

# Kontrola pristupa

# Literatura o kontroli pristupa

---

- David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli. *Role-Based Access Control*. Artech House, 2003. ISBN 1580533701

# Autentifikacija $\neq$ identifikacija $\neq$ autorizacija

---

- autentifikacija (provera identiteta) = utvrđivanje identiteta korisnika
- identifikacija = utvrđivanje da li je korisnik poznat sistemu
- autorizacija = utvrđivanje prava koja korisnik ima nad resursima u sistemu
  
- autorizacija zahteva uspešnu autentifikaciju
  - autentifikacija prethodi autorizaciji
- autentifikacija podrazumeva identifikaciju
  - identifikacija je sastavni deo postupka autentifikacije

# Kontrola pristupa

---

- "ko može da uradi šta"
- sigurnosni mehanizam koji je prisutan u svim delovima informacionih sistema
- prvi bezbedan sistem za računanje - registar kasa (Dayton, Ohio, 1879)
  - kupac vidi iznos koji je prodavac otkucao
  - fijkica se otvara samo prilikom unosa iznosa
  - kasa zapisuje istoriju naplata

# Rizici po bezbednost informacija

---

- „CIA“ klasifikacija
- *confidentiality* (poverljivost)
  - čuvanje podataka od neovlašćenog čitanja
- *integrity* (integritet)
  - čuvanje podataka od izmena
- *availability* (dostupnost)
  - informacije su dostupne u trenutku kada su i potrebne
- mehanizmi kontrole pristupa bave se poverljivošću i integritetom
  - onaj ko neovlašćeno pristupi nekom sistemu može da utiče i na dostupnost

# Razvoj mehanizama kontrole pristupa

---

- prvi radovi početkom 1970-tih
- standardizacija početkom 1980-tih
- role-based access control (RBAC) početkom 1990-tih

# Koncepti kontrole pristupa

---

- korisnik (*user*)
  - čovek koji koristi informacioni sistem
  - ima svoj identifikator
  - može imati više identifikatora
  - sistem može povezati više identifikatora sa istim korisnikom
- sesija (*session*): jedna instanca komunikacije korisnika sa sistemom

# Koncepti kontrole pristupa

---

- subjekat (*subject*)
  - računarski proces (~program) koji obavlja zadatke za korisnika
  - jedan korisnik, sa istim ID-jem, može imati više subjekata (email klijent, web klijent, ...)
  - kontrola pristupa sprovodi se za svaki subjekat posebno



# Koncepti kontrole pristupa

---

- objekat (*object*)
  - bilo koji resurs informacionog sistema koji je dostupan korisniku
    - fajl
    - štampač
    - baza podataka
    - pojedini slogovi u bazi podataka
  - tipično se tretiraju kao pasivni entiteti koji sadrže ili primaju podatke
  - stari modeli kontrole pristupa omogućavali su i tretman aktivnih entiteta (programa, ...) kao objekata

# Koncepti kontrole pristupa

---

- operacija (*operation*)
  - aktivan proces koga je pokrenuo subjekat
  - primer: bankomat
    - korisnik se autentifikuje karticom i PIN-om
    - program koji opslužuje korisnika je subjekat
    - subjekat može da pokrene više operacija
      - upit stanja
      - isplata
      - uplata

# Koncepti kontrole pristupa

---

- dozvola (*permission*)
  - dopuštenje da se obavi određena operacija u okviru sistema
  - kombinuje objekat i operaciju
    - dva objekta i ista operacija → različite dozvole
    - isti objekat i dve operacije → različite dozvole

# Koncepti kontrole pristupa

- minimalne privilegije (*least privilege*)
  - selektivno dodeljivanje dozvola korisnicima  
tako da nemaju više privilegija nego što je minimalno neophodno za obavljanje njihovog posla
  - ako korisnik ima mogućnost da izvrši nepotrebne ili štetne operacije → potencijalni problem
  - određivanje skupa minimalnih privilegija je zadatak administrativne prirode
    - identifikacija funkcija vezanih za jedno radno mesto ili korisnika
    - specifikacija dozvola potrebnih za obavljanje svake od funkcija
    - restrikcija korisnika na neki domen uz dodeljene privilegije
  - striktno pridržavanje ovog principa → korisnik može imati različite dozvole u različitim trenucima
    - skup dozvola se menja tokom vremena (dinamička priroda)

# Elementi kontrole pristupa

---

- **politika** kontrole pristupa
- **mehanizmi** kontrole pristupa
- **model** kontrole pristupa

# Politika kontrole pristupa

---

- politika: zahtevi visokog nivoa kojima se definiše ko može da pristupi čemu i pod kojim uslovima
- politika se može definisati posebno za različite aplikacije ali često je definisana u okviru realnog sistema (njegove organizacione strukture)
  - finansijska institucija
  - vojna institucija
  - zdravstvena institucija
- politika se menja tokom vremena, jer odslikava promene u načinu rada organizacije
  - pri tome ne moraju da se menjaju model ili mehanizmi kontrole pristupa

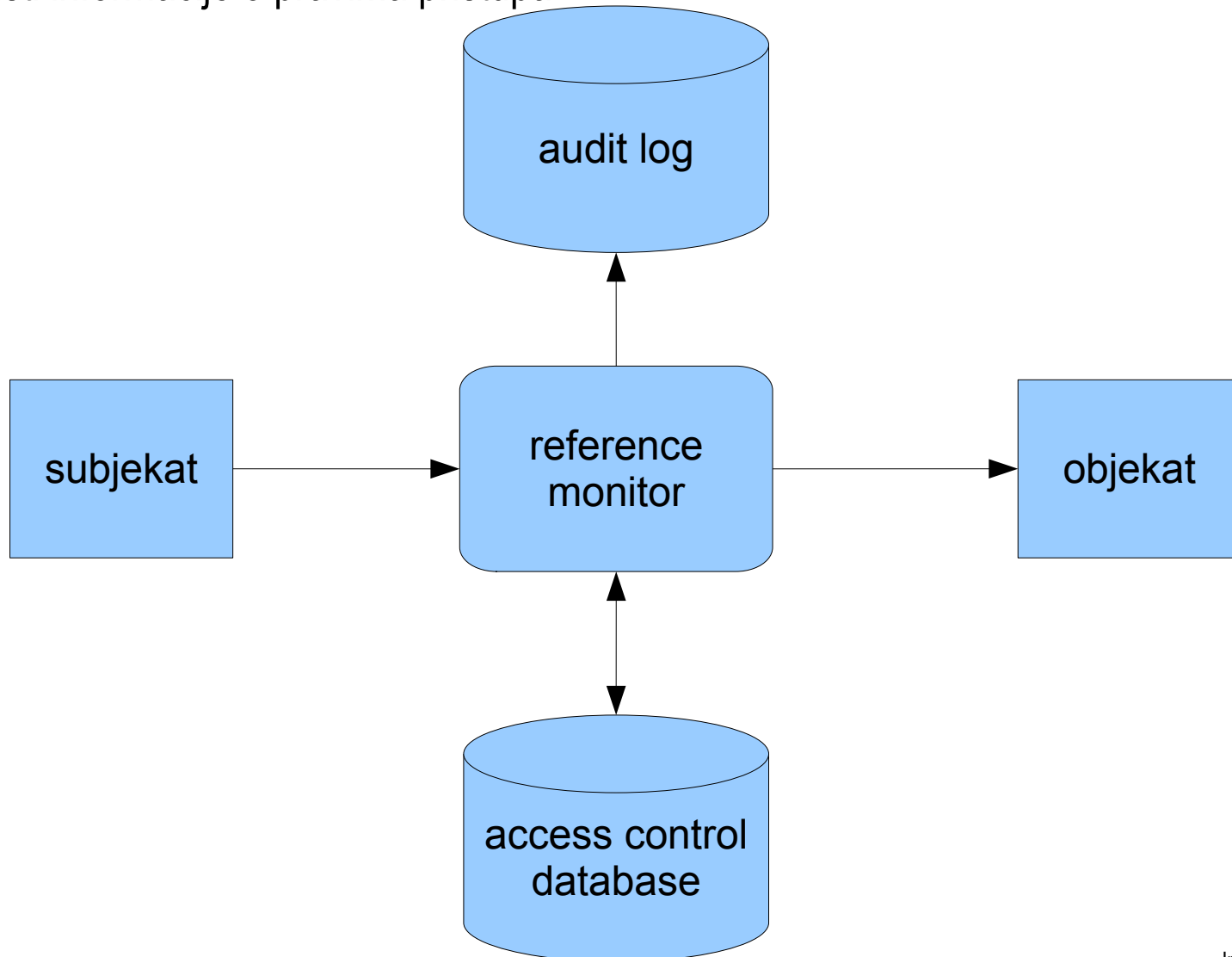
# Mehanizmi kontrole pristupa

---

- sprovode politiku kontrole pristupa
- vezuju bezbednosne attribute za korisnike i resurse
- mehanizam 1: poređenje vrednosti bezbednosnih atributa
  - za *read* operaciju: *clearance level*  $\geq$  *classification level*
- mehanizam 2: prisustvo u bezbednosnim atributima
  - za fajl je vezana lista parova (korisnik, pravo)
  - proverava obuhvata pretragu liste za korisnika koji traži pristup i operaciju koju zahteva
- ...

# Mehanizmi za kontrolu pristupa: reference monitor

- apstraktni pogled na (pod)sistem za kontrolu pristupa
- koristi informacije o pravima pristupa





# Mehanizmi za kontrolu pristupa: reference monitor

---

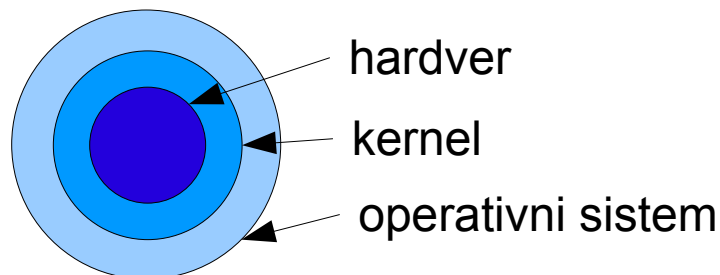
- principi implementacije reference monitora
  - kompletnost (*completeness*)  
uvek se mora pozvati i nije ga moguće zaobići
  - izolacija (*isolation*)  
mora biti otporan na neovlašćene izmene (*tampering*)
  - proverivost (*verifiability*)  
njegova korektna implementacija mora biti proveriva/dokaziva

# Mehanizmi za kontrolu pristupa: reference monitor

- kompletност
  - subjekat može da pristupi objektu **isključivo** preko RM-a
  - problem 1: šta su objekti?
    - očigledne stvari: fajlovi, memorija, baferi, ...
    - manje očigledne stvari: imena fajlova, poruke o greškama, ...
    - kompletност zahteva da se zaštite **svi** objekti, ne samo očigledni
  - problem 2: kako sprečiti zaobilaženje RM-a?
    - kako sprečiti pristup fajlu ako se pristup vrši preko fizičke adrese na disku?
    - kako SUBP da spreči pristup svojim fajlovima od strane operativnog sistema?

# Mehanizmi za kontrolu pristupa: reference monitor

- izolacija
  - mora biti nemoguće za napadača da pristupi/promeni RM tako da on više ne funkcioniše pravilno
  - potrebna je podrška i u hardveru i u softveru
  - jedno rešenje: security kernel
    - minimalna implementacija onih funkcija sistema koje su relevantne za bezbednost
    - oslanja se na hardver
    - pruža usluge delovima operativnog sistema na višem nivou
    - koristi se i za razdvajanje koda i podataka operativnog sistema od aplikacija
    - kernel softver je takođe podložan greškama u implementaciji



# Mehanizmi za kontrolu pristupa: reference monitor

- proverivost
  - ispravnost security kernela je potrebno proveriti
  - komplikovano testiranje
  - napraviti kernel što manjim
    - isključiti sve funkcije koje nisu potrebne za bezbednost sistema
    - definisati mali i jednostavan skup interfejsa
  - olakšati testiranje kernela
    - apstrakcija
    - skrivanje informacija
    - modularnost
  - formalno modeliranje kernela, formalne metode provere korektnosti

# Mehanizmi kontrole pristupa: reference monitor

---

- RM je potreban uslov za sprovođenje kontrole pristupa
- RM nije dovoljan uslov
  - najčešće se RM kupuje u okviru nekog većeg sistema
  - kako svoju politiku implementirati na kupljenom RM-u?
- tri dodatna uslova za sistem za kontrolu pristupa
  - fleksibilnost  
sistem mora da podrži politiku kontrole pristupa u organizaciji
  - upravljivost  
sistem mora biti jednostavan za korišćenje i upravljanje
  - skalabilnost  
funkcije sistema moraju raditi na isti način i za realan (veliki) broj korisnika i resursa u realnom (velikom) sistemu

# Modeli kontrole pristupa

---

- zasnivaju se na konceptima kontrole pristupa
- predstavljaju apstraktni pogled na mehanizme za sprovođenje kontrole pristupa
  - jednostavnija analiza
  - mogućnost izbora različitih implementacija
- razvoj od 1960-tih (vojne primene) do danas
- Lampson
- Bell-LaPadula
- US DoD standardi
  - discretionary access control (DAC)
  - mandatory access control (MAC)
- Clark-Wilson
- role-based access control (RBAC)

# Lampson model

- „matrica pristupa“
  - jedan red po subjektu
  - jedna kolona po objektu
- koristi koncepte subjekta i objekta
  - subjekti: procesi koje je pokrenuo korisnik
  - objekti: resursi sistema (fajlovi, ...)
  - operacije: operacije nad resursima (čitanje, pisanje)
- elementi matrice su skupovi dozvoljenih operacija

|       | Glavna knjiga | Obračun zarada | ... |
|-------|---------------|----------------|-----|
| Alice | RW            |                |     |
| Bob   |               | RW             |     |
| Carol | R             |                |     |

# Lampson model

- u realnim sistemima matrice su velike (puno korisnika, puno objekata)
- i retko popunjene (*sparse*)
- dva uobičajena mehanizma za implementaciju:
  - liste sposobnosti (*capability lists*)
    - za svaki subjekat čuva se lista njegovih „sposobnosti“ (objekat, pravo)
    - kako dobiti sve subjekte koji mogu da pristupe određenom objektu? samo prolazom kroz sve liste
      - npr. komplikovano brisanje fajla
  - liste kontrole pristupa (*access control lists, ACLs*)
    - za svaki objekat čuva se lista parova (subjekat, pravo)
    - ne mora biti puno ACL lista u sistemu ako se korisnici raspodele u grupe



# Bell-LaPadula model

- formalizacija vojnih pravila za kontrolu pristupa
- objekti = dokumenti
- objekti imaju svoj nivo poverljivosti (*classification level*)
  - poverljivo, strogo poverljivo, državna tajna
- korisnici imaju svoj nivo pristupa (*clearance level*)
- osnovna pravila:
  1. „no read up“  
subjekat ima pristupa samo onim dokumentima čiji nivo je manji ili jednak njegovom
  2. „no write down“  
subjekat može da piše samo u dokumente čiji nivo je veći ili jednak njegovom
- dodatni koncept: kategorija
  - svaki dokument spada u jednu ili više kategorija
  - korisnik mora imati odgovarajući nivo pristupa za svaku kategoriju

# Bell-LaPadula model

---

- problem ovog modela - sistem nije odlučiv (*undecidable*):  
ne može se znati da li će konfiguracija za koju se smatra da je ispravna ostati ispravna
  - Harrison, Ruzzo, Ullman 1976. - formalan dokaz
- korisnici mogu (čak i bez namere) dodeliti prava pristupa kroz mehanizme za delegiranje prava

# TCSEC standard

---

- standardizacija modela kontrole pristupa u okviru US Department of Defense
- *Trusted Computer System Evaluation Criteria* („Orange Book“) 1983.
- definiše dva modela kontrole pristupa
  - Discretionary Access Control (DAC)
    - vlasnici objekata dodeljuju prava pristupa
    - posebno pravo delegiranja sopstvenog prava
    - model nije odlučiv
  - Mandatory Access Control (MAC)
    - model jeste odlučiv
    - višenivojski model, na osnovu Bell-LaPadula

# Discretionary Access Control

---

- ograničavanje pristupa objektima na osnovu identiteta korisnika ili grupe korisnika
- „diskrecija“: korisnik koji ima odgovarajuće pravo može delegirati pravo pristupa objektu drugim korisnicima
- koncept „vlasništva“ nad objektom
  - vlasnik objekta ima pravo da dodeljuje prava pristupa drugim korisnicima
- jedan od mehanizama za implementaciju DAC modela: ACLs

# Discretionary Access Control

- dve osnovne slabosti:
  - dodeljivanje prava čitanja je tranzitivno
    - Alice dozvoli Bobu da čita određeni fajl (Alice je vlasnik fajla)
    - Bob iskopira sadržaj fajla u svoj fajl (Bob je vlasnik novog fajla)
    - Bob dozvoli Carol da čita novi fajl
    - Alice ne zna da Carol ima pristup podacima iz njenog fajla!
  - ranjivost na napade trojanskim konjem
    - programi nasleđuju identitet korisnika koji ih je pokrenuo
    - Bob napiše program za Alice koji će sadržaj Alicinog fajla iskopirati na neko mesto koje je dostupno i Bobu i Alice
    - Alice će pokrenuti program, ne znajući šta sve on radi
    - Bob će dobijeni fajl skloniti na svoje privatno mesto
    - Bobov program može čak i obrisati Alicine fajlove
      - u audit logu piše da je Alice pokrenula program koji je obrisao fajlove

# Discretionary Access Control

- primer implementacije - *protection bits*: kontrola pristupa fajl-sistemu na UNIX/Linux operativnim sistemima
- svaki objekat u fajl-sistemu ima dodeljen atribut koji definiše pravila za kontrolu pristupa
- tri kategorije korisnika:
  - vlasnik objekta
  - grupa - grupa korisnika koji imaju zajednički pristup objektu
  - ostali - svi ostali korisnici
- za svaku kategoriju definisana su po tri bita
  - r - dozvola čitanja
  - w - dozvola pisanja
  - x - dozvola izvršavanja (za direktorijume: dozvola listanja)
- primer: `rwXr-X--X`
  - vlasnik ima sva tri prava na fajlu
  - grupa ima pravo čitanja i izvršavanja
  - ostali imaju pravo izvršavanja
- postoji samo jedna grupa za svaki fajl
- sistem administrator uređuje grupe korisnika

# Discretionary Access Control

---

- primer: relacije baze podataka
- objekti su
  - tabele
  - pogledi
- objekat ima vlasnika
- operacije su
  - select
  - insert
  - update
  - delete

# Discretionary Access Control

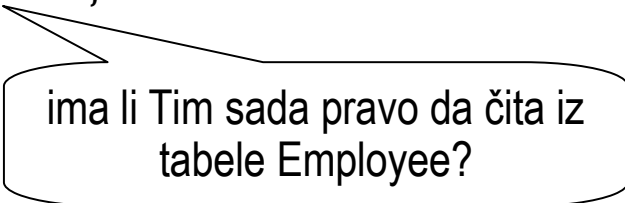
- kako organizovati administraciju?
  - centralizovana administracija
    - samo određeni privilegovani korisnici mogu da daju i oduzimaju prava pristupa
  - „vlasnička“ administracija
    - davanje i oduzimanje prava pristupa može da izvrši samo vlasnik objekta
    - delegiranje vlasničkih prava – vlasnik objekta može i drugim korisnicima dodeliti pravo da daju i oduzimaju prava pristupa

`GRANT Select ON Employee TO Tim WITH GRANT OPTION;`



# Discretionary Access Control

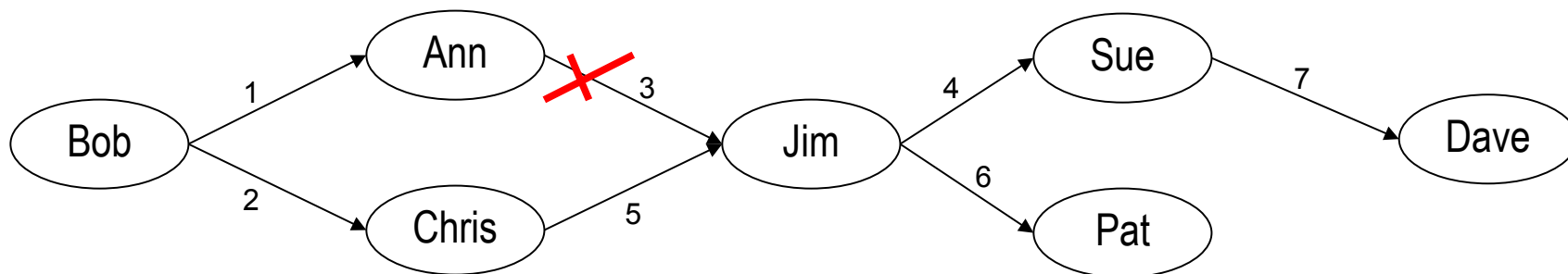
- SQL GRANT naredba: dodeljivanje prava pristupa  
(Bob): GRANT Select ON Employee TO Ann **WITH GRANT OPTION**;  
(Ann): GRANT Select ON Employee TO Tim;  
(Bob): GRANT Update, Insert ON Employee TO Tim;
- SQL REVOKE naredba: uklanjanje prava pristupa  
(Bob): REVOKE Select ON Employee TO Tim;



ima li Tim sada pravo da čita iz  
tabele Employee?

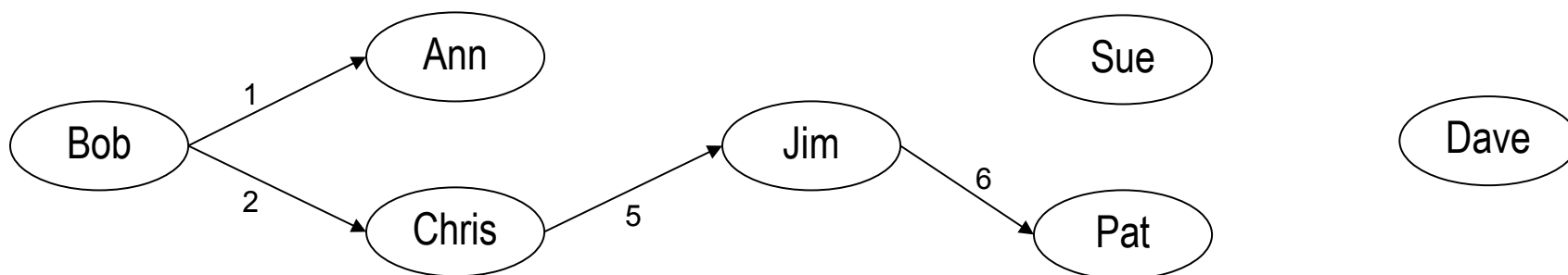
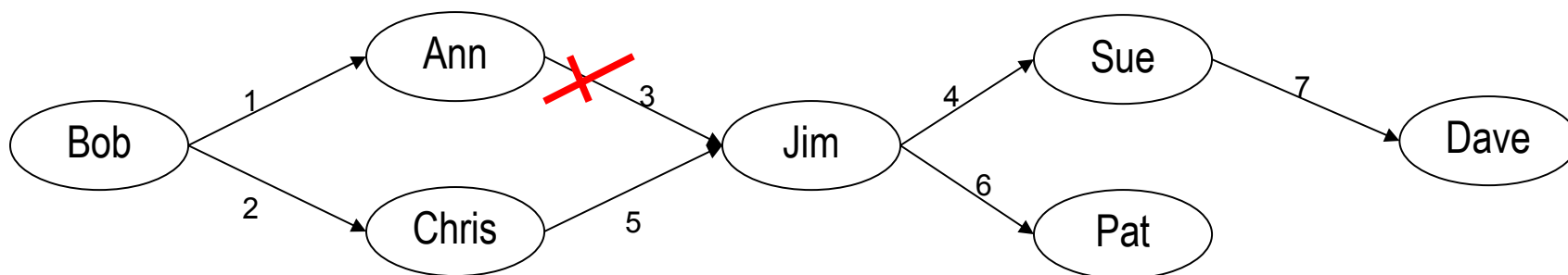
# Discretionary Access Control

- ukidanje prava može biti
  - kaskadno
  - kaskadno bez vremenske odrednice
  - nekaskadno
- primer
  - (brojevi predstavljaju hronologiju dodeljivanja prava)
  - Ann ukida pravo koje je dodelila Jimu



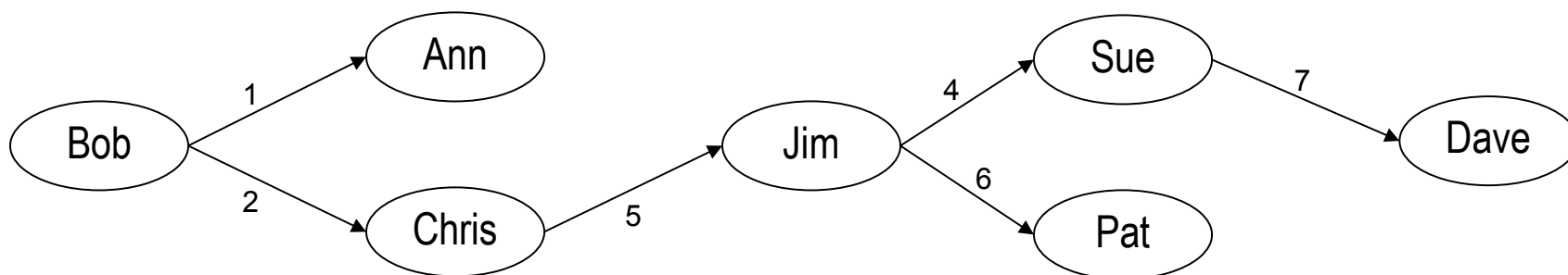
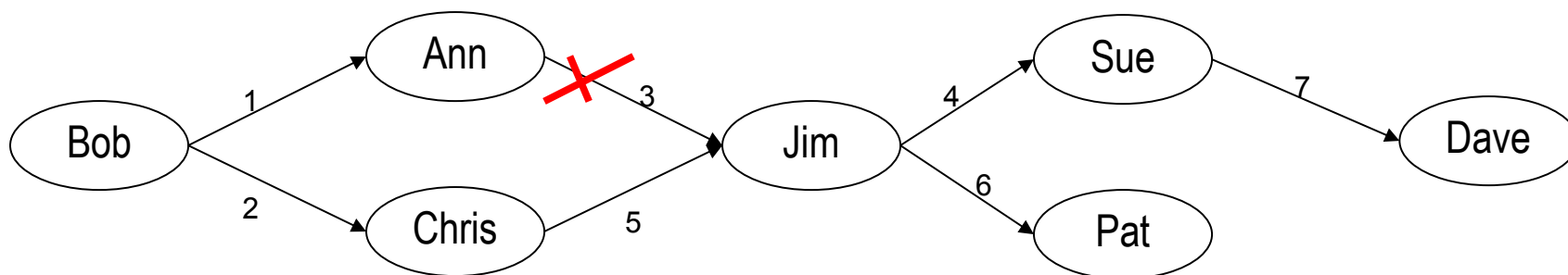
# Discretionary Access Control

- kaskadno ukidanje prava
  - prava se ukidaju tako da novo stanje odgovara situaciji kada se sprovede ista sekvenca dodela izuzimajući ukinutu
  - mora se voditi računa o redosledu dodeljivanja prava, odnosno o trenutku (timestamp) kada je neko pravo dodeljeno



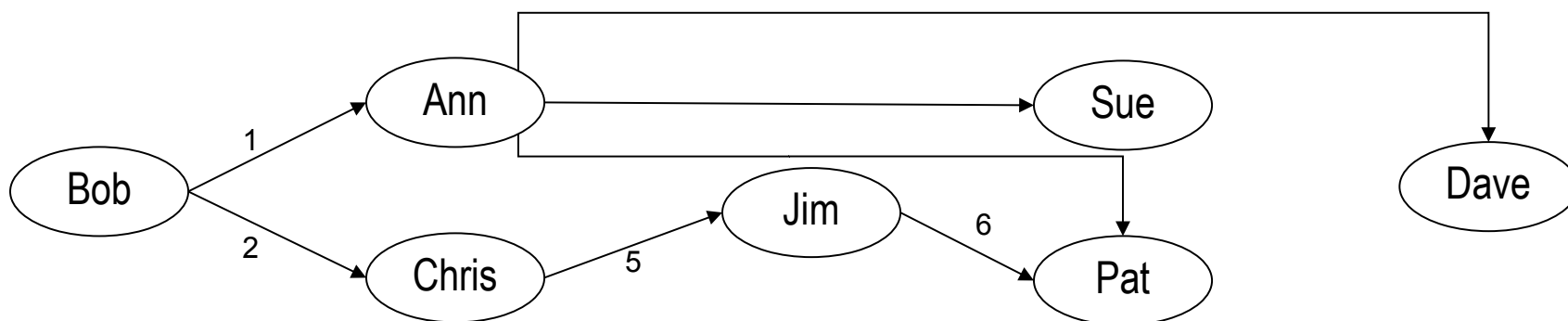
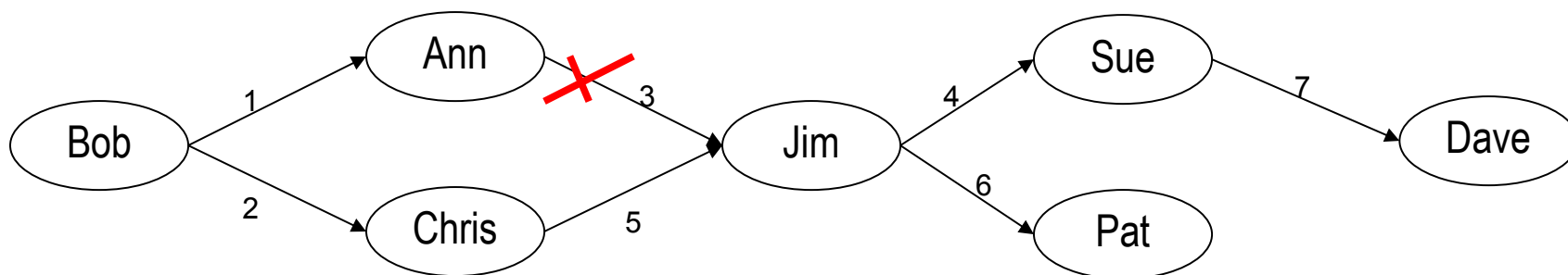
# Discretionary Access Control

- kaskadno bez vremenske odrednice
  - prava se ukidaju tako da se ne vodi računa o trenutku kada je pravo i dodeljeno



# Discretionary Access Control

- nekaskadno
  - ukidanje prava ne sme utiče na prava koja je dodelio korisnik koji ih je upravo izgubio (Jim)



# Mandatory Access Control

- bazirano na Bell-LaPadula modelu
- bezbednosni atributi korisnika i objekata imaju
  - hijerarhijsku komponentu - bezbednosni nivo (*clearance level*)
    - unclassified (U)
    - confidential (C)
    - secret (S)
    - top secret (TS)
  - nehijerarhijsku komponentu - dodeljene kategorije, npr.
    - NATO
    - NUCLEAR
    - ...
- primer
  - $TS \geq S \geq C \geq U$
  - $S(\text{NATO}, \text{NUCLEAR}) \geq S(\text{NUCLEAR}) \geq S$
- pravila „no read up“ i „no write down“
  - korisnik sa atributima  $S(\text{NUCLEAR})$  može da pristupa objektima sa atributima  $S(\text{NUCLEAR}), S, C, U$

# Mandatory Access Control

- model je odlučiv
- važno je razlikovati korisnika i subjekta (program)
  - Alice ima TS nivo: trebalo bi da može da čita i piše sve dokumente
  - Alice (korisnik) neće odavati TS informacije na nižim nivoima, ali možda program koji ona koristi (subjekat) hoće
  - Alice mora da promeni („spusti“) svoju sesiju na S nivo da bi pisala u fajlove na S nivou
- „no write down“ pravilo obezbeđuje od napada trojanskim konjem
  - korisnik ne može da piše u objekat koji je dostupan korisnicima sa nižim nivoom od njegovog
  - Alice ima nivo S(NUCLEAR)
  - Bob ima nivo S
  - Bob podmeće trojanskog konja Alice
    - program će moći da čita podatke sa nivoa S(NUCLEAR) kada ga pokrene Alice
    - ali neće moći da ih piše u fajl nivoa S (koga Bob može da čita)
- zaštita od brisanja trojanskim konjem nije rešena
  - Alice može Bobovim trojanskim konjem da pobriše sve svoje fajlove

# Mandatory access control

- primer: baze podataka
  - korisnici X, Y i Z sa nivoima poverljivosti:  
clearance(X) = TS  
clearance(Y) = S  
clearance(Z) = U
  - podaci u tabeli baze podataka su sledeći

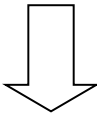
| Project Name | Topic                | Location        |  |
|--------------|----------------------|-----------------|--|
| Black, TS    | Databases, TS        | Los Angeles, TS |  |
| Silver, S    | Supply Chain, S      | New York, S     |  |
| Gold, U      | Inventories, S       | Atlanta, S      |  |
| Indigo, U    | Telecommunication, U | Austin, U       |  |



# Mandatory access control

- primer: podacima pristupa Y,  $\text{clearance}(Y) = S$

| Project Name | Topic                | Location        |  |
|--------------|----------------------|-----------------|--|
| Black, TS    | Databases, TS        | Los Angeles, TS |  |
| Silver, S    | Supply Chain, S      | New York, S     |  |
| Gold, U      | Inventories, S       | Atlanta, S      |  |
| Indigo, U    | Telecommunication, U | Austin, U       |  |

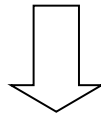


| Project Name | Topic                | Location    |  |
|--------------|----------------------|-------------|--|
| Silver, S    | Supply Chain, S      | New York, S |  |
| Gold, U      | Inventories, S       | Atlanta, S  |  |
| Indigo, U    | Telecommunication, U | Austin, U   |  |

# Mandatory access control

- primer: podacima pristupa Z,  $\text{clearance}(Z) = U$

| Project Name | Topic                | Location        |  |
|--------------|----------------------|-----------------|--|
| Black, TS    | Databases, TS        | Los Angeles, TS |  |
| Silver, S    | Supply Chain, S      | New York, S     |  |
| Gold, U      | Inventories, S       | Atlanta, S      |  |
| Indigo, U    | Telecommunication, U | Austin, U       |  |



| Project Name | Topic                | Location  |  |
|--------------|----------------------|-----------|--|
| Gold, U      | -, U                 | -, U      |  |
| Indigo, U    | Telecommunication, U | Austin, U |  |

# Mandatory access control

- primer: Z hoće da doda novi red (Silver, Linear Programming, Omaha)

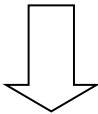
| Project Name | Topic                 | Location        |  |
|--------------|-----------------------|-----------------|--|
| Black, TS    | Databases, TS         | Los Angeles, TS |  |
| Silver, S    | Supply Chain, S       | New York, S     |  |
| Gold, U      | Inventories, S        | Atlanta, S      |  |
| Indigo, U    | Telecommunication, U  | Austin, U       |  |
| Silver, U    | Linear Programming, U | Omaha, U        |  |

- problem: ponavljanje podataka sa istim ključem!

# Mandatory access control

- primer: Z hoće da zameni NULL vrednosti konkretnim podacima (Markov Chain, New Jersey)

| Project Name | Topic                | Location  |  |
|--------------|----------------------|-----------|--|
| Gold, U      | -, U                 | -, U      |  |
| Indigo, U    | Telecommunication, U | Austin, U |  |



| Project Name | Topic                | Location        |  |
|--------------|----------------------|-----------------|--|
| Black, TS    | Databases, TS        | Los Angeles, TS |  |
| Silver, S    | Supply Chain, S      | New York, S     |  |
| Gold, U      | Inventories, S       | Atlanta, S      |  |
| Indigo, U    | Telecommunication, U | Austin, U       |  |
| Gold, U      | Markov Chain, U      | New Jersey, U   |  |

# Biba model

- MAC model se fokusira na poverljivost podataka, a zapostavlja integritet
- Biba model (1977) fokusira se na integritet a zapostavlja poverljivost
- koncept nivoa integriteta (sa hijerarhijskom i kategorizacijskom komponentom)
  - za korisnika: indikacija nivoa poverenja u korisnika u pogledu menjanja podataka na datom nivou
  - za objekat: osetljivost objekta na izmene
- primer nivoa
  - critical (C)
  - important (I)
  - ordinary (O)
- pravila za kontrolu pristupa su inverzna u odnosu na Bell-LaPadula model
  - subjekat S može da čita objekat O ako je  $\text{clearance}(S) \leq \text{classification}(O)$
  - subjekat S može da piše u objekat O ako je  $\text{clearance}(S) \geq \text{classification}(O)$

# Clark-Wilson model

- Clark-Wilson 1987: u komercijalnim (ne-vojnim) primenama daleko je važniji integritet nego poverljivost
  - integritet: podaci se menjaju samo na ispravan način od strane autorizovanih korisnika
- novi koncepti
  - dobro formirana transakcija (*well-formed transaction*, WFT)
    - sistem ograničava korisnika na promene podataka samo pomoću odgovarajućih transakcija
    - podaci iz jednog validnog stanja mogu preći u drugo validno stanje
  - razdvajanje zaduženja (*separation of duty*, SoD)
    - osigurava konzistentnost izmena u podacima
    - npr. nabavku zahteva korisnik A, odobrava korisnik B, kontroliše (nadgleda) korisnik C

# Clark-Wilson model

---

- osnovna jedinica kontrole pristupa je uređena trojka
  - korisnik (*user*)
  - transformaciona procedura (*transformation procedure*, TP)
  - podatak sa ograničenim pristupom (*constrained data item*, CDI)
- pored toga, postoji i
  - podatak bez ograničenja pristupa (*unconstrained data item*, UDI)
  - procedura za proveru integriteta (*integrity verification procedure*, IVP) - utvrđuje da li je podatak u validnom stanju

# Clark-Wilson model

- devet pravila za obezbeđivanje integriteta podataka
  1. za svaki CDI mora postojati IVP koja proverava da li je CDI u validnom stanju
  2. svaka TP koja menja CDI mora biti sertifikovana da ga menja isključivo na validan način
  3. CDI može da menja samo sertifikovana TP
  4. svaka TP mora da vodi dnevnik promena koje sprovodi nad CDI
  5. svaka TP koja kao ulaz ima UDI mora da transformiše UDI u CDI isključivo na validan način
  6. samo sertifikovane TP mogu da menjaju UDI
  7. korisnik može da pristupi CDI samo kroz TP za koju je autorizovan
  8. svaki korisnik mora biti autentifikovan pre pozivanja TP
  9. samo bezbednosni administrator može da autorizuje korisnika da poziva TP
- TP - transformaciona procedura (*transformation procedure*)
- CDI - podatak sa ograničenim pristupom (*constrained data item*)
- UDI - podatak bez ograničenja pristupa (*unconstrained data item*)
- IVP - procedura za proveru integriteta (*integrity verification procedure*)



# Clark-Wilson model

---

- Bell-LaPadula model kontroliše tok podataka pomoću kontrole operacija čitanja i pisanja niskog nivoa
- Clark-Wilson model teži da se podaci menjaju samo na autorizovan način od strane autorizovanih korisnika
  - ovo se ne može implementirati samo na nivou kernela
  - primer: baza podataka gde tabele nisu neposredno dostupne korisnicima, nego samo uskladištene procedure

# Clark-Wilson model

- razdvajanje zaduženja (SoD) - način da sprečimo da autorizovani korisnici naprave pogrešne izmene u podacima
- npr. kupovina robe
  - formiranje i slanje porudžbine (korisnik A)
  - evidentiranje prispeća robe (korisnik B)
  - odobravanje plaćanja (korisnik C)
- ako bi sve ove podoperacije radila ista osoba moguće su prevare
- ako ove podoperacije rade različite osobe, moguća je prevara, uz „zločinačko udruživanje“

# Politika kineskog zida

- osnovna namera: sprečiti tok podataka koji mogu izazvati konflikt interesa
- primer
  - finansijski konsultanti dobijaju privatne podatke svojih klijenata
  - ako konsultant poznaje privatne podatke dvaju banaka može to da zloupotrebi
    - za privatni profit
    - za dobrobit jedne banke a na štetu druge
- privatni podaci o organizaciji se nalaze u jednoj od (međusobno disjunktih) kategorija za konflikt interesa (COI)
- svaka organizacija pripada u tačno jedan COI
- svaki COI sadrži bar dve organizacije, koje se bave istom ili sličnom delatnošću
- konsultant ne može da pristupi podacima više od jedne organizacije iz istog COI
  - ako pristupi privatnim podacima jedne organizacije iz datog COI, ne može da pristupa podacima drugih organizacija iz istog COI
- rešenje za kontrolu operacije čitanja, ne i pisanja
  - pisanjem se bavi sledeći model...

# Brewer-Nash model

---

- svaka organizacija je predstavljena skupom podataka
- skupovi podataka su smešteni u COI
- pravilo za čitanje podataka: subjekat S ima pravo da čita objekat O ako je zadovoljeno nešto od sledećeg
  - O je u istom skupu podataka kao i neki drugi objekat koga je S već čitao
  - O pripada COI iz koga S još nije čitao ništa
- pravilo za pisanje podataka: subjekat S ima pravo da piše u objekat O ako je zadovoljeno sve od sledećeg
  - S može da čita O prema pravilu za čitanje
  - nijedan objekat iz drugih skupova u odnosu na skup kome pripada O nije dostupan za čitanje

# Brewer-Nash model

- pravilo za pisanje je zamišljeno kao zaštita od trojanskih konja
  - Alice ima pravo
    - čitanja podataka energetske kompanije A i
    - čitanja i pisanja podataka banke A
  - Bob ima pravo
    - čitanja podataka energetske kompanije B i
    - čitanja podataka banke B
  - trojanski konj koji je pokrenula Alice bi mogao da pokuša da
    - čita podatke o banci A i
    - piše te podatke u banku B (ne može)
    - piše te podatke u energetska kompaniju B (ne može)

# Domain Type Enforcement model

---

- subjekti = aktivni entiteti (procesi, programi)
- subjektu se dodeljuje **domen**
- objekti = pasivni entiteti (fajlovi, uređaji, delovi memorije)
- objektu se dodeljuje **tip**
- dozvole se vezuju za domene i tipove
  - domen-domen dozvole
    - izražene tabelarno: domain-domain access control table (DDACT)
  - domen-tip dozvole
    - izražene tabelarno: domain-type acces control table (DTACT)
- u ćelijama tabele nalazi se skup dodeljenih prava

# Domain Type Enforcement model

---

- primer: fajl sistem
  - domen-domen dozvole: create (C) i kill (K)
  - domen-tip dozvole: read (R), write (W), execute (E), browse directory (T)
  - proces A može da pokrene proces B samo ako postoji pravo C u ćeliji tabele koja povezuje A i B
- slično kao i Lampson model (matrica pristupa) ali je matrica znatno manja zbog grupisanja procesa u domene i objekata u tipove

# Role Based Access Control

---

- uloga (role) ~ radno mesto u organizaciji
- razvoj krajem 1980-tih i početkom 1990-tih
  - Dobson-McDermid
  - Baldwin
  - Nash-Poland
- 1992: NIST studija
  - stanje u privatnom i javnom sektoru: koristi se DAC
  - DAC nije najbolje prilagođen potrebama:
    - stvarni vlasnik podataka nije korisnik već organizacija
    - diskreciona kontrola nije poželjna
  - konvencionalni MAC ne odgovara potrebama
    - potrebna je kontrola pristupa bazirana na kompetenciji
    - sprečavanje konflikta interesa



# Role Based Access Control

---

- inicijalni RBAC model: Ferraiolo-Kuhn 1992.
- tri pravila
  1. dodela uloga
    - subjekat može da izvrši transakciju samo ako mu je dodeljena uloga ili je izabrao neku ulogu
    - autentifikacija nije transakcija
    - sve posle autentifikacije se izvršava u obliku transakcija za koje su potrebne uloge
  2. autorizacija uloga
    - subjekat može da koristi samo uloge koje su mu autorizovane
  3. autorizacija transakcija
    - subjekt može da izvršava transakciju samo ako je transakcija autorizovana za korisnikovu aktivnu ulogu

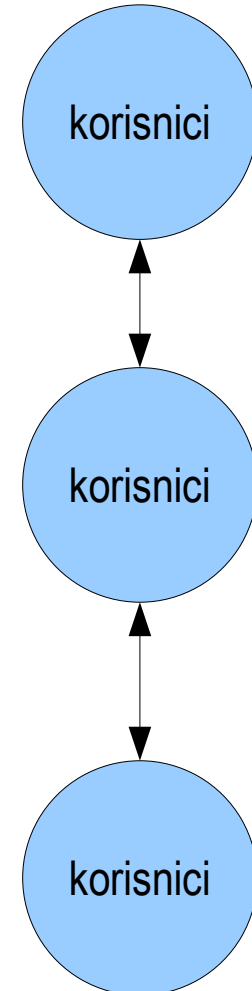
# Role Based Access Control

- formalna definicija pravila
  - aktivna uloga subjekta  $s$ :  $AR(s)$
  - skup uloga koje može izabrati  $s$ :  $RA(s)$
  - skup transakcija koje može pokrenuti uloga  $r$ :  $TA(r)$
  - $exec(s, t) = \text{true}$  akko subjekat  $s$  može da pozove transakciju  $t$
  - dodela uloga:  $exec(s, t) \Rightarrow AR(s) \neq \emptyset$
  - autorizacija uloga:  $AR(s) \subseteq RA(s)$
  - autorizacija transakcija:  $exec(s, t) \Rightarrow t \in TA(AR(s))$

ovo je implikacija, a ne ekvivalencija - to nam omogućava da uvedemo dodatna ograničenja na mogućnost pozivanja transakcija

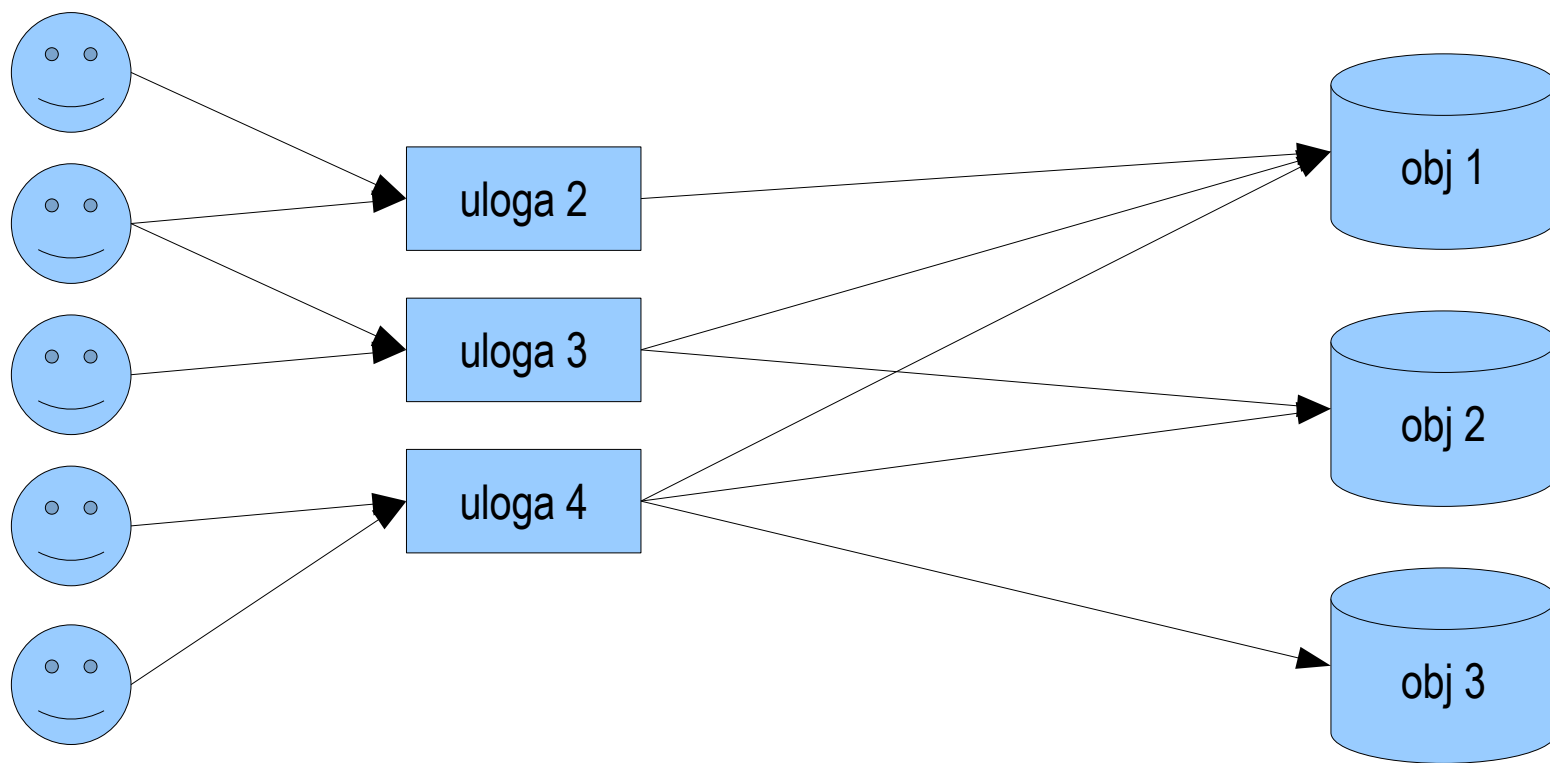
# Role Based Access Control

- uloga predstavlja skup dozvola
- korisnicima se dodeljuje jedna ili više uloga
- uloga  $\neq$  grupa
  - uloga je skup dozvola
  - grupa je skup korisnika



# Role Based Access Control

- u okviru organizacije uloge se retko menjaju
- dok se korisnici i dozvole mogu menjati češće
- pojednostavljena administracija



izbegava se „kloniranje korisnika“

# Role Based Access Control

- ušteda u administrativnim aktivnostima:
  - U - skup korisnika na istom radnom mestu
  - P - skup dozvola potrebnih za obavljanje zadataka na tom radnom mestu
  - broj veza potrebnih za direktno povezivanje svih korisnika i njihovih dozvola:  $|U| \cdot |P|$
  - uloga = skup dozvola
  - broj veza korisnik-uloga i uloga-dozvole je  $|U| + |P|$
  - ako je  $|U| + |P| < |U| \cdot |P|$  tada postoji ušteda
  - uslov je zadovoljen za  $|U| > 2, |P| > 2$
  - ako ima ukupno  $n$  radnih mesta, ušteda postoji kada je 
$$\sum_{i=1}^n (|U_i| + |P_i|) < \sum_{i=1}^n (|U_i| \cdot |P_i|)$$

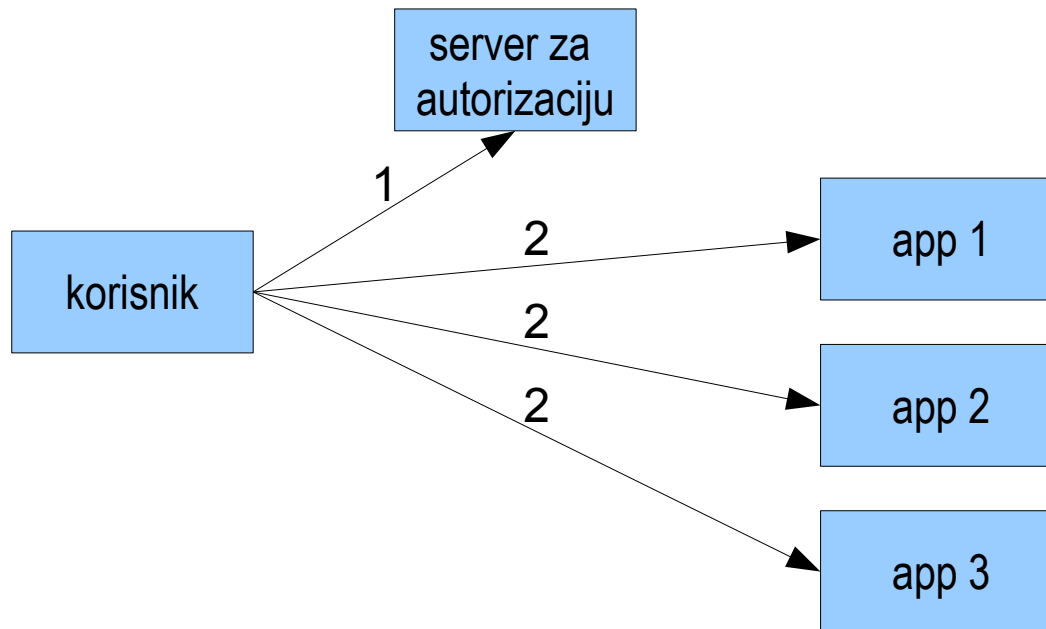
# Role Based Access Control

---

- administracija je obično centralizovana
  - na jednom serveru su definisane uloge, dozvole i korisnici
- dva pristupa centralizaciji
  - user pull
  - server pull

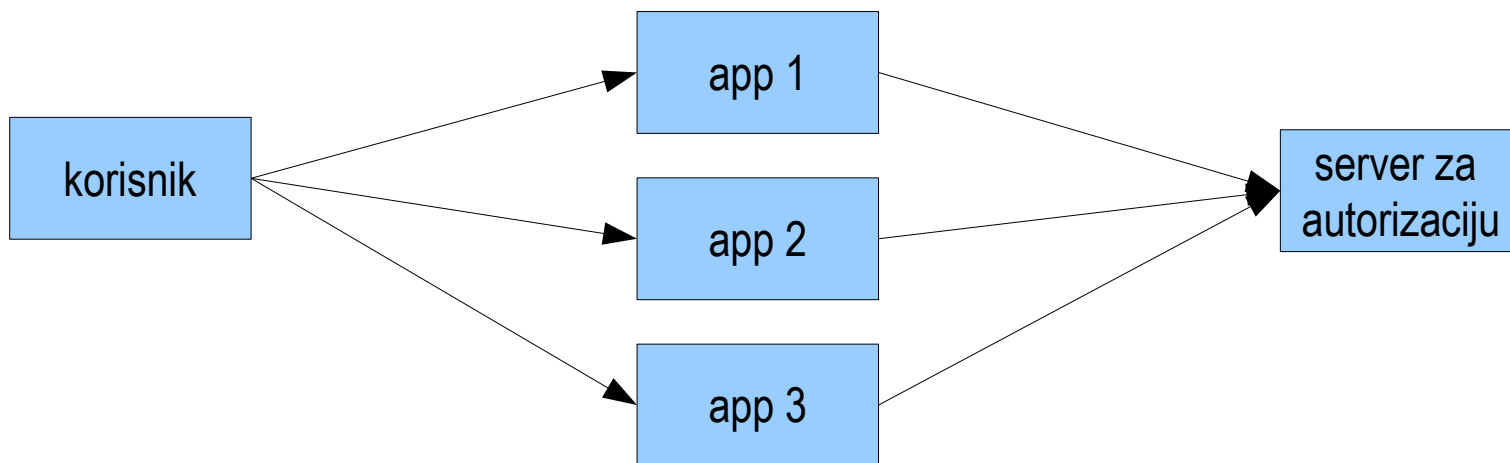
# Role Based Access Control

- *user pull* centralizovana autorizacija
  - korisnik se prijavljuje na server za autorizaciju
  - od njega dobija neke podatke kojima će se predstaviti aplikacijama
  - prilikom obraćanja aplikacijama dostavlja i te podatke



# Role Based Access Control

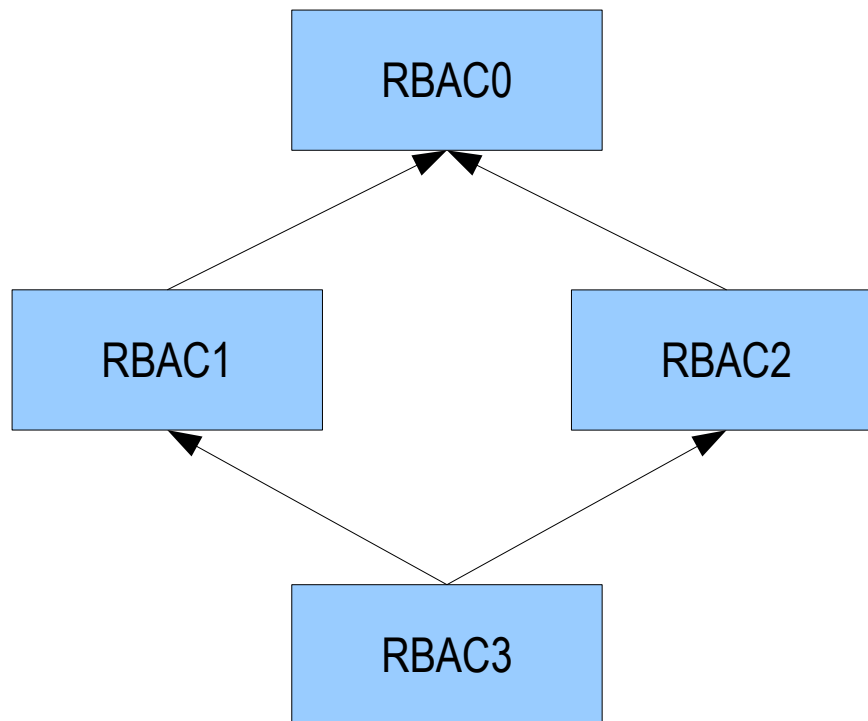
- server pull centralizovana autorizacija
  - aplikacije su zadužene za autentifikaciju
  - podaci o pravima su centralizovani na serveru
  - kada korisnik pristupi aplikaciji, aplikacija se obraća serveru radi dobijanja njegovih dozvola





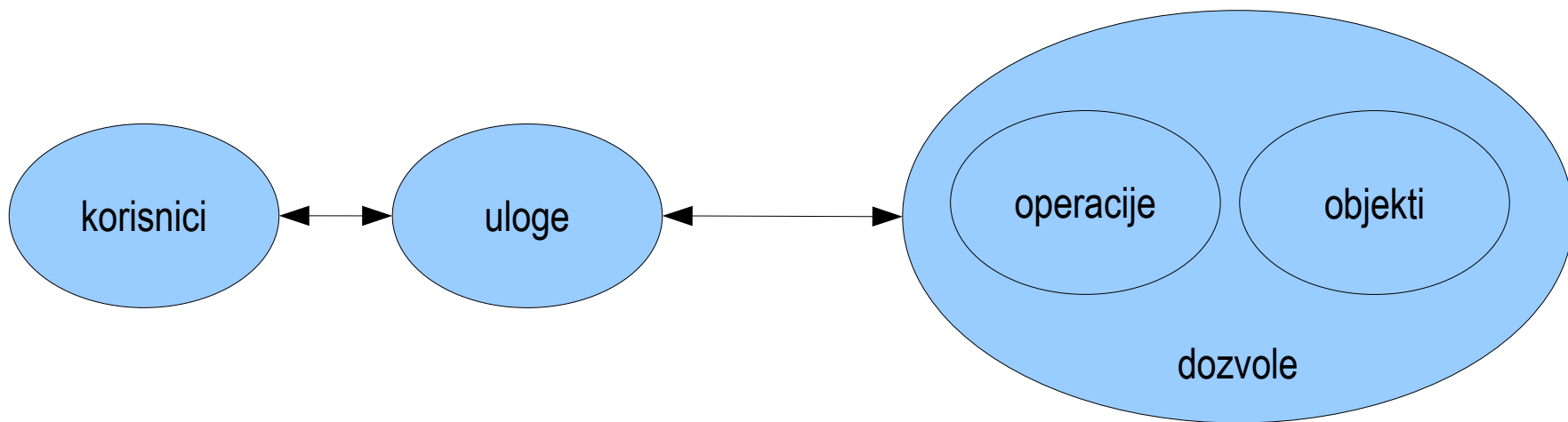
# RBAC varijante

- Sandhu 1996: četiri varijante RBAC-a
  - RBAC0: osnovni elementi RBAC sistema
  - RBAC1: RBAC0 + hijerarhije uloga
  - RBAC2: RBAC0 + ograničenja (SoD)
  - RBAC3: RBAC1 + RBAC2



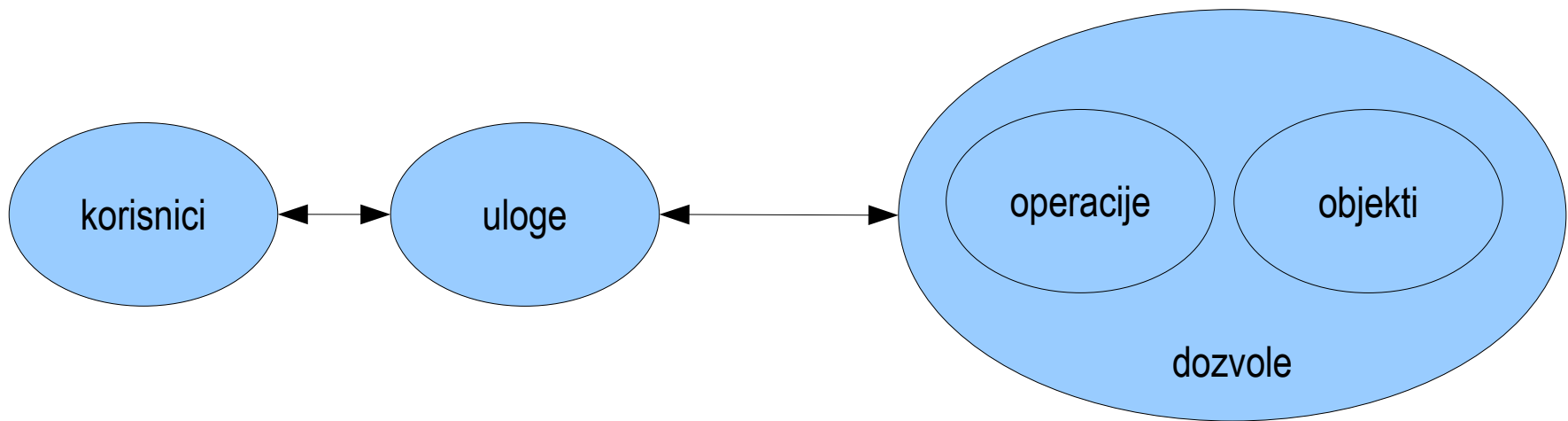
# Osnovni RBAC

- pet koncepata
  - korisnici
  - uloge
  - dozvole, koje se sastoje iz
    - operacija dopuštenih nad
    - objektima



# Osnovni RBAC

- veze su n:m
  - korisnik može imati više uloga
  - uloga može imati više korisnika
  - uloga može imati više dozvola
  - dozvola može biti u više uloga



# Osnovni RBAC

- granularnost dozvola se može birati prema potrebama
  - dozvola se može posmatrati kao atomična operacija u sistemu
  - operacije mogu biti implementirane kao transakcije
- primer: šalterski radnik u banci
  - može da isplati novac sa računa (*withdraw*) ili da uplati novac na račun (*deposit*)
    - trebaju mu read i write prava za podatke o računima
  - ne može da ispravlja ništa nakon obavljene transakcije
- supervizor u banci
  - može da ispravi rezultat neke transakcije
    - trebaju mu read i write prava za podatke o računima
  - ali ne može samostalno da pozove *withdraw* ili *deposit*

# Osnovni RBAC

---

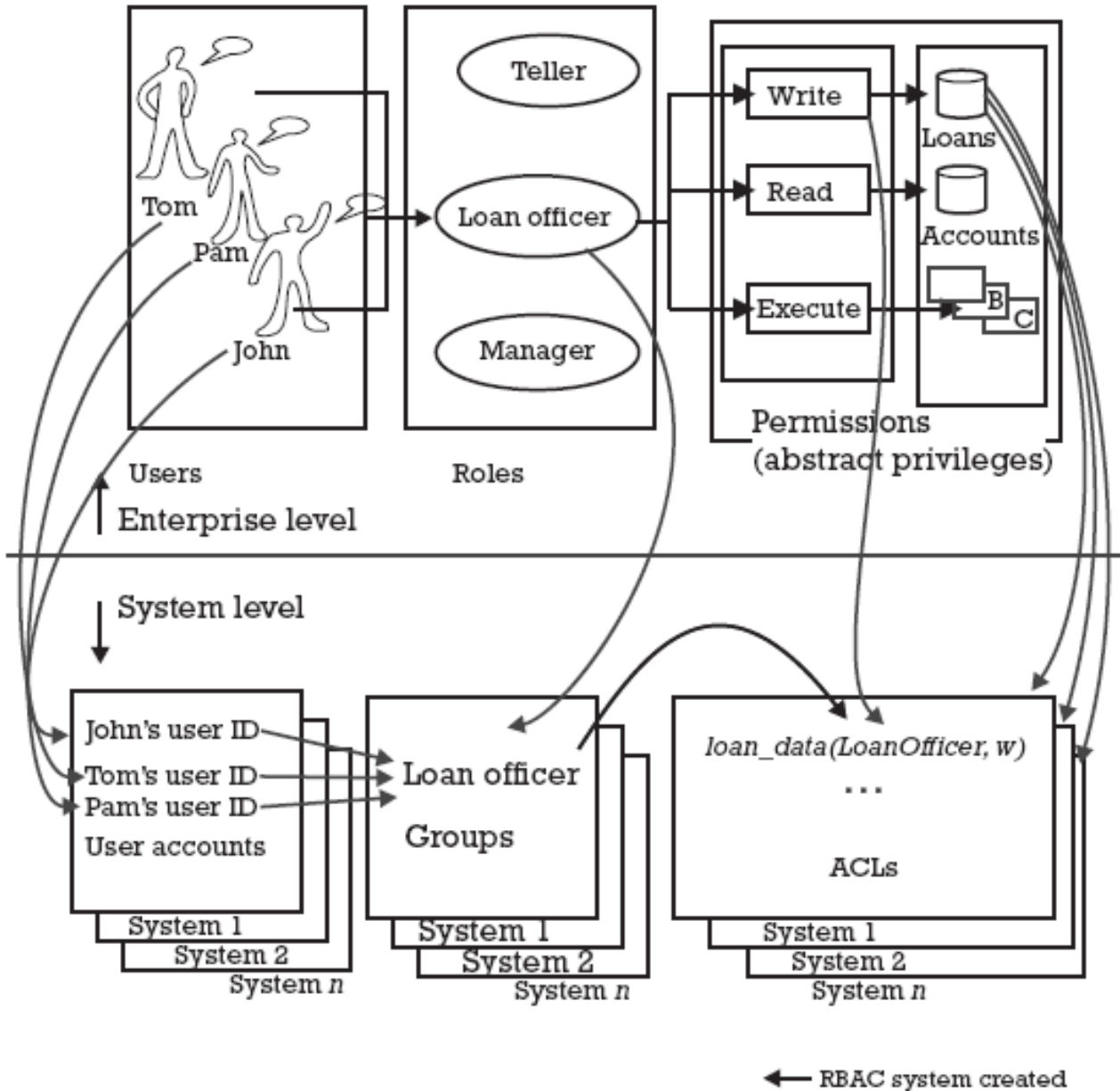
- veza korisnik - subjekat je 1:n
  - subjekat je tipično aktivni entitet - npr. program
  - subjekat se vezuje za jednu sesiju
  - korisnik može imati više istovremenih subjekata
- korisnik može imati više uloga
- subjekat može aktivirati podskup mogućih uloga dodeljenih korisniku
  - radi ispunjenja principa minimalnih privilegija
- mogućnost izbora aktivnih uloga za subjekta predstavlja dinamičku komponentu osnovnog RBAC-a

# Osnovni RBAC

---

- RBAC predstavlja apstraktan model
- njegova implementacija zavisi od korišćene tehnologije
- koncepti RBACa moraju se mapirati na koncepte sistema
  - prava pristupa fajlovima, korisnici, grupe, itd.

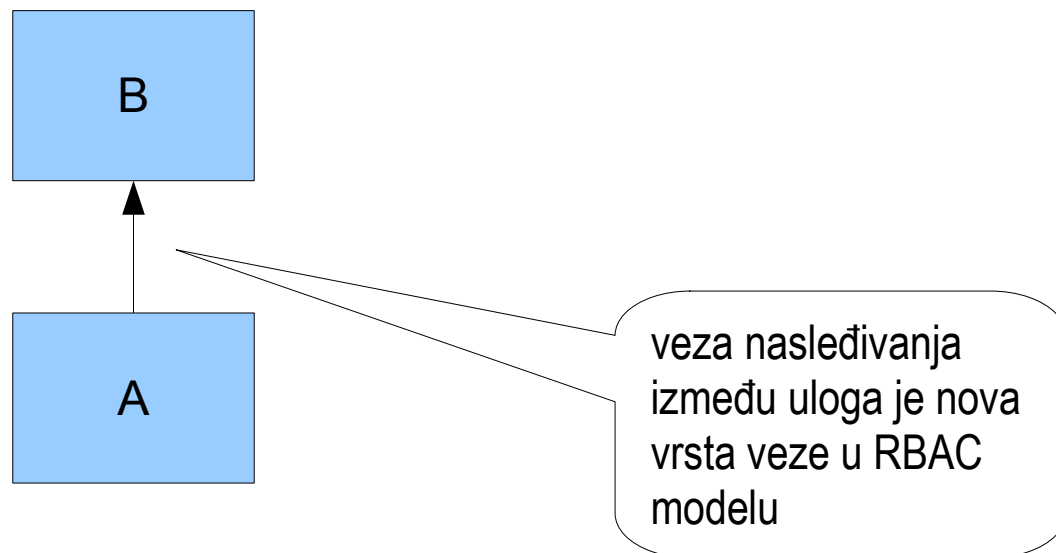
# Osnovni RBAC



mapiranje  
apstraktnih dozvola  
iz poslovnog modela  
na stvarne ACL liste  
na nivou sistema

# Hijerarhijski RBAC

- hijerarhija uloga
  - uloge mogu da nasleđuju dozvole od drugih uloga
  - ako uloga A nasleđuje ulogu B, tada sve dozvole koje ima uloga B pripadaju i ulozi A
    - tj. skup dozvola u B je podskup skupa dozvola u A





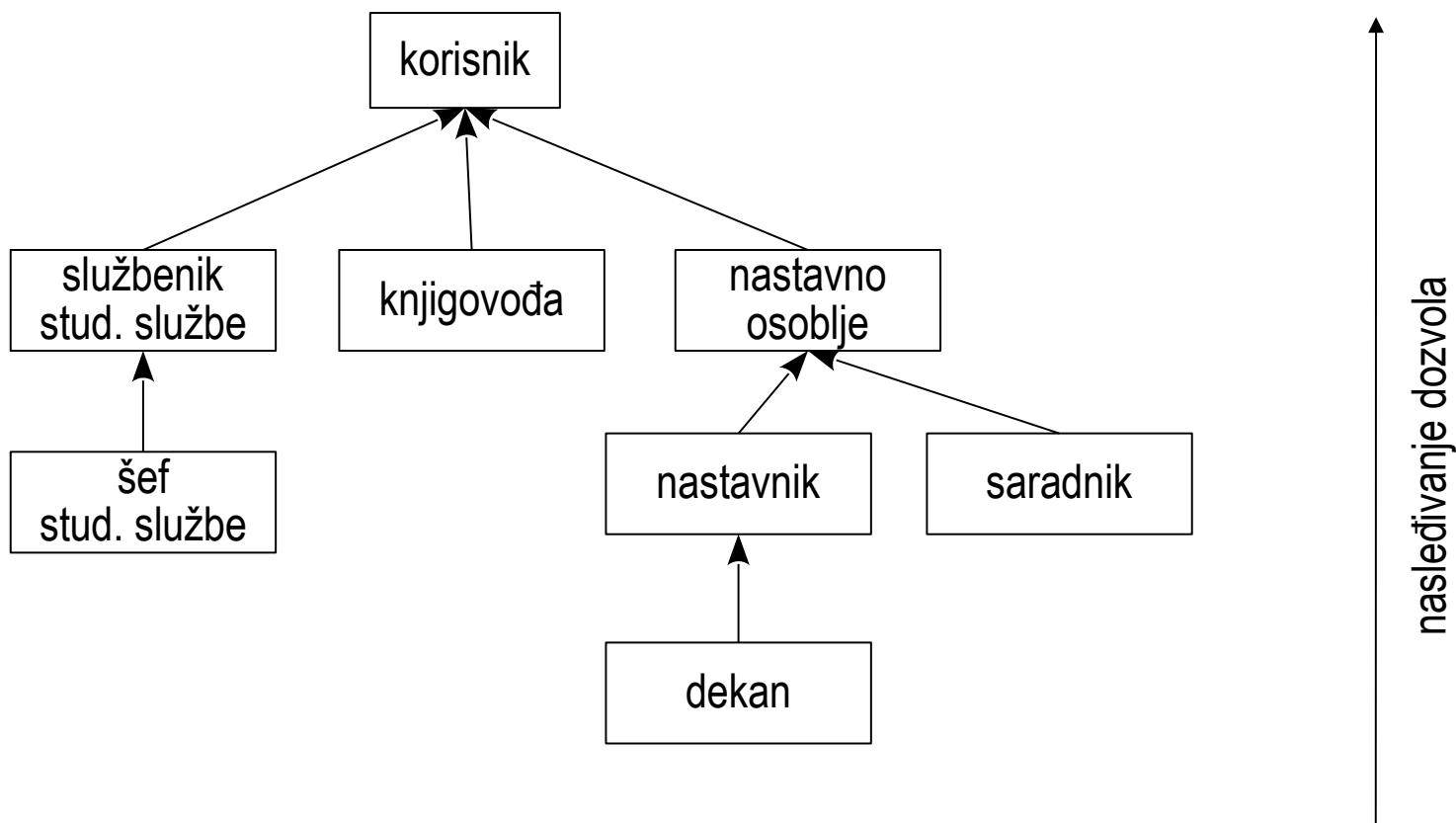
# Hijerarhijski RBAC

---

- hijerarhije uloga su prirodno sredstvo za strukturiranje uloga tako da odslikavaju organizaciju, raspodelu zaduženja i odgovornosti
- iako inicijalno zahteva složeniju pripremu, korišćenje hijerarhije uloga se isplati na duži rok kroz jednostavniju administraciju

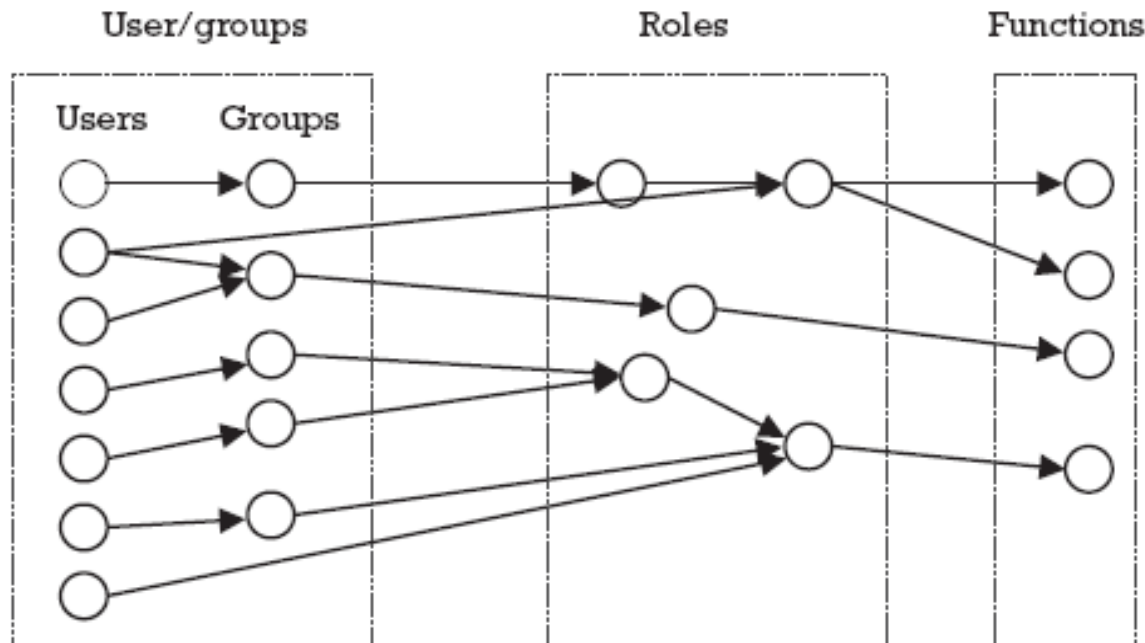
# Hijerarhijski RBAC

- motivacija za formiranje hijerarhije uloga: uloge u organizaciji često imaju preklapajuće funkcije



# Hijerarhijski RBAC

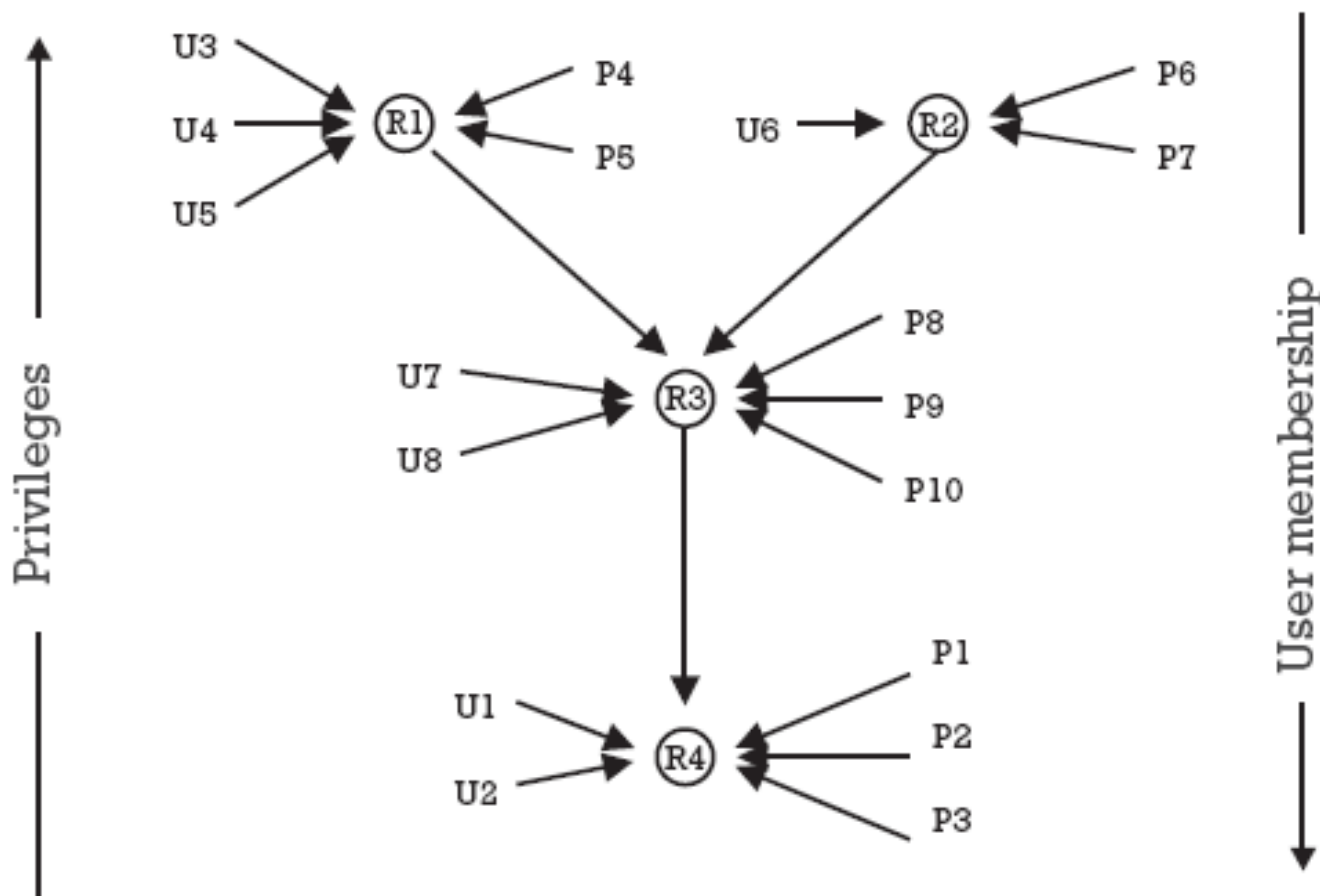
- šeme nasleđivanja
  - direktno nasleđivanje dozvola
    - uloga je imenovani skup dozvola
    - uloga  $r_2$  nasleđuje ulogu  $r_1$  ako je skup dozvola  $r_1$  podskup skupa dozvola  $r_2$
    - korisnici (i grupe korisnika) definišu se odvojeno
    - Baldwinov graf privilegija



moгуćnost redundanse:  
korisnik može dobiti  
ulogu direktno ili preko  
grupe (grupe se  
administriraju odvojeno  
od RBAC-a!)

# Hijerarhijski RBAC

- šeme nasleđivanja
  - nasleđivanje dozvola i korisnika
    - uloga obuhvata i dozvole i korisnike
    - nasleđivanje obuhvata i korisnike



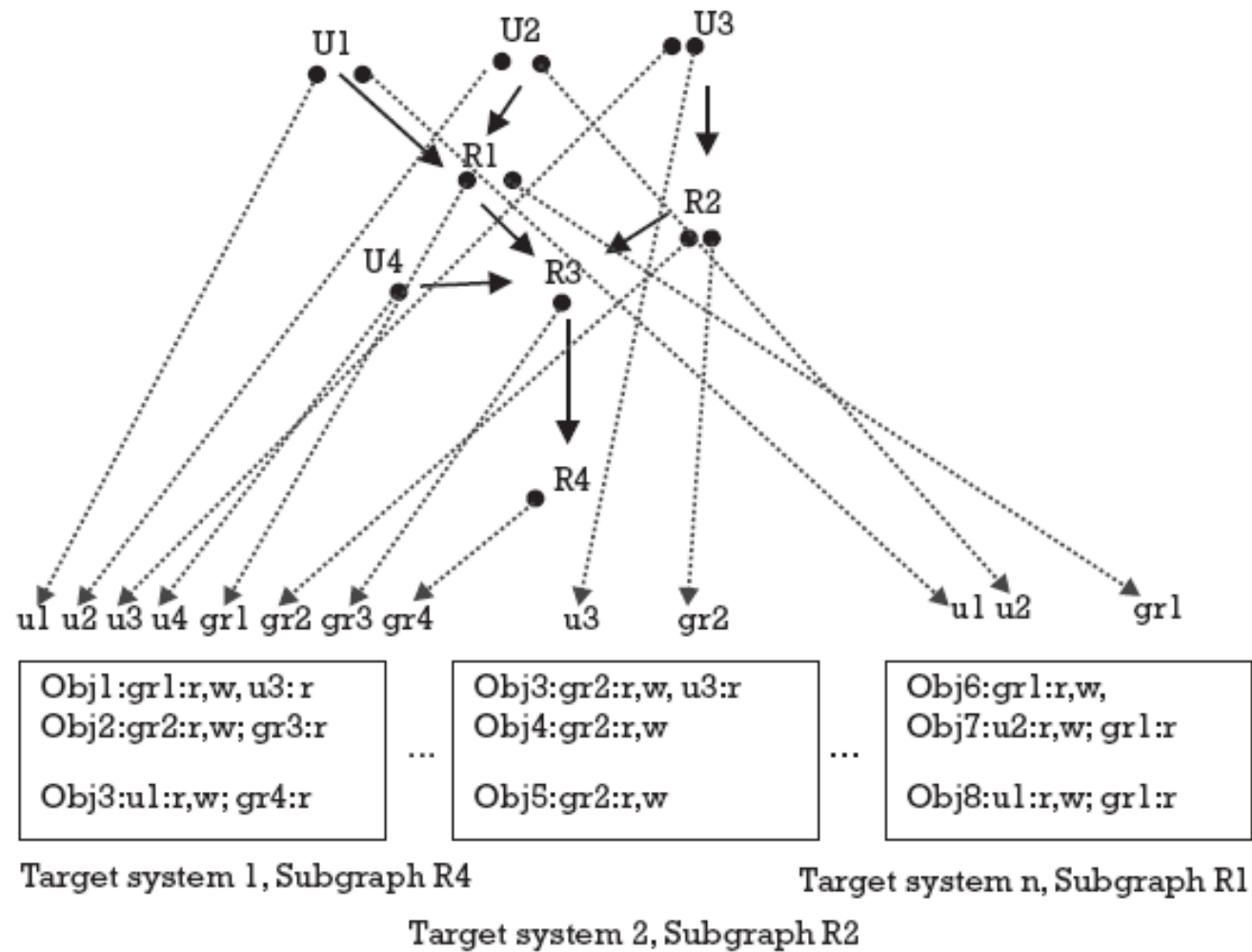
korisnik U3 ima  
dozvole P1-P5,  
P8-P10

# Hijerarhijski RBAC

- šeme nasleđivanja
  - 3. nasleđivanje korisnika
    - dozvole se dodeljuju grupama korisnika
    - grupe se mapiraju na uloge
    - uloge su vezane u hijerarhiju
    - uloga  $r_1$  „sadrži“ ulogu  $r_2$  ako svi korisnici koji imaju  $r_1$  imaju i  $r_2$
  - dodeljivanje uloge  $r$  korisniku obuhvata
    - dodeljivanje korisnika svim grupama koje se mapiraju na  $r$
    - i dodeljivanje korisnika svim grupama koje se mapiraju na role koje  $r$  sadrži

# Hijerarhijski RBAC

- šeme nasleđivanja
  - 3. nasleđivanje korisnika
    - primer:



Apstraktni korisnici U1-U4 se implementiraju u sistemu kao u1-u4.

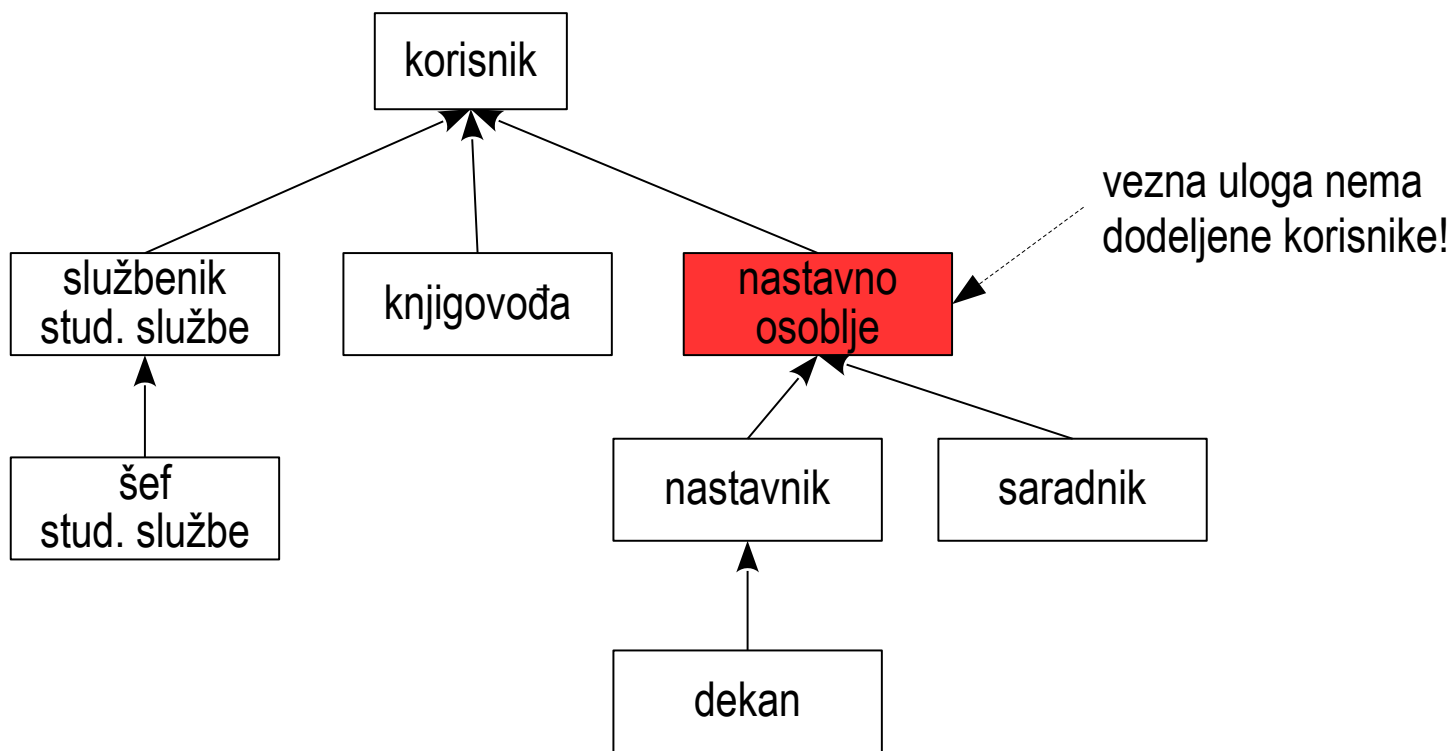
Grupe gr1-gr4 odgovaraju ulogama R1-R4.

Grupe sadrže sledeće:  
gr1 ima u1 i u2  
gr2 ima u2  
gr3 ima u1-u4  
gr4 ima u1-u4

Formalno ne postoji nasleđivanje dozvola, ali se nasleđivanjem korisnika postiže ekvivalentan rezultat!

# Hijerarhijski RBAC

- vezne uloge ~ apstraktne klase
  - zgodno za grupisanje dozvola
  - za ograničavanje nasledenih dozvola



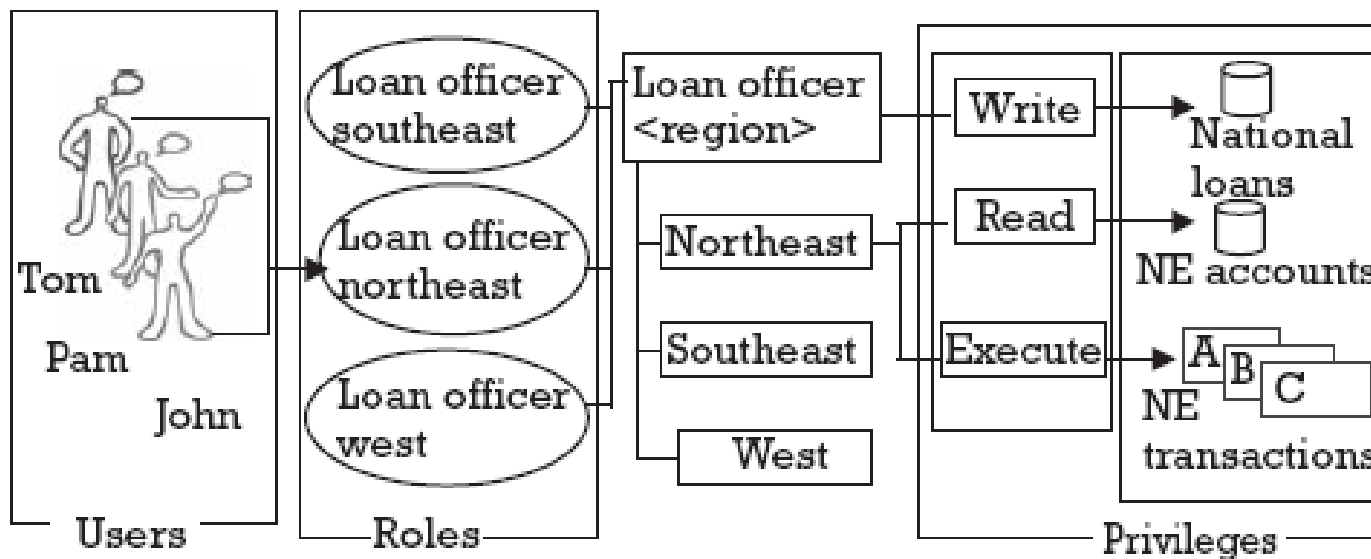
# Hijerarhijski RBAC

- više hijerarhija u jednoj organizaciji
  - organizaciona hijerarhija
  - teritorijalna hijerarhija
- npr. šalterski službenik ima read/write prava nad podacima o bankovnim računima, ali samo za korisnike koji su otvorili račun u jednoj filijali
- morali bismo definisati posebne uloge za jedno isto radno mesto ali za svaku filijalu posebno!
  - nepraktično
- uvodimo **tipove uloga** (*role types*): uloga koja ima kvalifikator
  - kvalifikatori: lokacija, org. jedinica, region, itd.



# Hijerarhijski RBAC

- tipovi uloga
  - primer
    - tip uloge: kreditni savetnik
    - konkretne uloge: kreditni savetnik vezan za konkretni region
    - globalnim podacima o kreditima mogu svi da pristupaju, a lokalnim samo pojedine uloge



implementacija tipova uloga može se napraviti pomoću običnog nasleđivanja!

# Hijerarhijski RBAC

---

- ograničene (*limited*) hijerarhije ~ jednostruko nasleđivanje
- opšte (*general*) hijerarhije ~ višestruko nasleđivanje
- ograničene su mnogo raširenije u komercijalnim proizvodima

# RBAC sa ograničenjima

- razdvajanje zaduženja (SoD): kritične operacije obavljaju dva ili više lica, tako da bezbednost ne zavisi od jedne osobe
  - otvaranje sefa u banci: klijent i službenik
  - „two-man rule“ za aktivaciju nuklearnog oružja
  - uplata i isplata u knjigovodstvu

## Separation of Duties

### I. Disbursement of Funds

The following minimum separation of duties applies to individuals in departments and accounting offices who are responsible for the disbursement of funds.

The following duties shall be performed by different individuals:

1. Check request reviewer—evaluates requests with respect to business purpose, applicable policy, backup documentation, and authorized signature.
2. Check preparer—prepares checks and ledger entries.
3. Check issuer—has checks signed and approves ledger entry.
4. Check deliverer—distributes checks or sends to payees.
5. Ledger reviewer—reconciles bank statement with general ledger cash account.

### II. Depository Funds

The following minimum separation of duties applies to individuals in departments and accounting offices who are responsible for depository funds.

The following duties shall be performed by different individuals:

1. Mail handler—opens mail, reviews, and endorses checks.
2. Cashier—processes cash, determines account coding, and deposits in bank account or delivers to another cashier.
3. Auditor—ensures that all checks received are deposited and accounts coded correctly; also receives checks returned to the office.
4. Ledger reviewer—reconciles department accounting records with accounting office records.

# RBAC sa ograničenjima

---

- statički SoD
  - ograničenja se postavljaju u trenutku kada se korisniku dodeli uloga
  - npr. ako je korisniku dodeljena uloga A, ne sme mu biti dodeljena uloga B
- dinamički SoD
  - ograničenja se postavljaju u trenutku kada korisnik koristi sistem (u toku sesije)
  - npr. korisnik ne sme imati istovremeno aktivne uloge A i B
    - obe uloge mu mogu biti statički dodeljene
    - ali ih ne može imati istovremeno aktivne

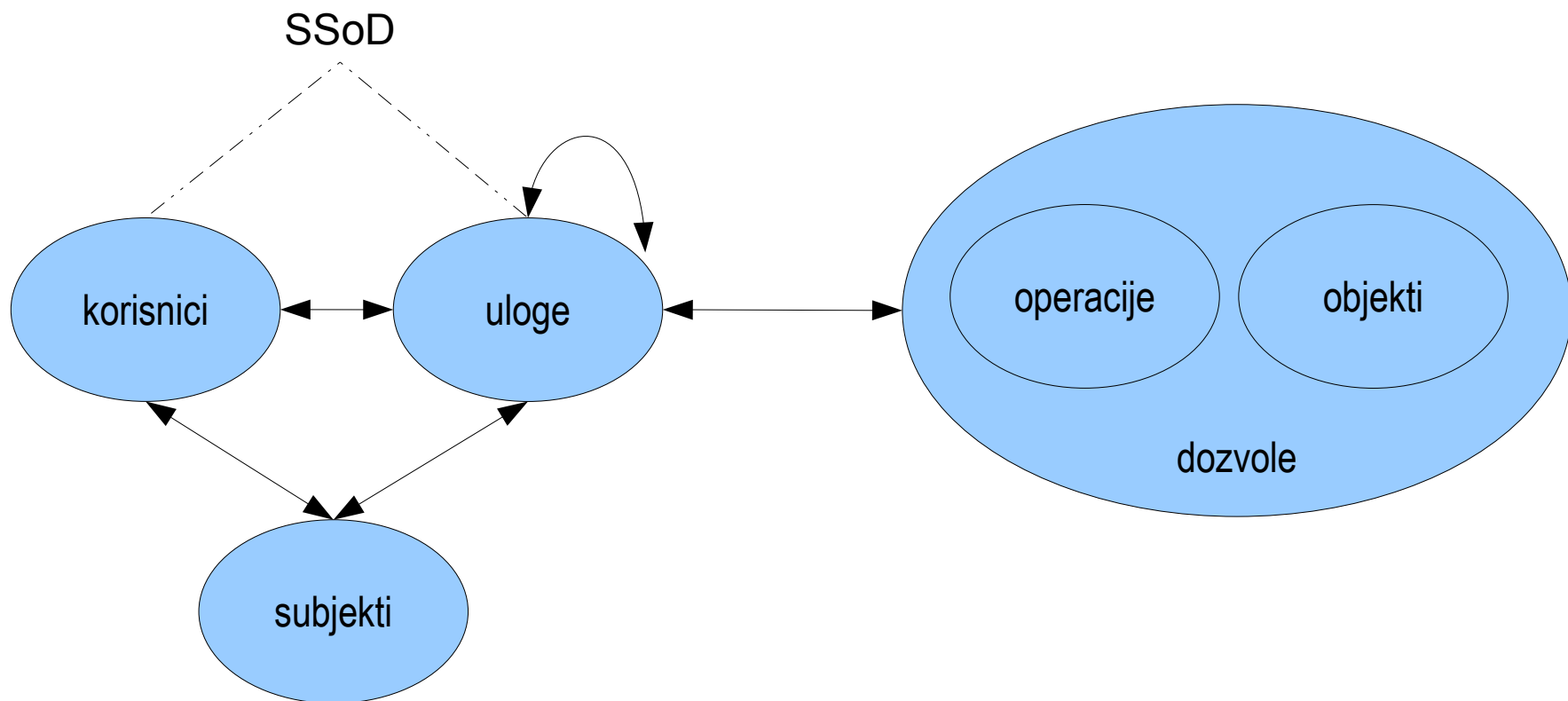
# RBAC sa ograničenjima

---

- statički SoD bez hijerarhije uloga
  - dodeljivanje jedne uloge može sprečiti dodeljivanje druge, u skladu sa SoD pravilima
- statički SoD sa hijerarhijom uloga
  - dodeljivanje jedne uloge može sprečiti dodeljivanje druge, i svih njenih potomaka
- statički SoD: uređeni par (*skup uloga*,  $n$ ) gde nijedan korisnik ne može imati više od  $n$  uloga iz ovog skupa (obično  $n=1$ )

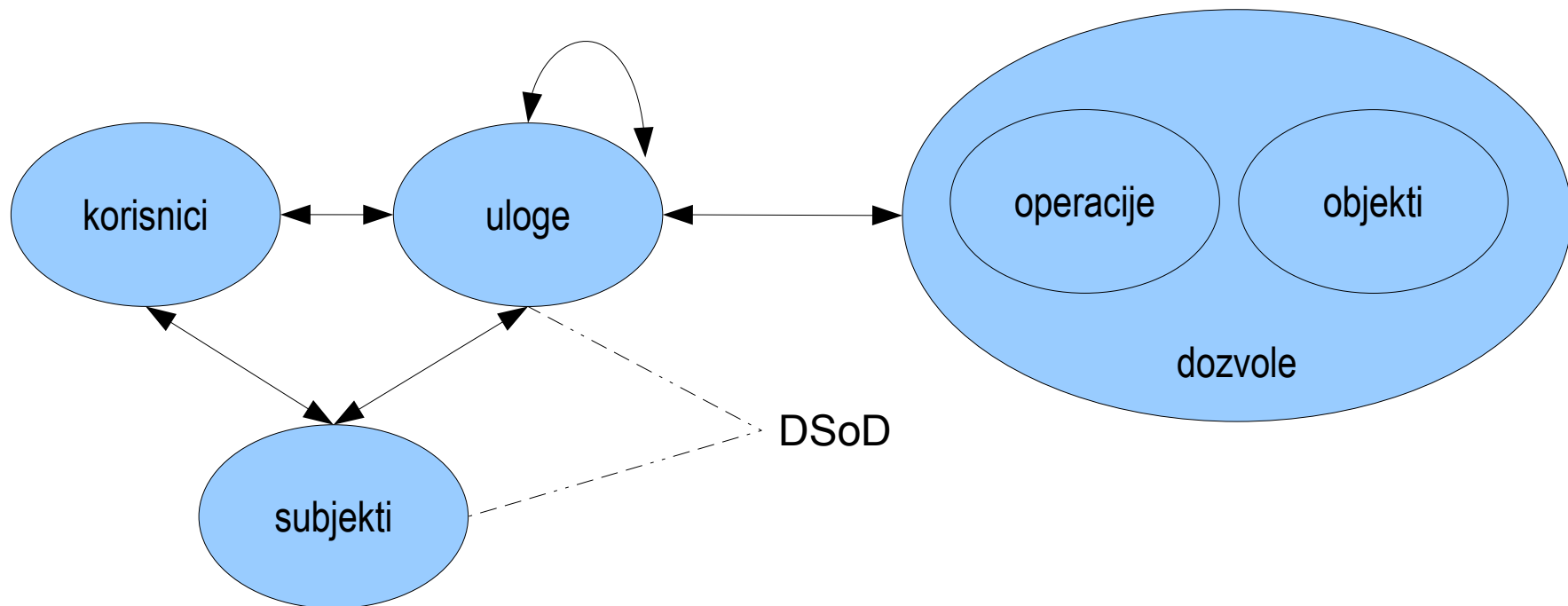
# RBAC sa ograničenjima

- statički SoD
  - ograničenja povezuju korisnike i uloge



# RBAC sa ograničenjima

- dinamički SoD
  - ograničenja povezuju subjekte i uloge
- dinamički SoD: uređeni par  $(\text{skup uloga}, n)$  gde nijedna korisnička sesija (subjekat) ne može imati više od  $n$  uloga iz ovog skupa (obično  $n=1$ )



# RBAC sa ograničenjima

---

- SoD baziran na objektima
  - primer: zahtev i odobravanje troškova
    - statički SoD: nijedna osoba ne sme imati obe uloge
    - nepraktično za male organizacije
    - rešenje: nijedna osoba ne sme imati obe uloge **za isti objekat** (isti zahtev za nabavku)
- SoD baziran na istoriji
  - korisnik može imati sve dozvole za obavljanje kritičnog zadatka (što inače ne bi smeo) ali ne može obaviti **sve delove** zadatka **nad istim objektom**



# RBAC sa ograničenjima

---

- međusobno isključivanje uloga
  - pomoću skupova uloga
    - dobijanje jedne uloge iz skupa onemogućava dobijanje drugih uloga iz istog skupa
  - pomoću parova uloga
    - nepraktično: za  $n$  uloga postoji  $n(n-1)/2$  mogućih parova

# RBAC sa ograničenjima

- dodeljivanje dozvola ulogama tako da se zadovolji SoD princip - nijedan korisnik ne može samostalno da izvrši kritičan zadatak - ne mora biti jednostavno

- primer

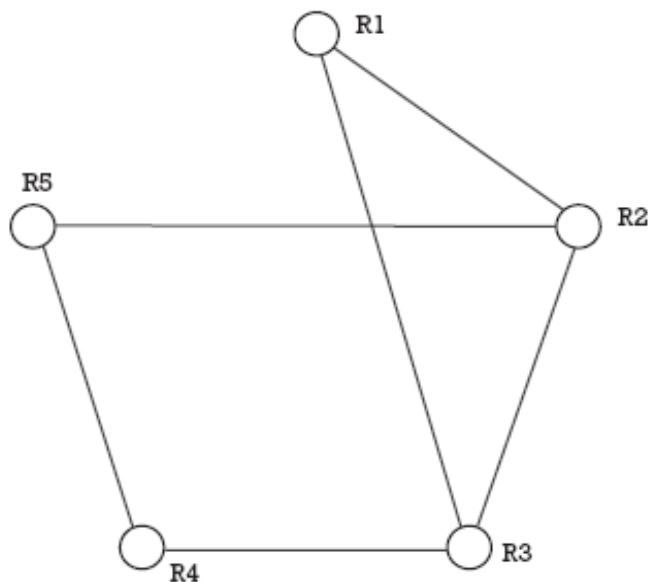
- tri kritična zadatka: T1, T2, T3
- potrebne dozvole: P1, P2, P3
- dve uloge nisu dovoljne!

|           | <i>P1</i> | <i>P2</i> | <i>P3</i> |
|-----------|-----------|-----------|-----------|
| <i>T1</i> | X         | X         | —         |
| <i>T2</i> | —         | X         | X         |
| <i>T3</i> | X         | —         | X         |

# RBAC sa ograničenjima

- dodeljivanje uloga korisnicima
  - za dati skup uloga i međusobno isključujućih parova uloga, koliko nam treba različitih korisnika?
  - primer: uloge R1-R5

|           | <i>R1</i> | <i>R2</i> | <i>R3</i> | <i>R4</i> | <i>R5</i> |
|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>R1</i> | —         | X         | X         | —         | —         |
| <i>R2</i> | X         | —         | X         | —         | X         |
| <i>R3</i> | X         | X         | —         | X         | —         |
| <i>R4</i> | —         | —         | X         | —         | X         |
| <i>R5</i> | —         | X         |           | X         | —         |



problem bojenja grafa: potrebne su tri boje, odnosno treba nam bar 3 korisnika

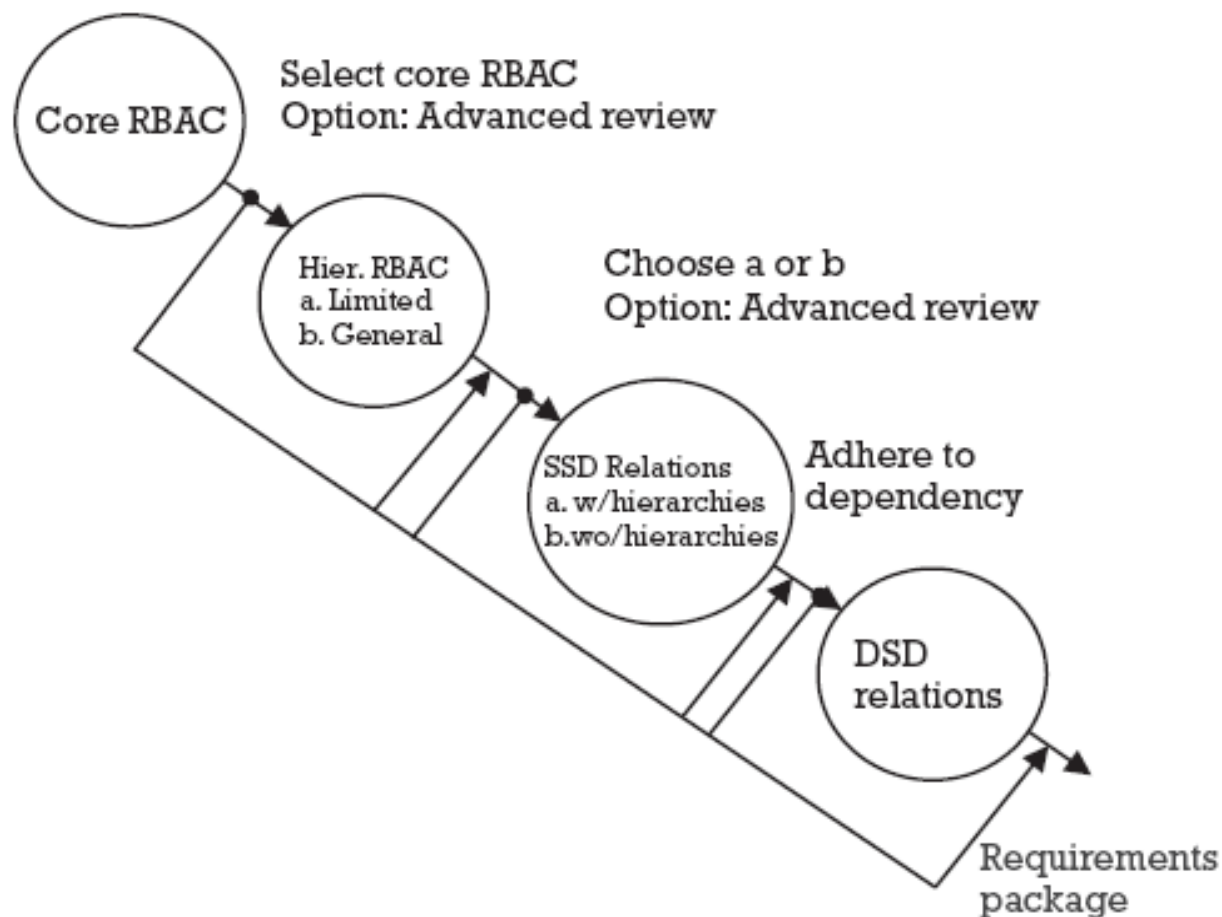
# RBAC sa ograničenjima

---

- pravila za bezbedno dodeljivanje privilegija u odnosu na SoD
    - potreban uslov: za svaki par međusobno isključivih uloga svaka uloga mora sadržati bar jednu dozvolu koju ne sadrži druga uloga
      - inače bi jedna uloga bila podskup druge, pa SoD nema smisla
    - dovoljan uslov: za svaki par međusobno isključivih uloga nijedna dozvola iz R1 ne nalazi se u R2
      - SoD je garantovan, ali je ovo često prestrog uslov
      -
- => opšte upozorenje: ako jednu dozvolu sadrži više uloga, postoji potencijalna opasnost za SoD

# RBAC standard

- NIST standard  
<http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf>
- omogućava izbor varijante RBAC-a



# Administracija RBAC-a RBAC-om

---