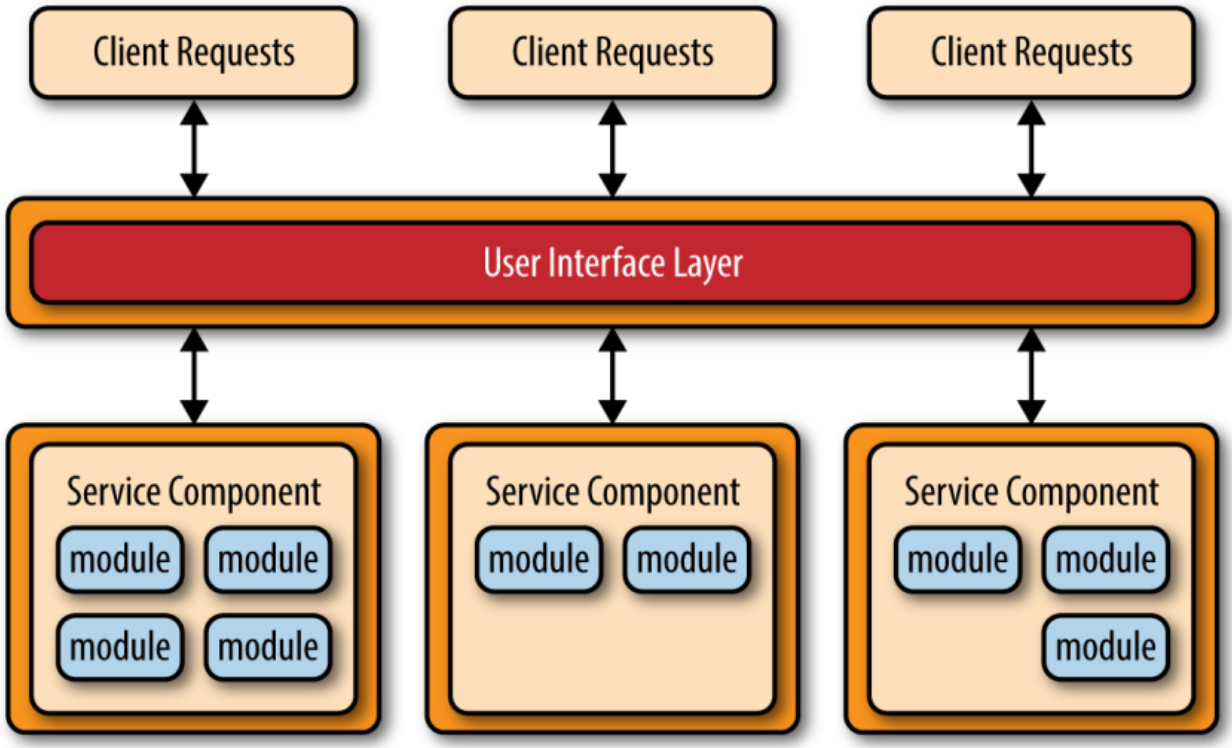
Tema 2

Microservices se referă, de fapt, la un tip de arhitectură software în care o aplicație este construită din bucăți mici, independente (microservicii).

Printre beneficiile abordării stilului arhitectural bazat pe microservicii putem enumera dimensiunile relativ mici ale microserviciului - această dimensiune face procesul de înțelegere a developerului să decurgă mai rapid, crește productivitatea teoretică a developerilor, mediile de dezvoltare nefiind supraîncărcate, și containerele web pornesc mai repede, mărind, din nou, productivitatea developerilor dar și mărind viteza de livrare și procesul din spate.

Fiecare serviciu poate fi livrat independent de celelalte servicii, fiind mult mai ușor să se livreze noi versiuni ale serviciilor, în mod frecvent. Este mai ușor să se scaleze procesul de development - se poate organiza efortul de dezvoltare în mai multe echipe, fiecare echipă, în funcție de dimensiune și capabilități fiind responsabilă de un singur serviciu sau un grup. Fiecare echipă își poate dezvolta, scala și livra serviciul sau grupul de servicii la care lucrează.

Un alt avantaj în constituie izolarea mai bună a bug-urilor/defectelor. De exemplu, dacă există un memory leak într-un serviciu, acesta este conținut doar în acel serviciu, adică doar acel serviciu va fi afectat.



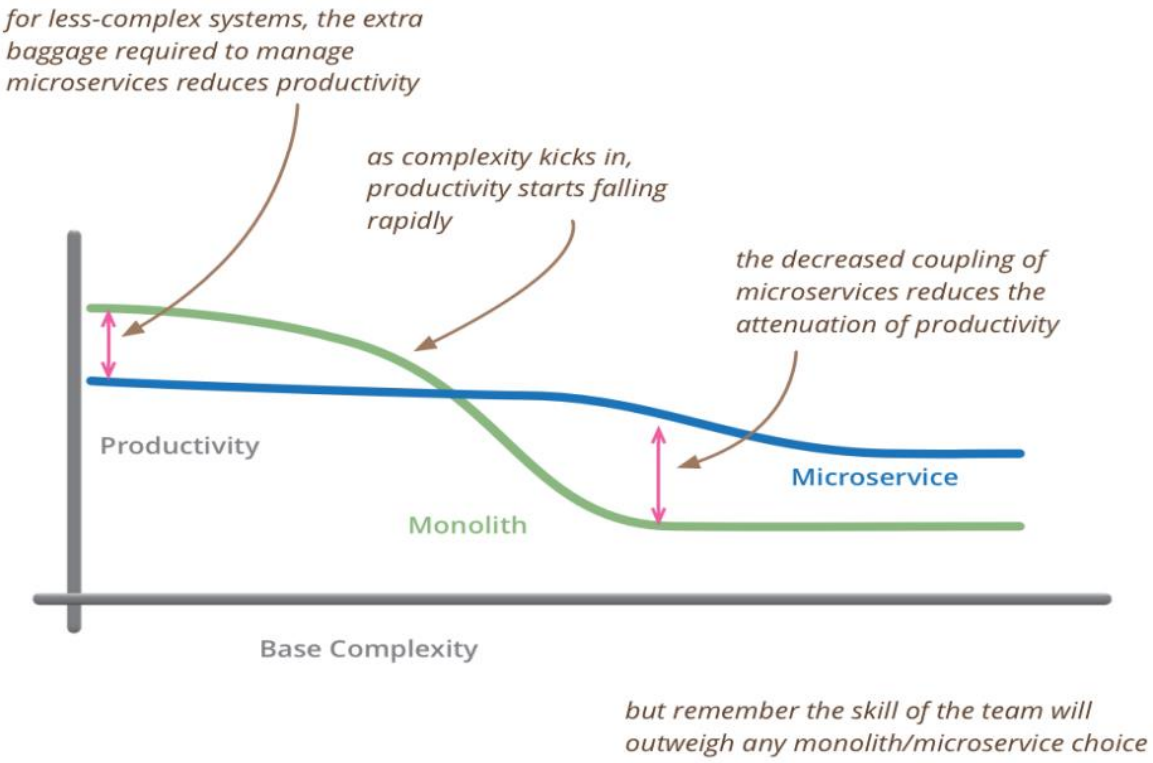
Sursa: Richards, M. (2015). Software Architecture Patterns - Understanding Common Architecture Patterns and When to Use Them.

Desigur, microserviciile nu sunt soluția perfectă, și acestea vin cu dezavantaje.

În primul rând, developerii trebuie să țină cont de complexitatea adițională asociată creării unui sistem distribuit.

Mai mult, developerilor le revine sarcina de a crea un sistem de comunicare inter-servicii. Implementarea use-case-urile ce implică folosirea a mai multe servicii fără utilizarea tranzacțiilor distribuite este foarte dificilă, iar coordonarea intre echipe în aceste situații trebuie făcută cu grijă.

Nu în ultimul rând, trebuie să fie menționat consumul mare de memorie. Arhitectura bazată pe microservicii înlocuiește N monoliți cu N * M servicii. Dacă fiecare serviciu își rulează propriul JVM (sau ceva echivalent), ce în mod obișnuit este necesar pentru izolarea instanțelor, atunci este un exces de M ori numărul de mașini virtuale. Mai mult, dacă fiecare serviciu își rulează propria mașină virtuală (instanță EC2), ca și în cazul Netflix, atunci excesul este mult mai mare



Microserviciile sunt folosite de marile companii de aplicatii web, cum este Netflix, Amazon, eBay care au evoluat de o arhitectură monolitică la o arhitectură de microservicii.

Netflix, care este un serviciu de streaming video foarte popular, care este responsabil pentru până la 30% din traficul pe Internet, are o arhitectură pe scară largă, orientată spre servicii. Pe langă serviciile de streaming video, Netflix oferă și servicii pentru dezvoltarea unei aplicații SOA, cum ar fi: Hystrix(serviciu pentru circuit-breaker), Eureka(Spring Cloud Service Discovery), Ribbon(Inter Process Communication), Zuul(Proxy and Reverse Proxy Service).