



Diplomski studij

**Informacijska i
komunikacijska tehnologija:**

Telekomunikacije i informatika

Računarstvo:

Programsko inženjerstvo i
informacijski sustavi

Računarska znanost

Ak.g. 2008./2009.

Raspodijeljeni sustavi

9.

Replikacija i konzistentnost podataka

13.11.2008.

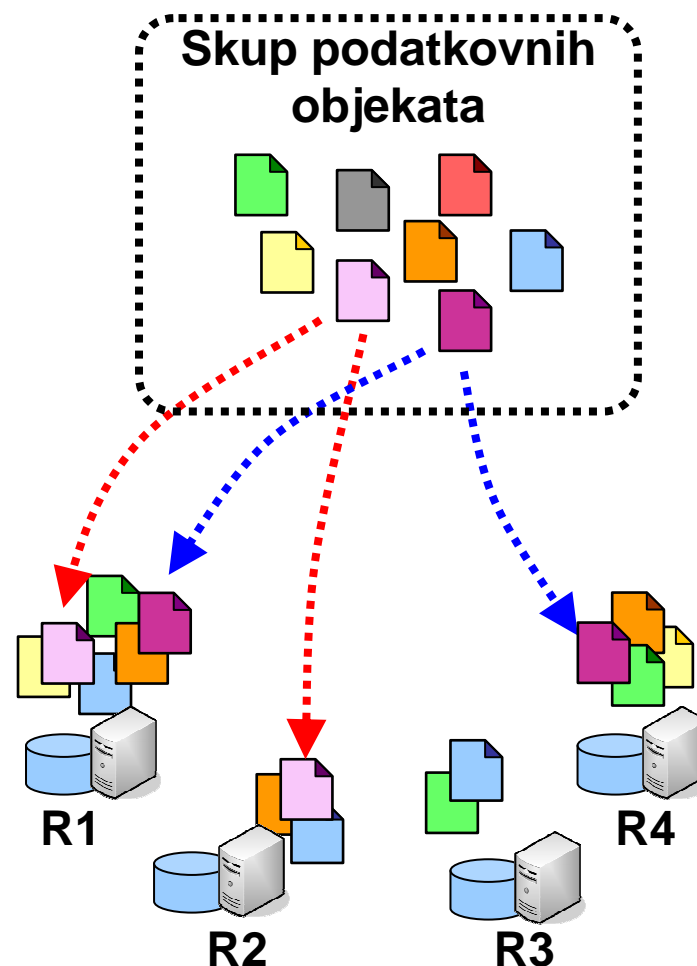
- ◆ Replikacija i konzistentnost podataka
- ◆ Svrha replikacije podataka
- ◆ Primjena dijeljenog spremničkog prostora
- ◆ Modeli održavanja konzistentnosti podataka
- ◆ Modeli uspostave replikacije podataka

Replikacija i konzistentnost podataka



◆ Što je replikacija podataka u raspodijeljenim sustavima?

- ◆ Podatci u raspodijeljenom sustavu ostvareni kao skup podatkovnih objekata postavljenih na računala
- ◆ Podatkovni objekti su spremjeni u obliku nekoliko kopija (**replika**) na različitim računalima
- ◆ Replikacija je postupak stvaranja i upravljanja kopijama podatkovnih objekata

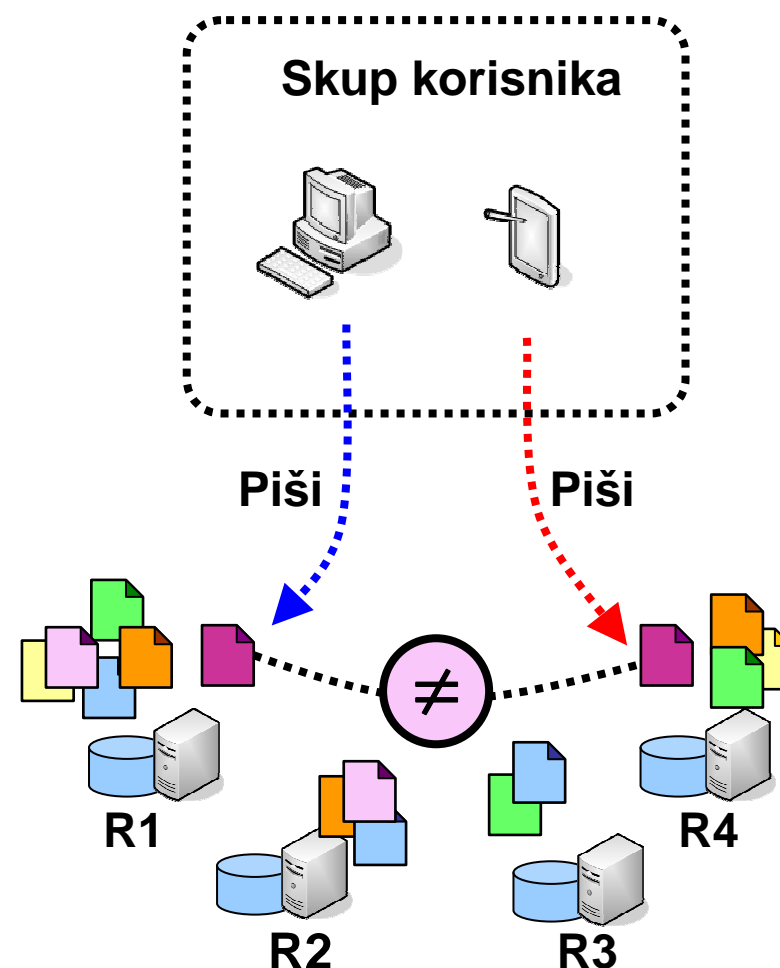


Replikacija i konzistentnost podataka



◆ Što je konzistentnost podataka u raspodijeljenim sustavima?

- ◆ Replike istog objekta mogu istodobno i nezavisno koristiti različiti korisnici
- ◆ Korisnici mogu stanje replika istog objekta mijenjati na različite načine
- ◆ Konzistentnost je **narušena** kada postoje replike nekog objekta koje nemaju ekvivalentno stanje



◆ Pouzdanost podataka

- ◆ U slučaju da neka od replika postane nedostupna, sustav prosljeđuje zahtjeve preostalim dostupnim replikama
- ◆ U slučaju da neka od replika ima pogrešne zapise, usporedbom zapisa više replika ostvaruje se otpornost na pogreške u zapisima

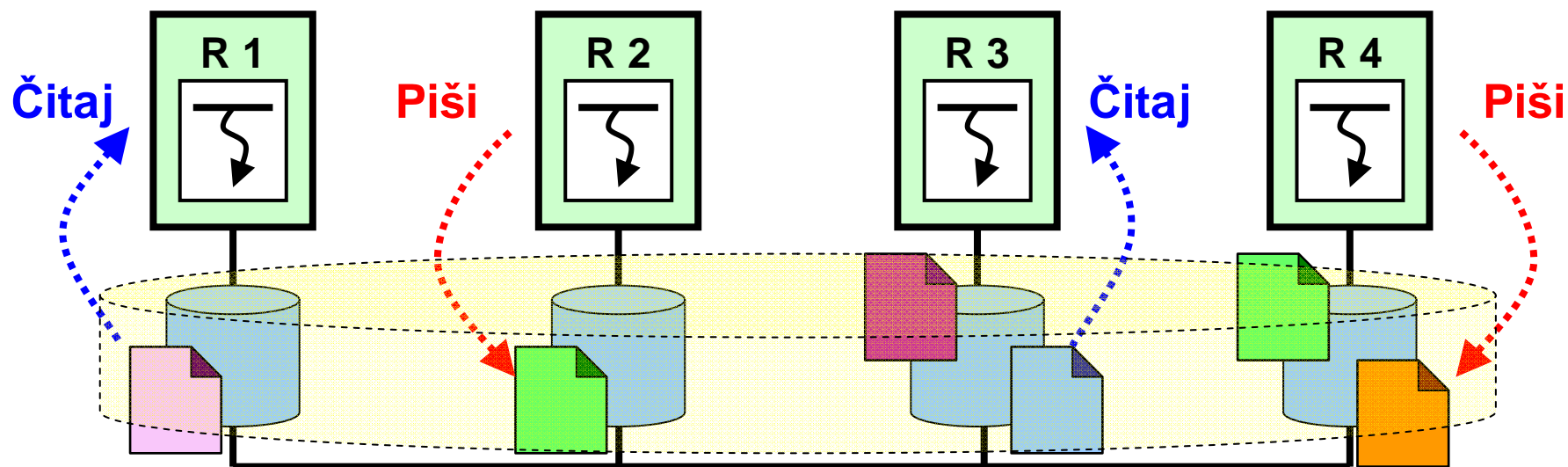
◆ Učinkovito posluživanje podatka

- ◆ U slučaju da neka od replika je preopterećena, pristigli zahtjevi prosljeđuju se ostalim replikama
- ◆ Replike je moguće predodrediti za posluživanje različitih razreda zahtjeva

Model dijeljenog spremničkog prostora



- ◆ Skup spremničkih prostora na računalima u raspodijeljenoj okolini
- ◆ Osnovne operacije
 - ◆ Čitanje podataka (**W**)
 - ◆ Pisanje podataka (**R**)



- ◆ Replikacija i konzistentnost podataka
- ◆ Svrha replikacije podataka
- ◆ Primjena dijeljenog spremničkog prostora
- ◆ Modeli održavanja konzistentnosti podataka
- ◆ Modeli uspostave replikacije podataka

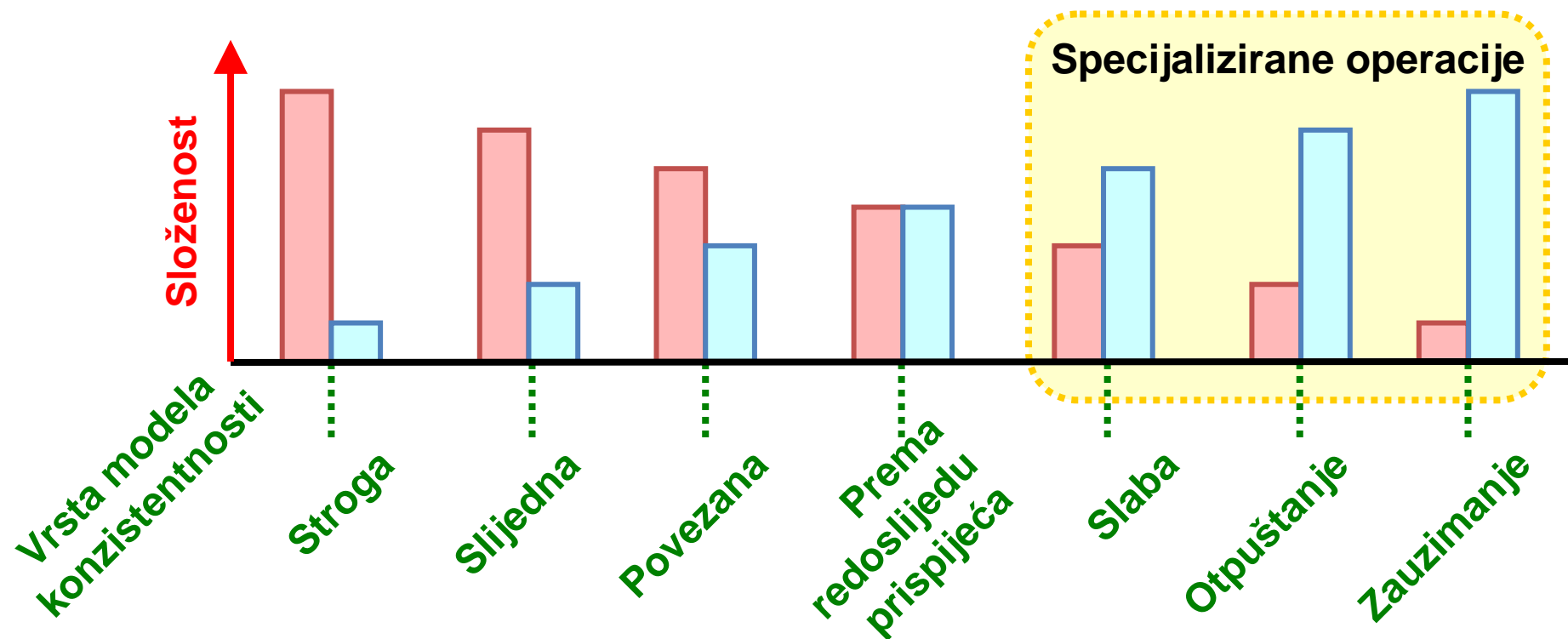
Razredba modela konzistentnosti



Složenost ostvarenja modela
u raspodijeljenoj okolini



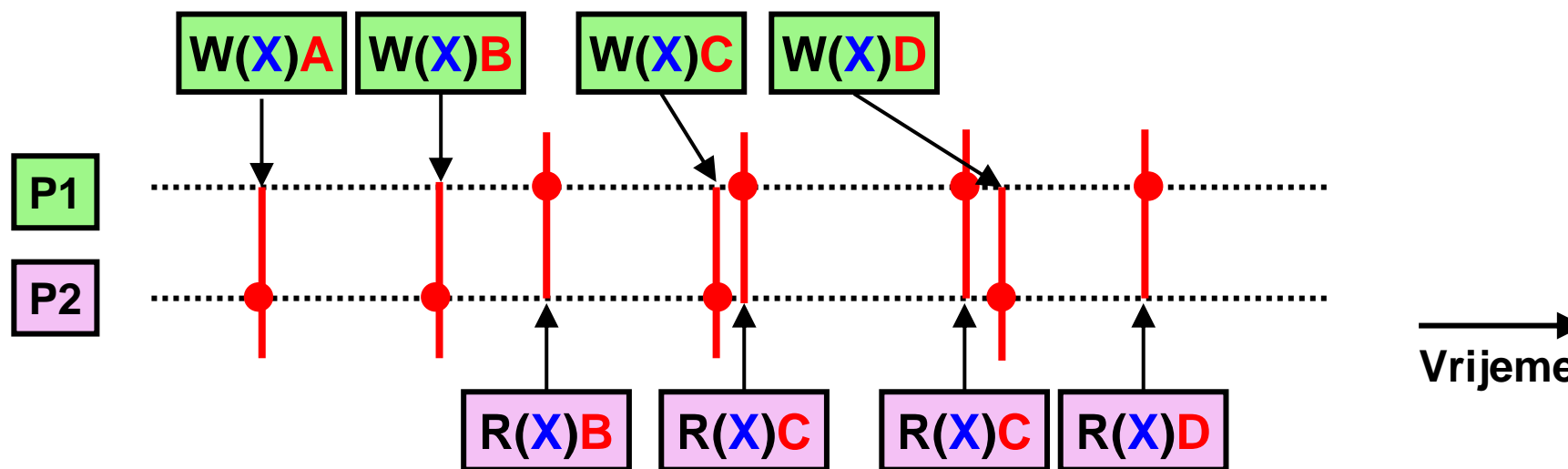
Složenost primjene modela
tijekom razvoja sustava



Stroga konzistentnost



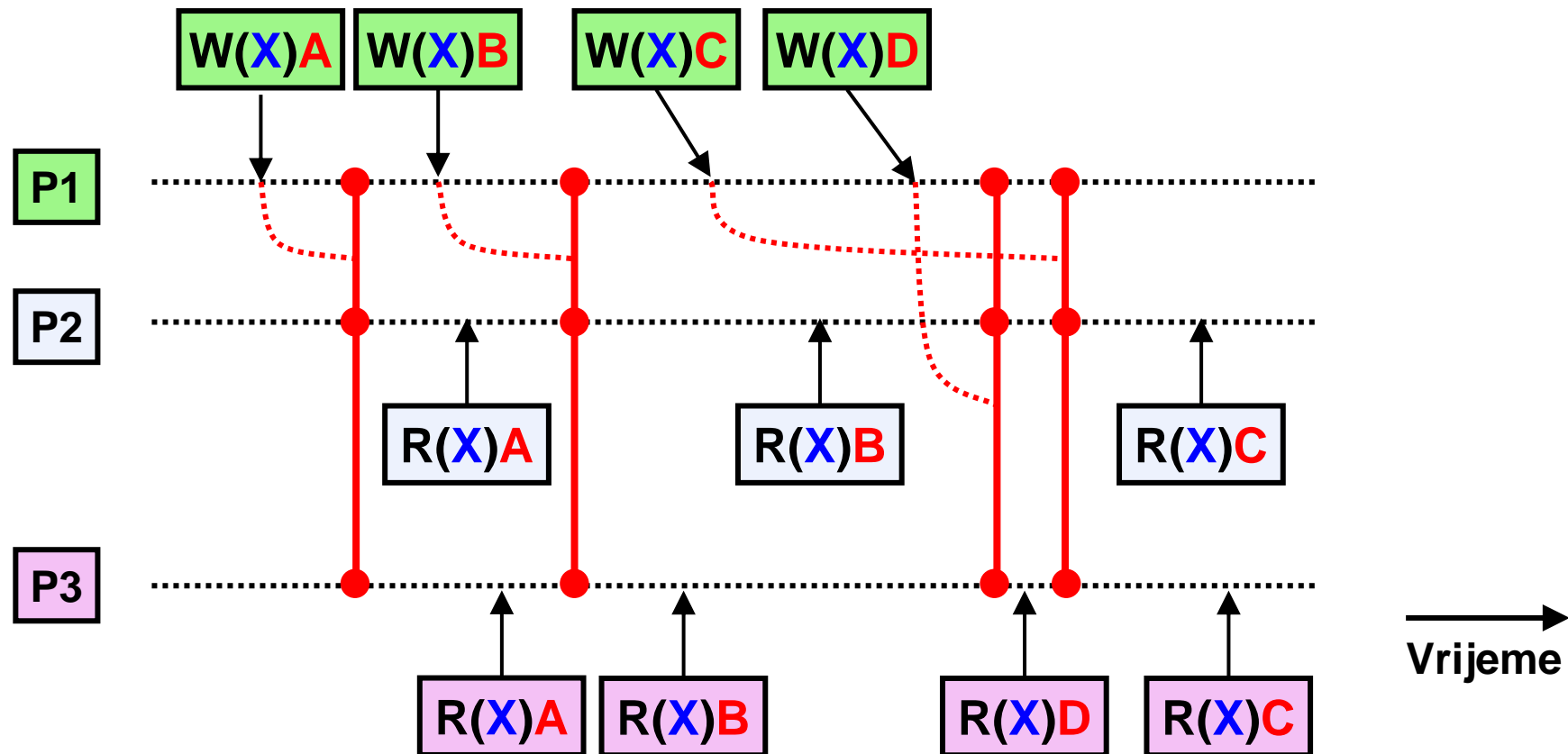
- ◆ Čitanje podatka na lokaciji **X** daje vrijednost koja je posljednja bila zapisana na lokaciju **X**
- ◆ Značajke modela
 - ◆ Primjena globalnog vremena
 - ◆ Ostvaren po uzoru na jednoprocesorske sustave



- ◆ Redoslijed izvođenja operacija može biti **proizvoljan** ali svi procesi moraju vidjeti **jednak** slijed izvođenja akcija u vremenu

- ◆ Značajke modela
 - ◆ Nije potrebno održavati globalni tijek vremena
 - ◆ Proces se moraju dogovoriti o globalnom redoslijedu izvođenja akcija u vremenu

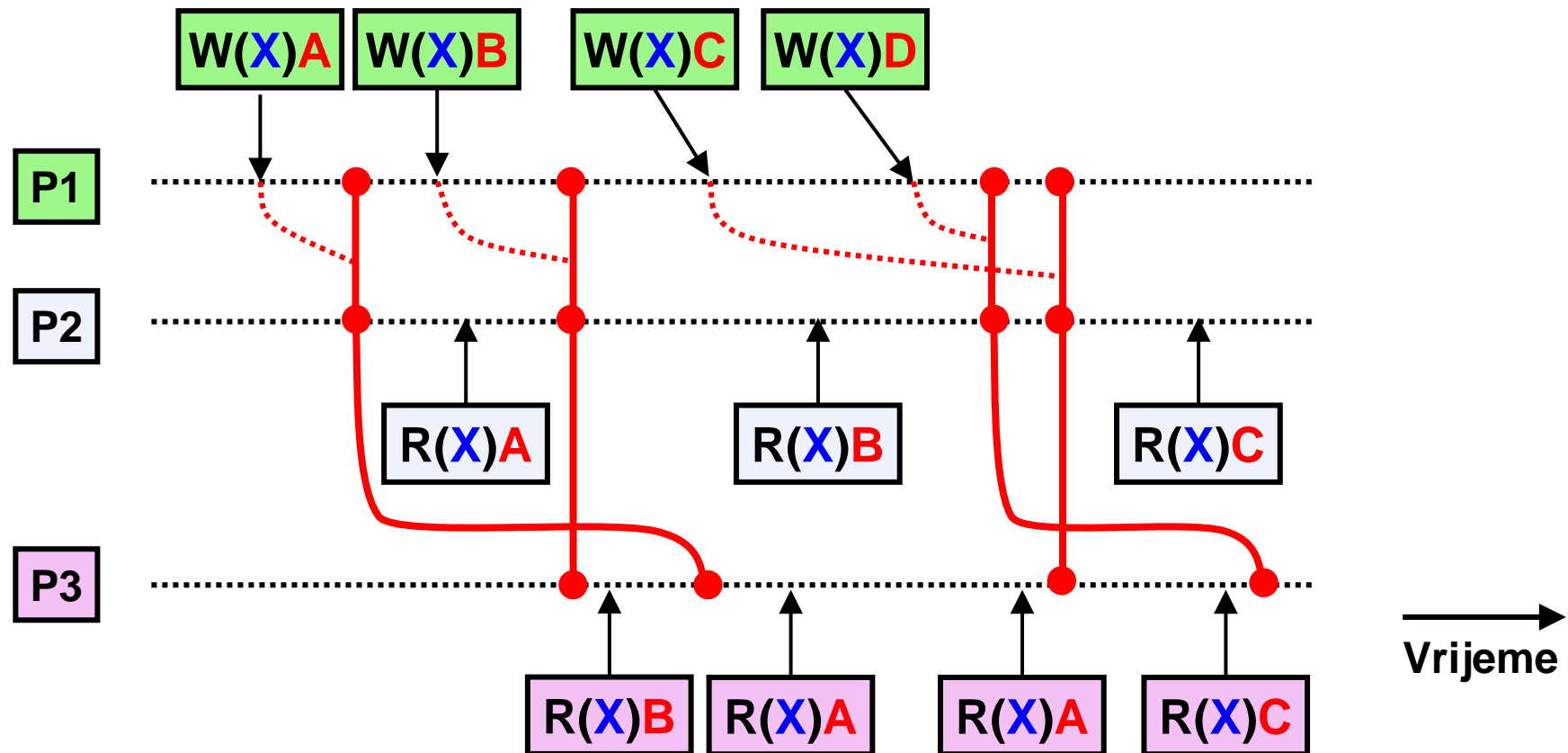
Primjer slijedne konzistentnosti



Čitanje P2: $A \Rightarrow B \Rightarrow D \Rightarrow C$

Čitanje P3: $A \Rightarrow B \Rightarrow D \Rightarrow C$

Primjer slijedne nekonzistentnosti



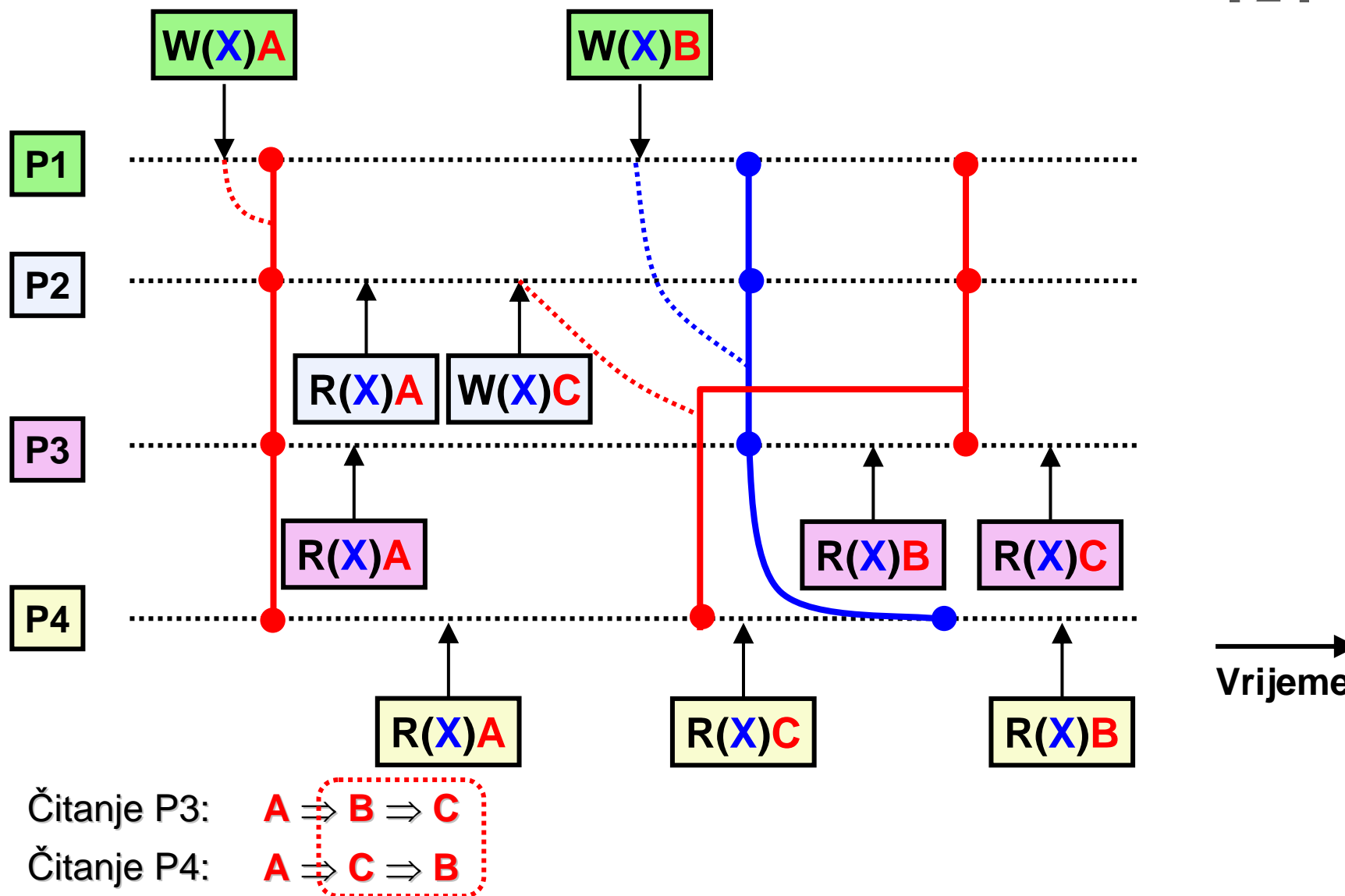
Čitanje P2: $A \Rightarrow B \Rightarrow D \Rightarrow C$

Čitanje P3: $B \Rightarrow A \Rightarrow A \Rightarrow C$

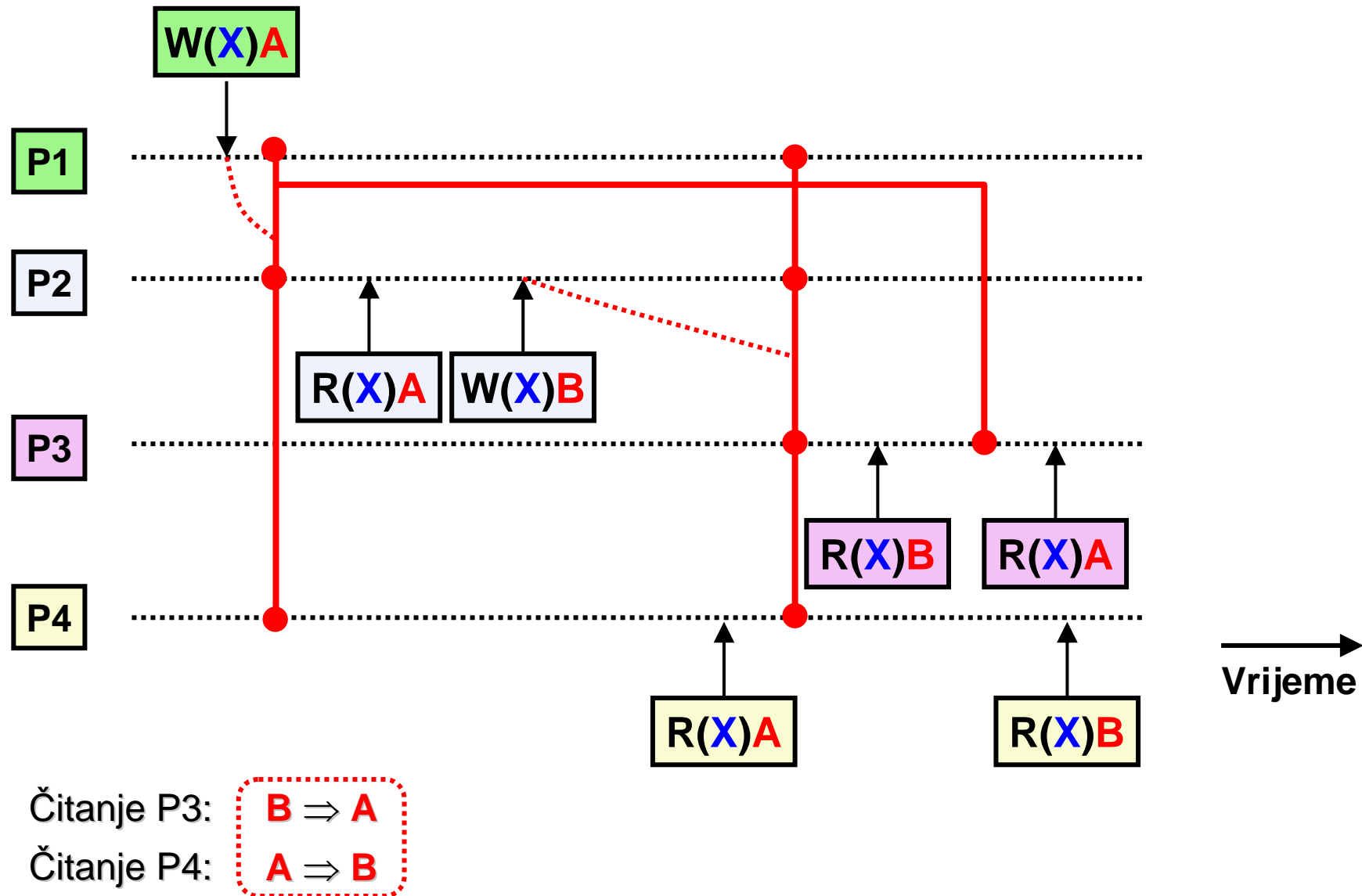
- ◆ Redoslijed izvođenja **povezanih** operacija pisanja vidljiv je svim procesima na jednak način, dok je redoslijed izvođenja operacija pisanja koje **nisu povezane** svakom procesu može biti prikazan na drugačiji način

- ◆ **Povezanost operacija**
 - ◆ Operacija pisanja **W** sadržaja u lokaciju **X** odvija se u vremenu prije operacije čitanja sadržaja **R** iz lokacije **X** čime je operacija **R** povezana s operacijom **W**
 - ◆ Dvije operacije pisanja nisu povezane ako ostvaruju istodobno zapisivanje sadržaja u različite lokacije dijeljenog spremnika

Primjer povezane konzistentnosti



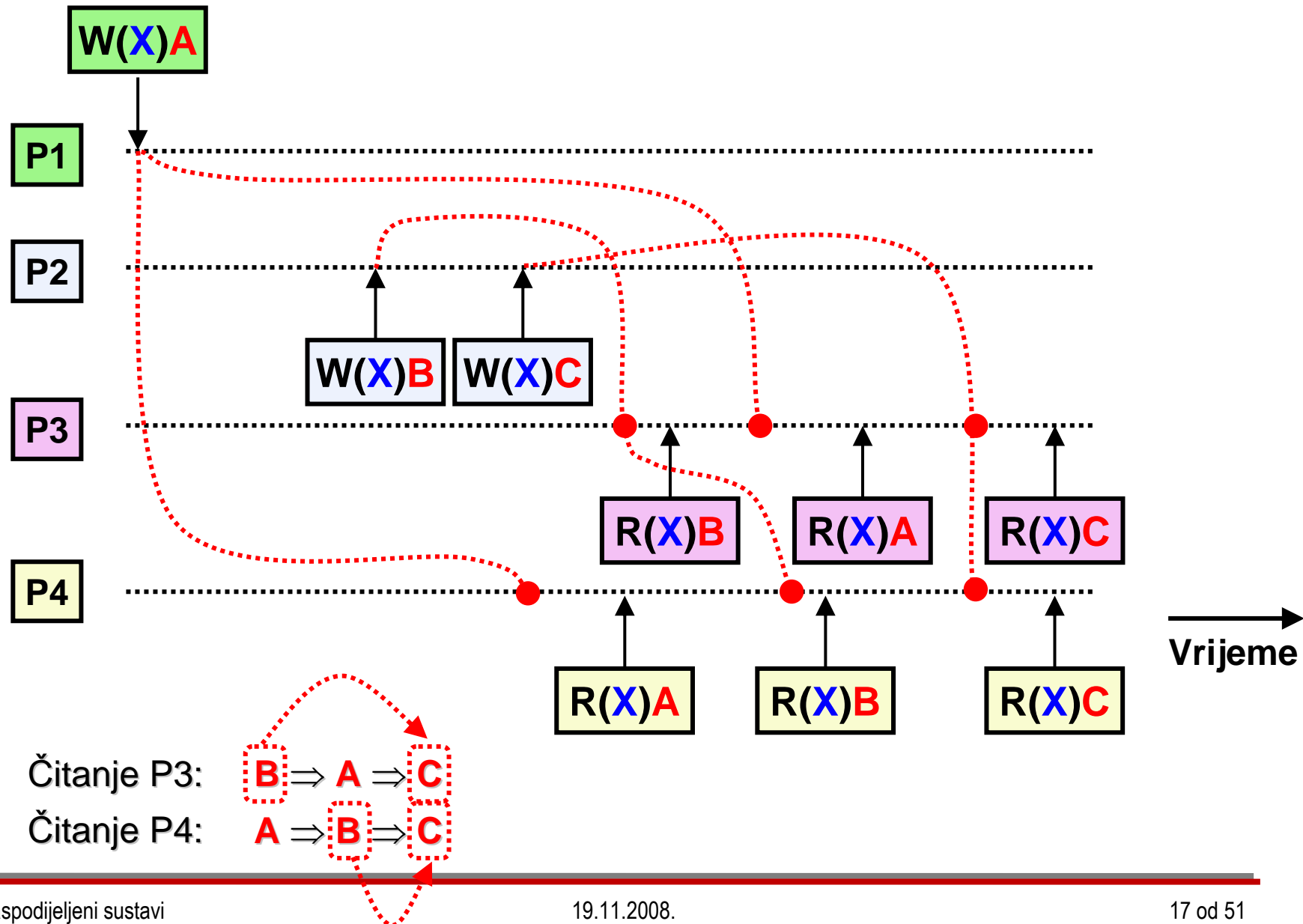
Primjer nepovezane konzistentnosti



- ◆ Redoslijed izvođenja operacija pisanja provedenih od strane **jednog** procesa vidljiv je na **jednak** način svim ostalim procesima, dok je redoslijed izvođenja operacija pisanja **različitih** procesa može biti vidljiv na **proizvoljan** način ostalim procesima

- ◆ **Značajke modela**
 - ◆ Jednostavno ostvarenje zasnovano na pridruživanju jedinstvenih oznaka svakom zahtjevu za pisanje
 - ◆ Jedinstvena oznaka uključuje identifikator procesa i redni broj izvođenja operacije

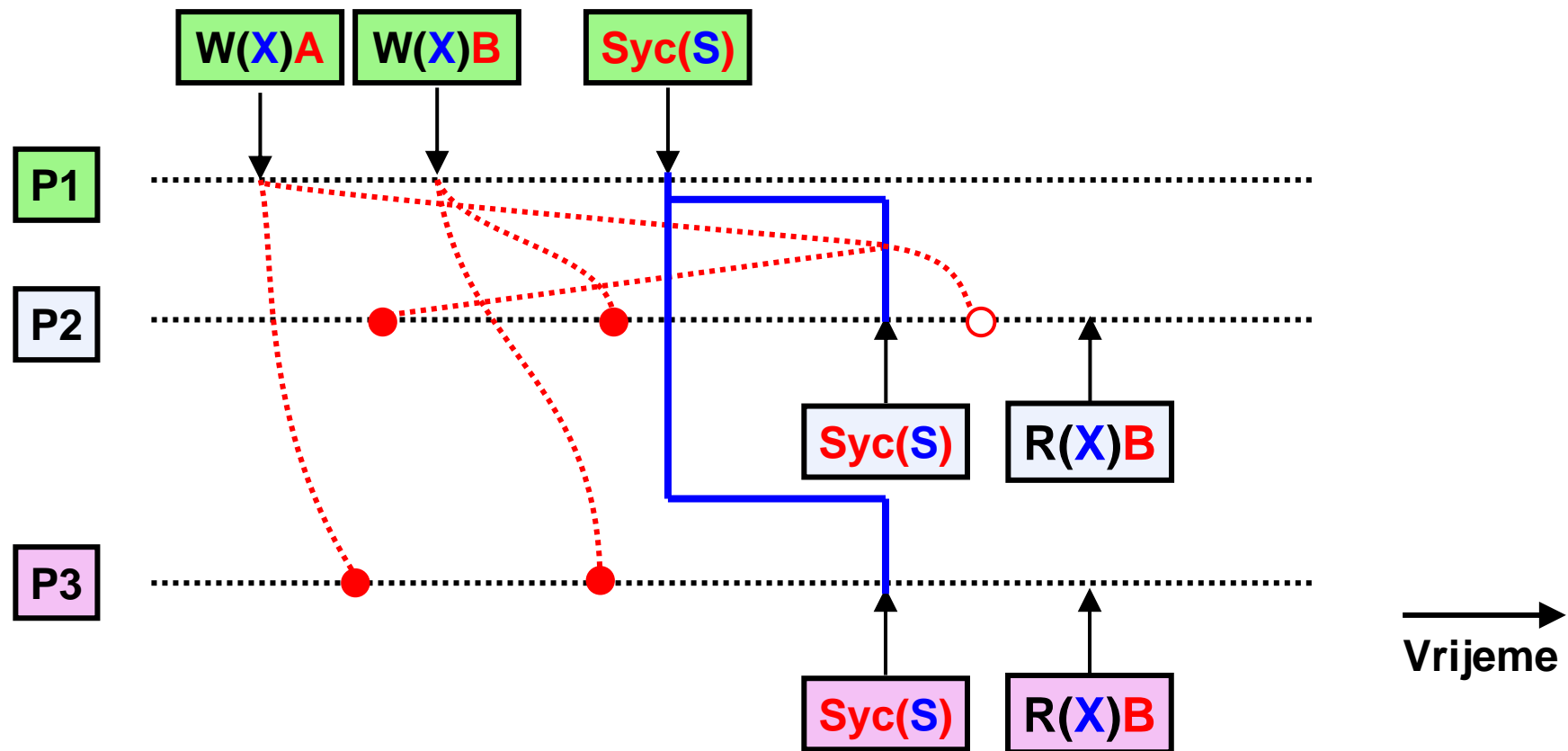
Primjer konzistentnosti redoslijeda



- ◆ Ostvaruje se primjenom sinkronizacijskih varijabli koje omogućavaju upravljanje trenutcima ostvarivanja sinkronizacije replika u raspodijeljenoj okolini
- ◆ Operacija sinkronizacije **Sync (S)**
- ◆ Primjena sinkronizacijskih varijabli
 - ◆ Omogućavaju usklađivanje svih replika zadanog skupa podataka
 - ◆ Slijed akcija nad sinkronizacijskom varijablama vidljiv je na jednak način svim procesima

- ◆ **Uvjeti primjene i ostvarenja operacije Sync**
 - ◆ Operacija nad sinkronizacijskom varijablom dovršava se tek nakon što su završene sve prethodno započete operacija pisanja
 - ◆ Operacije pisanja i čitanja mogu se izvoditi tek nakon završetka izvođenja operacije nad sinkronizacijskom varijablom

Primjer slabe konzistentnosti



Čitanje P2: **B**

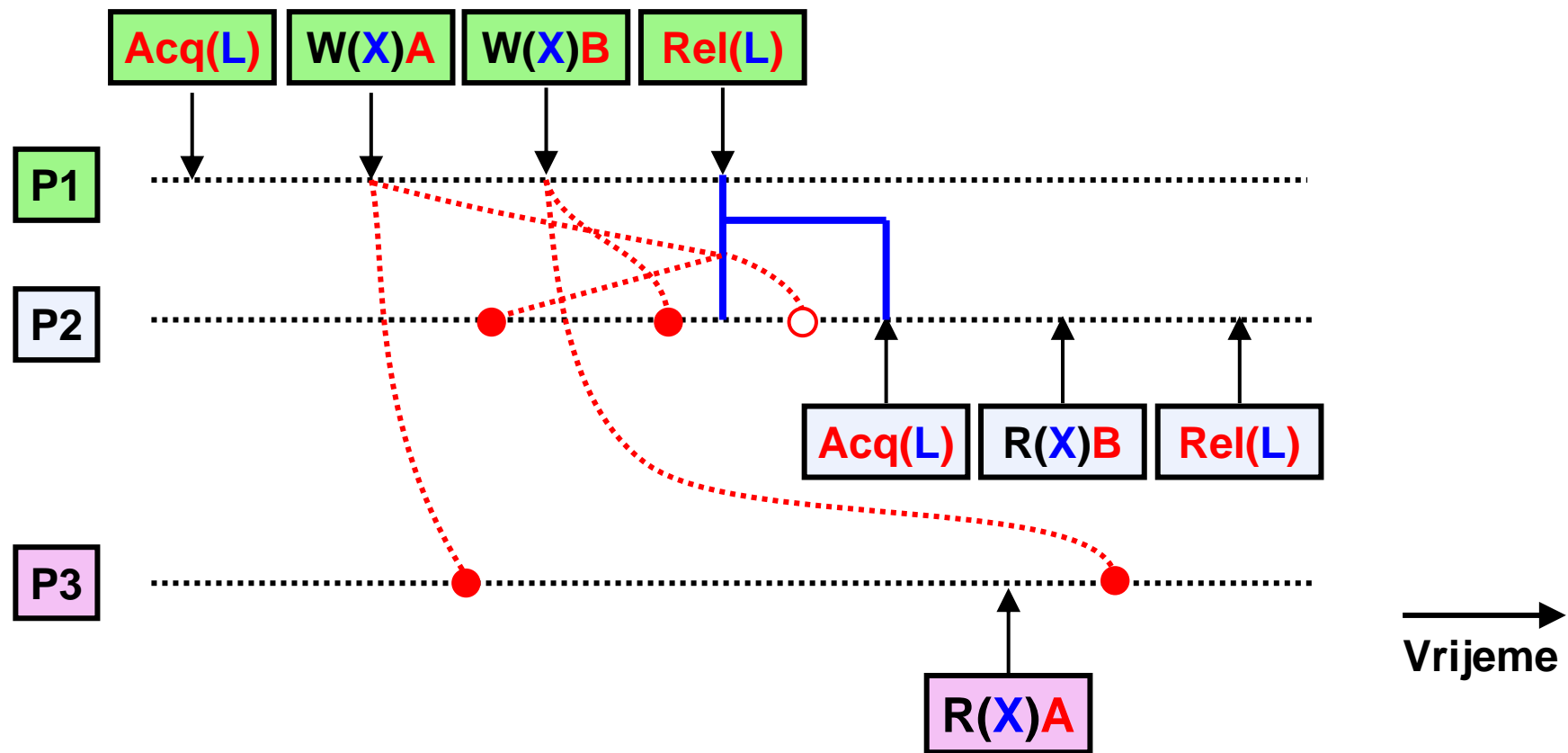
Čitanje P3: **B**

- ◆ **Vrsta konzistentnosti zasnovana na primjeni kritičnih odsječaka**
 - ◆ Konzistentnost se održava nakon izlaska iz kritičnog odsječaka
 - ◆ Izgradnja kritičnih odsječaka ostvarena je primjenom operacija **Zauzmi** i **Otpusti**

- ◆ **Operacija **Zauzmi****
 - ◆ Ulazak u kritični odsječak

- ◆ **Operacija **Otpusti****
 - ◆ Izlazak iz kritičnog odsječaka, prije izlaska sve lokalne promjene prosljeđuju se svim replikama podataka

Primjer konzistentnosti otpuštanja



Čitanje P2: **B**

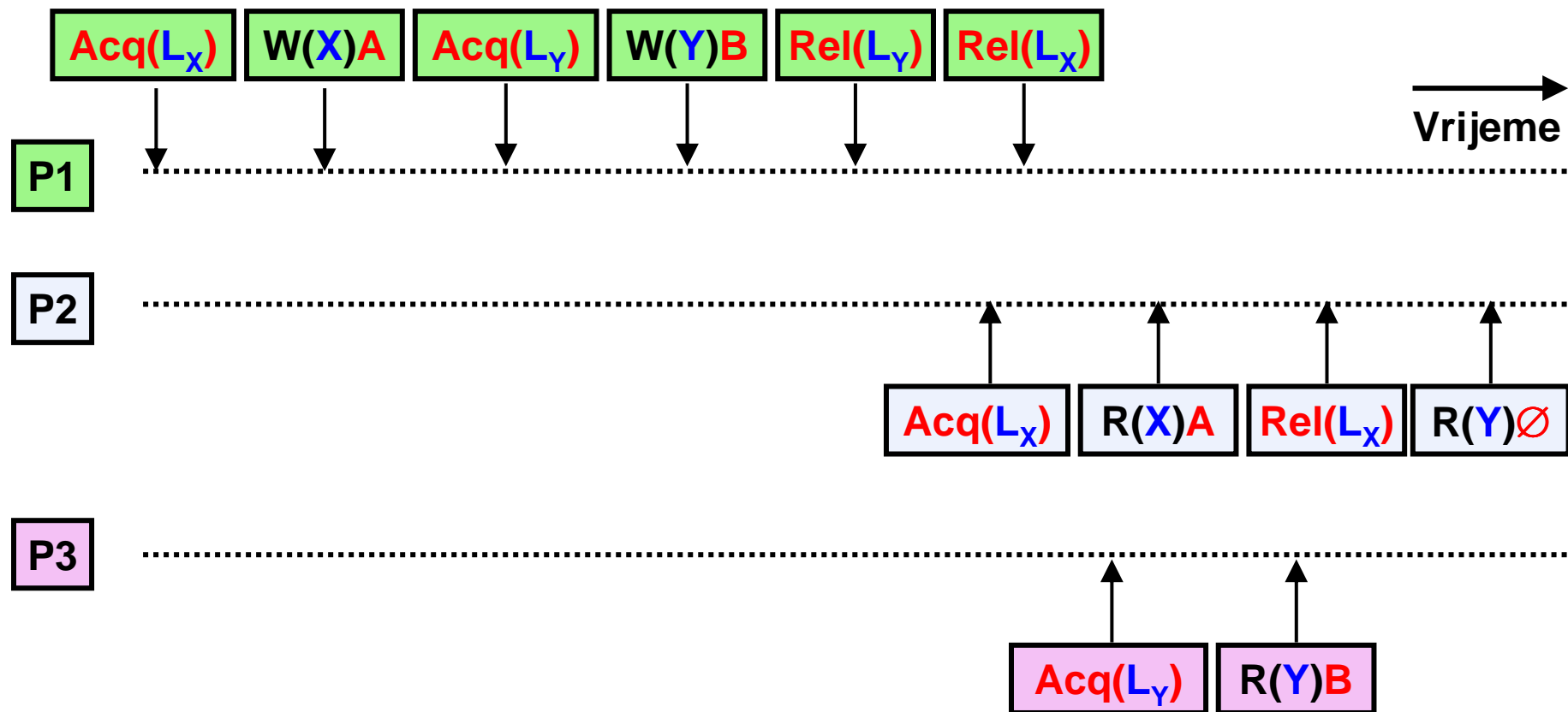
Čitanje P3: **A**

- ◆ Konzistentnost zasnovana primjeni sinkronizacijskih varijabli
 - ◆ Konzistentnost se održava u trenutku ulaska u kritični odsječak
 - ◆ Izgradnja kritičnih odsječaka ostvarenih primjenom operacija **Zauzmi** i **Otpusti**

- ◆ Operacija **Zauzmi**
 - ◆ Ulazak u kritični odsječak, početak izvođenja odsječka tek nakon što su usklađene sve vrijednosti replika s posljednjom promjenom

- ◆ Operacija **Otpusti**
 - ◆ Izlazak iz kritičnog odsječaka

Primjer konzistentnosti zauzimanja



Čitanje P2: $A \Rightarrow \emptyset$

Čitanje P3: B

- ◆ Replikacija i konzistentnost podataka
- ◆ Svrha replikacije podataka
- ◆ Primjena dijeljenog spremničkog prostora
- ◆ Modeli održavanja konzistentnosti podataka
- ◆ Modeli uspostave replikacije podataka

- ◆ **Organizacija sustava replika**

- ◆ Poslužiteljska računala, korisnička računala i spremnici replika

- ◆ **Razredba vrsta replika**

- ◆ Trajne replike, poslužiteljske replike, korisničke replike

- ◆ **Održavanje konzistentnosti replika**

- ◆ Dohvaćanje promjena stanja replika
- ◆ Prosljeđivanje promjena stanja replika

- ◆ **Ostvarivanje operacija**

- ◆ Operacije pisanja sadržaja
- ◆ Operacije čitanja sadržaja

Organizacija sustava replika

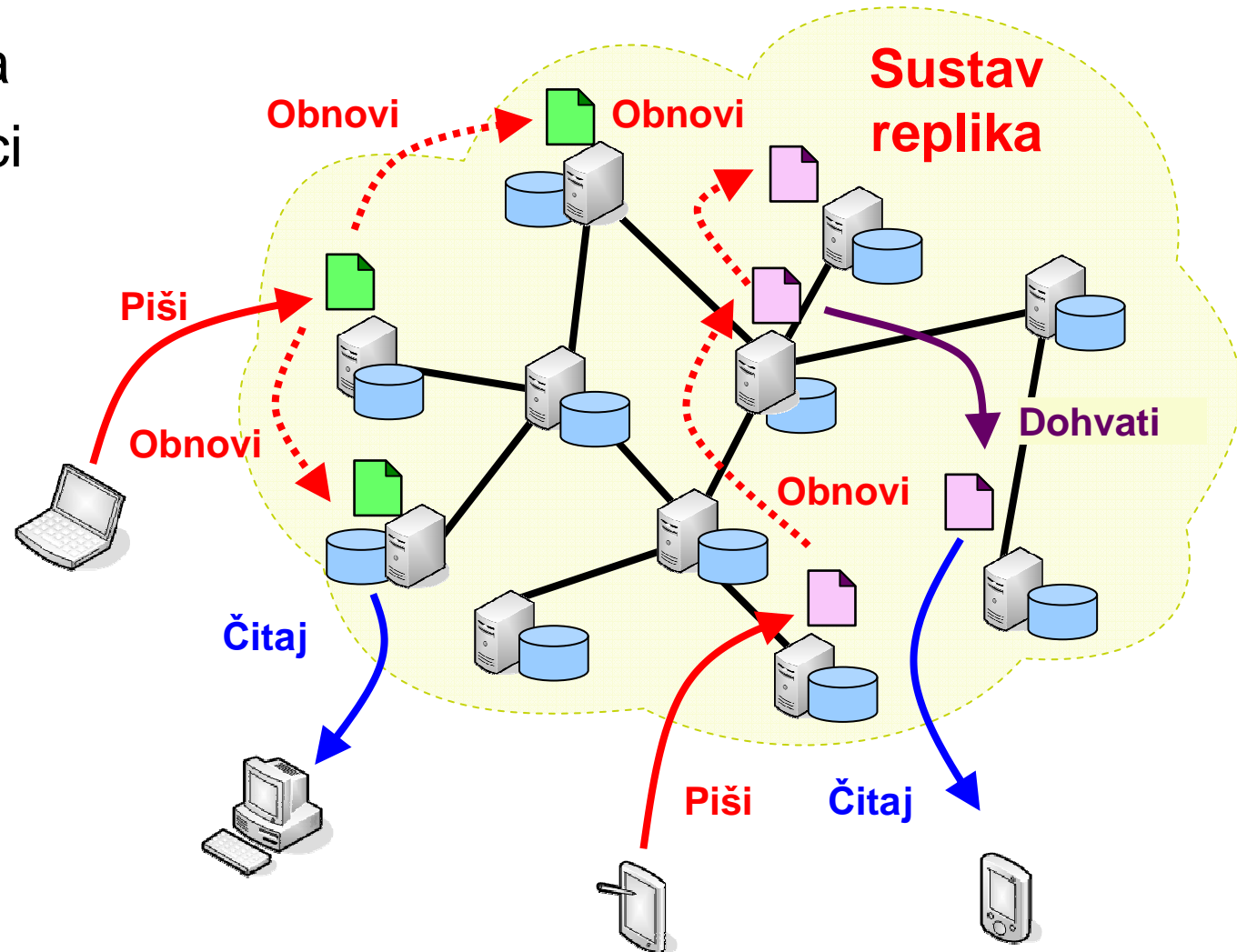


◆ Elementi sustava

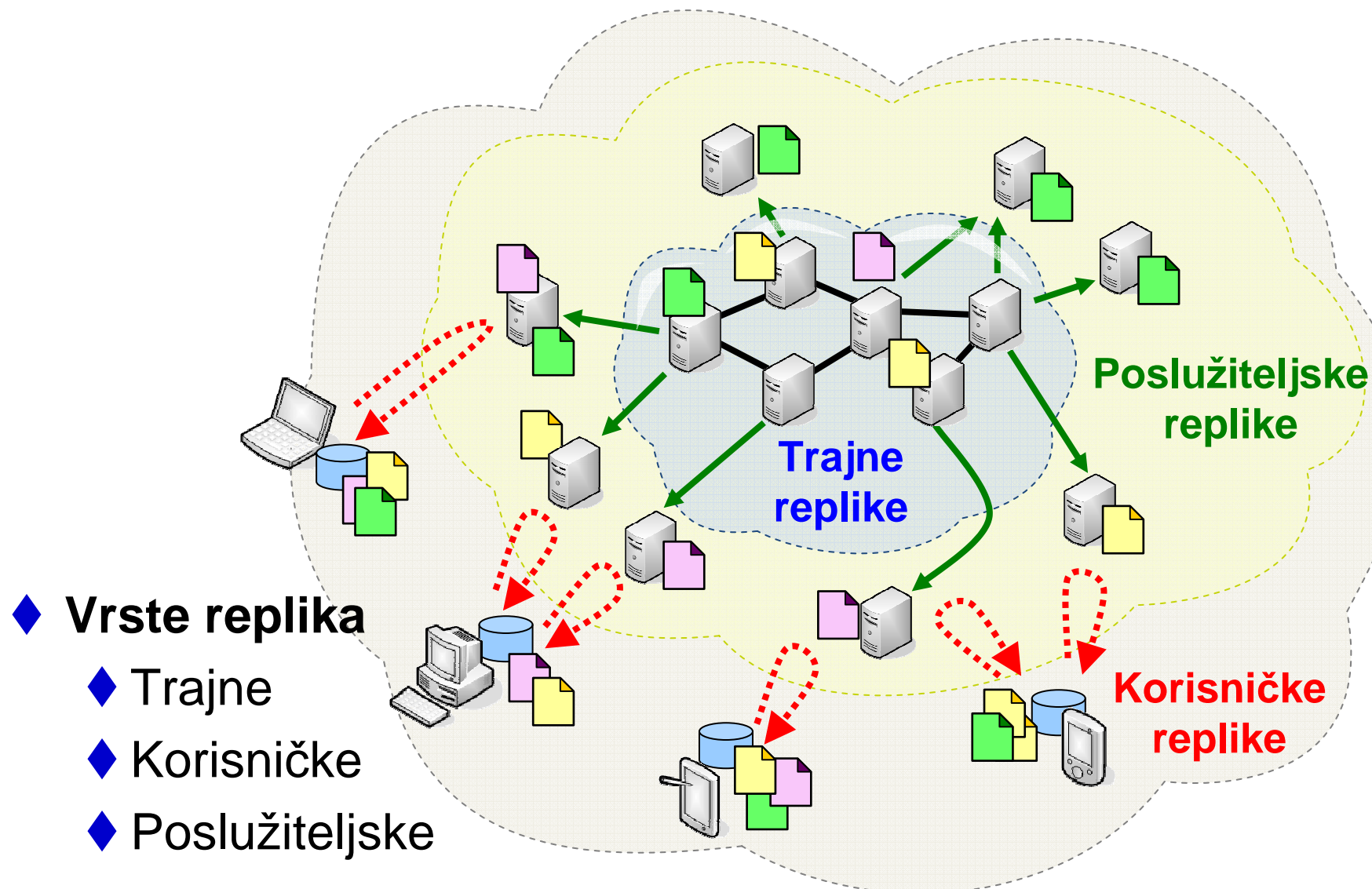
- ◆ Računala
- ◆ Spremnici
- ◆ Replike

◆ Operacije

- ◆ Čitaj
- ◆ Piši
- ◆ Dohvati
- ◆ Obnovi



Razredba vrsti replika

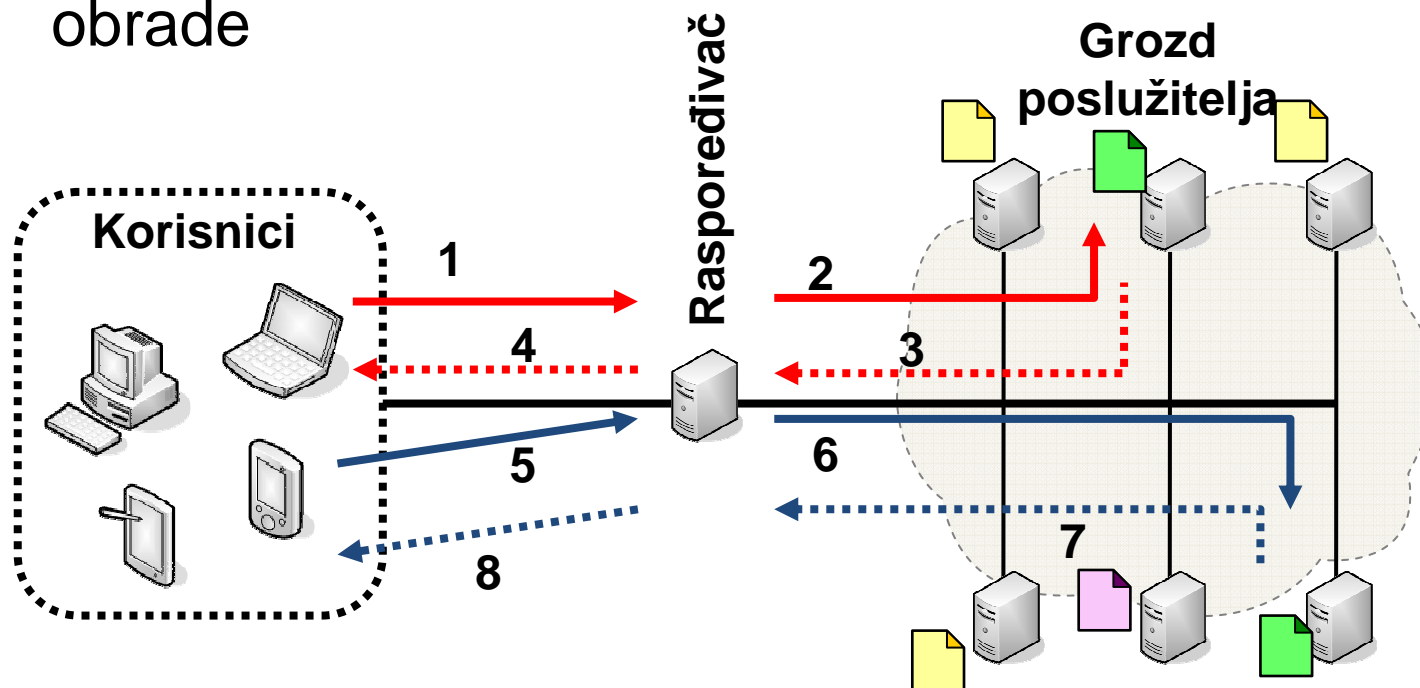


- ◆ **Početni skup replika postavljen na skupu računala povezanih lokalnom mrežom**
 - ◆ Grozdovi poslužitelja
 - ◆ Replike poslužitelja

- ◆ **Osnovne značajke**
 - ◆ Statička organizacija i postavke sustava
 - ◆ Većina zahtjeva je čitanje podataka
 - ◆ Raspoređivanje zahtjeva na dostupne replike

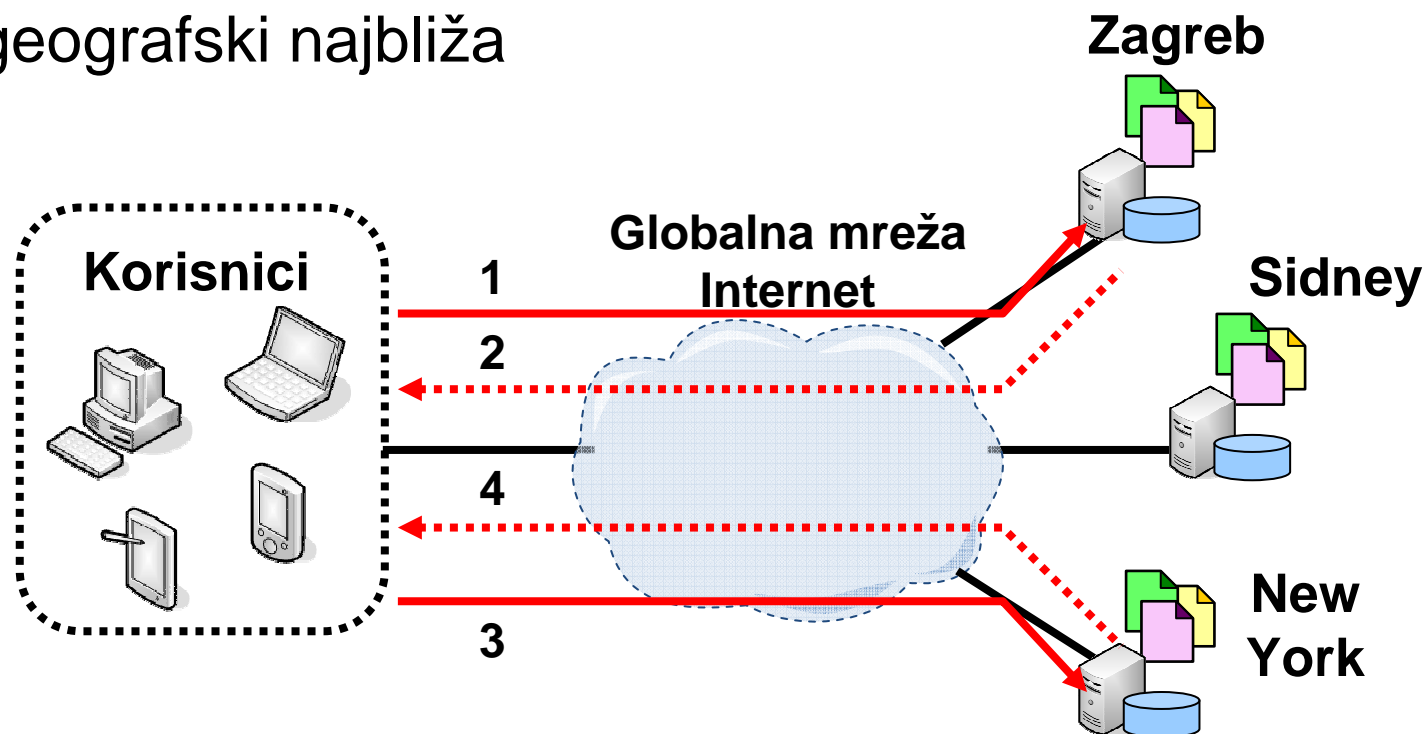
◆ Organizacija grozda poslužitelja

- ◆ Raspoređivač prihvata i prosljeđuje zahtjeve poslužiteljima
- ◆ Poslužitelji prihvataju zahtjev te prosljeđuju rezultate obrade



◆ Organizacija replika poslužitelja

- ◆ Kopije sadržaja poslužitelja postavljaju se nekoliko računala u globalnoj mreži Internet
- ◆ Korisnici sami odabiru repliku poslužitelja koja ima je geografski najbliža



◆ Poslužitelj sadrži trajne replike koje su dostupne korisnicima

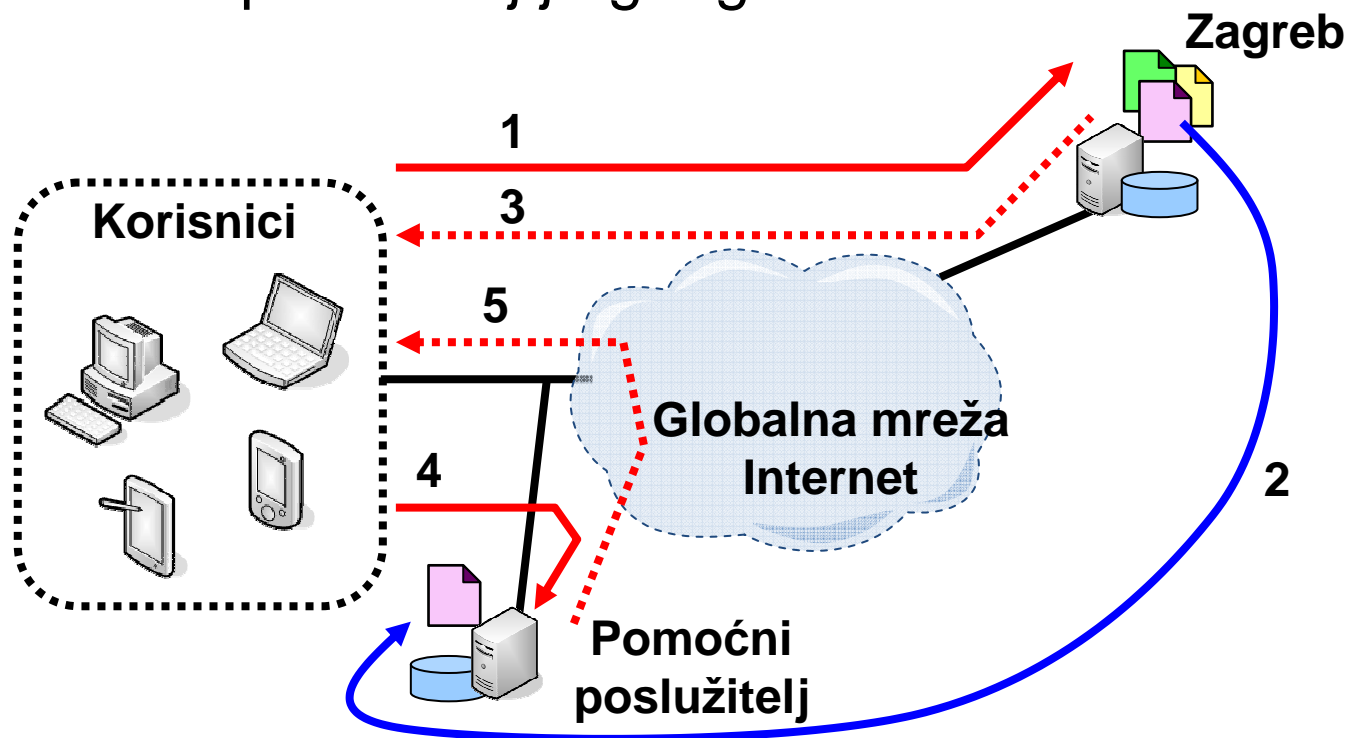
- ◆ U slučaju povećane potražnje za podacima, poslužitelj započinje postupak repliciranja podataka
- ◆ Poslužitelj prosljeđuje replike traženih podataka privremenim poslužiteljima

◆ Osnovne značajke

- ◆ Poslužitelj prati vlastito opterećenje
- ◆ Odabir i raspoređivanje replika ostvaruje se tijekom rada sustava

Poslužiteljske replike

- ◆ Poslužitelj ostvaruje replikaciju u slučaju velike potražnje za obrađenim podacima
 - ◆ Odabire se pomoćni poslužitelj na koji se postavljaju traženi podatak
 - ◆ Pomoćni poslužitelj je geografski bliži korisnicima



◆ Korisnički programi koriste lokalni spremnik

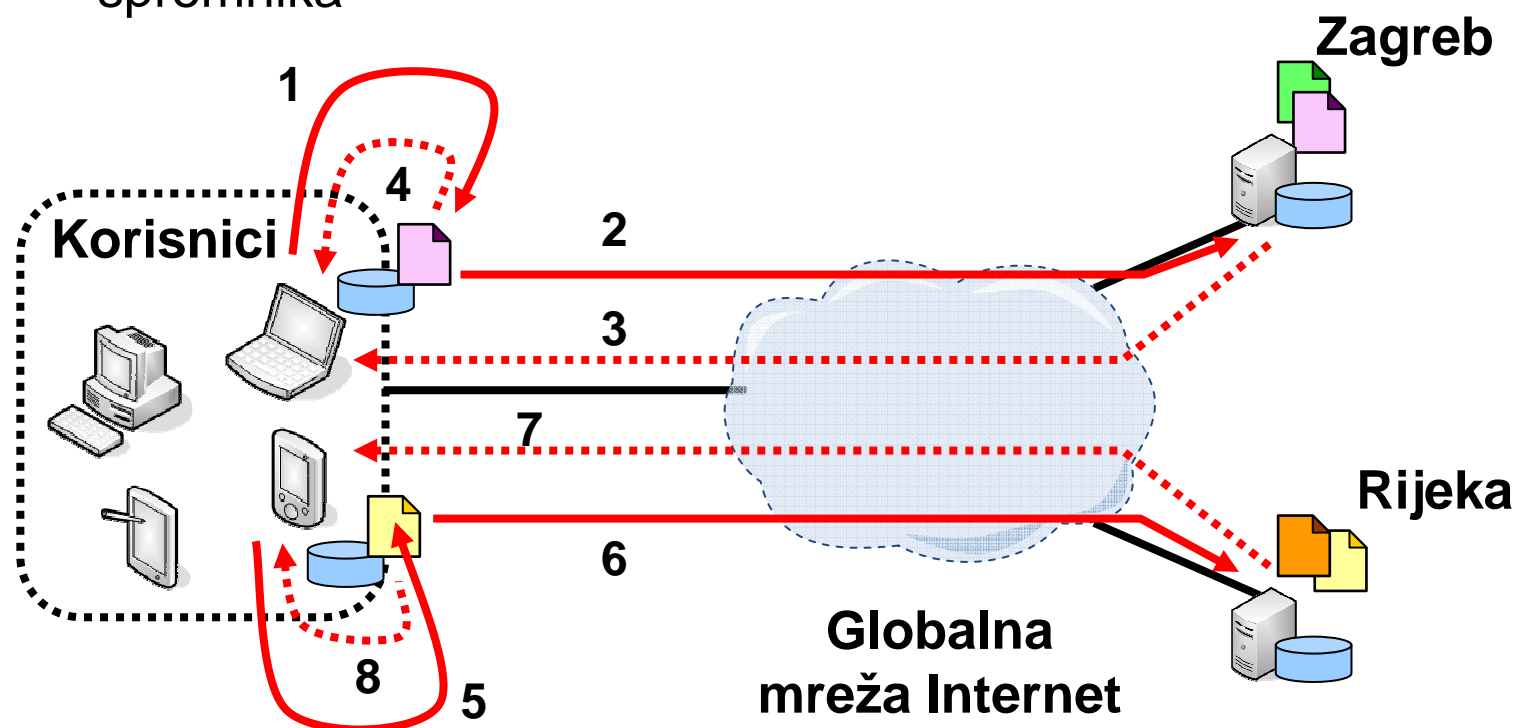
- ◆ Dohvaćeni dokumenti spremaju se u lokalni spremnik
- ◆ U slučaju potrebe za istim podacima, podatci se dohvaćaju iz lokalnog spremnika
- ◆ Potrebno je održavati konzistentnost lokalnog spremnika s poslužitelje od kojeg su podatci dohvaćeni
- ◆ Lokalni spremnik može biti na istom računalu kao i korisnički programi ili na dijeljenom računalu u lokalnoj mreži

◆ Osnovne značajke

- ◆ Najpovoljnije je koristiti u slučajevima kada se najčešće provode operacije čitanja
- ◆ Smanjuje se vrijeme dohvata podataka
- ◆ U slučaju kada nekoliko korisnika dijeli lokalni spremnik povećava se učinkovitost primjene korisničkih replika

Korisničke replike

- ◆ **Korisnici ostvaruju replikaciju dokumenata koje dohvaćaju od udaljenih poslužitelja**
 - ◆ Dokumenti se spremaju u lokalni spremnik
 - ◆ Za svaki sljedeći zahtjev, dokumenti se dohvaćaju iz lokalnog spremnika



- ◆ **Organizacija sustava replika**

- ◆ Poslužiteljska računala, korisnička računala i spremnici replika

- ◆ **Razredba vrsta replika**

- ◆ Trajne replike, poslužiteljske replike, korisničke replike

- ◆ **Održavanje konzistentnosti replika**

- ◆ Dohvaćanje promjena stanja replika
- ◆ Prosljeđivanje promjena stanja replika

- ◆ **Ostvarivanje operacija**

- ◆ Operacije pisanja sadržaja
- ◆ Operacije čitanja sadržaja

◆ Obnavljanje stanja replika

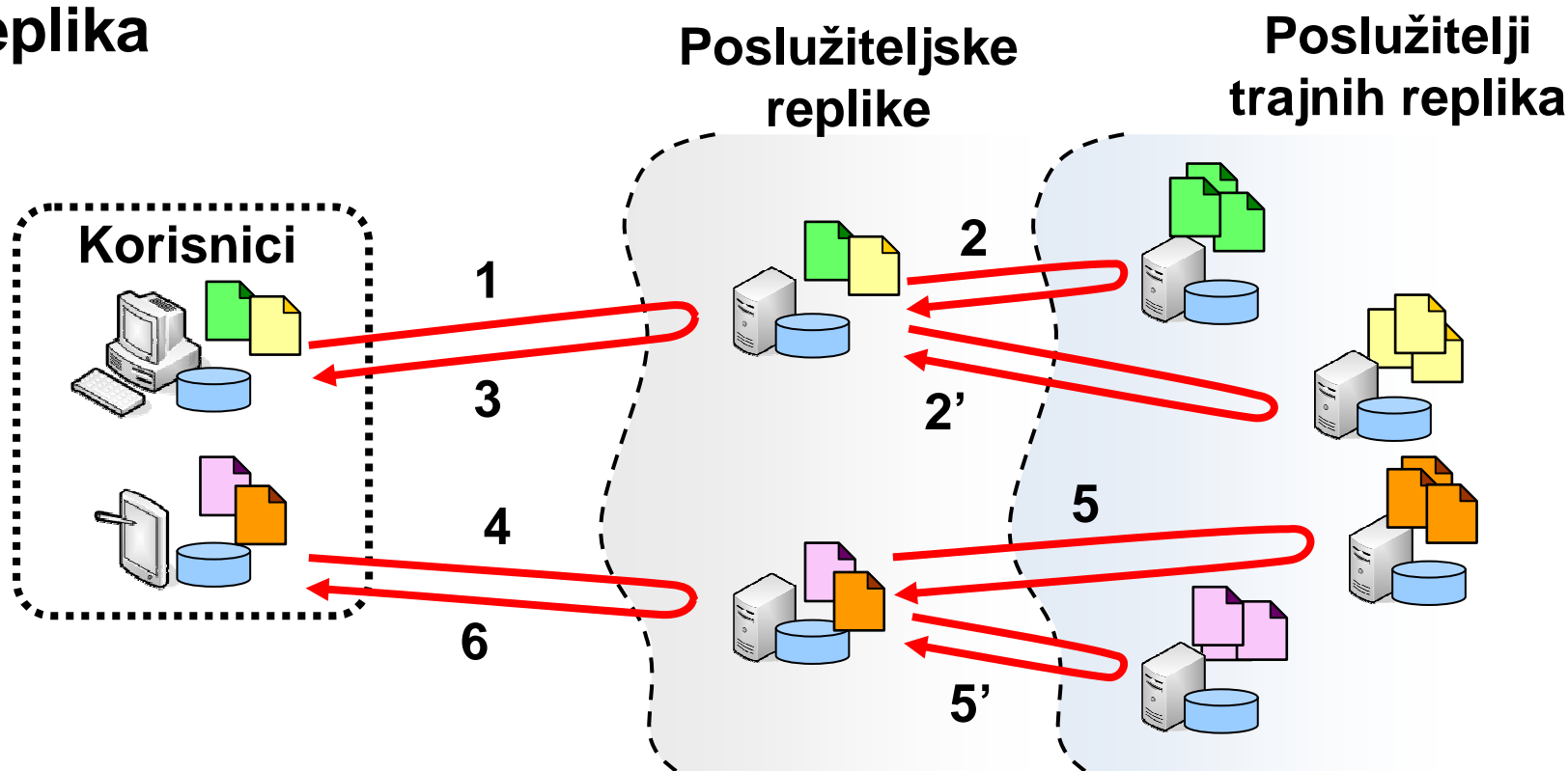
- ◆ Replike je potrebno usklađivati s promjenama stanja trajnih replika
- ◆ Obnavljanje sadržaja replika može biti ostvareno u trenutku promjene sadržaja ili u trenutak prije ostvarivanja pristupa replici

◆ Osnovne metode održavanja konzistentnosti sadržaja replika

- ◆ Dohvaćanje promjena sadržaja (*pull*)
- ◆ Prosljeđivanje promjena sadržaja (*push*)

Dohvaćanje promjena sadržaja

- ◆ Korisnici dohvaćaju promjene sadržaja trenutak prije pristupa replikama
- ◆ Primjenjuju se od strane korisnika prema poslužiteljskim replikama i poslužiteljima trajnih replika

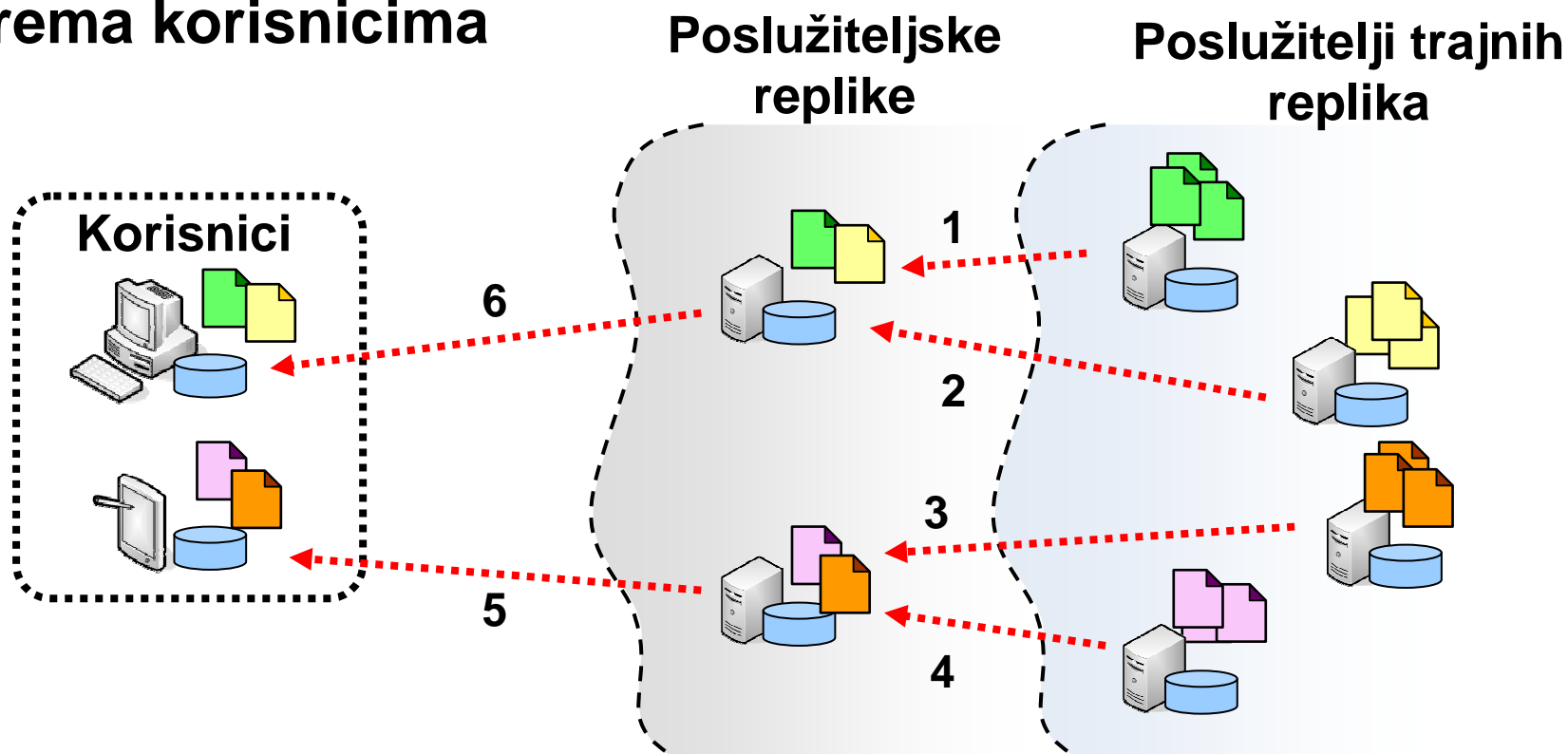


- ◆ **Značajke dohvaćanja promjena sadržaja**
 - ◆ Pogodno za korištenje u slučajevima rijetkih izmjena sadržaja replika
 - ◆ Poslužitelji trajnih replika ne moraju znati broj i identitet korisnika
 - ◆ Smanjuje se mrežno opterećenje i rasterećuje poslužitelj replika
 - ◆ U slučajevima da lokalno stanje replike nije obnovljeno povećava se vrijeme dohvata novog stanja replika

Prosljeđivanje promjena sadržaja



- ◆ Poslužitelji sadržaja prosljeđuju svim replikama promjene stanja sadržaja u trenutku nastanka promjene
- ◆ Primjenjuju se od strane poslužitelja trajnih replika prema korisnicima



- ◆ **Značajke primjene prosljeđivanja sadržaja**
 - ◆ Ostvarivanje velikog stupnja konzistentnosti
 - ◆ Stvaranje dodatnog mrežnog prometa
 - ◆ Poslužitelji trajnih replika moraju imati zabilježene adrese svih replika koje obnavljaju i opis njihova stanja
 - ◆ U slučaju da neka od replika ukloni dio svojeg stanja, replika obavještava o promjenama stanja poslužitelja trajnih replika
 - ◆ Najčešće se koristi u slučajevima kada **veliki** broj korisnika koristi dijeljenu repliku kada je vjerojatno da će uvijek neki od klijenata imati potrebu koristiti obnovljene podatke

- ◆ **Organizacija sustava replika**

- ◆ Poslužiteljska računala, korisnička računala i spremnici replika

- ◆ **Razredba vrsta replika**

- ◆ Trajne replike, poslužiteljske replike, korisničke replike

- ◆ **Održavanje konzistentnosti replika**

- ◆ Dohvaćanje promjena stanja replika
- ◆ Prosljeđivanje promjena stanja replika

- ◆ **Ostvarivanje operacija**

- ◆ Operacije pisanja sadržaja
- ◆ Operacije čitanja sadržaja

- ◆ **Održavanje konzistentnosti sadržaja replika tijekom provođenja operacija**
 - ◆ Operacije čitanja i pisanja
 - ◆ Učinkovitost ostvarenja operacija

- ◆ **Obnavljanje stanja udaljenih replika**
 - ◆ Zahtjev za obnavljanje stanja prosljeđuje se udaljenim replikama koje zajednički ostvaruju

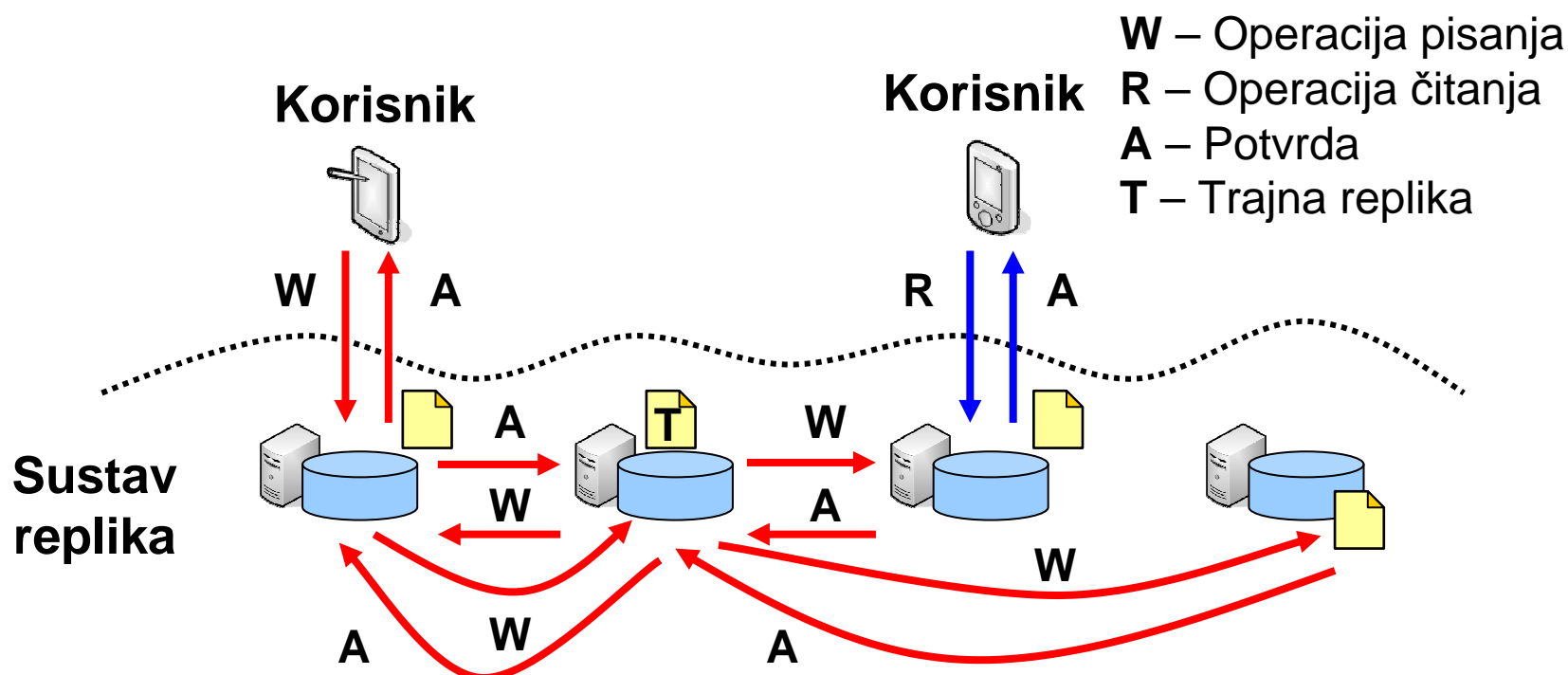
- ◆ **Lokalno obnavljanje stanja replika**
 - ◆ Replike se dohvaćaju na računalo domaćin te se operacije provode lokalno

Obnavljanje stanja udaljenih replika



◆ Model ostvarivanja operacija čitanja i pisanja

- ◆ Zahtjevi za obnavljanje stanja prosljeđuju se udaljenoj trajnoj replici za zadani sadržaj
- ◆ Operacije čitanja se provode na lokalnoj replici ili bilo kojoj drugoj replici u sustavu replika



◆ Značajke modela obnavljanja stanja udaljenih replika

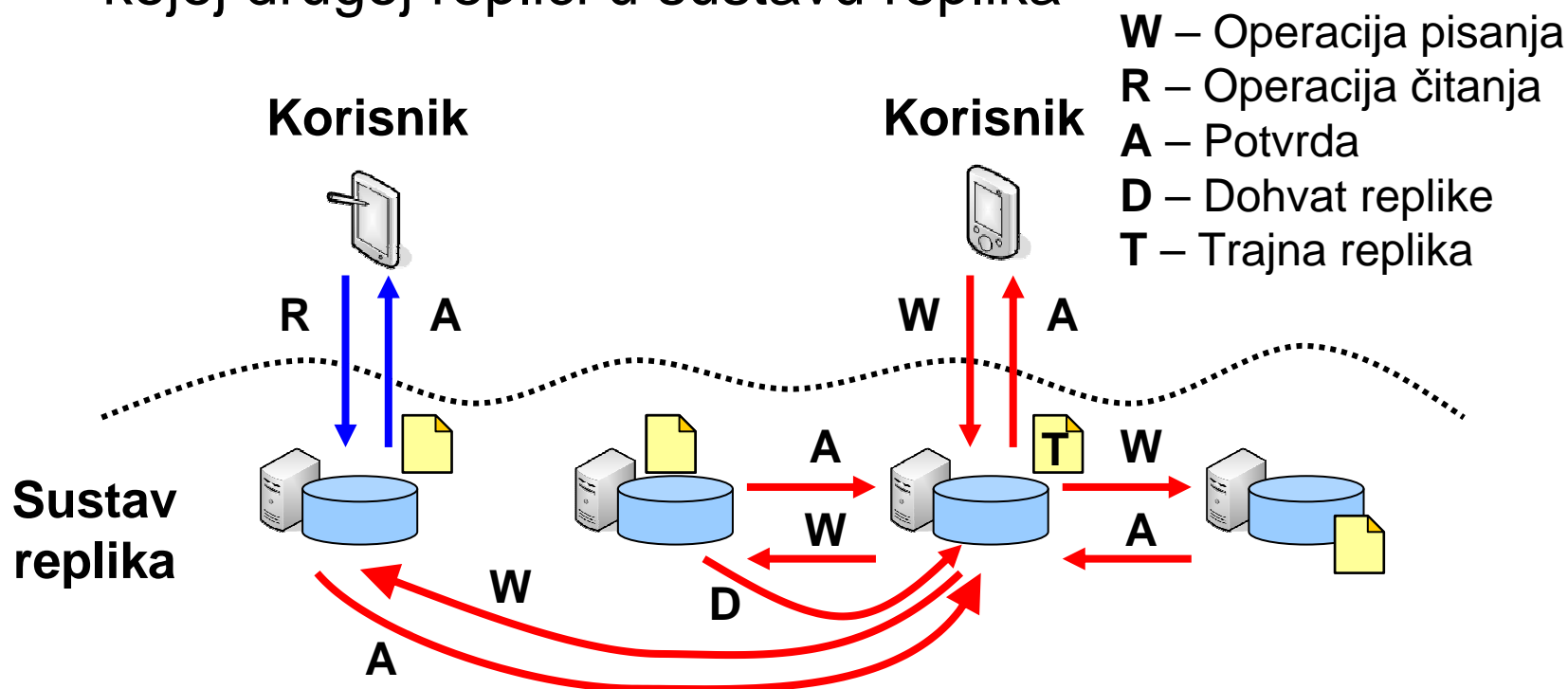
- ◆ Omogućava uspostavu slijedne konzistentnosti obzirom da samo glavna replika provodi operacije pisanja
- ◆ Svi korisnici doživljavaju jednak redoslijed izvođenja operacija pisanja u vremenu bez obzira putem koje replike dohvaćaju sadržaj
- ◆ U slučaju velikog broj pomoćnih replika, izvođenje operacije pisanja može zahtijevati značajnu količinu vremena potrebu za provođenje operacije pisanja na svim pomoćnim replikama

Lokalno obnavljanje stanja replika



◆ Model ostvarivanja operacija čitanja i pisanja

- ◆ Trajna replika se dohvaća na računalo domaćin i provodi se operacija pisanja
- ◆ Operacije čitanja se provode na lokalnoj replici ili bilo kojoj drugoj replici u sustavu replika



- ◆ **Značajke modela lokalnog obnavljanja stanja replika**
 - ◆ Uzastopne operacije pisanja mogu biti provedene u kratkom vremenu na računalu domaćinu
 - ◆ Rezultati provođenja uzastopnih operacija mogu biti agregirani u jednu operaciju pisanja koja se provodi na pomoćnim replikama u raspodijeljenoj okolini
 - ◆ Korisnici koji čitaju sadržaj mogu pristupiti vlastitim lokalnim replikama neovisno o trajnoj replici

Usporedba modela replikacije u sustavima



	DB	P2P	WWW
Upravljanje	Centralizirano	Raspodijeljeno	Centralizirano
Povezanost	Čvrsta	Labava	Čvrsta
Raspoređivanje	Centralizirano	Raspodijeljeno	Centralizirano
Vrsta većine operacija	Pisanje i čitanje	Čitanje	Čitanje
Pouzdanost	Predvidiva	Nepredvidiva	Predvidiva
Raznorodnost sredstava	Ne	Da	Ne

Primjer: Java Caching System



- ◆ **Raspodijeljeni sustava za upravljanje replikama**
 - ◆ Predviđen za aplikacije koje većinom provode operacije čitanja, dok su operacije pisanja rijetke
 - ◆ Ostvaren primjenom programskog jezika Java
 - ◆ <http://jakarta.apache.org/jcs>

- ◆ **Značajke**
 - ◆ Upravljanje spremničkim prostorom
 - ◆ Upravljanje trajnosti podataka
 - ◆ Sinkronizacija operacija procesa
 - ◆ Transakcije
 - ◆ Pouzdanost i dostupnost

◆ Knjige

- ◆ S. Tanenbaum, M. van Steen: “**Distributed Systems: Principles and Paradigms**”, Prentice Hall, 2002. (Poglavljje: *Consistency and Replication*)
- ◆ H. Attiya, J. Welch: “**Distributed Computing: Fundamentals, Simulations, and Advanced Topics**”, Wiley, 2004. (Poglavljje: *Distributed Shared Memory*)

◆ Znanstveni radovi

- ◆ S. Goel, S. Buyya: "**Data Replication Strategies in Wide-Area Distributed Systems**", Enterprise Service Computing

<http://www.buyya.com/papers/DataReplicationInDSChapter2006.pdf>

- ◆ P. Padmanabhan, L. Gruenwald, A. Vallur, M. Atiquzzaman: "**A Survey of Data Replication Techniques for Mobile and Ad-Hoc Network Databases**", Journal of Very Large Data Bases
<http://www.cs.ou.edu/~database/documents/VLDB08.pdf>