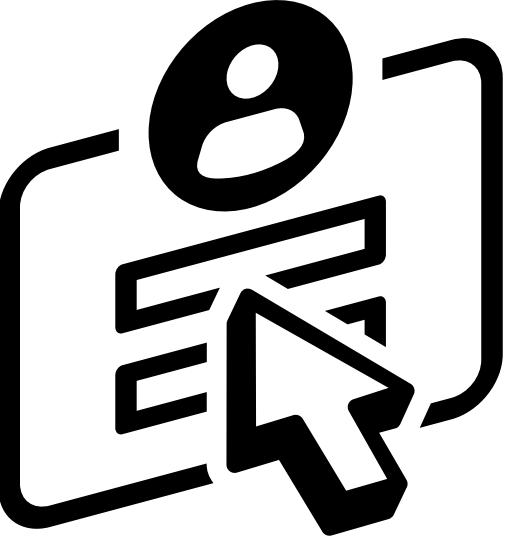
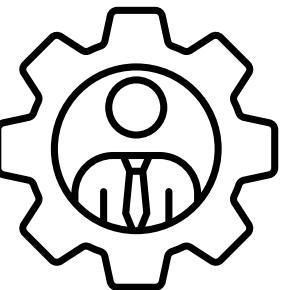
Détails du projet

Microservices



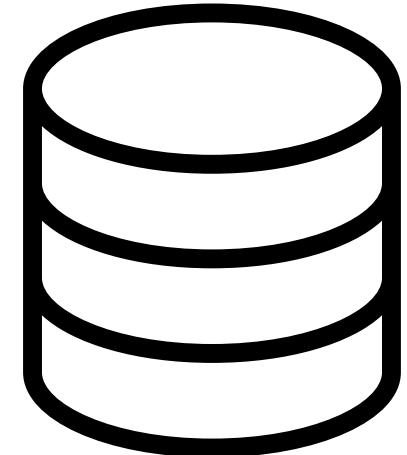
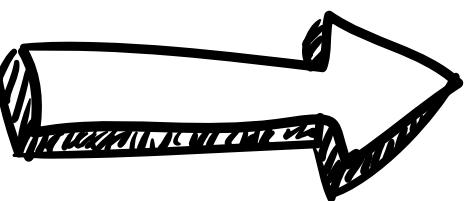
KEYCLOAK

Gestion des rôles :
Administrateurs
Utilisateurs
Modérateurs



**Micro service
Authentification**

Autogérée

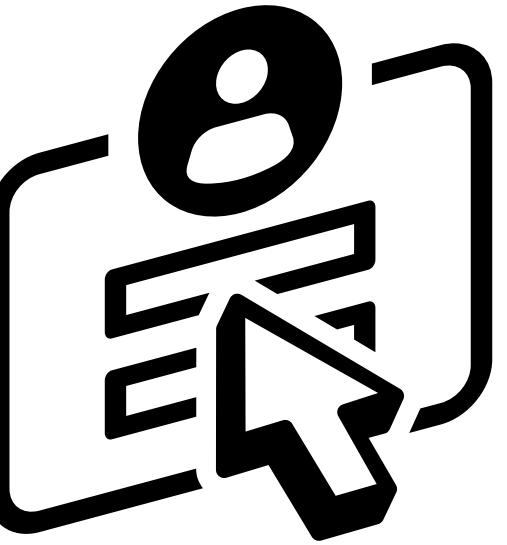
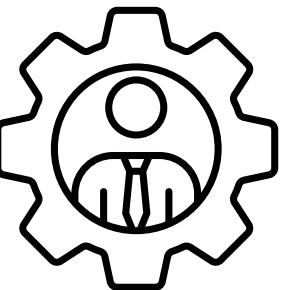


Détails du projet

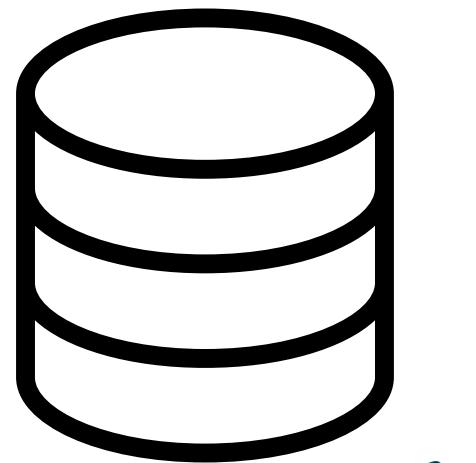
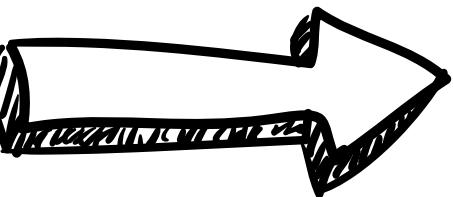
Microservices



Lien avec Keycloak par interception
d'évènement par SPI sur queue
RabbitMQ



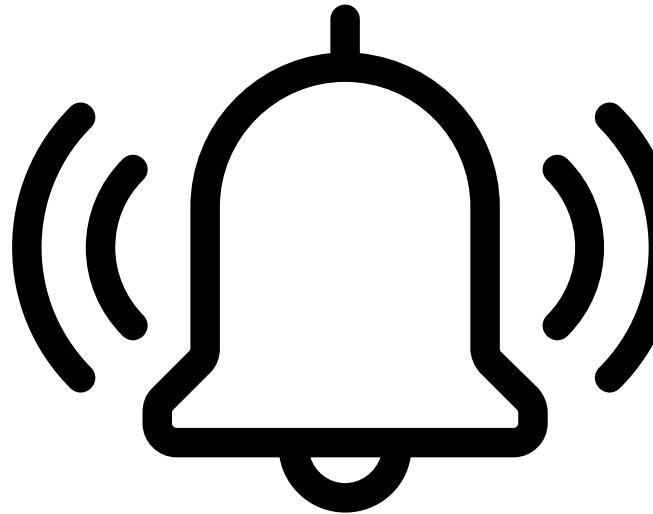
**Micro service
Utilisateurs**



MariaDB

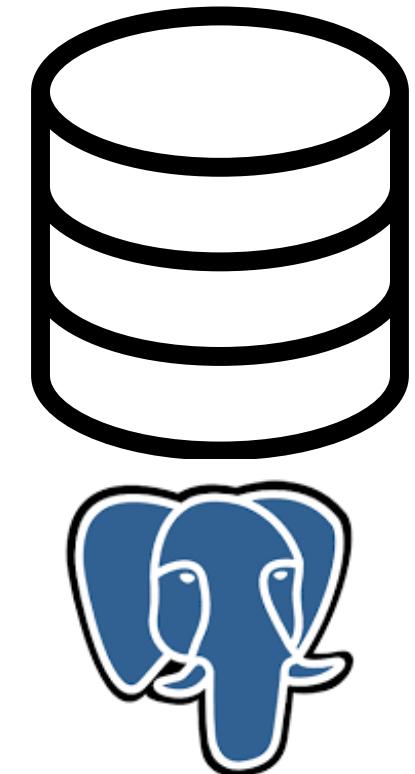
Détails du projet

Microservices



Microservice Notifications

clé (endpoint Navigateur)
liste des subscriptions de
notification



Fonctionnement :

- Récupération des événements Rabbit MQ
- Récupération des nouveaux challenges pour envoie de notifications Push et Email

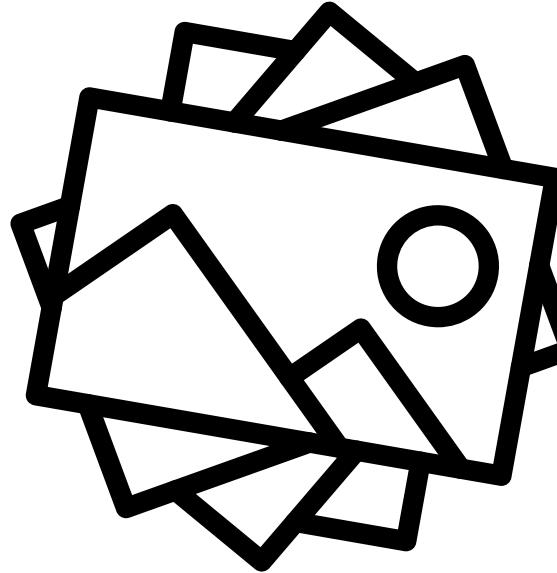
Détails du projet

Microservices

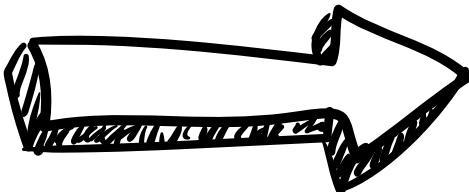


Fonctionnement :

- Upload des images
- Download des images
- Stockage des images



Microservice Gestion des images

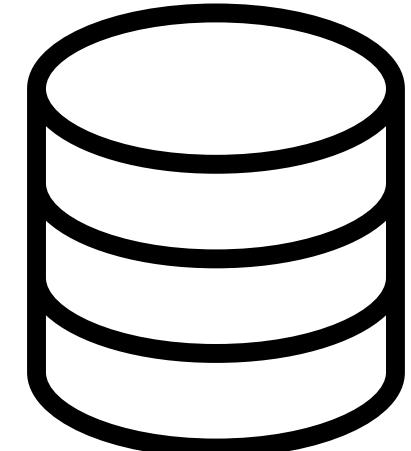


Utilisateur

Date

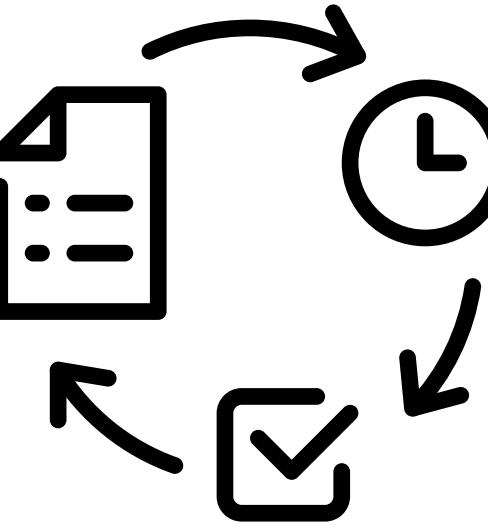
Image stocké sous FS

Snowflakes ID



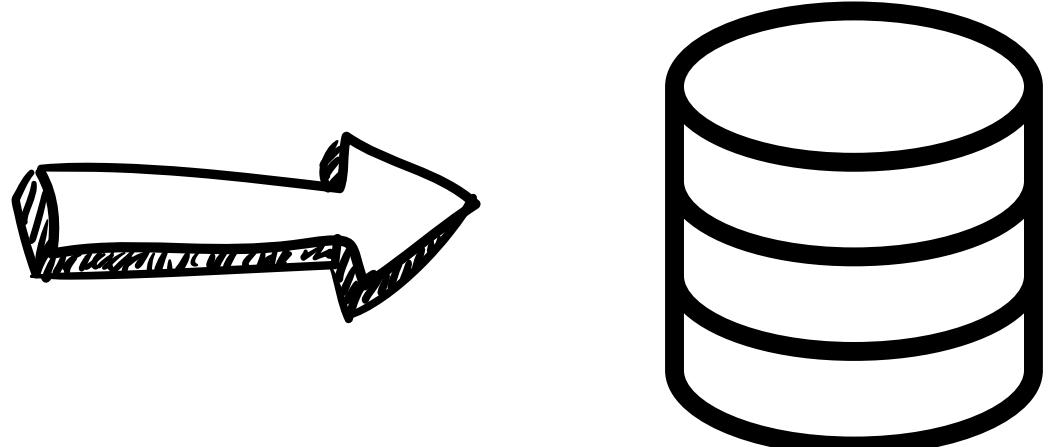
Détails du projet

Microservices



Microservice Gestion des challenges

Liste des challenges
des jours précédents



Fonctionnement :

- Générer une idée de challenge dans une liste
- Envoie d'un challenge aléatoire provenant de la liste défini par un cron job



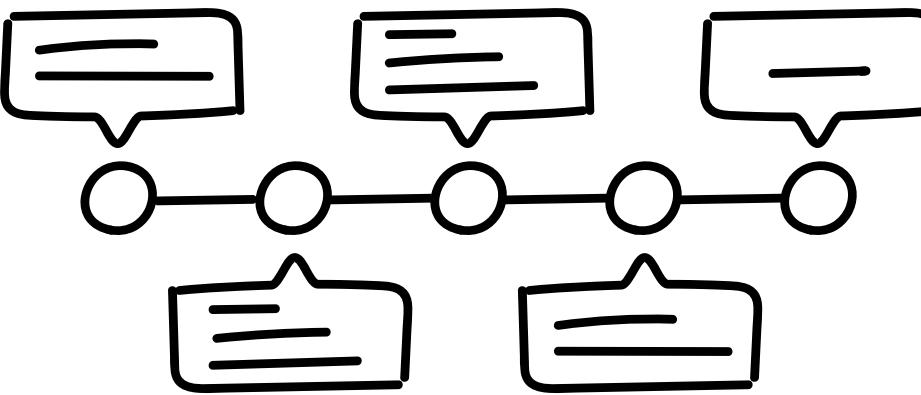
Détails du projet

Microservices

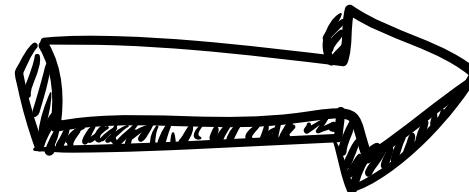


Fonctionnement :

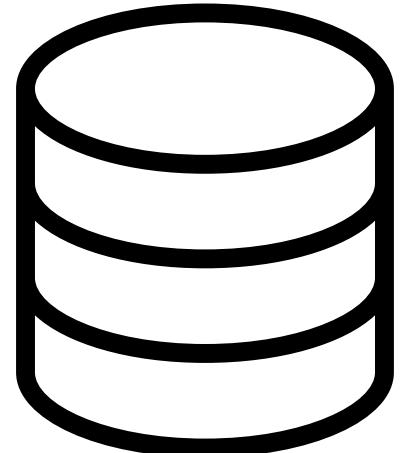
- Génération des fils d'actualité
 - Entre amis
 - Chronologie
 - Filtres d'intérêts



Microservice Timeline



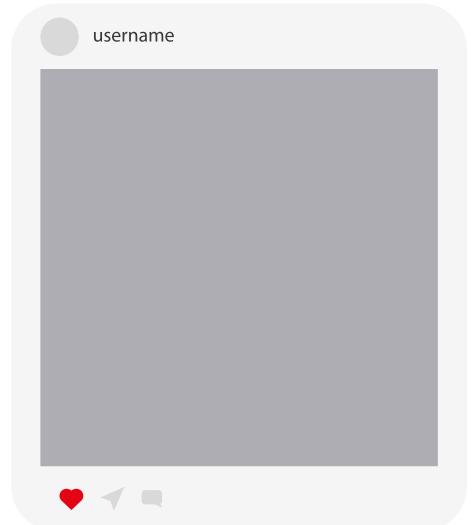
Post
Utilisateur (cache)



PostgreSQL

Détails du projet

Microservices

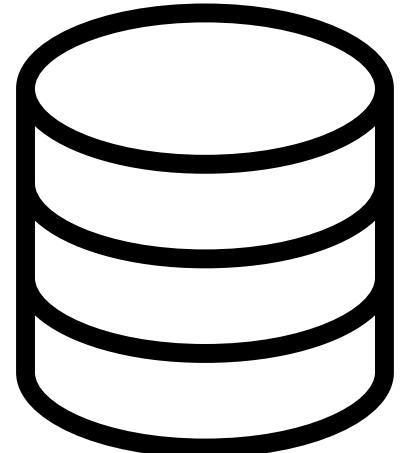


Microservice Publication

- Fonctionnement :
- Génération des posts
 - Titre
 - Description
 - Date
 - Heure
 - Modération

Titre
Description
Date

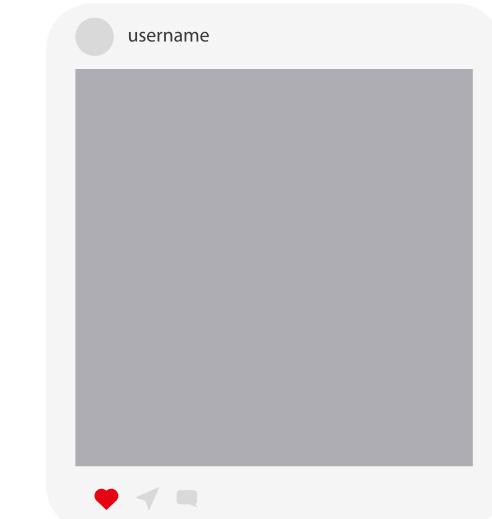
Image en bit (ID)



mongoDB

Détails du projet

Microservices



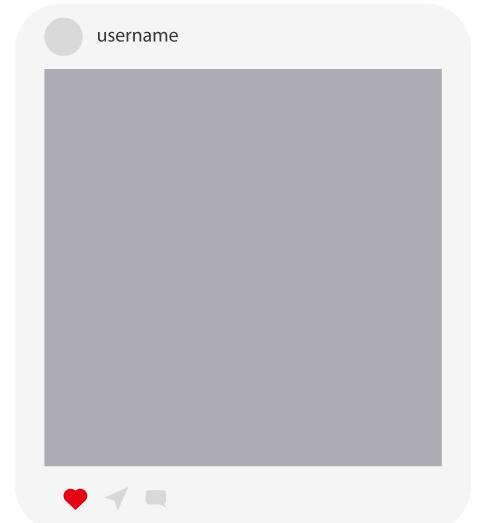
**Microservice
Frontend**



Interface utilisateur servit par un Nginx

Détails du projet

Microservices



**Microservice
Files d'échanges amqp**

 RabbitMQ

Interface utilisateur servit par un Nginx

Détails du projet

Microservices



Microservice Monitoring



Monitoring des logs de tous les services par Loki
Analyses des métrics par Prometheus
Affichage sous Grafana

Détails du projet

Microservices

Gateway



Fonctionnement :

- Rooting des requêtes vers les services

Détails du projet

Tests d'intégrations

Karate lab et JUnit 5



Gestion centralisé Maven pour permettre une implémentation future dans un processus automatisé, CI/CD, job Jenkins etc ..

Fonctionnement :

- Lancement et gestion par le runneur JUnit 5
- API testing
- Scénario de Test en langage BDD avec Karate
- Lancement local ou distant possible

Découpage et deux “Feature”:

- Test d'un module dépendant d'un service externe : Keycloak
- Test API