

WEB APLIKACIJE I KLASTERI



POJAM KLASTERA

2

◆ ŠTA SU KLASTERI?

- Grupa međusobno povezanih računara koji funkcionišu tako da se mogu posmatrati kao jedan sistem koji pruža neki servis

◆ UPOTREBA KLASTERA

- Sredstvo za unapređenje performansi
- Sredstvo za unapređenje pouzdanosti
- Jeftinije rešenje u odnosu na jedan računar ekvivalentnih mogućnosti



POJAM KLASTERA

3

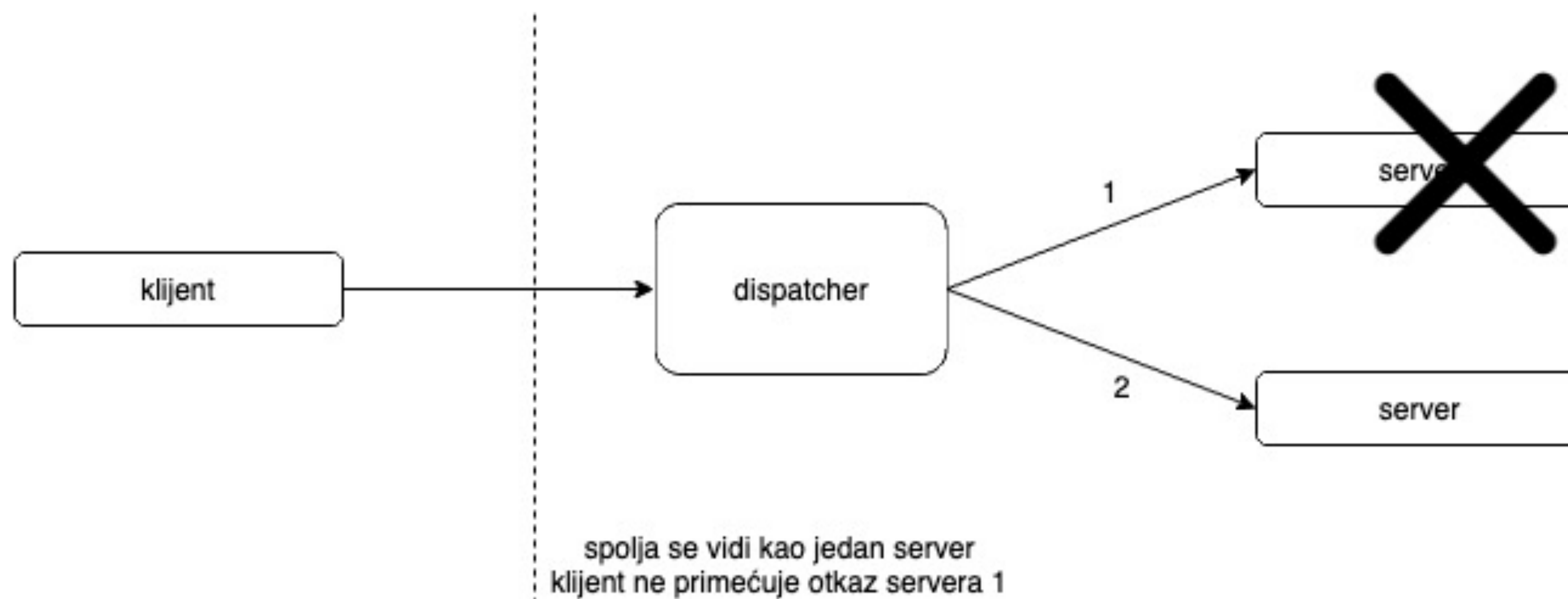


POVEZIVANJE ČVOROVA U KLASTERU

- Najčešće u lokalnoj mreži
- Može i distribuirano
- Dodavanjem novih čvorova u kluster povećavaju se i dostupnost i skalabilnost

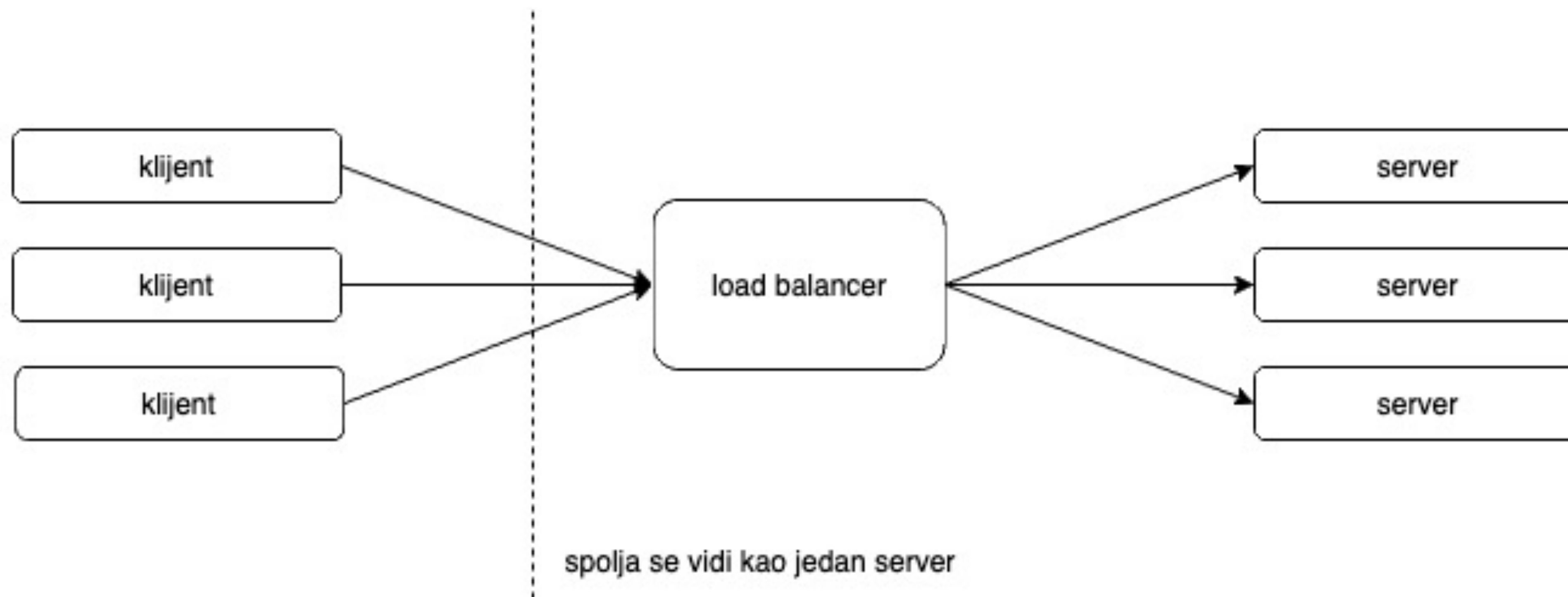
◆ KLASITER VISOKE DOSTUPNOSTI (HIGH-AVAILABILITY, FAILOVER)

- Redudantni hardver, veća pouzdanost
- Minimum dva računara
- Eliminiše SPoF (single point of failure)



◆ LOAD BALANCING

- Raspodela opterećenja na više čvorova
- Različiti algoritmi raspodele





VRSTE KLASTERA

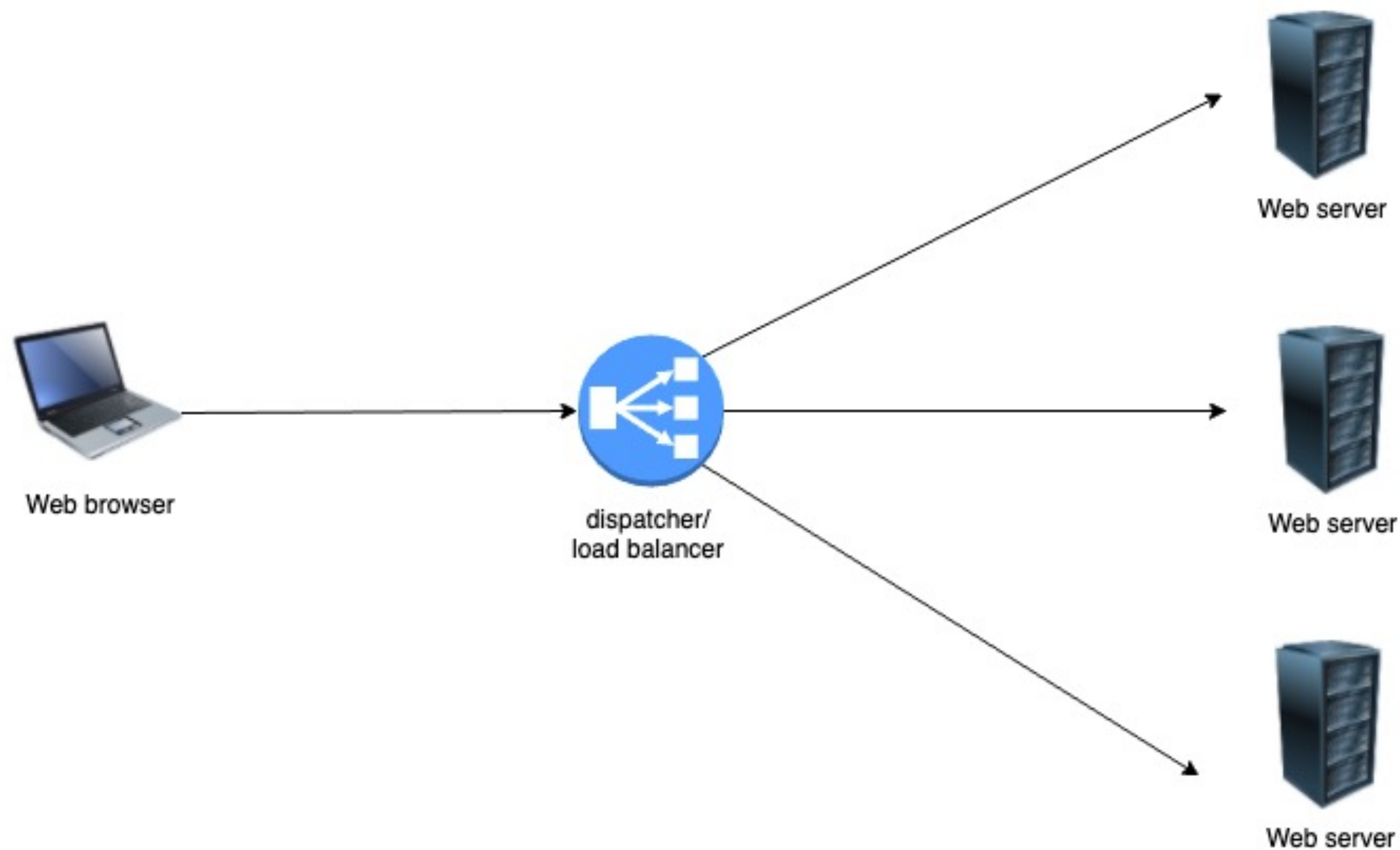
6



KLASTERI VISOKIH PERFORMANSI

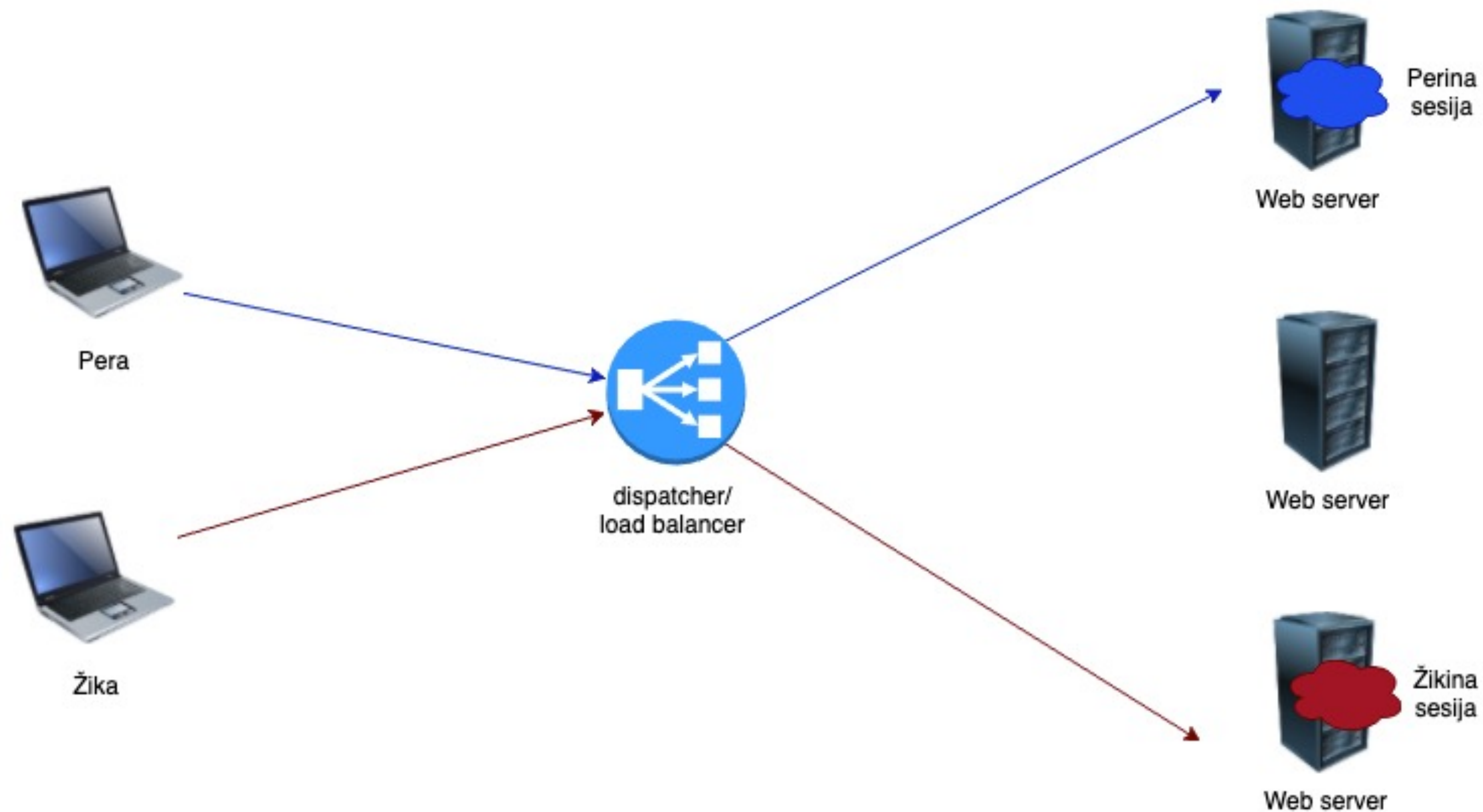
- Namenjeni za specifične poslove masovne paralelne obrade podataka
- Različiti načini za sprežanje čvorova:
 - Tightly coupled
 - Loosely coupled
 - Grid computing

◆ SERVERSKI RAČUNARI NA KOJIMA RADI WEB SERVER (NPR. TOMCAT)



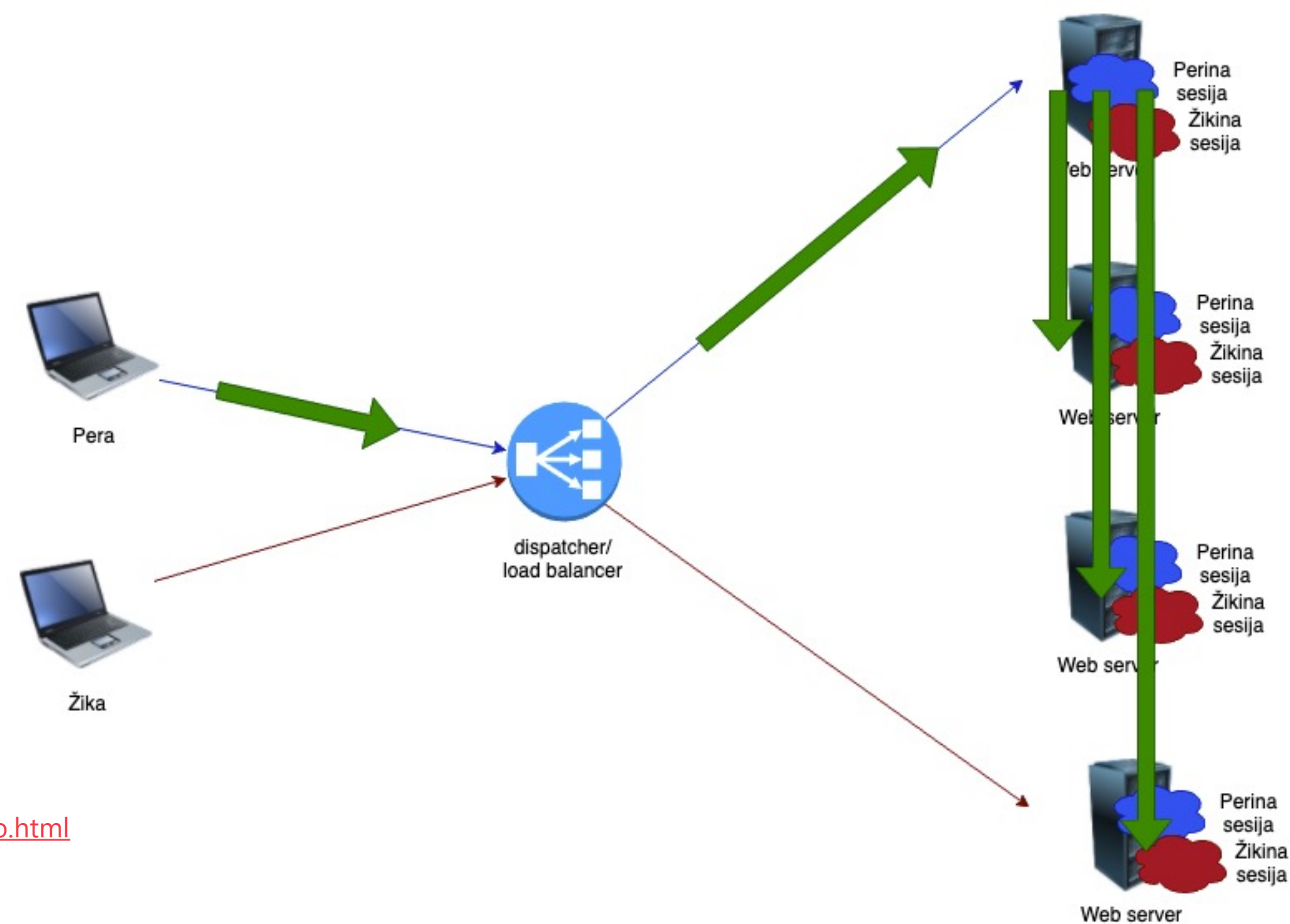
◆ NEMA REPLIKACIJE

- *Sticky sessions* režim rada
- Zahtev jednog klijenta uvek se upućuje na isti server u klasteru
- Jednostavno, ali nema failover



◆ SVE SESIJE NA SVIM SERVERIMA (TOMCAT¹)

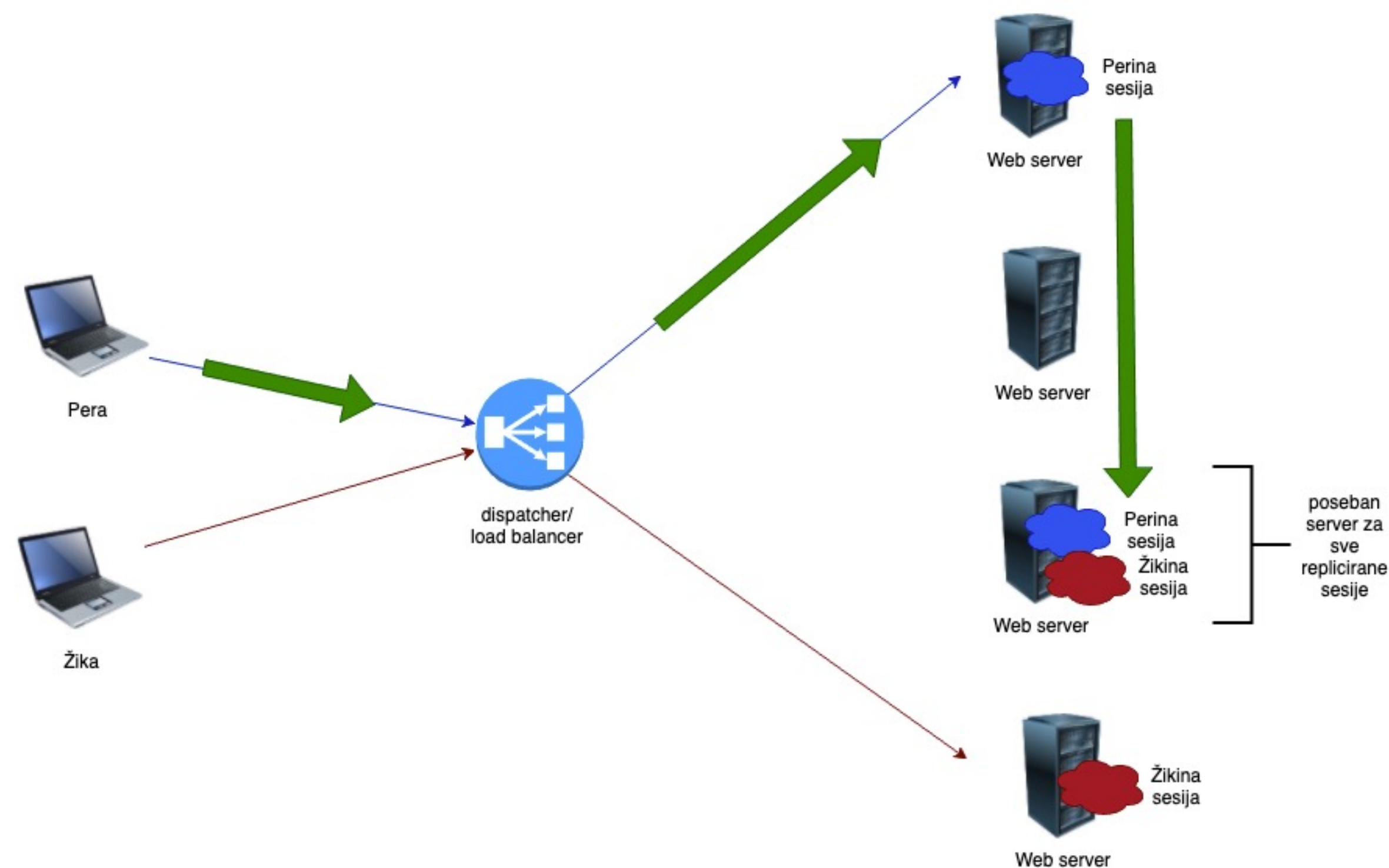
- Ima balansiranje, ima failover
- Replikacija sesija - veliki saobraćaj, nije za velike klastere ili velike sesije



¹ Poison message <https://tomcat.apache.org/tomcat-8.0-doc/cluster-howto.html>

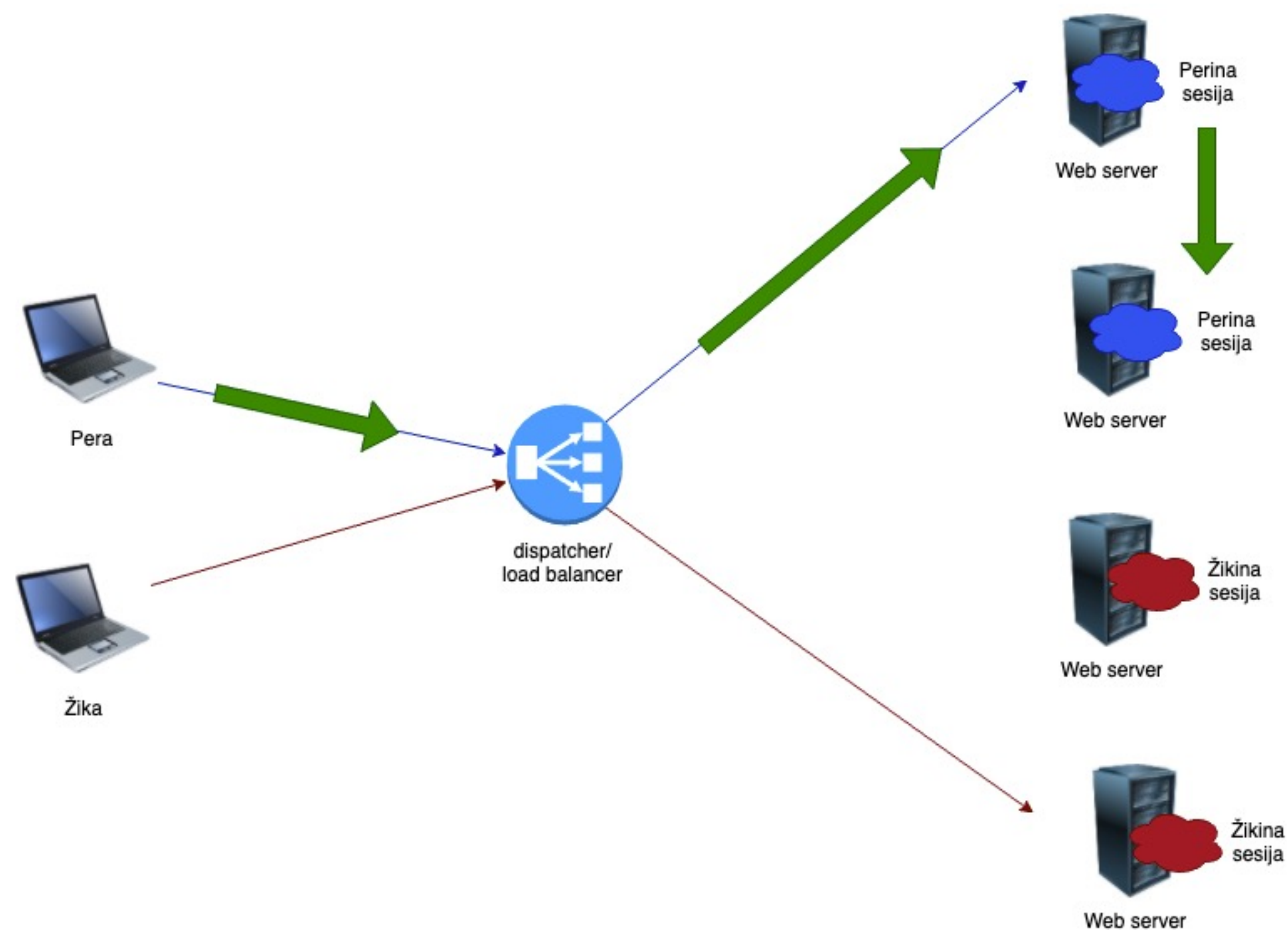
◆ SESIJA SE REPLICIRA SAMO NA POSEBAN ČVOR (TERRACOTA, IBM)

- Sesija je slabo vezana za čvor
- Load balancer radi sticky sessions dok je sve u redu
- SPoF?



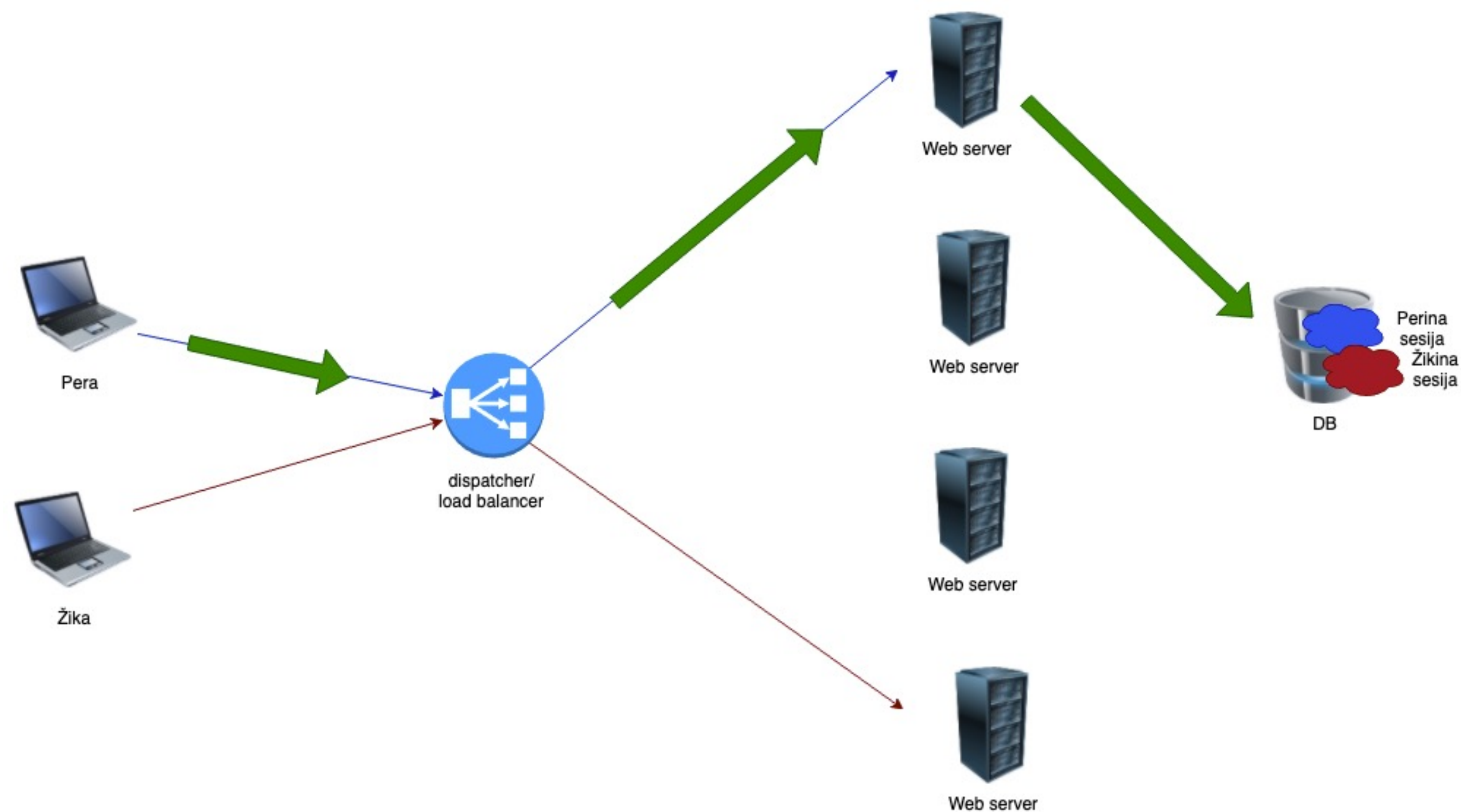
◆ SESIJA SE REPLICIRA NA JOŠ JEDAN SERVER (JBOSS, WEBLOGIC)

- Svaka sesija je na dva servera (primarni i backup)
- Dodavanje novih servera ne povećava saobraćaj



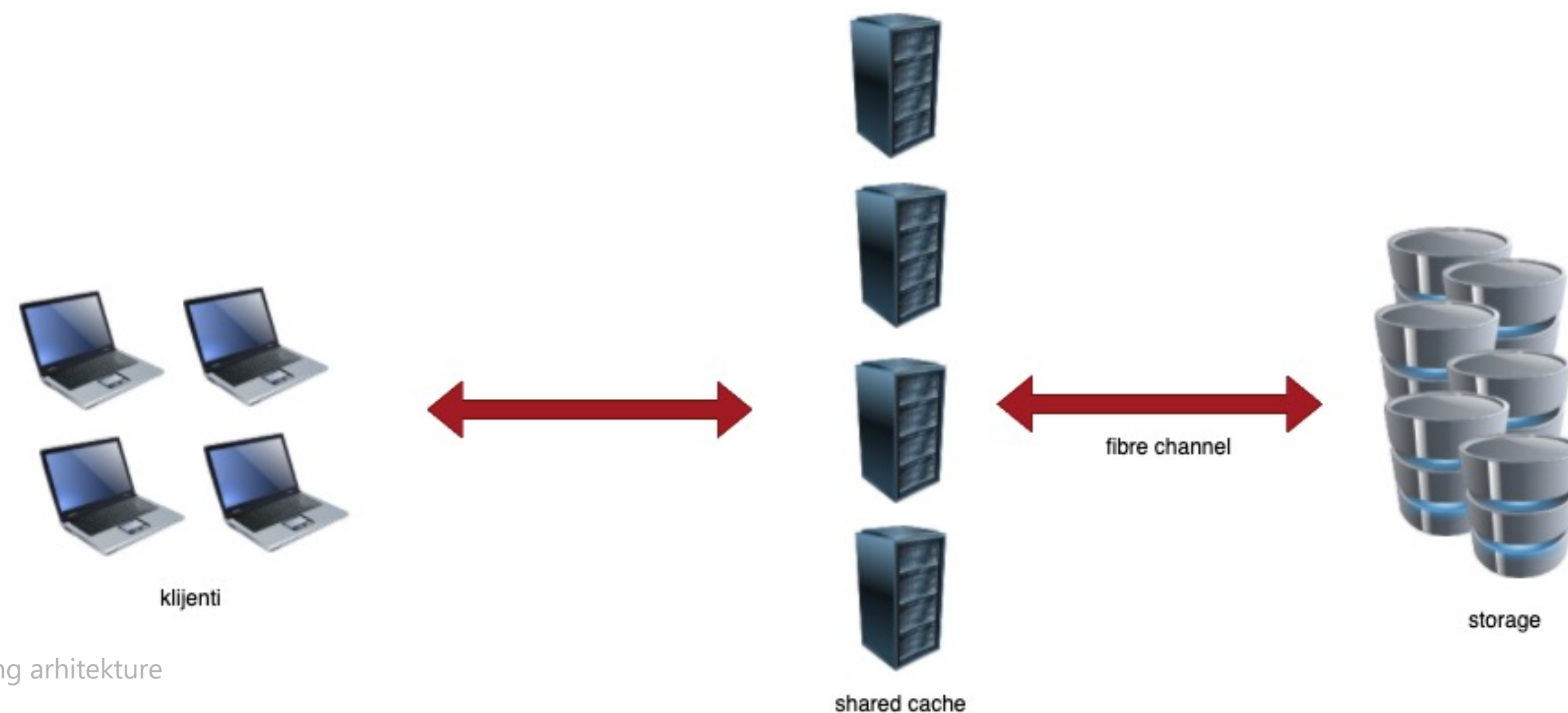
◆ SESIJA SE ČUVA U BAZI PODATAKA (SUN)

- Web server su *stateless*
- Potencijalno veliki saobraćaj prema bazi podataka



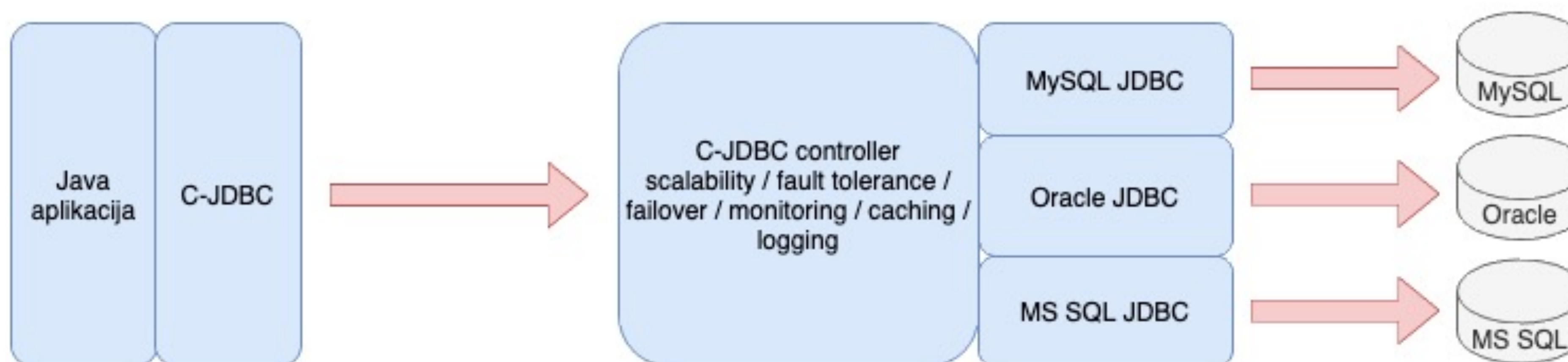
◆ SPECIFIČAN ZA KONKRETAN SUBP

- *Shared-nothing* arhitektura
 - Podaci na disku (ili u memoriji) se ne dele između nodova u klasteru
 - Svaki zahtev obrađuje jedan čvor (CPU/memorija/disk)
- *Shared-everything (shared-disk)* arhitektura
 - Podaci na disku (ili u memoriji) se dele između čvorova u klasteru



◆ KLASER POMOĆU JDBC DRAJVERA – C-JDBC

- *Cross-database* – može povezivati različite SUBP u jedan klaster





REFERENCE

15

- ◆ SLAJDOVI PO UZORU NA <https://github.com/mbranko/isa19/blob/master/09-arch/clustering.pdf>
- ◆ GOOGLE CLOUD. ARCHITECTURES FOR HIGH AVAILABILITY OF MYSQL CLUSTERS ON COMPUTE ENGINE
<https://cloud.google.com/architecture/architectures-high-availability-mysql-clusters-compute-engine>
- ◆ C-JDBC USER'S GUIDE. <https://c-jdbc.ow2.org/current/doc/userGuide/html/index.html>
- ◆ STONEBRAKER M. THE CASE FOR SHARED NOTHING
https://static.aminer.org/pdf/PDF/000/255/770/the_case_for_shared_nothing.pdf
- ◆ DEWITT D. AT AL. HOW TO BUILD A HIGH-PERFORMANCE DATA WAREHOUSE
http://db.csail.mit.edu/madden/high_perf.pdf
- ◆ SCALEDDB. SHARED-DISK VS. SHARED-NOTHING - COMPARING ARCHITECTURES FOR CLUSTERED DATABASES
https://web.archive.org/web/20150323110547/http://www.scaledb.com/pdfs/WP_SDvSN.pdf

**KOJA SU VAŠA
PITANJA?**