



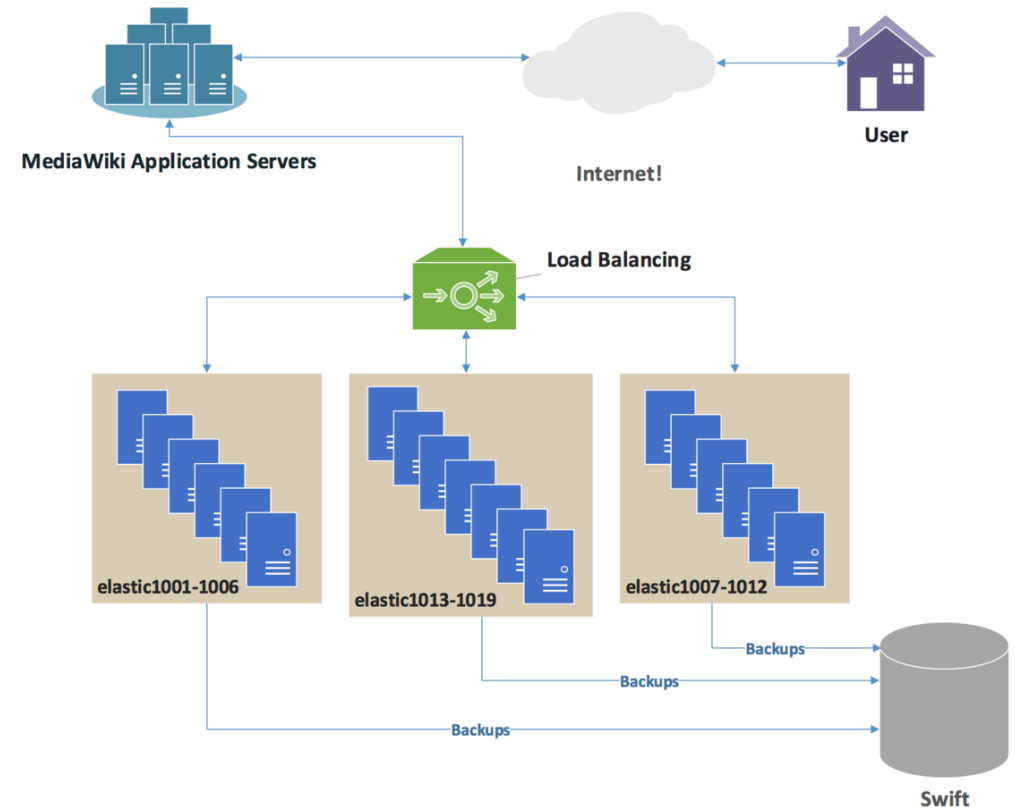
# SPRING CLOUD LOADBALANCER

Student: Miroslav Tomić  
Asistent: Marko Vještica  
Profesor: Vladimir Ivančević

# ŠTA JE LOAD BALANCING?

Load balancing se odnosi na proces raspodjele skupa zadataka preko skupa resursa sa ciljem da njihova obrada bude efikasnija.

Postoje različite tehnike balansiranja koje mogu optimizovati vrijeme odziva za svaki zadatak, ovim se izbjegava neravnomjerno opterećenje čvorova.



# ALGORITMI LOAD BALANCING- A

- Round Robin
- Weighted Round Robin
- Least Connection
- Weighted Least Connection
- Resource Based (Adaptive)
- Weighted Response Time
- URL Hash
- ...

# SPRING CLOUD LOADBALANCER

- Spring Cloud LoadBalancer je komponenta open source Spring Cloud biblioteke.
- Ova komponenta je trenutno u fazi razvoja i još uvijek nema mnogo članaka na internetu u cilju prikazivanja rada, odnosno nema mnogo korisničkih implementacija.
- Netflix Ribbon klijent (prethodnik spring cloud loadbalancera) je stable verzija LoadBalancer- a koja se još uvijek koristi u gotovo svim Spring Boot projektima međutim ova komponenta se više ne razvija.

# SPRING CLOUD LOADBALANCER

- Spring Cloud LoadBalancer trenutno ne nudi posebne (već implementirane) funkcionalnosti vezane za load balancing, ali će u narednim verzijama biti proširen raznim funkcionalnostima.
- Sav nedostatak posebnih funkcionalnosti može biti nadoknađen ručnom implementacijom.

# SPRING CLOUD LOADBALANCER

- Obezbjeđuje svoj client-side load-balancer apstrakciju i implementaciju.
- Za load-balancing mehanizam *ReactiveLoadBalancer* interfejs je dodat i Round-Robin kao podrazumjevana implementacija za njega.
- Da bi se dobile instance aplikacije kojima se dodjeljuju zadaci neophodno je koristiti *ServiceInstanceListSupplier*.
- Trenutno postoji podrška za service-discovery-based implementaciju *ServiceInstanceListSupplier* koja preuzima dostupne instance iz Service Discovery- a koristeći Discovery Client koji je dostupan u classpath- u.

# SPRING CLOUD LOADBALANCER - INTEGRATIONS

- U cilju lakšeg korišćenja Spring Cloud LoadBalancer-a obezbeđen je *ReactorLoadBalancerExchangeFilterFunction* koji može da se koristi sa *WebClient* i *BlockingLoadBalancerClient* koji radi sa *RestTemplate*.

# SPRING CLOUD LOADBALANCER - CACHING

- Pored klasične *ServiceInstanceListSupplier* implementacije koja preuzima instance preko *DiscoveryClient* svaki put kada bira instance, obezbjeđene su dvije caching implementacije:

Caffeine-backed LoadBalancer Cache Implementation

Default LoadBalancer Cache Implementation



# SPRING CLOUD LOADBALANCER — ZONE-BASED

- Za zone-based load-balancing se koriste posebni filteri kao što je *ZonePreferenceServiceInstanceListSupplier* koji obezbjeđuju preuzimanje instance iz Discovery Client ali samo pod uslovom da pripadaju traženoj zoni.
- Korisniku je prepuštena custom implementacija zone-based load-balancinga.

# SPRING CLOUD LOADBALANCER — HEALTH-CHECK

- Moguće je ugraditi I health-check unutar spring cloud load-balancera koji će provjeravati da li su "žive" instance specificirane putem *ServiceInstanceListSupplier*.
- Za ove potrebe obezbjeđen je *HealthCheckServiceInstanceListSupplier*.

# SPRING CLOUD LOADBALANCER — STARTER

- Za jednostavno pokretanje Spring Cloud LoadBalancer- a putem Spring Boot aplikacije koristi se dependency:  
`org.springframework.cloud:spring-cloud-starter-loadbalancer`

# SPRING CLOUD LOADBALANCER — PASSING CONFIGURATION

- Pomoću anotacije `@LoadBalancerClient` dajemo ime load-balancer klijenta i klasu u kojoj pišemo našu konfiguraciju.
- PRIMJER: <https://github.com/tmiroslav97/SMPUOS-spring-cloud-loadbalancer>

# IZVORI

- <https://cloud.spring.io/spring-cloud-commons/reference/html/#spring-cloud-loadbalancer>
- <https://spring.io/guides/gs/spring-cloud-loadbalancer/>
- [https://en.wikipedia.org/wiki/Load\\_balancing\\_\(computing\)](https://en.wikipedia.org/wiki/Load_balancing_(computing))