

Soluție bazată pe microservicii pentru platforme e-learning

Rezumat

Prin acest proiect se dorește studierea arhitecturii bazate pe Microservicii și implementarea unui prototip de aplicație care să poată fi scalată și întreținută ușor. Fiind o aplicație distribuită va oferi un timp de răspuns bun și pentru un număr mare de utilizatori care o accesează simultan. Aplicația va integra câteva microservicii, fiecare respectând principiul responsabilității singulare. Gestiunea accesului la funcționalitatea aplicației se va realiza prin intermediul unui gateway care reprezintă principalul punct de acces. Comunicarea dintre gateway și microservicii este realizată prin intermediul unui modul de tip Service Bus, bazat pe cozi de mesaje. Acest nivel este util pentru implementarea principiilor care stau la baza arhitecturii de tip microservicii: decuplarea, independența față de tehnologiile de implementare, dar și eventuala consistență a datelor. Prototipul realizat este o aplicație de tip e-Learning, deoarece aceasta se poate preta unei astfel de arhitecturi. Acest tip de aplicație, utilizată în facultăți și universități, poate avea un număr mare de utilizatori, deci și module supraîncărcate. Pentru a rezolva problemele de performanță ale unei aplicații construite după o abordare de tip monolit, este nevoie de scalarea întregii aplicații. Principala consecință a acestui fapt este utilizarea nejustificată de resurse. Un alt dezavantaj îl reprezintă faptul că o astfel de aplicație poate deveni complexă, și greu de întreținut. Orice modificare adusă unui modul din aplicație, poate afecta întreg sistemul, astfel devine necesară testarea aplicației în totalitate, dar și livrarea unei noi versiuni. O arhitectură de tip microservicii, poate fi o soluție pentru acest tip de aplicații, deoarece principiile pe care se bazează încearcă să rezolve dezavantajele enumerate anterior. Ideea principală este că funcționalitatea unei aplicații este împărțită în module mici, independente, care pot fi modificate și înlocuite cu ușurință. Astfel, modificările aduse unui modul, nu afectează celelalte module, testarea devenind mai ușoară. Principalul avantaj constă în faptul că într-o aplicație construită în această abordare, de tip microservicii, este posibilă scalarea unei singure componente, care poate deveni supraîncărcată la un moment dat. Totodată, utilizând acest tip de arhitectură, este ușoară și găzduirea în *Cloud*, aceasta fiind o tendință actuală. Toate microserviciile, de autentificare, autorizare, cele pentru funcționalitatea aplicației, gateway-ul, baza de date și componenta pentru cozi de mesaje, RabbitMq, sunt izolate în containere Docker. Pentru rularea aplicației s-a utilizat modulul Docker Compose. În final se dorește obținerea unei aplicații cu performanțe bune și în cazul componentelor suprautilizate, o aplicație ușor extensibilă și care să permită reutilizarea componentelor.