



UNIVERZITET U NOVOM SADU  
FAKULTET TEHNIČKIH NAUKA U  
NOVOM SADU



---

Aleksandar Maričić

# **INTEROPERABILNI ADAPTER ZA RAD SA *LDAP* SISTEMIMA**

MASTER RAD

- Master akademske studije -


Novi Sad, 2019.

## KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj, <b>RBR:</b>			
Identifikacioni broj, <b>IBR:</b>			
Tip dokumentacije, <b>TD:</b>	Monografska publikacija		
Tip zapisa, <b>TZ:</b>	Tekstualni štampani dokument/CD		
Vrsta rada, <b>VR:</b>	Master rad		
Autor, <b>AU:</b>	Aleksandar Maričić		
Mentor, <b>MN:</b>	Dr Nemanja Popović, docent		
Naslov rada, <b>NR:</b>	Interoperabilni adapter za rad <i>LDAP</i> sistemima		
Jezik publikacije, <b>JP:</b>	Srpski (latinica)		
Jezik izvoda, <b>Ji:</b>	Srpski		
Zemlja publikovanja, <b>ZP:</b>	Srbija		
Uže geografsko područje, <b>UGP:</b>	Vojvodina		
Godina, <b>GO:</b>	2019		
Izdavač, <b>IZ:</b>	Autorski reprint		
Mesto i Adresa, <b>MA:</b>	Fakultet Tehničkih Nauka (FTN), D. Obradovića 6, 21000 Novi		
Fizički Opis rada, <b>FO:</b> (poglavlja/strana/citata/tabela/Slika/grafika/priloga)	7/45/17/7/30/0/0		
Naučna oblast, <b>NO:</b>	Elektrotehničko i računarsko inženjerstvo		
Naučna disciplina, <b>ND:</b>	Primenjeno softversko inženjerstvo		
Predmetna odrednica/Ključne Reči, <b>PO:</b>	Informaciona-bezbednost, interoperabilnost, adapter, <i>LDAP</i>		
<b>UDK</b>			
Čuva se, <b>ČU:</b>	Biblioteka FTN, D. Obradovića 6, 21000 Novi Sad		
Važna napomena, <b>VN:</b>			
Izvod, <b>IZ:</b>	U ovom master radu istražen je način korišćenja dva različita sistema informacione-bezbednosti – <i>Microsoft Active Directory</i> zasnovan na <i>Windows</i> platformi i <i>OpenLDAP</i> zasnovanog na <i>Linux</i> platformi, uočene su njihove sličnosti i razlike, razvijen je, implementiran i verifikovan interoperabilni adapter za rad sa ova dva sistema informacione-bezbednosti koji omogućuje korišćenje administratorske i korisničke funkcionalnosti.		
Datum prihvatanja teme, <b>DP:</b>			
Datum odbrane, <b>DO:</b>			
Članovi komisije, <b>KO:</b>	Predsednik:		
	ČLAN:		Potpis mentora
	ČLAN, mentor:	Dr Nemanja Popović, docent	

## KEY WORDS DOCUMENTATION

Accession number, <b>ANO</b> :			
Identification number, <b>INO</b> :			
Document type, <b>DT</b> :	Monographic publication		
Type of record, <b>TR</b> :	Textual material, printed/CD		
Contents code, <b>CC</b> :	Master's thesis		
Author, <b>AU</b> :	Aleksandar Maričić		
Mentor, <b>MN</b> :	Dr Nemanja Popović, Assistant Professor		
Title, <b>TI</b> :	Interoperable adapter for managing LDAP systems		
Language of text, <b>LT</b> :	Serbian (latin script)		
Language of abstract, <b>LA</b> :	Serbian		
Country of publication, <b>CP</b> :	Serbia		
Locality of publication, <b>LP</b> :	Vojvodina		
Publication year, <b>PY</b> :	2019		
Publisher, <b>PB</b> :	Author reprint		
Publication place, <b>PP</b> :	Faculty of Technical Sciences, D. Obradovića 6, 21000 Novi Sad		
Physical description, <b>PD</b> : (chapters/pages/ref./tables/pictures/graphs/appendixes)	7/45/17/7/30/0/0		
Scientific field, <b>SF</b> :	Electrical and computer engineering		
Scientific discipline, <b>SD</b> :	Power softwer engineering		
Subject/Key words, <b>S/KW</b> :	Information Security, Interoperability, Adapter, LDAP		
<b>UC</b>			
Holding data, <b>HD</b> :	Library of the Faculty of Technical Sciences, D. Obradovića 6, 21000 Novi Sad		
Note, <b>N</b> :			
Abstract, <b>AB</b> :	In this Master's thesis the way of using two different information security system is researched – Microsoft Active Directory based on Windows platform and OpenLDAP based on Linux platform, their similarities and differences are identified, the interoperable adapter for those two directory services was developed, implemented and verified with supporting administration and user-related functionalities.		
Accepted by the Scientific Board on, <b>ASB</b> :			
Defended on, <b>DE</b> :			
Defended Board, <b>DB</b> :	President:		
	Member:		Menthor's sign
	Member, Mentor:	Dr Nemanja Popović, Assistant Professor	

	UNIVERZITET U NOVOM SADU • FAKULTET TEHNIČKIH NAUKA 21000 NOVI SAD, Trg Dositeja Obradovića 6	Broj:
	<b>ZADATAK ZA MASTER RAD</b>	Datum:

(Podatke unosi predmetni nastavnik - mentor)

STUDIJSKI PROGRAM:	PRIMENJENO SOFTVERSKO INŽENJERSTVO
UKOVODILAC STUDIJSKOG PROGRAMA:	Prof. dr Dragan Popović

Student:	Aleksandar Maričić	Broj indeksa:	E5 6/2018
Oblast:	Elektrotehničko i računarsko inženjerstvo		
Mentor:	Dr Nemanja Popović, docent		
NA OSNOVU PODNETE PRIJAVE, PRILOŽENE DOKUMENTACIJE I ODREDBI STATUTA FAKULTETA IZDAJE SE ZADATAK ZA DIPLOMSKI – MASTER RAD, SA SLEDEĆIM ELEMENTIMA: <ul style="list-style-type: none"> <li>- problem – tema rada;</li> <li>- način rešavanja problema i način praktične provere rezultata rada, ako je takva provera neophodna;</li> </ul>			

#### NASLOV DIPLOMSKOG- MASTER RADA:

**Interoperabilni adapter za rad sa *LDAP* sistemima**

#### TEKST ZADATKA:

Upoznati se sa sistemima *Microsoft Active Directory* i *OpenLDAP*, njihovim postavljanjem, bibliotekama za pristup i rad na zadatim sistemima, njihove sličnosti i razlike. Dizajnirati i implementirati interoperabilni adapter koji omogućuje rads sa dve zadate bezbednosne platforme kroz generički sloj, dok se odabir platforme konfiguriše na jednom mestu bez neophodnosti izmene svih specifičnih linija koda za jednu od platformi. Definirati i postaviti testno okruženje koje obuhvata *Micorsoft Active Directory* i *OpenLDAP*, razviti testnu aplikaciju i verifikovati adapter.

Rukovodilac studijskog programa:	Mentor rada:

Primerak za: o - Studenta; o - Dosije studenta; o - Mentora;

## Spisak korišćenih skraćenica

<i>Skraćenica</i>	<i>Značenje</i>
IT	<i>Information Technology</i>
AD	<i>Active Direcotry</i>
OS	<i>Operating System</i>
LDAP	<i>Lightweight Diretory Access Protocol</i>
DN	<i>Distinguished Name</i>
CN	<i>Common Name</i>
OID	<i>Object Identifier</i>
RDN	<i>Relative Distinguished Name</i>
DAP	<i>Directory Access Protocol</i>
GUID	<i>Global Unique Identifier</i>
DC	<i>Domain Controller</i>
FSMO	<i>Flexible Single Master Operator</i>
SLAPD	<i>Stand - alone LDAP Daemon</i>
SLURPD	<i>Stand - alone LDAP Update Replication Daemon</i>
LDIF	<i>LDAP Data Interchange Format</i>
PVM	<i>Process Virtual Machine</i>
IL	<i>Intermediate Language</i>

## Zahvalnica

*Najpre bih se zahvalio svojoj porodici – ocu Nenadu, majci Julkici, sestri Milani i devojci Katarini – na razumevanju i podršci tokom mog školovanja. Osim početne pomoći oko domaćih zadataka i savladavanja gradiva, roditelji su me naučili kako da učim, pristupam svojim obavezama, budem odgovoran i ne obeshrabrim se kada naiđem na prepreku. Na svoju porodicu uvek sam mogao da se oslonim u momentima nesigurnosti, kao i kada mi je bio potreban iskren savet.*

*Mentoru Nemanji Popoviću zahvaljujem se na posvećenosti koju mi je pružio pri izradi ovog rada, omogućivši mi da steknem nova interesovanja i saznanja koja nisu bila ograničena samo na temu master rada.*

*Za savetovanje i usmeravanje u toku studija posebno se zahvaljujem bratu Miti Čokiću, na koga nastojim da se ugledam kada je reč o marljivosti i akademskom zalaganju.*

*Zahvaljujem se i asistentu i tim lideru Bojanu Jelačiću za mnoštvo korisnih saveta i iskazano razumevanje.*

## Sadržaj

1	Uvod .....	1
2	Teorijske osnove.....	3
2.1	Kontrola pristupa.....	3
2.2	<i>Lightweight Directory Access Protocol</i> .....	5
2.3	<i>Microsoft Active Directory</i> .....	8
2.3.1	Domen, kontrolor domena, domensko stablo i šuma.....	8
2.3.2	Organizaciona jedinica, globalni katalog.....	10
2.3.3	Master kontrolori domena.....	10
2.3.4	Poverenje između domena .....	11
2.4	<i>OpenLDAP</i> .....	12
2.5	Interoperabilnost.....	14
2.6	Procesna virtuelna mašina, <i>Microsoft .Net Core</i> .....	14
3	Predloženo rešenje interoperabilnog adaptera .....	17
3.1	Arhitektura rešenja .....	17
3.2	Generički sloj .....	19
3.3	Interoperabilni adapter .....	20
3.3.1	Funkcionalnosti.....	20
3.3.2	Dizajn.....	22
3.4	<i>Active Directory</i> menadžer funkcija.....	24
3.4.1	Kreiranje novog korisnika.....	24
3.4.2	Brisanje korisnika .....	25
3.4.3	Izmena korisničkih podataka .....	25
3.4.4	Kreiranje grupe .....	25
3.4.5	Brisanje grupe .....	25
3.4.6	Izmena grupnih podataka.....	26
3.4.7	Dodavanje korisnika ili grupe u grupu.....	26
3.4.8	Dodeljivanje ugrađenih dozvola pristupa .....	26
3.4.9	Kreiranje, dodavanje i dodeljivanje novih dozvola pristupa .....	26
3.4.10	Logovanje korisnika i auditovanje.....	27
3.4.11	Proveravanje dozvole pristupa.....	27
3.5	<i>OpenLDAP</i> menadžer funkcija.....	28
3.5.1	Kreiranje korisnika.....	28

3.5.2	Kreiranje grupe .....	29
3.5.3	Promena korisničkog ili grupnog imena .....	29
3.5.4	Brisanje čvora .....	29
3.5.5	Izmena korisničkih podataka .....	29
3.5.6	Dodavanje korisnika/grupe u grupu .....	30
3.5.7	Proveravanje članstva .....	30
3.5.8	Dodeljivanje ugrađenih dozvola pristupa .....	30
3.5.9	Dodavanje i dodeljivanje novih dozvola pristupa.....	30
3.5.10	Logovanje korisnika i auditovanje .....	30
3.5.11	Proveravanje dozvole pristupa .....	30
4	Verifikacija rešenja.....	31
4.1	Testno okruženje .....	31
4.2	Testiranje.....	32
5	Zaključak .....	41
6	Literatura .....	43
7	Biografija.....	45



# 1 Uvod

Svakodnevnim razvojem informacionih tehnologija (*eng. Information Technologies – IT*) povećava se broj korisnika *IT* sistema, što dovodi do lakšeg otkrivanja slabosti i ugrožavanja bezbednosti. Bezbednost predstavlja zaštitu od različitih opasnosti po sistem, kao što su: fizička, personalna, komunikaciona, informaciona-bezbednost i dr [1]. U okviru izrade ovog master rada fokus je na istraživanju informaciono-tehnoloških aspekata informacione-bezbednosti.

Informaciona-bezbednost obezbeđuje zaštitu podataka značajnih za kompaniju i predstavlja jednu od ključnih komponenti za uspešan razvoj i poslovanje kompanija. Osnovni cilj informacione bezbednosti jeste obezbeđivanje zaštite poverljivosti, integriteta podataka i dostupnosti podataka [1]. Informaciono-bezbednosni sistem upravlja podacima o korisnicima, njihovim ličnim podacima, grupama kojima korisnici pripadaju, dozvolama pristupa i drugim specifičnim podacima. Podaci o entitetima organizuju se hijerarhijski radi lakše pretrage, na osnovu potreba kompanija.

U savremenim informaciono-tehnološkim rešenjima zahteva se sve veća standardizacija rešenja koja omogućuje fleksibilnost pri izboru *IT* komponenti. Svakom funkcionalnom celinom upravlja jedna ili više komponenti, a za kontinuirano funkcionisanje celina, potrebna je njihova međusobna integracija i usklađenost, koja se zove interoperabilnost. Izbor komponenti je ekonomski vođen uz mogućnost što jednostavnije promene *IT* komponente tokom eksploatacionog perioda.

Postoje različite informaciono-tehnološke platforme i nad njima zasnovana rešenja informacione-bezbednosti. U ovom istraživanju fokus je na dva različita informaciono-bezbednosna sistema – *Microsoft Active Directory* (*eng. ActiveDirectory - AD*), zasnovan na *Windows* operativnom sistemu (*eng. Operating System - OS*) i *OpenLDAP*, zasnovan na *Linux OS*-u. Ovi bezbednosni sistemi se oslanjaju na *Lightweight Directory Access Protocol* (*eng.LDAP*), ali i pored toga imaju brojne razlike koje se ogledaju u različitim tipovima entiteta, atributa, načinima povezivanja entiteta i u interfejsima za konekciju i pristup njihovim bazama podataka. *LDAP* je industrijski protokol za pristup i izmenu podataka u bazi informacionih podataka koja čuva podatke o entitetima i obezbeđuje njihovo deljenje preko mreže u okviru kompanija. Podaci o entitetima u bazi informacionih podataka organizovani su hijerarhijski u zavisnosti od lokacije, uloge, dozvola pristupa i drugim parametrima specifičnim za oblast poslovanja kompanija [2].

Prilikom migracije sa jedne platforme informacione bezbednosti na drugu, ove razlike zahtevaju promenu svih linija koda zaduženih za pristup i upravljanje podacima na jednoj od platformi. Što je informacioni sistem veći, to je broj specifičnih linija veći i promena je složenija. Kako bi se obezbedila jednostavna migracija sa jedne informaciono-bezbednosne platforme na drugu, potrebno je razviti konfigurabilni interoperabilni adapter koji će lokalizovati sve pozive funkcija ka sistemima informacione bezbednosti (uključujući i one specifične). Generički sloj bi korisniku obezbedio pristup bezbednosnim sistemima bez detaljnog znanja o sistemima, dok bi se izbor baze odredio postavkom konfiguracije na jednom mestu.

U ovom radu istraženi su principi funkcionisanja sistema informacione bezbednosti *AD*, zasnovanog na *Windows* platformi i *OpenLDAP*, zasnovanog na *Linux* platformi, njihove postavke i instalacije, biblioteke za pristup i za rad sa navedenim platformama. Razvijene su funkcije za upravljanje informaciono-bezbednosnim podacima, proučene su strukture podataka koje se koriste na oba bezbednosna sistema, uočene njihove sličnosti i razlike i mapirani su atributi entiteta koji imaju isto značenje. Dizajniran je i implementiran konfigurabilni interoperabilni adapter koji omogućuje rad sa obe bezbednosne platforme kroz generički sloj. Izbor i promena platforme vrši se na jednom mestu kroz konfiguracionu datoteku, koja se iščitava na početku rada adaptera. Definisano je i postavljeno testno okruženje, koje obuhvata *AD* i *OpenLDAP* bezbednosni sistem i razvijena je testna aplikacija, povezana sa interoperabilnim adapterom, kojom je verifikovan rad adaptera.

U prvom poglavlju prikazan je uvod u kojem je opisana motivacija za sprovođenje istraživanja u okviru ove oblasti, zatim problem koji se pojavljuje i šta je urađeno kako bi se on rešio. Drugo poglavlje daje pregled teorijskih osnova korišćenih za izradu ovog master rada. Opisani su osnovni pojmovi informacione bezbednosti, kontrole pristupa, objašnjen je *LDAP* protokol i dva informaciono bezbednosna sistema zasnovanih na *LDAP* protokolu. U trećem poglavlju predstavljeno je predloženo rešenje interoperabilnog adaptera. Objašnjena je struktura i dizajn sistema, kako je implementiran interoperabilni adapter zasnovan na generičkom sloju i kako su razvijene funkcionalnosti za rad sa *AD* i za *OpenLDAP* bezbednosnim sistemima. Četvrto poglavlje opisuje proces verifikacije rešenja. Navedene su korišćene tehnologije i njihove verzije, predstavljena je testna aplikacija i vizuelno prikazana verifikacija rešenja. U petom poglavlju dat je zaključak, kao i smernice za dalji rad i buduća istraživanja implementiranog rešenja. Literatura je navedena u šestom poglavlju, a biografija u sedmom.

## 2 Teorijske osnove

Informaciona-bezbednost predstavlja vrlo široku i multidimenzionalnu oblast jedne organizacije koja obuhvata ljude, procese i tehnologije [1].

Informaciona-bezbednost oslanja se na sledeće pojmove [1]:

- a) Kontrola pristupa predstavlja praćenje i kontrolisanje pristupa subjekata prema objektima.
- b) Resursi jedne kompanije predstavljaju sve što je u njenom vlasništvu: fizički predmeti, softver, osobe, informacioni i računarski sistemi i dr.
- c) Napad predstavlja akciju koja nanosi štetu podacima ili vrši neovlašćene izmene podataka. Napad može da bude aktivan ili pasivan, sa namerom ili slučajan i direktan ili indirektan.
- d) Kontrola, zaštita i kontra-mere su pojmovi informacione-bezbednosti čijim razvijanjem se priprema odbrana sistema, smanjuje rizik od napada i ranjivost sistema i na druge načine se povećava bezbednost sistema.
- e) Izloženost resursa je stanje kada je napadač prisutan i sistem ranjiv.
- f) Rizik je verovatnoća da će se desiti nepoželjna situacija.
- g) Pretnja je subjekat ili entitet koje predstavlja opasnost za sistem.
- h) Ranjivost je slabost ili postojeća greška u sistemu ili zaštitni mehanizam koji je podložan napadima.
- i) Gubitak je bilo kakvo nanošenje štete ili neovlašćeno pristupanje resursima.

Osnovni cilj informacione-bezbednosti je da zaštiti karakteristike informacija koje imaju vrednost za organizaciju, a te karakteristike su [1]:

- a) Poverljivost podataka predstavlja zaštitu pristupa podacima od neovlašćenih korisnika. Poverljivost se postiže preduzimanjem određenih mera: klasifikacijom informacija, zaštitom baze podataka, kontrolisanjem dozvola pristupa i edukacijom krajnjih korisnika i poverljivosti informacija.
- b) Integritet podataka predstavlja zaštitu od neautorizovanih izmena i brisanja.
- c) Dostupnost je mogućnost ovlašćenog korisnika da neometano pristupa podacima i da su mu dostupni u svakom trenutku.

U ovom poglavlju dat je pregled koncepata kontrole pristupa, *LDAP* protokola i tehnoloških rešenja *AD* i *OpenLDAP* koji su korišćeni tokom istraživanja.

### 2.1 Kontrola pristupa

Kontrola pristupa se oslanja na pojmove subjekata, objekata, operacija i dozvola da se operacija izvrši. Subjekt je entitet koji ima aktivnu ulogu u sistemu bezbednosti, dokazuje svoj identitet, pristupa objektima i inicira prenos informacija do drugih subjekata ili objekata.

Subjekat može da bude [3]:

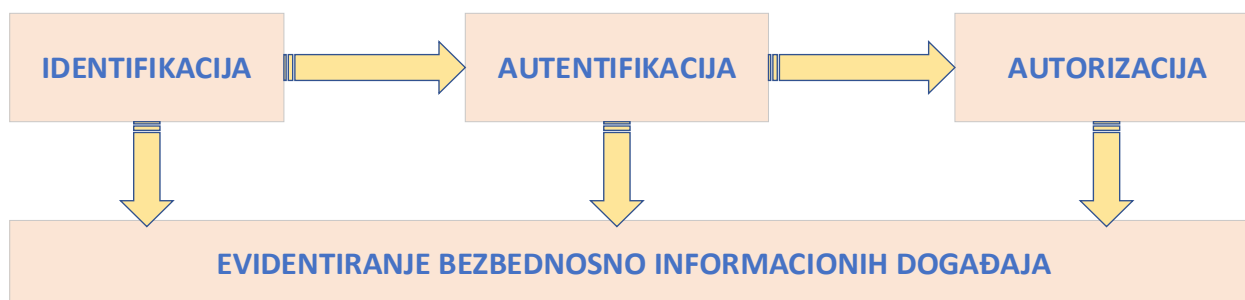
- a) autorizovani ili neovlašćeni korisnik,
- b) aplikacija,
- c) sistem,
- d) mreža.

Subjektu na nekoliko različitih načina može da se ograniči pristup objektu, a neka od tih ograničenja su: vreme pristupanja, lokacija autentifikacije subjekta, pristup van lokalne mreže, specijalno dodeljena prava pristupa i dr. Objekat je entitet koji nema aktivnu ulogu kao subjekat. On prima ili čuva podatke, a može da bude [3]:

- a) aplikacija,
- b) mreža,
- c) fizički prostor za čuvanje memorije,
- d) podatak.

Operacija predstavlja vrstu obrade informacije, dok dozvole predstavljaju pravila po kojima subjekat pristupa objektu. Svaki subjekat ima listu dozvola pristupa i ta lista je definisana u listi kontrole pristupa [3].

Kontrola pristupa definiše kojim objektima subjekat sme da pristupa, kakva su njegova prava i šta korisnik sme da radi sa tim objektima. Da bi se uspostavila potpuna kontrola pristupa, subjekat mora da prođe kroz tri faze: identifikacija, autentifikacija i autorizacija uz evidentiranje bezbednosno informacionih događaja, koje se redom mogu videti na Slici 1.



Slika 1 - Proces predstavljanja korisnika i utvrđivanja prava

Identifikacija subjekta je njegovo predstavljanje. Autentifikacija je dokazivanje identiteta korisnika. Postoje tri načina autentifikacije subjekta [3]:

- a) nešto što subjekat zna (npr. lozinka),
- b) nešto što poseduje (npr. hardverski uređaj),
- c) nešto što jeste, (npr. biometrijski otisak prsta).

Prosta autentifikacija koristi samo jedan od tri načina dokazivanja identiteta. Za veću sigurnost implementira se dvostepena autentifikacija, gde je subjekat uz ono što zna, dužan da priloži i ono što ima ili ono što jeste. Najveća sigurnost se postiže kod višestepene (eng. *multifactor*) autentifikacije, koja koristi sva tri načina [3].

Autorizacija korisnika je mehanizam kojim se određuju prava subjekta i njegovi nivoi pristupa objektima, a iz liste kontrole pristupa mogu da se proveriti dozvole subjekta za pristupanje. Autorizacija je moguća tek kada je subjekat identifikovan i autentifikovan.

Iz Tabele 1 može se videti primer uprošćene liste kontrole pristupa. U prvoj koloni su izlistani primeri subjekata, a u drugoj i trećoj koloni primeri objekata sa dozvolama pristupa subjekata prema objektu. *Korisnik 1* ima dozvolu da čita i upisuje podatke na *Disk C*, ali da nema dozvolu za čitanje ni za pisanje u *Skladište A*. *Korisnik 2* može i da čita i upisuje nove podatke u *Skladište A*, a samo da čita podatke sa diska i dr.

Subjekti	Objekti	
	<i>Skladište A</i>	<i>Disk C</i>
<i>Korisnik1</i>		<i>čitanje / pisanje</i>
<i>Korisnik2</i>	<i>čitanje / pisanje</i>	<i>čitanje</i>
<i>Proces A</i>		<i>čitanje</i>
<i>Proces B</i>	<i>pisanje</i>	

Tabela 1 – Primeri subjekata i objekata

Evidentiranjem bezbednosno-informacionih događaja se prate, zapisuju i čuvaju aktivnosti bezbednosnih sistema, čime se zna kojim objektima se pristupa, ko im pristupa, u koje vreme i koje aktivnosti primenjuje, a analizom ovih podataka mogu se uočiti eventualne greške ili neovlašćeni pristupi u sistemu [3].

U velikim sistemima sa mnogo subjekata i objekata, veliki izazov je kako će se organizovati kontrola pristupa. Zato se subjekti sa istim dozvolama pristupa grupišu. Administrator ne dodaje subjektu sve dozvole pojedinačno, već će ga učlanjuje u grupu. Pored subjekata, i objekti mogu da se grupišu. Primer grupisanja objekata je dodavanje datoteka u isti direktorijum. Administrator tada definiše prava pristupa za ceo direktorijum, a ne za datoteke pojedinačno [3].

## 2.2 *Lightweight Directory Access Protocol*

*LDAP* je industrijski protokol standard zasnovan na Internet Protokolu za pristup i izmenu distribuiranih podataka u bazi informacionih podataka koja čuva specifične i uređene informacije o subjektima i objektima [2].

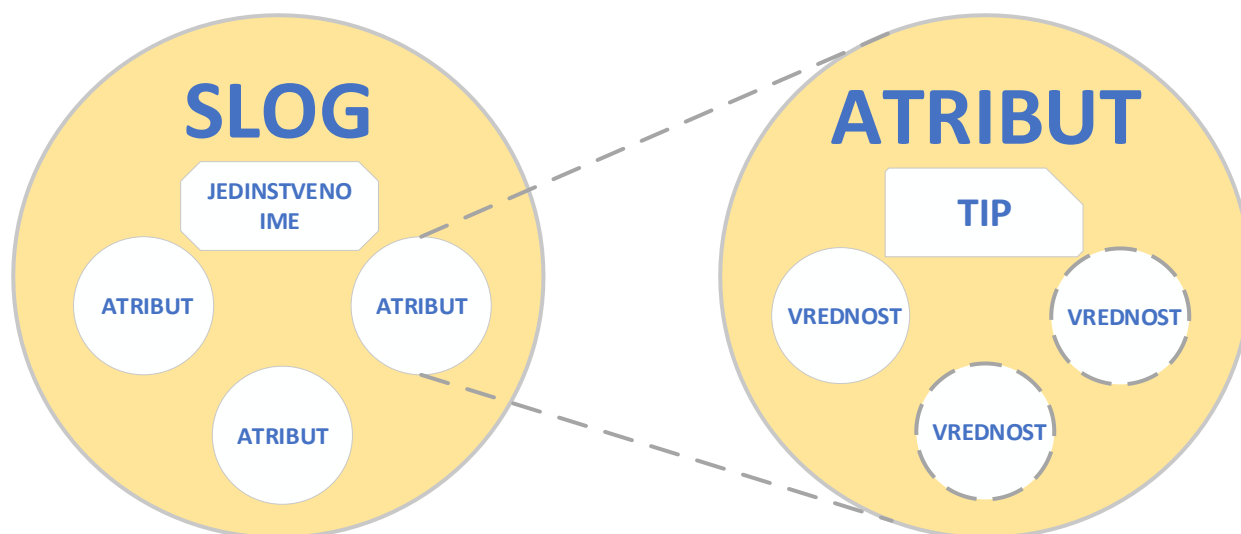
Osnovni pojmovi *LDAP* – a su [2]:

- a) slog,
- b) atribut,
- c) jedinstveno ime,
- d) klasa objekata.

**Slog** predstavlja sve podatke o jednom entitetu. Svaki slog se sastoji od tri glavne komponente: jedinstveno ime (*eng. Distinguished Name - DN*), objektnih klasa i atributa. Slogovi su organizovani u strukturu stabla. Više stabala koji su sačinjeni od slogova čine šumu [2].

**Atributi** predstavljaju osobine subjekata i objekata, imaju svoj tip i vrednost ili listu vrednosti. Slogovi se sastoje od atributa, npr. *cn=Pera Peric*. *Cn (common name)* je naziv tipa, a *Pera Peric* je vrednost atributa. *Cn* je naziv tipa, a *Pera Peric* je vrednost atributa. Atributi su definisani u šemi baze podataka jedinstveno su identifikovani objektnim identifikatorom (*eng. Object Unique Identifier – OID*). Imaju svoju sintaksu, koja definiše koji tipovi podataka mogu da se čuvaju u atributu i koja su ograničenja vrednosti. Vrednost atributa može biti i niz vrednosti [2].

Na Slici 2 prikazana je konekcija sloga i atributa u vidu dva kruga. Prvi krug je slog, koji je identifikovan jedinstvenim imenom i sadrži set atributa. Drugi krug je atribut kojeg definiše tip i sadrži vrednost ili skup vrednosti.

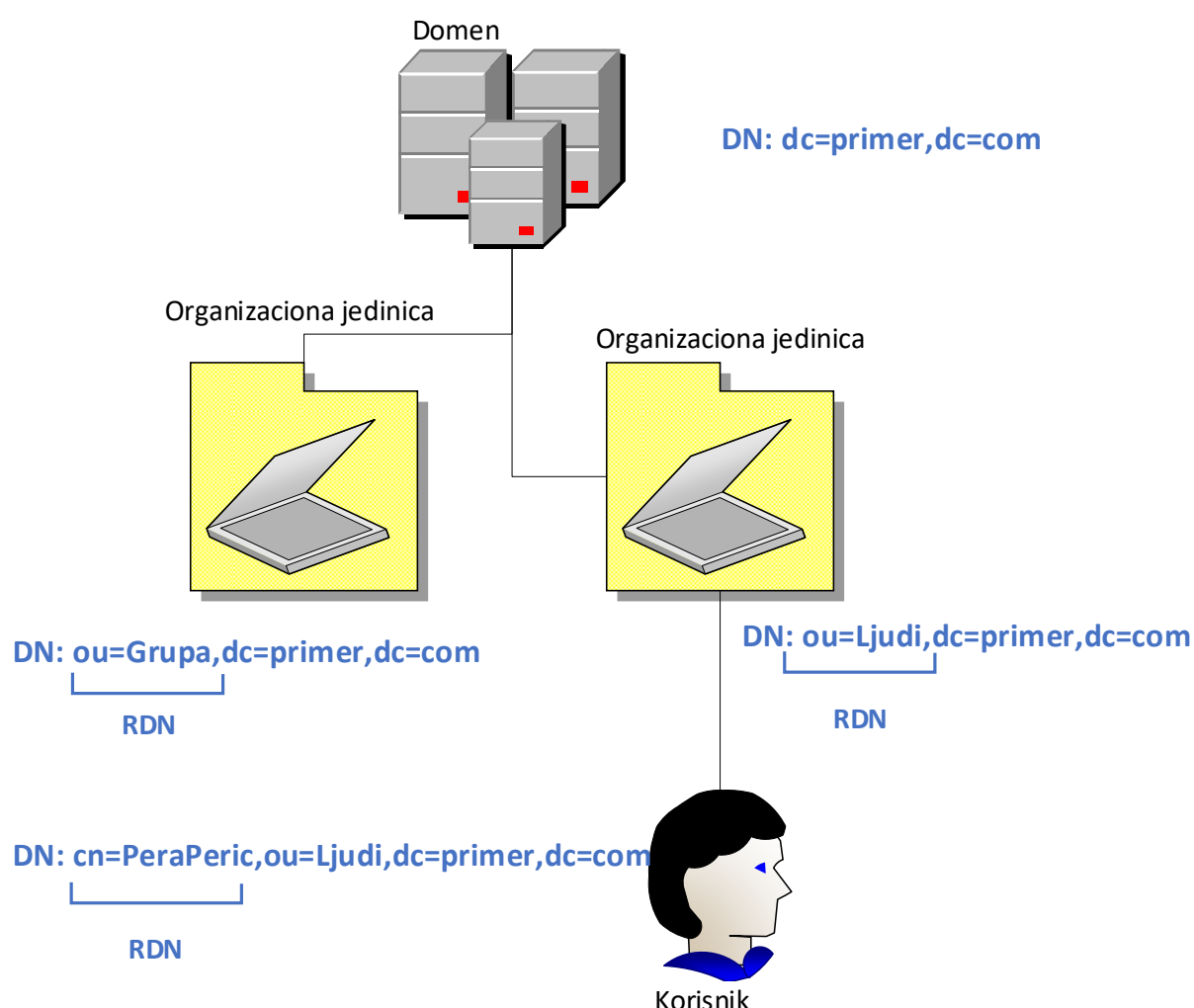


Slika 2 - Veza slogova i atributa, slika preuzeta sa [2]

**Jedinstveno ime** je atribut kojim se na jedinstven način identifikuje svaki slog. *DN* se sastoji od relativnog jedinstvenog imena (*eng. Relative Distinguished Name - RDN*) i putanje u stablu, gde se slog nalazi [2].

Na Slici 3 se vidi primer stabla sa: jednim domenom, dve organizacione jedinice i osobom, a svaki čabla čini jedan slog u bazi podataka.. Ako posmatramo najniži čvor stabla, to je osoba *Pera Peric* sa jedinstvenim imenom: *cn=PeraPeric,ou=Ljudi,dc=primer,dc=com*. Prvi deo jedinstvenog imena - *cn=PeraPeric* je relativno jedinstveno ime, *ou=Ljudi* je *RDN* organizacione jedinice u stablu roditelja posmatranog čvora. Paralelno je druga organizaciona jedinica, čiji je *RDN*: *ou=Grupa*. Koren stabla je domen, sa jedinstvenim imenom: *dc=primer,dc=com*.

**Klasa objekata** je specijalan atribut zajednički za slogove u bazi, jedinstveno identifikovan objektnim identifikatorom i definiše koji se atributi koriste u okviru jednog sloga. Primer klasa objekata bi bila klasa *Person*, koja definiše osobu. Slogovi kojima se dodeljuje ova klasa mogu da imaju attribute poput: imena, prezimena, adrese, broja telefona i dr. Klase objekata se definišu u šemi, koja opisuje i koji su atributi obavezni, koji su opcionalni, definiše sintaksu i ograničenja atributa [2].



Slika 3 - LDAP stablo

Standardizovani port za *LDAP* je 389 za nekriptovanu komunikaciju, a 636 za kriptovanu komunikaciju. Klijent inicira komunikaciju sa serverom i šalje mu zahtev koji želi da se ispuni. Server je nakon toga odgovoran da napravi odgovarajuće izmene u bazi informacionih podataka. Kada izvrši zadatak, server šalje odgovor klijentu. Iako se zahteva odgovor serverske strane, komunikacija ne mora da bude sinhrona (po mogućstvu se može kontrolisati sa klijentske strane).

Za korisnika i dizajnera *LDAP* protokola, nije od suštinskog značaja da zna kako se prenose poruke između klijenta i servera i kakva je struktura poruke, već koji logički model podataka se koristi, kakvi se podaci čuvaju na serverskoj strani, kako su podaci organizovani, kako se štite i koje operacije su moguće nad podacima. Serverska strana čuva informacije o subjektima i objektima – korisnicima, grupama, organizacijama, koji tipovi veza postoje, kako su objekti povezani i dr. *LDAP* se može implementirati na različitim operativnim sistemima, sa različitim tipovima baza podataka. *LDAP* je jednostavnija verzija protokola za pristupanje direktorijuma (*eng. Directory Access Protocol – DAP*), koji pripada X.500 standardu [2].

X.500 *Directory Service* je standard za elektronsko čuvanje podataka o ljudima unutar organizacija. Standard je nastao kako bi se olakšala pretraga subjekata po atributu koji lako može da se zapamti. To može da bude korisničko ime, adresa elektronske pošte, grupa kojoj korisnik pripada ili neki drugi atribut. Svi podaci se čuvaju na centralnom serveru organizacije, kako bi tehnički svima u organizaciji bio omogućen pristup [2].

### 2.3 Microsoft Active Directory

Predstavlja implementaciju *LDAP* protokola kompanije *Microsoft*. Prvi put je implementiran na *Windows 2000* operativnom sistemu, a danas se koristi na *Windows Server* operativnim sistemima (poslednja verzija je *Windows Server 2019*). *AD* pruža bazu informacione-bezbednosti i tehnološko rešenje kontrole pristupa koje se koriste u ovom radu. Ono omogućuje upravljanje korisnicima, grupama, organizacionim jedinicama, štampačima, aplikacijama i servisima. Podaci se nalaze na centralnom serveru i njima mogu da pristupaju subjekti kojima je dodeljena dozvola pristupa [4]. Osim upravljanja podacima, administratori mogu da organizuju podatke na osnovu biznis zahteva kompanija i da prošire funkcionalnosti *AD* – a, ali to prevazilazi obim ovog istraživanja.

Svaki objekat u *AD* ima jedinstveni globalni identifikator (*eng. Global Unique Identifier - GUID*, 128bitni broj), koji mu se dodeli prilikom kreiranja. Kada se čvor prebacuje sa jednog nivoa u stablu na drugi ili ako se prebacuje na drugo stablo, jedinstveni identifikator se ne menja. Pretraga po jedinstvenom identifikatoru nije praktična, pa se za jedinstveno identifikovanje objekata koristi *DN*. Kao što je spomenuto, *DN* označava poziciju sloga u stablu, pa kada se slog prebaci sa jedne pozicije na drugu, *DN* mora da se ažurira [4].

#### 2.3.1 Domen, kontrolor domena, domensko stablo i šuma

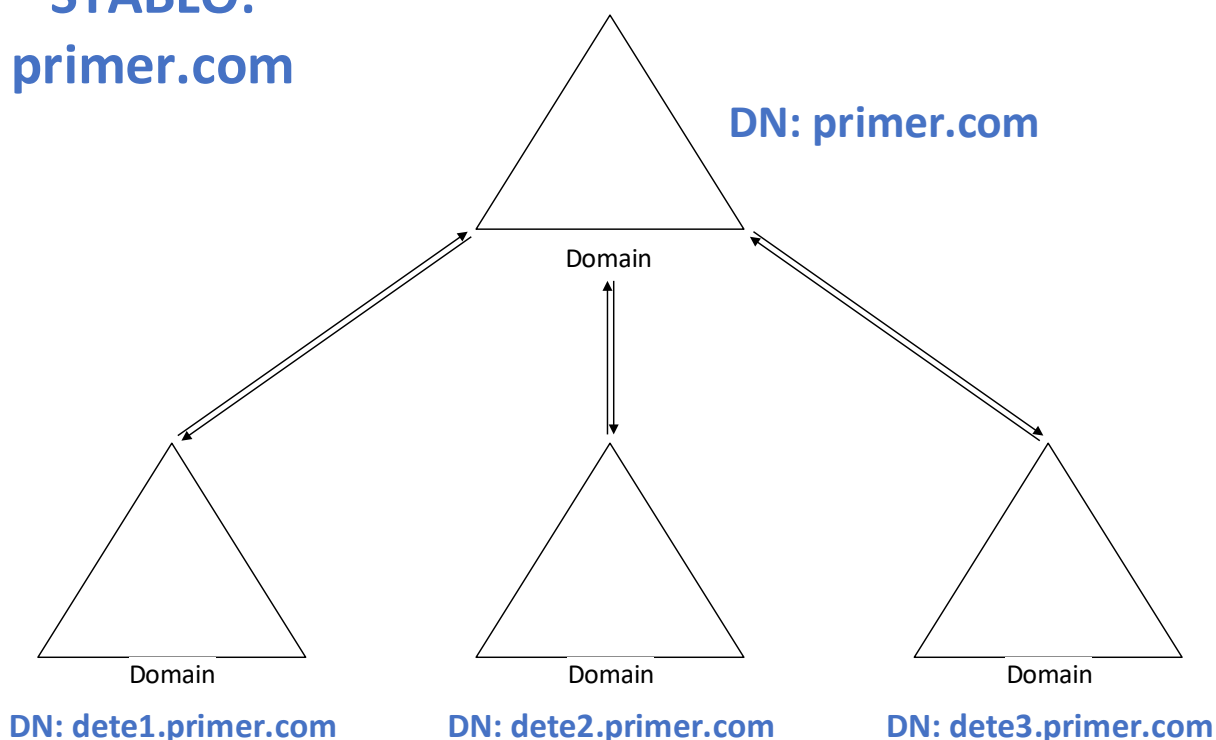
Organizacija slogova u stablu je hijerarhijska, ali se podaci u bazi podataka čuvaju serijski po redosledu kreiranja. Prvi slog u tabeli i koren stabla je uvek domen. Domen predstavlja logičku



grupu oko koje se formiraju objekti. Za upravljanje domenom su zaduženi kontrolori domena (eng. *Domain Controller - DC*). Kontrolor domena zadužen je za upravljanje samo jednim domenom, dok za jedan domen može da bude zaduženo više kontrolora domena. Ako više kontrolora domena upravlja jednim domenom, podaci se repliciraju između kontrolora domena. Kada se kreira prvi domen u *AD*, kreira se i stablo sa jednim elementom. U okviru tog stabla se mogu dodavati i drugi domen, ali stablo nosi naziv prvog domena koji je kreiran [4].

Na Slici 4 prikazano je stablo sa četiri domena, gde je prvo kreiran domen sa jedinstvenim imenom: *primer.com* i on je koren stabla, a njegova deca su listovi sa jedinstvenim imenima: *dete1.primer.com*, *dete2.primer.com* i *dete3.primer.com*. Naziv stabla je isti kao naziv prvog domena u stablu: *primer.com*. Ako novokreirani domen ne pokriva istu oblast informisanja sa ostalim domenima u stablu, novi domen kreira u drugom stablu. Više stabala u *AD* se zove šuma. Šuma se formira kada se kreira prvi domen, a pa i prvo stablo nosi naziv prvog domena [5].

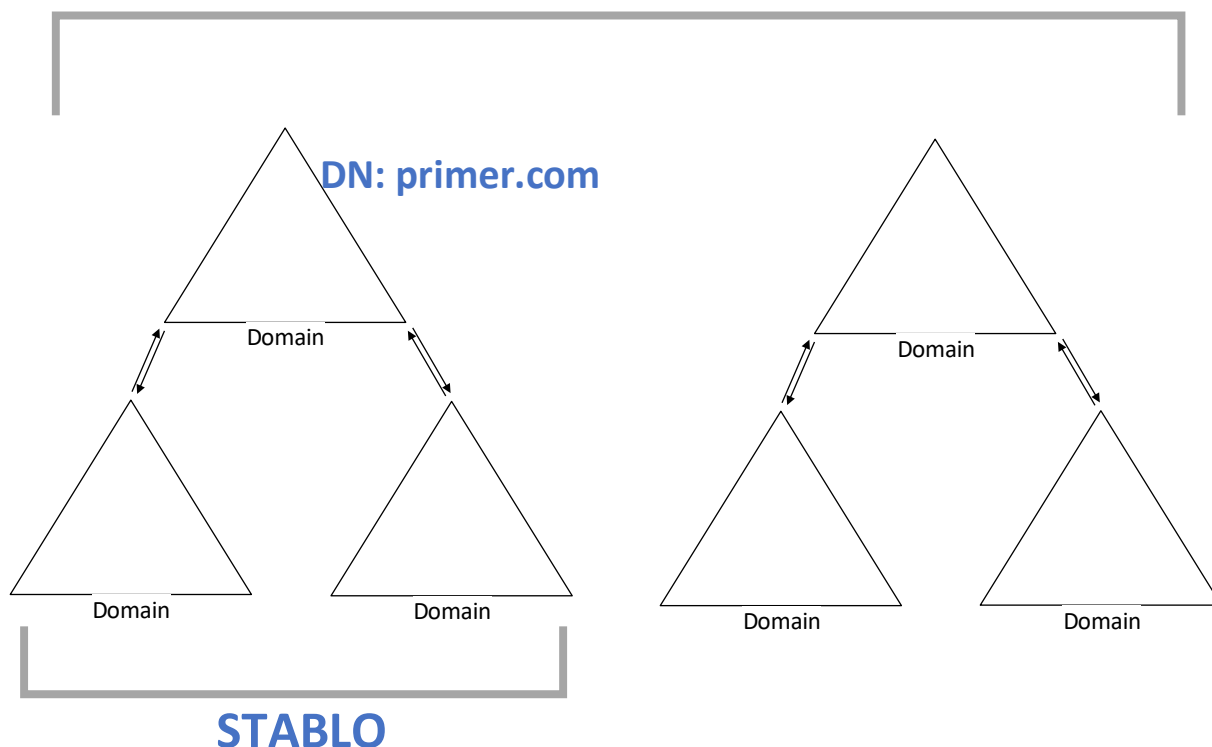
## STABLO: primer.com



Slika 4 - AD stablo - nosi naziv prvog domena

Primer šume sa dva stabla po tri domena može se videti na Slici 5. Na prvoj polovini slike je jedno stablo, na drugoj polovini slike je drugo stablo. Prvi domen koji je kreiran u šumi je domen u stablu na prvoj polovini slike sa jedinstvenim imenom *primer.com*, zato je naziv šume identičan.

## ŠUMA: primer.com



Slika 5 – AD šuma – nosi naziv prvog domena

### 2.3.2 Organizaciona jedinica, globalni katalog

Hijerarhijska struktura u *AD* – u, zahteva postojanje kontejnera, objekata koji mogu imati potomke. Organizaciona jedinica je osnovni tip kontejnera u *AD* koji administratoru omogućava bolju preglednost i lakšu organizaciju slogova. Kada se prvi put pokrene, *AD* podrazumevano ima nekoliko organizacionih jedinica, kao što su: *Users* i *Computers*.

Globalni katalog omogućava lakši pronalazak slogova u domenskoj šumi, kreirajući grupu slogova koji se potencijalno nalaze na različitim stablima, kako bi se olakšala pretraga. Globalni katalog ne čuva sve attribute slogova, već samo one koji češće koriste i oni se dodaju u novu parcijalnu kolekciju atributa i odatle se mogu dodavati i brisati [4].

### 2.3.3 Master kontrolori domena

Jedan domen mogu da uslužuju više kontrolora domena, a dozvolu za upravljanje domenom (poput izmene i dodavanja novih podataka) ranije su imali svi kontrolori jednog domena. To je često dovodilo do konflikta i nekonzistentnih podataka prilikom izmene podataka u bazi. Rešenje je uvođenje specijalnih uloga koje se dodeljuju pojedinačnim, master kontrolorima (*eng.Flexible Single Master Operator - FSMO*) i postoje pet različitih tipova [4]:

- a) Šema master ima dozvolu da menja postojeću šemu i nijedan drugi kontrolor domena ne može da vrši promene šeme. Inicijalni šema master je prvi kontrolor domena koji se kreira u šumi.
- b) Master imenovanja dodaje, briše, premešta domene i menja jedinstvena imena na osnovu nove pozicije u stablu ili šumi. Inicijalni master imenovanja je prvi kontrolor domena koji se kreira u šumi.
- c) Primarni master čuva i održava konzistentnost lozinke, u slučaju da se promena lozinke korisnika vrši sa više različitih kontrolora domena u isto ili približno sličnom vremenskom periodu. Kontrolor domena svaku izmenu lozinke prosleđuje i za svaku validaciju lozinke kontaktira primarni master. Druga uloga primarnog mastera je poravnanje vremena u domenu.
- d) Master identifikovanja obezbeđuje jedinstvenost bezbednosnih objekata u domenu.
- e) Infrastrukturni master čuva i održava reference između objekata iz različitih domena.

Tri od pet funkcionalnosti važe za jednog kontrolora domena na svakom domenu: primarni master, master identifikovanja i infrastrukturni master, a dva važe za čitavu šumu: šema master, master imenovanja. Ako jedan kontrolor domena prestane sa radom, dodeljivanje njegove master funkcionalnosti se ne vrši automatski, već je za to zadužen administrator [4].

### 2.3.4 Poverenje između domena

Objekti koji se nalaze u jednom domenu mogu da pristupaju resursima iz drugog domena samo ako je izgrađeno poverenje između dva domena. Poverenje može da bude izgrađeno između domena istog stabla, između domena različitog stabla i između domena u različitim domenskim šumama, a može imati sledeće putanje i karakteristike:

- a) Jednosmernost – pristup resursima je omogućen samo u jednom pravcu, npr. subjekti iz domena A mogu pristupati samo objektima iz domena B,
- b) Dvosmernost – pristup objektima omogućen u oba pravca,
- c) Netranzitivnost – ako se kreira poverenje od domena A prema domenu B i od domena B prema domenu C, domen A nema poverenje prema domenu C,
- d) Tranzitivnost – ako se kreira poverenje od domena A prema domenu B i od domena B prema domenu C, tada i domen A ima poverenje prema domenu C.

U Tabeli 2 prikazani su tipovi poverenja koji postoje u *AD* – u. U prvoj koloni su prikazani svi tipovi poverenja. U drugoj je definisana karakteristika poverenja, a u trećoj smer poverenja. U poslednjoj koloni je ukratko opisan svaki tip poverenja.

Tip poverenja	Karakteristika	Smer	Opis
<i>Parent – child</i>	Tranzitivnost	Dvosmeran	Dodavanjem novog deteta se uspostavlja poverenje između roditelja i deteta
<i>Tree – root</i>	Tranzitivnost	Dvosmeran	Poverenje između dva stabla
<i>External</i>	Netranzitivnost	Jednosmeran ili dvosmeran	Omogućava pristup resursima na <i>Windows NT 4.0</i> domenu ili domenu u drugoj šumi koji nije uključen u <i>Forest trust</i>
<i>Realm</i>	Tranzitivnost ili netranzitivnost	Jednosmeran ili dvosmeran	Formira poverenje između oblasti na <i>Windows</i> operativnom sistemu koji ne podržava <i>Kerberos</i> i <i>AD</i>
<i>Forest</i>	Tranzitivnost	Jednosmeran ili dvosmeran	Omogućava poverenje između domena jedne šume i domena druge šume
<i>Shortcut</i>	Tranzitivnost	Jednosmeran ili dvosmeran	Skraćuje putanju poverenja između dva domenu u šumi

Tabela 2 – Tabela poverenja između domena, preuzeta sa [6]

## 2.4 OpenLDAP

Na *Linux* operativnom sistemu postoji više implementacija *LDAP* protokola, kao što su: *Sun Microsystems's SunOne directory server*, *Novell's eDirectory*, *389 Directory Server*, *OpenLDAP*... Za ovo istraživanje je izabran *OpenLDAP*.

*OpenLDAP* predstavlja implementaciju informaciono-bezbednosnog sistema na *Linux OS*-u, pružajući bazu informacione-bezbednosti. Šema predstavlja način definisanja pravila za konstruisanje baze i formalno definiše: atribut, klase objekata i pravila za pristupanje bazi informacione-bezbednosti [7].

Sastoji od kolekcije konceptualno povezanih definicija šema, koje uređuju konstruisanje slogova u bazi informacione-bezbednosti. Postoje četiri osnovna tipa definicija šema [7]:

- a) definicija klasa objekata – definiše različite klase objekata, imena, jedinstvene identifikatore i moguće attribute,
- b) definicija atributa – definiše različite attribute, imena, tipove, vrednosti i ograničenja vrednosti,
- c) definicija identifikatora – povezuje vrednost i tip jedinstvenog identifikatora,
- d) definicija stabla – definiše pravila povezivanja čvorova u stablu.

Osim osnovnih tipova, pravila šeme su definisana i kroz druge definicije:

- a) Pravila upoređivanja – definiše pravila poređenja vrednosti atributa. Npr. pravilo za poređenje jedinstvenog imena sloga poredi vrednost atributa *member* sa jedinstvenim imenom sloga.
- b) Korišćenje definisanih pravila upoređivanja – primenjuje definisana pravila upoređivanja na attribute. Npr. vrednost atributa *member* može da bude samo jedinstveno ime.
- c) Definicija sintakse – definiše tip i vrednost atributa.
- d) Forme imena – definiše tip atributa koji se koristi kao *RDN*.

Osnovne šeme *OpenLDAP*-a, omogućuju upravljanje korisnicima, grupama i organizacionim jedinicama. *OpenLDAP* pruža mogućnost proširenja šeme i samim tim se otvara mogućnost za proširenje funkcionalnosti. Prilikom instalacije *OpenLDAP* sistema, inicijalno su uključene tri šeme koje obezbeđuju osnovne definicije za najčešće korišćene ekstenzije[7]:

- a) *Core.schema*,
- b) *Cosine.schema*,
- c) *Inetorgperson.schema*.

*OpenLDAP* se sastoji od dva pozadinska procesa koja obrađuju zahteve sistema:

- a) *Stand-alone LDAP Deamon* (eng. *SLAPD*) predstavlja *LDAP* servis kome klijent šalje zahteve, pozadinski proces ga obradi i šalje odgovor. Komunicira sa *LDAP* bazom podataka i vrši autentifikaciju, pretragu, izmenu podataka i dr.
- b) *Stand-alone LDAP Update Replication Deamon* (eng. *SLURPD*) pozadinski proces koji je zadužen za kopiranje podataka sa glavne *LDAP* baze podataka. Prati sve akcije *SLAPD* - a i replicira ih na druge servere.

Jedan od osnovnih načina za upis i izmenu podataka u informacionu-bazu je kreiranjem *LDAP Data Interchange Format* (eng. *LDIF*) datoteka. To je format za zapisivanje slogova u tekstualnu datoteku. Sa ovim formatom datoteke, informacioni podaci na serveru mogu se menjati, brisati ili dodavati novi. Primer *LDIF* datoteke:

```
dn: cn=peraPeric,dc=primer,dc=com
cn: Pera Peric
mail: pperic@primer.com
givenName: Pera
sn: Peric
objectClass: inetOrgPerson
```

Organizacija *OpenLDAP* – a se može predstaviti kroz četiri celine [7]:

- a) Serveri čine prvu celinu, a glavni server je *SLAPD*. Klijenti se preko mreže povezuju sa *SLAPD* serverom preko *LDAP* – a.
- b) Klijenti čitaju, upisuju, brišu i modifikuju *LDAP* podatke, a pre toga se konektuju i autentifikuju, a na samom kraju raskidaju konekciju.
- c) Uslužni programi održavaju *LDAP* servere, ali ne koriste protokol *LDAP*, već manipulišu servisima na nižim nivoima.
- d) Biblioteke obezbeđuju programske interfejsa za pristupanje *OpenLDAP* – u. Serveri, klijenti i uslužni programi koriste ove biblioteke.

## 2.5 Interoperabilnost

U današnjem svetu raste potreba za složenim sistemima, koji su sačinjeni od više funkcionalnih, međusobno usklađenih, manjih komponenti. Usklađenost komponenti predstavlja sposobnost razmene podataka između različitih komponenti i korišćenje primljenih podataka, a naziva se interoperabilnost. Postoje dve vrste interoperabilnosti [8]:

- a) sintaksna interoperabilnost – sistemi komuniciraju i razmenjuju podatke, čak iako im interfejsi ili programski jezici nisu isti,
- b) semantička interoperabilnost – razmenjeni tipovi podataka su razumljivi u oba sistema.

## 2.6 Procesna virtuelna mašina, *Microsoft .Net Core*

Procesna virtuelna mašina (*eng.Process Virtual Machine - PVM*) predstavlja okruženje koje apstrahuju *OS* i njegove primitive. Namenjeno je za izvršavanje aplikacije uvođenjem dodatnog nivoa apstrakcije od platforme kako bi se ista aplikacija mogla izvršavati na više različitih operativnih sistema i procesorskih arhitektura. *PVM* tipično obuhvata više komponenti – od prevođenja programskog rešenja, učitavanja i izvršavanja, automatskog rukovanja memorijom kao i skupa često korišćenih biblioteka.

Prevođenje koda se može vršiti iz jedne ili više faza. Moguće je interpretirati kod u mašinski u toku izvršavanja, kada korisnik pokreće datoteku sa izvornim kodom kao što je kod *Python* programskog jezika. Drugi pristup je da se prilikom razvoja prevodi u među asembler koji će se prevoditi u mašinski kod prilikom pokretanja. Ovakav pristup jer primenjen kod *.NET C#*, *.NET VB* i *Java* programskih jezika. Kod *.NET C#* programskog jezika se prvi nivo prevođenja vrši u među-asemblerki jezika - na primer *Intermediate language (eng.IL, kod C# programskog jezika)*. Prilikom izvršavanja se vrši prevođenje u mašinski kod koje obavlja *JIT (eng. Just-in-Time)* prevodilac. Među-assembler može se prevoditi na svim platformama za koje postoji odgovarajući *JIT* prevodilac [9].

Na Slici 6 prikazani su koraci prevođenja programskog koda kod *.NET C#* programskog jezika. Prvi korak prevođenja je u među-asmblerski kod, a zatim se iz među-asmblerskog koda prevođenje vrši u mašinsko.



Slika 6 - Koraci prilikom prevođenja C# koda u mašinski jezik

Rukovanje memorijom predstavlja zauzimanje i oslobađanje programske memorije. Memorijski prostor na kojem se dinamički zauzima i oslobađa memorija naziva se *heap*.

Alokacija memorije je zauzimanje memorijskog prostora i postiže se na dva načina: u toku prevođenja i u toku rada aplikacije. Dealokacija memorije predstavlja oslobađanje memorijskog prostora, može da bude [10]:

- a) eksplicitna – programer sam vrši oslobađanje memorijskog prostora,
- b) automatska – programsko okruženje oslobađa memorijski prostor uz specijalni mehanizam obrade memorijskog prostora.

*Garbage collector* predstavlja vrstu automatskog rukovanja memorijom. U zavisnosti od potreba aplikacije, *GC* se može pokretati u sledećim situacijama [11]:

- a) popunjen predefinisani deo *heap* – a
- b) velika fragmentacija memorije
- c) nakon alociranja velike količine memorije.
- d) nedostatak slobodne memorije na računaru.

*Mark Sweep* je *GC* algoritam za automatsko rukovanje memorijom. Osmišljen je 1960. godine (tvorac je *John McCarthy*), a sastoji se od tri faze:

- a) Pronađu se svi bazni objekti: objekti koji se čuvaju u registrima, globalne i statičke promenljive, lokalne varijable na *stack* – u, argumenti funkcija na *stack* – u i dr.
- b) Pronađu se i označe svi objekti koji su preko referenci povezani sa baznim objektima.
- c) Obrišu se svi objekti koji nisu označeni u prethodnoj fazi.

Mana *Mark Sweep* algoritma je fragmentovanje memorije. Fragmentovanje memorije je nastanak ‘iscepanih’ delova memorije, koji su neupotrebljivi i smanjuju kapacitet i performanse sistema. Fragmentovanje se rešava kompaktovanjem memorije, što predstavlja smanjivanje fragmentovanih delova, grupisanjem svih alociranih objekata na jednoj strani i svih slobodnih fragmenata memorije sa druge strane *heap* – a, omogućujući brže alociranje [11].

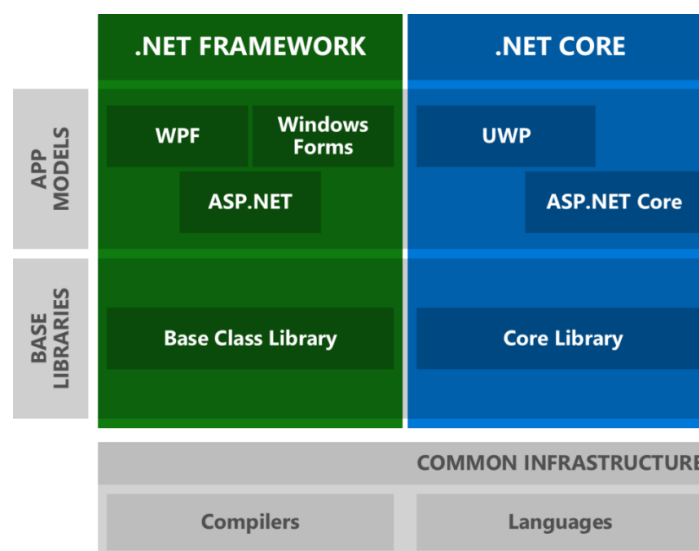
*Microsoft .NET Core* je procesna virtuelna mašina sa velikim brojem pomoćnih klasa. Prevođenje izvornog koda *.NET Core* – a oslanja se na među-asmblerski jezik. Osobine *.NET Core* – a su sledeće [12]:

- funkcionalnost na različitim platformama (*Windows, macOS, Linux*),
- konzistentnost na različitim arhitekturama (*x64, x86, ARM*),
- fleksibilan razvoj – može biti u okviru aplikacije ili instaliran *side-by-side* principom,
- kompatibilnost sa *.NET Framework, Xamarin* i *Mono* preko *.NET Standard* – a,
- .NET Core* platforma je besplatna i koristi *MIT* i *Apache 2* licence,
- podržana od strane Majkrosofta.

Podržava više programskih jezika kao što su: *C#, Visual Basic, C++* i *F#*, a mogu da se realizuju na programskim okruženjima: *Visual Studio, Visual Studio Code, Sublime Text, Vim* [12].

*Microsoft .NET Core* implementira *GC* sa *Mark Sweep Compact* algoritmom sa particionisanjem (deljenjem) *heap* – a. Particionisanje *heap* – a organizuje memorijski prostor, a vrši se po nekoliko različitih kriterijuma: veličina objekata, životni ciklus objekata, procesorskom jezgru, i dr. Povećavanje slobodnog memorijskog prostora na *.NET* – u se u zavisnosti od potrebe vrši komaktovanjem ili povećanjem veličine *heap* – a.

Na Slici 7 predstavljeno je poređenje *.NET Framework* i *.NET Core* procesnih virtuelnih mašina. Prikazana je razlika aplikativnih modela i različite kolekcije klasa na koje se oslanjaju, a donjem delu slike prikazana je zajednička infrastruktura, kompajleri i programski jezici koje koriste.



Slika 7 - *.Net Framework* i *.NET Core*

Za razvijanje predloženog rešenja za interoperabilni adapter u ovom master radu izabran je *.NET Core*, zato što i sam podržava različite platforme *OS* – a. Implementacijom interoperabilnog adaptera i testnom aplikacijom na *.NET Core* – u, omogućen je rad i testiranje rada sa bezbednosnim sistemima na različitim platformama *OS* – a.



### 3 Predloženo rešenje interoperabilnog adaptera

U ovom radu razvijen je interoperabilni adapter zasnovan na generičkom sloju, funkcionalan sa različitim informaciono-bezbednosnim sistemima:

- a) *AD* – om na *Windows* platformi,
- b) *OpenLDAP* – om na *Linux* platformi.

Identifikovane su i istražene biblioteke za rad sa bezbednosno-informacionim sistemima, funkcionalnostima koje nude i koje strukture podataka se koriste za razvijanje adaptera. Prilikom pokretanja adaptera kroz konfiguracionu datoteku se podešava informaciono-bezbednosni sistem sa kojim adapter komunicira, a izmenom na samo jednom mestu se menja komunikacija ka drugom bezbednosnom sistemu, bez dodatnih izmena u postojećem kodu. Dizajn rešenja zasnovan je na Fabričkim dizajn paternom, koji od korisnika sakriva način pristupanja *AD* i *OpenLDAP* informacionim bazama podataka.

Kako bi se konektovao na oba servera, adapter koristi biblioteke *System.DirectoryServices* i *System.DirectoryServices.AccountManagement* za komunikaciju sa *AD* – om, a *Novell.Directory.Ldap* za rad sa *OpenLDAP* – om.

#### 3.1 Arhitektura rešenja

Arhitektura sistema, prikazana na Slici 8, sastoji se od sledećih komponenti:

- a) *AD* i *OpenLDAP* bezbednosni servisi,
- b) aplikacija,
- c) konfiguraciona datoteka,
- d) interoperabilni adapter,
- e) datoteka za evidenciju događaja.

***AD* i *OpenLDAP*** su na Slici 8 prikazani kao dve baze podataka, zato što čuvaju informaciono-bezbednosne podatke. Strelica koja povezuje bezbednosne servise sa interoperabilnim adapterom ima dva pravca, zato što adapter šalje zahteve, a bezbednosni sistemi u suprotnom pravcu prosleđuju odgovore.

Implementirana je korisnička **aplikacija**, koja je povezana sa interoperabilnim adapterom i nudi korisnicima popunjavanje podataka za kreiranje novih, izlistavanje i modifikaciju postojećih slogova, pozivajući funkcije sa interoperabilnog adaptera. Aplikacija šalje zahteve interoperabilnom adapteru, a zatim prima i odgovore sa adaptera (takođe predstavljeno strelicom sa dva smeru).

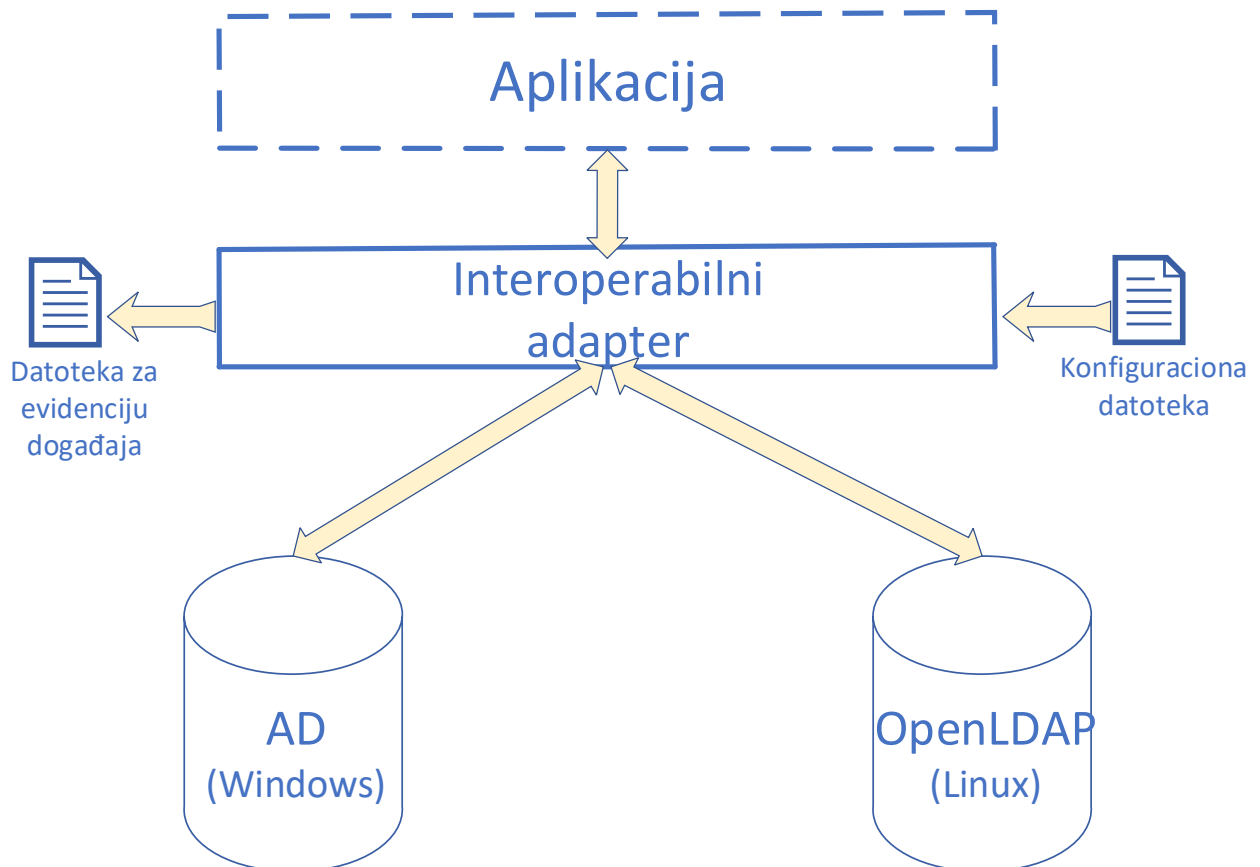
**Konfiguraciona datoteka** sadrži neophodne podatke za pokretanje sistema, a to su:

- a) izbor bezbednosnog servisa,
- b) naziv ciljanog domena (npr. *primer.com*),
- c) *IP* adresa i port bezbednosnog sistema.

Ako je prvi izbor bezbednosnog servisa *OpenLDAP*, korisnik je dužan da unese i internet protokol (eng. *Internet Protocol - IP*) adresu i port na kome je pokrenut server. Adresa i port nisu potrebni da se upisuju u konfiguraciona datoteka ako je izbor *AD*, već je uslov da se adapter pokreće na *Windows Server OS* – u i da korisnik bude ulogovan kao administrator ili da ima administratorska prava.

**Interoperabilni adapter** je smešten na generičkom sloju i predstavlja centralnu komponentu sistema, zato što je povezan sa svim ostalim komponentama. Prilikom pokretanja sistema, iščitavaju se podaci iz konfiguracione datoteke i adapter uspostavlja konekciju sa željenim informaciono-bezbednosnim servisom. Kada se uspostavi konekcija, komande se šalju sa aplikacije, do adaptera, koji ih prosleđuje do jednog od dva bezbednosna sistema. Bezbednosni sistem generiše odgovor i šalje adapteru.

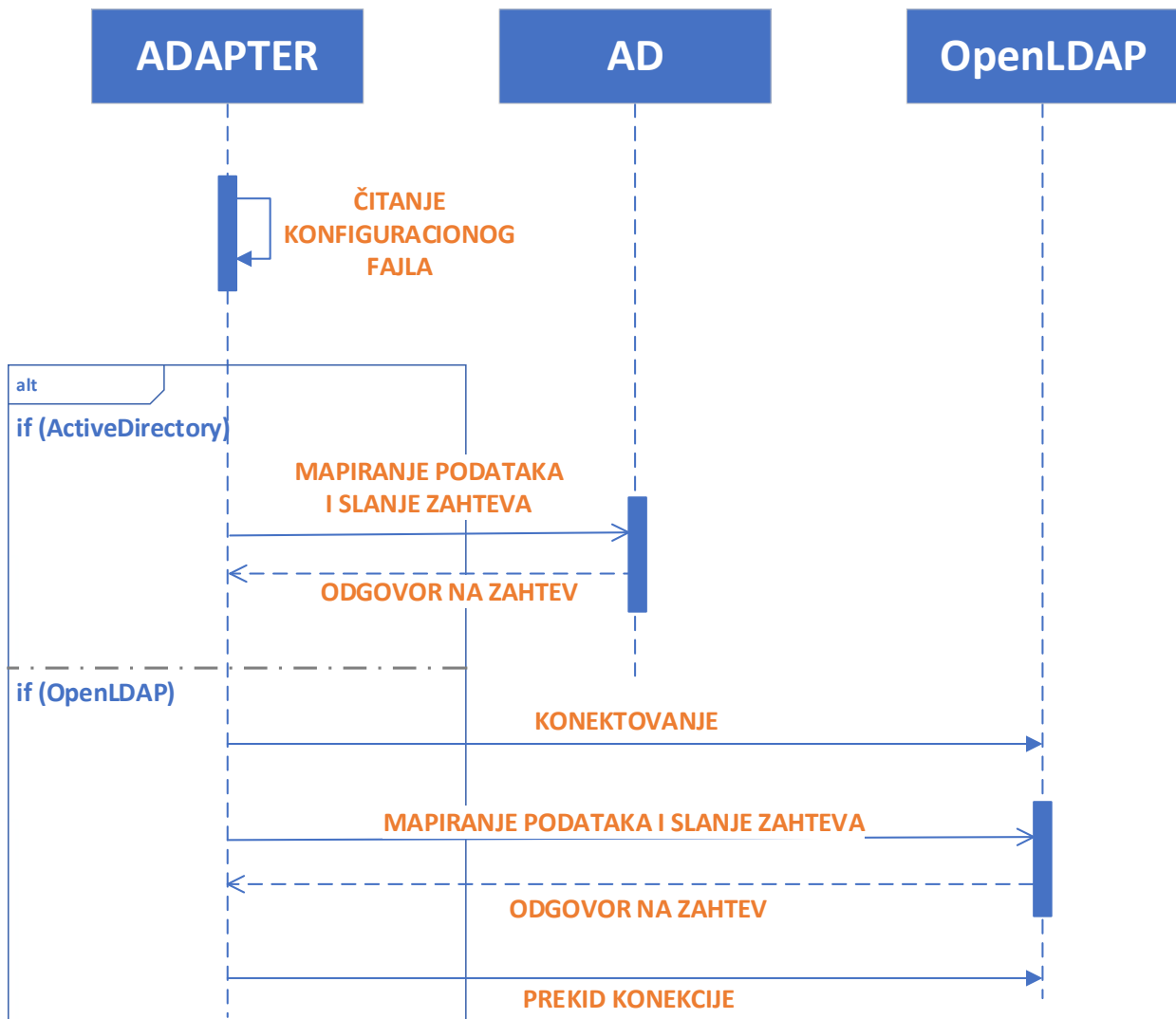
**Datoteka za evidenciju događaja** čuva zapis o poslednjem ulogovanom klijentu i vremenu kada se ulogovao.



Slika 8 – Arhitektura predloženog rešenja

Tok podataka opisan je sekvencijalnim dijagramom na Slici 9. Prikazan je komunikacioni ciklus između interoperabilnog adaptera i bezbednosnih servisa. Adapter iščitava podatke iz konfiguracione datoteke i na osnovu njih pristupa jednom od dva bezbednosna servisa. Ako je

izbor *OpenLDAP*, adapter šalje zahtev za konekcijom i uspostavom veze, dostavljajući korisničko ime i lozinku administratora. Kada je uspešno uspostavljena veza, adapter može da šalje zahteve za čitanje i upis podataka u bazi podataka.



Slika 9 - Sekvencijalni dijagram komunikacije adaptera i bezbednosnih sistema

### 3.2 Generički sloj

Interoperabilni adapter nalazi se na generičkom sloju, funkcionalnom na različitim platformama *OS* - a. Da bi se postigla nezavisnost od platforme na koju se oslanja sistem, rešenje je implementirano na *.NET Core* radnom okruženju koje se sastoji od procesne virtualne mašine i kolekcije klasa i sam je nezavisan je od platforme *OS* - a na koju se oslanja.

Generički sloj odvojen je od sloja bezbednosnih sistema i korisničkog sloja i funkcionalno odvaja ove dve celine. Zavisan je od bezbednosnih sistema, ali nije orijentisan samo prema jednom

bezbednosnom sistemu i lako je proširiv za druge sisteme, dok je u potpunosti nezavisan od korisničke aplikacije. Funkcije koje korisnik poziva sa generičkog sloja jednostavne su za korišćenje i omogućuju korisniku da bez znanja o komunikaciji interoperabilnog adaptera sa *AD* – om i *OpenLDAP* – om, intuitivno poziva funkcije svoje korisničke aplikacije.

### 3.3 Interoperabilni adapter

Za razvijanje interoperabilnog adaptera, prvo su istražene karakteristike bezbednosnih sistema:

- interfejsi za povezivanje sa *AD* i *OpenLDAP* bezbednosnim sistemima,
- tipovi podataka,
- način povezivanja podataka u informacionim bazama podataka,
- sličnosti i razlike tipova podataka i njihovih atributa između dva različita sistema.

U sledećem koraku mapirane su iste ili slične strukture podataka različitih bezbednosnih sistema. U trećem koraku su razvijene različite funkcije za pristup i izmenu podataka u informacionim bazama.

#### 3.3.1 Funkcionalnosti

*AD* u svojoj bazi podataka čuva informacije o korisnicima, grupama, računarima, kompjuterima, štampačima, aplikacijama i servisima. Za razliku od *AD*, *OpenLDAP* u svojoj osnovnoj šemi čuva podatke samo o korisnicima i grupama, pa su se prilikom implementacije rešenja realizovale funkcije za rad samo sa tim entitetima. Iako imena entiteta na dva različita sistema imaju određene sličnosti, imena tipova i vrednosti atributa se razlikuju. Zato je uvedeno mapiranje atributa entiteta generičkog sloja, sa semantički jednakim atributima entiteta *AD* i *OpenLDAP* sistema.

U Tabeli 3 predstavljeno je mapiranje atributa *AD* korisnika i *OpenLDAP* korisnika sa atributima generičkog korisnika sa interoperabilnog adaptera. U prvoj koloni su predstavljeni atributi generičkog korisnika, lako razumljivi korisniku interoperabilnog adaptera koji ne mora da poznaje specifičnosti informaciono-bezbednosnih sistema. U drugoj koloni izlistani su atributi na *AD* – u, a u poslednjoj atributi korisnika na *OpenLDAP* – u.

Generički korisnik	<i>AD</i> korisnik	<i>OpenLDAP</i> korisnik
<i>username</i>	<i>userPrincipalName</i>	<i>cn</i>
<i>name</i>	<i>givenName</i>	<i>givenName</i>
<i>surname</i>	<i>surname</i>	<i>sn</i>
<i>password</i>	<i>password</i>	<i>userPassword</i>
<i>phoneNumber</i>	<i>voiceTelephoneNumber</i>	<i>mobile</i>
<i>emailAddress</i>	<i>emailAddress</i>	<i>mail</i>
<i>description</i>	<i>description</i>	<i>description</i>
	<i>DistinguishedName</i>	<i>DistinguishedName</i>

Tabela 3 - Mapiranje korisničkih atributa generičkog sloja na na *AD* i *OpenLDAP* attribute

U Tabeli 4 predstavljeno je mapiranje atributa *AD* grupe i *OpenLDAP* grupe sa atributima generičke grupe. U prvoj koloni su izlistani atributi generičke grupe, u drugoj koloni atributi *AD* grupe, a u poslednjoj atributi *OpenLDAP* grupe. Na generičkom sloju i *AD* grupi struktura sadrži atribut koji predstavlja listu članova, dok *OpenLDAP* grupa za svakog člana ima poseban atribut.

Generička grupa	<i>AD</i> grupa	<i>OpenLDAP</i> grupa
<i>groupName</i>	<i>name</i>	<i>cn</i>
<i>members&lt;string&gt;</i>	<i>members&lt;PrincipalCollection&gt;</i>	<i>member&lt;string&gt;</i>
<i>Description</i>	<i>description</i>	<i>description</i>
	<i>DistinguishedName</i>	<i>DistinguishedName</i>

Tabela 4 - Mapiranje grupnih atributa generičkog sloja na *AD* i *OpenLDAP* attribute grupe

U obe tabele *AD* i *OpenLDAP* kolone sadrže jedan atribut više nego druge dve kolone, a to je jedinstveno ime koje se na generičkom sloju ne prikazuje korisniku zbog jednostavnije upotrebe.

Za upravljanje entitetima razvijene su dve grupe funkcija: administratorske i korisničke. Administratorske funkcije izlistane su u Tabeli 5 i Tabeli 6, a korisničke funkcije izlistane su u Tabeli 7.

U Tabeli 5 su prikazane i opisane administratorske funkcije dodavanje, izmenu i brisanje korisnika i grupa. U prvoj koloni su nazivi funkcija, a u drugoj koloni je opis funkcije.

Naziv funkcije	Opis
<i>CreateUser</i>	Kreiranje korisnika
<i>DeleteUser</i>	Brisanje korisnika
<i>ChangeUserUsername</i>	Izmena jedinstvenog imena korisnika
<i>ChangeUserName</i>	Izmena imena korisnika
<i>ChangeUserSurname</i>	Izmena prezimena korisnika
<i>ChangeUserPassword</i>	Izmena lozinke korisnika
<i>ChangeUserEmail</i>	Izmena adrese elektronske pošte korisnika
<i>ChangeUserPhoneNumber</i>	Izmena broja telefona korisnika
<i>ChangeUserDescription</i>	Izmena opisa korisnika
<i>CreateGroup</i>	Kreiranje grupe
<i>ChangeGroupName</i>	Izmena jedinstvenog imena grupe
<i>DeleteGroup</i>	Brisanje grupe

Tabela 5 – Administratorske funkcije za manipulisanje korisnicima i grupama

U Tabeli 6 su prikazane i opisane administratorske funkcije za menjanje i proveru članstva u grupi i izlistavanje slogova. U prvoj koloni je naziv funkcije, a u drugoj opis funkcije.

Naziv funkcije	Opis
<i>AddUserToGroup</i>	Dodavanje korisnika u grupu
<i>AddGroupToAnotherGroup</i>	Dodavanje grupe u drugu grupu
<i>IsMemberOf</i>	Proveravanje članstva
<i>GetAllGroupMembers</i>	Dobavljanje članova
<i>AssignPermission</i>	Dodeljivanje ugrađene dozvole pristupa
<i>RemovePermission</i>	Oduzimanje ugrađene dozvole pristupa
<i>CreateCustomPermission</i>	Kreiranje nove dozvole pristupa
<i>AssignCustomPermission</i>	Dodeljivanje nove dozvole pristupa
<i>RemoveCustomPermission</i>	Oduzimanje nove dozvole pristupa
<i>ListAllUsers</i>	Izlistavanje korisnika
<i>ListAllGroups</i>	Izlistavanje grupe

Tabela 6 – Administratorske funkcije članstva i dozvola pristupa

U Tabeli 7 su prikazane implementirane korisničke funkcije, u prvoj koloni je naziv, a u drugoj opis funkcije.

Naziv funkcije	Opis
<i>Login</i>	Logovanje korisnika
<i>CheckPermission</i>	Proveravanje dozvole prisutpa

Tabela 7 - Korisničke funkcije

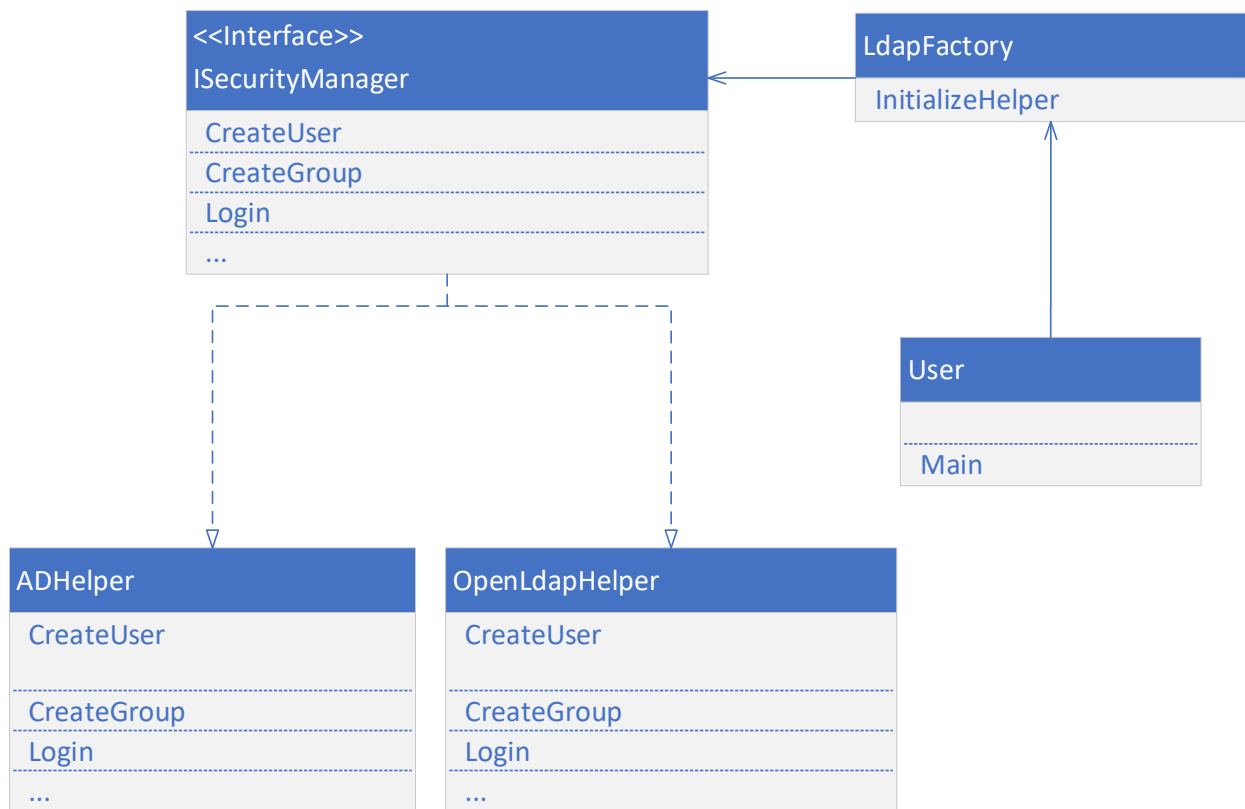
### 3.3.2 Dizajn

Prilikom dizajniranja interoperabilnog adaptera primenjen je fabrički dizajn patern. Fabrički dizajn patern se koristi u slučaju postojanja većeg broja sličnih klasa, kada programer ne zna koji objekat je potreban u specifičnom trenutku, do startovanja aplikacije. Komponente implementiranog dizajn patern su:

- LdapFactory* – komponenta zadužena za povezivanje adaptera sa korisničkom aplikacijom, na osnovu podataka iz konfiguracione datoteke kreira *AD* i *OpenLDAP* menadžere funkcija,
- ISecurityManager* – komponenta koja sadrži spisak funkcionalnosti,
- ADHelper* – *AD* menadžer funkcija, nasleđuje *ISecurityManager*,
- OpenLdapHelper* – *OpenLDAP* menadžer funkcija, nasleđuje *ISecurityManager*.

Fabrički dizajn patern interoperabilnog adaptera, prikazan na Slici 10, od korisnika sakriva logiku kreiranja objekata, tako što *LdapFactory* i *ISecurityManager* komponente preuzimaju svu odgovornost za kreiranje *ADHelper* i *OpenLDAPHelper* menadžera funkcionalnosti. Fabričkim dizajn paternom interoperabilni adapter je lako proširiv i u slučaju dodavanja novog menadžera funkcionalnosti za pristup novom informaciono-bezbednosnom servisu, postojeće rešenje se neće menjati. Novi menadžer funkcionalnosti će naslediti *ISecurityManager* interfejs i implementirati svoje funkcionalnosti za pristup bazi informacionih podataka.

Na Slici 10 prikazan je implementirani dizajn patern. Korisnik instancira *LdapFactory* klasu i poziva funkciju *IntializeHelper*, kojoj prosleđuje podatke iščitane iz konfiguracione datoteke (izbor bezbednosnog sistema, domen, korisničko ime i lozinka admina), a povratna vrednost ove funkcije je instance klase *ADHelper* ili *OpenLDAP*, koja se dostavlja korisniku. Uspostavljena komunikacija i kreirani menadžeri korisniku obezbeđuju funkcije upravljanja bazama informacionih-bezbednosti, opisanih u tabelama 5, 6 i 7.



Slika 10 - Fabrički dizajn patern primenjen kod interoperabilnog adaptera

### 3.4 Active Directory menadžer funkcija

*System.DirectoryServices* biblioteka omogućava pristup i rad sa AD bezbednosnim sistemom. Za potrebe ovog rada korišćene su dve klase: *DirectoryEntry* i *DirectorySearcher*, koje se oslanjaju na *Active Directory Services Interface (ADSI)* tehnologiju. *ADSI* je skup interfejsa koje koriste administratori za pristupanje i lociranje resursa na mreži [13].

Informaciono-bezbednosni sistemi organizovani su u strukturu stabla i *DirectoryEntry* klasa predstavlja jedan čvor u stablu, dok *DirectorySearcher* klasa vrši upite i pretragu baze informacione-bezbednosti [14].

Drugo biblioteci pripadaju *Principal* klase:

- a) *UserPrincipal* – zadužene za rad sa korisnicima,
- b) *GroupPrincipal* – zadužene za rad sa grupama,
- c) *ComputerPrincipal* – zadužene za rad sa kompjuterima,
- d) *PrincipalCollection* – kolekcija *Principal* objekata,
- e) *PrincipalContext* – definiše u kom domenu se vrši manipulisanje objektima.

*OpenLDAP* u inicijalnim šemama ne koristi objekte koji označavaju kompjutere, pa klasa *ComputerPrincipal* nema značaj u okviru AD menadžera funkcionalnosti. Za pronalazak sloga u bazi informacionih podataka korisnik navodi u kom domenu se traženi entitet nalazi i to se postiže kreiranjem *PrincipalContext* klase, koja se prosleđuje kao parametar konstruktora ostalih *Principal* klasa [15].

Za izradu ovog master rada koristile su se *Principal* klase pre klasa koje obezbeđuje *DirectoryServices* biblioteka, kad god je to bilo moguće, zbog lakšeg razumevanja i preglednijeg kodiranja.

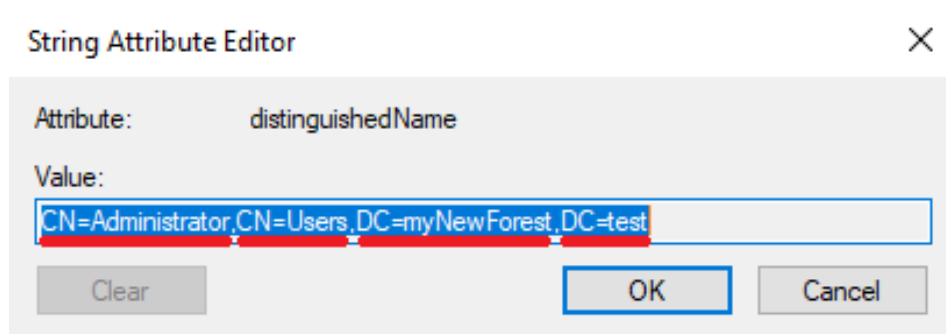
#### 3.4.1 Kreiranje novog korisnika

Za kreiranje novog korisnika koristi se *UserPrincipal* klasa. Podatke koji se proslede funkciji za kreiranje potrebno je spojiti sa odgovarajućim atributima koje nudi AD. Kada korisnik unese korisničko ime, očekivano je da ne postoji nijedan drugi korisnik sa tim korisničkim imenom i zato je bilo potrebno da se istraži koji atribut identifikuje korisnika, a da je jedinstven u domenu kome objekat pripada. Izbor je spao na atribut *sAMAccountName* i *userPrincipalName*. *sAMAccountName* se koristio za neke od prethodnih verzija OS-a (kao što je *Windows NT 4.0*), pa se zbog toga korisničko ime sa generičkog sloja mapira na atribut *userPrincipalName* [16]. Osim korisničkog imena, ime i prezime korisnika i lozinka su obavezni parametri koji se šalju sa generičkog sloja, a opcioni parametri su broj telefona, adresa elektronske pošte i opis korisnika. Mapiranje atributa može se videti u Tabeli 3.

Atribut koji jedinstveno identifikuje objekat je jedinstveno ime korisnika, a AD olakšava rad, u odnosu na *OpenLDAP* automatskim generisanjem jedinstvenog imena korisnika, na osnovu atributa *userPrincipalName* i pozicije u stablu gde se objekat nalazi, specijalno naznačeno *PrincipalContext* klasom. Na Slici 11 može se videti DN administratora. *CN=Administrator* je



*RDN* administratora, *CN=Users* je organizaciona jedinica, a domen je *myNewForest.test*. Organizaciona jedinica *Users* se podrazumevano kreira sa instalacijom *AD* – a.



Slika 11 - Jedinstveno ime AD korisnika

### 3.4.2 Brisanje korisnika

Za pronalazak korisnika u bazi informacione-bezbednosti se koristi *UserPrincipal* klasa, tako što joj se prosledi instanca klase *PrincipalContext* i korisničko ime traženog korisnika. *UserPrincipal* ima ugrađenu funkcionalnost za brisanje, a mogu da se obrišu samo korisnici koji u stablu predstavljaju listove.

### 3.4.3 Izmena korisničkih podataka

Pretraga korisnika se izvrši na isti način kao kod brisanja, a zatim se pristupa atributima *UserPrincipal* klase, izlistani u Tabeli 3, i dodeljuju im se nove vrednosti. Prilikom izmene korisničkom imena, novo korisničko ime mora da bude jedinstveno.

Za izmenu lozinke se poziva funkcija *SetPassword*, koja je deo *Principal* klase.

### 3.4.4 Kreiranje grupe

Funkcionalnost za kreiranje grupe koristi *GroupPrincipal* klasu. Korisnik upisuje naziv grupe, koji je jedinstven u okviru domena gde se nalazi. U *AD* – u postoje dva tipa grupa: *Distribution* i *Security*. U okviru istraživanja za ovaj master rad su korišćene *Security* grupe, zato što korisnicima mogu da se dodele dozvole pristupa, za razliku od *Distribution* grupa. Osim naziva grupe grupe, korisnik prosleđuje i opis grupe

### 3.4.5 Brisanje grupe

Grupa se prvo locira u domenu *PrincipalContext* klasom i nazivom grupe (na isti način kako se pronalazi i korisnik). *GroupPrincipal* klasa ima ugrađenu funkcionalnost za brisanje, ali, kao i za korisnika, ne sme se obrisati grupa koja nije list u stablu.

### 3.4.6 Izmena grupnih podataka

Grupa se prvo locira u domenu *PrincipalContext* klasom i nazivom grupe. Pristupa se atributima *GroupPrincipal* klase i postavljaju se nove vrednosti. Novo ime grupe mora biti jedinstveno u domenu. Atributi su izlistani u Tabeli 4.

### 3.4.7 Dodavanje korisnika ili grupe u grupu

U bazi informacionih podataka se lociraju traženi objekti, a zatim se pristupi članovima grupe preko *GroupPrincipal* klase i doda se novi član. Lista članova je generičkog tipa *PrincipalCollection*, što znači da i *UserPrincipal* i *GroupPrincipal* mogu da se dodaju kao novi član.

### 3.4.8 Dodeljivanje ugrađenih dozvola pristupa

Proširivanje liste dozvole pristupa subjekata nad objektima, ne mogu se koristiti *Principal* klase iz *AccountManagement* biblioteke, zato što nemaju ugrađene metode za pristupanje listi kontrole pristupa. Zbog toga se koristi *DirectoryEntry* klasa. Osim ove klase, koriste se:

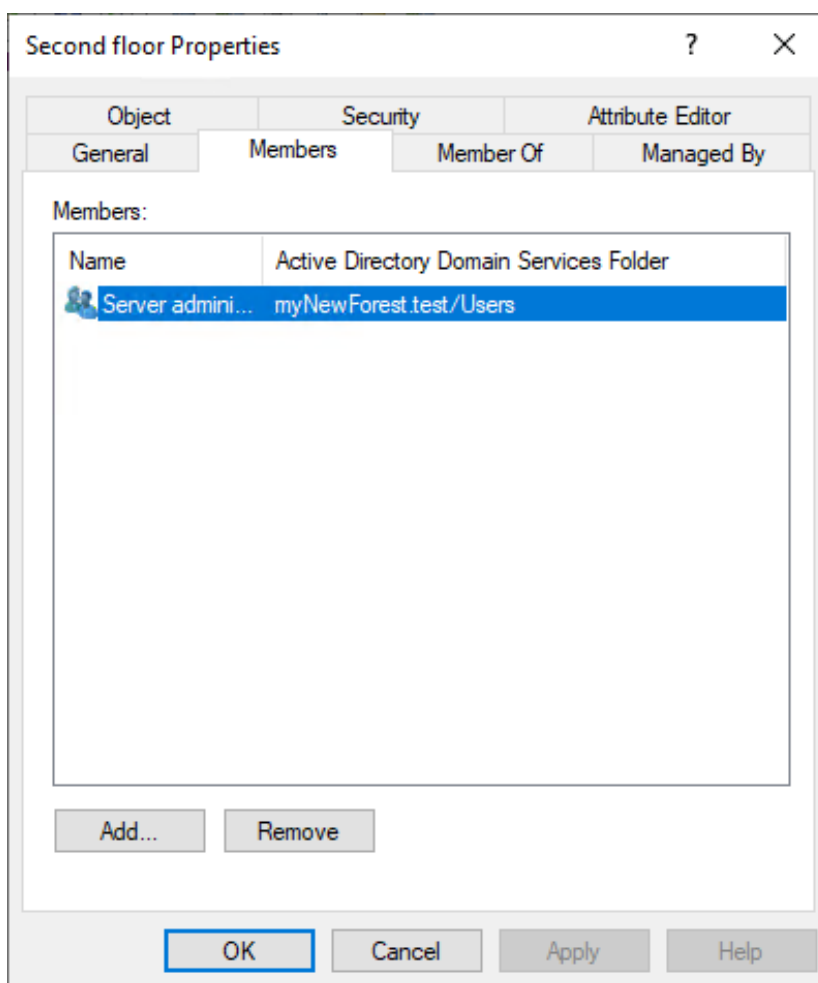
- a) *NTAccount* klasa – deo biblioteke *System.SecurityPrincipal*, označava korisnički ili grupni nalog (nije vezana samo za AD),
- b) *ActiveDirectoryAccessRule* – deo biblioteke *DirectoryServices* koji definiše vrstu objekta dodele pristupa. Opisuje da li subjekat ima ili nema dozvolu da pristupi objektu. To je jedan objekat iz liste kontrole pristupa.

AD ima listu ugrađenih dozvola: dozvola za upisivanje i čitanje objekta, dozvola da kreira, briše i pristupa deci objekta, dozvola za čitanje i pisanje atributa objekta, dozvola za brisanje objekta i brisanje svih potomaka objekta i dr. Za svaki objekat dozvole pristupa subjektu se odobrava ili zabranjuje pristup i dodeljuje mu se tip nasleđivanja, koji može da bude:

- a) *None* – dozvola pristupa važi samo za objekat kome je dodeljeno,
- b) *All* – dozvola pristupa važi za objekat i za sve njegove potomke,
- c) *Descendants* – dozvola pristupa važi samo za njegove potomke,
- d) *SelfAndChildren* – dozvola pristupa važi za objekat i za njegovu decu (direktne potomke),
- e) *Children* – dozvola pristupa važi samo za direktne potomke objekta.

### 3.4.9 Kreiranje, dodavanje i dodeljivanje novih dozvola pristupa

Kako bi se kompanijama omogućilo da dodeljuju dozvole pristupa koje prate oblast poslovanja, u okviru rada je istraženo kako se dodaju nove dozvole pristupa u listu ugrađenih dozvola. Izveden je zaključak da nije moguće menjati postojeću listu dozvola pristupa. Ovaj problem se rešava kreiranjem nove grupe. Dodeljivanje ovih dozvola korisniku vrši se dodavanjem korisnika kao člana te grupe. Na Slici 12 može se videti primer kreirane grupe *Second floor* (drugi sprat) i administratora kao člana te grupe. U praksi bi to značilo da administrator ima pravo pristupa drugom spratu.



Slika 12 - Administrator član grupe *Second floor*

#### 3.4.10 Logovanje korisnika i auditovanje

Prilikom logovanja, korisnik unosi svoje korisničko ime i lozinku. Ako lozinka prođe verifikaciju, u datoteku za evidentiranje logovanja zapisuje se koji korisnik se ulogovao i kada se ulogovao.

#### 3.4.11 Proveravanje dozvole pristupa

Kako bi se proverila da li korisnik ima dozvolu pristupa koja nije ugrađena, proverava se da li je subjekat član grupe. Za proveravanje ugrađenih dozvola pristupa, objekat kome subjekat pristupa se iz baze podataka čita *DirectoryEntry* klasom, koja sadrži polje sa listom subjekata i dozvolama koje imaju subjekti nad objektom.

### 3.5 *OpenLDAP* menadžer funkcija

*Novell.Directory.Ldap* biblioteka obezbeđuje konekciju i manipulisanje podacima na *OpenLDAP* bezbednosnom sistemu. Klase koje se koriste za razvijanje rešenja su: *LdapEntry*, *LdapConnection*, *LdapAttributeSet*, *LdapAttribute* i *LdapModification*.

*LdapEntry* je klasa koja predstavlja čvor u stablu i slog u bazi podataka.

*LdapConnection* klasa je zadužena za uspostavljanje konekcije sa *OpenLDAP* bezbednosnim sistemom. Uspostavljanje konekcije započinje povezivanjem sa *OpenLDAP* – om pozivanjem *Connect*, *LdapConnection* klase. Zatim sledi autentifikacija, pozivanjem funkcije *Bind* i prosleđivanjem *DN* – a i lozinke administratora, a nakon uspešnog dokazivanja identiteta, pozivaju se funkcije za manipulaciju podataka u bazi informacionih-bezbednosti. Na samom kraju sesije zatvara se konekcija pozivanjem funkcije *Disconnect*.

Atributi slogova kreiraju se nezavisno od sloga, instanciranjem *LdapAttribute* klase sa tipom atributa i vrednosti atributa. Svi atributi jednog sloga ugrađuju se u *LdapAttributeSet* klasu koja predstavlja kolekciju atributa i u tom obliku se prosleđuju *LdapEntry* klasi za kreiranje sloga.

*LdapModification* klasa definiše kakve izmene će se izvršiti nad čvorom. Može da doda, obriše ili zameni već postojeći atribut novim atributom, a za izmenu se koristi opcija zamene.

Slogovi se pretražuju pozivanjem funkcije *Search*, *LdapConnection* klase na dva načina [17]:

- a) sinhrono – povratna vrednost je struktura *LdapSearchResults*,
- b) asinhrono – povratna vrednost je struktura *LdapSearchQueue*.

Prednosti biblioteka za rad sa *AD* – om, u odnosu na *OpenLDAP* biblioteke su:

- a) biblioteke Majkrosofta za rad sa *AD* bezbednosnim sistemom ne zahtevaju ručnu uspostavu konekcije i povezivanje sa serverom (*Connect* i *Bind*),
- b) *Principal* klase automatski generišu *DN*,
- c) lakši pristup i izmena atributa izmena atributa,

što omogućuje jednostavnije razvijanje *AD* menadžera, u odnosu na *OpenLDAP* menadžer.

#### 3.5.1 Kreiranje korisnika

Za istraživanje atributa dostupnih kod korisnika na *OpenLDAP* sistemu, u okviru ovog master rada istraživalo se koji tipovi klasa objekata postoje. Rezultat istraživanja je pokazao da u osnovnim šemama korisnici mogu imati: *organizationalPerson*, *inetOrgPerson*, *posixAccount* i druge klase objekata.

*InetOrgPerson* je klasa objekata koja obezbeđuje najširi skup različitih atributa, u odnosu na druge dve klase, a nazivi atributa se najviše poklapaju sa atributima korisnika iz *AD* šeme, zbog toga je *inetOrgPerson* klasa izabrana za korisnika. Kao i kod *AD* menadžera funkcionalnosti, i u ovom slučaju je bilo potrebno da se istraži atribut koji jedinstveno identifikuje korisnika i koji će biti predstavljati korisničko ime.

Rezultat istraživanja pokazao je da atributi *uid* (eng. *unique identifier*) i *cn* jedinstveno identifikuju korisnika, ali je korisničko ime mapirano na *cn*, zato što je ovaj atribut obavezan u izabranoj klasi objekata. Kada se kreira novi objekat, klasa *LdapEntry* prima dva parametra: *DN* i listu atributa koji sadrži objekat.

### 3.5.2 Kreiranje grupe

Prilikom kreiranja grupe istražene su klase objekata, a kao rezultat istraživanja ukazuje da grupe mogu da imaju nekoliko različitih klasa objekata: *posixGroup*, *groupOfNames*, *groupOfUniqueNames* i dr.

*PosixGroup* nema atribut koji pokazuje koji su članovi grupe, pa zbog toga nije korišćen prilikom izrade ovog rešenja. *GroupOfUniqueNames* zahteva da se lista članova unosi po njihovom jedinstvenim identifikatorima – *uid*, a s obzirom na to da korisnici za jedinstvenu identifikaciju koriste samo *cn*, a da se *uid* ne koristi kod izrade rešenja, kao izbor je preostala objektna klasa *groupOfNames*, koja ujedno predstavlja klasu objekata inicijalno dodeljenu grupama [7]. U Tabeli 4 mogu se videti atributi grupe.

### 3.5.3 Promena korisničkog ili grupnog imena

Prilikom promene imena potrebno je proveriti da li to ime već postoji. *LdapConnection* klasa sadrži funkciju za promenu imena – *Rename*, i koristi se za sve čvorove, nezavisno od tipa. Kao parametri funkcije se prosleđuju:

- a) trenutni *DN* čvora,
- b) novi *cn* čvora,
- c) *bool* vrednost – da li da se obriše staro ime.

### 3.5.4 Brisanje čvora

Funkcionalnost brisanja čvorova razvijena je isto za korisnike i za grupe. *LdapConnection* klasa sadrži funkciju za brisanje – *Delete*. Kao parametar se prosledi *DN* čvora koje se briše, pod uslovom da predstavlja list u stablu.

Za promenu jedinstvenog imena i za brisanje čvora je potrebno kompletan *DN* korisnika (npr. *cn=PeraPeric,dc=primer,dc=com*). Zadatak generičkog sloja je da korisniku omogući upravljanje uz minimalna znanja o bezbednosnim sistemima, pa korisnik unosi korisničko ime ili ime grupe, a adapter generiše *DN* čvora, na osnovu imena i poziciju u stablu.

### 3.5.5 Izmena korisničkih podataka

Izmena atributa se realizuje pozivanjem *Modify* funkcije *LdapConnection* klase, a prosleđuju se dva parametra: jedinstveno ime čvora nad kojim se vrše izmene i instanca klase *LdapModification*, sa definisanim izmenama koje se primenjuju na čvoru.

### 3.5.6 Dodavanje korisnika/grupe u grupu

Funkcionalnost se implementira izmenom atributa grupe, korišćenjem *LdapModification* klase, dodavanjem novog atributa tipa *member*, a vrednost je jedinstveno ime korisnika ili grupe koji se učlanjuju (npr. ako se u grupu doda pet novih članova, grupa će imati pet *member* atributa).

### 3.5.7 Proveravanje članstva

*LdapSearch* filtrira sve attribute sa tipom *member*. Proveravanje članstva vrši se iteriranjem kroz sve vrednosti *member* atributa.

### 3.5.8 Dodeljivanje ugrađenih dozvola pristupa

Razvijanje rešenja u ovom master radu obuhvata samo osnovne šeme na oba informaciono-bezbednosna sistema. Osnovne šeme na *OpenLDAP* – u ne sadrže listu ugrađenih kontrola pristupa i zbog toga se subjektima ne može dodeliti dozvola pristupa nekom objektu. Funkcionalnost je ostala neimplementirana.

### 3.5.9 Dodavanje i dodeljivanje novih dozvola pristupa

Dodavanje novih dozvola pristupa se rešava identično kao rešenje na *AD* – u, učlanjenjem subjekta u grupu, čiji naziv definiše dozvolu koju imaju subjekti. Provera da li korisnik ima dozvolu za pristup objektu se rešava proverom članstva u grupi.

### 3.5.10 Logovanje korisnika i auditovanje

Prilikom logovanja, korisnik unosi svoje korisničko ime i lozinku. Ako lozinka prođe verifikaciju, u datoteku za evidentiranje logovanja se zapisuje koji korisnik se ulogovao i kada se ulogovao.

### 3.5.11 Proveravanje dozvole pristupa

Kako bi se proverila da li korisnik ima dozvolu pristupa koja nije ugrađena, proverava se da li je subjekat član grupe. Ugrađene dozvole pristupa nisu implementirane u okviru osnovnih šema *OpenLDAP* bezbednosnih sistema.

## 4 Verifikacija rešenja

Za verifikaciju rešenja kreirana je testna aplikacija, koja simulira akcije korisnika rešenja, pozivajući funkcije sa interoperabilnog adaptera.

### 4.1 Testno okruženje

Za testno okruženje u ovom istraživanju instalirani su *Windows Server 2019* i *Linux* operativni sistem *Ubuntu 18.04*. Bezbednosni sistem *AD* je instaliran na *Windows Server OS* i koristi ugrađenu *ActiveDirectory* šemu. Drugi sistem na kojem je testirano rešenje je *OpenLDAP*, verzija 2.4.48, sa osnovnim šemama:

- a) *Core.schema*,
- b) *Cosine.schema*,
- c) *Inetorgperson.schema*.

Za verifikaciju, kao i za kreiranje interoperabilnog adaptera korišćena je procesna virtuelna mašina *Microsoft.NETCore 2.1*, na programskom jeziku *C# 7.3*, da bi testna aplikacija bila kompatibilna sa adapterom i kako bi mogla da se primeni na dva različita *OS* – a.

Korišćene su i sledeće verzije programskih biblioteka:

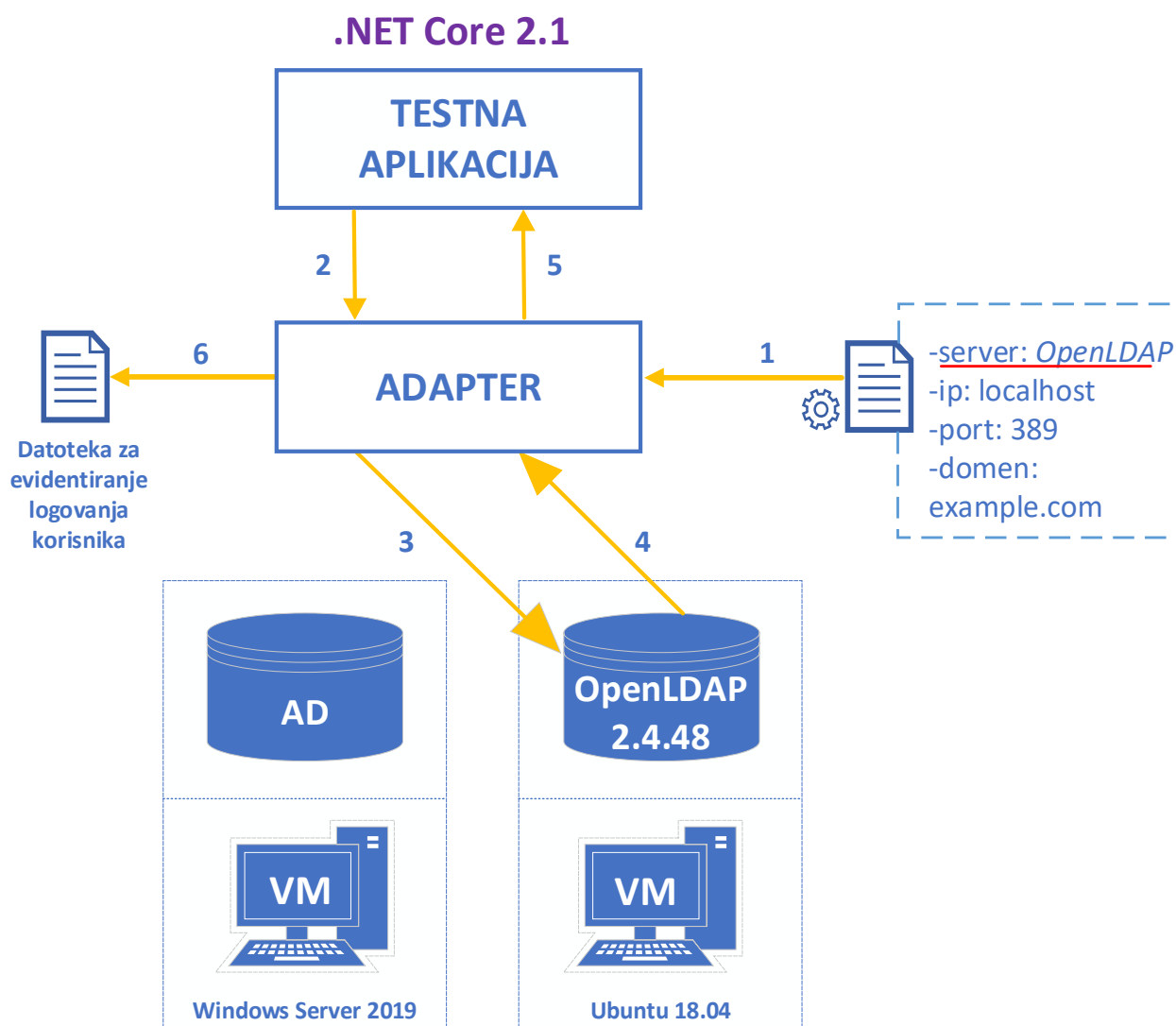
- a) *Novell.Directory.Ldap 3.1.0*,
- b) *System.DirectoryServices 4.6.0*,
- c) *System.DirectoryServices.AccountManagement 4.6.0*.

Na Slici Slika 13 prikazano je testno okruženje sa numerisanim tokom podataka. U gornjem desnom uglu prikazan je konfiguraciona datoteka koja kao prvi parametar sadrži bezbednosni sistem na koji se šalju podaci, a u ovom primeru testiranja je to *OpenLDAP*, drugi i treći parametar su *IP* adresa - *localhost* i port - 389, a četvrti parametar je domen - *primer.com*.

Testna aplikacija je razvijena u *.Net Core* – u i povezana je sa adapterom, koji ima konekcije ka *AD* i *OpenLDAP* bezbednosnim sistemima. Korišćeni bezbednosni sistemi instalirani su na *Windows Server 2019* i *Ubuntu 18.04 OS*, pokrenutim na virtuelnim mašinama, *Hyper – V Manager* alatom.

Tok podataka testne aplikacije je sledeći:

1. adapter čita podatke iz konfiguracione datoteke, koji popunjava korisnik,
2. testna aplikacija šalje testne primere se na adapter,
3. adapter prosleđuje zahtev do *OpenLDAP* – a,
4. *OpenLDAP* obrađuje zahtev i odgovor šalje do adaptera,
5. adapter prosleđuje odgovor do testne aplikacije koja vizuelno prikazuje odgovor,
6. u slučaju logovanja korisnika, nakon odgovora bezbednosnog servisa se zapisuju korisničko ime i vreme logovanje u datoteku za evidentiranje logovanja.



Slika 13 - Testna aplikacija povezana sa komponentama

## 4.2 Testiranje

Za verifikaciju rešenja razvijena je konzolna testna aplikacija u *.Net Core* – u, kako bi testna aplikacija bila funkcionalna na *Windows* i *Linux* platformama *OS* – a. Testna aplikacija povezana je sa interoperabilnim adapterom omogućuje slanje zahteva adapteru i prikaz odgovora adaptera za kreiranje novih, izlistavanje, pretragu, modifikaciju i brisanje entiteta.

Prilikom startovanja testne aplikacije, korisniku je data mogućnost logovanja kao administrator ili kao običan korisnik. Kada se loguje kao običan korisnik, korisnik aplikacije može da proveri svoje dozvole pristupa, a kada se loguje kao administrator, korisniku aplikacije se prikazuje meni sa ponuđenim administratorskim funkcionalnostima. Korisnički meni prikazan je na Slici 14. Administrator bira jednu od šest mogućnosti: upravljanje korisničkim ili podacima o grupama,



upravljanje članstvom i dozvolama pristupa, izlistavanje svih korisnika i grupa ili pretragu korisnika i grupa po korisničkom imenu ili imenu grupe.

```

-----
1. Manage users and groups
2. Manage membership and permissions
3. List all users
4. List all groups
5. Search users
6. Search groups
0. Exit
-----
Enter number:

```

Slika 14 - Korisnički meni testne aplikacije

Korisnički meni za upravljanje podacima o korisnicima i grupama prikazan je na Slici 15 i omogućuje pozivanje funkcija za kreiranje, izmenu i brisanje korisnika i grupa.

Meni za dodavanje članova grupa, proveravanje članstva, dodeljivanje i brisanje dozvola pristupa prikazan je na Slici 16.

```

-----
1. Manage users and groups
2. Manage membership and permissions
3. List all users
4. List all groups
5. Search users
6. Search groups
0. Exit
-----
Enter number: 1
1. Create user
2. Change user username
3. Change user name
4. Change user surname
5. Change user password
6. Change user phone number
7. Change user email address
8. Change user description
9. Delete user
-----
10. Create group
11. Change group name
12. Change group description
13. Delete group

```

Slika 15 - Funkcije za dodavanje, brisanje i izmenu entiteta

```

-----
1. Manage users and groups
2. Manage membership and permissions
3. List all users
4. List all groups
5. Search users
6. Search groups
0. Exit
-----
Enter number: 2
1. Add user to group
2. Add group to another group
3. Is member of
4. Get all group members
5. Assign permission
6. Remove permission
7. Create custom permission
8. Assign custom permission
9. Remove custom permission

```

Slika 16 - Funkcije za upravljanje odnosima između entiteta

Za testiranje i verifikaciju razvijenih funkcionalnosti informacionu bazu podataka treba popuniti entitetima. Kreirano je pedeset novih korisnika sa korisničkim imenima: *username0*, *username1*, *username2* i dr. Nakon kreiranja novih korisnika, kreirano je i pet novih grupa sa imenima: *group0*, *group1*, *group2*, *group3* i *group4*. Za verifikaciju kreiranje novih slogova pozvana je funkcija izlistavanja svih korisnika i grupa. Na Slici 17 prikazana je lista novo-kreiranih korisnika na AD – u, a na Slici 18 lista kreiranih korisnika na *OpenLDAP* – u.

```
-----
Enter number: 3
Administrator
Guest
krbtgt
username0
username1
username2
username3
username4
username5
username6
username7
username8
username9
username10
username11
username12
username13
username14
username15
username16
username17
username18
username19
username20
username21
username22
username23
username24
username25
username26
username27
username28
username29
username30
username31
username32
username33
username34
username35
username36
username37
username38
username39
username40
username41
username42
username43
username44
username45
username46
username47
username48
username49
-----
```

Slika 17 - Verifikacija rešenja - izlistani korisnicima AD - u

```
-----
Enter number: 3
Users:
cn=username0,ou=People,dc=example,dc=com
cn=username1,ou=People,dc=example,dc=com
cn=username2,ou=People,dc=example,dc=com
cn=username3,ou=People,dc=example,dc=com
cn=username4,ou=People,dc=example,dc=com
cn=username5,ou=People,dc=example,dc=com
cn=username6,ou=People,dc=example,dc=com
cn=username7,ou=People,dc=example,dc=com
cn=username8,ou=People,dc=example,dc=com
cn=username9,ou=People,dc=example,dc=com
cn=username10,ou=People,dc=example,dc=com
cn=username11,ou=People,dc=example,dc=com
cn=username12,ou=People,dc=example,dc=com
cn=username13,ou=People,dc=example,dc=com
cn=username14,ou=People,dc=example,dc=com
cn=username15,ou=People,dc=example,dc=com
cn=username16,ou=People,dc=example,dc=com
cn=username17,ou=People,dc=example,dc=com
cn=username18,ou=People,dc=example,dc=com
cn=username19,ou=People,dc=example,dc=com
cn=username20,ou=People,dc=example,dc=com
cn=username21,ou=People,dc=example,dc=com
cn=username22,ou=People,dc=example,dc=com
cn=username23,ou=People,dc=example,dc=com
cn=username24,ou=People,dc=example,dc=com
cn=username25,ou=People,dc=example,dc=com
cn=username26,ou=People,dc=example,dc=com
cn=username27,ou=People,dc=example,dc=com
cn=username28,ou=People,dc=example,dc=com
cn=username29,ou=People,dc=example,dc=com
cn=username30,ou=People,dc=example,dc=com
cn=username31,ou=People,dc=example,dc=com
cn=username32,ou=People,dc=example,dc=com
cn=username33,ou=People,dc=example,dc=com
cn=username34,ou=People,dc=example,dc=com
cn=username35,ou=People,dc=example,dc=com
cn=username36,ou=People,dc=example,dc=com
cn=username37,ou=People,dc=example,dc=com
cn=username38,ou=People,dc=example,dc=com
cn=username39,ou=People,dc=example,dc=com
cn=username40,ou=People,dc=example,dc=com
cn=username41,ou=People,dc=example,dc=com
cn=username42,ou=People,dc=example,dc=com
cn=username43,ou=People,dc=example,dc=com
cn=username44,ou=People,dc=example,dc=com
cn=username45,ou=People,dc=example,dc=com
cn=username46,ou=People,dc=example,dc=com
cn=username47,ou=People,dc=example,dc=com
cn=username48,ou=People,dc=example,dc=com
cn=username49,ou=People,dc=example,dc=com
-----
```

Slika 18 - Verifikacija rešenja - izlistani korisnici na *OpenLDAP* - u

Na Slici 19 i Slici 20 verifikovane su novo-kreirane grupe.

```
Protected Users
Key Admins
Enterprise Key Admins
DnsAdmins
DnsUpdateProxy
Multiplication permission
group0
group1
group2
group3
group4
```

Slika 19 - Verifikacija rešenja – izlistane grupe na AD - u

```
1. Manage users and groups
2. Manage membership and permissions
3. List all users
4. List all groups
5. Search users
6. Search groups
0. Exit

Enter number: 4
Groups:
cn=group0,ou=Groups,dc=example,dc=com
cn=group1,ou=Groups,dc=example,dc=com
cn=group2,ou=Groups,dc=example,dc=com
cn=group3,ou=Groups,dc=example,dc=com
```

Slika 20 - Verifikacija rešenja – izlistane grupe na OpenLDAP - u

Za testiranje članstva u grupama, svaki peti korisnik je učlanjen u istu grupu. Na slikama: 21 i 22 može se videti grupa pod imenom *group0*. Prva slika pokazuje rešenje na AD – u, a druga slika na *OpenLDAP* – u. Razlika u strukturama na jednom i drugom bezbednosnom sistemu može se videti u tipovima atributa koji predstavljaju članove grupa. *OpenLDAP* za svakog člana ima poseban atribut *member*, dok *AD* ima listu članova u okviru jednog atributa.

```
Enter number: 6
Enter number of groups you want to search: 1
Enter group name [0]:
group0

Group name: group0
Description:
Members:
username0
username5
username10
username15
username20
username25
username30
username35
username40
username45
```

Slika 21 - Testiranje i verifikacija članstva u grupama na AD - u

```
Enter number: 6
Enter number of groups you want to search: 1
Enter group name [0]:
group0

Groups:
cn=group0,ou=Groups,dc=example,dc=com
objectClass: groupOfNames
cn: group0
Member: cn=username0,ou=People,dc=example,dc=com
Member: cn=username5,ou=People,dc=example,dc=com
Member: cn=username10,ou=People,dc=example,dc=com
Member: cn=username15,ou=People,dc=example,dc=com
Member: cn=username20,ou=People,dc=example,dc=com
Member: cn=username25,ou=People,dc=example,dc=com
Member: cn=username30,ou=People,dc=example,dc=com
Member: cn=username35,ou=People,dc=example,dc=com
Member: cn=username40,ou=People,dc=example,dc=com
Member: cn=username45,ou=People,dc=example,dc=com
```

Slika 22- Testiranje i verifikacija članstva u grupama na OpenLDAP - u

Za verifikaciju izmene korisničkih podataka, korisniku su dodatno upisani atributi: elektronska pošta, broj telefon i opis. Uspešno izmenjeni atributi na oba bezbednosna sistema prikazani su na Slici 23 i Slici 24.

```
-----
Enter number: 5
Enter number of users you want to search: 1
Enter username [0]:
username0
-----
DN: CN=username0,CN=Users,DC=myNewForest,DC=test
Given name: name0
Surname: surname0
User principal name: username0
Description: Verification user
Telephone number: 123123123
Email address: username0@email.com
-----
```

Slika 23 - Izmena korisničkih atributa na AD - u

```
-----
Enter number: 5
Enter number of users you want to search: 1
Enter username [0]:
username0
-----
Users:
cn=username0,ou=People,dc=example,dc=com
sn: surname0
givenName: name0
userPassword: test12
mobile: 123123123
objectClass: inetOrgPerson
cn: username0
description: Validation user
mail: username0@email.com
-----
```

Slika 24 - Izmena korisničkih atributa na OpenLDAP - u

Izmena atributa grupe je, takođe, uspešno verifikovana izmenom opisa grupe i na *AD* i *OpenLDAP* bezbednosnim sistemima. Uspešno realizovana izmena na *AD* – u, vidi se na Slici 25, a izmena na *OpenLDAP* bezbednosnom servisu prikazana je na Slici 26.

```
-----
Enter number: 6
Enter number of groups you want to search: group0
You must enter a number!
1
Enter group name [0]:
group0
-----
Group name: group0
Description: Verification group
Members:
username0
username5
username10
username15
username20
username25
username30
username35
username40
username45
-----
```

Slika 25 - Izmjena atributa grupe na AD - u

```
-----
Enter number: 6
Enter number of groups you want to search: 1
Enter group name [0]:
group0
-----
Groups:
cn=group0,ou=Groups,dc=example,dc=com
cn: group0
objectClass: groupOfNames
description: Verification group
Member: cn=username0,ou=People,dc=example,dc=com
Member: cn=username5,ou=People,dc=example,dc=com
Member: cn=username10,ou=People,dc=example,dc=com
Member: cn=username15,ou=People,dc=example,dc=com
Member: cn=username20,ou=People,dc=example,dc=com
Member: cn=username25,ou=People,dc=example,dc=com
Member: cn=username30,ou=People,dc=example,dc=com
Member: cn=username35,ou=People,dc=example,dc=com
Member: cn=username40,ou=People,dc=example,dc=com
Member: cn=username45,ou=People,dc=example,dc=com
-----
```

Slika 26 -Izmjena atributa grupe na OpenLDAP - u

Zapisi o logovanju korisnika se zapisuju u datoteci za evidentiranje logovanja korisnika. Evidencija logovanja je prikazana na Slici 27, a evidentira se korisnik i vreme logovanja.

User: username0	Last login: 11/1/2019 3:50:03 PM
User: username25	Last login: 11/3/2019 2:31:46 PM
User: username33	Last login: 11/5/2019 7:01:52 PM
User: username10	Last login: 11/5/2019 8:15:03 PM
User: username12	Last login: 11/9/2019 3:33:41 PM
User: username19	Last login: 11/11/2019 1:51:57 PM
User: username0	Last login: 11/11/2019 3:27:03 PM
User: username22	Last login: 11/11/2019 3:30:46 PM
User: username33	Last login: 11/11/2019 8:14:52 PM
User: username39	Last login: 11/12/2019 6:13:03 PM
User: username40	Last login: 11/12/2019 8:37:13 PM
User: username41	Last login: 11/12/2019 8:50:52 PM

Slika 27 - Evidentiranje logovanja korisnika

Verifikovano je rešenje i za dodeljivanja ugrađenih dozvola pristupa. Rešenje je testirano samo na AD bezbednosnim sistemom, zato što funkcija dodavanja dozvole pristupa nije implementirana na *OpenLDAP* – u zbog ograničenja osnovnih šema. Na Slici 28 prikazana je grupa sa imenom: *group0* i korisnici sa korisničkim imenima: *username0* i *username1* kojima je dodeljena dozvola za čitanje sadržaja grupe.

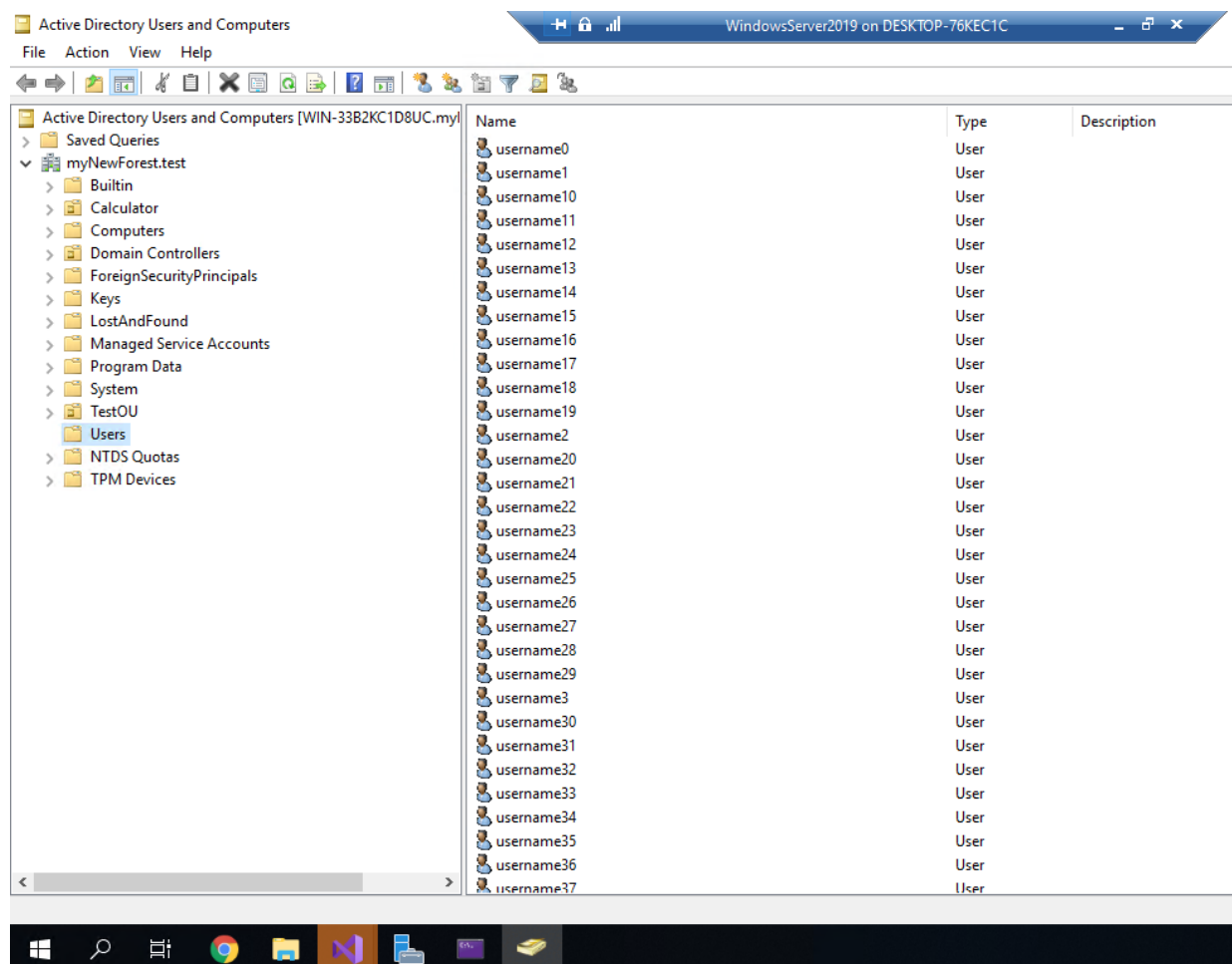
```
-----
Enter number: 6
Enter number of groups you want to search: 1
Enter group name [0]:
group0
-----
Group name: group0
Description: Verification group
Members:
username0
username5
username10
username15
username20
username25
username30
username35
username40
username45
-----
Group access rights:
NT AUTHORITY\SELF: GenericRead
NT AUTHORITY\Authenticated Users: GenericRead
NT AUTHORITY\SYSTEM: GenericAll
BUILTIN\Account Operators: GenericAll
MYNEWFOREST\Domain Admins: GenericAll
MYNEWFOREST\username0: GenericRead
MYNEWFOREST\username1: GenericRead
NT AUTHORITY\Authenticated Users: ExtendedRight
```

Slika 28 - Dodeljivanje dozvole pristupa

Kako bi se pokazala konzistentnost i verifikovao rad testne aplikacije, podaci u bazama informacione-bezbednosti ručno su provereni na AD – u i *OpenLDAP* – u. AD bazi podataka se pristupa ugrađenom aplikacijom *Users and Computers*.

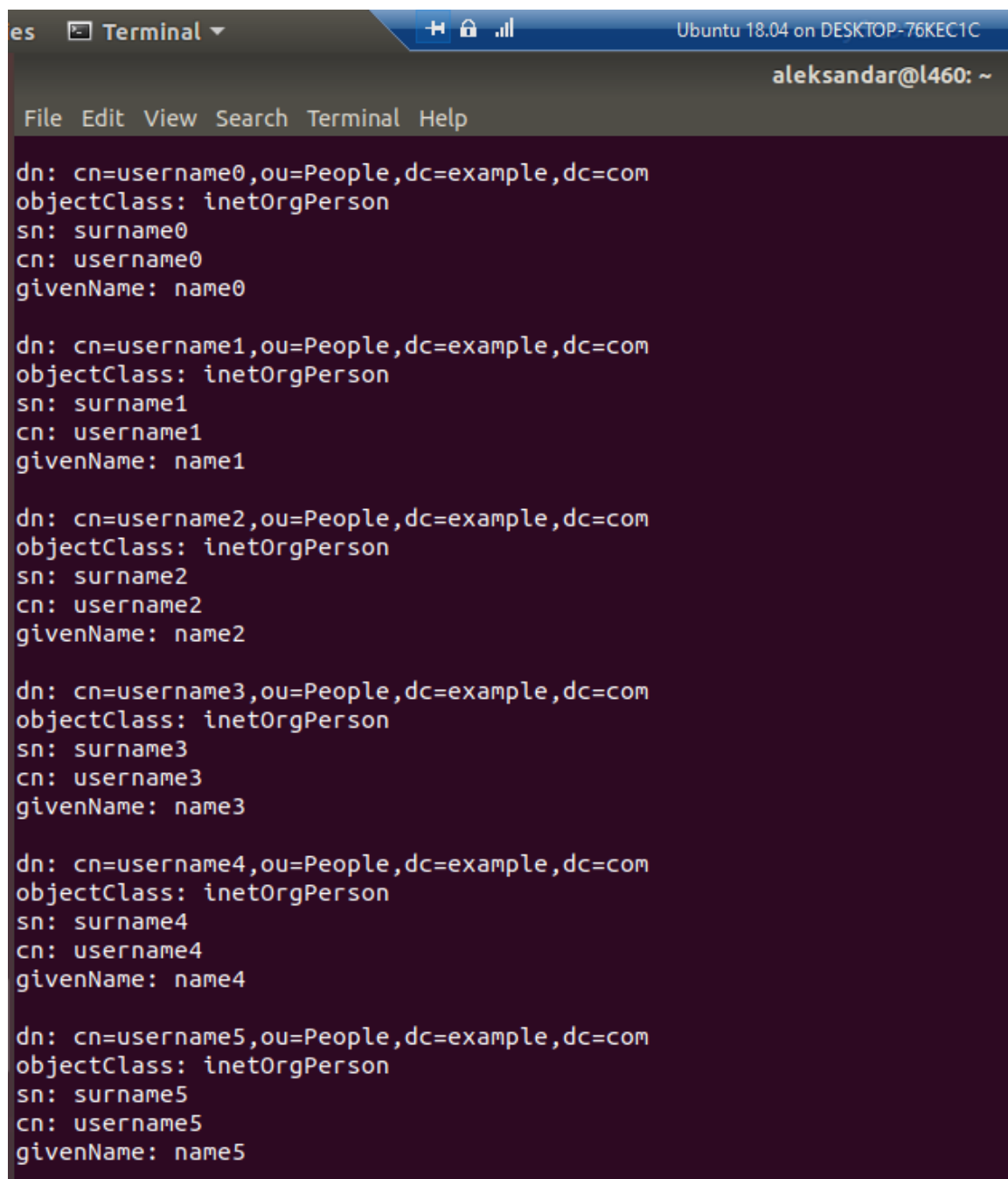


Na Slici 29 prikazano je stanje informacione baze podataka, gde vide korisnici sa korisničkim imenim: *username0*, *username1* i dr., ispunjavaju sadržaj te baze podataka, a kreirani su pozivanjem funkcija testne aplikacije. Kreirani korisnici pripadaju organizacionoj jedinici *Users*, koja predstavlja inicijalnu organizacionu jedinicu u kojoj se podrazumevano čuvaju novi korisnici i grupe na AD – u, ako se eksplicitno ne navede nova organizaciona jedinica.



Slika 29 - Izlistani korisnici na AD - u, verifikacija testnog slučaja

Prilikom izrade i verifikacije rešenja ovog master rada, pristup bazi informaciono-bezbednosnih podataka na *OpenLDAP* – a vršio se pozivanjem komande *ldapsearch* preko terminala Slici 30 mogu se videti izlistani korisnici na *OpenLDAP* – u.



```
es  Terminal ▾  Ubuntu 18.04 on DESKTOP-76KEC1C  aleksandar@l460: ~  
File Edit View Search Terminal Help  
dn: cn=username0,ou=People,dc=example,dc=com  
objectClass: inetOrgPerson  
sn: surname0  
cn: username0  
givenName: name0  
  
dn: cn=username1,ou=People,dc=example,dc=com  
objectClass: inetOrgPerson  
sn: surname1  
cn: username1  
givenName: name1  
  
dn: cn=username2,ou=People,dc=example,dc=com  
objectClass: inetOrgPerson  
sn: surname2  
cn: username2  
givenName: name2  
  
dn: cn=username3,ou=People,dc=example,dc=com  
objectClass: inetOrgPerson  
sn: surname3  
cn: username3  
givenName: name3  
  
dn: cn=username4,ou=People,dc=example,dc=com  
objectClass: inetOrgPerson  
sn: surname4  
cn: username4  
givenName: name4  
  
dn: cn=username5,ou=People,dc=example,dc=com  
objectClass: inetOrgPerson  
sn: surname5  
cn: username5  
givenName: name5
```

Slika 30 - Izlistani korisnici na *OpenLDAP* - u, verifikacija testnog slučaja



## 5 Zaključak

Organizacija informacione-bezbednosti u današnje vreme predstavlja neizbežan izazov za sve kompanije. Održavanje poverljivosti, integriteta i dostupnosti podataka od ključne je važnosti za poverenje svih korisnika usluga kompanija, ali i za zaštitu ličnih podataka zaposlenih.

Informaciono-bezbednosni sistem omogućava rad sa podacima vezanim za resurse kompanija, definiše hijerarhiju među entitetima, ograničava prava i dozvole pristupa subjektima nad objektima i time olakšava organizaciju, čitanje i upravljanje informaciono-bezbednosnim podacima. Dobro organizovani bezbednosni sistem daje nedvosmislena prava korisnicima u okviru kompanije.

Interoperabilnost dva sistema dobija na značaju razvijanjem mrežnih tehnologija. Povezanost različitih sistema omogućava upotrebu podataka koje je prikupio ili obradio drugi sistem, čime se ograničava uloga sistema.

Ograničavanje uloga sistema samo na funkcije za koje je zadužen, predstavlja bitan ekonomski faktor, smanjujući nepotrebna ulaganja u proširenja i razvijanje sistema.

U ovom radu istraženi su bezbednosni-informacioni sistemi za upravljanje informacionim podacima o korisnicima, grupama i njihovim vezama. Izabrana su dva različita informaciono-bezbednosna sistema, koji implementiraju protokol *LDAP*:

- a) *AD* koji je zasnovan na *Windows* platformi,
- b) *OpenLDAP* koji je zasnovan na *Linux* platformi.

*AD* pruža intuitivan i jednostavan korisnički interfejs, olakšavajući organizaciju resursa. *OpenLDAP* je besplatan, ali zahtevniji za instalaciju i organizaciju podataka od *AD*-a. Baza informacionih podataka na *OpenLDAP*-u je prazna nakon instalacije i ostavlja korisniku mogućnost da organizuje svoje stablo od samog početka, dok *AD* sa instalacijom, između ostalog, formira osnovnu strukturu sa domenom i organizacionim jedinicama korisnika i računara. Istraživanjem bezbednosnih sistema zaključuje se da je *AD* uže orijentisan i jednostavniji za korišćenje.

Istraženi su interfejsi za pristupanje informacionim bazama podataka na oba sistema, dizajniran je i implementiran interoperabilni adapter koji je kompatibilan sa obe *LDAP* implementacije bezbednosnih sistema. Interoperabilni adapter zasnovan je na generičkom sloju, funkcionalnom sa oba *OS* – a. Omogućen je izbor platforme za rad sa informaciono-bezbednosnim sistemima, ali i jednostavna promena platforme i zamena bezbednosnog sistema. Razvijeni adapter omogućava korisniku da bez izmena koda i bez složenih promena konfiguracije zameni bazu informaciono-bezbednosnog sistema sa kojim komunicira, a da funkcionalnosti ostanu nepromenjene.

Rešenje je implementirano na procesnoj virtuelnoj mašini *.NET Core*, koja je i sama interoperabilna i funkcionalna na obe *OS* platforme, čime je tehnički omogućena upotreba interoperabilnog adaptera na *Windows*-u i *Linux*-u.

Testiranjem rešenja, uspešno je verifikovan rad interoperabilnog adaptera. Testirane su funkcionalnosti za kreiranje novih korisnika, grupa, dozvola pristupa i dr. Sve funkcionalnosti testirane su konzolnom aplikacijom, da bi se uspešno verifikovala rešenja na obe OS platforme.

Predloženo rešenje adaptera se u budućnosti može proširiti i unaprediti na sledeće načine:

- a) Omogućavanjem pristupa još jednom informaciono-bezbednosnom sistemu. Za realizaciju bi bio iskorišćen fabrički dizajn patern za kreiranje novog menadžera funkcija, za novi bezbednosni sistem. Dodatni sistem bi proširio opseg mogućih korisnika i usluga.
- b) Proširenjem postojeće šeme na *AD*-u i *OpenLDAP*-u, čime bi se novi tipovi entiteta kreirali u toku rada adaptera. Da bi se ovakvo proširenje ostvarilo, potrebno je da se istraži kakve dodatne šeme postoje u *AD*, a kakve na *OpenLDAP* sistemu, kakvi subjekti i objekti postoje i kakve attribute sadrže. Implementacija ovakvog proširenja bi zahtevala dodatan nivo apstrakcije, koji bi nezavisno od toga koji novi tip entiteta se dodaje u bazu, mogao da obezbedi sve funkcionalnosti.
- c) Implementiranjem interoperabilnog adaptera na drugim programskim jezicima, čime bi se obezbedila lakša integracija sa bezbednosnim sistemima koji se ne oslanjaju na *C#* biblioteke za pristup bazama podataka.

## 6 Literatura

- [1] M. E. Whitman and H. J. Mattord, *Principles of Information Security Fourth Edition*, no. January 2015., Course Technology, 20 Channel Center Boston, MA 02210 USA, 2011.
- [2] S. Tuttle *et al.*, *Understanding LDAP Design and Implementation*, IBM RedBooks, 2006.
- [3] E. Harold F. Tripton, *Official (ISC) Guide To The SSCP CBK*. 1385., CRC Press-Taylor and Francis Group, 2011.
- [4] B. Desmond, J. Richards, R. Allen, and A. Lowe-Norris, *Active Directory, 5th Edition*, vol. 91, no. 5. 2012, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, April 2013.
- [5] B. Svidergol and A. Robbie, *Active Directory Cookbook*, O'Reilly Media, 2013.
- [6] Microsoft, *Trusts* 2005. Accessed on: Oct. 20, 2019. [Online]. Available: <https://msdn.microsoft.com/en-us/windows/desktop/cc775736>.
- [7] M. Butcher, *Mastering OpenLDAP - Configuring, Securing and Integrating Directory Services*, Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK, 2007.
- [8] Techopedia, *Interoperability*, Accessed on: Oct. 22, 2019. [Online]. Available: <https://www.techopedia.com/definition/631/interoperability>.
- [9] Microsoft, *Manage Code*, Accessed on: Dec. 2, 2019. [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/standard/managed-code>.
- [10] Chapman & Hall, *The Art of Automatic Memory Management*, CRC Press-Taylor and Francis Group, 2012.
- [11] Microsoft, *Garbage Collector and Mark Sweep*, 2009. Accessed on: Nov. 15, 2019 [Online]. Available: <https://blogs.msdn.microsoft.com/abhinaba/2009/01/30/back-to-basics-mark-and-sweep-garbage-collection/>.
- [12] Microsoft, *.Net Core*, 2019. Accessed on: Oct. 31, 2019 [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/core/about>.
- [13] Microsoft, *DirectoryEntry Class*, [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.directoryentry?view=netframework-4.8>.
- [14] Microsoft, *System.DirectoryServices Namespace*, Accessed on: Oct. 13, 2019. [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices?view=netframework-4.8>.
- [15] Microsoft, *System.DirectoryServices.AccountManagement Namespace*, Accessed on: Oct.13,2019,. [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.accountmanagement?view=netframework-4.8>.
- [16] Microsoft, *User Naming Attributes*, 2018. Accessed on: Oct. 13, 2019. [Online]. Available: <https://docs.microsoft.com/en-us/windows/win32/ad/naming-properties#samaccountname>.

- [17] Novell, *Novell Documentation*. Accessed on: Oct. 22, 2019. [Online]. Available: <https://www.novell.com/documentation/developer/ldapcsharp/?page=/documentation/developer/ldapcsharp/cnet/data/b3u1s5w.html>.

## 7 Biografija



Kandidat Aleksandar Maričić je rođen 30.5.1995. godine u Zrenjaninu. Završio je osnovnu školu „Vuk Karadžić“ kao nosilac diplome „Vuk Karadžić“ i proglašen je za člaka generacije. Završio je Zrenjaninsku gimnaziju, opšti smer, 2014. godine kao nosilac diplome „Vuk Karadžić“. Fakultet tehničkih nauka u Novom Sadu, smer Primijenjeno softversko inženjerstvo, upisao je 2014. godine, a osnovne akademske studije završio je 2018. godine, odbranivši diplomski rad pod nazivom „Replikaciona infrastruktura za heterogene baze podataka“. Potom je iste godine upisao master akademske studije Primijenjenog softverskog inženjerstva. Nakon godinu dana ispunio je sve obaveze i položio sve ispite predviđene studijskim programom. Stipendista je firme „*Schneider Electric DMS NS*“ postao je 2016. godine.