



Diplomski studij

**Informacijska i
komunikacijska tehnologija:**

Telekomunikacije i informatika

Računarstvo:

Programsko inženjerstvo i
informacijski sustavi

Računarska znanost

Ak.g. 2008./2009.

Raspodijeljeni sustavi

1.

Uvod u raspodijeljene sustave

10.9.2008.

- ◆ Definicija, obilježja i vrste raspodijeljenih sustava
- ◆ Otvorenost, transparentnost i skalabilnost
- ◆ Arhitektura raspodijeljenih sustava
- ◆ Modeli raspodijeljene obrade:
 - klijent – poslužitelj
 - objavi – pretplati
 - ravnopravni sudionici
 - pokretni kod i programski agenti

Definicija raspodijeljenog sustava (1)



Andrew S. Tannenbaum:

- ◆ Skup neovisnih računala koji korisniku izgleda kao jedan cjeloviti sustav.

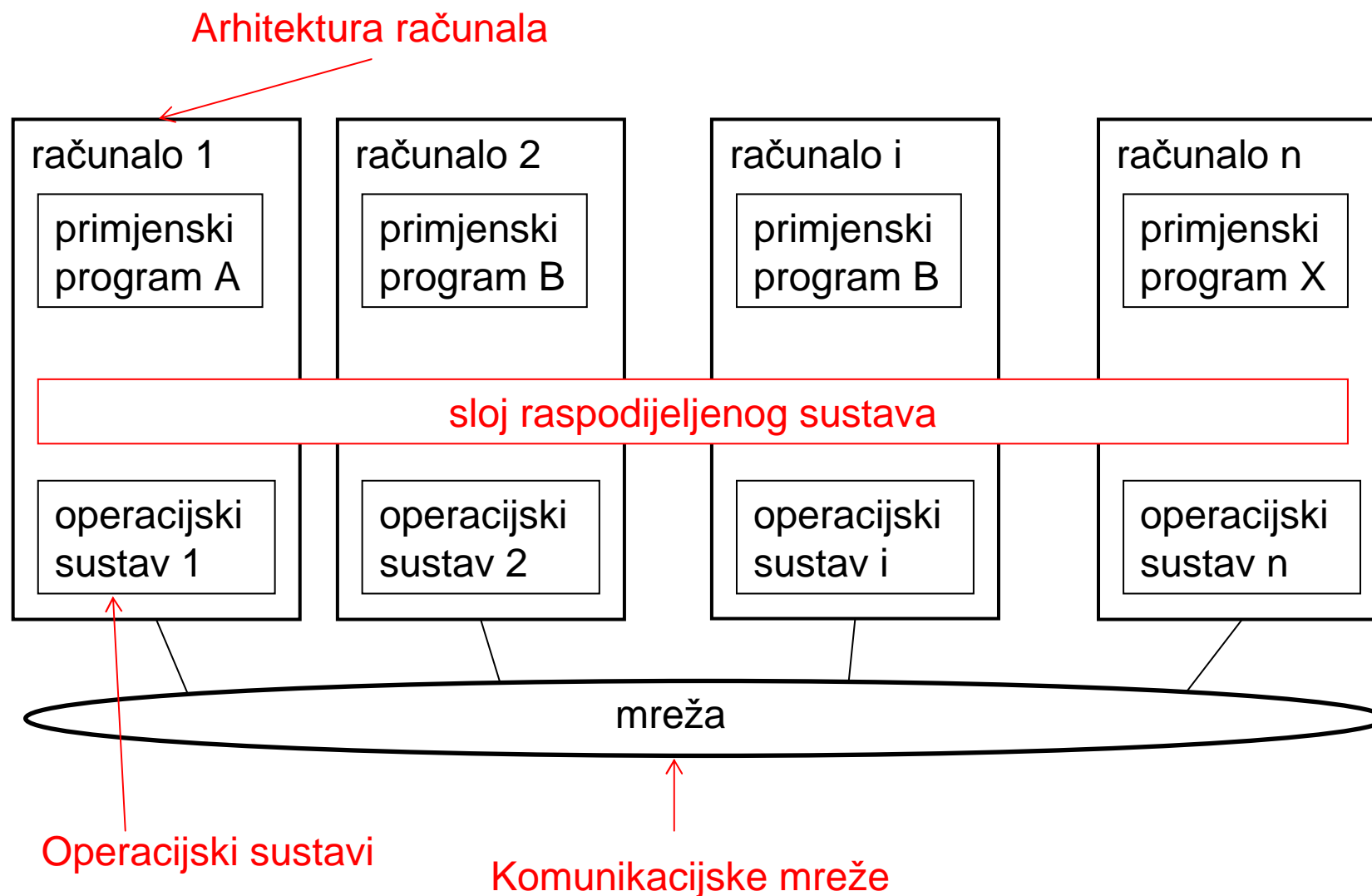
George Coulouris:

- ◆ Sustav u kojem programske i sklopovske komponente umreženih računala komuniciraju i usklađuju svoje aktivnosti isključivo razmjenom poruka.

Leslie Lamport:

- ◆ Sustav u kojem kvar računala za koje uopće ne znate da postoji može učiniti vaše računalo neupotrebljivim.

Definicija raspodijeljenog sustava (2)



Programski posrednički sloj, međuoprema (engl. *middleware*)

- ◆ prikriva činjenicu da su procesi i sredstva (resursi) raspodijeljeni na više računala u mreži
- ◆ omogućuje povezivanje i suradnju aplikacija, sustava i uređaja,
- ◆ omogućuje interakciju programa na aplikacijskoj razini,
- ◆ nalazi se između operacijskog sustava i primjenskih programa,
- ◆ nalazi se iznad mrežnog, a ispod aplikacijskog sloja

Paralelne aktivnosti

- ◆ autonomne komponente sustava istodobno izvode aktivnosti

Komunikacija razmjenom poruka:

- ◆ bez zajedničke memorije

Dijeljenje sredstava:

- ◆ zajedničkim sredstvima pristupa više komponentata

Nema globalnog stanja:

- ◆ niti jedan proces ne zna globalno stanje sustava

Nema globalnog vremenskog takta:

- ◆ ograničena mogućnost vremenskog usklađivanja

Razlozi za raspodijeljene sustave



Inherentna raspodijeljenost:

- ◆ Korisnika, informacija, sredstava, ...

Funkcionalno odvajanje:

- ◆ različite namjene, različite mogućnosti, različite uloge (korisnik – davatelj usluge, proizvođač-potrošač,)

Opterećenje:

- ◆ raspodjela i uravnoteženje opterećenja

Pouzdanost:

- ◆ više komponenata na različitim mjestima

Cijena, troškovi

....

Raspodijeljeni računalni sustavi:

- ◆ grozd, nakupina (engl. *cluster*)
- ◆ spleto (engl. *grid*)

Raspodijeljeni informacijski sustavi:

- ◆ sustav za obradu transakcija (engl. *Transaction Processing System*)
- ◆ integracija poslovnih primjena (engl. *Enterprise Application Integration*)

Sustavi za pružanje informacijskih i komunikacijskih usluga:

- ◆ sloj usluga i aplikacija (engl. *Service & Application Layer*)

Prožimajući sustavi (engl. *pervasive system*)

- ◆ senzorska mreža (engl. *sensor network*)

◆ Otvorenost

- otvoreni sustav (engl. *open system*): pruža usluge sukladno normiranim pravilima te definiranoj sintaksi i semantici

◆ Transparentnost

- prikrivanje odabranih značajki raspodijeljenog sustava
- utječe na složenost, performanse i troškove sustava
- oblikuje se sukladno korisničkim zahtjevima

◆ Skalabilnost

- sposobnost razmjerne prilagodbe veličini (broj korisnika – količina sredstva), rasprostranjenosti (lokalno, regionalno, globalno, ...) i načinu upravljanja (jedna ili više administrativnih domena)

Norma (standard) je specifikacija koja je:

- ◆ Široko prihvaćena u industriji (**de facto standard**) ili zastupana od normizacijskog tijela (**de jure standard**),
- ◆ Dobro definirana,
- ◆ Neutralna, tj. vlasnički neovisna i
- ◆ Javno dostupna.

Otvorenost je pretpostavka za:

- ◆ međudjelovanje (engl. *interoperability*)
- ◆ prenosivost (engl. *portability*)
- ◆ proširljivost (engl. *extensibility*)

Transparentnost (1)



Osnovne značajke raspodijeljenog sustava s motrišta transparentnosti:

- ◆ Transparentnost pristupa
- ◆ Lokacijska transparentnost
- ◆ Migracijska transparentnost
- ◆ Transparentnost premještanja
- ◆ Replikacijska transparentnost
- ◆ Konkurencijska transparentnost
- ◆ Transparentnost na kvar

Transparentnost pristupa (engl. *access transparency*)

- ◆ Prikrivanje različitih načina pristupa sredstvima (arhitektura) i razlika u predočavanju podataka (arhitektura, operacijski sustav)
- ◆ primjer: organizacija datoteka

Lokacijska transparentnost (engl. *location transparency*)

- ◆ Prikrivanje položaja sredstva: položaj sredstva u sustavu ne treba biti i nije poznat korisniku
- ◆ Primjer: naziv umjesto lokacije resursa

Transparentnost (3)



Migracijska transparentnost (engl. *migration transparency*)

- ◆ Prikrivanje promjene lokacije: promjena lokacije sredstva ne utječe na način pristupa sredstvu
- ◆ Primjer: nomadski pristup

Transparentnost premještanja (engl. *relocation transparency*)

- ◆ Prikrivanje premještanja/kretanja sredstva: sredstvo smije mijenjati položaj tijekom pristupa sredstvu, tj. tijekom uporabe
- ◆ Primjer: pokretni pristup (Mobile CARNet)

Transparentnost (4)



Replikacijska transparentnost (engl. *replication transparency*)

- ◆ Prikrivanje više istovrsnih sredstava ili preslika nekog sredstva
- ◆ Pretpostavka: lokacijska transparentnost
- ◆ Primjer: sustav s više poslužitelja

Konkurencijska transparentnost (engl. *concurrency transparency*)

- ◆ Prikrivanje uporabe istog resursa od drugih korisnika: zajednička/dijeljena uporaba sredstva
- ◆ Problem: očuvanje konzistentnosti
- ◆ Primjer: svaki poslužitelj

Transparentnost (5)



Transparentnost na kvar (engl. *failure transparency*)

- ◆ Prikrivanje kvara: otkrivanje kvara i obnavljanje sustava nakon kvara nije uočljivo korisnicima

Oblikovanje raspodijeljenih sustava:

- ◆ Kakva je transparentnost potrebna?
- ◆ Kako transparentnost utječe na performanse?

Da bi se uz promjenu broja korisnika održale performanse sustava uz prihvatljive troškove treba:

više (istovrsnih) dijelova

Koliko?

prostorno raspodijeljenih

Gdje?

koji komuniciraju asinkrono

Kako?

Programska arhitektura:

- ◆ logička organizacija sustava: programske komponente sustava, njihova organizacija i interakcija

Sustavka arhitektura:

- ◆ smještaj programskih komponenata na jedno računalo (centralizirana arhitektura) ili više njih (decentralizirana arhitektura)

Modeli raspodijeljene obrade:

- ◆ klijent-poslužitelj, objavi-pretplati, ravnopravni sudionici, pokretni kod/prograsmki agent

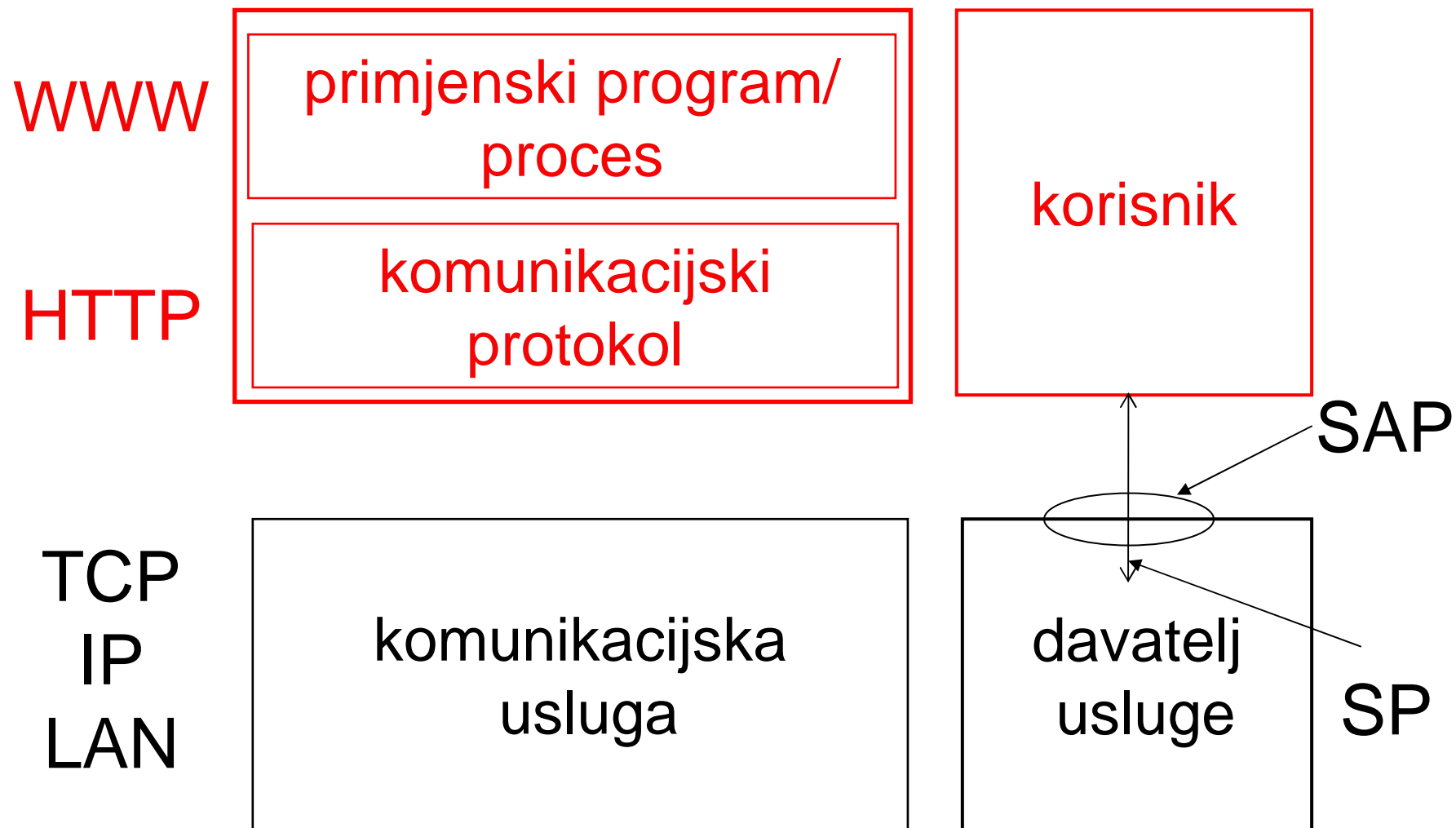
Slojeviti modeli

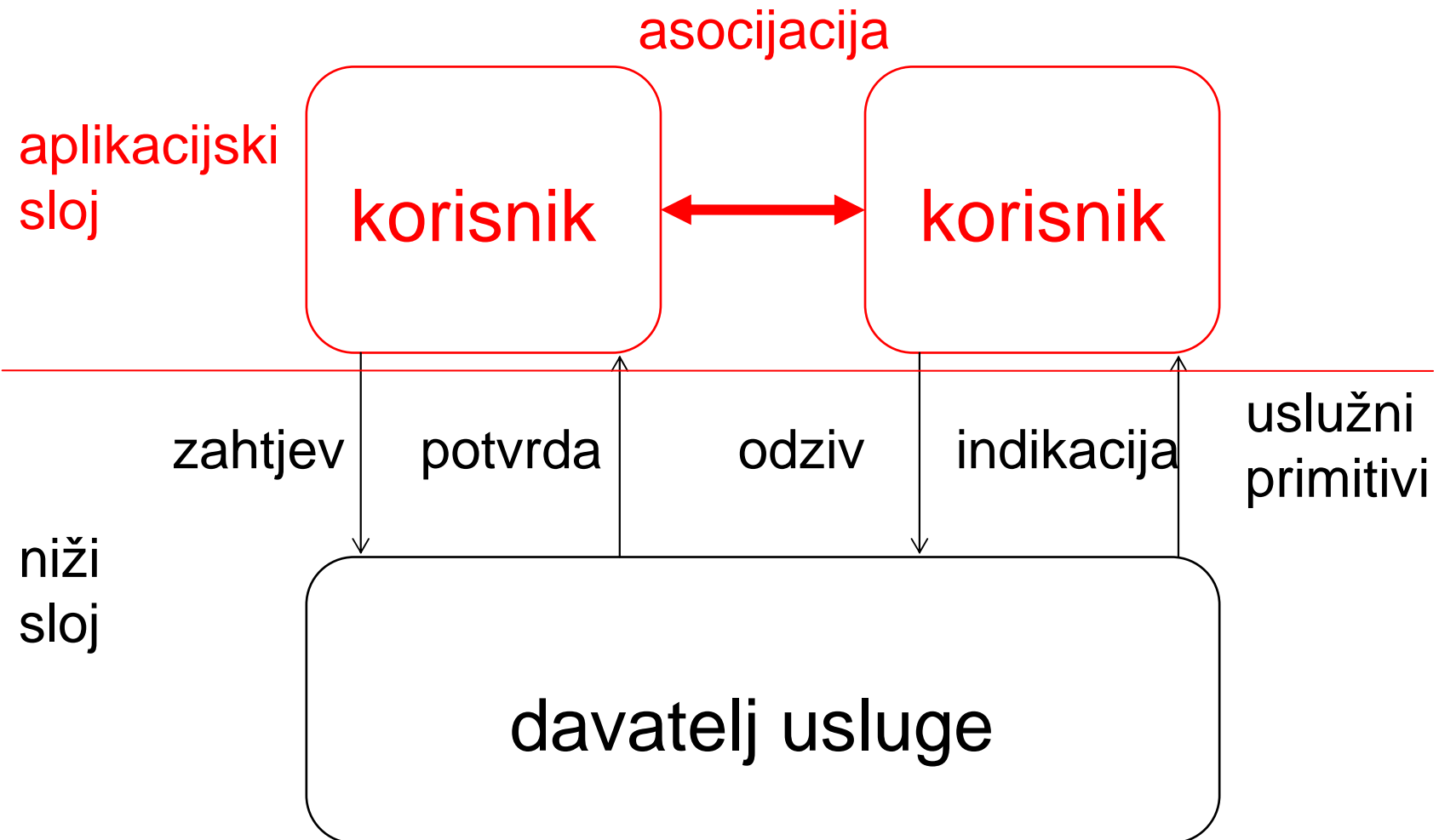
- ◆ U središtu pozornosti aplikacijski sloj (sloj primjene)
 - primjenski programi i primjenski procesi te
 - usluge koje im pružaju niži slojevi

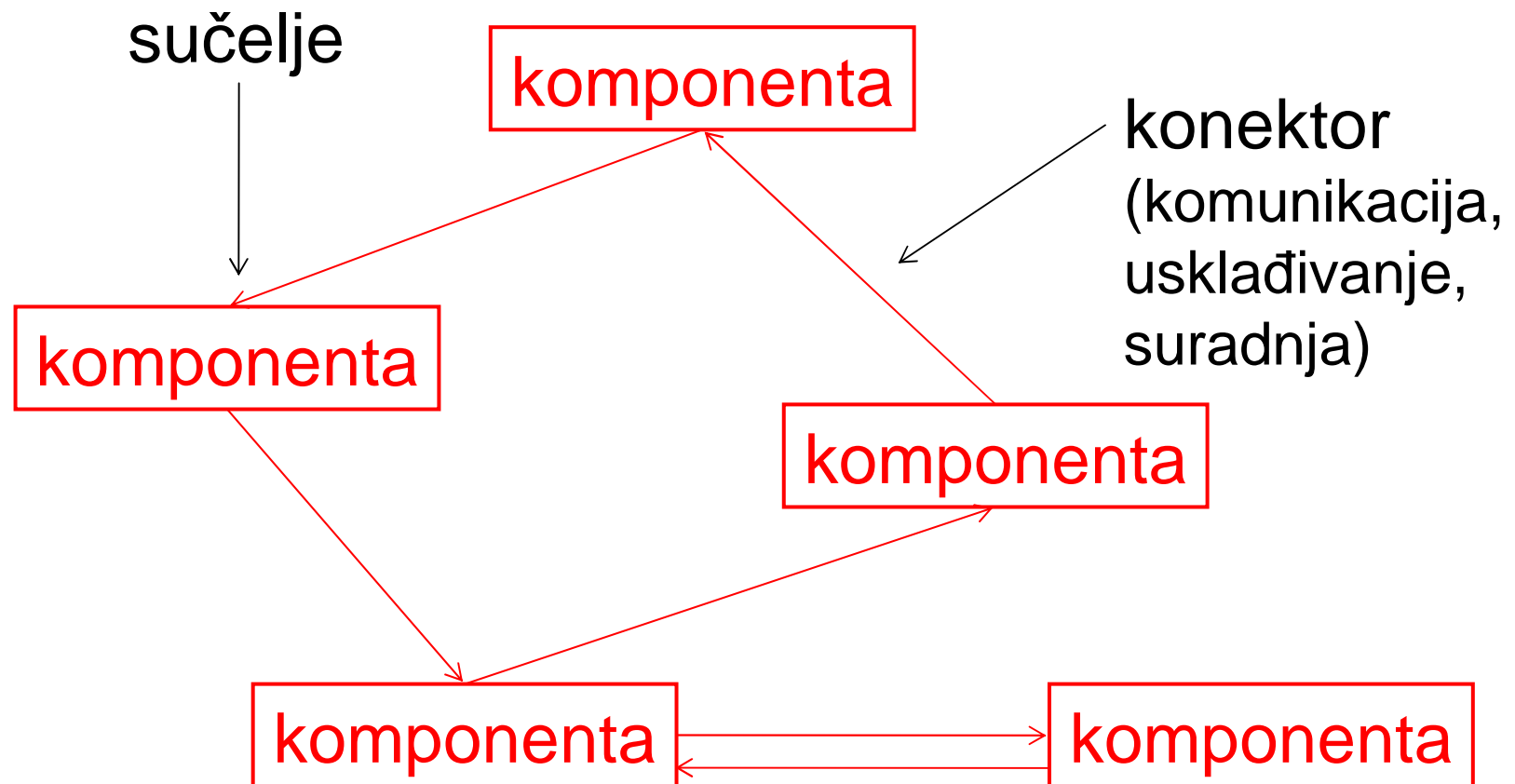
Objektni modeli

- ◆ objekti: komponente sustava s dobro definiranim sučeljem
- ◆ mehanizam komunikacije, usklađivanja i suradnje komponenata

Jezik: UML



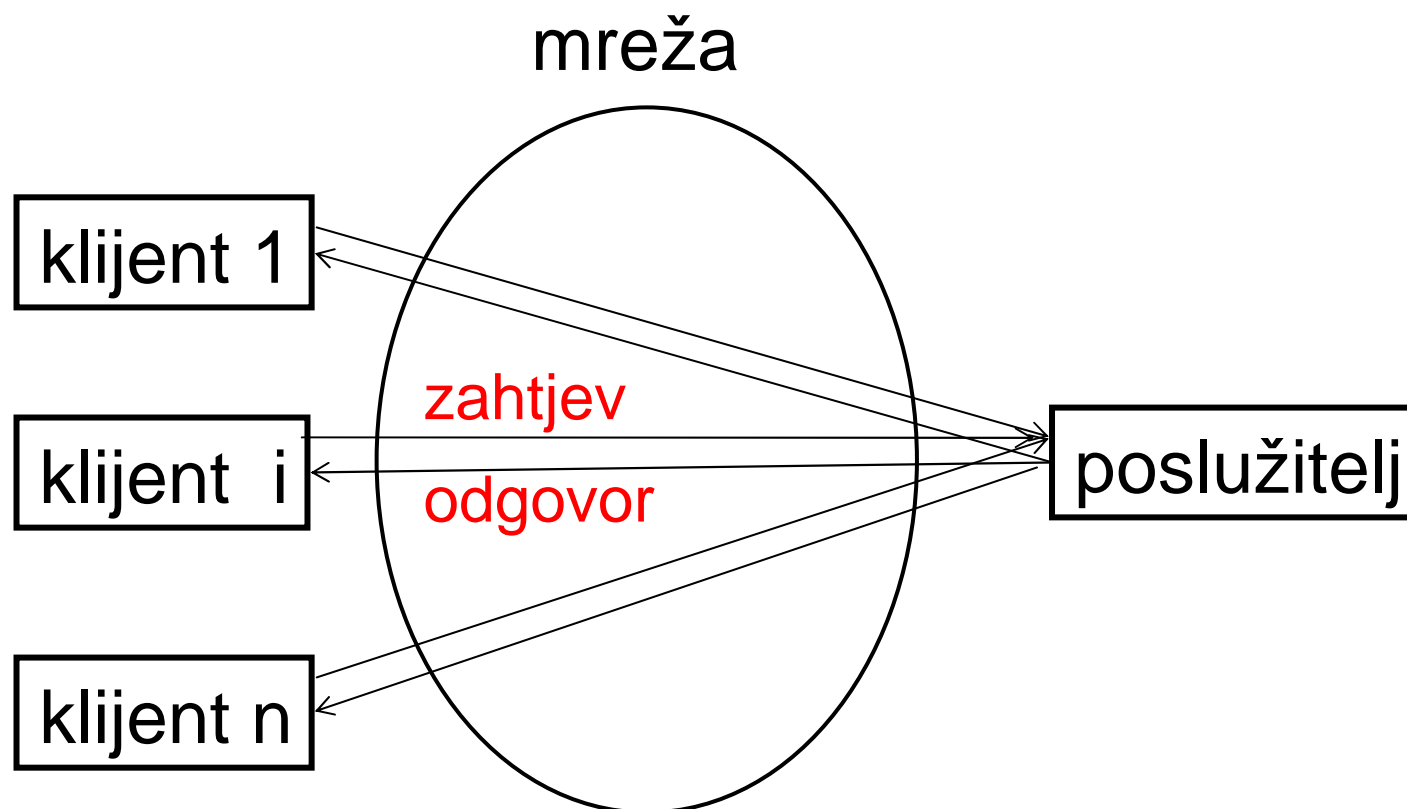




Model klijent – poslužitelj (1)



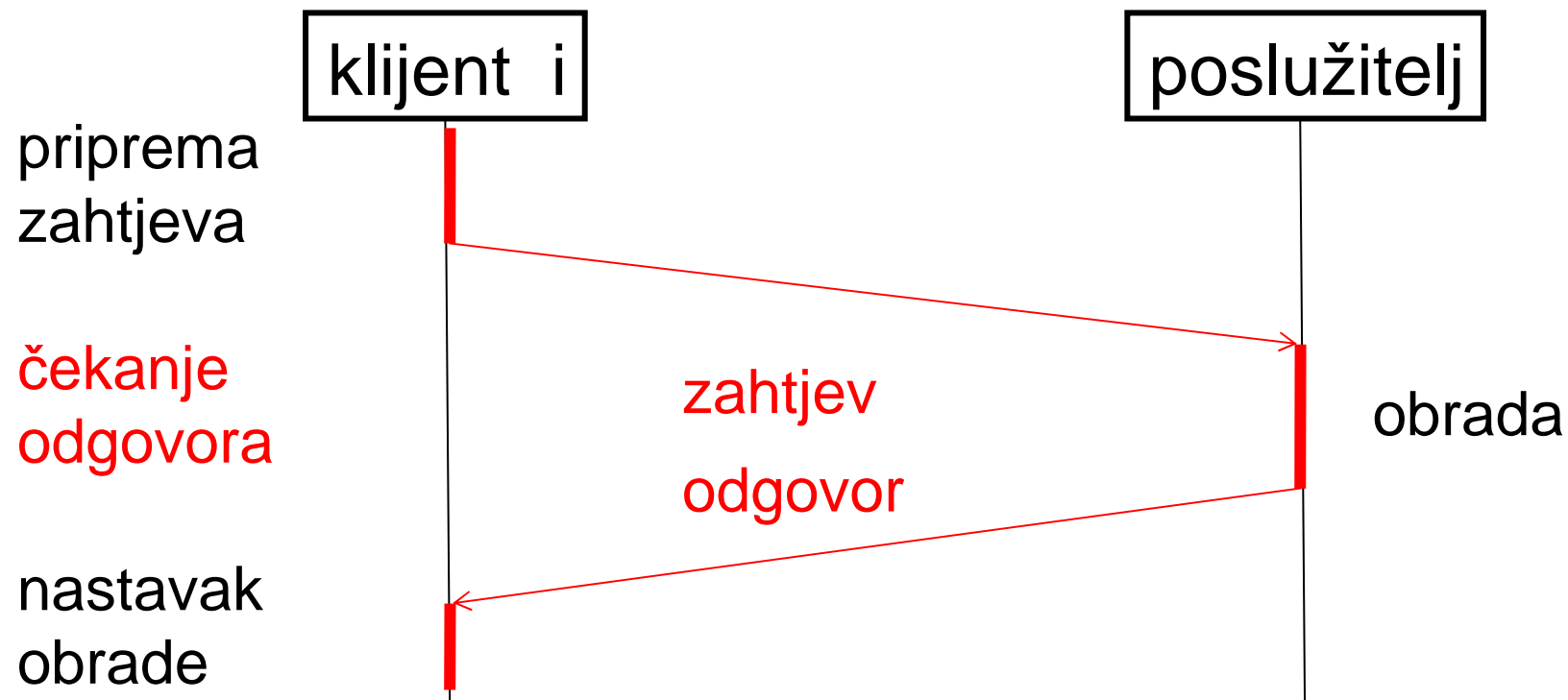
- ◆ Klijent: traži uslugu (zahtjev)
- ◆ Poslužitelj: pruža uslugu (odgovor) za više/mnogo klijenta



Model klijent – poslužitelj (2)



- ◆ Klijent šalje zahtjev i čeka odgovor
- ◆ Poslužitelj: prihvaća i obrađuje zahtjev te vraća odgovor



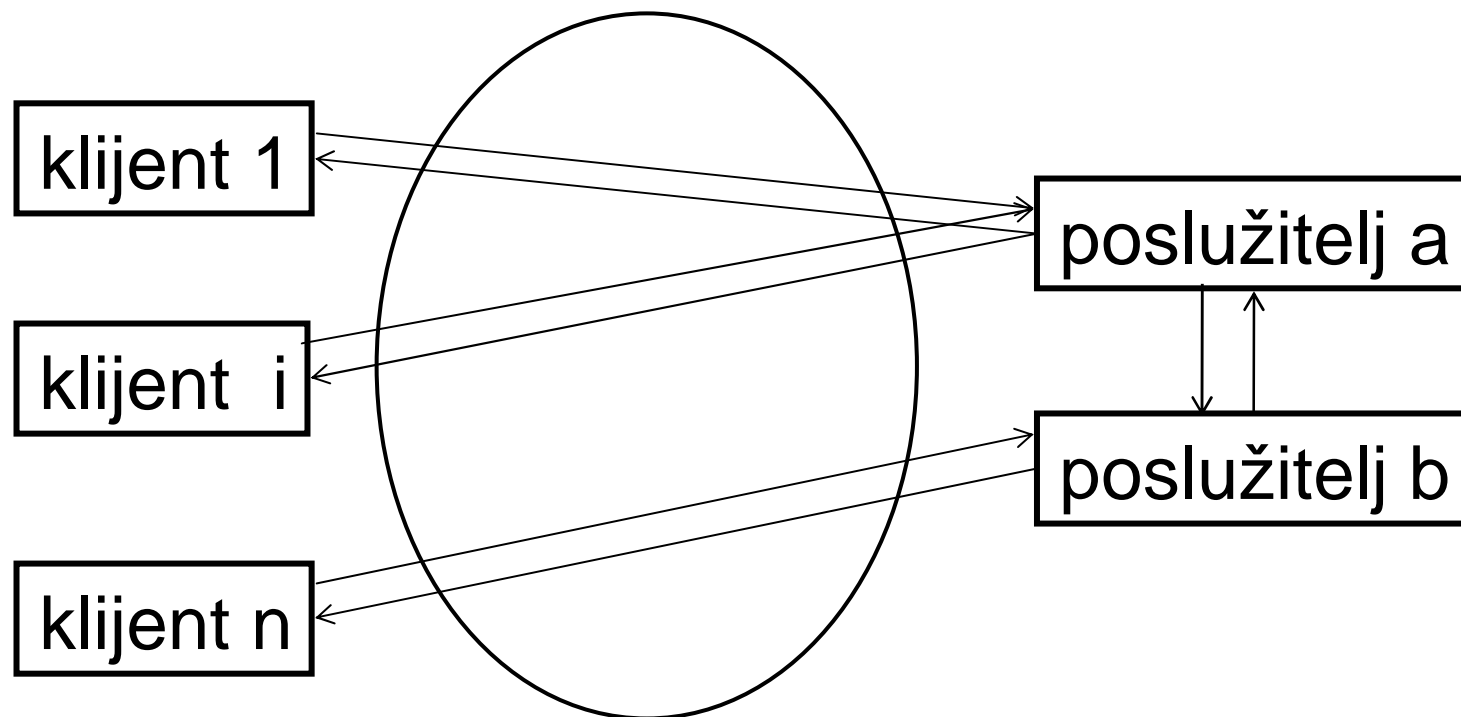
Model s više poslužitelja (1)



Istovrsni poslužitelji:

- posluživanje većeg/velikog broja klijenata
- prebacivanje zahtjeva na manje opterećene poslužitelje

mreža



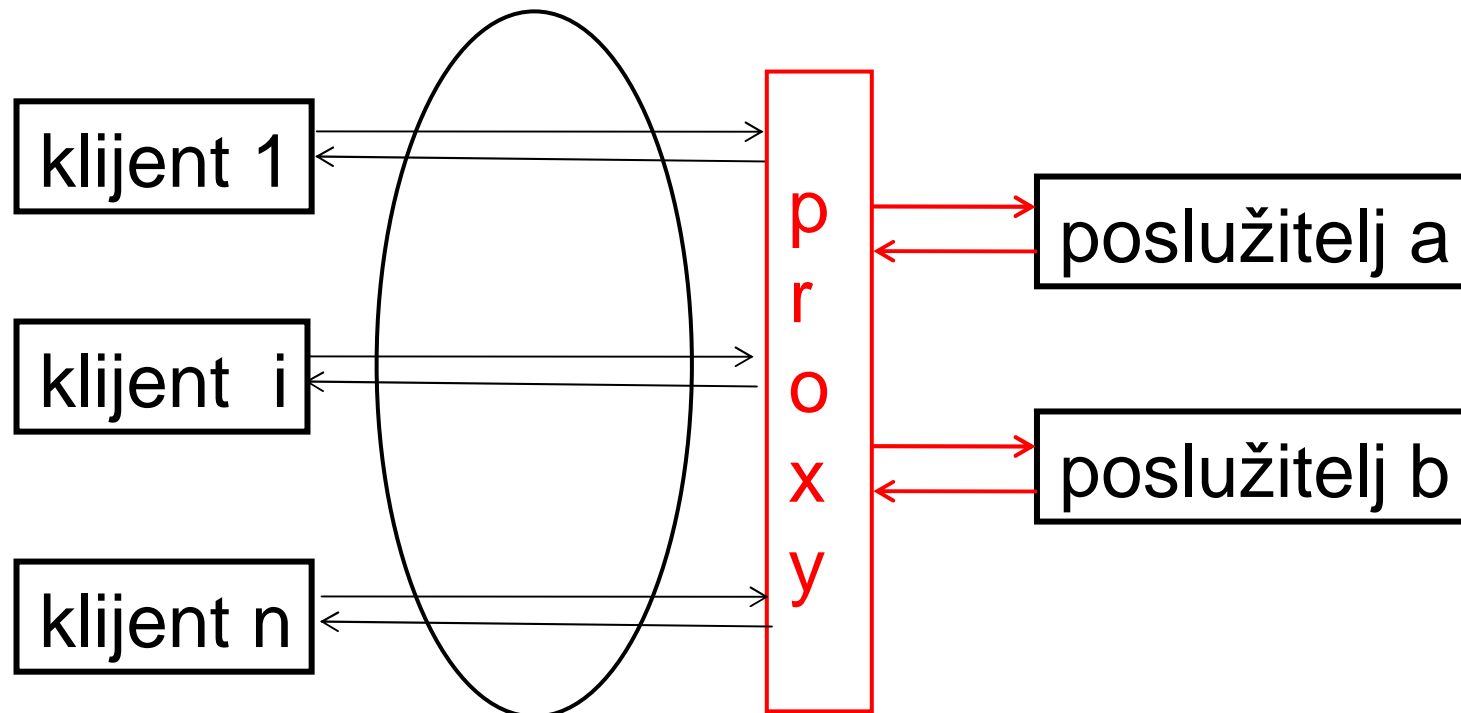
Model s više poslužitelja (2)



Zastupnik (engl. *proxy*) poslužitelja

- posredovanje: replikacijska transparentnost
- dodatna mogućnost: priručna pohrana podataka (engl. *cache*)

mreža



Programska izvedba sustava klijent – poslužitelj obuhvaća tri razine:

- ◆ **korisničko sučelje**, npr. kod weba: unos zahtjeva, pregled stranice odgovora

- ◆ **primjenski program**, npr. kod weba: pretraživač koji na temelju zahtjeva generira upite bazi podataka, preuzima stranice i predodčuje ih korisniku

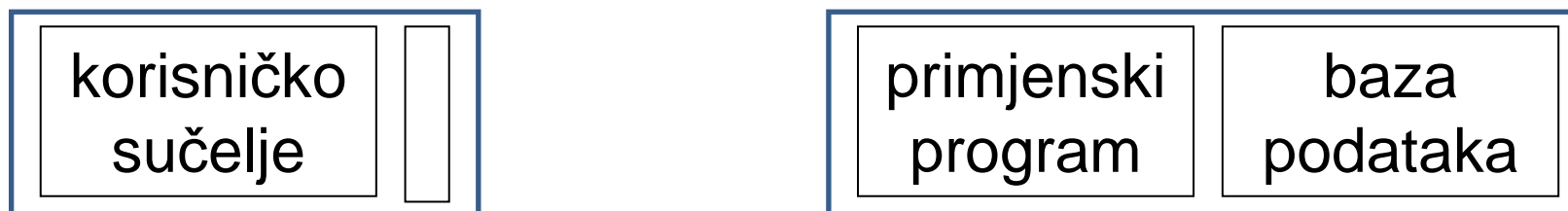
- ◆ **baza podataka**, npr. kod weba (indeksirane) stranice

Korisničko sučelje, primjenski program i baza podataka mogu biti na različite načine raspodijeljeni između klijenta i poslužitelja.

Dvoredna i višeredna arhitektura



Dvoredna arhitektura klijent – poslužitelj (engl. *two-tier*)



mršavi klijent (engl. *thin client*)

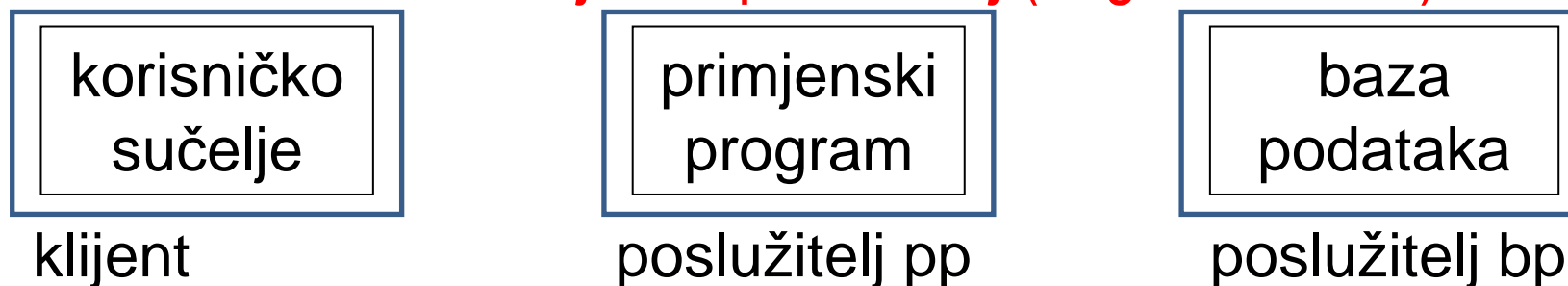
poslužitelj



debeli klijent (engl. *fat client*)

poslužitelj

Troredna arhitektura klijent – poslužitelj (engl. *three-tier*)



Model objavi-pretplati (1)

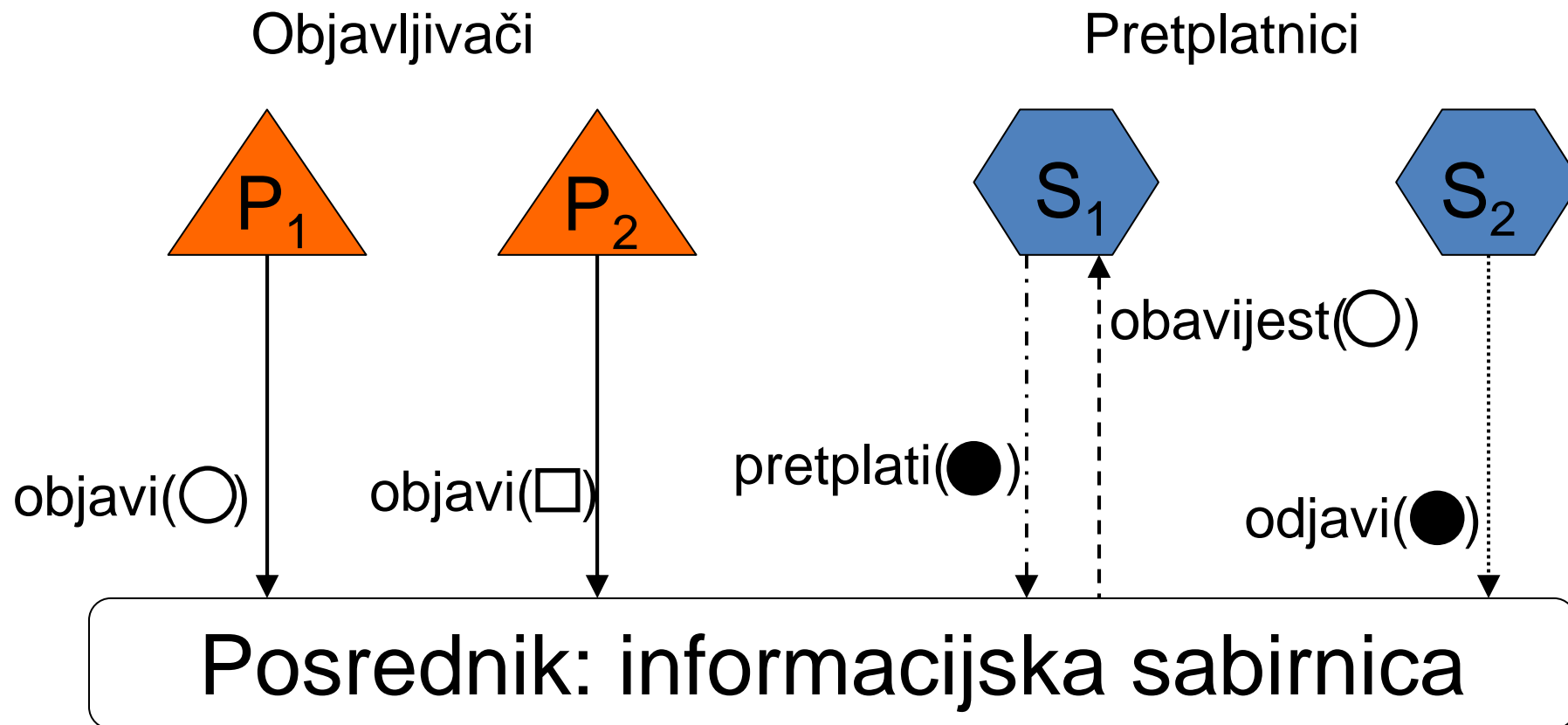


- ◆ objekt-izvor: objavljiivač obavijesti (engl. *publisher*)
- ◆ objekt-odredište: pretplatnik na obavijest (engl. *subscriber*)
- ◆ posrednik između objekta-izvora i objekta-odredišta:
 - informacijska sabirnica koja povezuje objavljiivače obavijesti s pretplatnicima

Analogija:

- ◆ novinske agencije (Reuters, HINA, ...)
- ◆ usluge za isporuku sadržaja (sličan interakcijski model, na drugoj razini apstrakcije)

Model objavi-pretplati (2)



◆ Arhitektura:

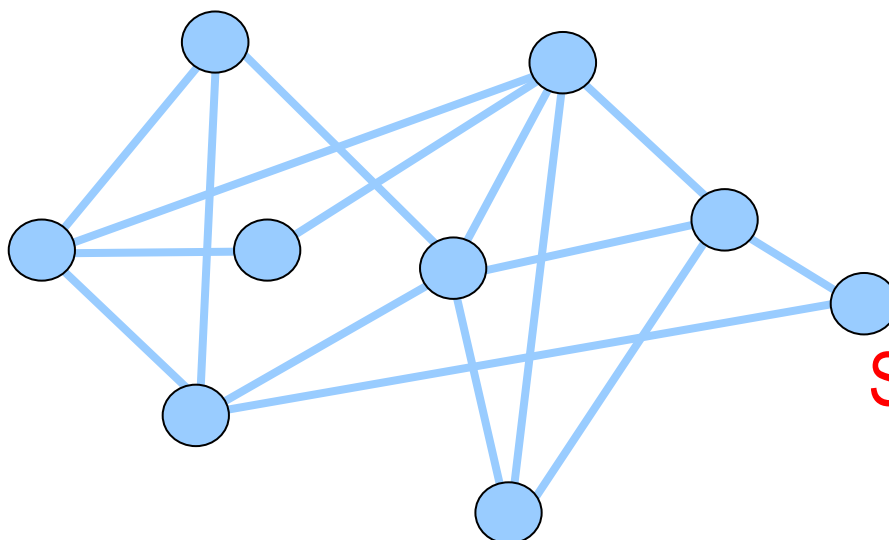
- centralizirana: jedan posrednik
- raspodijeljena: više posrednika

◆ Vrste pretplata:

- na kanal: tematsko grupiranje obavijesti (npr. vrijeme)
- na sadržaj: pretplata se definira ovisno o svojstvima i sadržaju obavijesti (npr. knjiga nekoga autora, po nekoj cijeni)

- ◆ ravnopravni sudionik (engl. *peer*) je onaj koji istodobno obavlja funkciju poslužitelja i klijenta
- ◆ ravnopravni sudionici međusobno komuniciraju (engl. *Peer-to-Peer*, P2P) tako da se povezuju u mrežu na aplikacijskom sloju koja predstavlja “prekrivajuću mrežu” (engl. *overlay network*) nad stvarnom mrežnom topologijom
- ◆ svaki čvor “plaća” sudjelovanje u mreži nudeći dio vlastitih sredstava ostalim čvorovima
- ◆ model ravnopravnih sudionika potencijalno nudi neograničena sredstva u velikim mrežama s puno čvorova

Decentralizirani distribuirani sustav



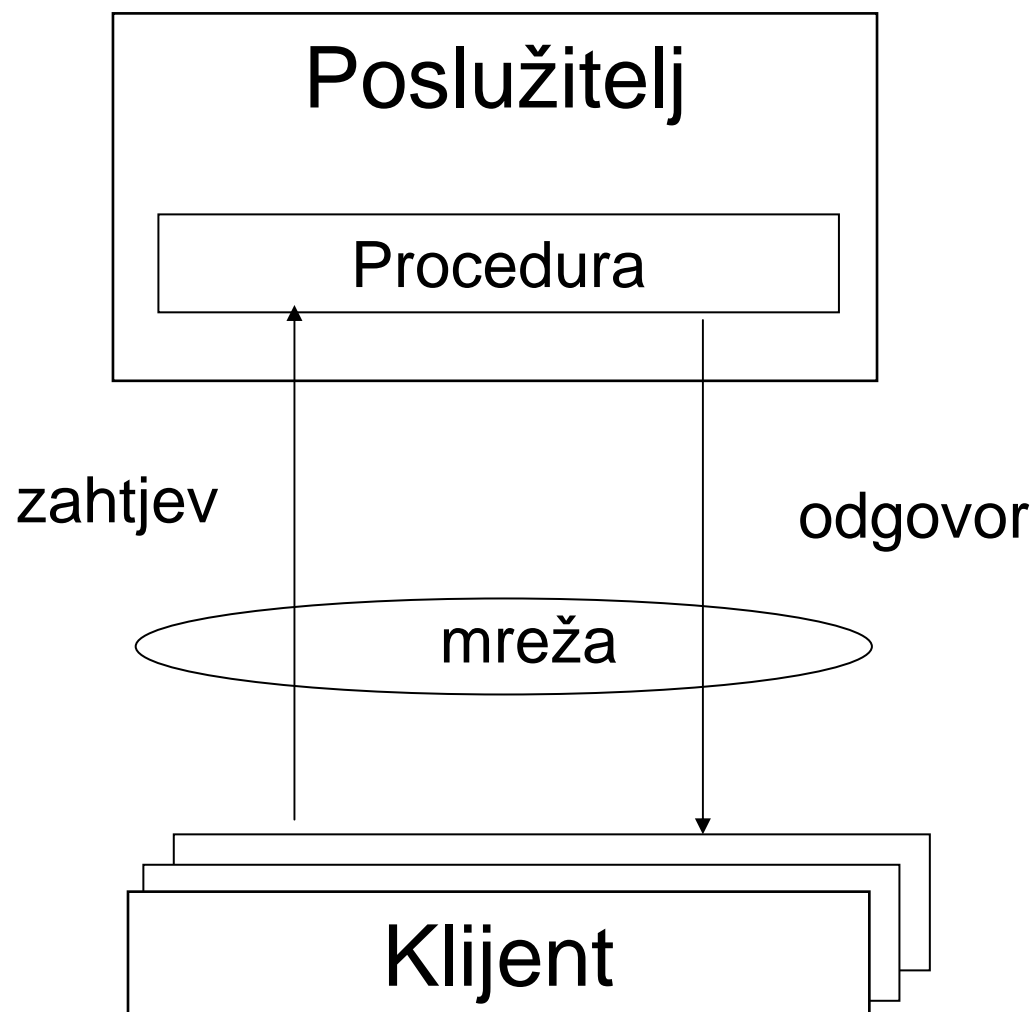
- nema centralizirane koordinacije među *peerovima*
- ne postoji jedna točka ispada

Samoorganizirajuća mreža čvorova

- *peerovi* su međusobno neovisni

- ◆ **Migracija programa** (engl. *code migration*): razmjena programa između umreženih čvorova;
 - Migracija procesa s jednog računala na drugo zbog uravnoteženja opterećenja, (uvođenja) konkurentnosti, (proširenja) funkcionalnosti,
- ◆ **Pokretni agent** (engl. *mobile agent*) je program koji predočuje korisnika u mreži i za njega obavlja neki posao krećući se samostalno između čvorova u mreži;
 - Umjesto razmjene upita i odgovora između klijenta i poslužitelja, pokretni s klijenta migrira na poslužitelja, provodi obradu i vraća se s rezultatom;
 - Mogu se kombinirati različita svojstva agenata (pokretljivost, inteligencija, autonomnost, reaktivnost, proaktivnost, ...);

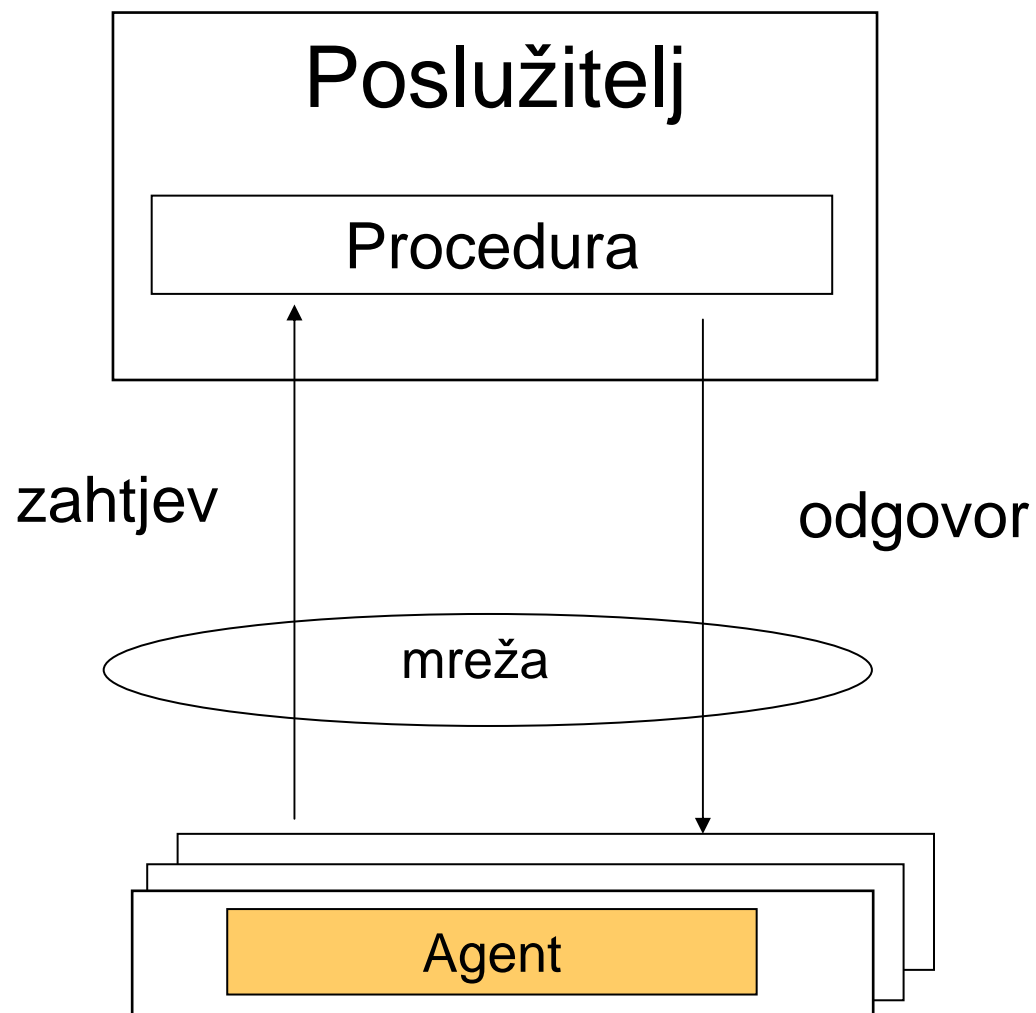
Od modela klijent-poslužitelj ...

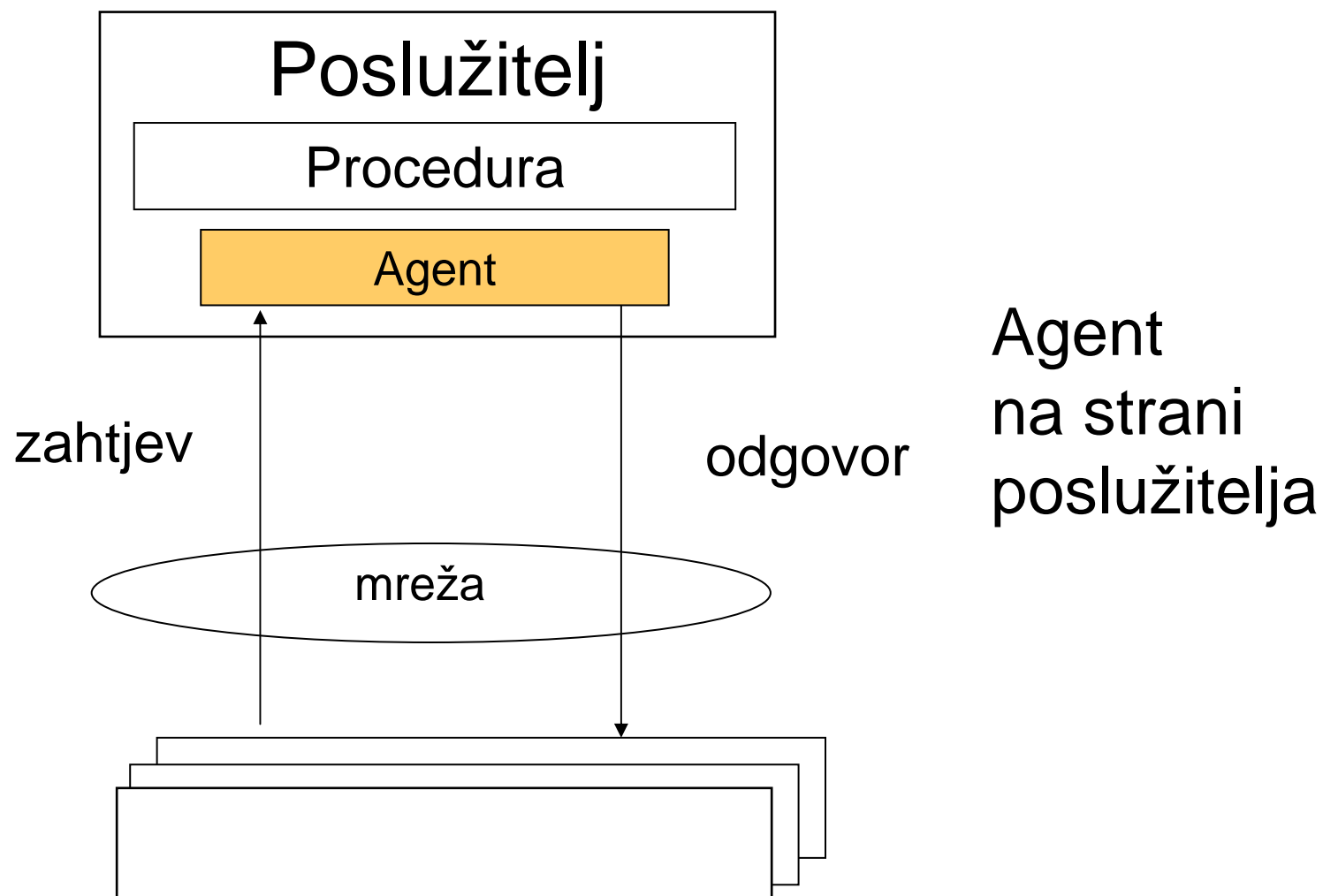


Programski model:

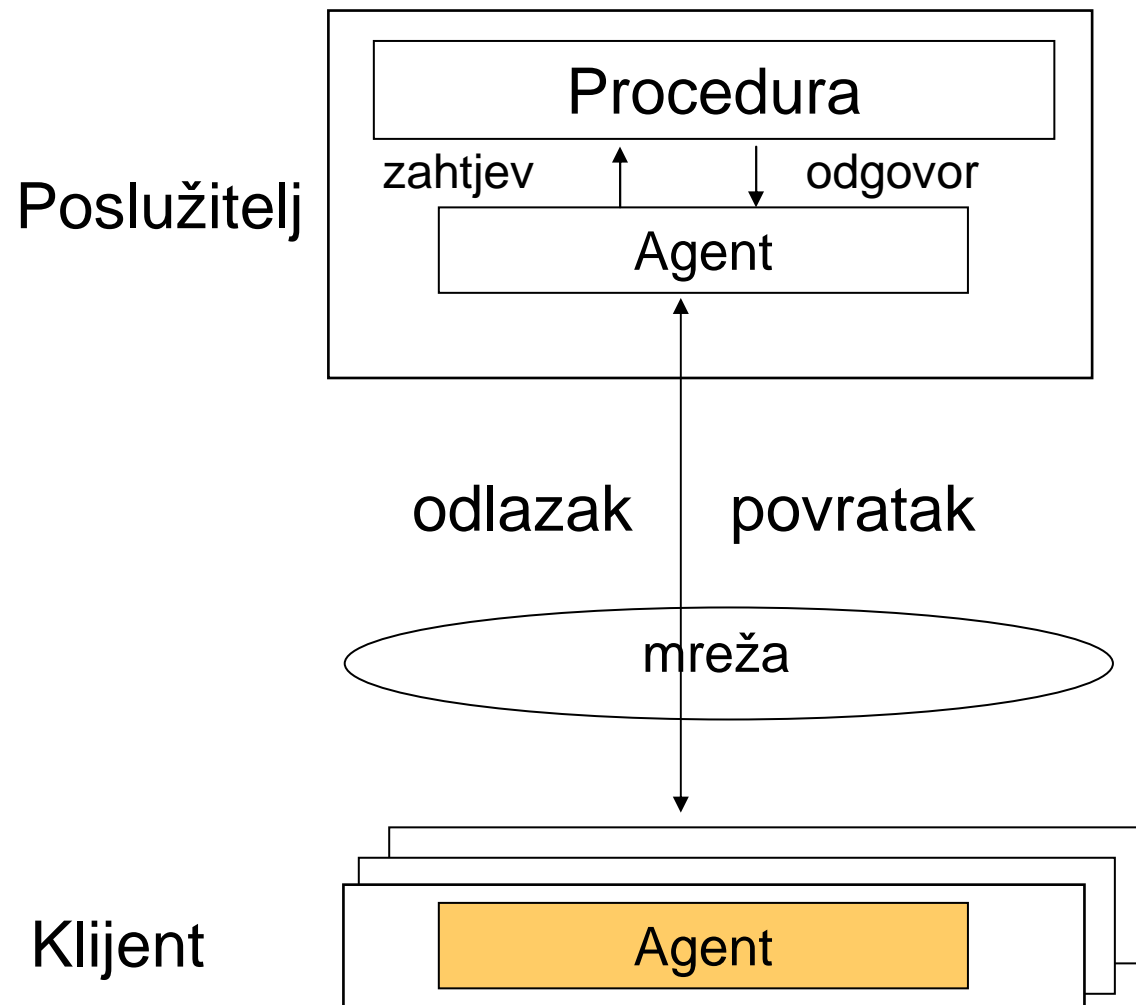
Poziv udaljene procedure
(RPC – *Remote Procedure Call*)

... preko stacionarnog programskog agenta ...





... do pokretnog programskog agenta



Programski model:

Udaljeno programiranje
(RP – *Remote Programming*)

- ◆ Svi arhitekturni modeli sastoje se od procesa koji međusobno komuniciraju razmjenom poruka

- ◆ Temeljni formalizmi:
 - Interakcijski model: procesi, komunikacija, vremenska usklađenost odvijanja i komunikacije procesa
 - Model kvara: kvarovi i njihov utjecaj na odvijanje i komunikaciju procesa
 - Model sigurnosti: prijetnji odvijanju i komunikaciji procesa te mjere zaštite