



FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII
ȘI TEHNOLOGIA INFORMAȚIEI

Sesiunea de comunicări științifice
2016

Arhitecturi pentru sisteme auto-scalabile: microservicii

Andrei Mihăescu

Conducători științifici:

Conf. dr. ing. Eduard-Cristian POPOVICI

Ș.l. dr. ing. Radu BADEA

Capitolul 1

Motivație

Datorită avansurilor tehnologice uriase în domeniul informaticii și al telecomunicațiilor astăzi, mai mult ca niciodată, o foarte mare parte a activităților coditene (fie ele profesionale sau personale) sunt stâns legate de domeniul virtual. Revoluția digitală a schimbat complet modul în care folosim sistemele informatice, care acum stau la baza foarte multor afaceri din toate domeniile (plecând de la cel bancar și terminând cu cel agricol) în proporție de 24% cu previziuni ca această cifră să se dubleze până în 2020 - *"Forrester Research - The State of Digital Business, 2015 to 2020"*.

Această expansiune a pus o presiune enormă asupra inginerilor din domeniul informatic, care se confruntă cu o adevărată provocare pentru a crea sisteme robuste capabile să satisfacă această nevoie crescândă. Pornind de la această nevoie, conceptul de aplicație de tip "enterprise" a luat naștere. O astfel de aplicație trebuie să răspundă anumitor standarde și cerințe pentru a putea asigura buna funcționare a unei afaceri:

- **flexibilitate** - o aplicație bine gândită trebuie să permită modificări, ce apar ca urmare a evoluției domeniului în care este folosită; acestea trebuie ușor implementate, pentru a crea valoare pentru utilizatori.
- **robustețe** - în proiectarea unei aplicații trebuie luate în calcul cât mai multe cazuri în care aceasta ar putea fi exploatată și compromisă, creând astfel întreruperi în furnizarea serviciilor.
- **scalabilitate** - având în vedere dorința fiecărei afaceri de a prospera și a crește este de așteptat ca și aplicațiile folosite să se poată adapta la volume din ce în ce mai mari de procesări.
- **securitate** - dat fiind faptul că o aplicație poate deveni principala sursă de venit din cadrul unei companii este nevoie ca aceasta să respecte cel mai înalte standarde de securitate astfel încât datele utilizatorilor să fie bine protejate.

Pentru a răspunde tuturor acestor nevoii, un sistem informatic trebuie să fie slab cuplat (eng. *loosely-coupled*), pentru a asigura flexibilitate, să aibă mecanisme ce asigură redundanța, pentru robustețe, să aibă un grad mare de modularizare pentru a fi scalabil și să implementeze principiul eng. AAA - *Authentication, Authorization & Accounting* pentru a fi sigur.

Capitolul 2

Aplicații monolit

Aplicațiile **monolit** reprezintă modul tradițional de dezvoltarea a unei aplicații, fie ea chiar și de tip enteprise. În cadrul aplicațiilor monolit toată funcționalitatea este dezvoltată în cadrul unui artefact care este livrat la final și trimis în producție. Acest model prezintă o serie de avantaje și dezavantaje.

Printre primele avantaje ale acest tip de aplicație se numără faptul că este ușor de dezvoltat; din punct de vedere arhitectural nu implică eforturi mari. Totul este dezvoltat ca un singur bloc a cărui componente pot fi organizate pe module, însă sunt reunite și livrate sub forma unui singur produs final. În mod tipic o astfel de aplicație va rula pe unul sau mai multe servere. Fiecare dintre acestea fiind identic toate componentele din cadrul aplicație vor comunica între ele la nivel de proces, fără a implica vreun fel de comunicare cu exteriorul, exceptând prin intermediul punctelor terminale (*eng. end point*) special construite. Acest tip de aplicație prezintă un grad relativ scăzut de complexitate, aceasta din punct de vedere arhitectural, deoarece complexitatea logici de business nu este corelată cu arhitectura aplicației. Mentenanța unei astfel de aplicații este facilă, întreg codul situându-se într-un singur loc, procesul de depanare se poate desfășura ușor.

În ciuda acestor avantaje care ar putea înclina balanța în favoarea aplicațiilor monolit,ele prezintă o serie de dezavantaje care îngreunează procesul de creare de valoare prin implementarea nevoilor unei afaceri. Faptul că toate componentele sunt în cadrul unui singur artefact determină un nivel înalt de cuplare între acestea - *eng. tightly coupled* - ceea ce duce la o scădere a flexibilității și deci o reducere a ritmului de implementare a schimbărilor. Legat de acest fapt este randamentul slab în ceea ce privește scalabilitatea. Aplicația nu poate scala decât cu o granularitate de o instanță, nu de un modul al său de care poate fi nevoie. Astfel spus, pentru a îmbunătății performanța unui modul toată aplicație trebuie instalată pentru un alt server. Un dezavantaj, ar fi legat de modul de instalare a aplicațiilor pe diverse medii (dezvoltare, testare sau producție). Adesea o nouă instalare (*eng. deploy*) vizează o singură componentă, însă pentru a livra schimbarea toată aplicația trebuie recompilată și instalată.

Deși avantajele acestor aplicații nu sunt de neglijat, ele prezintă dezavantaje care le fac să nu mai fie relevante în contextul actual al tendințelor, care orientate pe flexibilitate și pe livrarea în producție cât mai rapidă a produselor informatice.

Capitolul 3

Aplicații bazate pe microservicii

Aplicațiile bazate pe microservicii fac parte dintr-o categorie mai largă de aplicații care sunt înscrise în capitolul aplicațiilor orientate pe servicii (*eng. SOA - Service Oriented Architecture*). Caracteristicile arhitecturii SOA sunt:

- Serviciile sunt autonome. Fiecare serviciu este menținut, dezvoltat, instalat și versionat în mod independent.
- Serviciile pot fi distribuite. Serviciile pot fi localizate oriunde pe rețea, local sau la distanță, atât timp cât rețeaua dispune de protocoalele necesare de comunicare.
- Serviciile sunt slab cuplate. Fiecare serviciu este independent față de celălalte și poate fi înlocuit și actualizat fără a întrerupe celălalte aplicații care îl folosesc atât timp cât interfețele sunt compatibile.
- Serviciile împart schema și contractul, nu clasa. Serviciile partajează contracte și scheme atunci când comunică, și nu clase interne.

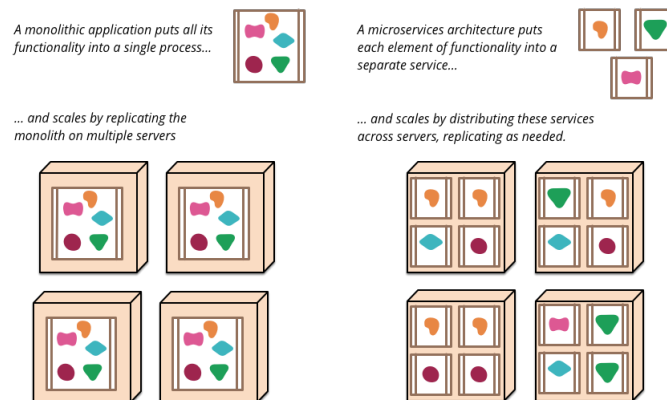


Figura 3.1: Aplicații monolit și aplicații bazate pe microservicii

Plecând de la aceste caracteristici ideea microserviciilor duce mai departe distribuirea și slaba cuplare a serviciilor, astfel că dezvoltarea unei aplicații este realizată ca o suită de mici servicii, fiecare rulând un proces independent și comunicând prin mecanisme simple, adesea resurse expuse prin API-uri via HTTP. Pe lângă faptul că serviciile pot fi instalate și scalate în mod independent, fiecare crează o limită fermă a modulului, permițând chiar și folosirea limbajelor de programare diferite, acestea putând fi gestionate de echipe diferite.