

*Baze podataka*

*Katedra za računarstvo  
Elektronski fakultet u Nišu*

# Arhitekture sistema baza podataka

Prof.dr Leonid Stoimenov

# Pregled

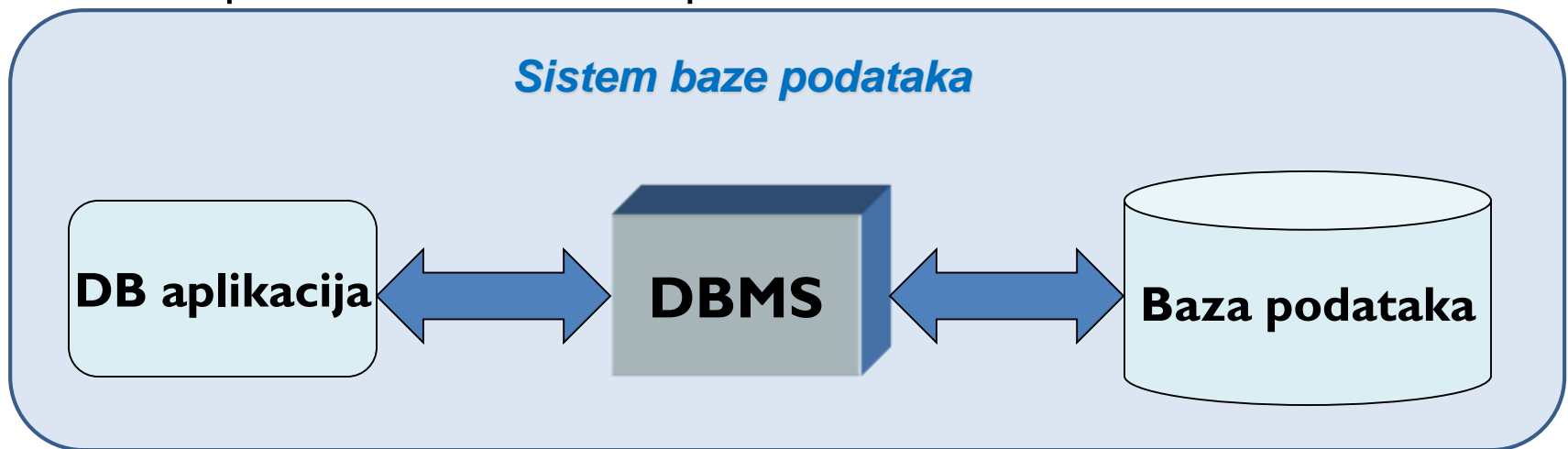
---

- ▶ Podsetnik
- ▶ Arhitekture višekorisničkih DB sistema



# Podsetnik

- ▶ **Baza podataka** predstavlja kolekciju **povezanih** podataka organizovanih u logičke celine predstavljene **tabelama**.
- ▶ **Sistem za upravljanje bazama podataka** (DBMS) - Softverski sistem koji omogućava definisanje, kreiranje i manipulisanje bazom podataka
- ▶ **Aplikacija baze podataka** - Program koji interaguje sa bazom podataka u toku svog izvršenja
- ▶ **Sistem baze podataka** - Kolekcija aplikacionih programa koji interaguju sa bazom podataka, DBMS i baza podataka



# Struktura aplikacije nad bazom podataka

---

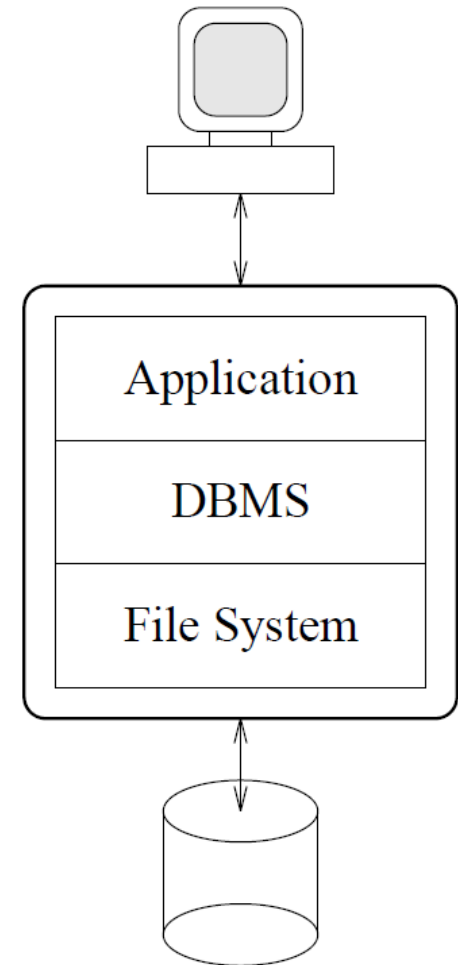
- ▶ Komponente aplikacije nad BP su
    - ▶ Baza podataka – čuva podatke
    - ▶ Transakciona logika – obrada podataka (kod BP), kontrolisana od strane DBMSa, na nivou read-write operacija
    - ▶ Poslovna logika i logika obrade podataka – obrada podataka (kod aplikacija) koja je definisana poslovnim procesima
    - ▶ Korisnički interfejs – przentacija podataka korisniku (forme, izveštaji, validacija i kontrola unosa podataka, poruke)
- 



# Monolitni sistemi

---

- ▶ Aplikacije
  - ▶ Interakcija sa korisnikom
  - ▶ Aplikaciono-specifični zadaci
- ▶ DBMS
  - ▶ Optimizacija upita
  - ▶ Obrada upita
  - ▶ Upravljanje Transakcijama
  - ▶ Sigurnost i upravljanje integritetom
- ▶ Fajl sistem
  - ▶ Čuvanje podataka



# Arhitekture višekorisničkih sistema baze podataka

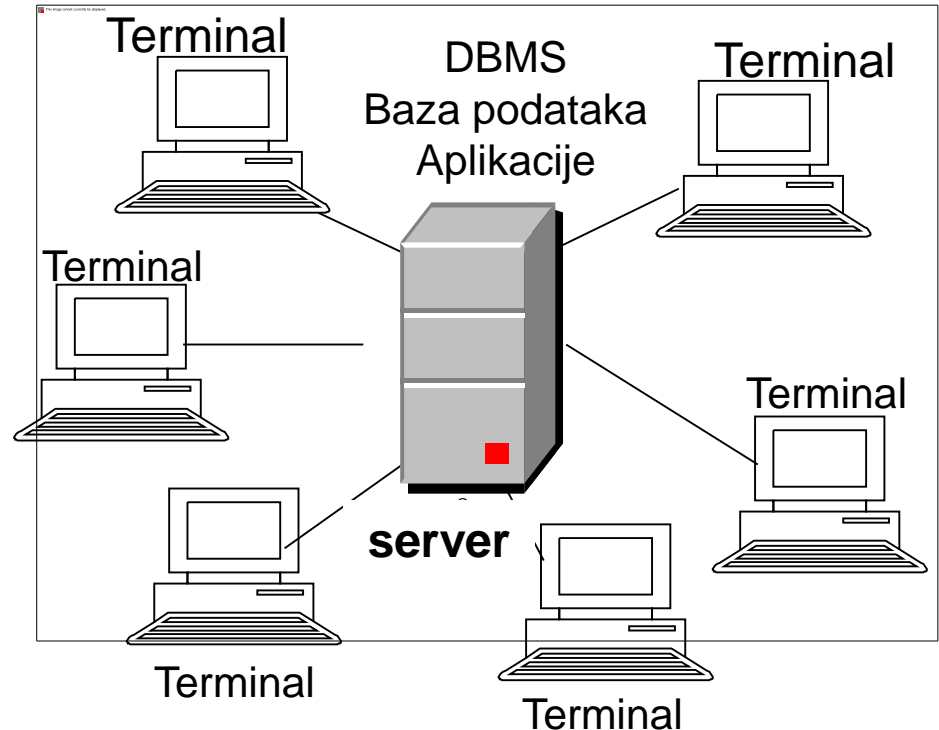
---

- ▶ Teleprocesing
- ▶ Fajl-server
- ▶ Klijent-server
  - ▶ u 2 nivoa (two-tier) - tradicionalni
  - ▶ u 3 nivoa (three-tier)

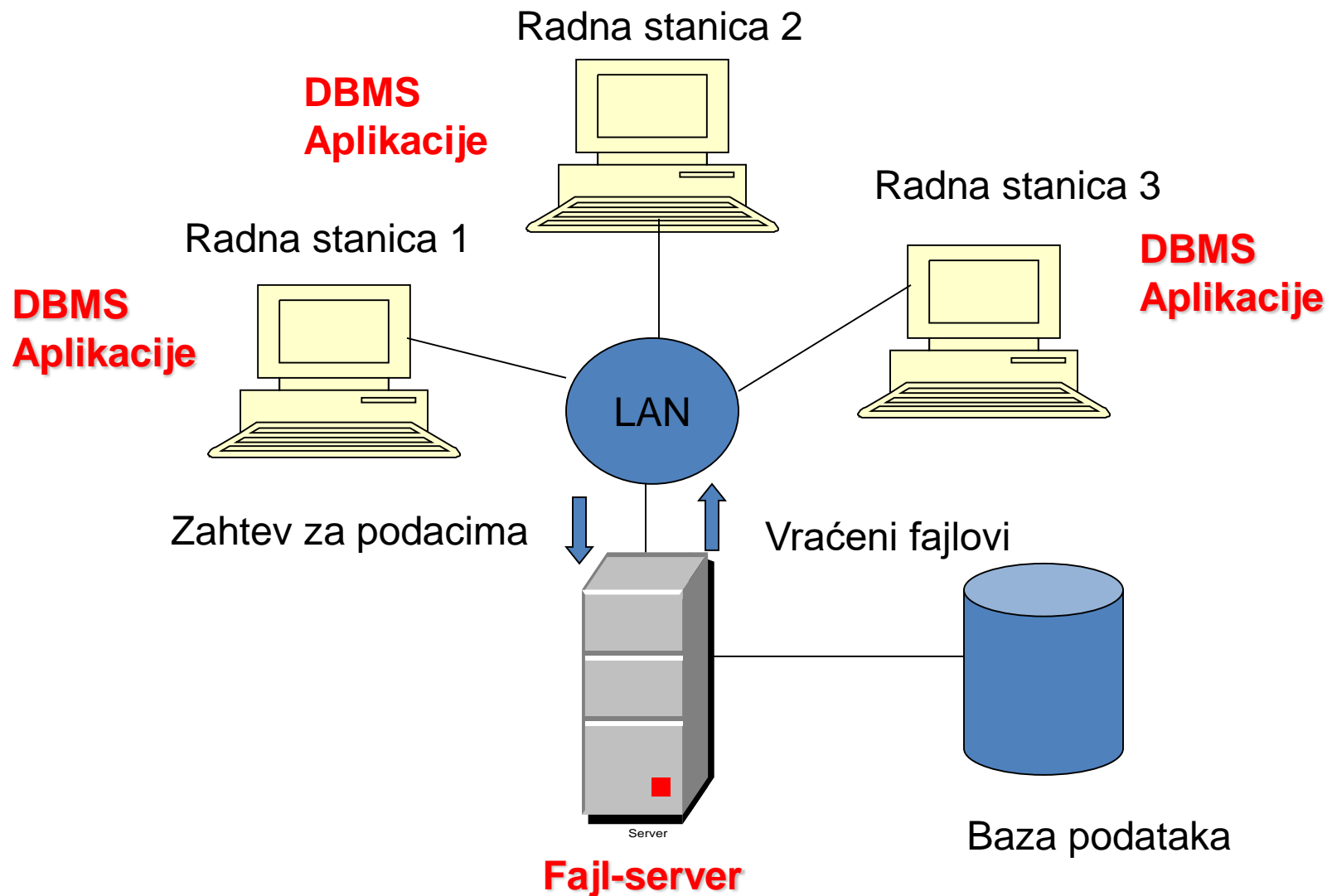


# Teleprocessing

- ▶ Postoji jedan centralni procesor i veći broj terminala
- ▶ Sva obrada se vrši na centralnom procesoru
- ▶ Terminali preko komunikacionog podsistema OS-a šalju poruke aplikacionom programu koji koristeći usluge DBMS-a obrađuje zahtev i vraća poruku korisnikovom terminalu



# Fajl-server arhitektura (1)





# Klijent-server sistemi

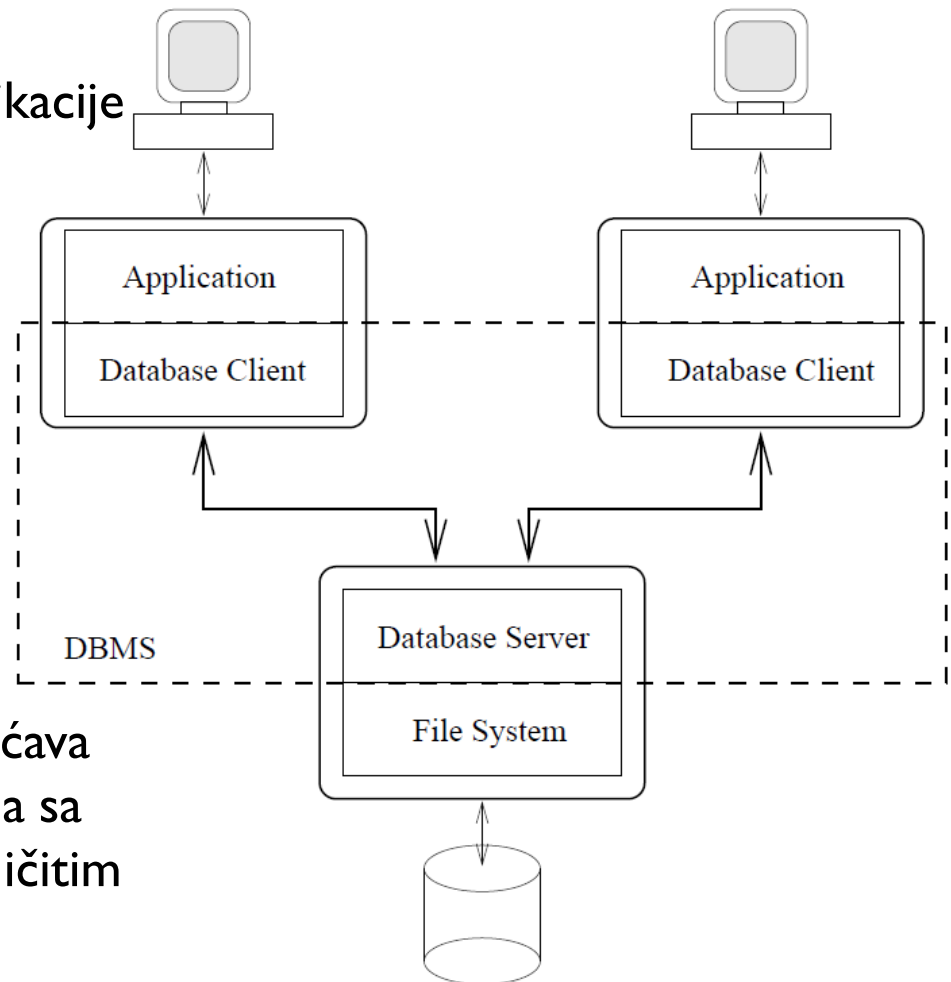
## ▶ **DBMS Klijent**

- ▶ Prihvata zahteve od strane aplikacije
- ▶ Pakuje ih u poruke i
- ▶ Šalje serveru BP
- ▶ Prihvata odgovore
- ▶ Prosleđuje ih aplikaciji

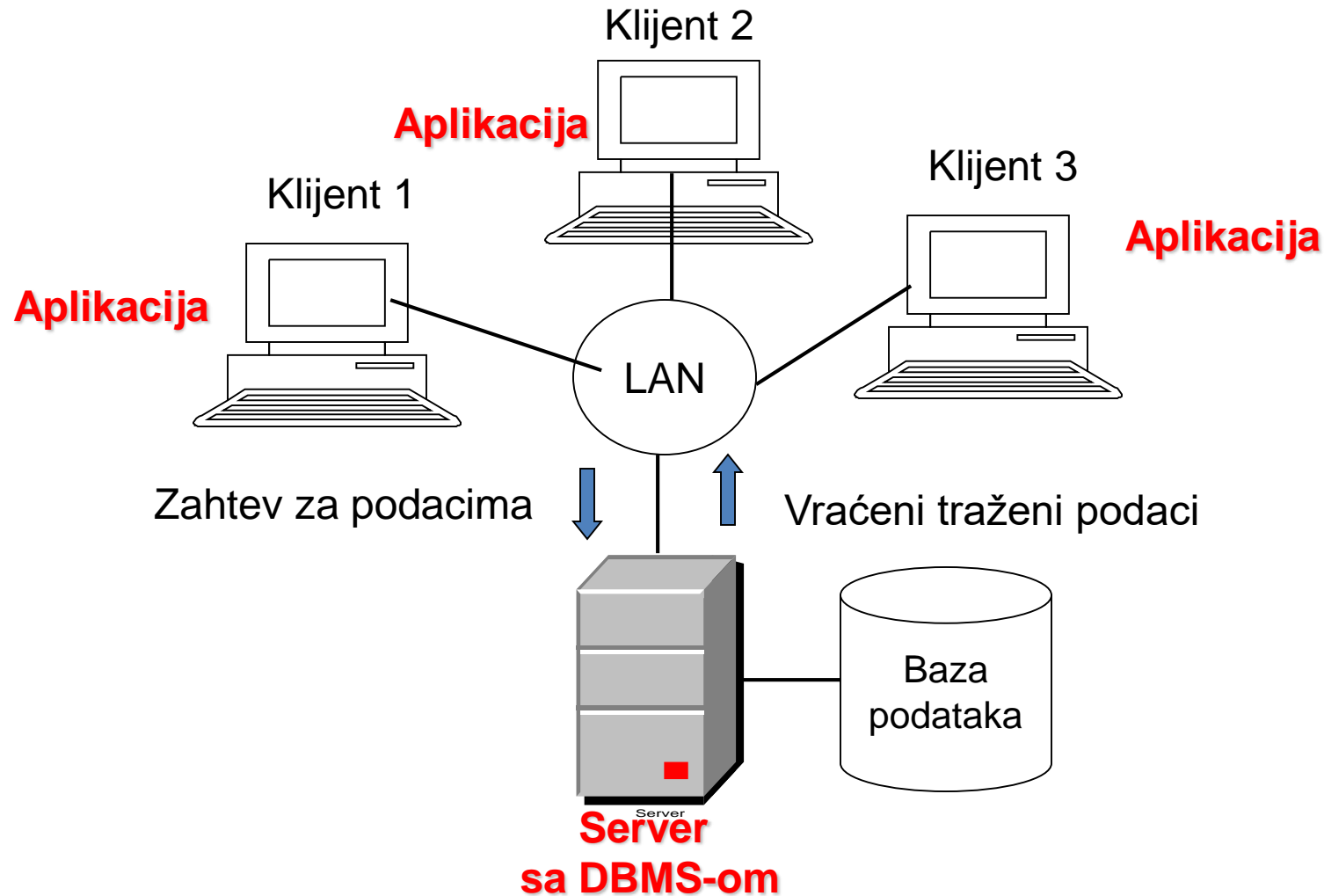
## ▶ **DBMS Server**

- ▶ DBMS funkcionalnost

- ▶ Odvajanje klijenta i servera omogućava da upravljanje podacima i interakcija sa korisnikom bude izvršavano na različitim procesorima

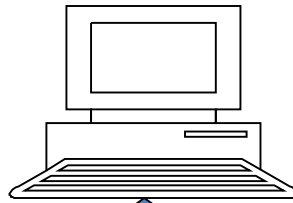


# Implementacija tradicionalne klijent-server arhitekture u dva nivoa (two-tier)



# Dva nivoa (two-tier)

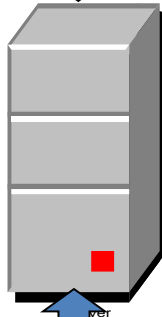
Prvi nivo  
Klijent



## Zadaci

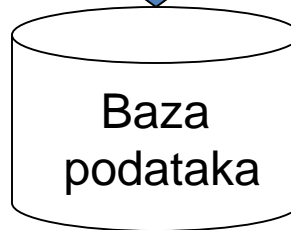
- Korisnički interfejs
- Poslovna logika
- Logika obrade podataka

Drugi nivo  
Server baze podataka  
(sa DBMS-om)



## Zadaci

- Validacija podataka
- Pristup podacima



# Dvoslojna klijent-server arhitektura

---

- ▶ Klijent je odgovoran za prezentaciju podataka korisnicima
- ▶ Server obezbeđuje servise podataka klijentima
  - ▶ Podaci mogu dolaziti od relacionog DBMS-a, objektno-relacionog DBMS-a, objektno-orijentisanog DBMS-a, “legacy” DBMS-a ili od nekog sistema za pristup podacima
- ▶ Klijent se obično izvršava na PC-u i umrežen je sa centralizovanim serverom baze podataka
- ▶ Tipična interakcija između klijenta i servera:
  - ▶ Klijent prihvata zahtev korisnika, proverava sintaksu, generiše zahteve bazi podataka u formi SQL-a ili nekog drugog jezika koji odgovara aplikacionoj logici, šalje poruku serveru i čeka na odgovor
  - ▶ Server prihvata zahtev od klijenta i obrađuje ga, a zatim odgovor šalje natrag klijentu
    - ▶ Obrada obuhvata proveru autorizacije, obezbeđenje integriteta, kontrolu konkurencije i oporavka, održavanje sistemskog kataloga, pretraživanje i ažuriranje baze podataka



# Prednosti dvoslojne klijent server arhitekture

---

- ▶ Omogućava širi pristup postojećim bazama podataka
  - ▶ Poboljšava performanse
    - ▶ Klijentski i serverski proces se mogu paralelno izvršavati na različitim mašinama
    - ▶ Mogu se bolje podesiti performanse servera pošto su mu zadaci preciznije formulisani
  - ▶ Redukuju se troškovi hardvera
    - ▶ Samo server treba da bude dovoljno moćan da upravlja bazama podataka
  - ▶ Redukuju se komunikacioni troškovi
    - ▶ Deo operacija se izvodi na klijentu, pa se kroz mrežu šalju samo zahtevi bazi podataka
  - ▶ Veća je konzistentnost podataka
    - ▶ Server rukuje proverom integriteta, dok su aplikacije oslobođene
  - ▶ Sasvim prirodno se preslikava na arhitekturu otvorenih sistema
- 



# Nedostaci dvoslojne klijent-server arhitekture

---

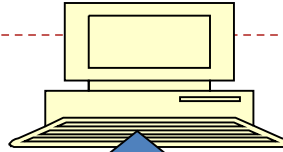
- ▶ “Debeo” klijent iziskuje više resursa na klijentskim mašinama (moćniji CPU, veći kapacitet RAM-a i diskova)
- ▶ Značajni troškovi administriranja na klijentskoj strani
- ▶ Ovi nedostaci su otklonjeni kod troslojne klijent-server arhitekture



# Tri nivoa (three-tier)

## Prvi nivo

Klijent

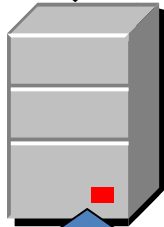


## Zadaci

- Korisnički interfejs

## Drugi nivo

Aplikacioni server

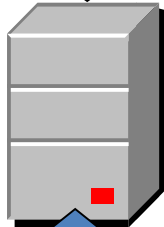


## Zadaci

- Poslovna logika
- Logika obrade podataka

## Treći nivo

Server baze podataka  
(sa DBMS-om)



## Zadaci

- Validacija podataka
- Pristup podacima



# Troslojna klijent server arhitektura

---

- ▶ Klijent je odgovoran jedino za korisnički interfejs aplikacije i može izvoditi samo prostu logiku obrade (npr. Validaciju ulaznih podataka)
  - ▶ Radi se o “tankom” klijentu
- ▶ Poslovna logika je smeštena u srednjem sloju koji je fizički povezan sa klijentom i serverom baze podataka preko LAN-a i WAN-a
  - ▶ Jedan aplikacioni server se projektuje za više klijenata





# Prednosti troslojne klijent server arhitekture

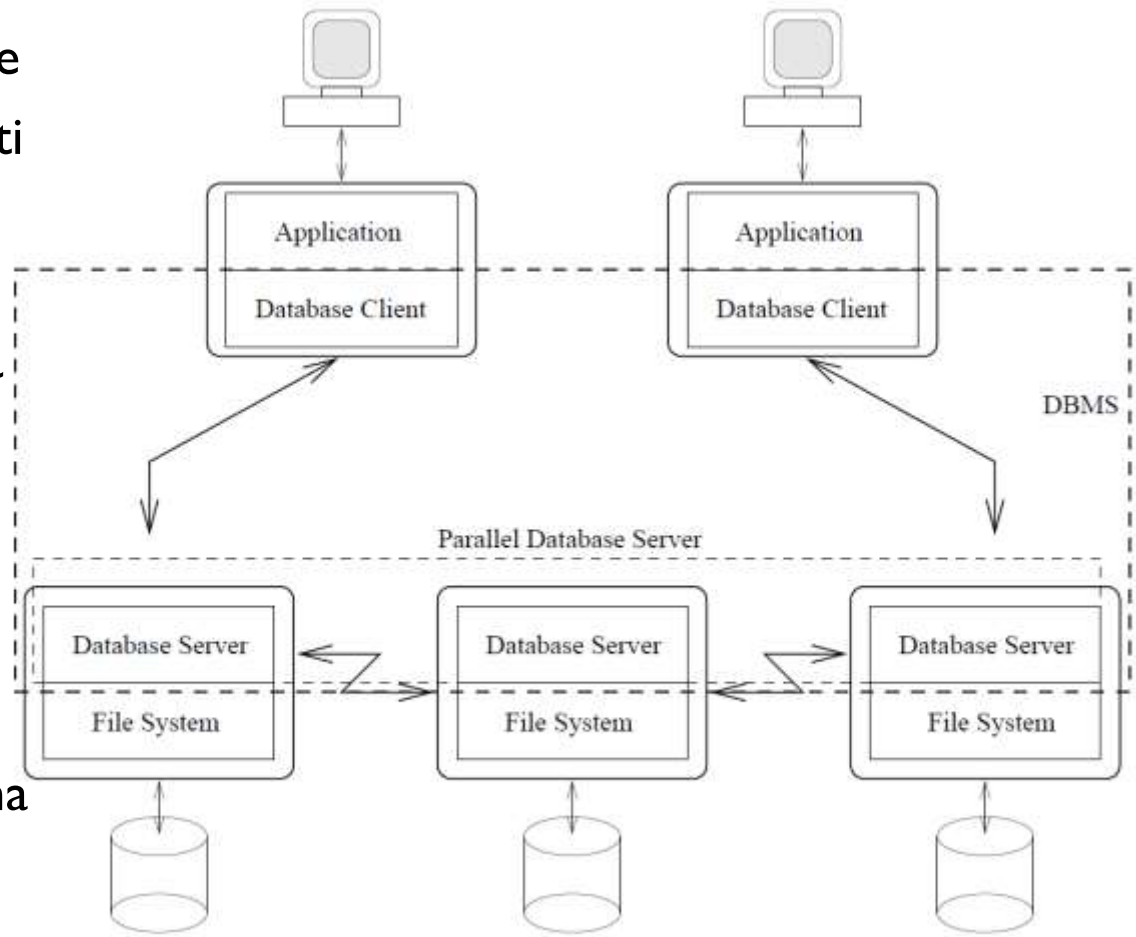
---

- ▶ Manji troškovi za hardver (klijenti su tanki)
- ▶ Održavanje aplikacije je centralizovano
  - ▶ Gro poslovne logike je na jednom mestu (aplikacionom serveru) umesto na više klijenata
  - ▶ Nema distribucije softvera, što je bio izvor mnogih problema i troškova
- ▶ Dodatna modularnost omogućava lakšu zamenu jednog sloja bez uticaja na ostale
- ▶ Lakše je balansiranje opterećenja pošto je poslovna logika odvojena od funkcija baze podataka
- ▶ Sasvim prirodno se preslikava na **Web okruženje**
  - ▶ Klijent je **Web browser**
  - ▶ Aplikacioni server je **Web server**
- ▶ Pogodna je za Internet i intranet okruženje



# Paralelni/distribuirani server BP

- ▶ Podaci su distribuirani
- ▶ Relacije mogu biti fragmetisane
- ▶ Relacije (ili fragmenti) mogu biti replicirane na nekoliko mesta
- ▶ **Serveri-komponente treba da rade zajedno**
  - ▶ Postoji jedna **zajednička šema**
  - ▶ Distribucija podataka je transparentna
  - ▶ Replikacija je transparentna
  - ▶ Fragmentacija je transparentna



# Paralelni u odnosu na Distribuirani server

---

## ▶ **Paralelni server BP**

- ▶ Serveri se nalaze blizu jedan drugog
- ▶ Mora da postoji brza, stalna komunikacija između servera, obično preko LANa ili deljive memorije
- ▶ Upiti se obično obrađuju kooperativno od svih servera

## ▶ **Distribuirani server BP**

- ▶ Serveri mogu da se nalaze bilo gde
- ▶ Server-to-server komunikacija može da bude spora, obično preko WANa
- ▶ Upiti se obrađuju obično na jednom od servera

# Horizontalna fragmentacija

## Kompletna relacija

Vno	Vname	City	Vbal
1	Sears	Toronto	200.00
2	Kmart	Ottawa	671.05
3	Eatons	Toronto	301.00
4	The Bay	Ottawa	162.99

## Horizontalna fragmetacija relacije (na 2 lokacije):

### Lokacija 1: Ottawa

Vno	Vname	City	Vbal
2	Kmart	Ottawa	671.05
4	The Bay	Ottawa	162.99

### Lokacija 2: Toronto

Vno	Vname	City	Vbal
1	Sears	Toronto	200.00
3	Eatons	Toronto	301.00

# Vertikalna fragmentacija

## Kompletna relacija

Vno	Vname	City	Vbal
1	Sears	Toronto	200.00
2	Kmart	Ottawa	671.05
3	Eatons	Toronto	301.00
4	The Bay	Ottawa	162.99

## Vertikalna fragmetacija relacije (na 2 lokacije):

### Lokacija 1

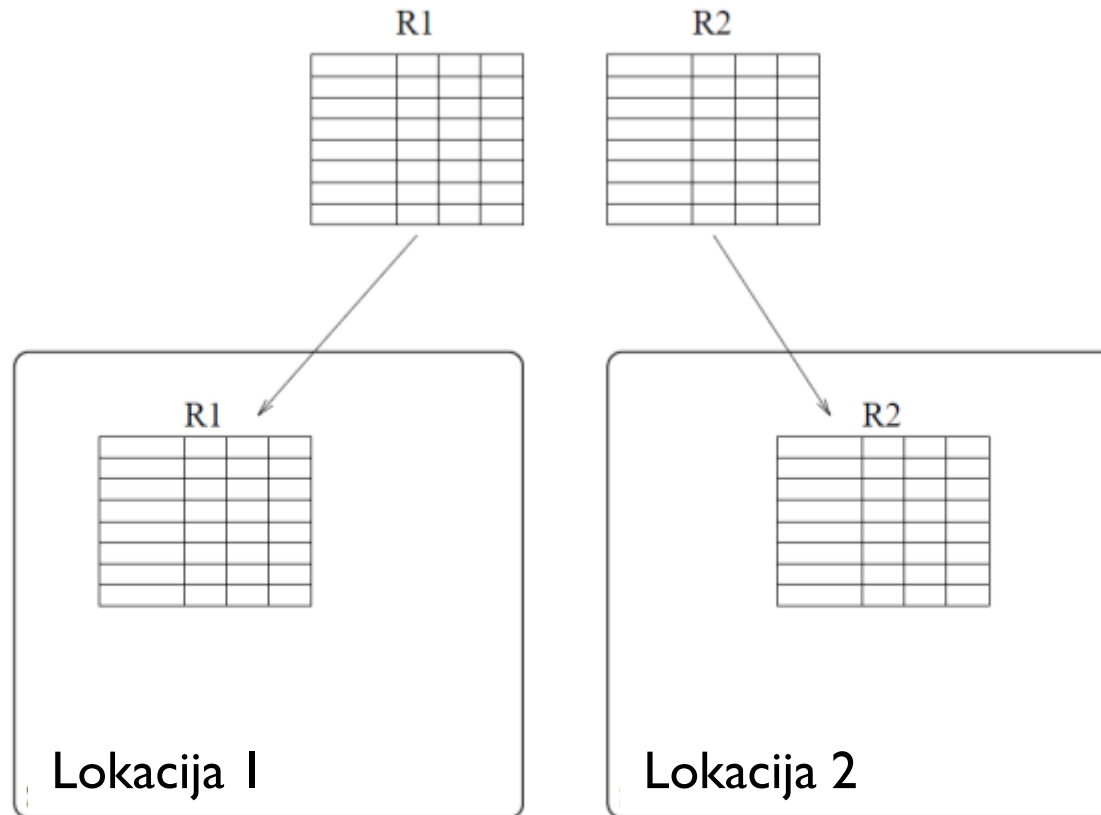
Vno	Vname	City
1	Sears	Toronto
2	Kmart	Ottawa
3	Eatons	Toronto
4	The Bay	Ottawa

### Lokacija 2

Vno	Vbal
1	200.00
2	671.05
3	301.00
4	162.99

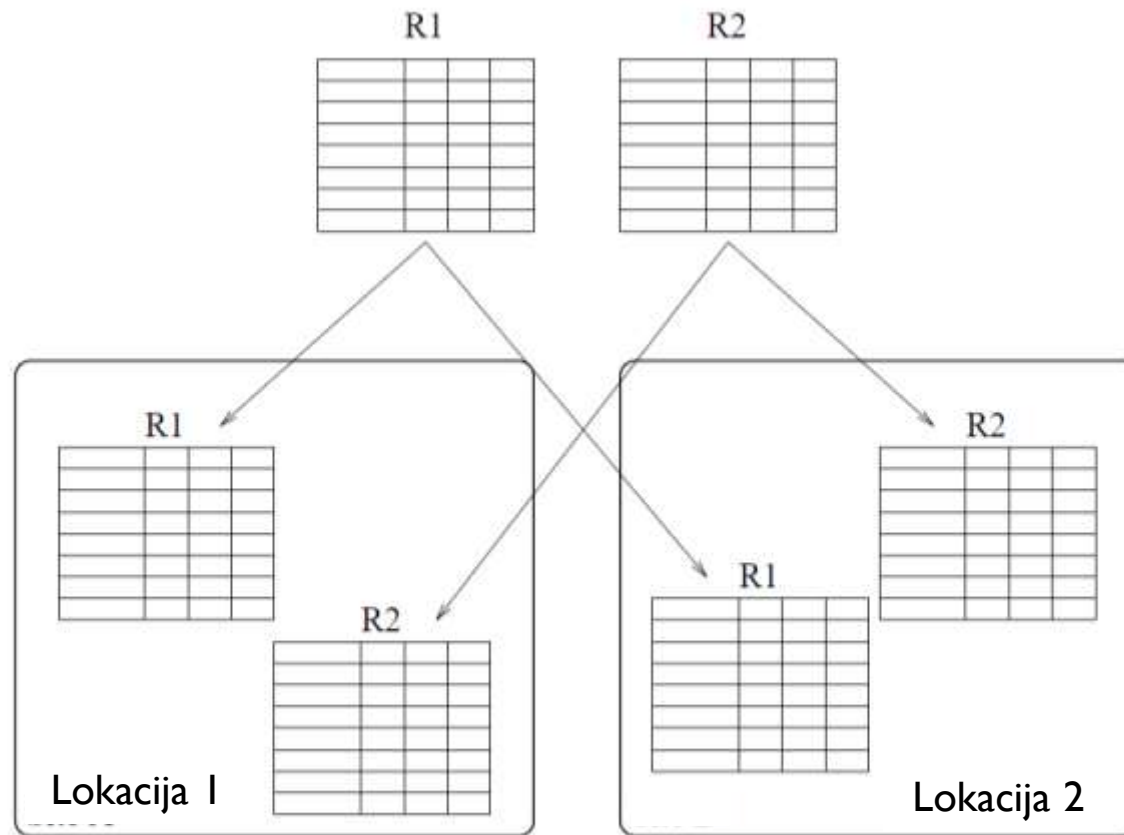
# Distribucija podataka

## Relacije (ili fragmenti)



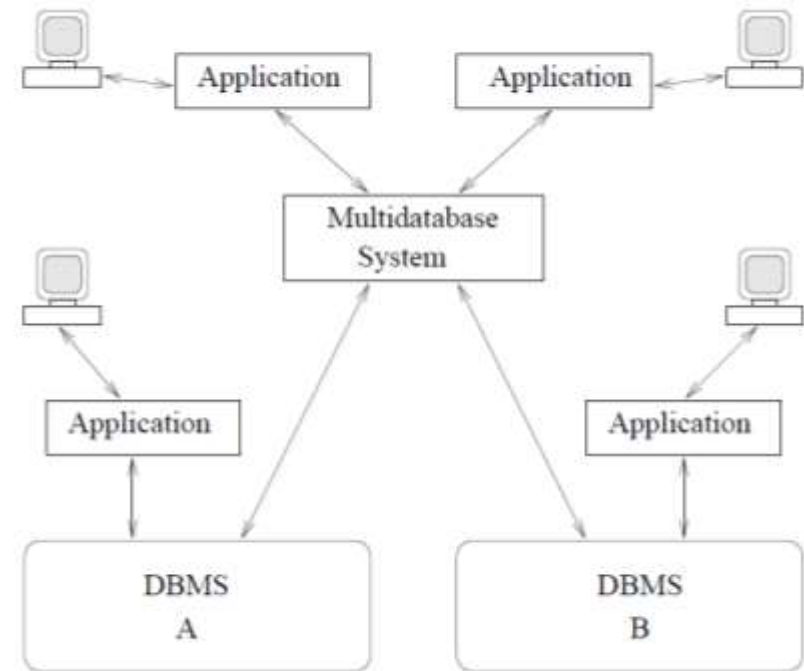
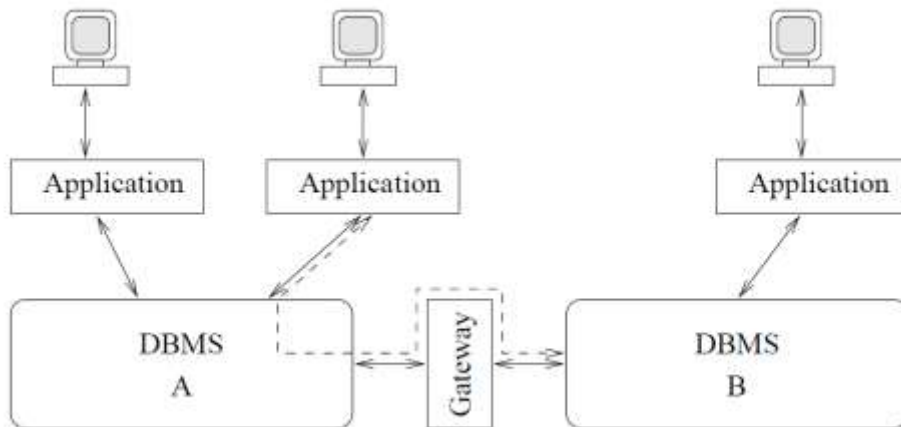
# Replikacija podataka

Relacije (ili fragmenti)



# Sistemi multi-baza podataka

- ▶ Aplikacije “vide” jedan sistem BP
- ▶ Sistemi BP su autonomni
- ▶ Pristup može da ide i preko komponente tzv “gateway”





Arhitecture sistema baza podataka

Pitanja ???