

Tema 2 PSSC

Microservices

Descriere:

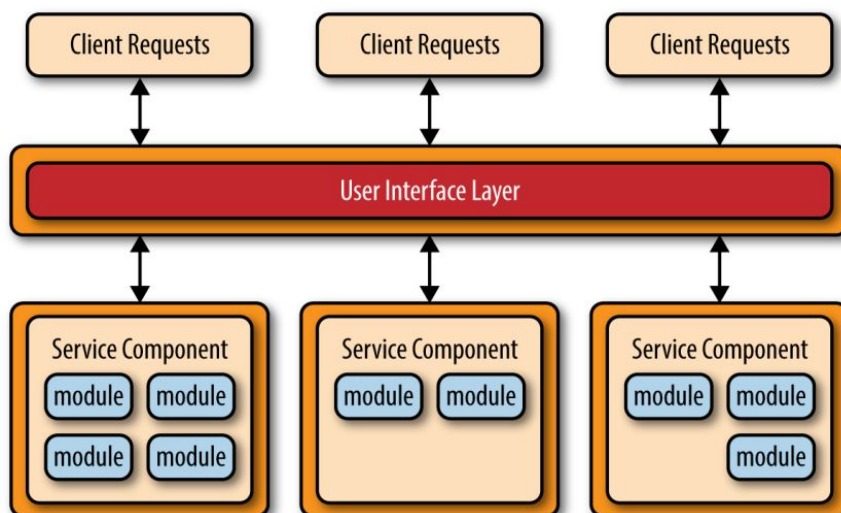
Microserviciile , cunoscute ca si Arhitectura Microservicii este un stil arhitectural in care aplicatia este structurata dintr o colectie de servicii cuplate in mod liber .Arhitectura lor permite implementarea unor aplicatii complexe mult mai usor. Aceasta tehnica paralelizeaza dezvoltarea , permitand echipelor mici autonome sa se dezvolte si sa implementeze independent servicii .
Pana in momentul de fata ,nu s-a realizat un consens care sa defineasca proprietatile microserviciilor si lipseste o definitie oficiala.

Implementare :

De obicei serviciile intr o arhitectura de microservicii sunt procese care comunica pentru printr o retea pentru as indeplini scop folosind protocoale cum ar fii HTTP.In retea OSGI ele functioneaza in cadrul aceluiasi proces.

Pentru a folosi aceasta tehnica se aplica o structura modulara , astfel incat modificarea unei parti din aplicatie va necesita reconstruirea unui numar mic de servicii.

Dupa cum se poate observa in schema de mai jos , inlocuirea unui modul nu le va influenta si pe celelalte.



Sursa: Richards, M. (2015). Software Architecture Patterns - Understanding Common Architecture Patterns and When to Use Them.

O arhitectura bazata pe microservicii respecta principiile “interfetele cu granulatiefina”,dezvoltarea bazata pe afaceri , arhitecturile aplicatiilor cloud ,programarea si persistenta programelor poliglot.

Deși există multe de modalități de implementare a unui model de arhitectură orientată pe microservicii, se remarcă trei topologii principale:

- topologia API-REST (Application Programming Interface – REpresentational State Transfer),
- topologia REST (REpresentational State Transfer),
- topologia mesageriei centralizate.

Istoric:

Prima data termenul de microservicii a aparut la Venetia in 2011 folosit de un grup de arhiecti pentru a descrie tehnicile observate de ei.James Lewis a prezentat un studiu de caz in 2012 in care a prezentat cateva idei despre acest stil arhitectural.

Avantaje:

1. Este mai usor de inteles deoarece serviciile si relationarea dintre ele pot fii analizate pe rand ;
2. Este usor de testat , se pot efectua teste pentru verificarea fiecarui serviciu dupa care se poate testa integrarea lor ;
3. Sunt usor de dezvoltat , permit mai multor persoane sa dezvolte servicii diferite dupa care ele vor fii integrate pentru a functiona impreuna , fiind usor de adaugat/inlocuit/eliminat un anumit modul;
4. Permite utilizarea unor limbaje diferite , specifice fiecarui tip de serviciu ;

Dezavantaje:

In ciuda acestor multe avantaje , aceasta tehnica are si unele dezavantaje , principalul ar fii limbajul utilizat in aplicatia, e posibil ca acesta sa nu permita utilizarea anumitor servicii si procese necesare implementarii acestui stil .

Concluzie:

Aceasta tehnica de programare este solutia optima pentru dezvoltarea unor aplicatii mai mari , care vor suferi multe modificari in timp , in special in domeniul Web.