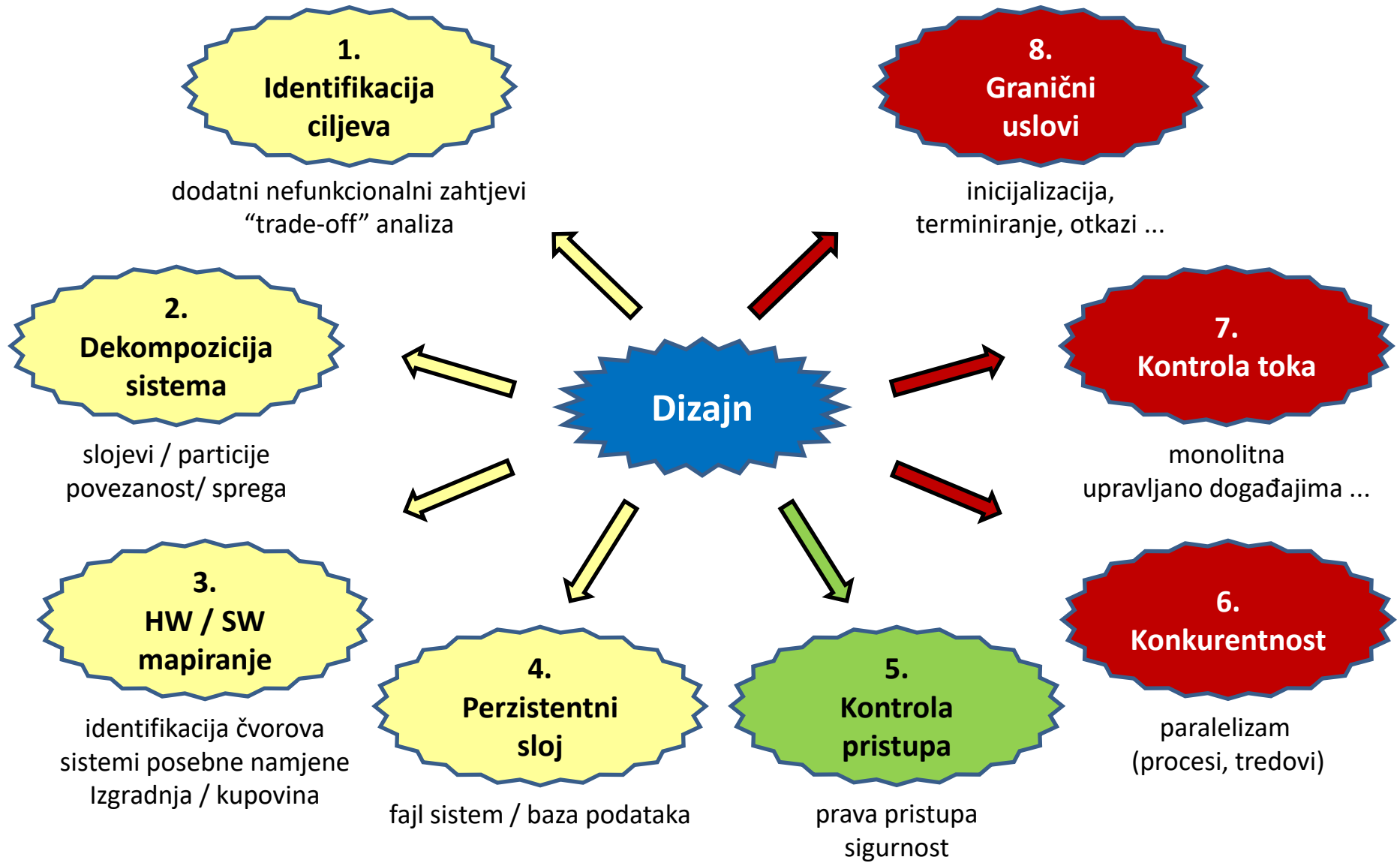
UNIVERZITET U BANJOJ LUCI
ELEKTROTEHNIČKI FAKULTET

Prof. dr Dražen Brđanin

PROJEKTOVANJE SOFTVERA
/kontrola pristupa/

Banja Luka
2024.

8 bitnih aktivnosti u projektovanju

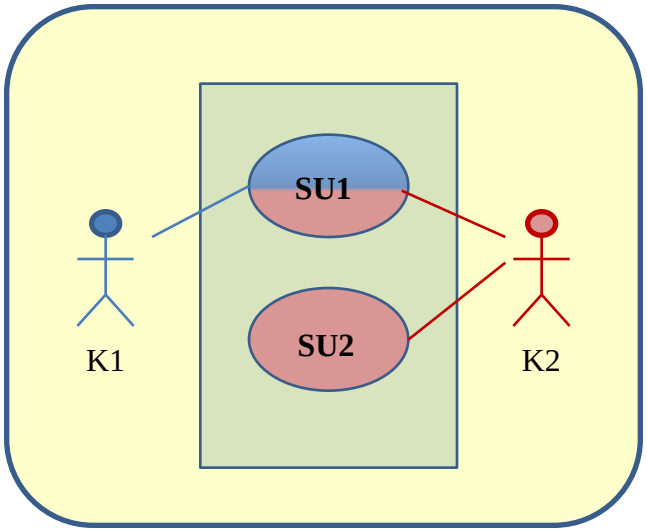


5. Kontrola pristupa

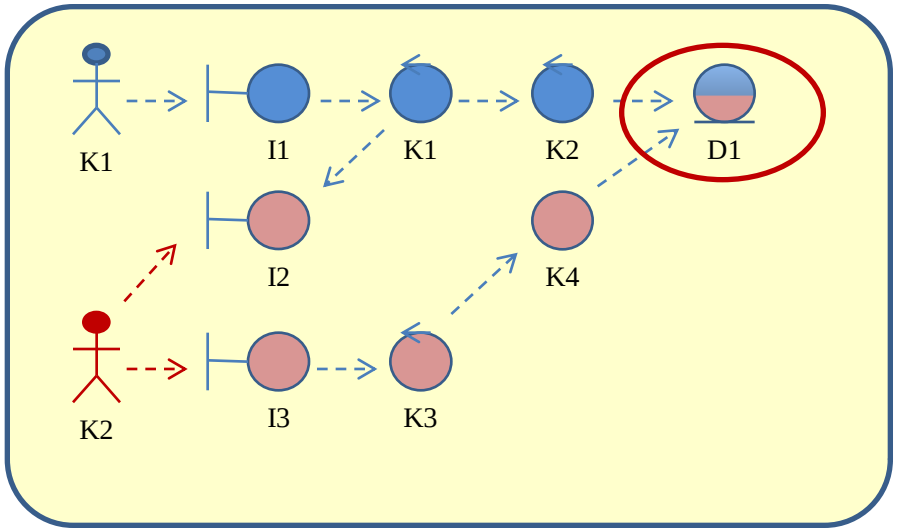
Kontrola pristupa objektima

- U višekorisničkim sistemima **različiti učesnici imaju različita prava pristupa** različitim funkcionalnostima i podacima.
  - **Koji su mehanizmi za autentikaciju korisnika?**
    - kredencijali (*username/password*), hardverski tokeni, ...
  - **Koji korisnici imaju pravo pristupa kojim klasama?**
  - **Kako se objekti štite od neautorizovanog pristupa?**
- **Kako se to modeluje?**
  - **Tokom analize**:
    - različiti učesnici vežu se sa pripadajućim slučajevima upotrebe
  - **Tokom projektovanja sistema**:
    - identifikacija objekata sa dijeljenim pristupom (višekorisnički pristup)
    - zavisno od sigurnosnih zahtjeva, definišu se mehanizmi za autentikaciju učesnika i enkripciju podataka

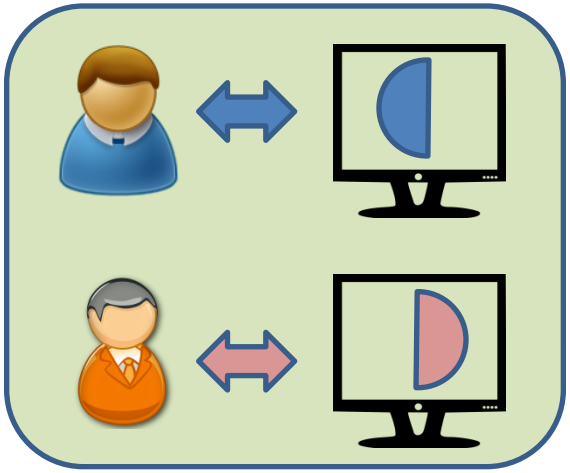
5. Kontrola pristupa



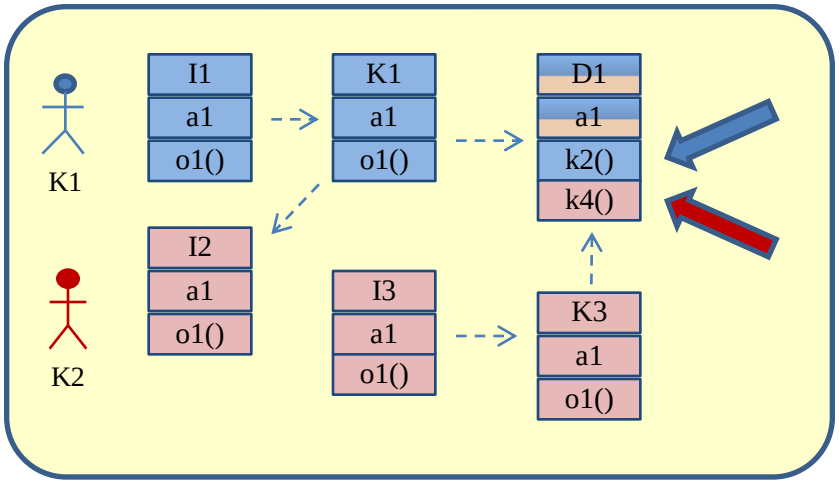
analiza



projektovanje



implementacija



5. Kontrola pristupa

Globalna matrica pristupa (*Global Access Matrix*) [Lampson, 1971]

- Služi za modelovanje statičke kontrole prava pristupa učesnika pojedinim klasama
- **Tipično: red = učesnik, kolona = klasa** (za koju definišemo prava pristupa)
- **Pravo pristupa**: lista operacija (u ćeliji matrice) koje učesnik može da izvrši nad objektima date klase

učesnici	**klase**			
	Klasa 1	**Klasa 2**	**...**	**Klasa *m***
Učesnik 1	operacija1() operacija2() ...	operacija3()		×
...				
Učesnik *n*	operacija3() operacija5()	operacija1()		operacija1()

- **Mehanizmi za reprezentaciju (implementaciju) matrice pristupa:**
  - **globalna tabela prava pristupa** (*global access table – GAT*)
  - **lista prava pristupa** (*access control list – ACL*)
  - **lista mogućnosti korisnika** (*user capability list – UCL*)

5. Kontrola pristupa

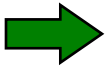
Mehanizmi za reprezentaciju (implementaciju) matrice pristupa

1. Globalna tabela prava pristupa (*Global Access Table – GAT*)

– Svaka ćelija matrice pristupa reprezentuje se trojkom

`< učesnik,klasa,operacija >`

učesnici	klase	
	Klasa 1	Klasa 2
Učesnik 1	operacija1() operacija3()	operacija2()
Učesnik 2	operacija2() operacija3()	operacija1()



učesnik	klasa	operacija
Učesnik 1	Klasa 1	operacija1()
Učesnik 1	Klasa 1	operacija3()
Učesnik 1	Klasa 2	operacija2()
Učesnik 2	Klasa 1	operacija2()
Učesnik 2	Klasa 1	operacija3()
Učesnik 2	Klasa 2	operacija1()

– **Provjera da li učesnik ima pravo pristupa nekoj operaciji vrši se pretragom trojki u globalnoj tabeli – ako nema odgovarajuće trojke, učesnik nema pravo pristupa**

– Implementacija globalne tabele (tipično) zahtijeva mnogo prostora (zavisi od broja učesnika, klasa i operacija u klasama)

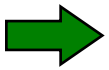
5. Kontrola pristupa

Mehanizmi za reprezentaciju (implementaciju) matrice pristupa

2. Lista prava pristupa (*Access Control List – ACL*)

- **Svaku klasu karakteriše lista parova** `< učesnik,operacija >`

učesnici	klase	
	Klasa 1	**Klasa 2**
Učesnik 1	operacija1() operacija3()	operacija2()
Učesnik 2	operacija2() operacija3()	operacija1()



Klasa 1	
Učesnik 1	operacija1()
Učesnik 1	operacija3()
Učesnik 2	operacija2()
Učesnik 2	operacija3()

Klasa 2	
Učesnik 1	operacija2()
Učesnik 2	operacija1()

- **Svaki put kad se pristupa nekom objektu, provjerava se da li pripadajuća lista prava pristupa sadrži odgovarajući par** `<učesnik,operacija>` – ako ne sadrži, učesnik nema pravo pristupa (kao što npr. recepcioner provjerava da li se na spisku rezervacija nalazi ime nekog gosta – ako se nalazi, gost će biti smješten)
- Kontrolne liste omogućavaju brzo dobijanje odgovora na pitanje
  "Ko ima pravo pristupa?"
- **Tipične primjene:**
  - operativni sistemi (kontrola pristupa datotekama, *active directory*)
  - DBMS (kontrola pristupa tabelama)

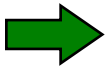
5. Kontrola pristupa

Mehanizmi za reprezentaciju (implementaciju) matrice pristupa

3. Lista mogućnosti korisnika (*user capability list – UCL*)

– **Svakog učesnika karakteriše lista parova** `< klasa,operacija >`

učesnici	klase	
	Klasa 1	**Klasa 2**
Učesnik 1	operacija1() operacija3()	operacija2()
Učesnik 2	operacija2() operacija3()	operacija1()



Učesnik 1	
Klasa 1	operacija1()
Klasa 1	operacija3()
Klasa 2	operacija2()

Učesnik 2	
Klasa 1	operacija2()
Klasa 1	operacija3()
Klasa 2	operacija1()

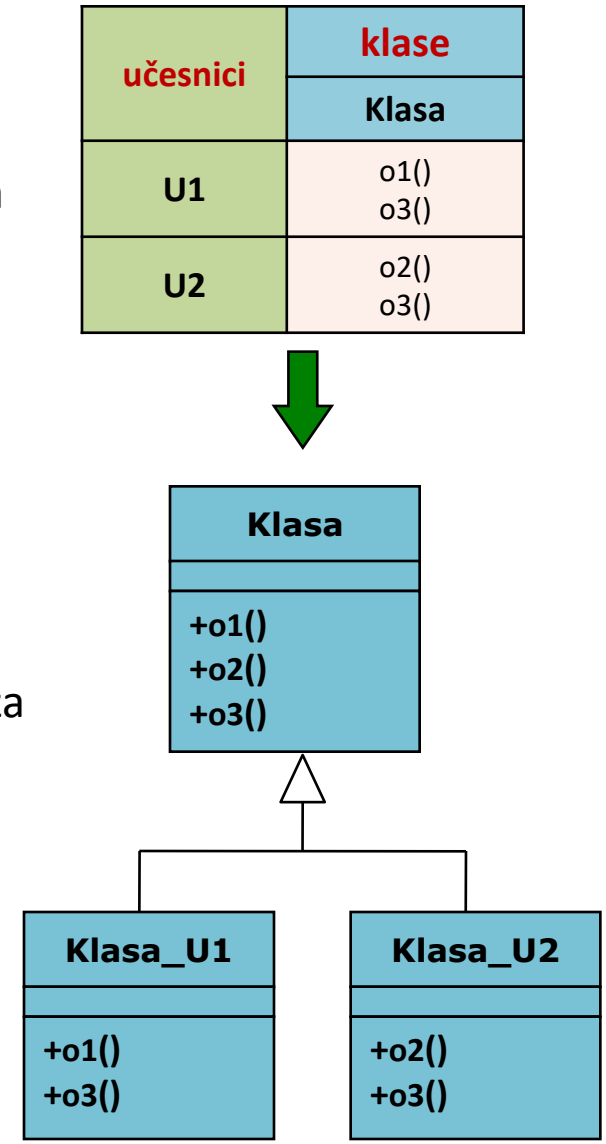
– **Učesnik može da izvrši operaciju nad nekim objektom ako lista njegovih mogućnosti sadrži odgovarajući par** `<klasa,operacija>` – ako ne sadrži, učesnik nema pravo pristupa (kao što npr. **obezbjeđenje na stadionu omogućava ulaz na stadion posjetiocima koji imaju ulaznicu**)

– Liste mogućnosti omogućavaju brzo dobijanje odgovora na pitanje
“Kojim objektima učesnik ima pravo pristupa?”

5. Kontrola pristupa

Strukturna (statička) implementacija prava pristupa

- Svaki red u globalnoj matrici pristupa predstavlja pogled na sistem iz perspektive jednog učesnika – **koliko učesnika toliko pogleda**. Svi pogledi moraju da budu konzistentni.
- **Pogledi se često implementiraju specijalizacijom klasa** za svaki različit tip para `<učesnik,operacija>`
- **Prednosti**:
  - manja vjerovatnoća za neautorizovani pristup
- **Nedostaci**:
  - nefleksibilnost, potrebne izmjene modela i aplikacije za svaki novi tip para `<učesnik,operacija>`



5. Kontrola pristupa

Dinamička kontrola prava pristupa

– **Često učesnici istog tipa nemaju ista prava pristupa objektima iste klase!**

Npr. u IS banke, brokeri (lični bankari) imaju pristup većem broju klijentskih računa.

Pravo na transakcije (uplate, isplate) na nekom klijentskom računu ima samo jedan broker (tačno određeni broker), dok drugi brokeri nemaju pravo transakcija na tom računu.

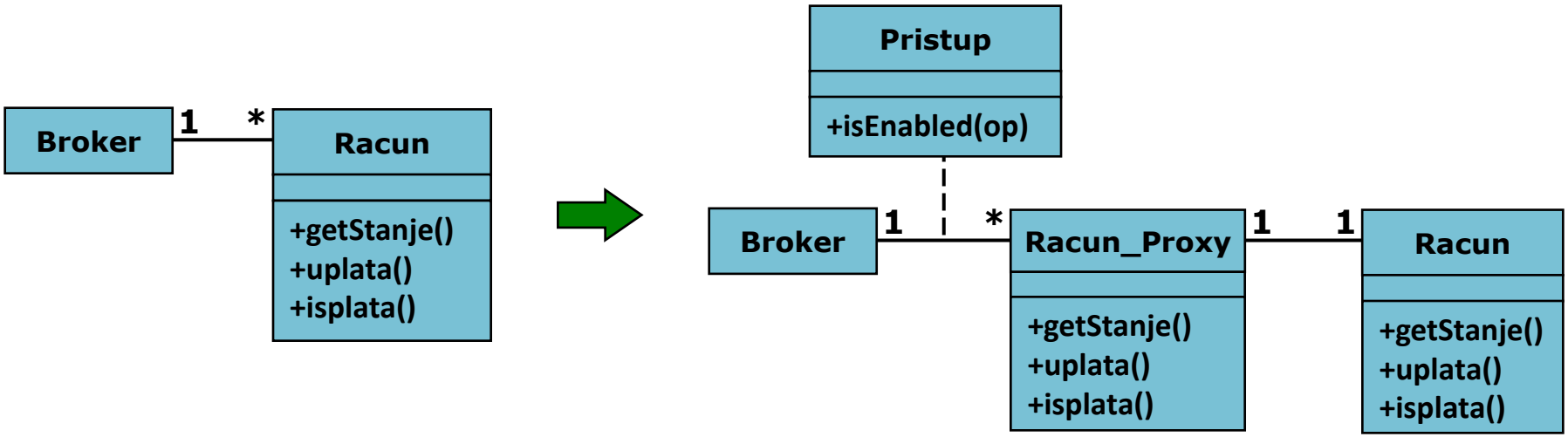
U ovom slučaju treba **dinamička kontrola prava pristupa**.

– Moguće rješenje: **proxy šablon**

Za svaki račun kreira se odgovarajući proksi koji ima ulogu kontrole pristupa.

Asocijacija Pristup između brokera i proksija pokazuje kojem računu broker ima pravo pristupa.

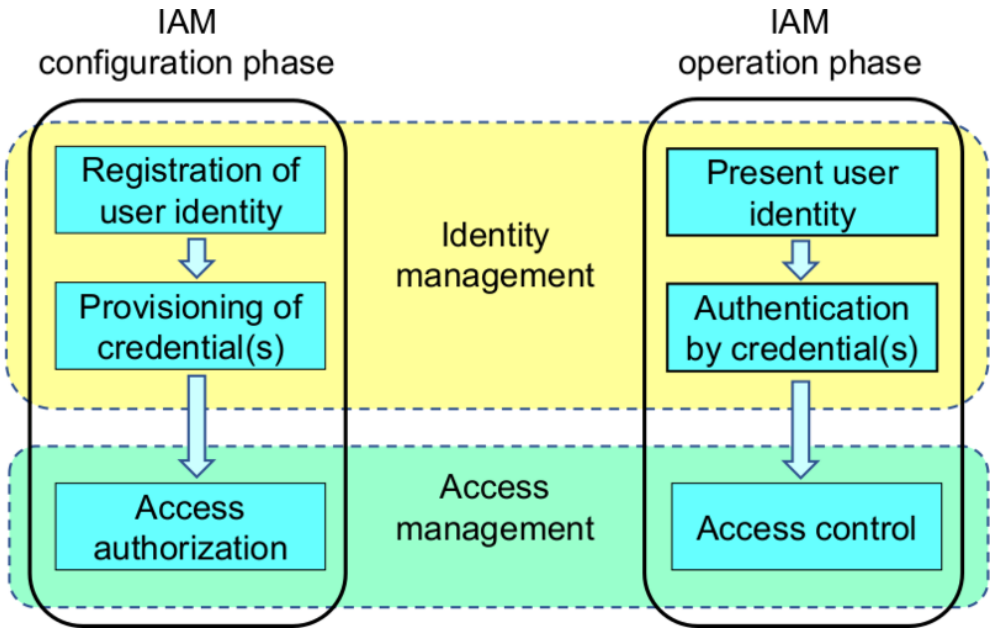
Da bi pristupio računu, broker prvo šalje poruku odnosnom proksiju. Potom proksi provjerava da li dati broker ima odgovarajuću asocijaciju sa proksijem (pristup sadrži listu dozvoljenih operacija), odnosno da li ima pravo izvršavanja tražene operacije. Ako ima broker to pravo, proksi prosljeđuje poruku računu.



5. Kontrola pristupa

Napredna rješenja za kontrolu pristupa

- **Identity Management (IdM) / Identity and Access Management (IAM or IdAM)**
  - is a framework of policies and technologies to ensure that the right users have the appropriate access to technology resources



- **Configuration phase**
  - first registering and authorizing access rights (assisted / self-service)
- **Operation phase**
  - identifying, authenticating and controlling individuals or groups of people to have access to applications, systems or networks based on previously authorized access rights

Typical IAM System capabilities

- **Authentication & Authorization & Roles & Delegation**

5. Kontrola pristupa

Napredna rješenja za kontrolu pristupa

- **Identity Management (IdM) / Identity and Access Management (IAM or IdAM)**

Standardization

- ISO/IEC 24760-1 A framework for identity management – Part 1: Terminology and concepts
- ISO/IEC 24760-2 A Framework for Identity Management – Part 2: Reference architecture and requirements
- ISO/IEC 24760-3 A Framework for Identity Management – Part 3: Practice
- ISO/IEC 29115 Entity Authentication Assurance
- ISO/IEC 29146 A framework for access management
- ISO/IEC CD 29003 Identity Proofing and Verification
- ISO/IEC 29100 Privacy framework
- ISO/IEC 29101 Privacy Architecture
- ISO/IEC 29134 Privacy Impact Assessment Methodology

Tools

- AuthO, SpectralOPS
- AWS Identity & Access Management
- Microsoft Azure Active Directory
- Google Cloud IAM
- IBM IAM
- Oracle Identity Management
- ...
- Apache Syncope (open-source)

5. Kontrola pristupa

Napredna rješenja za kontrolu pristupa

Apache Syncope

–Open-source IAM system

–Admin UI / End-user UI

–Third-party applications

- Eclipse IDE plug-in
- Netbeans IDE plug-in

–Višeslojna arhitektura

–STORAGE (različiti DBMSs)

–API