



**UNIVERZITET U NOVOM SADU**  
**FAKULTET TEHNIČKIH NAUKA**  
**KATEDRA ZA AUTOMATIKU I UPRAVLJANJE SISTEMIMA**

# Arhitektura

Distribuirani sistemi

Distribuirano programiranje

# Uvod

- Distribuirani sistem (DS) često sačinjava veoma složen softver koji je po definiciji razruđen na više računara
- Neophodna je propisna organizacija softvera
  - Za lakše rukovanje sistemom
- Treba razlikovati
  - Logičku organizaciju softverskih komponenti – **Softverska arhitektura**
    - Podela posla u DS po celinama - komponentama
  - Fizičku realizaciju softverskih komponenti – **Sistemska arhitektura**
    - Realizacija gde se primerci komponenti raspoređuju na stvarne računare

# Softverske komponente

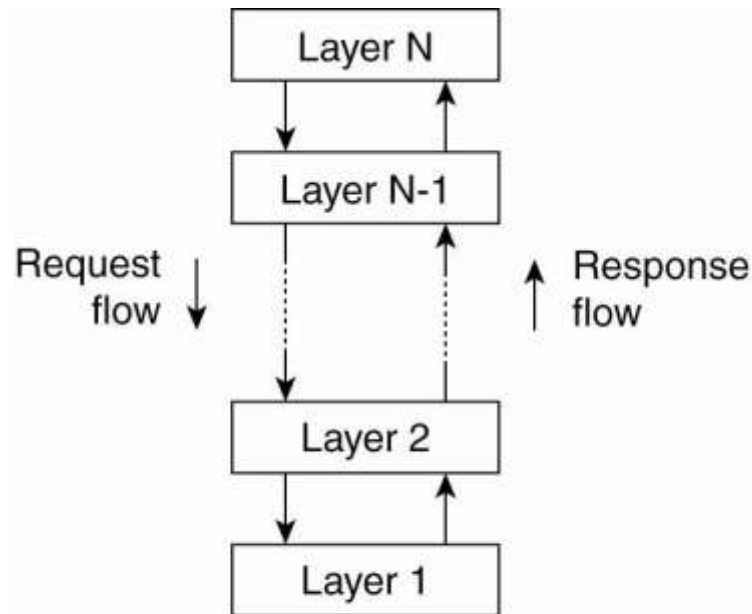
- Komponenta je modularna softverska jedinica
  - Sa jasno definisanim **interfejsom**
  - Implementacija je izmenjiva
- Softverske komponente su gradivne jedinice DS
  - Komponente su povezane i razmenjuju podatke
  - Način povezivanja komponenti definiše stil softverske arhitekture

# Stilovi softverske arhitekture

- Cilj je uspostaviti distribucionu transparentnost.  
Otežavaju je poteškoće koje se odnose na:
  - Performanse
  - Tolerantnost otkaza
  - Lakoću programiranja
  - ...
- Važni stilovi arhitekture DS:
  1. Slojevite arhitekture - *Layered architectures*
  2. Objektno-orijentisane arhitekture - *Object-based architectures*
  3. Podacima usresređene arhitekture - *Data-centered architectures*
  4. Arhitekture zasnovane na događajima - *Event-based architectures*
- Ne postoji jedan stil koji se može upotrebiti i većini DS.

# Stil 1 - Slojevita arhitektura

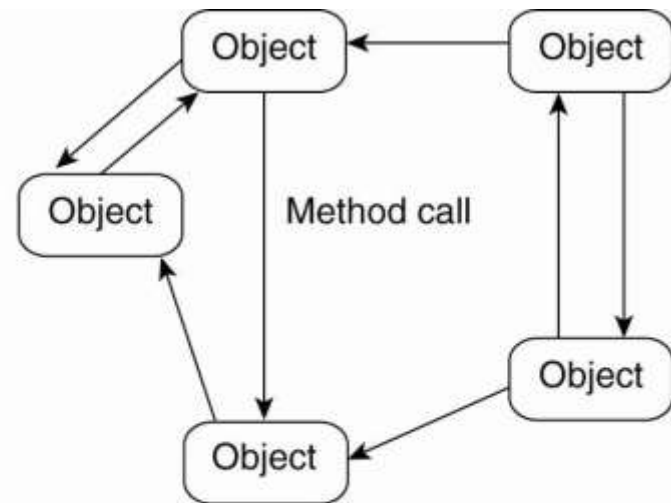
- Komponente su organizovane po slojevima
- Postoji hijerarhija slojeva
  - Samo susedni slojevi komuniciraju



(a)

# Stil 2 - Objektno-orijentisana arhitektura

- Slobodnija organizacija komponenti u odnosu na slojevitu arhitekturu
- Svaki objekat odgovara komponenti
- Objekti su tipično dislocirani i komuniciraju upotrebom *Remote Procedure Call* (RPC)
  - Veze su po modelu klijent-server



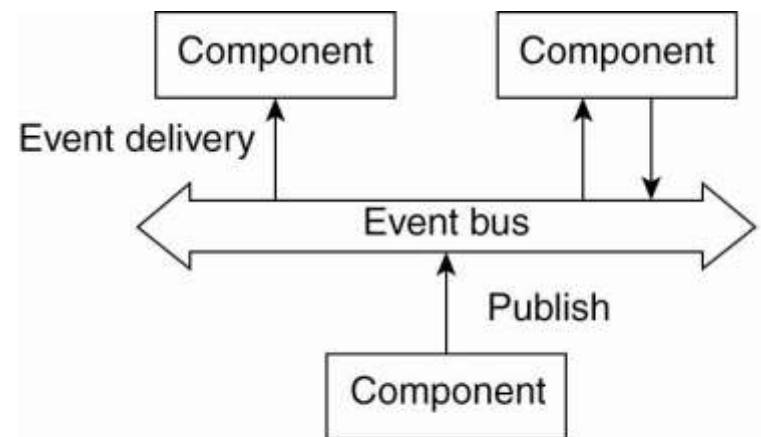
(b)

# Stil 3 - Podacima usresređena arhitektura

- Komponente (tačnije, procesi) komuniciraju preko zajedničkih skladišta podataka
  - Aktivnih ili pasivnih
- Postoje implementacije DS gde se sva komunikacija odvija preko deljenih datoteka.

# Stil 4 - Arhitektura zasnovana na događajima

- Komponente komuniciraju (uglavnom) preko propagacije događaja
  - Pri tome se prenose i podaci koji se odnose na događaj
- Ovi DS se nazivaju i **Publish/Subscribe** sistemi
  - Neke komponente emituju poruke
  - Druge komponente primaju poruke
    - pretplate se na neke tipove poruka i onda samo takve poruke primaju
  - *Middleware* se brine o mehanizmima razmene poruka
  - Efekat je **slaba povezanost** komponenti
    - Komponente ne moraju eksplicitno da se obraćaju jedna drugoj

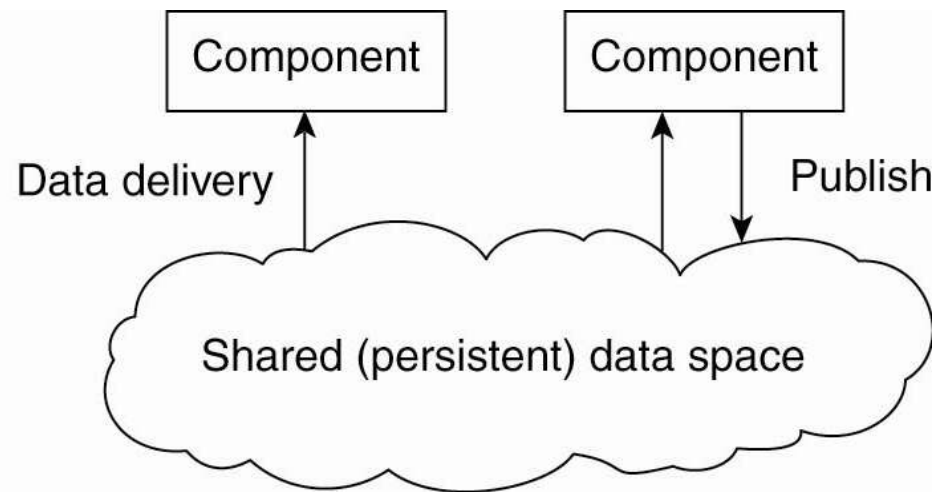


(a)



# Stilovi 3 i 4 – *Shared data spaces*

- Kombinovanje arhitekture zasnovane na događajima i podacima usresređene arhitekture
  - Ne zahteva se da komponente koje komuniciraju budu jednovremeno aktivne



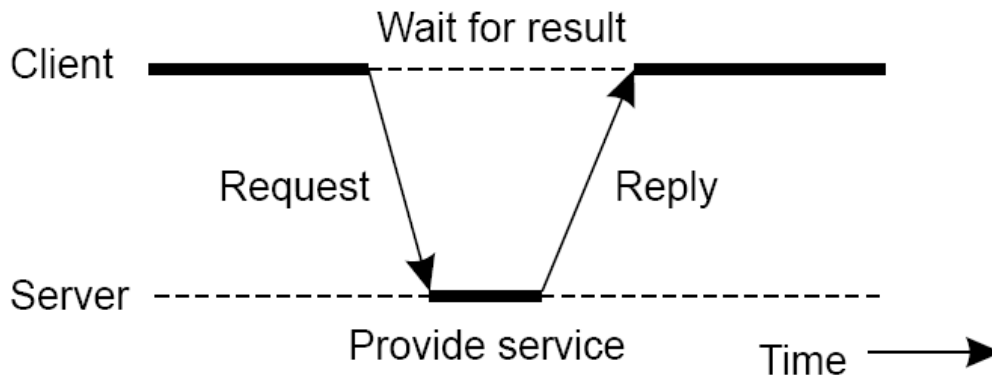
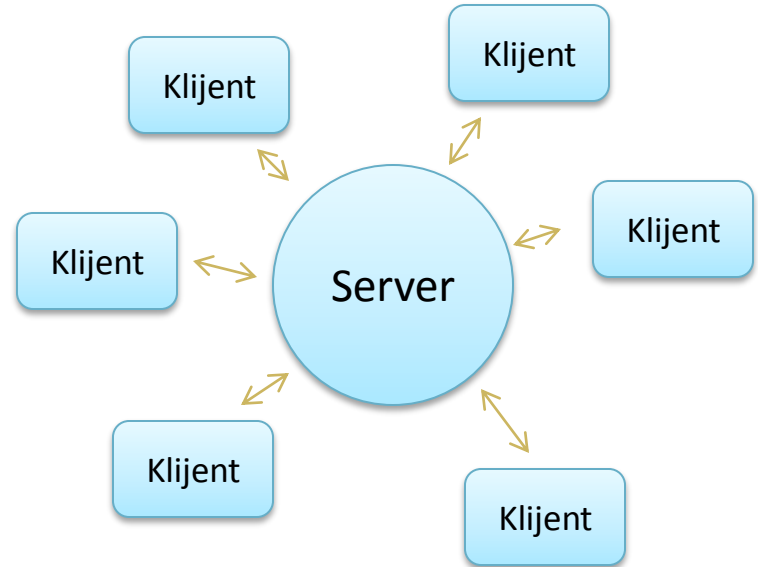
(b)

# Sistemske arhitekture

- Govori o fizičkom rasporedu komponenti
- Organizacije komponenti
  - Centralizovana organizacija
  - Decentralizovana organizacija
  - mešovita

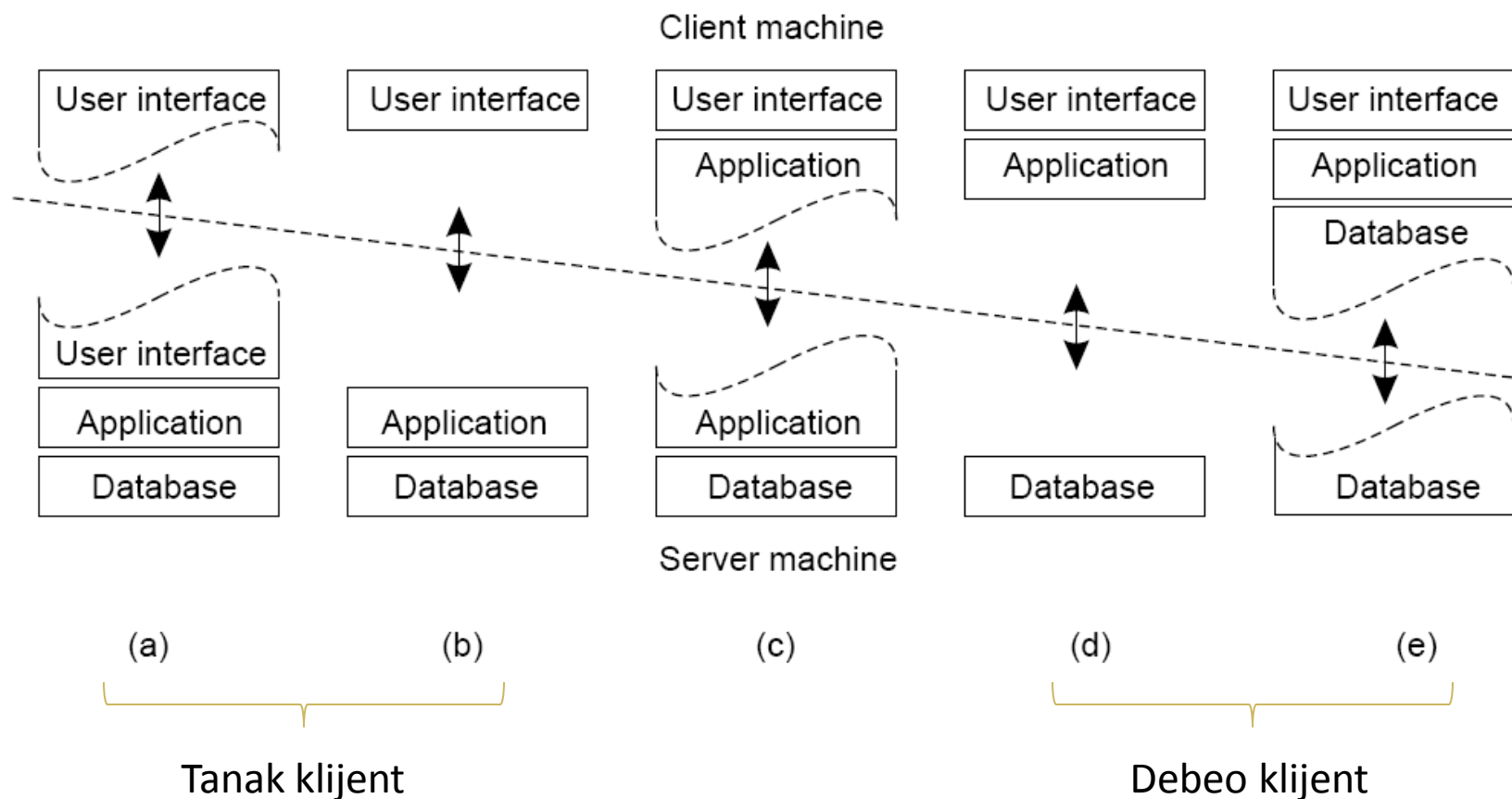
# Centralizovana arhitektura

- Poznatija kao **Klijent-Server** model
- Učesnici su dve komponente (procesa)
  - **Server** – implementira servis
  - **Klijent** – koristi uslugu servisa
- **Upit-odgovor** ponašanje



# Klijent-server arhitektura

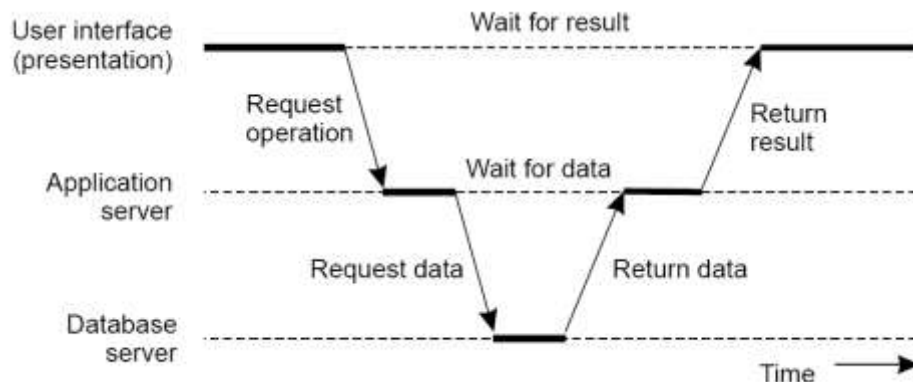
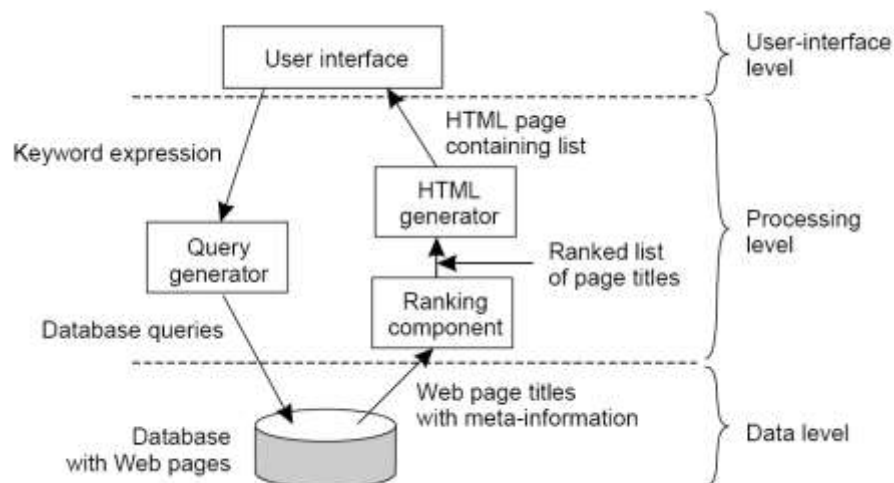
- (Fizička) dvoslojna arhitektura (Klijent-Server) ima svoje alternative
- Primer troslojne logičke organizacije



# Višeslojna arhitektura softvera

- Slojevitost prati logičnu organizaciju aplikacije
- Često se koristi troslojna arhitektura softvera
  1. sloj korisničkog interfejsa (*UI – User Interface*)
    - unos i prikaz podataka
  2. sloj obrade (*processing*)
    - obrada zahteva ili rezultata
  3. sloj podataka (*data*)
    - skladištenje (perzistencija) podataka
- Komunikacija između slojeva se odvija po Klijent-Server modelu

## Primer: Pretraga preko Interneta



# Decentralizovane arhitekture

- **Vertikalna distribucija** – razmeštanje komponenti iz logički razdvojenih slojeva na različite računare
  - Logička i fizička arhitektura su izjednačene
- **Horizontalna distribucija** – klijent ili server se deli na logički jednake delove koji se izvršavaju na različitim računarima
  - Balansira se opterećenje
  - Često se nazivaju *peer-to-peer* sistemim
  - Iz globalne perspektive:
    - Svi čvorovi su jednaki
    - Međusobna interakcija je simetrična – svaki je i server i klijent

# Arhitektura i *middleware*

- *Middleware* je sloj između distribuiranih aplikacija i platformi (operativnih sistema)
  - Treba da pomogne u postizanju (distributivne) transparentnosti
  - Sakriva
    - Distribuiranost podataka
    - Distribuiranost procesiranja
    - Upravljanje iz aplikacija
- Postojeći *middleware*-i su pratili specifične stilove u arhitekturi
  - OMG CORBA; Microsoft DCOM – adaptiran objektno-orijentisan stil
  - TIBCO – prati stil arhitekture zasnovane na događajima
  - ...
- Upotreba *middleware* nameće arhitektonski stil
  - Savremena rešenja DS traže prilagodljiv *middleware*