



Facultatea de Automatică și Calculatoare  
Sisteme distribuite

Documentație Assignment 2  
**Sensor Monitoring System and Real-Time Notification**  
An 2022/2023

**Student:**

Coblișan George

GR. 30241



## 1. Arhitectura conceptuală a unui sistem distribuit

Un sistem distribuit poate fi un aranjament de diferite configurații, cum ar fi mainframe, calculatoare, stații de lucru și minicalculatoare. Partajarea resurselor precum hardware, software și date este unul dintre principiile cloud computing. Cu diferite niveluri de deschidere către software și concurență, este mai ușor să procesați date simultan prin mai multe procesoare.

Organizațiile au apelat la sisteme de calcul distribuite pentru a gestiona explozia de generare de date și nevoile crescute de performanță a aplicațiilor. Aceste sisteme distribuite ajută companiile să se extindă pe măsură ce volumul de date crește. Acest lucru este valabil mai ales deoarece procesul de adăugare a hardware-ului la un sistem distribuit este mai simplu decât actualizarea și înlocuirea unui întreg sistem centralizat format din servere puternice.

Sistemele distribuite constă din mai multe noduri care lucrează împreună pentru un singur obiectiv. Aceste sisteme funcționează în două moduri generale și ambele au potențialul de a face o diferență uriașă într-o organizație.

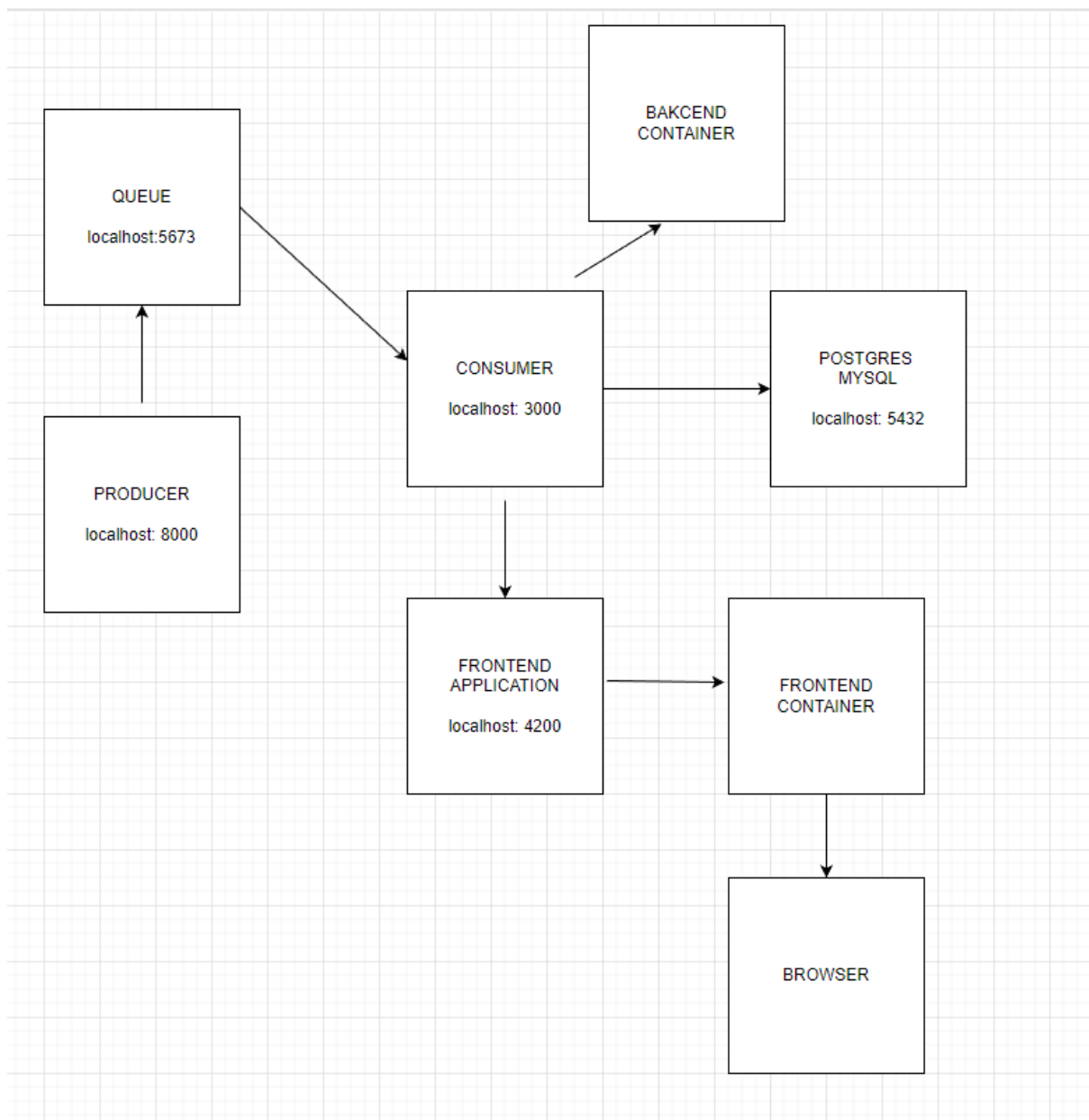
- Primul tip este un sistem coerent în care clientul are fiecare mașină, iar rezultatele sunt direcționate dintr-o singură sursă.
- Al doilea tip permite fiecărui nod să aibă un utilizator final cu propriile nevoi, iar sistemul distribuit facilitează partajarea resurselor sau comunicarea.

Pentru acest assignment am integrat RabbitMQ și Web Sockets. Pe scurt, RabbitMQ este un software de așteptare a mesajelor, cunoscut și ca broker de mesaje sau manager de cozi. Sau mai simplu, este un software la care sunt definite cozi, la care se conectează aplicațiile pentru a transfera un mesaj sau mesaje. Un mesaj poate include orice fel de informații. Ar putea, de exemplu, să aibă informații despre un proces sau sarcină care ar trebui să înceapă pe o altă aplicație (care ar putea fi chiar pe alt server) sau ar putea fi doar un simplu mesaj text. Software-ul de gestionare a cozii stochează mesajele până când o aplicație de primire se conectează și scoate un mesaj din coadă. Aplicația care primește apoi procesează mesajul. În cazul nostru, am creat o nouă aplicație așa numită "producer" în care am integrat RabbitMQ și procesează datele dintr-un fișier csv, urmând ca la un interval de câteva minute să trimită câte o valoare în coadă. Când aplicația anterioară care reprezintă backend-ul proiectului va fi pornită va începe să consume datele din coadă, rând pe rând.

WebSocket este bidirecțional, un protocol full-duplex care este utilizat în același scenariu de comunicare client-server, spre deosebire de HTTP, începe de la ws:// sau wss://. Este un protocol cu stare, ceea ce înseamnă că conexiunea dintre client și server va menține vie până când este terminată de oricare dintre părți (client sau server). După închiderea conexiunii de către client și server, conexiunea este încheiată de la ambele capete.



## 2. UML deployment diagram



MINISTERUL EDUCAȚIEI NAȚIONALE



---

**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

---