Diplomski studij

Informacijska i komunikacijska tehnologija:

Telekomunikacije i informatika

Računarstvo:

Programsko inženjerstvo i informacijski sustavi

Računarska znanost

Ak.g. 2008./2009.

Raspodijeljeni sustavi

4.

Splet računala, P2P, programski agent

14.10.2008.

Sadržaj predavanja

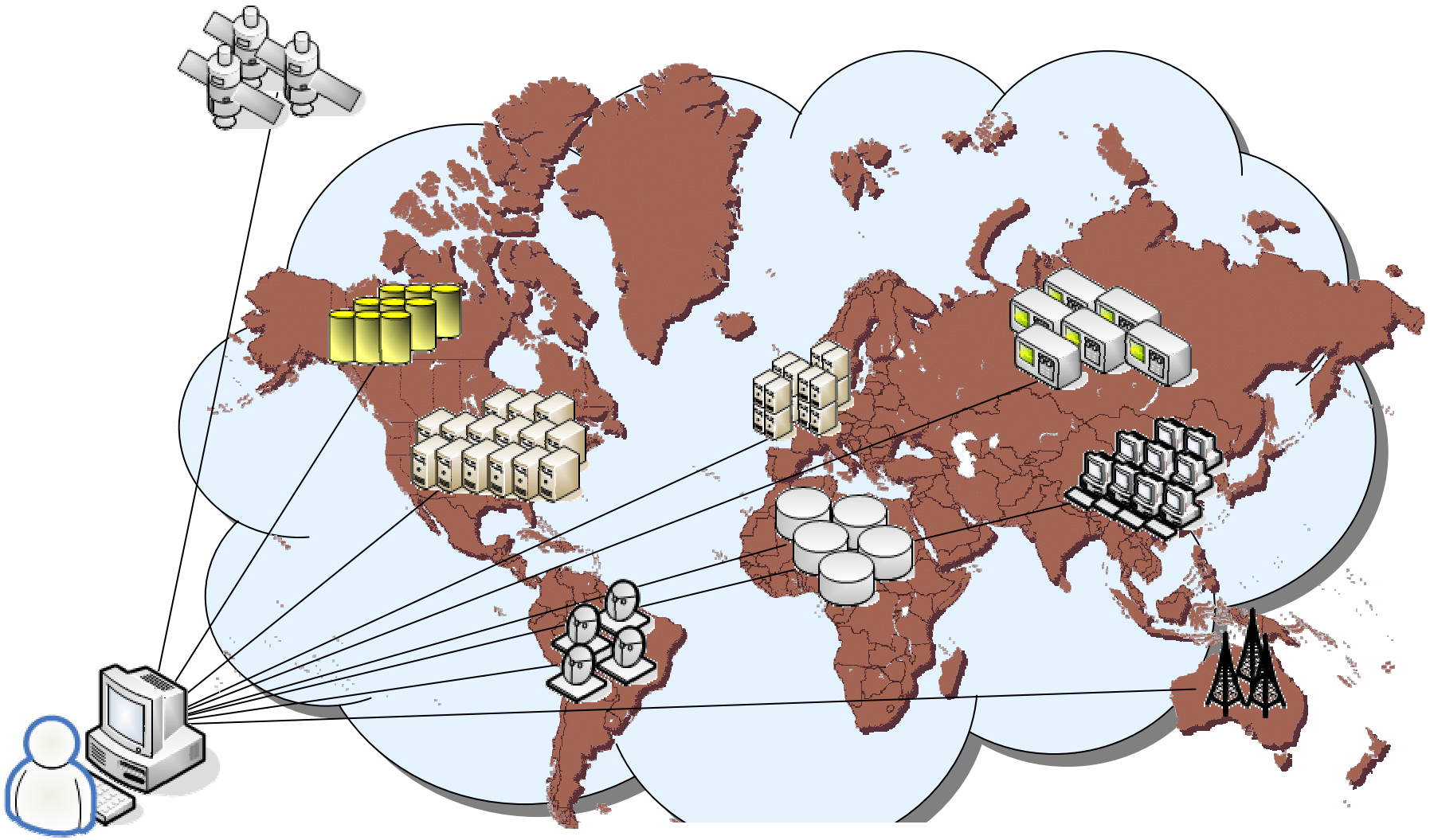
- Splet računala
  - Motivacija
  - Osnovni elementi spleta računala
  - Grozd računala
  - Splet računala
  - Usporedba grozda i spleta računala
- P2P
  - Centralizirani i decentralizirani raspodijeljeni sustavi
  - Definicija sustava P2P
  - Nestrukturirani sustavi P2P
  - Strukturirani sustavi P2P
- Programski agent

Splet računala

Motivacija

- Globalna mreža Internet
  - Uspostava komunikacije između geografski raspodijeljenih korisnika
  - Objavljivanje, pretraživanje i razmjena sadržaja
- Internet osnova za izgradnju nove vrste aplikacija ?
- **Splet računala**
  - Raspodijeljeni sustav za usklađeno **upravljanje**, **razmjenu** i **korištenje raznovrsnih** sredstava u globalnoj mreži Internet

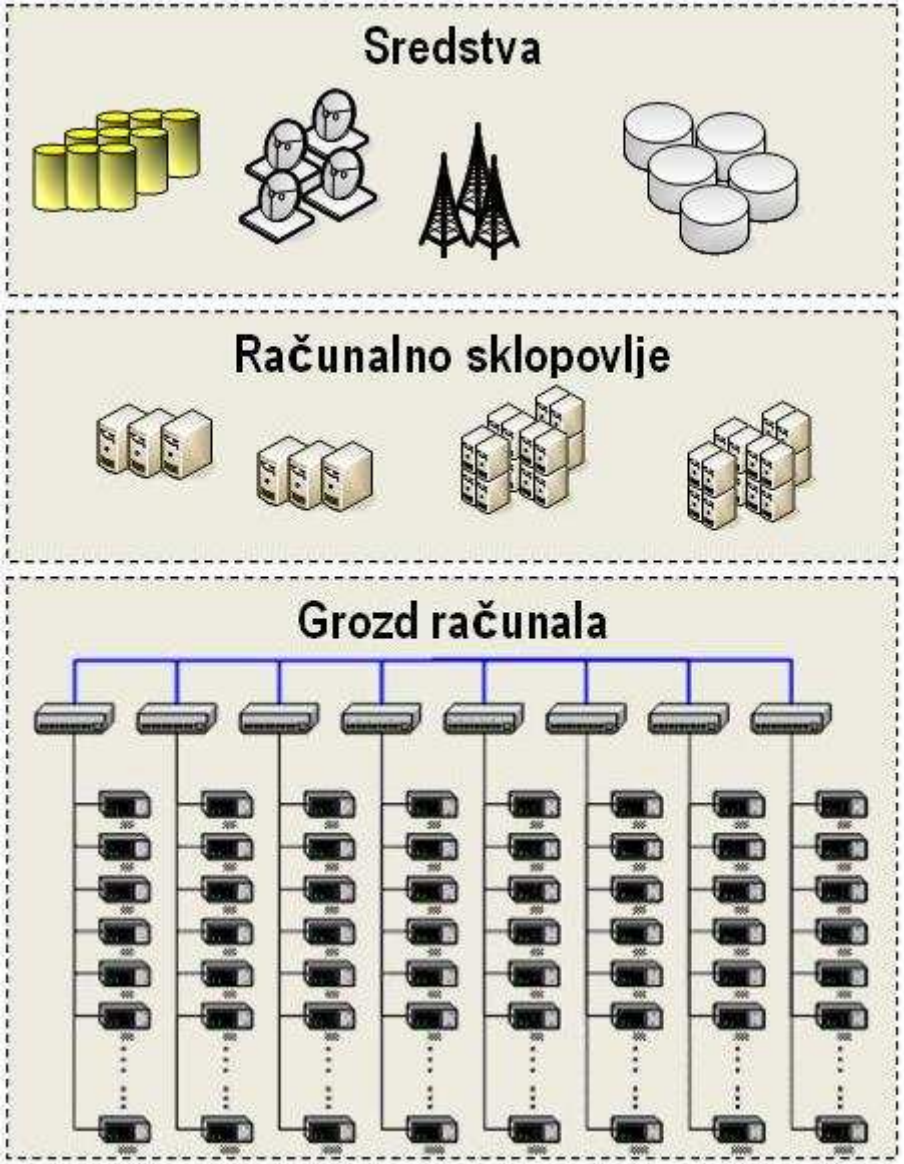
Osnovni elementi spleta računala



Splet računala

Osnovni elementi spleta računala

- Struktura spleta računala
  - Sredstva
    - Podaci, aplikacije, računalni procesi, spremnički prostor, računalna snaga
  - Računalno sklopovlje
    - Procesorske jedinice, radna računala, mrežno sklopovlje, osjetila, aktuatori
  - Grozd računala
    - Čvrsto povezani skup sklopovlja i sredstava



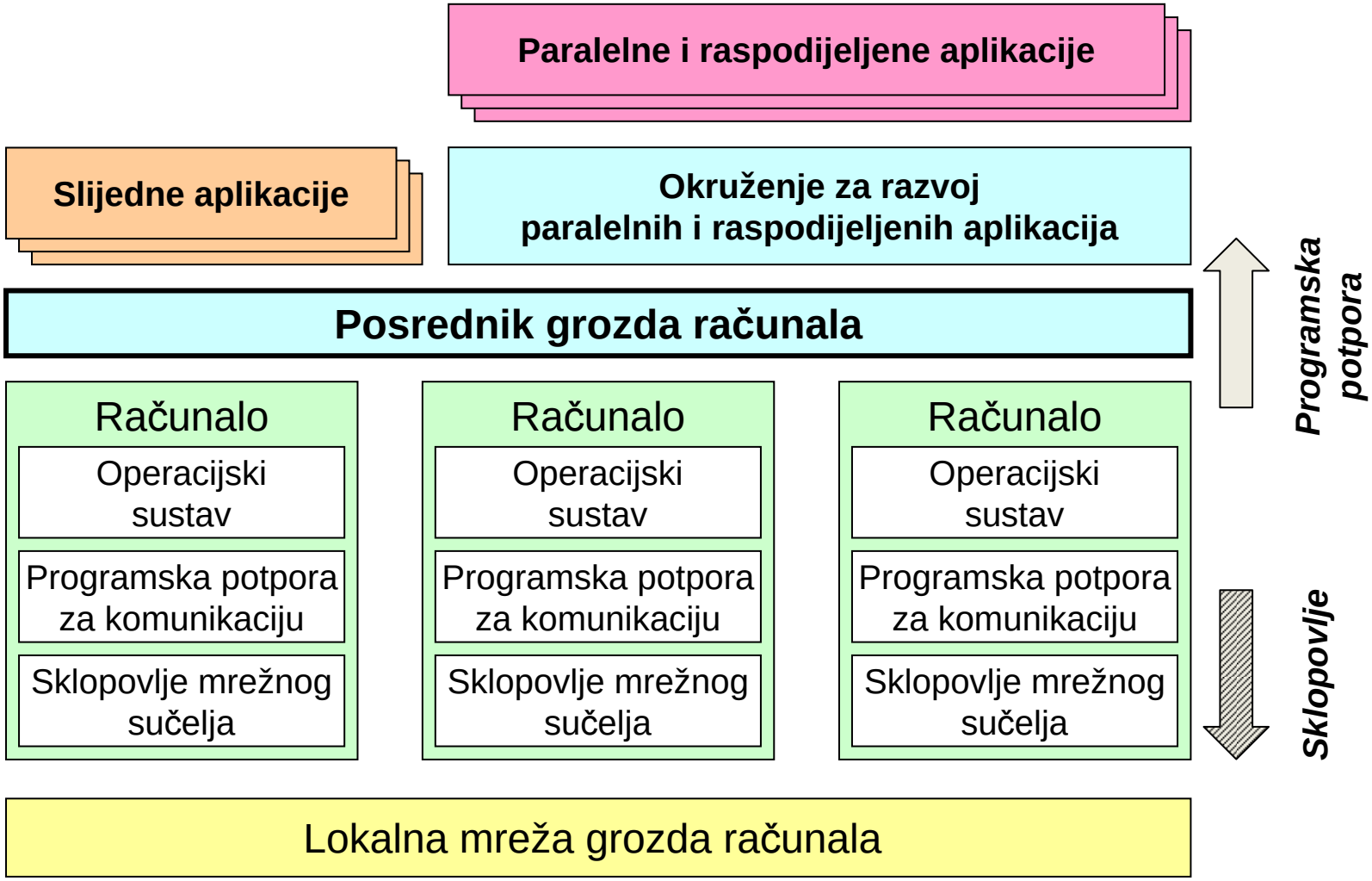
Grozd računala

- Definicija grozda računala
- Arhitektura grozda računala
- Aplikacije grozda računala
- Primjeri grozda računala

Definicija grozda računala

- Sustav za izvođenje paralelnih ili raspodijeljenih aplikacija zasnovan na skupu računala koja su povezana lokalnom mrežom i zajednički djeluju kao objedinjeno računalno sredstvo

Arhitektura grozda računala



Sklopovlje grozda računala

- Okolina za izvođenje
  - Simetrični višeprocesorski sustavi (engl. *symmetric multiprocessor systems*)
  - Skup radnih stanica (engl. *cluster of workstations*)
  - Skup osobnih računala (engl. *cluster of personal stations*)
- Komunikacijska mreža
  - Sabirnica (engl. *bus*)
  - Komunikacijska matrica (engl. *crossbar*)
  - Lokalna mreža visoke propusnosti (*Gigabit Ethernet*)
  - Raznovrsne vlasničke tehnologije (*Quadrics QsNet, Myrinet*)

Programska potpora grozda računala

- Operacijski sustav
  - Linux, Hewlett Packard UniX (HPUX)
  - Sun Solaris
  - Microsoft Windows
- Posrednički sustav grozda računala
  - Podsustav za upravljanje poslovima
  - Podsustav za nadgledanje rada
  - Razvojno okruženje
  - Podsustav dijeljenog spremničkog prostora

Programska potpora grozda računala

- Podsustav za upravljanje poslovima
  - Korisničko sučelje za upravljanje procesima i aplikacijama
  - Definiranje značajki korisničkih procesa i aplikacija
  - Korisničke postavke za upravljanje sredstvima
  - Primjeri: *Sun Grid Engine* (*SGE*), *Portable Batch System* (*PBS*), *Condor*
- Podsustav za nadgledanje rada
  - Stanje sredstava
  - Statistika zauzeća sredstava
  - Primjeri: *Ganglia*, *Supermon*

Programska potpora grozda računala

- Razvojno okruženje
  - Razvoj logike procesa
  - Razvoj logike međudjelovanja procesa
  - Usmjeravanje procesa na izvođenje
  - Komunikacija zasnovana na razmjeni poruka
  - Primjeri: *Message Passing Interface* (*MPI*), *Linda,*
    High Performace Fortran (*HPF*),
    Z-Level Programming Language (*ZPL*)
- Podsustav dijeljenog spremničkog prostora
  - Raspodijeljeno spremanje i dohvat velike količine podataka
  - Primjeri: *Network File System* (*NFS*), *AFS*, *Lustre*

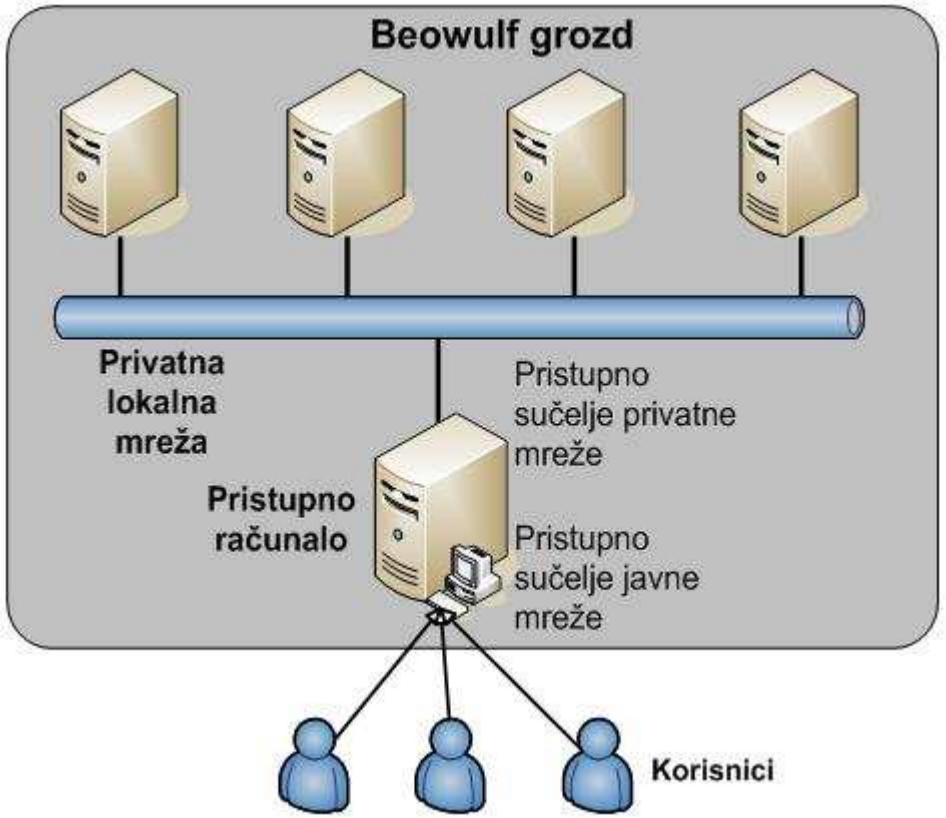
Aplikacije grozda računala

- Primjer aplikacija
  - Znanstvene aplikacije u biologiji, medicini, fizici
    (*složeni izračuni velikog skupa podataka*)
  - Baze podataka velikih razmjera
    (*pouzdanost* i *dostupnost podataka*)
  - Poslovne aplikacije
    (*dubinska analiza korisničkih obrazaca ponašanja*)

- Osnovni razredi aplikacija
  - Računalno zahtjevne aplikacije
  - Komunikacijski zahtjevne aplikacije
  - Aplikacije visokog stupnja dostupnosti

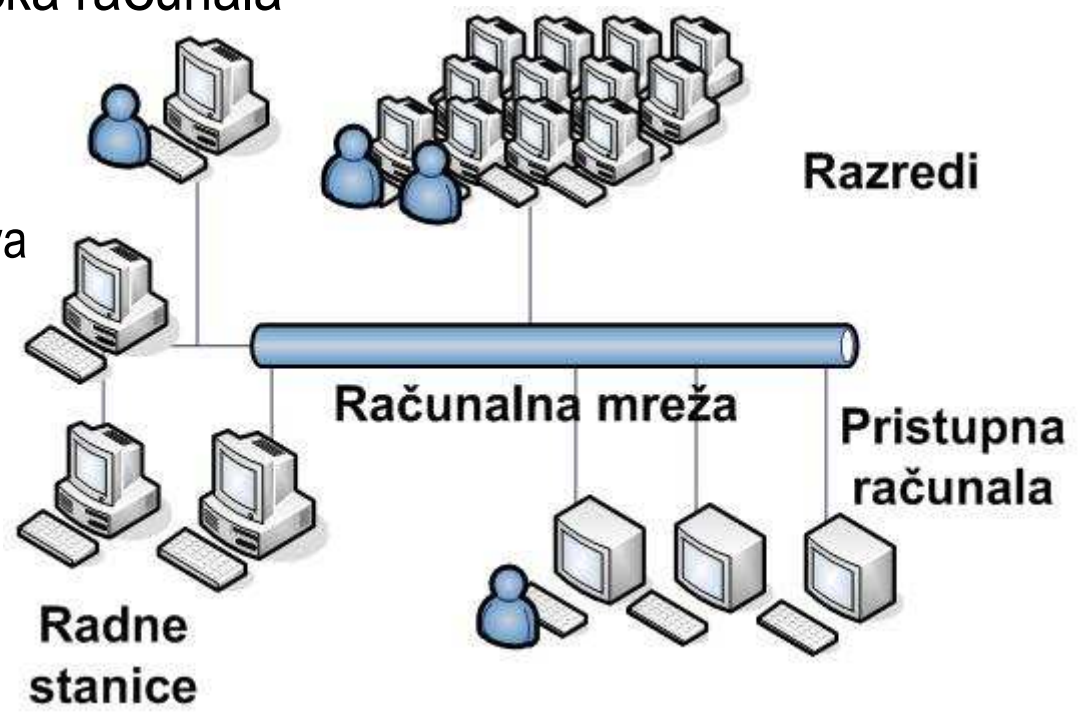
Aplikacije grozda računala

- Računalno zahtjevne aplikacije
  - Složene računske operacije
  - Potrebna velika količina procesorske snage
  - Skup radnih jedinica
    - Privatna mreža
  - Pristupno računalo
    - Posrednik između privatne i javne mreže
    - Upravljanje i nadgledanje izvođenja poslova



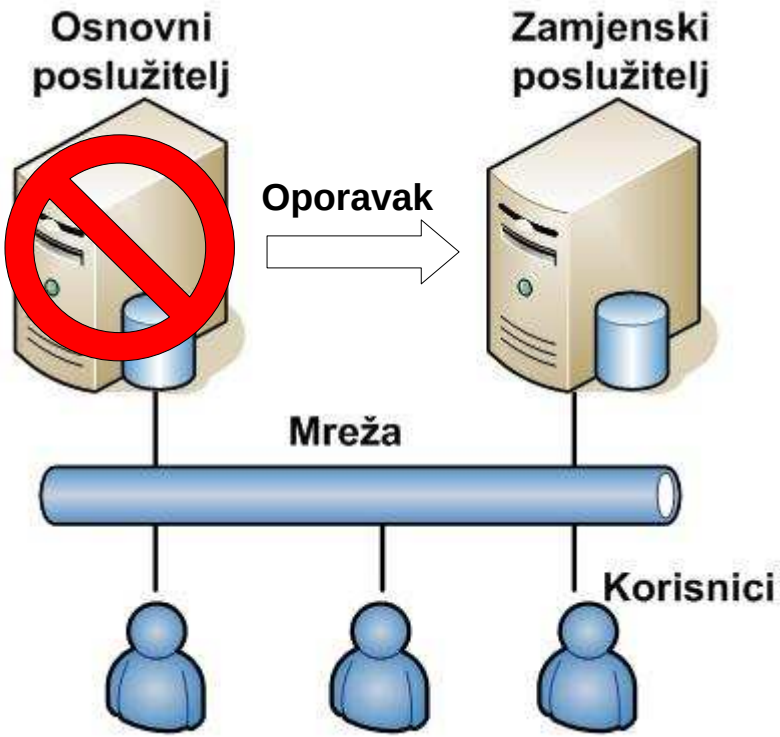
Aplikacije grozda računala

- Komunikacijski zahtjevne aplikacije
  - Razmjena velike količine podataka ili suradnja u stvarnom vremenu
    - Razmjena rezultata eksperimenata, video konferencije, dijeljenje aplikacija
  - Raznovrsna korisnička računala povezana mrežom
    - Velika propusnost
    - Kratko vrijeme odziva



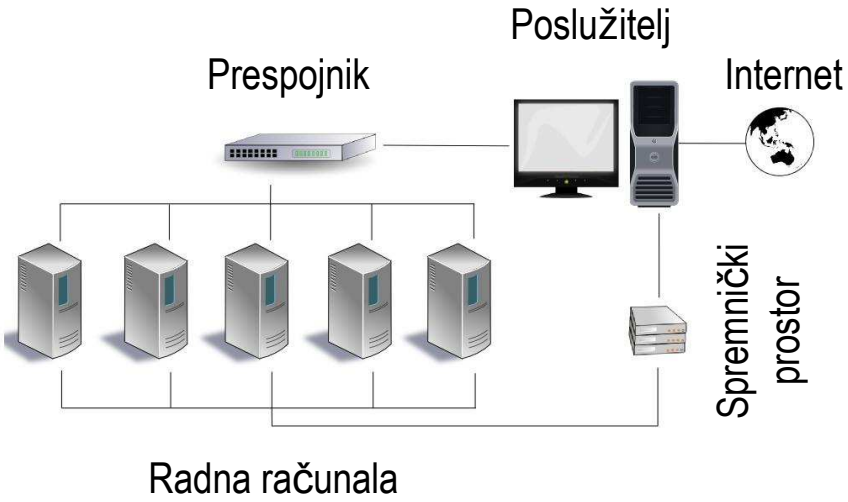
Aplikacije grozda računala

- Aplikacije visokog stupnja dostupnosti
  - Osnovni sustavi za poslovanje
    - Mrežni poslužitelji
    - Poslužitelji pošte
    - Baze podataka
  - Dostupnost i pouzdanost
    - Bez središnjeg upravljanja
    - Raspoređivanje zahtjeva sredstvima
    - Izbjegavanje nedostupnih sredstava u trenutku kvara



Primjer grozda računala

- **Beowulf** (www.beowulf.org)
  - Do 1024 radnih računala
    - Intel, AMD procesorske jedinice
  - Standardne mrežne tehnologije
    - Fast Ethernet, Gigabit Ethernet
  - Otvoreni OS (Linux)



- **Isabella** (www.srce.hr/isabella)
  - 100 radnih računala
    - HP Blades, Dual Intel Xeon, AMD Opteron
  - Lokalna mreža (10 Gb Infiniband)
  - 20 projekta s nekoliko desetaka članova

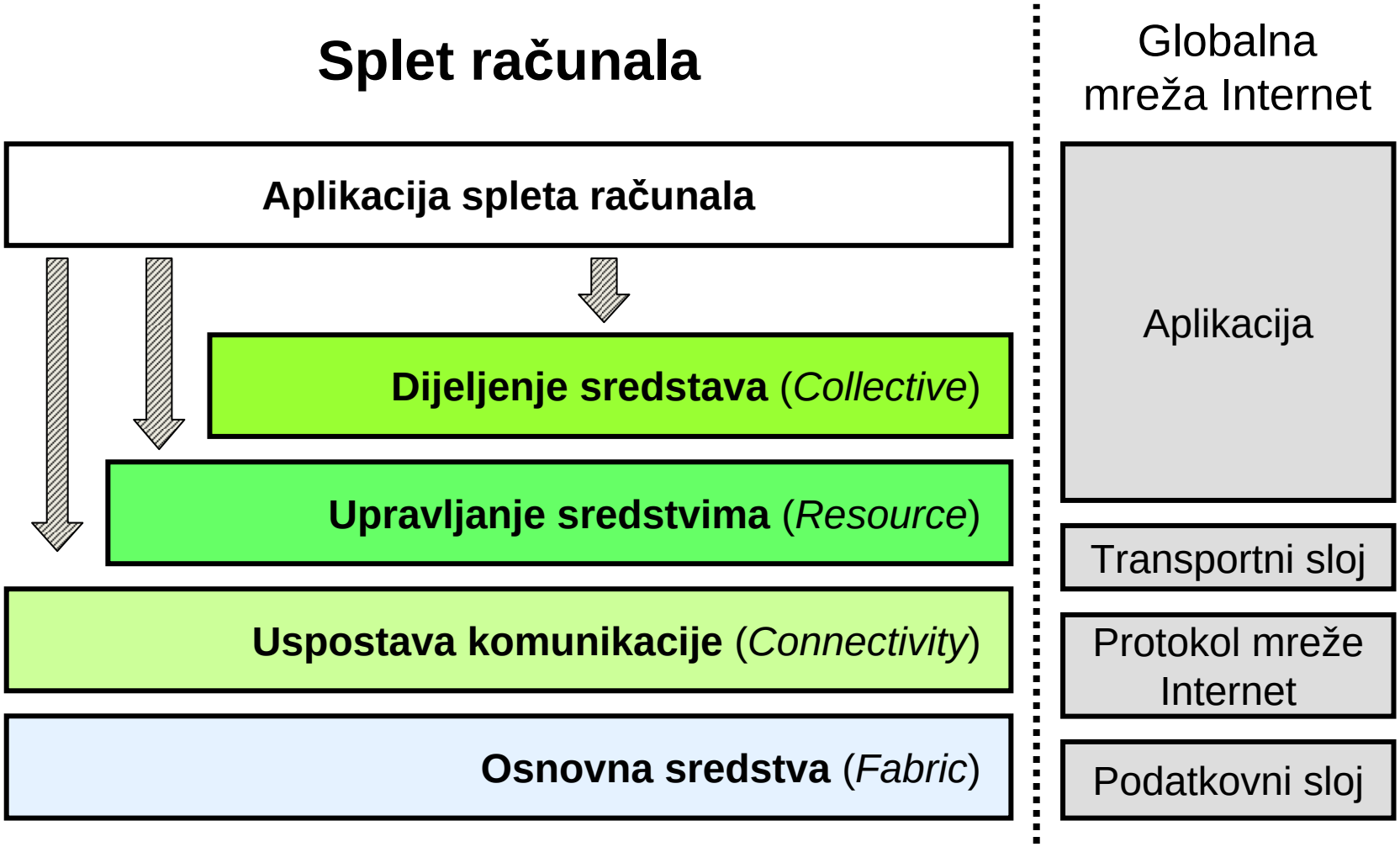
Splet računala

- Definicija spleta računala
- Arhitektura spleta računala
- Vrste spletova računala
- Primjeri spleta računala

Definicija spleta računala

- Splet računala je raspodijeljena računalna okolina koja omogućava
  - Usklađeno dijeljenje heterogenih i geografski raspršenih sredstava (računalna snaga, spremnički prostor, mrežna propusnost, aktuatori, osjetila)
  - Usklađeno dijeljenje informacija, procesa i aplikacija unutar i između dinamičnih virtualnih organizacija raznovrsnih institucija

Arhitektura spleta računala



Osnovna sredstva

- Uspostava raznovrsnih i raznorodnih sredstva koja ostvaruju osnovne funkcionalnosti spleta računala

- Osnovne vrste sredstava
  - Računalna sredstva (engl. *computational resources*)
  - Spremnički prostor (engl. *storage systems*)
  - Katalozi sredstava (engl. *resource catalogues*)
  - Mrežna sredstva (engl. *network resources*)
  - Osjetila i aktuatori (engl. *sensors and actuators*)

Uspostava komunikacije

- Osnova za uspostavu sigurne i pouzdane komunikacije između sredstava dostupnih u spletu računala

- Osnovne značajke komunikacije
  - Prijenos velike količine podataka
  - Učinkovito usmjeravanje sadržaja
  - Naslovljavanje sredstava i sudionika komunikacije
  - Sigurnost
  - Pouzdanost

Upravljanje sredstvima

- Osnovni protokoli za udaljeno postavljanje, nadgledanje, korištenje sredstava
  - Informacijski protokoli
  - Upravljački protokoli

- Informacijski protokoli
  - Dohvaćanje, pregled i analiza stanja sredstava u spletu računala

    (zauzeće sredstava, broj kvarova, broj korisnika,…)
- Upravljački protokoli
  - Upravljanje postavkama za dijeljenje, korištenje i upravljanje sredstvima spleta računala

    (prava pristupa, dostupnost, udruživanje, usmjeravanje,…)

Dijeljenje sredstava

- Protokoli i usluge za učinkovito dijeljenje, usklađivanje rada i upravljanje grupom sredstava spleta računala

- Usluge za dijeljenje sredstava
  - Imeničke usluge
  - Usluge za raspoređivanje zahtjeva za pristup sredstvima
  - Usluge za posredovanje u suradnji sredstava
  - Usluge za nadgledanje rada skupine sredstava
  - Okruženja za razvoj logike suradnje sredstava
  - Usluge za naplatu korištenja sredstava

Vrste spletova računala

- Spletovi za složene proračune (engl. *Computational grids*)
  - Modeliranje i simuliranje složenih znanstvenih eksperimenata
- Podatkovni spletovi računala (engl. *Data grids*)
  - Spremanje i obrada velike količine podataka
- Poslovni spletovi računala (engl. *Business grids*)
  - Dubinska analiza velike količine podataka
- Bežični spletovi računala (engl. *Wireless grids*)
  - Potpora bežičnim korisničkim uređajima

Primjeri spleta računala

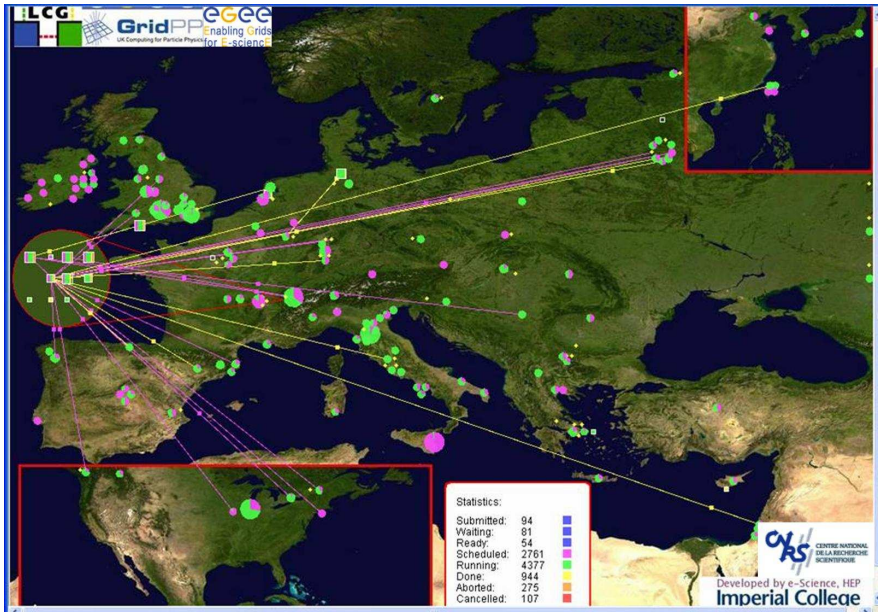
- TeraGrid
- Enabling Grids for E-science in Europe (EGEE)
- CROGrid

TeraGrid

- Nacionalni splet računala SAD-a
- Razvoj najveće okoline spleta računala za znanstvena istraživanja
  - Istraživanje lijekova za rak
  - modeliranje i predviđanje vremenskih uvjeta
- Sredstva
  - 20 Tflops računalne snage
  - 1 Pbyte spremničkog prostora
  - 40 Gbit/s mrežne propusnosti

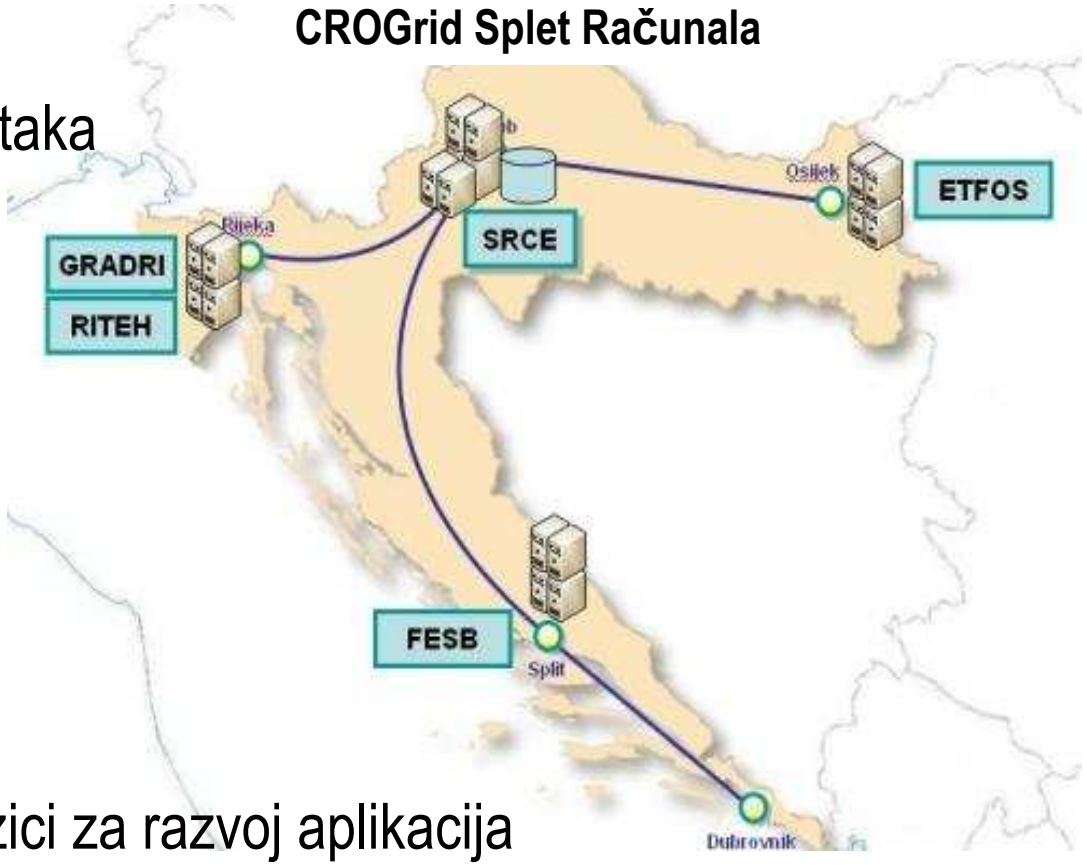
Enabling Grids for E-science in Europe (EGEE)

- Splet računala u EU
- Splet računala za izvođenje znanstvenih aplikacija
  - Obrada podataka sustava *Large Hadron Collider* (LHC)
  - Obrada biomedicinskih podataka
  - Modeliranje i simuliranje prirodnih pojava
- EGEE-III sredstva
  - 72,000 procesnih jedinica
  - 20 Pbyte spremničkog prostora
  - 10,000 reg. korisnika



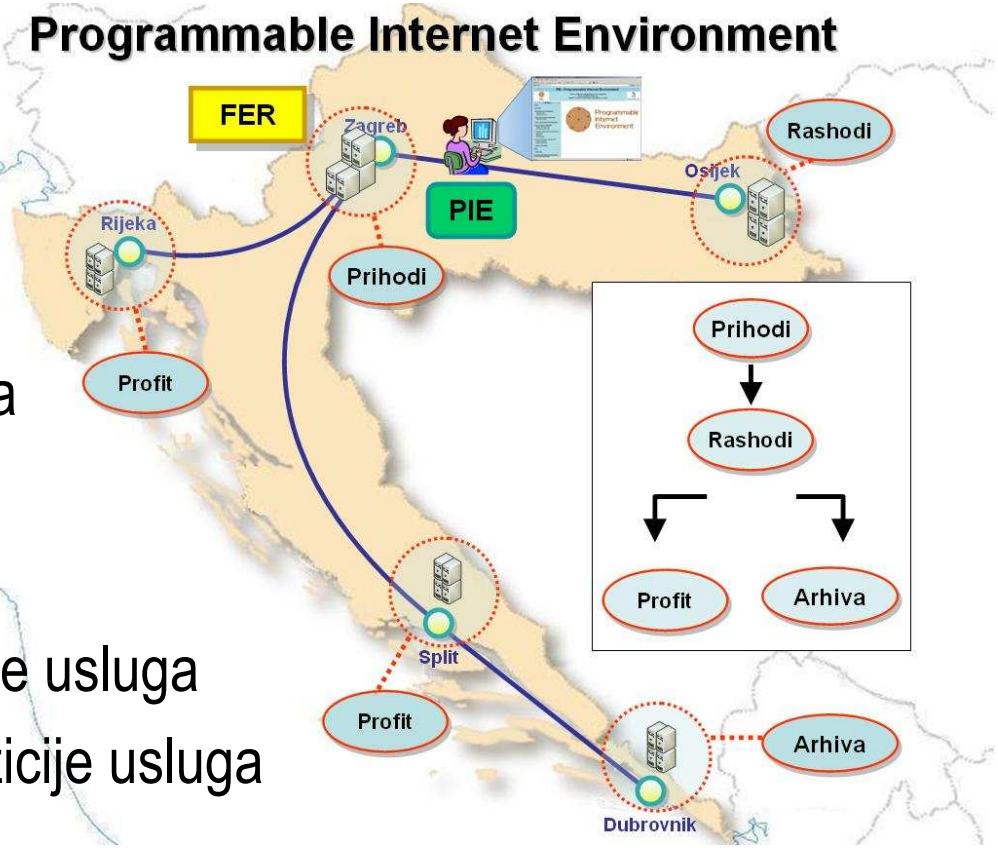
CroGRID

- Hrvatski splet računala (www.cro-grid.hr, www.cro-ngi.hr)
  - FER, ETK, SRCE, ETFOS, FESB, GRADRI, RITEH
- **Aplikacije**
  - Dubinska analiza podataka
  - Analiza proteina
  - Pametni transport
- **Infrastruktura**
  - Grozdovi računala
- **Posrednički sustav**
  - Programski modeli i jezici za razvoj aplikacija



CroGRID posrednički sustav

- Programmable Internet Environment (PIE)
  - Razvoj raspodijeljenih aplikacija zasnovanih na kompoziciji usluga
- Struktura PIE sustava
  - Prividna mreža
  - Imenik i spremnik usluga
  - Mehanizmi suradnje i natjecanja usluga
  - Razvoj opisa kompozicije usluga
  - Izvođenje opisa kompozicije usluga



Usporedba grozda i spleta računala

	Grozd računala	Splet računala
Aplikacije	Izvođenje računalno zahtjevnih aplikacija	Dijeljenje raznovrsnih sredstava u globalnoj mreži
Tehnologija	Primjena vlasničkih i standardiziranih tehnologija	Primjena standardiziranih tehnologija
Geografska raspodijeljenost	Elementi sustava na bliskoj geografskoj udaljenosti	Elementi sustava globalno raspodijeljeni
Upravljanje	Središnje upravljanje sredstvima sustava	Više administrativnih domena za upravljanje sustavom
Povezanost	Čvrsto povezane strukture	Labavo povezane strukture
Prilagodljivost	Zatvorena okolina	Otvorena okolina

Dodatne informacije

- Knjige
  - I. Foster, C. Kesselman: **"The Grid 2: Blueprint for a New Computing Infrastructure"**, Morgan Kaufmann, 2004.
- Web
  - **Open Grid Forum** (www.ogf.org)
  - **Grid Computing**, IEEE DS Online (dsonline.computer.org/gc)
  - **The Globus Alliance** (www.globus.org)
- FER Kolegiji
  - **Računarstvo zasnovano na uslugama**

    1. sem, dipl., PI&IS, RI, RZ
  - **Posrednici umreženih sustava**

    2. sem, dipl., PI&IS, RI, RZ , T&I

Diplomski studij

Informacijska i komunikacijska tehnologija:

Telekomunikacije i informatika

Računarstvo:

Programsko inženjerstvo i informacijski sustavi

Računarska znanost

Ak.god. 2008./2009.

Raspodijeljeni sustavi

4.

Sustavi P2P

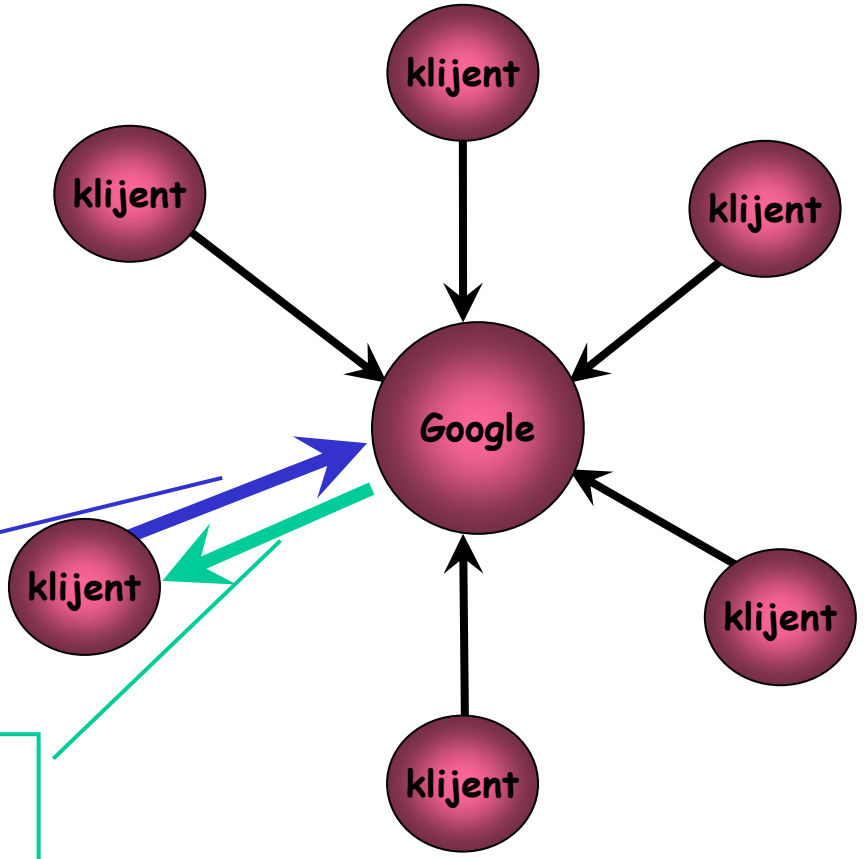
Sadržaj predavanja

- Centralizirani i decentralizirani raspodijeljeni sustavi
- Definicija sustava P2P
- Nestrukturirani sustavi P2P
- Strukturirani sustavi P2P

Centralizirani distribuirani sustavi

Primjer - Web tražilice

- npr. Google
- 91,000,000 upita po danu
- oko 2,480,000,000 indeksiranih dokumenata



Upit
"FER adresa"

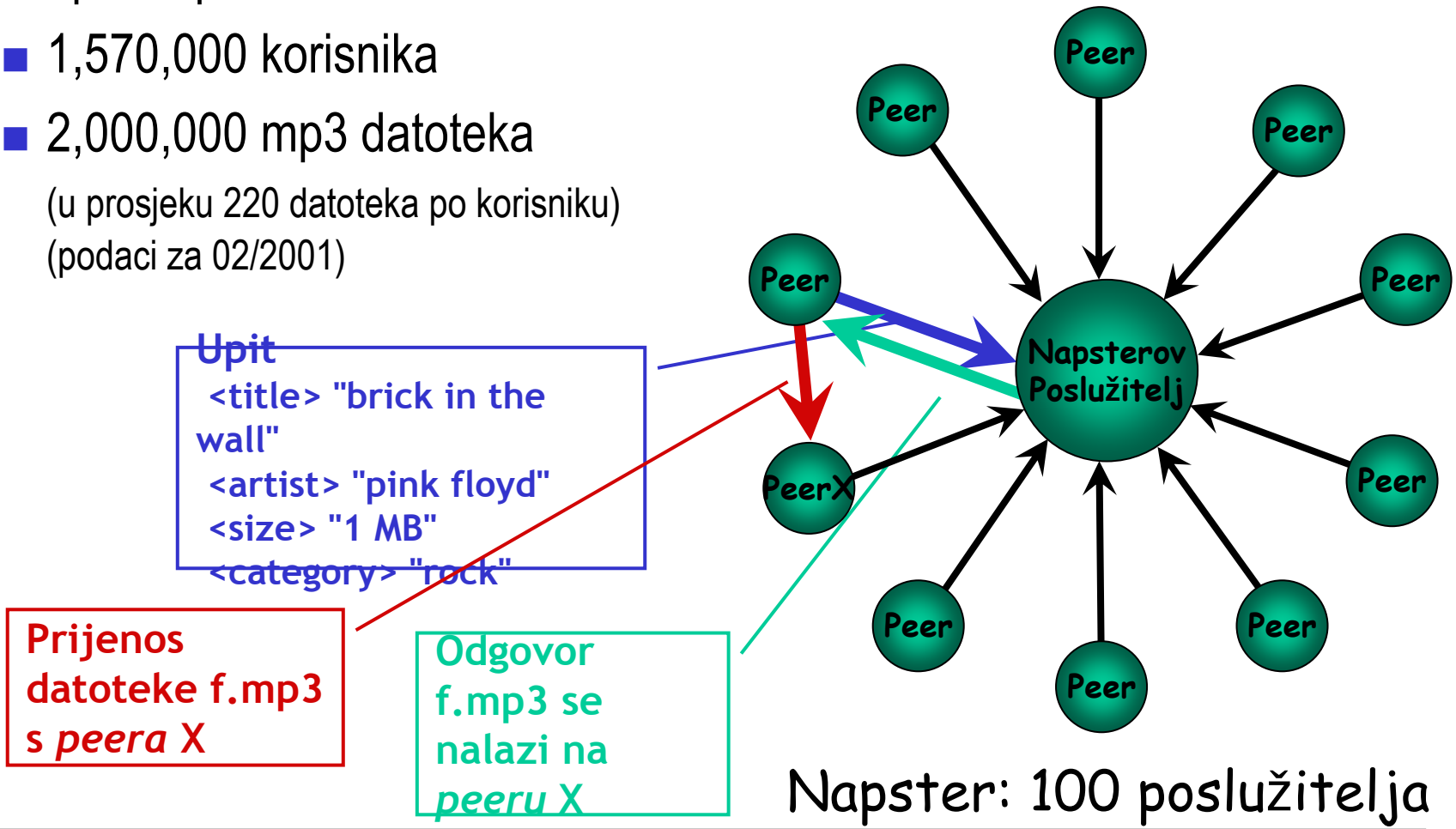
Lista rangiranih odgovora
- www.fer.hr....

Grozd računala: 15000 poslužitelja
(podatak iz 2003)

Decentralizirani distribuirani sustavi (1)

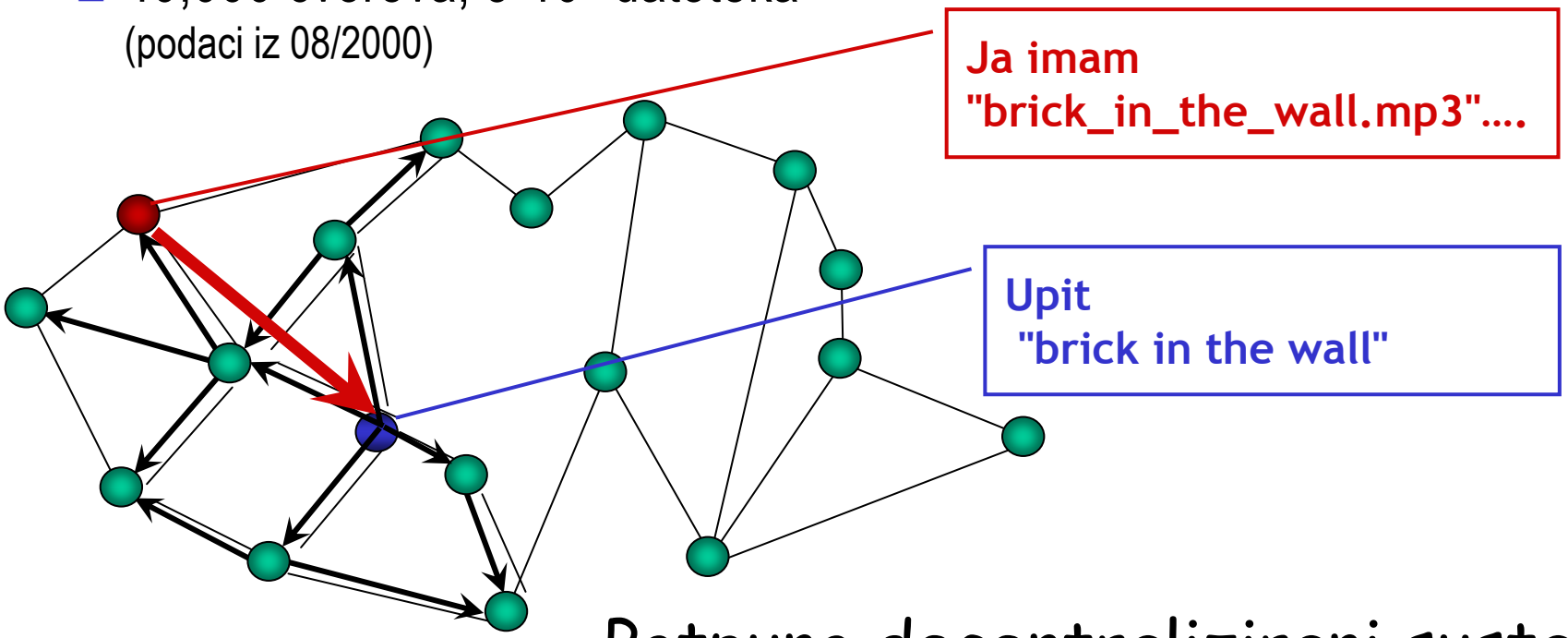
- Primjer – aplikacija za razmjenu mp3 datoteka
  - npr. Napster
  - 1,570,000 korisnika
  - 2,000,000 mp3 datoteka

    (u prosjeku 220 datoteka po korisniku)
    (podaci za 02/2001)



Decentralizirani distribuirani sustavi (2)

- Primjer: aplikacija za razmjenu datoteka
  - npr. Gnutella
  - 40,000 čvorova, $3 \cdot 10^6$ datoteka
    (podaci iz 08/2000)



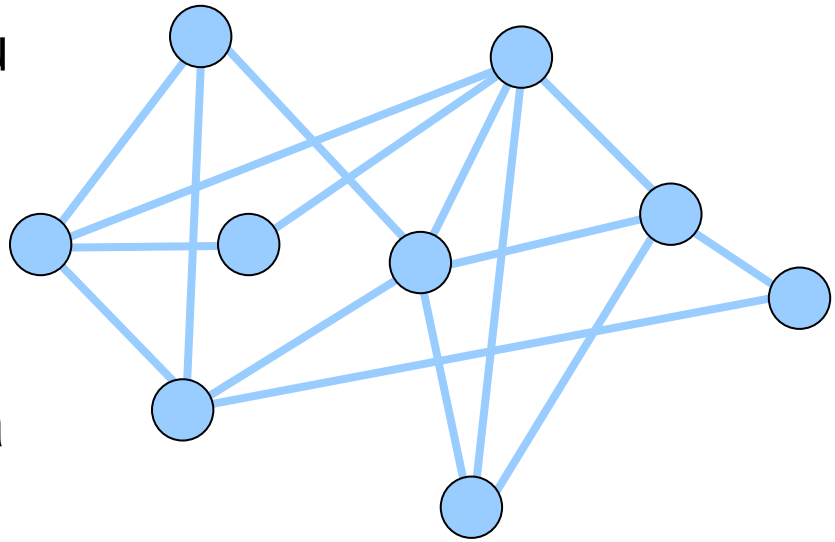
Potpuno decentralizirani sustav

Sadržaj predavanja

- Centralizirani i decentralizirani distribuirani sustavi
- Definicija sustava P2P
- Nestrukturirani sustavi P2P
- Strukturirani sustavi P2P
- Primjeri aplikacija

Definicija sustava peer-to-peer (P2P)

- mreža istovrsnih “čvorova” - *peerova*
- svaki *peer* istovremeno vrši funkciju poslužitelja i klijenta
- svaki čvor “plaća” sudjelovanje u mreži nudeći dio vlastitih resursa (memorija, CPU) ostalim čvorovima
- potencijalno sustav P2P nudi neograničene resurse (broj *peerova* nije ograničen)



Mreža peerova

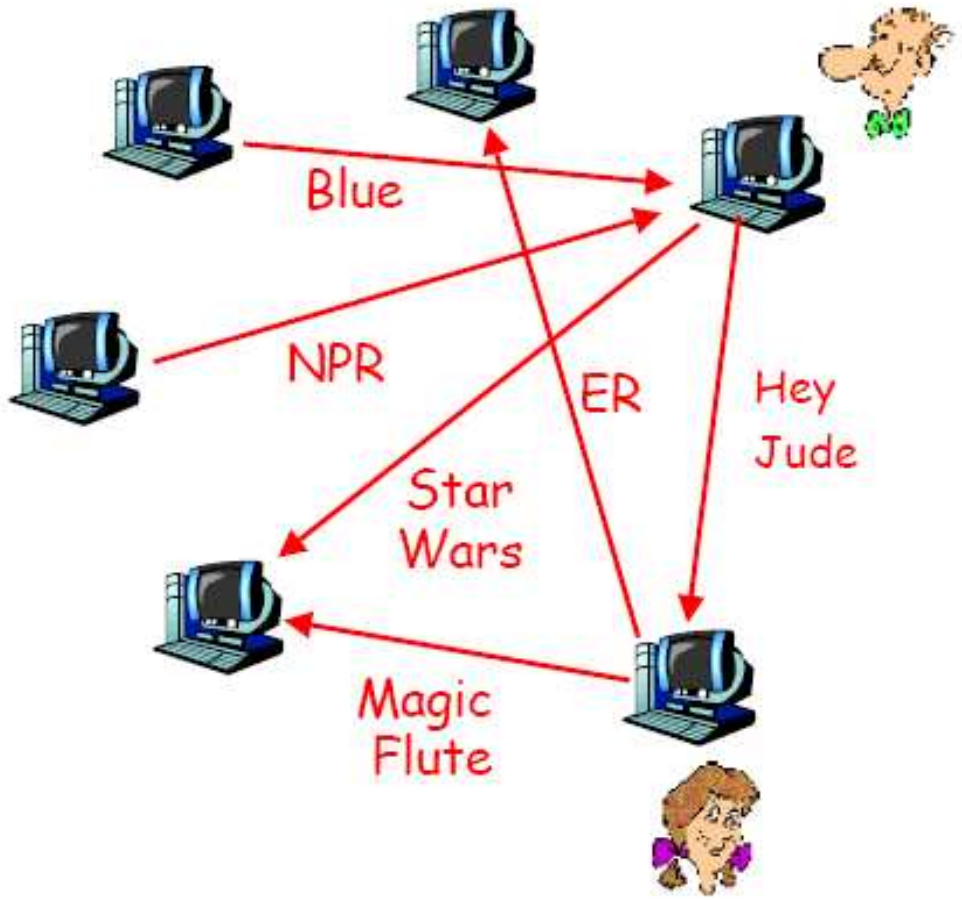
- Kada su 2 *peera* susjedi?
  - otvorena TCP konekcija ili
  - virtualne grane među *peerovima*, *peer* zna IP adresu drugog *peera*
- Kako se održava mreža *peerova*?
  - mreža je izrazito nestabilna
  - npr. *peer* periodički provjerava stanje susjeda (ping porukama)
  - ako je susjed nedostupan, briše se iz liste susjeda
  - potreban je poseban algoritam za otkrivanje novih susjeda
  - poseban algoritam za dodavanje novog peera u postojeću mrežu (najčešće poznaje listu peerova za inicijalni kontakt)

Obilježja sustava P2P

- decentralizirani distribuirani sustav
  - nema centralizirane koordinacije među *peerovima*
  - ne postoji jedna točka ispada
- samoorganizirajuća mreža čvorova
  - *peerovi* su međusobno neovisni
- skalabilan sustav
  - dodavanje novih čvorova i ispad čvorova je podržano organizacijom P2P mreže i definiranim protokolima
- globalni informacijski sustav bez velikih ulaganja
  - raspodijeljena instalacija i održavanje

Osnovna zadaća sustava P2P (1)

- Pronalaženje resursa (npr. datoteka) u sustavima P2P!



Osnovna zadaća sustava P2P (2)

Kako pronaći podatak ***d*** u mreži peerova?

- "naivno rješenje": poslati upit svim peerovima u mreži
  - problemi: moram znati adrese svih peerova, što je s mrežnim prometom?
- "manje naivno rješenje": poslati upit odabranim peerovima u mreži
  - problemi: kako odabrati peerove, hoću li sigurno pronaći podatak ***d***
- "pametnije" rješenje
  - *peer* s adresom ***p*** pohranjuje podatak ***d*** (npr. datoteka) koji se može jedinstveno identificirati ključem ***k*** (npr. ***k*** = hash(***d***))
  - za dani ključ ***k*** pronaći *peera* ***p*** na kome je pohranjen podatak ***d***
  - dovoljno je znati funkciju **f:*k*** $\rightarrow$ ***p*** da bismo pronašli ***d*** jer ta funkcija određuje identifikator *peera* na kome je pohranjen ***d***
  - Kako definirati i implementirati funkciju **f** u distribuiranoj i decentraliziranoj okolini?

Vrste sustava P2P

- nestrukturirani sustavi
  - mrežna topologija nema definiranu strukturu
  - mrežu *peerova* čini slučajan graf, npr. peer "poznaje" svoja četiri susjeda i preko njih pretražuje cijelu mrežu
  - primjeri: Freenet, Gnutella, KaZaA, BitTorrent
- strukturirani sustavi
  - mrežna topologija je definirana i ima posebnu strukturu
  - podatak ***d*** jedinstveno određuje ključ ***k*** (svaki peer može odrediti ***k*** za ***d***)
  - podatak ***d*** je pohranjen na *peeru* koji je "zadužen" za ključ ***k***, a ne na peeru koji ga kreira
  - primjeri: CAN, Chord, P-Grid, Pastry

Sadržaj predavanja

- Centralizirani i decentralizirani distribuirani sustavi
- Definicija sustava P2P
- Nestrukturirani sustavi P2P
- Strukturirani sustavi P2P

Nestrukturirani sustavi P2P

- podatak (npr. datoteka) je pohranjen na *peeru* koji ga kreira, ne postoji veza između podatka ***d*** i peera ***p***
- moguće je pohraniti kopiju podatka na peerovima koji ga kopiraju s originalnog peera
- pretraživanje se izvodi preplavljivanjem ili slučajnim izborom (random walk), itd.

Obilježja nestrukturiranih sustava P2P

- jednostavnost (jednostavan protokol)
- robustnost (ne postoji jedna točka ispada)
- niska cijena objavljivanja novog podatka (podatak ostaje na peeru koji ga objavljuje)
- velika cijena prilikom pretraživanja, generira se veliki mrežni promet
- dobro rješenje za pronalaženje podataka koji su replicirani na velikom broju *peerova*, ali ne za podatke pohranjene na malome broju *peerova*

Sadržaj predavanja

- Centralizirani i decentralizirani distribuirani sustavi
- Definicija sustava P2P
- Nestrukturirani sustavi P2P
- Strukturirani sustavi P2P

Strukturirani sustavi P2P

- podatak ***d*** je pohranjen na *peeru* koji je “zadužen” za ključ ***k***, a ne na *peeru* koji ga kreira
- svaki *peer* ima jedinstveni identifikator (adresu) ***p*** te je moguće definiranim distribuiranim algoritmom jednoznačno povezati ***p*** i ***k***, a stoga i pronaći ***d***
- sustav P2P implementira metodu `lookup`(***k***) koja vraća identifikator *peera* za dani ključ ***k***

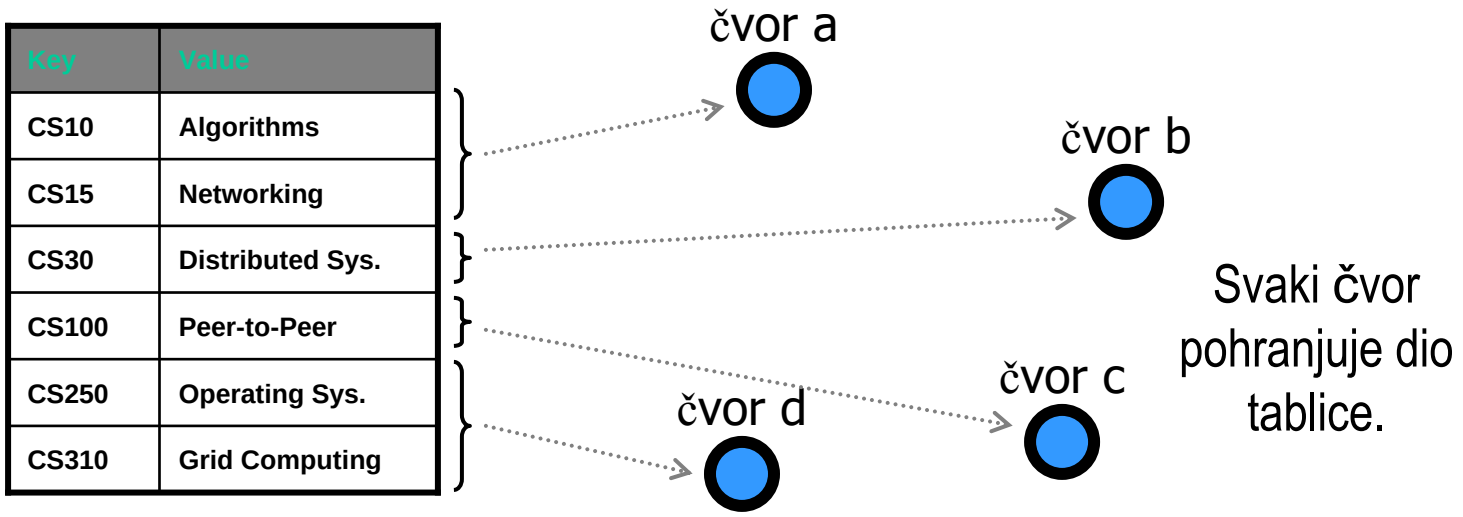
Osobine strukturiranih sustava P2P

- garantira pronalaženje podatka u O(log n) koraka (n je broj *peerova* u mreži)
  - skalabilno rješenje u odnosu na nestrukturirane sustave
- povećana cijena objavljivanja novog podatka u odnosu na nestrukturirane sustave P2P
  - podatak se pohranjuje na peeru koji je za njega “zadužen”
- potrebno je održavati dodatne strukture podataka (tablice usmjeravanja) radi umjeravanje upita prema *peerovima* koji pohranjuju tražene podatke

Raspodijeljena hash tablica - Distributed Hash Table (DHT)

Tipičan primjer strukturiranog sustava P2P

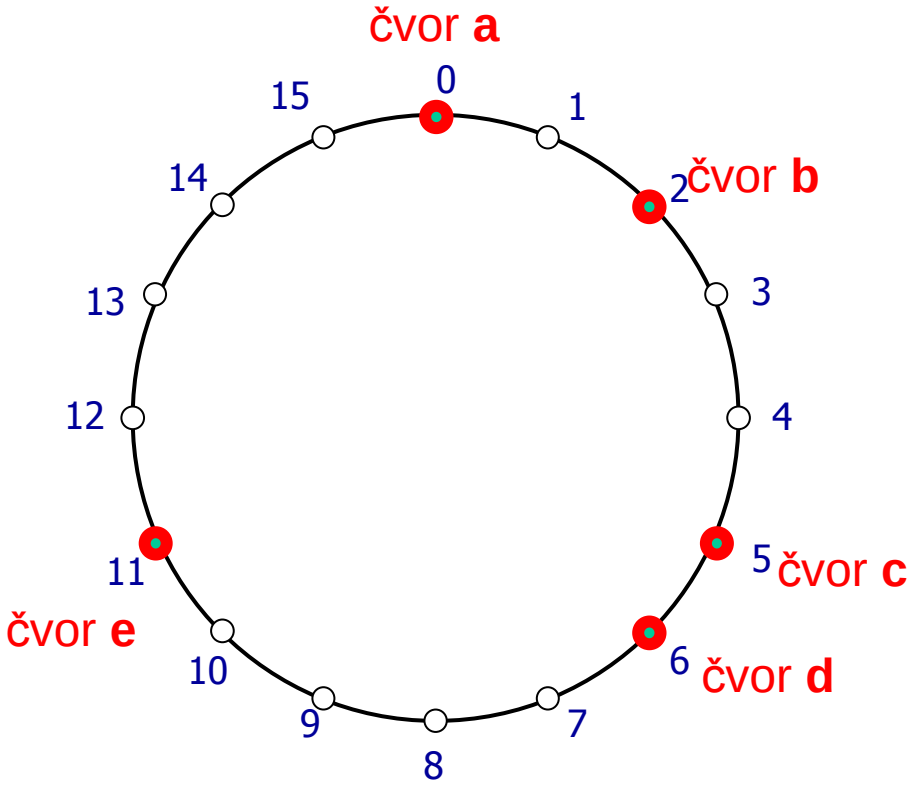
- hash tablica je raspodijeljena na više čvorova.



- *lookup* omogućuje svakom čvoru pronalaženje vrijednosti povezane s nekim ključem
- primjer:

  lookup("*CS30*"), odgovor: *"Distributed Sys."*

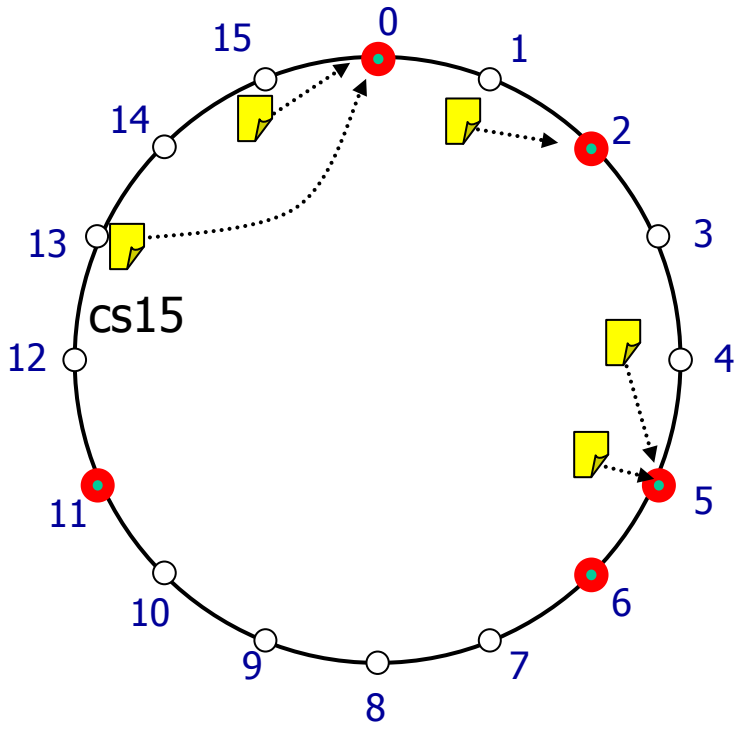
Primjer: Kako raspodijeliti hash tablicu?



Mogući identifikatori čvorova:
{0,1, …, 15} , N = 16

- Primjer mreže s 5 čvorova **{a,b,c,d,e}**
- Koristi se prostor identifikatora veličine N = 16 (dovoljna su 4 bita za kodiranje)
- Čvorovima se jednoznačno pridjeljuju identifikatori uz pomoć posebne funkcije H_1,
  npr $H_1(a) = 0$.

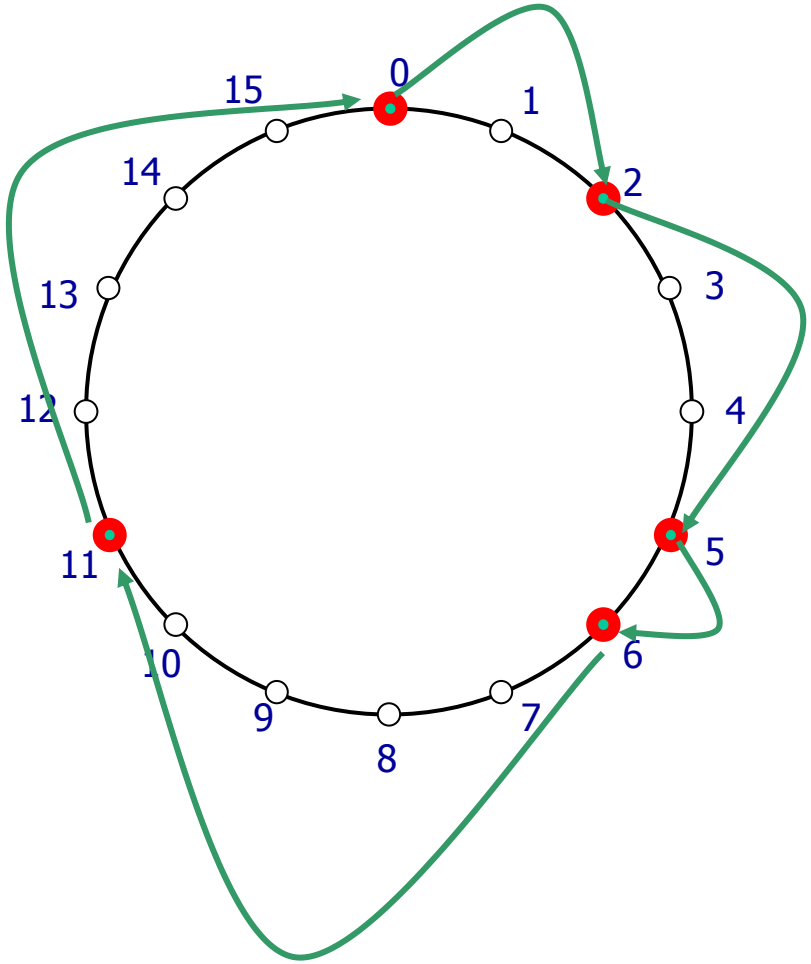
Kako se podatak dodjeljuje čvoru?



- Podacima se pridjeljuju ključevi (tj. identifikatori) iz istog prostora {0,1, …, 15} koristeći funkciju H_2
- Npr. podatak (”**cs15**”, ”**networking**”) dobiva ključ/ identifikator 13 jer vrijedi H_2(”**cs15**”) = 13
- Kako u mreži ne postoji čvor s id = 13, podatak se pohranjuje na prvom sljedećem čvoru (to je u ovom slučaju čvor a kojem je id = 0)

Kako ćemo povezati čvorove?

- Svaki čvor održava jedan pokazivač na sljedbenika, tj. na prvi sljedeći čvor u smjeru kazaljke na satu



sljedbenik čvora 0 → čvor 2

sljedbenik čvora 2 → čvor 5

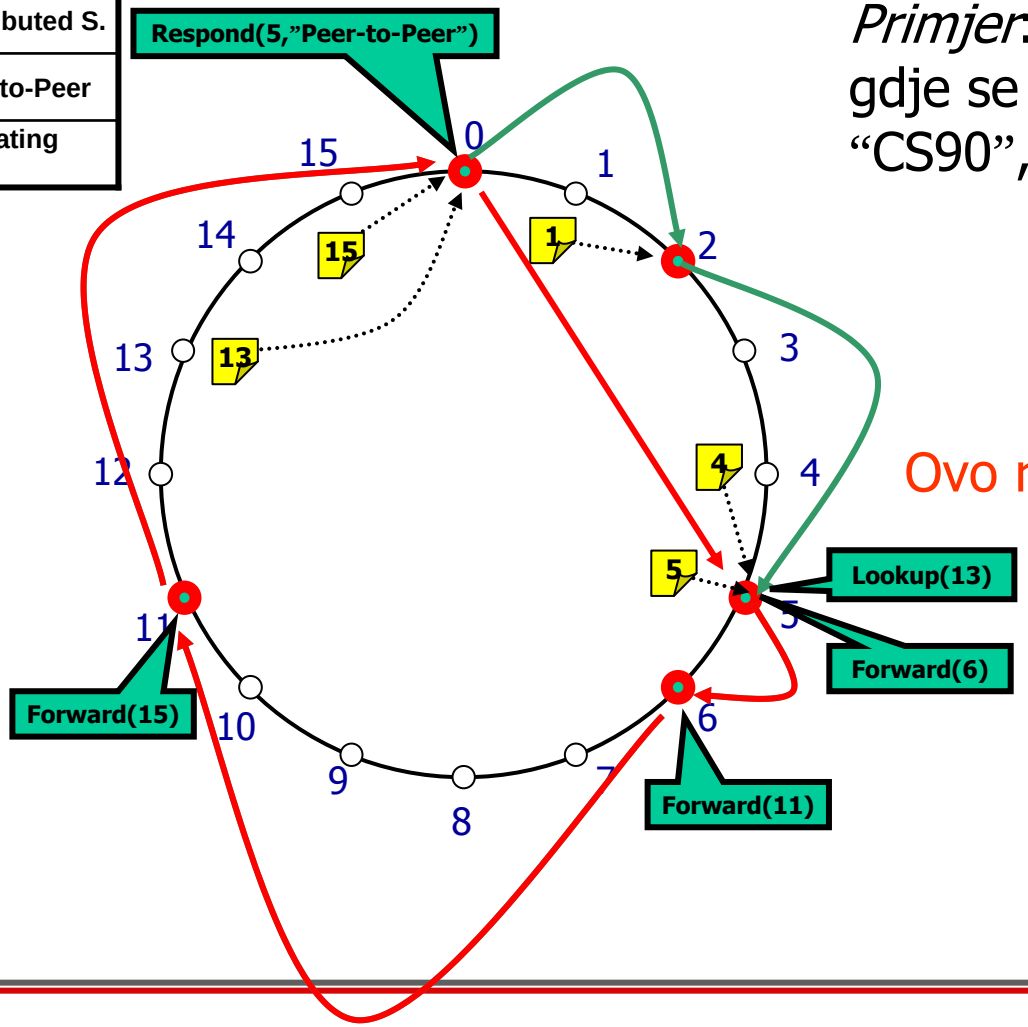
sljedbenik čvora 5 → čvor 6

sljedbenik čvora 6 → čvor 11

sljedbenik čvora 11 → čvor 0

Jednostavno pretraživanje

Key	Value
1 (CS10)	Algorithms
4 (CS15)	Networking
5 (CS30)	Distributed S.
13 (CS90)	Peer-to-Peer
15 (CS95)	Operating Sys.



Primjer: Dolazi upit s čvora 5 gdje se traži vrijednost za ključ "CS90", H_2("CS90")=13

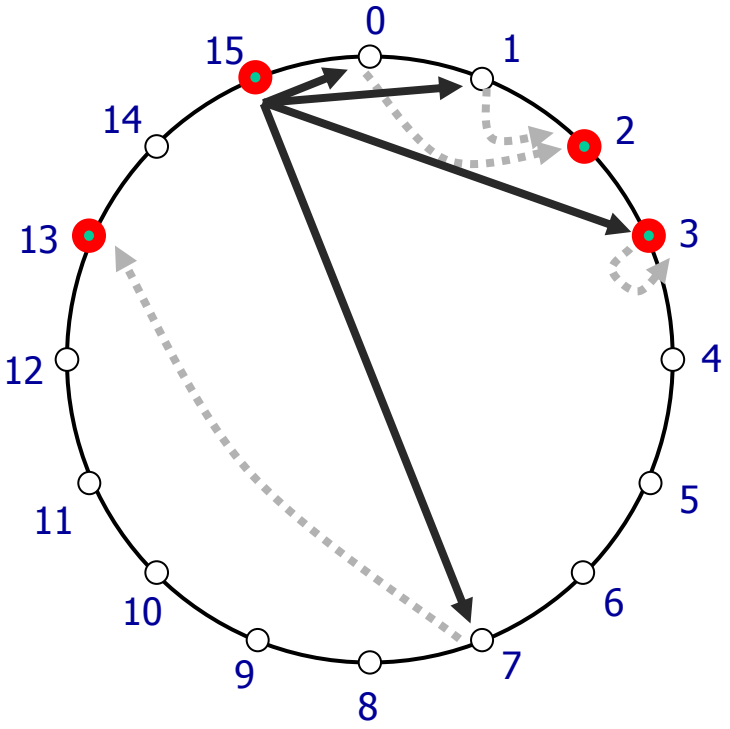
Ovo rješenje je vrlo sporo!

Kako ubrzati pretraživanje (1)

Primjer: tablica usmjeravanja za čvor n=15

$15+2^0=$	0	→ čvor 2
$15+2^1=$	1	→ čvor 2
$15+2^2=$	3	→ čvor 3
$15+2^3=$	7	→ čvor 13

Za čvorove koji ne postoje, pokazivač se postavlja na prvog sljedbenika!

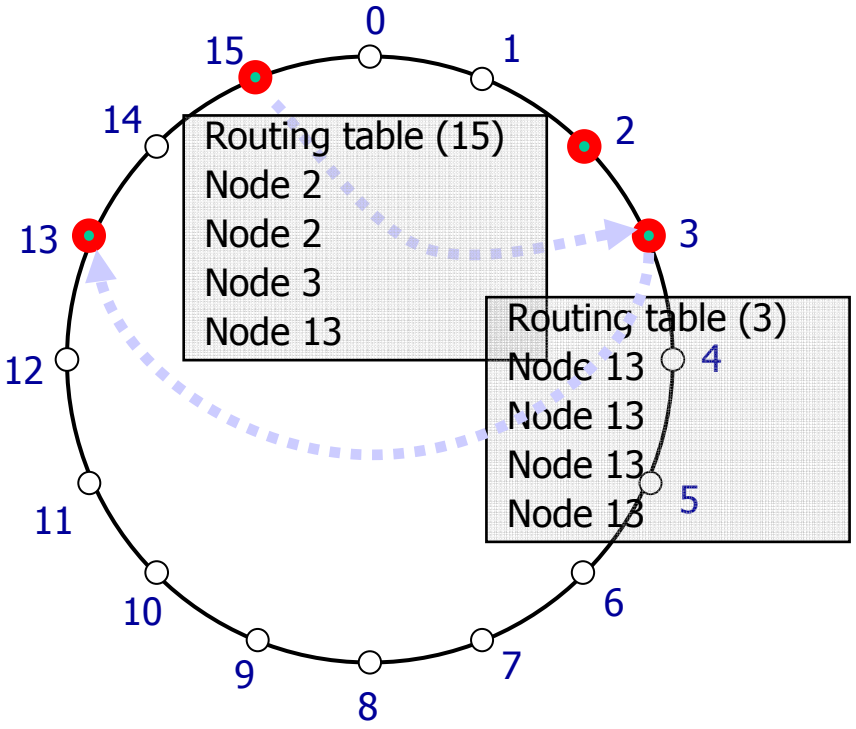


Kako ubrzati pretraživanje (2)

Primjer: na čvoru 15 je postavljen upit za podatkom čiji je identifikator jednak 10, podatak s id=10 je pohranjen na čvoru 13

čvor 15 usmjerava upit od čvora 3 (id 13 je veći od 10!)

čvor 3 usmjerava upit do čvora 13



Literatura

- Eng Keong Lua Crowcroft, J. Pias, M. Sharma, R. Lim, S., A survey and comparison of peer-to-peer overlay network schemes, IEEE Communications Surveys & Tutorials, 7(2), Second Quarter 2005, pp. 72- 93. http://www.cl.cam.ac.uk/teaching/2005/AdvSysTop/survey.pdf

Programski agent

Agent je jedinka koja djeluje u ime svojeg vlasnika ili korisnika:

- sa sposobnošću djelovanja u okružju, promatranja okružja i njegovom djelomičnom predodžbom,
- vođena skupom namjera ili ciljeva,
- s vlastitim resursima i vještinama,
- koja može komunicirati s drugim agentima,
- koja se može reproducirati.

Agenti se dijele na:

- fizikalne ili virtualne,
- ljudske,sklopovske ili programske.

Obilježja agenta:

- inteligencija – znanje, rasuđivanje, učenje
- samostalnost,
- reaktivnost – pobuda iz okružja,
- proaktivnost - usmjerenost cilju,
- pokretljivost,
- komunikativnost, kooperativnost, društvenost, ...

- Informacijski (traženje informacije u mreži: pretraživanje i filtriranje, snabdijevanje informacijama, savjetovanje i fokusiranje, …).
- Kooperacijski (rješavanje složenog problema: skupni posao, zabava, upravljanje mrežom/komunikacijom, …).
- Transakcijski (obrada i nadzor transakcije: e-trgovina, poslovni procesi, proizvodni procesi, …).

Jedan agent ili višeagentski sustav!

Mikro razina

- Dva ili nekoliko agenata u međusobnoj interakciji

Skupina

- Više agenata s različitim ulogama

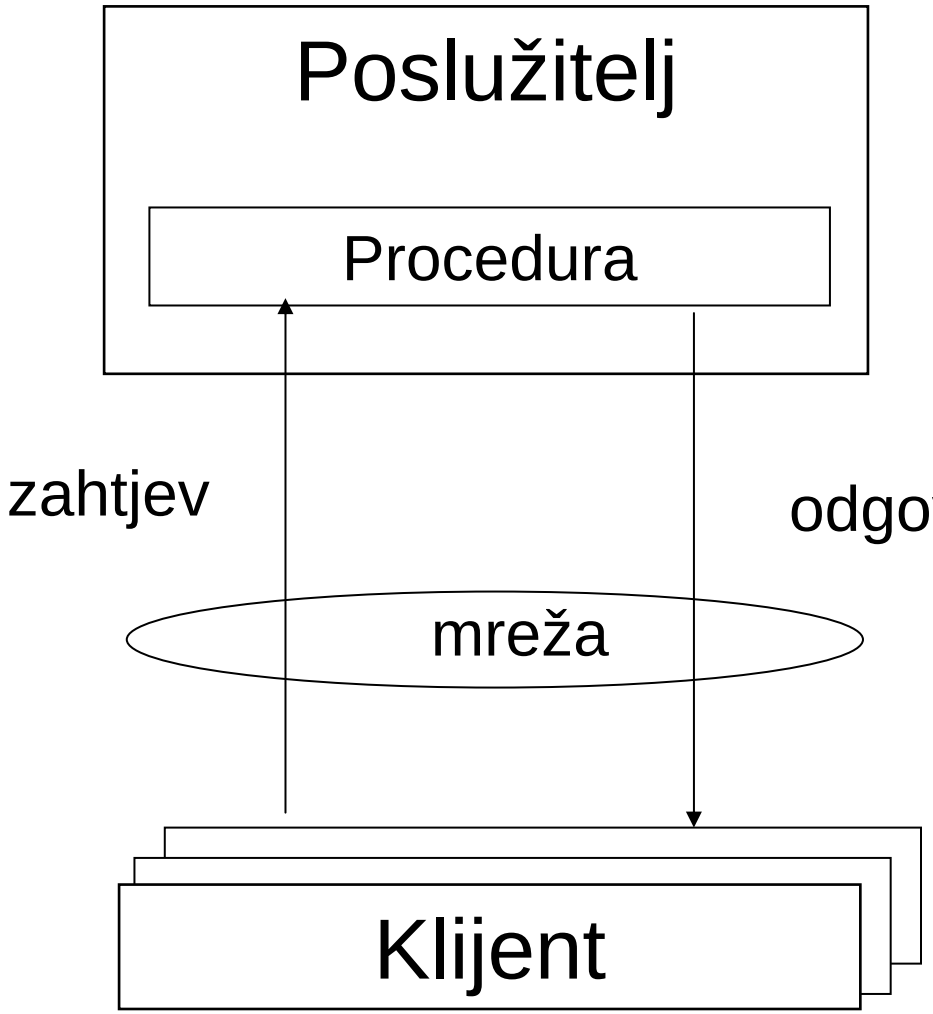
Društvo

- Mnogo agenata u dinamičnim odnosima

Uzor: sociološki (npr. tim) ili
biološki (npr. kolonija mrava)

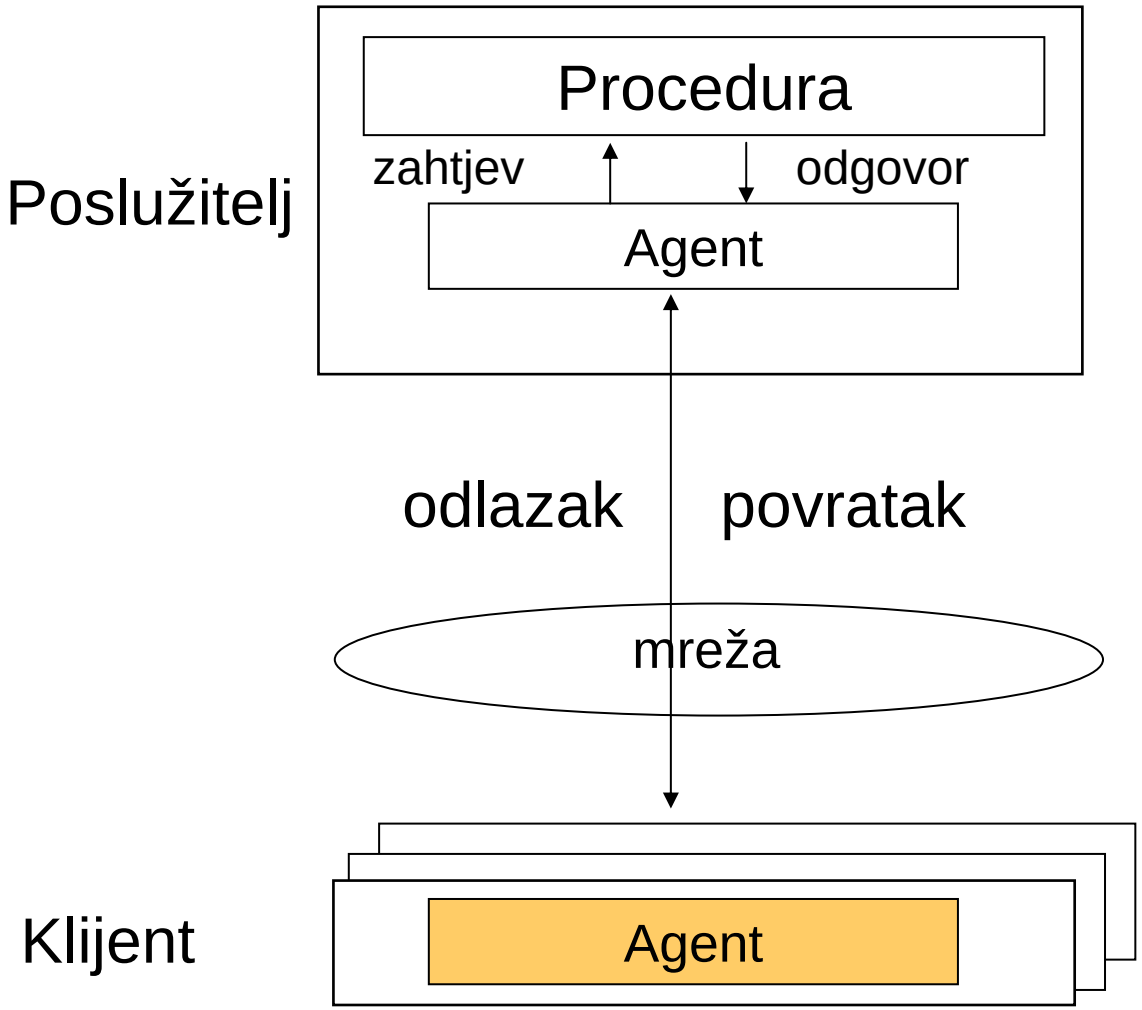
Pokretni agent:

- program koji predočuje korisnika u mreži i može migrirati samostalno iz čvora u čvor zbog izvedbe neke obrade u ime korisnika.
- Agentova aplikacija opisuje neki posao, ubacuje agenta u mrežu dopuštajući mu kretanje i povratak u početni čvor po obavljenom poslu.
- Agent se može kretati mrežom po prethodno određenom putu ili na temelju dinamički prikupljene informacije.



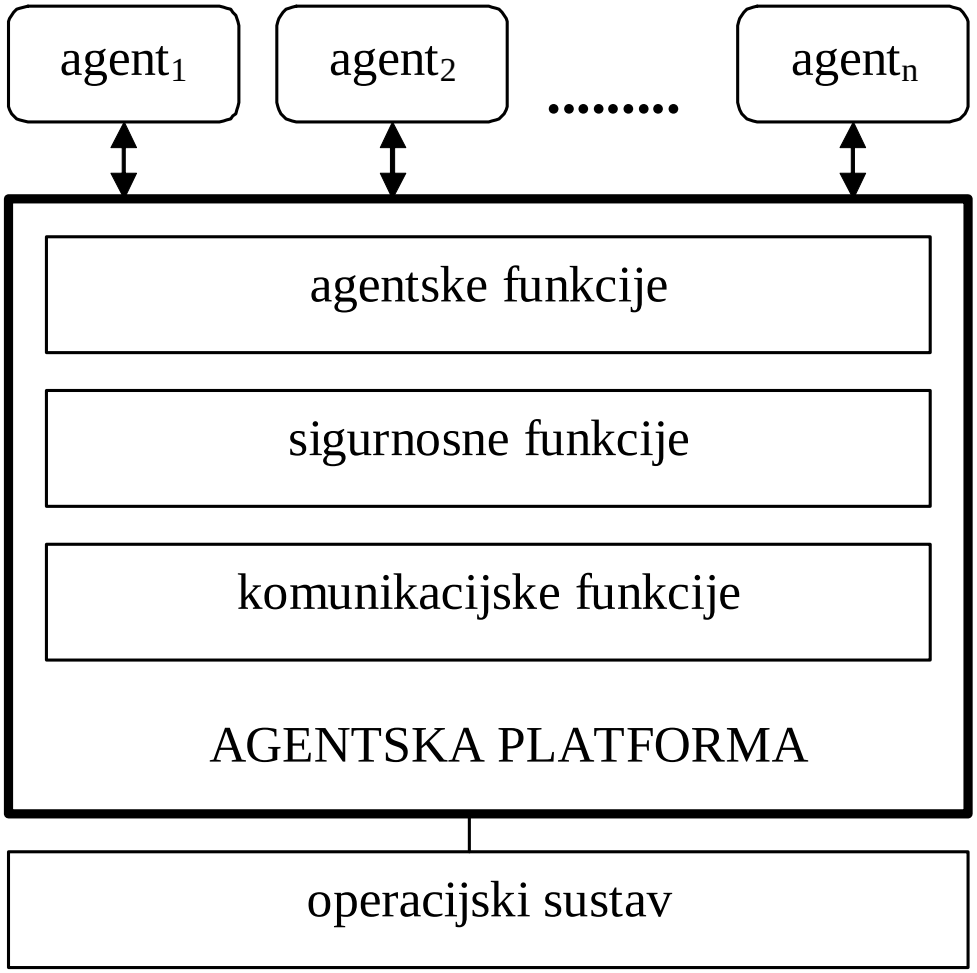
Programski model:

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



Programski model:

Udaljeno programiranje (RP – *Remote Programming*)



- Operacijski sustav
- Agentska platforma (osnovna programska oprema za pokretljivost) koja sadrži tri sloja s agentskim, sigurnosnim i komunikacijskim funkcijama,
- Agenti

- Agentski sloj (izvođenje i nadzor aktivnih agenata, snabdijevanje agenata s funkcijama za pokretljivost, komunikaciju, identifikaciju, ...);
- Sigurnosni sloj (zaštita od promjene/čitanja i nedopuštenog ulaza u sustav, funkcije kriptografije, digitalnog potpisa i certifikacije, zaštitna pregrada, ...),
- Komunikacijski sloj (protokoli, formati, RP, serijalizacija/deserijalizacija, ...)

Primjer:

- JADE – Java Agent DEvelopmemt Framework (jade.tilab.com)

Identifikacija agenta

- identify (*agent-id*, *personal-key*, *agent-type*)
  - *agent-id* – identifikator agenta
  - *personal-key* – osobni ključ agenta
  - *agent-type* – vrsta agenta

Traženje raspoloživog agenta

- getAvailableAgent (*agent-id*, *agent-type*)

Kontaktiranje agenta

- contact (*agent-id*, *agent-id*)

Odašiljanje naloga

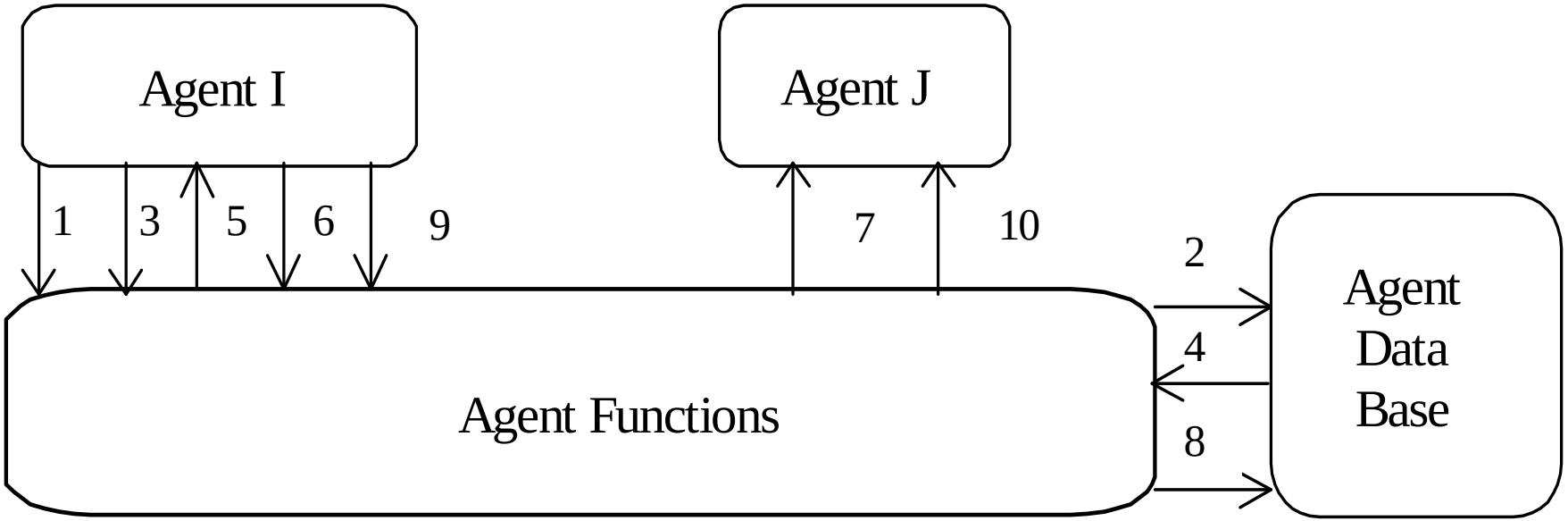
- sendCommand (*agent-id*, *agent-id*, *command*)
  - *command* - nalog

Premještanje (migracija) agenta

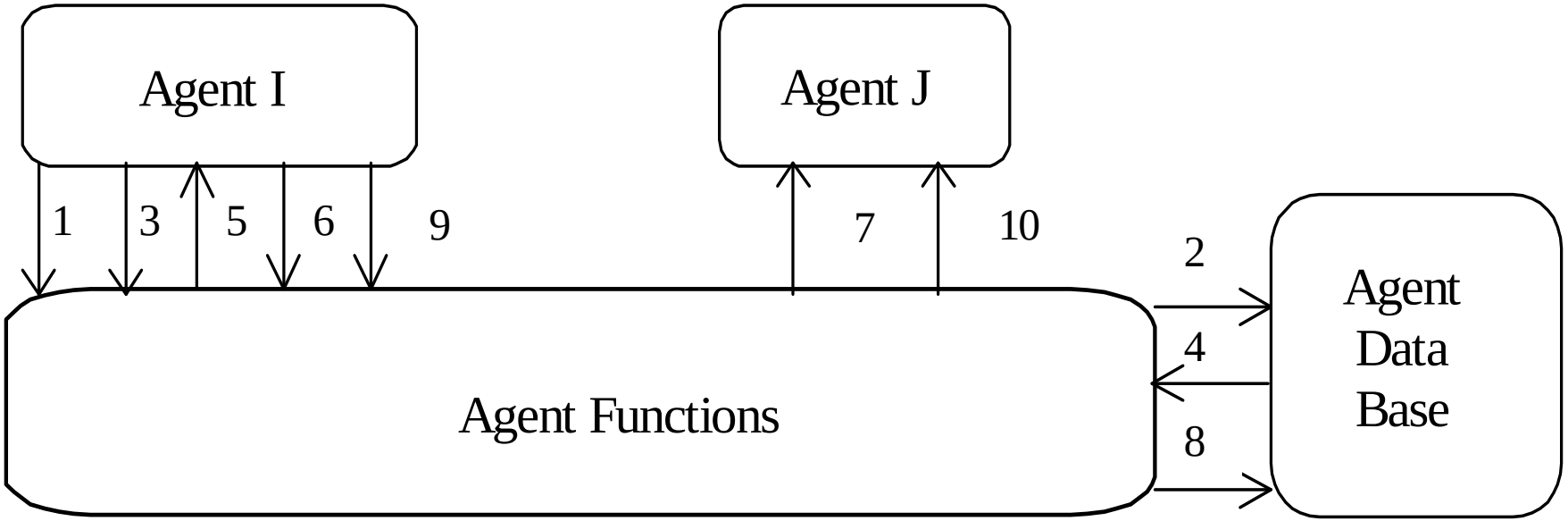
- move (*agent-id*, *address*)

Izmjena poruka

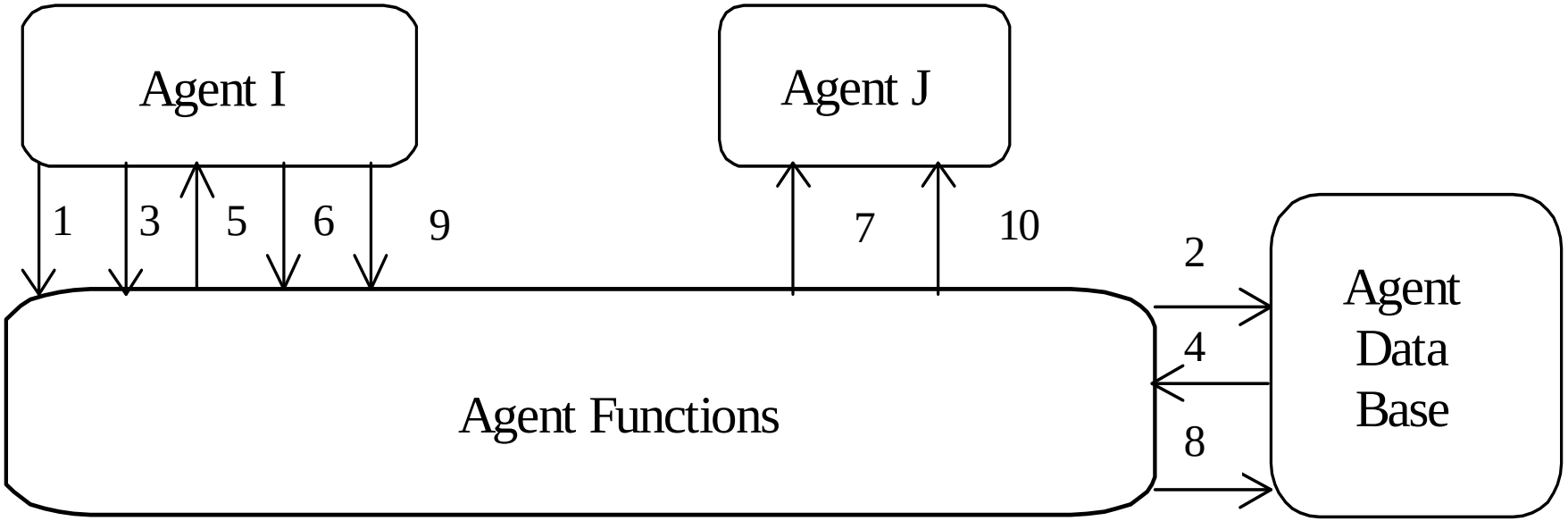
- sendMessage (*agent-id*, *agent-id/agent-type*, (*address*), *message*)
  - *address* – adresa mrežnog čvora
  - *message* - poruka



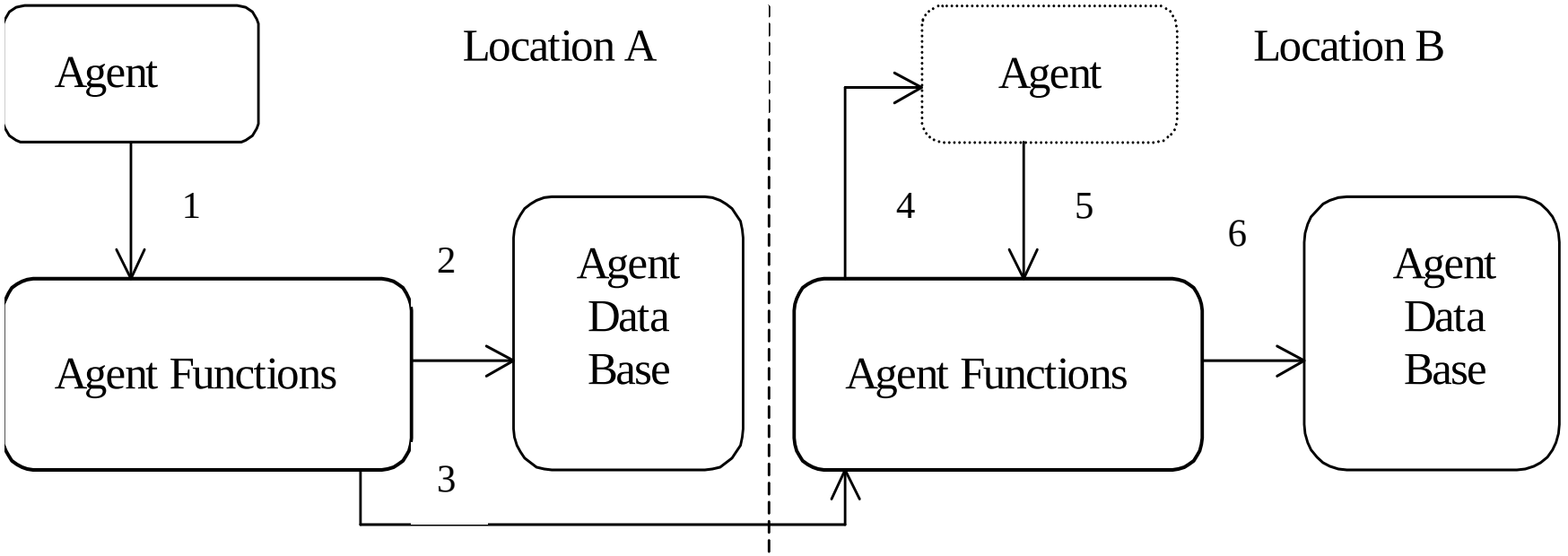
1. Agent I identificira samog sebe s *agent-id=I*, *personal-key=pkI* i *agent-type=atI* s naredbom *identify* (*I*, *pkI*, *atI*).
2. Agentski sloj provjerava identifikacijsku informaciju agenta I u lokalnoj bazi agenata.
3. Agent I traži drugog raspoloživog agenta definirajući njegov *agent-type=atx* i pokreće naredbu *getAvailableAgent* (*I*, *atx*).



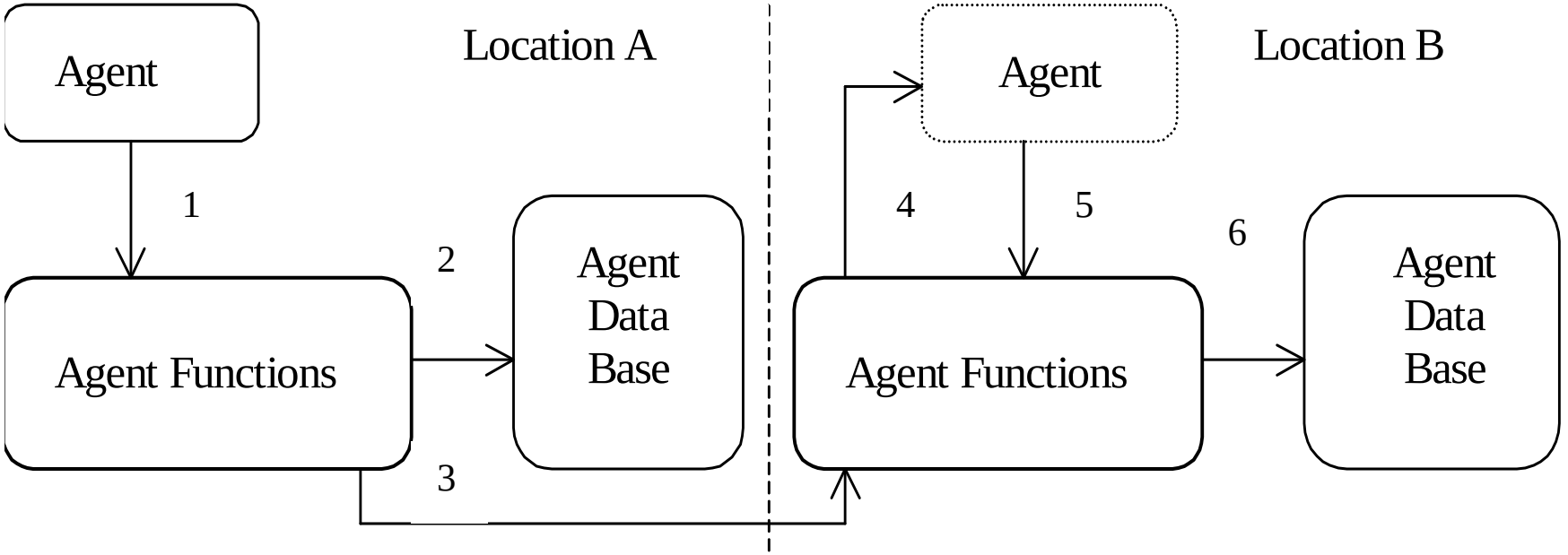
4. Agentski sloj pronalazi agenta partnera J vrste *atx* s njegovim *agent-id=J*.
5. Agentski sloj informira o agentu J s njegovim *agent-id=J*.
6-7. Agent I kontaktira agenta J naredbom *contact* (*I*, *J*).



8. Agentski sloj označava agenta J kao zauzetog u lokalnoj bazi agenata.
9. Agent I definira zahtijevani nalog *command* i šalje ga agentu J naredbom *sendCommand* (*I*, *J*, *command*).
10. Agent J prima zahtijevani nalog s naredbom *sendCommand* (*I*, *J*, *command*).

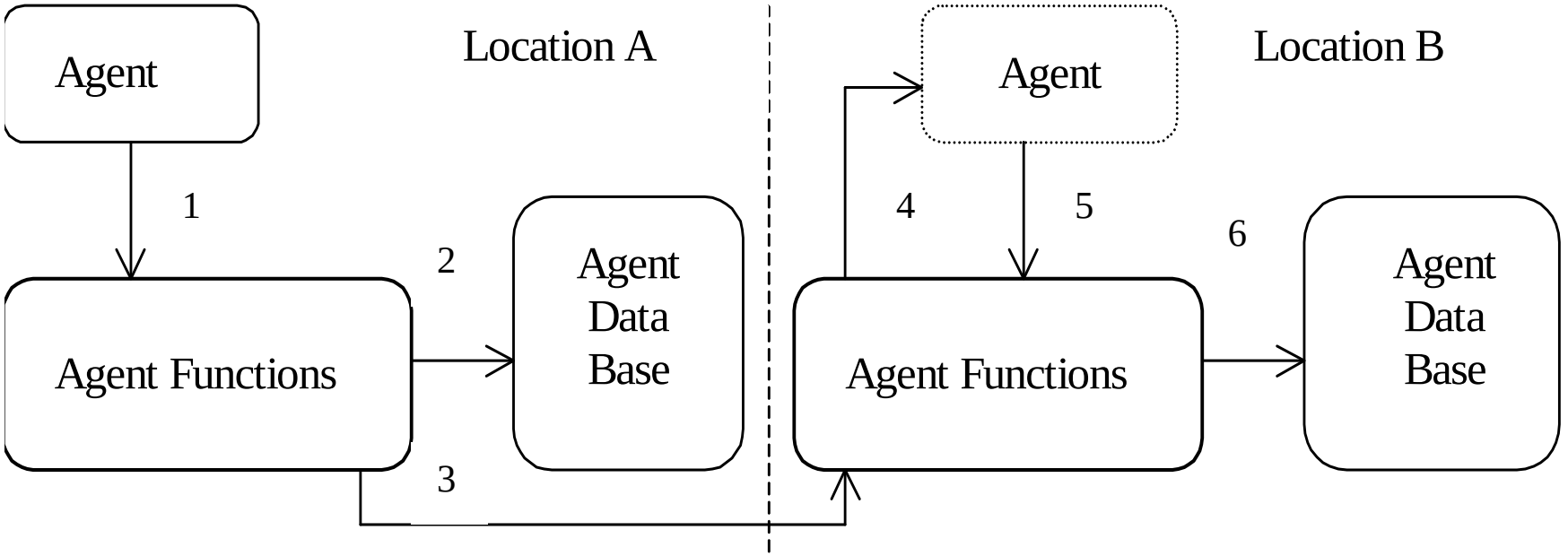


1. Agent I identificira samog sebe s *agent-id=I*, *personal-key=pkI* i *agent-type=atI* s naredbom *identify* (*I*, *pkI*, *atI*).
2. Agentski sloj provjerava identifikacijsku informaciju agenta I u lokalnoj bazi agenata na izvorištu A.

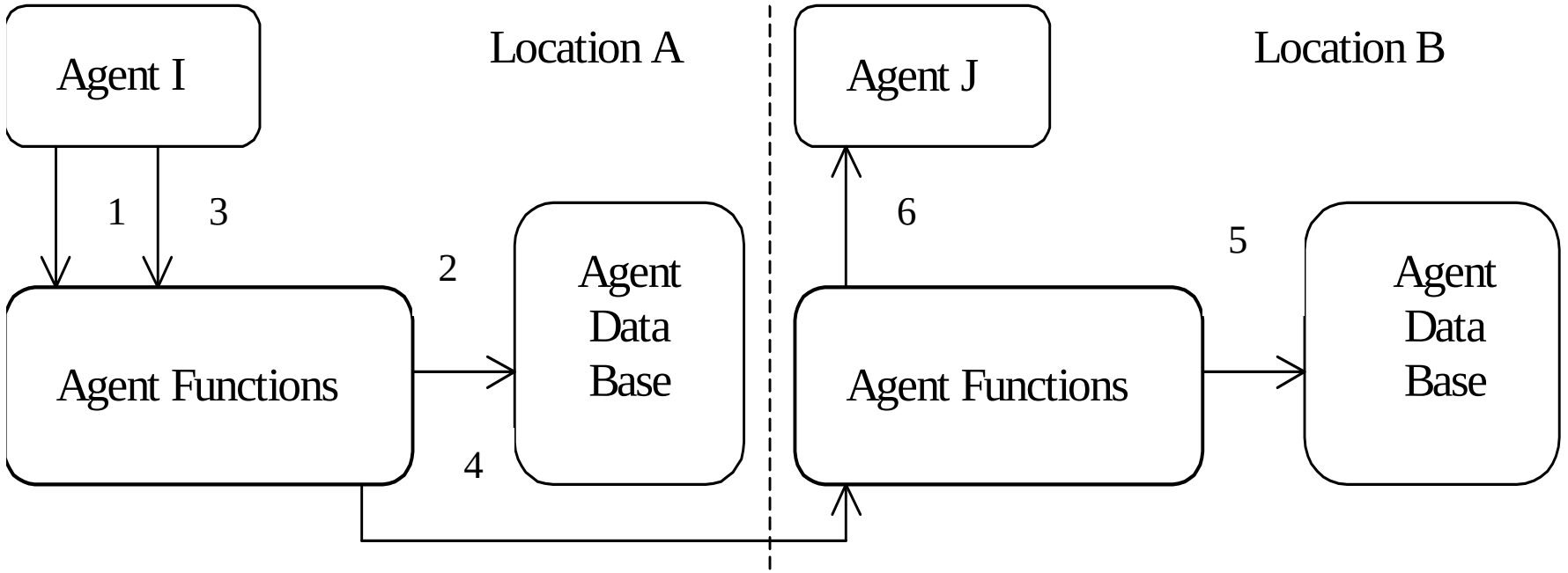


3-4. Agentski slojevi na izvorišnoj lokaciji A i odredišnoj lokaciji B izvode naredbu *move* (*I*, *B*) kojom se provodi migracija agenta I. Agent I se obnavlja na lokaciji B.

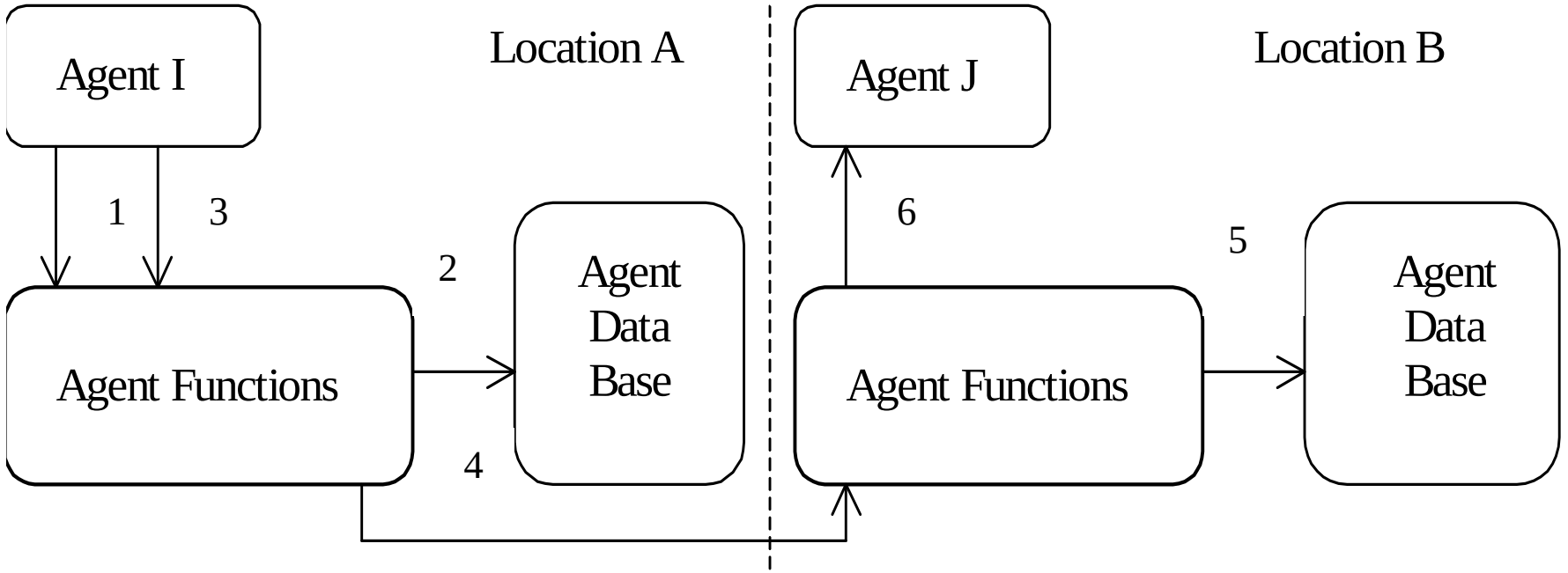
5-6. Agent I identificira samog sebe s *agent-id=I*, *personal-key=pkI* i *agent-type=atI* s naredbom *identify* (*I*, *pkI*, *atI*) u lokalnoj bazi agenata na odredištu B.



1. Agent I identificira samog sebe s *agent-id=I*, *personal-key=pkI* i *agent-type=atI* s naredbom *identify* (*I*, *pkI*, *atI*).
2. Agentski sloj provjerava identifikacijsku informaciju agenta I u lokalnoj bazi agenata.



3. Agent I pripravlja poruku *message* koju će poslati poznatom udaljenom agentu J na lokaciji *address=B* s naredbom *sendMessage* (*I*, *J*, *B*, *message*).
4. Agentski sloj na izvorišnoj lokaciji A šalje naredbu agentu J na odredišnoj lokaciji B.



5. Agentski sloj označava agenta J kao zauzetog u lokalnoj bazi agenata.
6. Agent J prima poruku prenesenu naredbom *sendMessage* (*I*, *J*, *B*, *message*).

- FIPA (*The Foundation for Intelligent Physical Agents*), standardizacijsko tijelo o okviru IEEE Computer Society (www.fipa.org)
- Mobile Agent Reseach Activities (agents.tel.fer.hr)

- FER kolegiji
  - **Pokretni programski agenti**

    2. sem, dipl., T&I
  - **Konkurentno programiranje**

    2. sem, dipl., T&I