

# Platformă Distribuită pentru Organizarea Evenimentelor și Sincronizarea Calendarului de Grup

Alexandru Alin-Ioan

Grupa: 344C3

23 noiembrie 2025

## Cuprins

<b>1</b>	<b>Introducere</b>	<b>1</b>
<b>2</b>	<b>Obiectivele aplicației</b>	<b>1</b>
<b>3</b>	<b>Descrierea arhitecturii</b>	<b>2</b>
3.1	Componente principale . . . . .	2
3.2	Rețele și comunicare între microservicii . . . . .	2
<b>4</b>	<b>Modulele aplicației</b>	<b>2</b>
<b>5</b>	<b>Diagrama arhitecturii</b>	<b>3</b>

## 1 Introducere

Această aplicație web propune o soluție pentru un calendar distribuit prin care utilizatorii să poată crea, edita și sincroniza evenimente în cadrul unor grupuri. Platforma va oferi funcționalitatea de gestionare a disponibilității tutror participanților și de găsire automată de intrevale libere pentru evenimentele planificate.

## 2 Obiectivele aplicației

- Oferirea unui calendar multi-utilizator cu sincronizare în timp real.
- Gestionarea disponibilității și a evenimentelor unui participant sau unui grup.
- Prevenirea conflictelor de rezervare prin mecanisme de distributed locking.
- Generarea de propuneri automate de meeting folosind procesare asincronă și workeri replicați.
- Implementarea unor mecanisme robuste de autentificare, autorizare și management al rolurilor folosind Keycloak.
- Livrarea proiectului sub forma unui set de servicii Docker.

## 3 Descrierea arhitecturii

### 3.1 Componente principale

1. **Serviciul de autentificare (Keycloak)** – Folosește OAuth2 și emite token-uri JWT.
2. **Serviciul de profil utilizatori** – Preia informațiile oferite de Keycloak și creează profiluri locale în baza de date.
3. **Baza de date (PostgreSQL)** – destinată managementului și persistenței entităților și setărilor aplicației.
4. **Serviciul Calendar** – expune un API pentru management de grupuri, operații CRUD pentru evenimente, verificarea disponibilității.
5. **Serviciul Suggestion Worker (replicat)** – preia joburi de la broker și procesează în paralel cererile de găsire a unui interval comun pentru un grup.
6. **Message Broker (RabbitMQ)** – preia cererile de procesare a unui interval de la serviciul calendar și le trimită către workeri.
7. **Serviciul de locking distribuit (Redis)** – gestionează mecanismele de *distributed locking* necesare evitării conflictelor la nivel de interval orar prin algoritmul *RedLock*.

### 3.2 Rețele și comunicare între microservicii

## 4 Modulele aplicației

1. **Autentificare** – funcție de autorizare utilizând o tehnologie de Single Sign-On (SSO).
2. **Profil utilizator + roluri** – Keycloak pentru managementul utilizatorilor, rolurilor și token-urilor.
3. **Baza de date** – gestionată prin ORM.
4. **Conflict resolution + Distributed Locking** – gestionează situațiile concurente în care mai mulți utilizatori creează sau modifică evenimente în același interval temporal.
5. **Suggestion Worker replicat** – calculează automat intervalele comune libere pentru un grup. Joburile sunt transmise de către serviciul Calendar către broker, care mai apoi le redistribuie către workeri.

## 5 Diagrama arhitecturii

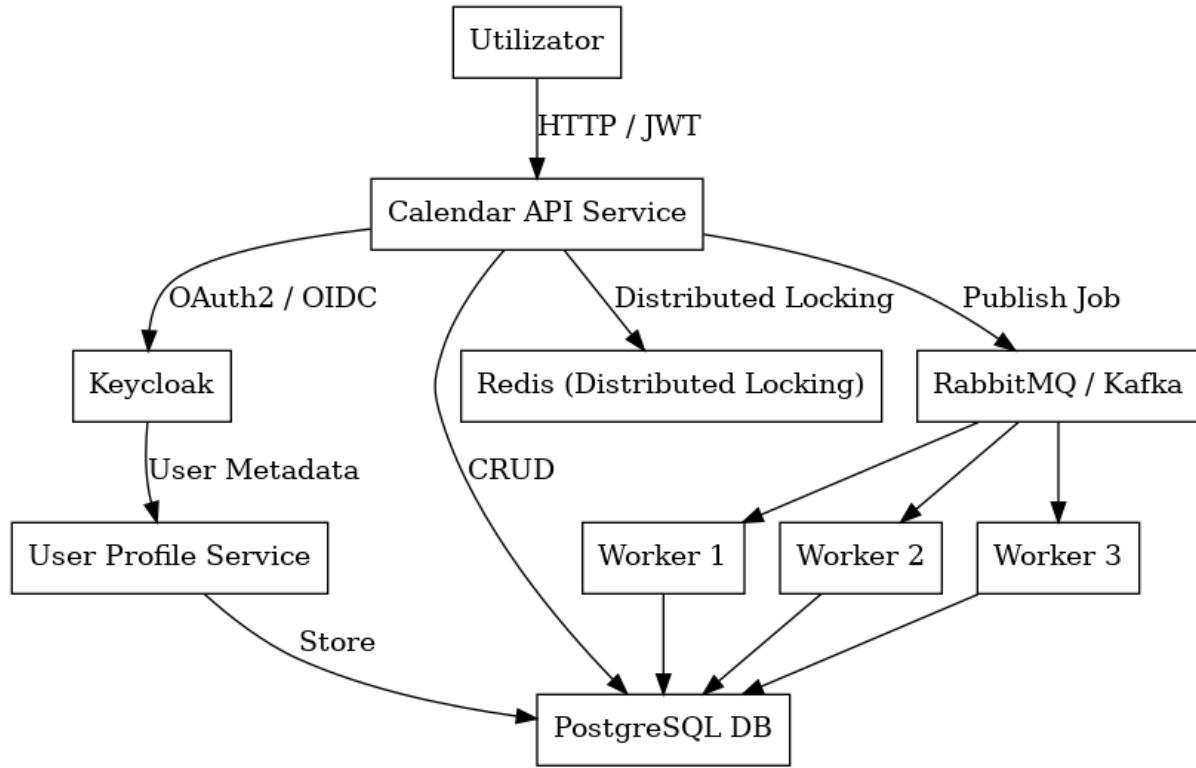


Figura 1: Arhitectura platformei de calendar distribuit