



Preddiplomski studij

Računarstvo

Modul:

**Telekomunikacije i
informatika**

Višemedijske usluge

**Distribuirani sustavi i
modeli distribuiranog procesiranja**

Ak.god. 2007./2008.

03.04.2008.

- ◆ Distribuirani sustav
 - definicija i svojstva
 - međuoprema
 - karakteristike komunikacijske međuopreme
- ◆ Komunikacijska međuoprema
 - komunikacija korištenjem priključnica (*socket*)
 - poziv udaljene procedure/metode
 - komunikacija porukama
 - model objavi-pretplati

Definicija

- ◆ Distribuirani (raspodijeljeni) sustav je **skup računala povezanih mrežom** koji djeluje kao jedinstveni sustava te krajnjem korisniku pruža definiranu uslugu
- ◆ Sastoji se od:
 - sklopovlja = (autonomna) računala + mreža
 - programske opreme (omogućuje razvoj jedinstvenog distribuiranog sustava)
- ◆ Sa stajališta korisnika riječ je o **jedinstvenom sustavu**

Zahtjevi i dobre karakteristike

- ◆ Otvorenost
- ◆ Skalabilnost
- ◆ Transparentnost
- ◆ Otpornost na neispravnosti i ispade
- ◆ Pouzdanost
- ◆ Raspoloživost

- ◆ Za distribuirani sustav se kaže da je otvoren ako nudi usluge slijedeći standardna pravila koja opisuju sintaksu i semantiku tih usluga
- ◆ Usluga se obično definira pomoću sučelja
- ◆ Primjeri standardnih jezika za opis sučelja
 - Interface Definition Language (IDL) – CORBA
 - Web Services Description Language (WSDL)

- ◆ Sposobnost sustava da se prilagodi i zadovolji zahtjeve povećanog broja korisnika
- ◆ Problemi neskalkabilnih rješenja
 - centralizirana usluga, podaci, algoritmi
- ◆ Rješenje neskalkabilnosti
 - raspodijeljenost usluge, podataka, algoritama
 - primjeri skalabilnih sustava: DNS, WWW
 - model s ravnopravnim procesima (engl. *peer-to-peer*)

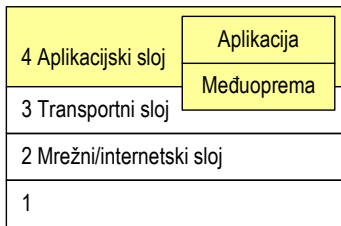
- ◆ Osnovni cilj
 - sakriti činjenicu da su procesi i resursi fizički raspodijeljeni na više računala
 - npr. distribuirana baza podataka ponaša se kao i centralizirana
 - klijent ne mora biti svjestan da je izvedba sustava distribuirana (doživljava ga kao jedinstveni sustav)

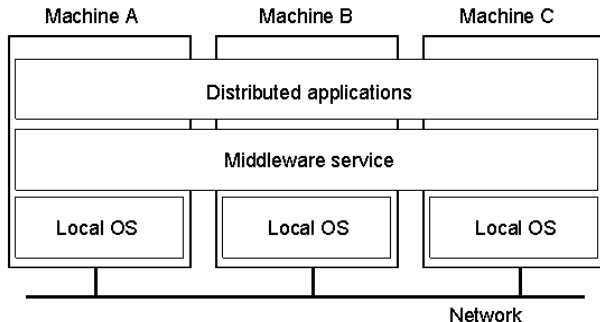
- ◆ postiže se dodavanjem posebnog aplikacijskog sloja koji nudi usluge višim slojevima i olakšava razvoj distribuiranih sustava, a naziva se **programska međuoprema** (engl. *middleware*)

- ◆ Lokacija
 - prikrivanje stvarne lokacije resursa
 - npr. naziv mrežne usluge neovisan je lokaciji resursa koji realizira uslugu
- ◆ Migracija
 - promjena lokacije resursa se ne odražava na krajnjeg korisnika
- ◆ Replikacija
 - skrivanje činjenice da postoji više kopija određenog resursa

- ◆ Distribuirani sustav
 - definicija i svojstva
 - **međuprema**
 - karakteristike komunikacijske međupreme
- ◆ Komunikacijska međuprema
 - komunikacija korištenjem priključnica (*socket*)
 - poziv udaljene procedure/metode
 - komunikacija porukama
 - model objavi-pretplati

- ◆ Distribuirani sustavi koriste **međuopremu** (*middleware*), programsku infrastrukturu koja pruža **generičke usluge za jednostavniji razvoj distribuiranih aplikacija**
- ◆ U internetskom modelu međuoprema je smještena na aplikacijskom sloju između transportnog sloja i aplikacije





Različite vrste
međupreme

- ♦ nude usluge, različite kompleksnosti
- ♦ primjer: komunikacijska međuprema

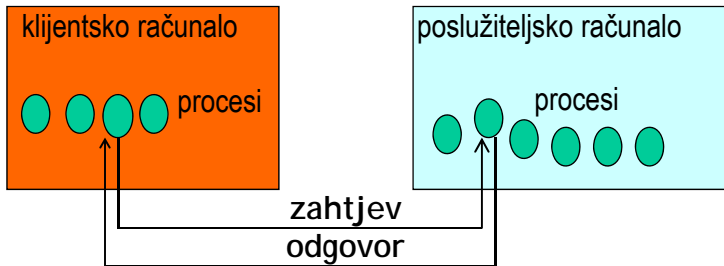
- ♦ vrsta međuopreme za realizaciju komunikacije među udaljenim računalima
- ♦ protokoli za komunikaciju distribuiranih procesa na višem nivou apstrakcije od transportnog sloja
- ♦ omogućuje jednostavniji razvoj distribuiranih aplikacija, sakriva kompleksnost i heterogenost nižih slojeva

- ◆ Postojeća rješenja za komunikaciju distribuiranih procesa
 - komunikacija korištenjem priključnica (*socket*)
 - poziv udaljene procedure (*remote procedure call*, RPC)
 - distribuirani objekti - poziv udaljene metode (*remote method invocation*, RMI)
 - komunikacija razmjenom poruka (*message-oriented interaction*)
 - model objavi-pretplati (*publish/subscribe*)

Osnovni model komunikacije: klijent-poslužitelj



Zavod za telekomunikacije



◆ KLIJENT

- ◆ zahtjeva uslugu
- ◆ šalje zahtjev poslužitelju i čeka odgovor

◆ POSLUŽITELJ

- ◆ nudi usluge
- ◆ prima i obrađuje dolazne zahtjeve te šalje odgovor klijentima

- ◆ Distribuirani sustav
 - definicija i svojstva
 - međuoprema
 - **karakteristike komunikacijske međuopreme**
- ◆ Komunikacijska međuoprema
 - komunikacija korištenjem priključnica (*socket*)
 - poziv udaljene procedure/metode
 - komunikacija porukama
 - model objavi-pretplati

♦ vremenska (ne)ovisnost

- vremenski ovisni procesi moraju biti istovremeno dostupni za realizaciju komunikacije
- vremenski neovisni procesi mogu komunicirati i ako nisu istovremeno dostupni

♦ ovisnost o referenci “sugovornika”

- proces je ovisan o referenci “sugovornika” ako mora znati jedinstveni identifikator udaljenog procesa s kojim želi komunicirati
- proces može biti i neovisan o referenci, tj. ne mora znati jedinstveni identifikator udaljenog procesa

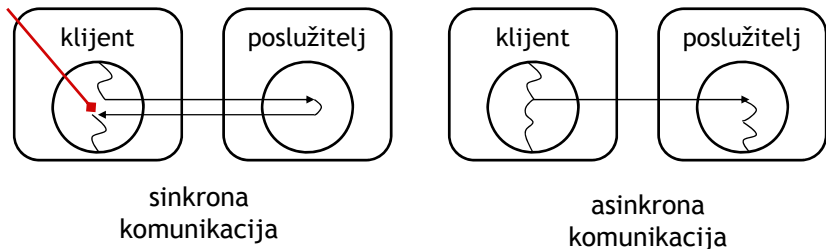
◆ perzistentnost komunikacije

- perzistentna komunikacija garantira isporuku poruke, poruka se pohranjuje u sustavu i isporučuje na odredište kada ono postane dostupno
- neperzistentna komunikacija je nepouzdana, garantira isporuku poruke samo ako su pošiljalatelj i primatelj istovremeno dostupni

♦ sinkrona ili asinkrona komunikacija

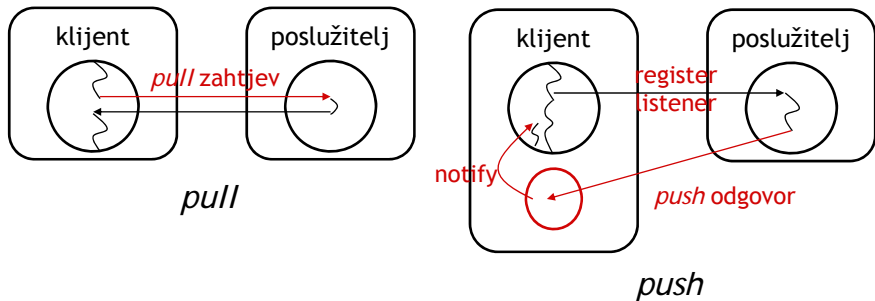
- sinkrona komunikacija blokira klijenta dok ne dobije odgovor od strane poslužitelja (konekcijska)
- asinkrona komunikacija omogućuje klijentu nastavak procesiranja odmah nakon slanja zahtjeva (beskonekcijska)

blokiranje



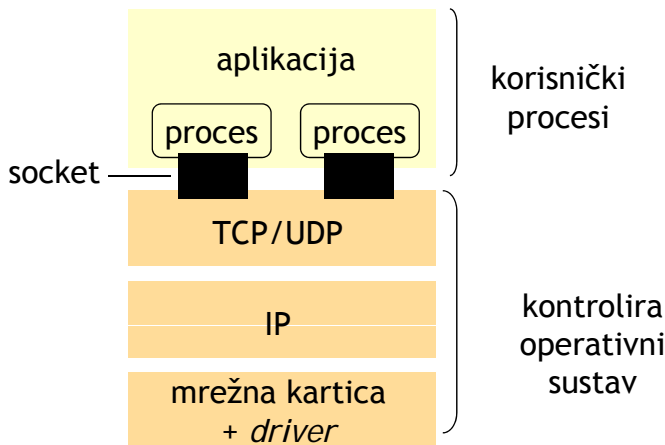
♦ *pull* ili *push* pokretanje komunikacije

- *pull* – klijent eksplicitno šalje zahtjev poslužitelju
- *push* – poslužitelj šalje podatke, a klijent “sluša”



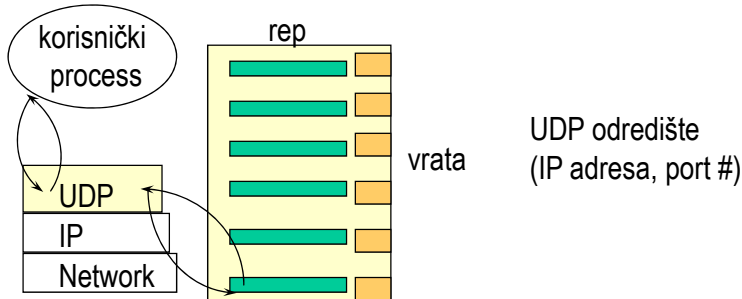
- ◆ Distribuirani sustav
 - definicija i svojstva
 - međuoprema
 - karakteristike komunikacijske međuopreme
- ◆ Komunikacijska međuoprema
 - komunikacija korištenjem priključnica (*socket*)
 - poziv udaljene procedure/metode
 - komunikacija porukama
 - model objavi-pretplati

- ◆ jednostavno korištenje funkcionalnosti transportnog sloja
 - TCP – konekcijski protokol, pouzdan prijenos podataka
 - UDP – prijenos nezavisnih paketa (*datagrami*), nepouzdan prijenos
- ◆ *Socket*
 - komunikacijska točka preko koje aplikacija šalje podatke u mrežu i iz koje čita primljene podatke
 - viši nivo apstrakcije nad komunikacijskom točkom koju operativni sustav koristi za pristup transportnom sloju

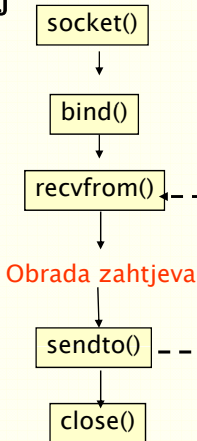


User Datagram Protocol (UDP)

- komunikacija se odvija preko vrata (engl. *portova*) koje dodjeljuje operativni sustav



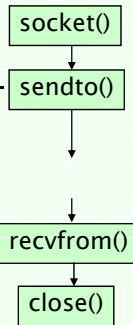
POSLUŽITELJ



podaci (zahtjev)

podaci (odgovor)

KLIJENT



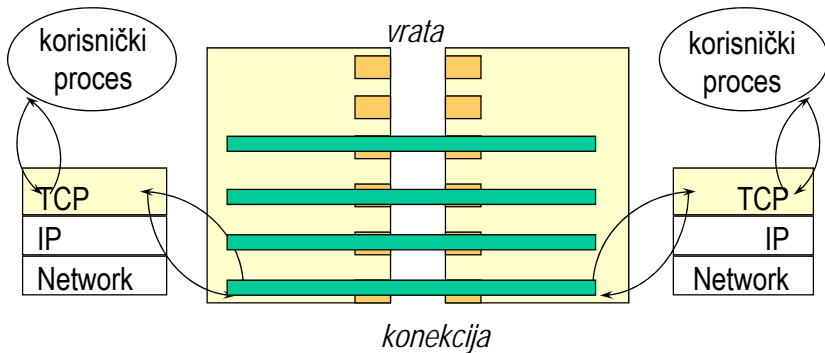
Funkcija	Značenje
socket	Kreiranje nove komunikacijske točke (<i>socket</i>)
bind	Povezivanje transportne adrese* i <i>socket</i> -a
recvfrom	Primanje datagrama (zahtjeva ili odgovora)
sendto	Slanje datagrama (zahtjeva ili odgovora)
close	Zatvaranje konekcije

*Transportna adresa = (IP adresa, port)

- ◆ vremenska ovisnost
 - poslužitelj mora biti aktivan za primanje datagrama
- ◆ klijent mora znati identifikator poslužitelja
- ◆ komunikacija nije perzistentna
- ◆ asinkrona komunikacija
 - klijent šalje datagram i nastavlja procesiranje

Transmission Control Protocol (TCP)

- konekcija između dvije krajnje točke koje se moraju dogovoriti o uspostavi konekcije



Konekcijska komunikacija pomoću TCP *socket*a (*Socket API*)



Zavod za telekomunikacije

POSLUŽITELJ

socket()

bind()

listen()

accept()

blokiran čeka
klijentski zahtjev

accept() stvara novi
socket za prihvati zahtjeva

read()

write()

close()

obrada zahtjeva

uspostava
asocijacije

podaci (zahtjev)

podaci (odgovor)

KLIJENT

socket()

connect()

write()

read()

close()

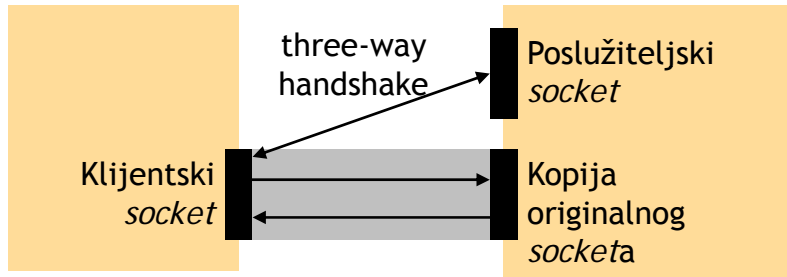
Funkcija	Značenje
socket	Kreiranje nove komunikacijske točke (<i>socket</i>)
bind	Povezivanje transportne adrese sa <i>socket</i> -om
listen	Najava spremnosti za komunikaciju
accept	Poslužitelj prima zahtjev za inicijalizaciju konekcije
connect	Pokušaj uspostave konekcije
send	Slanje podataka
receive	Primanje podataka
close	Zatvaranje konekcije

POSLUŽITELJ

- ♦ **socket** – kreira komunikacijsku točku, operativni sustav rezervira resurse koji će omogućiti slanje i primanje podataka koristeći odabrani transportni protokol
- ♦ **bind** – povezuje adresu sa socketom. Poslužitelj povezuje IP adresu računala i broj *porta* sa socketom.
- ♦ **listen** – omogućuje operativnom sustavu rezerviranje resursa (spremnika) za specificirani maksimalni broj konekcija.
- ♦ **accept** – poslužitelj prima zahtjev za iniciranje konekcije od strane klijenta (connect). Poslužitelj stvara novi identičan socket koji se koristi za komunikaciju s klijentom. Originalni socket se koristi za “oslušivanje” novih zahtjeva.
- ♦ **read i write** – slanje i primanje podataka

KLIJENT

- ♦ **socket** - kreira komunikacijsku točku, bind nije potreban jer OS dinamički alokira port socketu pri kreiranju konekcije.
- ♦ **connect** – klijent šalje zahtjev za kreiranje konekcije. Klijent mora definirati transportnu adresu na koju se šalje zahtjev za kreiranje konekcije. Klijent je blokiran do uspostave konekcije.
- ♦ **read i write** – slanje i primanje podataka
- ♦ **close** – zatvaranje konekcije



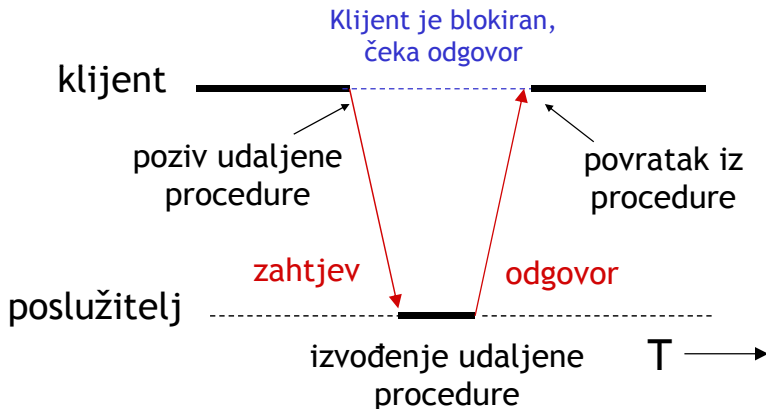
- ◆ za svaki novi korisnički zahtjev kreira se novi socket koji je kopija originalnog
- ◆ originalni poslužiteljski socket mora konstantno biti u stanju "oslušivanja"

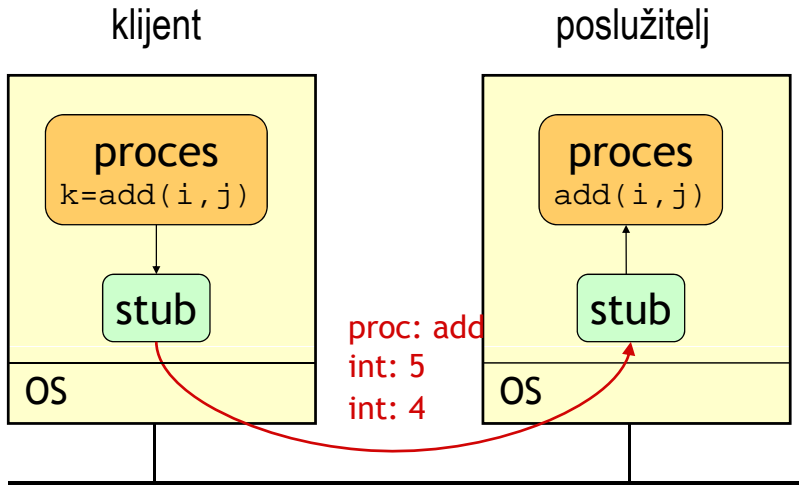
- ◆ vremenska ovisnost
 - klijent i poslužitelj moraju biti istovremeno dostupni
- ◆ klijent mora znati identifikator poslužitelja
- ◆ komunikacija nije perzistentna
- ◆ sinkrona komunikacija
 - klijent šalje zahtjev za kreiranje konekcije i blokiran je do uspostave konekcije
- ◆ *pull* pokretanje komunikacije

- ◆ Distribuirani sustav
 - definicija i svojstva
 - međuoprema
 - karakteristike komunikacijske međuopreme
- ◆ Komunikacijska međuoprema
 - komunikacija korištenjem priključnica (*socket*)
 - poziv udaljene procedure/metode
 - komunikacija porukama
 - model objavi-pretplati

Remote Procedure Call (RPC)

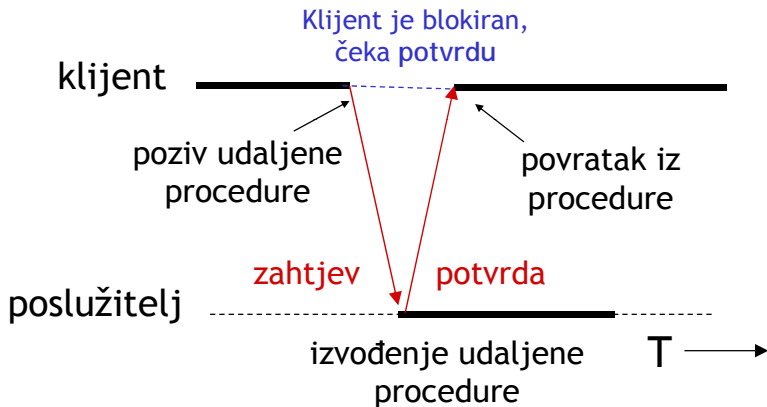
- ◆ Omogućiti procesima pozivanje i izvođenje procedura na udaljenom računalu
- ◆ Proces na računalu A poziva proceduru koja se izvodi na računalu B.
- ◆ Pozivajući proces na računalu A šalje parametre za izvođenje procedure na računalu B i blokiran je čekajući rezultate izvođenja procedure.
- ◆ Računalu B izvodi proceduru koristeći primljene parametre i šalje odgovor računalu A.
- ◆ **Transparentnost**: za proces na računalu A poziv udaljene procedure jednak je pozivu lokalne procedure.



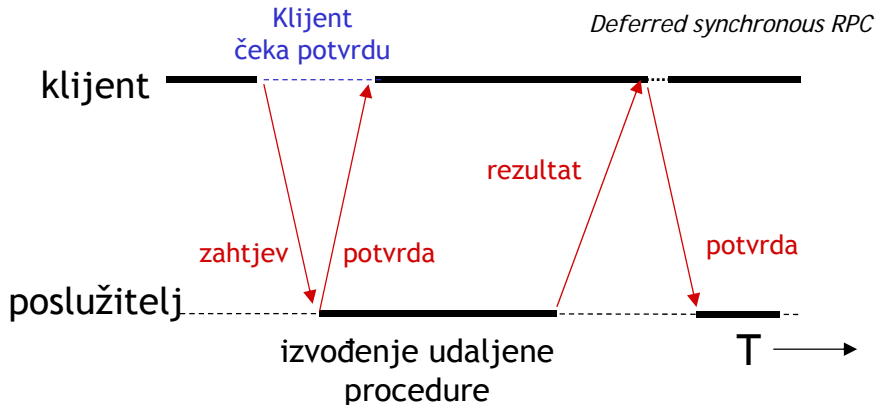


1. Klijent poziva proceduru `add(i, j)` koristeći *stub*.
2. *Stub* "pakira" parametre i identifikator procedure i poziva OS.
3. OS klijentskog računala šalje poruku na udaljeno računalo.
4. OS udaljenog računala predaje poruku *stubu* poslužitelja.
5. *Stub* poslužitelja "raspakira" parametre i poziva proceduru `add(i, j)` koristeći primljene parametre.
6. Procedura vraća rezultat izvođenja poslužiteljskom *stub*-u.
7. *Stub* poslužitelja "pakira" rezultat u poruku i poziva OS.
8. OS poslužitelja šalje poruku OS-u klijenta.
9. OS klijenta predaje poruku *stubu*.
10. *Stub* "raspakira" rezultat i predaje ga klijentskom procesu.

- ◆ *Marshaling* – “pakiranje” parametara ili rezultata u poruku
- ◆ *Unmarshaling* – čitanje parametara ili rezultata iz poruke
- ◆ Prenošenje vrijednosti parametra
 - navodi se tip (npr. int, char, long) i vrijednost
 - različiti OS koriste različite prikaze znakova
- ◆ Prenošenje parametara koristeći reference
 - referenca ima smisla samo u adresnom prostoru procesa koji je koristi!
 - Kako prenijeti string na udaljeno računalo?
 - nije moguće koristiti referencu na string!
 - kopiranje cijelog stringa i “pakiranje” u poruku



- ◆ Klijent šalje zahtjev poslužitelju, ali ne očekuje odgovor, tj. rezultat izvođenja procedure, već samo potvrdu o primitku zahtjeva



- ♦ klijent šalje zahtjev poslužitelju i nastavlja s procesiranjem nakon što primi potvrdu.
- ♦ rezultat izvođenja procedure poslužitelj šalje koristeći drugi asinkroni RPC.

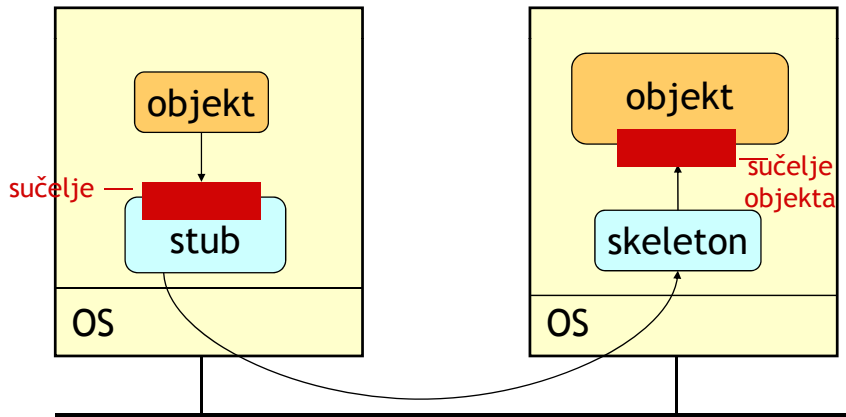
Remote Method Invocation (RMI)

- ◆ “nasljednik” poziva udaljene procedure, poziva se metoda udaljenog objekta
- ◆ distribuirani objekt
 - proširenje osnovnog objektnog modela na distribuirane objekte
 - odvajanje sučelja i implementacije objekta
- ◆ objekt (klijent) poziva metodu udaljenog objekta (poslužitelja) na transparentan način
 - identično pozivu metode lokalnog objekta

- ◆ Postoje reference na lokalne i udaljene objekte
- ◆ Svaki udaljeni objekt ima globalno jedinstven identifikator
 - npr. [ref: [endpoint:[161.53.19.24:1251](local),objID:[0]]]]
- ◆ Potrebna je usluga za registriranje i pronalaženje udaljenih objekata (*directory service*)

kljent

poslužitelj

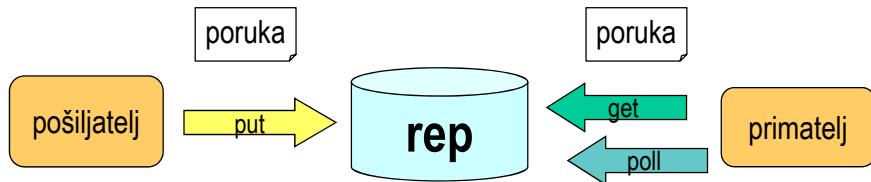


1. Klijentski objekt se povezuje s distribuiranim objektom.
 - pronalaženje udaljenog objekta
 - kreiranje *stuba* na klijentskom računalu
2. Klijent poziva metodu *stuba* (*stub* ili *proxy* - ima isto sučelje kao i poslužiteljski objekt).
3. *Stub* “pakira” poziv metode u poruku i šalje je *skeletonu*.
4. *Skeleton* čita primljenu poruku i poziva odgovarajuću metodu poslužiteljskog objekta.
5. *Skeleton* prima rezultat izvođenja metode, “pakira” ga u poruku i prosljeđuje *stubu*.
6. *Stub* čita poruku i vraća rezultat izvođenja metode klijentskom objektu.

- ◆ vremenska ovisnost
- ◆ klijent mora znati identifikator poslužitelja
- ◆ komunikacija nije perzistentna
- ◆ sinkrona komunikacija
 - klijent je blokiran dok ne primi odgovor od strane poslužitelja
- ◆ *pull* pokretanje komunikacije
 - klijent eksplicitno šalje zahtjev poslužitelju

- ◆ Distribuirani sustav
 - definicija i svojstva
 - međuoprema
 - karakteristike komunikacijske međuopreme
- ◆ Komunikacijska međuoprema
 - komunikacija korištenjem priključnica (*socket*)
 - poziv udaljene procedure/metode
 - **komunikacija porukama**
 - model objavi-pretplati

- ◆ Procesi komuniciraju razmjenjujući poruke.
- ◆ U komunikaciji sudjeluju izvor (pošiljatelj poruke) i odredište.
- ◆ Izvor šalje poruku, poruka se pohranjuje u rep koji je pridijeljen odredištu.
- ◆ Odredište čita poruku iz repa.
- ◆ Poruke sadrže podatke, važna je adresa odredišnog repa.
- ◆ Adresiranje se izvodi najčešće na nivou sustava, svaki rep ima jedinstven identifikator u sustavu.

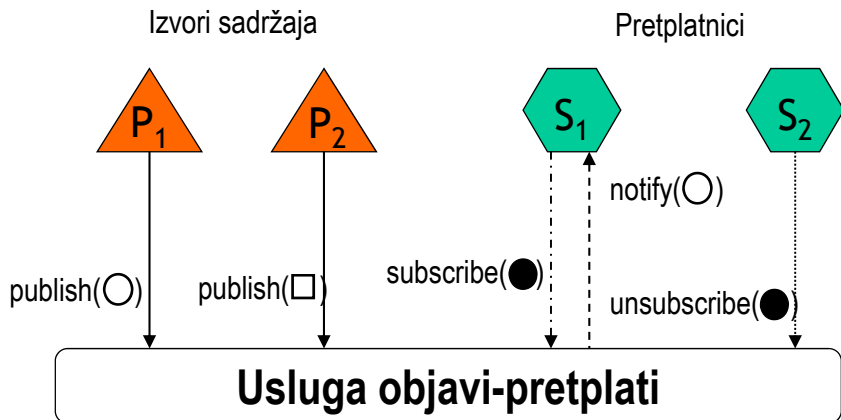


1 izvor : 1 odredište

- ♦ `put` – dodaj poruku u rep
- ♦ `get` – pročitaj poruku iz repa, primatelj je blokiran ako je rep prazan
- ♦ `poll` – provjeri postoje li poruke u repu i pročitaj prvu poruku ako takva postoji, primatelj nije blokiran

- ◆ **vremenska neovisnost**
 - primatelji i pošiljatelji ne moraju istovremeno biti aktivni, poruka se sprema u rep
- ◆ pošiljatelj mora znati identifikator odredišta, tj. njegovog repa
- ◆ komunikacija je **perzistentna**
- ◆ asinkrona komunikacija
 - pošiljatelj šalje poruku i nastavlja procesiranje neovisno o odgovoru od strane primatelja
- ◆ *pull* pokretanje komunikacije
 - primatelj provjerava postoji li poruka u repu

- ◆ Distribuirani sustav
 - definicija i svojstva
 - međuoprema
 - karakteristike komunikacijske međuopreme
- ◆ Komunikacijska međuoprema
 - komunikacija korištenjem priključnica (*socket*)
 - poziv udaljene procedure/metode
 - komunikacija porukama
 - **model objavi-pretplati**



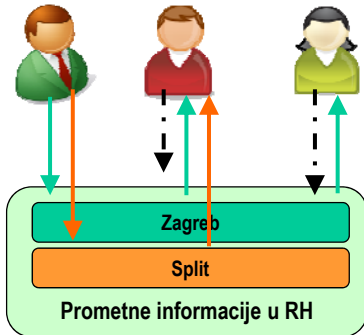
- ◆ izvori i pretplatnici razmjenjuju **obavijesti**
- ◆ **izvori** (*publishers*) objavljuju obavijest preko posrednika
- ◆ **pretplatnici** (*subscribers*) se pretplaćuju na određene vrste obavijesti
- ◆ kada izvor objavi novu obavijest, posrednik ga dostavlja svim odredištima pretplaćenim na taj tip obavijesti (**višeodredišni način komunikacije**)
- ◆ izvori i pretplatnici su neovisni, međusobno anonimni
 - posrednik vodi računa o pretplatnicima i njihovim pretplatama

◆ Pretplata na kanal

- tematsko grupiranje obavijesti (npr. vrijeme)
- hijerarhijski odnos kanala (npr. vrijeme u Europi, Hrvatskoj, Zagrebu)
- kanal – logička veza između izvora i odredišta

◆ Pretplata na sadržaj

- pretplata se definira ovisno o svojstvima i sadržaju obavijesti (skup atributa i vrijednosti)



Pretplatnik odabirom kanala
definira "filter" za obavijesti

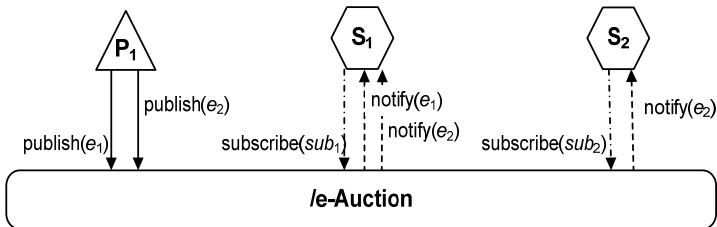


$e_1 = (\text{category} = \text{"books"} \\ \& \text{author} = \text{"D. Adams"} \\ \& \text{title} = \text{"The Hitchhiker's Guide through the Galaxy"} \\ \& \text{price} = 9.99 \text{ EUR})$

$e_2 = (\text{category} = \text{"books"} \\ \& \text{author} = \text{"J.R.R. Tolkien"} \\ \& \text{title} = \text{"The Lord of the Rings"} \\ \& \text{price} = 19.99 \text{ EUR})$

$sub_1 = (\text{category} == \text{"books"} \\ \& \text{price} < 20 \text{ EUR})$

$sub_2 = (\text{category} == \text{"books"} \& \\ \text{author} == \text{"J.R.R. Tolkien"} \\ \& \text{price} < 20 \text{ EUR})$

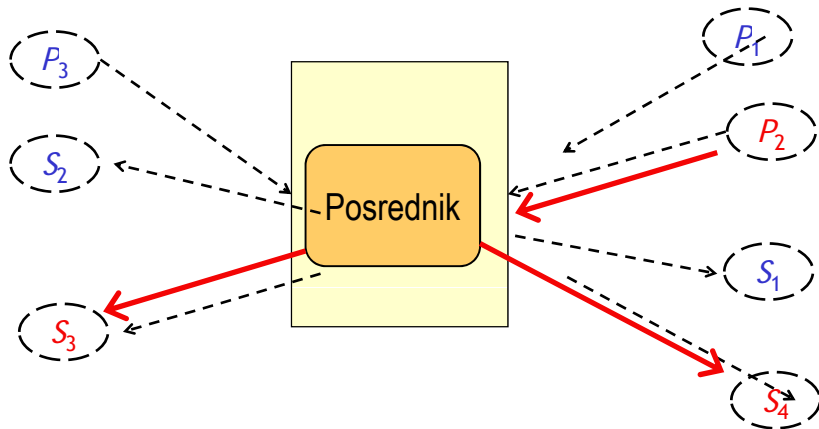


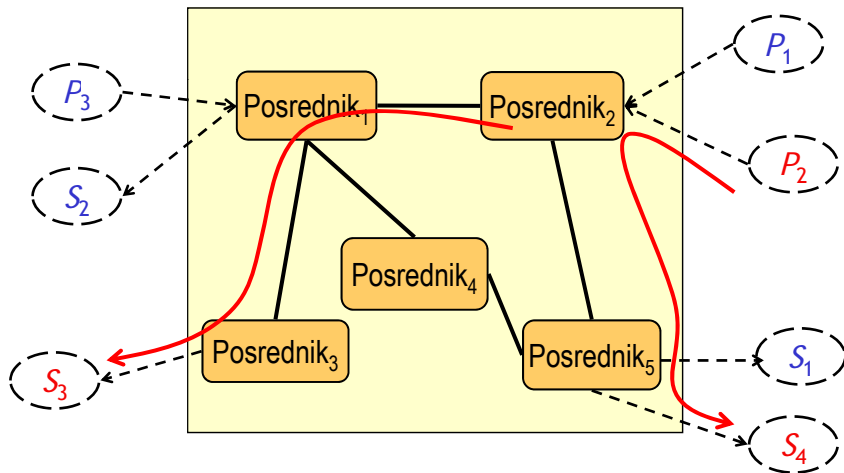
◆ Centralizirana

- svi izvori i odredišta razmjenjuju obavijesti preko jednog poslužitelja posrednika
- poslužitelj pohranjuje sve pretplate i prosljeđuje objavljene obavijesti

◆ Distribuirana

- za veliki broj izvora, odredišta i pretplata
- skup poslužitelja, svaki je poslužitelj zadužen za izvore i odredišta u svojoj domeni
- algoritmi za usmjeravanje informacija o pretplatama i usmjeravanje objavljenih događaja





- ◆ **vremenska neovisnost**
 - izvori i odredišta ne moraju istovremeno biti aktivna, posrednik pohranjuje poruku
- ◆ izvor ne mora znati identifikator odredišta (**anonimnost**), o tome se brine posrednik
- ◆ komunikacija je **perzistentna**
- ◆ **asinkrona komunikacija**
 - izvor šalje poruku i nastavlja procesiranje neovisno o odgovoru od strane odredišta
- ◆ **push** pokretanje komunikacije
 - izvor šalje poruku posredniku koji je prosljeđuje odredištima bez prethodnog eksplicitnog zahtjeva

- ◆ **personalizacija primljenog sadržaja**
 - filtriranje objavljenih poruka prema pretplatama
- ◆ **proširivost sustava**
 - dodavanje novog izvora ili odredišta ne utječe na ostale strane u komunikaciji
- ◆ **skalabilnost**
 - implementacija distribuiranog sustava (distribuirana arhitektura)